International Journal of Engineering Research & Management Technology ISSN: 2348-4039
January-2016 Volume 3, Issue-1
Email: editor@ijermt.org www.ijermt.org

Profit and Availability Analysis of Single Module with Perfective Maintenance

Navneet Malik
Research Scholar
Dept. of CSS
Mewar University
Rajasthan

Pardeep Goel
Visiting Faculty
Mewar University
Rajasthan

ABSTRACT:
The paper discusses Profit Analysis of software consisting of a single Module System. This single module can work in full working state and partial working state. This is one module model which can work in reduced state which means the software is having partial features and need corrective, adaptive maintenance to reach the full capacity working state. This model also considers perfective maintenance. It deals with two type of failures: Partially failure and completely failure. This model consists of module 'A'. The module 'A' can fail partially and hence can be in up state. 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. Repairs are unadulterated .The service a time are considered exponential and is different for both type of failures(S/W and H/W). Regenerative Point Graphical Technique (RPGT) is used to calculate different parameters. These parameters are used to measure software system performance.

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

Tables and graph are prepared to represent to study the behavior.

KEY WORDS: Availability, Module, Base-State, Regenerative Point Graphical Technique (RPGT), Busy Period of Repairman, MTSF, Simple paths, Primary Circuits, S/W Failure, H/W Failure, Probability, Reliability, Failed State.

1. INTRODUCTION:

Reliability is defined as the ability of an item to perform a required function, under given environmental and operational condition and for a stated period of time as from ISO8402. Term item can be considered for a system or subsystem and required function may be single function or group of functions. Reliability is the significant parameter due to industry needs and huge operating cost involved in development and maintenance. 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. Maintenance can be considered as a tool for system reliability. J.Endrenyi et al. [1] discussed about different maintenance strategies. They discussed about the cost benefit analysis for applying the maintenance, as maintenance is increased reliability improves

But cost is more. Jim Grey [2] discussed about fault tolerant system. According to this report the major source of system un availability is software as compared to hardware, Gunther A. Hoffman [3] et al. discussed about the preventive measures to increase the software availability, F J Heemstra[4] discussed about pros and cons of different cost estimation models. Sachin Garg [5] et al. discussed about two motivational factors for finding new techniques for software fault tolerance. The first factors is Reliability/Availability Versus cost and second is nature of failure, Barry W.Boehm [6] et al. discussed about the importance to understand and control the software cost, Goel Pradeep and Satpal [7] discussed the availability analysis of dairy plant using RPGT, Gupta Rakesh et al.[8] discussed the reliability and cost benefit analysis for two non identical unit model, Malik S.C. [9] et al. discussed about reliability modeling of computer system by giving priority to H/W repair as compared to S/W up gradation, Gupta Sanjay and Gupta Suresh Kumar [10]discussed the reliability measures using Semi Markov process and RPGT, Sureria J K et al.[11] elaborated the cost benefit analysis of computer system with priority to software development as compared to hardware repair, Erkoyuncu John Ahmad at al. [12] discussed the system cost estimation for service, Asthana Abhaya and Okumoto Kazu [13] elaborated integrated design for software reliability which fill the gap between expected and actual behavior of system, Kim Dong Seong et al.[14] 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, Malik SC et al.[15] discussed the model of reliability for single unit. Three states are considered normal state, partial failure and complete failure, Navneet Malik and Pardeep Goel [16] et al. discussed about the availability of a single module under software and hardware failure, S.C Malik and Jyoti Anand [17] discussed the reliability and economic measures using semi markov process and RPGT.

In this paper the availability analysis of single module software system is done. This model takes the consideration of partial failure and perfective maintenance. Regenerative Point Graphical Technique **(RPGT)** is used to measure the mean time to system failure (MTSF),Availabilty and other key parameters of the software system.

The paper confers the Availability Analysis of a software system consisting of Single Module System, in which module can work in full, reduced and perfective category also. This is a one module model which can work in reduced state which means the software is having partial features and need corrective maintenance for 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' which can work in reduced state after failure. The module 'A' can work partially and hence can be in up state, 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 independent for both type of failure i.e. software and hardware failure. Using Regenerative Point Graphical Technique **(RPGT)**, the following software system features have been assessed to study the system performance.

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

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

2. CONVENTIONS AND SYMBOLIZATIONS:

The following conventions and symbols are used:

1) The software contains single module which can work in reduced state after partial failure but cannot work in completely failed state.

2) The module 'A' can work partially and hence can be in up state and this state is considered to be corrective maintenance category to make the system according to requirement of user. The software can work with reduced capacity in a partially failed state.
3) Single server facility is there for completely failed state and reduced capacity state of any module.
4) Repair facility never does any damage to the software.
5) A repaired module works like a new-one.
6) The software is down if module is in completely failed state. Replica of modules can be used to up the software.
7) When the software system is in failed state due to H/W failure it can not fail further due to S/W failure. H/W failure is replaced with new one.
8) The software is discussed for steady state conditions.
9) The spreading of repair time and failure time are exponential.
10) The repair time and failure time are independent for S/W and H/W failure.
11) The software can be considered for the corrective maintenance state also.

pr/p : 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 represents 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.

$\left(x \xrightarrow{s_r} y\right)$: 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.

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

$V_{m,m}$: 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 represents 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

μ_x : It represents the mean time to stay in state x, before moving to any other states;

$$\mu_x = \int_0^\infty R_x(t)dt.$$

μ_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

η_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.

λ_1/λ_2 : It represents the constant failure rate of the module 'A' to a partially failed state/ from partially failed state to a totally failed state.

λ : 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' 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' 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.

A/A'/a/AP : Module 'A' in full capacity working/ partially failed state/completely failed state/perfective maintenance state.

A(H) : Module 'A' failure due to H/W failure from full capacity working state.

A'(H) : Module 'A' failure due to H/W failure from reduced capacity working state.

AP(H) : Failure due to H/W failure from perfective working state.

AP : Module 'A' working in perfective state.

Using above notations system can be considered in the following regenerative states:

S_0 = A
S_1 = A(H)
S_2=A'
S_3 = a
S_4 = A'(H)
S_5 = AP(H)
S_6 = AP

3. TRANSITION DIAGRAM OF THE SYSTEM:

Using above assumption and symbols given in table 1, transition diagram is given in fig.1.

Table – 1

State	Symbol
Regenerative state	◉
Up-state	○
Failed state	□
Reduced state	⬭
Perfective state	△

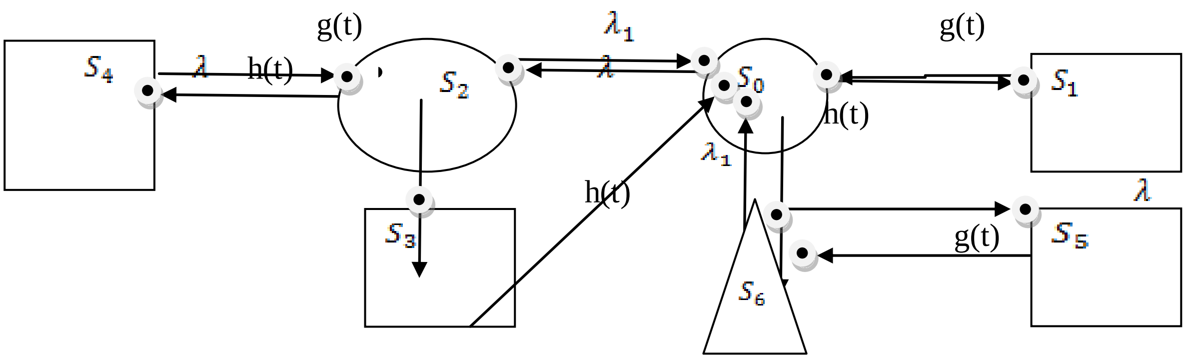

Fig. 1

4. EVALUATION OF PARAMETERS OF THE SYSTEM:

The key parameters of the software system are calculated by finding the 'base-state' and applying RPGT. The MTSF is determined w.r.t. the initial state '0' and the other parameters are obtained by using base-state. Table 2 presents the transition probabilities of the model.

4.1 MEAN STAY TIMES AND TRANSITION PROBABILITIES:

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.

$\mathbf{p_{x,y} = q^*_{x,y}(0)}$; where the symbol * denotes Laplace transformation.

Table - 2

$q_{x,y}(t)$	$p_{x,y} = q^*_{x,y}(0)$
$q_{0,1}(t) = \lambda e^{-(\lambda+\lambda_1)t}.\bar{H}(t)$ $q_{0,2}(t) = \lambda_1 e^{-(\lambda+\lambda_1)t}.\bar{H}(t)$ $q_{0,6}(t) = h(t)e^{-(\lambda+\lambda_1)t}$	$p_{0,1}(t) = \frac{\lambda}{\lambda+\lambda_1}(1-h^*(\lambda+\lambda_1))$ $p_{0,2}(t) = \frac{\lambda_1}{\lambda+\lambda_1}(1-h^*(\lambda+\lambda_1))$ $p_{0,6} = h^*(\lambda+\lambda_1)$
$q_{1,0}(t) = g(t)$	$p_{1,0} = g^*(0)$
$q_{2,0}(t) = h(t)e^{-(\lambda+\lambda_2)t}$ $q_{2,3}(t) = \lambda_2 e^{-(\lambda+\lambda_2)t}\bar{H}(t)$ $q_{2,4}(t) = \lambda e^{-(\lambda+\lambda_2)t}\bar{H}(t)$	$p_{2,0} = h^*(\lambda+\lambda_2)$ $p_{2,3}(t) = \frac{\lambda_2}{\lambda+\lambda_2}(1-h^*(\lambda+\lambda_2))$ $p_{2,4}(t) = \frac{\lambda}{\lambda+\lambda_2}(1-h^*(\lambda+\lambda_2))$
$q_{3,0} = h(t)$	$p_{3,0}(t) = h^*(0) = 1$

$q_{4,2}(t) = g(t)$	$p_{4,2}(t) = g^*(0)$=1
$q_{5,6}(t) = g(t)$	$p_{5,6}(t) = g^*(0)$ =1
$q_{6,0}(t) = \lambda_1 e^{-(\lambda+\lambda_1)t}$ $q_{6,5}(t) = \lambda e^{-(\lambda+\lambda_1)t}$	$p_{6,0}(t) = \frac{\lambda_1}{\lambda+\lambda_1}$ $p_{6,5}(t) = \frac{\lambda}{\lambda+\lambda_1}$

MEAN STAY TIMES:

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

μ_x :mean stay time spent in state x, before visiting any other states;

$$\mu_x = \int_0^\infty R_x(t)dt = R_x^*(0).$$

TABLE 3 PRESENTS THE RELIABILITY CALCULATION.

Table - 3

$\mathbf{R_x(t)}$	$\boldsymbol{\mu_x = R_x^*(0)}$
$R_0(t) = e^{-(\lambda+\lambda_1)t}H(t)$	$\mu_0 = \frac{1-h^*(\lambda+\lambda_1)}{\lambda+\lambda_1}$
$R_1(t) = \bar{G}(t)$	$\mu_1 = -g^{*\prime}(0)$
$R_2(t) = e^{-(\lambda_2+\lambda)t}H(t)$	$\mu_2 = \frac{1-h^*(\lambda_2+\lambda)}{\lambda_2+\lambda}$
$R_3(t)=\bar{H}(t)$	$\mu_3 = -h^{*\prime}(0)$
$R_4(t) = \bar{G}(t)$	$\mu_4 = -g^{*\prime}(0)$
$R_5(t) = \bar{G}(t)$	$\mu_5 = -g^{*\prime}(0)$
$R_6(t) = e^{-(\lambda+\lambda_1)t}$	$\mu_6 = \frac{1}{\lambda+\lambda_1}$

4.3 ESTIMATION OF PARAMETERS:

Regenerative Point Graphical Technique **(RPGT)** is used to calculate the different parameters. '0' is considered as the base-state for these calculations:

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

$V_{0,0} = \left[(0,1,0) + \frac{(0,2,0)}{1-L_1} + \frac{(0,2,3,0)}{(1-L_1)} + \frac{(0,6,0)}{(1-L_2)}\right] = 1$

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

$V_{0,2} = \frac{(0,2)}{1-L_1}$

$V_{0,3} = \frac{(0,2,3)}{1-L_1}$

$V_{0,4} = p_{0,2}p_{2,4}$

$V_{0,5} = p_{0,6}p_{6,5}$

$V_{0,6} = \dfrac{p_{0,6}}{1 - L_2}$

Where
$1 - L_1 = 1 - \{2,4,2\} = 1 - p_{2,4}p_{4,2}$
$1 - L_2 = 1 - \{6,5,6\} = 1 - p_{6,5}p_{5,6}$

(a). MTSF(T_0): It represent 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 = 0,2,6. For 'θ' = '0', MTSF is given by

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

T_0=[(0,0) μ_0+(0,2) $\mu_2 + (0,6)\mu_6$]÷ ([1–(L_1+L_2)])= N ÷ D

Where, L_1=(0,2,0)=$p_{0,2}\,p_{2,0}$ L_2=(0,6,0)=$p_{0,6}\,p_{6,0}$

N=((0,0) μ_0+(0,2) $\mu_2 + (0,6)\mu_6$

D = ([1–(L_1+L_2)])

(B). 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 = 0,2,6. Numbers of regenerative states are 0 to 6. For 'θ' = '0', the total span of time for which the system is available is given by:

A_0=

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

$A_0 = \left[\sum_y V_{\theta,y}\,.f_y.\mu_y\right] \div \left[\sum_x V_{\theta,x}\,.\mu_x^1\right]$

A_0=

$$\left[V_{0,0}\,.f_0.\mu_0 + V_{0,2}\,.f_2.\mu_2 + V_{0,6}\,.f_6.\mu_6\right] \div \left[V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1 + V_{0,5}\mu_5^1 + V_{0,6}\mu_6^1\right]$$

(C). 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 =1,3,4,5.For 'θ' = '0', the total span of time for which the repairman is busy is given by the following equation. Perfective maintenance is considered as a special case. As perfective maintenance is endless process due to which busy period of repairman can not be calculated exactly.

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

B_0=$\left[\sum_y V_{\theta,y}\,.\eta_y\right] \div \left[\sum_x V_{\theta,x}\,.\mu_x^1\right]$

$$\left[V_{0,1}\,.\eta_1 + V_{0,3}.\eta_3 + V_{0,4}.\eta_4 + V_{0,5}.\eta_5\right] \div \left[V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1 + V_{0,5}\mu_5^1 + V_{0,6}\mu_6^1\right]$$

(D). EXPECTED NUMBER OF REPAIRMAN'S VISITS:

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

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

$$V_0=\left[\sum_y V_{\theta,y}\right]\div\left[\sum_x V_{\theta,x}\,.\mu_x^1\right]$$

$$=(V_{0,1}+V_{0,2}+V_{0,6})\div\left[V_{0,0}\mu_0^1+V_{0,1}\mu_1^1+V_{0,2}\mu_2^1+V_{0,3}\mu_3^1+V_{0,4}\mu_4^1+V_{0,5}\mu_5^1+V_{0,6}\mu_6^1\right]$$

5. ANALYTICAL DISCUSSION:

5.1 Availability (A_0) vs. Repair Rate (ω):

The Availability of the software system is calculated for different values of the repair rate by taking ω = 0.80, 0.85, 0.90, 0.95 and 1.0 and different values of the Failure Rate (λ_1) by taking λ_1 = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.01 and The data so obtained are shown in Table 4.

Table-4

λ_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	0.994843	0.995144	0.995411	0.995651	0.995866
0.006	0.993651	0.994020	0.994348	0.994643	0.994908
0.007	0.992457	0.992894	0.993284	0.993633	0.993948
0.008	0.991264	0.991770	0.992220	0.992623	0.992987
0.009	0.990074	0.990647	0.991157	0.991615	0.992028
0.01	0.988887	0.989527	0.990097	0.990609	0.991070

Table 4 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 (λ_1). It is observed that Availability of software system increases with increase in the values of the Repair Rate (ω).

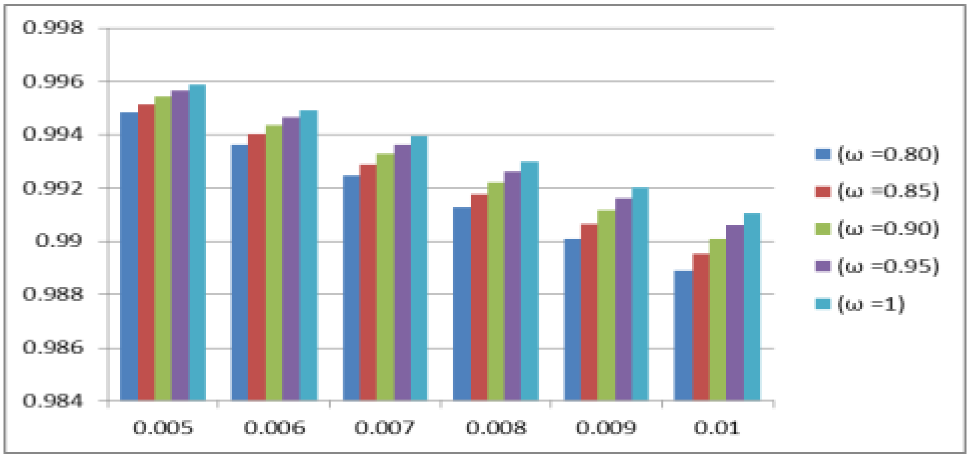

Fig. 3

Further it can be observed from the table and graph 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 (λ_1).

5.2 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 λ= 0.001and λ_2= 0.002. By taking λ_1 = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.010. The data so obtained are shown in Table 5.

Table-5

λ_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	.00582264	.00582326	.00582381	.00582430	.00582474
0.006	.00684395	.00684471	.00684538	.00684599	.00684653
0.007	.00785925	.00786015	.00786095	.00786166	.00786231
0.008	.00887053	.00887157	.00887250	.00887332	.00887407
0.009	.00987897	.00988016	.00988121	.00988215	.00988300
0.01	.01088534	.01088667	.01088785	.01088890	.01088985

5.3 COST CALCULATION:

The net profit can be calculated by the difference of revenue generated and cost involved in number of repairman visits. Table 6 represents the profit calculation.

$P(t) = S_0A_0 - S_1V_0$

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 λ_2= 0.002, λ= 0.001;

Table-6

λ_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	99426	99456	99483	99507	99528
0.006	99297	99334	99366	99396	99422
0.007	99167	99211	99250	99285	99316
0.008	99038	99088	99133	99174	99201
0.009	98909	98966	99017	99063	99104
0.010	98780	98844	98901	98952	98998

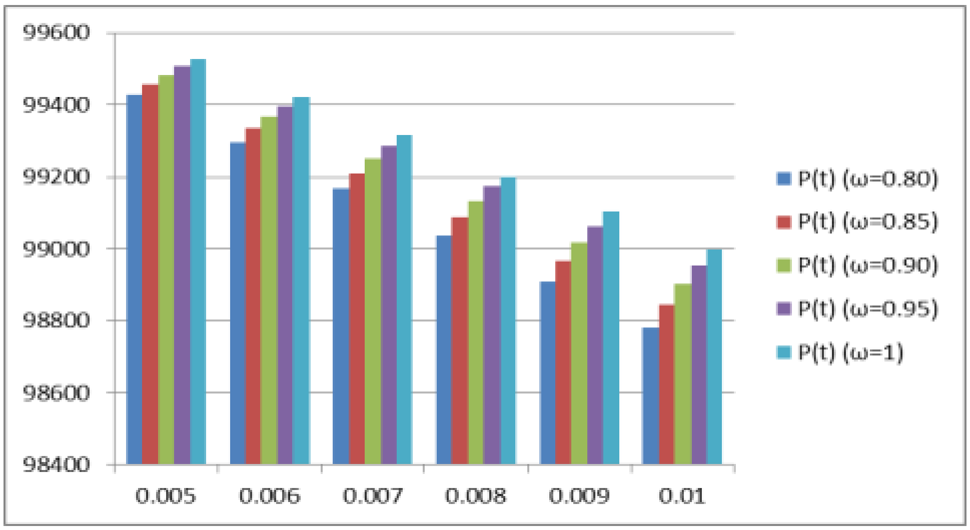

Fig. 4

6. CONCLUSION:

From the Tables and graphs, we see that, as the Repair Rate(ω) increases the Availability of the Software System is increases. The study can be extended for two or more modules software system with perfective maintenance .Profit earned increases with increase in repair rate and decreases with increase in the failure rate of the module. Based on these calculation of profit we can decide for requirement of repairman. In future, parameters can be evaluated by considering Repair rate and Failure rate as a 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.

REFERENCES:

1. J. Endrenyi, S. Aboresheid, R. Allan, G. Anders, S. Asgarpoor, R. Billinton, et al., "The present status of maintenance strategies and the impact of maintenance on reliability," Power Systems, IEEE Transactions on, vol. 16, pp. 638-646, 2001.
2. J. Gray, "A census of Tandem system availability between 1985 and 1990," Reliability, IEEE Transactions on, vol. 39, pp. 409-418, 1990.
3. G. A. Hoffmann, "Advanced Failure Prediction in Complex Software Systems: a Probabilistic Approach," ed, 2005.
4. F. J. Heemstra, "Software cost estimation," Information and software technology, vol. 34, pp. 627-639, 1992.
5. S. Garg, A. Puliafito, M. Telek, and T. Trivedi, "Analysis of preventive maintenance in transactions based software systems," Computers, IEEE Transactions on, vol. 47, pp. 96-107, 1998.
6. B. W. Boehm, "Understanding and controlling software costs," Journal of Parametrics, vol. 8, pp. 32-68, 1988.
7. P. Goel, "Availability Analysis of a Dairy Plant With Priority in Repair Using RPGT," 2014.

8. R. Gupta, S. Kujal, and M. Mahi, "A two dissimilar unit parallel system with two phase repair by skilled and ordinary repairmen," International Journal of System Assurance Engineering and Management, vol. 5, pp. 554-561, 2014.
9. S. Malik, "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, vol. 4, pp. 94-100, 2013.
10. [S. Gupta and S. K. Gupta, "STOCHASTIC ANALYSIS OF A RELIABILITY MODEL OF ONE-UNIT SYSTEM WITH POST INSPECTION, POST REPAIR, PREVENTIVE MAINTENANCE AND REPLACEMENT," IJMERR, vol. 2, pp. 178-188, 2013.
11. J. Sureria, S. Malik, and J. Anand, "Cost-benefit analysis of a computer system with priority to S/w replacement over H/w repair," Applied Mathematical Sciences, vol. 6, pp. 3723-3734, 2012.
12. J. A. Erkoyuncu, R. Roy, E. Shehab, and K. Cheruvu, "Understanding service uncertainties in industrial product–service system cost estimation," The International Journal of Advanced Manufacturing Technology, vol. 52, pp. 1223-1238, 2011.
13. A. Asthana and K. Okumoto, "Integrative Software Design for Reliability: Beyond Models and Defect Prediction," Bell Labs Technical Journal, vol. 17, pp. 37-59, 2012.
14. D. S. Kim, S. M. Lee, J.-H. Jung, T. H. Kim, S. Lee, and J. S. Park, "Reliability and availability analysis for an on board computer in a satellite system using standby redundancy and rejuvenation," Journal of mechanical science and technology, vol. 26, pp. 2059-2063, 2012.
15. S. Malik, A. Dhankar, and A. Kumar, "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, vol. 47, pp. 27-33, 2012.
16. P. G. Navneet Malik, "Behavioral Analysis of Single Module under Software and Hardware Failure using Regenerative Point Graphical Technique," INTERNATIONAL JOURNAL OF INFORMATIVE AND FUTURISTIC RESEARCH (IJIFR), vol. 1, 2013.
17. S. Malik and J. Anand, "Reliability and economic analysis of a computer system with independent hardware and software failures," Bulletin of Pure & Applied Sciences-Mathematics and Statistics, vol. 29, pp. 141-154, 2010.