STEADY STATE ANALYSIS OF BULK ARRIVAL, FIXED BATCH SERVICE QUEUE WITH INSPECTION, REWORK AND MULTIPLE VACATIONS

[1]S. Karpagam, [2]R. Lokesh

•

[1,2]Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,
Tamil Nadu, India.
[1]karpagammaths19@gmail.com, [2]rlokesh1020@gmail.com

Abstract

A non-Markovian bulk queueing system receives bulk arrivals through a single server that handles fixed-size batch services. The server conducts quality checks after each service to identify defective items, which are then sent to rework for quality control purposes. The server takes random-length vacations until new arrivals reach the threshold "K" before returning to service. Through the supplementary variable technique, the key performance metrics are computed together with steady-state governing equations. Using graphical representations and numerical tables, the model demonstration shows essential information about system performance through calculated data.

Keywords: bulk arrival, batch service, inspection, rework, performance measures

1. Introduction

Service industries, together with modern manufacturing, adopt batch service systems to maximize operational efficiency while maintaining high-quality control standards. A system that operates with fixed batch sizes enables better resource management by designing server operations to reduce operational expenses. Product quality maintenance reaches optimal levels through inspections that reveal defective items that require rework after each batch service. The systematic implementation of this approach results in lower manufacturing faults and waste output while achieving better production dependability rates. Batch processing systems that integrate inspection procedures and rework activities serve to maximize quality control for automated production environments as well as additive manufacturing and assembly lines. The batch service framework helps industries achieve both productivity growth and defect-free output, which results in better customer satisfaction and production cost decreases.

Lee et al. [16] investigated fixed-batch service queues with Poisson arrival patterns and vacation periods. The study used decomposition methods to generate queue length distributions both for basic vacation systems and multiple vacation systems. Ayyappan et al. [1] applied fixed batch service together with multiple vacations and gated service to study transportation and manufacturing processes. By combining generating functions and Rouche's theorem, they developed precise solutions for queue probabilities as well as performance metrics. Goswami et al. [8] established a novel method to study sojourn-time distributions in $M/G^a/1$ queues with batch service through their derivation of probability density function expressions using partial fractions and the residue theorem. Yuvarani and Vijayalakshmi [24] studied a multistage batch arrival, batch service queue system that incorporated multiple vacation policies and server breakdowns and delayed repairs. They developed solutions for transient and steady-state conditions by using the supplementary variable technique.

Koyama et al. [12] conducted their study based on services from sharing economies, including ride-sharing and shared-taxi services. Their model presented a $M + M(K)/M/1$ queue system that allowed customers to decide between individual or combined services. Ghosh et al. [7] conducted a thorough analysis of a finite-buffer queueing system that incorporates correlated batch arrivals and batch-size-dependent bulk service. Van Ommeren et al. [23] studied bulk-service multi-server queues featuring heterogeneous server capacities that perfectly fit passenger transport and package delivery scenarios. Their decomposition-based method using renewal theory principles delivered a major breakthrough in analyzing multi-type server systems operating in complex service environments.

Chen et al. [4] investigated a modified Markovian queue system that used bulk arrivals and bulk services with general state-dependent control protocols. The study began with analyzing a bulk-arrival and bulk-service queue system in a resting state to reveal its relationship with broader queueing models. Kumar [13] investigated a single-server queue slideshow network that incorporated discouragement features during multiple vacation periods. D'Arienzo et al. [5] investigated a single-server retrial queue system that utilized MAP flow and PH service times and featured a finite capacity pool for customer accumulation and group service. The model implemented a threshold system that started the next group service only after reaching a predetermined customer count in the pool. Their main contribution involved demonstrating that group service durations can be shorter than the combined lengths of individual services because of batch processing advantages.

Pradhan et al. [20] investigated an infinite-buffer batch-service queue with Markovian arrivals and batch-size-dependent service times to derive performance metrics while examining arrival process correlation effects. Germs and Van Foreest [6] investigated state-dependent finite-buffer bulk queues through semi-regenerative analysis to derive queue length distributions. Kumar [14] examined a population model under catastrophic conditions, which he modeled through renewal growth processes along with Poisson catastrophes, utilizing geometric patterns to represent batch-sized population effects. Somasudaram et al. [21] investigated a batch arrival single service queue that included optional service and working breakdown. Chaudhry et al. [3] explored the queueing system with Poisson arrival patterns along with bulk service characteristics within limited capacity constraints.

Niranjan and Indhira [17] developed a comprehensive analysis of the batch service model. In more recent surveys, Panta et al. [19] and Upadhyaya [22] discussed additional types of models for vacation queueing systems, each with a different service policy. Niranjan et al. [18] studied a bulk queueing model with vacations and queue length and service request-based load adjustments in data center architecture to optimize cloud computing traffic distribution through power business intelligence and queueing theory. Haridass and Arumuganathan [9] developed a new batch service queuing framework that includes three essential features: multiple server vacations as well as setup durations followed by server-controlled re-service entry decisions.

Karpagam [11] developed a batch arrival bulk service queue model that incorporates unreliable servers and service resumption and rework for faulty items and multiple vacations with threshold policies for manufacturing systems. Ayyappan and Karpagam [2] developed a non-Markovian model that integrates batch arrivals with general bulk service and server breakdowns and multiple vacations and re-service and a standby server. Jeyakumar and Arumuganathan [10] developed a non-Markovian single-server queueing model featuring Poisson batch arrivals and general bulk service with controlled optional re-service based on probabilities and queue length-dependent vacation policies that trigger multiple vacations until reaching a threshold.

This paper focuses on analyzing the $M^X/G^K/1$ queueing system with inspection, rework, and multiple vacations where bulk arrivals adhere to a compound Poisson process denoted by M^X and fixed-size batch service durations follow a general distribution marked by G^K alongside a single server identified by 1.

The rest of the paper has been divided into eight sections. Section 2 gives practical application for the suggested model. Section 3 presents the model description, which includes Figure 1 schematic diagram and the model notation. Section 4 explores the steady-state queue size distribution, and Section 5 introduces the probability-generating function that describes queue-size distribution in steady state. Section 6 establishes key performance metrics. Numerical results together with graphical comparisons appear in Section 7. Finally, Section 8 concludes the paper.

2. Real-world application of the proposed model

The implementation of a fixed batch processing system in industrial 3D printing (Additive Manufacturing - AM) ensures printing automation through waiting for at least K parts to prepare while both optimizing machine performance and decreasing energy consumption. The system remains in vacation mode when the queue contains fewer than (K-1) parts. The AI driving inspection system ensures that the quality checks of the printing are met by detecting misalignment of the layer and other problems through CT analysis and optical scanning methods. The defective components need rework that includes laser remelting for metal parts and UV curing for resin prints together with heat treatment for thermoplastics to minimize reprinting requirements. The system processes final finishing and packaging that follows quality inspections of the high-quality outputs before they move to shipment. Manufacturing through this process benefits sectors including aerospace (turbine blades), medical (implants, prosthetics), automotive (3D-printed brackets), and fashion (customized footwear), while enhancing both production speed and waste reduction and providing accurately precise results.

3. Detailed model description

In this section, the proposed system is modeled using the supplementary variable technique, incorporating the remaining service time of the batch in service and the remaining vacation time of the server as supplementary variables at an arbitrary time. The schematic representation of the model is shown in Figure 1.

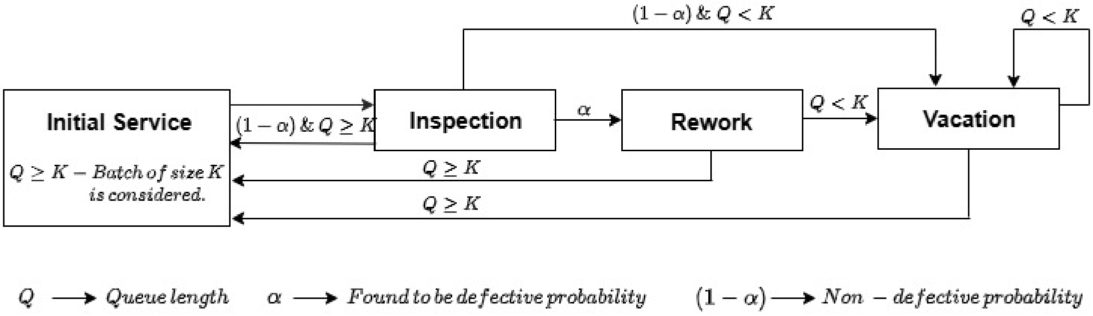
$(1-\alpha)\ \&\ Q < K$
$Q < K$
Initial Service
$Q \geq K$ – Batch of size K is considered.
$(1-\alpha)\ \&\ Q \geq K$
Inspection
α
Rework
$Q < K$
Vacation
$Q \geq K$
$Q \geq K$
$Q \longrightarrow$ Queue length
$\alpha \longrightarrow$ Found to be defective probability
$(1-\alpha) \longrightarrow$ Non – defective probability

Figure 1: *Pictorial depiction of the model*

The server operates with batch services of a fixed size 'K'. The initial service begins only when the queue reaches 'K'. Upon completing a service, the batch undergoes an inspection. If defects are found, the batch moves to rework with probability 'α' If no defects are detected, the system checks the queue length:

- If the queue length meets or exceeds 'K', the next service starts immediately with probability '$(1-\alpha)$'.
- If the queue length is less than 'K', the server goes on vacation with probability '$(1-\alpha)$'.

After rework, a similar decision process follows:

- If the queue length reaches or exceeds 'K', the next service starts.
- If the queue length remains below 'K', the server enters a vacation.

This vacation cycle continues until the queue accumulates 'K' parts, at which point service starts.

3.1. Notations

The model parameters include the arrival rate λ, the group size random variable X with probability $g_i = \Pr(X = i)$, and its probability generating function $X(z)$. The mean batch size is denoted as $E(X) = X_1$. Let $\aleph_s(t)$ and $\aleph_q(t)$ denote the number of products in service and queue at time 't'. Let $\chi(t)$ denote the server state at time 't'. The basic notations are listed in Table 1:

Table 1: *The list of notations used in this paper*

Function Type	Initial service time	Inspection time	Rework time	Vacation Time
Cumulative Density Function (CDF)	$M_b(\hbar)$	$H_1(\hbar)$	$H_2(\hbar)$	$V(\hbar)$
Probability Density Function (PDF)	$m_b(\hbar)$	$h_1(\hbar)$	$h_2(\hbar)$	$v(\hbar)$
Laplace-Stieltjes Transform (LST)	$\tilde{M}_b(\mho)$	$\tilde{H}_1(\mho)$	$\tilde{H}_2(\mho)$	$\tilde{V}(\mho)$
Remaining Service Time	$M_b^0(t)$	$H_1^0(t)$	$H_2^0(t)$	$V^0(t)$

1. $\chi(t) \ = 1, 2, 3$ and 4 denote the server is on initial service, inspection, rework and vacation, respectively.
2. $\pounds(t) \ = l$, if the server is on l^{th} vacation.

We define the following state probabilities as follows :

$$F_{K,j}(\hbar, t)\Delta t = Pr\{\aleph_s(t) = K,\ \aleph_q(t) = j,\ \hbar \leq {M_b}^0(t) < \hbar + \Delta t,\ \chi(t) = 1\},\ j \geq 0;$$
$$Q_n(\hbar, t)\Delta t = Pr\{\aleph_q(t) = n,\ \hbar \leq H_1^0(t) < \hbar + \Delta t,\ \chi(t) = 2\}, \text{where } n \geq 0;$$
$$R_n(\hbar, t)\Delta t = Pr\{\aleph_q(t) = n,\ \hbar \leq {H_2}^0(t) < \hbar + \Delta t,\ \chi(t) = 3\}, \text{where } n \geq 0;$$
$$W_{l,j}(\hbar, t)\Delta t = Pr\{\aleph_q(t) = j,\ \hbar \leq V^0(t) < \hbar + \Delta t,\ \chi(t) = 4, \pounds(t) = l\}, \text{where } j \geq 0, l \geq 1.$$

4. Steady state queue size distribution

This section develops the probability-generating function (PGF) for the queue size at an arbitrary time point to determine essential performance metrics. The first step involves developing steady-state difference-differential equations. The Laplace - Stieltjes transform applied to these equations combined with the method from Lee et al. [15] leads to a PGF solution through algebraic steps.

Server is in initial service

$$-F_{K,0}^{'}(\hbar) = -\lambda F_{K,0}(\hbar) + R_K(0)m_b(\hbar) + (1-\alpha)Q_K(0)m_b(\hbar) + \sum_{l=1}^{\infty} W_{n,K}(0)m_b(\hbar), \tag{1}$$

$$-F_{K,j}^{'}(\hbar) = -\lambda F_{K,j}(\hbar) + R_{K+j}(0)m_b(\hbar) + (1-\alpha)Q_{K+j}(0)m_b(\hbar) + \sum_{l=1}^{\infty} W_{n,K+j}(0)m_b(\hbar)$$
$$+ \sum_{i=1}^{j} F_{K,j-i}(\hbar)\lambda g_i,\ j \geq 1, \tag{2}$$

Server is in inspection (3)

$$-Q_0(\hbar) = -\lambda Q_0(\hbar) + F_{K,0}(0)h_1(\hbar), \tag{4}$$

$$-Q_n(\hbar) = -\lambda Q_n(\hbar) + F_{K,n}(0)h_1(\hbar) + \sum_{i=1}^{n} Q_{n-i}(\hbar)\lambda g_i, \quad n \geq 1, \tag{5}$$

Server is in rework (6)

$$-R_0(\hbar) = -\lambda R_0(\hbar) + \alpha Q_0(0)h_2(\hbar), \tag{7}$$

$$-R_n(\hbar) = -\lambda R_n(\hbar) + \alpha Q_n(0)h_2(\hbar) + \sum_{i=1}^{n} R_{n-i}(\hbar)\lambda g_i, \quad n \geq 1. \tag{8}$$

Server is in vacation

$$-W_{1,0}^{'}(\hbar) = -\lambda W_{1,0}(\hbar) + R_0(0)v(\hbar) + (1-\alpha)Q_0(0)v(\hbar), \tag{9}$$

$$-W_{1,n}^{'}(\hbar) = -\lambda W_{1,n}(\hbar) + R_n(0)v(\hbar) + (1-\alpha)Q_n(0)v(\hbar) + \sum_{i=1}^{n} W_{1,n-i}(\hbar)\lambda g_i,$$
$$1 \leq n \leq K-1, \tag{10}$$

$$-W_{1,n}^{'}(\hbar) = -\lambda W_{1,n}(\hbar) + \sum_{i=1}^{n} W_{1,n-i}(\hbar)\lambda g_i,\ n \geq K \tag{11}$$

$$-W_{l,0}^{'}(\hbar) = -\lambda W_{l,0}(\hbar) + W_{l-1,0}(0)v(\hbar),\ l \geq 2, \tag{12}$$

$$-W_{l,n}^{'}(\hbar) = -\lambda W_{l,n}(\hbar) + W_{l-1,0}(0)v(\hbar) + \sum_{i=1}^{n} W_{l,n-i}(\hbar)\lambda g_i,\ 1 \leq n \leq K-1,\ l \geq 2, \tag{13}$$

$$-W_{l,n}^{'}(\hbar) = -\lambda W_{l,n}(\hbar) + \sum_{i=1}^{n} W_{l,n-i}(\hbar)\lambda g_i,\ n \geq K,\ l \geq 2. \tag{14}$$

Using the LST on both sides of equations (1) to (14), we get

$$\mho\tilde{F}_{K,0}(\mho) - F_{K,0}(0) = \lambda\tilde{F}_{K,0}(\mho) - R_K(0)\tilde{M}_b(\mho) - (1-\alpha)Q_K(0)\tilde{M}_b(\mho) - \sum_{l=1}^{\infty} W_{l,K}(0)\tilde{M}_b(\mho), \tag{15}$$

$$\mho\tilde{F}_{K,j}(\mho) - F_{K,j}(0) = \lambda\tilde{F}_{K,j}(\mho) - R_{K+j}(0)\tilde{M}_b(\mho) - (1-\alpha)Q_{K+j}(0)\tilde{M}_b(\mho) - \sum_{l=1}^{\infty} W_{l,K+j}(0)\tilde{M}_b(\mho)$$
$$- \sum_{i=1}^{j} \tilde{B}_{K,j-i}(\mho)\lambda g_i,\ j \geq 1, \tag{16}$$

$$\mho\tilde{Q}_0(\mho) - Q_0(0) = \lambda\tilde{Q}_0(\mho) - F_{K,0}(0)\tilde{H}_1(\mho), \tag{17}$$

$$\mho\tilde{Q}_n(\mho) - Q_n(0) = \lambda\tilde{Q}_n(\mho) - F_{K,n}(0)\tilde{H}_1(\mho) - \sum_{i=1}^{n} \tilde{Q}_{n-i}(\mho)\lambda g_i, n \geq 1, \tag{18}$$

$$\mho\tilde{R}_0(\mho) - R_0(0) = \lambda\tilde{R}_0(\mho) - \alpha Q_0(0)\tilde{H}_2(\mho), \tag{19}$$

$$\mho\tilde{R}_n(\mho) - R_n(0) = \lambda\tilde{R}_n(\mho) - \alpha Q_n(0)\tilde{H}_2(\mho) - \sum_{i=1}^{n} \tilde{R}_{n-i}(\mho)\lambda g_i, n \geq 1, \tag{20}$$

$$\mho\tilde{W}_{1,0}(\mho) - W_{1,0}(0) = \lambda\tilde{W}_{1,0}(\mho) - R_0(0)\tilde{V}(\mho) - (1-\alpha)Q_0(0)\tilde{V}(\mho) \tag{21}$$

$$\mho\tilde{W}_{1,n}(\mho) - W_{1,n}(0) = \lambda\tilde{W}_{1,n}(\mho) - R_n(0)\tilde{V}(\mho) - (1-\alpha)Q_n(0)\tilde{V}(\mho) - \sum_{i=1}^{n}\tilde{W}_{1,n-i}(\mho)\lambda g_i, \quad 1 \le n \le K-1, \tag{22}$$

$$\mho\tilde{W}_{1,n}(\mho) - W_{1,n}(0) = \lambda\tilde{W}_{1,n}(\mho) - \sum_{i=1}^{n}\tilde{W}_{1,n-i}(\mho)\lambda g_i, \ n \ge K, \tag{23}$$

$$\mho\tilde{W}_{l,0}(\mho) - W_{l,0}(0) = \lambda\tilde{W}_{l,0}(\mho) - W_{l-1,0}(0)\tilde{V}(\mho), \ l \ge 2, \tag{24}$$

$$\mho\tilde{W}_{l,0}(\mho) - W_{l,0}(0) = \lambda\tilde{W}_{l,0}(\mho) - W_{l-1,0}(0)\tilde{V}(\mho) - \sum_{i=1}^{n}\tilde{W}_{l,n-i}(\mho)\lambda g_i, \ 1 \le n \le K-1, l \ge 2, \tag{25}$$

$$\mho\tilde{W}_{l,n}(\mho) - W_{l,n}(0) = \lambda\tilde{W}_{l,n}(\mho) - \sum_{i=1}^{n}\tilde{W}_{l,n-k}(\mho)\lambda g_i, \ n \ge K, \ l \ge 2. \tag{26}$$

The following probability-generating functions (PGFs) are defined in order to derive the PGF for the queue size at an arbitrary time epoch:

$$\left.\begin{aligned}
\tilde{F}_K(z,\mho) &= \sum_{j=0}^{\infty}\tilde{F}_{K,j}(\mho)z^j, \quad F_K(z,0) = \sum_{j=0}^{\infty}F_{K,j}(0)z^j,\\
\tilde{Q}(z,\mho) &= \sum_{j=0}^{\infty}\tilde{Q}_j(\mho)z^j, \quad Q(z,0) = \sum_{j=0}^{\infty}Q_j(0)z^j,\\
\tilde{R}(z,\mho) &= \sum_{j=0}^{\infty}\tilde{R}_j(\mho)z^j, \quad R(z,0) = \sum_{j=0}^{\infty}R_j(0)z^j,\\
\tilde{W}_l(z,\mho) &= \sum_{j=0}^{\infty}\tilde{W}_{l,j}(\mho)z^j, \quad W_l(z,0) = \sum_{j=0}^{\infty}W_{l,j}(0)z^j, \quad l \ge 1.
\end{aligned}\right\} \tag{27}$$

By multiplying equations (15) to (26) with suitable powers of z^j and summing over $'j'$ ($j = 0$ to ∞), then by using (27) and after some mathematical manipulations, we obtain

$$(\mho - u(z))\tilde{W}_1(z,\mho) = \big(\tilde{V}(u(z)) - \tilde{V}(\mho)\big)\sum_{n=0}^{K-1}\Big[R_n(0) + (1-\alpha)Q_n(0)\Big] \tag{28}$$

$$(\mho - u(z))\tilde{W}_n(z,\mho) = \big(\tilde{V}(u(z)) - \tilde{V}(\mho)\big)\sum_{n=0}^{K-1}W_{l-1,n}(0), \ l \ge 2, \tag{29}$$

$$(\mho - u(z))\tilde{Q}(z,\mho) = \big(\tilde{H}_1(u(z)) - \tilde{H}_1(\mho)\big)F_K(z,0), \tag{30}$$

$$(\mho - u(z))\tilde{R}(z,\mho) = \big(\tilde{H}_2(u(z)) - \tilde{H}_2(\mho)\big)\alpha Q(z,0)\tilde{H}_3(u(z)), \tag{31}$$

$$\begin{aligned}
y(z)(\mho - u(z))\tilde{F}_K(z,\mho) = \big(\tilde{M}_b(u(z)) - \tilde{M}_b(\mho)\big)\Bigg[&R(z,0) - \sum_{n=0}^{K-1}R_n(0)z^n + (1-\alpha)Q(z,0)\\
&- (1-\alpha)\sum_{n=0}^{K-1}Q_n(0)z^n + \sum_{l=1}^{\infty}\Big(W_l(z,0) - \sum_{n=0}^{K-1}W_{l,n}(0)z^n\Big)\Bigg]
\end{aligned} \tag{32}$$

5. Probability generating function of the queue size

Let P(z) be the PGF of the queue size at an arbitrary time epoch. Then,

$$P(z) = \tilde{F}_K(z,0) + \tilde{R}(z,0) + \tilde{Q}(z,0) + \sum_{l=1}^{\infty}\tilde{W}_l(z,0). \tag{33}$$

Substituting $\mho = 0$ in equations (28) - (32), then (33) becomes

$$P(z) = \frac{\sum_{n=0}^{K-1} c_n z^n (\tilde{V}(u(z)) - 1) a(z)}{u(z) y(z)} \tag{34}$$

where

$$r_n = R_n(0),\ q_n = Q_n(0), v_n = \sum_{l=1}^{\infty} W_{l,n}(0),\ c_n = r_n + (1-\alpha)q_n + v_n,\ u(z) = \lambda - \lambda X(z),$$
$$y(z) = z^K - (1-\alpha)\tilde{M_b}(u(z))\tilde{H_1}(u(z)) - \alpha\tilde{M_b}(u(z))\tilde{H_1}(u(z))\tilde{H_2}(u(z)) \text{ and}$$
$$a(z) = z^K + 1 - 2(1-\alpha)\tilde{M_b}(u(z))\tilde{H_1}(u(z)) - 2\alpha\tilde{M_b}(u(z))\tilde{H_1}(u(z))\tilde{H_2}(u(z)).$$

Remark. The PGF must obtain $P(1) = 1$. This criteria is satisfied by the result obtained by equating a term to 1 and applying the L'Hopital's rule in (34). Then $\rho = (M_{b1} + H_{11} + \alpha H_{21}/K) < 1$ is the condition for the steady state for the model under consideration existence.

5.1. Computational aspects

Rouche's theorem on complex variables can be utilized to determine these constants. $y(z)$ has one on the unit circle $|z| = 1$ and $(K-1)$ zeroes within, according to Rouche's theorem. $P(z)$ is analytic both on and within the unit circle; therefore, at these points the numerator must vanish, which gives 'K' equations in 'K' unknowns. These equations can be solved using any suitable numerical technique.

6. Performance measures

Some useful results of our model are listed below.

6.1. Expected queue length

Mean number of customers in the queue is the equation (34) is obtained by using

$$E(Q) = (\frac{dP(z)}{dz})_{z=1}$$
$$E(Q) = \frac{6\sum_{n=0}^{K-1} c_n \left[W_1(2na_1 + a_2) + W_2 a_1\right](-\lambda X1)y_1 - 6W_1\sum_{n=0}^{K-1} c_n a_1 \left[(-\lambda X1)y_2 - (\lambda X2)y_1\right]}{3(3\left[(-\lambda X1)y_2 - (\lambda X2)y_1\right])^2} \tag{35}$$

where

$$y_1 = K - M_{b1} - H_{11} - \alpha H_{21}, M_{b1} = \lambda X_1 E(M_b), M_{b2} = \lambda X_2 E(M_b) + (\lambda X_1)^2 E({M_b}^2),$$
$$y_2 = K(K-1) - M_{b2} - H_{12} - 2M_{b1}H_{11} - \alpha(2H_{21}H_{11} + 2H_{21}M_{b1} + H_{21}),$$
$$a_1 = K - 2(M_{b1} + H_{11} + \alpha H_{21}), H_{11} = \lambda X_1 E(H_1), H_{21} = \lambda X_1 E(H_2)),$$
$$a_2 = K(K-1) - 2[M_{b2} + H_{12} + 2M_{b1}H_{11} + \alpha(2H_{21}H_{11} + 2H_{21}M_{b1} + H_{21})],$$
$$H_{12} = \lambda X_2 E(H_1) + (\lambda X_1)^2 E({H_1}^2), H_{22} = \lambda X_2 E(H_2) + (\lambda X_1)^2 E({H_2}^2),$$
$$W_1 = \lambda X_1 E(V), W_2 = \lambda X_2 E(V) + (\lambda X_1)^2 E(V^2).$$

6.2. Expected waiting time

Using Little's formula, we get expected waiting time at arbitrary epoch is

$$E(W) = \frac{E(Q)}{\lambda X_1} \tag{36}$$

7. Numerical Illustration

This section presents a numerical illustration of the proposed queueing model by analyzing parameter variations using MATLAB software. The computed results are displayed in both tabular and graphical formats, assuming that the batch size distribution of arrivals follows a geometric distribution with a mean of 2. For the numerical demonstration, we assume that the initial service time, inspection, rework, and vacation times follow general distributions with parameters μ_s, μ_i, μ_r, and γ, respectively. Numerical values satisfying the stability criteria are selected randomly.

λ	**Exponential**		**Erlang-2**		**Hyperexponential**	
	E(Q)	**E(W)**	**E(Q)**	**E(W)**	**E(Q)**	**E(W)**
5.00	28.0325	2.8032	27.8501	2.7850	25.4377	2.5438
5.25	31.8792	3.0361	31.8001	3.0286	29.3675	2.7969
5.50	37.9381	3.4489	38.1386	3.4671	35.7476	3.2498
5.75	48.4498	4.2130	49.5277	4.3068	47.6078	4.1398
6.00	70.4681	5.8723	75.6091	6.3008	76.9288	6.4107

Table 2: *Performance metrics for different service time distributions*

The expected queue length $E(Q)$ and expected waiting time $E(W)$ are computed for various values of λ and presented in Table 2. The default parameter values for numerical analysis are:

- Exponential distribution: $\mu_s = 6$, $\mu_i = 5$, $\mu_r = 4$, $\alpha = 0.7$, $K = 7$, $\gamma = 3$.
- Erlang-2 distribution: $\mu_s = 9$, $\mu_i = 7$, $\mu_r = 5$, $\alpha = 0.1$, $K = 7$, $\gamma = 3$.
- Hyperexponential distribution: $\mu_s = 9$, $\mu_i = 7$, $\mu_r = 5$, $\alpha = 0.1$, $K = 7$, $c = 0.9$, $d = 0.7$, $e = 0.6$, $p = 6$, $q = 4$, $r = 4$, $f_1 = 0.3$, $s = 2$, $\gamma = 3$.

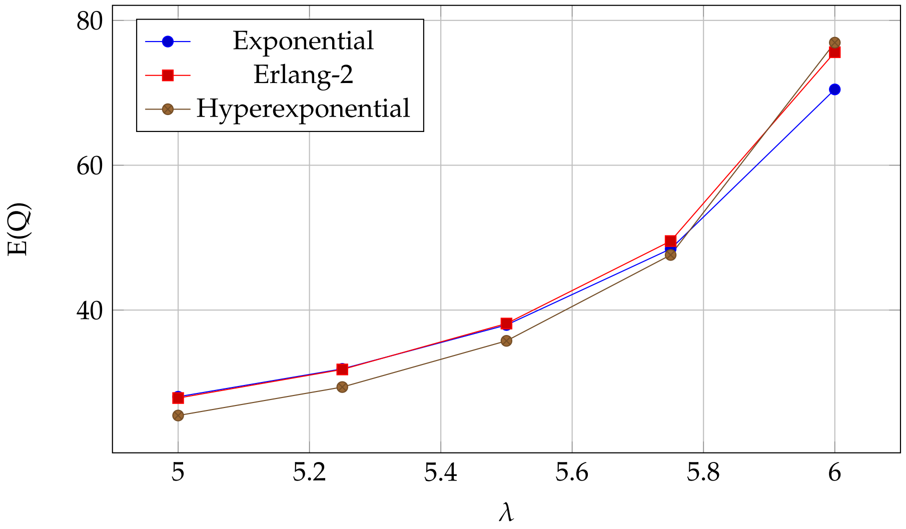

Figure 2: *Mean queue length E(Q) vs. arrival rate λ for different service time distributions.*

Figure 2 shows how system workload and queue length (E(Q)) react when production demand (λ) rises using various service time distributions. The system workload E(Q) experiences a substantial increase when demand rises, which suggests potential manufacturing, inspection, and rework stage congestion. Increased service time variability produces longer item waiting lines, which creates processing inefficiencies as well as delays in the fulfillment of orders.

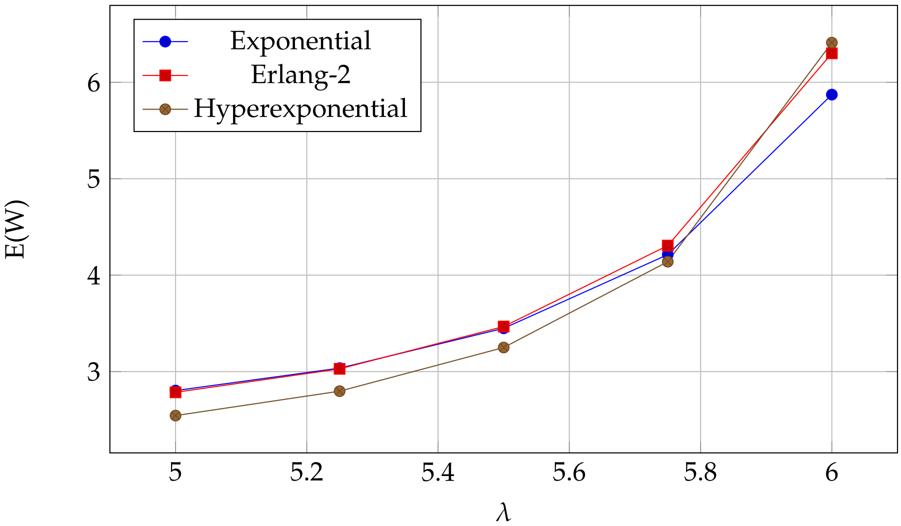

Figure 3: *Mean waiting time E(W) vs. arrival rate λ for different service time distributions.*

Figure 3 illustrates how raising demand levels prolongs waiting time (E(W)). Raising the λ value produces increased waiting durations mostly in systems whose service times show high variation. Long waiting times affect system productivity through disrupted scheduling operations. Optimizing scheduling combined with predictive maintenance systems and adaptable batch processing enables better control of operational difficulties to improve performance quality.

μ_s	$\lambda = 5$		$\lambda = 6$	
	E(Q)	**E(W)**	**E(Q)**	**E(W)**
9	24.0400	2.4040	35.3044	2.9420
10	11.0944	1.1094	22.7451	1.8954
11	10.5894	1.0589	18.0331	1.5028
12	10.3115	1.0312	15.5947	1.2996
13	10.1810	1.0181	14.1277	1.1773

Table 3: *E(Q) and E(W) for different service rates (μ_s)*

Table 3 employs Erlang-2 distributions for μ_s initial service rate along with $\mu_i = 7$ inspection rate and $\mu_r = 5$ rework rates while the vacation rate follows an exponential distribution with parameters $\gamma = 3$, $K = 7$, and $\alpha = 0.1$. The queue performance indicators E(Q) and E(W) appear in Table 3, where different service rate assumptions demonstrate system dynamic variances.

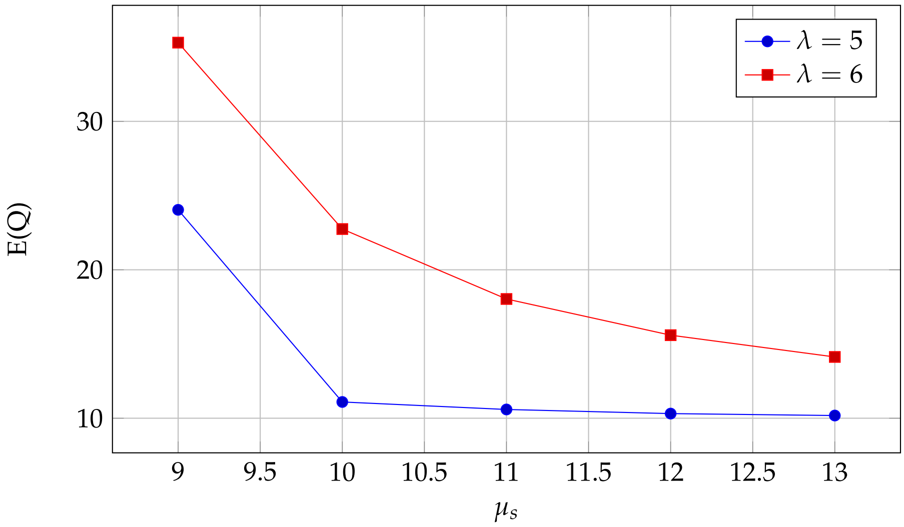

Figure 4: *E(Q) vs. Service Rate (μ_s)*

Figure 4 depicts the relationship between E(Q) and μ_s while λ varies. The E(Q) decreases when μ_s increases because this action minimizes queue congestion in all stages. E(Q) decreases through decreased queues, which improves part and order processing efficiency, especially when demand is high ($\lambda = 6$).

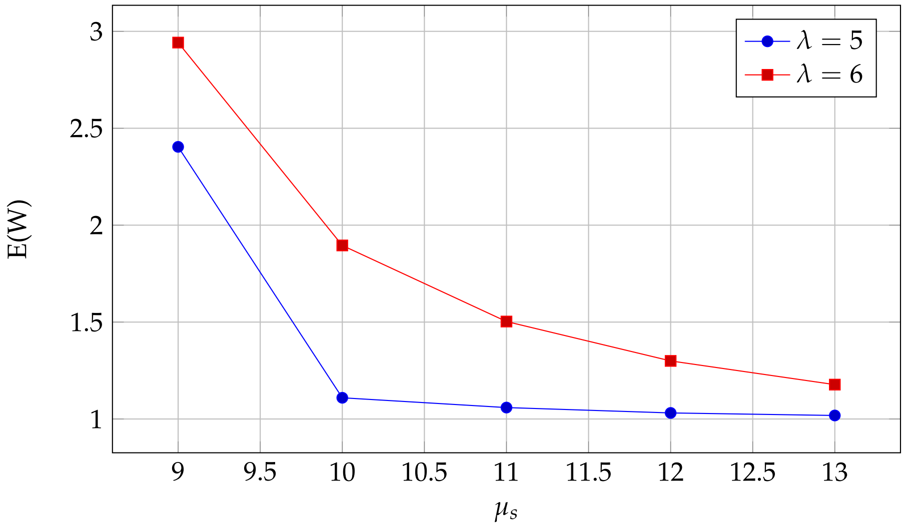

Figure 5: *E(W) vs. Service Rate (μ_s)*

Figure 5 shows how processing efficiency (μ_s) relates to the mean waiting time (E(W)) for orders. The mean waiting time decreases when processing efficiency (μ_s) rises because faster processing times generate fewer delays in order completion. Service rates play a vital role in reducing delays because higher demand levels ($\lambda = 6$) produce more pronounced effects.

Table 4 shows system parameters which contain the defect rate probability (α), service rate (μ_s), expected queue length ($E(Q)$), and expected waiting time ($E(W)$). The system adopts a fixed batch size $K = 7$ with an arrival rate $\lambda = 5$. Initial service time follows an Erlang-2 distribution with mean service rate at μ_s and inspection and rework services use Erlang-2 distributions at $\mu_i = 7$ and $\mu_r = 6$ respectively. The vacation period follows an exponential distribution with rate $f = 3$.

α	$\mu_s = 10$		$\mu_s = 11$		$\mu_s = 12$	
	E(Q)	**E(W)**	**E(Q)**	**E(W)**	**E(Q)**	**E(W)**
0.1	21.4589	2.1459	20.5613	2.0561	20.1177	2.0118
0.2	24.5557	2.4556	22.8812	2.2881	21.8684	2.1868
0.3	29.7274	2.9727	26.7080	2.6708	24.8457	2.4846
0.4	39.3315	3.9331	33.3550	3.3355	29.9064	2.9906
0.5	62.6701	6.2670	46.9792	4.6979	39.3757	3.9376
0.6	195.4627	19.5463	88.9730	8.8973	62.4679	6.2468

Table 4: *Queueing metrics for different service rates (μ_s) and defect rates (α)*

Figure 6 demonstrates how defect probability rates (α) affect expected queue length ($E(Q)$) when processing rates ($\mu_s = 10, 11, 12$) are used. An increase in the α value produces substantial growth in $E(Q)$ especially when processing rates remain low. The system experiences congestion because defect rates which increase cause more rework combined with longer inspection times. Processing rates that run efficiently work to reduce the accumulation of work in queue thereby sustaining a smooth production flow while minimizing batch manufacturing bottlenecks.

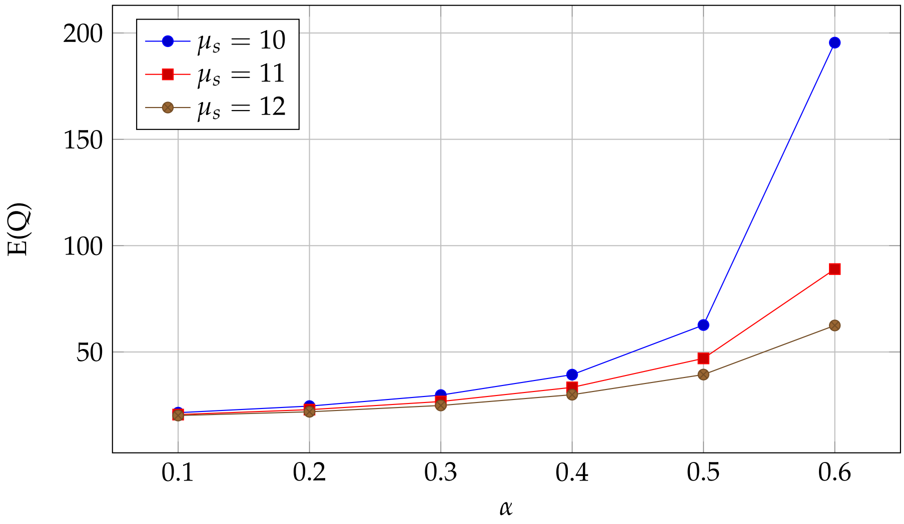

Figure 6: *E(Q) vs. Defect probability rate (α)*

Figure 7 displays how defect probabilities influence the expected waiting time ($E(W)$). The processing requirements along with rework increase because of the value increment in α. The combination of reduced service rates with μ_s intensifies these delays because it indicates the need to maintain high processing efficiency while implementing effective quality control measures. Higher quality inspection management as well as better rework procedures enhance both production speed and system operational effectiveness.

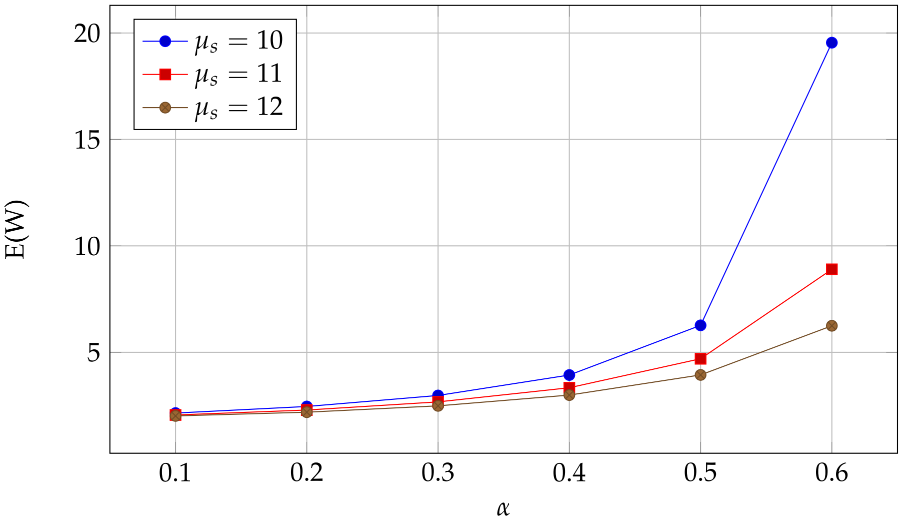

Figure 7: *E(W) vs. Defect probability rate* (α)

8. Conclusion and further scope for extension of work

The proposed model investigates a single-server non-Markovian bulk queueing system that employs bulk arrivals accompanied by fixed-size batch services and includes inspection capabilities and vacation features and rework operations. The supplementary variable technique enables steady-state analysis of the system to derive essential performance metrics. The combination of batch service policies with post-service inspection and rework operations leads to substantial improvements in both production efficiency and product quality. Numerical calculations and graphical presentation tools help reveal system behavior, allowing users to understand how batch sizes combine with arrival rates and service rates and how rework policies affect system results. Future research should expand this model through investigations of multi-server systems and dynamic batch size adaptation and priority-based rework strategies to achieve maximum efficiency. The implementation of machine learning techniques alongside adaptive vacation policies would boost the practical use of this system. The integration of cost-based optimization methods shows promise to reduce operational expenses while achieving maximum throughput and quality levels.

References

[1] Ayyappan, G., Devipriya, G., and Subramanian, A. M. G. (2014). Analysis of single server fixed batch service queueing system under multiple vacations with gated service. *International Journal of Computer Applications*, 89(5), 15–19.

[2] Ayyappan, G., and Karpagam, S. (2018). An $M^{[X]}/G(a,b)/1$ queueing system with breakdown and repair, stand-by server, multiple vacation and control policy on request for re-service. *Mathematics*, 6(6), 101.

[3] Chaudhry, M., Datta Banik, A., Barik, S., and Goswami, V. (2023). A novel computational procedure for the waiting-time distribution (in the queue) for bulk-service finite-buffer queues with Poisson input. *Mathematics*, 11(5), 1142.

[4] Chen, A., Wu, X., and Zhang, J. (2020). Markovian bulk-arrival and bulk-service queues with general state-dependent control. *Queueing Systems*, 95(3), 331–378.

[5] D'Arienzo, M. P., Dudin, A. N., Dudin, S. A., and Manzo, R. (2020). Analysis of a retrial queue with group service of impatient customers. *Journal of Ambient Intelligence and Humanized Computing*, 11:2591–2599.

[6] Germs, R., and Van Foreest, N. (2013). Analysis of finite-buffer state-dependent bulk queues. *OR Spectrum*, 35(3), 563–583.

[7] Ghosh, S., Banik, A. D., Walraevens, J., and Bruneel, H. (2022). A detailed note on the finite-buffer queueing system with correlated batch-arrivals and batch-size-/phase-dependent bulk-service. *4OR*, 20(2), 241–272.

[8] Goswami, V., Chaudhry, M., and Banik, A. D. (2022). Sojourn-time Distribution for $M/G^a/1$ Queue with Batch Service of Fixed Size-Revisited. *Methodology and Computing in Applied Probability*, 24(4), 2897–2912.

[9] Haridass, M., and Arumuganathan, R. (2012). A batch service queueing system with multiple vacations, setup time and server's choice of admitting reservice. *International Journal of Operational Research*, 14(2), 156–186.

[10] Jeyakumar, S., and Arumuganathan, R. (2011). A non-Markovian bulk queue with multiple vacations and control policy on request for re-service. *Quality Technology & Quantitative Management*, 8(3), 253–269.

[11] Karpagam, S. (2022). Analysis of bulk service queuing system with rework, unreliable server, resuming service and two kinds of multiple vacation. *Yugoslav Journal of Operations Research*, 33(1), 17–39.

[12] Koyama, Y., Nakamura, A., and Phung-Duc, T. (2024). Sojourn Time Analysis of a Single-Server Queue with Single-and Batch-Service Customers. *Mathematics*, 12(18), 2820.

[13] Kumar, A. (2023). Single server multiple vacation queue with discouragement solved by confluent hypergeometric function. *Journal of Ambient Intelligence and Humanized Computing*, 14(5), 6411–6422.

[14] Kumar, N. (2024). Analysis of a renewal arrival process subject to geometric catastrophe with random batch killing. *RAIRO-Operations Research*, 58(1), 933–946.

[15] Lee, H. W, Lee, S. S, Park, J. O., and Chae, K. C. (1994). Analysis of the $M^x/G/1$ queue by N-policy and multiple vacations. *Journal of Applied Probability*, 31(2), 476–496.

[16] Lee, H. W., Lee, S. S., and Chae, K. C. (1996). A fixed-size batch service queue with vacations. *International Journal of Stochastic Analysis*, 9(2), 205–219.

[17] Niranjan, S. P., and Indhira, K. (2016). A review on classical bulk arrival and batch service queueing models. *International Journal of Pure and Applied Mathematics*, 106(8), 45–51.

[18] Niranjan, S. P., Latha, S. D., Vlase, S., and Scutaru, M. L. (2024). Analysis of Bulk Queueing Model with Load Balancing and Vacation. *Axioms*, 14(1), 18.

[19] Panta, A. P., Ghimire, R. P., Panthi, D., and Pant, S. R. (2021). A review of vacation queueing models in different frameworks. *Ann. Pure Appl. Math*, 24, 99–121.

[20] Pradhan, S., and Gupta, U. C. (2019). Analysis of an infinite-buffer batch-size-dependent service queue with Markovian arrival process. *Annals of Operations Research*, 277, 161–196.

[21] Somasundaram, B., Karpagam, S., Lokesh, R., and Mary, A. K. S. (2023). An $M^{[X]}/G/1$ queue with optional service and working breakdown. *Ural Mathematical Journal*, 9(1), 162–175.

[22] Upadhyaya, S. (2016). Queueing systems with vacation: an overview. *International Journal of Mathematics in Operational Research*, 9(2), 167–213.

[23] Van Ommeren, J. K., Baer, N., Mishra, N., and Roy, D. (2020). Batch service systems with heterogeneous servers. *Queueing Systems*, 95(3), 251–269.

[24] Yuvarani, C., and Vijayalakshmi, C. (2020). Analysis of multistage $M^{[X]}/G^K/1$ queue with different server's interruptions and its application in energy consumption. *International Journal of Ambient Energy*, 41(6), 691–698.