COMBINED HARDWARE AND SOFTWARE MODELS OF RELIABILITY & AVAILABILITY FOR CONFIGURATION WITH REDUNDANT HARDWARE

Xu Bin
(Jiangxi Province Patent Office, Nanchang, China, 330046)
Yao Yiping
(No.303, Beijing University of Aeronautics & Astronautics,
Beijing, China, 100083)

ABSTRACT

This paper consists of three parts:

In the first part, the development of combined hardware and software reliability analysis methods are summarised.

In the second part, the prerequisite of modeling and two combined HW / SW reliability & availability models are presented. The theoretical analytical model based on Markov renewal processes, the numerieal model based on Markov processes. The former is very complex for solving the equations, but the latter is a practical model in which the series section is divided into sections for treating and the periods between each renewal time points are described by smooth Markov processes.

In the third part, the numerical model has been used to analyze the HW / SW reliability and availability for Fly–By–Wire (FBW) flight control system configuration with redundant hardware in numerical quantities.

I. INTRODUCTION

With the rapid development of the computer techniques, the computer systems have been applied in wide field and become more complicated. In order to improve their reliability and availability, the redundancy technology has been used for designing computer system, especially the FBW flight control system in which the configuration with redundant hardware. Modern FBW flight control system required the use of the digital computers with HW / SW redundancy and the software for implementing control laws and logics. The software may fail in errors remaining in the software, so we should take account of the influence of software failures. For this reason, it is our goal to give out a method to analyze the combined HW / SW reliability for this kind of computer systems.

There are three jobs to be done: the first is Modeling or selection models; the second is selection algorithms, and the third is calculation of reliability performances. Obviously, Modeling is the key of the analysis method. Therefore, we give out two combined HW / SW reliability & availability models first.

II. DEVELOPMENT OF AVAILABLE MODELS

In the Introduction, we have discussed the necessity of the HW / SW reliability modeling. In the foreign countries, the study on HW / SW combined reliability models just started in the early of 1970s, and several combined models have been put foward uniting with engineering problems since then. In accordance with the opinions of M.Z.Shooman, the starting work of the study on the reliability & availability for systems with repair should be the study of Markov probability models. It is the widely accepted method to study HW / SW reliability model by ways of Markov stochastic process theory, and the corresponding models are called Markov models. The typical models of which are Landrault–Lapre model(1977), Goel–Okumoto model(1981), Angus–Jamus model(1982), and Shooman model (1988), etc.. The other scholars have tried to study the same problem employing other methods, examples for Thompson's Bayes model and the series model used to analyze the systems without repair.

III. PREREQUISITE OF MODELING

Because of the concrete characteristics of modern computer control systems, some problems have to be discussed before modeling.

1. Handling of configuration with redundant hardware

In this paper, we take equivalently treatment method to deal with the hardware subsystem. After doing that, the hardware subsytem become a single – component system equating to the redundant hardware. The typical treatment methods are; analytical methods; network analysis method; state transition chain method and fault tree analysis (FTA) method, etc..

2. Handling of the failure time distribution of the software

According to the failure data collected, we can apply various models to fit into the data and choose appropriate model; then, estimating the model parameters by using the data; finally, giving out functional relation of software failure rate

$$\lambda_s(t) = f_s(t)$$

3. Handling of the valid time for software testing

The valid operating time of the software is the sum of each time intervals. For hardware subsystem, assuming that the composing elements have the virtue of ″New as Good″ and the corrections are merely with replacement and without adjustment. The total hardware subsystem should have the characteristics of ″New as Good″ at the time points for failure and repair.

Being processed as above, the system can be treated as two HW / SW parts system.

Both the handling of configuration with redundant software and the interaction between hareware and software have not discussed in this paper.

Ⅳ.THEORETICAL ANALYTICAL MODELING BASED ON MARKOV RENEWAL PROCESSES

As a general rule, the failure and repair times for both the hardware and the software are random distribution, so we try to analyse the reliability of system with repair by means of the Markov renewal Processes.

We recognize that both the hardware and the software have the disposition of Markov at the points of time for the software failure and repair. Assuming that zero time should be the time of initial state, we may regard the points of time of the software failure as the regeneration points, then define X(t) as a stochastic process getting its values at the state set space E.

Assuming that the points of time of software failures are $t_1, t_2, \cdots, t_N$, initial number of software errors is N,then the renewal points are $0 = t_0 < t_1 < t_2 < \cdots < t_N$.

The set of states of Markov Process that the system may assume is subdivided into three subsets:

—State i corresponds to the system operating state with remaining software errors i;

—State ih corresponds to a repair state following a hardware failure with remaining software errors i;

—State is corresponds to a repair state following a software failure with remaining software errors i.

The graph representing the system state transition is given in Fig. 1.

Assuming $W_h(t)$, $G_h(t)$to be the time distributions of failure and repair of hardware, $G_s(t)$to be the time distribution of software repair. Moreover, assuming that time distributions of software failure at different renewal points are different, expressed as $W_{si}(t), i = 1,\cdots,N$.

Assuming $Q_{k,j}(t)$to be half–Markov Core, then:

$$\begin{cases} Q_{i,ih}(t) = \int_0^t [1 - W_{si}(t)] dW_h(u) & i = 1,\cdots,N \\ Q_{i,is}(t) = \int_0^t [1 - W_h(t)] dW_{si}(u) & i = 1,\cdots,N \\ Q_{ih,i}(t) = G_h(t) & i = 0,1,\cdots N \\ Q_{is,i-1}(t) = G_h(t) & i = 1,2,\cdots N \\ Q_{0,0h}(t) = W_h(t) \end{cases} \tag{1}$$

Given: $A_j(t) = P\{0(t) / X_0 = j\}$

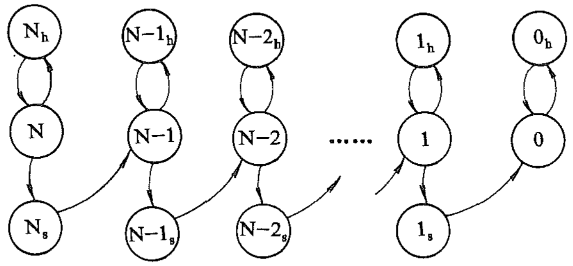

Figure 1. State transition diagram of renewal model

The renewal coupled equation of system availability is:

$$\begin{cases} A_j(t) = [1 - Q_{i,ih}(t) - Q_{i,is}(t)] \\ \quad + Q_{i,ih}(t) * A_{ih}(t) + Q_{i,is}(t) A_{is}(t) & i = 1,\cdots,N \\ A_{ih}(t) = Q_{ih,i}(t) * A_i(t) & i = 0,1,\cdots,N \\ A_{is}(t) = Q_{is,i-1}(t) * A_{i-1}(t) & i = 1,2,\cdots,N \end{cases} \tag{2}$$

The Laplace transform of the $A_i(t)$ is:

$$\begin{aligned}A_i^*(S) &= \frac{1-\hat{Q}_{i,ih}(S)-\hat{Q}_{i,is}(S)}{S[1-\hat{Q}_{i,ih}(S)\hat{Q}_{ih,i}(S)]} \\ &\quad + \frac{\hat{Q}_{i,is}(S)\hat{Q}_{is,i-1}(S)}{1-\hat{Q}_{i,ih}(S)\hat{Q}_{ih,i}(S)} A_{i-1}(S) \\ &= a_i(S)+b_i(S)A_{i-1}(S) \quad i=1,2,\cdots,N \quad (3)\end{aligned}$$

$\hat{Q}_{i,ih(S)}$, $\hat{Q}_{i,is(S)}$, $\hat{Q}_{is,i-1(S)}$ and $\hat{Q}_{ih,i(S)}$ can be coupled from equation (1) by means of Laplace–stijete transform.

On the assumption that the software doesn't fail after the time t_N, then:

$$A_0^*(S) = [1-\hat{Q}_{0,0h(S)}] / S[1-\hat{Q}_{0,0h(S)}\hat{Q}_{0h,0(S)}] \quad (4)$$

System availability $A_N(t)$ with initial state N can be deriven from the relation (5) and (6).

$$\begin{cases} A_N^*(S) = \sum_{k=1}^{N-1} a_k(S) \prod_{m=k+1}^{N} b_m(S) \\ \qquad + A_0^*(S)\prod_{j=1}^{N} b_j(S) + a_N(S) \qquad (5) \\ A_0^*(S) = [1-Q_{0,0h}(S)] / S[1-\hat{Q}_{0,0h}(S)\hat{Q}_{0h,0}(S)] \end{cases}$$

$$A_N(t) = \pounds^{-1}[A_N^*(S)] \quad (6)$$

In general, failure and repair are random distribution, so it may be very complicated to solve $A_N^*(S)$ or $A_N(t)$. In order to deal with actual engineering problems, we have put forward a numerical analytical model in the next section.

V.THE NUMERICAL ANALYTICAL MODELING BASED ON MARKOV PROCESSES

The numerical model is derived from the renewal model. With the continuous time distribution handled piece by piece, we can describe the system behavior employing the steady Markov Processes between two neighbouring times.

Having been handled piece by piece, we can get the software failure rate relations $\lambda_s(t_0)$, $\lambda_s(t_1)\cdots$, $\lambda_s(t_N)$ in the failure interval.

Assuming the hardware subsystem as a equivalent exponential distribution system, its equivalent failure rate can be gained from relation (7).

$$\int_0^\infty e^{-\lambda_H \cdot t} dt = \int_0^\infty e^{-\int_0^t \lambda_H(u)du} dt \quad (7)$$

In order to establish the model, the following assumptions are made:

1. The software initially possesses n'design errors;
2. The failures and repairs of both the hardware and software are independent;
3. The repairs of the hardware and the software are exponentially distributed with repair rates μ_H, $\mu_s$respectively;
4. The probability occurring with two or more hardware or software failures is negligible;
5. Both the hardware and the software can be repaired after its failure respectively, and no new errors can be introduced during the repairs;
6. Both the hardware failure and the software failure can lead to the system failure.

Based on the assumptions summarized as above, we define system states i, is, ih in the same way as we did in the last section. Fig. 2. shows its state transition relation.

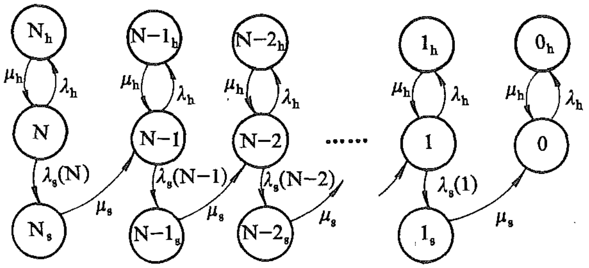

Figure 2. State transition diagram of numerical model

System one–step transition probabilities of Markov Processes are:

$$\begin{cases} P_{i,is}(\Delta t) = \lambda_s(i) \cdot \Delta t + 0(\Delta t), & i=1,2,\cdots,N \\ P_{i,ih}(\Delta t) = \lambda_H \cdot \Delta t + 0(\Delta t), & i=0,1,\cdots,N \\ P_{is,i-1}(\Delta t) = \mu_s \cdot \Delta t + 0(\Delta t), & i=1,2,\cdots,N \\ P_{ih,i}(\Delta t) = \mu_H \cdot \Delta t + 0(\Delta t), & i=0,1,\cdots,N \end{cases} \quad (8)$$

Define system state vector as P(t)

$$\begin{aligned}P(t) = (&P_N(t),P_{Nh}(t),P_{NS}(t),\cdots, \\ &P_1(t),P_{1h}(t),P_{1s}(t),P_0(t),P_{0h}(t)) \quad (9)\end{aligned}$$

The probability values of various states can be evaluated by the following differential coupled equation:

$$\dot{P}(t) = P(t) \cdot A$$
$$P(0) = (1,0,\cdots,0) \qquad (10)$$

The state transition rate matrix A can be derived from relation (9).

Calculating in computer, we can obtain system probabilities of any states at a series times, and evaluate the different system reliability performances in the end.

VI. ANALYTICAL EXAMPLE OF HW / SW RELIABILITY & AVAILABILITY FOR A FBW FLIGHT CONTROL SYSTEM

In this section, we just introduce the results associated with each analytical stage.

1. Calculation of the hardware subsystem failure rate of equivalent exponential distribution

For the complicated hardware subsystem, we have choden the method of the state transition chain with three– step end off, and obtained the failure rate ehange curve of the hardware subsystem shown in Figure 3.

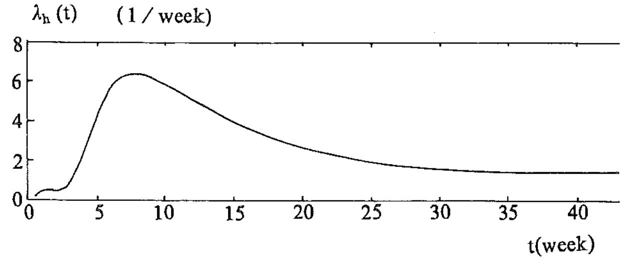

Figure 3. Failure rate change curve of the hardware subsystem

Then we have acquired the failure rate of hardware subsystem with equivalent exponential distribution λ_H as a result of the relation (7).

2. Calculation of software subsystem failure rate

Applying the numerical model, we have processed the software failure data collected and run our SRPP(Software Reliability Prediction Program) by computer. The SRPP have chosen the Goel–Okumoto model automatically and drawn out the failure rate change curve of the software subsystem shown in Figure 4.

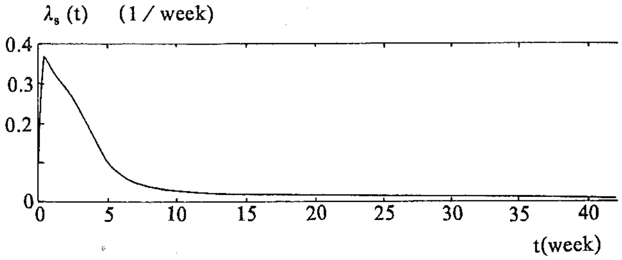

Figure 4. Failure rate change curve of the software subsystem

3. Calculation of system reliability & availability performance

Having processed as above, we have obtained the failure rate of equivalent hardware subsystem λ_H and the failure rates of the software subsystem at various renewal time points, $\lambda_s(i)$, $i = 1,\cdots,N$. The repair rate both of the hardware and the software subsystems are constant.

Having calculation the relating parameters, we have run our SHRAP (SW / HW Reliability Analysis Program) and calculated the each performances of system reliability and availability.

A. System availability

$$A(t) = \sum_{i=0}^{N} P_i(t)$$

Varification of A(t) is shown in Figure 5.

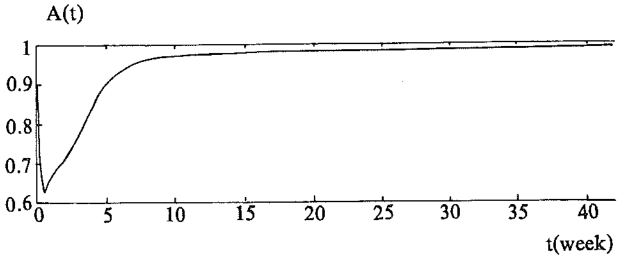

Figure 5. Varification diagram of system availability

B. System reliability

When the number of errors remaining in the software is i, the system reliability is determined by

$$R(t) = exp[-(\lambda_H + \lambda_S (i)).t]$$

C. The number of errors remaining in the software

$P_i(t)$ —the probability where system is in state j at time t

$$Q_j(t) = P_i(t) + P_{is}(t) + P_{ih}(t) , i = 1,\cdots,N$$

Taking the search method by computer, the number of errors m remaining in the software can be evaluated.

$$Q_i(t) = \underset{0 \leqslant i \leqslant N}{MAX}\{Q(t)\}$$

$$Q_0(t) = P_0(t) + P_{0h}(t)$$

The time distribution of errors remaining in the software during the software testing phase is shown in Figure 6.

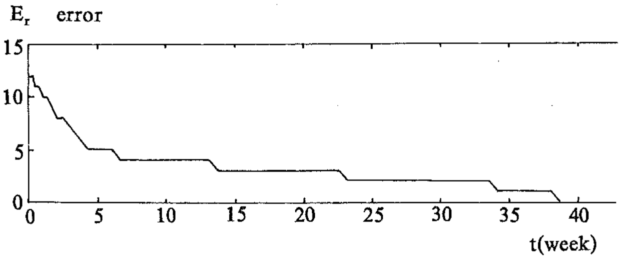

Figure 6. Time distribution of errors remaining in the software

D. The reliability performances to be released

The software testing didn't stop until all of the errors m remaining in the software had been removed. The system reliability and availability performances to be released : λ(T), MTTF, and R(t) can be acquired.

REFERENCE

1. Cao.J.H, Cheng.K, *Reliability Mathematics guide*, Scientific. Publishing House, 1986, Beijing
2. Yao.Yiping, Li Paiqiang, *Reliability and Redundancy*, Aero. Industry Press, 1991, 7 Beijing
3. Daniel.L.Palumbo, *The SIFT Hardware / Software System,*—Volume 1, A Detail Description, NASA—TM—875741, september, 1985
4. M.L.Shooman, *Research on combined Hardware / Software Reliability Model,* Polyteohnic No—84—003,004,005,006
5. E.R.Berg, W.G.Ness, *Digital Flight Control Software Validation Study,* ADA076021, Aune, 1988
6. John E.Angus, Larry E.James, *Combined Hardware / Software Reliability Model,* 1982 Proceeding Annual Reliability and Maintainability Symposium
7. Amrit Goel, Jopie Soenjoto, *Models for Hardware / software system operationaler Performance Evaluation,* 1981, Proceeding Annual Reliability and Maintainability Symposium
8. A.L.Goel, J.Soenjoto, *A cost operational study for Hardware / Software System,* Thchnical report, October, 1982