Journal of Manufacturing Systems 68 (2023) 325–337

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

A Markovian model of asynchronous multi-stage manufacturing lines fabricating discrete parts

M.C. Magnanini *, T. Tolio

Politecnico di Milano, Dipartimento di Meccanica, via La Masa 1, 20156 Milano, Italy

ARTICLE INFO

Keywords:
Manufacturing systems
Markov modeling
Performance evaluation
Continuous model
Discrete flow

ABSTRACT

Asynchronous serial manufacturing lines that fabricate discrete parts are traditionally used in mass production, which represents a key sector in the global economy. Recent technological solutions for the modularization and standardization of manufacturing stations have led to this type of manufacturing system being reconfigured more often than in the past. Therefore, synthetic but accurate performance-evaluation models have become relevant as kernels in decision supporting tools for the continuous improvement of manufacturing systems. This paper presents a novel analytical model for the performance evaluation of asynchronous unreliable manufacturing lines fabricating discrete parts with finite buffers and deterministic processing times. This approach is based on continuous-time continuous-flow Markov chains. The general concept of operational cycles in discrete production is integrated into the modeling. The proposed model was validated using a discrete event simulation. The results demonstrate the accuracy and robustness of this model in evaluating a wide set of performance measures. The advantages of using this approach with respect to a purely continuous model were demonstrated. The applicability of the model to actual industrial scenarios was also demonstrated in a use case involving a high-volume assembly line.

1. Introduction

Modeling manufacturing systems for performance evaluation is concerned with developing models that accurately capture the dynamics of manufacturing systems and using these models to evaluate their performance. Manufacturing systems are complex systems that involve a range of interconnected components, such as machines, materials, and workers, that operate in a dynamic and uncertain environment. Modeling these systems is essential for gaining insights into their behavior and optimizing their performance, under varying production scenarios. Alternative modeling techniques may be used to capture system complexity and dynamics, such as discrete event simulation [1], data-driven meta-models [2], and analytical models [3]. Each modeling technique has different level of details and advantages to be used according to the situation and the system assumptions [4]. Detailed performance evaluation models are exploited for in-line control purposes, by for example integrating them into reinforcement learning algorithms [5]. Similarly, synthetic performance evaluation models become relevant as evaluation kernels within decision-supporting tools. Examples refer to configuration optimization [6], proactive identification of bottlenecks [7], integration of quality feedbacks [8], production loss analysis [9], identification of improvement actions [10] and reconfiguration decisions [11].

Among manufacturing systems, automated manufacturing lines that fabricate discrete products represent the backbone of global manufacturing. Their application sectors vary from automotive to machinery and from semiconductor fabrication to customized assembled products. This type of multi-stage line is composed of automated processing stations that perform various operations, such as welding, riveting, assembling, and inspection [12]. Technological evolution has caused automated stations to be highly repetitive; hence, they are characterized by a relatively steady processing time and high reliability. Furthermore, each station has a cycle time, meaning that the resulting manufacturing line is asynchronous. Moreover, the overall system configuration includes inter-operational buffers with finite capacity, which is frequently small, causing the stations to be coupled in their behavior. Consequently, system performance, such as throughput, is strongly affected by the current configuration.

This paper proposes a novel analytical model for evaluating the performance of asynchronous multi-stage serial lines that fabricate discrete parts and are characterized by deterministic processing times, stochastic reliability, and finite buffers. The modeling of stages is based on continuous-time discrete state Markov chains, whereas buffers are modeled as continuous variables. Control mechanisms mock the peculiar dynamics of discrete production and create the basis for the

* Corresponding author.
E-mail address: mariachiara.magnanini@polimi.it (M.C. Magnanini).

https://doi.org/10.1016/j.jmsy.2023.04.006
Received 14 December 2022; Received in revised form 1 March 2023; Accepted 12 April 2023
Available online 26 April 2023
0278-6125/© 2023 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

integration of control policies at the system-level. Thus, this paper aims to provide a methodology based on hypotheses that are extremely close to the actual industrial context and can be used to solve problems involving the design and operation of manufacturing systems.

The remainder of this paper is organized as follows. Section 2 describes the reference system and modeling assumptions, as well as the peculiar characteristics of discrete production. Section 3 introduces the proposed methodology with respect to the main modeling blocks and the performance measures. Section 4 provides a numerical analysis with respect to model validation, robustness and convergence, as well as the comparison with approaches that neglect the dynamics of discrete production. The application of the proposed model to an actual industrial case is presented in Section 5. Section 6 provides the final comments and future research directions.

1.1. Contribution to the state of the art

Analytical and approximate analytical methods represent a class of models for performance evaluation of manufacturing systems. The main output of these models is the evaluation of the joint influence of phenomena propagating along the line owing to disruptive events, such as machine stoppages, on the system performance. Owing to the availability of production data that support model validation, there has been renewed interest in analytical performance-evaluation models for serial manufacturing lines. Recent studies have focused on the accuracy and robustness [13], integration of analytical models and simulation models for multi-fidelity modeling [14] or hybrid performance evaluation [4], analysis of lines with special features as residence time constraints [15], mixed continuous-discrete production [16], and parallel machines in each stage [17], as well as integration of explicit process models into system-level evaluation [18].

Over the years, many contributions have been made to this field. Some relevant material and detailed reviews are presented in [19–23] including an in-depth analysis of problem formulations used in different approaches, such as *aggregation* and *decomposition* methods, as well as more in general multi-stage manufacturing system models.

The concept of aggregation involves aggregating every two machines into a new aggregated machine, and continue until the end of the line. This procedure is performed backward and forward repeatedly until the parameters of the aggregated machines converge, which can be proven analytically [24]. This method is based on Bernoulli machine characterization and has been widely used in many different applications [13,15]. Current research focuses on the improvement of the model accuracy with respect to restrictive assumptions which may hinder the applicability to real systems [25], considering also peculiar characteristics of manufacturing lines [26].

Similarly, the concept of decomposition involves decomposing the line into two-machine one-buffer short lines (building blocks), each of which represents the whole line centered in one buffer [27]. Decomposition equations link various building blocks, and a recursive algorithm is used to align the system performance as computed from alternative perspectives. Several decomposition approaches have been developed over the years based on the same iterative algorithm. A two-level decomposition, in which the machine and buffer levels are intertwined to guarantee the conservation of flow throughout the system was introduced based on discrete-time Markov model [28].

With respect to Markovian models, synchronous manufacturing systems are traditionally modeled using discrete-time models [29,30], whereas asynchronous manufacturing systems are traditionally modeled using continuous-time models [31].

In continuous models, material flows instantaneously from one machine to another, which function as valves. Hence, when continuous models are used for the performance evaluation of discrete manufacturing systems, they may introduce significant inaccuracies in the modeling [32,33], neglecting the part space on the machine and introducing dynamics not existing in the original system, i.e., slowdown.

Several works have investigated the difference between discrete and continuous models, and suggested methods to translate one model into another [34]. For example, the loss of space at the machines can be compensated for by making all buffers in the continuous model larger by one unit compared with the corresponding buffer capacities in the discrete model. Alternatively, the utilization of each machine in the line can be maintained to better approximate the work-in-progress of the continuous model with respect to the discrete model with the same buffer capacities. These approximations perform well when the buffer capacities are large. For small buffers, the approximation of continuous models with regard to discrete manufacturing systems increases because production mechanisms have a relevant effect on the dynamics. This is particularly evident in relation to the average buffer level [35].

[36] showed that continuous models can be considered as a limiting case of discrete models. They also provide proof of some asymptotic properties for systems characterized by communication blocking. These results are based on sample path analysis, which enhances the use of continuous models to approximate discrete systems. Earlier, [37] provided proof of the convexity of continuous models derived from discrete models. In recent years, interest in the use of continuous models to approximate discrete production systems has spread, e.g., [38] simulated a continuous model and subjected a machine to a delay to mock the discrete material flowing out of a machine waiting for a period before arriving at its downstream buffer.

More recently, analytical models of multi-stage manufacturing systems based on Markovian representation successfully addressed diverse system conditions and features, from remote quality integrated as feedback to the performance evaluation model [39] to analysis of manufacturing systems under nonstationary conditions [40]. Mostly, Markovian models are used when buffers are infinite or extremely large, thus the blocking effect is almost negligible [41], or for modeling stages with variable cycle time.

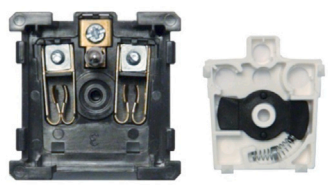
The aim of this study is to provide a general stochastic approximate analytical model based on a continuous-time Markovian model that can integrate control mechanisms for the accurate performance evaluation of asynchronous unreliable serial lines producing discrete parts. With respect to existing state of the art, this paper focuses in particular on addressing a sub-set of restrictive assumptions including deterministic cycle times, finite buffer capacities and asynchronous machines.

2. System description and analysis

In this section, a reference system and its characteristics and specific features are presented.

The reference system consists of a highly automated multi-stage manufacturing system (MMS), where each stage is decoupled by buffers from the others. Frequently, for this type of MMS, pick-and-place robots are used for loading/unloading operations among stations and buffers, making the MMS fully automated. Parts often travel on dedicated rails, which also serve as buffers between the stations. One of the main characteristics of this type of MMS is modularity, i.e., each station is considered a single module performing one operation on a part, such as welding, stamping, filling, and assembling. Automation advancements over the years have resulted in high repeatability of operations in time. Therefore, the processing time is repeatable and precise; hence, it can be considered to be deterministic. Section 5 proves this by presenting data from an actual MMS characterized by highly repeatable operations. Note that, given the wide variety of operations, the processing time is hardly equal among stations. Stoppages may occur along the line because of the mechanical and electronic dynamics of all the stations and feeding components. Operation-dependent failures are frequently dependent on mechanical dynamics, such as tool wear and degradation of machines and components, whereas time-dependent failures are frequently dependent on the dynamics of electronics, such as programmable logic controllers [42].

(a)



(b)

Fig. 1. MMS for the assembly of domestic sockets.

High-volume MMSs, such as assembly lines, packaging lines, and general fast automated manufacturing lines, belong to this set. An example of a modular MMS used for the assembly of domestic sockets is shown in Fig. 1[1] This line assembles six components in the main body of the socket. Vibration feeders convey them to the insertion mechanisms. The production rate of this line is 1800 parts/hour. Typical stoppages are related to vibration feeders, where components may become stuck, and some parts on the linear rails may be misplaced.

The following characteristics describe the reference MMS:

1. The first machine is never starved and the last machine is never blocked;
2. Processing times of the machines may be different between machines;
3. Machines are unreliable and may fail in different modes;
4. Time to failure and time to repair have a general distribution;
5. Load and unload times are negligible;
6. Parts arrive from outside and leave the system after being processed;
7. The capacities of the buffers are finite.
8. The system is asynchronous i.e. each machine can start or finish at any time without synchronization with the other machine.

2.1. Cyclic production dynamics

The asynchronous MMS fabrication of discrete parts is characterized by peculiar cyclic production dynamics. [35] defined a blocking-operational cycle and a starvation-operational cycle for two-machine lines.

The *blocking-operational* cycle occurs when the upstream machine is faster than the downstream machine. If both machines are operational and do not fail, the buffer soon becomes full. Subsequently, the upstream machine alternates the operational and blocking periods equal to the difference in processing times between the two machines.

The *starvation-operational* cycle occurs when the downstream machine is faster than the upstream one. If both machines are operational and do not fail, the buffer soon becomes empty. Subsequently, the downstream machine alternates operational and starvation periods equal to the difference in processing times between the two machines.

The system remains in these cyclic production dynamics, which are deterministically known until a disruptive event occurs, such as stochastic failures. Subsequently, the system will go out of this condition, but will tend to return to it and start over again. Therefore, explicit modeling of these dynamics enables a better adherence to reality.

In MMS, deterministic cyclic dynamics no longer involve only two machines, but an entire line, resulting in an increased complexity in modeling it. In an MMS, system-level dynamics imply the propagation of effects, leading to the additional analysis required to identify and understand possible cyclic production dynamics.

The intuition of this behavior can be explained using a short example that is based on a three-stage two-buffer (3M2B) line, where the same assumptions described in Section 2 and the system production mechanism is blocking after service (BAS).

Let us consider a 3M2B line with increasing cycle times, i.e., the first machine is faster than the second one, which is faster than the third one. In this case, the third machine represents the bottleneck of the line with respect to production cycle time. The sample path of this dynamic is shown in Fig. 2, where the system state in terms of machine states $\left(S_1(t), S_2(t), S_3(t)\right)$ and buffer levels $\left(x_1(t), x_2(t)\right)$ are provided at each time unit t. Machines can be either operational (green state), failed (red state) or idle because of blocking (orange state). When all three machines are operational, the two buffers soon become full. Subsequently, the entire line enters a *blocking-operational* cycle.

We can observe that the upstream machines depend on the bottleneck machine speed, and the entire system tends to remain in the system-level *blocking-operational* cycle. Indeed, machine M_1 completes

[1] Photo courtesy: www.cosberg.com.

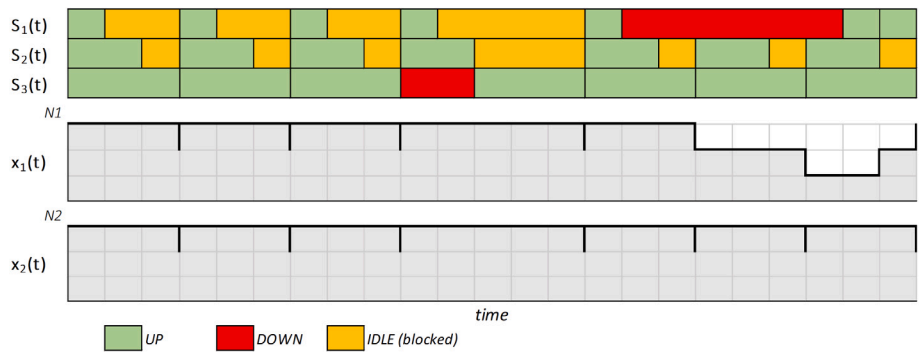


Fig. 2. Sample path of a 3M2B line with increasing cycle times at full-buffer dynamics ($ct_1 = 1, ct_2 = 2; ct_3 = 3$).

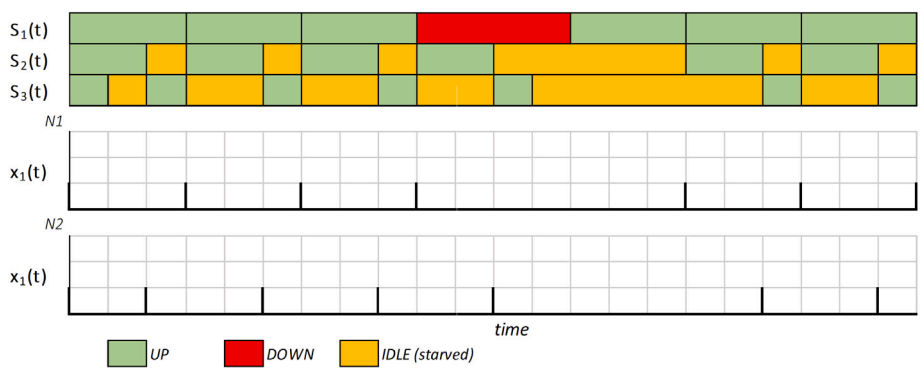


Fig. 3. Sample path of a 3M2B line with decreasing cycle times at empty-buffer dynamics ($ct_1 = 3, ct_2 = 2; ct_3 = 1$).

its part first, and it cannot unload it in the downstream buffer because machine M_2 has not completed its part yet. Therefore, machine M_1 becomes blocked. Subsequently, as soon as machine M_2 completes its part, it becomes blocked because machine M_3 has not yet completed its part. Only when machine M_3 completes its part and takes a new one from the upstream buffer can the other two machines M_1 and M_2 simultaneously unload their parts, take new ones, and start over again. Even when stochastic events occur, such as failures, the system will eventually synchronize again to start over the deterministic cyclic production dynamics of the *blocking-operational* cycle.

Similarly, a 3M2B line with decreasing cycle times can be considered, i.e., the third machine is faster than the second machine, which is faster than the first one. In this case, the first machine represents the bottleneck of the line with respect to production cycle time. The sample path of this dynamic is shown in Fig. 3. When all three machines are operational, the two buffers soon become empty. The entire line then enters a *starvation-operational* cycle.

The downstream machines depend on the bottleneck machine speed, and the entire system tends to remain in the system-level *starvation-operational* cycle. Indeed, machine M_3 completes its part first, and it cannot start a new one because machine M_2 has not completed its part yet and the upstream buffer is empty. Therefore, machine M_3 becomes starved. Then, as soon as machine M_2 completes its part, it becomes starved as well because machine M_1 has not completed its part yet, but machine M_3 can process the part coming from M_2. When machine M_1 completes its part and takes a new one, machine M_2 can load the new part, whereas machine M_3 must wait for this to be completed. Even when stochastic events occur, such as failures, the system will eventually synchronize again to start over the deterministic cyclic production dynamics of the *starvation-operational* cycle.

3. Methodology

In this section, a methodology for modeling the reference MMS described in 2 is presented. Additional modeling assumptions are provided in 3.1, and the characterization of the single-stage and multi-stage models by means of Markovian modeling is introduced in subsequent sections.

3.1. Assumptions

The system model is composed by K machine decoupled by $K - 1$ buffer. Each machine M_k corresponds to a single stage of the reference system. Each upstream machine M_k processes parts and puts them in the buffer B_k, and each downstream machine M_{k+1} takes the parts from the buffer B_k and processes them. In our notation, machines are represented by squares and buffers by circles, as shown in Fig. 4.

The characteristics listed in 2 apply to the model. The following additional specifications are introduced to define a subcategory of systems that can be analyzed with the proposed methodology.

1. Only one part type is produced;
2. Machines are characterized by one operational mode;
3. Failures are characterized by general repair time distributions;
4. The dispatching policy is first-in first-out (FIFO);
5. Parts are not scrapped or reworked.

To summarize, the proposed model is capable of addressing asynchronous MMS fabricating discrete parts, characterized by buffers with finite capacities, both Operation-dependent and Time-dependent failures. Therefore, the category addressed in the paper is quite large since system with the characteristics described can be found in many real applications.

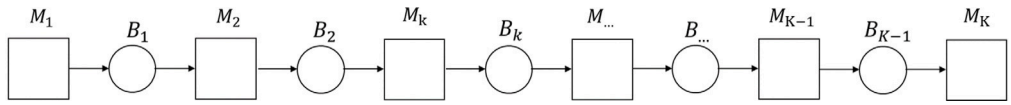


Fig. 4. Serial manufacturing line.

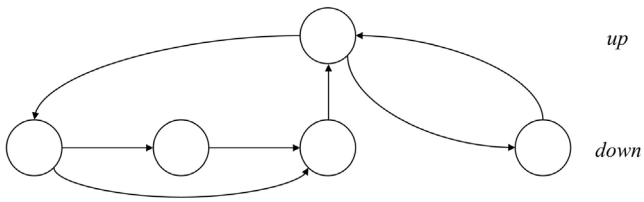


Fig. 5. Markov chain for single up — multiple down machine.

3.1.1. Single-stage modeling

Each machine in the isolation of the model corresponds to one stage of the original reference MMS. It is characterized by its own behavior, which is not influenced by the remainder of the system. A continuous-time discrete-state Markov chain is used to describe the behavior of the machine in isolation, which therefore acts as valve with respect to the upstream and downstream flows.

The machine in isolation M_k is characterized by the vectors of states $S_k = S[i]$ of size I. By convention, the first element in the vector of states represent the operational state and it is named 'up' state. When machine M_k is in the up state $S_k = S[1]$, it produces at rate $\mu^k(S[1])$. Conversely, the other elements in the vector of states represent the failure states and they are named 'down' states. When machine M_k is failed in a down state $S[i], i \neq 1$, it produces at rate $\mu^k(S[i]) = 0$.

An example of the Markov Chain of a machine is proposed in Fig. 5. The transition rates among the states of machine M_k are contained in the matrix Q^k with transition rates $q^k[i,j]$.

A machine in isolation is characterized by the efficiency in isolation e_k, which represents the probability that machine M_k is operational because it is not failed. The production rate in isolation $\rho_k = \mu \cdot e_k$ represents the production rate of machine M_k if it were never impeded by the other machines or buffers.

3.2. Outline of the method

The objective of the approach presented herein is to provide a method to accurately evaluate the steady-state performance of multi-stage asynchronous manufacturing systems for fabricating discrete parts based on single-stage dynamics. The main performance measures are the system throughput, average inventories, and steady-state probabilities of the machines. The steady-state probabilities of the machines include the probability that each machine is upstream- or downstream-limited by another machine in the system.

The proposed model is based on approximate analytical methods, particularly on a two-level decomposition approach, as depicted in Fig. 6.

In the two-level decomposition approach, the manufacturing system is decomposed according to two perspectives:

- At machine level, the manufacturing line is decomposed into Integrated Machines $M[k]$. Each integrated machine represents the original machine inserted into the system. Its state-based representation includes limiting states from the upstream and downstream parts of the line with respect to the considered machine. Therefore, the integrated machine models the dynamics of the overall system centered on the considered machine. The cyclic production dynamics discussed in Section 2.1, they are modeled approximately within the Integrated Machine using Markovian transitions. The aim of two-level decomposition is to accurately estimate the transition parameters from the buffer level.
- At buffer level, the manufacturing line is decomposed into Building Blocks $BB(k)$. Each building block is composed of an upstream pseudo-machine and a downstream pseudo-machine decoupled by a buffer, where the upstream pseudo-machine represents the upstream part of the line with respect to the buffer concerned and the downstream pseudo-machine represents the downstream part of the line in relation to the buffer concerned. Therefore, the building block represents the inflow and outflow of the overall system centered on the buffer, which is regulated by the dynamics of the upstream and downstream pseudo-machines. The cyclic production dynamics discussed in Section 2.1 are modeled exactly within the building blocks using threshold-based control mechanisms.

The model is based on continuous-time semi-Markov chains, and at the machine level, the integrated machines are modeled using continuous-time discrete-state Markov chains; at the buffer level, the building blocks are modeled by continuous-time mixed continuous- and discrete-state Markov chains.

3.3. Machine level: Integrated machine

The Integrated Machine $M[k]$ represents the machine as it is inserted into the system. An Integrated Machine $M[k]$ adds to the behavior of the corresponding machine in isolation M_k with the limiting phenomena resulting from the upstream part of the line, i.e., starvation, and from the downstream part of the line, i.e., blocking. Therefore, the integrated machine represents the entire line centered on machine $M[k]$.

The limiting phenomena descending from the upstream part of the line with respect to the considered integrated machine $M[k]$ consists of starvation phenomena, i.e., scenarios in which the upstream buffer B_{k-1} with respect to the considered Integrated Machine $M[k]$ becomes empty owing to the interruption of flow in the upstream part of the line. The interruption of the flow may be owing to stoppages, such as machine failures, as well as starvation-operational cycles with upstream bottlenecks.

The limiting phenomena descending from the downstream part of the line with respect to the considered integrated machine $M[k]$ consist of blocking phenomena, i.e., scenarios in which the downstream buffer B_k with respect to the considered Integrated Machine $M[k]$ becomes full because of the interruption of flow in the downstream part of the line. The interruption of the flow may be owing to stoppages, such as machine failures, as well as blocking-operational cycles with downstream bottlenecks.

The Integrated Machine $M[k]$ adds to the behavior of the original machine in isolation, namely *Local* states $\mathcal{L}^{[k]}$, two state partitions:

- The remote *starvation* states $\mathcal{S}^{[k]}$ represent the states in which the Integrated Machine $M[k]$ is upstream limited.
- The remote *blocking* states $\mathcal{B}^{[k]}$ represent the states in which the Integrated Machine $M[k]$ is downstream limited.

Each machine in the line M_k can be represented by the corresponding Integrated Machine $M[k]$ using a continuous-time discrete-state Markov Chain, acting as valves in modeling the upstream and downstream flows with respect to the considered machine.

Therefore, the Integrated Machines $M[k], k = 1, \ldots, K$ represent the same system from different perspectives, i.e. the single stages. Because all integrated machines represent the same system, they provide the same system performance, i.e., system throughput, with different steady-state probabilities. Hence, the conservation of flow is guaranteed, as shown in the following sections.

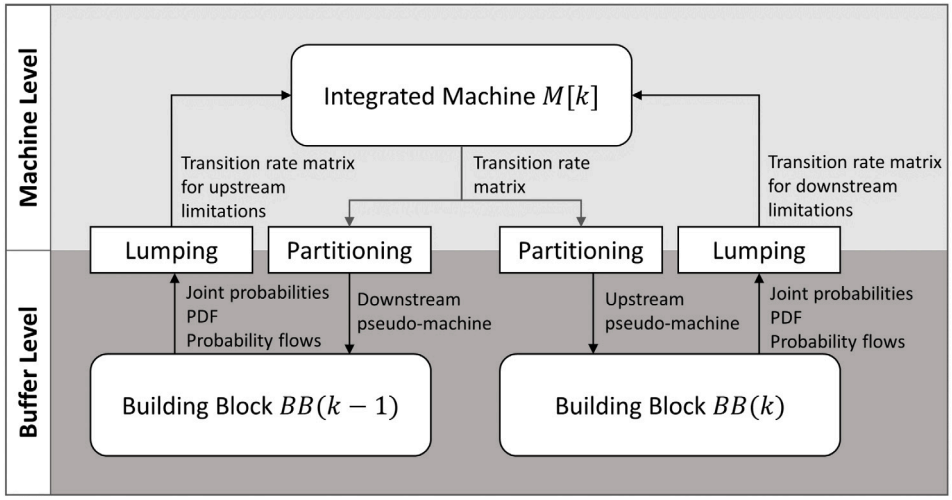


Fig. 6. Schematical representation of the proposed method.

3.4. Buffer level: Building block

The Building Block $BB(k)$ is a two-machine one-buffer line representing the inflow and outflow of the overall system centered on the considered buffer. The inflow is modeled using the upstream pseudo-machine $M^u(k)$, and the outflow is modeled using the downstream pseudo-machine $M^d(k)$.

The state $S(k)$ of the identified Building Block $BB(k)$ is represented by the triplet $S(k) = (x, S^u, S^d)$, where x is a continuous variable representing the buffer level, S^u is a discrete variable representing the states of the upstream pseudo-machine and S^d is a discrete variable representing the states of the downstream pseudo-machine. The duplet (S^u, S^d) denotes the joint machine states.

To account for the peculiar dynamics of discrete manufacturing, the two-stage modeling proposed by [35] is considered in this paper. Controlled mechanisms mock the operational cycles at buffer level, by modeling explicitly the blocking state for the upstream machine and the starvation state for the downstream machine.

Hence, the total number of joint machine states includes

- The joint machine states when no limitation occurs: $S^u_{(B)} \otimes S^d_{(S)}$, where $S^u_{(B)}$ denotes all possible upstream states excluding the blocking state B, $S^d_{(S)}$ denotes all possible downstream states excluding the starvation state S, and $\otimes$ denotes the Kronecker product.
- The joint machine states when downstream limitations occur, i.e. the upstream machine is blocked: $B \otimes S^d_{(S)}$.
- The joint machine states when upstream limitations occur, i.e. the downstream machine is starved: $S^u_{(B)} \otimes S$.

The solution method of the building block used in this paper is that proposed by [43], namely, the Generalized Threshold Method (GTM). The GTM enables the evaluation of the steady state performance of a continuous two-machine system with a finite buffer characterized by ranges.

According to the selected Building Block model, the solution method returns the steady state probability density functions $f(x, S^u, S^d)$ (PDF) for each joint machine state (S^u, S^d) as a function of the buffer level x. The proposed method is valid if other similar two-stage models are considered, which are based on the Markovian representation of machines, and for which the solution method returns the PDF.

Based on the PDF, the following main output can be computed: (i) the steady-state probabilities of the joint machine states $\pi(S^u, S^d)$ as

$$\pi(S^u[z], S^d[j]) = \int_0^N f(x, S^u[z], S^d[j])dx \qquad \forall z \forall j \tag{1}$$

and (ii) the boundary probability flows between joint machine states as a function of the empty or full buffer levels, $G(0, S^u S^d \to S^u S^d)$ and $G(N, S^u S^d \to S^u S^d)$ respectively, as

$$G(0, S^u S^d \to S^u S^d) = \left(\mu(S^u) - \mu(S^d)\right) \cdot B_2 \cdot f(0, S^u_{(B)}[z], S^d[j])dx \qquad \forall z \forall j \tag{2}$$

$$G(N, S^u S^d \to S^u S^d) = \left(\mu(S^u) - \mu(S^d)\right) \cdot B_2 \cdot f(N, S^u[z], S^d_{(S)}[j])dx \qquad \forall z \forall j \tag{3}$$

where B_2 is a Boolean matrix that defines the possible boundary transitions between joint machine states.

3.5. From buffer-level to machine-level: Lumping

The objective of this step is to characterize the integrated machine based on the output provided by the building block solution, in particular, (i) characterization of the state space and (ii) characterization of the transition rate matrix.

3.5.1. State space and state probabilities

The state space of each Integrated Machine $M[k]$ is defined from the corresponding machine in isolation M_k and from the Building Blocks $BB(k-1)$ and $BB(k)$.

$$\mathcal{L}^{[k]} = S_k \tag{4}$$

$$\mathcal{B}^{[k]} = B(k) \otimes S^d_{(S)}(k) \tag{5}$$

$$\mathcal{S}^{[k]} = S^u_{(B)}(k-1) \otimes S(k-1) \tag{6}$$

For large scale systems, i.e. with a number of stages higher than 10, the state space could become extremely large, thus leading to problems of state explosion. This problem can be addressed by means of lumping. An example related to large systems has been investigated by the same authors [4].

From the output of the building blocks, the steady-state probabilities of the states of the integrated machine $M[k]$ can be computed through partial lumping:

$$\Pi_{\mathcal{L}}[i] = \Sigma_j \pi_{(k)}(S^u[i], S^d[j]) \qquad \forall i \tag{7}$$

$$\Pi_{\mathcal{L}}[i] = \Sigma_z \pi_{(k-1)}(S^u[z], S^d[i]) \qquad \forall i \tag{8}$$

$$\Pi_{\mathcal{B}}[j] = \pi_{(k)}(B, S^d[j]) \qquad \forall j \tag{9}$$

$$\Pi_{\mathcal{S}}[z] = \pi_{(k-1)}(S^u[z], S) \qquad \forall z \tag{10}$$

We can observe that the steady-state probabilities of local states in $M[k]$ can be derived equally from both $BB(k-1)$ and $BB(k)$, where the original machine M_k is included either in the downstream pseudo-machine $M^d(k-1)$ or the upstream pseudo-machine $M^u(k)$. In contrast, the steady-state probabilities of the blocking states in the Integrated

Machine $M[k]$ are derived from the Building Block $BB(k)$, where the original machine M_k is downstream limited by the remainder of the system. Similarly, the steady-state probabilities of the starvation states in the Integrated Machine $M[k]$ are derived from the Building Block $BB(k-1)$, where the original machine M_k is upstream limited by the remainder of the system.

3.5.2. Transition rate matrix

At the machine-level, the goal is to define the transition rates in order to have a complete characterization of the transition rate matrix $Q^{[k]}$ belonging to the continuous-time discrete-state Markov Chain of each Integrated Machine $M[k]$.

Let us recall the definition of the transition rate matrix $Q^{[k]}$:

$$Q^{[k]} = \begin{bmatrix} Q_{\mathcal{L}\mathcal{L}} & Q_{\mathcal{L}\mathcal{S}} & Q_{\mathcal{L}\mathcal{B}} \\ Q_{\mathcal{S}\mathcal{L}} & Q_{\mathcal{S}\mathcal{S}} & Q_{\mathcal{S}\mathcal{B}} \\ Q_{\mathcal{B}\mathcal{L}} & Q_{\mathcal{B}\mathcal{S}} & Q_{\mathcal{B}\mathcal{B}} \end{bmatrix} \tag{11}$$

In the following, the decomposition equations used to define the submatrices are introduced. There are two sets of equations: the first defines the transition rates to enter and exit the limiting states, and represents the computation of transition rates at the machine level from the controlled transitions at the buffer level, and the second defines the transition rates among remote states.

Entering and exiting the limiting states. This set of equations defines the transition rates for entering and exiting the limiting states. These equations are based on the balance equations for the continuous-time Markov chain (CTMC). The balance equations are based on probability flow, and it is generally computationally intractable to solve this system of equations for most queueing models [44]. However, in this case the output from the Building Block evaluation contributes to the solution of these equations.

The corresponding transition rate matrices can be computed as

$$Q_{LS}^{[k]} = G_{LS}(k-1) \odot \left[\Pi_L(k-1)\right]^{-1} \tag{12}$$

$$Q_{LB}^{[k]} = G_{LB}(k) \odot \left[\Pi_L(k)\right]^{-1} \tag{13}$$

$$Q_{SL}^{[k]} = G_{SL}(k-1) \odot \left[\Pi_S(k-1)\right]^{-1} \tag{14}$$

$$Q_{BL}^{[k]} = G_{BL}(k) \odot \left[\Pi_B(k)\right]^{-1} \tag{15}$$

Where $\odot$ indicates the Hadamard product,[2] $[v]^{-1}$ indicates the Samelson inverse[3] of a vector v and the subscript of probability flows denotes the corresponding state partitions.

Transitions among limiting states. This set of equations define the transition rates among limiting states of the same type. Because there is a bi-unique relation between Integrated Machine $M[k]$ and pseudo-machine $M^{d(k-1)}$ and in turn in $BB(k-1)$, pseudo-machine $M^{d(k-1)}$ can only be limited by $M^{u(k-1)}$, and a change of limitation corresponds to a change in the state of pseudo-machine $M^{u(k-1)}$ therefore,

$$Q_{SS}^{[k]} = Q^u(k-1) \tag{16}$$

Similarly,

$$Q_{BB}^{[k]} = Q^d(k) \tag{17}$$

3.6. From machine-level to buffer-level: Partitioning

Based on the characterization of the machine level, the input to the buffer level can be defined in terms of the state space and transition rate matrix of the pseudo-machines for each building block $BB(k)$.

A schematic representation of the relation between the pseudo-machines at buffer-level and the Integrated Machines at machine-level is provided in Figure 7.

In particular, the upstream pseudo-machine $M^u(k)$ is characterized by state space $S^u(k) = [\mathcal{L}^{[k]}, \mathcal{S}^{[k]}]$. The corresponding transition rate matrix $Q^u(k)$ is

$$Q^u(k) = \begin{bmatrix} Q_{\mathcal{L}\mathcal{L}}^{[k]} & Q_{\mathcal{L}\mathcal{S}}^{[k]} \\ Q_{\mathcal{S}\mathcal{L}}^{[k]} & Q_{\mathcal{S}\mathcal{S}}^{[k]} \end{bmatrix} \tag{18}$$

Similarly, the downstream pseudo-machine $M^d(k)$ is characterized by the state space $S^d(k) = [\mathcal{L}^{[k+1]}, \mathcal{B}^{[k+1]}]$. The corresponding transition rate matrix $Q^d(k)$ is

$$Q^d(k) = \begin{bmatrix} Q_{\mathcal{L}\mathcal{L}}^{[k+1]} & Q_{\mathcal{L}\mathcal{B}}^{[k+1]} \\ Q_{\mathcal{B}\mathcal{L}}^{[k+1]} & Q_{\mathcal{B}\mathcal{B}}^{[k+1]} \end{bmatrix} \tag{19}$$

3.7. Algorithm for convergence of the performance evaluation

Step 0 For $m = 1...M$, Integrated Machine $M[k]$ is initialized based on $M\{m\}$.

Step 1 For $k = 1...K-1$:

1. Characterization of upstream and downstream pseudo-machines $M^u(k)$ and $M^d(k)$ from $M[k]$ and $M[k+1]$.
2. Evaluation of Building Block $BB(k)$, based on $M^u(k)$, $M^d(k)$ and $B(k)$.
3. Characterization of Integrated Machine $M[k+1]$ based on the downstream pseudo-machine $M^d(k)$.

Step 2 For $k = K-1...1$:

1. Characterization of upstream and downstream pseudo-machines $M^u(k)$ and $M^d(k)$, from $M[k]$ and $M[k+1]$.
2. Evaluation of Building Block $BB(k)$, based on $M^u(k)$, $M^d(k)$ and $B(k)$.
3. Characterization of Integrated Machine $M[k]$ based on the upstream pseudo-machine $M^u(k)$.

Step 1 and 2 to be repeated until throughput $TH(k)$ calculated in the various Building Blocks $BB(k), k = 1, \dots, K-1$ and Integrated Machines $M[k], k = 1, \dots, K$ converge.

This algorithm is based on the DDX algorithm proposed by [48]. The convergence has not been proved yet, however the algorithm converged in all the analyzed cases, as will be shown in the following Sections.

3.8. Performance evaluation

At steady state, each Building Block $BB(k)$ and each Integrated Machine $M[k]$ represent the entire line centered in buffer B_k and in machine M_k, respectively.

In particular, the solution method provides the steady-state probabilities of the integrated machines $\Pi\left(S_k\right)$. The main performance measures can be evaluated based on these results.

For each Integrated Machine $M[k]$ in the line, the steady-state probabilities can be computed by solving the corresponding Markov Chain. The following performance measures can be evaluated.

- System-level efficiency $E[k]$: This represents the probability that machine $M[k]$ is operational because it is not failed or limited by the other resources in the system:

$$E[k] = \Pi_{[k]}(U) \tag{20}$$

[2] The Hadamard product, also known as vector multiply, is commonly used in matrix computation [45].

[3] The Samelson inverse was firstly introduced in [46] and then in [47]. It defines the inverse of a vector v as: $[v]^{-1} = \frac{\bar{v}}{\bar{v} \cdot v} = \frac{\bar{v}}{\|v\|^2}$.

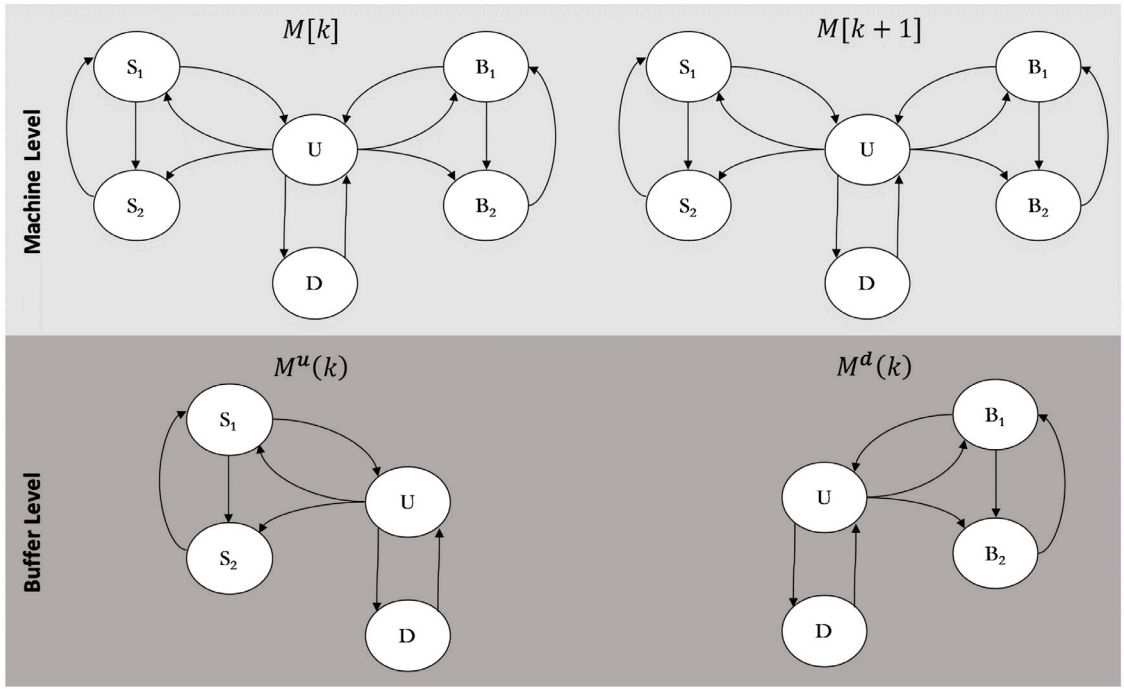


Fig. 7. Relation between the Markov Chains of Integrated Machines (machine-level) and pseudo-machines (buffer-level).

- Machine throughput $TH[k]$, expressed in $[parts/t.u.]$: This represents the average production rate of Integrated Machine $M[k]$:

$$TH[k] = \mu_k \cdot \Pi_{[k]}(U) \tag{21}$$

- Failure probability $\Pi_{[k]}(D_i), i = 1, \dots I$: The probability that at steady-state the Integrated Machine $M[k]$ is failed in failure mode i:

$$\Pi_{[k]}(D_i), i = 1, \dots I \tag{22}$$

- Blocking probability $\Pi_{[k]}(B)$: The probability that at steady-state the Integrated Machine $M_{[k]}$ is idle because the downstream buffer B_k is full:

$$\Pi_{[k]}(B) = \sum_{S \in Bl} \Pi_{[k]}(S) \tag{23}$$

- Starvation probability $\Pi_{[k]}(S)$: The probability that at steady-state the Integrated Machine $M_{[k]}$ is idle because the upstream buffer B_{k-1} is empty:

$$\Pi_{[k]}(S) = \sum_{S \in St} \Pi_{[k]}(S) \tag{24}$$

- Mean Time to Blocking $MTTB_{[k]}$ and Mean Time to Unload $MTTU_{[k]}$, expressed in $[t.u.]$: These represent the average time before machine $M[k]$ becomes blocked when operational and returns operational when being blocked, respectively:

$$MTTB_{[k]} = \frac{\Pi_{[k]}(B)}{\Pi_{[k]}(U)} \cdot q_{BL} \tag{25}$$

$$MTTU_{[k]} = \frac{1}{q_{BL}} \tag{26}$$

- Mean Time to Starvation $MTTS_{[k]}$ and Mean Time to Load $MTTL_{[k]}$, expressed in $[t.u.]$: These represent the average time before machine $M[k]$ becomes starved when operational and returns operational when being starved, respectively:

$$MTTS_{[k]} = \frac{\Pi_{[k]}(S)}{\Pi_{[k]}(U)} \cdot q_{SL} \tag{27}$$

$$MTTL_{[k]} = \frac{1}{q_{SL}} \tag{28}$$

The main system performance measures that can be computed by integrating the machine-level output and the buffer-level output are as follows:

Throughput. The Throughput TH, expressed in $[parts/t.u.]$ (where $t.u.$ stands for $timeunit$): the average production rate produced by the system at steady state. The overall system throughput is evaluated as:

$$TH = TH[k] \qquad \forall k = 1, \dots, K \tag{29}$$

Given the conservation of flow, it also holds that

$$TH(k) = TH[k], \forall k \tag{30}$$

where $TH(k)$ is the throughput calculated in Building Block $BB(k)$.

Work in progress. The average buffer level $\bar{x}_k$, expressed in $[parts]$ represents the average content of each buffer B_k. The average buffer level $\bar{x}_k$ is an output of the Building Block solution. The total work in progress (WIP) of the system can be computed from the total average content of buffers and the total average content of machines:

$$\bar{WIP} = \sum_{k=1}^{K-1} \bar{x}_k + \sum_{k=1}^{K} \left(1 - \Pi_{[k]}(S)\right) \tag{31}$$

Average system time. The average system time $\bar{T}$, expressed in $[t.u.]$: The average time that a part spends in the system. It can be evaluated by applying the Little's Law. However, it must be considered that the total average number of parts in the system is given by the total average content of buffers and the total average content of machines; thus:

$$\bar{T} = \frac{\sum_{k=1}^{K-1} \bar{x}_k + \sum_{k=1}^{K} \left(1 - \Pi_{[k]}(S)\right)}{TH} \tag{32}$$

Taken together, the proposed performance measures provide a detailed evaluation of manufacturing systems to support their optimization and continuous improvement.

4. Numerical results

The performance results obtained with the proposed approximate analytical model (indicated as $MMTT$ in the Tables) are compared with those obtained using a discrete event simulation model (indicated as Sim in the Tables) to test the ability of the proposed continuous approach to consider the peculiar dynamics of manufacturing systems fabricating discrete parts. Furthermore, the robustness of the proposed model was tested by exploring a wide range of system parameters that enable its applicability in different scenarios. A specific section is devoted to an analysis of the convergence of the proposed algorithm. In addition, a comparison with a purely continuous model is proposed (indicated as $Cont$ in the tables) to demonstrate the improvement in the accuracy of the results obtained with the proposed model in comparison with classical continuous analytical models.

The discrete event simulation model was implemented using SimEvents© – a MATLAB tool within Simulink – and was built based on the reference system introduced in Section 2. For each system configuration, the results were derived from a five-replicate simulation of the duration of 1.000.000[$t.u.$]($t.u.$ = time unit). The warm-up period was computed following the Welch method [49]. For all reported results, a 95% confidence interval was calculated.

Four types of manufacturing lines were evaluated, i.e. three-machine line ($3M2B$), five-machine line ($5M4B$), seven-machine line ($7M6B$), and nine-machine line ($9M8B$). The machine parameters were randomly selected from the following intervals:

$$e_k \in [0.75, 0.95] \qquad MTTR_k \in [5, 50] \qquad \mu_k \in [0.5, 5] \tag{33}$$

For each serial layout, three buffer capacities were tested:

$$N_k = \{2, 10, 30\} \tag{34}$$

In total, 4800 cases were evaluated on a PC with Intel® Core™ $i7-6820HQ$2.7 GHz and 8.00 RAM installed. A detailed analysis of these cases is presented in the following Sections.

The errors of the model compared with the discrete event simulation were computed as follows:

$$err\%_{TH} = \frac{|TH_{MMTT} - TH_{Sim}|}{TH_{Sim}} \cdot 100 \tag{35}$$

$$err\%_{WIP} = \frac{|WIP_{MMTT} - WIP_{Sim}|}{N} \cdot 100 \tag{36}$$

$$err_{Probabilities} = \left(|Prob_{MMTT} - Prob_{Sim}|\right) \cdot 100 \tag{37}$$

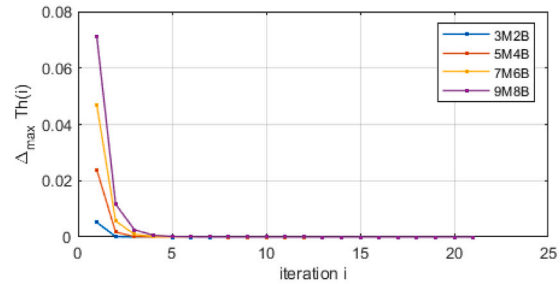
4.1. Validation

The numerical results of the main performance measures are summarized in Table 1. The model is proven to be precise with respect to discrete event simulation, not only for throughput and steady-state probabilities, but also for the average buffer level. Both the mean error and the standard deviation were considerably low. When the buffer capacity was small, i.e. $N = 2$, the average difference between the simulation and the $MMTT$ model is less than half of the absolute value. Moreover, the mean error as well as standard deviation do not seem to be affected by the size of the system, i.e. number of stages. We can then argue that the proposed model maintains its accuracy with respect to the underlying state-space within the defined boundaries.

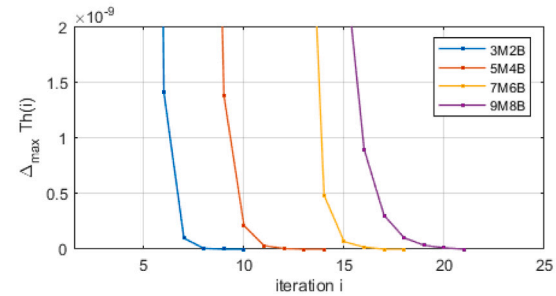
4.2. Convergence

Although the proof of convergence is currently under study, all calculated cases reached convergence in a limited number of iterations. The details are presented in Table 2.

The number of iterations required to reach convergence strongly depends on buffer size. In particular, small buffers enable a short convergence path with respect to larger buffers without a significant difference with respect to the number of machines in the line.



(a) Convergence paths.



(b) Convergence details.

Fig. 8. Analysis of convergence performance.

For all the calculated cases, the maximum difference in the throughput among the Building Blocks was computed for each iteration i as follows:

$$\Delta_{max} TH(i) = max\{TH(1,i), \dots, TH(K-1,i)\} - min\{TH(1,i), \dots, TH(K-1,i)\} \tag{38}$$

Fig. 8(a) shows the average difference in throughput per iteration for all the tested cases. Fig. 8(b) focuses on a smaller scale in order to show the details of the iteration measures also for small values. The algorithm is set to reach a precision of 10^{-14} on Matlab. We observed that a fair value of precision, e.g. 10^{-8}, was already reached for a low number of iterations in all the tested layouts. This can result in clear advantages when the proposed model is used as the evaluation kernel for optimization algorithm.

4.3. Comparison with continuous models

The continuous model is the one proposed in [50] and well known in literature. In this model, the buffer capacity has been set to $N + 1$ in order to account for the space of the downstream machine [19].[4] Hereafter, the continuous model is referred to as $Cont$. The purely continuous model does not introduce a significant approximation in modeling discrete manufacturing systems with respect to throughput. However, this introduces a relevant approximation when evaluating the average inventory level. Fig. 9 shows boxplots of the errors with respect to the simulation for the average inventory level.

[4] The output of the continuous model has been calculated as follows:

• The blocking probability includes the contribution of the pure blocking and the portion of blocking during the blocking-operational cycle, which corresponds in the continuous model to the slowdown state.

• The average buffer level has been computed according to the real buffer capacity. Therefore, when the average inventory level in the modeled system is greater than the buffer capacity, it means that the buffer is full and the average inventory level has been set equal to N.

Table 1
Summary of validation results.

Line	Group	TH		WIP		$\Pi(S)$		$\Pi(B)$	
		$err\%_{TH}$	stddev	$err\%_{WIP}$	stddev	$err\%_{Prob}$	stddev	$err\%_{Prob}$	stddev
3M2B	$N = 2$	0.40	0.21	8.53	5.22	<0.05	<0.05	<0.05	<0.05
	$N = 10$	0.20	0.10	2.45	1.13	<0.05	<0.05	<0.05	<0.05
	$N = 30$	0.09	0.07	1.07	0.72	<0.05	<0.05	<0.05	<0.05
5M4B	$N = 2$	1.07	0.32	7.99	5.96	<0.05	<0.05	<0.05	<0.05
	$N = 10$	0.96	0.33	2.99	2.28	<0.05	<0.05	<0.05	<0.05
	$N = 30$	0.70	0.50	2.79	3.03	<0.05	<0.05	<0.05	<0.05
Line	Group	TH		WIP		$\Pi(S)$		$\Pi(B)$	
		$err\%_{TH}$	stddev	$err\%_{WIP}$	stddev	$err\%_{Prob}$	stddev	$err\%_{Prob}$	stddev
7M6B	$N = 2$	0.83	0.48	7.06	5.91	<0.05	<0.05	<0.05	<0.05
	$N = 10$	1.17	0.58	2.65	1.65	<0.05	<0.05	<0.05	<0.05
	$N = 30$	0.63	0.51	1.92	1.67	<0.05	<0.05	<0.05	<0.05
9M8B	$N = 2$	1.51	0.23	7.59	5.38	<0.05	<0.05	<0.05	<0.05
	$N = 10$	1.60	0.06	2.50	1.47	<0.05	<0.05	<0.05	<0.05
	$N = 30$	0.78	0.27	1.54	1.55	<0.05	<0.05	<0.05	<0.05

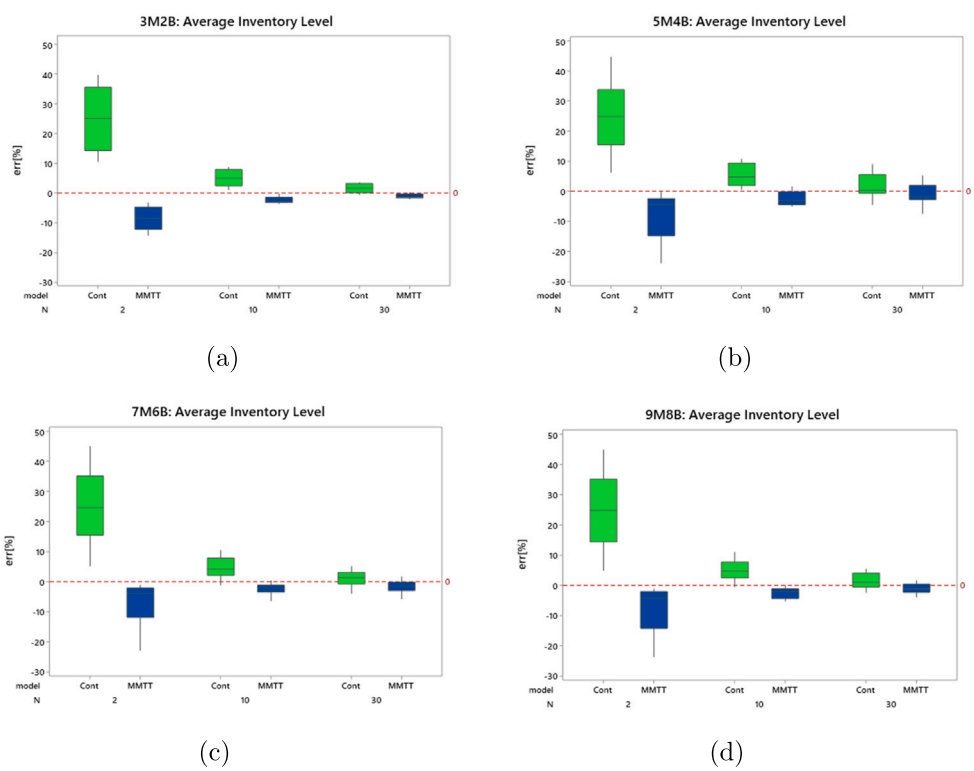


Fig. 9. Comparison in the Average Inventory Level estimation for the proposed $MMTT$ model and the $Cont$ model.

Table 2
Number of iterations to obtain the convergence.

Line	Group	$iter_{max}$	$iter_{min}$	$avg.iter$	Line	Group	$iter_{max}$	$iter_{min}$	$avg.iter$
3M2B	$N = 2$	5	4	4.5	7M6B	$N = 2$	7	7	7
	$N = 10$	7	6	6.08		$N = 10$	12	11	11.5
	$N = 30$	9	5	7.08		$N = 30$	16	12	14
5M4B	$N = 2$	7	6	6	9M8B	$N = 2$	8	8	8
	$N = 10$	9	9	9		$N = 10$	14	13	13.5
	$N = 30$	12	9	10.75		$N = 30$	20	13	16.75

When the buffer capacity was small and equal to 2, the $MMTT$ model underestimated the average inventory level by −10% on average for all serial layouts, and then it decreased to an average error of −1.7% and −0.3% for the medium and large buffers, respectively. However, the average inventory level was constantly overestimated by the $Cont$ model for small, medium, and large buffer capacities. In particular, when the buffer capacity was small, the error was higher than 25% for all serial layouts, with maximum error of 48% when the line was composed of more than three machines.

The proposed $MMTT$ model outperformed the purely continuous model in evaluating the average inventory level in all cases. Moreover, it showed a reduced variance compared with the continuous model. This means that applying the $MMTT$ model for the performance evaluation of manufacturing systems that produce discrete parts results in more reliable metrics.

5. Real case

The $MMTT$ model was used for performance evaluation in an actual manufacturing line with the aim of identifying critical stages; therefore suggesting improvement directions to the company.

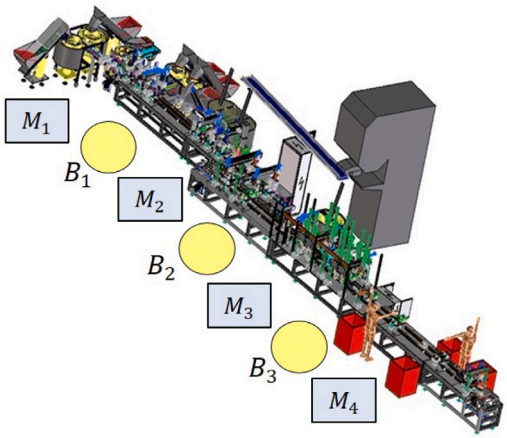


Fig. 10. Multi-stage manufacturing line producing drawer side walls.

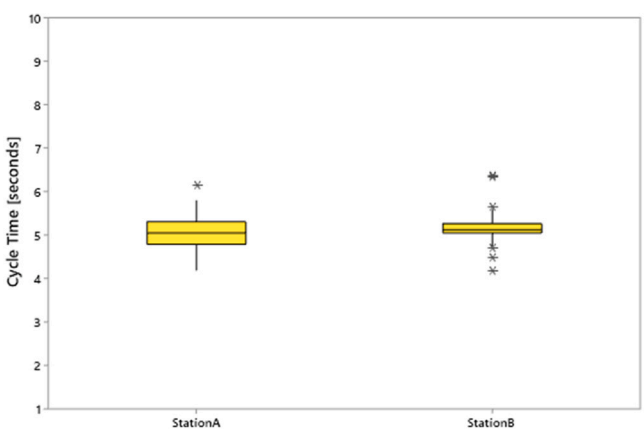


Fig. 11. Boxplots of cycle times for two random stations.

The reference case has been described previously [51]. The company is an Italian manufacturer that produces drawer side walls for personalized kitchens. More than 20 components are assembled to the main body of the drawers by the production line shown in Fig. 10,[5] with a high level of automation. Four stages can be identified in the line according to inter-operational buffers. Linear guide-rails serve as buffers in the line, where the capacity is given by the total length. Each stage comprises different stations performing elementary operations. The first two stages (stages M1 and M2) assembles components to the main body of the drawer. The second part of the line (stages M3 and M4) welds the components to the body and performs some final operations including a visual quality check, though parts are not scrapped within the line. Each stage is modeled as a single up — multiple down machine. The data were not provided for confidentiality reasons.

5.1. Analysis of processing times

In this section, data samples from the production log are analyzed with respect to the production rate. Fig. 11 shows boxplots of the cycle times for two different stations in the line.

The boxplots indicate that automation guarantees that each operation is repeatable. The cycle times can then be considered deterministic without introducing any approximation.

[5] Photo courtesy of Cosberg (www.cosberg.com/en/solutions/furniture-accessories/24/).

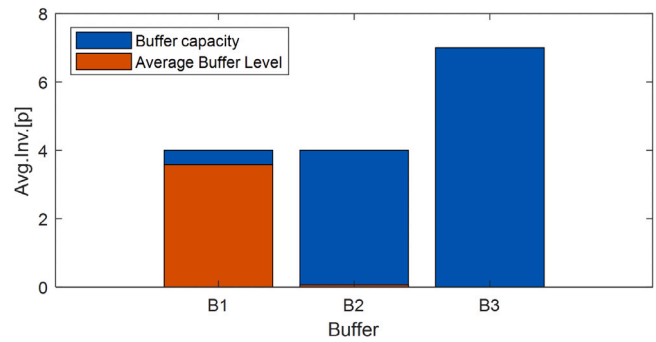


Fig. 12. Average buffer level in the drawers line.

5.2. Performance evaluation and analysis

The $MMTT$ model was used to evaluate drawer line performance. The difference in throughput was estimated to be less than 1.5%. Detailed results for the drawer lines are reported in Table 3.

Each stage was evaluated as Integrated Machine. The steady-state probabilities for the stages were computed. This also included the probability that each stage is limited by other stages in the line. For instance, stage $M1$ is idle, i.e. blocked, because of the downstream stage on 1.09% of the time. However, this blocking probability depends primarily on stages $M2$ and $M3$, since the last stage $M4$ never causes the first stage to be blocked. On the other hand, stage $M3$ can be starved because of upstream stages $M1$ and $M2$ for 35% of the time. It is interesting to notice that the effect of stage $M1$ on $M3$ is higher than the effect of $M2$. Despite the proximity of stage $M2$ to stage $M3$, the reason for limitations of stage $M3$ is mainly due to stage $M1$. This can be traced to the actual dynamics occurring in the system, in particular those related to the repair times, which are higher in stage $M1$ on average than in stage $M2$. Thus, the propagation of limitation is more severe for stage $M1$. Indeed, $M1$ has the highest limiting effect on the other machines. However, if the average buffer level is considered, additional comments can be added. The average buffer level for each buffer compared with the buffer capacity is shown in Fig. 12.

From these results, stage $M2$ appears to be the one with highest inventory level before and lowest inventory level after, which, according to many methods [52], indicates the bottleneck of the line.

Therefore, three main outcomes can be derived from this analysis:

- Stage $M1$ has the most severe effect on the downstream stages. This occurs because of the stoppages propagating along the line.
- Stage $M2$ limits the line performance because it results in the highest inventory level in the previous buffer. This occurs because stage $M2$ represents the bottleneck from the perspective of the cycle time.
- The buffer allocation of the line can be improved, because the largest buffer is placed where it is not required; in fact, it is always empty.

6. Conclusion

In this paper, a novel analytical model is introduced for the performance evaluation of asynchronous and unreliable multi-stage manufacturing lines for fabricating discrete parts. The model is based on a continuous-time Markovian model. Owing to the control mechanisms introduced, the model proved effective in terms of performance evaluation accuracy with respect to a purely continuous model also when small buffer capacity is considered.

The proposed two-level decomposition decouples the buffer-level analysis from the machine-level analysis, by means of lumping and partitioning different Markovian representations. As a consequence, it is

Table 3
Evaluation of the real system with Integrated Machines.

M1			M2		
State $S^{[1]}$	Type	Probability $\Pi(S)$	State $S^{[2]}$	Type	Probability $\Pi(S)$
U	Local	0.9241	U	Local	0.6185
D_1	Local	0.0100	D_1	Local	0.0350
D_2	Local	0.0394	D_2	Local	0.0165
D_3	Local	0.0156	D_3	Local	0.0084
Idle because of M2	Blocking	0.0097	D_4	Local	0.0027
Idle because of M3	Blocking	0.0012	Idle because of M1	Starvation	0.3139
Idle because of M4	Blocking	0.0000	Idle because of M3	Blocking	0.0051
			Idle because of M4	Blocking	0.0000
M3			**M4**		
State $S^{[3]}$	Type	Probability $\Pi(S)$	State $S^{[4]}$	Type	Probability $\Pi(S)$
U	Local	0.6185	U	Local	0.6185
D_1	Local	0.0182	D_1	Local	0.0003
D_2	Local	0.0022	Idle because of M3	Starvation	0.0949
Idle because of M2	Starvation	0.1449	Idle because of M2	Starvation	0.1217
Idle because of M1	Starvation	0.2163	Idle because of M1	Starvation	0.1646
Idle because of M4	Blocking	0.0000			

expected to be more easily generalized for further system architectures, as disassembly systems, as well as for more complex system features, as in-process scrap and rework.

The concept of Integrated Machine is in fact introduced as key element for the two-level decomposition. The Integrated Machine consists in a Markovian meta-model of the entire multi-stage system, centered in a specific perspective, i.e. stage. Thus, the proposed method provides not only the estimates of a wide set of performance measures, but also a set of meta-models which can be used as synthetic computationally-efficient models of the entire multi-stage system. Additionally, these meta-models are explicit, in the sense that there exist explicit relations between the dynamics in a certain stage and its effect on another stage. This is beneficial for the fast identification of critical stages as well as of improvement actions, as shown in the analyzed industrial case study.

The meta-models, i.e. the Integrated Machines, can then be used for further analysis at single-stage level without neglecting the effect at system-level, thus pursuing global optimization rather than local optimization. Hence, ongoing and future research focuses on how to exploit such meta-models for design and operations of complex manufacturing systems. For instance, at single-stage level control policies could be integrated, as maintenance-oriented policies or energy-efficiency policies. In this case, the resulting Integrated Machines could be used as evaluation kernels for reinforcement learning algorithms. Similarly, when configuration analysis is considered, meta-models of the system can be more easily integrated into efficient optimization methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Detailed markovian representation of integrated machine

In this Appendix, additional details are provided for the Markovian representation of Integrated Machines, based on single-up multiple-down representation. For each partition $\mathcal{S}^{[k]}$ and $\mathcal{B}^{[k]}$, two types of states can be distinguished:

- 0^- represents the limiting state in which the upstream pseudo-machine $M^u(k-1)$ with respect to the considered machine $M[k]$ is operational. Similarly, 0^+ represents the limiting state in which the downstream pseudo-machine $M^d(k)$ with respect to the considered machine $M[k]$ is operational. These states are used to model the cyclic production dynamics which have been shown to be relevant to the overall system behavior.
- R^- represents all the other states for upstream remote limitations. Similarly, R^+ represent all the other states for downstream remote limitations. These states are used to model the remote limitations which may occur to the considered machine.

Thus, the transition rate matrix $Q^{[k]}$ of the generic Integrated Machine $M[k]$ can be detailed as follows:

$$Q^{[k]} = \begin{bmatrix} Q_{\mathcal{L}\mathcal{L}} & Q_{\mathcal{L}\mathcal{S}} & Q_{\mathcal{L}\mathcal{B}} \\ Q_{\mathcal{S}\mathcal{L}} & Q_{\mathcal{S}\mathcal{S}} & Q_{\mathcal{S}\mathcal{B}} \\ Q_{\mathcal{B}\mathcal{L}} & Q_{\mathcal{B}\mathcal{S}} & Q_{\mathcal{B}\mathcal{B}} \end{bmatrix} \tag{39}$$

$$= \begin{bmatrix} \begin{bmatrix} Q_{UU} & Q_{UD} \\ Q_{DU} & Q_{DD} \end{bmatrix} & \begin{bmatrix} Q_{U0^-} & Q_{UR^-} \\ Q_{D0^-} & Q_{DR^-} \end{bmatrix} & \begin{bmatrix} Q_{U0^+} & Q_{UR^+} \\ Q_{D0^+} & Q_{DR^+} \end{bmatrix} \\ \begin{bmatrix} Q_{0^-U} & Q_{0^-D} \\ Q_{R^-U} & Q_{R^-D} \end{bmatrix} & \begin{bmatrix} Q_{0^-0^-} & Q_{0^-R^-} \\ Q_{R^-0^-} & Q_{R^-R^-} \end{bmatrix} & \begin{bmatrix} Q_{0^-0^+} & Q_{0^-R^+} \\ Q_{R^-0^+} & Q_{R^-R^+} \end{bmatrix} \\ \begin{bmatrix} Q_{0^+U} & Q_{0^+D} \\ Q_{R^+U} & Q_{R^+D} \end{bmatrix} & \begin{bmatrix} Q_{0^+0^-} & Q_{0^+R^-} \\ Q_{R^+0^-} & Q_{R^+R^-} \end{bmatrix} & \begin{bmatrix} Q_{0^+0^+} & Q_{0^+R^+} \\ Q_{R^+0^+} & Q_{R^+R^+} \end{bmatrix} \end{bmatrix} \tag{40}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & Q_{UD} \\ Q_{DU} & 0 \end{bmatrix} & \begin{bmatrix} Q_{U0^-} & Q_{UR^-} \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} Q_{U0^+} & Q_{UR^+} \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} Q_{0^-U} & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & Q_{0^-R^-} \\ Q_{R^-0^-} & Q_{R^-R^-} \end{bmatrix} & [0] \\ \begin{bmatrix} Q_{0^+U} & 0 \\ 0 & 0 \end{bmatrix} & [0] & \begin{bmatrix} 0 & Q_{0^+R^+} \\ Q_{R^+0^+} & Q_{R^+R^+} \end{bmatrix} \end{bmatrix} \tag{41}$$

The matrix $Q^{[k]}$ has been simplified according to the possible non-zero transition rates.

Given the continuous nature of the model, a machine cannot be contemporary starved and blocked therefore starvation and blocking states are indeed separate states. Moreover, it is impossible to go directly from an upstream limitation to a downstream limitation (or on the way round) without first being back in the operational state, because starvation and blocking depend on the level of the neighboring buffers and the only way to get into blocking or starvation is that machine $M[k]$ produces parts. This is indeed a peculiar dynamics of discrete manufacturing systems.

References

[1] Herps K, Dang Q-V, Martagan T, Adan I. A simulation-based approach to design an automated high-mix low-volume manufacturing system. J Manuf Syst 2022;64:1–18.
[2] Ruane P, Walsh P, Cosgrove J. Development of a digital model and metamodel to improve the performance of an automated manufacturing line. J Manuf Syst 2022;65:538–49.
[3] Bai Y, Tu J, Yang M, Zhang L, Denno P. A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: design, accuracy and robustness. Int J Prod Res 2021;59(13):4072–89.
[4] Magnanini MC, Mastrangelo M, Tolio TA. Hybrid digital modelling of large manufacturing systems to support continuous evolution. CIRP Ann 2022;71(1):389–92.

[5] Li C, Chang Q. Hybrid feedback and reinforcement learning-based control of machine cycle time for a multi-stage production system. J Manuf Syst 2022;65:351–61.
[6] Koren Y, Gu X, Guo W. Choosing the system configuration for high-volume manufacturing. Int J Prod Res 2018;56(1–2):476–90.
[7] Tu J, Zhang L. Performance analysis and optimisation of Bernoulli serial production lines with dynamic real-time bottleneck identification and mitigation. Int J Prod Res 2022;1–17.
[8] Du S, Xu R, Li L. Modeling and analysis of multiproduct multistage manufacturing system for quality improvement. IEEE Trans Syst Man Cybern 2016;48(5):801–20.
[9] Li C, Huang J, Chang Q. Data-enabled permanent production loss analysis for serial production systems with variable cycle time machines. IEEE Robot Autom Lett 2021;6(4):6418–25.
[10] Sun Y, Zhang L. Application of a novel approach of production system modelling, analysis and improvement for small and medium-sized manufacturers: a case study. Int J Prod Res 2022;1–21.
[11] Gu X. Modeling of reconfigurable manufacturing system architecture with geometric machines and in-stage gantries. J Manuf Syst 2022;62:102–13.
[12] Magnanini MC, Tolio TA. Robust improvement planning of automated multi-stage manufacturing systems. In: Selected topics in manufacturing. Springer; 2022, p. 61–75.
[13] Bai Y, Tu J, Yang M, Zhang L, Denno P. A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: design, accuracy and robustness. Int J Prod Res 2020;1–18.
[14] Kang Y, Mathesen L, Pedrielli G, Ju F, Lee LH. Multi-fidelity modeling for analysis and optimization of serial production lines. IEEE Trans Automat Control 2020.
[15] Lee J-H, Li J, Horst JA. Serial production lines with waiting time limits: Bernoulli reliability model. IEEE Trans Eng Manage 2017;65(2):316–29.
[16] Magnanini MC, Tolio T. Restart policies to maximize production quality in mixed continuous-discrete multi-stage systems. CIRP Ann 2020.
[17] Diamantidis A, Lee J-H, Papadopoulos CT, Li J, Heavey C. Performance evaluation of flow lines with non-identical and unreliable parallel machines and finite buffers. Int J Prod Res 2020;58(13):3881–904.
[18] Li C, Chang Q, Xiao G, Arinez J. Integrated process-system modeling and performance analysis for serial production lines. IEEE Robot Autom Lett 2022;7(3):7431–8.
[19] Dallery Y, Gershwin SB. Manufacturing flow line systems: a review of models and analytical results. Queueing Syst 1992;12(1–2):3–94.
[20] Buzacott J, Shanthikumar J. Stochastic models of manufacturing systems. Vol. 4, Prentice Hall Englewood Cliffs, NJ; 1993.
[21] Gershwin SB. Manufacturing systems engineering. Prentice Hall; 1994.
[22] Li J, Meerkov SM. Production systems engineering. Springer Science & Business Media; 2008.
[23] Papadopoulos CT, Li J, O'Kelly ME. A classification and review of timed Markov models of manufacturing systems. Comput Ind Eng 2018.
[24] Jia Z, Zhang L, Arinez J, Xiao G. Performance analysis for serial production lines with Bernoulli machines and real-time WIP-based machine switch-on/off control. Int J Prod Res 2016;54(21):6285–301.
[25] Yan F-Y, Wang J-Q, Li Y, Cui P-H. An improved aggregation method for performance analysis of Bernoulli serial production lines. IEEE Trans Autom Sci Eng 2020;18(1):114–21.
[26] Lee J-H, Zhao C, Li J, Papadopoulos CT. Analysis, design, and control of Bernoulli production lines with waiting time constraints. J Manuf Syst 2018;46:208–20.
[27] Gershwin SB. An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Oper Res 1987;35(2):291–305.
[28] Colledani M, Tolio T. A decomposition method to support the configuration/reconfiguration of production systems. CIRP Ann-Manuf Technol 2005;54(1):441–4.
[29] Helber S. Performance analysis of flow lines with non-linear flow of material. Springer Science; 1999.
[30] Le Bihan H, Dallery Y. A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers. Ann Oper Res 2000;93(1):265–97.
[31] Colledani M, Gershwin SB. A decomposition method for approximate evaluation of continuous flow multi-stage lines with general Markovian machines. Ann Oper Res 2013;209(1):5–40.
[32] David R, Xie X, Dallery Y. Properties of continous models of transfer lines with unreliable machines and finite buffers. IMA J Manag Math 1989;2(4):281–308.
[33] Brandimarte P, Sharifnia A, Von Turkovich B. Continuous flow models of manufacturing systems: a review. CIRP Ann 1996;45(1):441–4.
[34] Suri R, Fu B-R. On using continuous flow lines to model discrete production lines. Discrete Event Dyn Syst 1994;4(2):129–69.
[35] Magnanini MC, Tolio TA. Performance evaluation of asynchronous two-stage manufacturing lines fabricating discrete parts. CIRP J Manuf Sci Technol 2021;33:488–505.
[36] Fu B-R, Shi L, Suri R. Analysis of departure times in discrete and continuous tandem production lines. Discrete Event Dyn Syst 2002;12(2):159–86.
[37] Shi L, Fu B-R, Suri R. Sample path analysis for continuous tandem production lines. Discrete Event Dyn Syst 1999;9(3):211–39.
[38] Xie X, Hennequin S, Mourani I. Perturbation analysis and optimisation of continuous flow transfer lines with delay. Int J Prod Res 2013;51(23–24):7250–69.
[39] Du S, Xu R, Huang D, Yao X. Markov modeling and analysis of multi-stage manufacturing systems with remote quality information feedback. Comput Ind Eng 2015;88:13–25.
[40] Askin RG, Hanumantha GJ. Queueing network models for analysis of nonstationary manufacturing systems. Int J Prod Res 2018;56(1–2):22–42.
[41] Shin J, Grosbard D, Morrison JR, Kalir A. Decomposition without aggregation for performance approximation in queueing network models of semiconductor manufacturing. Int J Prod Res 2019;1–14.
[42] Matta A, Simone F. Analysis of two-machine lines with finite buffer, operation-dependent and time-dependent failure modes. Int J Prod Res 2016;54(6):1850–62.
[43] Tolio TA, Ratti A. Performance evaluation of two-machine lines with generalized thresholds. Int J Prod Res 2018;56(1–2):926–49.
[44] Grassmann WK. Computational probability. Vol. 24, Springer Science & Business Media; 2013.
[45] Golub GH, Van Loan CF. Matrix computations. Vol. 3, JHU Press; 2012.
[46] Lanczos C. Linear systems in self-ad joint form. Amer Math Monthly 1958;65(9):665–79.
[47] Wynn P. Acceleration techniques for iterated vector and matrix problems. Math Comp 1962;16(79):301–22.
[48] Dallery Y, David R, Xie X-L. An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. IIE Trans 1988;20(3):280–3.
[49] Welch PD. The statistical analysis of simulation results. In: The computer performance modeling handbook. Vol. 22, 1983, p. 268–328.
[50] Tolio T. Performance evaluation of two-machine line with multiple up and down states and finite buffer capacity. In: Eighth conference on stochastic models of manufacturing and service operations. Turkey, Kusadasi, 28 May-2 June 2011; 2011, p. 117–27.
[51] Colledani M, Magnanini MC, Tolio T. Impact of opportunistic maintenance on manufacturing system performance. CIRP Ann 2018.
[52] Chiang S-Y, Kuo C-T, Meerkov S. C-Bottlenecks in serial production lines: identification and application. Math Probl Eng 2001;7(6):543–78.