

Jain, Madhu; Meena, Rakesh Kumar

Article

Fault tolerant system with imperfect coverage, reboot and server vacation

Journal of Industrial Engineering International

Provided in Cooperation with:

Islamic Azad University (IAU), Tehran

Suggested Citation: Jain, Madhu; Meena, Rakesh Kumar (2017) : Fault tolerant system with imperfect coverage, reboot and server vacation, Journal of Industrial Engineering International, ISSN 2251-712X, Springer, Heidelberg, Vol. 13, Iss. 2, pp. 171-180, <https://doi.org/10.1007/s40092-016-0180-8>

This Version is available at:

<https://hdl.handle.net/10419/172559>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>

Fault tolerant system with imperfect coverage, reboot and server vacation

Madhu Jain¹ · Rakesh Kumar Meena¹

Received: 27 October 2014 / Accepted: 11 December 2016 / Published online: 26 December 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract This study is concerned with the performance modeling of a fault tolerant system consisting of operating units supported by a combination of warm and cold spares. The on-line as well as warm standby units are subject to failures and are send for the repair to a repair facility having single repairman which is prone to failure. If the failed unit is not detected, the system enters into an unsafe state from which it is cleared by the reboot and recovery action. The server is allowed to go for vacation if there is no failed unit present in the system. Markov model is developed to obtain the transient probabilities associated with the system states. Runge–Kutta method is used to evaluate the system state probabilities and queueing measures. To explore the sensitivity and cost associated with the system, numerical simulation is conducted.

Keywords Fault tolerant system · Server vacation · Machine repair · Queue length · Imperfect coverage · Reboot · Unreliable server · Runge–Kutta method

Introduction

The performance of any fault tolerant system such as computer or communication system, power plant or transmission lines, manufacturing or production and many more, is highly affected by the failure of its operating units. The breakdowns of the units may cause not only loss of production but also increase in cost. To avoid this loss and

extra cost, the organization or industry can make the provision of standbys and repair facility. In some cases, if the operating unit fails, the system remains operative and continues to perform its assigned job due to switching over of failed unit by the spare part. For the smooth functioning and to achieve desired availability of the concerned system, the concepts of redundancy and maintainability have drawn the attention of practitioners as well as researchers. To reduce the maintainability and operating cost, the provision of server vacation is a key feature which has been included in many queueing models to analyze the congestion problems in different contexts. The machine repair problems with server vacations have been investigated and applied extensively in many areas including the computer and communication networks, manufacturing and inventory control processes, transportation and service sectors, etc. In many queueing scenarios of machining systems, the server is allowed to go for vacation if the system is empty. A noticeable amount of recent past works on the server vacation concept has been done by many eminent researchers (cf. Doshi 1986). Gupta (1997) proposed several server vacation schemes such as multiple vacations, single vacations, hybrid multiple/single vacation for the machine interference problems. Machine repair models with different type of policies for the server vacation have been investigated and studied by a few researchers (cf. Ke and Wang 2007; Ke et al. 2011; Ke and Wu 2012). Jain and Upadhyaya (2009) investigated the performance of multi-component machining system with multiple vacations of the servers, multiple types of redundant components and operating under N-policy. They obtained the probabilities for the system states and various key metrics by implementing the matrix recursive method. The queue size distribution has been established by Kumar and Jain (2013) for both F-policy and N-policy by employing the

✉ Rakesh Kumar Meena
rakeshmeena3424@gmail.com

¹ Indian Institute of Technology Roorkee, Roorkee,
Uttarakhand, India

recursive method. Recently, Jain et al. (2015) developed multi-component machine repair model with two heterogeneous servers and having the facility of warm standby units along with on-line units. They derived the steady state queue size distributions and other measures of performance with the implementation of successive over relaxation technique.

The timings of reboot operation may vary from a few seconds to long hours, depending upon the complexity of the machining system. In many industries, an extensive loss of production as well as cost occurs due to the failure of some malicious components if not tackled properly with the help of suitable mechanism. But in some practical situations, the fault handling device may prove inadequate to recover a fault perfectly. These types of situations are called as imperfect coverage. In literature, some research works can be found on the reliability analysis of the machine repair problems with imperfect fault coverage. In this context, we cite some recent works which are relevant to present investigation. The concept of the reboot was discussed by Trivedi (2008) for the analysis of some reliability models in his book on ‘Probability and Statistics with Reliability, Queueing and Computer Science’. Ke et al. (2008) has done the performance analysis of a repairable system by including the features of detection, imperfect coverage and reboot. A statistical model for a standby system involving reboot, switch failure and unreliable repair was presented by Hsu et al. (2011). The queueing and reliability indices of the machine repair systems with imperfect coverage and reboot have been studied by Jain et al. (2012), Jain (2013), Jain and Gupta (2013), Wang et al. (2013), and many more. Further, Ke and Liu (2014) investigated the machine repair system with imperfect coverage incorporating the reboot delay concept by taking the illustration of gamma and exponential time distributions.

In the present scenario, we notice that most of the studies devoted to the queueing analysis of server vacation are restricted to the reliable server queueing model, but in real life, no server can be reliable as such the incorporation of an unreliable server concept will be helpful to portray the more versatile queueing scenarios. In this direction, a few researchers have contributed by developing Markovian models with unreliable and vacation servers. Ke et al. (2009) developed a Markovian model having a finite buffer of the multi-server queueing system in which the servers are unreliable and allowed to go for vacation on the basis of (d, c) vacation policy. Further, Ke et al. (2013) presented the queueing analysis of unreliable multi-repairmen machining system comprising of operating machines and warm spares by incorporating the concept of imperfect coverage and reboot action. They determined the queue size distributions by employing the matrix recursive approach.

Further, Jain et al. (2014) analyzed N-policy model for the multi-component machining system under the realistic features of imperfect coverage, reboot and unreliable server. Yang and Wu (2015) investigated the N-policy for M/M/1 working vacation queueing model by considering the server breakdowns. They employed the particle swarm optimization algorithm to optimize the cost function and determined the optimal parameters.

Among various available soft computing techniques, neuro-fuzzy technique which is a hybrid of fuzzy logic and neural network, has widely used for the performance of complex systems for which analytical formulae can not be framed. By using the feature of a neural network and fuzzy inference system, some researchers have developed the adaptive neuro-fuzzy inference system (ANFIS) controller for the performance analysis of various embedded systems in different frameworks (cf. Lin and Liu 2003; Yang and Zhao 2012). For the performance prediction of degraded multi-component system with standby switching failure, N-policy and multiple vacations, Jain and Kumar (2013) used ANFIS to match the soft computing based results with the results obtained numerically using successive over relaxation (SOR).

In the present investigation, we provide the performance indices of the machining system supported by a repair facility and mixed standby units operating under vacation policy. A few research papers on the machine repair problem with vacation policy in different frameworks have appeared in the past few years as mentioned earlier. But to the best of authors’ knowledge, no research article explores the transient study of the machine repair problem combined with vacation, mixed standbys, imperfect coverage and server breakdown. Further, the implementation of ANFIS technique to match soft computing results with the analytic results makes our study to deal with complex dynamic behavior of the machining system in efficient computational manner by incorporating many realistic features. The remaining contents of the paper are structured in different sections. In section. “[Model description](#)”, we provide the notations and assumptions to formulate Markov model. In section “[Governing equations](#)”, Chapman-Kolmogorov equations are constructed for the transient state to develop Markov model which are further solved numerically based on the Runge–Kutta method. In section “[Performance indices](#)”, some performance indices have been established explicitly by using the transient probabilities of the system states. In section “[Cost function](#)”, the cost function is constructed. In section “[Numerical illustration](#)”, numerical illustration and sensitivity analysis are provided. The outcomes of the numerical results are displayed in graphs and tables to explore the effect of system descriptors on the performance indices. Finally, the conclusion is drawn in section “[Conclusion](#)”.



Model description

In this section, we develop a Markov queueing model by defining the appropriate transition rates of the concerned birth–death process for the performance analysis of fault tolerant system. The assumptions and notations used for developing the model are as follows:

- The system consists of M operating and S_n ($1 \leq n \leq l$) of n th type standby units.
- The operating units are prone to failure and have the life time exponentially distributed with mean $1/\lambda$. When an operating unit fails, it is immediately backed up by an available standby unit. When a standby unit moves into an operating state, it has the same failure characteristic as that of an operating on-line unit.
- The n th type standby unit may also fail; the life time of n th type unit follows the exponential distribution with mean $1/\alpha_n$, ($0 \leq \alpha_n \leq \lambda$).
- When all the spares are used, the system operates in short (i.e., degraded) mode with at least m ($< N$) operating units and its failure rate becomes λ_d ($> \lambda$).
- The failed units are repaired by the repairman in the same order in which failure occur, i.e., repair is performed by following the FIFO service discipline. The repair of failed units is rendered by the server according to exponentially distribution with rate μ .
- The operating unit can be successfully recovered with probability c ; the recovery time of operating unit is exponentially distributed with parameter σ .
- The server is allowed to go for vacation if there is no work load of repair job of failed units in the system and returns back from the vacation as soon as any unit fails.

To develop the model, Markov process for the three mutually exclusive system states, i.e., (1) operating state, (2) recovery state and (3) reboot state is taken into account. The transient probabilities of the failed units at time t for different states are defined for the scenarios when the server is in (1) vacation state, (2) busy state, and (3) broken down state. Let $P_{i,j,k}(t)$ denote the probability that there are i , ($1 \leq i \leq L$) failed units when the server is in j th ($j = 0, 1, 2$) state and the system is operating in k th ($k = 0, 1, 2$) mode. Here indices $j = 0, 1, 2$ when the server is on vacation, busy and broken down states, respectively. Also indices $k = 0, 1, 2$ represent the system status in operating state, recovery state and reboot state, respectively.

The failure rate of the operating units which depends upon the number of already failed units is given by:

$$\lambda_i = \begin{cases} M\lambda & i < S^{(l)} \\ (M + S^{(l)} - i)\lambda_d & S^{(l)} \leq i < L \end{cases}, \quad (1)$$

where $S^{(l)} = \sum_{n=1}^l S^{(n)}$ and λ_d is the degraded failure rate.

The failure rate of the standby units is given by:

$$\alpha_i = \begin{cases} \sum_{x=2}^l S_x v_x + (S_1 - i)v_1, & i < S^{(x)} \\ \sum_{y=x+1}^l S_y v_y + (S^{(x)} - i)v_x, & S^{(x-1)} \leq i < S^{(x)} \\ (S^{(l)} - i)v_l, & S^{(l-1)} \leq i < S^{(l)} \\ 0, & S^{(l)} \leq i < L. \end{cases} \quad (2)$$

$x = 2, 3, \dots, l-1$

Governing equations

For evaluating the probabilities associated with different server states, Kolmogorov Chapman equations have been constructed using the transition flow rates of birth death process specifying the Markov model. The state transition for in-flow and out-flow rates of specific model when $M = 5$, $S_1 = 2$, $S_2 = 1$, $l = 2$, $m = 1$ is shown in Fig. 1.

1. Server vacation state when $j = 0$, $k = 0$:

As soon as the server becomes free when there is no job of repairing the failed machines in the system, it reaches to vacation state (0, 0, 0). In this case, the server is in vacation state and reaches to other state using appropriate transition rates. Now, we frame the Chapman-Kolmogorov equation for state (0, 0, 0) as follows:

$$\frac{dP_{0,0,0}(t)}{dt} = -(\lambda_0 + \alpha_0)P_{0,0,0}(t) + \mu P_{1,1,0}(t). \quad (3)$$

The server returns back to busy state with rate θ when a failed machine joins the system; in between, some more machines may fail so that during vacation period, the system states may be $(i, 0, 0)$, $i = 1, 2, \dots, L$. Using appropriate in-flow and out-flow rates, we formulate the governing equations for $(i, 0, 0)$, $i = 1, 2, \dots, L$ as follows:

$$\begin{aligned} \frac{dP_{i,0,0}(t)}{dt} = & -(\lambda_i + \theta + \alpha_i)P_{i,0,0}(t) + \alpha_{i-1}P_{i-1,0,0}(t) \\ & + \sigma P_{i-1,0,1}(t) + \beta P_{i-1,0,2}(t); \quad 1 \leq i \leq S^{(l)} - 1, \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{dP_{S^{(l)},0,0}(t)}{dt} = & -(\lambda_{S^{(l)}} + \theta)P_{S^{(l)},0,0}(t) + \alpha_{S^{(l)}-1}P_{S^{(l)}-1,0,0}(t) \\ & + \sigma P_{S^{(l)}-1,0,1}(t) + \beta P_{S^{(l)}-1,0,2}(t), \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{dP_{i,0,0}(t)}{dt} = & -(\lambda_i + \theta)P_{i,0,0}(t) + \sigma P_{i-1,0,1}(t) \\ & + \beta P_{i-1,0,2}(t); \quad S^{(l)} + 1 \leq i \leq L - 1, \end{aligned} \quad (6)$$



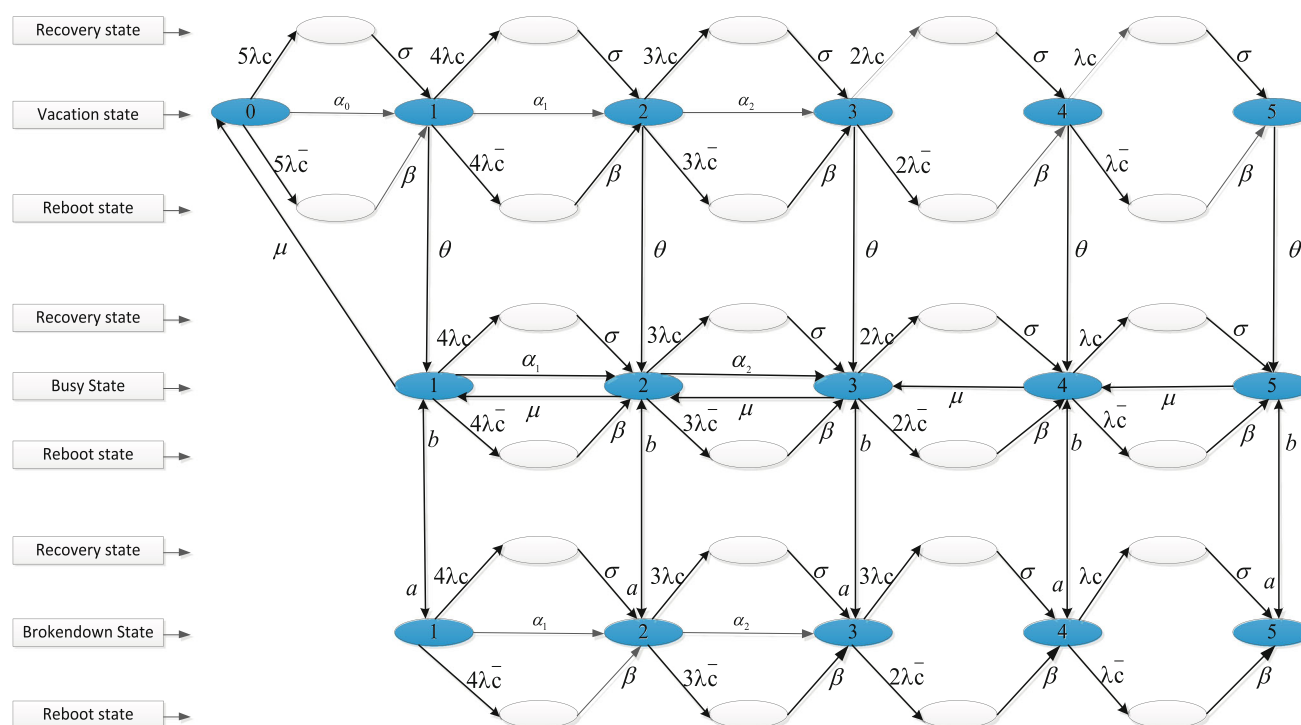


Fig. 1 Transition state diagram

$$\frac{dP_{L,0,0}(t)}{dt} = -\theta P_{L,0,0}(t) + \sigma P_{L-1,0,1}(t) + \beta P_{L-1,0,2}(t). \quad (7)$$

2. Server being in busy state when $j = 1, k = 0$:

When the server is busy in providing repair of the failed machines, the transient equations are framed using appropriate transition rates for states $(i, 1, 0), i = 1, 2, \dots, L$ as follows:

$$\frac{dP_{1,1,0}(t)}{dt} = -(\lambda_1 + a + \mu + \alpha_1)P_{1,1,0}(t) + \theta P_{1,0,0}(t) + \mu P_{2,1,0}(t) + b P_{1,2,0}(t), \quad (8)$$

$$\begin{aligned} \frac{dP_{i,1,0}(t)}{dt} = & -(\lambda_i + a + \mu + \alpha_i)P_{i,1,0}(t) + \theta P_{i,0,0}(t) \\ & + \mu P_{i+1,1,0}(t) + b P_{i,2,0}(t) + \alpha_{i-1} P_{i-1,1,0}(t) \\ & + \sigma P_{i-1,1,1}(t) + \beta P_{i-1,1,2}(t); \quad 1 < i \leq S^{(l)} - 1, \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{dP_{S^{(l)},1,0}(t)}{dt} = & -(\lambda_{S^{(l)}} + a + \mu)P_{S^{(l)},1,0}(t) + \theta P_{S^{(l)},0,0}(t) \\ & + \mu P_{S^{(l)}+1,1,0}(t) + b P_{S^{(l)},2,0}(t) + \alpha_{S^{(l)}-1} P_{S^{(l)}-1,1,0}(t) \\ & + \sigma P_{S^{(l)}-1,1,1}(t) + \beta P_{S^{(l)}-1,1,2}(t), \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{dP_{i,1,0}(t)}{dt} = & -(\lambda_i + a + \mu)P_{i,1,0}(t) + \theta P_{i,0,0}(t) \\ & + \mu P_{i+1,1,0}(t) + b P_{i,2,0}(t) + \sigma P_{i-1,1,1}(t) \\ & + \beta P_{i-1,1,2}(t); \quad S^{(l)} + 1 \leq i \leq L - 1, \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{dP_{L,1,0}(t)}{dt} = & -(\mu + a)P_{L,1,0}(t) + \theta P_{L,0,0}(t) + b P_{L,2,0}(t) \\ & + \sigma P_{L-1,1,1}(t) + \beta P_{L-1,1,2}(t). \end{aligned} \quad (12)$$

3. The failed server is under repair state when $j = 2, k = 0$:

In this case the server is broken down and the repairman is performing the repair job to restore it. Now for states $(i, 2, 0), i = 1, 2, \dots, L$, the transient equations are framed by law of conservation of flows as follows:

$$\frac{dP_{1,2,0}(t)}{dt} = -(\lambda_1 + b + \alpha_1)P_{1,2,0}(t) + a P_{1,1,0}(t), \quad (13)$$

$$\begin{aligned} \frac{dP_{i,2,0}(t)}{dt} = & -(\lambda_i + b + \alpha_i)P_{i,2,0}(t) + a P_{i,1,0}(t) \\ & + \sigma P_{i-1,2,1}(t) + \alpha_{i-1} P_{i-1,2,0}(t) \\ & + \beta P_{i-1,2,2}(t); \quad 1 < i \leq S^{(l)} - 1, \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{dP_{S^{(l)},2,0}(t)}{dt} = & -(\lambda_{S^{(l)}} + b)P_{S^{(l)},2,0}(t) + aP_{S^{(l)},1,0}(t) \\ & + \sigma P_{S^{(l)}-1,2,1}(t) + \beta P_{S^{(l)}-1,2,2}(t) \\ & + \alpha_{S^{(l)}-1}P_{S^{(l)},2,0}(t), \end{aligned} \quad (15)$$

$$\begin{aligned} \frac{dP_{i,2,0}(t)}{dt} = & -(\lambda_i + b)P_{i,2,0}(t) + aP_{i,1,0}(t) + \sigma P_{i-1,2,1}(t) \\ & + \beta P_{i-1,2,2}(t); \quad S^{(l)} + 1 \leq i \leq L-1, \end{aligned} \quad (16)$$

$$\begin{aligned} \frac{dP_{L,2,0}(t)}{dt} = & -bP_{L,2,0}(t) + aP_{L,1,0}(t) + \sigma P_{L-1,2,1}(t) \\ & + \beta P_{L-1,2,2}(t). \end{aligned} \quad (17)$$

4. For $k = 1, j = 0, 1, 2$, when the system is in recovery state:

From $(i, j, 0)$ state, due to perfect failure detection, the system can go to recovery state $(i, j, 1)$, for $i = 1, 2, \dots, L-1; j = 0, 1, 2$. For the recovery states, the transient equations are framed as:

$$\begin{aligned} \frac{dP_{ij,1}(t)}{dt} = & -\sigma P_{ij,1}(t) + \lambda_i c P_{ij,0}(t); \quad j \\ = & 0, 1, 2; \quad i = 1, 2, \dots, L-1. \end{aligned} \quad (18)$$

5. For $k = 2, j = 0, 1, 2$, when the system is in reboot state:

From $(i, j, 0)$ state, due to imperfect failure detection, the system can go to reboot state $(i, j, 2)$, for $i = 1, 2, \dots, L-1; j = 0, 1, 2$. For these state, the transient equations are constructed as:

$$\begin{aligned} \frac{dP_{ij,2}(t)}{dt} = & -\beta P_{ij,2}(t) + \lambda_i \bar{c} P_{ij,0}(t); \\ j = & 0, 1, 2; \quad i = 1, 2, \dots, L-1. \end{aligned} \quad (19)$$

The Eqs. (3)–(19) have been solved numerically using Runge–Kutta 4th order method, which is a powerful tool to solve the ordinary differential equations of first order. It is a good choice to employ this technique to solve the set of differential equations governing the system state probabilities. It is worth noting the R–K method is quite accurate, stable and easy to implement in comparison to other methods available to solve the differential equations. For the coding purpose, we have chosen this particular method here and MATLAB's 'ode45' function is exploited for the programming purpose.

Performance indices

To analyze the transient system behavior, we derive various performance indices using the probabilities which can be evaluated as described in previous section. The expressions for the expected number of failed units in the

system, failure frequency of the system, availability of the server and the system state probabilities for the server being in different states and other performance metrics are established as follows:

1. The average number of failed units in the system at time t , is:

$$\begin{aligned} E[N(t)] = & \sum_{i=1}^L \sum_{j=0}^2 i P_{ij,0}(t) \\ & + \sum_{i=1}^L \sum_{j=0}^2 i \{P_{ij,1}(t) + P_{ij,2}(t)\}. \end{aligned} \quad (20)$$

2. Failure frequency of the server at time t , is:

$$f(t) = a \sum_{i=1}^L P_{i,2,0}(t). \quad (21)$$

3. System availability of the at time t , is:

$$\begin{aligned} A(t) = & 1 \\ & - \left(\sum_{i=0}^L P_{i,2,0}(t) + \sum_{i=1}^{L-1} \{P_{i,2,1}(t) + P_{i,2,2}(t)\} \right). \end{aligned} \quad (22)$$

4. The transient probability that the system is in recovery state:

$$P_{RC}(t) = \sum_{i=0}^{L-1} P_{i,0,1}(t) + \sum_{i=1}^{L-1} P_{i,1,1}(t) + \sum_{i=1}^{L-1} P_{i,2,1}(t). \quad (23)$$

5. The transient probability that the system is in reboot state:

$$P_R(t) = \sum_{i=0}^{L-1} P_{i,0,2}(t) + \sum_{i=1}^{L-1} P_{i,1,2}(t) + \sum_{i=1}^{L-1} P_{i,2,2}(t). \quad (24)$$

6. The transient probability that the server is in broken down state:

$$P_{BD}(t) = \sum_{i=1}^L P_{i,2,0}(t) + \sum_{i=1}^{L-1} P_{i,2,1}(t) + \sum_{i=1}^{L-1} P_{i,2,2}(t). \quad (25)$$

7. The transient probability that the server is in busy state:

$$P_B(t) = \sum_{i=1}^L P_{i,1,0}(t) + \sum_{i=1}^{L-1} P_{i,1,1}(t) + \sum_{i=1}^{L-1} P_{i,1,2}(t). \quad (26)$$

8. The transient probability that the server being on vacation state:



$$P_V(t) = \sum_{i=0}^L P_{i,1,0}(t) + \sum_{i=1}^{L-1} P_{i,1,1}(t) + \sum_{i=1}^{L-1} P_{i,1,2}(t). \quad (27)$$

Cost function

The system designer may be interested to determine the proper combination of the spares and repairmen so as the total cost incurred on the system should be minimum. While making such decisions, it should be kept in mind that the organization should not bear the burden of excessive costs of keeping the spares and repairmen. To determine the optimal number of repairmen and spare machines, we construct the cost function by considering various cost elements involved in different activities. We denote the various cost elements incurred on different activities as follows:

- C_V Cost per unit time of the server when he is on vacation;
- C_V Cost per unit time of the server when he is in busy state;
- C_H Holding cost of one failed unit per unit time in the system;
- C_{BD} Cost of repairing of a broken down server per unit time

Now we evaluate the total cost function by considering the above cost elements and respective performance measures as follows:

$$C(t) = C_V P_V(t) + C_B P_B(t) + C_H E[N(t)] + C_{BD} P_{BD}(t). \quad (28)$$

Numerical illustration

To reveal the practical applicability of the underlying model in real time machining system, we consider an illustration from flexible manufacturing systems where robots are used for the packing purpose. In normal functioning mode, the system requires five robots, however, the system can work in degraded mode, i.e., in short mode if there are less than five but at least one operating robot is in active state. The system has two types of standby robots which acts as back up unit in case of failure of any operating unit and immediately put in place of broken down robot. The failure rate of the operating robot is $\lambda = 0.1$ robots per day. To maintain the desired level of availability, three standby robots having failure rate $\alpha_0 = 0.7$, $\alpha_1 = 0.4$, and $\alpha_2 = 0.2$ robots per day and one robot which cannot fail in active mode, are taken as warm and cold standby units, respectively. Whenever a robot

fails, its failure is detected, diagnosed and recovered with probability $c = 0.5$ and recovery rate is assumed to be $\sigma = 0.8$. If fault is not detected due to imperfect coverage, it is cleared by the reboot or reset operation with a rate $\beta = 10$ per day. To evaluate the performance results, coding of the program is done in MATLAB software. The subroutine ode45 is employed for solving the set of equations associated with the probabilities of different system states. For the computation of results, we fixed the various parameters as:

$$\begin{aligned} \text{Cost Set : } C_V &= \$10, C_B = \$100, C_H = \$20, C_{BD} = \$200 \\ M &= 5, S_1 = 2, S_2 = 1, l = 3, m = 1, c = 0.5, \\ a &= 0.02, b = 1, \mu = 2, \lambda = 0.1, \alpha_0 = 0.07, \\ \alpha_1 &= 0.04, \alpha_2 = 0.02, \sigma = 0.8, \beta = 10, v = 0.02. \end{aligned}$$

The sensitivity analysis has been done to explore the effect of varying parameters on different performance indices. The numerical results displayed in the form of tables and graphs are quite easy to understand the behavior of system. The numerical results obtained have been displayed in graphs and tables.

The numerical results for various performance indices for varying values of different parameters are presented in Tables 1, 2, 3. The effects of different parameters are examined by displaying the numerical results for the availability and average number of failed units in the system in Figs. 2, 3, 4, 5, 6, 7. From Figs. 2, 3, 4, 5, we observe that as time increases, the system availability initially decreases rapidly and then after becomes almost constant. In the case of curves of $E\{N(t)\}$ plotted in Figs. 6, 7, sharp increment is noticed up to $t = 20$, then after it becomes asymptotically stable as time passes.

The trends of various performance indices for varying different parameters are as follows:

1. Effect of failure rate of operating unit and repairman (λ, a): from Figs. 2 and 3, it is noted that the availability of the system decreases as λ and a increase. From Table 1, it is clear that the average number of failed units increases as λ increases.
2. Effect of reboot rate and service rate (β, μ): Table 2 displays that the average number of failed units $E\{N(t)\}$ exhibits the increasing trend as reboot rate β increases; on the contrary, the availability $A(t)$ of the system decreases as reboot rate β increases. It is seen in Fig. 5 that as μ increases, the availability $A(t)$ of the system increases while $E\{N(t)\}$ decreases.
3. Effect of repair rate of repairman and coverage factor (b, c): From Figs. 4 and 7, it is noticed that the availability $A(t)$ of the system increases as repair rate (b) increases. The average number of failed machines



Table 1 Effect of failure rate of operating unit (λ) on various performance indices

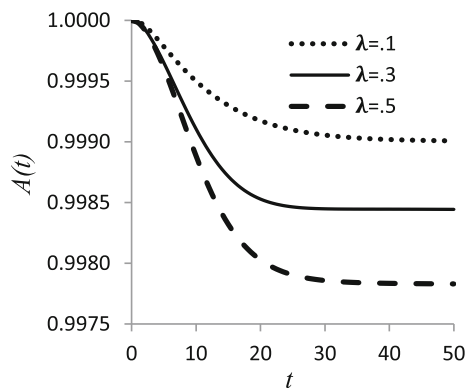
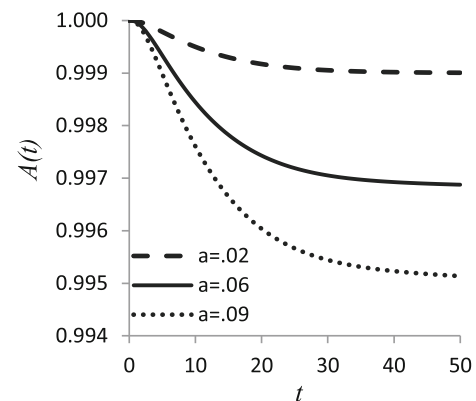
λ	t	$P_{RC}(t)$	$P_R(t)$	$P_{BD}(t)$	$EN(t)$	$f(t)$	$A(t)$	$P_B(t)$	$C(t)$
0.1	10	0.13	0.0094	0.00051	2.75	0.00049	0.99949	0.0289	67.6
	30	0.06	0.0045	0.00095	3.96	0.00085	0.99905	0.0486	93.9
	50	0.05	0.0041	0.00100	4.07	0.00088	0.99900	0.0506	96.2
0.3	10	0.14	0.0093	0.00090	4.08	0.00079	0.99910	0.0545	96.7
	30	0.07	0.0057	0.00156	4.41	0.00119	0.99844	0.0836	106.0
	50	0.07	0.0058	0.00156	4.41	0.00119	0.99844	0.0837	106.2
0.5	10	0.12	0.0073	0.00112	4.47	0.00096	0.99888	0.0731	106.1
	30	0.09	0.0074	0.00215	4.43	0.00152	0.99785	0.1238	110.2
	50	0.09	0.0075	0.00218	4.43	0.00153	0.99782	0.1250	110.3

Table 2 Effect of reboot rate (β) on various performance indices

β	t	$P_{RC}(t)$	$P_R(t)$	$P_{BD}(t)$	$EN(t)$	$f(t)$	$A(t)$	$P_B(t)$	$C(t)$
2	10	0.1255	0.0478	0.00049	2.6	0.00045	0.99951	0.0278	65.5
	30	0.0561	0.0222	0.00097	3.9	0.00083	0.99903	0.0494	93.3
	50	0.0501	0.0200	0.00102	4.0	0.00087	0.99898	0.0517	95.6
6	10	0.1266	0.0157	0.00051	2.7	0.00049	0.99949	0.0287	67.3
	30	0.0567	0.0074	0.00096	4.0	0.00084	0.99904	0.0488	93.8
	50	0.0509	0.0068	0.00100	4.1	0.00088	0.99900	0.0508	96.1
10	10	0.1268	0.0094	0.00051	2.7	0.00049	0.99949	0.0289	67.6
	30	0.0569	0.0045	0.00095	4.0	0.00085	0.99905	0.0486	93.9
	50	0.0510	0.0041	0.00100	4.1	0.00088	0.99900	0.0506	96.2

Table 3 Effect of failure rate of standby unit (α) on various performance indices

α	t	$P_{RC}(t)$	$P_R(t)$	$P_{BD}(t)$	$EN(t)$	$f(t)$	$A(t)$	$P_B(t)$	$C(t)$
0.03	10	0.127	0.0094	0.00051	2.75	0.00049	0.99949	0.0289	67.64
	30	0.057	0.0045	0.00095	3.96	0.00085	0.99905	0.0486	93.90
	50	0.051	0.0041	0.00100	4.07	0.00088	0.99900	0.0506	96.23
0.06	10	0.120	0.0088	0.00055	2.89	0.00053	0.99945	0.0310	70.79
	30	0.054	0.0043	0.00098	4.02	0.00087	0.99902	0.0498	95.03
	50	0.049	0.0039	0.00102	4.11	0.00090	0.99898	0.0517	97.18
0.09	10	0.114	0.0084	0.00059	3.02	0.00056	0.99941	0.0329	73.41
	30	0.052	0.0041	0.00100	4.06	0.00089	0.99900	0.0510	95.97
	50	0.047	0.0038	0.00104	4.15	0.00092	0.99896	0.0527	97.98

**Fig. 2** Variation in $A(t)$ for different values of λ **Fig. 3** Variation in $A(t)$ for different values of α

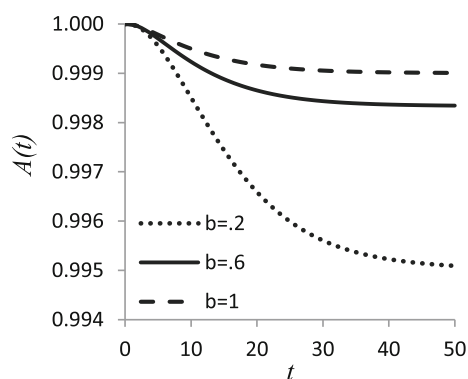


Fig. 4 Variation in $A(t)$ for different values of b

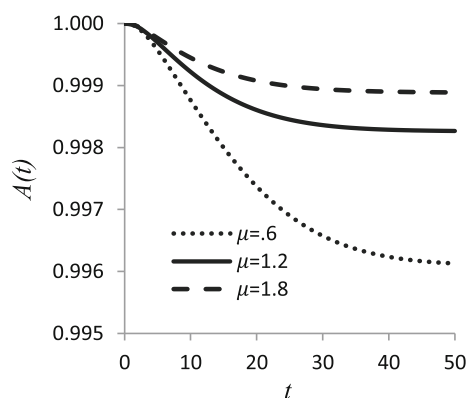


Fig. 5 Variation in $A(t)$ for different values of μ

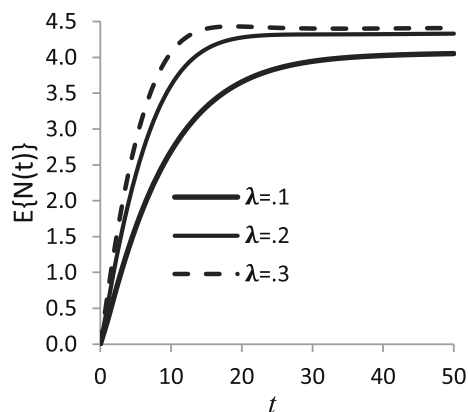


Fig. 6 Variation in $E\{N(t)\}$ for different values of λ

$E\{N(t)\}$ reveals the increasing trend as the coverage factor c increases.

Neuro-fuzzy technique is applied to compute the performance indices of the fault tolerant MRP. The membership function of input parameters λ , a and b are taken as trapezoidal function by taking the very low, low, average, high and very high values as depicted in Fig. 8. The numerical results by ANFIS approach have been computed by using the neuro-fuzzy tool in Matlab software. To

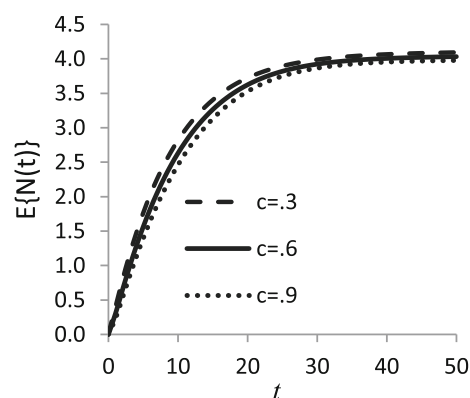


Fig. 7 Variation in $E\{N(t)\}$ for different values of c

facilitate the comparison of results obtained by Runge–Kutta method with neuro-fuzzy results, we plot the availability by both approaches in Figs. 9, 10, 11.

The sensitivity of availability obtained by R–K method is depicted by continuous line for varying parameters λ , a and b . The numerical results obtained using neuro-fuzzy technique are shown by tick marks. From these figures, we notice that both R–K and ANFIS results are quite close to each other.

Conclusion

In this paper, we have investigated the performance of fault tolerant system supported by mixed standbys and repair facility. Some realistic concepts such as reboot, recovery, unreliable server and vacation are incorporated in order to develop generalized queueing model so as to analyze the queueing and reliability characteristics of many real time fault tolerant systems. The transient probabilities of the system states and other performance measures evaluated

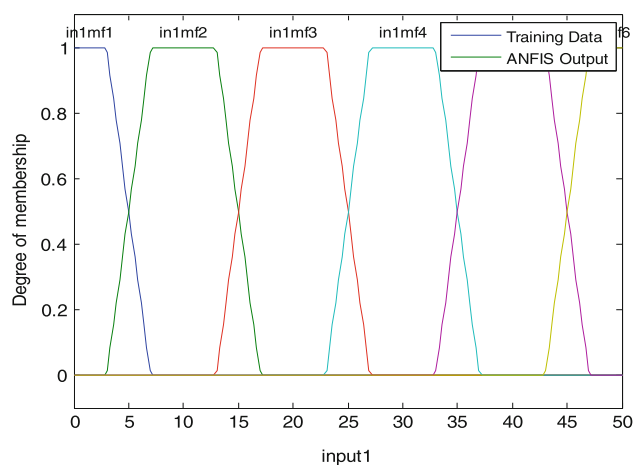


Fig. 8 Membership function for λ , a and b

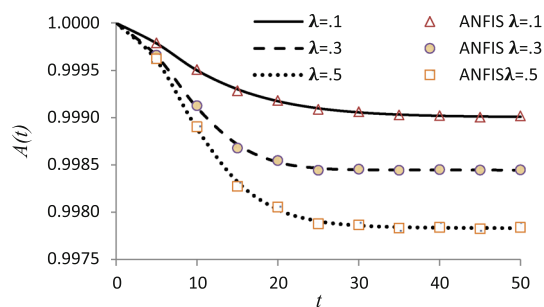


Fig. 9 $A(t)$ vs t for different values of λ

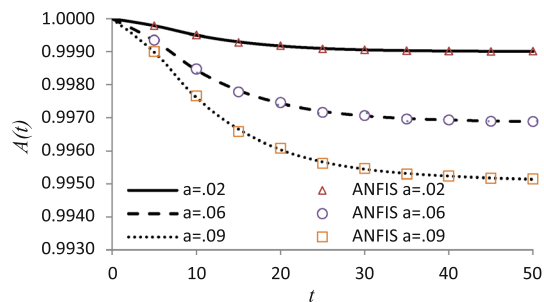


Fig. 10 $A(t)$ vs t for different values of a

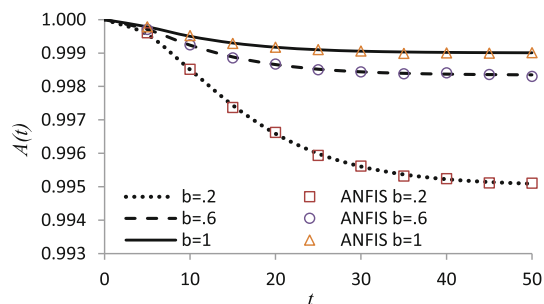


Fig. 11 $A(t)$ vs t for different values of b

using Runge–Kutta method can be easily implemented for the evaluation of cost function and to determine the optimal control parameters for many redundant systems such as embedded computer and communication systems, power plants, automated manufacturing systems, etc. The sensitivity analysis carried out may be helpful to the decision makers and system engineers for the further improvement of the concerned systems operating under machining environments. There is scope of further extension of present investigation by incorporating the correlated rates and/or bulk failure and work is under process in that direction.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a

link to the Creative Commons license, and indicate if changes were made.

References

- Doshi BT (1986) Queueing systems with vacations—a survey. *Queueing Syst Theory Appl* 1:29–66
- Gupta SM (1997) Machine interference problem with warm spares, server vacations and exhaustive service. *Perform Eval* 29:195–211
- Hsu YL, Ke JC, Liu TH (2011) Standby system with general repair, reboot delay, switching failure and unreliable repair facility—a statistical standpoint. *Math Comput Simul* 81:2400–2413
- Jain M (2013) Availability prediction of imperfect fault coverage system with reboot and common cause failure. *Int J Oper Res* 17:374–397
- Jain M, Gupta R (2013) Optimal replacement policy for a repairable system with multiple vacations and imperfect fault coverage. *Comput Ind Eng* 66:710–719
- Jain M, Kumar K (2013) Threshold N- policy for (M, m) degraded machining system with k- heterogeneous servers, standby switching failure and multiple vacation. *Int J Math Oper Res* 5:423–445
- Jain M, Upadhyaya S (2009) Threshold N-policy for degraded machining system with multiple type spares and multiple vacations. *Qual Technol Quant Manag* 6:185–203
- Jain M, Agrawal SC, Naresh P (2012) Fuzzy reliability evaluation of repairable system with imperfect coverage, reboot and common cause shock failure. *Int J Eng* 25:231–238
- Jain M, Shekhar C, Rani V (2014) N- policy for a multi—component machining system with imperfect coverage, reboot and unreliable server. *Prod Manuf Res* 2:457–476
- Jain M, Mittal R, Kumari R (2015) (m, M) Machining system with two unreliable servers, mixed spares and common cause-failure. *J Ind Eng Int* 11:171–178
- Ke JC, Liu TH (2014) A repairable system with imperfect coverage and reboot. *Appl Math Comput* 246:148–158
- Ke JC, Wang KH (2007) Vacation policies for machine repair problem with two type spares. *Appl Math Model* 31:880–894
- Ke JC, Wu CH (2012) Multi-server machine repair model with standbys and synchronous multiple vacation. *Comput Ind Eng* 62:296–305
- Ke JC, Lee SL, Hsu YL (2008) On repairable system with detection, imperfect coverage and reboot: Bayesian approach. *Simul Model Pract Theory* 16:353–367
- Ke JC, Lin CH, Yang JY, Zhang ZG (2009) Optimal (d, c) vacation policy for a finite buffer M/M/c queue with unreliable server and repairs. *Appl Math Model* 33:3949–3962
- Ke JC, Wu CH, Liou CH, Wang TY (2011) Cost analysis of a vacation repair model. *Procedia—Social Behav Sci* 25:246–256
- Ke JC, Hsu YL, Liu TH, Zhang ZG (2013) Computational analysis of machine repair problem with unreliable multi-repairmen. *Comput Oper Res* 40:848–855
- Kumar K, Jain M (2013) Threshold F-policy and N-policy for multi-component machining system with warm standbys. *J Ind Eng Int* 9:1–9
- Lin ZC, Liu CY (2003) Analysis and application of the adaptive neuro-fuzzy inference system in prediction of CMP machining parameters. *Int J Comput Appl Technol* 17:80–89
- Trivedi KS (2008) Probability and statistics with reliability, queueing, and computer science applications, 2nd edn. Wiley, New York
- Wang KH, Liou CD, Lin YH (2013) Comparative analysis of the machine repair problem with imperfect coverage and service pressure condition. *Appl Math Model* 37:2870–2880



- Yang DY, Wu CH (2015) Cost-minimization analysis of a working vacation queue with N-policy and server breakdowns. *Comput Ind Eng* 82:151–158
- Yang Y, Zhao Q (2012) Machine vibration prediction using ANFIS and wavelet packet decomposition. *Int J Model Ident Control* 15:219–226

