Production Systems with Cycle Overrun: Modeling, Analysis, Improvability, and Bottlenecks

Yongsoon Eun[a], Kang Liu[b] and Semyon M. Meerkov[b*]

[a]Department of Information and Communication Engineering, DGIST, Daegu 42988, Republic of Korea; [b]Department of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI, USA.

ABSTRACT

Production systems literature usually attributes throughput losses to two reasons: unreliable equipment and random part processing time (also referred to as machine cycle time). In practice, however, one more reason for throughput losses is observed: cycle overrun. The specificity of cycle overrun is that not all parts may require more time than allotted by a fixed cycle time and, if the overrun does occur, its duration is coupled with the part processing time: typically, it is a fraction or a small multiple of the cycle time. This paper is indented to develop methods for analysis and improvement of production systems with unreliable machines and cycle overrun. Specifically, it introduces a mathematical model of an unreliable machine with cycle overrun, develops its simplified version, explores the efficacy of machines' improvability by reducing either downtime or cycle overrun, and discusses the issue of bottleneck identification. The results obtained are illustrated by a case study based on an automotive transmission case machining line.

KEYWORDS

Production systems; Unreliable machines; Cycle overrun; Machine improvability; Bottleneck analysis; Continuous improvement.

1. Introduction

1.1. *Background and Motivation*

Throughput losses in production systems are usually attributed to two reasons: unreliable equipment and random part processing time (also referred to as the machine cycle time). The former is typically considered in system-theoretic literature devoted to production systems, where the machine up- and downtime are assumed to be random variables (often, exponentially distributed), while the cycle time is viewed as a constant (see for instance, Viswanadham and Narahari (1992), Askin and Standridge (1993), Gershwin (1994), Altiok (1997), Li and Meerkov (2009), Curry and Feldman (2010)). The latter is typically used in queuing-theoretic literature, where the processing time is assumed to be a random variable (often, also exponential), while up- and downtimes are not explicitly considered and may be viewed as "embedded" in the random processing time (see, for instance, Buzacott and Shanthikumar (1993), Papadopolous et al. (1993), Papadopoulos et al. (2009)).

In practice, however, one more reason for throughput losses is observed: cycle overrun. This term implies that the part processing time, τ, which is supposed to be constant, may, in fact, require additional time (i.e., overrun), τ_{OR}, leading to the total

*Corresponding author. Email: smm@umich.edu

machine processing time given by

$$\tau_{total} = \tau + \tau_{OR}. \tag{1}$$

Such situations occur, for instance, in automated operations with a constant cycle time, τ, and manual loading/unloading operations, which may have a random component in their duration. This scenario takes place in numerous machining and welding operations. Another scenario is typical in assembly operations, where a fixed cycle time, τ, is imposed by operational conveyors, and the overrun, τ_{OR}, is enabled by push-buttons, offering the operator a possibility to occasionally stop the conveyor in order to complete the job with the desired quality. This scenario takes place, for example, in automotive paint shops and general assembly.

Two main features characterize the cycle overrun. The first one is that it may or may not take place at every cycle time; this implies that overruns occur with a certain probability. The second feature is that, given that the overrun occurs, the conditional pdf of its duration is related to the part processing time, τ. Indeed, in most cases the overrun duration is either a fraction or a small multiple of τ. These features, exacerbated by the fact that the machines with cycle overrun may have equipment-dependent up- and downtimes, make the queuing-theoretic approach inapplicable to systems with cycle overrun. The system-theoretic approach is not applicable as well: while it does consider machine reliability models in terms of up- and downtime, the cycle time is assumed to be constant.

Given that the current literature offers no analytical methods for analysis and improvement of production systems with unreliable machines and cycle overrun and taking into account that these systems are often encountered in practice, developing such methods is of importance. This is carried out in the current paper.

1.2. *Paper Contribution*

Specific novel results reported here are:

- A mathematical model of an unreliable machine with cycle overrun is developed.
- A simplified version of this model is proposed, which enables analytical performance investigation.
- The relative effectiveness of a stand-alone machine throughput improvement by reducing its average downtime vs. its average overrun is investigated.
- The effect of cycle overrun on the performance of serial production line is analyzed.
- The bottleneck identification and throughput improvability in production systems with cycle overrun is investigated, and effectiveness of throughput improvement by downtime reduction vs. cycle overrun reduction is analyzed.
- The results obtained are illustrated by a case study based on an automotive transmission case machining line.

1.3. *Related Literature*

Concluding this section, it should be pointed out that, although the literature offers no analytical methods for analysis and improvement of systems with unreliable machines and cycle overrun, some of the related issues have been discussed in manufacturing and automation engineering literature. Specifically, Morrison and Martin (2007) developed

practical methods for approximating random cycle time of manufacturing systems modeled by a G/G/M-queue. Nadarajah and Kotz (2008) provided the cycle time distribution formula to characterize the cycle underrun and overrun, where the cycle time was modeled as a sum of production busy time and idle time of Pareto and gamma distributions. Kuo, Chien, and Chen (2011) proposed to use neural networks to exploit the production data and tool data of the semiconductor production systems, in order to predict and reduce the production cycle time. Millstein and Martinich (2014) developed Takt Time Grouping method to implement kanban-flow manufacturing in a production process with cycle underrun and overrun, where the variations of cycle time were due to manual operation and set-up time randomness. Kacar, Mönch, and Uzsoy (2016) et al. presented methods in non-integer linear programming to model the cycle time variation in production planning problems. Larco et al. (2017) provided methods to estimate the warehouse workers' discomfort and optimize the job assignment, in order to prevent long cycle overrun realizations. Casalino et al. (2019) proposed a scheduling method for human-robot collaborative assembly based on the cycle time duration data collected at runtime, adapting to the cycle time underrun and overrun of manufacturing processes. Ben-Ammar, Bettayeb, and Dolgui (2020) studied the integrated production planning and quality control strategies for serial production systems with machines having variable probability distributions of the cycle time. Roshani et al. (2020) proposed a hybrid adaptive neighborhood search approach to minimize cycle time variability in multi-sided assembly lines. Touzani et al. (2021) proposed methods for multi-robot task sequencing and automatic path planning to reduce cycle times of automotive production lines.

1.4. *Paper Outline*

The outline of this paper is as follows: Section 2 presents a mathematical model of an unreliable machine with cycle overrun. Section 3 investigates two avenues for reducing this model to that considered in the system-theoretic approach: either by embedding the average overrun time into the machine's downtime or into machine's cycle time. In Section 4, analysis of performance improvement of stand-alone machines with cycle overrun is carried out. In Section 5, a method for performance analysis of serial lines with unreliable machines and cycle overrun is investigated. In Section 6, a bottleneck identification technique for serial lines with unreliable machines and cycle overrun is discussed and the issue of improvability is investigated. A case study is described in Section 7. Finally, the conclusions and topics for future research are given in Section 8. The proofs are included in the Appendix. The list of abbreviations and notations is placed after Section 8.

2. Mathematical Model of Unreliable Machines with Cycle Overrun

This model is defined by the following three groups of parameters/assumptions:

(a) Nominal parameters:

- *Machine cycle time* (τ) – the nominal time necessary to process a part by a machine. The term "nominal" is used to imply that the machine operates in the ideal regime, e.g., with no overruns. In large volume manufacturing, τ is practically always constant or almost constant (i.e., random, but with a small coefficient of variation). If manual

loading and unloading operations are involved, their nominal durations are included in τ.

- *Machine capacity* (c) – the nominal number of parts a stand-alone machine produces per unit of time in the ideal regime, e.g., without breakdowns and cycle overruns. If the unit of time is an hour and the cycle time is in seconds, the machine capacity is

$$c = \frac{3600}{\tau} \text{ parts/hour.} \tag{2}$$

(b) Reliability assumption:

- *Exponential reliability model* – machine breakdown and repair rates, λ and μ are constant, implying that up- and downtime of the machine are distributed exponentially with parameters λ and μ, respectively. The inverses of λ and μ are average up- and downtime, T_{up} and T_{down}.

While the above machine characteristics are widely used in system-theoretic literature, the ones below are novel.

(c) Cycle overrun parameters/assumptions:

- *Overrun probability* (p_{OR}) – the probability that a cycle has an overrun. The complementary probability, $1 - p_{OR}$, is the probability that this cycle does not have an overrun.

- *Overrun distribution* ($f_{OR}(t)$) – the (conditional) pdf of the overrun duration, given that the cycle has an overrun. This distribution is assumed to be exponential with the expected value denoted as T_{OR}. To reflect the practical meaning of the overrun, it is assumed that T_{OR} is either a fraction or a small multiple of τ. Specifically, we assume that $T_{OR} = k_{OR}\tau$, where $k_{OR} \in (0, 2]$.

□

Based on the above, the conditional and unconditional pdf's of the overrun as well as its unconditional expected value (T_{OR}^{uc}) are:

$$f_{OR}(t) = \frac{1}{k\tau}\exp(-\frac{t}{k\tau}), \quad t \geq 0, \tag{3}$$

$$f_{OR}^{uc}(t) = p_{OR}\left(\frac{1}{k\tau}\exp(-\frac{t}{k\tau})\right) + (1 - p_{OR})\delta(t), \quad t \geq 0, \tag{4}$$

$$T_{OR}^{uc} = p_{OR}k_{OR}\tau = p_{OR}T_{OR}. \tag{5}$$

Thus, according to the above model, an exponential unreliable machine with cycle overrun is defined by five independent parameters $\{\tau, T_{up}, T_{down}, p_{OR}, k_{OR}\}$.

Note that model (a)-(c) can be extended to non-exponential distributions of uptime, downtime, and cycle overrun; initial results in this direction are mentioned in Section 8. Note also that, according to the above formulation, τ_{OR} in expression (1) is exactly T_{OR}^{uc}.

3. Simplified Mathematical Model of Unreliable Machines with Cycle Overrun

In the system-theoretic literature, a machine is usually characterized by three independent parameters $\{\tau, T_{up}, T_{down}\}$. They are used to evaluate machine's efficiency, e, and stand-alone throughput, SAT, as follows:

$$\begin{aligned} e &= \frac{T_{up}}{T_{up} + T_{down}}, \\ SAT &= \frac{3600}{\tau} \frac{T_{up}}{T_{up} + T_{down}} \text{ parts/hour.} \end{aligned} \tag{6}$$

To simplify the description and performance analysis of a machine with cycle overrun, two avenues are possible: either embedding the unconditional expected value of the overrun, $p_{OR}T_{OR}$, into T_{down} or into τ. The former leads to

$$\begin{aligned} e_1 &= \frac{T_{up}}{T_{up} + T_{down} + p_{OR}T_{OR}}, \\ SAT_1 &= \frac{3600}{\tau} \frac{T_{up}}{T_{up} + T_{down} + p_{OR}T_{OR}} \text{ parts/hour.} \end{aligned} \tag{7}$$

The latter results in

$$\begin{aligned} e_2 = e &= \frac{T_{up}}{T_{up} + T_{down}}, \\ SAT_2 &= \frac{3600}{\tau + p_{OR}T_{OR}} \frac{T_{up}}{T_{up} + T_{down}} \text{ parts/hour.} \end{aligned} \tag{8}$$

Which one of these simplifications is more precise from the point of view of approximating the stand-alone throughput (SAT)?

To answer this question, we evaluate SAT of the exact stochastic model of Section 2 and compare it with (7) and (8).

Theorem 3.1. *The stand-alone throughput of an unreliable machine with cycle overrun, defined by assumptions (a)-(c), is given by*

$$SAT_{exact} = \frac{3600}{\tau + p_{OR}T_{OR}} \frac{T_{up}}{T_{up} + T_{down}} \text{ parts/hour.} \tag{9}$$

Proof. See the Appendix. □

In other words, $SAT_{exact} = SAT_2$, i.e., no error is introduced by embedding overruns into the machine cycle time.

To quantify the accuracy of SAT_1, consider an example of a machine defined by $\tau = 1$, $T_{up} = 8$, $T_{down} = 2$, $p_{OR} \in \{0.1, 0.2, \ldots, 0.9\}$, $T_{OR} = 0.5\tau$, and calculate SAT_1 and SAT_{exact}. The result is shown in Figure 1. As one can see, the error, defined by

$$\epsilon_1 = \frac{|SAT_{exact} - SAT_1|}{SAT_{exact}} \cdot 100\%, \tag{10}$$

varies as a function of p_{OR} between 4% and 27%.

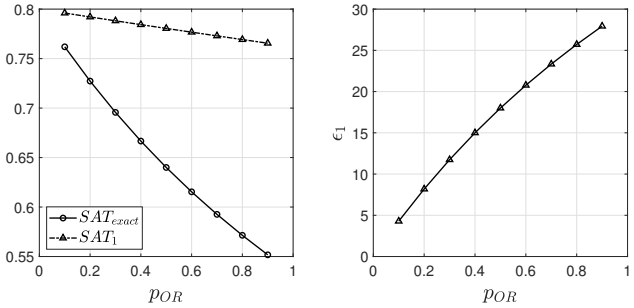


Figure 1. Accuracy of stand-alone throughput evaluation with cycle overrun embedded into downtime.

Based on the above, we conclude that from the point of view of stand-alone throughput, embedding the overrun into the cycle time is preferable. Thus, we define the simplified model of a machine with cycle overrun by a triple $\{\tau + p_{OR}T_{OR}, T_{up}, T_{down}\}$. Similar to $\{\tau, T_{up}, T_{down}\}$, which is the parametric model of a machine without overruns, $\{\tau + p_{OR}T_{OR}, T_{up}, T_{down}\}$ is referred to as the *simplified parametric model of unreliable machine with cycle overrun.*

4. Improvability of Stand-alone Unreliable Machine with Cycle Overrun

Production losses of an unreliable machine with cycle overrun can be decreased by either decreasing its downtime or cycle overrun. Which one of these options is preferable?

To formalize this question, consider a simplified model of unreliable machine defined by $\{\tau + p_{OR}T_{OR}, T_{up}, T_{down}\}$. Assume that its downtime is reduced to become rT_{down}, where the $r \in (0, 1)$ is the *reduction coefficient.* In this case, the machine is characterized by $\{\tau + p_{OR}T_{OR}, T_{up}, rT_{down}\}$ and referred to as *downtime-reduced machine.* Alternatively, assume that the unconditional mean of the overrun is reduced by the same fraction. This results in a machine defined by $\{\tau + rp_{OR}T_{OR}, T_{up}, T_{down}\}$ and is referred to as *overrun-reduced machine.* The statement below specifies which of these machines has a larger SAT.

Theorem 4.1. *For any value of the reduction coefficient, if*

$$\frac{T_{down}}{T_{up}} < \frac{p_{OR}T_{OR}}{\tau}, \tag{11}$$

the overrun-reduced machine has a larger SAT than the downtime-reduced machine. If this inequality is reversed, the downtime-reduced machine is more productive than the overrun-reduced one.

Proof. See the Appendix. □

Condition (11), which we refer to as the *SAT Improvability Indicator*, can be interpreted as follows: Small $\frac{T_{down}}{T_{up}}$ implies that the efficiency of a machine if no overruns are taking into account, is high. Thus, in this situation, the best way of SAT improvement is by decreasing the overruns. On the other hand, if $\frac{T_{down}}{T_{up}}$ is large, the overruns play a

minor role in systems' performance, and the best way to increase SAT is by decreasing machine's downtime. In fact, our research in systems with cycle overrun has been motivated by a continuous improvement project at the underbody assembly system of an automotive assembly plant, where no useful results have been obtained until the cycle overruns were taken into account. By the way, in that project, the overrun time has been embedded into the downtime, which, as it is follows from Section 3, is not the best way to approach the problem.

5. Performance Analysis of Serial Lines with Unreliable Machines and Cycle Overrun Based on their Simplified Parametric Model

While the simplified parametric model of an unreliable machine with cycle overrun precisely predicts its performance in terms of the stand-alone throughput, in a multi-machine production system with finite buffers this may not be the case. Therefore, in this section we investigate the accuracy of serial lines performance evaluation using the simplified parametric model of Section 3 viz-a-viz the exact stochastic model of Section 2. In addition to the throughput (TP), we investigate the accuracy of work-in-process (WIP) and probabilities of blockages (BL) and starvations (ST). First we address the case of two-machine lines (where closed formulas for all performance metrics are available in Li and Meerkov (2009)) and then the general case of ($M > 2$)-machines (using the aggregation procedure of Bai et al. (2020)). In both cases, the accuracy of using the simplified model is evaluated in terms of the errors defined by:

$$\begin{aligned}
\epsilon_{TP} &= \frac{|TP - TP_{sim}|}{TP_{sim}} \cdot 100\%, \\
\epsilon_{WIP} &= \frac{1}{M-1} \sum_{i=1}^{M-1} \frac{|WIP_i - WIP_{i,sim}|}{N_i} \cdot 100\%, \\
\epsilon_{BL} &= \frac{1}{M-1} \sum_{i=1}^{M-1} |BL_i - BL_{i,sim}|, \\
\epsilon_{ST} &= \frac{1}{M-1} \sum_{i=2}^{M} |ST_i - ST_{i,sim}|,
\end{aligned} \tag{12}$$

where the symbols with subscript 'sim' refer to the performance metrics evaluated by simulations, and the symbols without the subscript refer to the same performance metrics calculated analytically (either by formulas or by aggregation). Note that in the case of two-machine lines the summation signs in (12) should be omitted.

5.1. *Two-machine Lines*

In the two-machine case, we generate 100 lines, with machine and buffer parameters selected randomly and equiprobably from the following sets:

$T_{down,i} \in [3, 10]$, $e_i \in [0.6, 0.95]$, $c_i \in [1, 2]$, $p_{OR,i} \in [0, 1]$, $T_{OR} = k_{OR,i}\tau_i$, where $k_{OR,i} \in [0.2, 2]$, $i = 1, 2$; $N = \lceil h \max(c_1 T_{down,1}, c_2 T_{down,2}) \rceil$, where $h \in [2, 4]$, denotes the level of buffering, which protects a machine against job losses during the adjacent machine's downtime.

The analytical calculations for each of these lines have been carried out using expressions (11.13)-(11.17) of Li and Meerkov (2009). The results obtained are summarized in Tables 1. Based on these results, we conclude that the simplified parametric machine model is acceptable for two-machine systems evaluation.

Table 1. Accuracy of performance metrics evaluation using the simplified parametric model in two-machine lines.

(a) Accuracy of TP evaluation.

	Mean values
TP_{sim}	40.7941
TP	40.8795
ϵ_{TP}	0.29%

(b) Accuracy of WIP evaluation.

	Mean values
WIP_{sim}	17.8947
WIP	17.8821
ϵ_{WIP}	1.72%

(c) Accuracy of BL evaluation.

	Mean values
$BL_{1,sim}$	0.1140
BL_1	0.1120
ϵ_{BL}	0.0021

(d) Accuracy of ST evaluation.

	Mean values
$ST_{2,sim}$	0.1140
ST_2	0.1122
ϵ_{ST}	0.0021

5.2. *($M > 2$)-machine Lines*

In the ($M > 2$) case, we consider $M \in \{3, 5, 10, 15, 20\}$, and for each M generate 100 lines, with machine and buffer parameters selected randomly and equiprobably from the following sets:

$T_{down,i} \in [3, 10]$, $e_i \in [0.6, 0.95]$, $c_i \in [1, 2]$, $p_{OR,i} \in [0, 1]$, $T_{OR} = k_{OR,i}\tau_i$, where $k_{OR,i} \in [0.2, 2]$, $i = 1, \ldots, M$; $N_j = \lceil h_j \max(c_j T_{down,j}, c_{j+1} T_{down,j+1}) \rceil$, where $h_j \in [2, 4]$, and $j = 1, \ldots, M-1$.

The results obtained are presented in Tables 2. From these results, we observe that the accuracy of performance metrics evaluation is decreasing as a function of M, and for large M the errors become relatively large. This is because in the former case the errors are not only due to the reduction of the exact model to a simplified one, but also due to the errors introduced by the aggregation procedure of Bai et al. (2020). Nevertheless, since in most practical cases the data of machine parameters is rarely available with high precision, we conclude that the simplified machine reliability model is still acceptable in most practical systems evaluations with $M \leq 20$.

6. Bottleneck Identification and Improvability of Serial Lines with Unreliable Machines and Cycle Overrun

6.1. *Bottleneck Identification*

The notion of bottlenecks in serial lines is formulated in Li and Meerkov (2009) as follows:

Table 2. Accuracy of performance metrics evaluation using the simplified parametric model in $(M > 2)$-machine lines.

(a) Accuracy of TP evaluation.

Mean values	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
TP_{sim}	34.374	30.426	26.514	23.906	23.262
TP	34.414	30.484	26.797	24.588	24.119
ϵ_{TP}	0.39%	0.53%	1.44%	3.32%	4.26%

(b) Accuracy of WIP evaluation.

Mean values	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
$\frac{1}{M-1}\sum_{i=1}^{M-1} WIP_{i,sim}$	19.209	18.609	16.856	17.315	17.970
$\frac{1}{M-1}\sum_{i=1}^{M-1} WIP_i$	19.164	18.904	18.026	21.433	23.091
ϵ_{WIP}	2.08%	2.13%	4.80%	11.68%	14.37%

(c) Accuracy of BL evaluation.

Mean values	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
$\frac{1}{M-1}\sum_{i=1}^{M-1} BL_{i,sim}$	0.1620	0.1661	0.1630	0.1811	0.1869
$\frac{1}{M-1}\sum_{i=1}^{M-1} BL_i$	0.1627	0.1681	0.1765	0.2254	0.2436
ϵ_{BL}	0.0041	0.0058	0.0163	0.0452	0.0564

(d) Accuracy of ST evaluation.

Mean values	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
$\frac{1}{M-1}\sum_{i=2}^{M} ST_{i,sim}$	0.1234	0.1509	0.1841	0.1973	0.1997
$\frac{1}{M-1}\sum_{i=2}^{M} ST_i$	0.1231	0.1501	0.1697	0.1447	0.1319
ϵ_{ST}	0.0036	0.0050	0.0185	0.0546	0.0686

Definition 6.1. Machine m_i is the bottleneck (BN) of a serial line with M unreliable machines if

$$\frac{\partial TP}{\partial c_i} > \frac{\partial TP}{\partial c_j},\ \forall j \neq i, \tag{13}$$

where c_k is the capacity of the k-th machine, $k = 1, \ldots, M$.

Since evaluating analytically the partial derivatives involved in (13) is all but impossible, Li and Meerkov (2009) provide the following:

BN Identification Procedure:

- Evaluate BL and ST of all machines in the system (either by calculation or by measurements on the factory floor.)
- Assign arrows in each pair of consecutive machines according to the rule:
  - if $BL_i < ST_{i+1}$, assign the arrow pointing from m_i to m_{i+1};
  - if $BL_i > ST_{i+1}$, assign the arrow pointing from m_{i+1} to m_i;
- Then,
  - if there is a single machine with no emanating arrows, it is the BN in the sense of (13);

- if there are multiple machines with no emanating arrows, the one with the largest severity is the Primary BN (P-BN), where the severity is defined by

$$\begin{aligned} &S_j = |ST_{j+1} - BL_j| + |ST_j - BT_{j-1}|, j = 2, \ldots, M-1, \\ &S_1 = |ST_2 - BL_1|, \\ &S_M = |ST_M - BL_{M-1}|. \end{aligned} \tag{14}$$

It is shown in Li and Meerkov (2009) that this identification procedure determines the BNs in systems with machines having no overruns with high accuracy. In this subsection, we verify by simulations whether this approach works for serial lines with machines having overruns, modeled by the simplified parametric model. This is carried out as follows:

We consider ($M > 2$)-machine serial lines with cycle overrun for $M \in \{3, 5, 10, 15, 20\}$. For each M, we generate 20 lines, with parameters selected randomly and equiprobably from the following sets:

$T_{down} \in [3, 10]$, $e_i \in [0.6, 0.95]$, $c_i \in [1, 2]$, $p_{OR,i} \in [0, 0.6]$, $T_{OR,i} = k_i \tau_i$, where $k_i \in [0.2, 2]$, $i = 1, \ldots, M$; $N_j = \lceil h_j \max(c_j T_{down,j}, c_{j+1} T_{down,j+1}) \rceil$, where $h_j \in [2, 4]$, $j = 1, \ldots, M-1$.

For each of the 100 lines obtained, we identify the BN using BN Identification Procedure, and assess its accuracy by comparing it with that identified using numerical evaluation of the derivatives involved in (13). As it turns out, in all 100 lines considered, the BN identified numerically is one of the BN's identified by BN Identification Procedure. Table 3 shows the number of lines, where P-BN identified by BN Identification Procedure is the same as the one identified numerically.

Based on the above, we conclude that the BN Identification Procedure has an acceptable accuracy in serial lines with cycle overrun modeled by the simplified parametric model.

Table 3. The number of lines where P-BN identified by BN Identification Procedure is the same as the one identified numerically.

$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
20	19	18	18	18

6.2. *Throughput Improvability*

Section 4 provided the SAT Improvability Indicator for a single machine with overruns. Will this indicator hod in serial lines for BN machine improvement?

We answer this question by simulations. In the above-mentioned 100 lines, we reduce either downtime or unconditional mean overrun of the BN by 10% using the SAT Improvability Indicator. Table 4 shows the number of lines for each M, where the SAT Improvability Indicator turned out to be optimal in the framework of serial lines as well. Based on this result, we conclude that this indicator may be used for improvement of serial lines with overruns described by the simplified parametric model of the machines.

Table 4. The number of cases, where SAT Improvability Indicator leads to the largest serial line improvement.

$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
18	19	18	20	20

7. Case Study

7.1. *Preliminaries*

In this case study, we consider a production system motivated by an automotive transmission case machining line. Due to confidentiality reasons, all machine and buffer parameters have been modified, preserving, however, qualitative features of the system performance. Due to these modifications, the system throughput that has been measured on the factory floor, cannot be used for validating the mathematical model of this system. Therefore, we have created what is currently referred to as a "digital twin" of the system at hand, populated it by the modified data, and used it as the "real system" in this case study.

The system at hand has been modeled as a serial line consisting of 12 operations separated by finite buffers (see Figure 2, where the numbers in the rectangles represent the modified buffers capacity). Since the original data have been available to us for a period of eight weeks, the case study was carried out using eight weeks of modified data (discussed in Subsection 7.2 below).

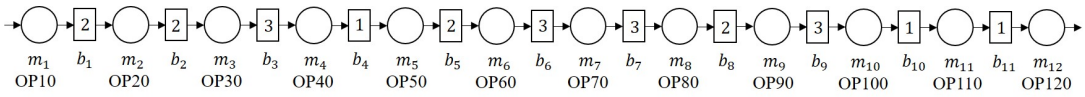


Figure 2. Structural model of the modified production line.

7.2. *Raw Data and Model Validation*

This subsection presents the modified data (referred to thereafter as *raw data*) and uses it to validate the mathematical model of the system at hand.

For each week, the raw data (shown in Table 5) provides the values of machines τ, T_{up}, T_{down}, p_{OR}, and k_{OR}. Based on these data, e and T_{OR}^{uc} have been calculated and included in the raw data tables. As on can see:

- machine cycle time varies from 105sec to 120sec;
- machine efficiency is in the range of 0.61 to 0.97;
- the smallest machine efficiency is that of OP30;
- seven machines experience cycle overruns;
- the longest overruns are in OP60 and OP10.

Using the raw data and the aggregation procedure of Bai et al. (2020), we have calculated the system throughput for all eight weeks (denoted as TP) and compared it with that evaluated by simulations using the digital twin populated by respective week raw data (denoted as TP_{sim}). The results, shown in Table 6, indicate that the mathematical model can be considered as validated.

Table 5. Weekly raw data.

(a) Week 1.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	8.3406	3.024	0.7339	120	0.3563	0.6215	26.5723
OP20	24.9552	4.5164	0.8468	119	0	0	0
OP30	49.9567	4.1475	0.9233	120	0.3592	0.1651	7.1172
OP40	19.9509	3.3174	0.8574	120	0.1644	0.0994	1.9621
OP50	50.0429	3.6922	0.9313	106	0	0	0
OP60	16.6674	3.5635	0.8239	120	0.3318	0.2778	11.0585
OP70	49.9549	3.4369	0.9356	120	0.2762	0.3389	11.2307
OP80	8.3601	2.57	0.7649	120	0.2529	0.0108	0.3268
OP90	24.9889	2.826	0.8984	120	0	0	0
OP100	14.286	3.0102	0.826	113	0	0	0
OP110	19.9701	2.4306	0.8915	120	0.3136	0.1366	5.1407
OP120	50.0198	4.1669	0.9231	105	0	0	0

(b) Week 2.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	16.683	3.5742	0.8236	120	0.6169	0.6012	44.5056
OP20	33.3561	4.3471	0.8847	119	0	0	0
OP30	16.6831	3.5722	0.8236	120	0.3214	0.1227	4.7307
OP40	16.6979	3.3335	0.8336	120	0.1818	0.4583	9.9947
OP50	49.9756	2.3848	0.9545	106	0	0	0
OP60	33.3354	3.4229	0.9069	120	0.3356	0.4993	20.1068
OP70	100.0144	3.7587	0.9638	120	0.2784	0.37	12.3582
OP80	20.0006	2.1139	0.9044	120	0.2557	0.0135	0.4154
OP90	25.0173	2.8394	0.8981	120	0	0	0
OP100	25.0066	3.5459	0.8758	113	0	0	0
OP110	25.013	2.1959	0.9193	120	0.2976	0.4032	14.3994
OP120	50.0034	4.1134	0.924	105	0	0	0

(c) Week 3.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	8.3052	3.109	0.7276	120	0.417	0.5691	28.4766
OP20	16.6606	3.6998	0.8183	119	0	0	0
OP30	19.995	3.3221	0.8575	120	0.3581	0.141	6.0611
OP40	20.0183	3.1142	0.8654	120	0.152	0.1147	2.0931
OP50	19.9967	4.3187	0.8224	106	0	0	0
OP60	25.0236	3.8374	0.8670	120	0.4213	0.38	19.2123
OP70	20.0048	3.4488	0.8530	120	0.266	0.4015	12.8192
OP80	20.0148	3.6823	0.8446	120	0.2989	0.1561	5.5977
OP90	33.3934	3.1262	0.9144	120	0	0	0
OP100	20.0199	3.3236	0.8576	113	0	0	0
OP110	24.9656	2.6729	0.9033	120	0.3389	0.1745	7.0953
OP120	33.397	6.6547	0.8338	105	0	0	0

(d) Week 4.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	7.6206	3.1877	0.7051	120	0.3697	0.6626	29.3931
OP20	6.5843	2.8958	0.6945	119	0	0	0
OP30	19.999	2.8337	0.8759	120	0.3317	0.1905	7.5803
OP40	20.0329	2.6112	0.8847	120	0.1607	0.1539	2.9673
OP50	11.1021	2.9251	0.7915	106	0	0	0
OP60	19.9859	2.8353	0.8758	120	0.4195	0.2644	13.3123
OP70	50.0824	3.2984	0.9382	120	0.2298	0.5568	15.3532
OP80	20.0935	3.7012	0.8445	120	0.3224	0.4793	18.5469
OP90	33.3307	3.1948	0.9125	120	0	0	0
OP100	20.0855	3.321	0.8581	113	0	0	0
OP110	33.3265	2.6876	0.9254	120	0.3454	0.192	7.9598
OP120	99.9323	5.8691	0.9445	105	0	0	0

(e) Week 5.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	14.2714	4.7327	0.7510	120	0.4465	0.5577	29.8829
OP20	7.6328	3.6655	0.6756	119	0	0	0
OP30	8.3027	1.7795	0.8235	120	0.3367	0.1548	6.2526
OP40	14.2686	4.7353	0.7508	120	0.1717	0.1309	2.6987
OP50	16.6899	2.7667	0.8578	106	0	0	0
OP60	24.9749	3.9629	0.8631	120	0.4232	0.5553	28.197
OP70	49.9563	5.252	0.9049	120	0.2733	0.4473	14.6694
OP80	16.7332	4.001	0.807	120	0.259	0.0371	1.1517
OP90	33.3859	4.1533	0.8894	120	0	0	0
OP100	25.0217	3.8278	0.8673	113	0	0	0
OP110	25.0765	4.1712	0.8574	120	0.323	0.1853	7.183
OP120	99.9755	7.6868	0.9286	105	0	0	0

(f) Week 6.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	10.0586	3.7924	0.7262	120	0.4133	0.4509	22.3628
OP20	14.3617	3.3406	0.8113	119	0	0	0
OP30	5.8879	3.6987	0.6142	120	0.3322	0.0682	2.7178
OP40	12.4703	3.3201	0.7897	120	0.1734	0.1725	3.5902
OP50	20.0145	2.8557	0.8751	106	0	0	0
OP60	25.0407	3.8424	0.867	120	0.3954	0.5403	25.6384
OP70	50.0236	3.5653	0.9335	120	0.2965	0.4262	15.1605
OP80	20.0894	3.0354	0.8687	120	0.3154	0.3272	12.3836
OP90	33.2873	4.9632	0.8702	120	0	0	0
OP100	12.4909	2.7645	0.8188	113	0	0	0
OP110	33.3266	2.9013	0.9199	120	0.3151	0.2117	8.0047
OP120	99.9662	7.6247	0.9291	105	0	0	0

(g) Week 7.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	6.6949	2.2091	0.7519	120	0.5653	0.9904	67.1901
OP20	12.5163	3.5632	0.7784	119	0	0	0
OP30	5.8558	2.2167	0.7254	120	0.3218	0.1063	4.1054
OP40	16.6304	3.0973	0.843	120	0.1695	0.1131	2.2999
OP50	33.2742	3.4126	0.907	106	0	0	0
OP60	24.9705	3.8251	0.8672	120	0.3981	0.4726	22.5791
OP70	99.9469	3.1556	0.9694	120	0.3125	0.3861	14.4802
OP80	14.2309	2.9193	0.8298	120	0.2843	0.1452	4.9556
OP90	24.9441	3.1127	0.8891	120	0	0	0
OP100	24.9722	3.6785	0.8716	113	0	0	0
OP110	33.3025	3.8017	0.8975	120	0.3	0.1983	7.1391
OP120	100.0185	9.001	0.9174	105	0	0	0

(h) Week 8.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	7.7444	2.7936	0.7349	120	0.3993	0.6681	32.0122
OP20	19.9792	7.6559	0.7230	119	0	0	0
OP30	7.136	2.4903	0.7413	120	0.3354	0.0917	3.6886
OP40	33.3248	3.8121	0.8974	120	0.1654	0.1311	2.6033
OP50	33.2792	2.8491	0.9211	106	0	0	0
OP60	14.3036	2.2152	0.8659	120	0.5069	0.9835	59.8269
OP70	33.3474	6.6485	0.8338	120	0.2851	0.3096	10.5952
OP80	20.0423	3.4521	0.8531	120	0.2603	0.1068	3.3357
OP90	25.0204	4.3444	0.8521	120	0	0	0
OP100	50.0636	3.6599	0.9319	113	0	0	0
OP110	16.6557	3.1142	0.8425	120	0.2919	0.1568	5.4932
OP120	100.0047	8.3081	0.9233	105	0	0	0

Table 6. Model validation.

Throughput (JPH)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
TP_{sim}	16.4970	16.9290	15.7003	15.2434	14.5075	14.4710	13.8117	13.8078
TP	16.6910	17.0372	16.0725	15.3409	14.9029	14.6863	13.7300	14.1172
ϵ_{TP}	1.18%	0.64%	2.37%	0.64%	2.73%	1.49%	0.59%	2.24%

7.3. *Weekly Performance Analysis*

The weekly performance analysis has been carried out based on the weekly raw data and the aggregation procedure developed in Bai et al. (2020). The results are shown in Figure 3 for Weeks 1 to 8. As one can see:

- in Weeks 2, 3, 4, 5, and 7, OP10 has the smallest SAT and is the system's BN;

- in Week 1, OP80 is the P-BN due to its small SAT and buffering;
- in Week 6, OP30 has the smallest e and SAT value and is the system's BN;
- in Week 8, OP60 is the P-BN, although its e is relatively high, its SAT is small due to the large overruns;
- in all eight weeks, TP is substantially below the smallest SAT.

7.4. *Data for Continuous Improvement Project Design*

The weekly raw data exhibit substantial variability. This is obvious from Tables 5 and is supported by the coefficients of variation of e and T_{OR}^{uc} calculated using all eight weeks data and shown in the first column of Tables 7. This observation makes it necessary to "pre-process" the raw data in order to decrease its variability and use the less variable data for the continuous improvement project design. This is accomplished by averaging machine parameters using two or four weeks' data. The resulting coefficients of variation are shown in Tables 7. Clearly, the averaging over four weeks results in a relatively low variability of machine parameters, and this data (shown in Table 8) is used for continuous improvement project design. (Note that averaging over all eight weeks, i.e., over two months, might not be desirable, since machine parameters are not stationary and evolve in time.)

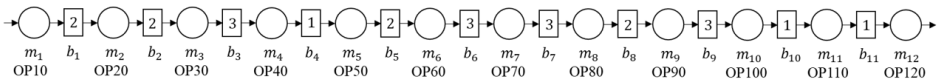
Table 7. Coefficients of variation the original and averaged raw data.

(a) Data for e.

	$CV_{e,0}$	$CV_{e,2}$	$CV_{e,4}$
OP10	0.0444	0.0300	0.0044
OP20	0.0900	0.0645	0.0411
OP30	0.1163	0.0905	0.0902
OP40	0.0545	0.0499	0.0238
OP50	0.0597	0.0584	0.0087
OP60	0.0243	0.0029	0.0015
OP70	0.0506	0.0230	0.0067
OP80	0.0462	0.0044	< 0.0001
OP90	0.0220	0.0186	0.0172
OP100	0.0375	0.0264	0.0104
OP110	0.0316	0.0189	0.0171
OP120	0.0347	0.0169	0.0100

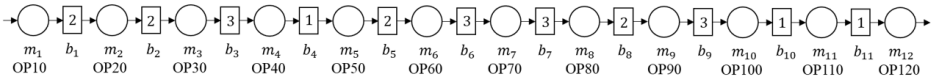
(b) Data for T_{OR}^{uc}.

	$CV_{T_{OR}^{uc},0}$	$CV_{T_{OR}^{uc},2}$	$CV_{T_{OR}^{uc},4}$
OP10	0.3863	0.2588	0.0802
OP30	0.3068	0.2186	0.2065
OP40	0.7068	0.4087	0.2065
OP60	0.569	0.4155	0.3629
OP70	0.129	0.0924	0.0295
OP80	1.0368	0.729	0.0655
OP110	0.3431	0.1596	0.1085



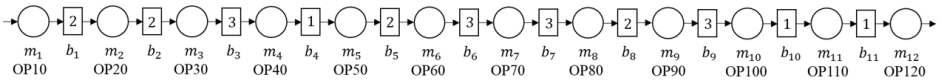
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.73	0.85	0.92	0.86	0.93	0.82	0.94	0.76	0.90	0.83	0.89	0.92
τ_{OR}^{uc} (sec)	26.57	0.00	7.12	1.96	0.00	11.06	11.23	0.33	0.00	0.00	5.14	0.00
SAT (JPH)	18.03	25.62	26.15	25.31	31.63	22.63	25.67	22.88	26.95	26.31	25.65	31.65
WIP	0.43	0.65	1.08	0.28	1.05	1.12	0.48	0.22	0.46	0.42	0.18	
ST	0.00	0.27	0.27	0.20	0.33	0.16	0.19	0.16	0.31	0.28	0.31	0.44
BL	0.05	0.04	0.08	0.12	0.19	0.07	0.18	0.06	0.06	0.03	0.01	0.00
S	0.22					0.15		0.27				
TP	16.6910JPH											

(a) Week 1.



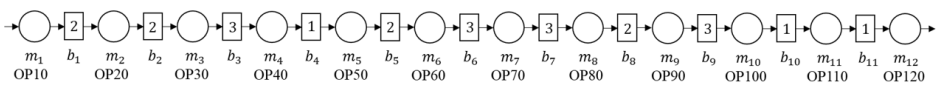
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.82	0.88	0.82	0.83	0.95	0.91	0.96	0.90	0.90	0.88	0.92	0.92
τ_{OR}^{uc} (sec)	44.51	0.00	4.73	9.99	0.00	20.11	12.36	0.42	0.00	0.00	14.40	0.00
SAT (JPH)	18.02	26.76	23.77	23.08	32.42	23.30	26.21	27.04	26.94	27.90	24.62	31.68
WIP	0.35	0.82	0.85	0.09	0.55	0.41	0.22	0.19	0.34	0.53	0.17	
ST	0.00	0.29	0.20	0.20	0.43	0.23	0.29	0.29	0.30	0.32	0.28	0.43
BL	0.04	0.05	0.04	0.03	0.06	0.02	0.06	0.06	0.04	0.04	0.01	0.00
TP	17.0372JPH											

(b) Week 2.



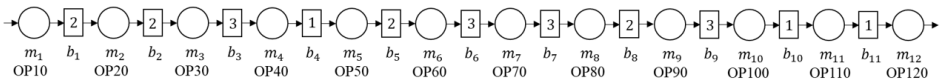
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.73	0.82	0.86	0.87	0.82	0.87	0.85	0.84	0.91	0.86	0.90	0.83
τ_{OR}^{uc} (sec)	28.48	0.00	6.06	2.09	0.00	19.21	12.82	5.60	0.00	0.00	7.10	0.00
SAT (JPH)	17.64	24.75	24.49	25.52	27.93	22.42	23.12	24.21	27.43	27.32	25.59	28.59
WIP	0.53	0.95	1.26	0.40	1.03	1.14	0.31	0.18	0.39	0.47	0.44	
ST	0.00	0.24	0.21	0.17	0.24	0.19	0.17	0.25	0.35	0.33	0.31	0.37
BL	0.06	0.06	0.10	0.18	0.17	0.08	0.11	0.06	0.05	0.05	0.05	0.00
TP	16.0725JPH											

(c) Week 3.



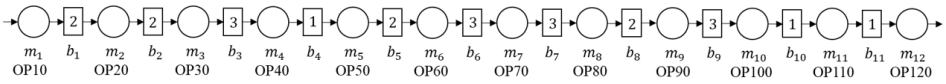
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.71	0.69	0.88	0.88	0.79	0.88	0.94	0.84	0.91	0.86	0.93	0.94
τ_{OR}^{uc} (sec)	29.39	0.00	7.58	2.97	0.00	13.31	15.35	18.55	0.00	0.00	7.96	0.00
SAT (JPH)	16.99	21.01	24.72	25.90	26.88	23.65	24.95	21.94	27.38	27.34	26.03	32.38
WIP	0.69	0.47	0.72	0.28	0.59	0.73	0.39	0.14	0.29	0.24	0.13	
ST	0.00	0.18	0.31	0.28	0.31	0.28	0.27	0.22	0.38	0.37	0.37	0.50
BL	0.07	0.01	0.04	0.12	0.06	0.04	0.13	0.05	0.04	0.02	0.01	0.00
TP	15.3409JPH											

(d) Week 4.



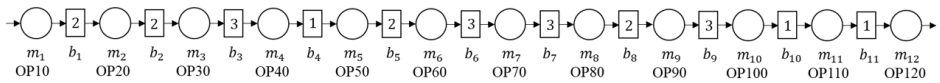
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.75	0.68	0.82	0.75	0.86	0.86	0.90	0.81	0.89	0.87	0.86	0.93
τ_{OR}^{uc} (sec)	29.88	0.00	6.25	2.70	0.00	28.20	14.67	1.15	0.00	0.00	7.18	0.00
SAT (JPH)	18.04	20.44	23.48	22.03	29.13	20.97	24.19	23.98	26.68	27.63	24.27	31.84
WIP	0.89	1.00	1.30	0.32	0.96	0.97	0.34	0.20	0.40	0.48	0.17	
ST	0.00	0.14	0.20	0.15	0.33	0.19	0.23	0.25	0.36	0.37	0.32	0.50
BL	0.12	0.04	0.12	0.11	0.16	0.08	0.16	0.08	0.06	0.07	0.02	0.00
TP	14.9029JPH											

(e) Week 5.



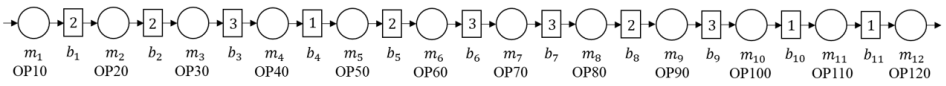
c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.73	0.81	0.61	0.79	0.88	0.87	0.93	0.87	0.87	0.82	0.92	0.93
τ_{OR}^{uc} (sec)	22.36	0.00	2.72	3.59	0.00	25.64	15.16	12.38	0.00	0.00	8.00	0.00
SAT (JPH)	18.36	24.54	18.02	23.00	29.72	21.43	24.86	23.62	26.11	26.09	25.87	31.86
WIP	0.96	1.87	0.93	0.21	0.73	0.63	0.33	0.19	0.39	0.26	0.16	
ST	0.00	0.16	0.07	0.23	0.39	0.25	0.30	0.27	0.36	0.35	0.39	0.50
BL	0.14	0.20	0.05	0.08	0.11	0.03	0.12	0.09	0.05	0.02	0.02	0.00
S	0.02		0.31									
TP	14.6863JPH											

(f) Week 6.



c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.75	0.78	0.73	0.84	0.91	0.87	0.97	0.83	0.89	0.87	0.90	0.92
τ_{OR}^{uc} (sec)	67.19	0.00	4.11	2.30	0.00	22.58	14.48	4.96	0.00	0.00	7.14	0.00
SAT (JPH)	14.46	23.55	21.04	24.81	30.80	21.90	25.95	23.91	26.67	27.77	25.41	31.46
WIP	0.38	0.65	0.47	0.13	0.57	0.40	0.26	0.15	0.28	0.32	0.19	
ST	0.00	0.31	0.24	0.34	0.47	0.31	0.39	0.32	0.41	0.42	0.40	0.52
BL	0.03	0.02	0.02	0.06	0.08	0.02	0.11	0.06	0.04	0.04	0.03	0.00
TP	13.7300JPH											

(g) Week 7.



c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.73	0.72	0.74	0.90	0.92	0.87	0.83	0.85	0.85	0.93	0.84	0.92
τ_{OR}^{uc} (sec)	32.01	0.00	3.69	2.60	0.00	59.83	10.60	3.34	0.00	0.00	5.49	0.00
SAT (JPH)	17.40	21.87	21.58	26.35	31.28	17.33	22.98	24.90	25.56	29.69	24.17	31.66
WIP	0.75	1.04	0.71	0.46	1.38	0.83	0.25	0.18	0.20	0.40	0.18	
ST	0.00	0.21	0.23	0.30	0.28	0.09	0.25	0.32	0.36	0.46	0.34	0.51
BL	0.14	0.06	0.04	0.17	0.32	0.08	0.10	0.08	0.03	0.05	0.02	0.00
S	0.07					0.34						
TP	14.1172JPH											

(h) Week 8.

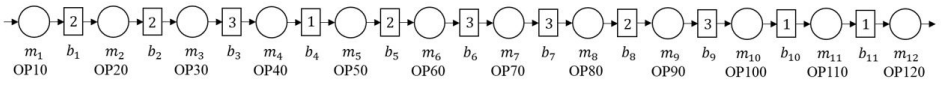
Figure 3. Performance analysis based on weekly data.

Table 8. Averaged raw data for Weeks 5-8.

	T_{up} (min)	T_{down} (min)	e	τ (sec)	p_{OR}	k_{OR}	T_{OR}^{uc} (sec)
OP10	9.6923	3.382	0.7413	120	0.4561	0.6668	36.495
OP20	13.6225	4.5563	0.7494	119	0	0	0
OP30	6.7956	2.5463	0.7274	120	0.3315	0.1052	4.1861
OP40	19.1735	3.7412	0.8367	120	0.17	0.1369	2.7935
OP50	25.8145	2.971	0.8968	106	0	0	0
OP60	22.3224	3.4614	0.8658	120	0.4309	0.6379	32.9867
OP70	58.3186	4.6553	0.9261	120	0.2918	0.3923	13.7395
OP80	17.774	3.352	0.8413	120	0.2798	0.1541	5.1724
OP90	29.1594	4.1434	0.8756	120	0	0	0
OP100	28.1371	3.4827	0.8899	113	0	0	0
OP110	27.0904	3.4971	0.8857	120	0.3075	0.188	6.9386
OP120	99.9912	8.1551	0.9246	105	0	0	0

7.5. *System Performance Analysis Using Four-weeks Averaged Data and Project Goal*

The performance of the system at hand has been evaluated using the averaged data of Table 8 and the aggregation procedure of Bai et al. (2020). The result is shown in Figure 4. As one can see the bottleneck is OP10 and $TP = 14.66$JPH. Since the nominal throughput (defined by the longest cycle time ($\tau = 120$sec) under the assumption that there are no machine breakdowns or cycle overruns) is 30JPH, i.e., the throughput losses are over 50%.



c (JPH)	30.00	30.25	30.00	30.00	33.96	30.00	30.00	30.00	30.00	31.86	30.00	34.29
e	0.81	0.77	0.75	0.84	0.90	0.87	0.93	0.84	0.88	0.89	0.89	0.92
τ_{OR}^{uc} (sec)	36.50	0.00	4.19	2.79	0.00	32.99	13.74	5.17	0.00	0.00	6.94	0.00
SAT (JPH)	16.91	22.67	21.09	24.53	30.46	20.23	24.93	24.14	26.27	28.35	25.11	31.70
WIP	0.67	1.07	0.72	0.22	0.87	0.63	0.29	0.18	0.27	0.35	0.18	
ST	0.00	0.21	0.18	0.27	0.40	0.22	0.30	0.29	0.37	0.41	0.36	0.50
BL	0.09	0.06	0.04	0.08	0.14	0.04	0.12	0.07	0.04	0.04	0.02	0.00
TP	14.6557JPH											

Figure 4. Performance analysis based on the averaged data.

It is of interest to evaluate what fraction of these losses are due to machine downtime and due to cycle overrun. The former can be calculated assuming that T_{down} of each machine in Table 8 is zero, and the latter assuming that p_{OR} is zero. The throughputs in the first case turns out to be 23.00JPH and in the second 17.26JPH. Thus, the production losses due to machine downtimes are 8.34JPH and due to cycle overrun 2.60JPH. Recovering these losses is the goal of this case study. More precisely, the goal is to design four options for a continuous improvement project leading to 5%, 10%, 20% and 30% of throughput improvement.

7.6. *Designing Continuous Improvement Projects*

Based on the results described in Section 4-6, we formulate:

Procedure for Continuous Improvement Projects Design:

(a) Using the BN Identification Procedure, determine the system's BN or P-BN.
(b) Using the SAT Improvability Indicator, decrease either T_{down} or T_{OR}^{uc} of the bottleneck.
(c) Calculate TP of the improved system (using, for instance, the aggregation procedure of Bai et al. (2020)).
(d) If $TP_{imp} \geq TP_{des}$, stop; else go to (a).

Applying this procedure to the production system at hand, we obtain steps for continuous improvement resulting up to 30% TP increase, i.e., $TP_{imp} = 19.05$JPH (see Table 9). Based on these steps, we specify the activities related to each machine leading to the desired throughput improvement. This results in continuous improvement project to ensure 5%, 10%, 20% and 30% TP improvement shown in Table 10. Note that for 5% improvement, only one machine must be improvement; for 10% improvement, parameters of three machines should be modified; for 20% improvement, five machines must be modified; and for 30% improvement, eight machines must be improved.

This information is intended to allow the Operations Manager to decide which of these continuous improvement projects should be implemented on the factory floor.

Table 9. Improvement steps.

	Improvement steps	TP	ΔTP	Percentage increase
0	initial state	14.6557	0	0%
1	reduce OP10 downtime by 35%, cycle overrun by 20%	15.6961	1.0404	6.63%
2	reduce OP60 overrun by 20%	15.7960	1.1403	7.22%
3	reduce OP10 downtime by 10%	15.9698	1.3141	8.23%
4	reduce OP20 downtime by 10%	16.2481	1.5924	9.80%
5	reduce OP60 cycle overrun by 20%	16.3311	1.6754	10.26%
6	reduce OP10 cycle overrun by 20%	16.5204	1.8647	11.29%
7	reduce OP30 downtime by 10%	16.7583	2.1026	12.55%
8	reduce OP60 downtime by 10%, cycle overrun by 20%	17.3995	2.7438	15.77%
9	reduce OP20 downtime by 10%	17.1351	2.4794	14.47%
10	reduce OP30 downtime by 10%	17.3457	2.6900	15.51%
11	reduce OP60 cycle downtime by 10%, cycle overrun by 20%	17.4619	2.8062	16.07%
12	reduce OP80 downtime by 10%	17.5078	2.8521	16.29%
13	reduce OP10 downtime by 10%	17.6579	3.0022	17.00%
14	reduce OP30 downtime by 10%	17.8537	3.1980	17.91%
15	reduce OP60 downtime by 10%	17.91	3.26	18.18%
16	reduce OP10 downtime by 10%	18.0441	3.3884	18.78%
17	reduce OP80 downtime by 10%	18.0959	3.4402	19.01%
18	reduce OP20 downtime by 10%	18.3165	3.6608	19.99%
19	reduce OP30 downtime by 10%	18.4906	3.8349	20.74%

20	reduce OP80 downtime by 10%	18.5484	3.8927	20.99%
21	reduce OP60 downtime by 10%	18.6059	3.9502	21.23%
22	reduce OP10 cycle overrun by 20%	18.7783	4.1226	21.95%
23	reduce OP30 downtime by 10%	18.9232	4.2675	22.55%
24	reduce OP80 downtime by 10%	18.9832	4.3275	22.80%
25	reduce OP60 cycle overrun by 20%	19.0356	4.3799	23.01%
26	reduce OP20 downtime by 10%	19.2177	4.5620	23.74%
27	reduce OP40 downtime by 10%	19.3518	4.6961	24.27%
28	reduce OP80 downtime by 10%	19.4144	4.7587	24.51%
29	reduce OP30 downtime by 10%	19.5501	4.8944	25.04%
30	reduce OP40 downtime by 10%	19.6622	5.0065	25.46%
31	reduce OP80 downtime by 10%	19.7181	5.0624	25.67%
32	reduce OP10 downtime by 10%	19.8333	5.1776	26.11%
33	reduce OP80 downtime by 10%	19.8883	5.2326	26.31%
34	reduce OP60 downtime by 10%	19.9476	5.2919	26.53%
35	reduce OP10 downtime by 10%	20.0497	5.3940	26.90%
36	reduce OP70 cycle overrun by 20%	20.0506	5.3949	26.91%
37	reduce OP80 downtime by 10%	20.0979	5.4422	27.08%
38	reduce OP60 downtime by 10%, cycle overrun by 20%	20.1957	5.5400	27.43%
39	reduce OP80 downtime by 10%	20.2315	5.5758	27.56%
40	reduce OP10 cycle overrun by 20%	20.3683	5.7126	28.05%
41	reduce OP30 downtime by 10%	20.4985	5.8428	28.50%
42	reduce OP40 downtime by 10%	20.6070	5.9513	28.88%
43	reduce OP90 downtime by 10%	20.6884	6.0327	29.16%
44	reduce OP20 downtime by 10%	20.8385	6.1828	29.67%
45	reduce OP40 downtime by 10%	20.9336	6.2779	29.99%
46	reduce OP90 downtime by 10%	21.0197	6.3640	30.28%

8. Conclusions and Future Work

This paper provides analytical methods for analysis and improvement of serial lines with unreliable machines and cycle overrun. These methods offer the analytics for modelling, analysis and improvement for a relatively large class of real-world serial lines, which has not been thus far explored in production systems literature.

Results reported here can be extended in at least two directions. The first one is to develop similar methods for assembly systems. The second is to extend the current results to systems with non-exponential machine reliability and overrun models. To-date, we have obtained a few results in this direction. Specifically, we have shown that Theorems 3.1 and 4.1 hold for Weibull, gamma and log-normal distributions of T_{up}, T_{down}, and cycle overrun. However, many other issues remain so far unexplored. In addition to these two areas of future research, a very important one is the application of the methods developed to systems on the factory floor.

Table 10. Continuous improvement projects.

(a) For 5% throughput improvement (resulting, in fact, in 6%).

Machine	Machine improvement
OP10	Reduce downtime by 35%, reduce cycle overrun by 20%.

(b) For 10% throughput improvement, i.e., $TP_{imp} = 16.12$JPH.

Machine	Machine improvement
OP10	Reduce downtime by 35%, reduce cycle overrun by 20%.
OP20	Reduce downtime by 10%.
OP60	Reduce cycle overrun by 36%.

(c) For 20% throughput improvement, i.e., $TP_{imp} = 17.59$JPH.

Machine	Machine improvement
OP10	Reduce downtime by 47%, reduce cycle overrun by 36%.
OP20	Reduce downtime by 27%.
OP30	Reduce downtime by 35%.
OP60	Reduce downtime by 27%, reduce cycle overrun by 59%.
OP80	Reduce downtime by 19%.

(d) For 30% throughput improvement, i.e., $TP_{imp} = 19.05$JPH.

Machine	Machine improvement
OP10	Reduce downtime by 57%, reduce cycle overrun by 59%.
OP20	Reduce downtime by 41%.
OP30	Reduce downtime by 52%.
OP40	Reduce downtime by 27%.
OP60	Reduce downtime by 47%, reduce cycle overrun by 74%.
OP70	Reduce cycle overrun by 20%.
OP80	Reduce downtime by 61%.
OP90	Reduce downtime by 10%.

Abbreviations and Notations

ϵ	error
λ	machine breakdown rate
μ	machine repair rate
τ	cycle time
τ_{OR}	cycle overrun
τ_{total}	part processing time
BL	blockage
BN	bottleneck
c	machine capacity
CV	coefficient of variation
des	desired
e	machine efficiency
$\exp(\cdot)$	exponential function

$f_{OR}(t)$	overrun distribution
h	level of buffering
imp	improved
k_{OR}	overrun multiple
M	number of machine
N	buffer capacity
OP	operation
p_{OR}	overrun probability
P-BN	primary bottleneck
r	reduction coefficient
sim	simulated
SAT	stand-alone throughput
ST	starvation
T_{OR}	mean time of overrun
T_{up}	average uptime
T_{down}	average downtime
$t_{up,i}$	i-th realization of uptime
$t_{down,i}$	i-th realization of downtime
TP	throughput
uc	unconditional
WIP	work-in-process

Funding

This work has been supported in part by the DGIST RD Program of the Ministry of Science and ICT (18-EE-00).

References

Altiok, T. 1997. *Performance Analysis of Manufacturing Systems.* Springer-Verlag, New York, NY.

Askin, R. G., and C. R. Standridge. 1993. *Modeling and Analysis of Manufacturing Systems.* Vol. 29. Wiley New York.

Bai, Yishu, Jiachen Tu, Mengzhuo Yang, Liang Zhang, and Peter Denno. 2020. "A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: design, accuracy and robustness." *International Journal of Production Research* 1–18.

Ben-Ammar, Oussama, Belgacem Bettayeb, and Alexandre Dolgui. 2020. "Integrated production planning and quality control for linear production systems under uncertainties of cycle time and finished product quality." *International Journal of Production Research* 58 (4): 1144–1160.

Buzacott, J. A., and J. G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems.* Vol. 4. Prentice Hall Englewood Cliffs, NJ.

Casalino, Andrea, Andrea Maria Zanchettin, Luigi Piroddi, and Paolo Rocco. 2019. "Optimal scheduling of human-robot collaborative assembly operations with time petri nets." *IEEE Transactions on Automation Science and Engineering* 18 (1): 70–84.

Curry, G. L., and R. M. Feldman. 2010. *Manufacturing systems modeling and analysis.* Springer Science & Business Media.

Gershwin, S. B. 1994. *Manufacturing Systems Engineering.* Prentice Hall, Englewood Cliff, NJ.

Kacar, Necip Baris, Lars Mönch, and Reha Uzsoy. 2016. "Modeling cycle times in production planning models for wafer fabrication." *IEEE Transactions on Semiconductor Manufacturing* 29 (2): 153–167.

Kuo, Chung-Jen, Chen-Fu Chien, and Jan-Daw Chen. 2011. "Manufacturing intelligence to exploit the value of production and tool data to reduce cycle time." *IEEE Transactions on Automation Science and Engineering* 8 (1): 103–111.

Larco, José Antonio, Rene De Koster, Kees Jan Roodbergen, and Jan Dul. 2017. "Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions." *International Journal of Production Research* 55 (21): 6407–6422.

Li, J., and S. M. Meerkov. 2009. *Production Systems Engineering*. Springer. (Chinese translation, 2012).

Millstein, Mitchell A, and Joseph S Martinich. 2014. "Takt Time Grouping: implementing kanban-flow manufacturing in an unbalanced, high variation cycle-time process with moving constraints." *International Journal of Production Research* 52 (23): 6863–6877.

Morrison, James R, and Donald P Martin. 2007. "Practical extensions to cycle time approximations for the $g/g/m$-queue with applications." *IEEE Transactions on Automation Science and Engineering* 4 (4): 523–532.

Nadarajah, Saralees, and Samuel Kotz. 2008. "The cycle time distribution." *International Journal of Production Research* 46 (11): 3133–3141.

Papadopolous, H. T., C. Heavey, J. Browne, and B. A. 1993. *Queueing Theory in Manufacturing Systems Analysis and Design*. Springer Science & Business Media.

Papadopoulos, H. T., Michael E. J. O'Kelly, M. J. Vidalis, and D. Spinellis. 2009. *Analysis and Design of Discrete Part Production Lines*. Springer.

Roshani, Abdolreza, Massimo Paolucci, Davide Giglio, and Flavio Tonelli. 2020. "A hybrid adaptive variable neighbourhood search approach for multi-sided assembly line balancing problem to minimise the cycle time." *International Journal of Production Research* DOI: 10.1080/00207543.2020.1749958.

Touzani, Hicham, Hicham Hadj-Abdelkader, Nicolas Seguy, and Samia Bouchafa. 2021. "Multi-Robot Task Sequencing & Automatic Path Planning for Cycle Time Optimization: Application for Car Production Line." *IEEE Robotics and Automation Letters* 6 (2): 1335–1342.

Viswanadham, N., and Y. Narahari. 1992. *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall, Englewood Cliff, NJ.

Appendix A. Proof of Theorem 3.1

Consider an unreliable machine with cycle overrun defined by $\{\tau, T_{up}, T_{down}, p_{OR}, T_{OR}\}$ at the end of the N-th up- and downtime realization. Denote the number of parts produced during this time period as K. This time period equals to $\sum_{i=1}^{N}(t_{up,i}+t_{down,i})$, and the machine's total uptime is $\sum_{i=1}^{N} t_{up,i}$. Denote the duration of the processing time of k-th part as $\tau_{total,k} = \tau+\tau_{OR,k}$, where τ_{OR} has mean value $p_{OR}T_{OR}$. Since there are K parts produced, $\sum_{i=1}^{N} t_{up,i} \geq \sum_{k=1}^{K} \tau_{total,k}$. Let $\sum_{i=1}^{N} t_{up,i} = \sum_{k=1}^{K} \tau_{total,k} + \check{\tau}$, where $\check{\tau} < \tau_{total,K+1}$.

Assume $K \to \infty$, then, obviously, $N \to \infty$ as well. Hence, the stand-alone through-

put of the machine can be evaluated as follows:

$$\begin{aligned} SAT_{exact} &= \lim_{K\to\infty} \frac{K}{\sum_{i=1}^{N}(t_{up,i}+t_{down,i})} \\ &= \lim_{K\to\infty} \left(\frac{K}{\sum_{k=1}^{K}\tau_{total,k}+\check{\tau}}\right)\left(\frac{\sum_{i=1}^{N} t_{up,i}}{\sum_{i=1}^{N}(t_{up,i}+t_{down,i})}\right) \\ &= \lim_{K\to\infty} \left(\frac{K}{\sum_{k=1}^{K}\tau_{total,k}+\check{\tau}}\right)\left(\lim_{N\to\infty}\frac{\sum_{i=1}^{N} t_{up,i}}{\sum_{i=1}^{N}(t_{up,i}+t_{down,i})}\right). \end{aligned} \tag{A1}$$

Clearly, for the first of the above limits, the following bounds hold:

$$\begin{aligned} \lim_{K\to\infty} \frac{K}{\sum_{k=1}^{K+1}\tau_{total,k}} &\equiv \lim_{K\to\infty} \frac{K}{K+1}\frac{K+1}{\sum_{k=1}^{K+1}\tau_{total,k}} < \lim_{K\to\infty} \frac{K}{\sum_{k=1}^{K}\tau_{total,k}+\check{\tau}} \\ &\leq \lim_{K\to\infty} \frac{K}{\sum_{k=1}^{K}\tau_{total,k}}, \end{aligned} \tag{A2}$$

Then, by the law of large numbers, the following takes place with probability 1:

$$\lim_{K\to\infty} \frac{K+1}{\sum_{k=1}^{K+1}\tau_{total,k}} = \lim_{K\to\infty} \frac{K}{\sum_{k=1}^{K}\tau_{total,k}} = \lim_{K\to\infty} \frac{K}{\sum_{k=1}^{K}\tau_{total,k}+\check{\tau}} = \frac{1}{\tau+p_{OR}T_{OR}}. \tag{A3}$$

As far as the second term in (11) is concerned, by the law of large numbers we have:

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} t_{up,i}}{\sum_{i=1}^{N}(t_{up,i}+t_{down,i})} = \frac{\lim_{N\to\infty}\frac{\sum_{i=1}^{N} t_{up,i}}{N}}{\lim_{N\to\infty}\frac{\sum_{i=1}^{N} t_{up,i}}{N}+\lim_{N\to\infty}\frac{\sum_{i=1}^{N} t_{down,i}}{N}} = \frac{T_{up}}{T_{up}+T_{down}}. \tag{A4}$$

Thus, combining (A3) and (A4), and assuming T_{up} and T_{down} are in seconds, we obtain:

$$\begin{aligned} SAT_{exact} =& \frac{1}{\tau+p_{OR}T_{OR}}\frac{T_{up}}{T_{up}+T_{down}} \text{ parts/second} \\ =& \frac{3600}{\tau+p_{OR}T_{OR}}\frac{T_{up}}{T_{up}+T_{down}} \text{ parts/hour.} \end{aligned} \tag{A5}$$

□

Appendix B. Proof of Theorem 4.1

In the following, we prove that $\frac{T_{down}}{T_{up}} < \frac{p_{OR}T_{OR}}{\tau}$ is a necessary and sufficient condition for the overrun-reduced machine having a larger SAT than that of the downtime-reduced one.

To show necessity, assume that the overrun-reduced machine has a larger SAT than

that of the downtime-reduced one. In this case, we have:

$$\begin{aligned} \frac{3600}{\tau + rp_{OR}T_{OR}} \frac{T_{up}}{T_{up} + T_{down}} &> \frac{3600}{\tau + p_{OR}T_{OR}} \frac{T_{up}}{T_{up} + rT_{down}} \\ \Leftrightarrow (\tau + rp_{OR}T_{OR})(T_{up} + T_{down}) &< (\tau + p_{OR}T_{OR})(T_{up} + rT_{down}) \\ \Leftrightarrow (1-r)\tau T_{down} &< (1-r)p_{OR}T_{OR}T_{up} \\ \Leftrightarrow \frac{T_{down}}{T_{up}} &< \frac{p_{OR}T_{OR}}{\tau}. \end{aligned} \tag{B1}$$

Note that the chain of inequalities in (B1) is bi-directional, thus, $\frac{T_{down}}{T_{up}} < \frac{p_{OR}T_{OR}}{\tau}$ is both necessary and sufficient.

Similarly, it can be shown that the downtime-reduced machine has a larger SAT than that of the overrun-reduced one, i.e., $\frac{3600}{\tau+rp_{OR}T_{OR}} \frac{T_{up}}{T_{up}+T_{down}} < \frac{3600}{\tau+p_{OR}T_{OR}} \frac{T_{up}}{T_{up}+rT_{down}}$ if and only if $\frac{T_{down}}{T_{up}} > \frac{p_{OR}T_{OR}}{\tau}$ holds. □