DE GRUYTER OPEN

Int. J. Appl. Math. Comput. Sci., 2015, Vol. 25, No. 4, 849–862
DOI: 10.1515/amcs-2015-0061

amcs

TOWARDS ROBUST PREDICTIVE FAULT–TOLERANT CONTROL FOR A BATTERY ASSEMBLY SYSTEM

LOTHAR SEYBOLD [a], MARCIN WITCZAK [b,*], PAWEŁ MAJDZIK [b], RALF STETTER [c]

[a] Department of Research & Development
RAFI GmbH Co. KG, Ravensburger Straße, 128–134, D-88276 Berg/Ravensburg, Germany
e-mail: lothar.seybold@rafi.de

[b] Institute of Control and Computation Engineering
University of Zielona Góra, ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, Poland
e-mail: {m.witczak,p.majdzik}@issi.uz.zgora.pl

[c] Faculty of Mechanical Engineering
University of Applied Sciences Ravensburg–Weingarten, Building D., Doggenriedstraße, Weingarten, Germany
e-mail: stetter@hs-weingarten.de

The paper deals with the modeling and fault-tolerant control of a real battery assembly system which is under implementation at the RAFI GmbH company (one of the leading electronic manufacturing service providers in Germany). To model and control the battery assembly system, a unified max-plus algebra and model predictive control framework is introduced. Subsequently, the control strategy is enhanced with fault-tolerance features that increase the overall performance of the production system being considered. In particular, it enables tolerating (up to some degree) mobile robot, processing and transportation faults. The paper discusses also robustness issues, which are inevitable in real production systems. As a result, a novel robust predictive fault-tolerant strategy is developed that is applied to the battery assembly system. The last part of the paper shows illustrative examples, which clearly exhibit the performance of the proposed approach.

Keywords: max-plus algebra, interval analysis, battery assembly, model predictive control, fault-tolerant control.

1. Introduction

Manufacturing systems, in particular, flexible ones, should be capable of adapting to change in product demands, shorter product life cycles, a higher product variety, requirements for shorter delivery times, and a higher quality and should allow responding to the fast changes in the economical market. Indeed, modern manufacturing systems proceed towards agile manufacturing, which significantly increases these demands (Gunasekaran, 1999). Model predictive control (MPC) is able to meet these demands in many practical production systems, especially in the continuous-time framework (Rossiter, 2013; Prodan *et al*., 2013). Thus, as the number of MPC applications constantly proliferates, this recommends its application the industry-oriented control tasks.

Indeed, there exist many reasons for applying MPC for control in the process industry. It is an easy-to-tune model-based controller design procedure that can handle multi-input multi-output processes and constraints imposed on the inputs and outputs of the system. Moreover, MPC can be employed for structural changes, such as sensor and/or actuator faults (Camacho and Bordons, 1997) and variations in the system parameters (it is of special importance in flexible manufacturing systems (De Schutter and Van Den Boom, 2001)) by using a moving horizon approach, in which the model and control strategy are continuously updated. Such a policy is called fault-tolerant control (FTC) (Blanke *et al*., 2006; Witczak, 2014).

A permanent increase in the complexity, efficiency, and reliability of modern industrial system necessitates continuous development of control and fault diagnosis. A moderate combination of these

*Corresponding author

two paradigms is intensively studied under the name of FTC. FTC systems are divided into two distinct classes (Zhang and Jiang, 2008): passive and active. In passive FTC, controllers are designed to be robust against a set of predefined faults; therefore, there is no need for fault diagnosis, but such a design usually degrades the overall performance. In contrast to passive ones, active FTC schemes react to faults actively by reconfiguring control actions in such a way that the system stability and acceptable performance are maintained. To attain this objective the control system relies on fault detection and isolation (FDI) (Korbicz *et al.*, 2004; Li *et al.*, 2007; Mrugalski, 2013; Witczak, 2007; Chen *et al.*, 2011) as well as an accommodation technique (Blanke *et al.*, 2006). Most of the existing works treat the FDI and FTC problems separately. Unfortunately, perfect FDI and fault identification are impossible, and hence there is always an inaccuracy related to this process.

The paper deals with the design and implementation task of FTC for a battery assembly system, which is described using the discrete event max-plus algebra framework (Baccelli *et al.*, 1992; Butkovic, 2010). This strategy is further extended to deal with various uncertainties, which are inevitable in certain production systems. It should be pointed out that, to the authors' knowledge, there are no works present in the literature that deal with FTC for production systems described within max-plus algebra framework.

The battery assembly system being investigated is under construction at the RAFI company, which is one of the leading electronic manufacturing service providers in Germany. The paper considers a part of this system that contains two transportation robots, assembly stations, and input and output buffers. Such components are typical for manufacturing processes, which belong to the class of discrete-event systems (DESs) (Polak *et al.*, 2004). The DES is a discrete-state, event-driven system while its state evolution depends entirely on the occurrence of discrete events over time. Thus, in DESs, the state-space of a system is naturally described by a discrete set like $\{0, 1, 2, \ldots\}$ and the transitions are only observed at the discrete instants in time. There are many different modeling techniques for discrete-event systems, such as Petri nets, extended state machines, event-graphs, formal languages, generalized semi Markov processes, nonlinear programming, automata, computer simulation models and so on (see, e.g., the works of Sahner *et al.* (2012), Abrams *et al.* (1992), Hillion and Proth (1989), Yan *et al.* (2013) and the references therein). It should be underlined that models describing a DES are nonlinear in the conventional algebra. However, it is possible to define a class of discrete-event systems, mostly called max-plus linear discrete-event systems, in which there is a synchronization without concurrency or selection. This class of DES can be described by a model that is linear within the max-plus algebra (Baccelli *et al.*, 1992; Butkovic, 2010). This recommends its application (instead of the more traditional approaches (Witczak, 2007)) for the task being undertaken.

Thus, the contribution of the paper is the design and implementation of a robust FTC framework for a battery assembly station, which makes it possible to minimize energy consumption of autonomous robots driving the system while satisfying all production-process-related constraints. The proposed strategy has also an appealing property of being able to deal with faults regarding mobile robots, processing and transportation.

The paper is organized as follows. Section 2 introduces elementary definitions and concepts. The battery assembly system is carefully described in Section 3. Subsequently, Section 4 introduces the MPC algorithm along with its implementation issues. Section 5 presents the details of the FTC algorithm. First, it is shown how to design and implement FTC for mobile robot faults. The proposed approach is then extended to processing and transportation faults. Given a complete FTC structure, the design strategy is suitably extended to cope with the robustness problem which is outlined in Section 6. Section 7 presents the results regarding the abilities of the proposed approach and clearly exhibits its performance. Finally, the last section concludes the paper.

2. Preliminaries

The main objective of this section is to provide essential definitions and concepts that will be exploited in further deliberations.

Definition 1. A *fault* is an unpermitted deviation of at least one characteristic performance time of the system from the nominal condition.

Definition 2. A *failure* is a permanent interruption of the system ability to perform a required mission under specified operating conditions.

Note that Definition 1 is based on the classical fault definition provided in the well-known textbooks (see, e.g., Blanke *et al.*, 2006), where the part "characteristic time" is replaced by "characteristic property", which has more universal meaning. Thus, the provided definition can be perceived as a special case of the general one.

After giving elementary definitions, it is possible to explain the main mathematical concepts related to the max-plus formalism as well as the max-plus linear system framework.

2.1. Max-plus algebra and max-plus linear systems.

The (max,+) algebraic structure $(\mathbb{R}_{\max}, \oplus, \otimes)$ is defined as follows:

- $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, where $\mathbb{R}$ is the field of real numbers,
- $\forall a, b \in \mathbb{R}_{\max}, a \oplus b = \max(a, b)$,
- $\forall a, b \in \mathbb{R}_{\max}, a \otimes b = a + b$.

The operations $\oplus$ and $\otimes$ are called the max-plus algebraic addition and max-plus algebraic multiplication, respectively. Let $k \in \mathbb{N}$. Then the k-th max-plus algebraic power of $a \in \mathbb{R}$ is denoted by $a^{\otimes^k}$. For $k > 0$, let us define ε such that $\varepsilon^{\otimes^k} = \varepsilon$ and $\varepsilon^{\otimes^0} = 0$. The rules related to the order of evaluation of max-plus algebraic operators are the same as those of the conventional algebra. Thus, the max-plus algebraic power has the highest priority, while the max-plus algebraic multiplication has a higher priority than the max-plus algebraic addition. The main properties are

$$\begin{aligned} &\forall a \in \mathbb{R}_{\max} : a \oplus \varepsilon = a \text{ and } a \otimes \varepsilon = \varepsilon, \\ &\forall a \in \mathbb{R}_{\max} : a \otimes e = a, \end{aligned} \tag{1}$$

where $\varepsilon = -\infty$ and $e = 0$ are the neutral elements for the max-plus-algebraic addition and max-plus-algebraic multiplication operations, respectively.

For matrices $X, Y \in \mathbb{R}_{\max}^{m \times n}$ and $X \in \mathbb{R}_{\max}^{n \times p}$,

$$(X \oplus Y)_{ij} = x_{ij} \oplus y_{ij} = \max(x_{ij}, y_{ij}), \tag{2}$$

$$\begin{aligned} (X \otimes Z)_{ij} &= \bigoplus_{k=1}^{n} x_{ik} \otimes z_{kj} \\ &= \max_{k=1,\ldots,n} (x_{ik} + z_{kj}), \end{aligned} \tag{3}$$

for all i, j. The matrix E_n is an $n \times n$ max-plus algebraic identity matrix – $(E_n)_{ii} = 0$ and $(E_n)_{ij} = \varepsilon$ for $i \neq j$, $i, j = 1, \ldots, n$. Thus, the matrix power of $A \in \mathbb{R}_{\max}^{m \times n}$ is defined as follows:

$$A^{\otimes^0} = E_n, \quad A^{\otimes^k} = A \otimes A^{\otimes^{k-1}} \tag{4}$$

for $k = 1, 2, 3, \ldots$. Further definitions and details related to the max-plus algebra formalism are given by Baccelli *et al*. (1992) and Butkovic (2010).

If the max-plus algebra framework is provided, then it is possible to introduce discrete-event systems that can be described by a model of the following form:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{5}$$

$$y(k) = C \otimes x(k), \tag{6}$$

where the index k is the event counter, while

- $x(k) \in \mathbb{R}_{\max}^{n}$ represents the state typically containing the time instants at which the internal events occur for the k-th time,
- $u(k) \in \mathbb{R}_{\max}^{r}$ is the input vector containing the time instants at which the input events occur for the k-th time,
- $y(k) \in \mathbb{R}_{\max}^{m}$ are states for the output vector containing the time instants at which the output events occur for the k-th time,

and the system matrices are $A \in \mathbb{R}_{\max}^{n \times n}$, $B \in \mathbb{R}_{\max}^{n \times r}$, and $C \in \mathbb{R}_{\max}^{m \times n}$.

It should be also noted that the max plus framework being employed is fully time-oriented. Indeed, the parameters of matrices A, B, C denote specific time instants describing the system behaviour. Similarly, input $u(k)$, state $x(k)$ and output $y(k)$ also represent the time of the predefined operations.

2.2. Interval max-plus algebra. The objective of this section is to provide a novel methodology that can be employed to settle the robustness to parameter uncertainties of the matrices A, B and C associated with the system (5)–(6). This idea was initially employed to analyze uncertain production systems (Cechlárová, 2005) and will be here used for robust FTC purposes.

The (imax,+) algebraic structure $(\mathcal{I}(\mathbb{R}_{\max}),\oplus,\otimes)$ is defined as follows:

- $\mathcal{I}(\mathbb{R}_{\max})$ is a set of real compact intervals of the form $a = [\underline{a}, \bar{a}]$,
- $\forall a, b \in \mathcal{I}(\mathbb{R}_{\max}), a \oplus b = \max(\bar{a}, \bar{b})$,
- $\forall a, b \in \mathcal{I}(\mathbb{R}_{\max}), a \otimes b = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$.

Similarly as in the previous point, for matrices $X, Y \in \mathcal{I}(\mathbb{R}_{\max})^{m \times n}$ and $X \in \mathcal{I}(\mathbb{R}_{\max})^{n \times p}$,

$$(X \oplus Y)_{ij} = x_{ij} \oplus y_{ij} = \max(\bar{x}_{ij}, \bar{y}_{ij}), \tag{7}$$

$$\begin{aligned} (X \otimes Z)_{ij} &= \bigoplus_{k=1}^{n} x_{ik} \otimes z_{kj} \\ &\quad + \max_{k=1,\ldots,n} (\bar{x}_{ik} + \bar{z}_{kj}), \end{aligned} \tag{8}$$

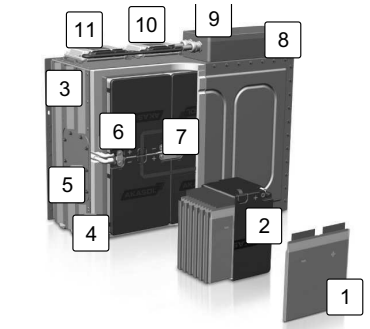
for all i, j. The linear system description (5)–(6) remains almost the same; the only modification is that $y(k) \in \mathcal{I}(\mathbb{R}_{\max})^{m}$. Since the general mathematical framework is provided, it is possible to introduce the battery assembly system for which a comprehensive design and performance evaluation study will be performed.

3. Battery assembly station

Currently, the RAFI company is assembling a low number of battery systems, which is realised by hand mostly. Further regulations and predicted numbers of high performance batteries will not allow this procedure in future (Nair and Garimella, 2010). Therefore, a flexible battery assembly system with autonomous robots will be introduced for high-volume serial production. This new production system is based on transport and manipulation robots, with additional hand assembly stations. RAFI's

goals are to set up a battery assembly system providing a maximal flexibility for upcoming variants of further battery system products as well as a maximum quality level and protection for products and staff (Chan, 2002).

At the first stage of expansion, only the transport robots are implemented. The actual product to assemble is a high performance battery system for domestic and private usage to buffer renewable energy sources and provide independent energy supply. An overview of the



1. Cell
2. AKAMODULE
3. High-strength battery tray
4. Thermal Isolation
5. Liquid Coolant Port
6. Coolant Interconnects
7. Electrical Interconnects
8. Main Connector Box
9. High Voltage Connection
10. BMS Master
11. Safety Control Unit

Fig. 1. Overview of the battery assembly system.

components of the battery system is shown in Fig. 1. Currently, two different main systems with two different voltage ratings are built. The two main formats are a rack-based and a box-based system. These systems are built by either a rack or box housing, two battery modules and a main battery management system. For the rack-based system, two voltage ratings (1000 V and 400 V) are available. This makes two different sets of cables and insulation material necessary for the assembly. The production system contains two assembly cycles (Fig. 3). The first cycle is operated with two types of transportation robots (transportation robots type 1 and 2). This task will be realised by KUKA omniRob (Fig. 2) and covers the assembly of the battery frames for the two different

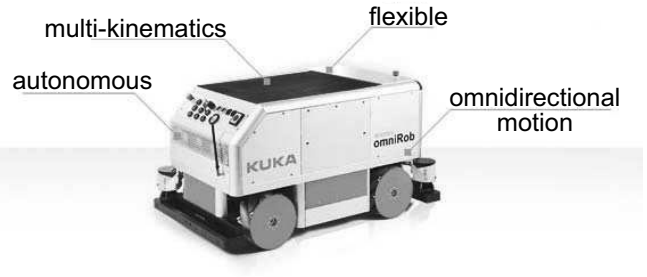


Fig. 2. KUKA omniRob: the transportation robot.

products. The second cycle is operated with three types of transportation robots (transportation robots type 3, 4 and 5) and covers the final assembly of the products.

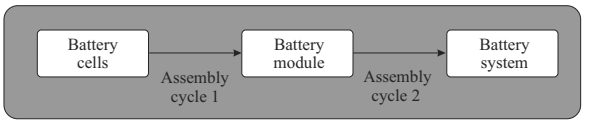


Fig. 3. Assembly process.

The sequence of the first production cycle (Fig. 4) starts with the robots in a starting setting. The robot of type 1 moves in the next step to the frame storage to pick up an empty battery module frame. In the next step the battery module controller is assembled into the frame. In the next step the robot moves with the battery frame aboard to the cell mounting station. The basic cells are fed by the robot of type 2 from the cell storage. It is assumed that this robot, in its own cycle, is able to get the number of basic cells needed to produce one battery. The next step is the assembly of an appropriate number of basic lithium-ion cell packages into the battery frames. The number of basic cells assembled into the module frame depends on the rated voltage of the battery module. For the 1000 V external voltage battery system, an internal 100 V battery module is used. 45 cells packages are mounted into the 100 V battery module. For the 400 V external voltage, a 24 V battery module is used, and therefore 11 cells are assembled into the 24 V battery module.

The final assembly of the battery module is moved to the relevant battery stores. Finally, the robot returns to the starting position.

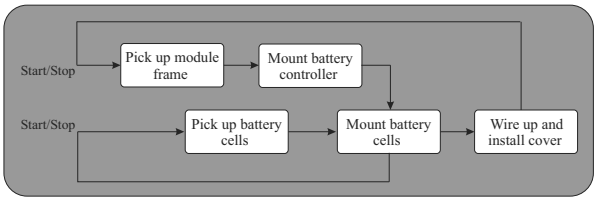


Fig. 4. Assembly process cycle 1.

The sequence of the second production cycle (Fig. 5) also starts with the robots in a starting position. Depending on the final product, the robot of type 3 moves either to the storage of the box housings or alternatively to a hand assembly station to pick up additional parts and wiring for the 1000 V rack version, and then to the rack housing storage to pick up the housings and bring them to the assembly station. In parallel, the robot of type 4 moves to the battery storage and picks up either 24V or 100V battery modules and brings them to the assembly station. The next step is the assembly of the battery controllers into the housings. Finally, the assembled product is moved

with a robot of type 5 to either the final rack or the box storage. From this point, the robot returns to the starting position. It should also be pointed out that the

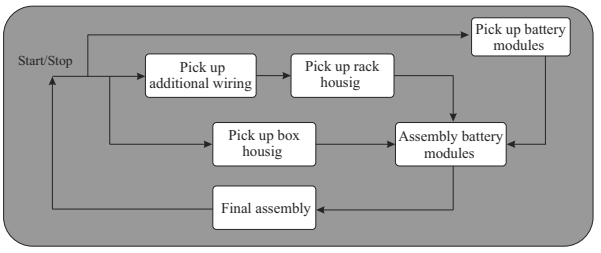


Fig. 5. Assembly process cycle 2.

storages for cell packages and frames can only be accessed by one robot at a time. Since the general description is provided, it is possible to use a max-plus framework to provide its mathematical model that will be used for further deliberations related to predictive control and its further extensions towards fault tolerance.

3.1. Max-plus linear model. Due to the space constraints, let us consider the first production cycle in the battery assembly system (see Fig. 4). In this part of the system, there are two types of transportation robots and five assembly processing units:

- P_1 represents the "Pick up module frame" unit,
- P_2 represents the "Mount battery controller" unit,
- P_3 is the "Pick up battery cells" unit,
- P_4 is the "Mount battery cells" unit,
- P_5 stands for the "Wire up and install cover" unit.

The processing times for $P_1, P_2, \ldots, P_5$ are d_1, d_2, d_3, d_4, d_5, respectively. The transportation times are as follows:

- t_1 is the transportation time from start to P_1,
- t_2 is the transportation time from start to P_3,
- t_3 is the transportation time from P_1 to P_2,
- t_4 is the transportation time from P_2 to P_4,
- t_5 is the transportation time from P_3 to P_4,
- t_6 is the transportation time from P_4 to P_5.

These times are clearly depicted in Fig. 6. It is defined that

- $u_i(k)$ denotes the time instant at which the i-th robot reaches the individual assembly station,
- $x_i(k)$ denotes the time instant at which the i-th processing unit starts performing a desired task,

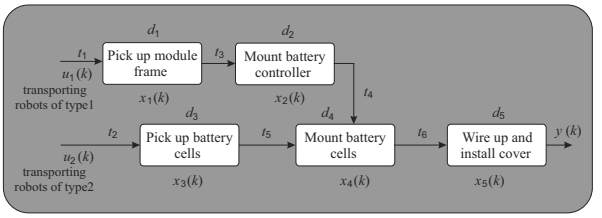


Fig. 6. Details of the assembly process cycle 1.

- $y(k)$ stands for the time of delivering the final product.

Note that a processing unit starts performing its operation on a new product (battery) if it has finished performing the previous one. If it is assumed that each operation starts as soon as all components of the assembly operation are available, then the system can be described by the following state space model:

$$
\begin{aligned}
x_1(k+1) &= \max(x_1(k)+d_1, u_1(k)+t_1),\\
x_2(k+1) &= \max(x_1(k+1)+d_1+t_3, x_2(k)+d_2)\\
&= \max(x_1(k)+2d_1+t_3, x_2(k)+d_2, u_1(k)\\
&\quad +t_1+d_1+t_3),\\
x_3(k+1) &= \max(x_3(k)+d_3, u_2(k)+t_2),\\
x_4(k+1) &= \max(x_2(k+1)+d_2+t_4, x_3(k+1)+d_3\\
&\quad +t_5 x_4(k)+d_4)\\
&= \max(x_1(k)+2d_1+t_3+d_2+t_4, x_2(k)\\
&\quad +2d_2+t_4, x_3(k)+2d_3+t_5, x_4(k)\\
&\quad +d_4, u_1(k)+t_1+d_1+t_3+d_2+t_4,\\
&\quad u_2(k)+t_2+d_3+t_5),\\
x_5(k+1) &= \max(x_4(k+1)+d_4+t_6, x_5(k)+d_5)\\
&= \max(x_1(k)+2d_1+t_3+d_2+t_4+d_4+t_6,\\
&\quad x_2(k)+2d_2+t_4+d_4+t_6, x_3(k)+2d_3\\
&\quad +t_5+d_4+t_6, x_4(k)+2d_4+t_6, x_5(k)\\
&\quad +d_5, u_1(k)+t_1+d_1+t_3+d_2+t_4\\
&\quad +d_4+t_6, u_2(k)+t_2\\
&\quad +d_3+t_5+d_4+t_6),\\
y(k) &= x_5(k)+d_5.
\end{aligned}
$$

The above equations can be described within the max-plus algebra framework (5)–(6):

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \tag{9}$$

$$y(k) = C \otimes x(k), \tag{10}$$

while a detailed description of the system matrices is given in Fig. 7. Since the analytical description of the system is given, it is possible to introduce the processing and transportation times. Table 1 describes these times in the form of exact values and intervals, respectively. The

$$A = \begin{bmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2d_1 + t_3 & d_2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & d_3 & \varepsilon & \varepsilon \\ 2d_1 + t_3 + d_2 + t_4 & 2d_2 + t_4 & 2d_3 + t_5 & d_4 & \varepsilon \\ 2d_1 + t_3 + d_2 + t_4 + d_4 + t_6 & 2d_2 + t_4 + d_4 + t_6 & 2d_3 + t_5 + d_4 + t_6 & 2d_4 + t_6 & d_5 \end{bmatrix},$$

$$B = \begin{bmatrix} t_1 & \varepsilon \\ t_1 + d_1 + t_3 & \varepsilon \\ \varepsilon & t_2 \\ t_1 + d_1 + t_3 + d_2 + t_4 & t_2 + d_3 + t_5 \\ t_1 + d_1 + t_3 + d_2 + t_4 + d_4 + t_6 & t_2 + d_3 + t_5 + d_4 + t_6 \end{bmatrix},$$

$$C = [\varepsilon,\ \varepsilon,\ \varepsilon,\ \varepsilon,\ d_5].$$

Fig. 7. Matrices of the battery assembly system.

Table 1. Nominal as well as interval processing and transportation times for the battery assembly system.

	Nominal time	Interval time [min]
d_1	6	[4,7]
d_2	3	[3,4]
d_3	5	[4,6]
d_4	8	[6,10]
d_5	3	[3,4]
t_1	4	[3,5]
t_2	4	[3,5]
t_3	2	[2,3]
t_4	4	[3,5]
t_5	4	[3,5]
t_6	4	[3,5]

first case corresponds to the perfect knowledge of these times and results in the system matrices of the form

$$A = \begin{bmatrix} 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 14 & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon \\ 21 & 10 & 14 & 8 & \varepsilon \\ 33 & 22 & 26 & 20 & 3 \end{bmatrix},$$

$$B = \begin{bmatrix} 4 & \varepsilon \\ 12 & \varepsilon \\ \varepsilon & 4 \\ 19 & 13 \\ 31 & 25 \end{bmatrix}, \tag{11}$$

$$C = [\varepsilon, \varepsilon, \varepsilon, \varepsilon, 3].$$

In the case of the second one, it is assumed that these times are known up to a given interval, which results in

$$A = \begin{bmatrix} [4,7] & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ [10,17] & [3,4] & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & [4,6] & \varepsilon & \varepsilon \\ [16,26] & [9,13] & [11,17] & [6,10] & \varepsilon \\ [25,41] & [18,28] & [20,32] & [15,25] & [3,4] \end{bmatrix},$$

$$B = \begin{bmatrix} [3,5] & \varepsilon \\ [9,15] & \varepsilon \\ \varepsilon & [3,5] \\ [15,24] & [10,16] \\ [24,39] & [19,31] \end{bmatrix}, \tag{12}$$

$$C = [\varepsilon, \varepsilon, \varepsilon, \varepsilon, [3,4]].$$

Note that, due to the properties of max-plus algebra, most of the constraints involved in the system (depicted in Fig. 6) are naturally incorporated in (9)–(10). However, specific performance constraints have to be introduced separately. This is the purpose of the next section.

3.2. Handling constraints. The system constraints are as follows:

- First of all, the designed system has to follow some predefined time trajectory that can be defined as scheduling constraints of the form

$$x_j(k) \le t_{\text{ref},j}(k), \quad j = 1, \ldots, n, \tag{13}$$

  where $t_{\text{ref},j}(k)$ is the upper bound of $x_j(k)$ at time k.

- The second constraint is related to the robot performance,

$$\underline{u}_i \le u_i(k) \le \bar{u}_i, \quad i = 1, \ldots, r. \tag{14}$$

  The lower bound $\underline{u}_i$ corresponds to the maximum speed of the robot. The upper bound $\bar{u}_i$ corresponds to the minimum speed of the robot. Crossing this limit means that the energy consumption of robot drives rises drastically.

- The last constraint is the change rate one,

$$u_j(k+1) - u_j(k) \ge z_j, \quad j = 1, \ldots, r, \tag{15}$$

  where $z_j > 0$ is the upper bound of the change rate.

Since the system is described within max-plus algebra along with suitable constraints, it is possible to develop a control strategy that will enable its optimal performance.

4. Constrained model predictive control

Irrespective of the system type (continuous or discrete), constraints and control quality measures are inevitable in modern industrial systems. As mentioned in the introductory part of this paper, MPC is a perfect candidate to settle this challenging problem. Indeed, one of the core advantages of MPC is its natural ability of handling constraints. The proposed framework is based on the general idea of MPC for max-plus linear systems described by De Schutter and Van Den Boom (2001). Note that, according to Definition 1, a violation of a scheduling constraint (13) means faulty behaviour of the system, which will be analyzed within the subsequent sections of this paper. Here, it is assumed that the system is fault-free, and hence all constraints being imposed (i.e., (13)–(15)) can be satisfied.

Thus, within the proposed framework, MPC, along with max-plus algebra, is to be used to minimize the robot's energy consumption. This can be perceived as a kind of economic MPC for which the energy consumption is the most important goal. Finally, the problem boils down to finding the input sequence $u(k), \ldots, u(k+N_p-1)$ that minimizes the cost function $J(u)$,

$$J(u) = -\sum_{j=0}^{N_p-1}\sum_{i=1}^{r} q_i u_i(k+j), \tag{16}$$

where $q_i > 0$, $i = 1, \ldots, m$, is a positive weighting constant corresponding to the relative importance of the energy consumption of the i-th robot, while N_p stands for the prediction horizon. The main advantage of (16) over the quadratic criteria employed in the case of continuous systems is that there is no need for using the relatively time consuming quadratic programming. Instead, taking into account the linear constraints (13)–(15), an efficient linear programming framework can be used.

The first task towards the computational framework is to eliminate direct influence of $x(k+1), \ldots, x(k+N_p-1)$ on the scheduling constraints (13). For this purpose, let

$$\tilde{x}(k+N_p-1) = M \otimes x(k) \oplus H \otimes \tilde{u}(k), \tag{17}$$

where

$$\tilde{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix},$$

$$\tilde{x}(k+N_p-1) = \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+N_p-1) \end{bmatrix}.$$

Using (5)–(6), it can be shown that

$$H = \begin{bmatrix} B & \varepsilon & \cdots & \varepsilon \\ A \otimes B & B & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ A^{\otimes k+N_p-2} \otimes B & A^{\otimes k+N_p-3} \otimes B & \cdots & B \end{bmatrix},$$

$$M = \begin{bmatrix} A \\ A^{\otimes 2} \\ \vdots \\ A^{\otimes k+N_p-1} \end{bmatrix}.$$

Thus, substituting (17) into the scheduling constraints (13) allows formulating a linear optimisation problem of the following form: Given an initial condition $x(k)$, obtain the optimal input sequence $\tilde{u}(k)^*$ by solving

$$\tilde{u}(k)^* = \arg\min_{\tilde{u}(k)} J(u) \tag{18}$$

under the constraints (13)–(15).

To summarize, the control algorithm is structured as Algorithm 1.

Algorithm 1. Max-plus MPC.

Step 0. Set $k = 0$.

Step 1. Measure the state $x(k)$ and obtain $\tilde{u}(k)^*$ by solving the constrained optimization problem (18).

Step 2. Use the first vector element of $\tilde{u}(k)^*$ (i.e., $u(k)^*$) and feed it into the system (5)–(6).

Step 3. Set $k = k+1$ and go to *Step 1*.

Since the max-plus MPC algorithm is provided, it is possible to implement it for the battery assembly system described by (9)–(10).

4.1. Implementation details. In order to guarantee full production effectiveness, the battery assembly system has to be connected tightly to the manufacturing execution system (MES), which itself is connected to the RAFI advanced planning and scheduling system (APS) Felios, where all dispositive factors, like raw materials and production capacity, are optimally allocated, while at the top level the APS is connected to the enterprise resource planning (ERP) system SAP. The RAFI APS is in place to fulfill RAFI's made-to-order manufacturing strategy, due to a certain risk in storing partly assembled or fully assembled battery systems (Vincent, 1999). The MES is an intermediate system that will take care of product definition and scheduling as well as resource management.

With detailed information on product configuration and all general production conditions, the MES provides current production state $x(k)$ to the MPC max-plus control system as well as scheduling constraints t_{ref} for

the entire prediction horizon N_p. Having this information, the MPC max-plus control framework calculates the input $u(k)^* = [u_1^*(k), u_2^*(k)]^T$, which is composed of the time instant at which the first and second robots reach the individual assembly station, respectively. This task is realised according to Algorithm 1.

An overview of the MPC max-plus control strategy for the battery assembly system is portrayed in Fig. 8. Finally, it should be noted that the MES also provides

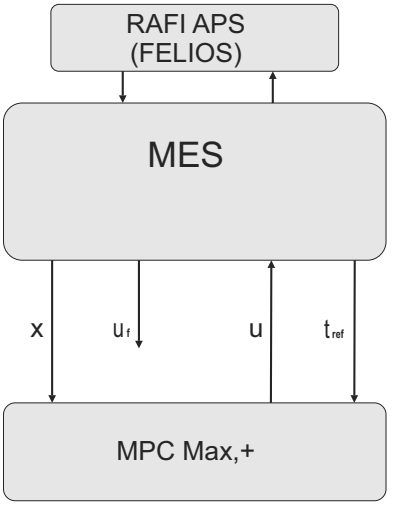


Fig. 8. Overview of MPC max-plus: MES configuration.

the actual time at which the robot reaches an individual assembly station that is denoted by $u_f(k)$. However, this information will be used in a subsequent part of this paper, which is devoted to FTC.

5. Fault-tolerant control of the battery assembly station

The main objective of this section is to provide tools that are to be useful while handling faults that can appear in the battery assembly station. These faults are divided into two groups:

- mobile robot faults,
- process faults.

A detailed discussion of these emerging issues is provided in the subsequent parts of this section.

5.1. Handling mobile robot faults. The objective of this point is to enhance the proposed MPC max-plus strategy with mobile robot fault-tolerance features. The causes of such faults may have different roots, which can be divided into

mechanical issues: power loss, tyre problems, drive problems, etc.,

infrastructure issues: bad surface, slow charging, obstacles, etc.

Figure 9 provides an outline of the FTC system regarding the above-mentioned faults.

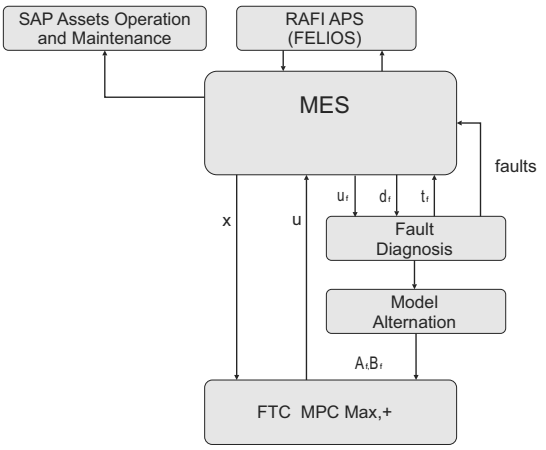


Fig. 9. Predictive FTC scheme.

As in the case of Algorithm 1, the MES provides the current production state $x(k)$ to the FTC system as well as scheduling constraints t_{ref} for the entire prediction horizon N_p. In contrast to the above case, the MES provides also the actual time at which the robots reaches an individual assembly station that is denoted by $u_f(k)$. Note that the faulty behaviour should be perceived as a delayed reaction of the robot compared to the calculated transportation time $u(i,k)^*$. Having this information, the FTC system decides about the faulty or fault-free status of the robots, which is realised by a simple residual-based decision threshold:

$$\text{if } s_i > \delta_i, \text{ then the } i\text{-th mobile robot is faulty,} \tag{19}$$

where the residual is

$$s_i = u_f(i,k) - u(i,k)^* \tag{20}$$

for all $i = 1, \ldots, r$ and $\delta_i > 0$ being a small positive constant that is robot-dependent and should be set by the designer. If a mobile robot fault is detected, then the matrix B should be replaced by B_f, which in the case of the i-th fault is defined as

$$b_{f,j,i} = b_{j,i} \otimes s_i, \quad j = 1, \ldots, n, \tag{21}$$

while the resulting system is

$$x(k+1) = A \otimes x(k) \oplus B_f \otimes u(k), \tag{22}$$

$$y(k) = C \otimes x(k). \tag{23}$$

This process is clearly illustrated in Fig. 9

Finally, it is evident that the mobile robot fault may have an influence on the scheduling constraints.

Indeed, the faulty behaviour may make the optimization problem (18) infeasible. Thus, it is proposed to relax the scheduling constraints as follows:

$$x_j(k) \leq t_{\text{ref},j}(k) + \alpha_j, \quad j = 1, \ldots, n, \tag{24}$$

where $\alpha_j \geq 0$, $j = 1, \ldots, n$, should be as small as possible in order to exhibit a minor divergence from the desired time schedule. In order to obtain the optimal values of α_j, a new cost function is proposed:

$$J(\alpha) = \sum_{i=1}^{n} \alpha_i, \tag{25}$$

and hence a new optimization framework can be described by

$$J(u, \alpha) = (1 - \beta)J(u) + \beta J(\alpha), \tag{26}$$

where $1 \leq \beta \leq 0$ is a constant set by the designer, which can be adjusted to reflect greater importance of either $J(u)$ or $J(\alpha)$, respectively. Thus, the optimization strategy is as follows: Given an initial condition $x(k)$, obtain the optimal input sequence $\tilde{u}(k)^*$ by solving

$$\tilde{u}(k)^* = \arg \min_{\tilde{u}(k), \alpha} J(u, \alpha) \tag{27}$$

for the faulty system (22)–(23) under the constraints (24), (14) and (15).

To summarize, the FTC strategy is detailed by Algorithm 2.

Algorithm 2. Mobile robot FTC.

Step 0. Set $k = 0$.

Step 1. Measure the state $x(k)$ and the actual input value $u_f(k)$, and then calculate the residual (20).

Step 2. If the fault test (19) indicates that there is no fault, then obtain $\tilde{u}(k)^*$ by solving the constrained optimization problem (18), otherwise obtain $\tilde{u}(k)^*$ by solving (27).

Step 3. Use the first vector element of $\tilde{u}(k)^*$ (i.e., $u(k)^*$) and feed it into the system (5)–(6).

Step 4. Set $k = k + 1$ and go to *Step 1*.

By applying it to Algorithm 2 for the battery assembly system, the inputs $u(k)^* = [u_1^*(k), u_2^*(k)]^T$ are calculated, which are composed of the time instant at which first and second robots reach the individual assembly station, respectively. Finally, it should be noted that the information about faults is fed into the RAFI SAP asset operation and maintenance system, which is a part of RAFI's preventive maintenance strategy (cf. Fig. 9). These data are further used in the RAFI condition monitoring module.

5.2. Handling production faults. It is obvious that mobile robot faults are not the only ones can appear in the system. Indeed, the production schedule can be significantly violated when the production and/or transportation times are not achieved. In particular, the fault diagnosis block is responsible for providing information about the processing and transportation time, whose actual values are fed to this block. The decision process is realised in the same way as the one for mobile robots, and, when the fault is indicated, then system matrices are suitably calculated. Finally, the new system has the following form:

$$x(k+1) = A_f \otimes x(k) \oplus B_f \otimes u(k), \tag{28}$$

$$y(k) = C_f \otimes x(k). \tag{29}$$

This implies that the FTC structure for a production fault is almost the same as that of Algorithm 2, and results in Algorithm 3.

Algorithm 3. Process FTC.

Step 0. Set $k = 0$.

Step 1. Measure the state $x(k)$ and the actual production and transportation times $\mathbb{P} = t_1, \ldots, t_{n_t}, d_1, \ldots, d_{n_d}$, and then calculate the residual:

$$s_i = p_i - p_{f,i}, \quad i = 1, \ldots, n_t + n_d, \tag{30}$$

where $p_i \in \mathbb{P}$ stands for the nominal production and/or transportation time.

Step 2. If the fault test

$$\text{if } s_i > \delta_i, \text{ then}$$
$$\text{the } i\text{-th system production component is faulty,} \tag{31}$$

indicates that there is no fault, then obtain $\tilde{u}(k)^*$ by solving the constrained optimization problem (18), otherwise obtain $\tilde{u}(k)^*$ by solving (27) with (28)–(29),

Step 3. Use the first vector element of $\tilde{u}(k)^*$ (i.e., $u(k)^*$) and feed it into the system.

Step 4. Set $k = k + 1$ and go to *Step 1*.

6. Towards robustness

The proposed FTC has to prevent the fault from causing a failure at the system level (Blanke *et al.*, 2006). It is evident that, in the case of a failure, constraints described in Section 3.2 will be violated, which will lead to the infeasibility of the proposed approach. Then the system will stop or run out of parts depending on where the failure occurs. The algorithm, as it is now, will not be able to organize detours or the restructuring of the assembly setting (using, for example, only one robot). This could also be an interesting issue for further research.

Another factor which may impair the quality of the proposed FTC is related to the robustness issue, which is usually perceived as the sensitivity of a given approach to parametric inaccuracies, a model and system structure mismatch, or external disturbances and noise. It is evident that in the proposed framework the structure of the model and system is exactly the same (in the discrete event sense). Thus, the model-system discrepancy can be only related to the uncertainty of the model matrices A, B and C. Indeed, the application of precise matrices may be inappropriate for the modeled situation, as the variability of the processing and transportation times, as well as measurement uncertainty, implies that the computations performed with the exact system description given by (11) do not correspond to the real behaviour of the system. Thus, an alternative description is proposed that outerbounds the above-mentioned uncertainties with suitable intervals (cf. Table 1), hence making the FTC algorithm robust to them. This leads directly to the system structure described by (12). However, this new system description requires a novel formulation of the max-plus paradigm, which is provided in Section 2.2 and called the imax-plus framework. If the model structure and computational tools are given, then the robust fault-tolerant control (RFTC) algorithm can be easily formulated. The structure of the algorithm is exactly the same as that of Algorithms 2 or 3 but instead of the max-plus algebra paradigm, an imax-plus one is employed.

Finally, it should be pointed out that the framework does not consider external disturbances and noise as it is usually the case in the continuous-time framework. These factors are mostly introduced in an additive manner both in the state and output equations. However, this can be a subject of further research.

7. Results

The main objective of this section is to validate the reliability of the proposed approaches applied to the battery assembly system. Thus, the following case studies were performed:

- application of Algorithm 1 for the nominal system,
- application of Algorithm 2 for the nominal system with an over-demanded schedule,
- application of Algorithm 2 for the mobile robot fault,
- application of Algorithm 3 for the production fault.

It should be pointed out that in all but the first two experiments a robust system description was used, which is described with (12). This implies that imax-plus algebra is used instead of the usual max-plus algebra framework. Thus, the objective of the subsequent part of this section is to provide a comprehensive description regarding the above-defined case studies.

7.1. MPC for the nominal system. Before applying Algorithm 1 to the battery assembly system, it is necessary to provide all associated constraints. Let us start with the scheduling constraints, which are defined with

$$\begin{aligned} t_{\text{ref}}(0) &= [4, 12, 4, 19, 31]^T, \\ t_{\text{ref}}(1) &= [14, 22, 14, 29, 41]^T, \\ t_{\text{ref}}(2) &= [24, 32, 24, 39, 51]^T, \\ &\vdots \end{aligned} \tag{32}$$

As already mentioned, Algorithm 1 will be illustrated with the system description (11). The robot performance constraint is neglected while the rate of the change constraint (15) is defined with $z_1 = 5$ and $z_1 = 6$. Moreover, the prediction horizon was set to $N_p = 4$ along with $q_1 = q_2 = 1$ shaping the cost function (16). As a result, the following control sequence, i.e., optimal values of $u(k)$, was obtained:

$$\begin{bmatrix}10\\10\end{bmatrix}, \begin{bmatrix}20\\20\end{bmatrix}, \begin{bmatrix}30\\30\end{bmatrix}, \begin{bmatrix}40\\40\end{bmatrix}, \begin{bmatrix}40\\40\end{bmatrix}, \begin{bmatrix}50\\50\end{bmatrix}, \begin{bmatrix}60\\60\end{bmatrix}, \tag{33}$$

which guarantees that all constraints are satisfied with the total cost $J(u) = -1100$. Note that such a control strategy can be intuitively derived from Fig. 6, which clearly exhibits the correctness of Algorithm 1.

7.2. FTC for the nominal system with an over-demanded schedule. The objective of this section is to show the performance of Algorithm 2 for the fault-free case but over-demanded schedule. This means that Algorithm 1 cannot be used for solving such a problem because of the infeasibility of the scheduling constraints, which are defined with

$$\begin{aligned} t_{\text{ref}}(0) &= [4, 12, 4, 19, 31]^T, \\ t_{\text{ref}}(1) &= [11, 19, 11, 26, 38]^T, \\ t_{\text{ref}}(2) &= [18, 26, 18, 33, 45]^T, \\ &\vdots \end{aligned} \tag{34}$$

The settings of the algorithm are the same as those in the preceding section, and additionally $\beta = 0.8$ in (26). Figure 10 portrays the difference between the actual state of the system and the schedule (34), while the associated control strategy is given in Fig. 11.

As can be observed, there is an increasing delay with respect to the schedule but the algorithm minimizes it with respect to (26). Indeed, since the schedule is over-demanded, it is vain to expect that the control algorithm will compensate its effect. The only solution is to minimize its effect as far as possible, which can be realized with Algorithm 2.

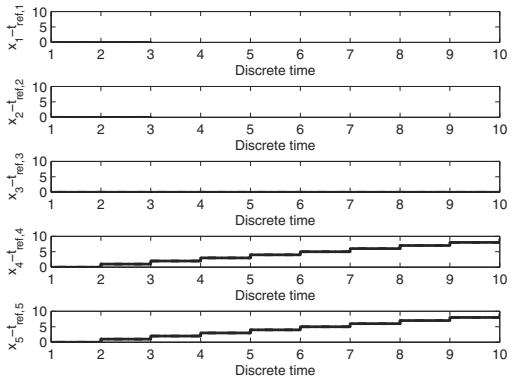


Fig. 10. Difference between the actual state and the reference trajectory for an over-demanded schedule.

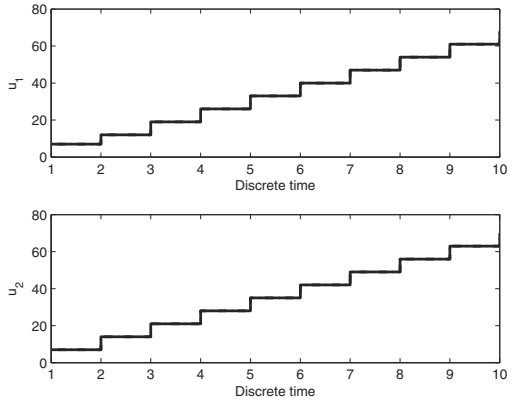


Fig. 11. Control strategy for an over-demanded schedule.

7.3. FTC for the mobile robot fault. The mobile robot fault is perceived as a four-minute delay of the first robot that begins from the fourth iteration. This fact is determined by the fault diagnosis system depicted in Fig. 9. Subsequently, the system matrices are updated in order to cope with this new situation. It should be also pointed out that the constraints are identical as those in Section 7.1. As a result of applying Algorithm 3, Fig. 12 presents the difference between the actual state and the upper bound of the scheduling constraints (32) while Fig. 13 presents the associated control strategies. It can be observed that after fault occurrence MPC without fault tolerance yields a permanent delay while the FTC strategy brings the system to a zero difference between the actual state and the reference trajectory. Finally, Fig. 14 portrays the final product outlet, which clearly exhibits the performance of the proposed approach.

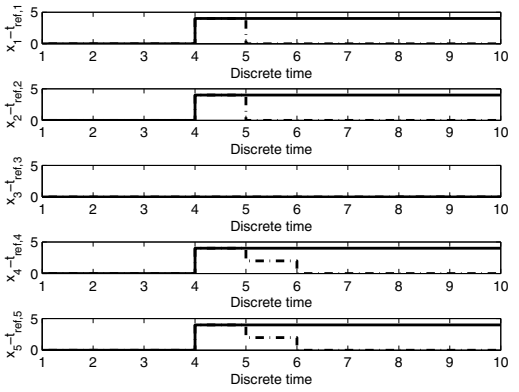


Fig. 12. Mobile robot fault: difference between the actual state and the reference trajectory with FTC (dashed line) and without it.

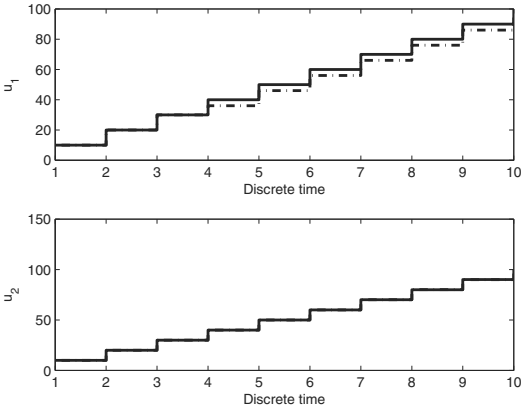


Fig. 13. Mobile robot fault: control strategy with FTC (dashed line) and without it.

7.4. FTC for production faults. This scenario concerns production faults. In particular, the fault is defined as a four-minute delay related to "Mount Battery Controllers". This means that the time d_2 is increased by 4 minutes. This fact is determined by the fault diagnosis system depicted in Fig. 9. Subsequently, the system matrices are updated in order to cope with this new situation. It should be also pointed out that the constraints are identical as those in Section 7.1.

As a result, Fig. 15 presents the difference between the actual state and the upper bound of the scheduling constraints (32) while Fig. 16 portrays the associated control strategies. It can be observed that after the fault occurrence MPC without fault tolerance yields a permanent delay while the FTC strategy brings the system to the zero difference between the actual state and the

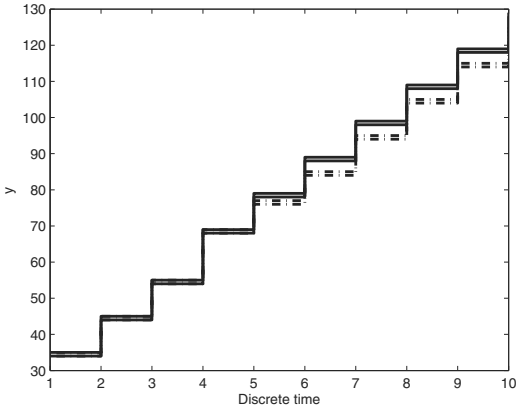


Fig. 14. Mobile robot fault: final bounds of the product outlet with FTC (dashed line) and without it.

reference trajectory. Finally, Fig. 17 portrays the final product outlet, which clearly exhibits the performance of the proposed approach.

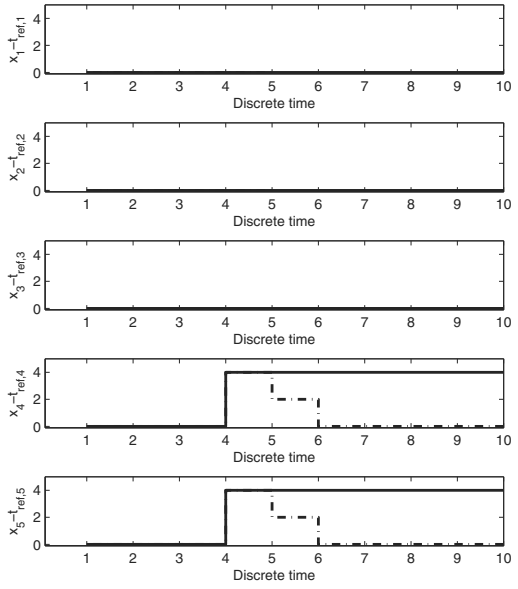


Fig. 15. Production fault: difference between actual state and the reference trajectory with FTC (dashed line) and without it.

8. Conclusions

The main objective of this paper was to propose a unified FTC MPC design procedure for a battery assembly

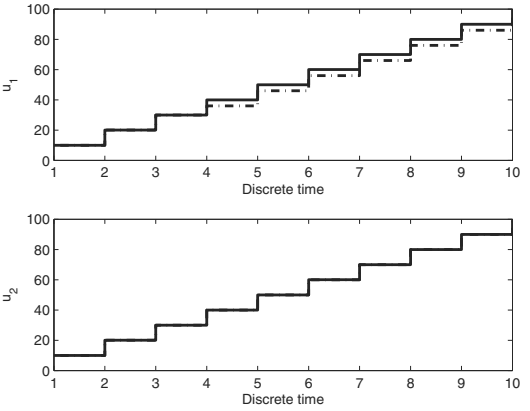


Fig. 16. Production fault: control strategy with FTC (dashed line) and without it.

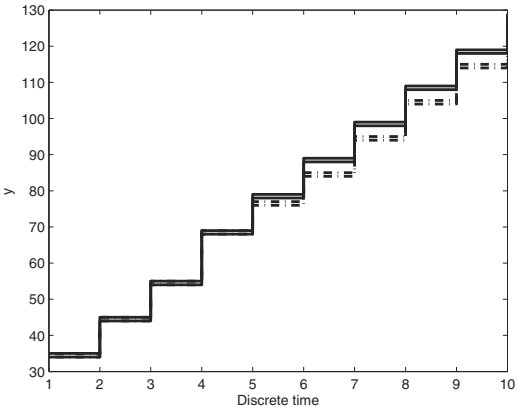


Fig. 17. Production fault: final bounds of the product outlet with FTC (dashed line) and without it.

system located at the RAFI company, allowing high production flexibility under real production conditions. In particular, one of the objectives was to describe the system within the max-plus algebra framework along with suitable constraints inevitably present in all real systems. It should be pointed out that max-plus algebra can be used to model and analyse the production system within the linear framework; e.g., the well-known Petri nets can be employed to simulate merely the system behaviour. The major advantage of the proposed approach is the avoidance of the non-linear optimization problem, which is the main drawback of the classical algebra framework applied to such tasks.

Apart from a suitable modeling framework, the FTC-MPC-based control procedure was provided. Indeed, a suitable control criterion and constraints were given. The optimization criteria take into account

robot energy consumption only, while all productivity demands are incorporated within the constraints. The main advantage, compared with the classical framework, is that the cost function is linear but not quadratic. This significantly reduces the computational burden, which makes it possible to apply the proposed approach for a large scale system. Due to the lack of space, the presented experimental results concern a selected part of the entire system but they clearly confirm its high performance. This approach allows the RAFI company to compete as an European manufacturer in the global market of renewable energy storages and to keep production labour at local production sites. Finally, it should be pointed out that the proposed strategy assumes that the whole state vector measurement is available. This condition can be relaxed by applying appropriate state observers. However, this will be a subject of future research.

Acknowledgment

The work was supported by the National Science Centre of Poland under a grant no. 2013/11/B/ST7/01110 for the years 2014–2017.

References

Abrams, M., Doraswamy, N. and Mathur, A. (1992). Chitra: Visual analysis of parallel and distributed programs in the time, event, and frequency domains, *IEEE Transactions on Parallel and Distributed Systems* **3**(6): 672–685.

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

Blanke, M., Schröder, J., Kinnaert, M., Lunze, J. and Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*, Springer, Berlin.

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

Camacho, E.F. and Bordons, C.A. (1997). *Model Predictive Control in the Process Industry*, Springer-Verlag New York, Inc., New York, NY.

Cechlárová, K. (2005). Eigenvectors of interval matrices over max-plus algebra, *Discrete Applied Mathematics* **150**(1): 2–15.

Chan, C. (2002). The state of the art of electric and hybrid vehicles, *Proceedings of the IEEE* **90**(2): 247–275.

Chen, W., Khan, A.Q., Abid, M. and Ding, S.X. (2011). Integrated design of observer-based fault detection for a class of uncertain non linear systems, *International Journal of Applied Mathematics and Computer Science* **21**(3): 423–430, DOI: 10.2478/v10006-011-0031-0.

De Schutter, B. and Van Den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems, *Automatica* **37**(7): 1049–1056.

Gunasekaran, A. (1999). Agile manufacturing: A framework for research and development, *International Journal of Production Economics* **62**(1): 87–105.

Hillion, H.P. and Proth, J.-M. (1989). Performance evaluation of job-shop systems using timed event-graphs, *IEEE Transactions on Automatic Control* **34**(1): 3–9.

Korbicz, J., Kościelny, J., Kowalczuk, Z. and Cholewa, W. (Eds.) (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*, Springer-Verlag, Berlin.

Li, H., Zhao, Q. and Yang, Z. (2007). Reliability modeling of fault tolerant control systems, *International Journal of Applied Mathematics and Computer Science* **17**(4): 491–504, DOI: 10.2478/v10006-007-0041-0.

Mrugalski, M. (2013). An unscented Kalman filter in designing dynamic GMDH neural networks for robust fault detection, *International Journal of Applied Mathematics and Computer Science* **23**(1): 157–169, DOI: 10.2478/amcs-2013-0013.

Nair, N.-K.C. and Garimella, N. (2010). Battery energy storage systems: Assessment for small-scale renewable energy integration, *Energy and Buildings* **42**(11): 2124–2130.

Polak, M., Majdzik, P., Banaszak, Z. and Wójcik, R. (2004). The performance evaluation tool for automated prototyping of concurrent cyclic processes, *Fundamenta Informaticae* **60**(1): 269–289.

Prodan, I., Olaru, S., Stoica, C. and Niculescu, S.-I. (2013). Predictive control for trajectory tracking and decentralized navigation of multi-agent formations, *International Journal of Applied Mathematics and Computer Science* **23**(1): 91–102, DOI: 10.2478/amcs-2013-0008.

Rossiter, J. (2013). *Model-based Predictive Control: A Practical Approach*, CRC Press, Boca Raton, FL.

Sahner, R., Trivedi, K. and Puliafito, A. (2012). *Performance and Reliability Analysis of Computer Systems: An Example-based Approach using the SHARPE Software Package*, Springer Publishing Company, Inc., New York, NY.

Vincent, C. (1999). Lithium batteries, *IEE Review* **45**(2): 65–68.

Witczak, M. (2007). *Modelling and Estimation Strategies for Fault Diagnosis of Non-linear Systems*, Springer-Verlag, Berlin.

Witczak, M. (2014). *Fault Diagnosis and Fault-tolerant Control Strategies for Non-linear Systems*, Lecture Notes in Electrical Engineering, Vol. 266, Springer International Publishing, Heidelberg.

Yan, F., Dridi, M. and El Moudni, A. (2013). An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach, *International Journal of Applied Mathematics and Computer Science* **23**(1): 183–200, DOI: 10.2478/amcs-2013-0015.

Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems, *Annual Reviews in Control* **32**(2): 229–252.

Lothar Seybold was born in Germany in 1970, received the Dipl-Ing. degree in electronic engineering from Hochschule Ravensburg–Weingarten (Germany) and the M.B.A. degree in business integration from the University of Wurzburg (Germany) in 1998 and 2003, respectively. In 2015 he received a Ph.D. degree in control engineering and robotics from the University of Zielona Góra (Poland). Lothar Seybold has worked as a design engineer and a consultant for several international companies like ABB AG (Switzerland) and Integrated Systems AG (Germany). Since 2005 he has been in charge of innovation and R&D activities for RAFI GmbH & Co. KG (Germany). His current research interests include industrial controls and automation, artificial intelligence, fault detection and isolation (FDI) and fault-tolerant control (FTC). Lothar Seybold has published more than 20 papers in international journals and conference proceedings.

Marcin Witczak was born in Poland in 1973, received the M.Sc. degree in electrical engineering from the University of Zielona Góra (Poland), the Ph.D. degree in automatic control and robotics from the Wrocław University of Technology (Poland), and the D.Sc. degree in electrical engineering from the University of Zielona Góra, in 1998, 2002 and 2007, respectively. In 2015 he received a full professorial title. Marcin Witczak has been an associate professor of automatic control and robotics at the Institute of Control and Computation Engineering, University of Zielona Góra, Poland, since 2008. His current research interests include computational intelligence, fault detection and isolation (FDI), fault-tolerant control (FTC), as well as experimental design and control theory. Marcin Witczak has published more than 140 papers in international journals and conference proceedings. He is an author of 4 monographs and 19 book chapters.

Paweł Majdzik was born in Poland in 1967, received the M.Sc. degree in electrical engineering from the Wrocław University of Technology (Poland) and the Ph.D. degree from the Poznań University of Technology (Poland), in 1992 and 1998, respectively. Paweł Majdzik 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 as well as modeling and control of DESs, and fault-tolerant control (FTC). Paweł Majdzik has published more than 36 papers in international journals and conference proceedings. He is an author of 2 monographs and 3 book chapters.

Ralf Stetter was born in Germany in 1970, received the M.Sc. degree in mechanical engineering from the Munich University of Technology (Germany) and the Ph.D. degree in product development from the same university in 1996 and 2000, respectively. Ralf Stetter has worked at Audi AG, Ingolstadt (Germany), as a team coordinator in the Interior Development Department. He has been a professor of design and development in automotive technology, University of Ravensburg–Weingarten (Germany), since 2004. Since 2006 he has also been a project leader at the Steinbeisni Transfer Center for Automotive Systems, concentrating on product development, testing and consulting for mainly automotive companies. His current research interests include mechatronic design, application of agent systems, fault detection and isolation (FDI), and fault-tolerant control (FTC). Ralf Stetter has published more than 130 papers in international journals and conference proceedings.

Received: 22 December 2014
Revised: 25 April 2015
Re-revised: 2 July 2015