Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 1, 51–63
DOI: 10.34768/amcs-2022-0005

A FEASIBLE SCHEDULE FOR PARALLEL ASSEMBLY TASKS IN FLEXIBLE MANUFACTURING SYSTEMS

PAWEŁ MAJDZIK [a]

[a]Institute of Control and Computation Engineering
University of Zielona Góra
ul. Szafrana 2, 65-516 Zielona Góra, Poland
e-mail: p.majdzik@issi.uz.zgora.pl

The paper concerns the design of a framework for implementing fault-tolerant control of hybrid assembly systems that connect human operators and fully automated technical systems. The main difficulty in such systems is related to delays that result from objective factors influencing human operators' work, e.g., fatigue, experience, etc. As the battery assembly system can be considered a firm real-time one, these delays are treated as faults. The presented approach guarantees real-time compensation of delays, and the fully automated part of the system is responsible for this compensation. The paper begins with a detailed description of a battery assembly system in which two cooperating parts can be distinguished: fully automatic and semi-automatic. The latter, nonderministic in nature, is the main focus of this paper. To describe and analyze the states of the battery assembly system, instead of the most commonly used simulation, the classic max-plus algebra with an extension allowing one to express non-deterministic human operators' work is used. In order to synchronize tasks and schedule (according to the reference schedule) automated and human operators' tasks, it is proposed to use a wireless IoT platform called KIS.ME. As a result, it allows a reference model of human performance to be defined using fuzzy logic. Having such a model, predictive delays tolerant planning is proposed. The final part of the paper presents the achieved results, which clearly indicate the potential benefits that can be obtained by combining the wireless KIS.ME architecture (allocated in the semi-automatic part of the system) with wired standard production networks.

Keywords: parallel system, synchronization, scheduling, discrete event system, wireless equipment.

1. Introduction

Nowadays, concurrent and distributed systems constitute one of the most important trends of scientific research, mainly but not only, in the field of computer science. This research focuses mainly on the design and implementation of concurrent and distributed applications where the main challenge is to ensure proper synchronization of concurrent/distributed processes while maintaining a high level of flexibility. In the real world, you can find many examples of concurrent/distributed systems such as: transport systems (railway, underground), logistics systems (conveyance and storage of goods, organization and delivery of services), air control systems, traffic control systems and, what the subject of this paper is, automated manufacturing systems (Seybold *et al*., 2015; Groover, 2014).

In such systems, at least two basic layers can be distinguished: a layer of devices that are connected by dedicated wired or wireless networks and a control layer, which includes, among others, an implementation of synchronization and scheduling methods. In many cases, the nature of a given system imposes time requirements on the system designers (Baruwa *et al*., 2015). These time requirements transfer our considerations to the sphere of real-time systems, in particular the firm real time ones. In these, apart from the synchronization, the tasks' schedule methods should be designed (Kopetz, 2011). These methods have to guarantee the execution of all real tasks according to the requirements for timeliness.

The subject of the paper is a battery assembly system (Fig. 1), in which two parts can be distinguished: the first one contains fully automated assembly and transportation robots, while in the second, semi-automatic part, some assembly and transportation tasks are performed by human operators. One of the core challenges in this type of systems is the design of fault-tolerant control (FTC).

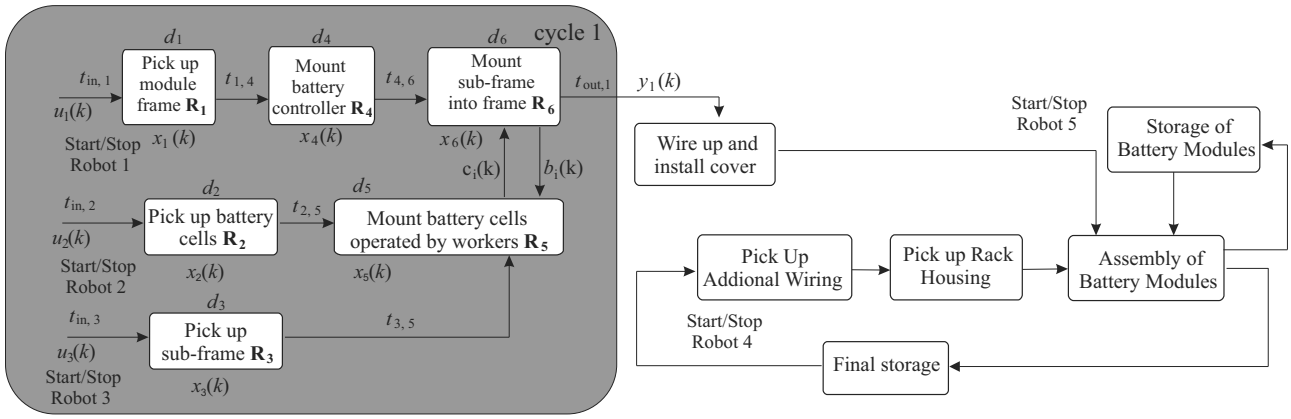

Fig. 1. Structure of the battery assembly system.

Another core challenge is the design and implementation of such scheduling methods that will guarantee the task's flow consistent with the assumed schedule, and will additionally compensate for the effects of probable violations of the schedule. Majdzik *et al*. (2021) and Witczak *et al*. (2020) present a FTC scheme for fully automated systems, guaranteeing the flow of parallel tasks according to a reference schedule. It is obvious that the proposed solutions concern a specific set of faults whose effects can be compensated for in a fully automated way by accelerating the execution of other tasks. However, in a system where tasks are performed by human operators, the solutions proposed by Majdzik *et al*. (2021) and Witczak *et al*. (2020) must be extended to take into account the impact of objective, non-deterministic factors, such as work shift length, tiredness, experience, skills (Nivolianitou and Konstantinidou, 2018) on the human operator's efficiency. These factors pose an inevitable uncertainty during the forming of a model of human operators. Thus, fuzzy logic (Segura *et al*., 2016; Mircetic *et al*., 2016; Salazar *et al*., 2020; Rutkowski *et al*., 2019) is a natural formalism for modelling such an uncertain behaviour. Therefore, a very current research issue is to elaborate a scheduling method that takes into account the factors described above and guarantees the flow of tasks in accordance with the desired schedule. One of the most effective analytical frameworks is max-plus algebra which can be combined with the predictive methods, as well as FTC. This algebra naturally allows for the modelling of concurrent tasks and synchronization of them (Baccelli *et al*., 1992; Majdzik *et al*., 2016). But the presented solutions concern deterministic systems with fixed synchronization rules.

One of the frequently applied approaches to deal with such a challenge is the simulation one. However, this approach demands significant costs related to the preparation of the simulation environment. In addition, any changes in the input parameters require time-consuming re-simulation (Ebrahimi *et al*., 2015; Rousset *et al*., 2016). A much more formidable challenge than the application of the simulation tool is to extend one of the existing mathematical formalisms with the necessary functionalities and incorporate it into the considered system. Therefore, the main contribution of this paper is to design and incorporate an analytical framework into the system, which is the composition of all the foregoing formalisms, and is able to maintain a schedule consistent with the desired one.

To enable communication between fully automated devices and human operators, the appropriate KIS.ME platform (Keep It Simple.Manage Everything) has been allocated in the battery assembly system (Fig. 1). It should be noted that the current progress in IoT development (Madakam *et al*., 2015) allows us to design and implement more efficient systems equipped with a necessary number of sensors and actuators connected by various networks. Such wireless devices enable efficient digitization through dedicated SCADA systems and thus communication between the monitoring layer and the device layer, including the operators (RAFI, 2021). Having the above infrastructure it is possible to incorporate the proposed strategy of task scheduling into the battery assembly system. The proposed approach extends the one of Dizdar and Koçar (2020) by providing an interpretable structure of model parameters. Contrarily to Dizdar and Koçar (2020), the proposed operator model operates in a time domain instead of a velocity one.

The paper is organized as follows. Section 2 describes the battery assembly system. In Section 3 mathematical modelling tools for the automated and semi-automatic parts of the system are presented. As a result, Section 4 contains the matrix-based representation of the system. Section 5 proposes the Takagi–Sugeno operator model. In Section 6 the comprehensive framework allowing to control the task flow is described. Finally, in Section 7, the effectiveness of the proposed

solution, using selected faults scenarios within the range of the tasks performed by human operators, is verified.

2. Overview of the battery assembly system

This section aims at describing a system that is utilized to assess the proposed scheduling strategy and the task synchronization strategy—a system for assembling battery cells used, for instance, to store electrical energy in houses equipped with solar systems. Additionally, the assembly system is equipped with KIS.ME. The main goal is to set up a battery assembly system providing a maximal flexibility for further variants of battery system as well as the efficiency for various battery variants.

These variants require different formats of frames, different numbers of Lithium-ion cells, different cables and, finally, different amounts of insulation material.

The considered system belongs to the class of flexible assembly systems (FASs), and therefore requires the application of the following layers: visualization and acquisition layers, a control layer and a device layer (Fig. 1). In the first, device layer, two kinds of workstations can be distinguished. The first kind covers fully automated devices such as a pick&place robot unloading Lithium-ion cell packages, industrial manipulators mounting controllers, wiring, covers and mobile transportation robots, while the second one consists of a semi-automatic assembly station operated by a set of human operators. It is obvious that the proper skills and experience of the operators are critical to achieving the assumed task schedule in the system. Therefore, monitoring of human tasks and their effective synchronization with fully automated devices requires installing additional equipment. The battery assembly system consists of two cycles integrated with each other. The first one is handled by three types of transportation robots (Types 1–3). Transportation tasks are performed by autonomous mobile robots that are able to provide different frames to manufacture different batteries. The second cycle is handled by two types of transportation robots and covers the final assembly of the products.

The initial operation performed in the first production cycle is to move the robots from the starting setting to the frame storage where the robots pick up an empty battery main-frame. Subsequently, the battery module controller is mounted into the main-frame. In the next step, three parallel transportation tasks are carried out. All of them deliver the needed components to the cell mounting station. The robots of Type 1 transport the main-frame, the robots of Type 2 deliver cells from the cell storage, and the last one delivers the sub-frame to various workplaces. These workplaces are allocated in the workspace of the cell mounting station that is handled by three human operators. Therefore, each operator has an own, individual workplace to which the cells are transported by the robots and from which they are picked up by the operators. In each cycle, the robots of Type 2 must provide the appropriate number of Lithium-ion cell packages that are assembled by human operators into the module main-frame. This number depends on the type of battery. The human operators assemble cell packages into the sub-frames, then assemble sub-frames into the main-frame and finally connect the sub-frames with the controller. Since the above tasks have to be performed at different stations, operators have to additionally transport components between them. The last step of the first cycle is the wiring and cover assembly.

The second cycle begins also with the robots in the starting position. The robot of Type 4 moves to the storage and picks up additional wiring, then moves to the rack housing storage. Subsequently, the robot delivers all the above-mentioned components to the last assembly station. At the same time, the robot of Type 5 brings battery modules to the assembly of battery modules station. At this station, the battery modules and additional wiring are assembled into the rack housings. In the last step, a Type 4 robot transports the finished product to the final storage.

However, due to the space limitations, only the first cycle is taken into account, while the description of the second one is omitted in this representation.

3. Preliminaries

As stated in the previous section, two layers can be distinguished in the system under consideration. The first layer consist of fully automated devices, while the other consists of two semi-automatic assembly stations operated by a set of human-operators.

In the fully automated part of the system, parallel tasks are synchronized by means of two basic modes, i.e., the mutual exclusion and the rendez-vous ones. Additionally, in the redundant human-operators layer, a concurrency has to be considered.[1] This section aims at introducing a mathematical framework underlying the classical max-plus algebra formalism, allowing to express both of the above synchronization modes as well as the concurrency.

The battery assembly system belongs to the class of discrete event dynamic system (DEDS) and can be described by the following parameters (Fig. 1):

- the set of work stations: $\mathbb{R} = \{R_1, R_2, \ldots, R_s\}$, where s is the number of stations;
- the set of processing times: $\mathbb{D} = \{d_1, d_2, \ldots, d_s\}$, where d_i is the processing time on the i-th station, d_i is the operation time on the i-th station,

[1] Concurrency understood as the choice of one of the competing human operators for access to the main-frame.

- $d_{i,j}$ stands for the operation time on the i-th station carried out by the j-th human operator, [2]
- the set of transportation input times: $\mathbb{T}_{in} = \{t_{in,1}, t_{in,2}, \ldots, t_{in,m}\}$, where $t_{in,i}$ is the transportation time from the i-th input towards the assembly unit,
- the set of transportation times in the automatic assembly layer: $\mathbb{T} = \{t_{j,l}\},\ \forall j, l \in \{1, 2, \ldots, r\} : j \neq l$ with a connection established between the j-th and the l-th assembly stations, respectively.

Additionally, to describe the assembly and transportation task flow, the following parameters are defined:

- k is the event counter,
- $u_i(k)$ is the time instant at which the robot transfers items towards the i-th input at the k-th counter,
- $x_i(k)$ denotes the time instant at which the i-th assembly station starts carrying out the task at the k-th counter,
- $y_1(k)$ is the time of delivering the i-th main-frame into Cycle 2.

3.1. Synchronization. As can be seen in Fig. 1, parallel assembly and transportation tasks need to be coordinated at selected points, denoted by the individual assembly station. At these points, the next operation can be executed only when two or more operations complete their execution. To explain this sort of synchronization, let us consider the assembly operation carried out by a human operator by using the station R_5 (Fig. 1). The operator can start his/her operation at the station R_5, at the k-th counter when the following conditions are met:

- cells have been transported from the cell storage (the station R_2) to the station R_5 by a robot of Type 2;
- a sub-frame has been delivered from the storage station R_3 to the assembly station R_5 by a robot of Type 3;
- the human operator has to complete the previous assembly task at the $(k-1)$-th counter being the sequence of the following operations: mounting the cells into the sub-frame on the assembly station R_5, transporting the sub-frame to the station R_6, connecting the sub-frame with the main-frame and, finally, the return of the human operator to the station R_5.

[2]The assembly of the cells in a sub-frame, the transport of the sub-frame and its assembly to the main-frame can be performed by one of the redundant operators, which forces the above time indexing.

Taking into account the above conditions, the time-evolution of $x_5(k)$ can be described as follows:

$$\begin{aligned} x_5(k) = \max(&x_2(k) + d_2 + t_{2,5},\ x_3(k) + d_3 + t_{3,5},\\ &x_4(k) + d_4 + t_{4,5},\ x_5(k-1) + d_5). \end{aligned} \tag{1}$$

To simplify this notation, it is assumed that this equation does not distinguish between single operators and that all actions of human operators are represented in (2) by one operation (duration d_5). This simplification will be omitted in the subsequent sections. Extending (1) for the other assembly station, the entire system can be described by the following form:

$$\begin{aligned} x_1(k) &= \max(x_1(k-1) + d_1,\ u_1(k) + t_{in,1}),\\ x_2(k) &= \max(x_2(k-1) + d_2,\ u_2(k) + t_{in,2}),\\ x_3(k) &= \max(x_3(k-1) + d_3,\ u_3(k) + t_{in,3}),\\ x_4(k) &= \max(x_1(k) + d_1 + t_{1,4},\ x_4(k-1) + d_4)\\ &= \max(x_1(k-1) + 2d_1 + t_{1,4},\\ &\quad x_4(k-1) + d_4, u_1(k) + t_{in,1} + d_1 + t_{1,4}),\\ x_5(k) &= \max(x_2(k) + d_2 + t_{2,5},\ x_3(k) + d_3 + t_{3,5},\\ &\quad x_5(k-1) + d_5)\\ &= \max(x_2(k-1) + 2d_2 + t_{2,5},\\ &\quad x_3(k-1) + 2d_3 + t_{3,5},\ x_5(k-1) + d_5,\\ &\quad u_2(k) + t_{in,2} + d_2 + t_{2,5},\\ &\quad u_3(k) + t_{in,3} + d_3 + t_{3,5}). \end{aligned} \tag{2}$$

Since the time instant at which the human operator starts assembly operation at the assembly station R_6 is entirely dependent on non-deterministic behaviour of human operators, the part related to the station R_6 has been omitted in (2).

3.2. Concurrency. As one of the assembly stations is handled by a set of human operators, there is a need to express a concurrency that is understood here as the selection of one of the operators to assembly cells into the main-frame in a given iteration. As was already mentioned, in order to tackle this challenge in terms of hardware, it is proposed to utilize the KIS.ME IoT infrastructure[3] that will be portrayed in detail in Section 5.

The main problem boils down to the scheduling of the automated tasks and human operators' tasks within the entire battery assembly system. In our previous works (Majdzik, 2020; Witczak *et al.*, 2020; Majdzik *et al.*, 2016) each assembly and transportation task was considered with some level of conservativeness. Introducing human operators to the system requires taking into account the inevitable delays. The mathematical

[3]This infrastructure was designed and manufactured by RAFI Gmbh & Co. KG.

representation of the parallel execution of the tasks which compete to access the shared work station is provided by a few well-known formalisms, e.g., automates, max-plus algebra (Van Den Boom and De Schutter, 2006). However, it is necessary to define a new approach to solve the representation of the concurrency.

Witczak *et al*. (2020) present the analytical descriptions of automated guided vehicles (AGVs), which enable to operate a set of items. But in the works of Majdzik (2020) and Witczak *et al*. (2020) it is assumed that the operation times are constant, which is also very hard to ensure by human operators. Removing this unrealistic assumption generates a need for an appropriate modification of the entire analytical framework and, as a consequence, generates an additional problem: What are the expected times of mounting and transportation tasks carried out by human operators? To deal with this problem, the standard operating procedures (SOPs) for human operators and a fuzzy logic human operator performance model are designed in Section 5. Let us start by providing a formal representation of the set of human operators by defining the following variables:

- $v_i(k)$ is a two-valued decision variable (i.e., its values belong to the set $\{e = 0, \varepsilon = -\infty\}$) specifying which human operator carried out the assembly operation for the the k-th frame,[4] where $i = 1, 2, \ldots, n_w$ and n_w is the number of operators;
- $x_{r,i}(k)$ stands for the time instant at which the i-th human operator is ready to start assembly task on the r-th station at the k-th counter;
- $c_i(k)$ denotes the transportation time of the k-th sub-frame by the i-th operator from assembly station R_5 to R_6 at the k-th counter;
- $b_i(k)$ stands for the time needed for the i-th operator to return from R_6 to R_5 at the k-th counter (Fig. 1).

In the battery assembly system, the set of human operators competes for an access to the main frame which is delivered to the assembly station R_6 (Fig. 1). Note that $v_i(k) = e$ denotes that the i-th human operator performs assembly operation related to the k-th main-frame, while $v_i(k) = \varepsilon$ means an opposite situation. Since the tasks performed by human operators at the station R_5 (where cells are mounted to the sub-frame) have an impact on which of the competing operators gets access to the currently available main-frame, the time evolution of the i-th operator has the following form:

$$\begin{aligned} x_{5,i}(k) = \max(&x_{5,i}(k-1) + d_5 + c_i(k-1) + d_6 \\ &+ b_i(k-1),\ x_4(k) + d4 + t_{4,6}), \\ &i = 1, 2, \ldots, n_w, \end{aligned} \tag{3}$$

[4]The term 'k-th counter' is equivalent to the term 'k-th battery', i.e., the k-th battery is manufactured in the k-th counter.

with the associated constraints

$$\begin{aligned} &c_i(k) = c(k), \qquad b_i(k) = b(k), \\ &c_j(k) = 0, \qquad b_j(k) = 0, \\ &\qquad\qquad \forall j \neq i, \quad j = 1, \ldots, n_w, \end{aligned} \tag{4}$$

and the exclusion rule

$$v_i(k) = e \quad \Leftrightarrow \quad v_j(k) = \varepsilon, \quad \forall i \neq j, \tag{5}$$

where d_5 and d_6 are the assembly time of the cells to the sub-frame and the assembly time of the sub-frame into the main-frame, respectively.

From (4) and (5) it follows that

- if the i-th human operator does not carry out assembly task at the k-th counter then $b_i(k)$ and $c_i(k)$ are set to zero;
- the assembly times of the i-th human operator are equal to their nominal values.

Summarizing, one of the contributions of this paper is the adaptation of the mathematical approach presented in (Witczak *et al*., 2020) that allows us to represent the non-deterministic behaviour of the human operators.

4. Analytical description

This section aims at describing a suitable mathematical formalism. As can be observed, the intuitive models (2) and (3) use only $\max$ and $+$ operators, which implies applying the so-called max-plus algebra (Baccelli *et al*., 1992; Butkovic, 2010).

The max-plus algebraic structure $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ is defined with

$$a \oplus b = \max(a, b), \tag{6a}$$

$$a \otimes b = a + b, \quad \forall a, b \in \mathbb{R}_\varepsilon \tag{6b}$$

with $\mathbb{R}_\varepsilon \triangleq \mathbb{R} \cup \{-\infty\}$ while $\mathbb{R}$ is the field of real numbers. Thus, $\oplus$ and $\otimes$ signify the max-plus addition and multiplication operators, respectively. In addition, $a \oplus \varepsilon = a$, $a \otimes \varepsilon = \varepsilon$ and $a \otimes e = a$, $\forall a \in \mathbb{R}_\varepsilon$, where $\varepsilon = -\infty$ and $e = 0$ are the neutral elements for the max-plus addition and max-plus multiplication operators, respectively. For matrices $\boldsymbol{D}, \boldsymbol{E} \in \mathbb{R}_\varepsilon^{m \times n}$ and $\boldsymbol{F} \in \mathbb{R}_\varepsilon^{n \times p}$ we have

$$(\boldsymbol{D} \oplus \boldsymbol{E})_{ij} = d_{ij} \oplus e_{ij} = \max(d_{ij}, e_{ij}), \tag{7}$$

$$(\boldsymbol{D} \otimes \boldsymbol{F})_{ij} = \bigoplus_{k=1}^{n} d_{ik} \otimes f_{kj} = \max_{k=1,\ldots,n} (d_{ik} + f_{kj}).$$

A general class of DEDSs has a nonlinear representation in the conventional algebra. But a certain class of DEDSs, called "max-plus linear systems", has a

linear representation based on the above framework with the state space equations as follows:

$$\boldsymbol{x}(k) = \boldsymbol{A} \otimes \boldsymbol{x}(k-1) \oplus \boldsymbol{B} \otimes \boldsymbol{u}(k), \tag{8}$$
$$\boldsymbol{y}(k) = \boldsymbol{C} \otimes \boldsymbol{x}(k). \tag{9}$$

Here the following notation is used:

- $\boldsymbol{x}(k) \in \mathbb{R}_{\varepsilon}^{n}$ is the state vector including the time instants that correspond to the internal events at the k-th counter,
- $\boldsymbol{u}(k) \in \mathbb{R}_{\varepsilon}^{r}$ is the input vector including the time instants that correspond to input events at the k-th counter,
- $\boldsymbol{y}(k) \in \mathbb{R}_{\varepsilon}^{m}$ stands for the output vector including the time instants that correspond to the output events occurring at the k-th counter,
- $\boldsymbol{A} \in \mathbb{R}_{\varepsilon}^{n \times n}$ denotes the state transition matrix, $\boldsymbol{B} \in \mathbb{R}_{\varepsilon}^{n \times r}$ denotes the control matrix and $\boldsymbol{C} \in \mathbb{R}_{\varepsilon}^{m \times n}$ denotes the output matrix.

4.1. Modeling the full-automatic and the redundant human operator parts. As stated in the previous section, the battery assembly system can be divided into two parts such as the deterministic part based on fully automated devices and the non–deterministic part containing two semi–automatic assembly stations operated by a set of human-operators. This distinction necessitates different approaches for modelling individual parts. Let us start with presenting the max-plus algebra state space equation for the deterministic part. Here (2) can be transformed into (8) and (9), where the system matrices are

$$\boldsymbol{A} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & d_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & \varepsilon & \varepsilon \\ 2d_1 + t_{1,4} & \varepsilon & \varepsilon & d_4 & \varepsilon \\ \varepsilon & 2d_2 + t_{2,5} & 2d_3 + t_{3,5} & \varepsilon & d_5 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} t_{in,1} & \varepsilon & \varepsilon \\ \varepsilon & t_{in,1} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ t_{in,1} + d_1 + t_{1,4} & \varepsilon & \varepsilon \\ \varepsilon & t_{in,2} + d_2 + t_{2,5} & \varepsilon \\ \varepsilon & \varepsilon & t_{in,3} \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & t_{in,3} + d_3 + t_{3,5} \end{bmatrix}.$$

According to a similar line of reasoning as in the case of the deterministic part, the matrix-based form of the part of human operators presented between the stations R_5 and R_6 in Fig. 1 should be elaborated. Max-plus algebra is a natural formalism for modeling a certain class of DEDS, where there is synchronization, but no concurrency.

The analytical description of concurrency presented in max-plus algebra is based on the switch max-plus algebra linear framework with an additional set of decision variables (i.e.$\{(v_i(k), v_i(k-1))\}_{k=1}^{N_p}$, where N_p denotes the predictive control horizon. The model (3)–(5) can be converted as follows:

$$x_5(k) = \boldsymbol{A}(v(k-1), v(k), k) \otimes x_5(k-1) \oplus \boldsymbol{B}(v(k), k) \otimes z(k), \tag{10}$$

where

$$x_5(k) = [x_{5,v_1}(k), x_{5,v_2}(k), \ldots, x_{5,n_w}(k), x_6(k)]^T,$$
$$v(k) = [v_1(k), \ldots, v_{n_w}(k)]^T,$$

$\boldsymbol{A}(\cdot,\cdot,\cdot) \in \mathbb{R}_{\max}^{n_w+1 \times n_w+1}$ signifies the state matrix and $\boldsymbol{B}(\cdot,\cdot) \in \mathbb{R}_{\max}^{n_w+1}$ is the control matrix.

It should be noted that (3) does not conflict with (10). In (10), $x_5(k)$ stands for the vector, where the i-th element stands for the instant at which the i-th human operator is ready to start the assembly task. On the other hand, in (3) $x_5(k)$ is a scalar and denotes the instant when the assembly task can be started because all the necessary components have been delivered to R_5. Taking into account (3)–(5), the delivery time of the k-th sub-frame to the station R_6 can be obtained as follows:

$$\begin{aligned} x_6(k) = \max(&x_{5,1}(k) + d_5 + c_1(k) + v_1(k),\ x_{5,2}(k) \\ &+ d_5 + c_2(k) + v_2(k),\ x_{5,3}(k) + d_5 \\ &+ c_3(k) \ldots,\ x_{5,n_w}(k) + d_6 + c_{n_w}(k) \\ &+ v_{n_w}(k),\ x_6(k-1) + d_6). \end{aligned} \tag{11}$$

Thus, substituting (3) into (11) yields

$$\begin{aligned} x_6(k) = \max(&x_{5,1}(k-1) + 2d_5 + c_1(k-1) + c_1(k) \\ &+ d_6 + b_1(k-1) + v_1(k),\ x_{5,2}(k) + 2d_5 \\ &+ c_2(k-1) + c_2(k) + d_6 + b_2(k-1) \\ &+ v_2(k), \ldots, \\ &x_{5,n_w}(k) + 2d_5 + c_{n_w}(k-1) + c_{n_w}(k) + d_6 \\ &+ b_{n_w}(k-1) + v_{n_w}(k),\ z(k) + d_5 + c_1(k) \\ &+ v_1(k),\ z(k) + d_5 + c_1(k) + v_1(k),\ z(k) \\ &+ d_5 + c_2(k) + v_2(k), \ldots, z(k) + d_5 \\ &+ c_{n_w}(k) + v_{n_w}(k),\ x_6(k-1) + d_6). \end{aligned} \tag{12}$$

where $z(k) = x_4(k) + d_4 + t_{4,6}$.

From (10)–(12) it follows that the structure of matrices $A(v(k-1), v(k), k)$, $B(v(k), k))$ depends on the decision variables $v_{i,j}(k)$, $v_{i,j}(k+1)$. To specify the content of the matrices $A(v(k-1)$, $B(v(k), k)$ for the

Table 1. KIS.ME parameters.

Name	Value
Luminous element color	RGB
Degree of protection	IP65
WLAN Standard	IEEE 802.11 b/g/n 2.4 GHz
Connection terminal	M12 8-pin A-coded
Operating voltage	$5 \pm 10\%V$, $24 \pm 20\%V$
GPIO	2 Inputs/ 2 Outputs

Table 2. KIS.LIGHT conditions.

KL	KL Input1	KL Input2
red	0	0
green	1	0
green	0	1
blue	1	1

k-th counter, the values of decision variables of both the $(k-1)$-th and the k-th counters have to be determined. The matrices $A(v(k-1), v(k), k)$ and $B(v(k), k)$, for notational convenience, are denoted by $A_v(k)$ and $B_v(k)$ and are given by (16).

5. Fuzzy model of human operator performance

The main purpose of this section is to outline a step-by-step procedure for obtaining human operator performance. To incorporate the human operator's procedure into the assembly battery system, the KIS.ME IoT infrastructure is used. It contains two components—the first one is the software IoT cloud KIS.MANAGER and the hardware constitutes the second. The hardware component is formed by

- **KIS.BOX (KB)**: a decentralized twin button box with an integrated WiFi interface,
- **KIS.LIGHT (KL)**: a signal light with an integrated WiFi interface.

All KIS.LIGHTs and KIS.BOXes can be easily virtualized with KIS.MANAGER. As the outlet points of the assembly station are stationary, KIS.MANAGER also allows a visualization of their digital twin on the scheme of battery system. Both the components (KIS.LIGHT and KIS.BOX) are equipped with a digital GPIO interface (general-purpose input/output) and their general parameters are given in Tab. 1.

The two digital inputs of the KL portrayed in Fig. 1 are connected with three optical sensors, which sets them in a binary way, if a subframe (Input 1) and cells package (Input 2) are available for mounting, respectively. This corresponds to the four possible states, which are listed in columns 2 and 3 of Table 2. Having KB and KL states, it is possible to implement their transitions, which can be easily done using the KIS.MANAGER rule engine (see the manual by RAFI (2021) for an example tutorial). Each operator's workspace has two KB buttons and one KL button. Assigning two separate buttons KL_1 and K_2 allows the main supervision system to monitor elementary operations performed by individual human operators. In addition, the use of two buttons allows for greater parallelism of the tasks performed by operators, and thus to reduce the total production time.

A necessary condition for installing the cells in the sub-frame is the availability of all elements (which is signalled by the blue color of KL) and the completion of the assembly task by the human operator in the previous iteration (which is signalled by the green illumination of KB_1 and KB_2 colour). Taking into account the above assumptions, the standard operating procedure (SOP) for a human operator can be defined by the steps shown in Table 3. In the first step, the human operator takes all the parts, which need to be fitted together and KL automatically turns red (see Table 2). The KB transitions from States 1–2 and 2–3 are made by a human by pressing the first one, performing the appropriate action. Similarly, transitions from States 3–4 and 4–5 are carried out by pressing the second button, respectively. Thus, the human operators cyclically execute the sequence states 1–5. The time evolution of all states can be easily measured in KIS.MANAGER with the *Trend widget* that allows a real-time analysis of operator performance. Historical data can be easily archived in the `csv` file, and thus processed further in external software.

The fuzzy model constitutes a set of three sub-models:

$$c^F = h_c(S_O, E, D, N_C), \tag{13}$$

$$b^F = h_b(S_O, E, D), \tag{14}$$

$$d_5^F = h_{d_5}(S_O, E, N_C), \tag{15}$$

where

- $h_{(\cdot)}$ stands for a given model structure;
- S_O is the operator's current time (ranging in 0–8 hours) during an 8 hour shift;
- E denotes the experience of a human operator;
- D is the distance (in meters) between a human workplace and the assembly station where the sub-frame is mounted in the main frame;
- N_C is the number of mounted cells to the sub-frame depending on the type of battery. This parameter can be overridden by the weight or the capacity of the sub-frame with installed cells.

Table 3. KIS.BOX states.

State	KB_1	KB_2	Action
1	blue	blue	Can start
2	red	blue	Cell mounting into the sub-frame
3	green	blue	Cell mounting completed
4	green	red	Transport to the station R_6 and mounting the sub-frame into the main-frame
5	green	green	Mounting completed and returning to station R_5

Table 4. Fuzzy premise variables.

Variable	Description	Intervals
E	Well-qualified	0.6-1.0
	Qualified	0.3-0.7
	Beginner	0.0-0.4
S_O	Long	5-8
	Medium	3-6
	Short	0-4

From (13) it follows that the transportation time of a sub-frame depends on the number of cells and the distance. In the context of an operator's return without a sub-frame, the parameter N_C in (14) is meaningless. A similar case concerns the lack of the parameter D in (15). The other three parameters have direct impact on the times of all tasks considered. Taking into account such nomenclature, it is proposed to model the human operators performance using a Takagi–Sugeno (T–S) fuzzy model (Tanaka and Sugeno, 1992). For that purpose, two fuzzy premise variables are introduced and described in Table 4. The fuzzy variables E and S_O are related with the triangular membership function spanned over the intervals given in Table 4. These membership functions create the fuzzy sets $\mathbb{M}_{E,i}$ $\mathbb{M}_{S,i}$, where $i = 1, \ldots, 3$.

In this paper, the T–S counterpart of (13)–(14) is formulated as follows:

$$\begin{aligned} &\text{IF } S \in \mathbb{M}_{S,i} \text{ and } \mathbb{W}_T \in \mathbb{M}_{W,j}, \text{ THEN} \\ &c^F = p_{1,c}D + p_{2,c}N_C, \\ &b^F = p_{1,b}D, \\ &d_5^F = p_{1,d}N_C, \end{aligned} \tag{17}$$

where $p_{i,j}$ are parameters which have to be determined. Using a conventional approach (Tanaka and Sugeno, 1992), model (17) can be transformed into

$$\begin{aligned} c^F &= \sum_{j=1}^{N} \mu_j(E, S_O)(p_{1,c}^j D + p_{2,c}^j N_C), \\ b^F &= \sum_{j=1}^{N} \mu_j(E, S_O)(p_{1,b}^j D), \\ d_5^F &= \sum_{j=1}^{N} \mu_j(E, S_O)(p_{1,d}^j N_C), \\ &\sum_{j=1}^{N} \mu_j(E, S_O) = 1, \quad \mu_j(E, S_O) \geq 0, \end{aligned} \tag{18}$$

where $\mu_j(E, S_T)$ is the normalized rule firing strength formed with the above-defined membership functions. Finally, the T–S model can be written in the final form

$$c^F = r_c^T p_c, \quad b^F = r_b^T p_b, \quad d^F = r_d^T p_d, \tag{19}$$

where

$$\begin{aligned} p_b &= [p_b^1, \ldots, p_b^N]^T, \\ p_c &= [p_c^1, \ldots, p_c^N]^T, \\ p_d &= [p_d^1, \ldots, p_d^N]^T, \\ p_x^i &= [p_{1,x}^i, p_{2,x}^i, p_{3,x}^i]^T, \\ r_c &= [\mu_1(\cdot,\cdot)[D, N_C]^T, \ldots, \mu_N(\cdot,\cdot)[D, N_C,]^T]^T, \\ r_b &= [\mu_1(\cdot,\cdot)D, \ldots, \mu_N(\cdot,\cdot)D]^T, \\ r_d &= [\mu_1(\cdot,\cdot)N_C, \ldots, \mu_N(\cdot,\cdot)N_C]^T. \end{aligned}$$

As a result, the recursive least-squares algorithm (Witczak, 2014) can be efficiently used for estimating p_b, p_c and p_c. The resulting model can be used for the assessment a nominal operator performance time $d_{\text{ref}} = d_5^F + c^F + d_6 + b^F$ (see Fig. 1). Thus, any breach of this reference can be assumed as a fault occurring in the system, which should be properly compensated.

6. Fault-tolerant control

In order to complete the description of the entire control strategy (i.e. the comprehensive synchronization of transportation and assembly tasks with full scheduling of them), a set of constraints specifying the additional rules of tasks' execution should be developed. These constraints can be represented in the following form:

- **Scheduling.** Assembly tasks have to fulfill time requirements defined by reference trajectory:

$$x_j(k) \leq x_{\text{ref},j}(k), \quad j = 1, \ldots, s, \tag{20}$$

  where $x_{ref,j}(k)$ is the upper bound of $x(k)$ at time k.

- **Transfer constraint.** This constrain ensures that the k-th main-frame is released from the station R_6 before the l-th main-frame ($l > k$) arrives:

$$\begin{aligned} &\text{IF } (l > k \text{ and } x_6(k) = x_6(l)) \\ &\text{THEN } x_6(k) \leq x_6(l) + \alpha(j), \end{aligned} \tag{21}$$

$$\boldsymbol{A}_v(k) = \begin{bmatrix} d_5 + d_6 + c_1(k-1) + b_1(k-1) & & \\ \varepsilon & & \\ \vdots & & \\ 2d_5 + d_6 + c_1(k-1) + b_1(k-1) + c_1(k) + v_1(k) & & \\ \varepsilon & \dots & \varepsilon \\ d_5 + d_6 + c_2(k-1) + b_2(k-1) & \dots & \varepsilon \\ \vdots & \ddots & \vdots \\ 2d_5 + d_6 + c_2(k-1) + b_2(k-1) + c_2(k) + v_2(k) & \dots & d_6 \end{bmatrix}, \tag{16}$$

$$\boldsymbol{B}_v(k) = [v_1(k), v_2(k), \dots, v_{n_w}(k), d_5 + \max(c_1(k) + v_1(k), c_2(k) + v_2(k), \dots, c_{n_w}(k) + v_{n_w}(k)]^T$$

where $\alpha(j) > 0$ is an unknown time between the k-th main-frame leaving the station and the l-th main-frame being delivered. The parameter $\alpha(j) > 0$ prevents the unrealistic situation, where the k-th and the l-th main-frame leaves and arrives, respectively, from R_6 at the same time respectively, i.e., $x_6(k) = x_5(l)$. However, it should be stressed that $\alpha(l)$ should be as small as possible.

- **Concurrency.** This constraint concerns the choice of a human operator who will guarantee a main-frame handling according to an assumed time trajectory.

$$v_i(k) = e, \quad v_j(k) = -\infty, \qquad \forall i \neq j. \tag{22}$$

Thus, under the above constraints, the problem reduces to determining a sequence of pairs

$$(x_5(k), v_i(k)), \quad k = 0, 1, 2, \dots \tag{23}$$

resulting in the k-th sub-frame start time $x_5(k)$ along with $v_i(k)$ the i-th human operators carrying out this action.

- **Transportation constraint.** The operational times of the i-th human operator's assembling cells and transporting the k-th sub-frame obey

$$b_i(k) = \max\left(e, b_i^F(k) + v_i(k)\right), \tag{24}$$

$$c_i(k) = \max\left(e, c_i^F(k) + v_i(k)\right) \tag{25}$$

where c^F and b^F are obtained using the fuzzy logic operator performance framework (13) and (14). It is obvious that if the i-th human operator does not perform his/her tasks for the k-th sub-frame, then $v_i(k) = \varepsilon = -\infty$, which implies that $b_i(k) = c_i(k) = \max(e, \varepsilon) = \max(0, \varepsilon) = 0$.

The task scheduling strategy is based on the MPC paradigm that has a natural ability of handling constraints. The main aim of this section is to provide the FTC scheme that is able to deal with human operators' faults. Before proceeding to the FTC algorithm, the two parts (deterministic and non-deterministic) of the system have to be coupled in the following form:

$$\hat{x}(k) = A_t(k) \otimes \hat{x}(k-1) \oplus B_t(k) \otimes \hat{u}(k), \tag{26}$$

where we have $\hat{x}(k) = [x(k)^T, x_5(k)^T]^T$, $\hat{u}(k) = [u(k)^T, z(k)^T]^T$ and

$$A_t(k) = \begin{bmatrix} A & \epsilon \\ \epsilon & A_v(k) \end{bmatrix}, \quad B_t(k) = \begin{bmatrix} B & \epsilon \\ \epsilon & B_v(k) \end{bmatrix}, \tag{27}$$

while ϵ stands for a matrix of appropriate dimension containing ε's only. The core of the problem is to determine the input sequence $\hat{u}(k), \dots, \hat{u}(k + N_p - 1)$ on a moving horizon $k, \dots, k + N_p - 1$ which minimizes two cost functions. The first one

$$J_{\hat{u}} = -\sum_{k=0}^{n_P - 1} (\hat{u})(k) \tag{28}$$

allows determining a possibly large average assembly and transportation start time. The second cost function

$$J_\alpha = \sum_{j=1}^{n_\alpha} \alpha(j) \tag{29}$$

allows us to minimize necessary times between the k-th main-frame leaving the station—operated by human operators—and the l-th main-frame being delivered to this station.

Criteria (21), (22) and (23) reflect different aims, but the impact of each criterion on an operator's performance can be determined by changing the values of the following parameters:

$$J = \gamma_1 J_{(\hat{u})} + \gamma_2 J_\alpha, \tag{30}$$
$$\gamma_1 + \gamma_2 = 1, \quad \gamma_i \geq 0, \quad i = 1, 2,$$

where each γ_i signifies the importance of a given criterion. It should be noted that a designer, by changing γ_i,

can attain a desired system behaviour. Taking into account (30) along with constraints (20)–(25), it is possible to get a comprehensive framework to predict the schedule of the operators' tasks according to the reference schedule. Such a model was proposed by Majdzik *et al*. (2016), Seybold *et al*. (2015) or Van Den Boom and De Schutter (2006) and is based on the max-plus algebra formalism with the definition of additional decision variables by Witczak *et al*. (2019) or Majdzik (2020) to support both deterministic synchronization and non-deterministic choice.

Taking into account (10) and (16), the input sequence $z(k), \ldots, z(k+N_p-1)$ on a moving horizon $k, \ldots, k+N_p-1$ should be selected in order to minimize the cost function given by (30). A recursive application of (26) yields

$$\tilde{x}(k) = \boldsymbol{M}(\tilde{v}(k)) \otimes \tilde{x}(k-1) \oplus \boldsymbol{H}(\tilde{v}(k)) \otimes \tilde{u}(k), \quad (31)$$

where the matrices $\boldsymbol{M}(\tilde{v}(k))$ and $\boldsymbol{H}(\tilde{v}(k))$ are specified by recursive substitutions (see the work of Witczak *et al*. (2019) for a complete description). Having an ability of calculating (31), a complete optimization problem reduces to

$$(\tilde{u}(k)^*, \tilde{v}(k)^*) = \arg \min_{\tilde{u}(k), \tilde{v}(k)} J(\hat{u})(k), \quad (32)$$

under the set constraints (20)–(25).

In this paper three of the four possible human operator's faults are taken into account. The considered potential faults concern the tasks which are most susceptible to the changes of the following parameters: the weight of individual elements, the different route between stations for each person, the different experiences of individual operators. These faults can be presented as follows:

$$\begin{aligned}
&\text{IF } c_i(k) \le c_i^{\text{ref}}(k), \text{ THEN } f_{i,c}(k) = 0,\\
&\text{ELSE } f_{i,c}(k) = c_i(k) - c_i^{\text{ref}}(k),\\
&\text{and}\\
&\text{IF } b_i(k) \le b_i^{\text{ref}}(k), \text{ THEN } f_{i,b}(k) = 0,\\
&\text{ELSE } f_{i,b}(k) = b_i(k) - b_i^{\text{ref}}(k),\\
&\text{and}\\
&\text{IF } d_{5,i}(k) \le d_{5,i}^{\text{ref}}(k), \text{ THEN } f_{i,d_5}(k) = 0,\\
&\text{ELSE } f_{i,d_5}(k) = d_{5,i}(k) - d_{5,i}^{ref}(k).
\end{aligned} \quad (33)$$

The final result of the proposed strategy constitutes Algorithm 1.

7. Case study

The purpose of this section is to conduct an experiment to evaluate the performance of the proposed solution. Let us start with analyzing the fault-tolerance control within the operational cycle with the number of human operators equals $n_w = 3$. As described in Section 3, the human operator performs the task consisting of four separate operations in each cycle. In this experiment, it was assumed that there were faults in three of them. These faults come down to the third operator and are equal to:

Algorithm 1. Integrated FTC.

Step 0. Initialize: $k = 1$, $v(0)$ and N_p.

Step 1. For the the i-th human operator mounting and transporting $k-1$ sub-frame, i.e., $v_i(k-1) = e$, get T–S based $d_5^F(k-1)$, $c^F(k-1)$, $b^F(k-1)$ and then measure $b_i(k-1)^m$, $c_i(k-1)^m$ and $d_{5,i}(k-1)^m$ using KIS.ME and get $x_5(k-1)$.

Step 2. Perform fault estimation with (33) and set $\hat{f}_{i,d_5}(k) = f_{i,d_5}(k-1)$, $\hat{f}_{i,c}(k) = f_{i,c}(k-1)$ and $\hat{f}_{i,b}(k) = f_{i,b}(k-1)$

Step 3. Determine $b_i^F(k)$, $c_i^F(k)$ and $d_{5,i}^F(k)$, $i = 1, \ldots, N_p$.

Step 4. If $\hat{f}_{i,d_5} \neq 0$ and/or $\hat{f}_{i,c} \neq 0$, update $A(\cdot,\cdot,\cdot)$ in (16), corresponding to $x_{5,i}(k)$ of the i-th human operator:

$$\begin{aligned}
A_{v,i,i}(\cdot,\cdot,\cdot)_{i,i} &= b_i(k-1) + c_i(k-1)\\
&\quad + \hat{f}_{i,b} + \hat{f}_{i,c} + \hat{f}_{i,d}, \quad (34)\\
A_{v,n_v+1,i}(\cdot,\cdot,\cdot) &= b_i(k-1) + c_i(k-1)\\
&\quad + c_i(k) + \hat{f}_{i,b} + 2\hat{f}_{i,c} \quad (35)\\
&\quad + \hat{f}_{i,d} + v_i(k),
\end{aligned}$$

and set

$$d_{5,i}(k) = \max\left(e, d_{5,i}^F(k) + \hat{f}_{i,d} + v_i(k)\right), \quad (36)$$

$$b_i(k) = \max\left(e, b_i^F(k) + \hat{f}_{i,b} + v_i(k)\right), \quad (37)$$

$$c_i(k) = \max\left(e, c_i^F(k) + \hat{f}_{i,c} + v_i(k)\right). \quad (38)$$

If $\hat{f}_{i,b} \neq 0$, then set

$$B_{v,n_v+1}(\cdot,\cdot) = b_v(k) \quad (39)$$

while

$$\begin{aligned}
b_v(k) &= \max(c_1(k) + v_1(k), \ldots, c_{n_v}(k)\\
&\quad + v_{n_v}(k)). \quad (40)
\end{aligned}$$

Step 5. Obtain $\tilde{v}(k)^*$, $\tilde{u}(k)^*$, $\tilde{\alpha}(k)$ by solving (32) under (21)–(25).

Step 6. Send $u(k)^*$ and $v(k)^*$ to the system.

Step 7. Update $k = k+1$ and go to Step 1.

- assembly operator faults:

$$f_{d_5,2}(k) = \begin{cases} 0 & k < 7, \\ 0.4 & \text{otherwise.} \end{cases} \tag{41}$$

- transportation faults:

$$f_{c,2}(k) = \begin{cases} 0 & k < 7, \\ 0.2 & \text{otherwise.} \end{cases} \tag{42}$$

$$f_{b,2}(k) = \begin{cases} 0 & k < 7, \\ 0.1 & \text{otherwise.} \end{cases} \tag{43}$$

The system is naturally initialized with $\hat{x}(0) = 0$ and the algorithm is operating with a prediction horizon equal to $N_p = 3$. Moreover, the scheduling constraint is related to the assembly station (represented by the R_6 station in Fig. 1) and it is formulated by:

$$t_{\text{ref},6} = [3.6, 4.6, 5.6, 6.6, 7.6 \ldots]. \tag{44}$$

Figures 2–5 present the obtained results. Two variants of Algorithm 1 are considered without (by removing Step 2 and Step 4 from Algorithm 1) and with fault tolerance control mechanism. As can be observed in Fig. 2, the initial states of without and with FTC are exactly the same. In 8-th even counter the second human operator performed the assembly operation at the station R_5, the transport operation to the station R_6 and the return operation to the station R_5 in 0.4 min, 0.2 min and 0.1 min time longer than the nominal time, respectively. It can be observed in Fig. 4 that the effect of the fault both for without and with FTC is the same and causes the important divergence from the reference schedule. But the FTC scheme eliminates the fault actively (Fig. 5). The Gantt diagram from Fig. 3 shows that the less experienced worker is less frequently selected to deliver the sub-frame to the station R_6.

8. Conclusion remarks

The paper aims at integrating a human operator within a partially automated assembly system. Indeed, the proposed framework allows coupling the deterministic, i.e., a fully automated, part and non-deterministic ones handled by human operators. To settle such a challenge, the KIS.ME IoT infrastructure was utilized, which is based on an intuitive push buttons and signalling lamps, which are communicated with a cloud-based KIS.MANAGER. Such a communication is realized with an existing WiFi network. Using such infrastructure, one can easily measure and monitor the performance of human operators. To form a reference for a human operator performance, a Takagi–Sugeno model is designed, which is based on the data gathered using KIS.MANAGER. Having such a digital twin of a human behaviour, one can identify the faults, which pertain possible delays, i.e., discrepancies between the human behaviour and the one provided with the digital twin. This makes it possible to form a unique fault-tolerant scheduling framework, which aims at minimizing the effect of faults. The paper is concluded with a set of validation results, which clearly exhibits the profits pertaining an integration KIS.ME and the fault-tolerant scheduling framework.

Table 5. Operation times.

Name	Time [min]
d_1	0.3
d_2	0.4
d_3	0.2
d_4	1.2
d_5	2.5 or 2.8
d_6	0.7

Table 6. Transportation times.

Name	Time [min]
$t_{\text{in},1}$	0.1
$t_{\text{in},2}$	0.2
$t_{\text{in},3}$	0.1
$t_{1,4}$	0.2
$t_{2,5}$	0.2
$t_{3,5}$	0.3
$t_{4,6}$	0.1
c_i^{ref}	0.2
b_i^{ref}	0.1

where $i = 1, 2, 3$.

Acknowledgment

The work was partially supported by the National Science Centre, Poland, under the grant UMO-2017/27/B/ST7/00620.

References

Baccelli, F., Cohen, G., Olsder, G.J. and Quadrat, J.-P. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*, John Wiley& Sons, Hoboken.

Baruwa, O.T., Piera, M.A. and Guasch, A. (2015). Deadlock-free scheduling method for flexible manufacturing systems based on timed colored Petri nets and anytime heuristic search, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **45**(5): 1–12.

Butkovic, P. (2010). *Max-Linear Systems: Theory and Algorithms*, Springer, London.

Dizdar, E.N. and Koçar, O. (2020). Fuzzy logic-based decision-making system design for safe forklift truck speed: Cast cobblestone production application, *Soft Computing* **24**(19): 1–14.

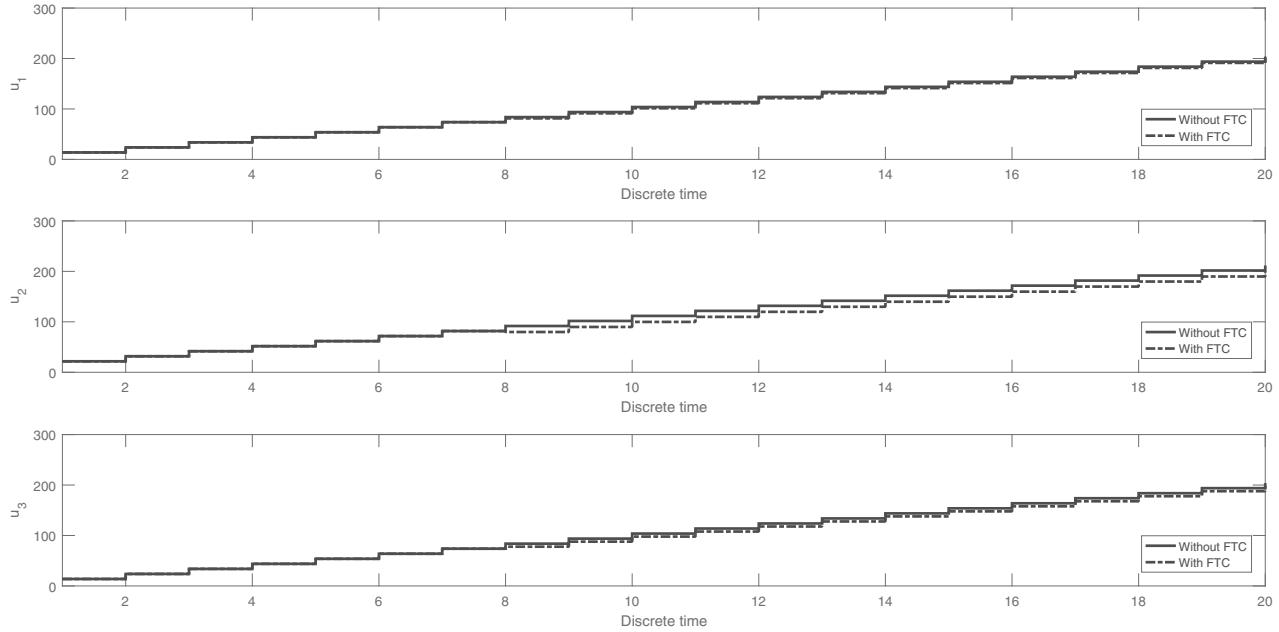

Fig. 2. Input strategy for the assembly system.

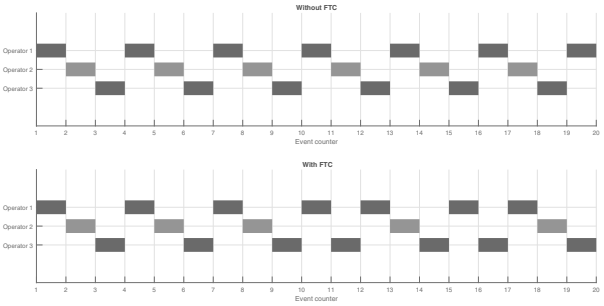

Fig. 3. Scheduling strategy for the operators.

Ebrahimi, A., Sajadi, S., Roshanzamir, P. and Azizi, M. (2015). Determining the optimal performance of flexible manufacturing systems using network analysis and simulation process, *International Journal of Management, Economics and and Social Sciences* **4**(1): 12–17.

Groover, M. (2014). *Automation, Production Systems, and Computer-Integrated Manufacturing, 3rd Edition*, Pearson, London.

Kopetz, H. (2011). *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Springer, Boston.

Madakam, S., Ramaswamy R. and Tripathi, S. (2015). Internet of things (IoT): A literature review, *Journl of Computing and Communications* **3**(05): 164.

Majdzik, P. (2020). Feasible schedule under faults in the assembly system, *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), Shenzhen, China*, pp. 1049–1054.

Majdzik, P., Akielaszek-Witczak, A., Seybold, L., Stetter, R. and Mrugalska, B. (2016). A fault-tolerant approach to the control of a battery assembly system, *Control Engineering Practice* **55**: 139–148.

Majdzik, P., Witczak, M., Lipiec, B. and Banaszak, Z. (2021). Integrated fault-tolerant control of assembly and automated guided vehicle-based transportation layers, *International Journal of Computer Integrated Manufacturing*: 1–18, DOI: 10.1080/0951192X.2021.1872103.

Mircetic, D., Ralevic, N., Nikolicic, S., Maslaric, M. and Stojanovic, D. (2016). Expert system models for forecasting forklifts engagement in a warehouse loading operation: A case study, *PROMET—Traffic&Transportation* **28**(4): 393–401.

Nivolianitou, Z. and Konstantinidou, M. (2018). A fuzzy modeling application for human reliability analysis in the process industry, *in* H. Pham (Ed.), *Human Factors and Reliability Engineering for Safety and Security in Critical Infrastructures*, Springer, London, pp. 109–154.

RAFI GmbH & Co. KG (2021). KIS.ME User Manual, RAFI GmbH & Co. KG, Berg, `https://kisme.rafi.de/documents/KISME-UserManual.pdf`.

Rousset, A., Herrmann, B., Lang, C. and Philippe, L. (2016). A survey on parallel and distributed multi-agent systems for high performance computing simulations, *Computer Science Review* **22**: 27–46.

Rutkowski, T., Łapa, K. and Nielek, R. (2019). On explainable fuzzy recommenders and their performance evaluation, *International Journal of Applied Mathematics and Computer Science* **29**(3): 595–610, DOI: 10.2478/amcs-2019-0044.

Salazar, J.C., Sanjuan, A., Nejjari, F. and Sarrate, R. (2020). Health-aware and fault-tolerant control of an octorotor

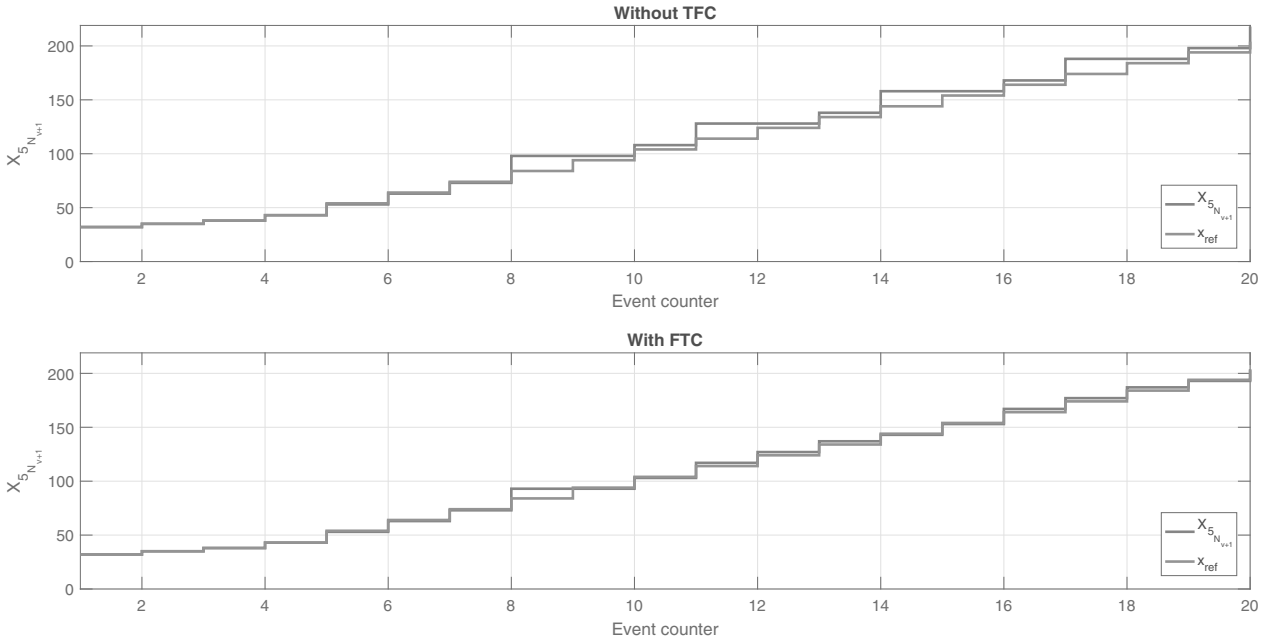

Fig. 4. Output times of the 5th station.

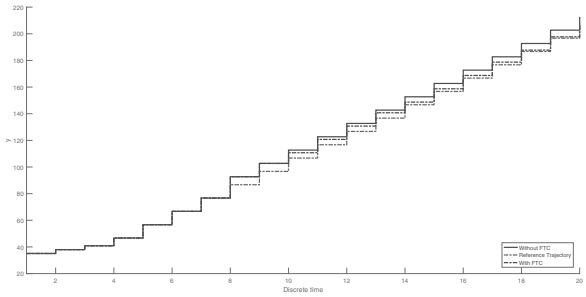

Fig. 5. Output sequence for the assembly system.

UAV system based on actuator reliability, *International Journal of Applied Mathematics and Computer Science* **30**(1): 47–59, DOI: 10.34768/amcs-2020-0004.

Segura, M.A., Hennequin, S. and Finel, B. (2016). Human factor modelled by fuzzy logic in preventive maintenance actions, *International Journal of Operational Research* **27**(1–2): 316–340.

Seybold, L., Witczak, M., Majdzik, P. and Stetter, R. (2015). Towards robust predictive fault-tolerant control for a battery assembly system, *International Journal of Applied Mathematics and Computer Science* **25**(4): 849–862, DOI: 10.1515/amcs-2015-0061.

Tanaka, K. and Sugeno, M. (1992). Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems* **45**(2): 135–156.

Van Den Boom, T. and De Schutter, B. (2006). Modelling and control of discrete event systems using switching max-plus-linear systems, *Control Engineering Practice* **14**(10): 1199–1211.

Witczak, M. (2014). *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems*, Springer, Heidelberg.

Witczak, M., Majdzik, P., Stetter, R. and Lipiec, B. (2019). Multiple AGV fault-tolerant within an agile manufacturing warehouse, *IFAC-PapersOnLine* **52**(13): 1914–1919.

Witczak, M., Majdzik, P., Stetter, R. and Lipiec, B. (2020). A fault-tolerant control strategy for multiple automated guided vehicles, *Journal of Manufacturing Systems* **55**(4): 56–68.

Paweł Majdzik was born in Poland in 1967. He received his MSc degree in electrical engineering from the Wrocław University of Technology (Poland) and his PhD degree from the Poznań University of Technology (Poland) in 1992 and 1998, respectively. He has been an assistant professor of computer science at the Institute of Control and Computation Engineering, University of Zielona Góra (Poland), since 1998. His current research interests include design and optimization of DESs, modelling and control of DESs, and fault-tolerant control. Paweł Majdzik has published more than 45 papers in international journals and conference proceedings. He is an author of two monographs.

Received: 7 November 2021
Revised: 14 January 2022
Accepted: 28 January 2022