

Eksploatacja i Niezawodnosc - Maintenance and Reliability

Volume 29 (2027), Issue 3

https://ein.org.pl

The Polish Maintenance Society

Semi-Markov analysis of a two-unit warm standby system with inspection-based decisions, preventive maintenance, and preemptive-resume priority

Hany Mohamed Hamouda[1,*]

[1] Department of Management Information Systems, College of Business and Economics, Qassim University, Buraidah 52571, Saudi Arabia
* Corresponding Author: h.hamouda@qu.edu.sa

Received: 24 February 2026
Revised: 28 September 2026
Accepted: 29 October 2026
Online: 16 November 2026

This is an open access article under the CC BY 4.0 license

Abstract

Reliable operation of software-intensive computer systems requires maintenance strategies that explicitly account for the distinct characteristics of hardware and software failures. This paper develops a stochastic reliability and economic model for a two-unit warm standby system subject to independent hardware and software failures, attended by a single server responsible for inspection, repair, replacement, software upgrading, and preventive maintenance activities. A key contribution of the proposed framework is the integration of inspection-based repair-or-replacement decisions with a preemptive-resume priority discipline, whereby software upgrading activities interrupt ongoing hardware maintenance and resume from the point of interruption after upgrading completion. This specific combination—integrating inspection-based decisions, competing-risk preventive maintenance, preemptive-resume software priority, and a full economic analysis within a unified semi-Markov framework—has not been previously addressed in a unified analytical framework. The stochastic behavior of the system is modeled using a semi-Markov process and regenerative point techniques. Closed-form expressions are derived for important performance measures, including the Mean Time to System Failure (MTSF), steady-state availability, server busy fractions, and expected profit per unit time. Numerical examples and sensitivity analyses are conducted to investigate the effects of preventive maintenance and software upgrading rates on system performance. A comparative study with a non-priority maintenance policy demonstrates that the proposed strategy yields measurable improvements in both availability and profitability. The developed model provides a useful analytical framework for evaluating maintenance policies and supporting decision-making in modern software-intensive computer systems.

Keywords: *warm standby system, semi-Markov process, inspection-based maintenance, preventive maintenance, preemptive-resume priority*

Article citation:
Hamouda H M, Semi-Markov analysis of a two-unit warm standby system with inspection-based decisions, preventive maintenance, and preemptive-resume priority, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2027: 29(3) http://doi.org/10.17531/ein/242320

Highlights

- Semi-Markov model of a two-unit warm standby system with hardware and software failures.
- Inspection decides whether a failed unit is repaired or replaced.
- Software upgrading has preemptive-resume priority over all hardware maintenance.
- Closed-form MTSF, availability, busy periods and profit via regenerative points.

1. Introduction

Modern software-intensive computer systems form the backbone of critical infrastructures such as telecommunications, cloud computing, healthcare services, financial networks, and industrial automation. The growing dependence on such systems has intensified the demand for reliable maintenance strategies capable of minimizing service interruptions and ensuring continuous availability. Because system failures may result in substantial economic losses and service degradation, reliability and maintainability analyses have become indispensable tools for evaluating system performance and supporting maintenance decision-making.

Unlike conventional engineering systems, computer systems are simultaneously exposed to hardware and software failures, which differ fundamentally in their failure mechanisms and recovery procedures. Hardware failures generally require physical repair or replacement actions, whereas software failures are commonly addressed through restarting or software

upgrading activities. Consequently, reliability models that jointly account for both failure modes provide a more realistic representation of operational behavior. Early contributions by Friedman et al. [1] and Welke et al. [2] established foundational frameworks for combined hardware/software reliability modeling. Lai et al. [3] subsequently proposed availability models for distributed software/hardware systems, demonstrating the value of integrated analytical treatments. Building on these foundations, Malik and Anand [4] and Munday and Malik [5] developed stochastic reliability and economic models for computer systems subject to independent hardware and software failures, highlighting the influence of maintenance and software upgrading policies on long-term system performance. Kumar et al. [6] further emphasized the importance of software stability by assigning preemptive priority to software upgrading over hardware repair tasks.

To enhance system reliability and operational continuity, standby redundancy has been extensively employed in engineering practice. Among available redundancy structures, warm standby configurations provide a practical balance between reliability improvement and resource efficiency. Goel et al. [7] conducted a cost-benefit analysis of a two-unit warm standby system with inspection, repair and post-repair. Mokaddis et al. [8] investigated the stochastic behaviour of a two-unit warm standby system with two types of repairmen and patience time, while Mahmoud and Hamouda [9] studied a two-unit cold standby system attended by regular and expert repairmen, with patience time and three types of failures. Rizwan et al. [10] analyzed the reliability of a two-unit hot standby PLC-based industrial system. More recently, Kumar [11] analyzed a cold standby computer system with hardware repair and software replacement subject to conditional arrival time of the server. Temraz [12] analyzed a discrete-time semi-Markov model for a two-unit warm standby system subject to degradation effects, and Qi et al. [13] proposed a condition-based maintenance policy for warm standby systems with imperfect switching. Furthermore, Danjuma et al. [14] studied reliability, availability, maintainability, and dependability characteristics of cold standby series-parallel systems, and Ragheb et al. [15] examined reliability and availability measures of a two-unit cold standby structure under imperfect switching conditions. Collectively, these contributions demonstrate the continuing research interest in standby system reliability modeling and maintenance optimization.

Preventive maintenance and inspection policies constitute additional mechanisms for improving system reliability and reducing unexpected failures. Inspection activities provide valuable information regarding system condition and support maintenance decision-making, while preventive maintenance aims to reduce the likelihood of future failures and extend operational life. Malik et al. [16] analyzed a two-unit parallel system in which failed units are inspected before repair, with priority given to repair. Notably, following inspection, the failed hardware unit may either be repaired or replaced depending on the inspection outcome, a decision mechanism that critically affects system availability and long-term cost performance. Kumar et al. [17] carried out a cost analysis of a two-unit cold standby system subject to degradation, inspection and priority. Monika and Chopra [18] analyzed profitability in an industrial food production system incorporating priority-based repair. More recently, Dabas and Rathee [19] employed a semi-Markov framework to analyze parallel systems incorporating inspection and priority mechanisms, and Wang et al. [20] proposed a hybrid inspection-replacement policy that explicitly accounts for imperfect inspections and degradation processes. These studies highlight the growing importance of integrating inspection and maintenance decisions within unified reliability models.

Various stochastic approaches have been adopted for analyzing repairable systems, including Markov processes, semi-Markov processes, regenerative techniques, and supplementary variable methods. Koutras [21] utilized Markov regenerative processes to evaluate the dependability and performance of two-unit multi-state systems under maintenance. Mittal et al. [22] demonstrated the effectiveness of semi-Markov modeling for reliability and availability evaluation of high-altitude platform stations. Bai et al. [23] similarly applied semi-Markov models to assess the dependability of service function chains. Yadav et al. [24] applied Markov process models to evaluate reliability and availability measures of multi-unit repairable systems, while Shivani et al. [25] conducted sensitivity analyses of series-parallel systems using Markov processes combined with the supplementary variable technique. Zhang et al. [26] also

employed the supplementary variable method to derive reliability characteristics of two-dimensional linear consecutive repairable systems. These contributions confirm the versatility and effectiveness of stochastic modeling approaches for complex repairable systems.

To the best of the authors' knowledge, no published study has simultaneously incorporated inspection-based repair-or-replacement decisions, competing-risk preventive maintenance, preemptive-resume software upgrading priority, and economic performance evaluation within a two-unit warm standby computer system modeled through a semi-Markov framework. In modern software-intensive systems, delays in upgrading may significantly degrade long-term performance and operational stability, further motivating the prioritization of software upgrading over hardware-related maintenance tasks.

Table 1. Comparison of related studies with the proposed model.

Study	Two-Unit Warm Standby	H/W & S/W Failures	Software Upgrading	Inspection Policy	Repair/Replace Decision	Preventive Maintenance	Preemptive-Resume Priority	Economic Analysis	Semi-Markov Model
Malik & Anand (2010) [4]	×	✓	✓	×	×	×	×	✓	×
Munday & Malik (2015) [5]	×	✓	✓	×	×	×	✓	✓	×
Kumar et al. (2013) [6]	×	✓	✓	×	×	×	✓	×	×
Kumar (2017) [11]	×	✓	×	×	×	×	×	✓	×
Temraz (2022) [12]	✓	×	×	×	×	×	×	×	✓
Dabas & Rathee (2022) [19]	×	×	×	✓	✓	✓	✓	✓	✓
Qi et al. (2024) [13]	✓	×	×	✓	×	✓	×	×	×
Wang et al. (2024) [20]	×	×	×	✓	✓	×	×	×	×
Ragheb et al. (2025) [15]	×	×	×	×	×	×	×	×	×
Proposed Study	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓ = Feature present; × = Feature absent; H/W = Hardware; S/W = Software

Among the studies considered in Table 1, the proposed model is the only framework that simultaneously incorporates all nine features within a unified semi-Markov framework.

This study proposes a two-unit warm standby computer system subject to independent hardware and software failures, attended by a single server responsible for inspection, repair, replacement, software upgrading, and preventive maintenance activities. A preemptive-resume priority discipline is adopted, whereby software upgrading interrupts any ongoing hardware-related service and resumes from the point of interruption after upgrading completion. The system is analyzed using a semi-Markov process and regenerative point techniques to derive closed-form expressions for the Mean Time to System Failure (MTSF), steady-state availability, server busy fractions, and expected profit per unit time. A comparative study with a non-priority maintenance policy is conducted to evaluate the operational and economic impact of the proposed strategy, thereby demonstrating the practical benefits of the proposed maintenance policy.

The main contributions of this study are summarized as follows:

(i) Development of a semi-Markov model that simultaneously integrates hardware failures, software failures, inspection-based repair-or-replacement decisions, competing-risk preventive maintenance, and

economic analysis within a unified analytical framework.

(ii) Incorporation of a preemptive-resume priority discipline for software upgrading in a two-unit warm standby structure, allowing interrupted hardware maintenance to resume from the point of interruption.

(iii) Analytical derivation of closed-form expressions for key reliability and economic performance measures, including the Mean Time to System Failure (MTSF), steady-state availability, server busy fractions, and expected profit per unit time, using regenerative point techniques.

(iv) Formulation of an expected profit per unit time function, enabling quantitative cost-benefit evaluation of the proposed maintenance strategy.

(v) Comparative assessment with a non-priority maintenance policy, demonstrating the measurable improvements in availability and profitability achieved by the preemptive-resume priority strategy through numerical sensitivity analysis.

2. Assumptions

The stochastic model is developed under the following assumptions:

1. The system consists of two identical units. At any time, one unit operates while the other remains in warm standby mode and can immediately replace the operative unit upon failure.
2. Hardware and software failures occur independently of each other and independently of all maintenance activities, including service interruptions due to the adopted priority discipline.
3. A single server performs all maintenance operations, namely hardware inspection, hardware repair, hardware replacement, software upgrading, and preventive maintenance.
4. Upon hardware failure of the operative unit, the failed unit first undergoes inspection. After inspection, it is sent for repair with probability a or for replacement with probability b = 1 − a. The probabilities a and b are assumed to be constant and independent of time, system age, and prior maintenance history. For clarity, the inspection-based repair or replacement decision described above applies exclusively to hardware failures of the operative unit. In contrast, hardware failures of the standby unit are assumed to be sent directly for repair without undergoing inspection.
5. Software failures are corrected exclusively through software upgrading; no repair or replacement action is considered for software components.
6. A strict preemptive-resume priority discipline is adopted: software upgrading interrupts any ongoing hardware-related activity (inspection, repair, or replacement). The interrupted hardware service resumes from the point of interruption after completion of the software upgrading.
7. At most one unit can receive service at any time. If both units require maintenance simultaneously, one unit is served while the other remains in a failed inactive state until the server becomes available.
8. Preventive maintenance (PM) initiation competes with hardware and software failures according to an exponential competing-risk structure. Specifically, PM for the operative unit is initiated after an exponentially distributed time with rate β. The first occurring event among hardware failure, software failure, or PM initiation determines the next system transition. During preventive maintenance, the standby unit becomes operative, and the PM duration follows a general distribution with CDF K(t).
9. Failure times are exponentially distributed with constant rates:
   - λ_1: hardware failure rate of the operative unit,
   - λ_2: software failure rate of the operative unit,
   - λ_3: failure rate of the standby unit.
10. Service times follow general distributions with CDFs: Inspection H(t), repair G(t), replacement V(t), software upgrading F(t), preventive maintenance K(t).

Special Case for Numerical Illustration

For the numerical study presented in Section 9, maintenance times are assumed to follow exponential distributions with constant rates $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$, corresponding to inspection, repair, upgrading, replacement, and preventive maintenance completion, respectively.

3. Notations

Let the system be modeled by a semi-Markov process with state space partitioned as follows:

$$\mathcal{E} = \{S_0, S_1, S_2, S_3, S_{10}, S_{11}, S_{14}\} \quad (1)$$

denotes the set of regenerative states, and

$$\bar{\mathcal{E}} = \{S_4, S_5, S_6, S_7, S_8, S_9, S_{12}, S_{13}\} \quad (2)$$

denotes the set of non-regenerative states.

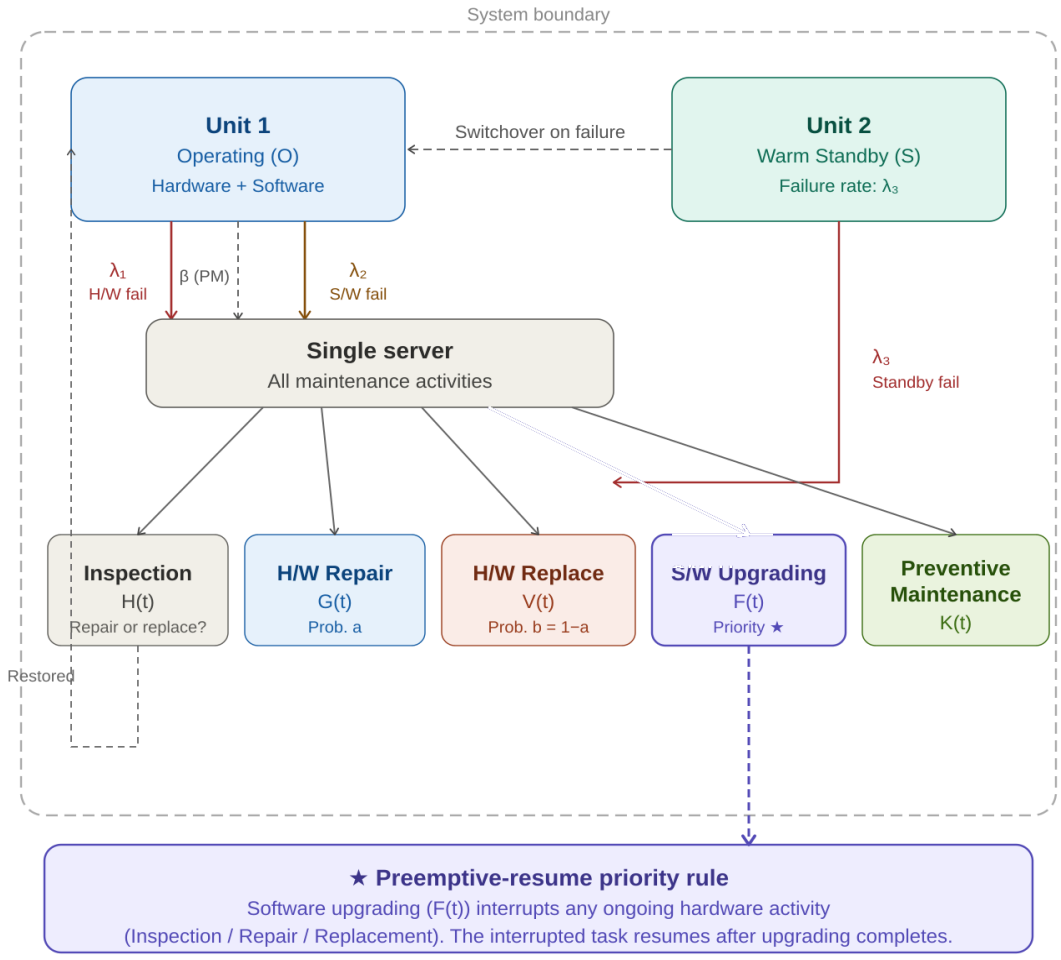

Figure 1. System configuration and maintenance flow of the proposed two-unit warm standby system.

The following symbols are used throughout the paper:

Operational Modes

- O: Operative mode
- S: Warm standby mode
- I: Hardware inspection
- R: Hardware repair
- P: Hardware replacement
- U: Software upgrading
- M: Preventive maintenance

Failure and Maintenance Rates

- λ_1: Hardware failure rate of the operative unit
- λ_2: Software failure rate of the operative unit
- λ_3: Failure rate of the standby unit
- β: Preventive maintenance initiation rate

Service-Time Distributions

Let

- $H(t)$: CDF of hardware inspection time
- $G(t)$: CDF of hardware repair time
- $V(t)$: CDF of hardware replacement time
- $F(t)$: CDF of software upgrading time
- $K(t)$: CDF of preventive maintenance time

Their corresponding probability density functions are denoted by $h(t), g(t), v(t), f(t), k(t)$, respectively.

Transition Functions

- $Q_{ij}(t)$: CDF of the direct transition time from regenerative state i to state j without visiting any other regenerative state (where j may be regenerative or non-regenerative).
- $Q_{ij}^{(k)}(t)$: CDF of the transition time from regenerative state i to regenerative state j via a visit to non-regenerative state k exactly once.
- $\tilde{q}_{ij}(s)$: Laplace–Stieltjes transform of the transition density $q_{ij}(t)$.

Sojourn Times

- μ_i: Mean sojourn time in state S_i

Exponential Special Case (Numerical Illustration)

For the numerical analysis, service times are assumed to follow exponential distributions with constant rates:

- α_1: Hardware inspection rate
- α_2: Hardware repair rate
- α_3: Software upgrading rate
- α_4: Hardware replacement rate
- α_5: Preventive maintenance completion rate

Under this assumption, the model reduces to a continuous-time Markov process.

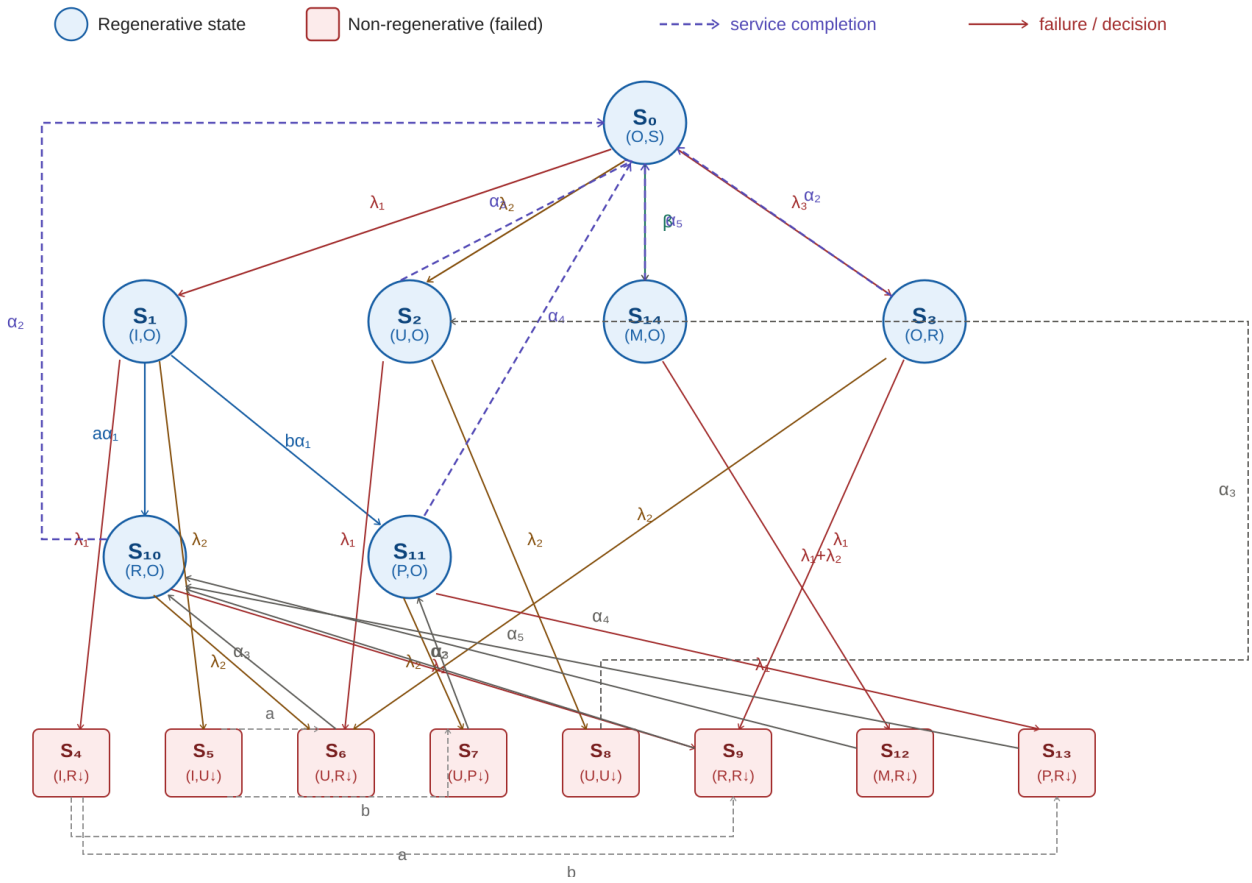

Figure 2. State transition diagram of the proposed two-unit warm standby system.

The Possible Transition States of the System

The system may occupy the following states:

S_0: Normal operation. One unit is operative (O), and the other is in warm standby (S).

S_1: Hardware failure. The failed operative unit is under inspection (I), and the standby unit becomes operative (O).

S_2: Software failure. The failed operative unit is under software upgrading (U), and the standby unit becomes operative (O).

S_3: Standby failure. The operative unit continues functioning (O), and the standby unit is sent directly for hardware repair (R).

S_4: Failed state (system down). Unit 1 is under hardware inspection (I), and Unit 2 is waiting for hardware repair (R). No unit is operative; therefore, this is a non-regenerative state.

S_5: Failed state (system down). Unit 1 is under hardware inspection (I), and Unit 2 is waiting for software upgrading (U). Since no unit is operative, this state is classified as non-regenerative.

S_6: Failed state (system down). Unit 1 is under software upgrading (U), and Unit 2 is waiting for hardware repair (R). The absence of an operative unit makes this a non-regenerative state.

S_7: Failed state (system down). Unit 1 is under software upgrading (U), and Unit 2 is waiting for hardware replacement (P) following an inspection decision. As no unit is operative, this state is non-regenerative.

S_8: Failed state (system down). Unit 1 is under software upgrading (U), and Unit 2 is waiting for software upgrading (U). With both units unavailable for operation, this state is non-regenerative.

S_9: Failed state (system down). Unit 1 is under hardware repair (R), and Unit 2 is waiting for hardware repair (R). Since neither unit is operative, this state is classified as non-regenerative.

S_{10}: Hardware failure. The failed unit is under repair (R) after inspection, and the standby unit is operative (O).

S_{11}: Hardware failure. The failed unit is under replacement (P) after inspection, and the standby unit is operative (O).

S_{12}: Failed state (system down). Unit 1 is under preventive maintenance (M), and Unit 2 is waiting for hardware repair (R). As no unit is operative during this configuration, the state is non-regenerative.

S_{13}: Failed state (system down). Unit 1 is under hardware replacement (P), and Unit 2 is in a failed inactive state awaiting repair (R). No unit is operative; hence, this is a non-regenerative state.

S_{14}: Preventive maintenance. The operative unit is under preventive maintenance (M), and the standby unit becomes operative (O).

The system states can be classified into regenerative and non-regenerative (failed) states. The regenerative states are $S_0, S_1, S_2, S_3, S_{10}, S_{11}$ and S_{14}, while the non-regenerative failed states include $S_4, S_5, S_6, S_7, S_8, S_9, S_{12}$ and S_{13}.

4. Transition probabilities and mean sojourn times

4.1. Definitions

Let $Q_{ij}(t)$denote the CDF of the transition time from state $S_i$to state S_j(directly, without visiting any other regenerative state), and let $q_{ij}(t) = \frac{d}{dt}Q_{ij}(t)$be the corresponding density (when it exists). The (one-step) transition probability is:

$$P_{ij} = Q_{ij}(\infty) = \int_0^\infty q_{ij}(t)\, dt \tag{3}$$

The mean sojourn time in state $S_i$is:

$$\mu_i = \int_0^\infty \Pr(T_i > t)\, dt \tag{4}$$

where $T_i$is the total holding time in S_i.

The three quantities $Q_{ij}(t)$, $q_{ij}(t)$, and p_{ij} are related but distinct within the semi-Markov framework. The kernel $Q_{ij}(t)$ gives the probability of transitioning from state i to state j within time t. Its derivative $q_{ij}(t) = \mathrm{d}Q_{ij}(t)/\mathrm{dt}$ is the transition rate density. The one-step probability $P_{ij} = Q_{ij}(\infty) = \int_0^\infty q_{ij}(t)\, dt$ is the unconditional transition probability regardless of time. In summary: $Q_{ij}(t)$ describes the joint distribution of destination and sojourn time; $q_{ij}(t)$ is its density; and p_{ij} is the marginal transition probability obtained by integrating out time.

4.2. Transition probabilities and mean sojourn time for

4.2.1. State $S_0 = (O, S)$

In state S_0, one unit is operative and the other is in warm standby. The system leaves state S_0 due to one of the following competing exponential events:

- Hardware failure of the operative unit with rate λ_1, leading to state S_1.
- Software failure of the operative unit with rate λ_2, leading to state S_2.
- Hardware failure of the standby unit with rate λ_3, leading to state S_3.
- Initiation of preventive maintenance with rate β, leading to state S_{14}.

Total exit rate:

$$D_0 = \lambda_1 + \lambda_2 + \lambda_3 + \beta \tag{5}$$

Transition densities:

$$q_{01}(t) = \lambda_1 e^{-D_0 t},\ t \geq 0 \tag{6}$$

$$q_{02}(t) = \lambda_2 e^{-D_0 t},\ t \geq 0 \tag{7}$$

$$q_{03}(t) = \lambda_3 e^{-D_0 t},\ t \geq 0 \tag{8}$$

$$q_{0,14}(t) = \beta e^{-D_0 t},\ t \geq 0 \tag{9}$$

Transition CDFs:

$$Q_{01}(t) = \int_0^t \lambda_1\, e^{-D_0 u}\, du = \frac{\lambda_1}{D_0}(1 - e^{-D_0 t}) \tag{10}$$

$$Q_{02}(t) = \frac{\lambda_2}{D_0}(1 - e^{-D_0 t}) \tag{11}$$

$$Q_{03}(t) = \frac{\lambda_3}{D_0}(1 - e^{-D_0 t}) \tag{12}$$

$$Q_{0,14}(t) = \frac{\beta}{D_0}(1 - e^{-D_0 t}) \tag{13}$$

One-step transition probabilities:

$$P_{01} = Q_{01}(\infty) = \frac{\lambda_1}{D_0} \tag{14}$$

$$P_{02} = Q_{02}(\infty) = \frac{\lambda_2}{D_0} \quad (15)$$

$$P_{03} = Q_{03}(\infty) = \frac{\lambda_3}{D_0} \quad (16)$$

$$P_{0,14} = Q_{0,14}(\infty) = \frac{\beta}{D_0} \quad (17)$$

$$P_{01} + P_{02} + P_{03} + P_{0,14} = 1 \quad (18)$$

Mean sojourn time:

$$\mu_0 = \int_0^\infty e^{-D_0 t}\, dt = \frac{1}{D_0} \quad (19)$$

4.2.2. State S_1 (inspection state)

In S_1, the failed operative unit is under inspection (rate α_1), while the other unit is operative. During inspection, the operative unit may fail due to hardware (λ_1) or software (λ_2). Inspection completion leads to repair with probability a or replacement with probability $b = 1 - a$.

Total exit rate:

$$D_1 = \alpha_1 + \lambda_1 + \lambda_2 \quad (20)$$

Transition densities:

$$q_{14}(t) = \lambda_1 e^{-D_1 t},\ t \geq 0 \quad (21)$$

$$q_{15}(t) = \lambda_2 e^{-D_1 t},\ t \geq 0 \quad (22)$$

$$q_{1,10}(t) = a\alpha_1 e^{-D_1 t},\ t \geq 0 \quad (23)$$

$$q_{1,11}(t) = b\alpha_1 e^{-D_1 t},\ t \geq 0 \quad (24)$$

Transition CDFs:

$$Q_{14}(t) = \frac{\lambda_1}{D_1}(1 - e^{-D_1 t}) \quad (25)$$

$$Q_{15}(t) = \frac{\lambda_2}{D_1}(1 - e^{-D_1 t}) \quad (26)$$

$$Q_{1,10}(t) = \frac{a\alpha_1}{D_1}(1 - e^{-D_1 t}) \quad (27)$$

$$Q_{1,11}(t) = \frac{b\alpha_1}{D_1}(1 - e^{-D_1 t}) \quad (28)$$

One-step transition probabilities:

$$P_{14} = \frac{\lambda_1}{D_1} \quad (29)$$

$$P_{15} = \frac{\lambda_2}{D_1} \quad (30)$$

$$P_{1,10} = \frac{a\alpha_1}{D_1} \quad (31)$$

$$P_{1,11} = \frac{b\alpha_1}{D_1} \quad (32)$$

Mean sojourn time:

$$\mu_1 = \int_0^\infty e^{-D_1 t}\, dt = \frac{1}{D_1} \quad (33)$$

4.2.3. State S_2 (software upgrading state)

In S_2, the failed operative unit is under software upgrading (rate α_3), while the other unit is operative. During upgrading, the operating unit may fail due to hardware (λ_1) or software (λ_2).

Total exit rate:

$$D_2 = \alpha_3 + \lambda_1 + \lambda_2 \quad (34)$$

Transition densities:

$$q_{20}(t) = \alpha_3 e^{-D_2 t},\ t \geq 0 \quad (35)$$

$$q_{26}(t) = \lambda_1 e^{-D_2 t},\ t \geq 0 \quad (36)$$

$$q_{28}(t) = \lambda_2 e^{-D_2 t},\ t \geq 0 \quad (37)$$

Transition CDFs:

$$Q_{20}(t) = \frac{\alpha_3}{D_2}(1 - e^{-D_2 t}) \quad (38)$$

$$Q_{26}(t) = \frac{\lambda_1}{D_2}(1 - e^{-D_2 t}) \quad (39)$$

$$Q_{28}(t) = \frac{\lambda_2}{D_2}(1 - e^{-D_2 t}) \quad (40)$$

One-step transition probabilities:

$$P_{20} = \frac{\alpha_3}{D_2} \quad (41)$$

$$P_{26} = \frac{\lambda_1}{D_2} \quad (42)$$

$$P_{28} = \frac{\lambda_2}{D_2} \quad (43)$$

Mean sojourn time:

$$\mu_2 = \frac{1}{D_2} \quad (44)$$

4.2.4. State S_3 (standby repair state)

In S_3, the standby unit is under hardware repair (rate α_2), while the operating unit continues operating. During this time, the operative unit may fail due to hardware (λ_1) or software (λ_2). A software failure preempts repair and triggers upgrading.

Total exit rate:

$$D_3 = \alpha_2 + \lambda_1 + \lambda_2 \quad (45)$$

Transition densities:

$$q_{30}(t) = \alpha_2 e^{-D_3 t},\ t \geq 0 \quad (46)$$

$$q_{39}(t) = \lambda_1 e^{-D_3 t},\ t \geq 0 \quad (47)$$

$$q_{36}(t) = \lambda_2 e^{-D_3 t},\ t \geq 0 \quad (48)$$

Transition CDFs:

$$Q_{30}(t) = \frac{\alpha_2}{D_3}(1 - e^{-D_3 t}) \quad (49)$$

$$Q_{39}(t) = \frac{\lambda_1}{D_3}(1 - e^{-D_3 t}) \quad (50)$$

$$Q_{36}(t) = \frac{\lambda_2}{D_3}(1-e^{-D_3 t}) \quad (51)$$

One-step transition probabilities:

$$P_{30} = \frac{\alpha_2}{D_3} \quad (52)$$

$$P_{39} = \frac{\lambda_1}{D_3} \quad (53)$$

$$P_{36} = \frac{\lambda_2}{D_3} \quad (54)$$

Mean sojourn time:

$$\mu_3 = \frac{1}{D_3} \quad (55)$$

4.2.5. State S_{10} (repair after inspection)

In S_{10}, the failed unit is under repair (rate α_2) after inspection, while the other unit is operative. During repair, the operative unit may fail due to hardware (λ_1) or software (λ_2). A software failure preempts repair.

Total exit rate:

$$D_{10} = \alpha_2 + \lambda_1 + \lambda_2 \quad (56)$$

Transition densities:

$$q_{10,0}(t) = \alpha_2 e^{-D_{10}t},\ t \geq 0 \quad (57)$$

$$q_{10,9}(t) = \lambda_1 e^{-D_{10}t},\ t \geq 0 \quad (58)$$

$$q_{10,6}(t) = \lambda_2 e^{-D_{10}t},\ t \geq 0 \quad (59)$$

Transition CDFs:

$$Q_{10,0}(t) = \frac{\alpha_2}{D_{10}}(1-e^{-D_{10}t}) \quad (60)$$

$$Q_{10,9}(t) = \frac{\lambda_1}{D_{10}}(1-e^{-D_{10}t}) \quad (61)$$

$$Q_{10,6}(t) = \frac{\lambda_2}{D_{10}}(1-e^{-D_{10}t}) \quad (62)$$

One-step transition probabilities:

$$P_{10,0} = \frac{\alpha_2}{D_{10}} \quad (63)$$

$$P_{10,9} = \frac{\lambda_1}{D_{10}} \quad (64)$$

$$P_{10,6} = \frac{\lambda_2}{D_{10}} \quad (65)$$

Mean sojourn time:

$$\mu_{10} = \frac{1}{D_{10}} \quad (66)$$

4.2.6. State S_{11} (replacement after inspection)

In S_{11}, the failed unit is under replacement (rate α_4) after inspection, while the other unit is operative. During replacement, the operating unit may fail due to hardware (λ_1) or software (λ_2). A software failure preempts replacement.

Total exit rate:

$$D_{11} = \alpha_4 + \lambda_1 + \lambda_2 \quad (67)$$

Transition densities:

$$q_{11,0}(t) = \alpha_4 e^{-D_{11}t},\ t \geq 0 \quad (68)$$

$$q_{11,13}(t) = \lambda_1 e^{-D_{11}t},\ t \geq 0 \quad (69)$$

$$q_{11,7}(t) = \lambda_2 e^{-D_{11}t},\ t \geq 0 \quad (70)$$

Transition CDFs:

$$Q_{11,0}(t) = \frac{\alpha_4}{D_{11}}(1-e^{-D_{11}t}) \quad (71)$$

$$Q_{11,13}(t) = \frac{\lambda_1}{D_{11}}(1-e^{-D_{11}t}) \quad (72)$$

$$Q_{11,7}(t) = \frac{\lambda_2}{D_{11}}(1-e^{-D_{11}t}) \quad (73)$$

One-step transition probabilities:

$$P_{11,0} = \frac{\alpha_4}{D_{11}} \quad (74)$$

$$P_{11,13} = \frac{\lambda_1}{D_{11}} \quad (75)$$

$$P_{11,7} = \frac{\lambda_2}{D_{11}} \quad (76)$$

Mean sojourn time:

$$\mu_{11} = \frac{1}{D_{11}} \quad (77)$$

4.2.7. State S_{14} (preventive maintenance)

In S_{14}, the operative unit is under preventive maintenance (completion rate α_5), while the standby unit becomes operative. During PM, the operative (now running) unit may fail due to hardware or software.

Total exit rate:

$$D_{14} = \alpha_5 + \lambda_1 + \lambda_2 \quad (78)$$

Transition densities:

$$q_{14,0}(t) = \alpha_5 e^{-D_{14}t},\ t \geq 0 \quad (79)$$

$$q_{14,12}(t) = (\lambda_1 + \lambda_2)e^{-D_{14}t},\ t \geq 0 \quad (80)$$

Transition CDFs:

$$Q_{14,0}(t) = \frac{\alpha_5}{D_{14}}(1-e^{-D_{14}t}) \quad (81)$$

$$Q_{14,12}(t) = \frac{\lambda_1+\lambda_2}{D_{14}}(1-e^{-D_{14}t}) \quad (82)$$

One-step transition probabilities:

$$P_{14,0} = \frac{\alpha_5}{D_{14}} \quad (83)$$

$$P_{14,12} = \frac{\lambda_1+\lambda_2}{D_{14}} \quad (84)$$

Mean sojourn time:

$$\mu_{14} = \frac{1}{D_{14}} \quad (85)$$

Non-regenerative states (start)

4.2.8. State S_4

In S_4, one unit is under inspection (rate α_1) and the other unit is failed and waiting for repair. Upon inspection completion, the inspected unit goes to repair with probability $a$or to replacement with probability b.

Mean sojourn time:

$$\mu_4 = \frac{1}{\alpha_1} \quad (86)$$

One-step transition probabilities:

$$P_{4,9} = a \quad (87)$$

$$P_{4,13} = b \quad (88)$$

4.2.9. State S_5

In S_5, one unit is under inspection (rate α_1) and the other unit is waiting for software upgrading. Upon inspection completion, the server immediately performs software upgrading (priority), so the next state depends on the inspection outcome.

Mean sojourn time:

$$\mu_5 = \frac{1}{\alpha_1} \quad (89)$$

One-step transition probabilities:

$$P_{5,6} = a \quad (90)$$

$$P_{5,7} = b \quad (91)$$

4.2.10. State S_6 (upgrading, waiting for repair)

In S_6, Unit 1 is under software upgrading, and Unit 2 is failed and waiting for hardware repair. Since no unit is operative, the only event is completion of upgrading (rate α_3). After upgrading is completed, Unit 1 becomes operative and the server starts repairing Unit 2, hence the system moves to state S_{10}.

Mean sojourn time:

$$\mu_6 = \frac{1}{\alpha_3} \quad (92)$$

Transition density:

$$q_{6,10}(t) = \alpha_3 e^{-\alpha_3 t},\ t \geq 0 \quad (93)$$

Transition CDF:

$$Q_{6,10}(t) = \int_0^t \alpha_3\, e^{-\alpha_3 u}\, du = 1 - e^{-\alpha_3 t} \quad (94)$$

One-step transition probability:

$$P_{6,10} = Q_{6,10}(\infty) = 1 \quad (95)$$

4.2.11. State S_7 (upgrading, waiting for replacement)

In S_7, Unit 1 is under software upgrading, and Unit 2 is failed and waiting for hardware replacement (this corresponds to a prior inspection decision). Since no unit is operative, the only event is completion of upgrading (rate α_3). After upgrading is completed, Unit 1 becomes operative and the server starts replacing Unit 2, hence the system moves to state S_{11}.

Mean sojourn time:

$$\mu_7 = \frac{1}{\alpha_3} \quad (96)$$

Transition density:

$$q_{7,11}(t) = \alpha_3 e^{-\alpha_3 t},\ t \geq 0 \quad (97)$$

Transition CDF:

$$Q_{7,11}(t) = 1 - e^{-\alpha_3 t} \quad (98)$$

One-step transition probability:

$$P_{7,11} = 1 \quad (99)$$

4.2.12. State S_8 (upgrading, waiting for upgrading)

In S_8, Unit 1 is under software upgrading and Unit 2 is failed and waiting for software upgrading. Since no unit is operative, the only event is completion of upgrading of Unit 1 (rate α_3). After completion, Unit 1 becomes operative and the server starts software upgrading for Unit 2, hence the system moves to state S_2.

Mean sojourn time:

$$\mu_8 = \frac{1}{\alpha_3} \quad (100)$$

Transition density:

$$q_{8,2}(t) = \alpha_3 e^{-\alpha_3 t},\ t \geq 0 \quad (101)$$

Transition CDF:

$$Q_{8,2}(t) = 1 - e^{-\alpha_3 t} \quad (102)$$

One-step transition probability:

$$P_{8,2} = 1 \quad (103)$$

4.2.13. State S_9 (repair, waiting for repair)

In S_9, Unit 1 is under hardware repair and Unit 2 is failed and waiting for hardware repair. Since no unit is operative, the only

event is completion of the ongoing repair (rate α_2). After the repair is completed, Unit 1 becomes operative and the server starts repairing Unit 2, hence the system moves to state S_{10}.

Mean sojourn time:

$$\mu_9 = \frac{1}{\alpha_2} \tag{104}$$

Transition density:

$$q_{9,10}(t) = \alpha_2 e^{-\alpha_2 t},\ t \geq 0 \tag{105}$$

Transition CDF:

$$Q_{9,10}(t) = 1 - e^{-\alpha_2 t} \tag{106}$$

One-step transition probability:

$$P_{9,10} = 1 \tag{107}$$

4.2.14. State S_{12} (preventive maintenance, waiting for repair)

In S_{12}, Unit 1 is under preventive maintenance and Unit 2 is failed and waiting for hardware repair. Since no unit is operative, the only event is completion of preventive maintenance (rate α_5). After completion, Unit 1 becomes operative and the server starts repairing Unit 2, hence the system moves to state S_{10}.

Mean sojourn time:

$$\mu_{12} = \frac{1}{\alpha_5} \tag{108}$$

Transition density:

$$q_{12,10}(t) = \alpha_5 e^{-\alpha_5 t},\ t \geq 0 \tag{109}$$

Transition CDF:

$$Q_{12,10}(t) = 1 - e^{-\alpha_5 t} \tag{110}$$

One-step transition probability:

$$P_{12,10} = 1 \tag{111}$$

4.2.15. State S_{13} (replacement, waiting for repair)

In S_{13}, Unit 1 is under hardware replacement and Unit 2 is failed inactive and waiting for hardware repair. Since no unit is operative, the only event is completion of replacement (rate α_4). After completion, Unit 1 becomes operative and the server starts repairing Unit 2, hence the system moves to state S_{10}.

Mean sojourn time:

$$\mu_{13} = \frac{1}{\alpha_4} \tag{112}$$

Transition density:

$$q_{13,10}(t) = \alpha_4 e^{-\alpha_4 t},\ t \geq 0 \tag{113}$$

Transition CDF:

$$Q_{13,10}(t) = 1 - e^{-\alpha_4 t} \tag{114}$$

One-step transition probability:

$$P_{13,10} = 1 \tag{115}$$

Table 2. Regenerative states: total exit rates, non-zero transition probabilities and mean sojourn times

State	Total Exit Rate D_i	Non-zero Transition Probabilities	Mean Sojourn Time μ_i
S_0	$\lambda_1 + \lambda_2 + \lambda_3 + \beta$	$P_{01} = \frac{\lambda_1}{D_0}$, $P_{02} = \frac{\lambda_2}{D_0}$, $P_{03} = \frac{\lambda_3}{D_0}$, $P_{0,14} = \frac{\beta}{D_0}$	$\frac{1}{D_0}$
S_1	$\alpha_1 + \lambda_1 + \lambda_2$	$P_{1,10} = \frac{a\,\alpha_1}{D_1}$, $P_{1,11} = \frac{b\,\alpha_1}{D_1}$, $P_{14} = \frac{\lambda_1}{D_1}$, $P_{15} = \frac{\lambda_2}{D_1}$	$\frac{1}{D_1}$
S_2	$\alpha_3 + \lambda_1 + \lambda_2$	$P_{20} = \frac{\alpha_3}{D_2}$, $P_{26} = \frac{\lambda_1}{D_2}$, $P_{28} = \frac{\lambda_2}{D_2}$	$\frac{1}{D_2}$
S_3	$\alpha_2 + \lambda_1 + \lambda_2$	$P_{30} = \frac{\alpha_2}{D_3}$, $P_{39} = \frac{\lambda_1}{D_3}$, $P_{36} = \frac{\lambda_2}{D_3}$	$\frac{1}{D_3}$
S_{10}	$\alpha_2 + \lambda_1 + \lambda_2$	$P_{10,0} = \frac{\alpha_2}{D_{10}}$, $P_{10,9} = \frac{\lambda_1}{D_{10}}$, $P_{10,6} = \frac{\lambda_2}{D_{10}}$	$\frac{1}{D_{10}}$
S_{11}	$\alpha_4 + \lambda_1 + \lambda_2$	$P_{11,0} = \frac{\alpha_4}{D_{11}}$, $P_{11,13} = \frac{\lambda_1}{D_{11}}$, $P_{11,7} = \frac{\lambda_2}{D_{11}}$	$\frac{1}{D_{11}}$
S_{14}	$\alpha_5 + \lambda_1 + \lambda_2$	$P_{14,0} = \frac{\alpha_5}{D_{14}}$, $P_{14,12} = \frac{(\lambda_1+\lambda_2)}{D_{14}}$	$\frac{1}{D_{14}}$

Table 3. Non-regenerative states: exit rates, next states and mean sojourn times

State	Exit Rate	Next State	Mean Sojourn Time μ_i
S_4	α_1	S_9 (prob. a), S_{13}(prob. b)	$\frac{1}{\alpha_1}$
S_5	α_1	S_6 (prob. a), S_7(prob. b)	$\frac{1}{\alpha_1}$
S_6	α_3	S_{10}	$\frac{1}{\alpha_3}$
S_7	α_3	S_{11}	$\frac{1}{\alpha_3}$
S_8	α_3	S_2	$\frac{1}{\alpha_3}$
S_9	α_2	S_{10}	$\frac{1}{\alpha_2}$
S_{12}	α_5	S_{10}	$\frac{1}{\alpha_5}$
S_{13}	α_4	S_{10}	$\frac{1}{\alpha_4}$

5. Reliability and mean time to system failure (MTSF)

To determine the system reliability and the Mean Time to System Failure (MTSF), the failed states $S_4, S_5, S_6, S_7, S_8, S_9, S_{12}$, and S_{13} are treated as absorbing states. The reliability equations (116) – (122) are derived using the regenerative point technique combined with first-step analysis. At each regenerative state, all possible direct transitions are identified. Conditioning on the first transition yields renewal-type integral equations via Stieltjes convolutions of $q_{ij}(t)$. Transitions to failed absorbing

states contribute zero, whereas transitions to non-failed states contribute terms of the form $q_{ij}(t) * R_j(t)$. The resulting system is solved in the Laplace transform domain, and the MTSF is subsequently obtained by applying the final-value theorem.

Let $R_i(t)$have the probability that the system is operational at time t, given that it started in state $S_i$at $t = 0$.

Reliability equations

$$R_0(t) = Z_0(t) + q_{01}(t)©R_1(t) + q_{02}(t)©R_2(t) + q_{03}(t)©R_3(t) + q_{0,14}(t)©R_{14}(t) \quad (116)$$

$$R_1(t) = Z_1(t) + q_{1,10}(t)©R_{10}(t) + q_{1,11}(t)©R_{11}(t) \quad (117)$$

$$R_2(t) = Z_2(t) + q_{20}(t)©R_0(t) \quad (118)$$

$$R_3(t) = Z_3(t) + q_{30}(t)©R_0(t) \quad (119)$$

$$R_{10}(t) = Z_{10}(t) + q_{10,0}(t)©R_0(t) \quad (120)$$

$$R_{11}(t) = Z_{11}(t) + q_{11,0}(t)©R_0(t) \quad (121)$$

$$R_{14}(t) = Z_{14}(t) + q_{14,0}(t)©R_0(t) \quad (122)$$

where:

- $Z_i(t)$is the probability that the system will remain in state $S_i$without transitioning to any other state,
- ©denotes the Stieltjes convolution.

Laplace transform and MTSF

Taking the Laplace Transform of equations (116) – (122), we obtain a system of algebraic equations in $R_i^*(s)$.

The Mean Time to System Failure (MTSF) is obtained by:

$$\text{MTSF} = \lim_{s\to 0} R_0^*(s) = \frac{N}{D} \quad (123)$$

Final expression for MTSF

$$N = \mu_0 + P_{01}(\mu_1 + P_{1,10}\mu_{10} + P_{1,11}\mu_{11}) + P_{02}\mu_2 + P_{03}\mu_3 + P_{0,14}\mu_{14} \quad (124)$$

$$D = 1 - P_{01}(P_{1,10}P_{10,0} + P_{1,11}P_{11,0}) - P_{02}P_{20} - P_{03}P_{30} - P_{0,14}P_{14,0} \quad (125)$$

Hence,

$$\text{MTSF} = \frac{\mu_0 + P_{01}\mu_1 + P_{02}\mu_2 + P_{03}\mu_3 + P_{0,14}\mu_{14} + P_{01}P_{1,10}\mu_{10} + P_{01}P_{1,11}\mu_{11}}{1 - [P_{02}P_{20} + P_{03}P_{30} + P_{0,14}P_{14,0} + P_{01}(P_{1,10}P_{10,0} + P_{1,11}P_{11,0})]} \quad (126)$$

This expression represents the expected operating time of the system before entering any failed state for the first time.

6. Availability analysis

To determine the system availability, the regenerative point technique combined with first-step analysis is employed. For each regenerative state i, all feasible transitions are identified and the corresponding availability equations (128) – (134) are established. Transitions to operative states $j$contribute terms of the form $q_{ij}(t) * A_j(t)$, where " $*$ " denotes the Stieltjes convolution operator. The probability that the system remains operative prior to the first transition is represented by $[1 - H_i(t)]$. This yields a set of renewal-type integral equations expressed through Stieltjes convolutions. The resulting system is transformed into an algebraic form using Laplace transforms, and the steady-state availability is subsequently obtained through the final-value theorem:

$$A = \lim_{s\to 0} s\,\tilde{A}_0(s) \quad (127)$$

To determine the steady-state availability of the system, let $A_i(t)$ denote the probability that the system is operational at time (t), given that it starts in state $S_i$at $t = 0$.The operational states of the system correspond to the regenerative states

$$S_0,\ S_1,\ S_2,\ S_3,\ S_{10},\ S_{11}\ and\ S_{14}$$

Availability equations

$$A_0(t) = Z_0(t) + q_{01}(t)©A_1(t) + q_{02}(t)©A_2(t) + q_{03}(t)©A_3(t) + q_{0,14}(t)©A_{14}(t) \quad (128)$$

$$A_1(t) = Z_1(t) + q_{1,10}(t)©A_{10}(t) + q_{1,11}(t)©A_{11}(t) \quad (129)$$

$$A_2(t) = Z_2(t) + q_{20}(t)©A_0(t) \quad (130)$$

$$A_3(t) = Z_3(t) + q_{30}(t)©A_0(t) \quad (131)$$

$$A_{10}(t) = Z_{10}(t) + q_{10,0}(t)©A_0(t) \quad (132)$$

$$A_{11}(t) = Z_{11}(t) + q_{11,0}(t)©A_0(t) \quad (133)$$

$$A_{14}(t) = Z_{14}(t) + q_{14,0}(t)©A_0(t) \quad (134)$$

where:

$Z_i(t)$is the probability that the process remains in state S_i without transition to any other state, and © denotes the Stieltjes convolution.

Laplace transform and solution

Taking Laplace transforms of equations (128) – (134), we obtain a system of algebraic equations in $A_i^*(s)$. Solving the system for $A_0^*(s)$yields:

$$A_0^*(s) = \frac{N_A(s)}{D(s)} \quad (135)$$

where:

$$N_A(s) = Z_0^*(s) + q_{01}^*(s)\left[Z_1^*(s) + q_{1,10}^*(s)Z_{10}^*(s) + q_{1,11}^*(s)Z_{11}^*(s)\right] + q_{02}^*(s)Z_2^*(s) + q_{03}^*(s)Z_3^*(s) + q_{0,14}^*(s)Z_{14}^*(s) \quad (136)$$

and

$$D(s) = 1 - q_{01}^*(s)\left[q_{1,10}^*(s)q_{10,0}^*(s) + q_{1,11}^*(s)q_{11,0}^*(s)\right] - q_{02}^*(s)q_{20}^*(s) - q_{03}^*(s)q_{30}^*(s) - q_{0,14}^*(s)q_{14,0}^*(s) \quad (137)$$

Steady-state availability

The steady-state availability is obtained by the final value theorem:

$$A = \lim_{t\to\infty} A_0(t) = \lim_{s\to 0} sA_0^*(s) \quad (138)$$

Using the standard limits:

$$\lim_{s\to 0} sZ_i^*(s) = P(\text{system is up while in } S_i) \quad (139)$$

and for up (regenerative) states the system is operational throughout the sojourn time, hence:

$\lim_{s\to 0} sZ_i^*(s) = 1 \text{for } S_i \in \mathcal{E}$

Final steady-state availability (expanded form)

$$A = \frac{U}{U+W} \quad (140)$$

where:

$$U = \mu_0 + P_{01}\mu_1 + P_{02}\mu_2 + P_{03}\mu_3 + P_{0,14}\mu_{14} + P_{01}P_{1,10}\mu_{10} + P_{01}P_{1,11}\mu_{11} \quad (141)$$

and

$$W = P_{01}[P_{1,4}(\mu_4 + a\mu_9 + (1-a)\mu_{13}) + P_{1,5}(\mu_5 + a\mu_6 + (1-a)\mu_7) + P_{1,10}(P_{10,9}\mu_9 + P_{10,6}\mu_6) + P_{1,11}(P_{11,13}\mu_{13} + P_{11,7}\mu_7)] + P_{02}(P_{2,6}\mu_6 + P_{2,8}\mu_8) + P_{03}(P_{3,9}\mu_9 + P_{3,6}\mu_6) + P_{0,14}(P_{14,12}\mu_{12}) \quad (142)$$

7. Busy period analysis of the server

7.1. Introduction

In this section, we determine the long-run fractions of time during which the server is engaged in different maintenance activities. Since the system has a single server responsible for inspection, repair, replacement, software upgrading, and preventive maintenance, evaluating the server utilization is essential for economic analysis.

Let

- B_I= steady-state probability that the server is busy with hardware inspection.
- B_R= steady-state probability that the server is busy with hardware repair.
- B_P= steady-state probability that the server is busy in hardware replacement.
- B_U= steady-state probability that the server is busy in software upgrading.
- B_M= steady-state probability that the server is busy with preventive maintenance.

Using the regenerative cycle starting from S_0, the busy fraction of each activity is:

$$B_k = \frac{T_k}{T}, \text{ for } k \in \{I, R, P, U, M\}$$

where:

- $T$is the expected cycle length (one cycle starting from S_0),
- $T_k$is the expected time spent by the server in activity $k$during the cycle.

The expected cycle length is

$$T = U + W \quad (143)$$

with $U$and Wgiven in Eqs. (141) and (142).

(1) Inspection Busy Fraction B_I

Inspection occurs in states S_1, S_4, S_5. Hence:

$$T_I = P_{01}\mu_1 + P_{01}P_{1,4}\mu_4 + P_{01}P_{1,5}\mu_5 \quad (144)$$

$$B_I = \frac{T_I}{T} \quad (145)$$

(2) Hardware Repair Busy Fraction B_R

Repair occurs in states S_3, S_{10}, S_9. The server spends repair time in S_9through all possible entries into S_9. Thus:

$$T_R = P_{03}\mu_3 + P_{01}P_{1,10}\mu_{10} + P_{01}P_{1,4}\,a\,\mu_9 + P_{03}P_{3,9}\mu_9 + P_{01}P_{1,10}P_{10,9}\mu_9 \quad (146)$$

$$B_R = \frac{T_R}{T} \quad (147)$$

(3) Hardware Replacement Busy Fraction B_P

Replacement occurs in states S_{11}, S_{13}. Hence:

$$T_P = P_{01}P_{1,11}\mu_{11} + P_{01}P_{1,4}(1-a)\mu_{13} + P_{01}P_{1,11}P_{11,13}\mu_{13} \quad (148)$$

$$B_P = \frac{T_P}{T} \quad (149)$$

(4) Software Upgrading Busy Fraction B_U

Upgrading occurs in states S_2, S_6, S_7, S_8. Therefore:

$$T_U = P_{02}\mu_2 + P_{02}(P_{2,6}\mu_6 + P_{2,8}\mu_8) + P_{03}P_{3,6}\mu_6 + P_{01}P_{1,10}P_{10,6}\mu_6 + P_{01}P_{1,5}(a\mu_6 + (1-a)\mu_7) + P_{01}P_{1,11}P_{11,7}\mu_7 \quad (150)$$

$$B_U = \frac{T_U}{T} \quad (151)$$

(5) Preventive Maintenance Busy Fraction B_M

PM occurs in states $S_{14}$and S_{12}. Hence:

$$T_M = P_{0,14}\mu_{14} + P_{0,14}P_{14,12}\mu_{12} \quad (152)$$

$$B_M = \frac{T_M}{T} \quad (153)$$

8. Profit analysis

8.1. Introduction

In this section, we evaluate the economic performance of the proposed warm standby system. The expected profit is obtained by combining the revenue earned during system uptime with the expenses incurred due to inspection, repair, replacement, software upgrading, and preventive maintenance activities performed by the single server.

8.2. Cost and revenue parameters

Let:

- K_0: revenue per unit up-time of the system.
- C_I: cost per unit time when the server performs hardware inspection.
- C_R: cost per unit time when the server performs hardware repair.
- C_P: cost per unit time when the server performs hardware replacement.
- C_U: **cost** per unit time when the server performs software upgrading.
- C_M : cost per unit time when the server performs preventive maintenance.
- C_D : penalty cost per unit down-time (optional, if downtime cost is considered).

Let:

- A: steady-state availability of the system (Section 6).
- B_I, B_R, B_P, B_U, B_M : steady-state busy fractions of the server (Section 7).

8.3. Expected profit per unit time

The expected profit per unit time in steady state is defined as:

$$\text{EP} = \text{Expected revenue rate} - \text{Expected cost rate} \quad (154)$$

Since revenue is generated only when the system is operational, the expected revenue rate is:

$$\text{Revenue rate} = K_0 A \quad (155)$$

The expected maintenance cost rate is the weighted sum of the busy server fractions:

$$\text{Maintenance cost rate} = C_I B_I + C_R B_R + C_P B_P + C_U B_U + C_M B_M \quad (156)$$

If a downtime penalty is included, the expected downtime cost rate is:

$$\text{Downtime cost rate} = C_D(1 - A) \quad (157)$$

Therefore, the expected profit per unit time becomes:

$$\text{EP} = K_0 A - (C_I B_I + C_R B_R + C_P B_P + C_U B_U + C_M B_M) - C_D(1 - A) \quad (158)$$

8.4. Special case (if downtime penalty is not considered)

If the cost of downtime is ignored (i.e., $C_D = 0$), then (158) reduces to:

$$\text{EP} = K_0 A - (C_I B_I + C_R B_R + C_P B_P + C_U B_U + C_M B_M) \quad (159)$$

8.5. Interpretation

Equation (158) provides a direct quantitative measure of the profitability of the proposed maintenance policy. It explicitly captures:

- the benefit of higher availability through A,
- the influence of the preemptive software priority through B_U,
- and the operational and maintenance expenses through the busy fractions of the server.

Final expanded expected profit per unit time

Let:

$$T = U + W \quad (160)$$

Then:

$$A = \frac{U}{T}, \text{ and } B_k = \frac{T_k}{T}$$

So, the expected profit per unit time is:

$$\text{EP} = \frac{k_0 U - (C_I T_I + C_R T_R + C_P T_P + C_U T_U + C_M T_M) - C_D W}{U + W} \quad (161)$$

Where U and W are:

Up-time term

$$U = \mu_0 + P_{01}\mu_1 + P_{02}\mu_2 + P_{03}\mu_3 + P_{0,14}\mu_{14} + P_{01}P_{1,10}\mu_{10} + P_{01}P_{1,11}\mu_{11} \quad (162)$$

Down-time term

$$\begin{aligned} W = {} & P_{01}[P_{1,4}(\mu_4 + a\mu_9 + (1-a)\mu_{13} \\ & + P_{1,5}(\mu_5 + a\mu_6 + (1-a)\mu_7) \\ & + P_{1,10}(P_{10,9}\mu_9 + P_{10,6}\mu_6) \\ & + P_{1,11}(P_{11,13}\mu_{13} + P_{11,7}\mu_7)] \\ & + P_{02}(P_{2,6}\mu_6 + P_{2,8}\mu_8) + P_{03}(P_{3,9}\mu_9 \\ & + P_{3,6}\mu_6) + P_{0,14}(P_{14,12}\mu_{12}) \end{aligned} \quad (163)$$

And the activity-time terms T_I, T_R, T_P, T_U, T_M are:

Inspection time

$$T_I = P_{01}\mu_1 + P_{01}P_{1,4}\mu_4 + P_{01}P_{1,5}\mu_5 \quad (164)$$

Hardware repair time

$$T_R = P_{03}\mu_3 + P_{01}P_{1,10}\mu_{10} + P_{01}P_{1,4}a\mu_9 + P_{03}P_{3,9}\mu_9 + P_{01}P_{1,10}P_{10,9}\mu_9 \quad (165)$$

Replacement time

$$T_P = P_{01}P_{1,11}\mu_{11} + P_{01}P_{1,4}(1-a)\mu_{13} + P_{01}P_{1,11}P_{11,13}\mu_{13} \quad (166)$$

Software upgrading time

$$T_U = P_{02}\mu_2 + P_{02}(P_{2,6}\mu_6 + P_{2,8}\mu_8) + P_{03}P_{3,6}\mu_6 + P_{01}P_{1,10}P_{10,6}\mu_6 + P_{01}P_{1,5}(a\mu_6 + (1-a)\mu_7) + P_{01}P_{1,11}P_{11,7}\mu_7 \quad (167)$$

Preventive maintenance time

$$T_M = P_{0,14}\mu_{14} + P_{0,14}P_{14,12}\mu_{12} \quad (168)$$

Special case (no downtime penalty)

If $C_D = 0$, then (161) reduces to:

$$\text{EP} = \frac{k_0 U - (C_I T_I + C_R T_R + C_P T_P + C_U T_U + C_M T_M)}{U+W} \quad (169)$$

9. Numerical illustration

To demonstrate the applicability of the proposed model, we present a numerical illustration by assigning representative values to the model parameters (failure rates, service rates, and cost parameters). Using the analytical expressions derived in Sections 4–8, we compute the key performance measures: MTSF, steady-state availability, server busy periods, and expected profit per unit time.

9.1. Input data (baseline parameter set)

The numerical parameter values adopted in this study are hypothetical but consistent with ranges commonly employed in the reliability literature [4,6,12,24]. Their purpose is to demonstrate the applicability of the proposed model rather than represent a specific industrial installation. Failure rates are of the order 10^{-3} per hour reflecting high-reliability hardware, while service rates are higher to represent efficient maintenance.

Failure / maintenance rates (per hour):

- $\lambda_1 = 0.002, \lambda_2 = 0.0015, \lambda_3 = 0.001, \beta = 0.0008$
- $\alpha_1 = 0.5, \alpha_2 = 0.4, \alpha_3 = 0.6, \alpha_4 = 0.3, \alpha_5 = 0.2$
- $a = 0.7$ (So $b = 1 - a = 0.3$)

Revenue / cost parameters (per hour):

- Revenue rate: $K_0 = 100$
- Inspection cost: $C_I = 5$
- Repair cost: $C_R = 12$
- Replacement cost: $C_P = 20$
- Upgrading cost: $C_U = 8$
- PM cost: $C_M = 6$
- Downtime penalty: $C_D = 30$

9.2. Numerical results

Using the closed-form expressions obtained in the previous sections, the following results are obtained:

Table 4. Baseline numerical results.

Measure	Value
Mean Time to System Failure (MTSF)	15919.0794 hours
Steady-state Availability A	0.9997984
Server busy fraction in Inspection B_I	0.0039274
Server busy fraction in Repair B_R	0.0058591
Server busy fraction in Replacement B_P	0.0019482
Server busy fraction in Upgrading B_U	0.0024837
Server busy fraction in PM B_M	0.0039274

The baseline results presented in Table 4 show that the mean time to system failure is estimated at 15,919 hours, indicating a relatively long operational lifetime under the adopted maintenance configuration. The steady-state availability is calculated as 0.999798, which corresponds to a very limited annual downtime. In addition, the total server utilization remains low (approximately 1.81%), implying that maintenance activities do not create operational bottlenecks. The expected profit per unit time is 99.80145, reflecting the combined effect of high availability and moderate maintenance costs. Overall, these results demonstrate the effectiveness of the proposed maintenance structure under the selected baseline parameters. The equality of B_I and B_M in Table 4 is exact rather than a rounding artefact. Because $D_1 = \alpha_1 + \lambda_1 + \lambda_2$, Eq. (144) reduces to $T_I = P_{01}/\alpha_1$, and, because $D_{14} = \alpha_5 + \lambda_1 + \lambda_2$, Eq. (152) reduces to $T_M = P_{0,14}/\alpha_5$. Hence $B_I/B_M = \lambda_1\alpha_5/(\beta\alpha_1)$, which equals 1 for the baseline values because $\lambda_1/\alpha_1 = \beta/\alpha_5 = 0.004$.

9.3. Sensitivity analysis with respect to the preventive maintenance rate β

In this subsection, we study the effect of the preventive maintenance initiation rate β on the key performance measures. For each value of β, the values of MTSF, steady-state availability A, expected profit per unit time (EP), and the server busy fraction in preventive maintenance B_M are computed using the closed-form expressions derived in Sections 5–8.

Table 5 illustrates the impact of varying the preventive maintenance rate β on system performance. It is observed that increasing β leads to a monotonic decrease in MTSF, indicating

that more frequent preventive interventions increase the total scheduled downtime, thereby reducing the effective operating lifetime before failure. Similarly, the steady-state availability A exhibits a gradual decline as β increases, reflecting the inherent trade-off between preventive actions and operational continuity. At the same time, the server busy fraction associated with preventive maintenance BM increases proportionally with β, confirming the intensified utilization of maintenance resources. The expected profit per unit time (EP) decreases slightly as β increases, primarily due to higher maintenance expenditures and reduced productive uptime. Overall, these results highlight the inherent trade-off between reliability management through preventive maintenance and the economic efficiency of the system.

Table 5. Sensitivity analysis with respect to the preventive maintenance rate β.

β	MTSF (hours)	Availability (A)	PM busy fraction B_M	EP (per hour)
0.0000	20203.4377	0.999865	0.000000	99.833140
0.0004	17803.0987	0.999832	0.001968	99.817264
0.0008	15919.0794	0.999798	0.003927	99.801450
0.0012	14400.9644	0.999765	0.005880	99.785698
0.0016	13151.5928	0.999732	0.007824	99.770008
0.0020	12105.4060	0.999699	0.009761	99.754379
0.0030	10108.5503	0.999617	0.014570	99.715571

9.4. Sensitivity analysis with respect to the software upgrading rate α_3

In this subsection, a sensitivity analysis is performed to evaluate the influence of the software upgrading rate α_3 on the principal system performance measures. All other parameters are maintained at their baseline values. For varying levels of α_3, the MTSF, steady-state availability A, expected profit per unit time (EP), and the server busy fraction associated with software upgrading B_U are computed using the analytical expressions derived in Sections 5–8.

* Baseline parameter value.

Table 6. Sensitivity analysis with respect to the software upgrading rate α_3.

α_3	MTSF	Availability (A)	Upgrading busy fraction B_U	EP
0.20	12621.49	0.9996296	0.0074	99.74081
0.30	14066.52	0.9997276	0.0050	99.77285
0.40	14931.79	0.9997665	0.0037	99.78759
0.50	15507.92	0.9997865	0.0030	99.79601
0.60*	15919.08	0.9997984	0.0025	99.80145
0.80	16466.84	0.9998118	0.0019	99.80805
1.00	16815.13	0.9998190	0.0015	99.81191

9.5. Comparative study: preemptive-resume priority vs. non-priority policy

To evaluate the practical benefit of the preemptive-resume priority discipline, a comparative study is conducted between the proposed model and a non-priority alternative. In the non-priority policy, software upgrading does not interrupt ongoing hardware-related activities; instead, the server completes its current hardware task before initiating software upgrading. All other model parameters and assumptions remain identical. The baseline parameter set of Section 9.1 is used for both models. The results are summarized in Table 7.

Table 7. Comparative results: preemptive-resume priority vs. non-priority policy.

Performance Measure	Proposed (Priority)	Non-Priority	Improvement
MTSF (hours)	15919.0794	15919.0794	—
Steady-state Availability (A)	0.9997984	0.9997387	+0.0000597
Server busy fraction – Upgrading (BU)	0.0024837	0.0025433	−0.0000596
Expected Profit per unit time (EP)	99.80145	99.79322	+0.00823

Table 7 confirms that the preemptive-resume priority policy yields measurable improvements over the non-priority alternative. The steady-state availability increases from 0.9997387 to 0.9997984, and the expected profit per unit time improves from 99.79322 to 99.80145. These gains are attributable to the faster resolution of software failures under the priority discipline, which reduces the duration of system downtime and lowers the server busy fraction associated with software upgrading from 0.0025433 to 0.0024837. The MTSF remains unchanged between the two policies, because the two policies differ only in the order of service when both units are failed. Such situations arise only in the down states S_4–S_9, S_{12} and S_{13}, which are treated as absorbing in the MTSF analysis; the up states S_0, S_1, S_2, S_3, S_{10}, S_{11} and S_{14}, their exit rates and the transition probabilities out of them are identical under both policies, so Eqs. (116)–(126) are unchanged. Overall, these results demonstrate that the preemptive-resume priority strategy provides both operational and economic advantages compared to a non-priority maintenance policy.

Having examined the influence of key maintenance

parameters on system performance, the principal insights and implications of the developed semi-Markov model are summarized in the concluding section.

10. Conclusion

In this study, a stochastic reliability model for a warm standby computer system incorporating preventive maintenance and a preemptive-resume priority discipline for software upgrading has been developed within a semi-Markov framework. The model integrates hardware and software failures, inspection-based repair and replacement decisions, and economic performance considerations, providing a comprehensive analytical structure for long-run system evaluation.

Closed-form expressions for key performance measures, including the *Mean Time to System Failure (MTSF)*, steady-state availability (A), maintenance resource utilization, and the *Mean Profit per unit time (EP)*, were derived. The numerical illustration under the baseline parameter configuration specified in Section 9.1 demonstrates that the proposed maintenance structure achieves high availability and stable operational performance.

The sensitivity analyses reveal important performance trade-offs. Increasing the preventive maintenance rate reduces the *Mean Time to System Failure (MTSF)* and slightly decreases profitability due to additional scheduled downtime. In contrast, increasing the software upgrading rate (α_3) improves both *MTSF* and availability (A), while enhancing the *Mean Profit per unit time (EP)* by reducing recovery durations and upgrading-related downtime. These results clearly demonstrate that the adoption of a preemptive-resume priority discipline for software upgrading positively influences both reliability and economic performance. Furthermore, the comparative study presented in Section 9.5 provides quantitative evidence of the superiority of the proposed policy: under the baseline parameter set, the preemptive-resume priority model achieves a steady-state availability of 0.9997984 and an expected profit per unit time of 99.80145, compared to 0.9997387 and 99.79322, respectively, under the non-priority alternative. The reduction in the software upgrading busy fraction from 0.0025433 to 0.0024837 further confirms that prioritizing software upgrading leads to faster recovery and reduced resource utilization.

The developed framework offers a flexible foundation for further extensions. Future research may consider common-cause and dependent failures, multiple maintenance servers, optimization of preventive maintenance policies, and more complex priority structures to achieve an improved cost–reliability balance. Extensions may also incorporate imperfect repair and replacement actions, state-dependent failure rates, or finite-buffer queue effects to more closely reflect the operational complexity of modern software-intensive systems.

Acknowledgment

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University (https://www.qu.edu.sa) for financial support (QU-APC-2026).

Note: During the preparation of this manuscript, the author used ChatGPT (OpenAI) and Claude (Anthropic) for language editing, formatting and reference checking. The author reviewed and verified all outputs and takes full responsibility for the content of the publication.

References

1. Friedman MA, Tran PY, Goddard PL. Reliability techniques for combined hardware and software systems. Final technical report RL-TR-92-15. Griffiss AFB (NY): Rome Laboratory; 1992.
2. Welke SR, Johnson BW, Aylor JH. Reliability modeling of hardware/software systems. *IEEE Transactions on Reliability* 1995; 44(3): 413-418. https://doi.org/10.1109/24.406575.
3. Lai CD, Xie M, Poh KL, Dai YS, Yang P. A model for availability analysis of distributed software/hardware systems. *Information and Software Technology* 2002; 44(6): 343-350. https://doi.org/10.1016/S0950-5849(02)00007-1.
4. Malik SC, Anand J. Reliability and economic analysis of a computer system with independent hardware and software failures. *Bulletin of Pure and Applied Sciences* 2010; 29E(1): 141-153.
5. Munday VJ, Malik SC. Reliability measures of a computer system with priority for repair and hardware redundancy. *International Journal*

of Computer Applications 2015; 125(8): 32-37. https://doi.org/10.5120/ijca2015906125.

6. Kumar A, Anand J, Malik SC. Stochastic modeling of a computer system with priority to up-gradation of software over hardware repair activities. *International Journal of Agricultural and Statistical Sciences* 2013; 9(1): 117-126.
7. Goel LR, Gupta R, Singh SK. Cost-benefit analysis of a 2-unit warm-standby system with inspection, repair, and post repair. *IEEE Transactions on Reliability* 1986; 35(1): 70. https://doi.org/10.1109/TR.1986.4335351.
8. Mokaddis GS, El-Sherbeny MS, Ayid YM. Stochastic behavior of a two-unit warm standby system with two types of repairmen and patience time. *Journal of Mathematics and Statistics* 2009; 5(1): 42-46. https://doi.org/10.3844/jmssp.2009.42.46.
9. Mahmoud MAW, Hamouda HM. Stochastic behavior of a two-unit standby system with two types of repairman, patience time and three types of failures. *International Journal of Applied Mathematics* 2010; 23(1): 99-111.
10. Rizwan SM, Khurana V, Taneja G. Reliability analysis of a hot standby industrial system. *International Journal of Modelling and Simulation* 2010; 30(3): 315-322.
11. Kumar J. Analysis of a computer system with hardware repair and software replacement subject to conditional arrival time of the server. *Journal of Reliability and Statistical Studies* 2017; 10(1): 113-122.
12. Temraz N. Analysis of discrete-time semi-Markov model for a two-unit warm standby system subject to degradation. *Journal of Quality in Maintenance Engineering* 2022; 28(1): 86-101. https://doi.org/10.1108/JQME-06-2020-0055.
13. Qi F, Wang Y, Wang X, Wu Y. A novel condition-based maintenance policy considering imperfect switching for warm standby systems. *Quality and Reliability Engineering International* 2024; 40(3): 1148-1164. https://doi.org/10.1002/qre.3465.
14. Danjuma MU, Yusuf B, Yusuf I. Reliability, availability, maintainability, and dependability analysis of cold standby series-parallel system. *Journal of Computational and Cognitive Engineering* 2022; 1(4): 193-200. https://doi.org/10.47852/bonviewJCCE2202144.
15. Ragheb NM, Solouma E, Alahmari AA, Saber S. Reliability and availability analysis of a two-unit cold standby system with imperfect switching. *Axioms* 2025; 14(8): 589. https://doi.org/10.3390/axioms14080589.
16. Malik SC, Bhardwaj RK, Grewal AS. Probabilistic analysis of a system of two non-identical parallel units with priority to repair subject to inspection. *Journal of Reliability and Statistical Studies* 2010; 3(1): 1-11.
17. Kumar J, Kadyan MS, Malik SC. Cost analysis of a two-unit cold standby system subject to degradation, inspection and priority. *Eksploatacja i Niezawodnosc – Maintenance and Reliability* 2012; 14(4): 278-283.
18. Monika, Chopra G. Profitability analysis of a food industrial system having make-and-pack production strategy with priority basis repair. *Reliability: Theory & Applications* 2023; 18(2): 441-455. https://doi.org/10.24412/1932-2321-2023-273-441-455.
19. Dabas N, Rathee R. Parallel system analysis with priority and inspection using semi-Markov approach. *Reliability: Theory & Applications* 2022; 17(2): 56-66.
20. Wang J, Ma X, Yang L, Qiu Q, Shang L, Wang J. A hybrid inspection-replacement policy for multi-stage degradation considering imperfect inspection with variable probabilities. *Reliability Engineering & System Safety* 2024; 241: 109629. https://doi.org/10.1016/j.ress.2023.109629.
21. Koutras VP. A Markov regenerative process model for the dependability and performance of a two-unit multi-state system under maintenance. *Reliability Engineering & System Safety* 2023; 238: 109433. https://doi.org/10.1016/j.ress.2023.109433.
22. Mittal N, Ivanova N, Jain V, Vishnevsky V. Reliability and availability analysis of high-altitude platform stations through semi-Markov modeling. *Reliability Engineering & System Safety* 2024; 252: 110419. https://doi.org/10.1016/j.ress.2024.110419.
23. Bai J, Chang X, Machida F, Jiang L, Han Z, Trivedi KS. Impact of service function aging on the dependability for MEC service function chain. *IEEE Transactions on Dependable and Secure Computing* 2023; 20(4): 2811-2824. https://doi.org/10.1109/TDSC.2022.3150782.
24. Yadav AD, Nandal N, Malik SC. Markov approach for reliability and availability analysis of a four unit repairable system. *Reliability: Theory & Applications* 2023; 18(1): 193-205.
25. Shivani, Ram M, Goyal N, Kumar A. Analysis of series-parallel system's sensitivity in context of components failures. *RAIRO - Operations Research* 2023; 57(4): 2131-2149. https://doi.org/10.1051/ro/2023068.
26. Zhang C, Yang J, Li M, Wang N. Reliability analysis of a two-dimensional linear consecutive-(r,s)-out-of-(m,n): F repairable system. *Reliability Engineering & System Safety* 2024; 242: 109792. https://doi.org/10.1016/j.ress.2023.109792.