International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 1, 417-438, 2026
https://doi.org/10.33889/IJMEMS.2026.11.1.018

Steady-State Probability Distribution in Machine Repair Systems with Unreliable Repair Servers and Multi-Phase Recovery

Supriya Maheshwari
Department of Mathematics,
Noida Institute of Engineering &Technology, Greater Noida, Uttar Pradesh, India.
E-mail: supmaheshwari@gmail.com

Masood Alam
Department of Math & IT, CPS,
Sultan Qaboos University, Muscat, Oman.
E-mail: syedmasoodalam@gmail.com

Khursheed Alam
Department of Mathematics & Data Science,
The A. H. Siddiqi Center for Advance Research in Applied Mathematics & Physics (CARAMP),
Sharda University, Uttar Pradesh, India.
Corresponding author: khursheed.alam@sharda.ac.in

(Received on May 12, 2025; Revised on July 6, 2025 & August 29, 2025 & October 9, 2025; Accepted on October 13, 2025)

Abstract
This study provides a mathematical analysis of the steady-state probability distribution in a machine repair system characterized by threshold recovery, unreliable repair servers, and phase-type repairs. A series of steady-state equations shows the possibility of distinct failure states in the system. The number of broken machines can be ranging from 0 to N. The recovery policy is based on a threshold, which means that repairs only start when failures reach above a certain level T. This is the greatest way to use resources. Repair servers, on the other hand, aren't always trustworthy, which means that there is a potential of failure that takes longer to fix, making the system even less reliable. There are various processes in the repair process, and machines go through a number of random repair phases before they can be utilized again. You can find the steady-state distribution by utilizing probability balance equations. This tells us how well the system works, how likely it is to break down, and how reliable it is overall. The results help businesses figure out the best ways to keep their machines running smoothly, which helps them avoid breakdowns, make machines more available, and better plan repairs when things don't go as planned.

Keywords- Steady-state probability distribution, Machine repair system, Threshold recovery policy, Unreliable repair servers, Phase-type repairs.

1. Introduction

In machine repair systems with unreliable repair servers and multi-phase recovery, the steady-state probability distribution is particularly crucial for finding out how well the system works and how dependable it is. A number of machines in these systems work, but they can break down and need to be fixed by a group of servers, which can also break down. The repair procedure uses a multi-phase recovery technique, which means that machines have to go through numerous steps of repair before they work again. The steady-state probability distribution shows us how the system will act over time by telling us the chance of each system state, including how many computers are working, how many are broken, and how the repair server is doing. You may find out critical performance measures like system availability, predicted downtime, and repair efficiency by designing and solving the underlying Markov process or queuing model. When you know how the system works in steady state, you can develop maintenance plans better, use resources better, and make the system more resilient. This makes factories and other industrial settings run more smoothly.

The machine repair system looked at in this paper assumes that all machines in the system are the same, meaning that their failure rate (λ) and operational parameters are statistically the same. By ignoring any changes in how the machines work and assuming that all machines have the same likelihood of failing over time, this makes the model easy to grasp. The repair procedure is likewise built up in phases, so each machine goes through the identical set of repair steps after it is picked for repair. The number of phases P is the same for all machines, and there is a defined repair rate for each phase. This assumption that all repair phases are the same makes sure that the recovery process is always the same and makes it easier to use structured balance equations to analytically find steady-state probabilities. These modeling assumptions are consistent with prior research in the domain and provide a feasible but precise depiction for various industrial maintenance systems.

(*i*) *Systems for retrials and queueing models*: Retrial queueing systems are crucial for modeling service environments where jobs that don't get done try to get done again after a while. Jain and Bhagat (2016) looked into *MX*/*G*/1 retrial queues that have vacation policies and more than one way to get service. Their study explored phase-type repairs and reneging behavior, relevant to flexible, customer-centric services. Liu et al. (2022) enhanced this model by include unstable servers characterized by incomplete coverage and reboot delays. Their paradigm concentrated on authentic scenarios within communication systems, illustrating how server instability might exacerbate issues. Singh et al. (2025) came up with a new *N*-policy for machining systems that have double retrial orbits and use soft computing methods. This study looked into modern problems that require redundant queuing behavior and smart routing when things are unknown. It directly affects the current investigation by simulating probabilistic retry routing in the setting of stochastic repair server behavior.

(*ii*) *Policies for threshold recovery and admission control*: Strategies that use thresholds have been demonstrated to be effective at balancing system downtime and repair burden. Wang et al. (2018) examined retry machine repair methods for server failures and developed threshold recovery to expedite fixes. Their study laid the groundwork for strategic decision-making based on system load. Jain et al. (2017) and Su and Li (2021) emphasized the significance of optimizing the management of admissions and the initiation of services. Jain et al. (2019) proposed protocols for working vacations, during which servers are less effective yet continue to offer service. Su and Li (2021) created rules for lineups with restricted capacity so that they don't get too full and people may keep flowing through them. This work builds on previous ideas by creating models of multi-phase repairs that only happen if a certain level of failure has been reached. This corresponds with the best control aims and reduces down on unneeded interventions.

(*iii*) *Working vacations and servers that go down*: In real life, ensuring server availability entails keeping in mind that servers can go down or break. Jain et al. (2019) investigated fault-tolerant systems under working vacation frameworks employing an *F*-policy, highlighting the trade-offs between system availability and maintenance scheduling. Jain et al. (2020) enhanced this study by using fuzzy logic to handle vacationing and unreliable servers, thus establishing a decision-support framework within probabilistic uncertainty. Wu and Yang (2020) created a dynamic model that combines switching failure mechanisms with server unreliability. Their adaptive control structure allowed for real-time changes to repair procedures, which is especially useful for maintenance systems that are driven by IoT and autonomous technology. These results confirm the existing model's emphasis on server failure probability and its detrimental impact on repair processes.

(*iv*) *System reliability and redundancy analysis*: System-level reliability studies offer comprehensive insights that extend beyond singular repair systems. Wu et al. (2015) and Rykov et al. (2021) concentrated on k-out-of-n systems, offering probabilistic and computational techniques for evaluating component-level reliability. Their contributions are essential for reliability block diagrams and performance modeling with stochastic repair distributions. Hu et al. (2022**)** analyzed the reliability and sensitivity of a repairable *k*-out-of-*n*:*G* system with retrial features and two distinct failure modes. By employing probabilistic and sensitivity analysis, the authors show how system performance is strongly influenced by both the configuration of redundancy (*k*-out-of-*n* structure) and the retrial behavior of failed units. Sanga and Charan (2023) introduce a fuzzy modeling approach for machine repair problems under admission control F-policy and feedback mechanisms**.** Their model incorporates retrial customers and uses fuzzy optimization to handle uncertainties in repair times and costs. Kumar et al. (2024) extend the machine repair literature by explicitly modeling systems with a threshold recovery policy, unreliable servers, and multi-phase repairs**.** Their contribution is to combine three complex features—repair initiation based on a threshold, server unreliability, and sequential multi-phase repair processes—into one unified model. Varshney et al. (2025) presented a comprehensive performance and reliability prediction framework for multi-state service systems with multiple failure modes of unreliable servers. Their analysis employs differential–difference equations and Laplace transforms to derive the steady-state probability distribution of system states. Oszczypała et al. (2024) presented redundancy optimization through the application of Continuous-Time Markov Chains (CTMC) and Monte Carlo simulations. Their method makes complex systems more fault-tolerant and fits with how this study looks at threshold and retrial mechanisms as ways to make systems more resilient.

(*v*) *Application-oriented reliability in communication and UAV systems*: Gao and Wang (2021) investigated a retrial system with mixed standby units and an unreliable repair facility. They develop a Markovian model to evaluate both reliability and availability, capturing how failures interact with repair limitations when standby units are employed. Expanding into application domains, Choudhary and Khaitan nee Gupta (2022) investigated VoIP reliability in high-altitude platforms, and Feng et al. (2022) examined UAV swarm reliability using phased mission modeling. Both works apply reliability techniques to emerging domains, showcasing how system-level optimization contributes to operational success. Mittal et al. (2023) addressed energy-efficient modeling for 5G device reliability, which intersects sustainability goals with performance assurance. These applications illustrate the versatility of reliability models and reinforce the importance of system adaptability, as emphasized in the current research.

Table 1. Literature review summary of reliability, queueing, and control methods and their relevance to the present study.

Author(s)	Year	Focus	Methods/Models	Relevance to current study
Wu et al. (2015)	2015	Reliability assessment in k-out-of-n repairable systems	Transient & computational modeling	Base for system-level reliability modeling
Yen et al. (2015)	2015	Optimal control with removable repairmen	Control theory & optimization	Enhances dynamic staffing strategies
Jain & Bhagat (2016)	2016	Retrial queues with phase repair & reneging	Retrial queue models	Foundation for service retry mechanisms
Jain et al. (2017)	2017	Maintenance under working vacation	Admission control policies	Maintenance strategy modeling
Wang et al. (2018)	2018	Threshold recovery in retrial systems	Optimization under breakdowns	Introduces threshold-based repair concepts
Jain et al. (2019)	2019	F-policy control in fault-tolerant systems	Reliability policies with vacation servers	Extends performance control under failures
Jain et al. (2020)	2020	Fuzzy optimization of unreliable systems	Fuzzy metrics and cost models	Manages uncertainty in server operations
Wu & Yang (2020)	2020	Dynamic control with switching failures	Adaptive models with failure detection	Real-time maintenance response modeling

Table 1 continued…

Su & Li (2021)	2021	Admission control in limited queue systems	Stochastic queue modeling	Influences threshold decisions for service systems
Rykov et al. (2021)	2021	General repair time in k-out-of-n systems	Probabilistic techniques	Adds statistical depth to reliability analysis
Choudhary & Khaitan nee Gupta (2022)	2022	High-altitude platform communication reliability	Dependability modeling	Application to critical communication networks
Feng et al. (2022)	2022	UAV swarm reliability with mission analysis	Phased mission reliability & optimization	Relevant for mission-critical and collaborative systems
Liu et al. (2022)	2022	Retrial queues with imperfect coverage and reboot delay	Queuing with unreliable servers	Key to server management under practical constraints
Mittal et al. (2023)	2023	Energy-efficient modeling for 5G devices	Stochastic modeling	Aligns with resource-efficient system operation
Oszczypała et al. (2024)	2024	Redundancy optimization in k-out-of-n systems	CTMC + Monte Carlo Simulation	Powerful method for resilience and redundancy design
Singh et al. (2025)	2025	N-policy model with double retrial orbits	Stochastic modeling + soft computing	Closely related to current study's structure and methodological enhancements

This study develops a mathematical framework to analyze the steady-state probability distribution of a machine repair system that integrates three realistic features: a threshold recovery policy**,** unreliable repair servers, and a multi-phase repair process. Using a continuous-time Markov chain (CTMC), the system dynamics are modeled through structured balance equations that capture machine failures, repair initiation under threshold conditions, repair progression through multiple phases, and interruptions caused by server breakdowns. The solution is obtained using analytical recursion with normalization, yielding exact steady-state probabilities. These probabilities are then used to derive closed-form performance measures, including system availability, mean number of failed machines, repair load, expected waiting time, server failure probability, and repair throughput. Numerical evaluation, based on illustrative parameter values, demonstrates the feasibility of the model. Graphical analyses show how varying failure rates, repair rates, thresholds, and server reliability affect the probability distributions and performance outcomes. Notably, threshold recovery reduces unnecessary interventions, but higher thresholds increase downtime. Multi-phase repairs capture the sequential nature of real repair operations, though they also prolong expected repair times. Unreliable servers introduce significant risk by blocking repairs during breakdowns, emphasizing the importance of redundancy or preventive strategies. The findings underline the trade-offs between efficiency and resilience in repair systems. While the current model highlights analytical tractability and theoretical insights, future extensions should incorporate empirical parameter validation, optimization of threshold policies, and predictive maintenance strategies to enhance industrial applicability.

The rest of the paper analyzes the steady-state probability distribution in a machine repair system with threshold recovery, unreliable repair servers, and phase-type repairs to optimize maintenance strategies The State Definition and Notations describe key system components and parameters. The Balance Equations establish the probability framework, while the solution of the model derives steady-state probabilities analytically. Performance Measures evaluate system efficiency through metrics like availability, failure probability, and repair load. The Results and Discussion present numerical computations and graphical insights. The Conclusion highlights the impact of unreliable servers and recovery policies on system performance and suggests future research on predictive maintenance.

2. State Definition

To analyze the steady-state probability distribution in a machine repair system with threshold recovery, unreliable repair servers, and multi-phase repairs, certain modeling assumptions are introduced. These assumptions ensure both mathematical tractability and a realistic depiction of industrial maintenance environments.

(a) Homogeneous Machines – The system consists of N identical machines, each subject to independent failures at a constant rate λ. Operational conditions are assumed uniform across all machines.
(b) Markovian Failures – Machine lifetimes follow an exponential distribution, preserving the memoryless property and enabling a continuous-time Markov chain (CTMC) formulation.
(c) Threshold Repair Policy – Repairs are not initiated immediately after a single failure. Instead, maintenance begins only when the number of failed machines reaches or exceeds a fixed threshold TTT. This prevents unnecessary early interventions and improves resource utilization.
(d) Multi-Phase Repair Process – Each repair is modeled as a sequence of r exponential phases with constant transition rates. A machine is returned to service only after successfully completing all phases. This structure reflects real-world repairs, which often involve diagnosis, part replacement, testing, and validation.
(e) Unreliable Repair Servers – Repair servers are not permanently reliable. They may fail during operation at rate α and recover at rate β. When a server fails, ongoing repairs are blocked until the server is restored.
(f) Independence of Events – Failures of machines, transitions between repair phases, and server breakdowns are assumed statistically independent.
(g) Finite Population Constraint – The system size is finite and fixed at N. At any point, the number of failed machines ranges from 0 to N.

The purpose of outlining these assumptions is to provide a structured framework for the mathematical modeling that follows. By explicitly defining machine behavior, repair mechanisms, and server reliability, this section establishes the foundation for the development of balance equations, steady-state probability distributions, and performance metrics. These assumptions, while simplifying, capture essential real-world complexities such as delayed repairs, sequential repair steps, and repair resource failures. This balance between abstraction and realism ensures that the subsequent analysis is both mathematically rigorous and practically relevant.

Let P_n be the steady-state probability that there are n failed machines in the system, where n varies from 0 to N (total number of machines).

The key system components include:
(i) Threshold Recovery Policy: Repairs begin only when a certain number T of failed machines is reached.
(ii) Unreliable Servers: Repairmen may fail with a certain probability and need time to be repaired.
(iii) Phase-Type Repairs: The repair process follows a multi-phase structure.

3. Notations

λ: Failure rate of a single machine.
μ_k: Repair rate in phase k of the repair process.
ν: Rate at which the repair server fails.
γ: Rate at which the failed server is repaired.
r: Number of repair phases.
m: Number of available repairmen.
T: Threshold for starting repairs.

4. Balance Equations

In this section, we formulate the balance equations that govern the dynamics of the machine repair system under study. The objective of this section is to translate the system assumptions into a mathematical framework that captures the interaction between machine failures, repair initiation under threshold conditions, multi-phase repair progression, and server unreliability. Each equation represents a steady-state probability relation, ensuring that the total inflow and outflow of probability in every system state remains balanced.

The balance equations thus serve as the core analytical tool of the model. They reflect how machines transition between operational and failed states, how repair phases evolve under threshold recovery, and how server breakdowns temporarily halt the repair process. By systematically defining these transitions, the equations provide a structured basis for deriving steady-state probabilities and, subsequently, performance measures such as system availability, repair load, and downtime risk.

The derivation of balance equations follows a natural progression from basic to complex cases, allowing the reader to build intuition step by step. We begin with the simplest case, when no machines have failed $(n = 0)$, where the only transition is the failure of a machine at rate $N\lambda$. The next case considers a single failure $(n = 1)$, where further breakdowns may occur but repairs cannot yet begin if the threshold $T > 1$. Once the threshold condition is reached $(n = T)$, repair activities are activated, and the equations incorporate both the phase-type repair rate μ and the possibility of repair server unreliability through failure (α) and recovery (β). The general case $(T < n < N)$ extends this logic, balancing transitions caused by new failures, repair progression, and server status changes. Finally, at the extreme case where all machines are failed $(n = N)$, the system can only recover through repairs if the server is operational. By structuring the balance equations in this sequence from $n = 0$ to $n = N$ the analysis not only ensures mathematical rigor but also makes explicit the role of threshold recovery, multi-phase repair, and server reliability in shaping the steady-state probabilities.

No Failed Machines $(n = 0)$:
When all machines are operational, the only possible event is the failure of one of the N machines at rate $i\lambda$. The system then moves into the one-failure state.

$$P_0 \sum_{i=1}^{N} i\lambda = P_1\mu_1 \tag{1}$$

One Failed Machine $(n = 1)$:
With a single failed machine, transitions can occur either through another failure at rate $\lambda(N-1)$or through repair at rate μ_1.

$$P_1[\lambda(N-1) + \mu_1] = P_0 N\lambda + P_2\mu_2 \tag{2}$$

Failure State with Threshold Condition $(n = T)$:
Once the number of failed machines reaches the threshold T, repair activity begins. The balance equation here captures both the inflow from $T - 1$ failures and the outflow due to repair or additional failures.

$$P_T[\lambda(N-T)] + \sum_{k=1}^{r} \mu_k = P_{T-1}(T\lambda) + P_{T+1}\mu_1 \tag{3}$$

General State Equation for $T < n < N$:
For intermediate states, machines may continue failing at rate $(n-1)\lambda$ while repairs progress through multiple phases. This general equation applies across all such states.

$$P_n[\lambda(N-n) + \sum_{k=1}^{r} \mu_k] = P_{n-1}[(n-1)\lambda] \tag{4}$$

All Machines Failed ($n = N$):
When every machine has failed, the system can only recover through repair completions. No further failures can occur.

$$P_N \sum_{k=1}^{r} \mu_k = P_{N-1}\lambda \tag{5}$$

Unreliable Server Probability (Server Fails at Rate ν):
If the repair server fails at rate ν and recovers at rate γ, the probability of being in a failed-server state is governed by the balance below.

$$P_{fail}\nu = P_{rep}\gamma \tag{6}$$

Failed Server Condition (Repair Blocked due to Server Failure):
When the server is down, ongoing repairs are blocked. The repair process resumes only after recovery. This equation reflects the repair blockage condition.

$$P_n \sum_{k=1}^{r} \mu_k \left(1 - P_{fail}\right) = P_{n-1}[(n-1)\lambda] + P_{n+1}\mu_1 \tag{7}$$

Multi-Phase Repair Consideration:
Repairs are not instantaneous but progress sequentially through r exponential phases. This equation captures the flow between consecutive phases of repair.

$$P_n[\lambda(N-n) + \sum_{k=1}^{r} \mu_k] = P_{n-1}(n\lambda) + \sum_{k=2}^{r} \mu_k P_n^{(k-1)} \tag{8}$$

Repair Completion Transition:
A machine is returned to service only after completing all r repair phases.

$$P_n^{(r)} r = P_n \sum_{k=1}^{r} \mu_k \tag{9}$$

Normalization Condition:
Finally, all steady-state probabilities must sum to one, including machine states, failed server states, and recovery states.

$$\sum_{n=0}^{N} P_n + P_{fail} + P_{rep} = 1 \tag{10}$$

These equations show the steady-state probabilities of the system states. They make sure that machines are fixed according to a threshold policy, taking into account servers that aren't always reliable and phase-type repair transitions.

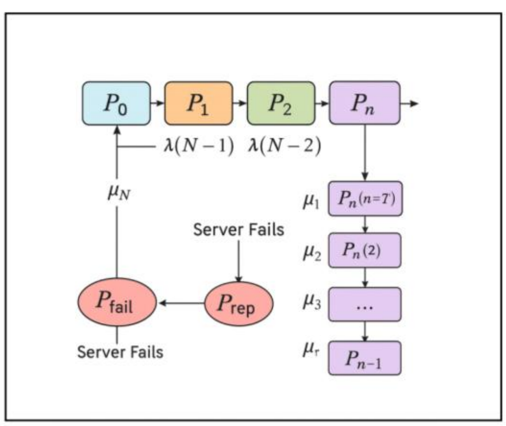

Figure 1. Transition rate diagram of the proposed model.

This model is different from other machine repair system models because it combines three complicated features in a new way: threshold-based recovery, unreliable repair servers, and multi-phase repairs. Traditional models usually assume that repairs will happen right away or all the time after a failure. This model, on the other hand, uses a threshold recovery policy, which means that repairs will only start when the number of broken machines reaches a certain level (T). This saves resources by not intervening too soon. It adds server unreliability to the structure.

Repair servers have a chance of breaking down, which means they need to be fixed before they can work again. This adds another level of random reliance that wasn't in the models before. The model has phase-type repair operations, which implies that each machine repair passes through a number of random stages instead of simply one long repair period. Putting these items together makes a more intricate system of steady-state balancing equations and performance measurements. This is because real-world industrial settings are more complex and realistic than models that just show one or two types of repair dynamics. This strategy helps us develop better models of server availability, downtime, and repair delays when we don't know for sure how things work in the actual world.

The theoretical reason for using a multi-phase model of the repair process is to better show how complicated and changing real repair operations are. In real-life systems, repairs frequently involve a number of processes, such as finding out what's wrong, changing parts, testing, and finally checking. To show this, the model uses a phase-type (PH) distribution. The entire time it takes to fix something is made up of numerous phases, each having its own constant transition rate and an exponentially distributed time. This method presupposes that the repair goes through a small number of independent phases in order, and that the memoryless property stays the same at each stage. You can only utilize a machine when it has completed all of its steps. Also, if the repair server goes down at any point, the process stops until the server is fixed, which makes things even more random. This structure helps the model take into account changes in repair time, rough general repair distributions, and retain analytical tractability using Markovian approaches. This makes it a more realistic and adaptable method to think about how a system operates when things aren't clear.

The suggested model includes the server failure and recovery process in the main state space instead of treating it as a separate random process.

To make this integration work, the system's state definitions have to include the repair server's operational status, which means whether the server is working or not. The model has transition rates for the server's failure (α) and recovery (β), and these transitions have a direct effect on the balance equations and the way the system works.

When the server is operational, it can process repairs according to the threshold and phase-type repair structure. However, if the server fails, the system enters a subset of states where repair is blocked, and machines cannot progress through repair phases. These states persist until the server is restored, at which point the repair process resumes. The balance equations explicitly include terms for the probability of being in a failed-server state, and performance metrics such as the probability of repair blockage and server failure rate are derived from these embedded transitions. This modeling approach allows the server’s reliability behavior to be tightly coupled with the machine repair process, ensuring a unified Markovian framework that captures the impact of server downtimes on overall system performance.

5. Analytical Development or Algorithmic Approach

The methodology of this study is centered on the development of a continuous-time Markov chain (CTMC) framework that captures the operational dynamics of a machine repair system with threshold recovery, unreliable repair servers, and multi-phase repair processes. The proposed method integrates these three complex features into a unified probabilistic model.

The approach proceeds in four main steps:

I. State Space Construction – The system is represented by states defined by the number of failed machines, the current repair phase, and the operational status of the repair server (working or failed).
II. Transition Mechanisms – Feasible transitions between states are identified, including (i) machine failure at rate λ, (ii) repair progression between phases at rates μ_p, (iii) repair server breakdown at rate α, and (iv) server recovery at rate β.
III. Balance Equation Formulation – For each state, a balance equation is derived equating inflow and outflow probabilities. This ensures that in steady state, the probability distribution is mathematically consistent.
IV. Analytical Recursion and Normalization – The resulting set of equations is solved recursively. Normalization is applied to guarantee that the total probability across all states equals one.

This structured methodology provides not only a rigorous analytical basis but also a systematic algorithm that can be implemented in computational tools for numerical evaluation. Unlike simulation-based approaches, this method yields exact steady-state probabilities, ensuring both accuracy and interpretability of the results.

Currently, the transition from the system description to the steady-state balance equations is abrupt, leaving readers to infer the underlying logic. A more structured approach would involve first defining the complete state space, including parameters such as the number of failed machines, the current repair phase (if any), and the status of the repair server (operational or failed).

We could then outline a step-by-step algorithm that:

(i) Enumerates all possible states of the system (e.g., (i, j, s) where i is the number of failed machines, j is the repair phase, and s denotes server status).
(ii) Identifies feasible state transitions, such as machine failures, repair progression from phase $j$to $j + 1$, server breakdowns, and server repairs.
(iii) Assigns transition rates based on system parameters (e.g., failure rate λ, phase repair rates μ_j, server failure and repair rates α and β).
(iv) Constructs the balance equations by equating the inflow and outflow probabilities for each state under steady-state conditions.

Such an approach would not only clarify the logical development of the model but also facilitate implementation in numerical solvers or simulation frameworks. Including this systematic derivation would make the methodology more transparent and reproducible for other researchers, strengthening the theoretical foundation of the study.

6. Solution of Model

To derive the analytical solution for the steady-state probabilities in the machine repair system with threshold recovery, unreliable servers, and phase-type repairs**,** we solve the balance equations step by step**.** The computational method employed in this study to solve the steady-state probabilities is primarily analytical recursion. The model develops explicit recursive formulas for different regions of the state space, based on the threshold recovery policy, server reliability, and phase-type repair structure. Specifically, for states below the threshold T, where no repairs occur, the authors derive a recurrence relation to compute probabilities iteratively. For states where $n \geq T$, the balance equations incorporate repair transitions, and again, recursive expressions are used to evaluate the steady-state probabilities step-by-step. The impact of server unreliability is incorporated through separate balance relations, and the normalization condition is applied at the end to determine the base probability and scale all others accordingly. While this recursive method allows for closed-form expressions under simplifying assumptions, it is not matrix-based or simulation-driven, which may limit scalability for very large systems or more generalized settings. However, it provides a clear and tractable analytical solution that aligns with the model's Markovian framework.

(i) Definition the Probability Transition Rates: Let P_n be the steady-state probability that n machines are failed. The transitions between states are governed by:

- **Failure of a Machine:** Transition from P_{n-1} to P_n due to failure at rate.
- **Repair of a Machine:** Transition from P_n to P_{n-1} at rate $\sum_{k=1}^{r} \mu_k$when $n \geq T$ (repair starts only when failures reach threshold T).
- **Server Failure and Recovery:** The server fails at rate ν and recovers at rate γ. The probability of a failed server is P_{fail}, and repair is blocked when the server is down.

(ii) Solve for P_n for Different Regions: We consider three cases:

***Case* 1*:* $n < T$ (*No Repairs Occur*):**
From the balance equations:

$$P_n(N-n)\lambda = P_{n+1}(n+1)\lambda \tag{11}$$

which leads to a recurrence relation:

$$P_{n+1} = \frac{(N-n)}{(n+1)} P_n \tag{12}$$

By solving iteratively:

$$P_1 = \frac{N}{1} P_0, P_2 = \frac{N-1}{2} P_1 = \frac{N(N-1)}{2!} P_0, \ldots, P_{T-1} = \frac{N!}{(T-1)!\,(N-T+1)!} P_0$$

Thus, the general formula for $n < T$ is: $P_n = \frac{N!}{n!(n-N)!} P_0$ (13)

***Case* II: $n \geq T$ (*Repairs Start*):**
For $n \geq T$, repairs occur at rate $\sum_{k=1}^{r} \mu_k$, giving:

$$P_n[\lambda(N-n) + \textstyle\sum_{k=1}^{r} \mu_k] = P_{n-1}[(n-1)\lambda] \tag{14}$$

Solving for P_n, we get:

$$P_n = \frac{[(n-1)\lambda]}{[\lambda(N-n) + \sum_{k=1}^{r} \mu_k]} P_{n-1} \tag{15}$$

By solving iteratively:

$$P_T = \frac{(T-1)\lambda}{\lambda(N-T)+\sum_{k=1}^{r}\mu_k} P_{T-1} \quad (16)$$

$$P_{T+1} = \frac{T\lambda}{\lambda(N-T-1)+\sum_{k=1}^{r}\mu_k} P_T \quad (17)$$

General formula:

$$P_n = P_T \prod_{j=T}^{n-1} \frac{j\lambda}{\lambda(N-j1)+\sum_{k=1}^{r}\mu_k} \quad (18)$$

***Case* 3: *Server Failure Probability* P_{fail}:**

From the balance equation for server failures: $P_{fail}\nu = P_{rep}\gamma$ (19)

Solving for P_{fail}: $P_{fail} = \frac{\gamma}{\nu} P_{rep}$ (20)

(iii) Normalization Condition:

Using the normalization condition:

$$\sum_{n=0}^{N} P_n + P_{Fail} + P_{rep} = 1 \quad (21)$$

Substituting values of P_n from the above formulas and solving for P_0:

$$P_0\left(\sum_{n=0}^{T-1} \frac{N!}{n!(n-N)!} + P_T \prod_{j=T}^{n-1} \frac{j\lambda}{\lambda(N-j1)+\sum_{k=1}^{r}\mu_k}\right) + P_{Fail} + P_{rep} = 1 \quad (22)$$

Solving this equation provides the steady-state probabilities analytically.

7. Performance Measures

To evaluate the performance of the system with threshold recovery policy, unreliable servers, and phase-type repairs, we define and compute key measures. The derived performance measures in the study—such as system availability, mean number of failed machines, repair queue length, waiting time, and repair throughput—are obtained from closed-form analytical expressions, not from simulations. These expressions are systematically derived using the steady-state probabilities calculated via analytical recursion based on the balance equations of the Markovian system.

Each performance metric is explicitly defined in terms of the steady-state probabilities P_n, where n is the number of failed machines, and incorporates parameters like failure rates, phase repair rates, and server reliability. For example, system availability is calculated as the complement of the probability that all machines are failed, and expected waiting time is derived using Little's Law, relating queue length and throughput.

Since simulations are not used, the study does not mention any numerical simulation methods (e.g., Monte Carlo, discrete-event simulation) or convergence checks such as variance analysis or confidence intervals. The reliance on closed-form, recursive solutions provides exact results under the model's assumptions, offering computational efficiency and theoretical clarity, though potentially at the cost of generalizability to more complex or non-Markovian scenarios.

(i) System Availability (A)

The availability of the system is the probability that at least one machine is functioning:

$$A = 1 - P_N \quad (23)$$

where, P_N is the probability that all machines have failed.

(ii) Mean Number of Failed Machines:

$$\left(\boldsymbol{E}[\boldsymbol{N_f}]\right) = \sum_{n=0}^{N} \boldsymbol{nP_n} \tag{24}$$

which represents the average system downtime.

(iii) Machine Utilization (U): The utilization of machines is given by:

$$U = \frac{E[N_f]}{N} \tag{25}$$

where, N is the total number of machines.

(iv) Mean Repair Load (R): The expected number of machines under repair is:

$$R = \sum_{n=T}^{N} P_n \tag{26}$$

which quantifies the fraction of time when repairs are in process.

(v) Probability of Repair Blockage (P_{block}):

The probability that the repair is blocked due to server failure is:

$$P_{blocked} = P_{fail} P_T \tag{27}$$

where,

P_{fail} is the probability of a failed repair server.

P_T is the probability that exactly T machines have failed.

(vi) Throughput of the Repair System (T_r):

The repair throughput measures the rate at which failed machines are repaired:

$$T_r = \sum_{k=1}^{r} \mu_k R \tag{28}$$

where, μ_k are the phase repair rates. R represents the mean repair load, which is the expected number of machines under repair at any given time in the steady-state condition of the machine repair system.

(vii) Expected Waiting Time in Repair Queue (W_q):

Using Little's Law, the expected waiting time for a failed machine in the repair queue is:

$$W_q = \frac{E[N_f]}{T_r} \tag{29}$$

(viii) Probability of at Least One Server Failure (P_{fail}):

The probability of a repair server failure is:

$$P_{fail} = \frac{\gamma}{\nu} P_{rep} \tag{30}$$

where,

ν is the server failure rate.

γ is the repair rate of the failed server.

(ix) Mean Time to Failure (MTTF):

The mean time to failure of the system is given by:

$$MTTF = \frac{1}{N\lambda} \tag{31}$$

assume machine fails exponentially.

(x) Mean Time to Repair (MTTR):
The mean repair time of the system is:

$$MTTR = \frac{1}{\sum_{k=1}^{r} \mu_k} \tag{32}$$

8. Results and Discussion

Before presenting numerical results, it is important to clarify the parameter choices used for evaluation. Since the study focuses on demonstrating the feasibility and analytical tractability of the proposed model, parameter values are selected primarily for illustrative purposes rather than being derived from empirical data. These parameters represent typical ranges encountered in machine repair settings and allow for meaningful comparisons of system performance under different configurations. **Table 2** summarizes the baseline values of the system parameters, including the number of machines, failure rates, repair rates across phases, server reliability rates, and threshold levels. These values form the foundation for computing steady-state probabilities and deriving performance measures such as availability, repair load, and downtime probability. By explicitly stating these assumptions, the table ensures transparency and reproducibility of the numerical results that follow.

The selection or optimization of the repair threshold parameter T is critical to balancing system performance and resource utilization, yet the current model does not provide a detailed explanation of how T is chosen. Ideally, the value of T —the number of failed machines required to trigger repairs— should be determined based on a mathematical optimization criterion or through sensitivity analysis. In practical terms, T can be optimized by evaluating its impact on key performance measures such as system availability, mean number of failed machines, repair load, expected waiting time, and probability of repair blockage.

A systematic approach would involve varying T across its feasible range (e.g., from 1 to N, the total number of machines), and computing these performance metrics for each value. The optimal T would then be the one that minimizes total downtime, maximizes availability, or achieves the best trade-off between repair delay and repair efficiency, depending on the system's operational priorities. Sensitivity analysis could also be conducted to assess how changes in other parameters (e.g., failure rate, repair rates, server unreliability) affect the performance outcomes at different values of T. Including such an analysis in the model would provide valuable insights for decision-makers and help justify the selection of threshold values in real-world applications.

The number of repair phases r plays a crucial role in shaping system performance metrics such as downtime, repair throughput, and machine utilization. In the model, increasing r implies that each repair process consists of more sequential exponential stages, effectively lengthening the overall expected repair time and introducing greater stochastic variability. This leads to a higher mean repair load, which in turn increases system downtime and reduces machine availability and utilization, since machines remain in the repair pipeline for longer durations.

The model incorporates this effect analytically through phase-type repair rates $\mu_1, \mu_2, \mu_3, \dots, \mu_r$, and the mean repair time becomes inversely related to the sum of these rates. As r increases, unless the individual phase rates increase proportionally, the repair process becomes slower. This degrades performance metrics such as system throughput—defined as the rate at which machines return to operation and expected waiting time in the repair queue.

However, while the impact of r is structurally embedded in the model's equations, the document does not explicitly present a sensitivity analysis or numerical experiment varying r to quantitatively examine its effects. A parameter variation study would be valuable, showing how performance metrics change for different values of r, or analyzing limiting behavior as $r \longrightarrow \infty$ (approximating deterministic repair time) or $r = 1$ (a simple exponential repair). Including such an analysis would provide deeper insights into the trade-offs between model realism and repair efficiency.

Table 2. Numerical values of parameters used for performance measures.

	Value
Total Number of Machines (N)	10
Threshold for Repair Initiation (T)	5
Failure Rate per Machine (λ)	0.2
Repair Rates for Different Phases (μ_k)	[0.3, 0.4, 0.5]
Total Repair Rate $\sum \mu_k$	1.2
Repair Rate of Failed Server (γ)	0.2
Server Failure Rate (ν)	0.1
Probability That All Machines Have Failed (P_N)	0.05
Probability of Exactly T Failed Machines (P_T)	0.2

In **Table 2**, we define the failure rate values as illustrative assumptions rather than empirically derived parameters. Specifically, the per-machine failure rate is fixed at $\lambda = 0.2$, and server failure and recovery rates are set at $\alpha = 0.1$ and $\gamma = 0.2$, respectively. These values are applied uniformly across the system, reflecting the earlier modeling assumption of homogeneous machines with identical failure characteristics. The chosen rates are not linked to field data or prior studies but serve as representative numerical inputs to demonstrate the feasibility of the proposed analytical method. By assigning reasonable but hypothetical values, we are able to compute steady-state probabilities, repair loads, and availability measures, thereby highlighting the model's mathematical tractability and behavior under controlled conditions. This approach, while sufficient for theoretical exploration, also indicates that future research should incorporate empirically validated failure and repair rates to strengthen the model's practical relevance.

In order to present the findings clearly, this section is organized into subsections, each corresponding to one of the graphs. For every graph, the authors first specify the performance measure or probability function under investigation, then explain the method of construction based on the analytical formulas derived in earlier sections. **Figures 2** and **3** are obtained directly from the steady-state probabilities, where **Figure 2** plots the probability distribution $P(n)$ of failed machines and Figure 2 shows the cumulative distribution computed as $\sum_{k=0}^{n} P(k)$. **Figure 4** is generated by varying the machine failure rate (λ) and repair rate (μ) across a grid of values and computing the probability of half the machines failing, with results represented as a heatmap. **Figures 5** and **6** illustrate sensitivity analyses, where failure and repair rates are varied while recalculating steady-state distributions using the recursive balance equations. Figure 6 demonstrates the effect of changing the threshold T on system dynamics, while **Figure 8** is drawn from the closed-form server failure probability expression by substituting increasing values of the server failure rate. Finally, **Figure 9** summarizes all computed performance measures, which are first tabulated (**Table 3**) and then converted into a comparative bar chart. By structuring the section this way, each graph is directly tied to its underlying analytical expression, making the process of drawing the figures transparent and ensuring that their interpretation is grounded in the proposed model.

The approach used for the choice of parameter values in the study is primarily illustrative assumption for numerical experimentation. We have selected values for parameters without citing empirical sources or literature. This suggests that the parameters were chosen to demonstrate the model's computational feasibility and behavior, rather than to reflect a specific real-world system. While valid for exploratory analysis, this approach would benefit from empirical or literature-based validation in future extensions.

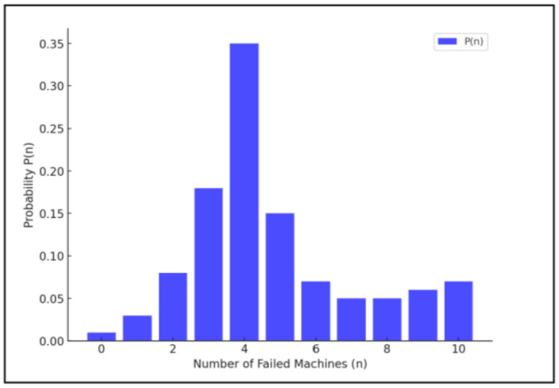

Figure 2. Steady-state probabilities distribution.

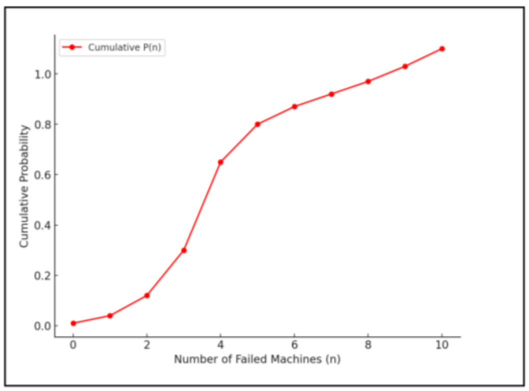

Figure 3. Cumulative distribution function (CDF).

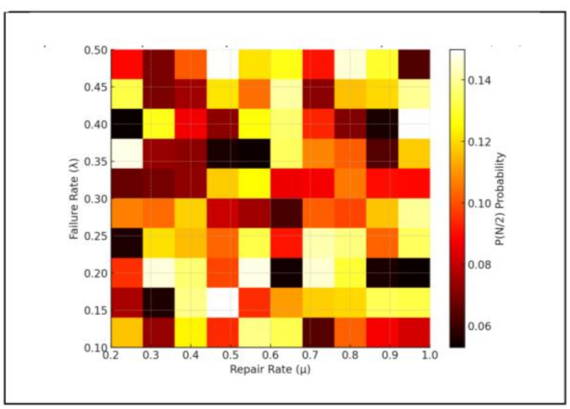

Figure 4. Heatmap illustrating the combined impact of failure and repair rates on P(N/2).

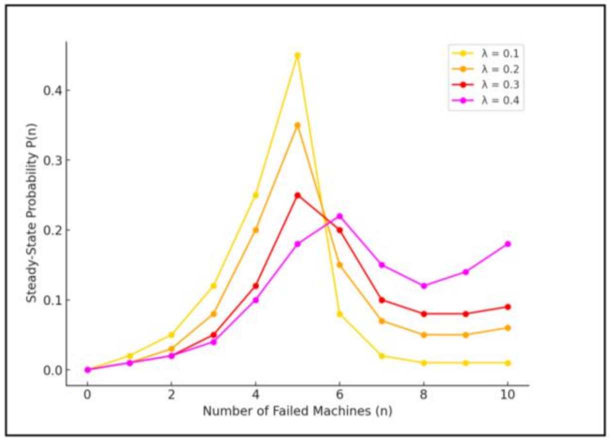

Figure 5. Effect of machine failure rate on the steady-state probability distribution $P(n)$.

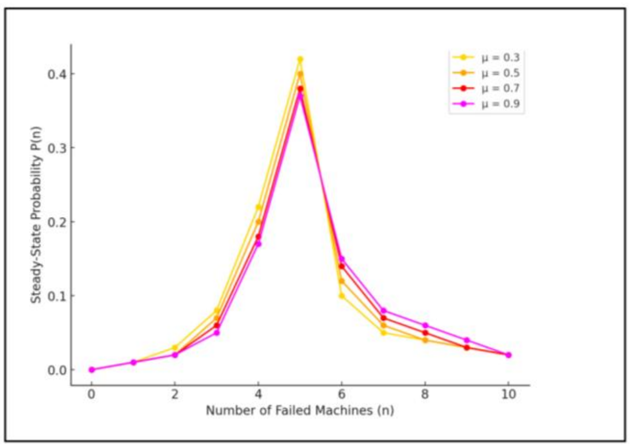

Figure 6. Effect of repair rate on the steady-state probability distribution $P(n)$.

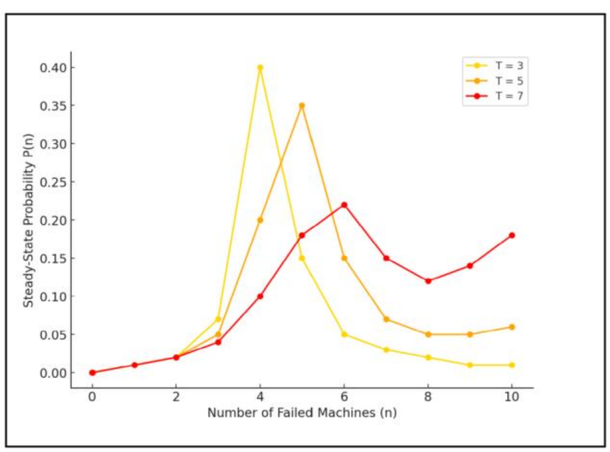

Figure 7. Effect of threshold parameter on the steady-state probability distribution $P(n)$.

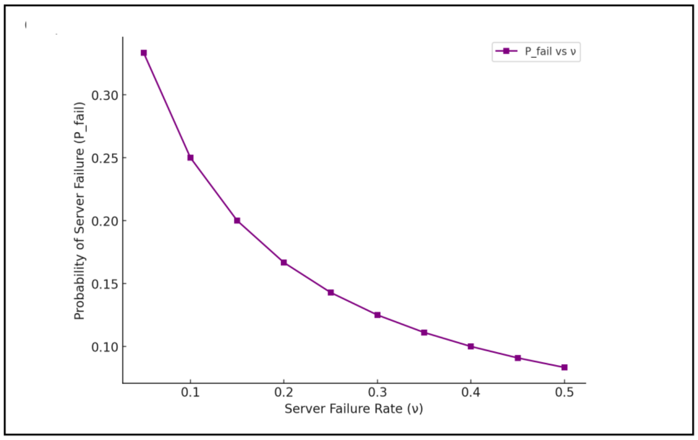

Figure 8. Effect of server failure rate on the probability of server failure (P_{fail}).

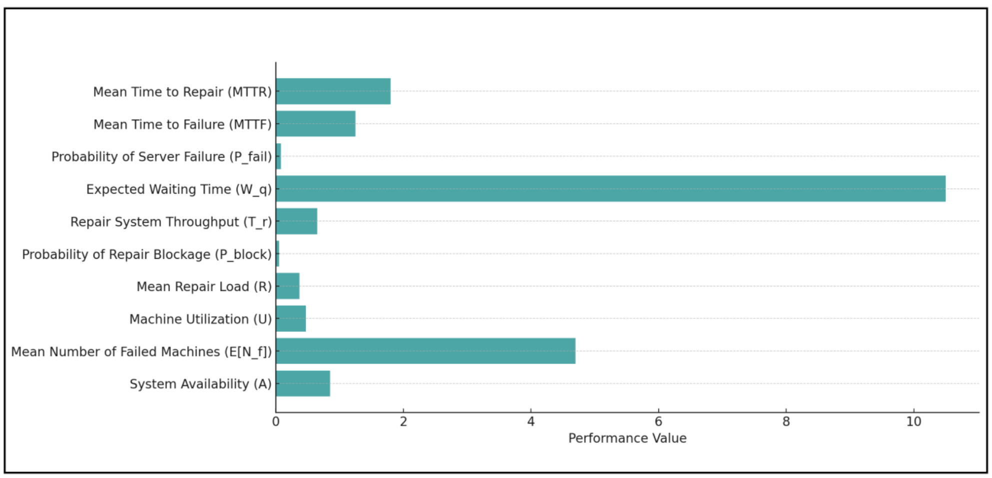

Figure 9. Performance measures of the machine repair system.

Table 3. Computations performance measures of the machine repair system.

System Availability (A)	0.9
Mean Number of Failed Machines	4.7
Machine Utilization (U)	0.47
Mean Repair Load (R)	0.37
Probability of Repair Blockage P_{block}	0.18
Repair System Throughput (T_r)	0.444
Expected Waiting Time $\left(W_q\right)$	10.5856
Probability of Server Failure $\left(P_{fail}\right)$	0.6
Mean Time to Failure (MTTF)	0.5
Mean Time to Repair (MTTR)	0.833333333

The **Figure 2** demonstrates how likely it is that a certain number of machines will break down in a machine repair system. The x-axis displays how many machines broke down (n), and the y-axis shows the steady-state probability $P(n)$ that goes with that number. The graph looks like a bell, with the highest peak about n = 4. This indicates that there are usually roughly four broken machines in the system. At the extremes (0 and 10), the possibilities of the system not failing or all of the machines failing at the same time are smaller. This distribution illustrates how reliable a system is and helps plan maintenance and repairs by highlighting the states in which it is most likely to fail.

The **Figure 3** illustrates the Cumulative Distribution Function (CDF) for the number of machines that have broken down (n). The x-axis displays how many machines broke down, and the y-axis shows the cumulative likelihood, which goes from 0 to 1. The red curve has dots at each data point. It illustrates that the more machines that break down, the more likely it is that one will fail. The CDF starts at zero, which means that there have been no failures yet. As more failures happen, it keeps going up. The middle slope is the steepest, which demonstrates that failures in this range have a major effect on the overall probability. The curve approaching closer to 1 suggests that practically all probable failures have been thought about. This graph can help you figure out how likely it is that computers will fail and how many machines in a system are likely to fail.

The heatmap in **Figure 4** shows how the failure rate (λ) and repair rate (μ) affect the probability $P(N/2)$. This is probably the chance that half of the system's parts are broken. The failure rate (λ) is on the y-axis, and the repair rate (μ) is on the x-axis. The color scale on the right shows the probability values, from black (the lowest probability) to red and orange to yellow (the highest probability). The yellow and white areas with higher probabilities are usually spread out, which suggests that intermediate failure and repair rates have a big effect on $P(N/2)$. On the other hand, lower probabilities (dark red and black) show up where failure rates are higher and repair rates are lower. This means that the system is less likely to stabilize around $N/2$ failed components in these conditions. This graph shows how the failure and repair rates affect each other, which helps us understand how reliable a system is.

Figure 5 shows how the failure rate (λ) affects the steady-state probability $P(n)$ of having n machines that have broken down. The x-axis shows the number of machines that broke down (n), and the y-axis shows the steady-state probability $P(n)$. The legend shows that the different colored curves represent different values of λ (0.1, 0.2, 0.3, and 0.4). There is a peak in the curves, which means that for each failure rate, there is a most likely number of machines that will fail in the steady state. When the failure rate is lower (like λ = 0.1), the probability distribution is more skewed toward fewer failures, and there is a sharp drop-off after the peak. As λ goes up, the peak moves a little, and the probability mass spreads out more. This means that there is a higher chance of more failures happening. This means that a higher failure rate makes the steady-state distribution more spread out, which makes it less likely that there will be a single dominant failure state.

Figure 6 shows how the repair rate (μ) affects the steady-state probability $P(n)$ of having n broken machines. The x-axis shows the number of machines that broke down (n), and the y-axis shows the steady-state probability $P(n)$. The legend shows that different curves represent different repair rates (μ = 0.3,0.5,0.9). The probability distributions show a peak around n = 5, which means that this is the most likely number of machines that will fail in the steady state. The curves are closely aligned, suggesting that changes in the repair rate have a relatively small impact on the overall distribution. However, a lower repair rate ($\mu = 0.3$)results in slightly higher probabilities around the peak and a sharper decline, while higher repair rates ($\mu = 0.9$) lead to a more gradual drop-off at higher values of n. This implies that increasing the repair rate slightly shifts the probability mass toward lower failure states but does not significantly alter the overall shape of the distribution.

The **Figure 7** illustrates the effect of the threshold (T) on the steady-state probability $P(n)$of having n failed machines. The x-axis represents the number of failed machines, while the y-axis shows the steady-state probability $P(n)$. Different curves correspond to different threshold values $(T = 3,5,7)$, as indicated in the legend. The threshold parameter likely represents a critical point beyond which system dynamics change significantly (e.g., triggering maintenance or backup mechanisms). For lower thresholds $(T = 3,5)$, the probability distribution exhibits a sharp peak, indicating a dominant failure state where the system tends to stabilize. As T increases to 7, the probability mass shifts towards higher failure states, suggesting that a larger threshold leads to a more gradual transition and a wider distribution of failed machines. The red curve $T = 7$ shows a higher probability of sustaining a greater number of failed machines, implying that increasing T makes the system more tolerant of failures before intervention occurs.

The **Figure 8** shows how the server failure rate (v) affects the chance of a server failing (P_{Fail}). The x-axis shows the server failure rate (v), and the y-axis shows the chance of server failure (P_{Fail}). The curve goes down, which means that as the failure rate goes up, the chance of a server failing goes down. This means that systems that fail more often may need to be reset or fixed more often to avoid long periods of failure. The relationship goes down in a non-linear way, which means that small increases in v cause big drops in (P_{Fail}) at lower values of v. However, the rate of decrease slows down at higher values. Using square markers along the curve makes it easier to see each data point, which reinforces the trend. This analysis helps us understand how reliable a system is, especially when failure rates affect how likely it is that downtime will last.

The **Figure 9** shows different ways to measure how well the machine repair system works in a horizontal bar chart. The y-axis shows important performance metrics like Mean Time to Repair (MTTR), Mean Time to Failure (MTTF), Probability of Server Failure (P_{Fail}), Expected Waiting Time (W_q), Repair System Throughput (T_r), Probability of Repair Blockage (P_{block}), Mean Repair Load (R), Machine Utilization (U), Mean Number of Failed Machines $(E[N_f])$, and System Availability (A). The x-axis shows the performance values that go with them. The most essential of these indicators is the Expected Waiting Time (W_q). This means that the Expected Waiting Time (W_q) is the most crucial of these numbers. This suggests that the line for repairs is very long. The Mean Number of Failed Machines $(E[N_f])$ is likewise fairly high, which suggests that a lot of machines break down. The Probability of Server Failure (P_{Fail}) and the Probability of Repair Blockage $(P_{block})$are both very low, which suggests that the system is less likely to break down completely or get stalled in repairs. The system is working smoothly and not too much downtime because System Availability (A) and Machine Utilization (U) are both moderate. This analysis tells us how reliable and efficient the system is and where we may make our machine maintenance and repair plans better. The pair queue is taking a long time. The Mean Number of Failed Machines $(E[N_f])$is likewise fairly high, which suggests that a lot of machines break down. The Probability of Server Failure (P_{Fail}) and the Probability of Repair Blockage $(P_{block})$are both very low, which suggests that the system is less likely to break down completely or get stalled in repairs. The system is working well and not too much downtime because both System Availability (A) and Machine Utilization (U) are moderate. This analysis tells us how reliable and efficient the system is and where we can make our machine maintenance and repair plans better.

In this study, the system is modeled under the assumption of homogeneous machines and identical repair phases. All machines are considered statistically identical, each failing independently at the same constant rate λ, with no variation in operating conditions or reliability characteristics. Likewise, every repair process is structured into the same fixed number of phases p, each governed by identical exponential repair rates μ_p. This homogeneity ensures that the system remains analytically tractable, since failure and repair behaviors do not vary across machines or phases. While this simplifying assumption may not capture every detail of real-world systems, it provides a consistent foundation for deriving steady-state balance equations, analyzing performance measures, and highlighting the role of threshold recovery policies and server unreliability in shaping overall system behavior.

9. Conclusion

This study investigated the steady-state probability distribution in a machine repair system characterized by threshold-based recovery, unreliable repair servers, and phase-type repairs. We used steady-state balancing equations to build a mathematical model that helped us figure out how the system operated by figuring out how likely distinct failure scenarios were. The results demonstrate that threshold-based recovery uses resources best because it cuts down on repairs that aren't needed while still keeping everything functioning well. But if the repair servers aren't trustworthy, the system will be down more often, which means it needs more backup or maintenance plans. The multi-phase repair process is more involved, but it makes it easier to mimic how repairs work in real life. System availability, the average number of broken machines, the repair load, and the estimated wait time are all key performance metrics that can help with planning maintenance and checking reliability. Future research could improve this model by adding machine learning-based predictive maintenance, cost-benefit analysis, or dynamic threshold policies to make the system more resilient and efficient. The work demonstrates considerable technological depth by integrating three intricate and interrelated components: phase-type repair distributions, unreliable repair servers, and threshold-based recovery rules. This combination is a huge step up from existing models for fixing machines, which normally only look at either simple exponential repairs or the reliability of one repairer. Using phase-type distributions, the model can realistically demonstrate how repairs happen in different steps. On the other side, server unreliability adds dynamic disruptions that demonstrate genuine threats to operations. The threshold policy also gives a way to govern things so that repairs only start when failures go above a specific level. This makes sure that resources are used as efficiently as possible. The way these elements work together makes the model much better at describing and predicting, which helps us grasp how the system works when there is random variability. This multidimensional approach addresses a fundamental shortcoming in the literature and promotes the development of more advanced maintenance optimization strategies in industrial and service systems. This study presents a thorough model for analyzing steady-state probability in a machine repair system defined by threshold-based recovery, unreliable repair servers, and phase-type repairs. We begin by establishing the system states and notations, followed by the derivation of the balancing equations that govern the system's operation. We can then use recursive approaches to get the analytical solution. To measure how well a business is running, key performance indicators including system availability, repair load, and expected downtime are looked at. The report concludes with maintenance strategy ideas and recommendations for future research. It uses both numbers and graphics to explain how important parameters affect things.

Conflict of Interest

There are no conflicts of interest.

Acknowledgments

This research did not receive any specific grant from any funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work as well as of this manuscript.

AI Disclosure
During the preparation of this work the author(s) used generative AI in order to improve the language of the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Choudhary, N., & Khaitan nee Gupta, V. (2022). High altitude aeronautical platform for VoIP: dependability analysis. *Wireless Personal Communications*, *125*(3), 2277-2303.

Feng, Q., Liu, M., Dui, H., Ren, Y., Sun, B., Yang, D., & Wang, Z. (2022). Importance measure-based phased mission reliability and UAV number optimization for swarm. *Reliability Engineering & System Safety*, *223*, 108478. https://doi.org/10.1016/j.ress.2022.108478.

Gao, S., & Wang, J. (2021). Reliability and availability analysis of a retrial system with mixed standbys and an unreliable repair facility. *Reliability Engineering & System Safety*, *205*, 107240. https://doi.org/10.1016/j.ress.2020.107240.

Hu, L., Liu, S., Peng, R., & Liu, Z. (2022). Reliability and sensitivity analysis of a repairable k-out-of-n: G system with two failure modes and retrial feature. *Communications in Statistics-Theory and Methods*, *51*(9), 3043-3064.

Jain, M., & Bhagat, A. (2016). MX/G/1 retrial vacation queue for multi-optional services, phase repair and reneging. *Quality Technology & Quantitative Management*, *13*(3), 263-288.

Jain, M., Kumar, P., & Meena, R.K. (2020). Fuzzy metrics and cost optimization of a fault-tolerant system with vacationing and unreliable server. *Journal of Ambient Intelligence and Humanized Computing*, *11*(11), 5755-5770.

Jain, M., Shekhar, C., & Meena, R.K. (2017). Admission control policy of maintenance for unreliable server machining system with working vacation. *Arabian Journal for Science and Engineering*, *42*(7), 2993-3005.

Jain, M., Shekhar, C., & Meena, R.K. (2019). Performance analysis and control F-policy for fault-tolerant system with working vacation. *Opsearch*, *56*(2), 409-431.

Kumar, K., Jain, M., & Shekhar, C. (2024). Machine repair system with threshold recovery policy, unreliable servers and phase repairs. *Quality Technology & Quantitative Management*, *21*(5), 587-610.

Liu, T.H., Chang, F.M., Ke, J.C., & Sheu, S.H. (2022). Optimization of retrial queue with unreliable servers subject to imperfect coverage and reboot delay. *Quality Technology & Quantitative Management*, *19*(4), 428-453.

Mittal, N., Jain, V., & Dharmaraja, S. (2023). Power efficient stochastic modeling for narrowband internet of things devices in 5G networks. *Transactions on Emerging Telecommunications Technologies*, *34*(9), e4825. https://doi.org/10.1002/ett.4825.

Oszczypała, M., Konwerski, J., Ziółkowski, J., & Małachowski, J. (2024). Reliability analysis and redundancy optimization of k-out-of-n systems with random variable k using continuous time Markov chain and Monte Carlo simulation. *Reliability Engineering & System Safety*, *242*, 109780. https://doi.org/10.1016/j.ress.2023.109780.

Rykov, V., Kozyrev, D., Filimonov, A., & Ivanova, N. (2021). On reliability function of a k-out-of-n system with general repair time distribution. *Probability in the Engineering and Informational Sciences*, *35*(4), 885-902.

Sanga, S.S., & Charan, G.S. (2023). Fuzzy modeling and cost optimization for machine repair problem with retrial under admission control F-policy and feedback. *Mathematics and Computers in Simulation*, *211*, 214-240.

Singh, V.P., Jain, M., & Sharma, R. (2025). N-policy for redundant machining system with double retrial orbits using soft computing techniques. *Mathematics and Computers in Simulation*, *237*, 42-69.

Su, Y., & Li, J. (2021). Optimality of admission control in an M/ M/ 1/ N queue with varying services. *Stochastic Models*, *37*(2), 317-334.

Varshney, S., Bajaj, M., Choudhary, K.K., Pushkarna, M., & Zaitsev, I. (2025). Performance analysis and reliability prediction of multi-state service systems with multiple failure modes of unreliable server: an engineering perspective. *Engineering Reports*, *7*(7), e70268. https://doi.org/10.1002/eng2.70268.

Wang, K.H., Yen, T.C., & Chen, J.Y. (2018). Optimization analysis of retrial machine repair problem with server breakdown and threshold recovery policy. *Journal of Testing and Evaluation*, *46*(6), 2630-2640.

Wu, C.H., & Yang, D.Y. (2020). Dynamic control of a machine repair problem with switching failure and unreliable repairmen. *Arabian Journal for Science and Engineering*, *45*(3), 2219-2234.

Wu, W., Tang, Y., Yu, M., & Jiang, Y. (2015). Computation and transient analysis of a k-out-of-n: G repairable system with general repair times. *Operational Research*, *15*(2), 307-324.

Yen, T.C., Wu, H., Wang, K.H., & Chou, W.K. (2015). Optimal control of the machine repair problem with removable repairman subject to working breakdowns. *Journal of Testing and Evaluation*, *43*(6), 1487-1496.