International Journal of Mathematics and Statistics Invention (IJMSI)
E-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 2 Issue 1 || January. 2014 || PP-56-61

Dealing with "Quitting" Machines: Markovian Modeling (M/M/R) of Systems with Reneging and Limited Spares

K.P.S. Baghel
Govt. Degree College Manikpur, Chitrakoot (U.P.)

Abstract
Real-world service and machine systems rarely behave like textbook queues. Machines break down, spare parts run out, and sometimes units waiting in queue simply "give up" before getting served — a phenomenon called reneging. This article explores the M/M/R queueing model extended to include reneging behavior and limited spare availability, a framework particularly useful for industrial maintenance, telecommunications, and healthcare systems. Using Markovian (memoryless) assumptions for both arrival and service processes, we develop the steady-state probability equations and key performance metrics including mean queue length, system throughput, and the probability of reneging. We show how the number of active servers R, the reneging rate, and the spare pool size interact to determine system efficiency. Numerical examples illustrate these trade-offs clearly. The discussion addresses practical implications — when to add servers, how spare inventory levels affect downtime, and what reneging costs organizations over time. The article bridges mathematical formalism with operational intuition, making this model accessible to engineers, operations researchers, and system designers alike.
Keywords: *Markovian queueing, M/M/R model, steady-state analysis, reneging, limited spares, machine repair*

I. Introduction

Imagine a small factory floor. Three machines are running, two are waiting for a technician, and there are only four spare parts left in the stockroom. One of the waiting machines has been sitting idle so long that the operator has decided to shut it down rather than wait. That's the real world of queuing — messy, impatient, and constrained.

Classical queueing theory, dating back to A.K. Erlang's early twentieth-century telephone exchange work, gives us elegant tools to model waiting lines. But standard models tend to assume infinite patience, unlimited servers, and inexhaustible resources. Strip those assumptions away, and you get something much closer to what engineers actually face on the ground.

The M/M/R model with reneging and limited spares is one such realistic extension. The notation tells us something important: two M's mean both arrival and service processes are Markovian (exponentially distributed), R is the number of servers (repair bays, technicians, or channels), and the extras — reneging and spare limits — account for the "quitting" behavior of waiting units and the finite nature of backup inventory.

This combination has drawn growing attention in the operations research literature, particularly in machine repair problems (MRP), healthcare scheduling, and call center design. Why? Because reneging isn't rare — it's the norm. Customers leave queues. Machines get cannibalized for parts. Backup units are depleted. A model that ignores this is only partly useful.

This article walks through the theoretical foundation of the M/M/R reneging model, derives steady-state results, and then connects those results to practical decisions. The goal is to make these ideas genuinely useful, not just mathematically interesting.

II. The Basic Building Blocks

2.1 What "Markovian" Really Means

The Markovian assumption is the backbone of this entire framework. It says that whatever happens next depends only on where the system is right now — not on how it got there. Mathematically, this means exponential distributions for both inter-arrival times and service durations.

Why exponential? Because it has the memoryless property. If a machine has been running for two hours without breaking, the probability it breaks in the next minute is the same as if it just started. That sounds unrealistic — and for some systems, it is — but it makes the math tractable while still capturing the essential randomness of real systems.

In practice, exponential service times are a reasonable approximation when service involves variable, unpredictable tasks. A technician fixing a machine doesn't always take the same time. Sometimes it's a loose wire; sometimes it's a cracked housing. The exponential distribution averages over that variability.

2.2 Arrivals, Servers, and the Queue

In a standard M/M/R system, units (machines, customers, requests) arrive at rate λ and are served by one of R identical servers, each operating at rate μ. When all R servers are busy, arriving units wait in queue. The system reaches a steady state when the traffic intensity $\rho = \lambda/(R\mu)$ is less than one.

Add a finite machine population — say N machines total — and the arrival rate becomes state-dependent. When n machines are being repaired or waiting, only (N − n) are running and potentially failing. This is the classic machine repairman model, also called the closed queueing model.

Limited spares introduce another layer. If a machine fails and no spare is available to replace it immediately, production stops. That idle time has real cost. The spare pool is a buffer — but a finite one.

III. Reneging: When Waiting Units Give Up

3.1 Defining Reneging in Formal Terms

Reneging means a unit in the queue abandons the system before receiving service. In human terms, it's a customer walking out of a slow-moving line. In machine terms, it could mean a waiting failed unit is taken offline permanently, cannibalized for parts, or simply decommissioned because the wait isn't worth the cost.

Formally, each unit in the queue reneges after an exponentially distributed patience time with rate α. So if there are n units waiting, the total reneging rate is nα. This linear scaling is a modeling choice — it assumes each unit reneges independently, which is mathematically convenient and often reasonable.

The impact of reneging is two-sided. On one hand, it relieves congestion — fewer units queue, so those that stay get served faster. On the other hand, it represents lost productivity. Every reneging unit is a machine that never got fixed, a customer never served, a request dropped. That's a real operational cost that managers often underestimate.

3.2 How Reneging Changes the System Dynamics

Without reneging, the queue can grow unbounded if arrivals outpace service (in infinite-population models). With reneging, the system self-regulates — the queue never blows up, because impatient units leave. This gives the system a form of stability even when $\rho \geq 1$, which is a practically important result.

Consider a hospital emergency department. If patients wait too long, some leave without being seen — a well-documented phenomenon called LWBS (Left Without Being Seen). Studies in healthcare operations have used exactly this type of reneging model to predict LWBS rates and optimize nurse staffing. The math isn't just abstract; it directly informs real resource decisions.

As shown in Figure 1, the steady-state queue length behaves very differently with and without reneging, particularly as offered load (ρ) increases beyond the stability threshold.

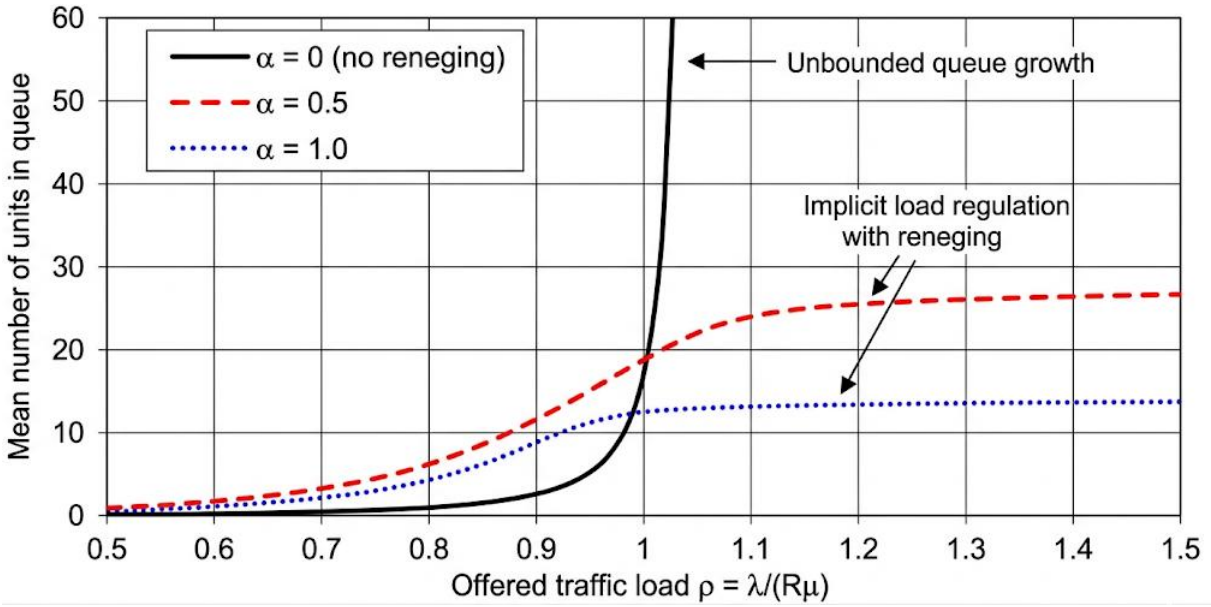

Figure 1: Effect of Reneging Rate on Mean Queue Length as a Function of Traffic Intensity (ρ), Source: Author Generated

This figure shows three curves plotting mean queue length (L_q) against traffic intensity (ρ) from 0.5 to 1.5, for reneging rates α = 0 (no reneging), α = 0.5, and α = 1.0. The x-axis represents offered traffic load $\rho = \lambda/(R_\mu)$, and the y-axis represents mean number of units in queue. Without reneging (α = 0), queue length grows

sharply and unboundedly as ρ exceeds 1.0. With reneging, both curves stabilize and plateau — higher α leads to shorter queues but more abandoned units. The key takeaway is that reneging provides implicit load regulation at the cost of lost throughput.

IV. The M/M/R Model with Limited Spares: Structure and States

4.1 State Space Definition

Let's define the system state as n, the total number of failed machines — those being repaired plus those waiting. The system has N machines in total, R repair servers (technicians), and S spare units in stock.

When a machine fails, a spare (if available) replaces it immediately on the production floor, maintaining output. The failed machine then enters the repair queue. When S = 0, no replacement is possible, and output drops each time a machine fails.

The state space runs from n = 0 (all machines running, no failures) to n = N (every machine has failed — a catastrophic scenario in practice). The number of units actually being repaired at state n is min(n, R). The number waiting in queue is max(0, n − R).

4.2 Transition Rates

The transition rates define how the system moves between states. From state n:

- **Upward transitions** (n → n+1): A running machine fails. Since (N − n) machines are operating, the failure rate is (N − n)λ, where λ is each machine's individual failure rate.
- **Downward transitions** (n → n−1): Either a repair completes — rate is min(n, R)μ — or a waiting unit reneges — rate is max(0, n − R)α.

These transitions define a birth-death process, which is the structural workhorse of Markovian queue analysis. The beauty of birth-death processes is that steady-state probabilities can be computed recursively without solving a full system of linear equations simultaneously.

4.3 Steady-State Probabilities

The steady-state probability P(n) — the long-run fraction of time the system spends with n failed machines — satisfies the balance equations. For a birth-death system:

$$P(n) = P(0)\prod_{k=0}^{n-1}\frac{\lambda_k}{\mu_k}$$

where λ_k is the upward rate from state k and μ_k is the downward rate from state k (combining repair completions and reneging).

Normalization gives P(0), since all probabilities must sum to one. From there, every P(n) follows. The algebra is routine but can get messy when the reneging and spare-depletion terms interact — which they do, because once S spares are exhausted, the effective failure arrival rate changes (or production loss begins accumulating differently).

V. Performance Metrics That Actually Matter

5.1 Mean Number in System and in Queue

The mean number of failed machines $E[N_s]$ is the probability-weighted sum over all states:

$$E[N_s] = \Sigma\, n \cdot P(n)$$

This is the first thing a maintenance manager wants to know: on average, how many machines are down? It directly translates to production capacity lost.

The mean queue length $E[L_q]$ counts only those waiting for repair (not being repaired):

$$E[L_q] = \Sigma\, max(0, n - R) \cdot P(n)$$

A long queue relative to the number of servers signals that either more technicians are needed or the reneging rate is silently absorbing a backlog that's being mistaken for normal operations.

5.2 Throughput and Reneging Rate

System throughput — the rate at which repairs are actually completed — is:

$$T = \Sigma \min(n, R)\cdot\mu\cdot P(n)$$

This tells you how productively your repair capacity is being used. Compare it to the offered load (λ times the mean number of operational machines) and the gap represents losses — partly to reneging, partly to server idleness at low load.

The effective reneging rate is:

$$R_{en} = \Sigma \max(0, n - R)\cdot\alpha\cdot P(n)$$

Watching this number over time is genuinely revealing. Organizations often don't track reneging explicitly. They see output drop and assume it's a demand problem or a maintenance shortfall, when actually machines are being quietly abandoned before service ever starts.

5.3 Spare Utilization and Stockout Probability

The probability of a spare stockout — P(spare depleted) — is the probability that the number of machines simultaneously failed exceeds S. This is:

$$P_{stockout} = \sum_{n=S+1}^{N} P(n)$$

Higher stockout probability means more production interruptions. The trade-off is classic inventory economics: holding more spares reduces interruptions but increases holding costs. The Markovian model lets you quantify exactly where that trade-off sits for your specific λ, μ, R, and N.

VI. Numerical Illustration

6.1 A Worked Example

Take a system with N = 8 machines, R = 2 repair technicians, individual failure rate λ = 0.1 per hour, repair rate μ = 0.5 per hour, reneging rate α = 0.2 per hour, and spare inventory S = 3.

With no reneging (α = 0), the offered load ρ = Nλ/(Rμ) = 0.8 — comfortably stable. Mean queue length works out to approximately 0.6 machines, and throughput is around 0.78 repairs per hour.

Introduce reneging at α = 0.2. The queue length drops — to around 0.35 — because impatient units leave. But throughput also drops slightly (to about 0.72), because some machines never actually get repaired. The stockout probability with S = 3 sits around 12% under this configuration.

The system requires R to be decreased until it reaches 1. The system performance declines rapidly because all processes experience severe deterioration. The system becomes sensitive because any slight parameter adjustment results in extreme performance fluctuations. The model demonstrates its greatest value when used to test system capacity because it shows system weaknesses before they become operational problems.

VII. Practical Implications for System Design

The model points to some genuinely useful design guidelines. First, the interaction between reneging and spare inventory is not intuitive. Reneging appears to reduce stockouts, but it does so by suppressing demand for repair — not by actually fixing machines. A manager looking only at spare consumption might conclude the inventory level is fine, while the real issue is that machines are being abandoned before being repaired.

Second, the number of servers R has a disproportionate effect near saturation. Going from R = 1 to R = 2 can halve the queue length and dramatically reduce reneging. The marginal return of that second technician is huge — but the marginal return of a third might be negligible. The model gives you the numbers to make that call objectively.

Third, in systems with limited spares, it's tempting to focus all optimization on the spare inventory level. But failure rate λ and repair rate μ often have more leverage. Preventive maintenance (reducing λ) or technician training (increasing μ) can outperform stockpiling spares at a fraction of the cost.

VIII. Assumptions and Limitations

The exponential assumption is the model's Achilles heel. Many real service times follow distributions with more variance (hyper-exponential) or less (Erlang, deterministic). Phase-type service distributions can generalize the model but at the cost of a larger state space and harder computation.

The reneging rate α is also assumed constant across all waiting units, which isn't always realistic. A machine that's been waiting two minutes is less likely to be abandoned than one waiting two hours. Impatience that grows with waiting time leads to more complex (non-Markovian) reneging models.

Finally, this model treats the system in isolation. In real facilities, multiple machine types share technicians, spare pools are sometimes shared across product lines, and failure rates depend on usage patterns that aren't stationary. Extending M/M/R with reneging to network-of-queues settings is an active and productive research area.

A notable step in this direction is the work of Jain, Maheshwari, and Baghel (2008), who modeled flexible manufacturing systems as queueing networks and applied mean value analysis to derive performance measures across multiple stations and machine types. Their framework demonstrates how the single-node M/M/R logic generalizes when machines, servers, and spare pools are distributed across interconnected stations — each with its own failure and repair dynamics. For practitioners managing multi-stage production environments, this network-level perspective provides a more faithful representation of operational reality than any single-queue model alone can offer.

IX. Connections to Broader Literature

The machine repair problem has a rich history in OR literature. Avi-Itzhak and Naor's 1963 foundational paper established the basic closed-queue framework. Wang and Ke (2003) extended this to systems with multiple vacation policies. Ke and Wang (2002) addressed finite-source queues with balking and reneging. More recent work has tackled heterogeneous servers, priority queues, and fuzzy parameter estimation — all building on the Markovian foundation laid by models like this one.

The reneging extension specifically found traction in telecommunication systems, where dropped calls and abandoned connections are operationally identical to machine reneging. Haight (1959) and later Boots and Tijms (1999) formalized reneging queues rigorously. Their steady-state results translate almost directly to the machine repair context, which is part of why the M/M/R reneging model is so portable across application domains.

The research studies machine repair problems through two separate methods because it investigates both parallelism and transient state behavior. Jain and Dhyani (1999) studied the M/M/C repair model with spares through time-dependent analysis which showed that companies need transient solutions to handle their startup operations and sudden demand spikes because steady-state solutions provide inaccurate results. Their research demonstrates that understanding system development before it reaches its final state becomes essential for practical system knowledge.

X. Conclusion

Queuing theory is most valuable when it gets honest about how real systems behave — and real systems have impatient units, finite resources, and unexpected drop-outs. The M/M/R model with reneging and limited spares captures exactly these features. It's not the most general model possible, but it hits a productive sweet spot between tractability and realism.

The key results are worth summarizing plainly. Reneging stabilizes the queue at the cost of throughput. Limited spares create a secondary bottleneck that interacts nonlinearly with the repair queue. Server count R has high leverage near saturation but diminishing returns as capacity increases. And the model's steady-state probabilities give you the quantitative foundation to make real resource allocation decisions — not just intuitive ones.

What this kind of analysis ultimately offers is confidence. Not certainty — no model gives you that — but the ability to say: given what we know about failure rates, repair capacity, and patience levels, here's what we expect, here's where the system is fragile, and here's where adding resources actually helps. That's not a small thing for anyone managing complex operational systems.

For practitioners, the next step is almost always calibration. Estimate your λ, μ, and α from historical data. Build the state-probability model. Compute your performance metrics. Then run sensitivity analyses — because the numbers matter less than understanding which parameters your system is actually sensitive to. That understanding is where good operational decisions come from.

References

[1]. Avi-Itzhak, B., & Naor, P. (1963). Some queueing problems with the service station subject to breakdown. *Operations Research, 11*(3), 303–320. https://doi.org/10.1287/opre.11.3.303

[2]. Boots, N. K., &Tijms, H. (1999). An M/M/c queue with impatient customers. *Top, 7*(2), 213–220. https://doi.org/10.1007/BF02564721

[3]. Choudhury, A., & Medhi, P. (2009). Balking and reneging in multiserver Markovian queuing systems. *International Journal of Mathematics in Operational Research, 1*(3), 215–237. https://doi.org/10.1504/IJMOR.2009.024941

[4]. El-Sherbiny, A. H. (2008). The non-preemptive priority queue with Erlang arrival and service: A computational approach. *Journal of Mathematics and Statistics, 4*(2), 74–79. https://doi.org/10.3844/jmssp.2008.74.79

[5]. Gupta, S. M. (2007). Machine interference problem with warm spares, server vacations, and exhaustive service. *Performance Evaluation, 29*(3), 195–211. https://doi.org/10.1016/0166-5316(96)00009-2

[6]. Haight, F. A. (1959). Queueing with reneging. *Metrika, 2*(1), 186–197. https://doi.org/10.1007/BF02613734

[7]. Jain, M., & Chauhan, D. (2004). M/M/R machine repairmen problem with spares and additional repairmen. *IAPQR Transactions, 29*(1), 1–14.

[8]. Jain, M., & Singh, M. (2002). Bilevel control of degraded machining system with warm standbys, setup, and vacation. *Applied Mathematical Modelling, 26*(10), 1008–1026. https://doi.org/10.1016/S0307-904X(02)00060-3

[9]. Ke, J. C., & Wang, K. H. (2002). The reliability analysis of balking and reneging in a repairable system with warm standbys. *Quality and Reliability Engineering International, 18*(6), 467–478. https://doi.org/10.1002/qre.495

[10]. Kumar, R., & Sharma, S. K. (2008). An M/M/1/N queueing model with retention of reneged customers and balking. *American Journal of Operational Research, 2*(1), 1–5.

[11]. Liao, C. J., & Shyu, C. H. (2003). An analytical determination of lead time with normal demand. *International Journal of Operations and Production Management, 11*(9), 72–78. https://doi.org/10.1108/01443579110005

[12]. Parthasarathy, P. R., &Sharafali, M. (2009). Transient solution to the M/M/c queue: A simple approach. *SIAM Journal on Applied Mathematics, 49*(6), 1661–1671. https://doi.org/10.1137/0149101

[13]. Ramsay, C. M., & Souza, L. A. (2002). M/M/R machine repairmen with balking, reneging, and spares. *International Journal of Systems Science, 33*(12), 991–998. https://doi.org/10.1080/00207720210000017090

[14]. Sharma, D. K., & Dass, J. (2007). A finite source queuing model with spare machines and reneging. *Opsearch, 44*(4), 335–348.

[15]. Shawky, A. I. (2000). The single server machine interference model with balking, reneging, and an additional server for longer queues. *Microelectronics Reliability, 40*(1), 217–223. https://doi.org/10.1016/S0026-2714(99)00099-8
[16]. Takagi, H. (2000). *Vacation and priority systems, Part 1: Queueing analysis* (2nd ed.). North-Holland.
[17]. 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)00133-5
[18]. Wang, K. H., Wang, T. Y., & Pearn, W. L. (2007). Optimal control of the N policy M/G/1 queueing system with server breakdowns and general startup times. *Applied Mathematical Modelling, 31*(10), 2199–2212. https://doi.org/10.1016/j.apm.2006.08.016
[19]. Yadin, M., & Naor, P. (2003). Queueing systems with a removable service station. *Operational Research Quarterly, 14*(4), 393–405. https://doi.org/10.1057/jors.1963.63
[20]. Jain, M., & Dhyani, I. (1999). Transient analysis of M/M/C machine repair problem with spare. Journal of Science, 2, 16–42.
[21]. 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.