Global Journal of Modeling and Intelligent Computing (GJMIC)

Vol. 1, No. 1, January, 2021 **ISSN: 2767-1917**

AVAILABILITY ANALYSIS OF REPAIRABLE SOFTWARE AND HARDWARE SYSTEM WITH SPARES

Madhu Jain, I.I.T. Roorkee, India (madhu.jain@ma.iitr.ac.in)
Anuradha Jain, IBS, Dr Bhim Rao Ambedkar University, India (anujain_ibs@yahoo.com)
Ritu Gupta, Amity University, Uttar Pradesh, India (rgupta@amity.edu)

ABSTRACT

This study deals with the availability assessment of software and hardware based distributed system supported by repair facility of failed components. The system is made of total N units wherein (m, M)units are required to operate the system properly. To improve reliability/availability of the system, S spare components are provided to replace the components failed due to hardware/software malfunctioning. A Markov process is developed for the performance modeling and availability prediction. To develop mathematical model,we construct the differential equations of system state probabilities. Runge-Kutta technique (RKT) is employed to establish the transient queue size distribution and evaluate various reliability metrics of the system. To examine the effects of the availability and other performance measures by changing values of descriptors, numerical experiment is performed. We compare RKT based results with the neuro-fuzzy results.

Keywords: Software/hardware system, Availability, Markov process, Software reliability, m-out-of-(M+S) system, Adaptive Neuro-Fuzzy Inference system (ANFIS).

1. INTRODUCTION

With the fast growing and tremendous developments in computer technology, software embedded systems find wide applications in distributed environments viz. internet communication systems, safety critical equipment, satellite, data centers, etc.. Since these systems are highly sensitive with respect to its time-dependent properties; the system failure due to components failures cannot be tolerated at any stage of the system process. The software embedded system may lead to overall shut down due to one or more faults; however this issue can be tackled by optimal facility of maintainability and redundancy. The reliability and availability are the main key concerns for the system organizer of any software embedded system. In past literature, the reliability modelling for software/hardware based systems where failures are due to different causes have been proposed by several researchers (Sumita and Masuda 1986;Welkeet al.1995). The system availability is a major requisite for the distributed hardware and software systems. Lai et al. (2002) and Chevance (2005) presented the availability analysis to study the impacts of software and hardware failureson the testing and resource allocation of the system. In the same direction, another work was done by Gokhale et al. (2005) by classifying hardware/software failures in multiple failure-severities levels. An embedded computer system with multiple hardware components and one software component has been analyzed by Jain et al. (2010) by considering the human error and common cause failure besides software and hardware failures. This study has been extended by Jain and Gupta (2014) to discuss the availability analysis for large and complex redundant computer-based systems by considering all possible failure combinations when the system failure occurs due to mainly three types of failures namely software failure, hardware failure and failure due to human error. It is realized that the continuous operation and ageing properties of the system components reflect the system performance that ultimately reduces the efficiency of the system. To keep this issue in mind, Tokuno and Yamada (2010) developed reliability model for the software system to analyze the degradation observed by clients. They considered the system with two operational states i.e. normal mode and other one with degraded performance level. Kumar et al. (2013) developed a reliability model for a computer system having single server who follows inspection policy to perceive software and hardware repair activities. Kumar et al. (2015) extended their work by including the concepts of imperfect fault detection of the units. The system reliability of the system under the multiple failures environment has been discussed by Montoro-Cazorla and Perez-Ocon (2018). In this context, Sinha et al. (2019) have studied the combined hardware-software systems to analyze the reliability model by incorporating the functional failures.

Performance analysis of repairable systems in reliability contexts with spare/redundant has imperative role on the software embedded systems and has been extensively studied by many researchers since long. Stochastic analysis of repairable systems with multiple critical as well as non-critical faults has been suggested by Chung (1995). In the context of system testing, Tokuno and Yamada (2003)have studied a Markovian software availability model in an imperfect debugging environment. The notable study on the availability model has been done by Siddiqi and Weck (2007) who have contributed towards the estimation of the requirement of reconfigurable spare parts in space missions like moon and mars surface exploration. At the other side, Jiang and Xu (2007) and Grottke and Schleich (2013) developed the stochastic models to examine the effects of software faults and hardware faults with software ageing characterized by progressive performance degradation. Jain and Gupta (2011) presented a survey article on various failure factors and their consequences, along with software and hardware standby techniques that are helpful to cope up the severe failures. A priority model of the computer system for the reliability prediction was established by Sureriaet al. (2012). By introducing switching failure concept related to hardware and software faults with warm spare parts, Jain and Rani (2014) have analyzed system availability and reliability indices. The study on two states repairable computer based systems taking partial-failed (degraded) state and completely failed state was due to Kumar and Tyagi (2009), El-Damcese and Shama (2015). Gonçalves (2018) presented the procedure to evaluate the reliability of a graphics processing unit using software and hardware based fault tolerant techniques. Recently, Seo et al. (2020) have worked on reliability evaluation in same direction for safety critical software system.

Keeping above observations and facts into consideration, in the present investigation is to study the reliability model of software embedded multi-component distributed systems by including the both hardware (H/w) and software (S/w) faults. In previous developed models on the same line, the common assumption is that the system either failed by software or hardware components is not affected with the change in time which is quite unpractical phenomenon. To keep this concern in our mind, we develop a Markov model for the H/w and S/w based system that consists of some identical operating components along with spare parts. Furthermore to deal with more real time problems, the system is analyzed under normal mode and short mode. Moreover, we assume that the repair activities for both the modes are different. The outcomes of the investigation done are organized in different sections. In sections 2 and 3, the description of the model and the governing equations are presented, respectively. Some performance indices are given in section 4. To explore the computation tractability of performance indices, numerical results and sensitivity analysis are given in section 5. The final section 6 is devoted to concluding remarks.

2. MODEL DESCRIPTION

A Markovian model is investigated for the reliability prediction of the software embedded multi-component distributed system. The embedded system has M operating and S standby units operating in the distributed environment. At least m components out of M operative components are required for the functioning of the distributed system.

The notations used for modeling purpose are as follows:

λ_h (α_h)	Failure rate of operating (spare) components due to H/w faults.
$\lambda_w(t)$	Failure rate of operating components at time t due to S/w faults.
μ_h (μ_w)	Repair rate of failed components when failure occurs due to H/w (S/w) faults and system is operating in normal mode.
μ'_h (μ'_w)	Repair rate of components, when failure occurs due to H/w (S/w) faults and all the spare components are exhausted.
(i, j)	The representation of the states of number of failed components when component i suffers from H/w failures and the component j suffers from S/w failures, where i, j =1,2,…,N.
N	Counting of total components in the system.
M(S)	Counting of operating (standby) components in the system.
m	Counting of required components for the system functioning properly.
$P_{i,j}(t)$	Probability of the state (i, j) at time t, where i, j =1,2,…,N.

2.1 Assumptions

- **(i)** All the failures which occur either due to software or hardware failure of the components are mutually independent.
- **(ii)** Each operating component has the same failure rate function $\lambda_w(t)$ when failure occurs due to software faults and the failure rate of software standbys are assumed as zero here.
- **(iii)** The system fails when there are less than m components in the system.
- **(iv)** When all standbys are used up then the failure rate of operating components will be changed.
- **(v)** Once all standbys are used and a component fails, the system deteriorates but still operates.
- **(vi)** The lifetime and repair times of the components are governed by exponential distribution.

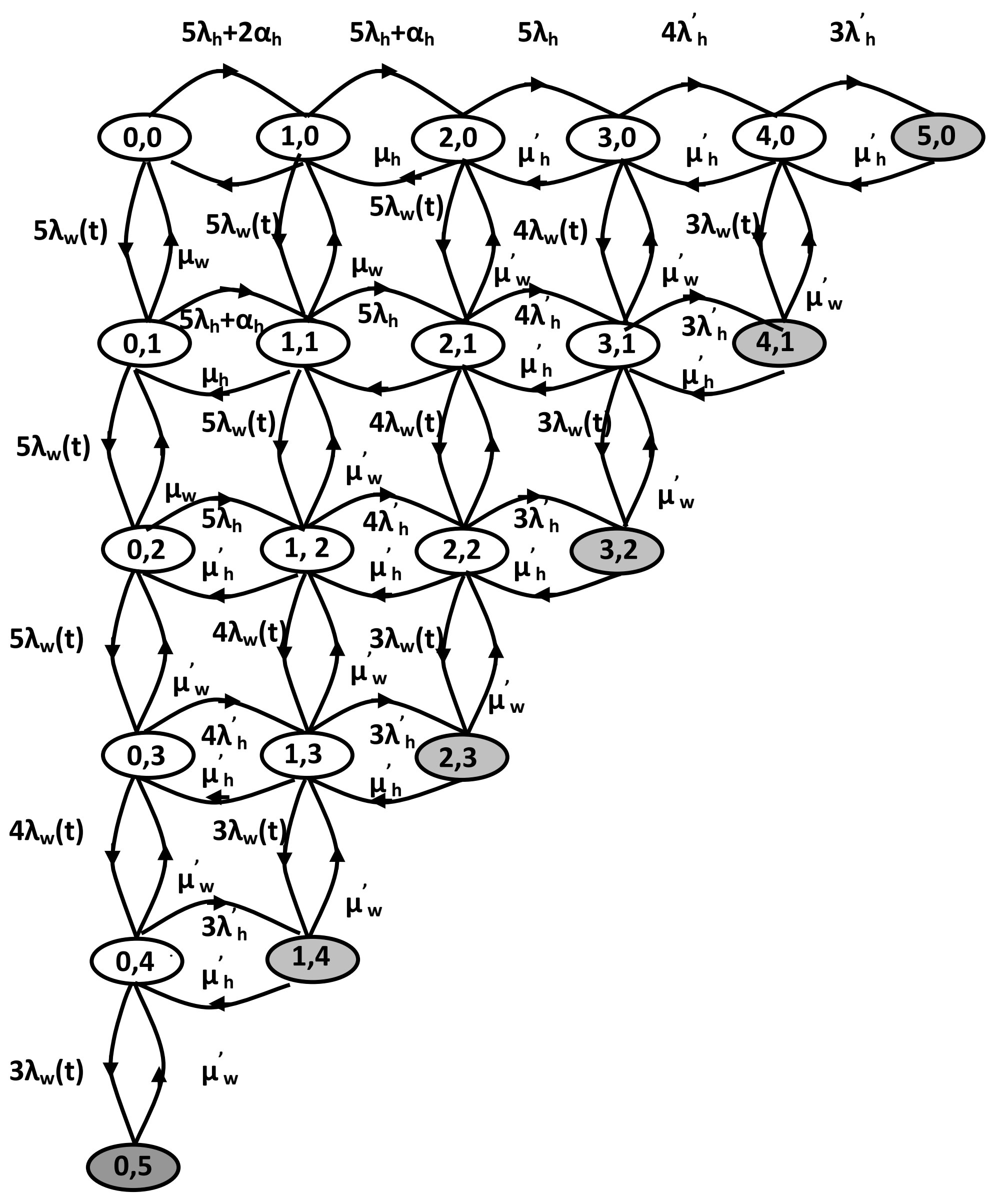


Figure 1: Markovian diagram of hardware and software system

The state dependent failure and repair rates are defined as follows:

Failure rates:

$$(i,j)\to(i+1,j);\ \Lambda_h(i,j)=\begin{cases}M\lambda_h+(S-i-j)\alpha_h; & k\le S,\ i+j=k\\ (N-i-j)\lambda_h'; & S<k<(N-m+1)\end{cases}$$

$$(i,j)\to(i,j+1);\ \Lambda_w(i,j)=\begin{cases}M\lambda_w(t); & k\le S;\\ (N-i-j)\lambda_w(t); & S<k<(N-m+1)\end{cases}$$

Repair rates:

$$\begin{aligned}&\mu_h\ \mathit{for}\,(i,j)\to(i-1,j); && k\le S\\ &\mu_h'\ \mathit{for}\,(i,j)\to(i-1,j); && k>S\\ &\mu_w\ \mathit{for}\ (i,j)\to(i,j-1); && k\le S\\ &\mu_w'\ \mathit{for}\ (i,j)\to(i,j-1); && k>S\end{aligned}$$

3. GOVERNING EQUATIONS

In this section, we use Markov process to analyze the hardware and software failure phenomenon of multi component distributed system. The state transition diagram is depicted in figure 1. For different system states, the equations governing the model are constructed by as follows:

$$\frac{d}{dt}P_{0,0}(t)=-\left[\{M\lambda_h+S\alpha_h\}+M\lambda_w(t)\right]P_{0,0}(t)+\mu_hP_{1,0}(t)+\mu_wP_{0,1}(t) \tag{1}$$

$$\begin{aligned}\frac{d}{dt}P_{i,0}(t)=&-\left[\{M\lambda_h+(S-i)\alpha_h\}+M\lambda_w(t)+\mu_h\right]P_{i,0}(t)+\{M\lambda_h+(S-i+1)\alpha_h\}P_{i-1,0}(t)\\&+\mu_hP_{i+1,0}(t)+\mu_wP_{i,1}(t),\ 1\le i\le S-1\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{d}{dt}P_{S,0}(t)=&-\left[\{M\lambda_h\}+\{M\lambda_w(t)\}+\mu_h\right]P_{S,0}(t)+\left(M\lambda_h+\alpha_h\right)P_{S-1,0}(t)\\&+\mu_h'P_{S+1,0}(t)+\mu_w'P_{S,1}(t)\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{d}{dt}P_{i,0}(t)=&-\left[\{(M+S-i)\lambda_h'\}+\{(N-i)\lambda_w(t)\}+\mu_h'\right]P_{i,0}(t)+\{(N-i+1)\lambda_h\}P_{i-1,0}(t)\\&+\mu_h'P_{i+1,0}(t)+\mu_w'P_{i,1}(t),\ (S+1)\le i\le(N-m)\end{aligned} \tag{4}$$

$$\frac{d}{dt}P_{i,0}(t)=-\mu_h'P_{i,0}(t)+(\{M+S-i+1\}\lambda_h')P_{i-1,0}(t),\ i=(M+S-m)+1 \tag{5}$$

$$\begin{aligned}\frac{d}{dt}P_{0,j}(t)=&-\left[\{M\lambda_h+(S-j)\alpha_h\}+\{M\lambda_w(t)\}+\mu_w\right]P_{0,j}(t)+\{M\lambda_w(t)\}P_{0,j-1}(t)\\&+\mu_wP_{0,j+1}(t)+\mu_hP_{1,j}(t),\ 1\le j\le S-1\end{aligned} \tag{6}$$

$$\frac{d}{dt}P_{0,S}(t) = -\left[M\lambda_h + M\lambda_w(t) + \mu_w\right]P_{0,S}(t) + \left\{M\lambda_w(t)\right\}P_{0,S-1}(t) + \mu'_h P_{1,j}(t) + \mu'_w P_{0,S+1}(t). \tag{7}$$

$$\frac{d}{dt}P_{0,j}(t) = -\left[\left\{(N-j)\lambda'_h\right\} + \left\{(N-j)\lambda_w(t)\right\} + \mu'_w\right]P_{0,j}(t) + \left\{(N-j)\lambda_w(t)\right\}P_{0,j-1}(t) + \mu'_h P_{1,j}(t) + \mu'_w P_{0,j+1}(t),\ (S+1) \le j \le (N-m-1). \tag{8}$$

$$\frac{d}{dt}P_{0,j}(t) = -\left[\left\{(N-j)\lambda'_h\right\} + \left\{(N-j)\lambda_w(t)\right\} + \mu'_w\right]P_{0,j}(t) + \left\{(N-j+1)\lambda_w(t)\right\}P_{0,j-1}(t) + \mu'_h P_{1,j}(t) + \mu'_w P_{0,j+1}(t),\quad j = (N-m). \tag{9}$$

$$\frac{d}{dt}P_{0,j}(t) = -[\mu'_w P_{0,j}(t)] + \left\{(M+S-j+1)\lambda_w(t)\right\}P_{0,j-1}(t),\quad j = (M+S-m)+1. \tag{10}$$

$$\frac{d}{dt}P_{i,j}(t) = -\left[\left\{(M\lambda_h + (S-i-j)\alpha_h\right\} + \left\{(M+S-i-j)\lambda_w(t)\right\} + \mu_h + \mu_w\right]P_{i,j}(t) + \left\{M\lambda_h + (S-i-j+1)\alpha_h\right\}P_{i-1,j}(t) + \left\{(M+S-i-j)\lambda_w(t)\right\}P_{i,j-1}(t) + \mu'_h P_{i+1,j}(t) + \mu'_w P_{i,j+1}(t),\ i,j \ne 0,\ 2 \le i+j \le S. \tag{11}$$

$$\frac{d}{dt}P_{i,j}(t) = -\left[\left\{(N-k)\lambda'_h\right\} + \left\{(N-k)\lambda_w(t)\right\} + \mu'_h + \mu'_w\right]P_{i,j}(t) + \left\{M\lambda_h + (S-k+1)\alpha_h\right\}P_{i-1,j}(t) + \left\{(N-k+1)\lambda_w(t)\right\}P_{i,j-1}(t) + \mu'_w P_{i,j+1}(t) + \mu'_h P_{i+1,j}(t),\ i,j \ne 0, (S+1) \le k \le (N-m-1). \tag{12}$$

$$\frac{d}{dt}P_{i,j}(t) = -\left[\left\{(N-k)\lambda'_h\right\} + \left\{(N-k)\lambda_w(t)\right\} + \mu'_h + \mu'_w\right]P_{i,j}(t) + \left\{N-k+1)\lambda_w(t)\right\}P_{i,j-1}(t) + \left\{(N-k+1)\lambda'_h\right\}P_{i-1,j}(t) + \mu'_w P_{i,j+1}(t) + \mu'_h P_{i+1,j}(t),\ i,j \ne 0,\ i+j \le (N-m). \tag{13}$$

$$\frac{d}{dt}P_{i,j}(t) = -\left[\mu'_h + \mu'_w\right]P_{i,j}(t) + \left\{(N-k+1)\lambda_w(t)\right\}P_{i,j-1}(t) + \left\{(N-k+1)\lambda'_h\right\}P_{i-1,j}(t),\ i,j \ne 0,\ k = N-m+1. \tag{14}$$

4. PERFORMANCE MEASURES

Once we obtain system's transient probabilities from the previous section then performance indices are formulated as follows:

1. The system availability at time t is

$$A_v(t) = \sum_{i+j=0}^{N-m} P_{i,j}(t)$$

2. Average counting of failed components because of H/w fault at time t is

$$E(N_h(t)) = \sum_{i=1}^{N-m} i \sum_{j=0}^{N-m-i} P_{i,j}(t)$$

3. Average counting of failed components because of S/w fault at time t

$$E(N_s(t)) = \sum_{j=1}^{N-m} j \sum_{i=0}^{N-m-j} P_{i,j}(t)$$

4. Average counting of standby components at time t

$$E(N_{sb}(t)) = \sum_{i+j=0}^{S} (S-i-j)P_{i,j}(t)$$

5. Failure frequency of the system is

$$\omega_f(t) = m(\lambda_h' + \lambda_w(t)) \sum_{\substack{i+j \in (N-m+1) \\ i\neq 0, j\neq 0}} P_{i,j}(t) + m[\lambda_h' P_{N-m+1,0}(t) + \lambda_w(t) P_{0,N-m+1}(t)]$$

4.1 Illustration

For illustration purpose we present a Markov model (see figure 1) for the hardware and software based distributed system. The system has the maximum number of software embedded component in operation as five (i.e. M=5). Number of standbys components which are used to replace the failed components are two (i.e. S=2). We consider that when at least three components (i.e. m=3) are in working stage then the system will work successfully otherwise system will fail. For repairing the failed component there is provision of one repairman in the system. The differential difference equations related to this particular case are as follows.

$$\frac{d}{dt}P_{0,0}(t) = \mu_h P_{1,0}(t) + \mu_w P_{0,1}(t) - (5\lambda_h + 2\alpha_h + 5\lambda_w(t))P_{0,0}(t) \tag{15}$$

$$\frac{d}{dt}P_{1,0}(t) = (5\lambda_h + 2\alpha_h)P_{0,0}(t) + \mu_h P_{2,0}(t) - (5\lambda_h + \alpha_h + \mu_h + 5\lambda_w(t))P_{1,0}(t) + \mu_w P_{1,1}(t) \tag{16}$$

$$\frac{d}{dt}P_{2,0}(t) = (5\lambda_h + \alpha_h)P_{1,0}(t) + \mu_h' P_{3,0}(t) + \mu_w' P_{2,1}(t) - (\mu_h + 5\lambda_h + 5\lambda_w(t))P_{2,0}(t) \tag{17}$$

$$\frac{d}{dt}P_{3,0}(t) = 5\lambda_h P_{2,0}(t) + \mu_h' P_{4,0}(t) + \mu_w' P_{3,1}(t) - (\mu_h' + 4\lambda_h' + 4\lambda_w(t))P_{3,0}(t) \tag{18}$$

$$\frac{d}{dt}P_{4,0}(t) = 4\lambda_h' P_{3,0}(t) + \mu_h' P_{5,0}(t) + \mu_w' P_{4,1}(t) - (\mu_h' + 3\lambda_h' + 3\lambda_w(t))P_{4,0}(t) \tag{19}$$

$$\frac{d}{dt}P_{5,0}(t) = 3\lambda_h' P_{4,0}(t) - \mu_h' P_{5,0}(t) \tag{20}$$

$$\frac{d}{dt}P_{0,1}(t)=5\lambda_w(t)P_{0,0}(t)+\mu_w P_{0,2}(t)+\mu_h P_{1,1}(t)-(\mu_w+5\lambda_h+\alpha_h+5\lambda_w(t))P_{0,1} \tag{21}$$

$$\begin{aligned}\frac{d}{dt}P_{1,1}(t)=&(5\lambda_h+\alpha_h)P_{0,1}(t)+\mu'_h P_{2,1}(t)+5\lambda_w(t)P_{1,0}(t)+\mu'_w P_{1,2}(t)\\&-(\mu_w+5\lambda_h+\mu_h+5\lambda_w(t))P_{1,1}(t)\end{aligned} \tag{22}$$

$$\begin{aligned}\frac{d}{dt}P_{2,1}(t)=&\mu'_h P_{3,1}(t)+5\lambda_h P_{1,1}(t)+5\lambda_w(t)P_{2,0}(t)+\mu'_w P_{2,2}(t)-(\mu'_w+4\lambda'_h+\mu'_h\\&+4\lambda_w(t))P_{2,1}(t)\end{aligned} \tag{23}$$

$$\begin{aligned}\frac{d}{dt}P_{3,1}(t)=&4\lambda'_h P_{2,1}(t)+\mu'_h P_{4,1}(t)+4\lambda_w(t)P_{3,0}(t)+\mu'_w P_{3,2}(t)-(\mu'_w+3\lambda'_h\\&+\mu'_h+3\lambda_w(t))P_{3,1}(t)\end{aligned} \tag{24}$$

$$\frac{d}{dt}P_{4,1}(t)=3\lambda'_h P_{3,1}(t)+3\lambda_w(t)P_{4,0}(t)-(\mu'_w+\mu'_h)P_{4,1}(t) \tag{25}$$

$$\frac{d}{dt}P_{0,2}(t)=5\lambda_w(t)P_{0,1}(t)+\mu'_w P_{0,3}(t)+\mu'_h P_{1,2}(t)-(\mu_w+5\lambda_h+5\lambda_w(t))P_{0,2}(t) \tag{26}$$

(27)

$$\frac{d}{dt}P_{2,2}(t)=\mu'_h P_{3,2}(t)+4\lambda'_h P_{1,2}(t)+4\lambda_w(t)P_{2,1}(t)+\mu'_w P_{2,3}(t)-(\mu'_w+3\lambda'_h+\mu'_h+3\lambda_w(t))P_{2,2}(t) \tag{28}$$

$$\frac{d}{dt}P_{3,2}(t)=3\lambda'_h P_{2,2}(t)+3\lambda_w(t)P_{3,1}(t)-(\mu'_w+\mu'_h)P_{3,2}(t) \tag{29}$$

$$\frac{d}{dt}P_{0,3}(t)=\mu'_h P_{1,3}(t)+5\lambda_w(t)P_{0,2}(t)+\mu'_w P_{0,4}(t)-(\mu'_w+4\lambda'_h+4\lambda_w(t))P_{0,3}(t) \tag{30}$$

$$\begin{aligned}\frac{d}{dt}P_{1,3}(t)=&4\lambda_w(t)P_{1,2}(t)+\mu'_w P_{1,4}(t)+\mu'_h P_{2,3}(t)+4\lambda'_h P_{0,3}(t)\\&-(\mu'_w+\mu'_h+3\lambda'_h+3\lambda_w(t))P_{1,3}(t)\end{aligned} \tag{31}$$

$$\frac{d}{dt}P_{2,3}(t)=3\lambda'_h P_{1,3}(t)+3\lambda_w(t)P_{2,2}(t)-(\mu'_w+\mu'_h)P_{2,3}(t) \tag{32}$$

$$\frac{d}{dt}P_{0,4}(t)=4\lambda_w(t)P_{0,3}(t)+\mu'_w P_{0,5}(t)+\mu'_h P_{1,4}(t)-(\mu'_w+3\lambda'_h+3\lambda_w(t))P_{0,4}(t) \tag{33}$$

$$\frac{d}{dt}P_{1,4}(t)=3\lambda'_h P_{0,4}(t)+3\lambda_w(t)P_{1,3}(t)-(\mu'_w+\mu'_h)P_{1,4}(t) \tag{34}$$

$$\frac{d}{dt}P_{0,5}(t)=3\lambda_w(t)P_{0,4}(t)-\mu'_w P_{0,5}(t) \tag{35}$$

The system availability is obtained by

$$A_v=\sum_{i=0}^{4}p_{i,0}(t)+\sum_{j=1}^{4}p_{0,j}(t)+\sum_{i=1}^{3}p_{i,1}(t)+\sum_{i=1}^{2}p_{i,2}(t)+p_{1,3}(t) \tag{36}$$

5. NUMERICAL RESULTS

The computations for the system state probabilities are done by using Runge-Kutta IVth order method. It is implemented by using the ode-45 function in software MATLAB. For illustration purpose, we consider 5 non-identical components of the software embedded system. There should be at least 3 components out of 5 components are required for the system working properly, otherwise system will fail. There is provision of two standby components which replace the failed components.

For the numerical experiment, the parameters are chosen as$\lambda_h = 0.1$, $\lambda_h^{'} = 0.8\,\lambda_h$, $\mu_h = 2$, $\mu_h^{'} = 1.2\,\mu_h$, $\lambda_w = 0.05$, $\mu_w = 2$, $\mu_w^{'} = 1.2\,\mu_w$,$\alpha_h = 0.13$and$\beta = 2$. Failure rate of the software component follows Weibull distribution and can be taken as $\lambda_w(t) = \lambda_w \beta t^{\beta-1}$ whereβ is the shape parameter.

Tables 1-4 reveal the impacts of parameters (λ_h, λ_w) and (μ_h, μ_w) on the system availability (A_v) by varying time t. From the tables 1-2 it is cleared that A_v lowers down as time passes which also depicts the real life phenomenon. We see the decreasing trend in the availability as we increase the failure rate of hardware/software components. Tables 3-4 show the impact of increments in μ_h and μ_w on A_v. It is clear that the A_v enhances by improving the repair rates (μ_h, μ_w).

Figures 2-5 show the effect of repair rates (μ_h, μ_w), shape parameter (β) and failure rates (α_h) on the availability by varying time t. Figure 2(b)-5(b) show the fuzzy membership function for the input parameter. In figures 2-5, the analytical results are shown by continuous lines and ANFIS results are shown by discrete lines.

In figs 2(a)-3(a), availability enhances as we increase the value of repair rates. Fig. 2(a)(fig. 3(a)) depicts that availability increases by speed up the repair of the hardware (software) component, but decreases constantly (steeply) for lower values of $\mu_h(\mu_w)$ and further becomes constant for higher values of $\mu_h(\mu_w)$. In figure 4(a), availability decreases for larger value of βbut remains almost constant for smaller values of β. The corresponding membership function is drawn in fig. 4(b). In fig. 5(a) there is no significant effect of failure rate of spare components (α_h) on A_v. In figures 2(b)-5(b), ANFIS provides closer values to analytical results.

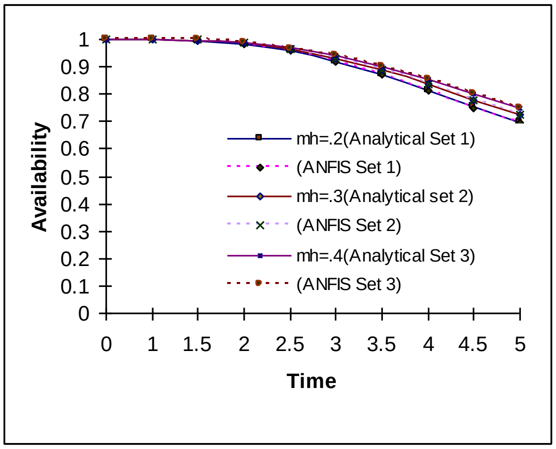
Table 1: System availability for different values of λ_h

Time(t)	Availability(A_v)			
	λ_h=0.2	λ_h=0.4	λ_h=0.6	λ_h=0.8
0.0	1	1	1	1
1.0	0.999474	0.993066	0.971791	0.930894
1.5	0.998563	0.983474	0.940834	0.871467
2.0	0.997565	0.974411	0.915927	0.831292
2.5	0.996702	0.967389	0.898972	0.807738
3.0	0.99603	0.962387	0.88816	0.794613
3.5	0.995526	0.958944	0.881432	0.787413
4.0	0.995147	0.956592	0.877262	0.783458
4.5	0.994853	0.954972	0.874656	0.781254
5.0	0.994613	0.953829	0.872992	0.779988

Table 2: System availability for different values of λ_w				
Time(t)	Availability(A_v)			
	λ_w=0.1	λ_w=0.2	λ_w=0.3	λ_w=0.4
0.0	1	1	1	1
1.0	0.999578	0.997641	0.992147	0.980939
1.5	0.997401	0.981169	0.937983	0.865796
2.0	0.990671	0.929277	0.802384	0.650563
2.5	0.97510	0.832938	0.627224	0.45901
3.0	0.946496	0.710816	0.481383	0.345911
3.5	0.903184	0.593389	0.38613	0.28372
4.0	0.847246	0.499495	0.326751	0.244385
4.5	0.783773	0.431083	0.286606	0.21605
5.0	0.718836	0.381727	0.256771	0.194143

Table 3: System availability for different values of μ_h				
Time(t)	Availability(A_v)			
	μ_h=0.5	μ_h=0.6	μ_h=0.7	μ_h=0.8
0.0	1	1	1	1
1.0	0.999514	0.999549	0.999582	0.999612
1.5	0.993747	0.994289	0.994778	0.995218
2.0	0.962089	0.965173	0.967917	0.970359
2.5	0.866562	0.874806	0.882064	0.888452
3.0	0.696338	0.707823	0.717724	0.726238
3.5	0.509543	0.5186	0.525973	0.531911
4.0	0.370749	0.375368	0.378564	0.380621
4.5	0.283996	0.285773	0.286465	0.286368
5.0	0.226914	0.227306	0.226922	0.226044

Table 4: System availability for different values of μ_w				
Time(t)	Availability(A_v)			
	μ_w=2	μ_w=4	μ_w=6	μ_w=8
0.0	1	1	1	1
1.0	0.999524	0.999537	0.999549	0.999561
1.5	0.992464	0.992813	0.99314	0.993447
2.0	0.942799	0.946155	0.949264	0.952146
2.5	0.766481	0.780349	0.793236	0.805218
3.0	0.468084	0.493752	0.518103	0.541195
3.5	0.231605	0.25653	0.280859	0.304576
4.0	0.128289	0.146657	0.164864	0.182898
4.5	0.086826	0.100712	0.114507	0.128212
5.0	0.064397	0.075457	0.086457	0.097395



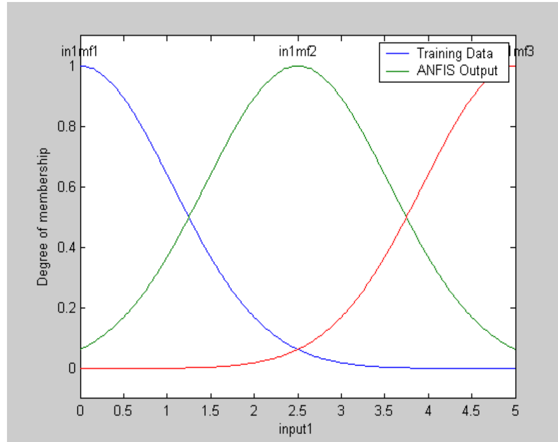
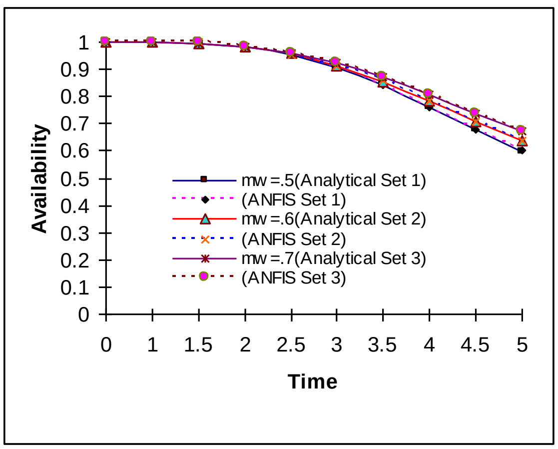


Figure 2(a): Trends of System Availability vs. μ_h **Figure 2(b):** Membership function for figure 2(a)



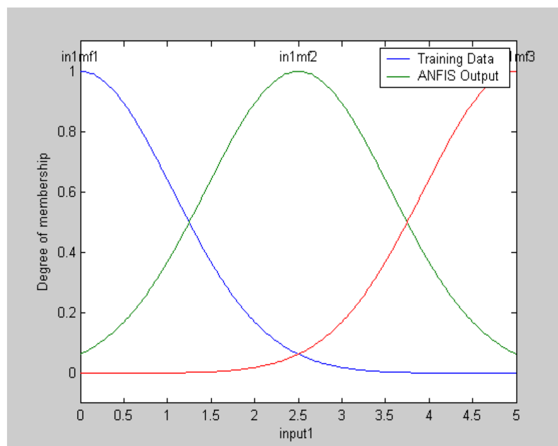
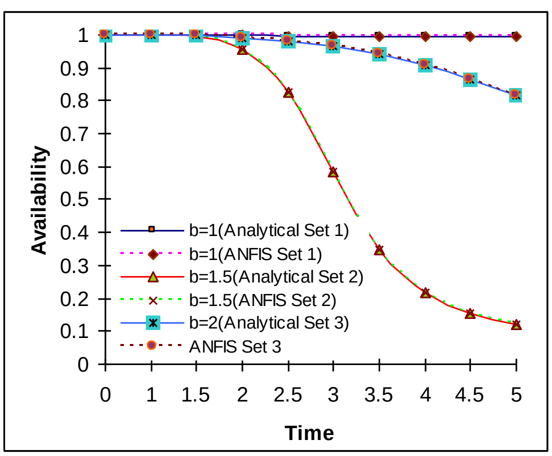


Figure 3(a): Trends of System Availability vs. μ_w **Figure 3(b):** Membership function for figure 3(a)



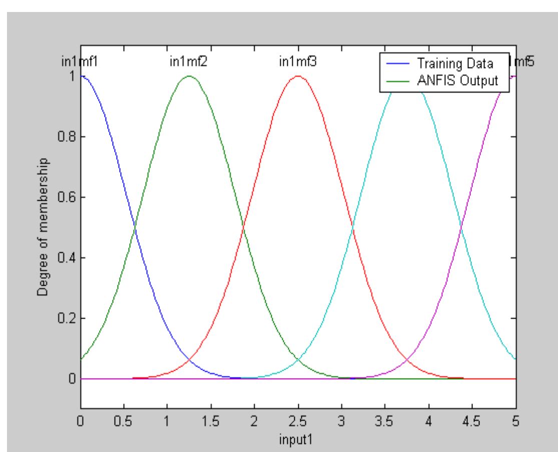
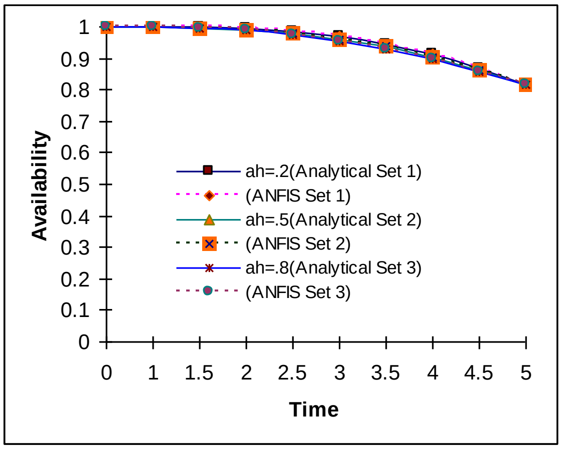


Figure 4(a): Trends of System Availability vs. β **Figure 4(b):** Membership function for figure 4(a)



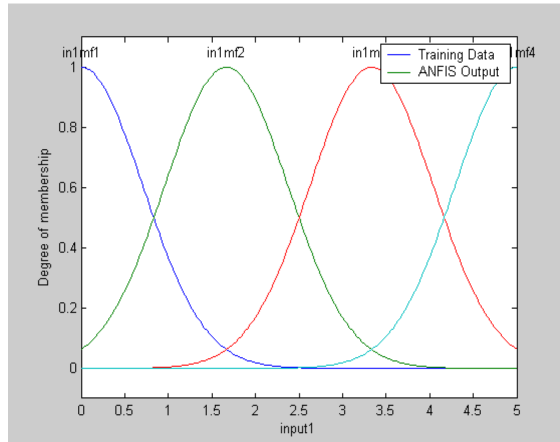


Figure 5(a): Trends of System Availability vs. α_h **Figure 5(b):** Membership function for figure 5(a)

6. CONCLUSIONS

In this paper, we have considered repairable hardware and software distributed system with standby provisioning. The suggested approach will be helpful for the system designers to improve the reliability of the embedded system involving hardware or software components. Our proposed stochastic model gives an insight for the understanding the factors which can be controlled during the development phase of software and hardware based system. The model enables new vision to improve the distributed system performance and the indices prediction may fulfill the requirements of many organization's mission in terms of efficiency, effectiveness, quality, safety and timelineness. The numerical results provided demonstrate the computational tractability as well as suggest how the system availability can better be achieved with the help of standbys and repairs. The results based on neuro-fuzzy model could also be used successfully for identifying the behavior of various parameters involved in the concerned systems which match profoundly with the many applications of real world.

REFERENCES

Chevance, R.J. (2005). Hardware and software solutions for high availability. *Server Architectures*, 609-652.

Chung, W.K. (1995). Stochastic analysis of k-out-of-n:G redundant systems with repair and multiple critical and non critical errors. *Microelectronics Reliability*, 35, 1429-1431.

El-Damcese, M.A., & Sharma, M.S. (2015). Reliability and availability analysis of a 2-state repairable system with two types of failure. *Engineering Mathematics Letters*, 2, 1-9.

Gokhale, S.S., Crigler, J.R., Farr, W.H.,& Wallace, D.R. (2005).System availability analysis considering hardware/software failure severities.*Software Engineering Workshop*, *29th Annual IEEE/NASA*, 7, 47 – 56.

Gonçalves, M., Saquetti, M., &Azambuja, R. (2018). Evaluating the reliability of a GPU pipeline to SEU and the impacts of software-based and hardware-based fault tolerance techniques, *Microelectronics Reliability*, 88-90, 931-935.

Grottke, M., &Schleich, B. (2013). How does testing effects the availability of aging software systems.*Performance Evaluation*, 70 (3), 179-196.

Jain, M., & Gupta, R. (2011). Redundancy issues in software and hardware systems: an overview. *International Journal of Reliability, Quality and Safety Engineering*. 18(1), 61–98.

Jain, M.,& Gupta, R. (2014).Availability analysis of repairable redundant system with three types of failures subject to common cause failure. *International Journal of Mathematics in Operational Research*, 6, 271-296.

Jain, M., & Rani, S. (2014). Transient analysis of hardware and software systems with warm standbys and switching failures. *International Journal of Operational Research*. 6(1), 1-28.

Jain, M., Agrawal, S.C., & Rani, S. (2010). Availability analysis of embedded computer system with two types of failure and common cause failure. *Journal of International Academy of Physical Sciences*, 14(4), 439-448.

Jiang, L., & Xu, G. (2007). Modeling and analysis of software aging and software failure. *Journal of Systems and Software*, 80(4), 590-595.

Kumar, A., Saini, M., & Malik, S.C. (2015). Performance analysis of a computer system with imperfect fault detection of hardware. *Procedia Computer Science*, 45, 602-610.

Kumar, S., & Tyagi, A. (2009). Evaluation of some reliability parameters of a three state repairable system with environmental failure. *International Journal of Research and Reviews in Applied Sciences*, 1(2), 96-103.

Kumar, V., Kumar, R.,& Sharma, A. (2013). Applying neuro-fuzzy approach to build the reusability assessment framework across software component releases-an empirical evaluation. *International Journal of Computer Application*, 70, 41-47.

Lai, C.D., Xie, M., Poh, K.L., Dai, Y.S.,& Yang, P. (2002). A model for availability analysis of distributed software/hardware systems. *Information and Software Technology*, 44, 343-350.

Montoro-Cazorla, D., & Pérez-Ocón, R. (2018). Constructing a Markov process for modelling a reliability system under multiple failures and replacements. *Reliability Engineering & System Safety*, 173, 34-47.

Seo, J., Kang, H.G., Lee, E.C., & Lee, S.J. (2020). Experimental approach to evaluate software reliability in hardware-software integrated environment.*Nuclear Engineering and Technology*, 52(7),1462-1470.

Siddiqi,A., &Weck, O.L.D. (2007). Spare parts requirements for space missions with reconfigurability and commonality. *Journal of Spacecraft and Rockets*, 44(1), 147-155.

Sinha, S., Goyal, N.K., & Mall, R. (2019). Early prediction of reliability and availability of combined hardware-software systems based on functional failures.*Journal of Systems Architecture*, 92, 23-38.

Sumita, U., &Masuda, Y. (1986). Analysis of software availability/reliability under the influence of hardware failures. *IEEE Transactions on Software Engineering*, 12, 32-41.

Sureria, J.K., Malik, S.C., & Anand, J. (2012). Cost-benefit analysis of a computer system with priority to s/w replacement over h/w repair. *Applied Mathematical Sciences*, 6(75), 3723-3734.

Tokuno, K., & Yamada, S. (2003). Relationship between software availability measurement and the number of restorations with imperfect debugging. *Computers & Mathematics with Applications*, 46, 1155-1163.

Tokuno, K.,& Yamada, S. (2010). Availability-based software performability model with user-perceived performance degradation. *International Journal of Software Engineering & Applications*, 4, 1-14.

Welke, S.R., Johnson, B.W., &Aylor, J.H. (1995). Reliability modeling of hardware/software systems. *IEEE Transactions on Reliability*, 44, 413-418.