

Performance and Cost Benefit Analysis of a Hardware-Software System Considering Hardware based Software Interaction Failures and Different Types of Recovery

Rajeev Kumar

Department of Mathematics,
M.D. University, Rohtak-124001, INDIA

Monu Kumar

Department of Mathematics,
M.D. University, Rohtak-124001, INDIA

ABSTRACT

In the present paper a hardware-software system that may have hardware, software and hardware based software interaction failures is taken. It is considered that a hardware and a software failure occurs purely due to respective hardware and software subsystems whereas a hardware based software interaction failure occurs whenever hardware degradation is not detected or repaired by the software method. It is assumed that, in addition to the software methods, recovery of the hardware and the software subsystems upon their failure may respectively be carried out by external hardware and software engineers. The measure of system performance such as mean time to system failure, expected up time of the system, expected down time due to pure hardware failure, expected down time due to pure software failure, expected down time due to hardware based software interactions failure, expected number of software repairs, expected number of hardware repairs, expected number of visits of hardware engineers, expected number of visits of software engineers, expected number of hardware repair by software method and expected number of software repair by re-execution are obtained using Markov process and regenerative point technique. Various conclusions regarding reliability and profit of the system are drawn on the basis of graphical studies.

General Terms

Stochastic Process

Keywords

Hardware-software system, Hardware based software interaction, MTSF, expected up time, expected down time, profit, Markov process, regenerative point technique

1. INTRODUCTION

Almost all the modern systems like computers, mobiles, robots, missiles, rockets, radars etc. are nothing but hardware-software systems. A few of the numerous applications of these systems are control of communication and transport systems, automated plants operation, space explorations and even in routine activities of today society e.g., reservation of tickets, computation of various bills, in banks and in business, etc. All these things demand high reliability since failure of any can be costly and hazardous. As a hardware-software system means a system consists of two major components- hardware and software and therefore for effective performance of the system, both its hardware and software sub-components must function with considerable reliability. Even though these systems consists of hardware and software

subsystems, to judge reliability of the system, the researchers in the past dealt separately with hardware reliability and software reliability. For the reliability analyses of the whole system, combined reliability models, i.e. including both hardware and software subsystems were discussed by a few researchers such as Friedman and Tran [1], Hecht and Hecht [2], Kumar and Malik [3], Welke et al.[4], etc. assuming in general that these subsystem are independent of each other. That is, the aspects of interactions between the hardware and software subsystems were not taken up by them. However, Boyd et al.[5] discussed the difficulties in modeling hardware-software interactions.

Some researchers in the past including Iyer and Velardi [6], Martin and Mathur [7], Kanoun and Ortalo-Borrel [8], Haung et al.[9], justified that there exists remarkable interactions between hardware and software components. Therefore while developing combined reliability models the interactions between hardware and software components should not be ignored. Keeping this in view, Teng et al. [10] established a reliability modeling of the combined computer system by considering hardware, software and hardware based software interactions failures. Here the reliability of the combined system has been obtained by considering different models for hardware, software and hardware based software failure and not in fact in integrated way. Moreover various recovery aspects on different failures for the overall system have not been considered. To fill up this gap, recently a model for the whole computer system, i.e. including hardware, software and hardware based software interactions failures taking in to account different recovery aspects for the overall system has been developed by Kumar and Kumar [11] and discussed its reliability and availability.

The present paper is an extension of the above work for hardware-software systems having hardware, software and hardware based software failures with different types of recovery methods. Besides this some other measures of system performance are obtained and cost-benefit analysis of the system is also carried out. In this paper a combined model for a hardware-software system that may have hardware, software or hardware based software interaction failures by taking different recovery aspects, as in Kumar and Kumar [11], is considered. In the model it is considered that hardware and software failures occurs purely due to respective hardware and software subsystems whereas hardware based software interaction failures occurs whenever hardware degradation is not detected or repaired by the software method. It is assumed that, in addition to the software methods, recovery of the hardware and the software subsystems upon their failure may respectively be carried out by external hardware and software engineers.

In other words, it is considered that the system from its initial normal mode may go either to total hardware failures mode or to partial hardware (hardware degradation) mode or to total software failure mode. The repair at total hardware failure mode is carried out by the hardware engineer and the system reaches its normal mode whereas at total software failure mode repair is done by the software engineer. It is also assumed that from partial hardware mode system may go to software fail safe mode or to unsafe mode due to hardware-software interaction. The recovery from the fail safe to normal mode is possible by re-execution of the software whereas from fail unsafe mode to partial hardware failure mode through re-installation of the software by the software engineer. Other assumptions are:

1. The partial hardware failure may or may not be immediately detected. Once a partial hardware failure is detected, it can be recovered using software tool if not recovered using software method then hardware engineer is called to repair it.
2. An undetected degradation may cause a hardware related software failure (fail unsafe) mode and a detected degradation may cause an execution abortion (fail safe) mode.
3. On repair of the software in the safe mode the system recovers completely whereas on repair in the fail unsafe mode due to undetected hardware failure, the system is degraded.
4. All the failure and repair time distributions are exponential.
5. The external service facilities, i.e. hardware and software engineers, reaches the system in negligible time.
6. Switching is perfect and instantaneous.
7. All random variables are independent of each other.

The measures of system performance such as mean time to system failure (MTSF), expected up time, expected down time due to pure hardware failure, expected down time due to pure software failure, expected down time due to hardware based software interactions failure, expected number of software repairs, expected number of hardware repairs, expected number of visits of hardware engineers, expected number of visits of software engineers, expected number of hardware repair by software method and expected number of software repair by re-execution and profit are computed by making use of semi-Markov Process and regenerative point techniques. Various conclusions regarding reliability and profit of the system are drawn on the basis of graphical studies.

2. NOTATIONS

- λ_{h1} : Hardware failure rate from normal mode to degradation partially failure mode.
 λ_{h2} : Hardware failure rate from partial hardware failure mode to total hardware failure mode.
 λ_{h3} : Hardware failure rate from normal mode to total hardware failure mode.
 λ_{s1} : Software failure rate from normal mode to total software failure mode.
 λ_{s2} : Software failure rate from undetected hardware partial failure mode to fail unsafe mode.
 λ_{s3} : Software failure rate from detected hardware partial failure mode to fail safe abortion mode.
 p_1 : Probability that hardware degradation is detected.

- q_1 : Probability that hardware degradation is not detected.
 p_2 : Probability that hardware degradation is recovered by software methods.
 q_2 : Probability that hardware degradation is not recovered by software methods.
 β_1 : Hardware repair rate when degradation is detected and recovered by software methods.
 β_2 : Hardware repair rate when degradation is detected and repaired by hardware engineer.
 β_3 : Hardware repair rate of total hardware-software mode by the hardware engineer.
 γ_1 : Software repair rate on total software failure by the Software engineer.
 γ_2 : Software repair rate by the software engineer on fail unsafe mode.
 γ_3 : Software repair rate by the software engineer on fail safe mode.

3. STATES OF THE SYSTEM

HS	Normal mode.
$\hat{H}S$	Partial (degraded) hardware mode.
$\bar{H}S$	Total (complete) hardware failed mode.
$H\bar{S}$	Total (complete) software failed mode.
$\hat{H}\bar{S}$	Fail unsafe mode.
$\hat{H}\hat{S}$	Fail safe mode.

The transition diagram showing the various states of the system is shown in the Fig.1. The epochs of entry into all the states are regenerative points and thus every state is regenerative state. The states 4, 5, 6 and 7 are the failed states.

4. MEASURES OF THE SYSTEM PERFORMANCE

4.1 Transition Probabilities and Mean Sojourn Time

The transition probabilities p_{ij} are given by:

$$\begin{aligned}
 p_{01} &= p_1 p_2 \lambda_{h1} / S & p_{02} &= q_1 \lambda_{h1} / S \\
 p_{03} &= p_1 q_2 \lambda_{h1} / S & p_{04} &= \lambda_{h3} / S \\
 p_{07} &= \lambda_{s1} / S & p_{10} &= \beta_1 / S_1 \\
 p_{14} &= \lambda_{h2} / S_1 & p_{24} &= \lambda_{h2} / S_2 \\
 p_{25} &= \lambda_{s2} / S_2 & p_{30} &= \beta_2 / S_3 \\
 p_{34} &= \lambda_{h2} / S_3 & p_{36} &= \lambda_{s3} / S_3 \\
 p_{40} &= p_{52} = p_{60} = p_{70} = 1
 \end{aligned}$$

Clearly, it can be verified that

$$\begin{aligned}
 p_{01} + p_{02} + p_{03} + p_{04} + p_{07} &= 1 & p_{10} + p_{14} &= 1 \\
 p_{24} + p_{25} &= 1 & p_{30} + p_{34} + p_{36} &= 1
 \end{aligned}$$

where

$$\begin{aligned}
 S &= \lambda_{h1} + \lambda_{s1} + \lambda_{h3}, \quad S_1 = \beta_1 + \lambda_{h2}, \quad S_2 = \lambda_{h2} + \lambda_{s2} \quad \text{and} \\
 S_3 &= \beta_2 + \lambda_{s3} + \lambda_{h2}
 \end{aligned}$$

Mean sojourn time μ_i in the i th state is the expected first passage time taken by the i th state before transiting to any other state.

$$\mu_i = \int_0^{\infty} \Pr(T_i > t) dt$$

where T_i is the p.d.f. of device life time.

$$\begin{aligned} \mu_0 &= 1/S, & \mu_1 &= 1/S_1, & \mu_2 &= 1/S_2, & \mu_3 &= 1/S_3 \\ \mu_4 &= \frac{1}{\beta_3}, & \mu_5 &= \frac{1}{\gamma_2}, & \mu_6 &= \frac{1}{\gamma_3}, & \mu_7 &= \frac{1}{\gamma_1} \end{aligned}$$

The unconditional mean time taken by the system to transition for any regenerative state j , when it (time) is counted from epoch of entrance into that state i is mathematically stated as:

$$m_{ij} = \int_0^{\infty} tq_{ij}(t)dt = -q_{ij}^{*'}(0)$$

Thus,

$$\begin{aligned} m_{01} &= p_1 p_2 \lambda_{h1} / S^2, & m_{02} &= q \lambda_{h1} / S^2, \\ m_{03} &= p_1 q_2 \lambda_{h1} / S^2, \\ m_{04} &= \lambda_{h3} / S^2, & m_{07} &= \lambda_{s1} / S^2 \end{aligned}$$

It is clear that $m_{01} + m_{02} + m_{03} + m_{04} + m_{07} = \mu_0$

Similarly,

$$\begin{aligned} m_{10} + m_{14} &= \mu_1, & m_{24} + m_{25} &= \mu_2 \\ m_{30} + m_{34} + m_{36} &= \mu_3, & m_{40} &= \mu_4, \\ m_{52} = \mu_5, & m_{60} = \mu_6 & \text{and} & m_{70} = \mu_7 \end{aligned}$$

4.2 Other Measures of System Performance

By probabilistic arguments for the regenerative process, we obtain the recursive relations for various measures of the system performance. On solving the recursive relations using Laplace and Laplace-Stieltjes transforms, we get the following measures in steady state:

$$\text{Mean Time to System Failure} \quad (T_0) = \frac{N_1}{D}$$

$$\text{Steady-State Availability} \quad (A_0) = \frac{N}{D_1}$$

Expected Down Time due to

$$\text{(a) Pure Hardware Failures} \quad (DH_0) = \frac{N_2}{D_1}$$

$$\text{(b) Pure Software Failures} \quad (DS_0) = \frac{N_3}{D_1}$$

$$\text{(c) Pure Hardware-Software Interaction Failures} \quad (DI_0) = \frac{N_4}{D_1}$$

Expected Number of

$$\text{(a) Software Repairs} \quad (RS_0) = \frac{N_5}{D_1}$$

$$\text{(b) Hardware Repairs} \quad (RH_0) = \frac{N_6}{D_1}$$

$$\text{(c) Hardware Repair by Software} \quad (HC_0) = \frac{N_9}{D_1}$$

$$\text{(d) Software Repair by Re-execution} \quad (SC_0) = \frac{N_{10}}{D_1}$$

Expected Number of Visits by

$$\text{(a) Hardware Engineer} \quad (VH_0) = \frac{N_7}{D_1}$$

$$\text{(b) Software Engineer} \quad (VS_0) = \frac{N_8}{D_1}$$

Where

$$D = 1 - p_{01}p_{10} - p_{03}p_{30}$$

$$D_1 = p_{24} \left[\mu_0 + \mu_1 p_{01} + \mu_3 p_{30} + \mu_7 p_{07} + \mu_4 \left(\frac{p_{01}p_{14} + p_{03}p_{34}}{p_{02} + p_{07}} \right) + \mu_6 p_{03}p_{36} \right] + \mu_2 p_{02} + \mu_5 p_{02}p_{25}$$

$$N = p_{24} [\mu_0 + \mu_1 p_{01} + \mu_3 p_{30}] + \mu_2 + p_{02}$$

$$N_1 = \mu_0 + \mu_1 p_{01} + \mu_2 p_{02} + \mu_3 p_{03}$$

$$N_2 = p_{24} \mu_4 [p_{02} + p_{01}p_{14} + p_{03}p_{34}]$$

$$N_3 = p_{24} p_{07} \mu_7$$

$$N_4 = p_{02} p_{25} \mu_5 + p_{24} p_{03} p_{36} \mu_6$$

$$N_5 = p_{02} p_{25} p_{52} + p_{24} p_{07} p_{70}$$

$$N_6 = p_{24} [p_{03} p_{30} + p_{04} p_{40}]$$

$$N_7 = p_{24} [p_{01} p_{14} + p_{03} p_{34} + p_{03} + p_{04}] + p_{02} p_{24}$$

$$N_8 = p_{02} p_{25} + p_{07} p_{24}$$

$$N_9 = p_{01} p_{10} p_{24}$$

$$N_{10} = p_{03} p_{36} p_{24}$$

5. COST-BENEFIT ANALYSIS

The expected total profit (P_0) incurred to the system in steady state is given by

$$P_0 = C_0 A_0 - C_1 (DH_0 + DS_0 + DI_0) - C_2 RH_0 - C_3 RS_0 - C_4 VH_0 - C_5 VS_0 - C_6 HC_0 - C_7 SC_0 - C_h - C_s$$

where

C_0 = revenue per unit up time of the system.

C_1 = cost per unit down time of the system.

C_2 = cost per unit of hardware repair.

C_3 = cost per unit of software repair.

C_4 = cost per visit of hardware engineer.

C_5 = cost per visit of software engineer.

C_6 = cost per unit of the hardware repair by software.

C_7 = cost per unit of the software repair by re-execution.

C_h = cost per unit of the hardware installation.

C_s = cost per unit of the software installation.

6. PARTICULAR CASE

The values of the various failures rates and probability of hardware degradation detection as given in Teng et al. [10] and Trivedi et al. [12] i.e $p_1=0$, $p_2=.95$, $q_1=1$, $q_2=.05$, $\lambda_{h1}=.000526$, $\lambda_{h2}=.000432$, $\lambda_{h3}=.000112$, $\lambda_{s1}=.000002$, $\lambda_{s2}=.000263$, $\lambda_{s3}=.00001$, are taken as a particular case.

For this case values of various measures of system performance are computed from the results given in preceding section for assumed values of repair rates and various costs as $\beta_1=2$, $\beta_2=.19$, $\beta_3=.18$, $\gamma_1=0.8$, $\gamma_2=0.9$, $\gamma_3=5$ $C_0=25000$, $C_1=1000$, $C_2=500$, $C_3=C_4=C_5=200$, $C_6=500$, $C_7=200$, $C_h=50000$, $C_s=150000$. These values are as under:

Mean Time to System Failure	= 3355.812
Steady-State Availability	= 0.9989373
Expected Up Time	=0.9989373
Expected Down Time due to	
(a) Pure Hardware Failures	= 4.30395
(b) Pure Software Failures	= .0058913
(c) Hardware based Software Interaction Failures	= .0524048
Expected Number of	
(a) Software Repairs	=.05010978
(b) Hardware Repairs	= .1649575
(c) Hardware Repair by Software	= 0
(d) Software Repair by Re-execution	= 0
Expected Number of Visits by	
(a) Hardware Engineer	= .9396684
(b) Software Engineer	=.05010978
Expected Profit	= -154507.4

7. GRAPHICAL ANALYSIS

For the analysis purpose various graphs are plotted for mean time to system failure and profit of the system for different values of hardware and software failures rates and probability of hardware degradation detection.

The behavior of MTSF with respect to various hardware failure rates λ_{h1} , λ_{h2} and hardware based software failure rate λ_{s2} are depicted in Fig. 2 to Fig. 4. It can be observed from the graphs that the MTSF decreases with the increase in the values of these failure rates. From Fig. 4, it can also be concluded that the MTSF increases with the increase in the values of the probability (p_1) when other parameters remain fixed.

Fig.5 shows the pattern of the Profit (P_0) with respect to the software failure rate λ_{s2} for different values of hardware failure rate λ_{h2} . The profit (P_0) decreases as the software

failure rate λ_{s2} increases and decreases for higher values of software failure rate λ_{h2} .

The curve in Fig. 6 gives the pattern of the profit (P_0) incurred from the system with respect to the revenue per unit up time (C_0) for different values of hardware failure rate λ_{h2} . The profit of the system increases as the revenue per unit up time (C_0) increases whereas it decreases for higher values of hardware failure rate λ_{h2} . It can also be observed from the graph that for $\lambda_{h2} = 0.0002$, P_0 is positive if $C_0 > 203694.036$. Therefore, the system is profitable whenever $C_0 > 203694.036$. Similarly for $\lambda_{h2} = 0.0006$ and $\lambda_{h2} = 0.001$, the system is profitable whenever $C_0 > 204099.85$ and 204534.86 respectively.

Fig.7 also reveals the pattern of the Profit (P_0) with respect to revenue per unit up time (C_0) for different values of the software failure rate λ_{s2} . The profit (P_0) increases as the revenue per unit up time (C_0) increases and decreases for higher values of software the failure rate λ_{s2} . It can also be observed from the graph that for $\lambda_{s2} = 0.0001$, P_0 is positive if $C_0 > 203694.036$. Therefore, the system is profitable whenever $C_0 > 203694.036$. Similarly for $\lambda_{s2} = 0.0006$ and $\lambda_{s2} = 0.0054$, the system is profitable whenever $C_0 > 204099.85$ and 204534.86 respectively.

8. CONCLUSIONS

A stochastic model is discussed for hardware-software system considering hardware, software and hardware based software failures along with different recovery methods. The performance and cost-benefit analyses of the model are carried out. It is concluded that the reliability and profit of the system decreases with the increase in the values of pure hardware, pure software and hardware based software failure rates when other parameters are kept fixed. However, these increase with the increase in the probability of detection of hardware degradation. The limits for revenue up time, hardware and software failure rates are/ can be obtained for the system to give positive profit that may be quite useful for both the system engineer and the system user.

9. REFERENCES

- [1] Friedman, M. A. and Tran, P., 1992. Reliability techniques for combined hardware/software systems, Proceeding Annual Reliability and Maintainability Symposium, pp.290-293
- [2] Hecht, H. and Hecht, M., 1986. Software Reliability in the System Context, *IEEE Transactions on Software Engineering*, Vol.12, pp.51-58.
- [3] Kumar, A. and Malik, S.C., 2012. Stochastic Modeling of a Computer System with Priority to PM over S/W Replacement Subject to Maximum Operation and Repair, *International Journal of Computer Applications*, Vol.43, No.3, pp. 27-34.
- [4] Welke, S.R., Johnson, B.W. and Aylor, J.H., 1995. Reliability Modeling of Hardware/Software Systems, *IEEE Trans. Reliability*, Vol.44, pp.413-418.
- [5] Boyd, M. A. and Monahan, C. M., 1995. Developing integrated hardware-software system reliability models: difficulties and issues [For Digital Avionics]. Proceeding of the digital avionics systems conference, 14th DASC, pp 193-198, Cambridge, USA.

- [6] Iyer, R.K. and Velardi, P., 1985. Hardware related Software Errors: Measurement and Analysis. IEEE Transactions on Software Engineering, Vol.11, pp.223-230.
- [7] Martin, R. and Mathur, A.P., 1990. Software and Hardware Quality Assurance: Towards a Common Platform for Increase the Usability of this Methodology, Proc. IEEE Conf. Comm.
- [8] Kanoun, K. and Ortalo-Borrel, M., 2000. Fault-Tolerant Systems Dependability-Explicit Modeling of Hardware and Software Component-Interactions, IEEE Trans. Reliability, Vol. 49, No. 4, pp. 363-376.
- [9] Huang, B., Li, X., Li, M., Bernstein, J. and Smidts, C., 2005. Study of the Impact of Hardware Fault on Software Reliability, Proc. 16th IEEE International Symp. Software Reliability Engineering (ISSRE'05).
- [10] Teng, X., Pham, H. and Jeske, D.R., 2006. Reliability Modeling of Hardware and Software Interactions and its Applications, IEEE Trans. Reliability, Vol.55, pp.571-577.
- [11] Kumar, R. and Kumar, M. 2011. Reliability and Availability Analysis of a Computer System with Hardware-Software Interactions and Different types of recovery, Proceeding of the International conference on Advances in Modeling, Optimization and Computing, I.I.T., Roorkee, pp.531-534.
- [12] Trivedi, K. S., Vasireddy, R., Trindade, D., Nathan, S., and Rick Castro., 2006. Modeling High Availability Systems. Proceeding Pacific Rim Dependability Conference PRDC.

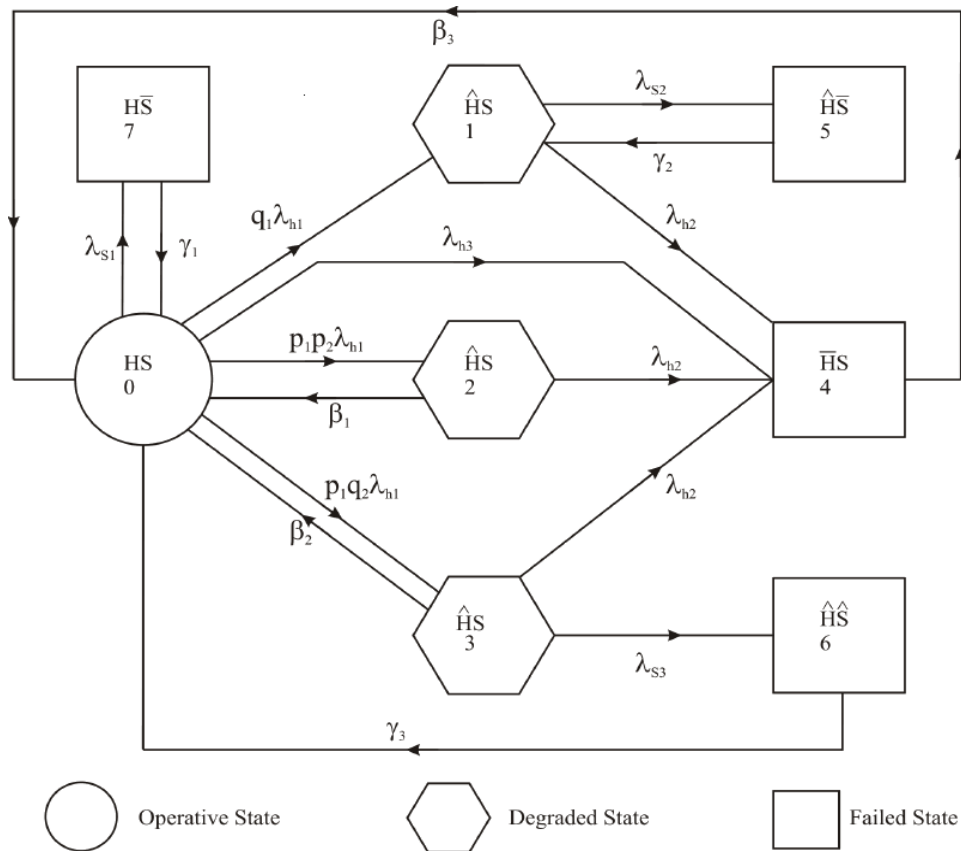


Fig. 1 State Transition Diagram

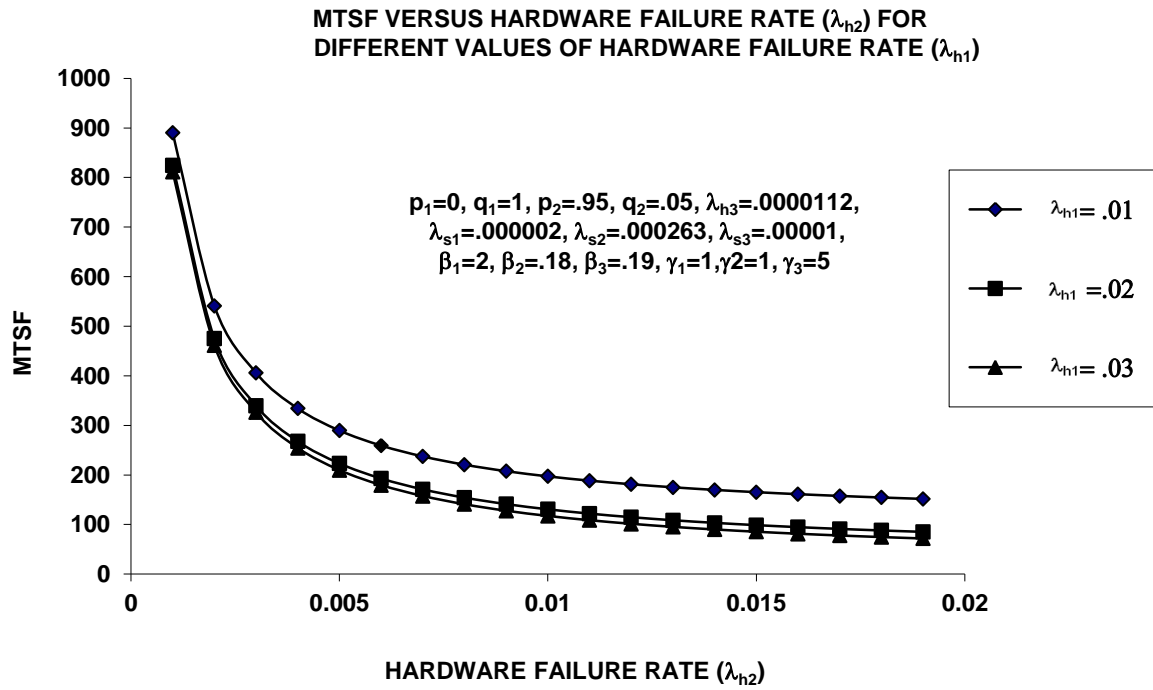


Fig.2

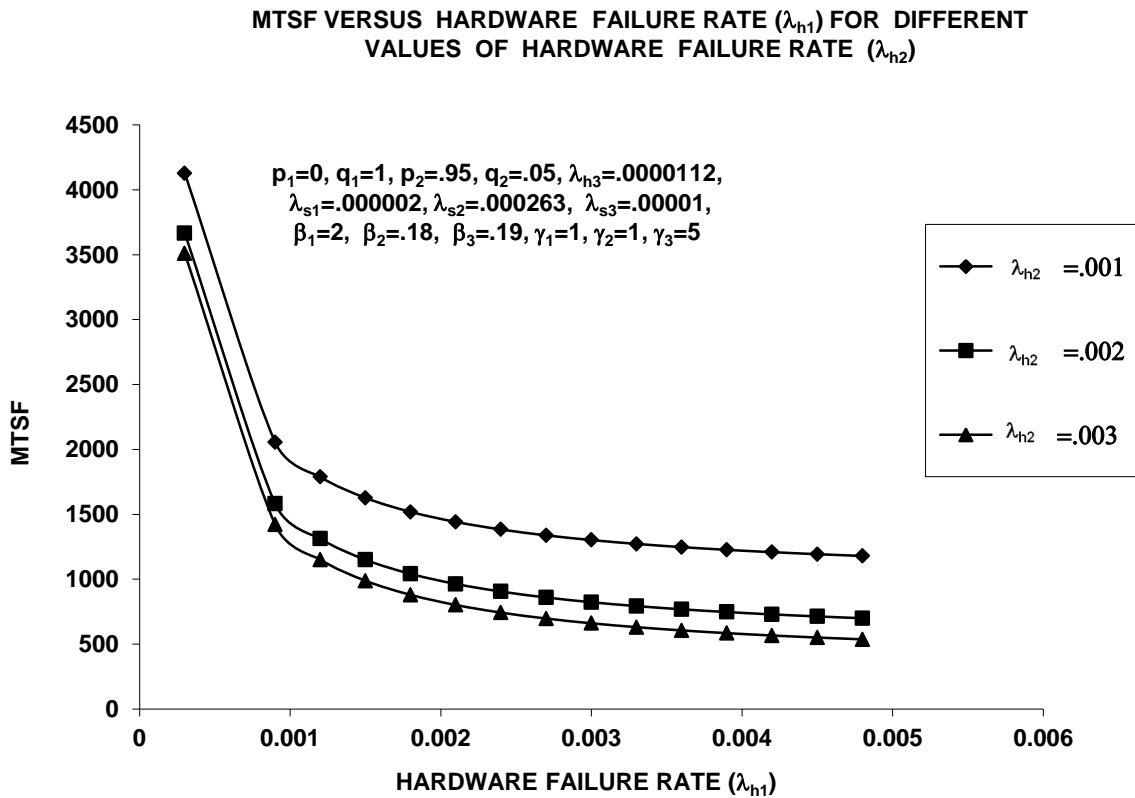


Fig. 3

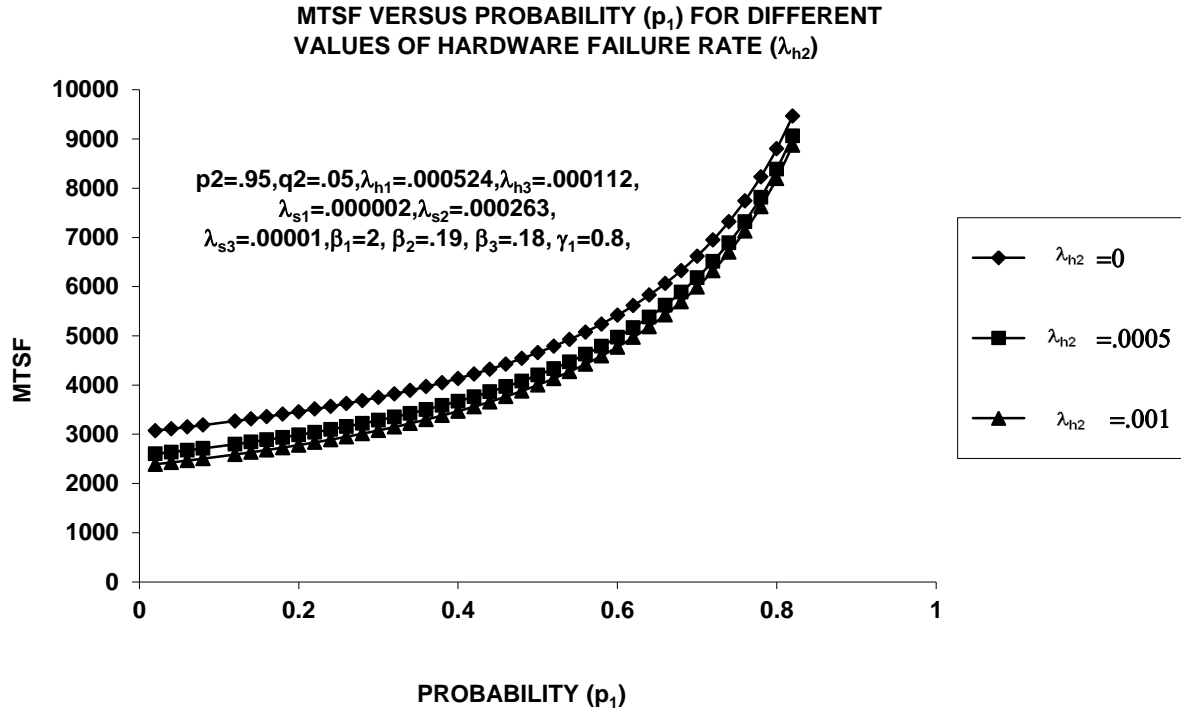


Fig.4

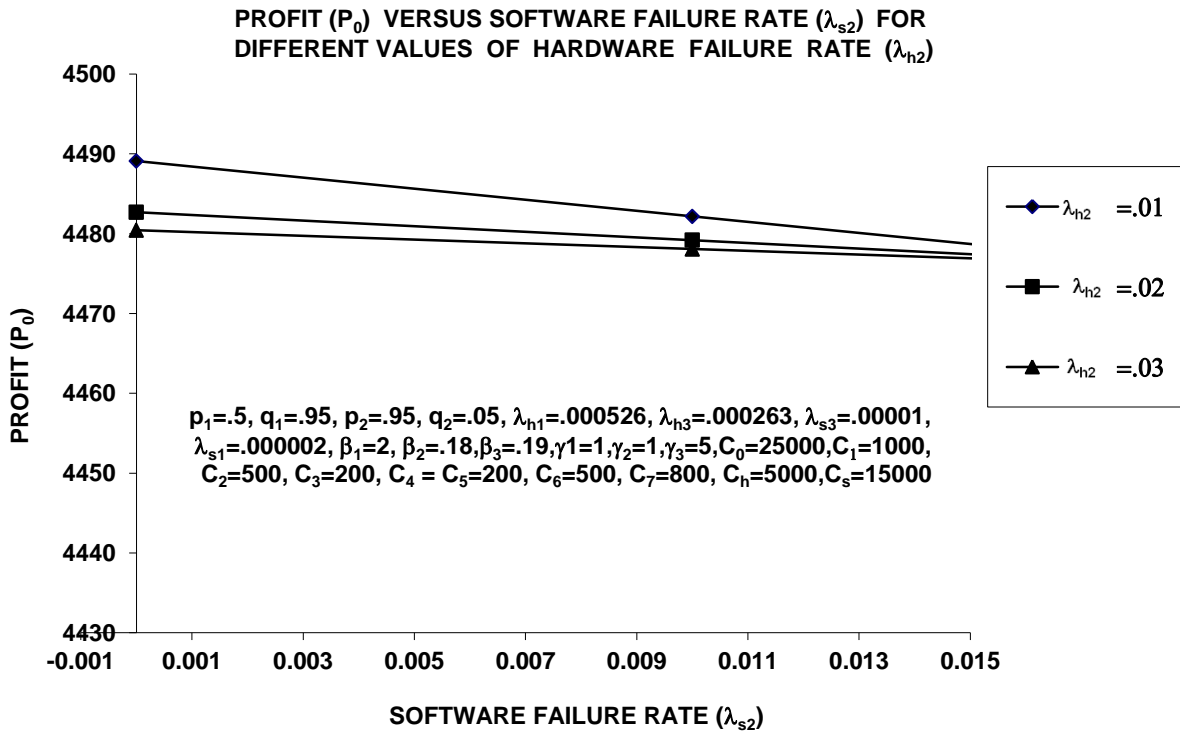


Fig. 5

PROFIT (P_0) VERSUS REVENUE PER UNIT UP TIME OF THE SYSTEM (C_0) FOR DIFFERENT VALUES OF SOFTWARE FAILURE RATE (λ_{s2})

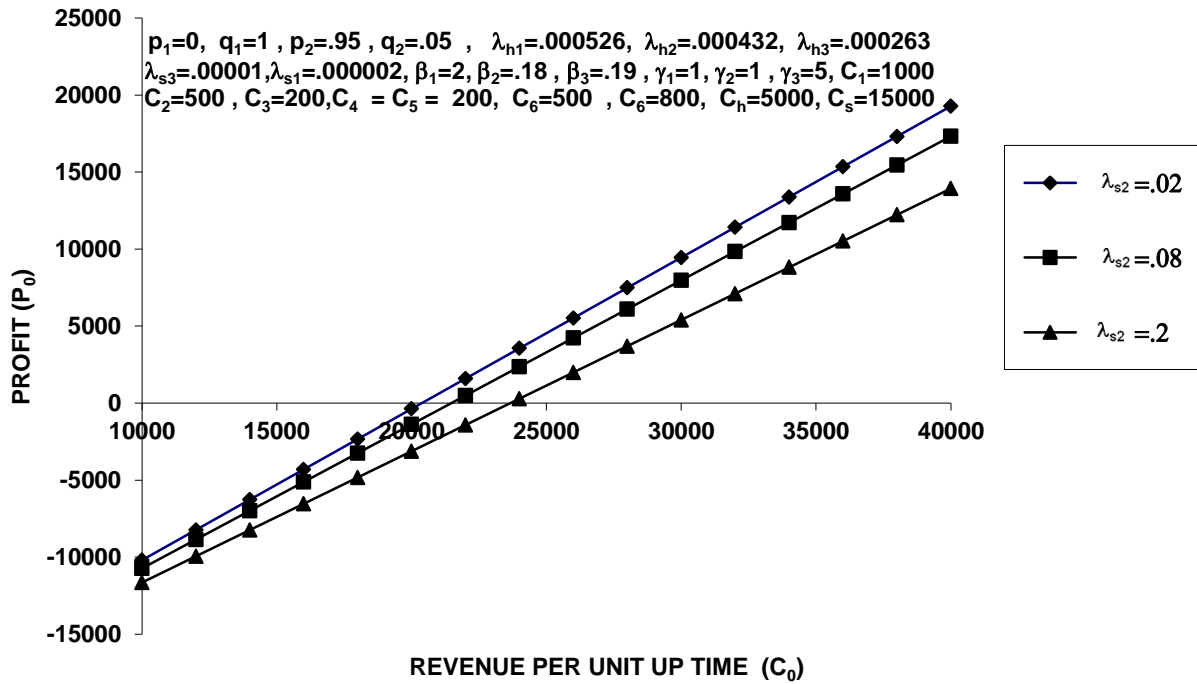


Fig.6

PROFIT (P_0) VERSUS REVENUE PER UNIT UP TIME OF SYSTEM (C_0) FOR DIFFERENT VALUES OF HARDWARE FAILURE RATE (λ_{h2})

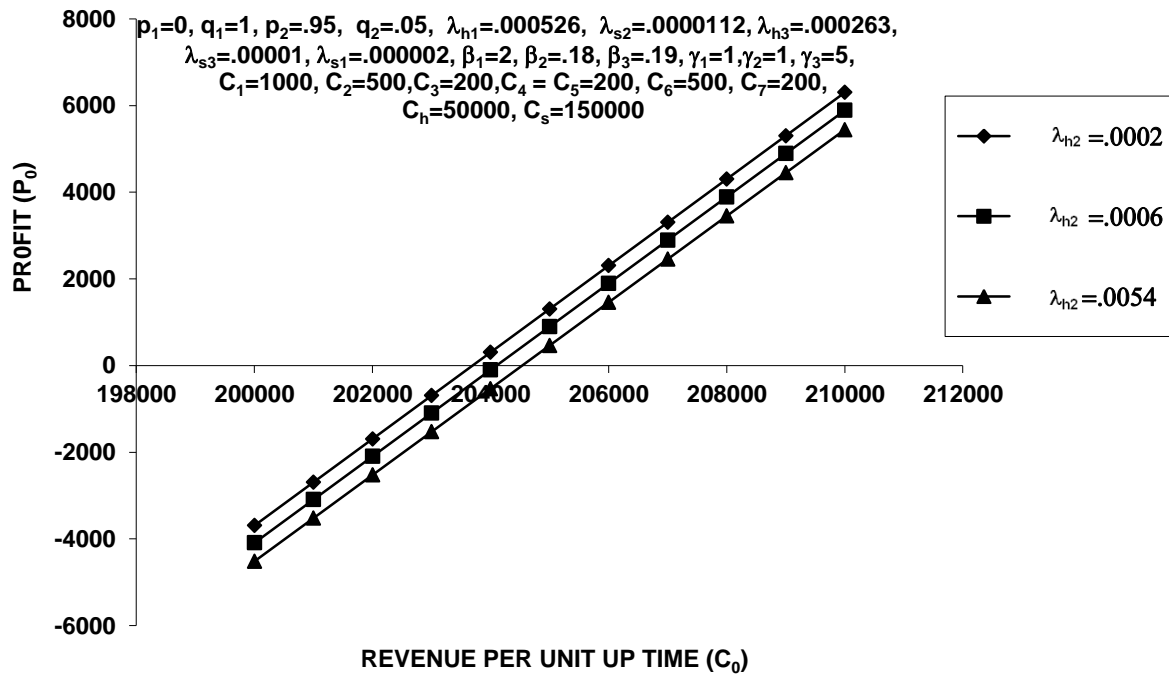


Fig. 7