mathematics

Article

Performance of a Synchronisation Station with Abandonment

Dieter Fiems

Department of Telecommunication Systems and Information Processing, Ghent University, St-Pietersnieuwstraat 41, 9000 Gent, Belgium; dieter.fiems@ugent.be

Abstract: The paper presents a Markovian queueing model for assessing the performance of synchronisation between stations in a production system. The system at hand consists of K distinct buffers, each buffer storing an item that is needed for the next production stage. Departures are immediate when all items are present. Due to the presence of multiple buffers, there is no reasonably fast way to calculate the stationary distribution of the Markov chain. Therefore, we focused on the series expansion of the stationary distribution in terms of the arrival rate. We provide a fast algorithm for calculating these terms. Comparing our results with stochastic simulation, we show that the expansion approach converges to the simulated values for a wide range of arrival rates.

Keywords: Markov chain; queueing system; synchronisation; Taylor series expansion

MSC: 60K25

check for updates

Citation: Fiems, D. Performance of a Synchronisation Station with Abandonment. *Mathematics* **2024**, *12*, 628. https://doi.org/10.3390/math12050628

Academic Editors: David Canca and Gabriel Villa

Received: 26 January 2024
Revised: 10 February 2024
Accepted: 18 February 2024
Published: 21 February 2024

Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The synchronisation of material flows is a key element in supply chain management and plays a pivotal role in ensuring efficiency in various industries including automotive [1] and electronics [2,3]. By coordinating different stages of a production process, resources such as machines, materials, and labour can be utilised more efficiently, which, in turn, leads to an increase in overall productivity. A synchronised production process allows for faster and more-predictable lead times. Nevertheless, due to the presence of uncertainty in production times at the different stages of the production process, some buffering may be required to streamline the process. For example, Borodin [4] and Romero-Silva and Hurtado-Hernàndez [5] note that the supply of components to assembly operations is subject to variability and uncertainty, as well as the task of assembling these components itself. In the absence of buffers, the subsequent production stages are prone to blocking, which reduces overall production efficiency. Such blocking is either caused by a lack of input materials or by the absence of storage at the output.

In this paper, we introduce a Markovian queueing model that captures the essential features and dynamics of synchronisation under uncertainty. The model at hand consists of multiple queues in parallel, each queue representing a distinct product inventory that is used to mitigate the blocking of the adjacent production stage. The different inventories are tied to distinct independent production stages, while the subsequent stage requires all items to be present. We, here, in particular, focus on the case where intermediate storage in the inventories is limited in time. The products cannot be used beyond a certain time limit, which is captured by abandonments in the different queues. Hence, the model at hand involves multiple queues with abandonments and synchronous departures from the different queues. From a mathematical perspective, the synchronisation system at hand, therefore, shares some properties with the well-investigated fork–join queues [6–9], as well as with queues with paired service [10–12]. In a fork–join queue, a job is split up between servers (forking) and, then, again, combined in a later stage (joining). Hence, in a fork–join queue, there is synchronisation between arrivals, which gives rise to a different type of dynamic. Queues with paired service, also referred to as assembly-like queues or kitting processes, also assume that the arrival processes are independent. However, departures are not immediate, as pairing requires service.

The synchronisation model at hand with only two buffers further corresponds to a so-called double-ended queue [13,14]. As service is immediate, one of the queues is empty, and the synchronisation station can be modelled by an integer-valued Markov chain. Positive states correspond to items in one queue, while negative states correspond to items in the other queue. Such double-ended queues are, e.g., motivated by taxi stands, where a trip starts once the taxi is present, as well as its passengers.

The main contribution of the present paper involves extending the analysis of the double-ended queue to Markovian models for the synchronisation of multiple streams. When the number of streams is limited and the provided buffer capacity per stream is limited, the corresponding Markov chain has a limited number of states, and the steady-state solution of the Markov chain can be readily obtained by solving the linear system of balance equations. As the state space of the synchronisation station is multidimensional, the model suffers from the so-called curse of dimensionality: the state space of the Markov chain grows exponentially in the number of dimensions, which makes the computational cost of directly solving the balance equations prohibitive for practical applications. A second contribution, therefore, lies in devising an efficient algorithm for numerically solving the Markovian synchronisation model. We, here, rely on a series expansion approach. Instead of calculating the stationary transition probabilities, the series expansion approach studies the stationary transition probabilities as functions of a system parameter (the arrival rate λ in the present setting) and calculates the terms in the series expansion of these functions. At first sight, this does not mitigate the computational burden of the large state space, as there are as many equations for each term in the series expansions as there are equations in the balance equations. However, if the chain at $\lambda = 0$ only allows for transitions in one direction (either upwards or downwards), the transition matrix is a triangular matrix, and the computational complexity for a chain with M states is reduced to at most $O(M^2)$. Moreover, in many practical applications, the number of possible transitions from a state is much smaller than the size of the state space, so that the numerical complexity for the calculation of each term is $O(M)$ and the complexity for the calculation of the first N terms of the expansion is only $O(MN)$.

The presented series expansion technique is known by various terms in the literature, including Markov chain perturbation, light-traffic approximation, and the power series algorithm. We refer to the surveys on series expansion techniques [15,16] and the references therein. Despite the lack of a fixed nomenclature, perturbation methods primarily originate from the necessity to analyse how a system's performance is affected by changes in specific parameters. Singular perturbations, which disrupt the class-based structure of the unperturbed chain, have been a significant focus in research, as seen in works such as [17–19]. When the perturbed parameter relates to the arrival rate of a queue, the series expansion is a light-traffic approximation. Lastly, the power series algorithm transforms a Markov chain of interest into a collection of Markov chains indexed by a potentially artificially introduced variable ρ. This transformation is chosen to allow for a straightforward computation of perturbations around $\rho = 0$, while as ρ approaches 1, the original Markov chain is regained. Hence, the series expansion can be used for approximating solutions of the original Markov chain. This holds true if the radius of convergence of the series expansion exceeds $\rho = 1$ [20–23]. The methodology in the present setting is a light-traffic approximation, though the numerical results reveal that the results are accurate for a wide range of arrival rates. The approach is primarily algorithmic, with a focus on a recursion for calculating many terms in the series expansion, rather than on closed-form expressions for the first few terms. Such an algorithmic approach previously allowed for analysing queueing systems with coupled service [11,12], retrial queues with heterogeneous retrials [24], and epidemic processes in finite populations [25]. In order to improve convergence, we further rely on convergence acceleration techniques. Such techniques not only accelerate the convergence in the region of convergence, but also extend the region of convergence [26–28]. We, here, in particular, apply Wynn's ϵ-algorithm to improve convergence. Wynn's ϵ-algorithm is an efficient implementation of Shank's transformation, which is the best all-purpose acceleration method according to [29]. Under non-stringent assumptions on the sequences, the transformation converges faster to the limit of the original series in the region of convergence and converges to the analytic continuation of diverging sequences [30].

The remainder of this paper is organised as follows. In the next section, we introduce the Markovian modelling assumptions and find the corresponding set of balance equations. We study the synchronisation station in the "light-traffic" regime in Section 3. In contrast to a direct solution, we show that the computational cost of calculating the terms in the light traffic approximation is considerably lower. We, then, extend the modelling assumptions by the inclusion of a Markovian environment variable, which modulates the arrival and abandonment rates in Section 4. We illustrate our approach by a set of numerical experiments in Section 5. Finally, the conclusions are drawn in Section 6.

2. Mathematical Model

We considered a synchronisation station with N finite or infinite capacity queues, each queue storing a particular part. Parts arrive at the nth queue in accordance with a Poisson process with rate $\lambda_n > 0$, $n \in \{1, \dots, N\}$. Let $C_n \in \mathbb{N} \setminus \{0\} \cup \{\infty\}$ denote the (possibly infinite) capacity of the nth queue. Whenever all queues are non-empty, a container with all parts immediately leaves the synchronisation station. Note that, at any point in time, at least one of the queues will be empty. If this is not the case, a container is ready and departs immediately. Apart from container departures, parts may also leave prior to being packed in a container. To be more precise, each part has a finite maximal residence time, which is exponentially distributed. Let $1/\alpha_n$ denote the average maximal residence time of parts in the nth queue. In other words, each part in the nth queue abandons the queue with rate $\alpha_n > 0$.

Considering the modelling assumptions outlined above, the Markovian state of the synchronisation station is completely determined by the number of parts in each queue. Let $X_n(t) \in \{0, \dots, C_k\} \doteq \mathcal{C}_k$ denote the number of parts in the nth queue at time t, and let $\boldsymbol{X}(t) = [X_1(t), \dots, X_N(t)]$ denote the vector of queue contents at time t. As one of the queues is empty for sure, the state space of the Markov process $\boldsymbol{X}(t)$ is given by

$$\mathcal{C} = (\mathcal{C}_1 \times \dots C_n) \setminus (\mathcal{C}_1^0 \times \dots \times \mathcal{C}_N^0)\,,$$

with $\mathcal{C}_k^0 = \mathcal{C}_k \setminus \{0\} = \{1, \dots, C_k\}$. Moreover, it is easy to verify that this Markov process is ergodic. First, the Markov chain is irreducible, as state $\boldsymbol{X} = \boldsymbol{0}$ can be reached by abandonments from any state and any state can be reached from state $\boldsymbol{0}$ by arrivals. Secondly, comparing the system at hand with the (ergodic) system without container departures ensures that state zero is recurrent. Indeed, in the absence of container departures, departures in the queues stem from abandonments, and each queue can be modelled by an infinite server queue.

We focus on the stationary distribution of the Markov chain. For $\boldsymbol{x} \in \mathcal{C}$, let $\pi(\boldsymbol{x})$ denote the stationary probability to be in state $\boldsymbol{x} = [x_1, x_2, \dots, x_N]$:

$$\pi(\boldsymbol{x}) = \lim_{t \to \infty} \mathsf{P}[\boldsymbol{X}(t) = \boldsymbol{x}]\,.$$

Moreover, to simplify the notation, it is convenient to set $\pi(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \neq \mathcal{C}$. State transitions are either triggered by part abandonments or by new part arrivals. In state $\boldsymbol{x}$, parts in the nth queue are abandoned with rate $x_n \alpha_n$, the new state being $\boldsymbol{x} - \boldsymbol{e}_n$. Here, $\boldsymbol{e}_n$ is a vector of all zeros, apart from its nth element, which is one. For $x_n < C_n$, a new part arrives at the nth queue with rate λ_n. If $\boldsymbol{x}_n + \boldsymbol{e}_n \in \mathcal{C}$, the arrival does not complete a container and the state after the arrival is $\boldsymbol{x}_n + \boldsymbol{e}_n$. If $\boldsymbol{x}_n + \boldsymbol{e}_n \notin \mathcal{C}$, the new arrival completes a container, such that there is an immediate departure of this container. Therefore, the state after the arrival is $\boldsymbol{x}_n + \boldsymbol{e}_n - \boldsymbol{e}$, where $\boldsymbol{e}$ is a vector of ones. In view of these observations, we find the following set of balance equations:

$$\pi(\boldsymbol{x}) \sum_{n=1}^{N} (\lambda_n 1_{\{x_n < C_n\}} + \alpha_n x_n) = \sum_{n=1}^{N} \pi(\boldsymbol{x} + \boldsymbol{e}_n)\alpha_n (x_n + 1) + \pi(\boldsymbol{x} - \boldsymbol{e}_n)\lambda_n + \pi(\boldsymbol{x} - \boldsymbol{e}_n + \boldsymbol{e})\lambda_n \,, \quad (1)$$

for $\boldsymbol{x} \in \mathcal{C}$. Here, $1_{\{\cdot\}}$ is the indicator function, which evaluates to 1 if its argument is true and to 0 if this is not the case. Note that the expression above holds both for finite and infinite C_n. For infinite C_n, the condition $x_n < C_n$ trivially holds for all x_n. Also, the assumption $\pi(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \notin \mathcal{C}$ greatly simplifies the balance equations as we can omit the indicators that the states $\boldsymbol{x} + \boldsymbol{e}_n$, $\boldsymbol{x} - \boldsymbol{e}_n$ and $\boldsymbol{x} - \boldsymbol{e}_n + \boldsymbol{e}$ are part of the state space.

3. Light-Traffic Analysis

For small N and C_n, the balance Equation (1) is readily solved by direct methods like, e.g., Gaussian elimination. However, already for moderate N and C_n, the size of the state space of the Markov process makes a direct calculation of the stationary probabilities computationally demanding. Therefore, we focus on the synchronisation system in the light-traffic regime. Our approach can approximate the exact distribution with high accuracy and low computational effort, whereby the computational effort for the calculation of the Tth term in the series expansion is $O(N \min(T, C)^N)$ with $C = \max_k C_k$, cf. infra.

To this end, we set $\lambda_n = \kappa_n \lambda$ and introduce the Maclaurin series expansion in λ of the solution of the balance equations:

$$\pi(\boldsymbol{x}) = \sum_{k=0}^{\infty} \pi_k(\boldsymbol{x})\lambda^k \,.$$

Before proceeding, we note that, for finite C_k, it is easy to check—by applying Cramer's rule, that one explicitly expresses $\pi(\boldsymbol{x})$ as a fraction of two determinants; that the stationary probabilities are rational functions of λ. Since rational functions have no singularities other than poles in the extended complex plane and there is no pole at $\lambda = 0$, we conclude that the stationary probabilities are analytic functions in a region around $\lambda = 0$. This, then, justifies the series expansion. However, analyticity is not necessarily required. In the following numerical examples, we rely on the ϵ method for convergence acceleration. This method is known to extend the region of convergence and even works when the radius of convergence is 0 [30].

Substituting the series expansions above in the balance equations and isolating the terms in λ^k leads to

$$\pi_k(\boldsymbol{x}) = \left(\sum_{n=1}^{N} \alpha_n x_n\right)^{-1} \Bigg(- \pi_{k-1}(\boldsymbol{x}) \sum_{n=1}^{N} \kappa_n 1_{\{x_n < C\}} + \sum_{n=1}^{N} \pi_k(\boldsymbol{x} + \boldsymbol{e}_n)\alpha_n (x_n + 1) + \pi_{k-1}(\boldsymbol{x} - \boldsymbol{e}_n)\kappa_n + \pi_{k-1}(\boldsymbol{x} - \boldsymbol{e}_n + \boldsymbol{e})\kappa_n \Bigg) \,, \quad (2)$$

for $k \in \{1, 2, \ldots\}$ and for $\boldsymbol{x} \in \mathcal{C} \setminus \{\boldsymbol{0}\}$. For $\boldsymbol{x} = \boldsymbol{0}$, the normalisation condition:

$$\sum_{\boldsymbol{x} \in \mathcal{C}} \pi(\boldsymbol{x}) = \sum_{\boldsymbol{x} \in \mathcal{C}} \sum_{k=0}^{\infty} \pi_k(\boldsymbol{x})\lambda^k = 1$$

yields

$$\pi_k(\boldsymbol{0}) = - \sum_{\boldsymbol{x} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \pi_k(\boldsymbol{x}) \,, \quad (3)$$

$k \in \{1, 2, \ldots\}$. Moreover, for $k = 0$, one easily finds

$$\pi_0(\boldsymbol{x}) \ = 1_{\{\boldsymbol{x}=\boldsymbol{0}\}}\,.$$

The latter result is also intuitively clear. In the absence of part arrivals, all part queues are empty.

In contrast to the balance Equation (1), the system of Equations (2) and (3) is easily solved. Indeed, close inspection of Equations (2) and (3) reveals that one can solve these equations in reverse lexicographical order: to calculate $\pi_k(\boldsymbol{x})$ by (2) and (3), one only needs $(k-1)$ the-order terms and terms in $\pi_k(\boldsymbol{y})$ for $\boldsymbol{y}$ lexicographically larger than $\boldsymbol{x}$. In addition, by the so-called k-events rule, one can show that $\pi_k(\boldsymbol{x}) = 0$ if the state $\boldsymbol{x}$ cannot be reached from state $\boldsymbol{0}$ with k arrivals (and possibly any number of other transitions). In this case, we have $\pi_k(\boldsymbol{x}) = 0$ if the total queue size exceeds k: $\sum_n x_n > k$. Summarising, we find that one needs to calculate at most $O(\min(T, C)^N)$ unknowns for the Tth-order terms, each term requiring $O(N)$ additions. Hence, the calculation of the Tth-order term is of order $O(N \min(T, C)^N)$. Note that the factor $\min(T, C)$ follows from the n-events rule used above. We have $\pi_T(\boldsymbol{x}) = 0$ and state that it cannot be reached from the empty queueing state with T arrivals.

Once the terms in the series expansions of the stationary probabilities have been calculated, various performance measures of interest can be studied. The Kth order expansion of the ℓth moment of the nth queue equals

$$\mathsf{E}[X_n^\ell] \approx \sum_{k=0}^{K} \left(\sum_{\boldsymbol{x}\in\mathcal{C}} \pi_k(\boldsymbol{x}) x_n^\ell \right) \lambda^k\,. \tag{4}$$

Similar expressions are readily obtained for the cross-moments of the different queue content as well.

The Kth-order expansion of the abandonment rate ν_n and loss probability η_n in the nth queue further equal

$$\nu_n \approx \sum_{k=0}^{K} \left(\sum_{\boldsymbol{x}\in\mathcal{C}} \pi_k(\boldsymbol{x}) x_n \alpha_n \right) \lambda^k\,, \quad \eta_n \approx \sum_{k=0}^{K} \left(\sum_{\boldsymbol{x}\in\mathcal{C}} \pi_k(\boldsymbol{x}) 1_{\{x_n=C_n\}} \right) \lambda^k\,.$$

Indeed, each of the x_n parts in the nth queue is abandoned with rate α_n, while there is a loss if there is an arrival in a full queue. Finally, the Kth-order expansion of the (simultaneous) departure rate μ from the queues equals

$$\mu \approx \sum_{k=0}^{K} \left(\sum_{\boldsymbol{x}\in\mathcal{C}} \pi_k(\boldsymbol{x}) \sum_{n=1}^{N} \kappa_n 1_{\{x_n=0, x_m>0, \forall m\neq n\}} \right) \lambda^{k+1}\,.$$

Here, we used the observation that we have a departure if there is an arrival in an empty queue, and the other queues or not empty.

Remark 1. *The expressions of the performance measures show that the terms in the series expansions of the performance measures can be calculated once the corresponding terms of the stationary probabilities are obtained. This observation allows for a considerable reduction of the memory requirements of the algorithm. Indeed, while close inspection of* (2) *shows that the nth terms need to be preserved during the calculation of the* $(n+1)$*th terms, once these calculations are complete and once the nth terms of the performance measures are calculated, the nth terms are no longer required, and the storage can be freed or overwritten. Summarising, the storage requirement is* $O(N \min(T, C)^N)$*, which only depends on the number of terms by the truncation of the state space, which follows from the n-events rule (cf. supra).*

4. Markovian Environment

We now extend our findings to a synchronisation station, where the arrival rates in the different queues, as well as the abandonment rates from the different queues depend on an exogenous Markov process with a finite state space. More precisely, let $Y(t)$ be an ergodic Markov process with finite state space $\mathcal{M}$ and with generator matrix $B = [\beta_{ij}]_{i,j\in\mathcal{M}}$. Here, β_{ij} denotes the transition rate from state i to state j ($i, j \in \mathcal{M}$ with $i \neq j$), while the diagonal elements are equal to

$$\beta_{ii} \doteq - \sum_{j\in\mathcal{M}\setminus\{i\}} \beta_{ij} \,,$$

as usual (for $i \in \mathcal{M}$). For further use, let $\boldsymbol{\nu}$ denote the invariant distribution of this Markov process. That is, $\boldsymbol{\nu}$ solves $\boldsymbol{\nu} B = 0$ and $\boldsymbol{\nu}\boldsymbol{e}' = 1$ or, equivalently,

$$\boldsymbol{\nu} = (B + \boldsymbol{e}'\boldsymbol{e})^{-1}\boldsymbol{e} \,.$$

When the exogenous state is $j \in \mathcal{M}$, new parts arrive at the nth queue with rate $\lambda_n^j > 0$, while parts in the nth queue are abandoned with rate $\alpha_n^j > 0$. Retaining the notation of the preceding section, it is now easily verified that the process $(\boldsymbol{X}(t), Y(t))$ constitutes an ergodic Markov process with state space $\mathcal{C} \times \mathcal{M}$. For $(\boldsymbol{x}, j) \in \mathcal{C} \times \mathcal{M}$, let

$$\pi(\boldsymbol{x}, j) = \lim_{t\to\infty} \mathsf{P}[\boldsymbol{X}(t) = \boldsymbol{x}, Y(t) = j]$$

denote the limiting distribution of the Markov process.

Remark 2. *For the ease of exposition, we assumed positive arrival and abandonment rates for all states of the environment. This is not strictly required. In the finite setting, a positive arrival and abandonment rate in one of the states suffices. In the infinite setting, it suffices to check that each queue is ergodic in the absence of container departures.*

Assuming $\pi(\boldsymbol{x}, j) = 0$ for $(\boldsymbol{x}, j) \notin \mathcal{C} \times \mathcal{M}$, the balance equations then read

$$\pi(\boldsymbol{x}, j)\left(\sum_{n=1}^{N}(\lambda_n^j 1_{\{x_n<C_n\}} + \alpha_n^j x_n) + \sum_{k\in\mathcal{M}\setminus\{j\}} \beta_{jk}\right) = \sum_{i\in\mathcal{M}\setminus\{j\}} \pi(\boldsymbol{x}, i)\beta_{ij}$$
$$+ \sum_{n=1}^{N}\Big(\pi(\boldsymbol{x} + \boldsymbol{e}_n, j)\alpha_n^j (x_n + 1) + \pi(\boldsymbol{x} - \boldsymbol{e}_n, j)\lambda_n^j + \pi(\boldsymbol{x} - \boldsymbol{e}_n + \boldsymbol{e}, j)\lambda_n^j\Big) \,, \quad (5)$$

for $\boldsymbol{x} \in \mathcal{C}$ and $j \in \mathcal{M}$. For convenience, let $\boldsymbol{\pi}(\boldsymbol{x}) = [\pi(\boldsymbol{x}, j)]_{j\in\mathcal{M}}$ denote the row vector that collects the stationary probabilities for a fixed $\boldsymbol{x} \in \mathcal{C}$. Moreover, let A_n and Λ_n denote diagonal matrices with jth diagonal entry α_n^j and λ_n^j, respectively. The balance equations can then be rewritten as follows

$$\boldsymbol{\pi}(\boldsymbol{x}) \sum_{n=1}^{N} (\Lambda_n 1_{\{x_n<C_n\}} + A_n x_n)$$
$$= \boldsymbol{\pi}(\boldsymbol{x})B + \sum_{n=1}^{N} (\boldsymbol{\pi}(\boldsymbol{x} + \boldsymbol{e}_n)A_n(x_n + 1) + \boldsymbol{\pi}(\boldsymbol{x} - \boldsymbol{e}_n)\Lambda_n + \boldsymbol{\pi}(\boldsymbol{x} - \boldsymbol{e}_n + \boldsymbol{e})\Lambda_n) \,. \quad (6)$$

We now express Λ_k in terms of a single rate λ, $\Lambda_k = \lambda K_k$, and as for the synchronisation station without an exogenous environment, we introduce the Maclaurin series expansion in λ of the solution of the balance equations:

$$\boldsymbol{\pi}(\boldsymbol{x}) = \sum_{k=0}^{\infty} \boldsymbol{\pi}_k(\boldsymbol{x})\lambda^k \,.$$

Plugging the series expansion above in (4) and comparing terms in equal powers of λ then yield

$$\boldsymbol{\pi}_0(\boldsymbol{x}) = \mathbf{0}\,, \tag{7}$$

and

$$\begin{aligned}\boldsymbol{\pi}_k(\boldsymbol{x}) = \Bigg(&- \boldsymbol{\pi}_{k-1}(\boldsymbol{x}) \sum_{n=1}^{N} K_n 1_{\{x_n<C_n\}} + \sum_{n=1}^{N} \boldsymbol{\pi}_k(\boldsymbol{x}+\boldsymbol{e}_n) A_n (x_n+1) \\ &+ \sum_{n=1}^{N} \boldsymbol{\pi}_{k-1}(\boldsymbol{x}-\boldsymbol{e}_n) K_n + \boldsymbol{\pi}_{k-1}(\boldsymbol{x}-\boldsymbol{e}_n+\boldsymbol{e}) K_n \Bigg) \left(\sum_{n=1}^{N} A_n x_n - B \right)^{-1}, \end{aligned} \tag{8}$$

for $k = 1, 2, \ldots$ and $\boldsymbol{x} \in \mathcal{C} \setminus \{\mathbf{0}\}$. Note that Equation (7) simply states that there are no parts in any of the queues in absence of arrivals.

For $\boldsymbol{x} = \mathbf{0}$, we further find

$$\begin{aligned} &\boldsymbol{\pi}_0(\mathbf{0}) B = \mathbf{0}\,, \\ &\boldsymbol{\pi}_k(\mathbf{0}) B = \boldsymbol{\pi}_{k-1}(\mathbf{0}) \sum_{n=1}^{N} K_n - \sum_{n=1}^{N} \left(\boldsymbol{\pi}_k(\boldsymbol{e}_n) A_n + \boldsymbol{\pi}_{k-1}(\boldsymbol{e}-\boldsymbol{e}_n) K_n \right). \end{aligned} \tag{9}$$

Combining these expressions with the normalisation conditions:

$$\boldsymbol{\pi}_0(\mathbf{0}) \boldsymbol{e}' = 1\,, \quad \sum_{\boldsymbol{x} \in \mathcal{C}} \boldsymbol{\pi}_k(\boldsymbol{x}) \boldsymbol{e}' = 0\,,$$

then yields $\boldsymbol{\pi}_0(\mathbf{0}) = \boldsymbol{\nu}$ and

$$\boldsymbol{\pi}_k(\mathbf{0}) = \left(\boldsymbol{\pi}_{k-1}(\mathbf{0}) \sum_{n=1}^{N} K_n - \sum_{n=1}^{N} \left(\boldsymbol{\pi}_k(\boldsymbol{e}_n) A_n + \boldsymbol{\pi}_{k-1}(\boldsymbol{e}-\boldsymbol{e}_n) K_n \right) + \kappa \boldsymbol{e} \right) (B + \boldsymbol{e}'\boldsymbol{e})^{-1},$$

with

$$\kappa = \boldsymbol{\pi}_k(\mathbf{0}) \boldsymbol{e}' = - \sum_{\boldsymbol{x} \in \mathcal{C} \setminus \{\mathbf{0}\}} \boldsymbol{\pi}_k(\boldsymbol{x}) \boldsymbol{e}'\,.$$

Note that we cannot simply obtain $\boldsymbol{\pi}_k(\mathbf{0})$ from Equation (9) as B is not invertible.

Again, the terms in the series expansion can be easily determined in reverse lexicographical order, where, by the k-events rule, one can show that $\boldsymbol{\pi}_k(\boldsymbol{x}) = \mathbf{0}$ if the state $\boldsymbol{x}$ cannot be reached from state $\mathbf{0}$ with k arrivals (and possibly any number of other transitions). The scalar operations of the preceding section are now replaced by matrix operations. Hence, following the same reasoning as in the preceding section, we find that the numerical complexity for calculating the Tth order term is $O(N \min(T, C) M^3)$, where M denotes the number of environment states. Once the series expansion of the stationary vectors $\boldsymbol{\pi}(\boldsymbol{x})$ is obtained, series expansions for various performance measures of interest are readily obtained. The expression (4) for the ℓth moment of the queue size remains valid, with $\pi(\boldsymbol{x}) = \sum_{j \in \mathcal{M}} \pi(\boldsymbol{x}, j)$. The Kth order expansion of the departure rate now reads

$$\mu \approx \sum_{k=0}^{K} \left(\sum_{\boldsymbol{x} \in \mathcal{C}} \sum_{j \in \mathcal{M}} \pi_k(\boldsymbol{x}) \sum_{n=1}^{N} \kappa_n^j 1_{\{x_n=0, x_m>0, \forall m \neq n\}} \right) \lambda^{k+1}\,.$$

5. Numerical Examples

We now illustrate our numerical approach by a set of examples. To assess the accuracy of the series approximations, we compare the results that were obtained by the analytic approach with stochastic simulation. We first demonstrate the accuracy for the synchronisation system without the random environment and, then, extend our examples by including an environment variable that allows for including burstiness in the arrival processes of one of the queues.

Preliminary experimentation showed that the series expansions only converge to the correct value for small λ values, which is an indication that the region of convergence of the series expansion is limited. Luckily, convergence acceleration techniques allow for extending the region of convergence [26–28]. We, here, implement Wynn's ϵ-algorithm to speed up convergence, as well as to extend the region of convergence. To this end, for a performance measure of interest, introduce the sequence of partial sums $S_n = \sum_{k=0}^{n} s_k \lambda^k$. Here, the sequence s_k corresponds to the coefficients in the expansion of the performance measure. Expressions for these terms are given in Section 3 and are readily calculated once the terms in the expansions of the stationary probabilities are determined. The ϵ-algorithm then approximates the limit $n \to \infty$ by $\epsilon_k^{(0)}$, for k even (odd indices are intermediate values), recursively defined as follows:

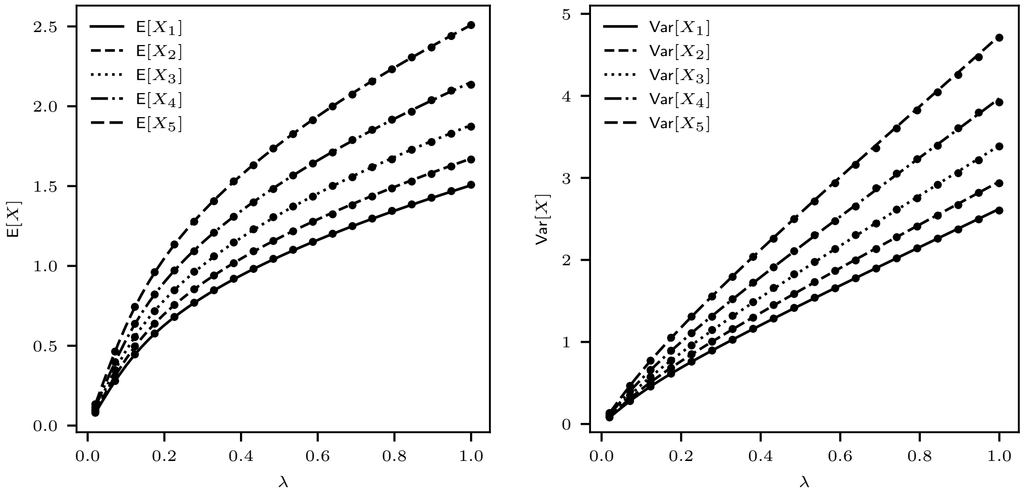
$$\epsilon_{-1}^{(n)} = 0\,, \quad \epsilon_0^{(n)} = S_n\,,$$
$$\epsilon_{r+1}^{(m)} = \epsilon_{r-1}^{(m+1)} + \frac{1}{\epsilon_r^{(m+1)} - \epsilon_r^{(m)}}\,,$$

for $k, n \in \{0, 1, 2, \ldots\}$. Our implementation includes a simple protection against division by zero, as well as the simple stopping rule of [31] (p. 213).

5.1. Basic Model

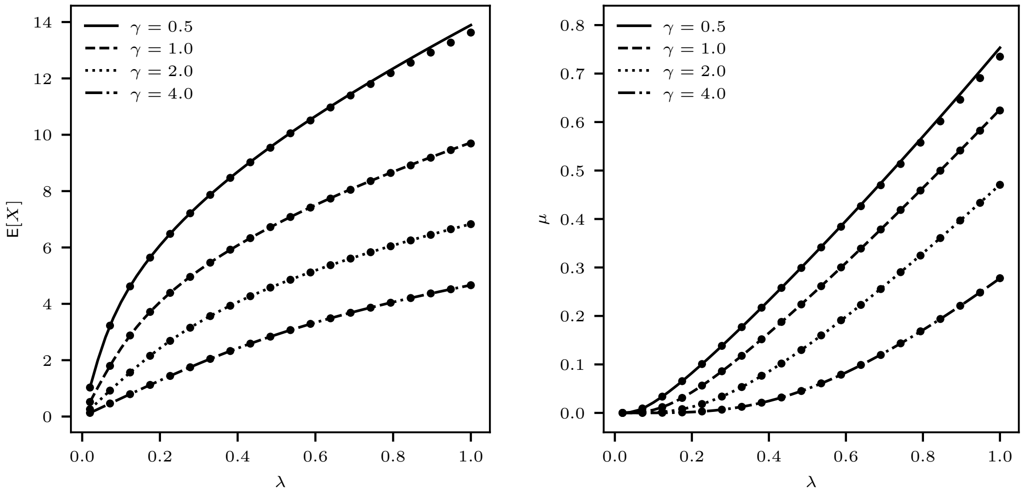
We first consider a synchronisation station with five queues, with equal arrival rates λ in all queues and with capacities $C_1 = C_2 = 10$, $C_3 = 15$, $C_4 = 20$, and $C_5 = 25$. The abandonment rates in the different queues equal $\alpha_1 = 0.25\gamma$, $\alpha_2 = 0.225\gamma$, $\alpha_3 = 0.2\gamma$, $\alpha_4 = 0.175\gamma$, and $\alpha_5 = 0.15\gamma$. Here, γ is a scale factor for the abandonment rates. This factor can be used to study how the abandonment rates affect performance, with higher γ corresponding to more abandonments. With these queue capacities, there are approximately 10^6 states. Figure 1a,b depict the mean and variance of the queue sizes versus the arrival rate λ, respectively. The scale parameter is fixed to $\gamma = 1$ for all curves. The different curves are calculated by applying the ϵ-algorithm on the 30th-order series expansion. Not all terms in the series expansion are necessarily used though, the expansion order being an upper bound for the number of terms in the stopping rule of [31]. The markers were obtained by means of stochastic simulation and are included to verify the accuracy of the series expansion approach. The simulation results were obtained by means of the replication–deletion approach. We simulated 100 replications of 5×10^6 state transitions. The obtained 99% confidence intervals are smaller than 1% of the simulated value and are, therefore, omitted from the plots. The figures show that the correspondence between the simulation results and the series expansion approach is excellent for the complete range of the λ-values. We would like to point out that the ϵ-algorithm is key for the accuracy of the results. Without convergence acceleration, the region of convergence of the series expansion is limited, and the results are only accurate for very small arrival rates ($\lambda < 0.05$). We excluded the curves without acceleration from the figures to enhance clarity.

To evaluate the effect of the abandonment rates on performance, Figure 2 depicts the mean total queue content and the departure rate from the queues vs. the arrival rate λ for different values of the scale factor γ, as indicated. Again, values from the (30th-order) series expansion approach were compared with the equivalent values obtained by stochastic simulation. As in Figure 1, the simulation results were obtained by means of the replication–deletion approach with 100 replications of 5×10^6 state transitions. The simulation study showed that the combination of the series expansion approach and the ϵ-algorithm leads to very accurate results. More arrivals translate into larger queue sizes, while more abandonments imply lower queue sizes. Analogously, the departure rate increases with the arrival rate and decreases with the abandonment rate. Recall that the departure rate only includes departures by synchronisation and excludes departures by abandonments. Ideally, the departure rate should equal the arrival rate, so that all arrivals can leave the queues by synchronisation. There are two impediments to such a perfect operation of the synchronisation station. First, some part arrivals may not enter the synchronisation station due to a lack of queue capacity. Secondly, parts may be abandoned, prior to synchronisation. The latter effect is dominant in the present example, especially for higher abandonment rates (that is, for higher γ).



(**a**) Mean queue size (**b**) Variance of the queue size

Figure 1. Mean (**a**) and variance (**b**) of the queue sizes vs. the arrival rate λ. The curves follow from the series expansion, and the markers were obtained by stochastic simulation.



(**a**) Mean total queue size (**b**) Departure rate

Figure 2. Total queue size $\mathsf{E}[X]$ (**a**) and departure rate μ (**b**) vs. the arrival rate λ for different values of the scale factor γ of the abandonment rates. The curves follow from the series expansion, and the markers were obtained by stochastic simulation.

5.2. *Markovian Environment*

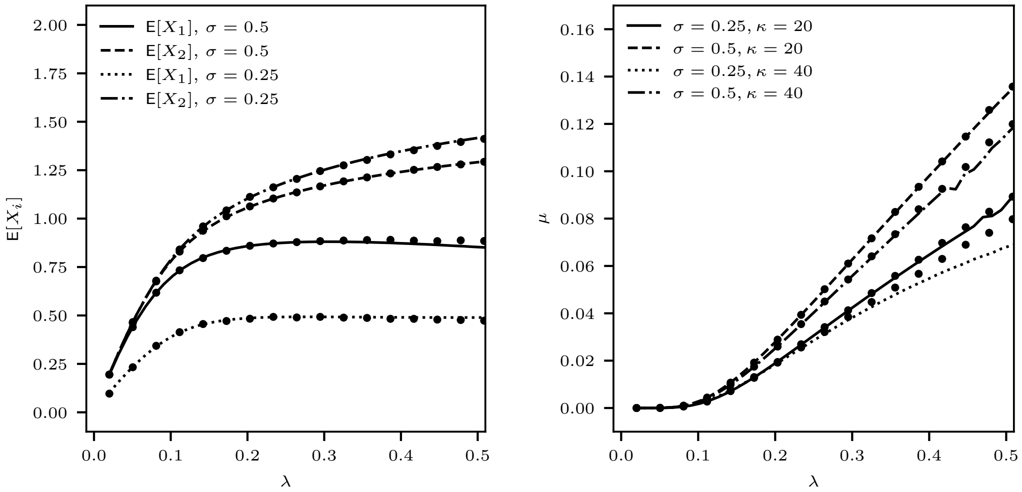
So far, the queueing model did not include the Markovian environment variable of Section 4. We now consider a model with 10 queues, all with capacity $C_n = 2$ and all with abandonment rate $\alpha_n = 0.1$, $n \in \{1, \ldots, N\}$. The arrivals in all but the first queue stem from independent Poisson processes with common rate $\lambda_n = 1$. In contrast, the part arrivals in the first queue stem from an interrupted Poisson process. More precisely, we introduce a random environment with two states, say on and off, and there are only arrivals in the on state. Let a denote the transition rate from the on to the off state, and let b denote the transition rate from the off state to the on state. The fraction of time σ the Markov environment is in the on state and the length κ of an on and off period then equal

$$\sigma = \frac{b}{a+b}, \quad \kappa = \frac{1}{a} + \frac{1}{b}.$$

The pair (σ, κ) uniquely determines the transition rates a and b. We further assume that the (Poisson) arrival rate in the on state equals $\hat{\lambda}_1 = 1/\sigma$, such that the average arrival rate in the first queue is 1 as well. One expects that smaller σ and larger κ affect performance negatively. For smaller σ, the arrivals are more concentrated, while for larger κ, the periods with and without arrivals are longer. This means that the queue can build up for a longer time.

Figure 3a shows the mean queue size as a function of the arrival rate for different values of σ and fixed $\kappa = 20$. Only the mean queue size of the first two queues is shown. Due to symmetry, the mean queue size of the remaining queues is identical to the mean queue size of the second queue. The curves were obtained by the 80th-order series expansion approach, complemented with the ϵ-algorithm. The markers again represent stochastic simulation results. The simulation results were obtained by means of the replication–deletion approach with 100 replications of 10^9 state transitions. The upper and lower bounds of the obtained 99% confidence intervals are visually indiscernible and, therefore, omitted. The correspondence between the analytic and simulation results is again good, though, by the additional complexity of the environment, a higher order expansion is needed to obtain accurate results. Recall that the arrivals in queue 1 are more concentrated in time for lower values of σ. This leads to a lower mean queue size for the first queue and a higher mean queue size for the second queue. The concentration of arrivals leads to considerable loss in the first queue, followed by periods without arrivals. The loss explains why there are fewer parts in this queue on average. On the other hand, the absence of parts in the first queue implies that there are fewer synchronised departures, which, in turn, means that there are more parts in the other queues. This latter observation is confirmed by Figure 3b, which shows the departure rate μ for different κ and σ. Both an increase of κ and a decrease σ lead to a decrease of the departure rate. Also, Figure 3b compares the analytic results (80th-order expansion) with the simulation results. In contrast to the preceding figures, the accuracy is not as good, especially for $\kappa = 40$ and $\sigma = 0.25$, which is the example where the arrivals are concentrated over long periods, followed by long periods without arrivals.

To conclude this numerical investigation, we briefly summarise some observations on the applicability and accuracy of the series expansion approach. Foremost, we found that the method works well for a large number of parameter sets, though the number of terms in the expansion that is required to obtain accurate results does depend on the specific parameter values. In general, there is no direct relation between the size of the state space and the required order. We, however, observed that the presence of the Markovian environment most often required higher-order expansions. As the calculation of the series expansion is linear in the order of the expansion, computational limitations did not prevent the use of higher-order expansions. However, we found that Wynn's algorithm may encounter numerical instability when dealing with higher-order series expansions for specific λ-values, so there is a practical limit on the order.



(**a**) Mean queue size (**b**) Departure rate

Figure 3. Mean queue size (**a**) and departure rate (**b**) vs. the arrival rate λ for different environment parameters σ and κ, as indicated. The curves follow from the series expansion, and the markers were obtained by stochastic simulation.

6. Conclusions

In this study, we introduced a Markovian queueing model to evaluate the performance of synchronisation among stations within a complex production system featuring K distinct buffers. These buffers store parts necessary for advancing to the subsequent production stages, and departures occur instantly upon the availability of all required items. The intricacies arising from multiple buffers render the calculation of the stationary distribution of the Markov chain a challenging task. To address this challenge, we applied a series expansion approach where the arrival rate was used as the parameter of the expansion.

Our primary contribution lies in the development of a rapid algorithm tailored to the computation of terms within the series expansion. By employing this algorithm, we mitigated the computational challenges associated with the calculation of the stationary distribution. Our numerical experiments confirmed that the proposed algorithm enables efficient and accurate computation of the stationary distribution. By comparing with stochastic simulation results, the accuracy was observed consistently across a broad spectrum of arrival rates, highlighting the robustness and versatility of our methodology in capturing the dynamic behaviour of the production system.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boudella, M.; Sahin, E.; Dallery, Y. Kitting Optimisation in Just-in-Time Mixed-Model Assembly Lines: Assigning Parts to Pickers in a Hybrid Robot–Operator Kitting System. *Int. J. Prod. Res.* **2018**, *56*, 5475–5494. [CrossRef]
2. Polat, O.; Kalayci, C.B.; Mutlu, O.; Gupta, S.M. A Two-Phase Variable Neighbourhood Search Algorithm for Assembly Line Worker Assignment and Balancing Problem Type-II: An Industrial Case Study. *Int. J. Prod. Res.* **2016**, *54*, 722–741. [CrossRef]
3. Guo, J.; Pu, Z.; Du, B.; Li, Y. Multi-Objective Optimisation of Stochastic Hybrid Production Line Balancing Including Assembly and Disassembly Tasks. *Int. J. Prod. Res.* **2021**, *60*, 2884–2900. [CrossRef]
4. Borodin, V.; Dolgui, A.; Hnaien, F.; Labadie, N. Component Replenishment Planning for a Single-Level Assembly System Under Random Lead Times: A Chance Constrained Programming Approach. *Int. J. Prod. Econ.* **2016**, *181*, 79–86. [CrossRef]
5. Romero-Silva, R.; Hurtado-Hernández, M. The effects of supply variability on the performance of assembly systems. *Int. J. Prod. Res.* **2023**, *61*, 4973–4990. [CrossRef]
6. Krishnamurthy, A.; Suri, R.; Vernon, M. Analysis of a fork/join synchronization station with inputs from coxian servers in a closed queuing network. *Ann. Oper. Res.* **2004**, *125*, 69–94. [CrossRef]

7. Sonmez, E.; Scheller-Wolf, A.; Secomandi, N. An Analytical Throughput Approximation for Closed Fork/Join Networks. *Informs J. Comput.* **2017**, *29*, 251–267. [CrossRef]
8. Goossens, N.; Krishnamurthy, A.; Vandaele, N. Analysis of a fork/join station with inputs from a finite population subnetwork with multi-server stations. *Spectrum* **2019**, *41*, 291–315. [CrossRef]
9. Barron, Y. The Delay Time Profile of Multistage Networks with Synchronization. *Mathematics* **2023**, *11*, 3232. [CrossRef]
10. Ramakrishnan, R.; Krishnamurthy, A. Analytical approximations for kitting systems with multiple inputs. *Asia-Pac. J. Oper. Res.* **2008**, *25*, 187–216. [CrossRef]
11. De Cuypere, E.; De Turck, K.; Fiems, D. A Maclaurin-series expansion approach to multiple paired queues. *Oper. Res. Lett.* **2014**, *42*, 203–207. [CrossRef]
12. Evdokimova, E.; Wittevrongel, S.; Fiems, D. A Taylor series approach for service-coupled queueing systems with intermediate load. *Math. Probl. Eng.* **2017**, *2017*, 3298605. [CrossRef]
13. Wu, H.; He, Q. Double-sided queues with marked Markovian arrival processes and abandonment. *Stoch. Model.* **2020**, *37*, 23–58. [CrossRef]
14. Liu, H.; Li, Q.; Chang, T.; Zhang, C. Double-ended queues with non-Poisson inputs and their effective algorithms. *Comput. Oper. Res.* **2022**, *144*, 105793. [CrossRef]
15. Błaszczyszyn, B.; Rolski, T.; Schmidt, V. Light-traffic approximation in queues and related stochastic models. In *Advances in Queueing: Theory, Methods, and Open Problems*; CRC Press: Boca Raton, FL, USA, 1995; pp. 379–406.
16. Kovalenko, I. Rare events in queueing systems—A survey. *Queueing Syst.* **1994**, *16*, 1–49. [CrossRef]
17. Altman, E.; Avrachenkov, K.; Núñez Queija, R. Perturbation analysis for denumerable Markov chains with application to queueing models. *Adv. Appl. Probab.* **2004**, *36*, 839–853. [CrossRef]
18. Avrachenkov, K.; Filar, J.; Howlett, P. *Analytic Perturbation Theory and Its Applications*; SIAM: Bangkok, Thailand, 2013.
19. Lasserre, J. A formula for singular perturbations of Markov chains. *J. Appl. Probab.* **1994**, *31*, 829–833. [CrossRef]
20. Blanc, J. Performance analysis and optimization with the power-series algorithm. In *Performance Evaluation of Computer and Communication Systems*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 53–80.
21. Blanc, J.; van der Mei, R. Optimization of polling systems with Bernoulli schedules. *Perform. Eval.* **1995**, *22*, 139–158. [CrossRef]
22. Koole, G. *On the Power Series Algorithm*; Technical Report; CWI: Amsterdam, The Netherlands, 1994.
23. van den Hout, W. The Power-Series Algorithm: A Numerical Approach to Markov Processes. Ph.D. Thesis, Tilburg University, Tilburg, The Netherlands, 1996.
24. Fiems, D.; Phung-Duc, T. Light-traffic analysis of random access systems without collisions. *Ann. Oper. Res.* **2019**, *277*, 311–327. [CrossRef]
25. De Cuypere, E.; De Turck, K.; Wittevrongel, S.; Fiems, D. Opinion propagation in bounded medium-sized populations. *Perform. Eval.* **2016**, *99–100*, 1–15. [CrossRef]
26. Weniger, E. Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. *Comput. Phys. Rep.* **1989**, *10*, 189–371. [CrossRef]
27. Brezinski, C.; Redivo-Zaglia, M. *Extrapolation Methods: Theory and Practice*; North-Holland: Amsterdam, The Netherlands, 1991.
28. Sidi, A. *Practical Extrapolation Methods: Theory and Applications*; Cambridge University Press: Cambridge, UK, 2003.
29. Graves-Morris, P.; Jenkins, C. Vector-valued rational interpolants III. *Constr. Approx.* **1986**, *2*, 263–289. [CrossRef]
30. Brezinski, C.; Redivo-Zaglia, M. A Survey of Shanks' Extrapolation Methods and Their Applications. *Comput. Math. Math. Phys.* **2021**, *61*, 699–718. [CrossRef]
31. Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B. *Numerical Recipes. The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.