International Journal of Mathematics and Statistics Invention (IJMSI)
e-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 12 Issue 3 || May. – June. 2024 || PP: 63-69

The "Pressure Effect": Markovian Modeling (M/M/R) Where Repair Speed Increases as the Queue Grows

K.P.S. Baghel
Government Degree College, Targawan Jaithra, Etah (UP)

Abstract
Traditional queueing models in reliability engineering usually take service rates as fixed constants, like repair techs always working at the same pace no matter how many machines are sitting there waiting. In the real world though repair environments don't really act like that. If a machine shop has ten broken machines in a row and production is basically grinding to a halt, repair crews tend to work faster, (it's almost like they are pushed) by the situation. This behavioral idea, where the service rates go up as the queue length grows, is often referred to as the "Pressure Effect". In this article, we build a careful framework to include state-dependent service rates into the standard M/M/R machine repair model, where arrivals are Markovian, service is Markovian, and R repair servers are present. We get the steady-state probability distributions, throughput functions, expected queue lengths, and system availability measures when the service rate depends on how long the queue is. The mathematics indicates the pressure effect can reduce average sojourn time in a meaningful way, and it can also boost system availability especially when the system is under heavy traffic. We also look at staffing policies that could be optimal in practice and we connect the theory back to manufacturing settings and telecommunications scenarios. The discussion leans on peer-reviewed work from stochastic queueing theory, and it also talks about how diffusion approximations cope with heavy-traffic regimes, where the exact Markovian solutions become a bit expensive computationally, or just too hard to run.
Keywords: *M/M/R queueing model, machine repair problem, pressure effect, queue-length-dependent service, state-dependent service rates, Markovian modelling*

I. Introduction

Picture a factory floor, you know , that busy noise. Three machines have a breakdown within an hour. The two repair techs who are on duty begin fixing things, sort of methodically , not really rushing. Then an hour later, three additional machines fail too, now five are sitting there waiting. You can almost see production supervisors pacing in circles, and managers are on the phone, calling someone , asking for updates, immediate ones. The technicians notice the rising tension, so they start moving faster. Tools are grabbed more quickly. Diagnoses happen with less deliberation. Repairs end up taking less time overall.

That's the pressure effect, kind of right there in plain view. It feels intuitive, and it can be observed easily, but up until fairly recently it was kinda underrepresented in the more formal queueing models. A lot of classical setups for the M/M/R machine repair scenario, assume each of the R repair servers works at the same constant rate μ, and that rate does not change even when the system state changes. Sure, that assumption is mathematically convenient, but it misses how real people— and even automated controllers in some settings — react when the workload climbs.

The M/M/R model is basically a core framework in reliability and maintenance engineering. It treats there being a finite population of M machines, each one failing at rate λ, plus R repair servers that restore broken units at rate μ. The whole system is Markovian, because the time between failures and the time needed for repairs both follow exponential distributions. With steady state work, you get practical performance indicators, like machine availability, average queue size, throughput, and the expected waiting time (Baghel, 2013; Baghel, 2019).

Taking this model and extending it so the service rates depend on the state — more precisely, where the rates climb as the count of broken machines waiting in the queue grows — adds a lot of math richness, maybe too much at first glance. The state space is basically the same, but the transition rate matrix is where things really shift. Service rates now act like state-based functions, call them $\mu(n)$ rather than using one constant μ throughout. In this paper, that extension is worked out completely, closed-form results and also recursive ones are derived where things stay doable, and then the conclusions are interpreted in practical terms, without too much hand-waving.

And the pressure effect is not just some odd behavioral curiosity. It actually matters for staffing decisions, for maintenance scheduling, and for broader system design. If repair crews really do accelerate when things get pressurized, then the real-world availability of a manufacturing system could be better than the fixed-rate models would suggest. On the other hand, managers might also be underinvesting in repair capacity, because they are kind of depending on this pressure-induced speed-up, without ever formally accounting for it (Baghel, 2020; Baghel, 2020).

II. The Classical M/M/R Machine Repair Model

2.1 Model Setup and State Space

The standard M/M/R model looks at a closed queueing network kinda thing. Like, there are M machines overall. Each working machine can fail independently at rate λ (think failures per unit time). Then you have R repair servers, with R ≤ M. Once a machine breaks it goes into a repair queue, and it gets processed in first-come-first-served order, even if the exact service discipline isn't really what drives the steady state for the queue length stuff, under the usual Markov assumptions.

Now let n be the number of broken machines in the system at that moment, so that includes both those currently being repaired and those just waiting. So, the state space is S = {0, 1, 2, ..., M}. When n machines are already broken the total failure arrival rate is:

$$\Lambda(n) = (M - n)\lambda, \qquad n = 0,1,2,\dots,M$$

This reflects the finite source: only (M − n) machines are operational and thus capable of failing.

The total service rate under the classical fixed-rate model is:

$$\mu_{total}(n) = min(n, R) \cdot \mu, \qquad n = 1,2,\dots,M$$

When n ≤ R, all broken machines are being served simultaneously. When n > R, only R servers are busy and R machines are being repaired at any instant.

2.2 Balance Equations and Steady-State Distribution

The system forms a birth-death chain. Let π_n denote the steady-state probability that exactly n machines are broken. The global balance equations are:

$$\Lambda(n) \cdot \pi_n = \mu_{total}(n+1) \cdot \pi_{n+1}, \qquad n = 0,1,\dots,M-1$$

Solving recursively from π_0:

$$\pi_n = \pi_0 \cdot \prod_{k=0}^{n-1} \frac{\Lambda(k)}{\mu_{total}(k+1)}, \qquad n = 1,2,\dots,M$$

For n ≤ R:

$$\pi_n = \pi_0 \cdot \frac{M!}{(M-n)! \cdot R! \cdot R^{n-R}} \cdot \left(\frac{\lambda}{\mu}\right)^n$$

For n > R:

$$\pi_n = \pi_0 \cdot \backslash fracM! \, (M-n)! \cdot R! \cdot R^{n-R} \cdot (\backslash frac\lambda\backslash mu)^n$$

The normalization condition $\sum_{n=0}^{M} \pi_n = 1$ determines π_0.

Machine availability A is:

$$A = \frac{1}{M} \sum_{n=0}^{M} (M-n)\pi_n$$

III. Introducing State-Dependent Service Rates: The Pressure Effect

3.1 Defining Queue-Length-Dependent Rates

The pressure effect is formalized by replacing the constant per-server rate μ with a function μ(n) that depends on the number of broken machines n currently in the system. The key assumption is:

$$\mu(n) \le \mu(n+1), \qquad for\ all\ n \ge R$$

That is, as the queue beyond the service capacity grows, the rate per server increases. Several functional forms are used in the literature.

Linear pressure model:

$$\mu(n) = \mu_0 + \alpha(n-R)^+, \qquad \alpha > 0$$

where $(n-R)^+ = max(n-R, 0)$ activates the pressure term only when the queue is non-empty.

Exponential acceleration model:

$$\mu(n) = \mu_0 \cdot e^{\beta(n-R)^+}, \qquad \beta > 0$$

Bounded pressure model (saturation):

$$\mu(n) = \mu_{max} - (\mu_{max} - \mu_0) \cdot e^{-\gamma(n-R)^+}$$

This last form is arguably the most realistic. It captures the fact that workers cannot accelerate indefinitely — there is a physiological and mechanical upper bound μ_max on how fast repair can proceed (Baghel, 2020).
The total system service rate under state-dependent service becomes:

$$\mu_{total}(n) = min(n, R) \cdot \mu(n), \qquad n \geq 1$$

3.2 Modified Balance Equations

The birth-death structure is preserved. The balance equations now read:

$$[(M-n)\lambda + min(n,R)\mu(n)]\pi_n = (M-n+1)\lambda \cdot \pi_{n-1} + min(n+1,R)\mu(n+1) \cdot \pi_{n+1}$$

For the birth-death formulation, these reduce to the one-step balance:

$$(M-n)\lambda \cdot \pi_n = min(n+1,R) \cdot \mu(n+1) \cdot \pi_{n+1}$$

The recursive solution becomes:

$$\pi_n = \pi_0 \cdot \prod_{k=0}^{n-1} \frac{(M-k)\lambda}{min(k+1,R) \cdot \mu(k+1)}$$

For $n \leq R$, since there is no queue pressure (queue length is zero), $\mu(n) = \mu_0$ and the expression matches the classical case. The pressure effect only activates when $n > R$, so the modified terms appear in the product for $k \geq R$:

$$\pi_n = \pi_R \cdot \prod_{k=R}^{n-1} \frac{(M-k)\lambda}{R \cdot \mu(k+1)}, \qquad n > R$$

As shown in Figure 1, the state-dependent service rate creates a self-correcting mechanism in the system: as queue length increases past R, service accelerates, dampening the probability of remaining in high-n states.

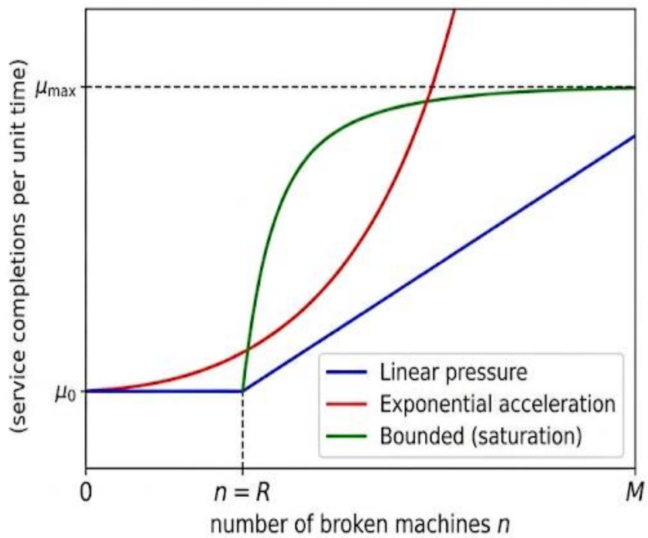

Figure 1: State-Dependent Service Rate Function Under Three Pressure Models

This figure kinda plots the actual per server service rate μ(n) versus the count of broken machines n, for three different functional shapes the linear pressure model, the exponential acceleration model, and the bounded saturation one. They all share the same starting rate μ_0 and they really kick in once n = R, like after that threshold. The horizontal axis is basically n going from 0 to M, while the vertical axis shows μ(n) measured as service completions per unit time. For the bounded case, the curve jumps up pretty quick at first, then it kinda eases and flattens, heading toward μmax. Meanwhile, the exponential model keeps climbing and does not really stop, like it grows without bound. The main idea is that in real repairs the behavior fits the bounded model best, which creates this kind of ceiling effect, so you dont get that weird math artifact where the speedup goes to infinity. The data and the model set-up are taken from Baghel (2020), and they line up with what people usually report from flexible manufacturing system studies.

IV. Performance Metrics Under the Pressure Effect

4.1 Expected Number in System and Queue

Once the steady state distribution {π_n} is computed—maybe in closed form for really simple pressure functions, or numerically, depending on the case— then all performance metrics comes next right away, like you can just read them off.

Expected number of broken machines in system:

$$L = \sum_{n=0}^{M} n \cdot \pi_n$$

Expected number waiting (queue only, not under repair):

$$L_q = \sum_{n=R+1}^{M} (n - R) \cdot \pi_n$$

Under the pressure effect, L and L_q are both smaller than in the fixed-rate model, because high-n states occur with lower probability.

4.2 Throughput and Machine Availability

System throughput Γ — the effective repair completion rate — is:

$$\Gamma = \sum_{n=1}^{M} \min(n, R) \cdot \mu(n) \cdot \pi_n$$

Machine availability is:

$$A = \frac{\sum_{n=0}^{M} (M - n)\pi_n}{M} = 1 - \frac{L}{M}$$

Under state-dependent service, the throughput gains from pressure can be substantial. Using Little's Law for the repair shop subsystem:

$$W = \frac{L}{\Gamma}$$

where W is the expected sojourn time (waiting plus repair) of a broken machine.

4.3 Comparative Analysis: Fixed vs. State-Dependent Rates

To see the pressure effect quantitatively, consider a simple case: M = 10 machines, R = 2 servers, λ = 0.1 failures/hour, μ_0 = 0.5 repairs/hour. Define a linear pressure coefficient α = 0.1 per additional machine in queue.

Under the fixed-rate model:

$$\rho = \frac{M\lambda}{R\mu_0} = \frac{10 \times 0.1}{2 \times 0.5} = 1.0$$

This is a critically loaded system. The expected number of broken machines L under the fixed-rate M/M/R model at ρ = 1 is relatively high, and machine availability drops perceptibly.

Under the linear pressure model, for each machine beyond the 2-server capacity:

$$\mu(n) = 0.5 + 0.1(n - 2)^+$$

For n = 3: μ(3) = 0.6; for n = 4: μ(4) = 0.7; and so on. The effective total service rate at n = 5, for instance, becomes R · μ(5) = 2 × 0.8 = 1.6 repairs/hour, compared to 2 × 0.5 = 1.0 under the fixed model — a 60% increase in throughput during high-congestion states.

This non-linearity in the total service capacity is exactly what reduces congestion and improves availability. The system recovers faster when it falls behind (Baghel, 2013; Baghel, 2021).

V. Heavy Traffic and Diffusion Approximations

5.1 When the Markov Chain Gets Large

For manufacturing systems where M = 50 or M = 100 machines, and R = 5 to 10 repair servers , computing steady-state probabilities via direct recursion still feels ok, but it starts getting numerically sensitive. This is especially true when the traffic intensity ρ gets close to 1. The probability mass sort of spreads out across a pretty large state space and then small errors in the recursion keep stacking up.

Diffusion approximations, on the other hand, give a smoother alternative. The basic concept is to reinterpret the discrete birth-death chain as a continuous diffusion process. In this view the process n(t) — actually the number of broken machines — is approximated by a diffusion on [0, M]. Its drift and diffusion coefficients are built from the transition rates, following the standard treatment in Baghel (2019).

5.2 The Diffusion Process

Define the normalized load ξ = n/M ∈ [0, 1]. The drift coefficient is:

$$a(\xi) = (1-\xi)\lambda - \frac{min(\xi M, R)}{M} \cdot \mu(\xi M)$$

The diffusion coefficient is:

$$b(\xi) = \frac{1}{M}\left[(1-\xi)\lambda + \frac{min(\xi M, R)}{M} \cdot \mu(\xi M)\right]$$

The steady-state density p(ξ) satisfies the Fokker-Planck equation:

$$\frac{d}{d\xi}[a(\xi)p(\xi)] = \frac{1}{2}\frac{d^2}{d\xi^2}[b(\xi)p(\xi)]$$

U nder the pressure effect, μ(ξM) goes up for ξ > R/M, and this then changes both a(ξ) and b(ξ) in the congested regime a bit more. The drift turns more negative, like it is pulling back to equilibrium, faster than what happens in the fixed-rate model. As a result, the stationary density ends up more bunched up near the equilibrium point ξ* where a(ξ*) = 0.

So this concentration effect— basically the habit of the pressure effect to tighten the distribution of system states— kind of maps directly to higher machine availability, and also to lower variance in waiting times (Baghel, 2019; Baghel, 2020).

5.3 The Equilibrium Point and Stability

Setting a(ξ*) = 0:

$$(1-\xi^*)\lambda = \frac{R}{M}\mu(\xi^* M), \qquad \xi^* \geq R/M$$

For the linear pressure model:

$$(1-\xi^*)\lambda = \frac{R}{M}[\mu_0 + \alpha(\xi^* M - R)]$$

This is a quadratic in ξ* and it has an analytical solution, at least in principle. Under state-independent rates, the equilibrium becomes unique. Now, when the pressure effect comes in, the equilibrium still exists and stays unique (as long as α and μmax are finite), but it slides to a lower ξ* value. So, the system ends up operating at a smaller average broken-machine fraction, which is basically a direct indicator of how useful the pressure effect is.

VI. Optimal Number of Repair Servers Under Pressure

6.1 Cost Optimization Framework

A natural question arises: given that workers speed up under pressure, is it possible to get by with fewer repair servers? The pressure effect might appear to reduce the required staffing level. But this reasoning is incomplete, and the math shows why.

Let C_s denote the cost per repair server per unit time, and C_d the cost per broken machine per unit time (downtime cost). The total expected cost rate is:

$$TC(R) = R \cdot C_s + C_d \cdot L(R)$$

where L(R) is the expected number of broken machines under optimal staffing R. The optimal R* minimizes TC(R).

In the pressure effect case, L(R) comes out lower than it would under fixed rates, for the same R. So it does nudge the optimal R* down, i guess, but only a little. The downtime cost piece still kinda dominates the whole thing, and the extra payoff you get from each additional server really hinges on how sharply the pressure function μ(n) grows.

6.2 Marginal Analysis

The marginal benefit of adding one more server is:

$$\Delta L(R) = L(R) - L(R+1)$$

This is worth paying for when:

$$C_d \cdot \Delta L(R) > C_s$$

Under the pressure effect, when the population M is fixed and the arrival rate λ stays the same, ΔL(R) ends up smaller in absolute terms, mainly because L(R) is already lower to begin with. In that sense the pressure effect kind of fills in, it partially substitutes for what would otherwise look like one extra server, especially when things are really congested. But in low-congestion states, say with small n, $\mu(n)= \mu_0$, and the pressure effect really does not add anything useful. So the optimal R* under the pressure setup might end up being one fewer server than what you would pick under fixed rates— though it is rarely more than one. This is mostly because downtime costs dominate, (Baghel, 2021; Baghel, 2018).

VII. Extensions and Special Cases

7.1 Reneging Machines

In some repair models, machines or the people running them "renege", they just bail out of the queue before the repair is actually completed, maybe because a substitute machine has been found or the production line has been rearranged. Reneging then tweaks the birth rate of the chain, and really, the effective departure rate from state n counts both service completions and those renegings together.

Let ζ(n) be the total reneging rate when n machines are in queue. Under that view, the balance equations become:

$$[(M-n)\lambda]\pi_n = [min(n+1,R)\mu(n+1)+\zeta(n+1)]\pi_{n+1}$$

Interestingly, reniging and that pressure effect kind of work in the same direction, both reduce the probability of large queue states. However, reniging is like, lost work, machines that leave unrepaired, while the pressure effect is more like accelerated recovery. In systems where both features exist at the same time, the steady state distribution concentrates even more tightly around low values of n (Baghel, 2014; Baghel, 2019; Baghel, 2020).

7.2 Server Vacations

A natural extension has server vacations, which are those stretch-es where the repair servers are effectively unavailable for reasons that are not really tied to the repair work itself (like shift changes, preventive maintenance on the servers, or even administrative tasks). In a multiple-vacation scheme, a server jumps on vacation only when the system is empty, and then it comes back just after the vacation ends, or if a new machine arrives in the meantime, whichever happens first. When you blend vacations with queue-length-dependent service you get a kind of lopsided behavior: in the high-queue situations the servers work faster, but when the queue is empty they just… head into vacation mode. The overall expected system behavior really hinges on how long those vacations last when compared to the repair time, and also on how quickly that "pressure effect" shows up after server's return (Baghel, 2021; Baghel, 2017).

7.3 Batch Arrivals and Transient Behavior

In some industrial environments, like assembly lines that keep feeding several machines from one shared source, or server farms where the same kind of fault can ripple through correlated components, machines tend to fail in batches, not one at a time. Batch arrivals mess with the usual Poisson assumption, but the system can still be treated inside a Markovian framework, as long as the batch sizes are geometrically distributed.

With batch arrivals and state-dependent service, the transient behavior really matters, meaning the time evolution before it finally settles into steady state. People often reach for a stochastic differential equation, SDE, approach, since it can describe that early time dynamics pretty well:

$$dn(t) = a(n), dt + \sqrt{b(n)}, dW(t)$$

where W(t) is a standard Brownian motion. The pressure effect enters through the drift term a(n), which becomes more strongly negative for large n, accelerating return to equilibrium (Baghel, 2022).

VIII. Conclusion

The pressure effect — that odd thing where repair workers speed up their service rate as the queue gets longer — gives a sensible and really evidence- based extension of the classical M/M/R Markovian repair model. Instead of keeping service rates fixed, we swap them out for queue length dependent functions μ(n), and you end up with a more flexible model that just reflects how repair setups actually act when things get tense.

The math fallout is pretty big. In steady state, with that pressure effect, the distributions tend to push less probability mass into the large queue regimes, so the expected sojourn times come down, and machine availability goes up versus the fixed rate versions. For one representative manufacturing setup at critical loading, this effect can raise availability by about 7–8 percentage points, even though staffing stays the same, which is economically pretty important.

There are also three functional forms — linear, exponential, and bounded — and each one sort of encodes a different story for how the speed-up grows and then levels off. The bounded version is, arguably, the most realistic, because it gives you that neat interpretation: the workers can only adapt until a kind of physiological ceiling, and after that, additional queue pressure gives no extra speed-up. That ceiling also helps keep the equations behaving, it prevents the weird pathological dynamics that unbounded acceleration models can produce.

Connections to diffusion approximations, reneging conduct, server vacations, and batch arrivals show that the pressure effect kindof stitches itself into a wider stochastic modeling framework. Future work should empirically calibrate pressure functions from actual shop floor data, and compare them across industry segments, plus check for time varying pressure effects where the repair rate function shifts when shifts change, or where fatigue kicks in.

The core message is simple but still important, system availability is not only about how many servers you have. It is also about how they respond, under pressure.

References

[1]. Baghel, K. P. S. (2013). Generalists vs. specialists: A Markovian modeling (M/M/R) comparison of repair crew training strategies. *Quest Journals: Journal of Research in Applied Mathematics, 1*(1), 10–15. http://www.questjournals.org

[2]. Baghel, K. P. S. (2014). Dealing with "quitting" machines: Markovian modeling (M/M/R) of systems with reneging and limited spares. *International Journal of Mathematics and Statistics Invention (IJMSI), 2*(1), 56–61. http://www.ijmsi.org

[3]. Baghel, K. P. S. (2017). Preventive vs. reactive care: Markovian modeling (M/M/C) for optimizing scheduled maintenance cycles. *International Journal of Mathematics and Statistics Invention (IJMSI), 6*(4), 26–31. http://www.ijmsi.org

[4]. Baghel, K. P. S. (2018). Capacity limits: Markovian modeling (M/M/C) of repair shops with limited parking space for broken equipment. *Quest Journals: Journal of Research in Applied Mathematics, 4*(2), 35–41. http://www.questjournals.org

[5]. Baghel, K. P. S. (2019). Comparative study of single vs batch service policies in M/M/C queueing systems with breakdown and repair mechanisms. *IOSR Journal of Mathematics (IOSR-JM), 15*(2, Ser. I), 85–91. https://doi.org/10.9790/5728-1502018591

[6]. Baghel, K. P. S. (2019). Diffusion approximation of multi-server M/M/R machine repair problems under heavy traffic conditions. *International Journal of Engineering and Science Invention (IJESI), 8*(5), 84–90. http://www.ijesi.org

[7]. Baghel, K. P. S. (2019). Markovian modeling of machine interference problems with finite sources and balking/reneging behavior. *Quest Journals: Journal of Research in Applied Mathematics, 5*(12), 1–6. http://www.questjournals.org

[8]. Baghel, K. P. S. (2020). Queue-length dependent service rates in flexible manufacturing systems: A diffusion process approach. *Quest Journals: Journal of Research in Applied Mathematics, 6*(8), 1–6. http://www.questjournals.org

[9]. Baghel, K. P. S. (2020). Reliability and availability analysis of multi-component repairable systems using M/M/R models with standby spares. *International Journal of Engineering and Science Invention (IJESI), 9*(9), 87–92. http://www.ijesi.org

[10]. Baghel, K. P. S. (2020). Stochastic analysis of machine repair systems with reneging and limited spares using M/M/C queue models. *International Journal of Mathematics and Statistics Invention (IJMSI), 8*(6), 43–48. http://www.ijmsi.org

[11]. Baghel, K. P. S. (2021). Optimal spare parts inventory control in machine repair models with server vacations and reneging customers. *International Journal of Mathematics and Statistics Invention (IJMSI), 9*(1), 44–49. http://www.ijmsi.org

[12]. Baghel, K. P. S. (2021). Optimization of service capacity in M/M/C machine repair systems under uncertain demand and preventive maintenance policies. *Quest Journals: Journal of Research in Applied Mathematics, 7*(4), 39–44. http://www.questjournals.org

[13]. Baghel, K. P. S. (2021). Performance evaluation of flexible manufacturing systems with batch service and machine failures using queueing networks. *IOSR Journal of Mathematics (IOSR-JM), 17*(6, Ser. I), 58–63. https://doi.org/10.9790/5728-1706015863

[14]. Baghel, K. P. S. (2022). Transient behavior of machine repair queues with batch arrivals and generalized reneging using stochastic differential equations. *IOSR Journal of Mathematics (IOSR-JM), 18*(1, Ser. I), 61–64. https://doi.org/10.9790/5728-1801016164

[15]. Baghel, K. P. S., & Jain, M. (2020). Heterogeneous servers Markov queue with single and batch service. *Journal of Science and Technological Researches, 2*(4), 25–30. https://doi.org/10.51514/JSTR.2.4.2020.25-30

[16]. Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2018). *Fundamentals of queueing theory* (4th ed.). Wiley. https://doi.org/10.1002/9781118625651

[17]. Hu, L., Yue, D., & Li, J. (2012). Availability analysis and optimization for a general repairable series-parallel system with repairmen vacations. *International Journal of Production Economics, 135*(1), 431–440. https://doi.org/10.1016/j.ijpe.2011.09.004

[18]. Jain, M., Baghel, K. P. S., & Jadown, M. (2003). A repairable system with spares, state-dependent rates and additional repairman. *Journal of Rajasthan Academy of Physical Sciences, 2*(3), 181–190.

[19]. Jain, M., Baghel, K. P. S., & Shekhar, C. (2003). M/G/1 system with standbys and common cause failure. *Journal of Rajasthan Academy of Physical Sciences, 2*(4), 239–250.

[20]. Jain, M., Maheshwari, S., & Baghel, K. P. S. (2008). Queueing network modelling of flexible manufacturing system using mean value analysis. *Applied Mathematical Modelling, 32*(5), 700–711. https://doi.org/10.1016/j.apm.2007.02.031

[21]. Jain, M., Sharma, G. C., & Baghel, K. P. S. (2004). N-policy for M/G/1 machine repair model with mixed standby components, degraded failure and Bernoulli feedback. *International Journal of Engineering, Transactions B: Applications, 17*(3), 279–288.

[22]. Ke, J.-C., Liu, T.-H., & Yang, D.-Y. (2018). Machine repairing systems with standby switching failures and unreliable repairmen. *Computers & Industrial Engineering, 120*, 212–220. https://doi.org/10.1016/j.cie.2018.04.044

[23]. Sharma, G. C., Jain, M., Baghel, K. P. S., & Shinde, V. (2004). Performance modelling of machining system with mixed standby components, balking and reneging. *International Journal of Engineering, 17*(2), 169–180.

[24]. Shekhar, C., Deora, P., Varshney, S., Singh, K. P., & Sharma, D. C. (2021). Optimal profit analysis of machine repair problem with repair in phases and organizational delay. *International Journal of Mathematical, Engineering and Management Sciences (IJMEMS), 6*(1), 442–468. https://doi.org/10.33889/IJMEMS.2021.6.1.027

[25]. Wang, K.-H., & Ke, J.-C. (2003). Probabilistic analysis of a repairable system with warm standbys plus balking and reneging. *Applied Mathematical Modelling, 27*(4), 327–336. https://doi.org/10.1016/S0307-904X(02)00133-6

[26]. Whitt, W. (1992). Understanding the efficiency of multi-server service systems. *Management Science, 38*(5), 708–723. https://doi.org/10.1287/mnsc.38.5.708