**International Journal of Advanced Research and Development**
**ISSN: 2455-4030**
**www.newresearchjournal.com/advanced**
**Volume 1; Issue 2; February 2016; Page No. 47-54**

# Profit and availability optimization of two modules with maintenance

**Parmod Kumar**
Assistant Professor of Maths, Rajiv Gandhi S. D. Commerce & Science College, Narwana, Haryana, India.

## Abstract

This paper discusses the Profit and Availability Optimization of Two Modules with Maintenance, in which modules can work in complete and reduced capability. In this two module model both the modules can work in reduced state which means the software is having partial features and need corrective maintenance to reach the full capacity working state. This model also contains the perfective maintenance. Proposed model deals with two type of states of failures: Partially failure and fully failure. This model consists of module ‘A’ and module ‘B’. The module ‘A’ or ‘B’ can fail partially and hence can be in upstate. The module failure can be due to S/W or H/W. The software system can work with reduced capacity in a partially failed state. There is a single repairman available for both type of failure. Up-dation are faultless i.e. during repair the repairman never does any damage to the modules and a repaired module is working like a new module. The service time are exponential and is different for both type of failures(S/W and H/W). The software system is considered for steady-state. To measure the performance of the software system Regenerative Point Graphical Technique (RPGT) is used to study system parameters.

**Keywords:** Availability, Circuits, Base state, RPGT

## 1. Introduction

Reliability is defined as the probability of software to perform the prerequisite function over a specified period of time. Reliability is the significant parameter due to industry needs and huge operating cost involved in development and maintenance of software. Performance of software and hardware depends on reliability and availability that are involved in processes, operating environment, maintenance actions, as well as efficiency, and technical expertise of operators and servers. When reliability is low, actions are needed to improve them by reducing the failure rate or increasing the service rate for the components or whole sub systems. In the present scenario of competitive market to cut down the production cost of delivery performance of software by end user require continuous and long term use of the software to meet the ever increasing demand at low cost. Availability of software can be improved by maintenance and inspection. Software may not be working with full functionality at particular instant of time, this state of affairs is dealt with by considering the fuzziness measure of the state which makes the problem more realistic. It also helps to decide when the software is to be declared in failed state which helps the management to take decision whether to replace or enhance the existing system.

Availability analysis and cost for availability has been done by many researchers such as Jain Madhu and Rani Sulekha [1] discussed about enhancement of reliability with the provision of standby and repair facility for an embedded system which contain hardware and software components, Gupta Rakesh *et al.* [2] discussed the reliability and cost benefit analysis for two non identical unit model, Goel Pardeep and Satpal [3] discussed the availability analysis of dairy plant using RPGT, Trivedi Kishore S *et al.* [4] discussed the availability of real system model by using analytical technique from Cisco and Motorola, Malik S C [5] discussed about reliability modeling of computer system by giving priority to H/W repair as compared to S/W upgradation, Gupta Sanjay and Gupta Suresh Kumar [6] discussed the reliability measures using Semi Markov process and RPGT, Chillar S K *et al.* [7] discussed about the relation between profit and failure rate, Sureria J K *et al.* [8] elaborated the cost benefit analysis of computer system with priority to software development as compared to hardware repair, Malik SC *et al.* [9] discussed the model of reliability for single unit. Three states are considered normal state, partial failure and complete failure, Kim Dong Seong *et al.* [10] discussed the reliability and availability of a satellite system. Different threats were discussed to the satellite such as environment problem, software aging and network attacks, Dalal Sandeep and Chillar R S [11] discussed about most common and severe type of software failures, Asthana Abhaya and Okumoto Kazu [12] elaborated integrated design for software reliability which fill the gap between expected and actual behavior of system, Taneja Gulshan *et al.* [13] discussed about the cut off point for cost and revenue so that product rate can be decided by company, Erkoyuncu John Ahmad at al. [14] discussed the service system cost estimation.

In this paper the availability analysis of two modules software system is done. This model takes the consideration of partial and full failure and takes the consideration of all type of maintenance including the perfective maintenance. The mean time to system failure, Availability, Busy period of repairman, Number of repair man’s visits is calculated for the software system by using the Regenerative Point Graphical Technique (RPGT).

The paper confers the Availability Analysis of a software system consisting of Two Module System, in which module can work in full, reduced and perfective category also. This is two module models which can work in reduced state which means the software is having partial features and need corrective maintenance to reach the full capacity working state. It also includes the state for perfective maintenance also. Thus there are two type of failure: Partially failure and completely failure. The software system consists of module ‘A’ and ‘B’ which can work in reduced state after failure. The module ‘A’ or ‘B’ can work partially and hence can be in upstate, partially failed state (reduced state), perfective maintenance state or completely failed state. The software system can work with reduced capacity in a partially failed state. There is a single repairman for both type of failure. Repairs are unadulterated. The software is down if software module is failed completely. Repair times are exponential and are different for both type of failure i.e software and hardware failure. This model is elaborated for stable state conditions. The following software system features have been assessed to study the system performance by using the technique Regenerative Point Graphical Technique (RPGT).

1) Mean Time To Software Failure (MTSF).
2) Total fraction of time for which the software is available.
3) The busy period of the repairman doing any given job.
4) The number of the repairman's visits.

Tables and graphs are prepared to represent the performance of the model.

2. **Assumptions and Notations**: - The following assumptions and notations/symbols are used:

1) The software contains two modules which can work in reduced state after partial failure but cannot work in completely failed state.
2) The module 'A' or 'B' can work partially and hence can be in upstate and this state is considered to be corrective/adaptive/perfective maintenance category to make the system according to requirement of user. The software can work with reduced capacity in a partially failed state.
3) Perfective and adaptive maintenance is endless.
4) Single server facility is there for completely failed state and reduced capacity state of any module.
5) Repair facility never does any destruction to the software.
6) A repaired module works like a new-one.
7) The software is down if module is in completely failed state. Replica of modules can be used to up the software.
8) H/W up gradation is different from hardware failure.
9) When the software system is in failed state due to H/W failure it cannot fail further due to S/W failure. H/W failure is replaced with new one.
10) The software is discussed for steady state conditions.
11) The distribution of failure time and repair time are exponential and general.
12) The failure time and repair time are independent for S/W and H/W failure.
13) Frequency of S/W perfective maintenance is more as compared to corrective maintenance.

pr/pf : Probability/transition probability factor.

$q_{x,y}(t)$: It represent the probability density function of the first passage time from a regenerative state x to a failed state or regenerative state y without visiting any additional regenerative state in the time (0,t].

$p_{x,y}$ : It represent the probability for steady state transition for regenerative state x to a regenerative state y without visiting any additional regenerative state. $p_{x,y} = q^{*}_{x,y}(0)$; where the symbol $*$ denotes Laplace transformation.

$\overline{cycle}$ : It represent the path formed through non failed states of the model.

m-cycle : It represent the path with terminals at the regenerative state m by considering the regenerative, failed or non-regenerative states for path formation.

m-$\overline{cycle}$ : It represent the path with terminals at the regenerative state m by considering only non-failed regenerative or non-regenerative states for path formation.

$\left(\theta \xrightarrow{sff} x\right)$ : It represent the directed simple failure free path from base state $\theta$ to another state x.

$\left(x \xrightarrow{s_r} y\right)$ : It represent the path from x-state to y-state which is $r$-th directed simple path, Where r can takes integral values of positive range for different paths from x to y state.

$V_{m,m}$ :*pf* of the state m accessible from the terminal state m of the m-cycle.

$\omega_x(t)$ : It represent the probability of server, which is busy doing a particular job at time t without transiting to any other regenerative state 'x' through one or more non- regenerative states, given that the system entered the regenerative state 'x' at t=0.

$V_{\overline{m,m}}$ : It represent the $pf$ of the state m accessible from the terminal state m of the m- $\overline{cycle}$.

$R_x(t)$ : It represent the reliability of the software system at given time t, where x is considered un-failed regenerative at t=0.

$A_x(t)$ : It represent the probability that the software system is available in working state at time t, by considering the software system entered regenerative state x at t=0.

$B_x(t)$ : It represent the probability that the repairman is busy doing a particular job at time t, by considering the software system entered regenerative state x at t=0

$V_x(t)$ : It represent the expected number of repairman visits for a given job in (0,t], by considering the software system entered regenerative state x at t=0

$\mu_x$ : It represent the mean time to stay in state i, before moving to any other states; $\mu_x = \int_0^\infty R_x(t)dt$.

$\mu_x^1$ : It represent the total un-conditional time spent before transiting to any other regenerative states, by considering the software system entered regenerative state x at t=0

$\eta_x$ : It represent the expected waiting time spent while doing a given job, given that the system entered regenerative state ‘x’ at t=0; $\eta_x = \omega_x^*(0)$.
$f_y$ : It represents the Fuzziness measure of the y-state and which is assumed to be 1.
$\lambda_1/\lambda_2$ : It represents the constant failure rate of the module ‘A’ or ‘B’ to a partially failed state/ from partially failed state to a totally failed state.
$\lambda$ : It represents the constant failure rate for hardware failure.
$g(t)/G(t)$ : It represents the probability density function/cumulative distribution function of the repair-time of the module ‘A’ or ‘B’ from the partially failed state due to H/W failure.
$h(t)/H(t)$ : It represents the probability density function/cumulative distribution function of the repair-time of the module ‘A’ or ‘B’ from the completely failed state or partially working state due to S/W failure. The system can be in any of the following states with respect to the above symbols.
$S_0$=A.B $S_1$=A.B(H) $S_2$=a.B $S_3$=a.B(H) $S_4$=a.b $S_5$=A.b $S_6$=(A.b)H $S_7$=ABP $S_8$=ABP(H)

A and B are two modules and states are described below:
$S_0$=Both the modules are working in normal mode., $S_1$=Modules not working due to H/W failure.
$S_2$=Module A is not working but B is working., $S_3$=Failure state after $S_2$ due to H/W failure.
$S_4$=Both modules are not working due to S/W failure.,
$S_5$= Module A is working but B is not working., $S_6$=Failure state after $S_2$ due to S/W failure.
$S_7$=System is in perfective maintenance state., $S_8$=H/W not supported after perfective maintenance.

State $S_0$, $S_1$, $S_2$, $S_3$, $S_4$, $S_5$, $S_6$, $S_7$,$S_8$ are regenerative states. The possible transitions between states along with transition time are shown in Fig. 1

**Transition diagram of the system**

**Table 1:** Transition States Symbols

| State | Symbol |
|---|---|
| Regenerative state/point | • |
| Up-state | ○ |
| Failed state | ▭ |
| Degenerated/Reduced state | ⬭ |
| Perfective state | △ |

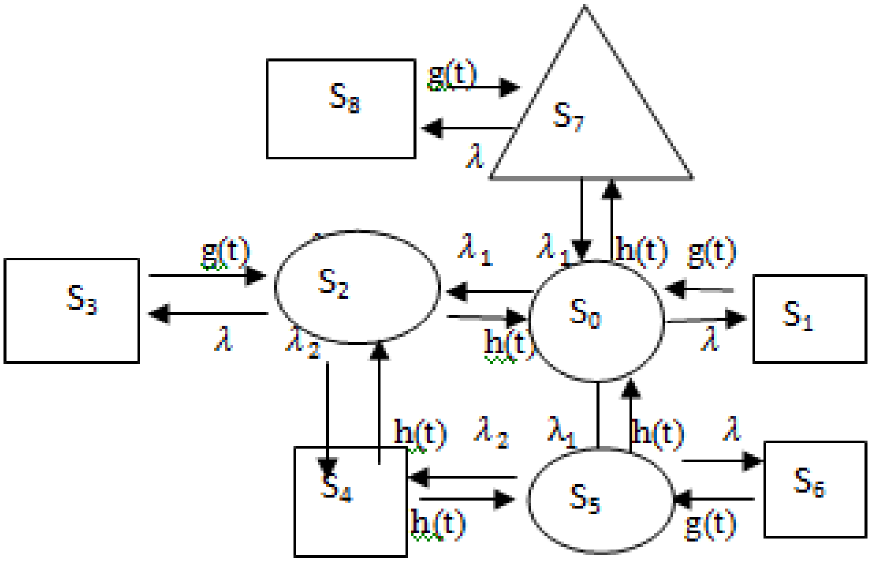

**Fig. 1:** Primary, Secondary, Tertiary Circuits w.r.t. the Simple Paths (Base-State ‘0’)

**Table 2:** Circuits for Base State ‘0’

| **Vertex y** | $\left(0 \overset{S_r}{\rightarrow} y\right)$: **(P0)** | **(P1)** | **(P2)** | **(P3)** |
|---|---|---|---|---|
| 1 | $\left(0 \overset{S_1}{\rightarrow} 1\right)$:{0,1} | NIL | Nil | Nil |
| 2 | $\left(0 \overset{S_1}{\rightarrow} 2\right)$:{0,2} | {2,3,2} | Nil | Nil |
| 3 | $\left(0 \overset{S_1}{\rightarrow} 3\right)$:{0,2,3} | {2,4,2} | Nil | Nil |
| 4 | $\left(0 \overset{S_1}{\rightarrow} 4\right)$:{0,2,4}<br>{0,5,4} | {2,3,2}<br>{5,6,5} | Nil | Nil |
| 5 | $\left(0 \overset{S_1}{\rightarrow} 5\right)$:{0,5}<br>{0,2,4,5} | {2,4,2}<br>{2,3,2} | Nil | Nil |
| 6 | $\left(0 \overset{S_1}{\rightarrow} 6\right)$:{0,5,6}<br>{0,2,4,5,6} | {5,4,5}<br>{2,3,2} {4,2,4} | {4,5,4} | Nil |
| 7 | $\left(0 \overset{S_1}{\rightarrow} 7\right)$:{0,7} | {7,8,7} | Nil | Nil |
| 8 | $\left(0 \overset{S_1}{\rightarrow} 8\right)$:{0,7,8} | NIL | Nil | Nil |

**Table 3:** Probability Function

| $\boldsymbol{p_{x,y} = q^*_{x,y}(0)}$ |
|---|
| $p_{0,1}(t) = \frac{\lambda}{\lambda + \lambda_1 + \lambda_1}\,(1 - h^*(\lambda + \lambda_1 + \lambda_1))$<br>$p_{0,2}(t) = \frac{\lambda_1}{\lambda + \lambda_1 + \lambda_1}\,(1 - h^*(\lambda + \lambda_1 + \lambda_1))$<br>$p_{0,5}(t) = \frac{\lambda_1}{\lambda + \lambda_1 + \lambda_1}\,(1 - h^*(\lambda + \lambda_1 + \lambda_1))$<br>$p_{0,7} = h^*(\lambda + \lambda_1 + \lambda_1)$ |
| $p_{1,0} = g^*(0) = 1$ |
| $p_{2,0} = h^*(\lambda + \lambda_2)$<br>$p_{2,3}(t) = \frac{\lambda}{\lambda + \lambda_2}\,(1 - h^*(\lambda + \lambda_2))$<br>$p_{2,4}(t) = \frac{\lambda_2}{\lambda + \lambda_2}\,(1 - h^*(\lambda + \lambda_2))$ |
| $p_{3,2}(t) = g^*(0) = 1$ |
| $p_{4,2}(t) = 1/2$<br>$p_{4,5}(t) = 1/2$ |
| $p_{5,0}(t) = h^*(\lambda + \lambda_2)$<br>$p_{5,4}(t) = \frac{\lambda_2}{\lambda + \lambda_2}\,(1 - h^*(\lambda + \lambda_2))$<br>$p_{5,6} = \frac{\lambda}{\lambda + \lambda_2}\,(1 - h^*(\lambda + \lambda_2))$<br>$p_{5,6}(t) = g^*(0) = 1$ |
| $p_{6,5}(t) = g^*(0) = 1$ |
| $p_{7,0}(t) = \frac{\lambda_1}{\lambda + \lambda_1}$<br>$p_{7,8}(t) = \frac{\lambda}{\lambda + \lambda_1}$ |
| $p_{8,7}(t) = g^*(0) = 1$ |

**Table 4:** Mean Stay Time

| **R$_x$(t)** | $\boldsymbol{\mu_x = R_x^*(0)}$. |
|---|---|
| $R_0(t) = e^{-(\lambda+\lambda_1+\lambda_1)t}H(t)$ | $\mu_0 = \dfrac{1 - h^*(\lambda + \lambda_1 + \lambda_1)}{\lambda + \lambda_1 + \lambda_1}$ |
| $R_1(t) = \bar{G}(t)$ | $\mu_1 = -g^{*\prime}(0)$ =1/$\omega_1$ |
| $R_2(t) = e^{-(\lambda_2+\lambda)t}H(t)$ | $\mu_2 = \dfrac{1 - h^*(\lambda_2 + \lambda)}{\lambda_2 + \lambda}$ |
| $R_3(t)$=$\bar{G}(t)$ | $\mu_3 = -g^{*\prime}(0)$=1/$\omega_1$ |
| $R_4(t) = H(t).H(t)$ | $\mu_4 = -h^{*'}(0) * -h*'(0)$<br>=1/$\omega_1$*1/$\omega_1$ |
| $R_5(t) = e^{-(\lambda+\lambda_2)t}H(t)$ | $\mu_5 = \frac{1-h^*(\lambda+\lambda_2)}{\lambda+\lambda_2} = \frac{1}{\lambda+\lambda_1+\omega 1}$ |
| $R_6(t)$=$\bar{G}(t)$ | $\mu_6 = -g^{*\prime}(0)$=1/$\omega_1$ |
| $R_7(t) = e^{-(\lambda+\lambda_1)t}$ | $\mu_7 = \frac{1}{\lambda+\lambda_1}$ |
| $R_8(t)$=$\bar{G}(t)$ | $\mu_8 = -g^{*\prime}(0)$=1/$\omega_1$ |

## 3. Transition Probabilities

$q_{x,y}(t)$ : It represent the probability density function of the first passage time from a regenerative state x to a regenerative state or failed state y without visiting any additional regenerative state in the time (0,t].

$p_{x,y}$ : It represent the probability for steady state transition for regenerative state x to a regenerative state y without visiting any additional regenerative state. $p_{x,y} = q_{x,y}^*(0)$; where the symbol $*$ denotes Laplace transformation.

### Mean Sojourn Times

$R_x(t)$ : reliability of the software time t, given that the software in regenerative state x.

$\mu_x$ : mean stay time spent in state x, before visiting any other states; $\mu_x = \int_0^\infty R_x(t)dt = R_x^*(0)$.

### Evaluation of Parameters

The mean time to software system failure and all the parameters of the system are calculated by using Regenerative Point Graphical Technique (RPGT) and using '0' as the base-state of the system as under:

The transition probability factors for all the states reachable from the base state '0' are:

$V_{0,0}$ = [(0,1,0)+(0,2,0)/(1-$L_1$)(1-$L_2$)(1-$L_3$)+(0,5,0)/(1-$L_5$)(1-$L_6$)(1-$L_7$)+(0,7,0)/(1-$L_8$) = 1

$V_{0,1}$ = (0,1) = $p_{0,1}$

$V_{0,2}$ = (0,2)/(1-$L_1$)(1-$L_2$)+(0,5,4,2)/(1-$L_5$)(1-$L_6$)

(1-$L_1$)(1-$L_7$)

$V_{0,3}$ = (0,2,3)/(1-$L_1$)(1-$L_2$)(1-$L_3$)+(0,5,4,2,3)/(1-$L_5$)(1-$L_3$)(1-$L_1$)(1-$L_7$)

$V_{0,4}$ = (0,5,4)/(1-$L_5$)(1-$L_6$)(1-$L_2$)+(0,2,4)/(1-$L_2$)

(1-$L_1$)(1-$L_3$)$V_{0,5}$ = (0,5)/(1-$L_5$)(1-$L_6$)(1-$L_2$)+(0,2,4,5)/(1-$L_2$)(1-$L_1$)(1-$L_3$)(1-$L_5$)

$V_{0,6}$ = (0,5,6)/(1-$L_5$)(1-$L_6$)(1-$L_2$)+(0,2,4,5,6)/(1-$L_2$)(1-$L_1$)(1-$L_3$)(1-$L_5$),

$V_{0,7}$ = (0,7)/(1-$L_8$), $V_{0,8}$ = (0,7,8)/(1-$L_8$)

Where

1-$L_1$ = 1 – {2,3,2} = 1-$p_{2,3}p_{3,2}$, 1-$L_2$ = 1 – {2,4,2} = 1-$p_{2,4}p_{4,2}$, 1-$L_3$ = 1 – {4,5,4} = 1-$p_{5,4}p_{4,5}$

1-$L_5$ = 1 – {5,6,5} = 1-$p_{5,6}p_{6,5}$, 1-$L_6$ = 1 – {5,4,5} = 1-$p_{5,4}p_{4,5}$, 1-$L_7$ = 1 – {4,2,4} = 1-$p_{4,2}p_{2,4}$

1-$L_8$ = 1 – {7,8,7} = 1-$p_{7,8}p_{8,7}$

4. **MTSF($T_0$):** It represents the regenerative un-failed states to which the system can travel from the base state before entering any failed state. As per given fig.1 these states are: x = 0,2,5,7. For '$\theta$' = '0', MTSF is calculated by:

$$\text{MTSF}=\left[\sum\nolimits_{x,s_r}\left\{\frac{\left\{pr\left(\theta\xrightarrow{s_r(sff)}x\right)\right\}.\mu_x}{\prod_{m_1\neq\theta}\{1-V_{\overline{m_1,m_1}}\}}\right\}\right] \div \left[1-\sum\nolimits_{s_r}\left\{\frac{\left\{pr\left(\theta\xrightarrow{s_r(sff)}\theta\right)\right\}}{\prod_{m_2\neq\theta}\{1-V_{\overline{m_2,m_2}}\}}\right\}\right]$$

$T_0$=[(0,0) $\mu_0$+(0,2) $\mu_2 + (0,5)\mu_5 + (0,7)\mu_7$]÷ ([1–($L_1$+$L_2 + L_3$)])= N ÷ D

Where, $L_1$=(0,2,0)=$p_{0,2}p_{2,0}$ $L_2$=(0,5,0)= $p_{0,5}p_{5,0}$, $L_3$=(0,7,0)=$p_{0,7}p_{7,0}$

N=(0,0) $\mu_0$+(0,2) $\mu_2 + (0,5)\mu_5 + +(0,7)\mu_7$, D = [1–($L_1$+$L_2 + L_3$)]

5. **Availability of the software system:** It represents the regenerative states, at which the software system is available. As per given fig.1 these states are: y = 0,2,5,7 and the regenerative states are x = 0 to 8. For '$\theta$' = '0', the total duration for which the system is available is given by:

$$A_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} y)\} f_y . \mu_y}{\prod_{m_1\neq\theta}\{1-V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} x)\} . \mu_x^1}{\prod_{m_2\neq\theta}\{1-V_{m_2,m_2}\}}\right\}\right] = \left[\sum_y V_{\theta,y} . f_y . \mu_y\right] \div \left[\sum_x V_{\theta,x} . \mu_x^1\right]$$

$$= [V_{0,0}f_0\mu_0+V_{0,2}f_2\mu_2+V_{0,5}f_5\mu_5+V_{0,7}f_7\mu_7]/[V_{0,0}\mu_0^1+V_{0,1}\mu_1^1+V_{0,2}\mu_2^1+V_{0,3}\mu_3^1+V_{0,4}\mu_4^1+V_{0,5}\mu_5^1+V_{0,6}\mu_6^1+V_{0,7}\mu_7^1+V_{0,1}\mu_8^8]$$

6. **Busy period of the Repairman:** It represents the regenerative states where repairman is busy while doing repairs. As per given fig.1 these state are: y =1,2,3,4,5,6,8; the regenerative states are: x = 0 to 8.For '$\theta$' = '0', the total duration for which the repairman is busy is given by:

$$B_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} y)\} . \eta_y}{\prod_{m_1\neq\theta}\{1-V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} x)\} . \mu_x^1}{\prod_{m_2\neq\theta}\{1-V_{m_2,m_2}\}}\right\}\right] = \left[\sum_y V_{\theta,y} . \eta_y\right] \div \left[\sum_x V_{\theta,x} . \mu_x^1\right]$$

$$B_0 = [V_{0,1}n_1+V_{0,2}n_2+V_{0,3}n_3+V_{0,4}n_4+V_{0,5}n_5+V_{0,6}n_6+V_{0,8}n_8]/[V_{0,0}\mu_0^1+V_{0,1}\mu_1^1+V_{0,2}\mu_2^1+V_{0,3}\mu_3^1+V_{0,4}\mu_4^1+V_{0,5}\mu_5^1 +V_{0,6}\mu_6^1+V_{0,7}\mu_7^1+V_{0,1}\mu_8^8]$$

7. **Expected number of Repairman's visits:** It represent the regenerative states where the repairman visits for repairs of the system. As per given fig.1 these states are: y = 1, 2, 5 and perfective maintenance should be considered as a special case. The regenerative states are: x = 0 to 6. For '$\theta$' = '0', the expected number of server's visits per unit time is:

$$V_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} y)\}}{\prod_{m_1\neq\theta}\{1-V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \xrightarrow{S_r} x)\} . \mu_x^1}{\prod_{m_2\neq\theta}\{1-V_{m_2,m_2}\}}\right\}\right] = \left[\sum_y V_{\theta,y}\right] \div \left[\sum_x V_{\theta,x} . \mu_x^1\right]$$

$$= (V_{0,1}+V_{0,2}+V_{0,5})/[V_{0,0}\mu_0^1+V_{0,1}\mu_1^1+V_{0,2}\mu_2^1+V_{0,3}\mu_3^1+V_{0,4}\mu_4^1+V_{0,5}\mu_5^1+V_{0,6}\mu_6^1 +V_{0,7}\mu_7^1 +V_{0,1}\mu_8^8]$$

8. **Analytical discussion** For Availability: $\lambda = 0.001$; $\lambda_2 = 0.002$;

**Availability ($A_0$) vs. Repair Rate(ω):** The Availability of the software system is calculated for different values of the Failure Rate ($\lambda_1$) by taking $\lambda_1$ = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.01 and for different values of the repair rate (ω)by taking ω = 0.80, 0.85, 0.90, 0.95 and 1.0. The data so obtained are shown in Table.5.

**Table 5:** Availability Vs Repair Rate

| $\lambda_1$ | $A_0$ (ω=0.80) | $A_0$ (ω=0.85) | $A_0$ (ω=0.90) | $A_0$ (ω=0.95) | $A_0$ (ω=1) |
|---|---|---|---|---|---|
| 0.005 | 0.998671 | 0.998758 | 0.998833 | 0.998900 | 0.998960 |
| 0.006 | 0.998635 | 0.998728 | 0.998808 | 0.998879 | 0.998941 |
| 0.007 | 0.998594 | 0.998693 | 0.998779 | 0.998854 | 0.998920 |
| 0.008 | 0.998546 | 0.998653 | 0.998745 | 0.998825 | 0.998895 |
| 0.009 | 0.998491 | 0.998608 | 0.998707 | 0.998793 | 0.998867 |
| 0.01 | 0.998431 | 0.998557 | 0.998664 | 0.998756 | 0.998836 |

Table 6 and graph in fig. 3 shows the behavior of the Availability ($A_0$) vs. the Repair Rate (ω) of the Unit of the System for different values of the Failure Rate ($\lambda_1$). It is observed that Availability of software system increases with increase in the values of the Repair Rate (ω).Further it can be observed from the table that values of Availability ($A_0$) shows the expected trend for different values of Failure Rates. Availability decreases with the increase in the values of Failure Rate($\lambda_1$).

**Expected number of repairman visits($V_0$) vs. Failure rate:** The expected number of repairman visits is calculated for different failure rate. Expected number of repairman visits are calculated for given value of $\lambda = 0.001$and $\lambda_2 = 0.002$. By taking $\lambda_1$ = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.010. The data so obtained are shown in Table 6.

**Table 6:** Repairman's Visits Vs Failure Rate

| $\lambda_1$ | $V_0$(ω=0.80) | $V_0$ (ω=0.85) | $V_0$ (ω=0.90) | $V_0$ (ω=0.95) | $V_0$ (ω=1.0) |
|---|---|---|---|---|---|
| 0.005 | 0.005704 | 0.005714 | 0.005724 | 0.005732 | 0.005739 |
| 0.006 | 0.006666 | 0.006681 | 0.006695 | 0.006707 | 0.006717 |
| 0.007 | 0.007610 | 0.007631 | 0.007650 | 0.007666 | 0.007680 |
| 0.008 | 0.008539 | 0.008567 | 0.008592 | 0.008613 | 0.008631 |
| 0.009 | 0.009454 | 0.009490 | 0.009521 | 0.009547 | 0.009571 |
| 0.01 | 0.010357 | 0.010400 | 0.010438 | 0.010471 | 0.010500 |

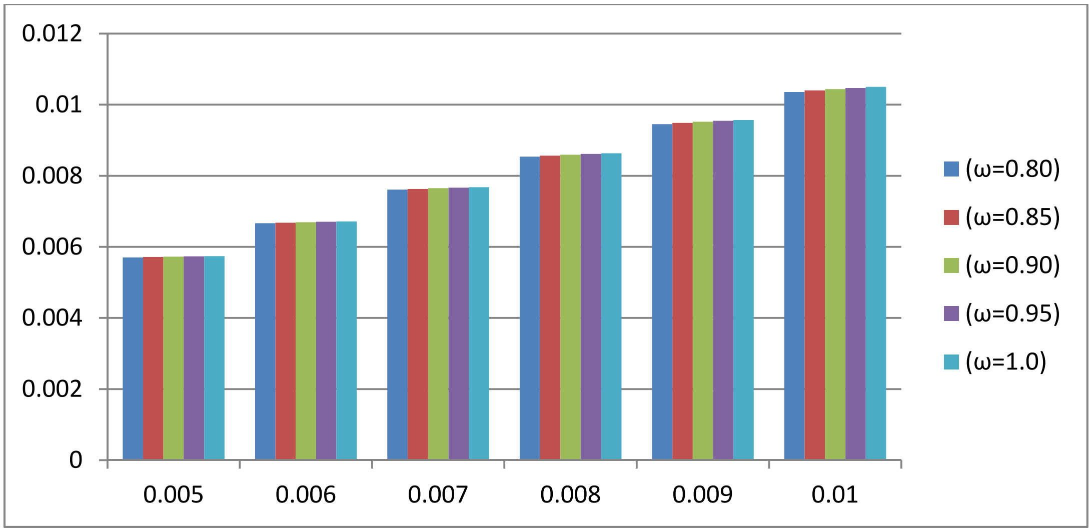

**Fig 2**

**Profit Optimization:** The net profit can be calculated by the difference of revenue generated and cost involved in number of repairman visits.

$P(t) = S_0A_0 - S_1V_0$

Where $S_0$ = revenue per unit when software is available.
$S_1$ = cost per visit of the repairman.
Considering the fixed values for $S_0$, $S_1$ net profit can be calculated from the software system. $A_0$= Availability of software and $V_0$= Number of repairman visits.

**Special case:** Assuming the values for $S_0 = 100000$, $S_1=10000$ and $\lambda_2= 0.002$, $\lambda= 0.001$;

**Table 7:** Profit Earned Vs Failure and Repair rate

| $\lambda_1$ | P(t) (ω=0.80) | P(t) (ω=0.85) | P(t) (ω=0.90) | P(t) (ω=0.95) | P(t) (ω=1) |
|---|---|---|---|---|---|
| 0.005 | 99810 | 99818 | 99826 | 99832 | 99838 |
| 0.006 | 99796 | 99805 | 99813 | 99820 | 99826 |
| 0.007 | 99783 | 99792 | 99801 | 99808 | 99815 |
| 0.008 | 99769 | 99779 | 99788 | 99796 | 99803 |
| 0.009 | 99754 | 99765 | 99775 | 99783 | 99790 |
| 0.010 | 99739 | 99751 | 99762 | 99770 | 99778 |

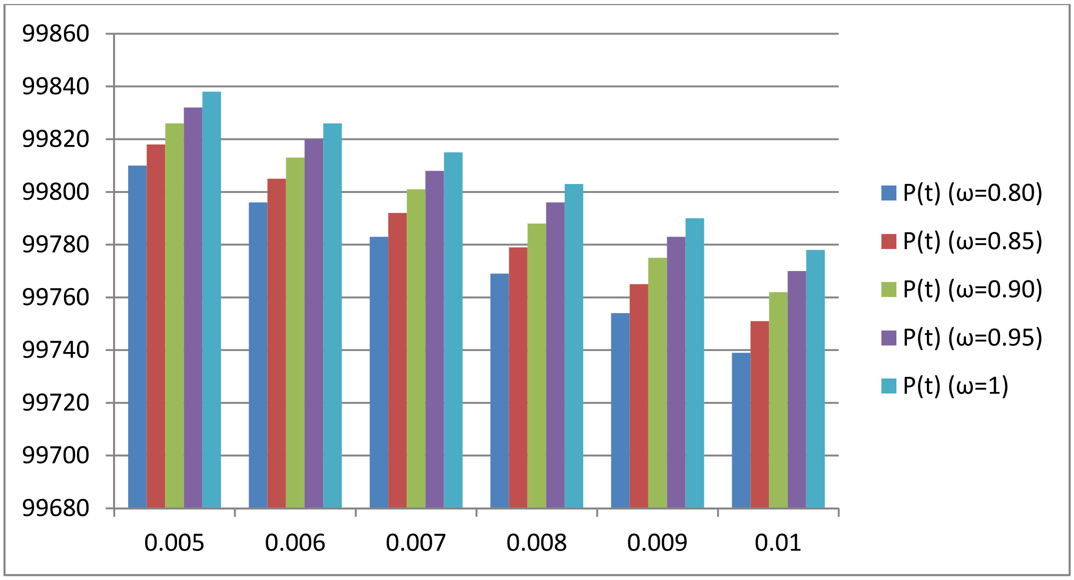

**Fig 3**

## 9. Conclusion

From the Tables and graphs, we see that, Availability of the Software System is increasing, as the Repair Rate(w) increases. Profit of the system increases with increase in repair rate and decrease with increase in failure rate. The study can be extended for more than two modules software system with perfective maintenance. In future, parameters can be evaluated when Repair rate and Failure rate are variable. Results obtained can be used for cost and benefit analysis. Any state can be taken as the Base-state to evaluate the various parameters. Study can also be extended for time dependent cases also.

## 10. References

1 Jain M, Rani S. Transient analysis of hardware and software systems with warm standbys and switching failures. International Journal of Mathematics in Operational Research. 2014; 6:1-28.

2 Gupta R, Kujal S, Mahi M. A two dissimilar unit parallel system with two phase repair by skilled and ordinary repairmen. International Journal of System Assurance Engineering and Management. 2014; 5:554-561.

3 Goel P. Availability Analysis of a Dairy Plant with Priority in Repair Using RPGT. 2014.

4 Trivedi KS, Kim DS, Ghosh R. System availability assessment using stochastic models. Applied Stochastic Models in Business and Industry. 2013; 29:94-109.

5 Malik S. Reliability modeling of a computer system with preventive maintenance and priority subject to maximum operation and repair times. International Journal of System Assurance Engineering and Management. 2013; 4:94-100.

6 Gupta S, Gupta SK. Stochastic analysis of a reliability model of one-unit system with post inspection, post repair, preventive maintenance and replacement. IJMERR. 2013; 2: 178-188.

7 Chhillar S, Barak A, Malik S. Analysis of a Parallel System with Priority to Repair over Maintenance Subject to Random Shocks. International Journal of Computer Science Issues (IJCSI). 2013, 10.

8 Sureria J, Malik S, Anand J. Cost-benefit analysis of a computer system with priority to S/w replacement over H/w repair. Applied Mathematical Sciences. 2012; 6:3723-3734.

9 Malik S, Dhankar A, Kumar A. Cost-Benefit Analysis of a System with Server Failure and Replacement of the Unit at Different Failure Modes Subject to Inspection. International Journal of Computer Applications. 2012; 47:27-33.

10 Kim DS, Lee SM, Jung JH, Kim TH, Lee S, Park JS. Reliability and availability analysis for an on board computer in a satellite system using standby redundancy and rejuvenation. Journal of mechanical science and technology. 2012; 26:2059-2063.

11 Dalal S, Chhillar RS. Case studies of most common and severe types of software system failure. International Journal of Advanced Research in Computer Science and Software Engineering. 2012, 2.

12 Asthana A, Okumoto K. Integrative Software Design for Reliability: Beyond Models and Defect Prediction. Bell Labs Technical Journal. 2012; 17:37-59.

13 Taneja G, Goyal A, Singh D. Reliability and cost-benefit analysis of a system comprising one big unit and two small identical units with priority for operation/repair to big unit. Mathematical Sciences. 2011; 5:235-248.

14 Erkoyuncu JA, Roy R, Shehab E, Cheruvu K. Understanding service uncertainties in industrial product–service system cost estimation. The International Journal of Advanced Manufacturing Technology. 2011; 52:1223-1238.