IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 18, 2021, accepted January 26, 2021, date of publication February 3, 2021, date of current version February 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056747

Reliability Assessment Model of IMA Partition Software Using Stochastic Petri Nets

WU ZHIJUN, MA HAOLIN, AND YUE MENG
School of Electronics and Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Wu Zhijun (zjwu@cauc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China and Civil Aviation Administration of China under Grant U1933108, in part by the Scientific Research Project of Tianjin Municipal Education Commission under Grant 2019KJ117, and in part by the Fundamental Research Funds for the Central Universities of China under Grant 3122020076 and Grant 3122019051.

ABSTRACT In order to reduce the failure rate of Integrated Modular Avionics (IMA) partition software, due to the reliability block diagram (RBD) method, fault tree analysis (FTA) method and GO method cannot describe the state transition process of partition software, according to the ARINC 653 standard and the actual running status of the partition software, this paper determines the state machine and conversion delay of the partition software, and establishes the stochastic Petri nets (SPN) reliability quantitative model of the partition software. By proving that each transition in the SPN model of the partition software approximately obeying exponential distribution, and according to the reachable state tree of the SPN isomorphic to a homogeneous Markov chain (MC), the steady-state probability of the partition software in the fault state is calculated to be $5.2778*10^{-9}$ by using MC stochastic process theory. The factors affecting the reliability of the partition software are obtained, and the sensitivity of each factor to the model is studied. Finally, the relevant conclusions are drawn to provide guidance for improving the reliability of partition software.

INDEX TERMS Integrated modular avionics, partition software, stochastic petri nets, reliability, fault tree analysis.

I. INTRODUCTION

In order to further lighten the weight of the aircraft and reduce the operation cost of the entire life cycle of the aircraft, the guidance document RTCA DO-297 from the Radio Technical Commission for Aeronautics (RTCA) resides the originally independent physical equipment in the IMA system in the form of software [1]. The emergence of IMA system solves the shortcomings of replacing hardware equipment and changing the original architecture due to the change of aircraft functional requirements, reduces the coupling between hardware and embedded software, and simplifies the development and verification process of avionics system software. At the same time, IMA system adopts the partition technology to provide an isolation mechanism to prevent the unexpected interference between the partition software. Therefore, IMA system is widely used in the system development and design of the new generation civil aircraft such as A380, B787 and C919 [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Dipankar Deb.

The reliability of the hardware equipment is much higher than that of the software. The software of airborne hardware brings many advantages, but it will make the reliability and safety of the aircraft decrease rapidly. Currently, the way to improve the reliability of IMA system is generally from the perspective of hardware and improve the reliability of the IMA system by improving the reliability of key hardware components. It is seldom considered that the main reason for the reliability reduction of IMA system is that the reliability of software is much lower than that of hardware. Wang *et al.* establish an SPN model of a single partition software based on the running state of the IMA partition software, but this model merges the startup state of the partition software defined by the ARINC 653 standard, which does not conform to the actual running state of the partition software [3]. The reliability of partition software is improved by increasing the Mean Time between Failures (MTBF) of the partition software. However, it does not notice that the increase of the MTBF of the partition software will have less and less effect on the reliability of the partition software, nor does it give the quantitative evaluation of partition software by various factors.

Therefore, this paper proposes a reliability modeling and quantitative analysis method based on SPN which is more in line with the running state of partition software by decomposing the interaction process and failure mode of partition software and according to ARINC 653 standard. On the basis of studying the average delay of partition software in different states and the reliability index of avionics, the steady-state probability of partition software in failure state is obtained. Finally, the factors that affect the reliability of the partition software are quantitatively evaluated, and the quantitative calculation analysis of the model is carried out to improve the reliability of the partition software.

The innovative work of this paper includes the following contents:

i. According to the actual operating status of the partition software, this paper establishes the SPN reliability evaluation model of the partition software of the IMA system based on the partition software state machine defined by the ARINC 653 standard, and determines the trigger delay of each transition.

ii. In order to be able to quantitatively evaluate the reliability of the partition software, we proved that each transition in the partition software SPN model approximately obeys an exponential distribution. According to the reachable state tree of SPN, it is isomorphic to a homogeneous Markov chain (MC), we us MC random process theory to calculate the steady-state probability of the partition software in a fault state.

iii. We conclude that there are 4 factors that affect the reliability of the partition software: (a) the MTBF of the partition software, (b) the average time of the partition software COLD_START, (c) the average time of the partition software HOT_START, (d) the average time of HM/FM to find a fault. The sensitivity of parameter changes is obtained to provide guidance for improving the reliability of partition software.

The rest of this paper is arranged as follows. Section II introduces related works. Section III establishes the fault tree model and dynamic SPN model of the partition software by analyzing the dynamic interaction process of the partition software of the IMA system. Section IV proves that the SPN model of the partition software is isomorphic to the MC, and gets the steady-state probability of the partition software in a failure state. Section V shows the experiments and results analysis. Section VI summaries this paper and discusses the future work.

II. RELATED WORKS

There have been many mature researches on the reliability and security of complex systems.

Reference [4] focuses on the effects of uncertainty on systems, a data-driven framework based on point estimate method and support vector machine is developed. In this way, the standard deviation of the uncertain parameters can be extracted, and its influence on the system operation problems can be reflected through the limited concentration points. Reference [5] proposes a distributed sustainable integration automation testing platform to improve the quality of software. Reference [6] proposes Security Constrained Unit Commitment (SCUC) incorporating Dynamic Thermal Line Rating (DTLR) of overhead transmission lines to improve the security of power system. In reference [7], a fault diagnosis method of the generator system is proposed to improve the reliability of the generator system. References [8] and [9] propose a method for data rate and specific signal format based peripheral security system. References [10] and [11] provide a detailed summary of the application of Bayesian networks in the field of reliability evaluation. Reference [12] contributes a remaining useful life RUL re-prediction method based on Wiener process combining the current monitoring status and historical degradation data of the system. This method can further improve the accuracy of the reliability model.

In recent years, IMA system is the main development direction of avionics system [13], scholars have mainly analyzed and improved the reliability of IMA system from the following three angles. The first category is from the perspective of the IMA system as a whole, the second category is from the perspective of the dynamic reconfiguration of the IMA system, and the last category is from the perspective of the IMA partition software to study the methods to improve the reliability of the IMA system.

Much work has been done on the reliability of the IMA system. In reference [14], the radar processing task of IMA system has been decomposed into a combination of software task and hardware sensor. By defining the adjacency matrix and the reachability matrix, and using the safety critical node decision method, the safety critical nodes that have the greatest impact on the task are obtained, and the reliability of the IMA system has been improved by improving the reliability of the critical nodes. The method is also applicable to other systems. In reference [15], the architecture analysis and design language (AADL) is used to model the interconnected architecture and fault information of IMA system, and the conversion rules of the qualitative model of AADL to the quantitative model of generalized stochastic Petri nets (GSPN) are studied, which provides a reference for the architecture design of IMA system, and it also provides a new method for reliability evaluation. In reference [16], the authors summarize IMA guidance documents issued by relevant industry and certification authorities, and provide 7 suggestions on development and certification of IMA, providing reference for relevant developers to improve the reliability of IMA system. In reference [17], the interaction mode of IMA system and the potential failure location of IMA system are analyzed.

In view of the reliability of the dynamic reconfiguration process of IMA system, the authors in reference [18] construct a reliability mathematical model more in line with the distributed integrated modular avionics (DIMA) dynamic reconfiguration characteristics by using joint k/n(G) mode, and obtain the relationship between the reliability and time of the system reconfiguration process. It provides guidance for improving the reliability and optimizing resource allocation of dynamic reconstruction process of DIMA system. In reference [19], AADL language is used to establish the component error state, system reconfiguration architecture and system reconfiguration behavior model of IMA system, and the model transformation mapping rules are used to transform the model into a computable model. Through parameter sensitivity analysis, suggestions are made to improve the reliability of the dynamic reconfiguration process of IMA system.

Regarding the reliability of IMA system partition software, the authors in references [3] and [20] construct the reliability mathematical model of partition software using SPN according to ARINC 653 standard and the operation process of partition software. It is verified that the steady-state probability of partition software in normal state has nothing to do with partition period, only with the initialization time of partition software and MTBF time of general processing module (GPM). However, this modeling method combines the CLOD_START state and WARM_START state of the partition software given by ARINC 653 standard, which is inconsistent with the actual running state of partition software. Industry standard proposes a hierarchical health monitoring/fault management (HM/FM) monitoring scheme to monitor the health status of partition software. The authors in reference [21] propose to use three HM/FM query methods: no subordinate layer query (NSQ), subordinate layer query with subordinate layer FM activation (SQSF) and subordinate layer query with current layer FM activation (SQCF). SPN and time scale decomposition (TSD) technology are used to model and analyze the blocking time and reliability of the three methods. Wang *et al.* summarize the environmental factors affecting the running of partition soft-ware, combined with the characteristics of the environmental factors affecting partition software, and establish the actual running model of partition software with consideration of external environmental factors by using stochastic differential equation [22]. The sensitivity of MTBF of partition software to different environmental factors is analyzed, but the method to improve the reliability of partition software is not given. Reference [23] describes the reliability of application software for integrated modular avionics systems using Markov chain state transition diagrams from the user's point of view, and provides a reference for the number of applications that reside within each IMA module.

In the field of reliability analysis, there are many literatures that use Petri nets to model and analyze the reliability of systems and software. Xie *et al.* establish the wireless communication model of high-speed train with SPN, which provides a new method for wireless communication reliability evaluation [24]. Li *et al.* establish the reliability evaluation model of cloud data center service with Petri nets, which make up the gap between the existing reliability evaluation model and the reality [25]. Wei *et al.* use Petri nets to establish a reliability evaluation method for aviation flight control systems, and take the advantage of the quantitative evaluation model of Petri nets to build a flight control system that meets the quantitative requirements of the system [26].

From the perspective of IMA partition software, it is found that few researchers have proposed suggestions and methods for partition software reliability index allocation and quantitative analysis. In this paper, through the partition software state machine defined by ARINC 653, combined with the unique running mode of the IMA system, the SPN is used to establish a quantitative assessment model of the reliability of the partition software, and to provide guidance and suggestions for the reliability index allocation and quantitative analysis of the partition software.

III. MODELING OF THE IMA SYSTEM PARTITION SOFTWARE STATE MACHINE

The development process of avionics system starts from independent equipment functions, and has experienced some avionics systems through the ARINC429 data bus for simple data interaction. At present, various avionics systems conduct complex and massive data interactions through the AFDX data bus. Among them, the most widely used and most mature is the IMA system.

Traditional avionics systems complete various aviation tasks by point-to-point connections between line replaceable units (LRU). In order to further reduce the weight of the aircraft and increase the data throughput of the avionics system, the LRU resides in different IMA platforms in software according to different security levels [14]. Aviation tasks are implemented by hosted application software, public IMA platforms, and high-speed networks. The IMA platform is composed of several IMA modules, and each IMA module is generally composed of a general processing module (GPM) and a core operating system (OS). GPM provides shared computing resources to applications, and core OS provides partitioned environments and basic services to applications. The partition software consists of hosted application software and basic partition operating system software. Each hosted application software can complete a basic function, and the basic partition operating system software provides support for the operation of the hosted application software. Each IMA module can be embedded with different numbers of partition software of the same security level. Each partition software obtains corresponding data through sensors and work systems. Different partition software transfers data through the aviation data network to complete all avionics system tasks. The description of the IMA system and partition software is shown in Figure 1 [15].

The core OS in each module of IMA system performs uniform scheduling and processing of partition software, which will inevitably lead to some new risks [29]. One is the risk in time, since the IMA system is an embedded real-time operating system, in order to ensure safety, the IMA system uses the time isolation mechanism. According to the number of partition software run by GPM and its running characteristics, the worst-case execution time (WCET) satisfying each partition software is determined, and the static scheduling table

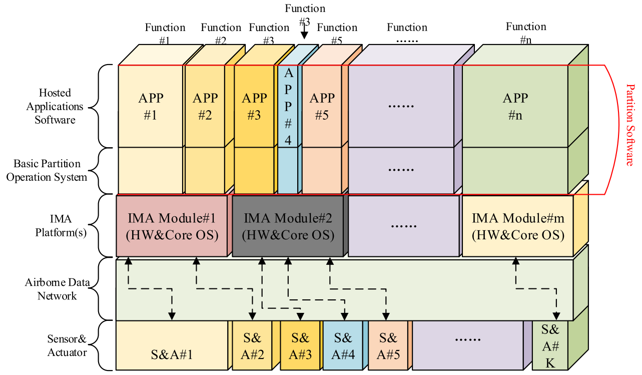


FIGURE 1. IMA system and partition software.

of GPM module is generated. The core OS schedules each partition according to the static scheduling table. In the initial stage of the IMA system design, according to the number of partitions running on the GPM and the operating cycle and duration of the partitions, the main time frame that can meet the cycle of each partition is determined, and the "idle time for the core OS to switch partitions" is reserved. As shown in Figure 2 is the static scheduling mechanism of the IMA system. This requires that the functions implemented by the avionics system must give processing results within a specified time. When a partition software maliciously occupies the processor's time, other partition software will be invalided. The other is the risk in space, since each partition software shares the same hardware, the storage address space of the processor is usually divided into different pages/segments. The memory management unit (MMU) unit configures the properties of these address space pages/segments, and configures the corresponding address space for each partition software. During the running of a certain partition software, when the core operating system incorrectly loads the running context of the partition, it will cause the partition software to erroneously read and write to the storage space of another partition software, resulting in other partition software failures. The IMA system MMU of the processor on the GPM according to the resource allocation information of the partition software, and the MMU checks whether the address space accessed by the partition software when it runs conforms to the access range of the partition software, thereby discover and prevent the address of the partition software from crossing the boundary.

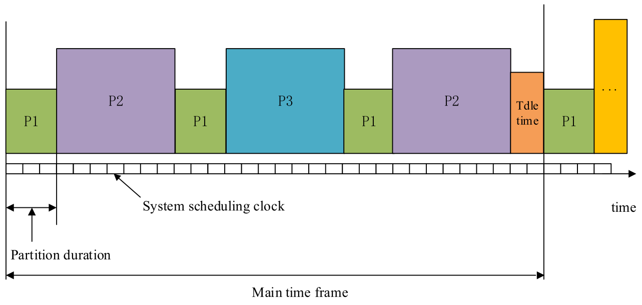


FIGURE 2. The static scheduling mechanism of IMA system.

The biggest difference between the IMA system and other systems is that the HM/FM mechanism is embedded inside. It is an important means for the IMA system to discover and deal with various failure of the partition software, realize the function recovery of the IMA system, and ensure the normal running of the avionics system [30]. The reliability of the IMA system hardware itself is already quite high. To further improve the reliability of the entire IMA system, it is necessary to analyze and identify the factors that affect the reliability of the partition software from the perspective of the partition software itself.

IMA partition software refers to the software running in a partition of the IMA system, including partition-resident applications and partitioned OS [31]. The common features of failure mode of IMA system partition software and that of general software are illegal requests, arithmetic errors and stack overflows. The difference is that there is interaction between the partition software and between the partition software and the IMA system. The interaction mainly includes resource allocation, two-level scheduling, partition communication, HM/FM, and system reconfiguration. During these interactions, it is inevitable that the unique faults of the partition software will occur. In order to build reliability model of partition software with SPN, it is necessary to find out the partition software state machine and analyze the transformation relationship between them.

There are many methods to model system reliability. As shown in Table 1, we summarize the advantages and disadvantages of the currently used system reliability modeling methods. Since the running state of the partition software has obvious timing characteristics, and in order to verify the accuracy of the established model through the software, the modeling method of Petri nets is used in this paper.

The partition software state machine defined in ARINC 653 standard has four states:
COLD_START, WARM_START, IDLE and NORMAL, as shown in Figure 3 [32]–[34].

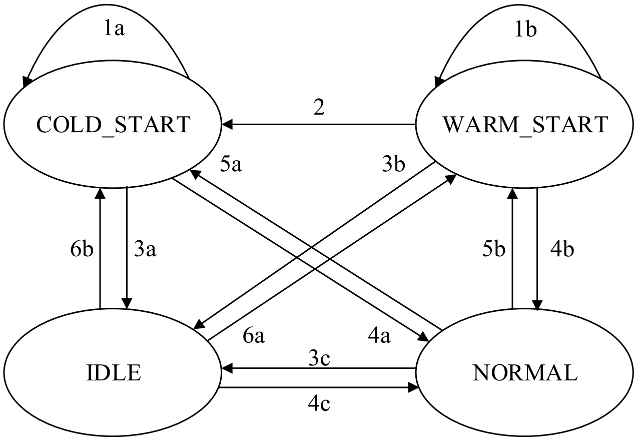


FIGURE 3. Partition state machine defined by ARINC 653.

In Figure 3, COLD_START, WARM_START and IDLE are the initial state of partition software. The main difference

TABLE 1. The advantages and disadvantages of the reliability modeling methods.

Method	Advantage	Disadvantage
RBD method	Simple, intuitive and logical	It is difficult to describe systems that are temporal, environmental, and human influenced
FTA method	It can conduct in-depth qualitative and quantitative analysis on the complex logical relationship between various events	It is difficult to describe sequential systems and fuzzy events
Petri nets method	It is convenient to model discrete time series dynamic systems, and has mature modeling software	When the system scale is complex, the exponential explosion problem exists in the state space
GO method	It is convenient for reliability modeling of ordered task and state variable system	The symbol is complex, and when the system scale is complex, the state space has exponential explosion problem

between COLD_START and WARM_START is that COLD_START can reload the code of all partitions and reinitialize the global data variables (GPM module system clock used by the partition, MMU page, etc.) that are not initialized for WARM_START. The IDLE is a state where the partition software has been initialized successfully but the application program has not yet been executed. NORMAL state indicates that the partition software is running normally. The description of each state transition of ARINC 653 partition software is shown in Table 2.

In order to establish the dynamic interaction model of the partition software and provide guidance for the development of the partition software for relevant developers, this paper obtains the state machine of partition software by combining the dynamic interaction process of partition software based on ARINC 653 standard.

The dynamic interaction process of partition software is mainly divided into the following steps:

i. INITIAL. Initialize IMA system hardware, initialize HM/FM mechanism, load the core OS code, initialize the core OS.

TABLE 2. Explanation of each state transition of ARINC 653 Partition software.

Transition	Description
1a	Partition software fails during COLD_START
1b	Partition software fails during WARM_START
2	The partition software hot start cannot recover the failure
3a	Initialize partition storage space, initialize permanent data variables and start threads
3b	Initialize the local variables of the partition software, start the scheduling function of the partition OS
3c	The scheduling partition goes into WAITING state
4a	Initialize partition storage space, initialize permanent data variables and start threads
4b	Initialize the local variables of the partition software, start the scheduling function of the partition OS
4c	The scheduling partition goes into NORMAL state
5a	A temporary failure of GPM or Core OS of an IMA system causes the partition software to fail
5b	Partition software failure
6a	Partition software failure
6b	A temporary failure of GPM or Core OS of an IMA system causes the partition software to fail

ii. COLD_START. Load the startup code of partition software, initialize global variables such as GPM system clock, MMU page and storage space pre-allocated by each partition software, start threads, start the scheduling function of partition OS, and complete the initialization of partition software.

iii. WARM_START. Initialize the local variables of the partition software, start the scheduling function of the partition OS, and jump to the entrance of the partition application.

iv. NORMAL. The processor resources of the core OS of the partition software perform arithmetic operation, and communication is conducted between threads in the partition, between partition software in the same IMA module, and between partitions of different IMA modules. When the time occupied by the partition software is used up, the context of the partition software is saved.

v. WAITING. The partition software is in a suspended state, waiting for the next invocation of the processor. When the processor invokes the partition software again, the recovery of the partition software context will be carried out.

Through the above dynamic interaction process of IMA partition software, INITIAL, COLD_START and WARM_START mainly complete the initialization of various system resources, which may cause partition software failure due to system resource allocation failure. When the partition software is in a normal state, in addition to regular errors such as arithmetic errors, stack overflows, and illegal requests, there must be data interaction processes between threads in the partition, between partition software of the same module, and between partition software of different modules, and

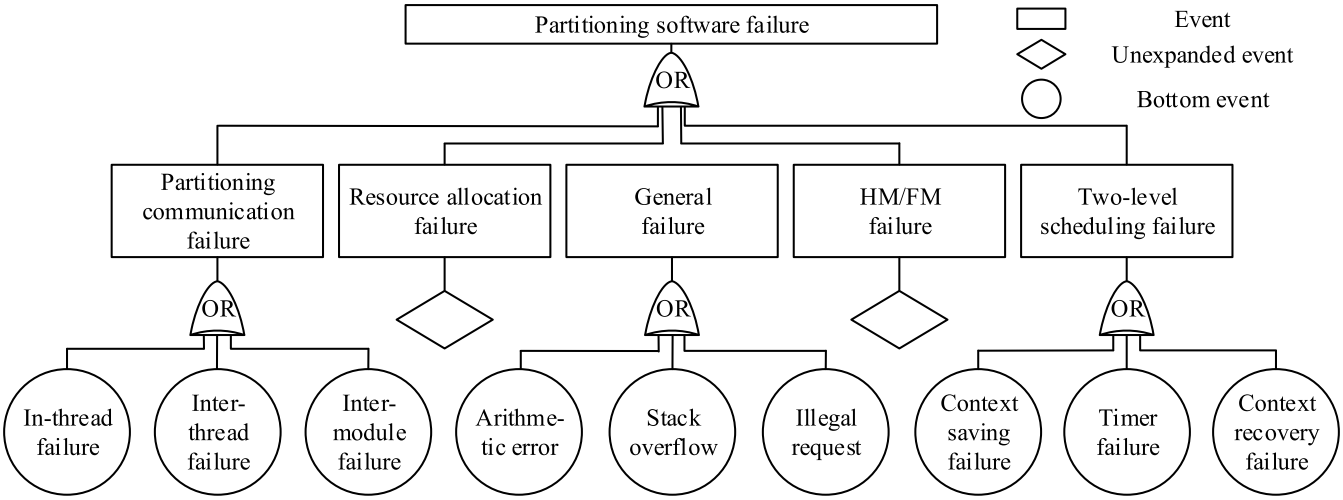


FIGURE 4. **Partition software failure mode fault tree.**

these processes may also cause partition software to failure. When the processor switches between partitions, it may cause partition context saving failure, partition timer failure, and partition context recovery failure. The failure mode fault tree of partition software can be obtained, which can provide a way to improve the reliability of partition software, as shown in Figure 4 [35]. Since this paper mainly analyzes the interaction mode and fault mode of partition software, it is assumed that the INITIAL, COLD_START, WARM_START and HM/FM mechanism cannot fail, so the possible fault modes in this process will not be decomposed. Although resource allocation failure will cause the failure of partition software, this failure mode is mainly caused by the failure of core OS, so this failure mode is not decomposed.

Arithmetic errors, stack overflows, and illegal requests are caused by the failure of the hosted application software itself. In the partition communication failure, the thread communication in the hosted application software (In-thread) is scheduled and managed by the basic partition operating system, so the thread communication failure in the hosted application software is caused by the failure of the hosted application software. The essence of thread communication between different partition software in the same IMA module (Inter-thread) is that the basic partition operating system communicates through a shared cache, that is, a storage area is configured as a cache space for communication data, and all partitions participating in the communication can access this space. This operation does not require the involvement of the core OS. Therefore, a failure of the hosted application software can result in failure of thread communication between different partition software within the same IMA module. The thread communication between different IMA modules must involve the core OS, because only the core OS can operate the bus. Therefore, the core OS failure can cause thread communication failure between different IMA modules (Inter-module failure). In two-level scheduling failures, partition context save failures and partition context recovery failures can be further decomposed into partition context unsaved, partition context saves failures, partition context unrecovered and partition context recovery failures according to the different modes of hardware and software failures. Because the failure of the core OS will lead to the error of global variables such as the context pointer variable and GPM system clock in the partition software, partition context unsaved, partition context unrecovered and the failure of the partition timer are all caused by the failure of the core OS. Partition context save failures and partition context recovery failures are caused by the MMU failure of the GPM module [36]. For the partition software failure (HARDWARE FAILURE) caused by the temporary failure of the core OS and GPM module, HM/FM mechanism is adopted to carry out COLD_START to recover, while the partition software failure itself (SOFTWARE FAILURE) will not cause the global variable failure of the partition software, HM/FM mechanism is adopted to carry out WARM_START to recover. Therefore, a temporary failure of the GPM or Core O/S of the IMA system causes the partition software to switch from the NORMAL state to the HARDWARE FAILURE state. A failure of the partition software causes the partition software to transition from the NORMAL state to the SOFTWARE FAILURE state.

In order to truly reflect the running state of partition software on the basis of simplifying the Petri net model, this paper makes the following assumptions and constraints on partition software [20].

i. There will be no failure in the INITIAL of IMA system, and the IMA system can only enter the COLD_START state immediately after INITIAL.

ii. There will be no failure during COLD_START and WARM_START of partition software.

iii. The partition software under NORMAL state will not enter COLD_START or WARM_START state.

iv. The partition software will be restarted once it enters the FAILURE (HARDWARE FAILURE and SOFT-WARE FAILURE) state.

v. The external conditions and storage conditions required by the partition software after restart will be satisfied.

The following describes the operation process of the partition software through an example of the operation process of the partition software. Assuming that two partition software (P1 and P2) are created and run in the GPM module of the IMA platform, Figure 5 shows the running process of the partition software.

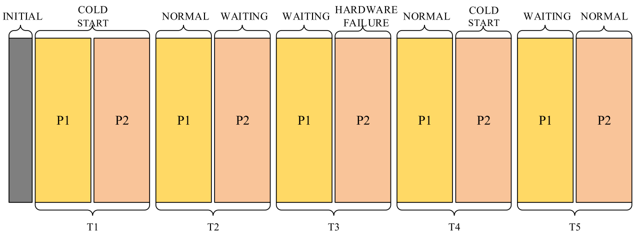


FIGURE 5. Operating process of partition software.

After the GPM module is powered on, the hardware and core OS are initialized first, and then partitions P1 and P2 are created. The partitions P1 and P2 are in the COLD_START state, the partition storage space is initialized, the permanent data variables are initialized, and the thread is started. From the start of the operating cycle T2, the partition software P1 and P2 alternately run, and then in the operating cycle T3, a temporary failure of the GPM or core OS of the IMA system causes a HARDWARE FAILURE of the partition software. In the operating cycle T4, HM/FM performs COLD_START recovery on P2 after detecting an error. Subsequently, partitions P1 and P2 both enter the normal state and alternately occupy processor resources periodically. When the partition SOFTWARE FAILURE at a certain moment, HM/FM will perform WARM_START recovery on the partition after finding the fault.

According to analysis, assumptions and constraints, the Petri SPN model of partition software can be obtained, as shown in Figure 6.

In order to further illustrate that the Petri nets model meets our expected operating state, we use TimeNET4.4 software to check the structure of the model and find that the model has 5 T invariants and no conflict sets. The T invariant meets our expectation, which further verifies the accuracy of the model. Figure 7 shows the results of the structure check of the partition software.

The description of the transition t_i, the firing rate λ_i and reference time corresponding to the transition t_i are shown in Table 3.

IV. RELIABILITY ASSESSMENT OF PARTITION SOFTWARE BASED ON SPN

By analyzing the interaction process and failure mode of partition software, the interaction process and failure mode of partition software are defined. According to the characteristics of partition software, combined with Petri nets suitable for describing asynchronous, concurrent, and uncertain systems, as well as having the advantages of intuitive,

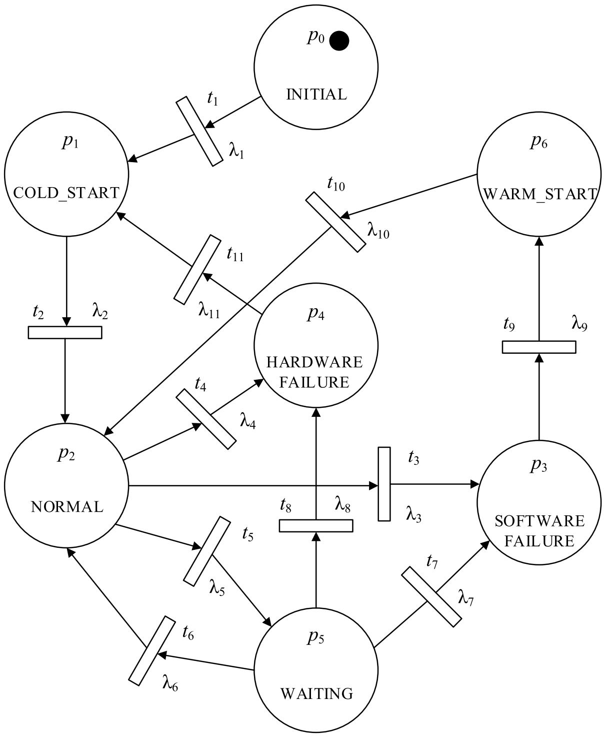
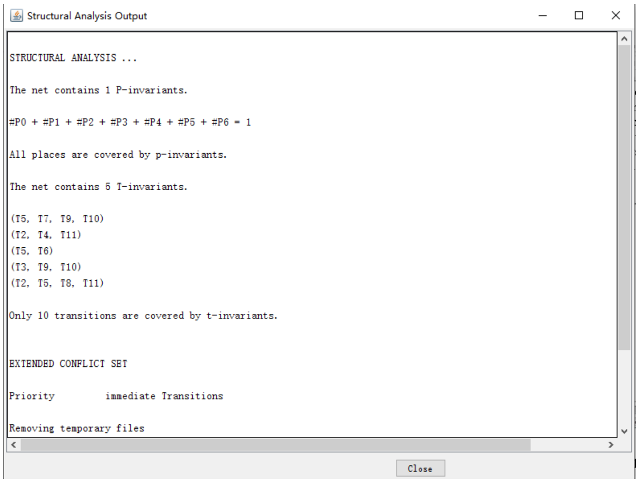


FIGURE 6. The Petri nets model of partition software.



```
STRUCTURAL ANALYSIS ...

The net contains 1 P-invariants.

#P0 + #P1 + #P2 + #P3 + #P4 + #P5 + #P6 = 1

All places are covered by p-invariants.

The net contains 5 T-invariants.

(T5, T7, T9, T10)
(T2, T4, T11)
(T5, T6)
(T3, T9, T10)
(T2, T5, T8, T11)

Only 10 transitions are covered by t-invariants.


EXTENDED CONFLICT SET

Priority        immediate Transitions

Removing temporary files
```

FIGURE 7. Result of a structure check.

visual, and perfect mathematical theory, this paper proposes a method for modeling and evaluating the reliability of partition software based on SPN, establishing a reliability mathematical model of partition soft-ware, and quantitatively analyzing the factors that affect the reliability of partition software [37].

Definition 1: The necessary and sufficient conditions for a seven-tuple $SPN = (P, T; F, K, W, M, \lambda)$ to be a SPN a follow.

(*i*)$P \cup T \neq \emptyset$ (Net is not empty).

(*ii*) $P \cap T = \emptyset$ (Duality).

(*iii*) $F \subseteq ((P \times T) \cup (T \times P))$. (The flow relationship is only between the elements of P and T).

(*iv*) $dom\,(F) \cup cod\,(F) = P \cup T$. (No isolated elements)

TABLE 3. The transition in the petri model of partition software.

Transition	Description	Firing rate	Related Reference Time
t_1	IMA system initialization	λ_1	GPM and Core OS initialization time
t_2	Initialize partition storage space, initialize permanent data variables and start threads	λ_2	Partition COLD_START initialization time
t_3	Partition software failure	λ_3	MTBF of partition software
t_4	A temporary failure of GPM or Core OS of an IMA system causes the partition software to fail	λ_4	MTBF of IMA module
t_5	The scheduling partition goes into WAITING state	λ_5	The time for partition running and partition switching.
t_6	The scheduling partition goes into NORMAL state	λ_6	The time for partition waiting and partition switching
t_7	Partition software failure	λ_7	MTBF of partition software
t_8	A temporary failure of GPM or Core OS of an IMA system causes the partition software to fail	λ_8	MTBF of IMA module
t_9	HM/FM find a failure	λ_9	The time for HM/FM find a failure
t_{10}	Initialize the local variables of the partition software, start the scheduling function of the partition OS	λ_{10}	Initialization time for WARM_START
t_{11}	HM/FM find a failure	λ_{11}	The time for HM/FM find a failure

(v) $K : P \rightarrow N^+ \cup \{\infty\}$ is the capacity function of the place.

(*vi*) $W : F \rightarrow N^+$ is the weight function of the directed arc.

(*vii*) $M : P \rightarrow N$ is the initial marking, satisfy: $\forall p \in P : M(p) \leq K(p)$.

(*viii*) $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ is the set of transition firing rates.

Where $dom(F) = \{x | \exists y : (x, y) \in F\}$ is the set of the first element of the order couple contained in F. $cod(F) = \{x | \exists y : (y, x) \in F\}$ is the set of the second element of the order couple contained in F. $P = \{p_1, p_2, \cdots, p_n\}$ is a limited set of places. $T = \{t_1, t_2, \cdots, t_n\}$ is a limited set of transition. N is A natural number. N^+ is a positive natural number [38].

In SPN, the transition T from enabling to implementing requires a delay, which can be regarded as a random variable x_i, and the random variable x_i obeys a distribution function: $F_i(x) = P\{x_i \leq x\}$ [39]. In the continuous-time SPN proposed by Molloy, the distribution function of each transition follows an exponential distribution with parameter λ_i, that is

$$\forall t \in T : F_t = 1 - e^{-\lambda_i x} \tag{1}$$

where the average firing rate $\lambda_i > 0$, variable $x \geq 0$. The following conclusions can be proved [40].

i. The probability that two transitions will be firing at the same moment is zero.

ii. The reachable state graph of SPN is isomorphic to a homogeneous MC, so it can be solved by Markov stochastic process.

Therefore, before using the SPN method, we need to prove that the delay associated with each transition in the Petri net model of the IMA partition software obeys an exponential distribution.

i. Delay of transition t_4, t_8

The research of electronic products is relatively mature, and the failure rate of electronic products has the characteristics of no memory. Therefore, the failure rate of electronic products obeys an exponential distribution. As a kind of electronic system, the IMA module also obeys an exponentially distributed random variable. So the firing rate of transition t_4 and t_8 meets the application conditions of SPN.

ii. Delay of transition t_3, t_5, t_6, t_7

In the Petri net model of partition software, the transitions associated with the MTBF of software are t_3 and t_7, while t_3, t_4, and t_5 have a competitive relationship, and t_6, t_7, and t_8 have a competitive relationship. The delays corresponding to transitions t_5 and t_6 are usually milliseconds. The MTBF of the IMA module corresponding to the transition t_4 and t_7 is generally 10^5, which is quite different from the delay (10^4) corresponding to the transition t_3 and t_8. Therefore, approximating the delay of transition t_3, t_5, t_6, t_7 to exponential distribution does not affect the change of the model state. So the firing rate of transition t_3, t_5, t_6, t_7 meets the application conditions of SPN.

iii. Delay associated with other transitions

The delay of GPM and Core O/S has nothing to do with the steady-state probability of the partition software in a failure state. The delay for HM/FM to find errors, the HOT_START of the partition and the initialization process of each variable in the COLD_START are all random Variable, the probability distribution of this variable decreases with time, which can be similar to an exponential distribution. So the firing rates of t_1, t_2, t_9, t_{10}, and t_{11} meet the application conditions of SPN.

A. QUANTITATIVE ANALYSIS OF RELIABILITY OF PARTITION SOFTWARE BASED ON SPN

The reachable status tree of partition software is shown in Figure 8. There are seven possible locations in the partition software. At the beginning, there is only one token in INITIAL, and the location of INITIAL is marked as $M_0 = (1, 0, 0, 0, 0, 0, 0)$. The firing rate in the SPN model of the partition software determines the position of token, and

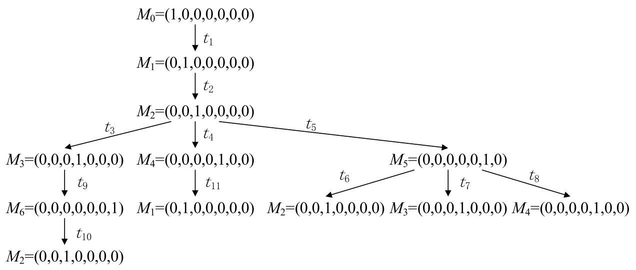


FIGURE 8. The reachable status tree of partition software.

the marking of the partition software in different positions can be obtained through the change of token.

In order to more intuitively reflect the transition relationship between various states in SPN, the reachable state tree of partition software can be further summarized to obtain the reachable state diagram of partition software, as shown in Figure 8.

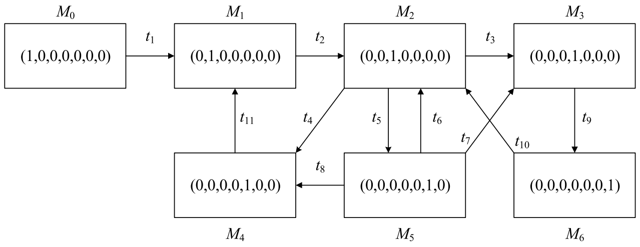


FIGURE 9. The reachable state diagram of partition software.

It can be seen from Figure 9 that the reachable state diagram of partition software is reversible, at least it can be cycled between M_1 and M_6. Then, the same status identification in the partition software reachable status diagram is merged, and the corresponding firing delay of each state is replaced by the corresponding firing rate. Refer to Table 2 for the conversion of firing delay and firing rate for each state, and a continuous-time MC isomorphic to the reachable state diagram of the partition software can be obtained, as shown in Figure 10.

In order to get the transition matrix of the MC, a $n \times n$ matrix $Q = [q_{ij}]|_{n \times n}$ is defined. The calculation method of each element is as follows.

$$q_{ij} = \begin{cases} \lambda_k, & i \neq j \wedge t_k \in MC_i[t_k > MC_j \\ -\sum\limits_{M_i[t_k>} \lambda_k, & i = j \\ 0, & i \neq j \wedge t_k \notin MC_i[t_k > MC_j \end{cases} \tag{2}$$

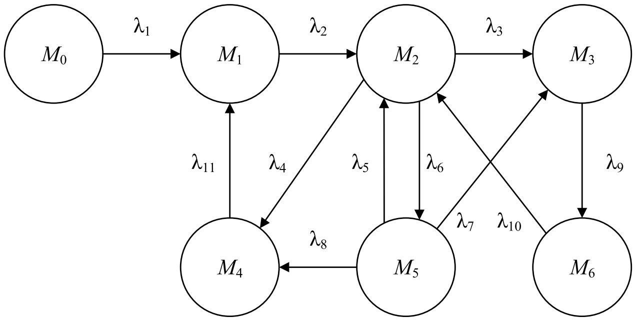


FIGURE 10. Isomorphic MC of partition software SPN model.

where $MC_i[t_k > MC_j$ is the transition t_k satisfies the partition condition to transfer to MC_j in state MC_i. According to formula (2), the transfer matrix $Q_{7\times7}$ of MC with isomorphic partition software SPN model can be obtained as follows (3), as shown at the bottom of the page.

Suppose the steady-state probability of partition software in each reachable state is $P[M_i] = x_{i+1}$. The element x_i in the steady-state probability set $X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ can be determined by the following system of equations.

$$\begin{cases} XQ = 0 \\ \sum x_i = 1, \quad 1 \leq i \leq n \end{cases} \tag{4}$$

Taking matrix Q and matrix X into the above formula (4), the following equations can be obtained

$$\begin{cases} -\lambda_1 x_1 = 0 \\ \lambda_1 x_1 - \lambda_2 x_2 + \lambda_{11} x_5 = 0 \\ \lambda_2 x_2 + (-\lambda_3 - \lambda_4 - \lambda_5) x_3 + \lambda_6 x_6 + \lambda_{10} x_7 = 0 \\ \lambda_3 x_3 - \lambda_9 x_4 + \lambda_7 x_6 = 0 \\ \lambda_4 x_3 - \lambda_{11} x_5 + \lambda_8 x_6 = 0 \\ \lambda_5 x_3 + (-\lambda_6 - \lambda_7 - \lambda_8) x_6 = 0 \\ \lambda_9 x_4 - \lambda_{10} x_7 = 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 1 \end{cases} \tag{5}$$

Assuming that the random variable t follows the exponential distribution with the parameter λ, and the corresponding average delay is $\overline{d}$, the probability distribution function of the random variable t is

$$F_t(t) = P[T \leq t] = 1 - e^{-\lambda t} \tag{6}$$

where the average delay is given by

$$\overline{d} = \int_0^\infty [1 - F_t(t)]\, dt = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda} \tag{7}$$

$$Q_{7\times7} = \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\lambda_3 - \lambda_4 - \lambda_5 & \lambda_3 & \lambda_4 & \lambda_5 & 0 \\ 0 & 0 & 0 & -\lambda_9 & 0 & 0 & \lambda_9 \\ 0 & \lambda_{11} & 0 & 0 & -\lambda_{11} & 0 & 0 \\ 0 & 0 & \lambda_6 & \lambda_7 & \lambda_8 & -\lambda_6 - \lambda_7 - \lambda_8 & 0 \\ 0 & 0 & \lambda_{10} & 0 & 0 & 0 & -\lambda_{10} \end{bmatrix} \tag{3}$$

TABLE 4. The relationship between the average time of each parameter and the firing rate.

Time	Average firing rate
Average time of COLD_START: T_C	$\lambda_2=1/T_C$
MTBF of partition software: $1/\lambda$	$\lambda_3=\lambda_7=\lambda$
MTBF of IMA module: $1/\lambda'$	$\lambda_4=\lambda_8=\lambda'$
Average time of partition software in normal state: T_N	$\lambda_5=1/T_N$
Average time of partition software in the WAITING state: T_H	$\lambda_6=1/T_H$
Average time for HM/FM finds a failure: T_F	$\lambda_9=\lambda_{11}=1/T_F$
Average time of WARM_START: T_W	$\lambda_{10}=1/T_W$

Therefore, the steady-state probability of each reachable state of partition software is as follows (8), as shown at the bottom of the page,where the relationship between the average time of each parameter and the firing rate is shown in Table 4.

V. EXPERIMENTAL VERIFICATION AND ANALYSIS

In order to find out which factors are related to the steady-state probability of partition software in failure state, it is necessary to separate the steady-state probability of the partition software in the working state and the failure state. The partition software can work normally only in the NORMAL state and the WAITING state and cannot work in other states. Therefore, the steady-state probability X_F of the partition software failure is as follows

$$\begin{aligned} X_F &= x_2 + x_4 + x_5 + x_7 \\ &= \frac{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F}{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1} \end{aligned} \tag{9}$$

The IMA system has N IMA modules, and each IMA module is embedded with several partition software. There is interaction between the partition software within the IMA module and between the partition software within different IMA modules. It is assumed that the running state of the partition software can meet the constraints and assumptions on the partition software given in section 3. The verification contents are as follows.

i. The impact of T_C on the reliability of the partition software is inversely proportional to the MTBF of the IMA module.

ii. The impact of T_F on the reliability of the partition software is almost independent of the MTBF of the IMA module.

iii. The impact of T_W on the reliability of the partition software is independent of the MTBF of the IMA module.

iv. The impact of λ on the reliability of the partition software is independent of the MTBF of IMA module.

v. T_C has little effect on the reliability of the partition software.

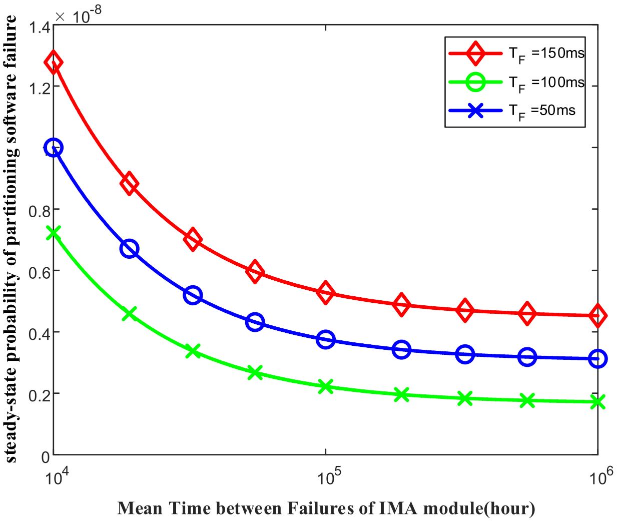


FIGURE 11. T_F impact on the reliability of partition software.

According to references [41]–[45], it can be obtained that: $T_C = 150ms$, $T_F = 150ms$, $T_W = 10ms$, $\lambda = 1/10^4 h = 10^{-4}$ and $\lambda' = 1/10^5 h = 10^{-5}$. The pseudo-code of the Partition software reliability analysis is given by Algorithm I.

$$\begin{cases} x_1 = 0 \\ x_2 = \dfrac{\lambda' T_C}{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1} \\ x_3 = \dfrac{T_N(T_H\lambda + \lambda' T_H + 1)}{(T_H + T_N + \lambda' T_H T_N + T_H T_N \lambda) \times (T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1)} \\ x_4 = \dfrac{T_F\lambda}{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1} \\ x_5 = \dfrac{\lambda' T_F}{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1} \\ x_6 = \dfrac{T_H}{(T_H + T_N + \lambda' T_H T_N + T_H T_N \lambda) \times (T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1)} \\ x_7 = \dfrac{T_W\lambda}{T_F\lambda + T_W\lambda + \lambda' T_C + \lambda' T_F + 1} \end{cases} \tag{8}$$

Algorithm 1: Partition Software Reliability Analysis Algorithm

```
Input: $x \in [T_F, T_C, T_W, \lambda]$
begin
    if $x$ is $T_F$ then
        $\lambda = 10^{-4}, T_C = 150\ ms, T_W = 10\ ms$
        foreach hardware failure rate
          $\lambda_i' \in [10^{-4}, \cdots, 10^{-6}]$ do
            foreach HM/FM
              $T_{Fj} \in [150ms, 100ms, 50ms]$ do
                $\mathrm{res}_{ij} = x2 + x4 + x5 + x7 = \frac{T_{Fj}\lambda + T_W\lambda + \lambda_i' T_C + \lambda_i' T_{Fj}}{T_{Fj}\lambda + T_W\lambda + \lambda_i' T_C + \lambda_i' T_{Fj} + 1}$
            end
        end
    elseif $x$ is $T_C$ then
        $\lambda = 10^{-4}, T_F = 150\ ms, T_W = 10\ ms$
        foreach hardware failure rate
          $\lambda_i' \in [10^{-4}, \cdots, 10^{-6}]$ do
            foreach COLD_START
              $T_{Cj} \in [150ms, 100ms, 50ms]$ do
                $\mathrm{res}_{ij} = x2 + x4 + x5 + x7 = \frac{T_{Fj}\lambda + T_W\lambda + \lambda_i' T_{Cj} + \lambda_i' T_F}{T_{Fj}\lambda + T_W\lambda + \lambda_i' T_{Cj} + \lambda_i' T_F + 1}$
            end
        end
    elseif $x$ is $T_W$ then
        $\lambda = 10^{-4}, T_F = 150\ ms, T_C = 150\ ms$
        foreach hardware failure rate
          $\lambda_i' \in [10^{-4}, \cdots, 10^{-6}]$ do
            foreach WARM_START
              $T_{Wj} \in [20ms, 10ms, 1ms]$ do
                $\mathrm{res}_{ij} = x2 + x4 + x5 + x7 = \frac{T_{Fj}\lambda + T_{Wj}\lambda + \lambda_i' T_C + \lambda_i' T_F}{T_{Fj}\lambda + T_{Wj}\lambda + \lambda_i' T_C + \lambda_i' T_F + 1}$
            end
        end
    else
        $T_F = 150, T_C = 150, T_W = 10$
        foreach hardware failure rate
          $\lambda_i' \in [10^{-4}, \cdots, 10^{-6}]$ do
            foreach MTBF of partition software
              $\lambda_j \in [10^{-3}, 5 * 10^{-4}, 10^{-4}, 10^{-5}]$ do
                $\mathrm{res}_{ij} = x2 + x4 + x5 + x7 = \frac{T_F\lambda_j + T_W\lambda_j + \lambda_i' T_C + \lambda_i' T_F}{T_{Fj}\lambda_j + T_W\lambda_j + \lambda_i' T_C + \lambda_i' T_F + 1}$
            end
        end
    end
end
Output: $\mathrm{res}_{ij}$
```

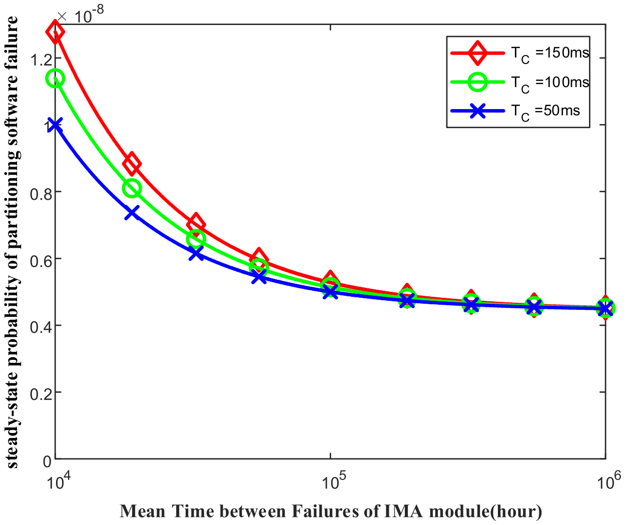


FIGURE 12. T_C impact on the reliability of partition software.

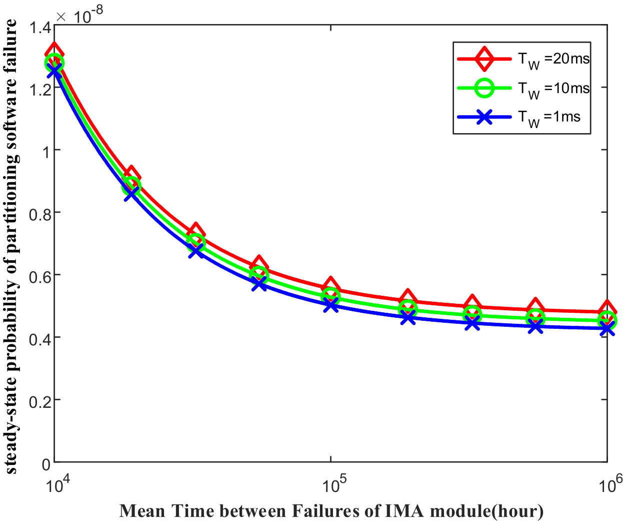


FIGURE 13. T_W impact on the reliability of partition software.

The impact of changes in T_F, T_C, T_W and λ on the reliability of partition software is shown in Figure 11, 12, 13 and 14. It can be seen in Figure 11 that the impact of T_F on the reliability of the partition software is almost independent of the MTBF of the IMA module. It can be seen in Figure 12 that as the MTBF of the IMA module increases, the benefits of reducing T_C to improve the reliability of the partition software will be significantly reduced. In Figure 13, it can be seen that the impact of T_W on the reliability of partition software is independent of the MTBF of IMA module. As can be seen in Figure 14, when the MTBF λ of the partition software is small, the increase of λ will significantly improve the reliability of the partition software. However, as the MTBF of the partition software λ increases, the benefits of increasing the MTBF of the partition software will significantly decrease and the impact of λ on the reliability of partition software is independent of the MTBF of IMA module.

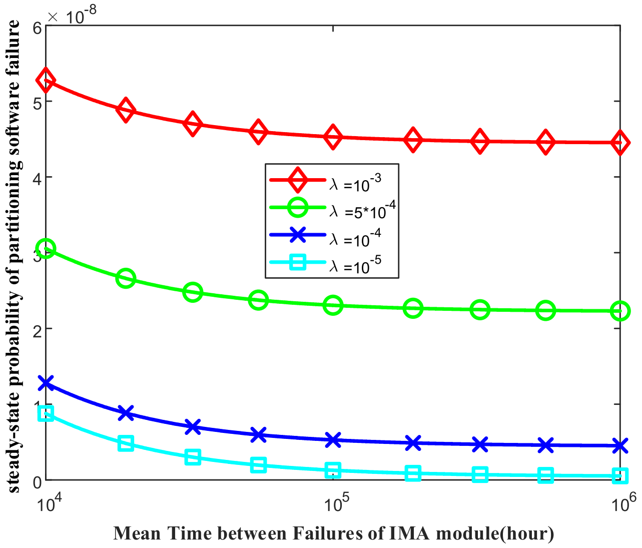


FIGURE 14. λ impact on the reliability of partition software.

In summary, the steady-state probability of the partition software in the failure state is independent of the average time T_N of the partition software in the NORMAL state and the average time T_H of the partition software in the WAITING state. When the MTBF of partition software is between 10^3h and 10^4h, increasing the MTBF of partition software will significantly improve the reliability of partition software. When the MTBF of the partition software is above 10^4h, the difficulty of increasing the MTBF of the partition software will be significantly increased, and the gain will be significantly reduced. Therefore, the reliability of the partition software can be improved by reducing the mean time of HM/FM fault detection, the mean time of CLOD_START and the mean time of WARM_STARM.

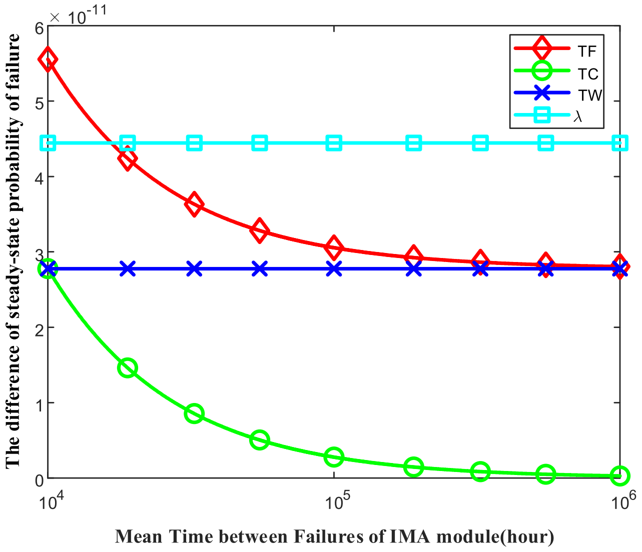
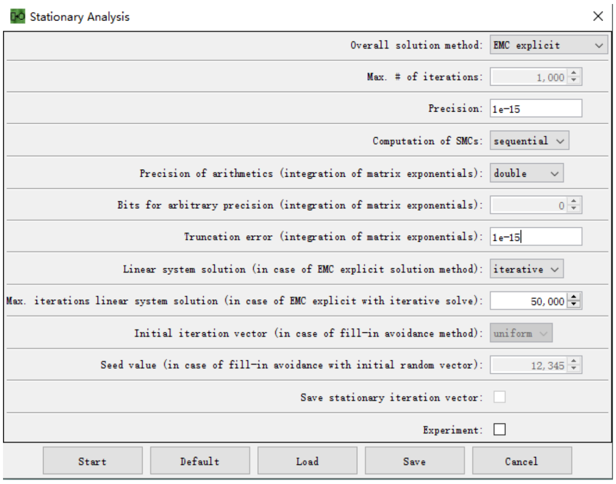


FIGURE 15. The influence of various factors on the reliability of partition software.

In order to obtain the influence of A on the reliability of partition software more clearly, the influence of various factors on the reliability of partition software is obtained, as shown in Figure 15. The curves *TF*, *TC* and *FW* represent the influence of T_F, T_C and T_W on the reliability of partition software for each decrease of 1ms, respectively. λ represents the impact of increasing the MTBF of partition software by an average of 1000 hours between 10^4h and 10^5h on the reliability of partition software.


Stationary Analysis

Overall solution method: EMC explicit

Max. # of iterations: 1,000

Precision: 1e-15

Computation of SMCs: sequential

Precision of arithmetics (integration of matrix exponentials): double

Bits for arbitrary precision (integration of matrix exponentials): 0

Truncation error (integration of matrix exponentials): 1e-15

Linear system solution (in case of EMC explicit solution method): iterative

Max. iterations linear system solution (in case of EMC explicit with iterative solve): 50,000

Initial iteration vector (in case of fill-in avoidance method): uniform

Seed value (in case of fill-in avoidance with initial random vector): 12,345

Save stationary iteration vector: □

Experiment: □

FIGURE 16. Simulation environment parameters.

The MTBF of the current IMA module is generally 10^5h. As can be seen from Figure 15, the average time for HM/FM to detect faults is reduced by $1ms$, the average time for WARM_STARM is reduced by $1ms$, and the MTBF of partition software is increased by 1000h, which will reduce the steady-state probability of partition software in the state of failure by the same order of magnitude (10^{-11}). The average time of CLOD_START has little effect on the reliability of the partition software (10^{-12}). To reduce the average time of HM/FM fault detection, the monitoring mode of HM/FM software can be used to optimize the monitoring mechanism of HM/FM. Optimizing the code architecture can reduce the time of WARM_START of the partition software. As shown in Figure 4, improving the MTBF of partition software can refer to methods such as improving the reliability of communication between partition software threads, the reliability of communication between partition software of different IMA modules, exception handling and the garbage disposal mechanism of partition software heap memory.

In order to further study the sensitivity of each parameter. We build the SPN model of partition software in TimeNET4.4. The simulation environment as shown in Figure 16 is set to obtain the sensitivity of different parameters to the model, as shown in Table 5. It can be seen from the table that as the MTBF of the partition software increases, the sensitivity of the model to the model becomes less and less for every 1000h increase in the MTBF of the partition software. In addition to λ, the average time for HM/FM to find a failure has the greatest impact on the partition software.

In order to improve the reliability of IMA partition software, different analysis methods and modeling methods have

TABLE 5. Sensitivity to parameter.

Parameter	The steady-state probability of the partition software in the fault state
T_C=150ms, T_F=150ms, T_W=10ms, λ=1/10000, λ'=1/100000	$5.2778*10^{-9}$
T_C=149ms, T_F=150ms, T_W=10ms, λ=1/10000, λ'=1/100000	$5.2750*10^{-9}$
T_C=148ms, T_F=150ms, T_W=10ms, λ=1/10000, λ'=1/100000	$5.2722*10^{-9}$
T_C=150ms, T_F=149ms, T_W=10ms, λ=1/10000, λ'=1/100000	$5.2472*10^{-9}$
T_C=150ms, T_F=148ms, T_W=10ms, λ=1/10000, λ'=1/100000	$5.2167*10^{-9}$
T_C=150ms, T_F=150ms, T_W=9ms, λ=1/10000, λ'=1/100000	$5.2500*10^{-9}$
T_C=150ms, T_F=150ms, T_W=8ms, λ=1/10000, λ'=1/100000	$5.2222*10^{-9}$
T_C=150ms, T_F=150ms, T_W=10ms, λ=1/11000, λ'=1/100000	$4.8737*10^{-9}$
T_C=150ms, T_F=150ms, T_W=10ms, λ=1/12000, λ'=1/100000	$4.5370*10^{-9}$

TABLE 6. Analysis and comparison of various modeling methods.

Modeling method	λ	T_C	T_W	T_F	environmental factor	Quantitative assessment
Wang[3]	√	√	×	×	×	×
Wang[22]	×	×	×	×	√	√
The method in this article	√	√	√	√	×	√

Quantitative assessment: The degree of change of the partition software failure rate with parameters.

been proposed in the academic community, mainly focusing on the environmental factors and the running state of IMA system software. On the basis of Wang *et al.* [3], according to ARINC 653 standard, this paper establishes a model more in line with the operating state of IMA partition software, and analyze the factors affecting the reliability of partition software, as shown in Table 6.

It can be seen from the above table that the method of Wang *et al.* [3] also proposes the SPN reliability evaluation model of the IMA system partition software, the model merges the COLD_START state and the WARM_START state defined in the ARINC 653 standard, which reduces the accuracy of the model. The model did not analyze the influence of the average initialization time of COLD_START, the average initialization time of WARM_START, and the average time of HM/FM to find errors on the model. The method of Wang and Sun [22] only analyzes the quantitative assessment of the MTBF of the partition software by specific environmental factors. This paper makes a quantitative assessment of the reliability factors of IMA partition software and obtains four factors that affect the reliability of IMA partition software to provide more detailed guidance to relevant developers.

VI. CONCLUSION

The IMA system is the future of the onboard avionics system. There are different numbers of partition software in the IMA system. In order to improve the reliability of the IMA system, this paper studies the SPN reliability quantitative evaluation model of partition software. On the basis of proving that the trigger rate of each transition in the model is similar to the exponential distribution, Markov stochastic process theory is used to obtain the steady-state probability of the partition software in the fault state. Under the assumptions presented, it is a function of the MTBF of the partition software, the mean time of the COLD_START of the partitioning software, the mean time of the HOT_START of the partition software, and the mean time of the HM/FM fault detection. It can be proved that the steady-state probability of partition software is independent from the schedule time. Through parameter sensitivity analysis, the sensitivity of different parameters to the model is obtained. Some suggestions for increasing the reliability of partition are discussed. In this paper, SPN model of single partition software is established, but in the actual running process of the IMA system, it is often multiple partition software programs are often used together to complete an aviation task.

In the future, the SPN model of multiple partition software interaction is considered, the influence of fault propagation mechanism between partitioned software on the reliability of partitioned software is determined, and quantitative analysis of module level software reliability is carried out.

REFERENCES

[1] *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, RTCA, Washington, DC, USA, RTCA DO-297, 2005.

[2] P. J. Prisaznuk, "ARINC 653 role in integrated modular avionics (IMA)," in *Proc. IEEE/AIAA 27th Digit. Avionics Syst. Conf.*, Saint Paul, MN, USA, Oct. 2008, pp. 1.E.5-1–1.E.-5-10.

[3] Y.-S. Wang, H. Lei, and X. Han, "The stochastic Petri net based reliability analysis for software partition integrated modular avionics," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 4, pp. 30–37, Apr. 2015.

[4] P. Wang, D. Wang, C. Zhu, Y. Yang, H. M. Abdullah, and M. A. Mohamed, "Stochastic management of hybrid AC/DC microgrids considering electric vehicles charging demands," *Energy Rep.*, vol. 6, pp. 1338–1352, Nov. 2020.

[5] J. S. Lei, X. Su, and M.-L. Jin, "A distributed sustainable integrated automated testing platform," *Comput. Modernization*, vol. 296, no. 4, pp. 14–18, 2020.

[6] M. A. Mohamed, E. M. Awwad, A. M. El-Sherbeeny, E. A. Nasr, and Z. M. Ali, "Optimal scheduling of reconfigurable grids considering dynamic line rating constraint," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 10, pp. 1862–1871, May 2020.

[7] D. Deb, "Method and engine controller for diagnosing waste gate valve malfunction and related power generation system," U.S. Patent Appl. 16546 264, Feb. 13, 2020.

[8] U. Prajapati, A. Rawat, and D. Deb, ''Integrated peripheral security system for different areas based on exchange of specific data rates,'' *Wireless Pers. Commun.*, vol. 111, no. 3, pp. 1355–1366, Apr. 2020.

[9] U. Prajapati, A. Rawat, and D. Deb, ''A novel approach towards a low cost peripheral security system based on specific data rates,'' *Wireless Pers. Commun.*, vol. 99, no. 4, pp. 1625–1637, Apr. 2018.

[10] B. Cai, M. Xie, Y. Liu, and Y. Liu, ''Availability-based engineering resilience metric and its corresponding evaluation methodology,'' *Rel. Eng. Syst. Saf.*, vol. 172, pp. 216–224, Apr. 2018.

[11] B. Cai, X. Kong, Y. Liu, J. Lin, X. Yuan, H. Xu, and R. Ji, ''Application of Bayesian networks in reliability evaluation,'' *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2146–2157, Apr. 2019.

[12] B. Cai, H. Fan, X. Shao, Y. Liu, G. Liu, Z. Liu, and R. Ji, ''Remaining useful life re-prediction methodology based on Wiener process: Subsea Christmas tree system as a case study,'' *Comput. Ind. Eng.*, Nov. 2020, Art. no. 106983, doi: 10.1016/j.cie.2020.106983.

[13] T. Robati, A. Gherbi, and J. Mullins, ''A modeling and verification approach to the design of distributed IMA architectures using TTEthernet,'' *Procedia Comput. Sci.*, vol. 83, pp. 229–236, Jan. 2016.

[14] H. Dong, X. Bao, T. Zhao, and Y. Xin, ''Integrated modular avionics fault propagation model establishment and reliability allocation,'' in *Proc. 12th Int. Conf. Rel., Maintainability, Saf. (ICRMS)*, Shanghai, China, Oct. 2018, pp. 377–382.

[15] P. Wang, R. Liu, and W. H. Liu, ''On reliability assessment method of integrated modular avionics system,'' *Electron. Opt. Control*, vol. 2015, pp. 60–65, Oct. 2015.

[16] L. K. Rierson, ''Best practices for certifying IMA systems in civil aircraft,'' *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 1, pp. 4–8, Jan. 2010.

[17] C. Philippa and M. John, ''High level failure analysis for integrated modular avionics,'' in *Proc. 6th Austral. Workshop Saf. Crit. Syst. Softw.*, vol. 3, 2001, pp. 13–21.

[18] F. Yan, P. Xing, C. Zhao, and P. Wang, ''Reliability modeling and analysis of DIMA system based on joint k/n(G) model,'' *Acta Aeronautica et Astronautica Sinica*, vol. 39, no. 6, pp. 321971-1–321971-9, 2018.

[19] Q. Zhang, S. Wang, and B. Liu, ''Approach for integrated modular avionics reconfiguration modelling and reliability analysis based on AADL,'' *IET Softw.*, vol. 10, no. 1, pp. 18–25, Feb. 2016.

[20] Y. H. Wang, ''Research on reliability of partition software in integrated modular avionics,'' Univ. Electron. Sci. Technol. China, Chengdu, China, Tech. Rep. 2017.

[21] J. Wan, X. Xiang, X. Bai, C. Lin, X. Kong, and J. Li, ''Performability analysis of avionics system with multilayer HM/FM using stochastic Petri nets,'' *Chin. J. Aeronaut.*, vol. 26, no. 2, pp. 363–377, Apr. 2013.

[22] W. Chong and S. Haiyan, ''A reliability model of integrated modular avionics (IMA) software considering with environment,'' in *Proc. Int. Conf. Manage. Eng., Softw. Eng. Service Sci. (ICMSS)*, Wuhan, China, Jan. 2017, pp. 49–53.

[23] I. Kabashkin and V. Filippov, ''Reliability of software applications in integrated modular avionics,'' *Transp. Res. Procedia*, vol. 51, pp. 75–81, Jan. 2020.

[24] Y. Xie and X. Zhang, ''Performance analysis of wireless communication for high-speed trains based on SPN,'' *J. Phys., Conf.*, vol. 1624, Oct. 2020, Art. no. 042044.

[25] X.-Y. Li, Y. Liu, Y.-H. Lin, L.-H. Xiao, E. Zio, and R. Kang, ''A generalized Petri net-based modeling framework for service reliability evaluation and management of cloud data centers,'' *Rel. Eng. Syst. Saf.*, vol. 207, Mar. 2021, Art. no. 107381.

[26] X. M. Wei, Z. Q. Dong, M. R. Xiao, and C. Tian, ''Failure probabilities allocation and safety assessment approaches based on AADL,'' *J. Softw.*, vol. 31, no. 6, pp. 1654–1671, 2020.

[27] *Airlines Electronic Engineering Committee. Avionics Application Standard Software Interface ARINC Specification 653-1*, Aeronautical Radio, Annapolis, MD, USA, 2003.

[28] Y. Wang, H. Lei, R. Hackett, and M. Beeby, ''Safety assessment process optimization for integrated modular avionics,'' *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 11, pp. 58–67, Nov. 2019.

[29] J. Windsor and K. Hjortnaes, ''Time and space partitioning in spacecraft avionics,'' in *Proc. 3rd IEEE Int. Conf. Space Mission Challenges Inf. Technol.*, Pasadena, CA, USA, Jul. 2009, pp. 13–20.

[30] D. Mazuk, ''IMA resource allocation process,'' *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 3, pp. 30–34, Mar. 2010.

[31] *National Aeronautics and Space Administration*, NASA Software Safety Standard NASA-STD-8719.13B, Ju. 2004.

[32] *ARINC Specification 653: Part 1. Avionics Application Software Standard Interface, Required Services*, ARINC, Annapolis, MD, USA, Mar. 2006. [Online]. Available: https://www.arinc.com/cf/store/index.cfm

[33] *ARINC Specification 653: Part 2. Avionics Application Software Standard Interface, Extended Services*, ARINC, Annapolis, MD, USA, Jan. 2007. [Online]. Available: https://www.arinc.com/cf/store/index.cfm

[34] *ARINC Specification 653: Part 3, Avionics Application Software Standard Interface, Conformity Test Specification*, ARINC, Annapolis, MD, USA, Oct. 2006. [Online]. Available: https://www.arinc.com/cf/store/index.cfm

[35] W. Yunsheng and L. Hang, ''Failure mode and effects analysis of partition software,'' (in Chinese), *Appl. Res. Comput.*, vol. 34, no. 8, pp. 2399–2403, 2017.

[36] M. D. Bennett and N. C. Audsley, ''Predictable and efficient virtual addressing for safety-critical real-time systems,'' in *Proc. 13th Euromicro Conf. Real-Time Syst.*, Delft, The Netherlands, 2001, pp. 183–190.

[37] C. A. Petri, ''Concepts of net theory,'' in *Proc. Symp. Summer School, Math. Found. Comput. Sci.*, High Tatras, Slovak, 1973, pp. 137–146.

[38] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi, ''Analysis of software rejuvenation using Markov regenerative stochastic Petri net,'' in *Proc. 6th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Toulouse, France, 1995, pp. 180–187.

[39] M. K. Molloy, ''Performance analysis using stochastic Petri nets,'' *IEEE Comput. Archit. Lett.*, vol. CAL-31, no. 9, pp. 913–917, Sep. 1982.

[40] W. M. Zuberek, ''Performance evaluation using unbounded timed Petri nets,'' in *Proc. 3rd Int. Workshop Petri Nets Perform. Models (PNPM)*, Kyoto, Japan, 1989, pp. 180–186.

[41] M. Ian, S. Allan, and J. Malcolm, *Civil Avionics Systems*. London, U.K.: Wiley, 2013, pp. 261–263.

[42] Y. H. Wang and H. Lei, ''Partition configuration and initialization in integrated modular avionics,'' (in Chinese), *J. Comput. Appl.*, vol. 37, no. 6, pp. 1808–1813, 2017.

[43] W. Q. Guo and X. Q. Wang, ''A method of capturing safety critical airborne software requirements,'' (in Chinese), *Aeronaut. Sci. Technol.*, vol. 24, no. 15, pp. 45–51, 2014.

[44] *SAE ARP 4754A, Guidelines for Development of Civil Aircraft and Systems*, SAE, Warrendale, PA, USA, 2010.

[45] *SAE ARP 4761 Guidelines and Methods for the Safety Assessment Process on Civil Airborne Systems Equipment*, SAE, Warrendale, PA, USA, 1996.

WU ZHIJUN received the B.S. and M.S. degrees in information processing from Xidian University, China, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, China. He was a Professor with the Department of Communication Engineering, Civil Aviation University of China. His research interests include aeronautical telecommunication networks and security in big data, and cloud computing.

MA HAOLIN received the B.S. degree in electronic information science and technology from the Zhengzhou University of Light Industry, China. He is currently pursuing the master's degree in information security with the Civil Aviation University of China. His research interest includes integrated modular avionics.

YUE MENG received the Ph.D. degree in information and communication engineering from Tianjin University, China, in 2017. He is currently an Associate Professor with the School of Electronics and Information Engineering and Automation, Civil Aviation University of China. His current research interests include information security and cloud computing.