Transient Analysis of Asynchronous Markovian Production Lines by Quasi Product Form

Alessio Angius
Computer Science Department, Universitá di Torino
Via Pessinetto 12 Turin, Italy
angius@di.unito.it

Marcello Colledani
Mechanical Engineering Department, Politecnico di Milano
Via la Masa 1 Milan, Italy
marcello.colledani@polimi.it

ABSTRACT

Analytical models have been extensively used to analyze the performance of production systems. Due to their enormous state spaces, the analysis of such models is very often approximated and limited to stationary first moment performance measures. However, in presence of randomness, the system performance observed in the short-medium run can be significantly different from the long term average performance. Moreover, modern systems often never reach the steady state due to the continuous product and process modifications that take place over time. Therefore, the analysis of higher order system performance measures in the short run has recently attracted more and more attention both by scientist and practitioners. This paper proposes an approximate model to analyze the performance of asynchronous production lines with finite capacity buffers. The transient probabilities of such model are analyzed by assuming a *quasi product form*. This assumption simplifies the dependency structure of the model and leads to a relatively small set of ordinary differential equations (ODE) that can be used to compute an approximation of the transient probabilities. The accuracy of this approximate method is studied by comparing the numerical results with those provided by simulation.

1. INTRODUCTION

Asynchronous unreliable production lines are a specific class of serial manufacturing systems where machines process material at non-identical processing rates and are prone to failures. Such systems are asynchronous in the sense that machines are allowed to start and finish their operations at different time instants. They are commonly found in several manufacturing contexts, among which semiconductor, automotive, food and white goods industries. In real settings, processing stages are typically separated by finite capacity buffers, temporarily storing the inventory or work in progress (WIP) flowing in the system. An exact analysis of these systems is almost impossible due to the state space explosion observed even for relatively small lines. For this reason, in the last two decades approximate analytical methods [13] have been developed to evaluate the performance of these systems and to support production managers in the phases of design/redesign, operation and control of these systems. These approaches typically apply decomposition techniques to generate smaller sub-systems, easier to be analyzed with exact techniques. Then, the connection between these subsystems is captured by means of specific non-linear decomposition equations. An application of these methods in the automotive industry can be found in [12]. Although they are very accurate in the evaluation of steady state first order performance measures of the system (average system throughput, average work in progress, and average lead time) at the state of the art higher order performance measures cannot be obtained by these approaches, except for very small systems with less then 5-6 machines [23]. Moreover, the transient analysis of asynchronous lines is still an almost unexplored area. Available models are focused on very small systems formed by two machines, with identical processing times, and one buffer ([20]).

However, there is industrial evidence that production variability due to random disturbances causes the observed production rate to be different from its average value, especially in the short run. Moreover, modern manufacturing systems undergo frequent reconfigurations due to continuous product and process modifications and improvements. Relevant industrial questions like " What is the expected time to produce a given lot?" or "What is the expected number of parts produced by the system in the next 2 hours, given the current system state and configuration?" remain unsolved. Motivated by these industrial needs, the objective of this paper is to propose an analytical method for the evaluation of higher order performance measures of asynchronous manufacturing systems, considering the transient period.

In detail, in this paper we apply an approximate transient analysis technique that maintains the state space of the model. The technique is based on assuming that the transient probabilities can be written in a *quasi product form* (QPF). The QPF assumption leads to a memory efficient description of the transient probabilities and determines a relatively small set of ODEs that provides an approximation of the transient probabilities. Such approach was proposed for the first time by Angius and Horvath in [2] to analyze reaction networks. These networks are characterized by the presence of infinite server-like mechanisms combined with switches. Subsequently, the same authors proposed guidelines to use QPF in the context open queuing networks hav-

ValueTools'13, December 10 – 12 2013, Turin, Italy

VALUETOOLS 2013, December 10-12, Torino, Italy

DOI 10.4108/icst.valuetools.2013.254392

ing finite server stations and *Stochastic Petri Nets* [3, 1]. It is also important to mention that a similar approach based on Bayesian network was presented earlier in [11] to compute passage time distributions of closed networks.

Instead, in this work we apply and experiment the approach in the context of production lines with finite number of stages.

The remainder of the paper is organized as follows. In the next section the reference model and its assumptions will be described. In section 3 the developed method is explained in detail. In section 4 we motivate the chosen decomposition and present the corresponding ODEs system. Numerical results are provided in section 5 comparing the QPF solution with the solution provided by simulation and by decomposition-based methods. In section 6 the conclusions are drawn and the future research steps are highlighted.

2. PRODUCTION LINE MODEL

The production line under analysis is composed of a set of K work centers (machines from now-on) separated by $K-1$ temporary storage areas having discrete finite dimension (buffers), $C_1, C_2, \ldots, C_{K-1}$, in which material flows in a fixed first in-first out (FIFO) sequence visiting each machine and each buffer only once.

Raw items start their processing at the first machine and then move forward through the system by receiving additional processing at each machine. Whenever a machine processes an item, it reduces the level of the buffer placed before it (upstream or source buffer) and increases the level of the next buffer (downstream or destination buffer). Finished products leaves the system after the last machine of the line. Figure 1 depicts a graphical representation of such system architecture where the squares represent the machines, and the circles represent the buffers.

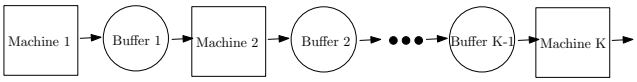

Figure 1: Machine-Buffer-Machine scheme

Every machine processes the items in sequence as long as the upstream buffer contains parts to work on and the downstream buffer has space to store the parts after the processing. The upstream of the first machine and the downstream of the last machine are considered as infinite; thus, we consider the situation where raw items are always available and there is always enough space to store finished products. Deliberate inactivity is not considered but machines are unreliable; thus, they are subject to failure. After a failure, the machine remains failed for an amount of time that corresponds to the time required to repair the failure.

Failures are the main cause of randomness of the number of items stored in the buffers; e.g. if the processing of the nth machine is stopped by a failure but the $(n+1)$th is operative then the nth buffer will tend to get empty, causing the starvation of the $(n+1)$th machine. Viceversa, the longer the time when $(n+1)$th machine is not operational while the nth is working and the more likely it is that the number of items stored in the nth buffer reaches its maximum capacity, thus causing the blocking of the nth machine. Therefore, the failure of a machine may cause the inactivity of its neighboring machines that, in turn, could generate the propagation of blockages and starvations respectively in the upstream and downstream portions of line. The blocking phenomenon can be mitigated by expanding the size of buffers in such a way that an adequate supply of items can be stored to be available when failures occur. On the other hand, larger buffers entail higher costs and may introduce delays into the production line. Thus, an oversizing of buffers capacities is undesired. Although the model presented in this paper can be coupled with an optimization algorithm to find the minimum buffer capacities to satisfy a given production rate requirement, the solution of this problem is out of the scope of this paper.

The state of the system is described by a set of random variables $X = \{M_1, B_1, \ldots, B_{K-1}, M_K\}$ where: M_i exists in $\{On, Off\}$, thus representing if machine M_i is operational or not, and B_i takes values in the interval $\{0, \ldots, C_i\}$, thus describing the level of the ith buffer.

We assume that:

- Every machine process one item at a time. The item is considered to be in the upstream buffer until its processing is completed, then the item is moved into the downstream buffer instantaneously.
- Machines works with *Blocking After Service* (BAS) policy. Thus machines gets idle immediately after the completion of a processing if the upstream buffer is empty or the destination buffer is full. Similarly, a machine is resumed immediately after the ending of the cause of the blocking or of the starvation, i.e. if the upstream buffer was empty and the preceding machine puts a new item in the buffer or if the destination buffer was full and the subsequent machine removes an item from it[1].
- Machine are subject to operation dependent failures that means that a machine can fail only if it is processing an item (not starved or blocked).

On the basis of the assumptions above, every machine is able to change the state of the system in three different ways; let $\{m_1 \ldots, b_{i-1}, m_i, b_i, \ldots, m_K\}$ be a given state of the system then:

1. if $m_i = On$, $b_{i-1} > 0$ and $b_i < C_i$ the system can move to $\{m_1, \ldots, b_{i-1}, Off, b_i, \ldots, m_K\}$ because of a failure of machine M_i,
2. if $m_i = Off$ the state $\{m_1, \ldots, b_{i-1}, On, b_i, \ldots, m_K\}$ can be reached because machine M_i returns operational,
3. if $m_i = On$, $b_{i-1} > 0$ and $b_i < C_i$ the system can move to $\{m_1, \ldots, b_{i-1} - 1, m_i, b_i + 1, \ldots, m_K\}$ because machine M_i ends the processing of an item,

where, in order to maintain short the conditional statements, we assumed b_0 and b_K as fixed to infinity and zero, respectively.

Another convention that will be adopted to simplify the notation is the use of x as placeholder for the set that represents a given state of the system. We make distinction

[1]Although we consider only the BAS policy, different blocking policies can be used as well (see for example those listed in [5]).

among two different states by applying primes. The same notation is used for their elements. Thus, given two sets x and x' representing two possible state of the system, m_i belongs to the set x whereas m_i' belongs to x' and both of them represent a possible state of machine M_i. We denote with $x' \xrightarrow{f,i} x$, $x' \xrightarrow{r,i} x$, $x' \xrightarrow{p,i} x$ the fact that if machine M_i fails (f), gets repaired (r), or end the processing of an item (p) then the set x' becomes equal to the set x.

We assume that every change of state occurs according to Markovian distributions such that the underlying stochastic process is a *Continuous time Markov chain* (CTMC). Therefore, in the most simple scenario, failures, repairs and processing times are associated with exponential distributions. In this situation the probability to find the system in a given state is associated with a single Chapman-Kolmogorov equation whose construction is straightforward by following the transition diagram of the system. Unfortunately, the exponential distribution is frequently a bad candidate for representing actual distributions of real systems mostly because "real" distribution have coefficients of variation far from 1 [17]. We circumvented this problem by considering phase-type distributions (PH) to model the dynamics of the processing times, failure and repair times of the machines. Let us remind that a random variable T is PH distributed if its cumulative distribution function (cdf) corresponds to the time till absorption of a CTMC given a pre-fixed initial distribution. More formally:

DEFINITION 1. *Let T be a random variable with cdf*

$$Pr\{T < t\} = 1 - \alpha e^{\mathbf{\Lambda} t}\mathbb{1}$$

and probability density function (pdf)

$$\frac{dPr\{T < t\}}{dt} = \alpha e^{\mathbf{\Lambda} t} a$$

where $\alpha = [a_i]_{1\times n}$ is an initial row vector of size n, $\mathbf{\Lambda} = [\lambda_{i,i}]_{n\times n}$ is a square matrix of size $n \times n$, $\mathbb{1}$ is the column vector of ones of size n and a is a row vector equal to $-\mathbf{\Lambda}\mathbb{1}$. We say that T is phase type distributed with representation $(\alpha, \mathbf{\Lambda})$, (or simply $PH(\alpha, \mathbf{\Lambda})$ distributed) if the following properties are satisifed:

1. *the vector a sums to one and all its entries are greater or equal to zero.*
2. *the entries of Λ are such that $\lambda_{i,i} < 0$, $\lambda_{i,j} \forall i \neq j$, and $\sum_{k=0}^{n} \lambda_{i,k} \leq 0$,*
3. *Λ is not singular.*

The more detailed structure of PH distributions allows the fitting of general distributions by matching their higher moments (see for example [16, 8, 7, 6]). Moreover, since PH distributions are represented through CTMCs, they can be used to extends exponential distributed models by preserving the use of traditional numerical methods for Markov chains.

Due to the introduction of PH distributions the structure of the underlying CTMC gets slightly more complicated because every state of the system is expanded in order to consider also the detailed information about the aging of those transitions that eventually will cause the change of the state. Figuratively, the process moves on two levels: the state of the system and the joint phases (the joint of the states associated with the PH distributions). On the basis of this structure memory-based policies, such as the resume of an interrupted activity, can be modelled. For sake of simplicity, we do not consider any of them; hence, when failures and processing are resumed after they have been interrupted, their distributions are resetted. Nevertheless, possible extensions of the model in these directions are straightforward by considering the works described in [22, 15]. Table 1 introduces the symbols that will be used in the remainder of this paper.

Matrices and vectors	
$\mathbb{1}_n$	column vector of ones composed of n entries
$\mathbf{I}_n$	identity matrix composed of $n \times n$ entries
$(\alpha_f^i, \mathbf{\Lambda}_i^f)$	representation of the failures of M_i
n_i^f	number of phases describing PH$(\alpha_i^f, \mathbf{\Lambda}_i^f)$
a_i^f	$-\mathbf{\Lambda}_i^f \mathbb{1}_{n_i^f}$
$(\alpha_i^r, \mathbf{\Lambda}_i^r)$	representation of the repairs of the M_i
n_i^r	number of phases describing PH$(\alpha_i^r, \mathbf{\Lambda}_i^r)$
a_i^r	$-\mathbf{\Lambda}_i^r \mathbb{1}_{n_i^r}$
$(\alpha_i^p, \mathbf{\Lambda}_i^p)$	representation of the processings of the M_i
n_i^p	number of phases describing PH$(\alpha_i^p, \mathbf{\Lambda}_i^p)$
a_i^p	$-\mathbf{\Lambda}_i^p \mathbb{1}_{n_i^p}$
$\otimes$	Kronecker product
$\oplus$	Kronecker sum
Scalars	
$v(t,x)$	distribution of the enabled PHs in x at time t
$\pi(t,x,l)$	prob. to observe state x and phases l at time t
$\mathbf{\Lambda}_i^z(k,k)$	entry (k,k) of matrix $\mathbf{\Lambda}_i^z$, $z \in \{f,r,p\}$
$\alpha_i^z(k)$	kth entry of vector α_i^z, $z \in \{f,r,p\}$
$a_i^z(k)$	kth entry of vector a_i^z, $z \in \{f,r,p\}$

Table 1: Symbols adopted in the paper.

PROPOSITION 1. *Let $v(t,x)$ be the vector describing the joint phases of the enabled PH distributions in the state x at time t; then its transient behaviour fulfills the following differential equation*

$$\begin{aligned}
\frac{dv(t,x)}{dt} &= v(t,x)\bigoplus_{i=1}^{K}\mathbf{S}_i(x) \\
&+ \sum_{i\,:\,x' \xrightarrow{f,i} x} v(t,x')\Big(\bigotimes_{j=1}^{i-1}\mathbf{A}_j(x)\otimes\mathbf{F}_i(x)\bigotimes_{j=i+1}^{K}\mathbf{A}_j(x)\Big) \\
&+ \sum_{i\,:\,x' \xrightarrow{r,i} x} v(t,x')\Big(\bigotimes_{j=1}^{i-1}\mathbf{A}_j(x)\otimes\mathbf{R}_i(x)\bigotimes_{j=i+1}^{K}\mathbf{A}_j(x)\Big) \\
&+ \sum_{i\,:\,x' \xrightarrow{p,i} x} v(t,x')\Big(\bigotimes_{j=1}^{i-2}\mathbf{A}_j(x)\otimes\mathbf{W}_{i-1}^{d}(x)\otimes\mathbf{P}_i(x) \\
&\otimes\mathbf{W}_{i+1}^{u}(x)\otimes\bigotimes_{j=i+2}^{K}\mathbf{A}_j(x)\Big) \qquad (1)
\end{aligned}$$

where

$$\mathbf{F}_i(x) = \begin{cases} a_i^f \otimes \alpha_i^r \otimes \mathbb{1}_{n_i^p} & m_i = Off \wedge b_{i-1} > 0 \wedge b_i < C_i \\ 0 & otherwise \end{cases}$$

$$\mathbf{R}_i(x) = \begin{cases} \alpha_i^f \otimes a_i^r \otimes \alpha_i^p & m_i = On \wedge b_{i-1} > 0 \wedge b_i < C_i \\ a_i^r & m_i = On \wedge (b_{i-1} = 0 \vee b_i = C_i) \\ 0 & otherwise \end{cases}$$

$$\mathbf{P}_i(x) = \begin{cases} \mathbf{I}_{n_i^f} \otimes (a_i^p \alpha_i^p) & m_i = On \wedge b_{i-1} > 0 \wedge b_i < C_i \\ a_i^p \otimes \mathbb{1}_{n_i^f} & m_i = On \wedge (b_{i-1} = 0 \vee b_i = C_i) \\ 0 & otherwise \end{cases}$$

$$\mathbf{S}_i(x) = \begin{cases} \mathbf{\Lambda}_i^f \oplus \mathbf{\Lambda}_i^p & m_i = On \wedge (b_{i-1} > 0 \wedge b_i < C_i) \\ 1 & m_i = On \wedge (b_{i-1} = 0 \vee b_i = C_i) \\ \mathbf{\Lambda}_i^r & m_i = Off \end{cases}$$

$$\mathbf{A}_i(x) = \begin{cases} \mathbf{I}_{n_i^f} \otimes \mathbf{I}_{n_i^p} & m_i = On \wedge b_{i-1} > 0 \wedge b_i < C_i \\ \mathbf{I}_{n_i^r} & m_i = Off \\ 1 & otherwise \end{cases}$$

$$\mathbf{W}_i^d(x) = \begin{cases} 1 & i = 0 \\ \alpha_i^f \otimes \alpha_i^p & i > 0 \wedge m_i = On \wedge b_{i-1} > 0 \wedge b_i = C_i - 1 \\ \mathbf{A}_i(x) & otherwise \end{cases}$$

$$\mathbf{W}_i^u(x) = \begin{cases} 1 & i = K + 1 \\ \alpha_i^f \otimes \alpha_i^p & i \leq K \wedge m_i = On \wedge b_{i-1} = 1 \wedge b_i < C_i \\ \mathbf{A}_i(x) & otherwise. \end{cases} \quad (2)$$

PROOF. We prove Theorem 1 by studying the dynamics of a single entry of the vector $v(t,x)$.

Unfortunately, this requires the introduction of further notation. We will denote the phases of a PH distribution by using latin letters whose superscripts indicate if they are associated with a failure (f), a repair (r), or a processing (p), whereas the subscripts refer to the machines; e.g. l_i^f refers to a phase of the failure of machine M_i. Furthermore, we assume them as equal to zero when their distributions are disabled, e.g. l_i^r is zero every time that machine M_i is On (no repair can take place). A given joint of the phases is considered as a set denoted with the same letter of its element, e.g. $l = \{\dots, l_i^f, l_i^r, l_i^p, \dots\}$. The entries of matrices $\mathbf{\Lambda}$ and vectors a will be indicated with parenthesis and they will be null if at least one of the arguments is equal to zero. The entries of the vectors α will be indicated in the same way but they are equal to one when the argument is zero.

The evolution over time of the probability to observe the system in the state x while the phases of the distributions are equal to l, denoted with $\pi(t,x,l)$, is determined by the following rates.

- The rates with which the system can change at least one between the joint of the phases l and the state x; these rates corresponds to the elements on the diagonal of the matrices $\mathbf{\Lambda}$.
- The rates with which the process can move to l from $e \neq l$ without changing the state of the system; these rates corresponds to the element out of the diagonals of the matrices $\mathbf{\Lambda}$.
- The rates with which machines fail, thus causing the activation of the PH distribution associated with the repair. Given the ith machine, these rates correspond to the products between $\alpha(l_i^r)$ and $a_i^f(k)$ for all $1 \leq k \leq n_i^f$.
- The rates with which the machines get repaired, thus causing the activation of the PH distributions associated with the failure and the processing (if the source buffer is not empty and the destination buffer is not full). Given the ith machine, these rates correspond to the products among $\alpha(l_i^f)$, $\alpha(l_i^p)$ and $a_i^r(k)$ for all $1 \leq k \leq n_i^r$.
- The rates with which the machines finish the processing of an item, thus causing the restart of a new processing (if the source buffer is not empty and the destination buffer is not full) and the possible awakening of the preceding machine (subsequent machine) if the source buffer was full (the destination buffer was empty). Given the ith machine, these rates are equal to the products among $\alpha(l_i^p)$,$w_{i-1}^d(x,l)$ $w_{i+1}^u(x,l)$ and $a_i^p(k)$ for all $1 \leq k \leq n_i^p$, where $w_{i-1}^d(x,l)$ ($w_{i+1}^u(x,l)$) is equal to $\alpha(l_{i-1}^p)$ ($\alpha(l_{i+1}^p)$) only if x is such that machine M_{i-1} (M_{i+1}) has been woken up by the end of the processing, and zero otherwise.

By weighting the rates described above with the proper probabilities and applying algebraic steps, we can write

$$\begin{aligned} \frac{d\pi(t,x,l)}{dt} = \sum_{i=1}^{K} \Bigg[& \\ & \pi(t,x,l)\Big(\mathbf{\Lambda}_i^f(l_i^f,l_i^f) + \mathbf{\Lambda}_i^r(l_i^r,l_i^r) + \mathbf{\Lambda}_i^p(l_i^p,l_i^p)\Big) \quad (3) \\ & + \sum_{(x,e)\xrightarrow{f,i}(x,l)} \pi(t,x,e)\mathbf{\Lambda}_i^f(e_i^f,l_i^f) \\ & + \sum_{(x,e)\xrightarrow{r,i}(x,l)} \pi(t,x,e)\mathbf{\Lambda}_i^r(e_i^r,l_i^r) \\ & + \sum_{(x,e)\xrightarrow{p,i}(x,l)} \pi(t,x,e)\mathbf{\Lambda}_i^p(e_i^p,l_i^p) \\ & + \sum_{(x',e)\xrightarrow{f,i}(x,l)} \pi(t,x',e)a_i^f(e_i^f)\alpha_i(l_i^r) \\ & + \sum_{(x',e)\xrightarrow{r,i}(x,l)} \pi(t,x',e)a_i^r(e_i^r)\alpha_i^f(l_i^f)\alpha_i^p(l_i^p) \\ & + \sum_{(x',e)\xrightarrow{p,i}(x,l)} \pi(t,x',e)a_i^p(e_i^r)\alpha_i^f(l_i^f)\alpha_i^p(l_i^p) \\ & \times w_{i-1}^d(x,l)w_{i+1}^u(x,l)\Bigg]. \quad (4) \end{aligned}$$

that is the scalar representation of the entries of the vector $v(t,x)$ whose dynamics have been described in equations (1) and (2). This completes the proof. □

3. QUASI PRODUCT FORM APPROACH

The model presented in Section 2 can, at least in principle, be treated by standard transient and steady-state analysis techniques developed for Markov chains [18]. In practice, this is possible only for very small systems since the number of states composing the state space grows exponentially with the number of machines and buffers that compose the production line. In this section, we present an approximated

approach, called *quasi product form*, that allows to decompose the state space in subsets that can be analysed one at a time by reducing the complexity of the analysis.

Let X be a CTMC describing the joint probabilities of N random variables, $\{X_1, X_2, \ldots, X_N\}$, that interact among each other. The quasi product form approach is based on the assumption that there exist sets of random variables (r.v.'s) whose conditional probabilities depend only on a subset of other r.v.'s instead of all the rest. For example, if we assume that the conditional probabilities of X_1 and X_2 depend only on X_3, X_4 and X_5 then we can write

$$Pr\{X_1 = x_1, X_2 = x_2 \mid X_3 = x_3, X_4 = x_4, ..., X_M = x_N\} = Pr\{X_1 = x_1, X_2 = x_2 \mid X_3 = x_3, X_4 = x_4, X_5 = x_5\}$$

A set of assumptions like the one expressed above allows us to decompose the probability $Pr\{X_1 = x_1, X_2 = x_2, ..., X_N = x_N\}$ into a product that is conveniently described by a *directed acyclic graph* (DAG), denoted by $\mathcal{G}$.

The set of the nodes of the graph is denoted by $\mathcal{V}$ and a given node, $v \in \mathcal{V}$, represents a subset of the r.v's. The index set of the r.v's represented by node v is denoted by $I(v)$. The set $\mathcal{V}$ must be such that it provides a partitioning of all the r.v.'s, i.e., $\cup_{v\in\mathcal{V}} I(v) = \{1, 2, ..., N\}$ and $\forall v_1, v_2 \in \mathcal{V}, v_1 \neq v_2 : I(v_1) \cap I(v_2) = \emptyset$.

The set of edges of the DAG, denoted by $\mathcal{E}$, provides the assumed dependency structure of the transient probabilities. Specifically, if $e = (u, v) \in \mathcal{E}$ then the conditional probability of the random variables in v *depends* on those that are present in u.

The set of r.v.'s present in the predecessors of v will be denoted by $P(v)$, i.e., $P(v) = \cup_{u:(u,v)\in\mathcal{E}} I(u)$. The conditional probability of the places in $I(v)$ is independent of those that are not present in $P(v)$, i.e.,

$$Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in\{1,2,...,N\}\setminus I(v)}(X_j = x_j)\} = Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in P(v)}(X_j = x_j)\}$$

where $\wedge$ denotes conjunction. By considering every node of the DAG, the probability of a given state of the CTMC, $\{x_1, ..., x_N\}$, can be written as

$$Pr\{\wedge_{1,2,...,N}(X_i = x_i)\} = \prod_{v\in\mathcal{V}} Pr\{\wedge_{i\in I(v)}(X_i = x_i) \mid \wedge_{j\in P(v)}(X_j = x_j)\} = \prod_{v\in\mathcal{V}} \frac{Pr\{\wedge_{i\in Q(v)}(X_i = x_i)\}}{Pr\{\wedge_{j\in P(v)}(X_j = x_j)\}} \quad (5)$$

where we applied the notation $Q(v) = I(v) \cup P(v)$. Note that the acyclicity of the graph guarantees that the unicity of the product in equation 5. In order to compute the transient probabilities based on the quasi product form assumption expressed by the DAG $\mathcal{G}$, the quantities appearing in (5) are needed. Since $P(v) \subseteq Q(v)$, the quantities in the denominator can be computed simply by appropriate summing of the quantities in the numerator. The quantities in the numerator can instead be computed by the proper summation of the Chapman-Kolmogorov equations , i.e.,

$$\frac{dPr\{\wedge_{i\in Q(v)}(X_i = y_i)\}}{dt} = \frac{d}{dt} \sum_{\substack{\{y_1,...,y_M\}: \\ k\in Q(v), y_k = x_k}} Pr\{\wedge_{\{1,2,...,N\}}(X_i = x_i)\} \quad (6)$$

The equations associated with a node v corresponds to a marginal distribution of the random variables belonging to the set $Q(v)$. In [1] it has been shown that every marginal distribution $Q(v)$ can be interpreted as a *time inhomogeneous* CTMC whose transient behaviour is determined by two type of transitions: those that are completely determined by the random variables belonging to $Q(v)$ and those that arises from the interaction with other marginal distributions. The first are considered in an exact way, whereas the second are considered as, possibly conditional, inhomogeneous Poisson processes. An exhaustive description of the structure of equation (6) in case of general models can be found in [1]. It is omitted since it is beyond the scope of this paper.

4. DECOMPOSITION STRATEGY

The advantages that can be obtained by using the quasi product form decomposition are strongly related to the choice of the DAG. Indeed, there is a trade-off between the accuracy and efficiency of the approximation that is determined by the computational effort that is required to analyze the model.

For most of the cases, the more dependencies are preserved and the more accurate is the approximation [2]. However, the preservation of several dependencies leads to the increasing of the number of states composing the marginal distributions. As a consequence, the number of equations to be considered could easily explode, eventually reaching the same complexity of the original problem. Furthermore, the gain obtained in terms of accuracy could be negligible with respect to that provided by a DAG that considers only few dependencies. In other words, we could overestimate the correlations among the random variables and, as a consequence, analyze more equations than what needed to achieve the desired accuracy. In this scenarios, given a maximum number of equations that we are able (or available) to analyze, the challenge is to find the DAG that minimizes the number of equations by providing the best possible accuracy. Unfortunately, due to the complexity of the transient analysis this can be done only through heuristics. Thus there is not guarantee that the selected DAG will be the optimal.

Neverthless, on the basis of the works [14, 19, 10, 9], where exact transient product form conditions are studied, satisfying policies have been found for queuing networks with finite server stations [3] and reaction networks of modulated infinite servers. Both the heuristics are based on the fact that the correlations among random variables that are cause or consequence of blocking must be preserved as much as possible. This is a consequence of the fact that two random variables cannot enjoy transient product form if the state of one causes the blocking of the other a.k.a. transient product form holds only in case of queuing networks composed omly of infinite server networks.

Since the system under investigation is a chain of modulated single server stations, the strategy we adopt to find an effective decomposition of the production line model is inspired by the same principle. By observing the structure and the dynamics of the model, it appears evident how: i) the random variables describing the buffers are completely passive components, whose states change on the basis of the

[2]Considering only the scenarios in which the number of dependencies is expanded in a rational way.

events generated by the machines; ii) a machine can produce only if two conditions are simultaneously verified on the buffers to which it is directly connected, i.e. the upstream buffer is not empty and the downstream buffer is not full; iii) the blocking of a machine is not a direct consequence of the states of the other machines but is the side-effect of the interaction between the other machines and the buffers.

In line with these observations, we penalize the detailed consideration of the correlations among the machines in favour of those between the machines and the buffers to which they are connected. Thus, a possible DAG that fits this purpose corresponds to that reported in Figure 2 where F_i, R_i and P_i

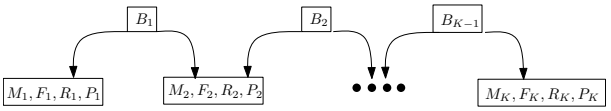

Figure 2: The DAG adopted to decomposed the production line model

denote the random variables describing the PH distributions of failures, repairs and processing of the ith machine.

The selected decomposition strategy implies that the marginal distribution of every machine is associated with a set Q that is composed, at the most, of six random variables that are B_{i-1},M_i,B_i, F_i, R_i and P_i, $1 \leq i \leq K$.

Let us denote the probability to observe $B_j = b_{j-1}$, $M_j = m_j$, $B_j = b_j$, $F_j = l_j^f$, $R_j = l_j^r$ and $P_j = l_j^p$ at time with $\pi(t, b_{j-1}, m_j, b_j, l_j^f, l_j^r, l_j^p)$, $2 \leq j \leq K-1$,. Then, by applying equation (6) to equation (4) and exploiting basic probability rules, we have that:

$$\begin{aligned}
&\frac{d\pi(t, b_{j-1}, m_j, b_j, l_j^f, l_j^r, l_j^p)}{dt} = \pi(t, b_{j-1}, m_j, b_j, l_j^f, l_j^r, l_j^p)\Big(\\
&\mathbf{\Lambda}_j^f(l_j^f, l_j^f) + \mathbf{\Lambda}_j^r(l_j^r, l_j^r) + \mathbf{\Lambda}_i^p(l_j^p, l_j^p) + \\
&-E[a_{j-1}^p | b_{j-1}] - E[a_{j+1}^p | b_j]\Big) + \\
&\sum_{k=1}^{n_j^f} \pi(t, b_{j-1}, m_j, b_j, k, l_j^r, l_j^p)\mathbf{\Lambda}_j^f(k, l_j^f) + \\
&\sum_{k=1}^{n_j^r} \pi(t, b_{j-1}, m_j, b_j, l_j^f, k, l_j^p)\mathbf{\Lambda}_i^r(k, l_j^r) + \\
&\sum_{k=1}^{n_j^p} \pi(t, b_{j-1}, m_j, b_j, l_j^f, l_j^r, k)\mathbf{\Lambda}_i^p(k, l_j^p) + \\
&\sum_{k=1}^{n_j^f} \pi(t, b_{j-1}, Off, b_j, k, l_j^r, l_j^p)a_j^f(k)\alpha_i(l_j^r) + \\
&\sum_{k=1}^{n_j^r} \pi(t, b_{j-1}, On, b_j, l_j^f, k, l_j^p)a_j^r(k)\alpha_i^f(l_i^f)\alpha_i^p(l_i^p) + \\
&\sum_{k=1}^{n_j^p} \pi(t, b_{j-1}+1, m_j, b_j-1, l_j^f, l_j^r, k)a_i^p(k)\alpha_i^f(l_i^f)\alpha_i^p(l_i^p) + \\
&\pi(t, b_{j-1}-1, m_j, b_j, l_j^f, l_j^r, l_j^p)E[a_{j-1}^p | b_{j-1}-1] + \\
&\pi(t, b_{j-1}, m_j, b_j+1, l_j^f, l_j^r, l_j^p)E[a_{j-1}^p | b_j+1] \qquad (7)
\end{aligned}$$

where the second to last and the last term have to be substituted with (8) and (9), respectively, if b_{j-1} ,m_j, b_j are such that the machine is restored by a production of one of the neighbours[3]

$$\pi(t, b_{j-1}-1, m_j, b_j, l_j^f, 0, 0)E[a_{j-1}^p | b_{j-1}-1]\alpha_j^f(l_j^f)\alpha_j^p(l_j^p) \qquad (8)$$

$$\pi(t, b_{j-1}, m_j, b_j-1, l_j^f, 0, 0)E[a_{j-1}^p | b_{j+1}+1]\alpha_j^f(l_j^f)\alpha_j^p(l_j^p). \qquad (9)$$

The value $E[a_{j-1}^p | b_{j-1}]$ is the expected rate with which machine M_{j-1} changes the level of buffer B_{j-1} from b_{j-1} to $b_{j-1}+1$ at time t. Symmetrically, $E[a_{j+1}^p | b_j]$ is the expected rate with which machine M_{j+1} decreases B_j, the level of buffer b_j, by one at time t. These rates are time dependent and corresponds to:

$$E[a_{j-1}^p | b_{j-1}] = \sum_{i=1}^{n_{j-1}^p} \sum_{j=1}^{n_{j-1}^f} \sum_{k=1}^{C_{j-2}} \frac{\pi(t, k, On, b_{j-1}, i, 0, j)}{\pi(t, b_{j-1})} a_{j-1}^p(i) \qquad (10)$$

$$E[a_{j+1}^p | b_j] = \sum_{i=1}^{n_{j+1}^p} \sum_{j=1}^{n_{j+1}^f} \sum_{k=1}^{C_{j+1}} \frac{\pi(t, k, On, b_j, i, 0, j)}{\pi(t, b_j)} a_{j+1}^p(i) \qquad (11)$$

where $\pi(t, b_{j-1})$ and $\pi(t, b_j)$ denote to the probability to observe the buffer B_{j-1} and B_j equal to b_{j-1} and b_j at time t respectively. Equations (10) and (11) can be easily computed from the marginal distributions describing the transient behaviour of machine M_{j-1} and M_{j+1}. The boundary cases with $j = 1$ and $j = K$ have the same structure of equation (7) but they do not have ingoing and outoging items respectively.

The resulting ODEs system can be interpreted as a set of interacting time inhomogeneous CTMCs in which a chain describes the transient behaviour of a machine and the buffers to which is connected. Given a chain, let's say the ith, the events whose intensity depends on the machine M_i are described in an exact way whereas those generated by the processing of the adjacent machines are considered from the point of view of their expected production rate at time t conditioned on the buffer level. It follows that numerical solution techniques developed for time inhomogeneous Markov chains, like the one proposed in [4], can be applied to calculate the transient probabilities.

The number of equations in the original set of ODEs given in (1) grows as the cartesian product between the active PH distributions and the level of the buffers, i.e. $\prod_{i=1}^{K-1}(C_i + 1) \times \prod_{i=1}^{K}(n_i^f \times n_i^p + n_i^r + 1)$. Given the DAG depicted in Figure 2, the number of equations describing the QPF (7) grows as $\sum_{i=1}^{K}((C_{i-1}+1) \times (C_i+1) \times (n_i^f \times n_i^p + n_i^r + 1))$ where $C_0 = C_k = 0$. Thus, the complexity of the analysis of the decomposed system grows linearly with the number of machines composing the production lines.

5. NUMERICAL RESULTS

In this section, we apply the QPF approximation to models with different number of machines and various settings

[3]The derivation of equation (7) is too long to be reported in the paper but is available at the URL www.di.unito.it/~angius/proofUnMach.pdf

of the parameters. In the first bulk of experiments we consider different production line lengths, with all the buffers having the same capacity. Moreover, we assume exponentially distributed transitions in order to compare the results of the approximation with the decomposition method described in [21]. The second set of tests considers, instead, PH distributed machines with different buffers settings. For all the experiments, we provide a comparison between the results obtained by the approximation and the Monte Carlo simulation of the original model. All the scenarios consider fully operative machines and empty buffers as initial state.

The algorithm based on the QPF assumption has been implemented in JAVA by using the odeToJava package3 for the solution of the system of ODEs. In particular, we applied the "explicit Runge-Kutta triple" solver of the package. The accuracy of this method is determined by two parameters, called relative and absolute tolerance, and we set these values to 10^{-8} and 10^{-6} , respectively. The reported run times refer to these settings and by choosing less restrictive values the computation times can be significantly reduced, by about one order of magnitude. All the experiments have been performed on an Intel Core i7 with 8Gb of RAM.

5.1 Machines with Exponentially Distributed Failures, Repairs and Processing Times.

The aim of the first set of experiments is to show how the length of the considered production line affects the accuracy of the approximation. In order to do this, we considered three systems composed of 5, 10 and 20 identical machines, respectively. All the machines have mean time to failure equal to 100, mean time to repair equal to 10 and mean processing time equal to 1.

Figure 3 depicts the mean and the variance of the number of items present on the last buffer for all the three systems; it is possible to observe that, although the error increases with the number of machines, the approximated curves are still very close to the one obtained by simulation along the entire time interval. Also, it is interesting to notice that the settling time of the transient period is larger for longer lines the for smaller lines and the proposed method well captures this phenomenon. Therefore, the transient analysis results to be even more significant in lines with a considerable number of stages, as those found in real manufacturing settings.

As second scenario, we keep the same failure and repair times but we increase the mean processing time to 0.2, i.e. five times faster than in the previous experiment. This test is motivated by previous numerical evidences that showed that the more the interactions between the marginals are fast with respect to the other rates of the ODEs system, the more the approximation looses accuracy ([2]). Figure 4 confirms this theoretical result by showing larger differences between approximated and original curves than in the previous test. This difference is particularly marked for system composed of twenty machines, where the approximated curves flatten around the time interval $(20, 50)$ and then re-start to follow the same slope of the original curves. Although these very fast machines are not frequently found in real manufacturing systems (except in water bottling lines, for example), these results will drive the future research activities aiming at improving the method accuracy in these specific cases. Despite these inaccuracies, even in the worst scenario the results obtained through the quasi product form provide a good picture of the behaviour of the system. This fact is

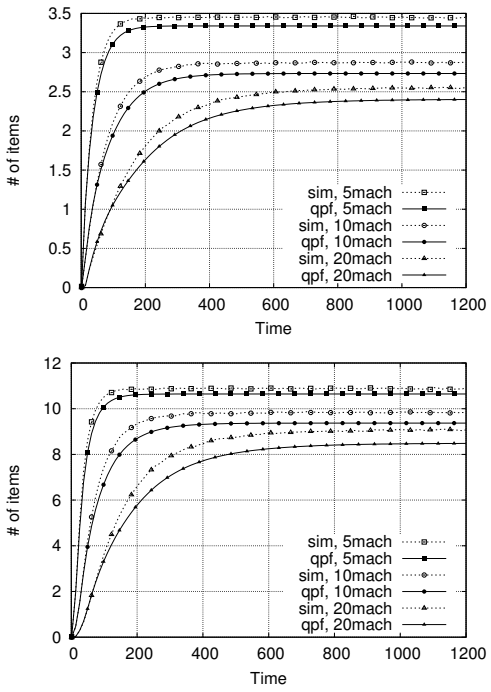

Figure 3: Mean (top) and variance (bottom) of the number of items present in the last buffer as function of the time with average processing times equal to 1 for different production line length.

confirmed by Figure 5 that depicts the comparison between the distribution of the number of items present in the 10th and the last buffer for the system composed of twenty machines. From Figure 5 it is possible to observe that the shape of both the distributions is well-represented by the approximation, although the error on the probability to observe the 10th buffer with 1 or 9 items is not negligible.

# mach	avg pr.	sim	qpf	dec
5	1	0.67161	0.67459	0.6694
10	1	0.61384	0.61918	0.6080
20	1	0.57612	0.58451	0.5752
5	0.2	3.09455	3.15537	3.0512
10	0.2	2.55960	2.68510	2.4940
20	0.2	2.15163	2.15253	2.1970

Table 2: Average throughput by observing the processes around steady state.

In the worst considered scenario (system composed of 20 machines) the ODEs system is composed of 4000 equations and the integration of 2000 time units required around 19 minutes. For the same case, the original ODEs system is composed of $\approx 2^{19} \times 11^{20}$ equation. It is evident that the computation of such system is not feasible by using traditional techniques. The trajectories corresponding to the original CTMC has been generated on the basis of 500000 of simulation runs, each of them required on average 0.0693 seconds in the worst case.

Although a thorough analysis of the advantages of the

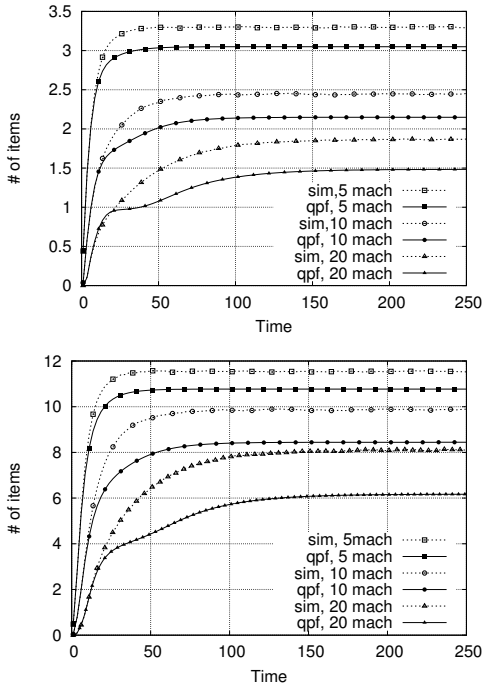

Figure 4: Mean (top) and variance (bottom) of the number of items present in the last buffer as function of the time with average processing times equal to 0.2 for different production line length.

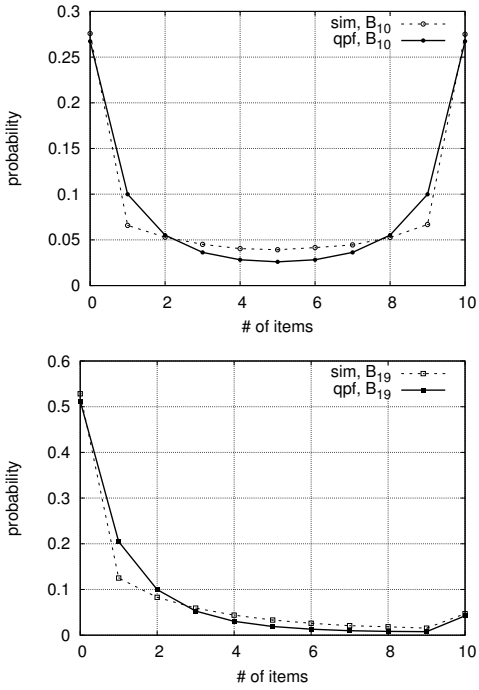

Figure 5: Distribution of the number of items present in the 10th and last buffer at time 300 with average processing times equal to 0.2 for a line composed of twenty machines.

# mach	avg pr.	sim	qpf	dec
5	1	20.008	20.001	20.780
10	1	44.991	44.999	45.256
20	1	95.010	94.970	92.120
5	0.2	19.910	19.999	20.360
10	0.2	44.991	44.999	45.170
20	0.2	95.052	94.991	96.133

Table 3: Average WIP by observing the processes around steady state.

proposed approach in regard to the decomposition method proposed in [21] is out the scope of this paper, we provide a preliminary comparison of the two methods.

The decomposition method proposed by Levantesi et al. splits the original line with K machines into a set of $K-1$ two-machine lines, each one being associated with a buffer of the original line.

The general idea of the decomposition method is to build the two-machine lines (building blocks) in such a way that the ith building block mimics, in the steady-state, the material flow dynamics through the corresponding buffer in the original line. The upstream and downstream machines in two machine line i represent the dynamics of the material flow respectively entering and leaving the buffer B_i. In this sense, they are called "pseudo-machine" as they do not have a direct connection to the corresponding machines in the original line, M_i and M_{i+1}.

To achieve this goal, additional failure and repair probabilities are properly assigned to the two machines in each building block, in order to mimic the propagation of starvation and blocking events throughout the system. In detail, the starvation of the ith machine is conveniently modelled into the ith building block by adding "remote" failures to the upstream pseudo-machine, a.k.a. additional states that are introduced to model those situations in which machine M_i is not operative because it is starved. Similarly, the blocking of the $i+1$th machine is conveniently modelled into the ith building block by adding "remote" failures to the downstream pseudo-machine, a.k.a. additional states that are introduced to model those situations in which machine M_{i+1} is not operative because it is blocked. The parameters of these pseudo-machines are iteratively updated by considering the performance of the neighboring sub-systems by decomposition equations, until convergence is met.

In Tables 2 and 3 we provide a coarse comparison of the steady-state average throughput and work in progress (WIP) for the two set of systems of the previous experiments as obtained by simulation, by the proposed approximate method and by the decomposition method. As it can be noticed, for all the experiments the estimated measures are very close among each other: the proposed approximate method is slightly less accurate than the decomposition method proposed in [21] in terms of average throughput estimates but is more accurate in terms of WIP estimate. In all the cases, the errors against simulation are very small. Due to the fact that QPF method provides information about both transient and steady state, these results highlight the practical applicability of the proposed approach in the analysis of real production lines and motivate future research.

5.2 Machines with PH Distributed Failures, Repairs and Processing times.

The second set of experiments introduces PH distributions by considering the processing times distributed according to an Erlang-5 distribution with mean 1 and the repair times distributed according to an hyper-exponential distribution with mean 10 and coefficient of variation equal to 10. The distribution of failures remains unchanged and the number of machines in the line is equal to 20 for all the reported experiments.

The first two tests aim at analyzing how the different buffer capacities affect the quasi product form approximation. We considered two different buffer settings, as reported in Table 4. It can be noticed that the buffer size distributions are reversed in the two cases, in the sense that the buffers with maximum capacity of the first corresponds to those with minimum capacity in the second and so on for the second to last, etc. Figure 6 shows the mean and the variance of the number of items present in the last buffer, for both system configurations. It is possible to observe that the mean values are well-approximated in both the cases, whereas the variance is underestimated for the second case.

#	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
1	5	5	5	7	7	7	7	10	10	15
2	15	15	15	10	10	10	10	7	7	5
#	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	
1	10	10	7	7	7	7	5	5	5	
2	7	7	10	10	10	10	15	15	15	

Table 4: Buffer capacities for the first the first two experiments with PH distributed machines.

In the last experiment, we considered a line composed of machines with different mean times to failures and mean processing times. In detail, for every machine, the mean values of these two distributions are randomly generated. In practice, this has been obtained for every machine by changing the mean values of their distributions of a random quantity that could not exceed the 15 per cent of the previous value. The buffer capacities have been set to 10 and the repair distributions were considered as in the two previous cases. In Figure 7 it is possible to observe that also in this very general case the approximation provides accurate results both for the mean and the variance of the number of items on buffer 19. As additional proof of the accuracy of the approximation, we provide in Figure 8 the comparison between the original and approximated joint distribution of the amount of items in buffer 9 and 10. Figure 8 shows that the approximation provides accurate results even if it tends to slightly overestimate regions where the probability mass is null.

By considering PH distributions the number of equation that have to be considered to carry on the computation of the quasi product form increases; in the worst case (the second case reported in Table 4) the largest marginal that was composed of $16^2 \times (5 + 2 + 1)$ equations and the overall ODEs system was composed of 18.034 equations. The integration of 2000 time units required about two and a half hours, Monte Carlo simulation required instead 0.1011 seconds per run on average. For all the tests, 500000 of simulation runs have been performed to generate the trajectories corresponding to the original CTMCs.

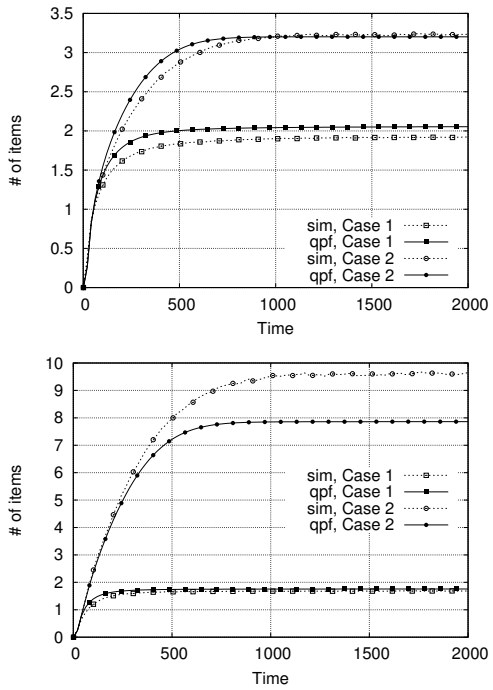

Figure 6: Mean (top) and variance (bottom) of the number of items present in the last buffer as function of the time with average processing times equal to 1, by considering different buffer settings.

6. CONCLUSIONS

This paper presents a new approximate method for the analysis of asynchronous production lines with finite capacity buffers and unreliable machines characterized by generally distributed failure, repair and processing events, modeled as phase type distributions. The proposed method is based on a quasi product form approximation of the state probabilities that is valid both in the transient and in the steady state of the system. This feature allows us to evaluate reasonably long production lines in very short time, with very accurate performance estimates. Therefore, the developed tool can be integrated within a global system design/ redesign platform to quantitatively estimate in advance the effect of potential alternative system configurations. Moreover, the transient analysis can be used to predict, in the short run, the output statistics of the system and to properly control the system performance in very dynamic production contexts, characterized by continuous product, process and system changes. Future development will be devoted to the analysis of more complex systems architectures involving multiple part types, multiple alternative routings of the parts in the system and multiple parallel machines.

7. ACKNOWLEDGMENTS

This work has been supported in part by project "AMALFI - Advanced Methodologies for the Analysis and management of Future Internet" sponsored by Universitå di Torino and

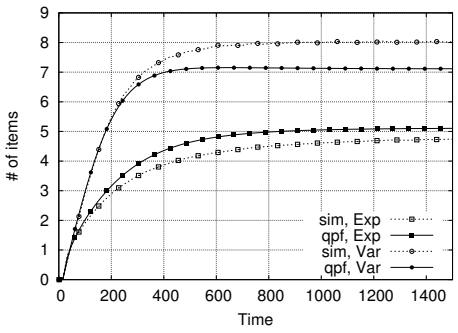

Figure 7: Mean and variance of the number of items present in the last buffer as function of time by considering a production line composed of machines with different failure and processing times.

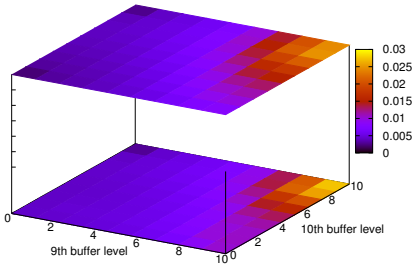

Figure 8: Comparison between the exact joint distribution of the 9th and 10th buffer content (bottom surface) and the approximated one (top surface) at time 2000 by considering a production line composed of machines with different failure and processing times.

Compagnia di San Paolo.

8. REFERENCES

[1] A. Angius. *Handling large state spaces in transient analysis of Markovian processes*. PhD thesis, Universita' degli studi di Torino, 2013.

[2] A. Angius, A. Horváth, and V. Wolf. Quasi product form approximation for Markov models of reaction networks. *Trans. on Comp. Systems Biology*, 7625(XIV), 2012.

[3] A. Angius, A. Horváth, and V. Wolf. Approximate Transient Analysis of Queuing Networks by Quasi Product Forms. In *Analytical and Stochastic Modeling Techniques and Applications*, volume 7984 of *LNCS*, pages 22–36. 2013.

[4] M. Arns, P. Buchholz, and A. Panchenko. On the Numerical Analysis of Inhomogeneous Continuous-Time Markov Chains. *Informs Journal on Computing*, 22(3):416–432, 2010.

[5] S. Balsamo, P. G. Harrison, and A. Marin. A unifying approach to product-forms in networks with finite capacity constraints. *SIGMETRICS Perform. Eval. Rev.*, 38(1):25–36, June 2010.

[6] A. Bobbio, A. Horvath, M. Scarpa, and M. Telek. Acyclic discrete phase type distributions: properties and a parameter estimation algorithm. *Perf. Eval.*, 54(1):1–32, 2003.

[7] A. Bobbio, A. Horváth, and M. Telek. The scale factor: a new degree of freedom in phase-type approximation. *Perf. Eval.*, 56(1-4):121–144, 2004.

[8] A. Bobbio, A. Horvath, and M. Telek. Matching three moments with minimal acyclic phase type distributions. *Stoch. Models*, 21:303–326, 2005.

[9] R. J. Boucherie. *Product-form in queueing networks*. PhD thesis, 1992. Th. : stochastic operations research.

[10] R. J. Boucherie and P. Taylor. Transient product form distributions in queueing networks. *Discr. Event Dyn. Syst.*, 3:375–396, 1993.

[11] G. Casale. Approximating passage time distributions in queueing models by Bayesian expansion. *Perform. Eval.*, 67(11):1076–1091, 2010.

[12] M. Colledani, M. Ekvall, T. Lundholm, P. Moriggi, A. Polato, and T. Tolio. Analytical methods to support continuous improvements at Scania. *Inter. Journal of Prod. Res.*, 48:1913–1945, 2010.

[13] Y. Dallery and S. B. Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems*, 12(1-2):3–94, 1992.

[14] J. M. Harrison and A. J. Lemoine. A note on networks of infinite-server queues. *J. Appl. Probab.*, 18(2):561–567, 1981.

[15] A. Horváth and M. Telek. Time Domain Analysis of Non-Markovian Stochastic Petri Nets with PRI Transitions. *IEEE Trans. Software Eng.*, 28(10):933–943.

[16] A. Horvath and M. Telek. Matching more than three moments with acyclic phase type distributions. *Stoch. Models*, 23:167–194, 2007.

[17] R. Inman. Empirical Evaluation of Exponential and Independence Assumptions in Queueing Models of Manufacturing Systems. *Production Operations Management*, 8:409–432, 1999.

[18] V. G. Kulkarni. *Modeling and analysis of stochastic systems*. Chapman & Hall, Ltd., London, UK, UK, 1995.

[19] W. A. Massey and W. Whitt. Networks of infinite-server queues with nonstationary Poisson input. *Queueing Systems*, 13:183–250, 1993.

[20] S. Meerkov, N. Shimkin, and L. Zhang. Transient Behavior of Two-Machine Geometric Production Lines. *Trans. on Automatic Control*, 55(2):453–458, feb. 2010.

[21] R. Levantesi, A. Matta, and T. Tolio. Performance Evaluation of Production Lines with Random Processing Times, Multiple Failure Modes and Finite Buffer Capacity. *Analysis and Modeling of Manufacturing Systems*, 60:201–219, 2003.

[22] M. Scarpa. *Non Markovian Stochastic Petri Nets with concurrent generally distributed transitions*. Ph.d. thesis, Universita' degli studi di Torino.

[23] B. Tan. Modeling and Analysis of Output Variability in Discrete Material Flow Production Systems. *International Series in Oper. Res. and Manag. Science*, 192(2):287–311, feb. 2013.