Int J Syst Assur Eng Manag (December 2023) 14(6):2218–2234
https://doi.org/10.1007/s13198-023-02055-3

ORIGINAL ARTICLE

A reward-based performability modelling of a fault-tolerant safety–critical system

Shakeel Ahamad[1] · Ratneshwer Gupta[1]

Received: 19 April 2023 / Revised: 22 June 2023 / Accepted: 19 July 2023 / Published online: 3 August 2023

Abstract Nowadays, various computer system carries out critical functions. The failure of these systems leads to unacceptable loss. Such systems are called Safety–Critical Systems (SCS). The Performance and Reliability of SCS should be high. So, the combined study of performance and reliability (called Performability) is an important issue. The testing of the system is also used to improve its performance. However, some issues might not be addressed in the testing procedure. Formal verification is used for developing secure software. In most of the research work, performability is obtained by operational systems or fail repair systems. Some studies have considered the fail-repair, including fault-tolerant systems. Safety–critical systems generally have fault-tolerant mechanisms to minimize the severity of the failure. This paper studies the safety–critical system's performability using the continuous-time Markov chain (CTMC) with a reward called the Markov reward model (MRM), keeping in mind the fail-repair, fault-tolerant characteristics of the systems. The various parameters of the performability have been analyzed. For mathematical calculation, python language is used. The case study illustrates the proposed approach.

Keywords Markov model · Markov reward model · Performability · Safety–Critical systems · Performance and reliability · Multi-states systems

✉ Shakeel Ahamad
ahamadshakeel608@gmail.com

Ratneshwer Gupta
ratnesh@mail.jnu.ac.in

[1] Software Quality Assurance Lab, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

1 Introduction

Many modern computers play a significant role in carrying out critical functionality. Failure of these systems leads to a massive loss of money, time, environment, and human life (John 2002). The SCSs are used in various domains like medical systems, finance systems, nuclear power plants, aviation, etc. (Pietrantuono and Russo 2013). The SCS include software and hardware, the essential factors in managing the high-level performance and reliability of the system. Performance and reliability are antagonists in nature (Tokuno and Yamada 2009). The combined study of reliability and performance is called performability (Mo et al. 2018). The performability analysis depends on various parameters that affect the reliability and performance of the system. During the analysis, one must formally understand the interrelationship of different parameters and their possible effects. So, the combined study of performance and reliability is essential to optimize the overall quality.

The quality assurance of such a system is essential for safety–critical systems. Quality is the umbrella term which is used in the various domain. The meaning of the quality in the software is a fitness of purpose. That explains the software system is working according to the Software system requirement (SRS). Fitness of the purpose is not only that represents the quality. There are various other parameters that show the quality of the system, like portability, maintainability, reusability, reliability, performance etc.

The safety–critical systems generally consist of various components. The failure of some parts does not lead to the collapse of the entire system (Singh et al. 2012). So, the safety–critical system works with the degraded system quality. So, the system has various operational states other than the extreme states (completely failed or operational). Such types of systems can be modelled using stochastic modelling. There are various stochastic models like the Petri net, Markov model, etc. These stochastic models efficiently handle performance and dependability-related issues (Gokhale et al. 2004). Markov model is of various types, i.e., discrete-time Markov model, continuous-time Markov model, semi-Markov, and hidden Markov model, based on the system orientation and design. The Markov reward method extends the Markov chain(Kwon and Agha 2007). The reward can be any effect (loss, cost, penalty), and it can be positive or negative.

Most researchers have used the Continuous-time Markov chain and Discrete-time Markov chain without reward to find the reliability or availability(Goel and Okumoto 1979; Trivedi et al. 2003; Wang 2004). The combined study of the performance and reliability needs more attention for the safety–critical system (Viktorova et al. 2018). This paper presents the continuous-time Markov reward model with the reward for the systems' performability analysis. In MRM, the reward matrices are changed according to the parameter under measurement. The accumulated reward on the particular state shows the measured parameter value. So, the different type of reward matrix for the different performability measure has been derived. The proposed methodology for the performability parameter analysis has been illustrated using the case study.

The paper is organized as follows: The related work has presented in Sect. 2. A brief introduction of the Markov chain with reward is given in Sect. 3. The performability parameters used for measuring performability are listed in Sect. 4. Section 5 presents the methodology for the Markov reward modelling for calculating the performability parameters. The step taken for calculating the system's performability is given in Sect. 6. Section 7 illustrates the case study to demonstrate performability measurement. Finally, Sect. 8 presents the future work and concludes the paper.

2 Related work

In the current scenario, various systems carry out critical functionality. So, the system's proper functioning mostly depends on the software, as the software is prone to errors and malfunctioning. There is a need to analyze the safety–critical system for the non-functional parameter properly. Reliability and performance are the most critical quality requirement of the system. The improvement of any one parameter may not be sufficient, e.g., in a fire alarm, If the system is highly reliable but has deficient performance. That leads to severe damage to the resource because the alarm signal is not timely. The same problem is also faced if high performance and reliability are low. So, the combined study of performance and reliability is significant and challenging for the researchers.

There is a large amount of research to find, predict, and estimate performance and reliability. Earlier, probability distributions are used considering the system's only binary state (up, down). But in recent times, the system is more complex and multitasking. So, the system has various other states than up and down states. That means if some component fails, it does not mean that the system completely fails. But the system work with degraded quality. Recently, the multistate stochastic model has determined performance and dependability parameters.

In the paper (Viktorova et al. 2018), a complex system is used to study performance, reliability, and performability. In this model, only the reliability-related parameter of the system is calculated. The distributed computer system's reliability and the number of failures are studied by Jin-Long Wang (Wang 2004) using the discrete Markov chain. The program's reliability runs on the particular terminal, and the overall system reliability is calculated. The directed graph is used to represent the distributed system that is taken for the case study. The two reliability parameters are used to calculate the system's reliability. One is Markov-chain distributed program reliability (MDPR) which calculates the reliability of the particular program of the distributed system. The second is Markov-chain distributed system reliability (MDSR), representing overall system reliability.

Lisnianski also derives the system's reliability with different capacities and demands. This paper finds the consolidated performance, but there is a lack of information about how much time is spent on different levels. The general approach is suggested to compute commonly used reliability measures(Lisnianski 2007). The general Markov reward model has been built according to the approach so that the corresponding reward matrix determination can calculate different reliability measures. The performability model for the wireless network is presented in the paper (Trivedi et al. 2003). The Erlang loss model creates composite and hierarchical Markov chains to derive loss formulas for a system with channel failures. The theory of queuing is used to create these formulas. For the blocking probability in a loss system (i.e., no waiting room in the system, that mean number of servers is equal to the number of customers), use the Erlang-B formula. For the wait probability in a delay system, use the Erlang-C formula. A reversible Markov process can be used to simulate a network of queues, as in the multidimensional Erlang-B formula for the blocking probability in a loss system with several classes of calls and various server occupations. In (Goel and Okumoto 1979) papers, a Markovian model has been presented for software error failures that are not removed, e.g., imperfect debugging. A compositional method for estimating the software reliability of the multithreads program is developed in the paper (Kwon and Agha 2007). The reliability is calculated based on the reliability of the individual component and the transition among them.

Only accurate data can be used to extract information for Condition-Based Maintenance (CBM), which is based on sensors(Martins et al. 2023).

Many approaches are used for the performability analysis of critical safety systems. But the stochastic method is efficient for measuring, predicting, and estimating the performability of the complex system. Various researchers used the Markov model for the individual parameter (reliability availability calculation). This paper presents an approach based on the combined study of reliability and performance using the Markov reward model. The major contribution of this research work is to analyze the performance and reliability of the software systems combined manner that perform the critical functionality. This study also helps in reducing the probability of a failure system and helps in taking decisions for enhancing performability. Because the Performability analysis is performed in the early stage of the system development, so, decisions can easily be incorporated into the systems.

3 Markov chain/process with reward

Markov chain is the stochastic model that shows all the possible events of the system (Norris 1998). Markov process has transitions and states. The states represent the possible system condition. The transition represents the events that carry the system from one state to another. The state's transition from one state to another depends on the current state. The state transition history to reach the present state does not affect the next transition. This property is known as the memoryless property (Mikosch and Kallenberg 1998). The Markov process's state space is the set of all possible states. The transition matrix shows the probability of the transition. The sum of the probability of each row is equal to one. There are two types of the Markov chain based on time, viz. discrete-time Markov chain (DTMC) and continuous-time Markov chain (CTMC). A statistical method known as a hidden Markov model (HMM) proceeds through a number of states that are 'hidden' from the observer (Sotelo et al. 2023).

The Markov chain can efficiently represent a real-world multistate stochastic problem (Bas 2019) if the time spent on the state (sojourn time) does not follow the exponential distribution. Then this Markov chain is called the semi-Markov chain. One interesting fact is that knowledge must not decide which state to enter next, although it remembers how long the current state has lasted. The stochastic model can be used for collective and individual performance and reliability.

The Markov reward method extends the Markov chain(Kwon and Agha 2007). The reward can be any effect, e.g., loss, cost, penalty, etc. A reward can be positive or negative. Using the reward in the Markov model for the analysis is called the Markov reward model (MRM). In the MRM, there is a reward associated with each state. The reward variable shows the stated reward up to the time t. Using the Markov models with reward to measure the performability parameters is more efficient than without reward.

4 A markov model representation of a system under study

The Markov model for a system has states and transitions among the states. The transition among the state is based on some rate parameters (failure rate and repair rate)(Karlin and M. Taylor 1975). A system and its different states are designed to demonstrate the failure and repair scenario with the help of the Markov model. A circle denotes the states of the system, and arrows denote the transition path between states, as illustrated in Fig. 1. Only the failure transitions are used for the reliability-related measure, while failure and repair transitions are used for the availability and performance measurement. Let suppose λ is the failure rate and the μ is the repair rate for the system. X(t), t ≥ 0 represents the system's state at t. The quality measurement of a system depends on the quality of its sub-systems. So, the system's quality can function of the states at time t (Mo et al. 2018). This case assumes that the system has a dual redundant fault-tolerant system. State p shows the primary component, r state shows the redundant, and b state shows the backup state. A state diagram of a system is drawn below to show the system's different failure and repair scenarios.

Here,

$\mu_{p:}$ the Repair rate of the primary component.

$\mu_{r:}$ the Repair rate of the redundant component.

$\mu_{b:}$ the Repair rate of the backup component.

λ_r: The failure rate of the primary component.

λ_r: The failure rate of the redundant component.

λ_b: The failure rate of the backup component.

The transition from state i to j takes place at some transition rates. In this study, two transition rates have been considered, i.e., failure rate λ and repair rate μ. These rates can also be represented using the Matrix.

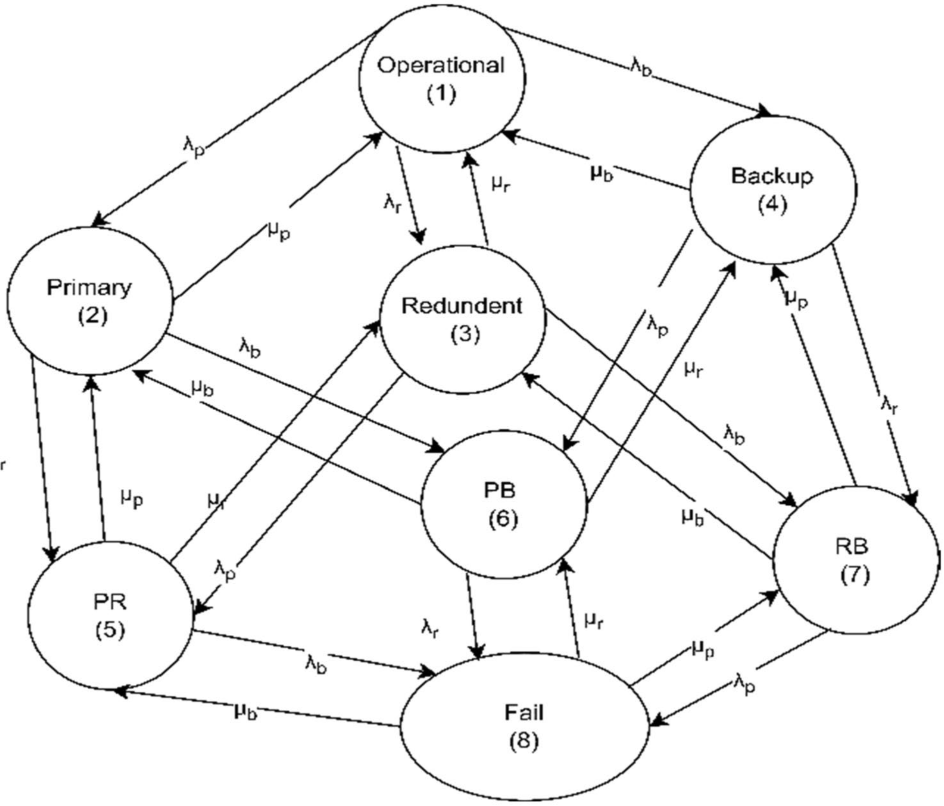
$\Lambda = \left[a_{i,j}\right]$, Λ is the $n * n$ Matrix and $a_{i,i} = -\sum_{j,j \neq i} a_{i,j}$

For the Continuous-Time Markov Chain, the diagonal element of the matric is the sum of the rest row elements with a minus sign. A specific cost is accumulated for each state while staying and transitioning to another state.

$W = \left[w_{ij}\right]_{n*n}$.

The reward matric depends on the quality parameter under measurement. The reward (w_{ii}) is the state's reward when there is no transition to the same state. The reward (w_{ij}) denotes the reward for the transition from i to j state. If all the rewards are assigned zero, the MRM is the same as the ordinary Markov chain. The total reward accumulated up

Fig. 1 State diagram of a system with backup and redundancy



to time t under the initial conditions represents the parameters. The transition intensities do not depend on the time t. That means remaining constant for the homogeneous Markov processes.

The next step is calculating the accumulated reward for the state i using the Howard differential equation (A. Howard 1960).

$$\frac{V_i(t)}{dt} = w_{ii} + \sum_{j=1,j\neq i}^{n} a_{ij}w_{ij} + \sum_{j=1}^{n} a_{ij}V_j(t), i = 1\dots\dots n \qquad (1)$$

Here,

$V_i(t)$ Denote the accumulated reward at time t at state i.

w_{ii} is the reward for state i

w_{ij} is the transition reward from state i to j

The transition and reward matrices are created to calculate the performability parameters. The set of differential equations is written based on Eq. 1.

5 The performability measures used for the study

For the performability analysis, various parameters have been defined in the study of a given system. Generally, performability is defined as a combination of performance and dependability (Mo et al. 2018). Dependability has the four-parameter viz reliability, availability, safety, and security. This paper has taken two dependability parameters (Reliability and Availability). Table 1 shows the study's performability parameters used in this paper.

The activity block diagram is given in Fig. 2. That describes the process of the performability analysis.

6 Design and development of the proposed performability model

For a system designed in Sect. 3 (Fig. 1), a methodology for performability measurement has been proposed in different steps. The model is depicted through the activity diagram, as shown in Fig. 2. Further, the steps are explained in the following subsections. Different dependability and performance measures are used to obtain performability. We used the reward process for different transitions mentioned in the respective reward matrices.

6.1 Reliability

For reliability, the first step is to define the system's reward and transition matrix. Besides, we consider the failed state as an absorbing state, so we delete all arcs in the Markov graph leading from the failed state to the working state (Viktorova et al. 2018). The elements of the reward matrix are as.

Table 1 The performability parameters and their definitions

Performability measure	Definition
Reliability	The probability that a system will produce correct outputs up to time t. (Wang et al. 2017)
Mean reliability	The ratio of operation time to the total time represents mean reliability
Mean No of failure	Number of times the system enters into the failure state in time T(Wang 2004)
Availability	The probability that the system will operate satisfactorily at a given point in time t.(Heddaya and Helal 1996)
Mean availability	Available time is divided by total time. (Lisnianski 2007)
Steady-State Availability	The availability does not change with time. (Viktorova et al. 2018)
Performance	The amount of work accomplished in time t.(Eshragh and Kargahi 2013)
Available Throughput	What is the available throughput during the operation time? (Viktorova et al. 2018)
Various Performance Level time	Various performance outputs like High, Medium, Low, and fail
Performability	A combined study of Performance and Dependability (Smith et al. 1988)

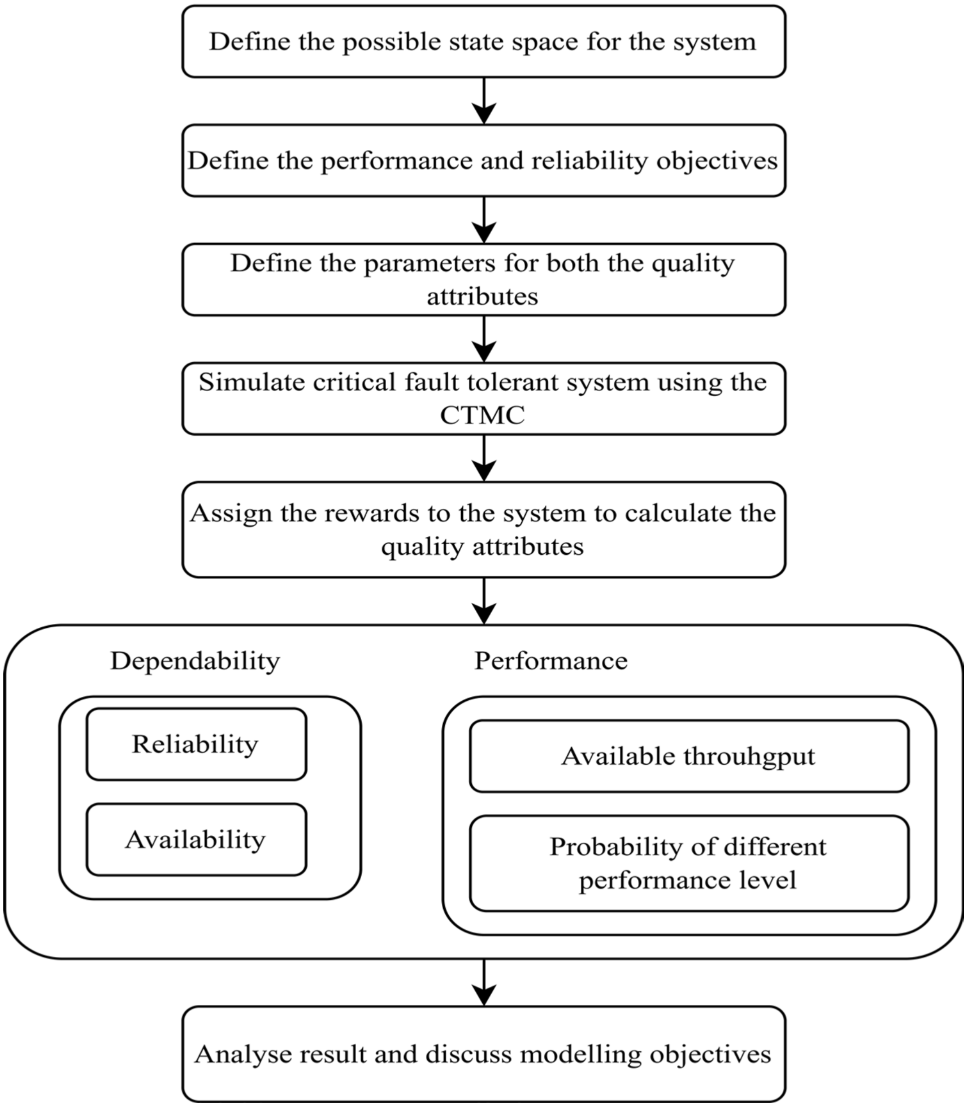

Define the possible state space for the system
Define the performance and reliability objectives
Define the parameters for both the quality attributes
Simulate critical fault tolerant system using the CTMC
Assign the rewards to the system to calculate the quality attributes
Dependability
Performance
Reliability
Availability
Available throuhgput
Probability of different performance level
Analyse result and discuss modelling objectives

Fig. 2 Activity diagram for the performability using continuous-time Markov chain with reward

For the reward matrix $r_{ii} = \begin{cases} 1, \forall i = jandi \in operational \\ 0, otherwise \end{cases}$

The reward matrix, according to the discussion

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For the safety–critical system, reliability should be very high. We consider that if at least two components working is considered acceptable, another failed state. So, the updated transition matrix is as follows.

For solving the differential equation, each state's accumulated reward obtains. At the initial time, we consider that all component is working. So, the accumulated reward at state 1 shows the system's Reliability (Kwon and Agha 2007).

6.2 Mean number of failures

For the Mean Number of failures, elements of the reward matrix are put as one if there is a transition from the operational state to the failure state. The failed state is considered the absorbing state (Viktorova et al. 2018). The differential equation is derived from Eq. (1).

$$\Lambda = \begin{bmatrix} -\left(\lambda_p+\lambda_r+\lambda_b\right) & \lambda_p & \lambda_r & \lambda_b & 0 & 0 & 0 & 0 \\ 0 & -\left(\lambda_r+\lambda_b\right) & 0 & 0 & \lambda_r & \lambda_b & 0 & 0 \\ 0 & 0 & -\left(\lambda_p+\lambda_b\right) & 0 & \lambda_p & 0 & \lambda_b & 0 \\ 0 & 0 & 0 & -\left(\lambda_p+\lambda_r\right) & 0 & \lambda_p & \lambda_r & 0 \\ 0 & 0 & 0 & 0 & -\left(\lambda_b\right) & 0 & 0 & \lambda_b \\ 0 & 0 & 0 & 0 & 0 & -\left(\lambda_r\right) & 0 & \lambda_r \\ 0 & 0 & 0 & 0 & 0 & 0 & -\left(\lambda_p\right) & \lambda_p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The differential equation for the total accumulated reward is written down according to Eq. 1.

$$\begin{aligned} \frac{dV_1(t)}{dt} &= 1-\left(\lambda_p+\lambda_r+\lambda_b\right)V_1(t)+\lambda_pV_2(t)+\lambda_rV_3(t)+\lambda_bV_4(t) \\ \frac{dV_2(t)}{dt} &= 1-\left(\lambda_r+\lambda_b\right)V_2(t)+\lambda_rV_5(t)+\lambda_bV_6(t) \\ \frac{dV_3(t)}{dt} &= 1-\left(\lambda_p+\lambda_b\right)V_3(t)+\lambda_pV_5(t)+\lambda_bV_7(t) \\ \frac{dV_4(t)}{dt} &= 1-\left(\lambda_p+\lambda_r\right)V_4(t)+\lambda_pV_6(t)+\lambda_rV_7(t) \\ \frac{dV_5(t)}{dt} &= -\left(\lambda_b\right)V_5(t)+\lambda_bV_8(t) \\ \frac{dV_6(t)}{dt} &= -\left(\lambda_r\right)V_6(t)+\lambda_rV_8(t) \\ \frac{dV_7(t)}{dt} &= -\left(\lambda_p\right)V_7(t)+\lambda_pV_8(t) \\ \frac{dV_8(t)}{dt} &= 0 \end{aligned} \tag{2}$$

6.3 Availability

Availability is defined as the system is operational at a time instant t. To calculate availability, we must calculate the average accumulated time spent in the operational states during the time interval (0, t) (Lisnianski 2007). The repair rate is also included in the transition matrix.

For the reward matrix $r_{ii} = \begin{cases} 1, \ \forall i = j and i \in operational \\ 0, \ otherwise \end{cases}$

The reward matrix is given below for the availability calculation.

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The transition matrix of the proposed Markov system

$$\Lambda = \begin{bmatrix} -\left(\lambda_p+\lambda_r+\lambda_b\right) & \lambda_p & \lambda_r & \lambda_b & 0 & 0 & 0 & 0 \\ \mu_p & -\left(\lambda_r+\lambda_b+\mu_p\right) & 0 & 0 & \lambda_r & \lambda_b & 0 & 0 \\ \mu_r & 0 & -\left(\lambda_p+\lambda_b+\mu_r\right) & 0 & \lambda_p & 0 & \lambda_b & 0 \\ \mu_b & 0 & 0 & -\left(\lambda_p+\lambda_r+\mu_b\right) & 0 & \lambda_p & \lambda_r & 0 \\ 0 & \mu_r & \mu_p & 0 & -\left(\lambda_b+\mu_p+\mu_r\right) & 0 & 0 & \lambda_b \\ 0 & \mu_b & 0 & \mu_p & 0 & -\left(\lambda_p+\mu_p+\mu_b\right) & 0 & \lambda_r \\ 0 & 0 & \mu_b & \mu_r & 0 & 0 & -\left(\lambda_p+\mu_b+\mu_r\right) & \lambda_p \\ 0 & 0 & 0 & 0 & \mu_b & \mu_r & \mu_p & -(\mu_b+\mu_r+\mu_{p)} \end{bmatrix}$$

The following differential equations were obtained using Eq. 1. That shows the accumulated reward for the availability of each state.

$$
\begin{aligned}
\frac{dV_1(t)}{dt} &= 1 - \left(\lambda_p + \lambda_r + \lambda_b\right)V_1(t) + \lambda_p V_2(t) + \lambda_r V_3(t) + \lambda_b V_4(t)\\
\frac{dV_2(t)}{dt} &= 1 + \mu_p V_1(t) - \left(\lambda_r + \lambda_b + \mu_p\right)V_2(t) + \lambda_r V_5(t) + \lambda_b V_6(t)\\
\frac{dV_3(t)}{dt} &= 1 + \mu_r V_1(t) - \left(\lambda_p + \lambda_b + \mu_r\right)V_3(t) + \lambda_p V_5(t) + \lambda_b V_7(t)\\
\frac{dV_4(t)}{dt} &= 1 + \mu_b V_1(t) - \left(\lambda_p + \lambda_r + \mu_p\right)V_4(t) + \lambda_p V_6(t) + \lambda_r V_7(t)\\
\frac{dV_5(t)}{dt} &= \mu_r V_2(t) + \mu_p V_3(t) - \left(\lambda_b + \mu_r + \mu_p\right)V_5(t) + \lambda_b V_8(t)\\
\frac{dV_6(t)}{dt} &= \mu_b V_2(t) + \mu_p V_4(t) - \left(\lambda_r + \mu_b + \mu_p\right)V_6(t) + \lambda_r V_8(t)\\
\frac{dV_7(t)}{dt} &= \mu_b V_3(t) + \mu_r V_4(t) - \left(\lambda_p + \mu_b + \mu_r\right)V_7(t) + \lambda_p V_8(t)\\
\frac{dV_8(t)}{dt} &= \mu_b V_5(t) + \mu_r V_6(t) + \mu_p V_7(t) - \left(\mu_b + \mu_r + \mu_p\right)(V_8(t)
\end{aligned}
\tag{3}
$$

Solving the above differential equation shows the availability of the system. After solving equations, accumulated reward ($V_i(t)$) is obtained for each state. We consider that all the components are working at the initial time, so the V_1(t) shows the average availability of the system (Viktorova et al. 2018). The average availability is obtained by dividing the total accumulated reward by the time.

The system is considered running for a long time for steady-state availability. At steady-state, system reliability does not change with time. So, the change rate of the differential equation is zero. So, the left-hand side of the differential Eqs. 3 should be set as zero to calculate the availability at the steady state.

6.4 Performance

First, the performance issue of the systems is described. Let's suppose a system has different levels of performance $l = \left\{l_1, l_2, l_3 \ldots l_i \ldots \ldots l_n\right\}$ where the performance level (l_i) is the level of performance of the system at state i. $\{L_i(t) \geq 0, L_i(t) \in l_j\}$ (Goševa-Popstojanova and Trivedi 2000). It is a stochastic process that shows the performance level of the at state i at time t. Let's suppose the Q is the quality associated with each state of the system. The state change with the time quality associated with the condition also changed (Goel and Okumoto 1979). So, the quality is the function of the S(t) (achieved state at time t), i.e., $Q(S(t)) = Q_{cr}(t)$ Or $Q_{cr}(t) = w_i$ The $Q_{cr}(t)$ is the current value of the quality at state (Toledano et al. 2016), or we can say it is a reward at state i. So, the performance level and the demand performance can be represented by the stochastic processes shown in Fig. 3.

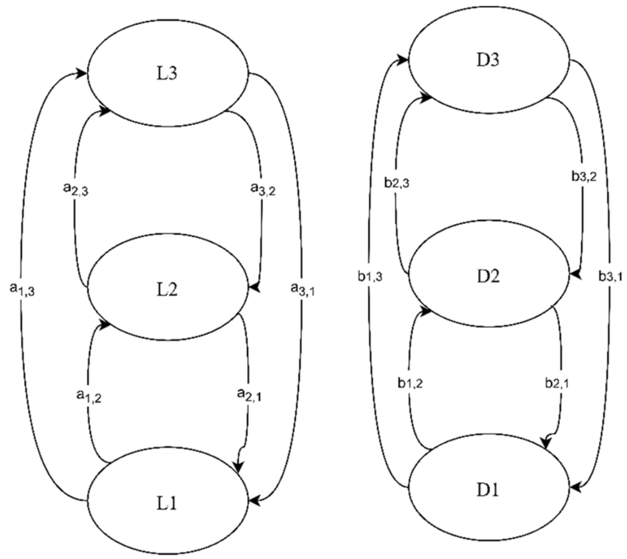


Fig. 3 stochastic processes showing throughput level and demand throughput of the system

Suppose we want to represent the weight for each state and change of weight from state i to j. The Matrix W can define that.

$W = [w_{i,j}]_{n*n}$,

The reward ($w_{i,j}$) is the effect on the system in terms of quality arising from the transition from state i to$j$. Transition state i to i i.e $w_{i,i} = w_i$. Suppose we consider the three performance levels: low, medium, and high. Another stochastic process $D_i(t) \geq 0$ shows the demand performance level at state$i$. For each state, there are two possibilities regarding whether demand may satisfy or not (Temraz and El-Dmcese 2011). The whole state space can be divided into two disjoint sets based on the condition. Demand satisfy states are the acceptable states, and that state not fulfilling the demand is categorized as a failed state. The acceptability function for the state is the function of the level of performance and the demand (Temraz and El-Dmcese 2011).

$$
Q\left\{L_i(t), D_i(t)\right\} = \begin{cases} L_i(t), -D_i(t), \left\{L_i(t) - D_i(t)\right\} \geq 0 \\ \qquad 0, else \end{cases}
$$

In Homogeneous Markov processes, the failure and repair rates remain independent of time t (Strielkina et al. 2018). Let's take the throughput as the demanded throughput as the performance parameter. The demand throughputs (D_1, $D_{2,}$ and D_3) and provided throughputs by the system ($T_{1,}$ $T_{2,}$ and $T_{3)}$ are shown in Fig. 3. Demand and capacity stochastic matrices are merged to get the combined transition matrix shown in Fig. 4.

The following transition matrix is created based on the generating and demand capacity transition matrices. The

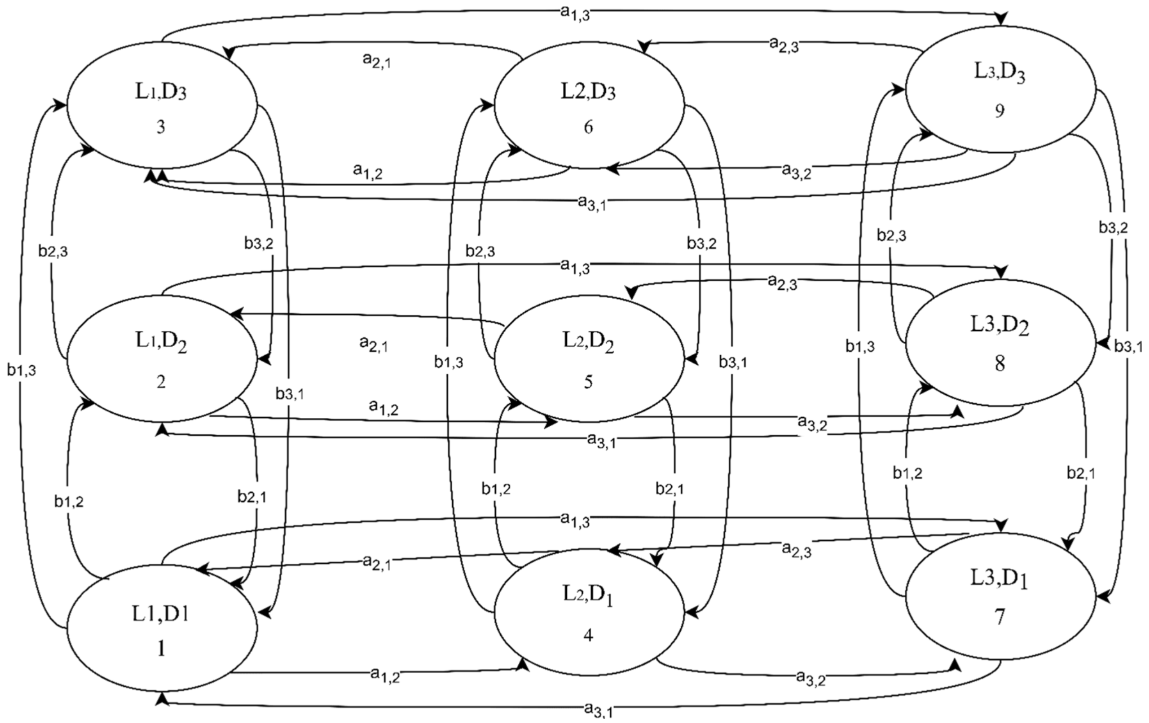


Fig. 4 stochastic process showing the demand and capacity of the system

transition rate assignment is done according to the rule in the paper (Lisnianski 2007). In this, only the horizontal and vertical transition is considered. The diagonal transition is ignored for the simplicity of calculating the transition rate.

$$\Lambda = \begin{bmatrix} -(y_1) & b_{1,2} & b_{1,3} & a_{1,2} & 0 & 0 & a_{1,3} & 0 & 0 \\ b_{2,1} & -(y_2) & b_{2,3} & 0 & a_{1,2} & 0 & 0 & a_{1,3} & 0 \\ b_{3,1} & b_{3,2} & -(y_3) & 0 & 0 & 0 & 0 & 0 & a_{1,3} \\ a_{2,1} & 0 & 0 & -(y_4) & b_{1,2} & b_{1,3} & a_{2,3} & 0 & 0 \\ 0 & a_{2,1} & 0 & b_{2,1} & -(y_5) & b_{2,3} & 0 & a_{2,3} & 0 \\ 0 & 0 & a_{2,1} & b_{3,1} & b_{3,1} & -(y_6) & 0 & 0 & a_{2,3} \\ a_{3,1} & 0 & 0 & a_{3,2} & 0 & 0 & -(y_7) & b_{1,2} & b_{1,3} \\ 0 & a_{3,1} & 0 & 0 & a_{3,2} & 0 & b_{2,1} & -(y_8) & b_{2,3} \\ 0 & 0 & a_{3,1} & 0 & 0 & a_{3,2} & b_{3,1} & b_{3,2} & -(y_9) \end{bmatrix}$$

$$availablethroughput = pr\{\varnothing(t) \geq 0\}$$

For calculating, we have defined the reward for each state. The acceptable states have the reward one. The unacceptable state and transition have the reward of zero. All the rewards associated with each state are determined using Eq. 1. Usually, the state with the greatest capacity level and minimum demand is considered as an initial state (Temraz and El-Dmcese 2011).

$$averageavailablethroughputTH(T) = \sum \frac{V_i(T)}{T}$$

A similar approach can be used for the response time and other performance parameters. Calculating the reward determines how much the system remains at the particular performance level. The reward related to the state with the required throughput or more is defined as 1, otherwise 0.

6.5 Performability

For critical safety systems, performability is very important to ensure that the system's quality is according to specification (Eshragh and Kargahi 2013). In this paper, performability only considers the reliability and performance of the system. Sections 6.1 and 6.2 measure the system's reliability, while subSect. 6.3 measures the combined performance and availability. The parameters measured in the above subsection show performability of the system. The combined parameter is named the available throughput. Another important performability measure is calculated to show the different performance levels of time spent during operations.

To explain how this methodology is beneficial. We must understand that the safety–critical system with high reliability and low performance or low reliability and high performance has no worthwhile performing the critical functionality. Because performance describes the system's responsiveness, and reliability shows how accurately the system produces the result. So, the performance and reliability should be combined to get the optimal solution for software reliability and performance. Second thing, Software security has a direct impact on both its reliability and performance. For example, security lapses may result in downtime, permit the entry of viruses that turn off essential functions, or give criminals access to your data. However, overly complicated security mechanisms can cause the system to lag and prevent integrations with other programs.

7 Case study

For the illustration of the proposed methodology in this work, The case study presented by (Kalaiarasi et al. 2017) is taken for study. The Markov chain graph is shown in Fig. 1. If at least two components out of three-component are in operational mode, then the system is considered operational; otherwise, the system is in a failed state. The system under study has three components; there are eight possible states. There are four operational states, and four are failed states. So, the state set is categorized as the set of failed and operational states. Figure 5 shows all-possible states of the system.

Assumptions

- A system with components is presented using CTMC, where states show operational and state transition indicates the movement to the other states.
- Component failure and repair rates are statistically independent and exponentially distributed.
- Link failures and repairs trigger transitions from one state into another.
- At most, one component can fail or repair at a particular time.
- All the components are working perfectly at the initial time.

The transition rate triggers the transition from one state to another. There are two types of transition rated failure rate and repair rate. The system has finite state spaces$s = \{0, 1, 2, 3, \ldots\ldots, 7\}$. The homogeneous CTMC $(X(t), t \geq 0)$ describes the stochastic nature. Transition rate matrix Λ

After identifying operational and failed states, the Markov transition diagram of the system with non-repairable capability was generated in Fig. 6, removing all the repair rate transitions from Fig. 1. If the two components are upstate, the state is considered the operational state; otherwise, fail state. So, the number of operational states is 4, and the number of failed states is 4. There is an 8*8 transition matrix Λ should be created. The failure rate $\lambda_1 = 0.03 + 0.002 * t$, $\lambda_2 = 0.03 + 0.001 * yand\lambda_3 = 0.01$, the repair rate$\mu_1 = 0.1$, $\mu_2 = 0.1$ And $\mu_3 = 0.05$ (Wang

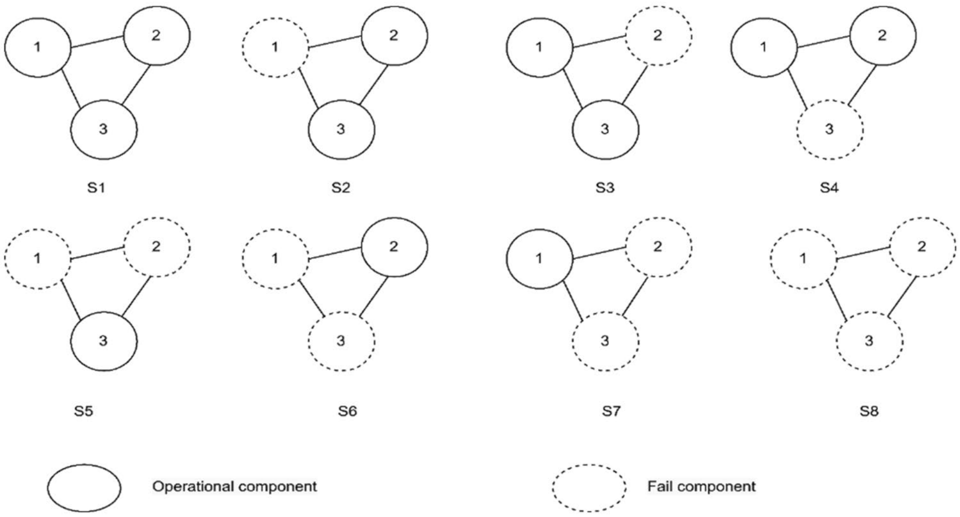
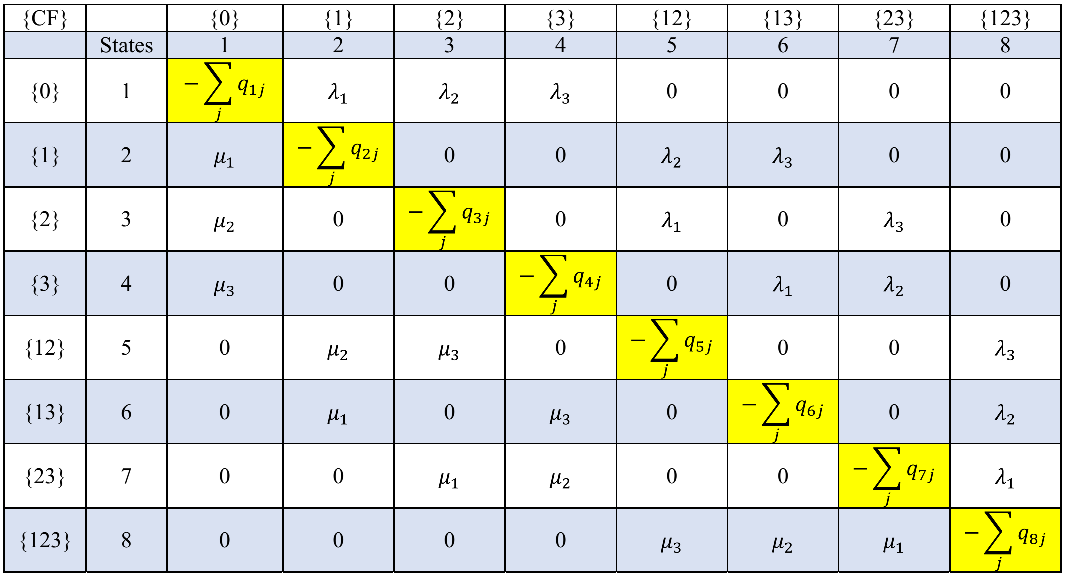


Fig. 5 All possible states of the system under consideration



{CF}		{0}	{1}	{2}	{3}	{12}	{13}	{23}	{123}
	States	1	2	3	4	5	6	7	8
{0}	1	$-\sum_j q_{1j}$	λ_1	λ_2	λ_3	0	0	0	0
{1}	2	μ_1	$-\sum_j q_{2j}$	0	0	λ_2	λ_3	0	0
{2}	3	μ_2	0	$-\sum_j q_{3j}$	0	λ_1	0	λ_3	0
{3}	4	μ_3	0	0	$-\sum_j q_{4j}$	0	λ_1	λ_2	0
{12}	5	0	μ_2	μ_3	0	$-\sum_j q_{5j}$	0	0	λ_3
{13}	6	0	μ_1	0	μ_3	0	$-\sum_j q_{6j}$	0	λ_2
{23}	7	0	0	μ_1	μ_2	0	0	$-\sum_j q_{7j}$	λ_1
{123}	8	0	0	0	0	μ_3	μ_2	μ_1	$-\sum_j q_{8j}$

Fig. 6 transaction matric of the system

2004). the first two failure rates are time-dependent. That means the failure of the component increases the time passes.

7.1 Mean Number of failures during the time interval (0, T)

The aim of calculating the mean number of failures is to put the trend of the system failure. The failure trend prediction is the challenge. This measure helps the fault tolerance technique make proactive action to manage the failure (Kalaiarasi et al. 2017). By putting all the failure and repair rates, the accumulated reward obtained in each state is given in Table 1. At the initial time system have all the component upstate. So, column zero shows the Mean no. of failure during the time (0, T) (Lisnianski 2007). Table 2 shows the mean number of failures during the time interval.

As the table shows, the mean number of failures is zero at time instance zero. As time increases, the number of failures also increases. This measure helps the system manager understand the system failure trend. That may allow the system manager to determine how much time the system goes to the failure stage. The graph representing the mean no of failures is given in Fig. 7.

7.2 Reliability

In the multistate system, the components have many objects. The mutual interaction influences the reliability of the system. Only Failure rates are considered for the reliability measurement (Lisnianski 2007). The set of Eq. 2 is used for the calculation of the reliability. Rewards related to all the operational states are considered 1, and all the failed states

Table 2 Table showing the Mean no. of failure

Time	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0.001459	0.037887	0.038353	0.058307	0.003613	0.003683	0.003689	0.000294
2	0.00568	0.072226	0.073969	0.113623	0.013121	0.013612	0.013656	0.002077
3	0.012445	0.103882	0.107567	0.166475	0.026934	0.028406	0.028525	0.006219
4	0.021561	0.13356	0.139733	0.217311	0.043899	0.047001	0.047234	0.013123
5	0.032853	0.161824	0.17094	0.266507	0.063186	0.068584	0.06896	0.022894
6	0.046169	0.189133	0.201569	0.31438	0.084209	0.092531	0.093073	0.035457
7	0.061373	0.215855	0.231922	0.361197	0.106556	0.118367	0.119091	0.050637
8	0.078343	0.242287	0.262241	0.40718	0.129948	0.145729	0.146647	0.068206
9	0.096975	0.268668	0.292716	0.452516	0.154198	0.174341	0.17546	0.087922
10	0.117175	0.29519	0.323494	0.497358	0.179189	0.203995	0.205319	0.109548

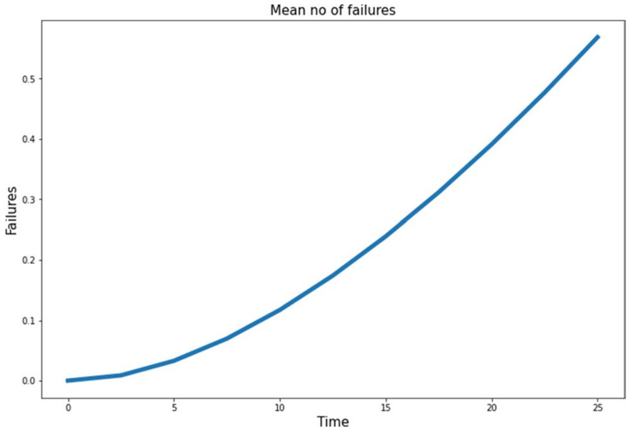


Fig. 7 Accumulated number of failures

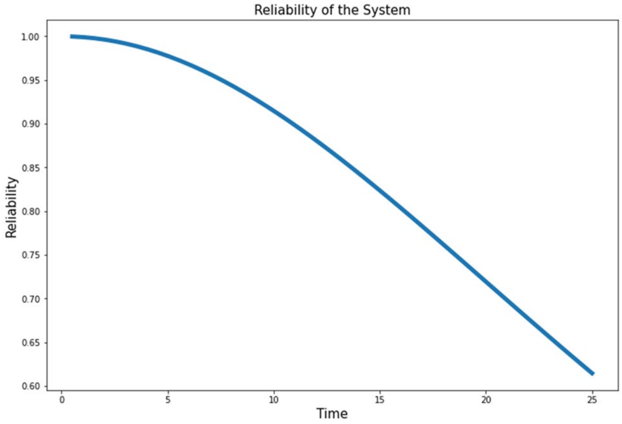


Fig. 8 Reliability of the system

and transitions are considered 0. The accumulated reward is given in Table 3 up to the time t.

Column zero shows the system's reliability because, at the initial time, all the components are working. That is represented by the state 0 (Temraz and El-Dmcese 2011). The reliability of state 1, state 2, and state 3 are less than state 1. These states are more prone to transit into the failure state. The reliability graph is shown in Fig. 8. As time passes, the reliability decreases exponentially. The system reaches zero after 140 years.

Although the reliability also greatly depends on the various characteristics of the systems. Software or specialized hardware monitoring can improve the overall system reliability. To improve the reliability of various faults, tolerate feature is added to the safety–critical system. In this case, the availability measures show the overall availability of the system. During the operation time, several failures and repairs occur. The fault-tolerance features can eliminate or reduce service disruption whenever the equipment fails by providing alternate routing and restoring lost connections(Wang 2004).

7.3 Availability

The time the system remains operational for the availability analysis is calculated. The system transition matrix has repair and failure transitions (Khvatskin and Frenkel 2017). The reward matrix and transition matrix are defined in subsection 5.2. Using the set of differential Eq. 3 used for the calculation. Table 4 shows the accumulated reward for each state that shows the amount of time spent.

The system's initial state $V_1(t)$ shows that all the components are working. Using the differential Eqs. 3, we obtain the average accumulated time spent on the working state (Lisnianski and Frenkel 2009). As time passes, the availability decreases, as shown in Fig. 9. At the steady-state, availability is 0.90. Similarly, the availability of state 2 is less than one as state 2 is more prone to failure.

Table 3 Table showing the reliability

Time	1	2	3	4	5	6	7	8	Reliability
0	0	0	0	0	0	0	0	0	0
1	0.999097	0.97027	0.969949	0.969628	0	0	0	0	0.999097
2	1.992771	1.882186	1.879717	1.877252	0	0	0	0	0.996385
3	2.975628	2.737461	2.729453	2.721474	0	0	0	0	0.991876
4	3.942402	3.537867	3.519646	3.50154	0	0	0	0	0.9856
5	4.888047	4.285229	4.251099	4.217301	0	0	0	0	0.977609
6	5.807821	4.981415	4.924901	4.869175	0	0	0	0	0.96797
7	6.69735	5.628319	5.542404	5.458106	0	0	0	0	0.956764
8	7.552686	6.227862	6.105189	5.985508	0	0	0	0	0.944086
9	8.370347	6.781975	6.615041	6.453217	0	0	0	0	0.930039
10	9.14735	7.292595	7.073922	6.863438	0	0	0	0	0.914735

Table 4 Table showing the availability

Time	1	2	3	4	5	6	7	8	Availability
0	0	0	0	0	0	0	0	0	0
1	0.999523	0.981732	0.981437	0.971542	0.092209	0.069508	0.069487	0.005911	0.999523
2	1.996356	1.932991	1.930902	1.891896	0.341583	0.258554	0.258256	0.042066	0.998178
3	2.988272	2.861206	2.854938	2.768729	0.714756	0.542809	0.541442	0.126701	0.996091
4	3.973497	3.771982	3.758741	3.608601	1.186446	0.903338	0.89942	0.268893	0.993374
5	4.950662	4.669526	4.646424	4.417118	1.737507	1.325439	1.316747	0.471731	0.990132
6	5.918755	5.556986	5.521246	5.199069	2.353437	1.797712	1.781299	0.734533	0.986459
7	6.877071	6.436711	6.385801	5.958557	3.023235	2.311327	2.283581	1.054381	0.982439
8	7.825174	7.310458	7.242173	6.699105	3.738551	2.85945	2.816171	1.427151	0.978147
9	8.762853	8.179552	8.092061	7.423755	4.493029	3.436797	3.373284	1.848202	0.97365
10	9.690099	9.045012	8.936882	8.135152	5.281827	4.039285	3.950421	2.312814	0.96901

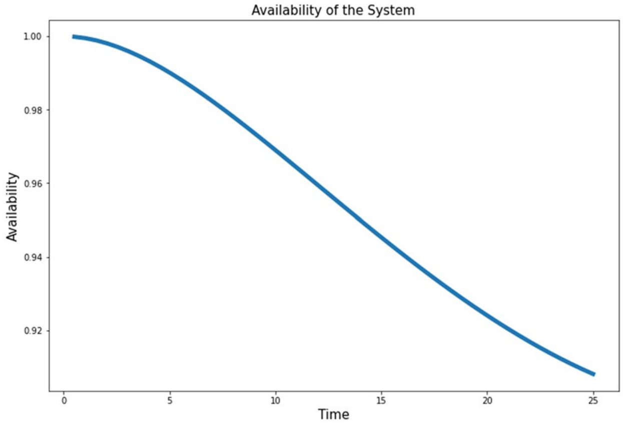


Fig. 9 The availability of the system vs. Time

7.4 Performance

The performance has various parameters, but the average available throughput and time spent on the different performance levels are considered. These are discussed in the following subsections.

7.4.1 Average available throughput

Subsection 5.3 presents a system with three different generating and demand capacities. The stochastic process represents demand and capacity, as shown in Fig. 3. For calculating the available throughput, these two stochastic processes are combined into a single stochastic process Fig. 4. The resulting process has nine states. Their failure and repair rates are given. If the demand is less than the generating capacity for any state, it is considered the acceptable state or else failed state. The numerical base data is taken from paper (Lisnianski 2007). We extended all two transitions to three transitions. Below is the transition matrix for the demanded throughput, generating throughput, and combined transition matrix.

$$a_{ij} = \begin{bmatrix} -500 & 100 & 400 \\ 200 & -1000 & 800 \\ 1 & 10 & -11 \end{bmatrix}$$

$$b_{ij} = \begin{bmatrix} -540 & 156 & 391 \\ 900 & -1110 & 210 \\ 1000 & 110 & -11 \end{bmatrix}$$

The combined stochastic transition matrix C is based on Matrix a and Matrix b.

$$C_{ij} = \begin{bmatrix} -1047 & 156 & 391 & 100 & 0 & 0 & 400 & 0 & 0 \\ 900 & -1610 & 210 & 0 & 100 & 0 & 0 & 400 & 0 \\ 1000 & 110 & -1610 & 0 & 0 & 100 & 0 & 0 & 400 \\ 200 & 0 & 0 & -1547 & 156 & 391 & 800 & 0 & 0 \\ 0 & 200 & 0 & 900 & -2110 & 210 & 0 & 800 & 0 \\ 0 & 0 & 200 & 1000 & 110 & -2110 & 0 & 0 & 800 \\ 1 & 0 & 0 & 10 & 0 & 0 & -1302 & 900 & 391 \\ 0 & 1 & 0 & 0 & 10 & 0 & 900 & -1121 & 210 \\ 0 & 0 & 1 & 0 & 0 & 10 & 1000 & 110 & -1121 \end{bmatrix}$$

Table 5 Average available throughput

Time	1	2	3	4	5	6	7	8	9	Available Th
0	0	0	0	0	0	0	0	0	0	
1	0.989783	0.990154	0.989751	0.991234	0.9913	0.99083	0.991772	0.991773	0.991769	0.991772
2	1.981537	1.981907	1.981504	1.982988	1.983053	1.982584	1.983525	1.983527	1.983523	0.991763
3	2.97329	2.973661	2.973258	2.974742	2.974807	2.974337	2.975279	2.97528	2.975276	0.99176
4	3.965044	3.965415	3.965012	3.966495	3.966561	3.966091	3.967033	3.967034	3.96703	0.991758
5	4.956798	4.957168	4.956765	4.958249	4.958314	4.957844	4.958786	4.958788	4.958784	0.991757
6	5.948551	5.948922	5.948519	5.950003	5.950068	5.949598	5.95054	5.950541	5.950537	0.991757
7	6.940305	6.940676	6.940273	6.941756	6.941822	6.941352	6.942294	6.942295	6.942291	0.991756
8	7.932058	7.932429	7.932026	7.93351	7.933575	7.933105	7.934047	7.934048	7.934045	0.991756
9	8.923812	8.924183	8.92378	8.925263	8.925329	8.924859	8.925801	8.925802	8.925798	0.991756
10	9.915566	9.915936	9.915533	9.917017	9.917083	9.916613	9.917555	9.917556	9.917552	0.991755

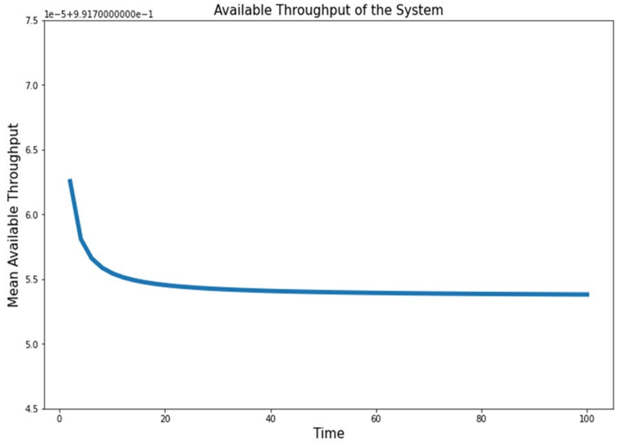


Fig. 10 Average available throughput

The throughput generating levels are L = {70, 89, 100} and the demand throughput level are D = {80, 85, 90}(Lisnianski 2007). When these values are put into the transition matrix represented in subSect. 6.3. The state where the demand satisfies (acceptable states) are 4,5,7,8,9, and failed states are 1,2,3,6.

To find the *average availability throughput* $A(T)$ according to the introduced approach. we should present the reward matrix **w**. By using differential Eq. 1, the nine-differential equation generated. The initial state is where demand is minimum and generating capacity is high. So, the initial state in our case is state 7. So, state 7 shows the available throughput of the system. Table 5 shows the accumulated reward of the states for the available throughput.

The table revealed that the available throughput is approximately 0.9917 as time passes. The throughput decreases slightly in Fig. 10. The range of the y-axis is (0.991745, 0.991775). The values of the y-axis are converted into the normalized form. The available throughput remained unchanged after 20 years.

Table 6 High-Performance time spent in percentage

Time	1	2	3	4	5	6	7	8	9	Highest Th
0	0	0	0	0	0	0	0	0	0	0
1	0.627488	0.627503	0.627334	0.627962	0.627736	0.627749	0.629015	0.628321	0.628373	0.629015
2	1.256248	1.256264	1.256095	1.256723	1.256496	1.25651	1.257776	1.257082	1.257134	0.628888
3	1.885009	1.885024	1.884856	1.885484	1.885257	1.88527	1.886537	1.885843	1.885895	0.628846
4	2.51377	2.513785	2.513617	2.514244	2.514018	2.514031	2.515297	2.514604	2.514656	0.628824
5	3.142531	3.142546	3.142377	3.143005	3.142779	3.142792	3.144058	3.143364	3.143416	0.628812
6	3.771291	3.771307	3.771138	3.771766	3.771539	3.771553	3.772819	3.772125	3.772177	0.628803
7	4.400052	4.400067	4.399899	4.400527	4.4003	4.400313	4.40158	4.400886	4.400938	0.628797
8	5.028813	5.028828	5.02866	5.029287	5.029061	5.029074	5.030341	5.029647	5.029699	0.628793
9	5.657574	5.657589	5.65742	5.658048	5.657822	5.657835	5.659101	5.658407	5.65846	0.628789
10	6.286334	6.28635	6.286181	6.286809	6.286582	6.286596	6.287862	6.287168	6.28722	0.628786

7.4.2 Time spent different performance levels

The complex system work on different performance level. How much time the system works in the different working states is also an important measure to show the performability of the safety–critical system (Januzaj et al. 2009). In this section, the system is classified into the four-performance level, i.e. (high, medium, low, and fail). The high-performance level state is state 7. The Medium

Table 7 Medium Performance Level (PL) time spent in percentage

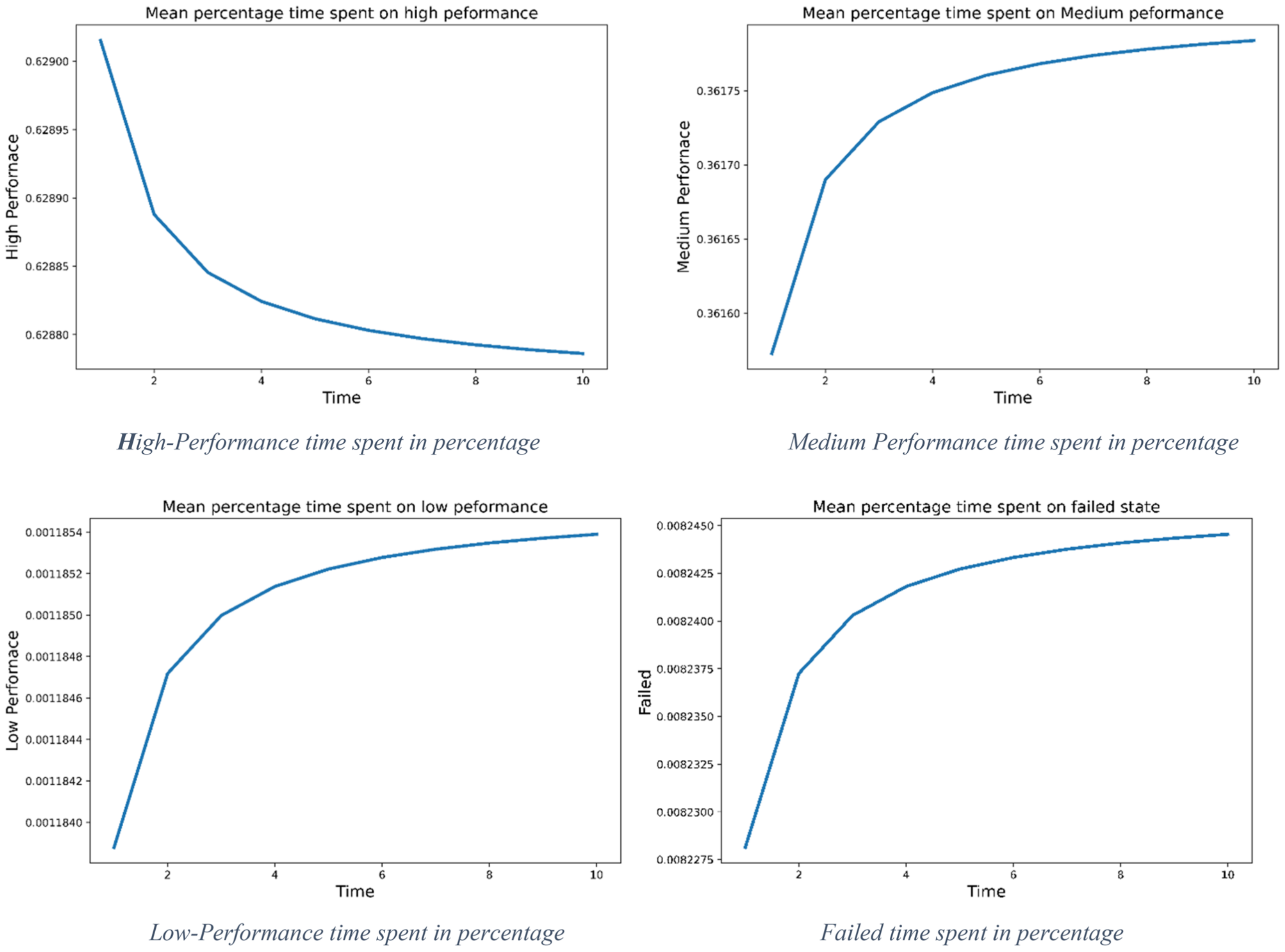
Time	1	2	3	4	5	6	7	8	9	Medium Th
0	0	0	0	0	0	0	0	0	0	0
1	0.361095	0.361432	0.361217	0.36202	0.361867	0.361837	0.361573	0.362264	0.362212	0.361573
2	0.722902	0.723239	0.723024	0.723827	0.723675	0.723645	0.72338	0.724072	0.724019	0.36169
3	1.08471	1.085046	1.084831	1.085635	1.085482	1.085452	1.085187	1.085879	1.085827	0.361729
4	1.446517	1.446853	1.446639	1.447442	1.447289	1.447259	1.446995	1.447686	1.447634	0.361749
5	1.808324	1.808661	1.808446	1.809249	1.809096	1.809067	1.808802	1.809494	1.809441	0.36176
6	2.170131	2.170468	2.170253	2.171057	2.170904	2.170874	2.170609	2.171301	2.171249	0.361768
7	2.531939	2.532275	2.532061	2.532864	2.532711	2.532681	2.532417	2.533108	2.533056	0.361774
8	2.893746	2.894083	2.893868	2.894671	2.894518	2.894488	2.894224	2.894916	2.894863	0.361778
9	3.255553	3.25589	3.255675	3.256479	3.256326	3.256296	3.256031	3.256723	3.25667	0.361781
10	3.617361	3.617697	3.617483	3.618286	3.618133	3.618103	3.617839	3.61853	3.618478	0.361784

Table 8 Low-Performance time spent in percentage

Time	1	2	3	4	5	6	7	8	9	Low Th
0	0	0	0	0	0	0	0	0	0	0
1	0.001201	0.001219	0.0012	0.001252	0.001697	0.001244	0.001184	0.001187	0.001184	0.001184
2	0.002386	0.002405	0.002385	0.002438	0.002883	0.002429	0.002369	0.002373	0.002369	0.001185
3	0.003572	0.00359	0.003571	0.003623	0.004068	0.003615	0.003555	0.003559	0.003555	0.001185
4	0.004757	0.004776	0.004756	0.004809	0.005254	0.004801	0.004741	0.004744	0.00474	0.001185
5	0.005943	0.005962	0.005942	0.005994	0.006439	0.005986	0.005926	0.00593	0.005926	0.001185
6	0.007128	0.007147	0.007127	0.00718	0.007625	0.007172	0.007112	0.007115	0.007112	0.001185
7	0.008314	0.008333	0.008313	0.008366	0.00881	0.008357	0.008297	0.008301	0.008297	0.001185
8	0.0095	0.009518	0.009499	0.009551	0.009996	0.009543	0.009483	0.009486	0.009483	0.001185
9	0.010685	0.010704	0.010684	0.010737	0.011182	0.010728	0.010668	0.010672	0.010668	0.001185
10	0.011871	0.011889	0.01187	0.011922	0.012367	0.011914	0.011854	0.011857	0.011854	0.001185

Table 9 Failed time spent in percentage

Time	1	2	3	4	5	6	7	8	9	Failed Time
0	0	0	0	0	0	0	0	0	0	0
1	0.010217	0.009846	0.010249	0.008766	0.0087	0.00917	0.008228	0.008227	0.008231	0.008228
2	0.018463	0.018093	0.018496	0.017012	0.016947	0.017416	0.016475	0.016473	0.016477	0.008237
3	0.02671	0.026339	0.026742	0.025258	0.025193	0.025663	0.024721	0.02472	0.024724	0.00824
4	0.034956	0.034585	0.034988	0.033505	0.033439	0.033909	0.032967	0.032966	0.03297	0.008242
5	0.043202	0.042832	0.043235	0.041751	0.041686	0.042156	0.041214	0.041212	0.041216	0.008243
6	0.051449	0.051078	0.051481	0.049997	0.049932	0.050402	0.04946	0.049459	0.049463	0.008243
7	0.059695	0.059324	0.059727	0.058244	0.058178	0.058648	0.057706	0.057705	0.057709	0.008244
8	0.067942	0.067571	0.067974	0.06649	0.066425	0.066895	0.065953	0.065952	0.065955	0.008244
9	0.076188	0.075817	0.07622	0.074737	0.074671	0.075141	0.074199	0.074198	0.074202	0.008244
10	0.084434	0.084064	0.084467	0.082983	0.082917	0.083387	0.082445	0.082444	0.082448	0.008245



High-Performance time spent in percentage

Medium Performance time spent in percentage

Low-Performance time spent in percentage

Failed time spent in percentage

Fig. 11 The obtained different performance levels spent time

performance states are 4,8,9. The low-performance state is state 5. State 1,2,3,6 represent the failure states of the system.

To calculate the high performance, we have to assign a reward $w_{77} = 1$ rest of the reward should be zero. The high-performance reward is shown in Table 6. For the medium performance level reward $w_{44} = w_{88} = w_{99} = 1$. The accumulated reward for the low-performance level is given in Table 7. For the low-performance level reward of state $w_{55} = 1$. The accumulated low reward is represented in Table 8. Similarly, for the failure time that the system spent, the reward for the failed states should be assigned 1, i.e., $w_{11} = w_{22} = w_{33} = w_{55} = 1$. The accumulated failed reward for the state is given in Table 9.

For calculating the percentage time spent on the different, the total accumulated reward is divided by time t. Table 6 shows the high-performance percentage time spent at the initial time is 0.629015. As time passes, the high-performance probability decreases. After some time, this reached to steady-state, as shown in Fig. 11. Similarly, the time spent between medium, low, and failed states is also given in Tables 6, 7, and 8.

The throughput availability in the medium is 0.3617 (approximately), low performance (0.001185), and at the failed state, 0.00824 (approximately). The sum of all the performance level probability equals one at each time. E.g., at time instance1, the sum of the probability of all the levels $(0.629015 + 0.361573 + 0.001184 + 0.008228 = 1)$ is 1.

The high, medium and low-performance levels show the system is operational. These performance levels show the system's availability calculated in subSect. 7.4.1. the failed state availability shows the system's unavailability during the operation.

The corresponding line graph for the different performance level probability is given the Fig. 11. Initially, the system is at a high-performance level. So, the high-performance graph in Fig. 11 shows the decrease in the probability. The other performance level at the initial time is zero. So, as time increases, the probability of the performance level increases with time. After some time, all the performance levels reached a steady state. The performability measures in this study show that the system is performing with high availability and throughput during the operational time. The primary purpose of this study is that a safety–critical system should accomplish a task within the deadline.

8 Limitations and possible scalability of the proposed approach

In investigations of predictability, Markov models are frequently used. For instance, Markov models empirically predict future fields from the present and past. It has been argued that utilizing Markov models to assess climate sensitivity can help solve some issues arising in time series research due to sampling errors. Markov models are widely used. However, they cannot be developed from deterministic, dynamical models in a precise way. This raises the question of when Markov models are suitable for modelling dynamical systems. The exponential expansion in the number of states with rising system complexity is one of the Markov model's drawbacks. It makes applying Markov-based state model generation for complex systems more difficult and increases computational resource consumption. This paper introduces a method for decomposing the target system into independent sub-systems and adopting system-level failure rates of the sub-systems estimated individually by the developed formulas. The method is based on decomposing the target system into independent sub-systems. A straightforward model of the target system can be created using the failure rates of the sub-systems.

9 Conclusion and future work

Safety Critical Systems should have high performance and reliability. If the system's reliability improves by adding some mechanism, it affects its performance and vice-versa. So, the combined study of performance and reliability (called performability) is essential and produces significant knowledge to improve the overall system quality. A combined study of performance and reliability is done based on the continuous-time Markov chain with rewards in this paper. The performability analysis is done based on the definition of the performability. The major part considered for performability analysis is dependability and performance. From the dependability, reliability and availability attributes are taken. For reliability, the mean number of failures was calculated. For the performance, the overall available throughput of the system is calculated. Further, time spent by the system during the operation at the different performance levels has been measured.

Future research can be dedicated to developing a more general dependability model for the safety–critical system. If state holding time does not follow the exponential distribution, the system cannot be modelled using the ordinary Markov chain. In practice, various systems do not hold the exponential distribution rule. This problem requires a more sophisticated modelling tool. This issue can be modelled using semi-Markov modelling. One more research dimension can be immersed as the automatic performability modelling using the suitable action based on the reward. Reinforced learning is the basic technique that can help to address this challenge. In reinforcement learning, the system can make a suitable decision based on the system requirement based on the reward.

Declarations

Conflict of interest I don't have any conflict of interest with any member of your journal. I don't have any affiliation with financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Ethical approval There are no human or animal studies are mentioned in this article.

Consent to participate Not applicable.

Consent to publish The manuscript does not contain any data from individuals. Hence it is not applicable.

References

Bas E (2019) An introduction to markov chains. Basics of Probability and Stochastic Processes, 1st edn. Springer International Publishing, Cham, pp 179–198

Eshragh F, Kargahi M (2013) Analytical architecture-based performability evaluation of real-time software systems. J Syst Softw 86:233–246. https://doi.org/10.1016/j.jss.2012.08.014

Goel AL, Okumoto K (1979) A Markovian model for reliability and other performance measures of software systems. 1979 international workshop on managing requirements knowledge. MARK 1979:769–774. https://doi.org/10.1109/MARK.1979.8817248

Gokhale SS, Wong WE, Horgan JR, Trivedi KS (2004) An analytical approach to architecture-based software performance and reliability prediction. Perform Eval 58:391–412. https://doi.org/10.1016/j.peva.2004.04.003

Goševa-Popstojanova K, Trivedi K (2000) Stochastic modeling formalisms for dependability, performance and performability. Lecture Notes Comput Sci 1769:403–422. https://doi.org/10.1007/3-540-46506-5_17

Heddaya A, Helal A (1996) Reliability, Availability, dependability and performability. A User-centered View, Boston

Howard AR (1960) Dynamic programming and markov process, 1st edn. The Massachusetts Institute of Technology, Cambridge

Januzaj V, Mauersberger R, Biechele F (2009) Performance modelling for avionics systems. International conference on computer aided systems theory. Springer-Verlag, Berlin Heidelberg, pp 833–840

John CK (2002) Safety critical system:challenges and directions. In: Proceedings of the 24th international conference on software engineering. IEEE, Orlando, FL, USA, pp 1–4

Kalaiarasi S, Merceline Anita A, Geethanjalii R (2017) Analysis of system reliability using markov technique. Global J Pure Appl Math 13:5265–5273

Karlin S, Taylor MH (1975) A first cource in stochatic prcess, 2nd edn. Elsevier, San Diego

Khvatskin L, Frenkel I (2017) Markov reward model for reliability assessment of aging markov reward model for reliability assessment of aging refrigeration

Koichi T, Shigeru Y (2009) Performability modeling for software system with performance degradation and reliability growth. IEICE Trans Fundam Electron Commun Comput Sci E92-A(7):1563–1571. https://doi.org/10.1587/transfun.E92.A.1563

Kwon YM, Agha G (2007) A Markov reward model for software reliability. In: Proceedings - 21st International parallel and distributed processing symposium, IPDPS 2007; Abstracts and CD-ROM. https://doi.org/10.1109/IPDPS.2007.370525

Lisnianski A (2007) The markov reward model for a multistate system reliability assessment with variable demand. Qual Technol Quant Manag 4:265–278. https://doi.org/10.1080/16843703.2007.11673150

Lisnianski A, Frenkel I (2009) Non-homogeneous Markov reward model for aging multistate system under corrective maintenance. Saf Reliab Risk Anal 1:551–557. https://doi.org/10.1201/9781482266481-84

Martins A, Fonseca I, Farinha JT et al (2023) Online monitoring of sensor calibration status to support condition-based maintenance. Sensors 23:2402. https://doi.org/10.3390/S23052402

Kallenberg O, Kallenberg O (1997) Foundations of modern probability (Vol. 2). New York: springer

Mo Y, Liu Y, Cui L (2018) Performability analysis of multistate series-parallel systems with heterogeneous components. Reliab Eng Syst Saf 171:48–56. https://doi.org/10.1016/j.ress.2017.10.023

Norris JR (1998) Markov Chains, Número 2008

Pietrantuono R, Russo S (2013) Introduction to Safety Critical Systems. In: Innovative technologies for dependable OTS-based critical systems. Springer Milan, pp 17–27

Singh L, Vinod G, Tripathi AK (2012) Modeling and prediction of performability of safety critical computer based systems using Petri nets. In: Proceedings - 23rd IEEE international symposium on software reliability engineering workshops, ISSREW 2012. pp 85–94

Smith RM, Trivedi KS, Ramesh AV (1988) Performability analysis: measures, an algorithm and a case study. IEEE Commun Surv Tutor 37:406–417

Sotelo M, Martins A, Mateus B et al (2023) Predicting the health status of a pulp press based on deep neural networks and hidden markov models. Energies 16:2651. https://doi.org/10.3390/EN16062651

Strielkina A, Kharchenko V, Uzun D (2018) A markov model of healthcare internet of things system considering failures of components. CEUR Workshop Proc 2104:530–543

Temraz NS, El-Dmcese MA (2011) Availability and reliability measures for multistate system by using markov reward model. Reliab Theory Appl 2:68–85

Toledano S, Gartsman I, Avitan G, et al (2016) On Markov reward approach to failure criticality importance assessment for aging multistate system. In: Proceedings - 2nd international symposium on stochastic models in reliability engineering, life science, and operations management, SMRLO, pp 375–379. https://doi.org/10.1109/SMRLO.2016.66

Trivedi KS, Ma X, Dharmaraja S (2003) Performability modelling of wireless communication systems. Int J Commun Syst 16:561–577. https://doi.org/10.1002/dac.605

Viktorova VS, Lubkov NV, Stepanyants AS (2018) A unified approach to reliability, availability, performability analysis based on markov processes with rewards. Adv Syst Sci Appl 18:13–38. https://doi.org/10.25728/assa.2018.18.4.624

Wang JL (2004) Markov-chain based reliability analysis for distributed systems. Comput Electr Eng 30:183–205. https://doi.org/10.1016/j.compeleceng.2002.02.001

Wang L, Tian Y, Pei Z (2017) Reliability analysis of 6-component lattice load-sharing Markov repairable system with spatial dependence. Int J Perform Eng 13:279–287. https://doi.org/10.23940/ijpe.17.03.p4.279287

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.