© 2018 г. Pooya Alavian, Research Associate
(University of Michigan, Ann Arbor, MI, USA),
Yongsoon Eun, Professor
(DGIST, Daegu, Republic of Korea),
Semyon M. Meerkov, Professor
(University of Michigan, Ann Arbor, MI, USA),
Liang Zhang, Associate Professor
(University of Connecticut, Storrs, CT, USA)

# Programmable Manufacturing Advisor – A Tool for Automating Decision-Making in Production Systems[1]

Programmable Manufacturing Advisor (PMA) is a device intended to automate decision-making in manufacturing environment. Programming and installing a PMA at any production system makes it smart: it becomes capable of self-diagnosing and providing the Operations Manager with an advice for achieving the desired productivity improvement. In this paper, theoretical/analytical foundations of PMA are outlined, its software/hardware implementations are commented upon, and demonstrations of PMA-based Smart Production Systems are provided using an automotive underbody assembly system and a hot-dip galvanization plant.

**A personal note:**

I first met Yakov Zalmanovich Tsypkin in 1964, shortly after joining IAT as a doctoral student (aspirant). By that time, I was familiar with his two books on relay systems and on impulsive control, both, in my opinion, classics in control theory. Although we did not have a personal relationship, I always admired his work and never missed his seminar presentations – until 1977 when I left IAT (by that time – IPU). The first two of his seminars I attended were on robust statistics and on statistical approach to pattern recognition (in the current terminology, machine learning). As I recalled, these lectures took place in the old IAT building at Kalanchevka, in the large conference room, filled to capacity, with standing room only. Later on, in the new building at Profsouznay, his lectures – always full of creativity, wisdom, and humor – were equally well attended and became special events for all at the Institute – old and young alike.

Our personal relationship began in the late 80's or early 90's, when Y.Z. visited us at the University of Michigan. He gave a wonderful lecture on robust control at my seminar in the EECS Department, again impressing the audience by his scientific results and engaging personality. Since then, we met on many occasions, both in Russia and other countries at various conferences.

[1]The work of P. Alavian and S. M. Meerkov was supported in part by the US National Institute of Standards and Technology Grant Number 70NANB17H214. The work of L. Zhang was supported in part by the National Institute of Standards and Technology Grant Number 70NANB18H024.

Through these contacts and from previous observations, I came to know Y.Z. quite well and perceived two main features of his personality: civil courage and scientific excitement. Here are two examples:

In the difficult times of the late 60's and early 70's, a number of well-known scientists lost their jobs due to political stands in support of the dissident movement and alike. In this situation, Y.Z. had the courage to invite two of them into his Lab and was able to secure their appointments through the Institute administration. Needless to say, this saved their scientific lives and brought additional prestige to the Institute and Russian research on control as a whole.

As for the scientific excitement, here is a story. After his visit to Michigan, I was giving Y.Z. a ride to the airport. While on the highway, we got into a discussion on periodic controllers and their ability to ensure infinite sector of absolute stability for closed-loop system with any causal time-invariant plant. The discussion became so intense that we missed the airport exit. So, we had to turn around, but ... missed the exit again. Only on the third attempt, we reached the airport. After decades in science, Y.Z. was still excited with technical issues as a young scientist!

Indeed, Y.Z. remained young in many respects until, unfortunately the premature, end of his life. His personality and his scientific results will always be a bright page of control theory world-wide.

Semyon Meerkov

## 1. Introduction

Production systems are machines and buffers arranged so as to produce a desired product. Typically, production systems in modern manufacturing environment are quite complex, consisting of hundreds or even thousands of people and machines. While in many cases manufacturing equipment is highly automated, decision-making in production systems is practically always "manual" – most of the decision-making in daily operations or in design of continuous improvement projects is based on managerial common sense and experience, assisted, in some cases, by computer simulations. In this situation, it is not surprising that production losses are very large: in dozens of case-studies at various industrial plants, we have discovered that throughput losses of 20%-30% are quite common in practice.

To help recover these losses, we have developed an analytical theory of Production Systems Engineering (PSE). Every problem addressed in this theory has its origin on the factory floor; almost every solution obtained has been implemented in practice. The main results of this theory have been summarized in the textbook by Li and Meerkov (2009) and subsequent publications (see, for instance, Meerkov and Yan (2016) and Alavian et al. (2017)).

During the last 30 years, PSE methods have been applied at numerous industrial plants, consistently leading to substantial reduction of production losses and corresponding productivity improvement, often in 10-20% range. These applications have been carried out "manually": a team of researchers would develop a mathematical model of the production system at hand, apply PSE methods, and calculate optimal steps for continuous improvement with rigorously predicted results. In most cases, the suggested improvements have been implemented on the factory floor and led to productivity improvements close to those predicted analytically. Such applications have been carried out at plants of General Motors, Ford, Chrysler, Toyota, Volvo, Tesla, Visteon, Harley-Davidson, General Electric, Kroger, Kraft, MillerCoors, Lexmark, etc.

A drawback of this “manual” approach is that after the end of the project, the systems would often return to an inefficient state, perhaps due to other reasons for performance losses, which might have not existed previously. This experience led us to the idea of automating decision-making in production systems by creating a device, which could be used for decision-making on a continuous basis and by factory floor managerial personnel without special training in PSE or analytics in general.

Recently, we have developed such a device and call it Programmable Manufacturing Advisor (PMA). Conceptually, PMA is similar to PLC (Programmable Logic Controller, see, for instance, Bolton (2015)). The difference is that PLC is intended to automate production systems equipment, while PMA automates decision-making. Programming and installing a PMA at any production system makes it smart: it becomes capable of self-diagnosing and autonomously developing continuous improvement projects, leading to the productivity improvement desired by the Operations Manager (if at all possible). We call such systems PMA-based Smart Production Systems (PMA-Based SPS). Note that by making production systems smart, PMA contributes to one of the four areas of emphasis of Industry 4.0 – Smart Manufacturing (see Kagermann et al. (2013), Schlechtendahl et al. (2015), and Liao et al. (2017)).

PMA consists of three units:

- Information Unit (IU), which constructs and, based on factory floor equipment status measurements, continuously updates a mathematical model of the production system at hand.
- Analytics Unit (AU), which autonomously evaluates system health and efficacy of potential improvement scenarios; this is carried out using the analytical methods of PSE.
- Optimization Unit (OU), which calculates optimal steps for achieving the desired productivity improvement and offers them as an advice to the Operations Manager; this is carried out based on search techniques, similar to those used in Artificial Intelligence.

Accordingly, the architecture of PMA-based SPS is shown in Fig. 1. As one can see, PMA has two inputs and two outputs. One input is represented by the Measurements of the production system’s equipment status. Another input is the Desired Productivity Improvement (DPI) and the Admissible Action Space (AAS), both provided by the Operations Manager (OM). The DPI indicates the performance metric to be improved and the extent to which it must be modified (e.g., the throughput increased by 10% or the production lead time and work-in-process decreased by 30%, etc.). The AAS indicates means for achieving the required improvement (e.g., machine cycle time adjustment, skilled trades priority assignment, raw material release policy modification, or changes in the number of carriers in closed systems). The outputs of PMA are the System Health and Optimal Advice for continuous improvement, both provided to OM. Finally, the output of PMA-based SPS as a whole is the Resulting Productivity obtained after the improvement project has been implemented.

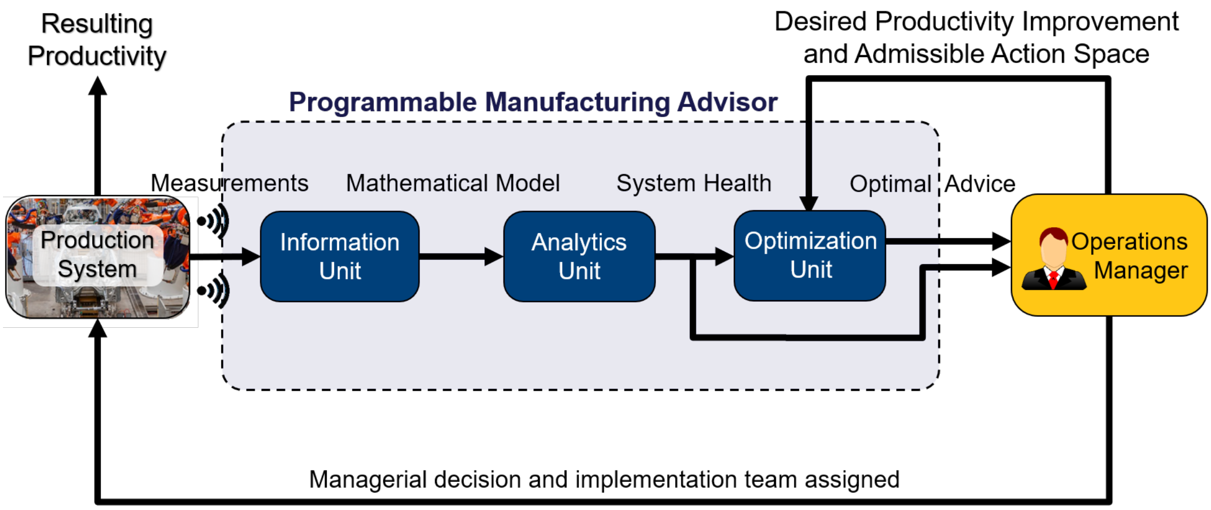



Рис. 1. PMA-based SPS architecture

The goals of this paper are to describe analytical foundations of PMA and its software/hardware implementations, as well as to demonstrate PMA-based SPS operation at an automotive underbody assembly system and a hot-dip galvanization plant.

To this end, Section 2 describes production system types, for which a PMA can be programmed, and the equipment parameters and performance metrics involved. Sections 3, 4, and 5 present analytical foundations of IU, AU and OU, respectively. In Section 6, the software/hardware implementations of PMA and workflow of PMA-based SPS are described. In Sections 7 and 8, screenshots demonstrating smart production systems operation are presented. Finally, Section 9 formulates the conclusion and topics for future work. The list of abbreviations and notations is given at the end of the paper.

## 2. Production Systems Types, Parameters, and Performance Metrics

### *2.1. Types of production systems*

The types of production systems addressed in SPS are:

- *Serial lines* (Fig. 2(a)), where the machines (circles) and buffers (rectangles) are arranged in a consecutive order to produce a desired product (part). If the processing times of all machines are the same, the line is called *synchronous*; otherwise, it is *asynchronous*.

- *Serial lines with product quality inspection devices* (Fig. 2(b)), where the black circles represent quality inspection devices, which are supposed to identify and remove defective parts, produced by the machines represented by shaded circles.

- *Serial lines with rework* (Fig. 2(c)), where the defective parts are repaired and returned for reprocessing.

- *Closed serial lines* (Fig. 2(d)), where the parts are transported on carriers, and after a part is produced, the carrier is returned to the return buffer (square), which makes the carriers available for the incoming parts.

- *Assembly systems* (Fig. 2(e)), where two or more serial lines produce subassemblies to be merged in the main line. (Note that serial lines may also contain assembly operations, but the subassemblies involved are either purchased parts or supplied by other departments.)
- *Multi-Job Production systems* (Fig. 2(f)), where different job-types (e.g., $J_1$ and $J_2$) are produced by the same sequence of manufacturing operations, perhaps with different processing times. In addition to other parameters (described in the next subsection), these systems are characterized by the product-mix of parts being manufactured.

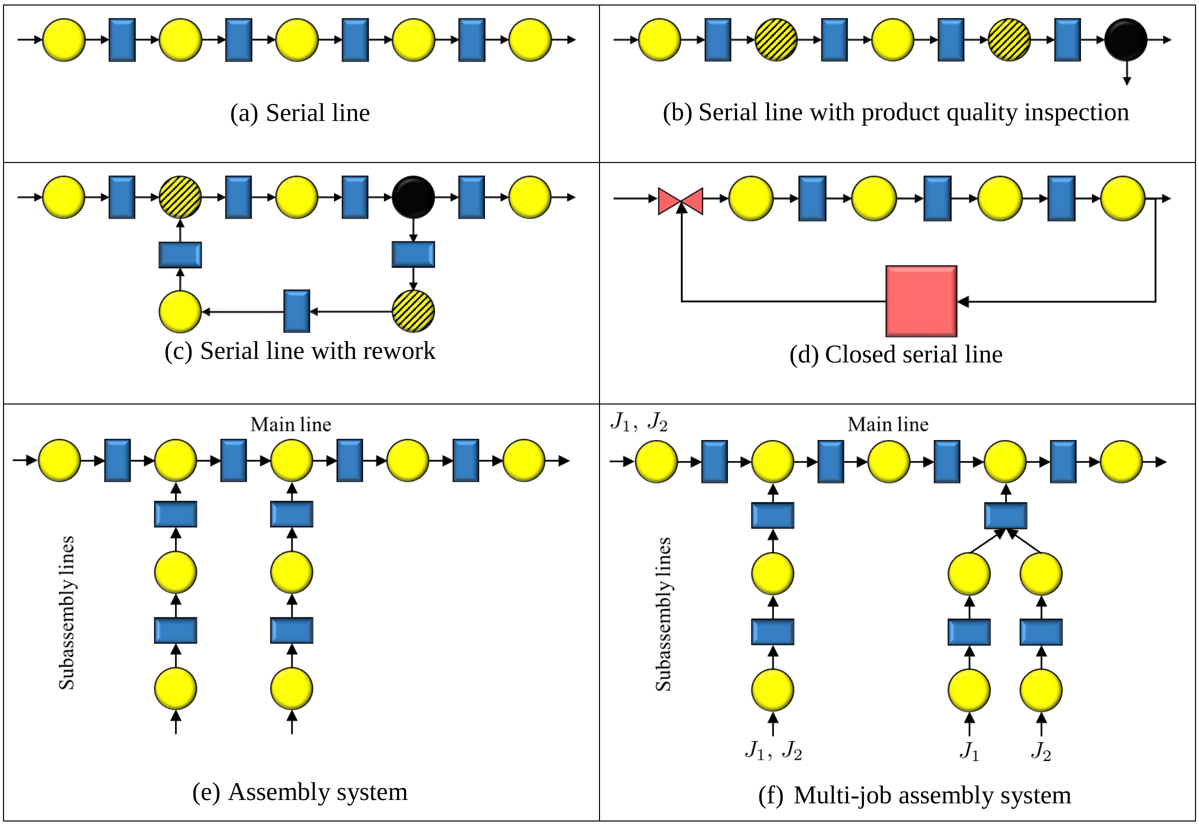



Рис. 2. Production system types addressed in SPS

### *2.2. Parameters of machines and buffers*

The following machine parameters are used in SPS analysis and design:

- *Machine cycle time* ($\tau$) – the time necessary to process a part by a machine (often measured in seconds). In many operations, $\tau$ is constant or almost constant (i.e., random, but with a small coefficient of variation).
- *Machine capacity* ($c$) – the number of parts a machine can produce per unit of time. If the unit of time is an hour and the cycle time is in seconds, the machine capacity is

(1) $$c = \frac{3600}{\tau} \text{parts/hour}.$$

- In most production systems, machines experience random downtime due to breakdowns. The *average uptime* and *average downtime* are denoted as $T_{up}$ and $T_{down}$ (typically, in minutes). In practice, $T_{up}$ and $T_{down}$ are often referred to as *mean time between failures* (MTBF) and *mean time to repair* (MTTR), respectively. In this paper, $T_{up}$ and $T_{down}$ or MTBF and MTTR are used interchangeably.
- *Machine quality parameter* ($g$) – the probability that a part produced is non-defective.

Thus, the machines are characterized by four independent parameters: $\{\tau, T_{up}, T_{down}, g\}$.

The following two non-independent parameters are also used as machine characteristics:

- *Machine efficiency* ($e$) – the fraction of time the machine is up:

$$e = \frac{T_{up}}{T_{up} + T_{down}}. \tag{2}$$

- *Machine stand-alone throughput* ($SAT$) – the average number of parts produced by the machine per unit of time (e.g., hour), when it is neither starved nor blocked:

$$SAT = ce. \tag{3}$$

  Note that if $g < 1$, the stand-alone throughput of non-defective parts is $SAT_g = ceg$.

As far as the buffers are concerned, we assume that each buffer is characterized by a single non-negative integer $N$, which represents its storing capacity.

### *2.3. Performance metrics*

Performance metrics are functions of machine and buffer parameters. The following are of importance in practice and, therefore, in PMA-based SPS.

- *Throughput* ($TP$) – the average number of (non-defective) parts produced by the system per unit of time (e.g., per hour). Throughput per machine cycle time in synchronous systems is referred to as *Production Rate* ($PR$). Obviously, $TP = H \cdot PR$, where $H$ is the number of cycles per unit of time (e.g., per hour).
- *Work-in-process* in the $i$-th buffer ($WIP_i$) – the average number of parts in the $i$-th buffer.
- *Blockage* of the $i$-th machine ($BL_i$) – the probability of the event that $m_i$ is up, $b_i$ is full, and $m_{i+1}$ takes no parts from the buffer.
- *Starvation* of the $i$-th machine ($ST_i$) – the probability of the event that $m_i$ is up and $b_{i-1}$ is empty.
- *Lead time* ($LT$) – the average time a part spends in the system, being processed or waiting for processing.

- *Scrap rate* ($SR_i$) – the average number of defective parts rejected by the $i$-th inspection machine per unit of time.

For systems operating on the factory floor, these performance metrics can be evaluated statistically, using factory floor measurements. However, since SPS is intended, in particular, to evaluate efficacy of potential improvement projects, a statistical approach is not applicable. Instead, either computer simulation or analytical methods must be used. The computer simulation approach requires exact models of the production systems at hand, with all the details involved (see, Law et al. (1991), Jerry (2005), and Altiok and Melamed (2010)). Sometimes, such models are referred to as "digital twins." Since creating digital twins for complex systems is practically impossible, we use an analytical approach, which is applicable to simplified models of the systems at hand.

Analytical methods for production systems analysis, improvement, and design have been under development for over 50 years, starting from the pioneering papers of Sevast'yanov (1962)[2] and Buzacott (1967) and continuing in the subsequent research, summarized in monographs by Viswanadham and Narahari (1992), Askin and Standridge (1993), Buzacott and Shanthikumar (1993), Papadopoulos et al. (1993), Gershwin (1994), Altiok (1997), Li and Meerkov (2009), Papadopoulos et al. (2009), and Curry and Feldman (2009). In the current work, we use the methods of PSE (Li and Meerkov (2009), Meerkov and Yan (2016) and Alavian et al. (2017)), mostly because they provide provable guarantees of convergence of recursive performance evaluation procedures and address, in a unified manner, various analysis and design problems (e.g., throughput evaluation, bottleneck identification, leanness, lead time analysis, characteristic curves, multi-job performance portraits, etc.). To enable PSE application on the factory floor, we have developed a software package referred to as *PSE Toolbox*. A screenshot of its home page is shown in Fig. 3. Its ten tools (with functionalities indicated in Fig. 3) can be used for production systems analysis and design. A demo of this toolbox can be found at `pse.smartproductionsystems.com`.


Рис. 3. Screenshot of *PSE Toolbox* homepage

[2]It should be pointed out that in 1957, A.N. Kolmogorov gave a lecture at a meeting of the Moscow Mathematical Society devoted to production systems. Unfortunately, no record of this presentation could be found. Since Sevast'yanov was, at the time, a graduate student of Kolmogorov, it is reasonable to assume that Kolmogorov's lecture contained ideas close to those of Sevast'yanov's paper.

## 3. Information Unit

This and two subsequent sections describe theoretical/analytical foundations of PMA units: IU, AU, and OU, respectively.

The theoretical foundations of IU stem from the coupling between IU and AU. This is because the model employed in AU dictates "what to measure" and "how to measure" for IU. Thus, the problem of mathematical modeling of production systems is at the core of IU design.

The mathematical model (MM) is intended to represent a *simplified version* of the production system, capturing, however, its main features. MM consists of a *structural model* and a *parametric model*. The structural model aims to reduce the system layout to one of the standard block diagrams of Fig. 2. The parametric model is intended to provide parameters of the machines and buffers included in the structural model. While the methods of mathematical modeling of production systems are described in Li and Meerkov (2009, Chapter 3), we note here that MM requires measurements leading to identification of the machine and buffer parameters, i.e., $\{\tau, T_{up}, T_{down}, g, N\}$.

Typically, the machine cycle time, $\tau$, can be easily identified by measuring the part processing time. If the machine involves loading/unloading operations, their duration must be added to the processing time, and, thus, the cycle time will include the total time of part processing and handling.

The identification of the average up- and downtime, $T_{up}$ and $T_{down}$, requires more efforts. To accomplish this, the duration of each randomly occurring up- and downtime, $t_{up,i}$ and $t_{down,i}$, (where $i$ denotes the $i$-th occurrence of the up- and downtime) must be measured. Then $T_{up}$ and $T_{down}$ can be calculated according to

$$T_{up} = \frac{\sum_{i=1}^{n} t_{up,i}}{n}, \quad T_{down} = \frac{\sum_{i=1}^{n} t_{down,i}}{n}, \tag{4}$$

where the number of downtime occurrences, $n$, should be sufficiently large to guarantee statistically reliable estimates. Recently, we have developed a theory for selecting the smallest number of measurements, $n^*$, which is necessary and sufficient to guarantee the desired accuracy of $T_{up}$ and $T_{down}$ estimates (Alavian et al. (2019)).

To identify the quality parameter, $g$, one must monitor the total number of parts, $Q_{total}$, produced, say, per shift, and the number of non-defective parts among them, $Q_g$, and then evaluate $g$ by

$$g = \frac{Q_g}{Q_{total}}. \tag{5}$$

Note that the values of $\tau$, $T_{up}$, $T_{down}$, and $g$ must be monitored continuously, since they do change in time.

The buffer capacity, $N$, is typically constant and can be obtained by evaluating the number of parts a buffer is capable of storing. If a conveyor serves as a buffer, its capacity can be evaluated using the method discussed in Li and Meerkov (2009, Chapter 3).

## 4. Analytics Unit

The Analytics Unit is at the heart of PMA operation. Therefore, it is described here in more details.

### *4.1. Analytical foundations*

The foundations of AU are the analytics of PSE. They are based on exact performance evaluation of two-machine systems (using Markov chain techniques) and recursive aggregation procedures for approximate evaluation of larger systems. An illustration of these techniques is given below in terms of synchronous serial lines with machines obeying the exponential reliability model. According to this reliability model, machine up- and downtimes are exponential random variables with parameters $\lambda$ and $\mu$, respectively.

A system of two exponential machines with parameters $(\lambda_1, \mu_1)$ and $(\lambda_2, \mu_2)$ and a buffer of capacity $N$ is described by a continuous-time Markov chain. Its stationary probability distribution can be calculated, leading to the following expressions for its production rate, $PR$, and probabilities of first machine blockage, $BL_1$, and the second machine starvation, $ST_2$. ($WIP$ also has been calculated, but is omitted here due to space limitations.)

$$\begin{aligned} PR &= e_2[1 - Q(\lambda_1, \mu_1, \lambda_2, \mu_2, N)] \\ &= e_1[1 - Q(\lambda_2, \mu_2, \lambda_1, \mu_1, N)], \end{aligned} \tag{6}$$

$$BL_1 = e_1 Q(\lambda_2, \mu_2, \lambda_1, \mu_1, N), \tag{7}$$

$$ST_2 = e_2 Q(\lambda_1, \mu_1, \lambda_2, \mu_2, N), \tag{8}$$

where

$$Q(\lambda_1, \mu_1, \lambda_2, \mu_2, N) = \begin{cases} \frac{(1-e_1)(1-\phi)}{1-\phi e^{-\beta N}}, & \text{if } \frac{\lambda_1}{\mu_1} \neq \frac{\lambda_2}{\mu_2}, \\ \frac{\lambda_1(\lambda_1+\lambda_2)(\mu_1+\mu_2)}{(\lambda_1+\mu_1)[(\lambda_1+\lambda_2)(\mu_1+\mu_2)+\lambda_2\mu_1(\lambda_1+\lambda_2+\mu_1+\mu_2)N]}, & \text{if } \frac{\lambda_1}{\mu_1} = \frac{\lambda_2}{\mu_2}, \end{cases} \tag{9}$$

$$e_i = \frac{\mu_i}{\lambda_i + \mu_i}, \quad i = 1, 2, \tag{10}$$

$$\phi = \frac{e_1(1 - e_2)}{e_2(1 - e_1)}, \tag{11}$$

$$\beta = \frac{(\lambda_1 + \lambda_2 + \mu_1 + \mu_2)(\lambda_1\mu_2 - \lambda_2\mu_1)}{(\lambda_1 + \lambda_2)(\mu_1 + \mu_2)}. \tag{12}$$

It turns out that a similar analysis for longer lines cannot be carried out in a closed-form. Therefore, approximations are necessary. As mentioned before, we have developed such approximations using a recursive aggregation procedure. To describe this procedure, consider a serial line with $M$ machines, denoted as $m_i$, each characterized by parameters $(\lambda_i, \mu_i)$, $i = 1, ..., M$, and $M - 1$ buffers, denoted as $b_i$, with capacity $N_i$, $i = 1, ..., M - 1$, separating each pair of consecutive machines. Aggregate the last two machines, $m_{M-1}$ and $m_M$, into a single exponential machine denoted as $m^b_{M-1}$, where the superscript $b$ stands for the *backward aggregation.* The parameters of $m^b_{M-1}$ are selected using the second expression in (6) (see Li and Meerkov (2009, Subsection 11.1.2) for details). Next, aggregate this machine, i.e., $m^b_{M-1}$, with $m_{M-2}$ to obtain another aggregated machine, $m^b_{M-2}$. Continue this procedure until all the machines are aggregated into $m^b_1$, which completes the backward phase of the aggregation procedure.

The subsequent *forward aggregation* consists of the following: Aggregate the first machine $m_1$ with the aggregated version of the rest of the line, i.e., $m_2^b$. This results in the aggregated machine $m_2^f$, where $f$ stands for the forward aggregation. The parameters of $m_2^f$ are selected using the first expression of (6). Next, aggregate $m_2^f$ with $m_3^b$, resulting in $m_3^f$ and so on until all the machines are aggregated into $m_M^f$, which completes the forward phase of the aggregation procedure. Then, iterate between backward and forward aggregations. Analytically, this recursive procedure can be represented as follows:

$$
\begin{aligned}
\mu_i^b(s+1) &= \mu_i(1 - Q(\lambda_{i+1}^b(s+1), \mu_{i+1}^b(s+1), \lambda_i^f(s), \mu_i^f(s), N_i)), \\
&\qquad\qquad\qquad\qquad i = 1, \dots, M-1, \\
\lambda_i^b(s+1) &= \lambda_i + \mu_i Q(\lambda_{i+1}^b(s+1), \mu_{i+1}^b(s+1), \lambda_i^f(s), \mu_i^f(s), N_i), \\
&\qquad\qquad\qquad\qquad i = 1, \dots, M-1, \\
\mu_i^f(s+1) &= \mu_i(1 - Q(\lambda_{i-1}^f(s+1), \mu_{i-1}^f(s+1), \lambda_i^b(s+1), \mu_i^b(s+1), N_{i-1})), \\
&\qquad\qquad\qquad\qquad i = 2, \dots, M, \\
\lambda_i^f(s+1) &= \lambda_i + \mu_i Q(\lambda_{i-1}^f(s+1), \mu_{i-1}^f(s+1), \lambda_i^b(s+1), \mu_i^b(s+1), N_{i-1}), \\
&\qquad\qquad\qquad\qquad i = 2, \dots, M, \\
& s = 1, 2, \dots,
\end{aligned}
\tag{13}
$$

with initial conditions

$$\lambda_i^f(0) = \lambda_i, \quad \mu_i^f(0) = \mu_i, \quad i = 2, \dots, M-1,$$

and boundary conditions

$$
\begin{aligned}
&\lambda_1^f(s) = \lambda_1, \quad \mu_1^f(s) = \mu_1, \quad s = 1, 2, \dots, \\
&\lambda_M^b(s) = \lambda_M, \quad \mu_M^b(s) = \mu_M, \quad s = 1, 2, \dots,
\end{aligned}
$$

where function $Q$ is defined by (9).

Теорема 1. *Recursive procedure* (13) *has the following properties:*

*(i) The sequences* $\lambda_2^f(s)$, ..., $\lambda_M^f(s)$, $\mu_2^f(s)$, ..., $\mu_M^f(s)$, *and* $\lambda_1^b(s)$, ..., $\lambda_{M-1}^b(s)$, $\mu_1^b(s)$, ..., $\mu_{M-1}^b(s)$, $s = 1, 2, \dots$ *are convergent with the limits denoted as* $\lambda_i^f$, $\mu_i^f$, $\lambda_i^b$ *and* $\mu_i^b$.

*(ii) These limits are unique solutions of the following equations:*

$$
\begin{aligned}
\mu_i^b &= \mu_i[1 - Q(\lambda_{i+1}^b, \mu_{i+1}^b, \lambda_i^f, \mu_i^f, N_i)], \quad i = 1, \cdots, M-1, \\
\lambda_i^b &= \lambda_i + \mu_i Q(\lambda_{i+1}^b, \mu_{i+1}^b, \lambda_i^f, \mu_i^f, N_i), \quad i = 1, \cdots, M-1, \\
\mu_i^f &= \mu_i[1 - Q(\lambda_{i-1}^f, \mu_{i-1}^f, \lambda_i^b, \mu_i^f, N_{i-1})], \quad i = 2, \cdots, M, \\
\lambda_i^f &= \lambda_i + \mu_i Q(\lambda_{i-1}^f, \mu_{i-1}^f, \lambda_i^b, \mu_i^f, N_{i-1}), \quad i = 2, \cdots, M.
\end{aligned}
\tag{14}
$$

*(iii) In addition, these limits satisfy the relationships:*

$$
\begin{aligned}
e_M^f &= e_1^b \\
&= e_{i+1}^b[1 - Q(\lambda_i^f, \mu_i^f, \lambda_{i+1}^b, \mu_{i+1}^b, N_i)] \\
&= e_i^f[1 - Q(\lambda_{i+1}^b, \mu_{i+1}^b, \lambda_i^f, \mu_i^f, N_i)], \quad i = 1, \dots, M-1,
\end{aligned}
\tag{15}
$$

*where*

$$e_i^f = \frac{\mu_i^f}{\lambda_i^f + \mu_i^f}, \quad e_i^b = \frac{\mu_i^b}{\lambda_i^b + \mu_i^b}, \quad i = 1, \dots, M.$$

□

*Proof*: See Li and Meerkov (2009, Chapter 11).

Statement (iii) implies that, from the point of view of each buffer $b_i$, $i = 1, .., M-1$, the upstream of the serial line is represented by the aggregated machine $m_i^f$ and the downstream by the aggregated machine $m_{i+1}^b$. Therefore, the performance metrics of such two-machine line can be evaluated using expressions (6)-(9). In other words, the estimates of $PR$, $BL_i$, and $ST_i$ can be introduced as follows:

$$\begin{aligned} \widehat{PR} &= e_{i+1}^b[1 - Q(\lambda_i^f, \mu_i^f, \lambda_{i+1}^b, \mu_{i+1}^b, N_i)] \\ &= e_i^f[1 - Q(\lambda_{i+1}^b, \mu_{i+1}^b, \lambda_i^f, \mu_i^f, N_i)], \qquad i = 1, \dots, M-1, \end{aligned} \tag{16}$$

$$\widehat{BL}_i = e_i Q(\lambda_{i+1}^b, \mu_{i+1}^b, \lambda_i^f, \mu_i^f, N_i), \quad i = 1, \dots, M-1, \tag{17}$$

$$\widehat{ST}_i = e_i Q(\lambda_{i-1}^f, \mu_{i-1}^f, \lambda_i^b, \mu_i^b, N_{i-1}), \quad i = 2, \dots, M. \tag{18}$$

The accuracy of these estimates has been evaluated numerically, and it has been shown that the error of $\widehat{PR}$ in most cases is well within 1%. This aggregation procedure and its generalizations for other types of production systems have been implemented in AU for its Performance Analysis functionality.

Equations (14) contain all qualitative and quantitative properties of production systems at hand. Deriving these properties, solutions to a number of industrially important problems have been obtained. Due to space limitations, only three of them, with a major role in PMA, are discussed below: the problem of bottleneck identification, the problem of product-mix performance portrait, and the problem of feedback control of raw material release to ensure the desired lead time.

### *4.2. Bottleneck identification problem*

In practice, the bottleneck is typically defined as the worst machine in the system as far as its $SAT$ is concerned. This definition does not take into account the structure of the system, buffers capacity, position of the machine in the system, etc. To account for these features, we define the bottleneck as follows:

**Definition 4.1:** The *bottleneck* (BN) is the machine with the largest effect on the system throughput ($TP$), quantified as

$$\frac{\partial TP}{\partial c_i} > \frac{\partial TP}{\partial c_j}, \forall j \neq i, \tag{19}$$

where, as mentioned in Section 2, $c_i$ is the $i$-th machine capacity.

As it turns out, the machine with the smallest $SAT$ is necessarily the BN only if the system has infinite buffers or is umimprovable with respect to workforce reallocation (i.e., has all buffers on the average half-full). In all other cases, any machine, including the one with the largest $SAT$, can be the BN.

Unfortunately, the derivatives involved in (19) cannot be calculated analytically because $TP$ (as a function of $c_1, \ldots, c_M$) for $M > 2$ cannot be represented in closed-form. Therefore, the following simplified procedure has been developed (see Li and Meerkov (2009, Chapters 5 and 13)):

- Evaluate $BL$ and $ST$ of all machines included in the MM's structural model.
- Assign arrows between 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 $m_i$.
- If there is only one machine with no emanating arrows, it is the BN (in the sense of (19)).
- If there are multiple machines with no emanating arrows, the one with the largest severity is the primary BN, where the bottleneck severity is defined by:

$$\begin{aligned} S_1 &= |ST_2 - BL_1|, \\ S_i &= |ST_{i+1} - BL_i| + |ST_i - BL_{i-1}|, i = 2, ..., M-1, \\ S_M &= |ST_M - BL_{M-1}|. \end{aligned} \tag{20}$$

This procedure has also been implemented in AU as a part of its Performance Analysis functionality.

### *4.3. Product-mix performance portrait of multi-job production systems*

The idea of product-mix performance portrait (PMPP) for multi-job production systems is motivated by the state-space portrait of dynamical systems (see, for instance, Andronov et al. (1959) and Khalil (2002)). Indeed, the latter allows to graphically represent the system trajectories for various initial conditions. Similarly, the former allows to graphically represent the system performance for various values of product-mix.

More specifically, PMPP represents $TP$ and BN as functions of the product-mix and, thus, allows the manager to assess the system behavior for all product-mixes, which often change on the daily basis. This leads to managerial actions corresponding to the changes in $TP$. The analytical methods for calculating $TP$ have been developed in Alavian et al. (2017) and implemented as an AU functionality.

### *4.4. Feedback control of raw material release*

An important characteristic of production systems is the Characteristic Curve (CC), which describes production lead time ($LT$) as a function of the raw material release rate or throughput. In systems with hardware-unlimited buffers, this function has a knee-type behavior (see Fig. 4, where the dot indicates the knee). Operating the system below the knee is not efficient, since $TP$ can be increased without an appreciable increase in $LT$. Operating above the knee is also counterproductive $-$ $LT$ becomes extremely large without a significant increase of $TP$. Thus, the desirable operating point is at the knee.

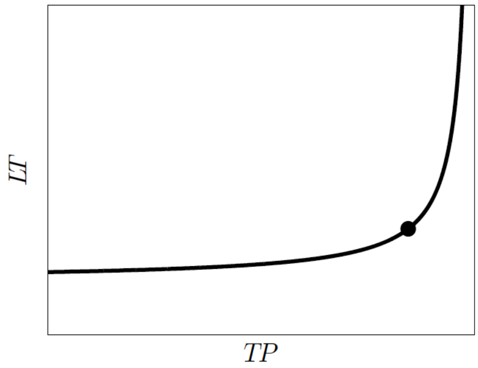



Рис. 4. Characteristic curve

To ensure the system operation at the knee or at any desired point on CC, the raw material release rate must be controlled. If the machine parameters are known precisely, this can be accomplished by calculating the release rate corresponding to the desired point on CC. However, when the machine parameters are not known precisely (e.g., changing in time), the open-loop control of the release rate would not work and a feedback approach is necessary. Recently, this has been developed in Meerkov and Yan (2016) for the case of serial lines. It turned out that a simple relay-type control law can be used for this purpose. Below is the description of this law.

Consider a serial line with infinite buffers and $M$ machines with exponential reliability model. An estimate of lead time in such a system is given by

$$\widehat{LT} = M\tau + \sum_{i=0}^{M-1} \left( \frac{e_i}{\mu_i} + \frac{e_{i+}}{\mu_{i+1}} \right) \left( \frac{1 - e_{i+1}}{e_{i+1} - e_0} \right), \tag{21}$$

where, as before, $e_i$, $i = 1, \dots, M$, is the machine efficiency, $\mu_i$, $i = 1, \dots, M$, is the inverse of MTTR, and $e_0$ is the efficiency of the virtual machine modeling the raw material release mechanism.

For any admissible desired lead time, $LT_d$, the release rate, $e_0^*$, which ensures this lead time, is the unique real root of the following $M$-th order polynomial equation:

$$\begin{aligned} &(LT_d - M\tau) \prod_{i=0}^{M-1} (e_{i+1} - e_0) - (1 - e_1) \left( \frac{e_0}{\mu_0} + \frac{e_1}{\mu_1} \right) \prod_{i=1}^{M-1} (e_{i+1} - e_0) \\ &- \prod_{i=1}^{M-1} \left( (1 - e_{i+1}) \left( \frac{e_i}{\mu_i} + \frac{e_{i+1}}{\mu_{i+1}} \right) \prod_{j=0, j \neq i}^{M-1} (e_{j+1} - e_0) \right) = 0. \end{aligned} \tag{22}$$

Based on this $e_0^*(LT_d)$, the deterministic hourly release rate, $E_H^*$ is defined as

$$E_H^* = \lfloor H e_0^*(LT_d) \rfloor, \tag{23}$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than $x$, and $H$ is the number of cycles in an hour.

Finally, to define the closed-loop control law of raw material release, represent the nominal work-in-process in the system as follows:

$$WIP_{nominal} = \frac{e_0^*}{\tau}(LT_d - M\tau). \tag{24}$$

Based on the above, the relay-type feedback release control law is given by

$$E(s+1) = \begin{cases} E_H^*, & \text{if } WIP_{total}(s) \leqslant WIP_{nominal}, \\ 0, & \text{otherwise,} \end{cases} \tag{25}$$

where $s = 0, 1, \dots,$ is the index of the release interval; $E(s+1)$ is the amount of raw material released at the beginning of release interval $s+1$; $E_H^*$ is defined in (23); and $WIP_{total}(s)$ is the real-time total work-in-process in the system at the end of release interval $s$.

This control law has been implemented in the Characteristic Curve functionality of AU.

## 5. Optimization Unit

The theoretical foundations of OU are based on search procedures typically used in the area of Artificial Intelligence. The reason is in the following:

As mentioned in Section 1, OU is intended to calculate an optimal advice to Operations Manager, based on the entered DPI/AAS and consistent with the System Health provided by AU. While this is indeed an optimization problem, the usual optimization tools, such as linear and nonlinear programming, cannot be used to find a solution. This is because the performance metrics to be optimized (e.g., throughput, work-in-process, production lead time, etc.) cannot be expressed as explicit functions of the system parameters (e.g., machine cycle time, MTBF, MTTR, buffer capacity, number of carriers, etc.). The situation here is similar to that in the area of computer chess games, which led to the development of various search techniques, guided by the knowledge of the game and intuition of the designers. Similarly, in the case of OU, the only available approach is to use search procedures in the parameter space, enhanced by the fundamental laws of Production Systems Engineering and, in particular, qualitative properties of the performance metrics (e.g., continuity, monotonicity, reversibility, and improvability). These properties have been investigated in Li and Meerkov (2009) and Meerkov and Yan (2016), and have been utilized in the algorithms developed for OU.

## 6. PMA Software/Hardware Implementations and PMA-based SPS Workflow

While Sections 3-5 outline analytical/theoretical foundations of PMA, the current section provides a few remarks on its software and hardware implementations as well as on the PMA-based SPS workflow.

### *6.1. Software and hardware implementations*

The PMA software is implemented as a web-based application. Its back-end, written in Node JS programming language, is responsible for storing and processing of production systems and users' data. All PMA calculations are implemented in the backend. The front-end is developed using JavaScript. It handles the data representation and visualization. The PMA software can be accessed from the cloud as the most economic and scalable option, or be installed on-premise, which brings faster performance and improved security.

The PMA is installed on the factory floor as a box consisting of a server and a display (see Fig. 5). The server is intended to store historical data on manufacturing equipment status (obtained through factory floor measurements) and to maintain the PMA's software. The display allows for entering the managerial inputs and presenting the PMA outputs.


Рис. 5. PMA box

*6.2. PMA-based SPS workflow*

The PMA-based SPS workflow is illustrated by a screenshot of Fig. 6. Its "Systems" block lists the production systems for which the PMA has been programmed. The five subsequent blocks represent the PMA per se and its interactions with the Operations Manager. Finally, the last block reports the measured productivity improvement, after the suggested plant modifications have been implemented. A few comments on the operation of these blocks are in order.

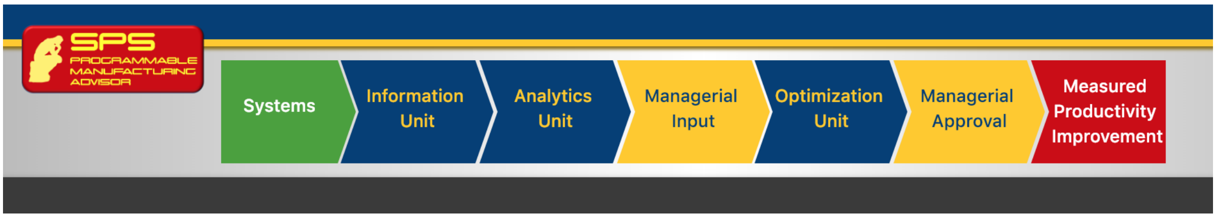



Рис. 6. PMA-based SPS workflow

As mentioned before, the IU block maintains and continuously updates mathematical models of the production systems for which the PMA has been programmed. These models are typically constructed by PMA programmers with the help of the Modeling module of the PSE Toolbox.

Using these models, AU carries out calculations in order to provide the following outputs:

- Performance Analysis: It shows the system's throughput, work-in-process, probabilities of blockages and starvations, and the bottleneck.

- System Diagnostics: It provides information on causes of production losses and buffering efficacy.

- What-if Analyses: It quantifies effects of potential machine and buffer parameter modifications on the overall system performance.

- System Health: It summarizes the main features of all above mentioned outputs; this information assists the Operations Manager in formulating the goals of the desired performance improvement.

If needed, the AU may be programmed to produce system-specific outputs such as:

- Product-mix Performance Portraits – for multi-job production systems.

- Characteristic Curves – for systems with hardware-unlimited buffers.

The Managerial Input block allows the Operations Manager to enter one or more scenarios of potential improvement. The OU output displays which of these scenarios can be implemented on the factory floor to achieve the desired improvements and which cannot, along with the optimal equipment modifications, necessary for achieving the predicted improvements. The Managerial Approval block allows the OM to select a specific scenario for implementation on the factory floor. Finally, the Measured Productivity Improvement block displays the results of the continuous improvement project implemented on the factory floor and compares them with those predicted by OU.

Two demonstrations of PMA-based SPS operation are presented in the subsequent sections. In both of these demonstrations, due to proprietary reasons, the system parameters have been modified, and the efficacy of the continuous improvement projects has been evaluated using discrete event simulations of the plants involved.

## 7. Demonstration: Smart Automotive Underbody Assembly System[3]

The automotive underbody assembly is a large-volume production system, manufacturing two job-types in multi-job regime. Its layout is shown in Fig. 7, where WH and SB denote Wheel Housing and Seat Bar, respectively. This section presents screenshots demonstrating PMA operation programmed for this system.

The output of IU is shown in Fig. 8. It provides the system's structural and parametric models, as well as the product-mix, $r_1 = 0.4$ and $r_2 = 0.6$, where $r_i$, $i = 1, 2$, is the fraction of job-type $i$ being manufactured.

One of the AU outputs, Product-mix Performance Portrait, is shown in Fig. 9. It characterizes the system's performance for all product-mixes. Specifically, it indicates that $TP$ is a non-monotonic (concave) function of $r_1$, reaching its maximum in the range $r_1 \in (0.31, 0.53)$. As for the BN, it is Op. 2 (Motor Compartment 2) for $r_1 \in [0, 0.31)$; Op. 6 (Wheel Housing) for $r_1 \in (0.31, 0.53)$; and Op. 1 (Motor Compartment 1) for $r_1 \in (0.53, 1]$.

Another AU output, Performance Analysis, shown in Fig. 10, displays the values of machines efficiency, stand-alone throughput, and blockages and starvations, along with the

[3]The authors acknowledge the assistance of industrial partners in carrying out this case-study.

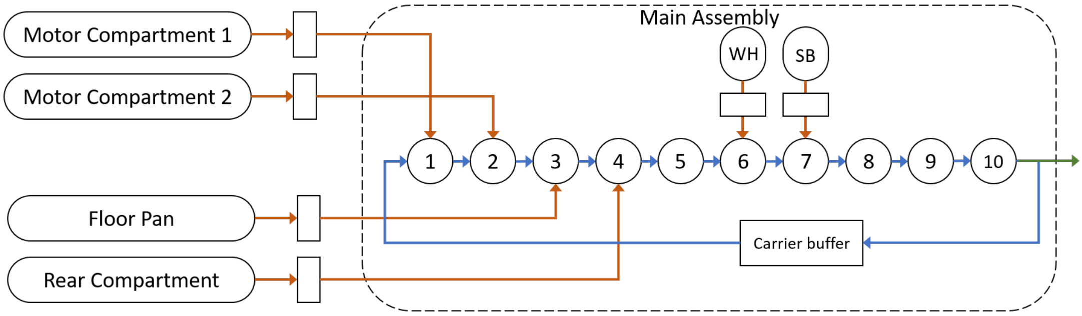
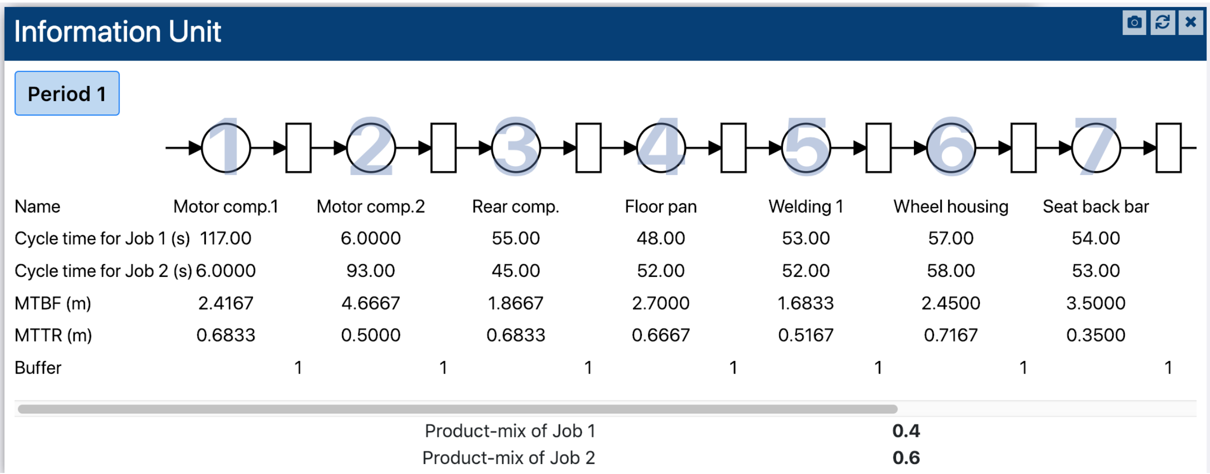



Рис. 7. Automotive underbody assembly system layout




| Name | Motor comp.1 | Motor comp.2 | Rear comp. | Floor pan | Welding 1 | Wheel housing | Seat back bar |
|---|---|---|---|---|---|---|---|
| Cycle time for Job 1 (s) | 117.00 | 6.0000 | 55.00 | 48.00 | 53.00 | 57.00 | 54.00 |
| Cycle time for Job 2 (s) | 6.0000 | 93.00 | 45.00 | 52.00 | 52.00 | 58.00 | 53.00 |
| MTBF (m) | 2.4167 | 4.6667 | 1.8667 | 2.7000 | 1.6833 | 2.4500 | 3.5000 |
| MTTR (m) | 0.6833 | 0.5000 | 0.6833 | 0.6667 | 0.5167 | 0.7167 | 0.3500 |
| Buffer | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | |
|---|---|
| Product-mix of Job 1 | **0.4** |
| Product-mix of Job 2 | **0.6** |

Рис. 8. IU output of PMA programmed for smart automotive underbody assembly system

buffers occupancy and system's throughout, all calculated for the product-mix (0.4,0.6). As one can see, $TP = 42.85$ JPH and $TP$s for job-types 1 and 2 are 17.14 JPH and 25.71 JPH, respectively. Also it shows that the BN, identified using the arrow-based rule of Subsection 4.2, is Op. 6.

The third AU output, System Diagnostics, quantifies the losses due to buffers (5.58 JPH) and losses due to machines (13.43 JPH), calculated according to:

$$\text{Losses due to buffers} = TP \text{ if buffers were infinite} - TP, \tag{26}$$

$$\text{Losses due to machines} = TP \text{ if machines had no breakdowns} - TP \text{ if buffers were infinite}. \tag{27}$$

Thus, the total $TP$ losses are 19.01 JPH, i.e., 30.7%. Also, this output quantifies buffering potency, which is a measure of buffers efficacy in rejecting perturbations due to machine breakdowns. The term "Weakly potent," shown in Fig. 11, indicates that the worst machine is indeed the BN, but the system $TP$ is too much lower than the $SAT$ of the BN machine.

The next AU output, What-if Analysis (Fig. 12), represents the effect of changing machine parameters on system's $TP$. It shows that if MTTR of Wheel Housing is decreased by about 25%, $TP$ increases, almost linearly, from 42.85 JPH to 45.7 JPH, but remains constant after that. The reason can be found looking at the right-hand side

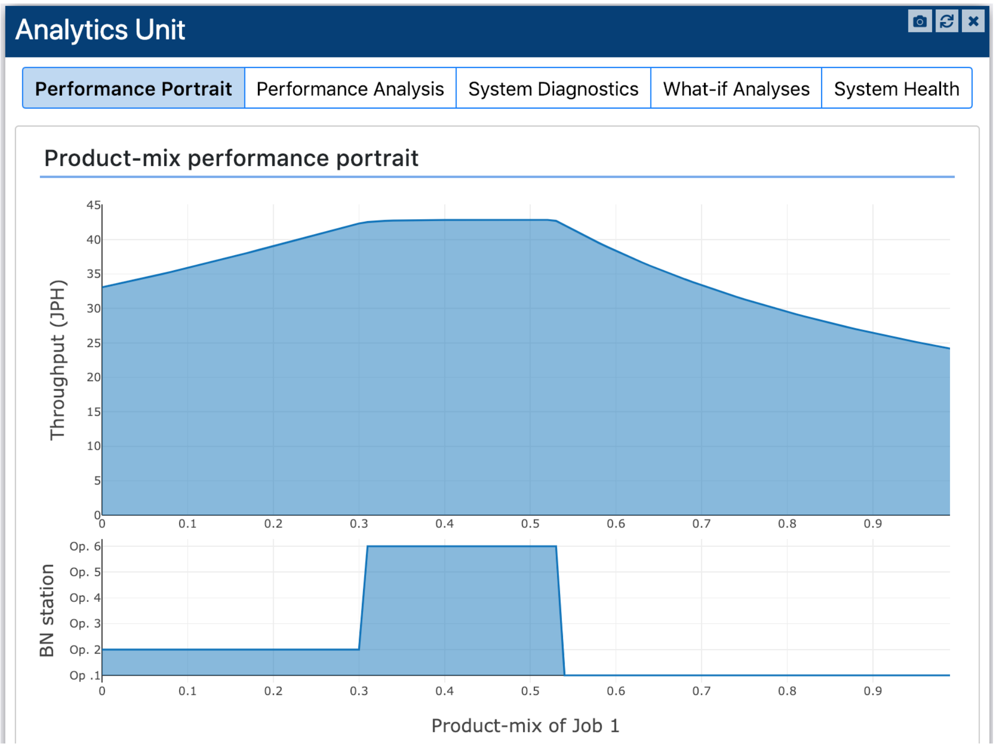
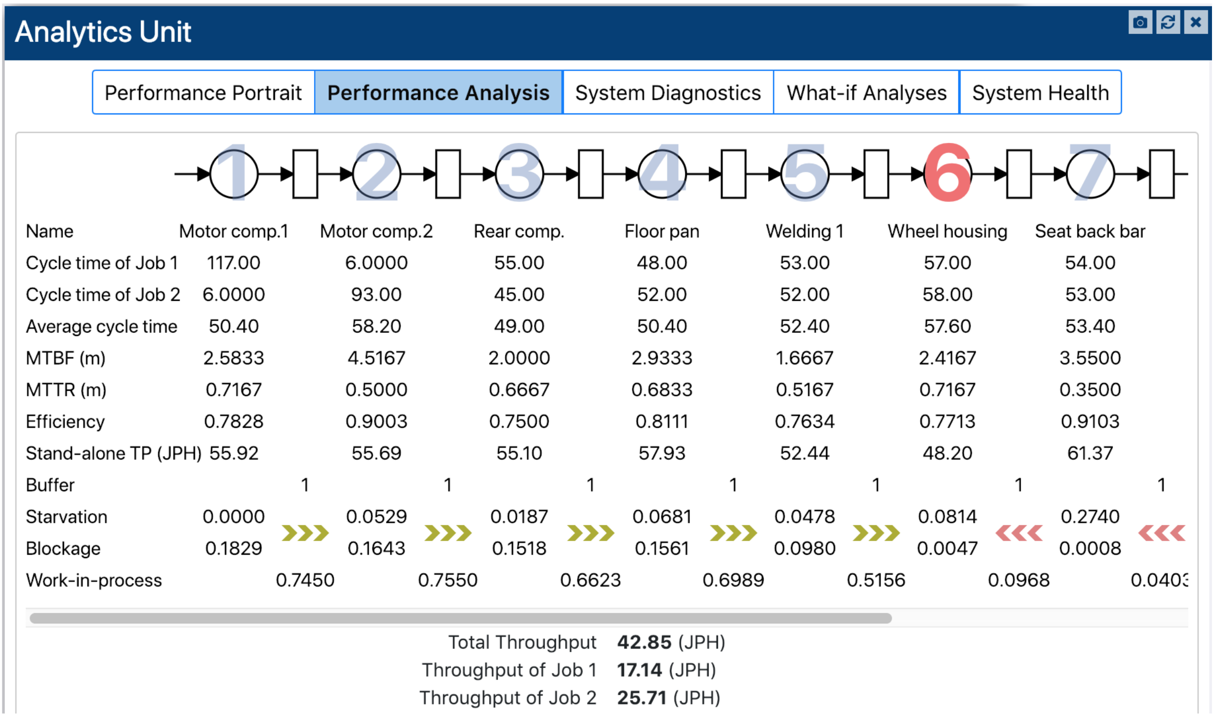



Рис. 9. AU output of PMA programmed for smart automotive underbody assembly system: Performance Portrait




| Name | Motor comp.1 | | Motor comp.2 | | Rear comp. | | Floor pan | | Welding 1 | | Wheel housing | | Seat back bar | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cycle time of Job 1 | 117.00 | | 6.0000 | | 55.00 | | 48.00 | | 53.00 | | 57.00 | | 54.00 | |
| Cycle time of Job 2 | 6.0000 | | 93.00 | | 45.00 | | 52.00 | | 52.00 | | 58.00 | | 53.00 | |
| Average cycle time | 50.40 | | 58.20 | | 49.00 | | 50.40 | | 52.40 | | 57.60 | | 53.40 | |
| MTBF (m) | 2.5833 | | 4.5167 | | 2.0000 | | 2.9333 | | 1.6667 | | 2.4167 | | 3.5500 | |
| MTTR (m) | 0.7167 | | 0.5000 | | 0.6667 | | 0.6833 | | 0.5167 | | 0.7167 | | 0.3500 | |
| Efficiency | 0.7828 | | 0.9003 | | 0.7500 | | 0.8111 | | 0.7634 | | 0.7713 | | 0.9103 | |
| Stand-alone TP (JPH) | 55.92 | | 55.69 | | 55.10 | | 57.93 | | 52.44 | | 48.20 | | 61.37 | |
| Buffer | | 1 | | 1 | | 1 | | 1 | | 1 | | 1 | | 1 |
| Starvation | 0.0000 | | 0.0529 | | 0.0187 | | 0.0681 | | 0.0478 | | 0.0814 | | 0.2740 | |
| Blockage | 0.1829 | | 0.1643 | | 0.1518 | | 0.1561 | | 0.0980 | | 0.0047 | | 0.0008 | |
| Work-in-process | | 0.7450 | | 0.7550 | | 0.6623 | | 0.6989 | | 0.5156 | | 0.0968 | | 0.0403 |

| | |
|---|---|
| Total Throughput | **42.85** (JPH) |
| Throughput of Job 1 | **17.14** (JPH) |
| Throughput of Job 2 | **25.71** (JPH) |

Рис. 10. AU output of PMA programmed for smart automotive underbody assembly system: Performance Analysis

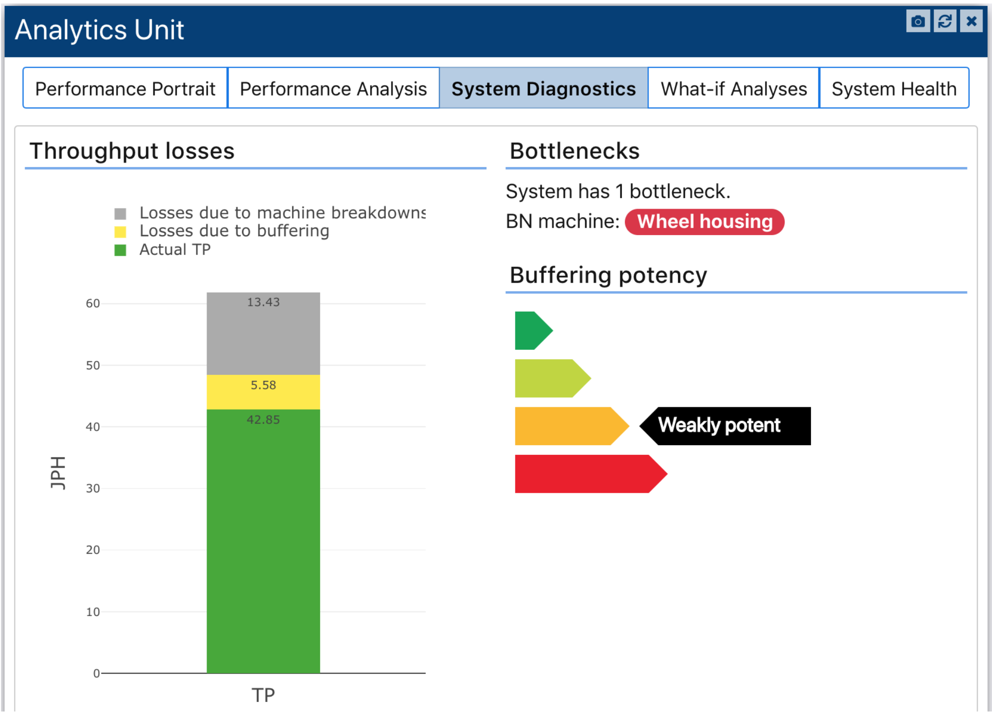



Рис. 11. AU output of PMA programmed for smart automotive underbody assembly system: System Diagnostics

of Fig. 12: when MTTR of Wheel Housing is reduced by 25%, the BN switches to Rear Compartment, and no further improvement of Wheel Housing could increase $TP$.

Finally, the last AU output, System Health (Fig. 13), summarizes the previous findings along with evaluating effectiveness of machines and buffers calculated according to

$$\text{Effectiveness of machines} = \frac{TP}{TP \text{ if machines had no breakdowns}}, \tag{28}$$

$$\text{Effectiveness of buffers} = \frac{TP}{TP \text{ if buffers were infinite}}. \tag{29}$$

This offers OM a possibility to quickly assess the problems with the system in order to formulate potential scenarios of continuous improvement.

These scenarios, entered in the Managerial Input block, are shown in Fig. 14. As one can see, two of them require increasing $TP$ to 47 JPH (about 10% improvement) and two others maximizing $TP$, all under different AAS constraints.

The outputs of OU for each of these scenarios are shown in Fig. 15. It indicates that Scenario 1 would not lead to the desired improvement, while Scenario 2 would (under the equipment changes indicated). Scenarios 3 and 4 would result in a substantial $TP$ improvement. Selecting a scenario (see Fig. 16) and implementing it in the plant leads to the $TP$ improvement close to that predicted by OU (Fig. 17).

This demonstration illustrates efficacy of PMA for automating decision-making in large-volume multi-job production systems.

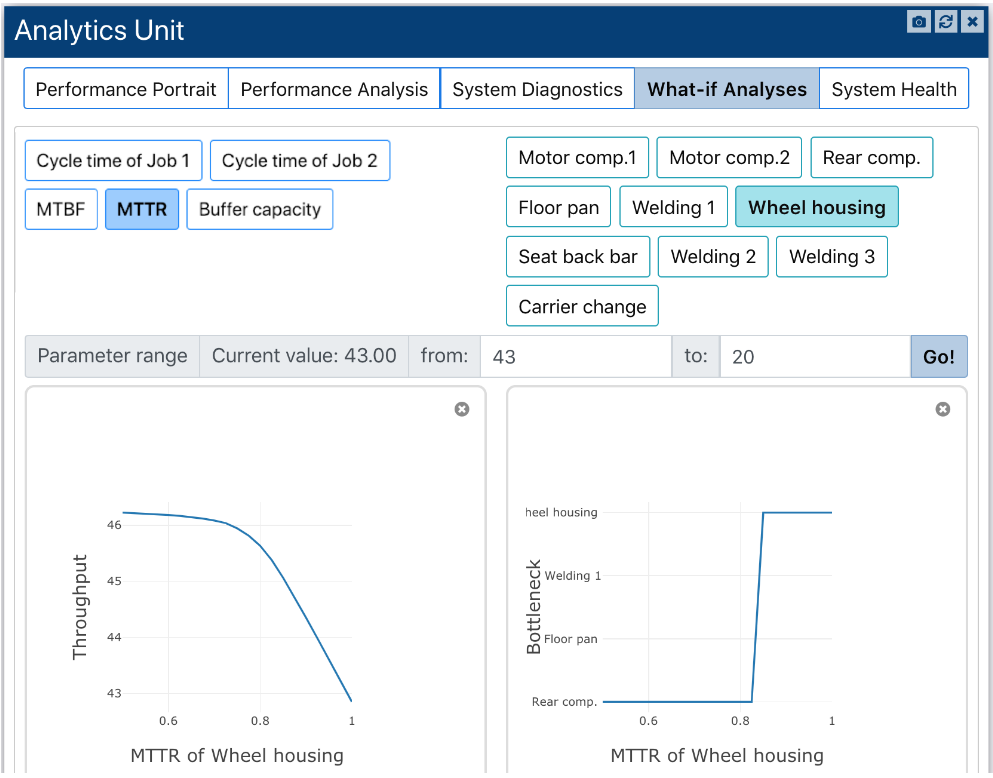



Рис. 12. AU output of PMA programmed for smart automotive underbody assembly system: What-if Analyses

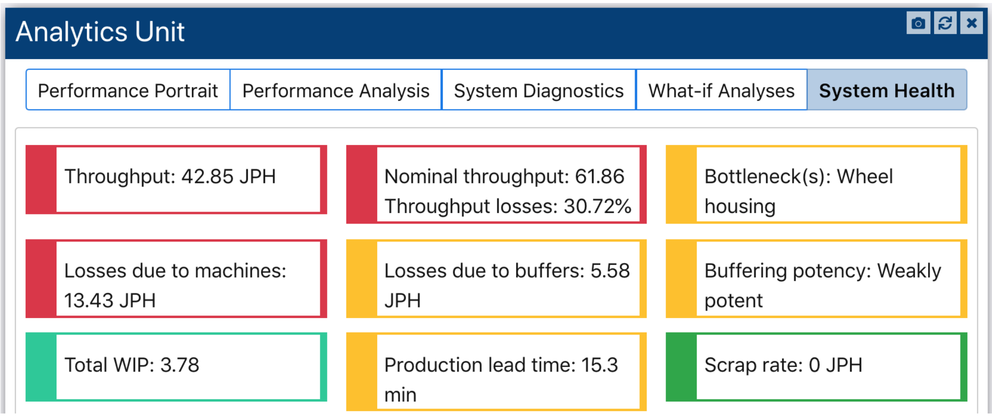



Рис. 13. AU output of PMA programmed for smart automotive underbody assembly system: System Health

## 8. Demonstration: Smart Hot-dip Galvanization Plant[4]

The hot-dip galvanization plant is a small-size manufacturing operation intended to coat iron sheets with a layer of Zinc to avoid oxidation and rusting. Based on the process steps, this production system is modelled as a serial line (see the output of IU in Fig. 18).

[4]The authors are grateful to Azarakhsh hot-dip galvanization factory for their help in this case-study.

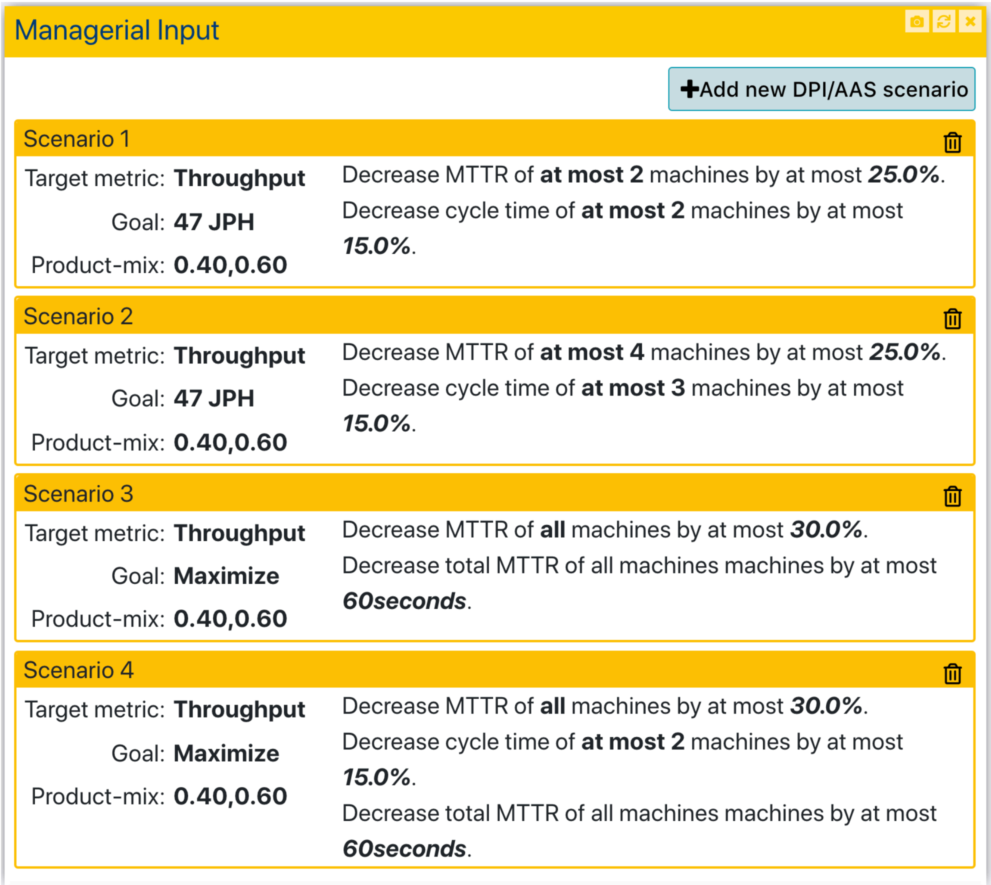
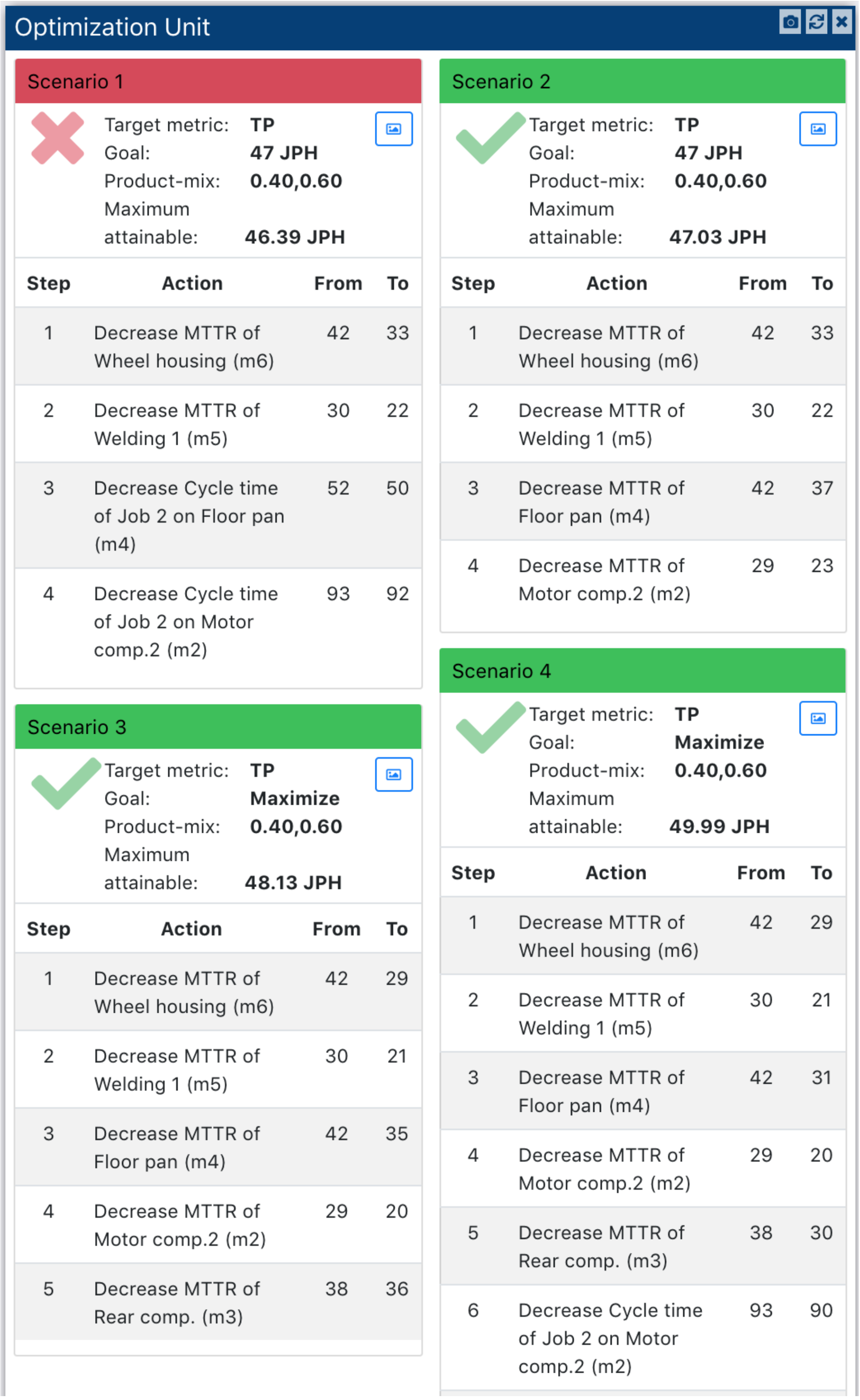



Рис. 14. Managerial Input of PMA programmed for smart automotive underbody assembly system

Its main feature is that there is no hardware-constrained buffering (as indicated by the open rectangles in Fig. 18). This leads to excessively large $WIP$ and unacceptable $LT$. Therefore, the goal of PMA-based SPS in this case is to both improve $TP$ and achieve a desired $LT$, using feedback release control of raw material based on the method described in Subsection 4.4. Below is an illustration of this SPS operation.

The AU outputs of the PMA programmed for this system are listed at the top of Fig. 19. Along with the usual outputs, a system-specific output of AU is the Characteristic Curve (CC), which represents the behaviour of $LT$ as a function of the raw material release rate; this information provides OM with a general understanding of the system's capability. The System Health output of AU is in Fig. 19(b); it shows, in particular, that $TP$ losses are over 20% and the production lead time is over 18 hours; it indicates that improvement of this system requires both increasing $TP$ and substantially decreasing $LT$. This is carried out using a two-stage procedure: First, OM enters the scenarios for $TP$ improvement (see Fig. 20, where a 20% $TP$ increase is requested). Based on this input, OU produces the output shown in Fig. 21, along with the managerial decision to submit Scenario 1 for implementation. The CC for the new (improved) system is shown in Fig. 22. In addition, this figure offers OM the possibility to enter the desired lead time ($LT_d = 20$ min, see the dot on the CC in Fig. 22) and the release interval ($RI = 60$ min). Clicking



Optimization Unit

**Scenario 1**

Target metric: **TP**
Goal: **47 JPH**
Product-mix: **0.40,0.60**
Maximum attainable: **46.39 JPH**

| Step | Action | From | To |
|---|---|---|---|
| 1 | Decrease MTTR of Wheel housing (m6) | 42 | 33 |
| 2 | Decrease MTTR of Welding 1 (m5) | 30 | 22 |
| 3 | Decrease Cycle time of Job 2 on Floor pan (m4) | 52 | 50 |
| 4 | Decrease Cycle time of Job 2 on Motor comp.2 (m2) | 93 | 92 |

**Scenario 2**

Target metric: **TP**
Goal: **47 JPH**
Product-mix: **0.40,0.60**
Maximum attainable: **47.03 JPH**

| Step | Action | From | To |
|---|---|---|---|
| 1 | Decrease MTTR of Wheel housing (m6) | 42 | 33 |
| 2 | Decrease MTTR of Welding 1 (m5) | 30 | 22 |
| 3 | Decrease MTTR of Floor pan (m4) | 42 | 37 |
| 4 | Decrease MTTR of Motor comp.2 (m2) | 29 | 23 |

**Scenario 3**

Target metric: **TP**
Goal: **Maximize**
Product-mix: **0.40,0.60**
Maximum attainable: **48.13 JPH**

| Step | Action | From | To |
|---|---|---|---|
| 1 | Decrease MTTR of Wheel housing (m6) | 42 | 29 |
| 2 | Decrease MTTR of Welding 1 (m5) | 30 | 21 |
| 3 | Decrease MTTR of Floor pan (m4) | 42 | 35 |
| 4 | Decrease MTTR of Motor comp.2 (m2) | 29 | 20 |
| 5 | Decrease MTTR of Rear comp. (m3) | 38 | 36 |

**Scenario 4**

Target metric: **TP**
Goal: **Maximize**
Product-mix: **0.40,0.60**
Maximum attainable: **49.99 JPH**

| Step | Action | From | To |
|---|---|---|---|
| 1 | Decrease MTTR of Wheel housing (m6) | 42 | 29 |
| 2 | Decrease MTTR of Welding 1 (m5) | 30 | 21 |
| 3 | Decrease MTTR of Floor pan (m4) | 42 | 31 |
| 4 | Decrease MTTR of Motor comp.2 (m2) | 29 | 20 |
| 5 | Decrease MTTR of Rear comp. (m3) | 38 | 30 |
| 6 | Decrease Cycle time of Job 2 on Motor comp.2 (m2) | 93 | 90 |

Рис. 15. OU outputs of PMA programmed for smart automotive underbody assembly system

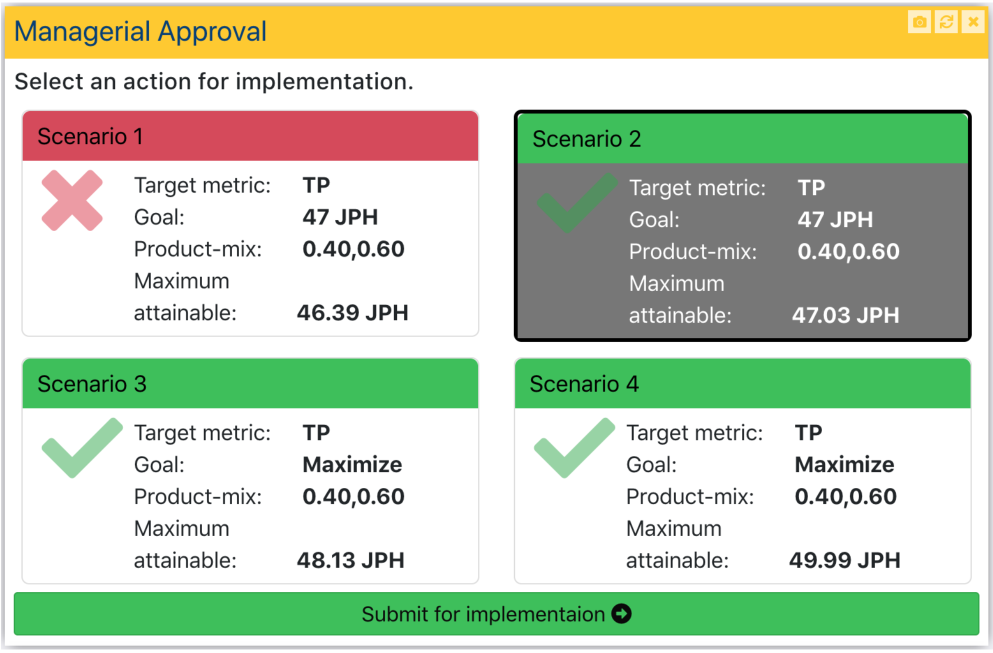
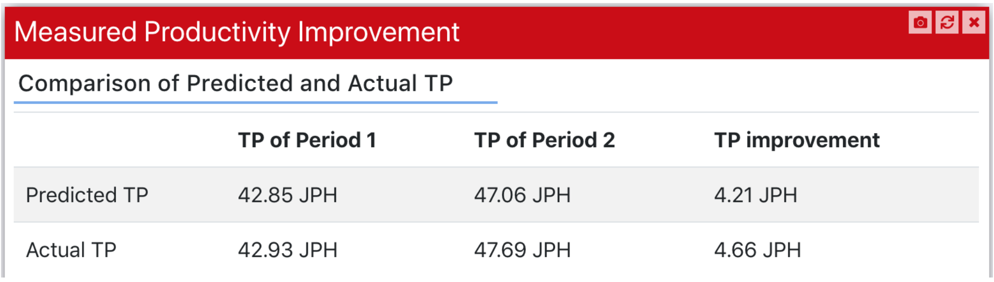



Рис. 16. Managerial Approval of PMA programmed for smart automotive underbody assembly system


Measured Productivity Improvement

Comparison of Predicted and Actual TP

| | TP of Period 1 | TP of Period 2 | TP improvement |
|---|---|---|---|
| Predicted TP | 42.85 JPH | 47.06 JPH | 4.21 JPH |
| Actual TP | 42.93 JPH | 47.69 JPH | 4.66 JPH |

Рис. 17. Measured Productivity Improvement output of PMA programmed for smart automotive underbody assembly system

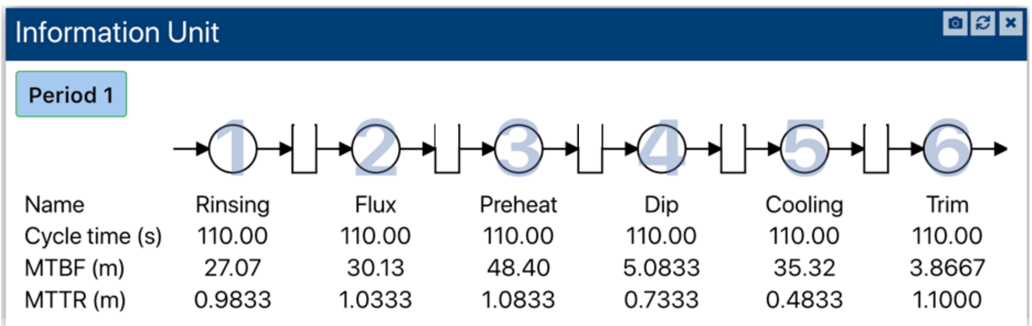
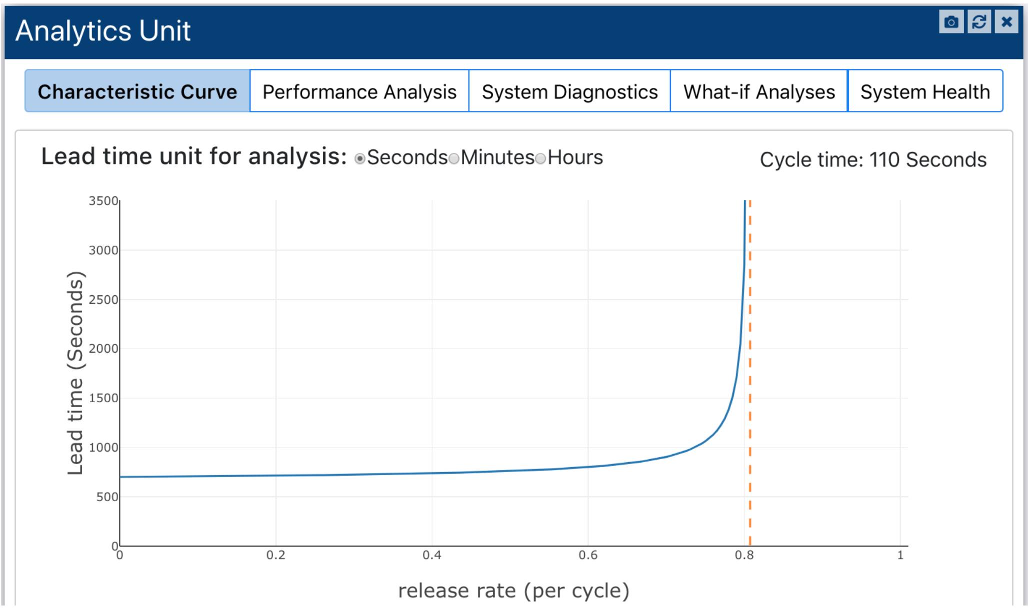
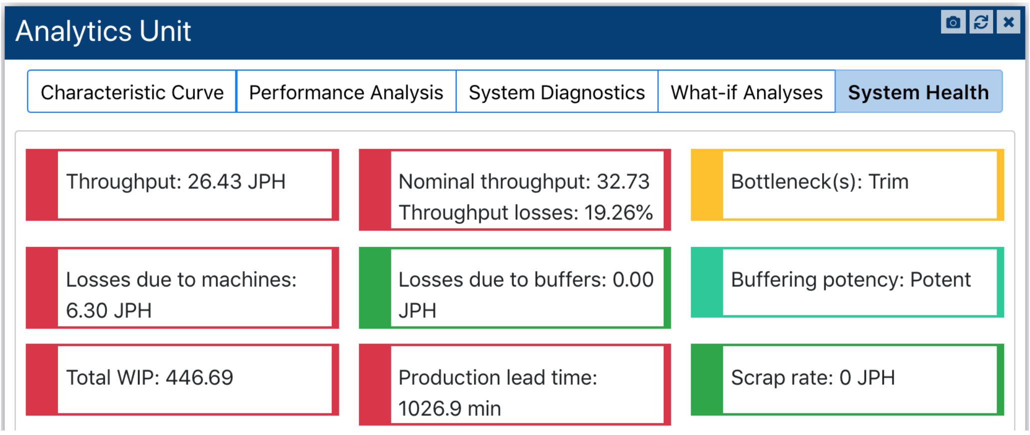



Рис. 18. IU output of PMA programmed for smart hot-dip galvanization plant




(a) Characteristic Curve




(b) System Health

Рис. 19. AU outputs of PMA programmed for smart hot-dip galvanization plant

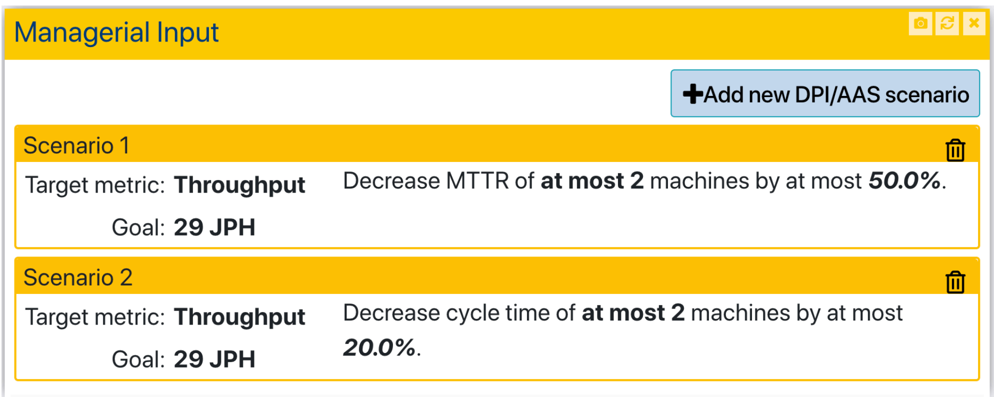



Рис. 20. Managerial Input of PMA programmed for smart hot-dip galvanization plant

on "Calculate" produces the parameters of the closed-loop raw material release policy, indicating that:

- The release rate per hour to ensure $LT_d$ is 28 parts.
- The release should take place only if the total $WIP$ at the beginning of an hour is less than 5 parts.
- This closed-loop release policy maintains this $LT_d$, while the resulting $TP$ is close to 28 JPH and the average $WIP$ in the system is 12.84 jobs.

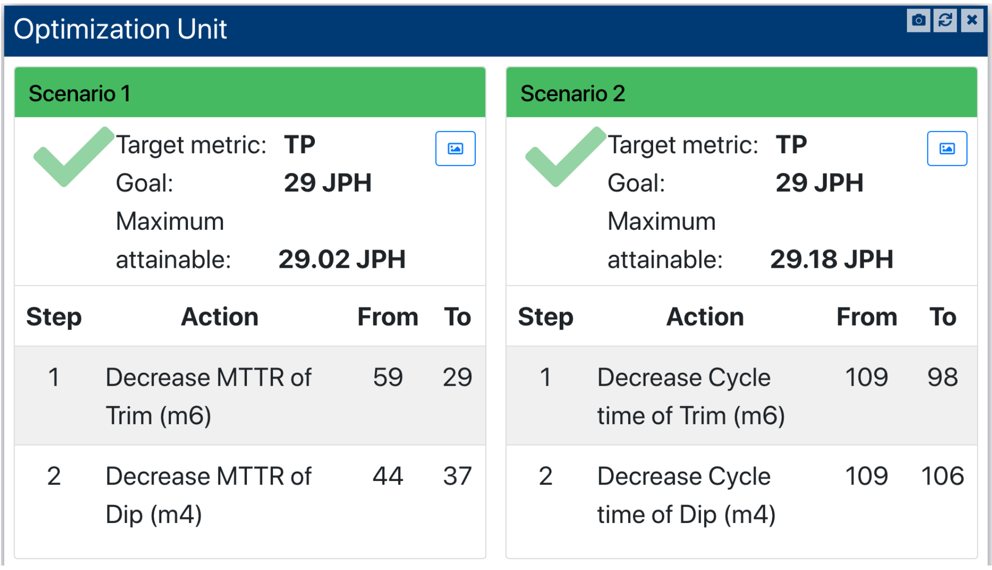



Рис. 21. OU output of PMA programmed for smart hot-dip galvanization plant

This demonstration illustrates efficacy of PMA for automating decision-making in production systems with hardware-unlimited buffers.

## 9. Conclusions and Future Work

This paper presents the structure and analytics of PMA as a central element of SPS. The main requirement for successful operation of PMA-based SPS is the availability of reliable data concerning the manufacturing equipment status in real time. In most cases, these data can be gleaned from PLCs used on the factory floor for equipment automation.

While the results reported are encouraging, a number of PMA-related problems remain open. They include:

- Develop a method for automated data cleaning and verification. It is quite common that the data collected on the factory floor contain errors. Failure to detect and correct these errors reduces the fidelity of the parametric model employed in IU.

- Develop analytical methods for PMA-based SPS operation in transient regimes, i.e., having AU operating not on the steady states of performance metrics, but on their real-time characteristics.

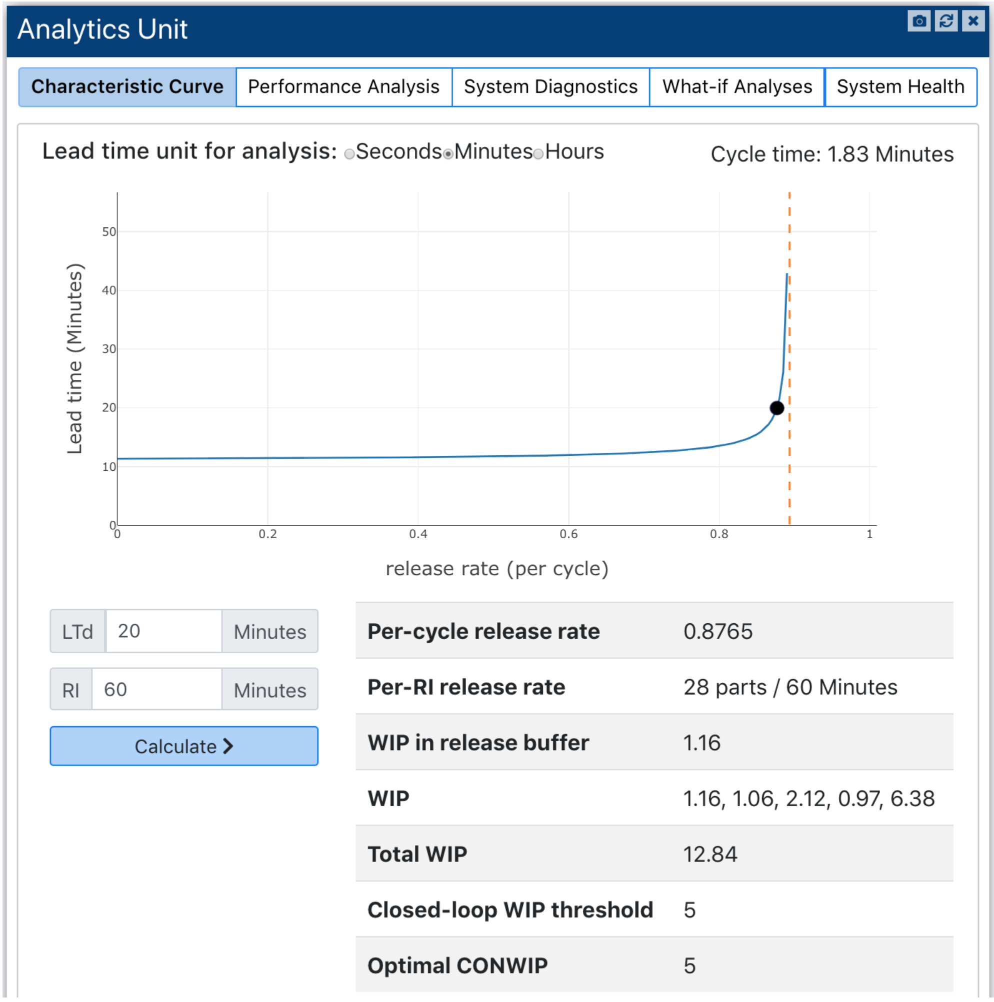



Рис. 22. AU output and lead time control of the improved system for smart hot-dip galvanization plant

- Analyze the sensitivity of $TP$ to various machine parameters (e.g., cycle time, MTBF, MTTR, etc.). This would increase the effciency of search algorithms employed in OU.
- Develop search procedures for simultaneous optimization of several performance metrics, e.g., $TP$ and $LT$ (in the framework of multi-criteria optimization).

Although solving each of these open problems is of importance, the overarching items of the future work are applications of PMA-based SPS in large, mid-size, and small manufacturing organizations.

## Abbreviations and Notations

Abbreviations: AAS – admissible action space; AU – analytics unit; BN – bottleneck machine; CC – characteristic curve; DPI – desired productivity improvement; IU – information unit; JPH – jobs-per-hour; MM – mathematical model; MTBF – mean time between failures; MTTR – mean time to repair; OM – operations manager;

OU – optimization unit; PLC – programmable logic controller; PMA – programmable manufacturing advisor; PMPP – product-mix performance portrait; PSE – production systems engineering; SPS – smart production system.

Notations: $b$ – buffer; $BL$ – probability of blockage; $c$ – machine capacity; $e$ – machine efficiency; $E$ – raw material release; $g$ – machine quality parameter; $H$ – number of cycles per unit of time; $\lambda$ – exponential distribution parameter of machine uptime; $LT$ – lead time; $m$ – machine; $\mu$ – exponential distribution parameter of machine downtime; $n$ – number of downtime occurrences; $N$ – buffer capacity; $PR$ – production rate; $Q_g$ – number of good parts produced; $Q_{total}$ – total number of parts produced; $SAT$ – standalone throughput; $ST$ – probability of starvation; $\tau$ – machine cycle time; $t_{down,i}$ – duration of $i$-th downtime; $t_{up,i}$ – duration of $i$-th uptime; $T_{down}$ – average downtime; $T_{up}$ – average uptime; $TP$ – throughput; $WIP$ – work-in-process.

СПИСОК ЛИТЕРАТУРЫ

P. Alavian, P. Denno, and S. M. Meerkov. Multi-job production systems: Definition, problems, and product-mix performance portrait of serial lines. *International Journal of Production Research*, 55(24):7276–7301, 2017.

P. Alavian, Y. Eun, K. Liu, S. M. Meerkov, and L. Zhang. The $(\alpha, \beta)$-precise estimates of MTBF and MTTR: Definitions, calculations, and induced effect on machine efficiency evaluation. Submitted to *the 9th IFAC Conference on Manufacturing Modelling, Management and Control (MIM'2019)*, 2019. Also, uploaded at `http://web.eecs.umich.edu/~smm/publications/mtbf_mttr_estimates.pdf`.

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

T. Altiok and B. Melamed. *Simulation Modeling and Analysis with Arena*. Elsevier, 2010.

A. A. Andronov, A. A. Vitt, and S. E. Khaikin. *Theory of Oscillators*. Fizmatgiz, Moscow, 1959.

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

W. Bolton. *Programmable Logic Controllers*. Newnes, 2015.

J. A. Buzacott. Automatic transfer lines with buffer stocks. *International Journal of Production Research*, 5(3):183–200, 1967.

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

G. L. Curry and R. M. Feldman. *Manufacturing Systems Modeling and Analysis*. Springer Berlin Heidelberg, 2009.

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

B. Jerry. *Discrete Event System Simulation*. Pearson Education, 2005.

H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry*. 2013. Final report of the Industrie 4.0 Working Group. Forschungsunion.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall Upper Saddle River, NJ, 2002.

A. M. Law, W. D. Kelton, and W. D. Kelton. *Simulation Modeling and Analysis*, volume 2. McGraw-Hill New York, 1991.

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

Y. Liao, F. Deschamps, E. F. R. Loures, and L. F. P. Ramos. Past, present and future of Industry 4.0 – A systematic literature review and research agenda proposal. *International Journal of Production Research*, 55(12):3609–3629, 2017.

S. M. Meerkov and C. B. Yan. Production lead time in serial lines: Evaluation, analysis, and control. *IEEE Transactions on Automation Science and Engineering*, 13(2):663–675, 2016.

H. T. Papadopoulos, C. Heavey, and J. Browne. *Queueing Theory in Manufacturing Systems Analysis and Design.* Chapman and Hill, London, UK, 1993.

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

J. Schlechtendahl, M. Keinert, F. Kretschmer, A. Lechler, and A. Verl. Making existing production systems Industry 4.0-ready. *Production Engineering*, 9(1):143–148, 2015.

B. A. Sevast'yanov. Influence of storage bin capacity on the average standstill time of a production line. *Theory of Probability & Its Applications*, 7(4):429–438, 1962.

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