Technical report bds:01-13

Properties of MPC for max-plus-linear systems*

T. van den Boom and B. De Schutter

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
Current URL: https://www.dcsc.tudelft.nl

---

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/01_13.html

Properties of MPC for max-plus-linear systems

Ton van den Boom and Bart De Schutter
Control Systems Engineering, Faculty of Information Technology and Systems,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
phone: +31-15-278.40.52, fax: +31-15-278.66.79,
email: {t.j.j.vandenboom,b.deschutter}@its.tudelft.nl

Abstract

Model predictive control (MPC) is a very popular controller design method in the process industry. One of the main advantages of MPC is that it can handle constraints on the inputs and outputs. Usually MPC uses linear discrete-time models. Recently we have extended this framework to max-plus-linear discrete event systems. In this paper we further explore this topic. More specifically, we focus on implementation and timing aspects, closed-loop behavior and tuning rules for model predictive control of max-plus-linear (MPL) systems.

Keywords: model predictive control, discrete event systems, max-plus algebra

1 Introduction

Model predictive control (MPC) [13, 18, 23] was pioneered simultaneously by Richalet *et al.* [22], and Cutler and Ramaker [8]. Since then, MPC has probably become the most applied advanced control technique in the process industry and many papers report successful applications [21]. MPC provides many attractive features. It can handle constraints in a systematic way and it can keep the system behavior as close as possible to the constraints without violating them. MPC is applicable to multi-variable systems and is capable of tracking pre-scheduled reference signals, using the concept of making predictions based on a model of the process. Finally, it is an easy-to-tune method. Basically three tuning parameters have to be chosen and adequate tuning rules are available [18, 23].

Conventional MPC uses (nonlinear) discrete-time models. The evolution of such systems is time-driven, i.e., in general the state of the system changes every sample step. In this paper we consider MPC for discrete event systems. In contrast to a discrete-time system, the evolution of a discrete event system is event-driven, i.e., the state of the system changes due to the occurrence of events. The class of discrete event systems essentially consists of man-made systems that contain a finite number of resources (such as machines, communications channels, or processors) that are shared by several users (such as product types, information packets, or jobs) all of which contribute to the achievement of some common goal (the assembly of products, the end-to-end transmission of a set of information packets, or a parallel computation) [1]. Typical examples of discrete event systems are flexible manufacturing systems, telecommunication networks, parallel processing systems, traffic control systems and logistic systems. Some examples of events for these systems are the arrival of a part at a buffer, a machine that has finished processing a part, a message that arrives at a router, a processor

that has finished a job, and a traffic signal that switches from red to green. Discrete event systems with synchronization but no concurrency can be described by models that are "linear" in the max-plus algebra [1, 7], and are called max-plus-linear systems. Typical examples of max-plus-linear discrete event systems are serial production lines, queuing systems, railway networks, and parallel computing systems [1].

In [11] we have extended the MPC framework to max-plus-linear systems and focused on efficient solution techniques for a single step of the max-plus-linear MPC algorithm. In this paper we investigate properties and implementation aspects of max-plus-linear MPC. More specifically, we first consider timing issues and the closed-loop behavior of the system and its MPC controller, i.e., we now look at the influence of applying MPC during the entire evolution of the system. In MPC for conventional linear discrete-time systems there exist rules of thumb for determining appropriate values for the MPC tuning parameters. In this paper we will show by several examples that these rules also more or less apply to MPC for max-plus-linear systems, with some minor but important changes.

The paper is organized as follows. In Section 2 we introduce max-plus-linear discrete event systems and we briefly recapitulate the MPC methodology for max-plus-linear systems. Section 3 discusses the timing aspects of the controller. In Section 4 we consider the closed-loop properties of max-plus-linear MPC. Next, we discuss the tuning of the parameters in MPC for max-plus-linear systems. We conclude with some illustrative examples.

2   The MPC problem for max-plus-linear systems

In [1, 6, 7] it has been shown that discrete event systems with only synchronization and no concurrency can be modeled by a max-plus-algebraic model of the following form:

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

$$y(k) = C \otimes x(k) \tag{2}$$

where k is the event counter, $x(k) \in \mathbb{R}_\varepsilon^{n \times 1}$ represents the state, $u(k) \in \mathbb{R}_\varepsilon^{m \times 1}$ is the input, $y(k) \in \mathbb{R}_\varepsilon^{l \times 1}$ is the output, with $\mathbb{R}_\varepsilon \overset{\text{def}}{=} \mathbb{R} \cup \{-\infty\}$, and where $A \in \mathbb{R}_\varepsilon^{n \times n}$, $B \in \mathbb{R}_\varepsilon^{n \times m}$, and $C \in \mathbb{R}_\varepsilon^{l \times n}$ are the system matrices. The operations $\oplus$ and $\otimes$ denote max-plus-algebraic addition and max-plus-algebraic multiplication respectively[1]:

$$x \oplus y = \max(x, y) \quad \text{and} \quad x \otimes y = x + y$$

for $x, y \in \mathbb{R}_\varepsilon$. For matrices $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ and $C \in \mathbb{R}_\varepsilon^{n \times p}$ we can extend the definition as follows:

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$$

$$(A \otimes C)_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_k (a_{ik} + c_{kj})$$

for all i, j. Define $\varepsilon = -\infty$ and the matrix $\mathcal{E}_{m \times n}$ as the $m \times n$ max-plus-algebraic zero matrix: $(\mathcal{E}_{m \times n})_{ij} = \varepsilon$ for all i, j. The matrix E_n is the $n \times n$ max-plus-algebraic identity matrix:

---

[1]The operations $\oplus$ and $\otimes$ are called the max-plus-algebraic addition and max-plus-algebraic multiplication respectively since many properties and concepts from linear algebra can be translated to the max-plus algebra by replacing $+$ by $\oplus$ and $\times$ by $\otimes$ (see [1, 7]).

$(E_n)_{ii} = 0$ for all i and $(E_n)_{ij} = \varepsilon$ for all i, j with $i \neq j$. The max-plus-algebraic matrix power of $A \in \mathbb{R}_\varepsilon^{n \times n}$ is defined as follows: $A^{\otimes^0} = E_n$ and $A^{\otimes^k} = A \otimes A^{\otimes^{k-1}}$ for $k = 1, 2, \ldots$

Systems that can be described by a model of the form (1)–(2) are called max-plus-linear (MPL) systems. In (1)–(2) the index k is the event counter. For a manufacturing system, $u(k)$ would typically represent the time instants at which raw material is fed to the system for the $(k+1)$th time, $x(k)$ the time instants at which the machines start processing the kth batch of intermediate products, and $y(k)$ the time instants at which the kth batch of finished products leaves the system. In [11] we have shown that prediction of future values of $y(k)$ for the system (1)–(2) can be done by successive substitution, leading to the expression

$$\tilde{y}(k) = \tilde{C} \otimes x(k) \ \oplus \ \tilde{D} \otimes \tilde{u}(k)$$

where $\tilde{C}$ and $\tilde{D}$ are given by

$$\tilde{C} = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes^2} \\ \vdots \\ C \otimes A^{\otimes^{N_{\mathrm{p}}}} \end{bmatrix}, \quad \tilde{D} = \begin{bmatrix} C \otimes B & \varepsilon & \ldots & \varepsilon \\ C \otimes A \otimes B & C \otimes B & \ldots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes^{N_{\mathrm{p}}-1}} \otimes B & C \otimes A^{\otimes^{N_{\mathrm{p}}-2}} \otimes B & \ldots & C \otimes B \end{bmatrix},$$

and $\tilde{u}(k)$, $\tilde{y}(k)$ are defined as:

$$\tilde{y}(k) = \begin{bmatrix} \hat{y}(k{+}1|k) \\ \hat{y}(k{+}2|k) \\ \vdots \\ \hat{y}(k{+}N_{\mathrm{p}}|k) \end{bmatrix}, \quad \tilde{u}(k) = \begin{bmatrix} u(k) \\ u(k{+}1) \\ \vdots \\ u(k{+}N_{\mathrm{p}}{-}1) \end{bmatrix}$$

where $\hat{y}(k{+}j|k)$ denotes the prediction of $y(k{+}j)$ based on knowledge at event step k and N_{p} is the prediction horizon.

The MPC problem for MPL systems is formulated as follows [11]:

$$\min_{\tilde{u}(k), \tilde{y}(k)} J(\tilde{u}(k), \tilde{y}(k)) = \min_{\tilde{u}(k), \tilde{y}(k)} J_{\mathrm{out}}(\tilde{y}(k)) + \lambda J_{\mathrm{in}}(\tilde{u}(k)) \tag{3}$$

subject to

$$\tilde{y}(k) = \tilde{C} \otimes x(k) \ \oplus \ \tilde{D} \otimes \tilde{u}(k) \tag{4}$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) \tag{5}$$

$$\Delta u(k + j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_{\mathrm{p}}{-}1 \tag{6}$$

$$\Delta^2 u(k + j) = 0 \quad \text{for } j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}}{-}1, \tag{7}$$

where $\Delta u(k) = u(k) - u(k{-}1)$ and $\Delta^2 u(k) = \Delta u(k) - \Delta u(k{-}1) = u(k) - 2u(k{-}1) + u(k{-}2)$, J_{out} is the output cost criterion and J_{in} the input cost criterion. Equation (5) reflects constraints on the input and output event separation times or maximum due dates for the output events, equation (6) guarantees a non-decreasing input signal, and equation (7) is due to the control horizon N_{c} (see Section 5). The above problem will be called the MPL-MPC problem.

In the remainder of this paper we consider the following output and input objective functions[2]:

$$J_{\text{out}}(k) = \sum_{i=1}^{mN_{\text{p}}} \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)$$

$$J_{\text{in}}(k) = -\sum_{i=1}^{lN_{\text{p}}} \tilde{u}_i(k).$$

These criteria can be interpreted as follows: J_{out} measures the tracking error or tardiness of the system, which is equal to the delay between the output dates $\tilde{y}_i(k)$ and due dates $\tilde{r}_i(k)$ if $\tilde{y}_i(k) - \tilde{r}_i(k) > 0$, and zero otherwise; J_{in} maximizes the input dates $\tilde{u}_i(k)$[3].

If we replace (4) by the following inequality:

$$\tilde{y}(k) \geqslant \tilde{C} \otimes x(k) \ \oplus \ \tilde{D} \otimes \tilde{u}(k) \tag{8}$$

we obtain the relaxed MPL-MPC problem, which is defined by the optimization of (3) subject to (8), (5), (6) and (7).

Theorem 2.1 *Let the mapping $\tilde{y} \to F(k)\tilde{y}$ be a monotonically non-decreasing function of $\tilde{y}$. Let $(\tilde{u}^*, \tilde{y}^*)$ be an optimal solution of the relaxed MPL-MPC problem. If we define $\tilde{y}^\sharp = \tilde{C} \otimes x(k) \oplus \tilde{D} \otimes \tilde{u}^*$ then $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MPL-MPC problem.*

Proof: See [11]. □

So if the linear constraints are monotonically non-decreasing as a function of $\tilde{y}(k)$, the MPL-MPC problem can be recast as a convex problem[4]. Moreover, by introducing some additional dummy variables the problem can even be reduced to a linear programming problem (see [11]).

MPC uses a receding horizon strategy. So after computation of the optimal control sequence $u(k), \ldots, u(k+N_{\text{c}}-1)$, only the first control sample $u(k)$ will be implemented, subsequently the horizon is shifted and the model and the initial state estimate can be updated if new measurements are available, then the new MPC problem is solved, etc.

3 Timing

MPC for MPL systems is different from conventional MPC in the sense that the event counter k is not directly related to a specific time. In the previous section and in [11] we have assumed that $x(k)$ is available when we optimize over the future control sequence. However, in practice not all components of $x(k)$ are known at the same time instant (Recall that $x(k)$ contains the time instants at which the internal activities or processes of the system start for the kth time). Therefore, we will now present a method to address the timing issues of MPL-MPC.

---

[2]Other cost criteria are also possible — see [10, 11].

[3]For a manufacturing system, this corresponds to a scheme in which raw material is fed to the system as late as possible. Note that this implies that the internal buffer levels are kept as low as possible.

[4]Note that (8) can be rewritten in conventional notation as a system of linear inequalities $\tilde{y}_i(k) \geqslant \tilde{c}_{ir} + x_r(k)$, $\tilde{y}(k) \geqslant \tilde{d}_{is} + \tilde{u}_s(k)$ for $i = 1, \ldots, l$, $r = 1, \ldots, n$ and $s = 1, \ldots, m$.

We consider the case of full state information[5]. Let $[x_{\text{true}}(k)]_i$ be the measured (true) occurrence time of the kth occurrence of internal event i, and let $[x_{\text{est}}(k, t)]_i$ be an estimation of the kth occurrence time of this event at time t. The estimation can be done using the following procedure: Let $\ell(t)$ be the smallest integer such that $[x_{\text{true}}(k - \ell(t))]_i < t$ for all $i = 1, \ldots, n$. Hence, $[x_{\text{true}}(k - \ell(t))]_i$ is completely known at time t. If we define $x_{\text{est}}(k - \ell(t), t) = x_{\text{true}}(k - \ell(t))$, we can reconstruct the unknown state components using the recursion

$$x_{\text{est}}(k - \ell(t) + j, t) = A \otimes x_{\text{est}}(k - \ell(t) + j - 1, t) \oplus$$
$$B \otimes \tilde{u}(k - \ell(t) + j - 1, t) \qquad \text{for } j = 1, \ldots, \ell(t),$$

where for the components of $\tilde{u}(k - \ell(t) + j - 1, t)$ that are less than t we take the actually applied input times, and for the other components we take the computed values. Finally, the value of the state $x(k)$ that can be used to compute the MPL-MPC controller at time t is given by $x(k)|_t$ with components $[x(k)|_t]_i$ for $i = 1, \ldots, n$ such that

$$[x(k)_t]_i \ = \ \begin{cases} [x_{\text{true}}(k)]_i & \text{if } [x_{\text{true}}(k)]_i \text{ is known at time } t \ \ (\text{so } [x_{\text{true}}(k)]_i \leqslant t) \\ [x_{\text{est}}(k, t)]_i & \text{otherwise.} \end{cases}$$

A second problem that arises in the implementation of MPL-MPC is to find the best time $t = t(k)$ to determine $x(k)|_t$ and to start the optimization to compute the optimal control sequence $\tilde{u}(k)$ with elements $u(k|k), u(k+1|k), \ldots, u(k+N_c-1|k)$. A good indication is found in the previous computed sequence $\tilde{u}(k-1)$ with elements $u(k-1|k-1), u(k|k-1), \ldots, u(k+N_c-2|k-1)$: an appropriate time instant to start the computation for the kth optimal input sequence is given by

$$t(k) = \min_j [u(k|k - 1)]_j - t_c \ ,$$

where t_c is an upper bound for the computation time of the optimization. If the computational effort for the optimization is not high, i.e.,

$$t_c \ll t(k + 1) - t(k) \ ,$$

then we can additionally do one or more intermediate optimizations with new information. If we have a system with more than one input, then if some of the kth input event operations have already started when we start to recompute the kth optimal input sequence, the inputs that have already been computed and applied should be fixed to their actual values in the optimization, which will lead to a number of additional equality constraints. Note that convexity of the relaxed MPL-MPC problem will not be lost by adding these constraints, so that Theorem 2.1 can still be applied.

4   Closed-loop behavior

In this section we will take a closer look at the closed-loop behavior of an MPL system and an MPC controller with a control law as derived in Section 2. We will only consider SISO systems, but most of the properties can be directly interpreted in the multi-variable case.

---

[5]Since the components of $x(k)$ correspond to event times, they are in general easy to measure. Also note that measurements of occurrence times of events are in general not as susceptible to noise and measurement errors as measurements of continuous-time signals involving variables such as temperature, speed, pressure, etc.

4.1 Closed-loop expression

In conventional MPC theory, in the absence of inequality constraints, the closed loop consisting of the (conventional) linear time-invariant (LTI) process with the MPC controller, is again an LTI system (in the conventional algebra). Unfortunately, there is no analogous property for MPL systems. For two special cases, a closed analytic expression for the optimal MPL-MPC input sequence was found in [9][6]. These expressions involve the operations minimization and addition. In fact, they are linear in the min-plus algebra, which is the dual of the max-plus algebra and which has minimization and addition as basic operations. So in these cases the closed-loop system would be a min-max-plus system. In the general case, the closed-loop system (consisting of the MPL process with the MPL-MPC controller) will not be an MPL system, but it will be piecewise affine in the state $x(k)$ and reference $r(k)$ and it can be described by max-min-plus-scaling functions (this follows directly from the results of [3, 12, 16]).

4.2 Stability

Stability in conventional system theory is concerned with boundedness of the states. In MPL systems however, k is an event counter and $x_i(k)$ refers to the occurrence time of an event. So the sequence $x_i(k), x_i(k+1), \ldots$ should always be non-decreasing, and for $k \to \infty$ the event time $x_i(k)$ will usually grow unbounded. We therefore adopt the following notion of stability for discrete event systems [20].

Definition 4.1 *A discrete event system is called stable if all its buffer levels remain bounded.*

Note that in our case we have due dates and that we assume that finished parts are removed from the output buffer at the due dates (provided that they are present). This means that there are delays if the parts are not produced before the due date. These delays should also remain bounded. Therefore, we add as an additional condition for stability that all delays between due dates and actual output dates remain bounded as well. If there are no internal buffers that are not (indirectly) coupled to the output of the system (observability), then it is easy to verify that all the buffer levels are bounded if the dwelling times of the parts or batches in the system remain bounded. This implies that for an observable SISO discrete event system with due date $r(k)$ closed-loop stability is achieved if there exist finite constants M_{yr}, M_{ry} and M_{yu} such that

$$y(k) - r(k) \quad \leqslant \quad M_{yr} \tag{9}$$

$$r(k) - y(k) \quad \leqslant \quad M_{ry} \tag{10}$$

$$y(k) - u(k) \quad \leqslant \quad M_{yu} \ . \tag{11}$$

Condition (9) means that the delay between the actual output date $y(k)$ and the due date $r(k)$ remains bounded. Condition (10) implies that the number of parts in the output buffer will remain bounded. Finally, condition (11) means that the time between the starting date $u(k)$ and the output date $y(k)$ (i.e., the throughput time) is bounded.

An important observation is that stability is not an intrinsic feature of the system, but it also depends on the input and the due dates (i.e., the reference signal) of the system.

---

[6]These expressions are only valid for another objective function than the one considered in this paper. In addition the parameter λ has to be set to 0, which is a bad choice, as will be demonstrated in Section 5.2.

More precisely, it depends on the asymptotic slope of the input and due date sequence. If by a wrong choice of the MPL-MPC tuning parameters the input rate is too fast, we can get input buffer overflows and unbounded delays between the output dates and the due dates. A similar effect will occur if the slope of the due date sequence is not steep enough, since then the system cannot complete the tasks in time (i.e., $y(k) \gg r(k)$ for large k) even if $u(k + j) = u(k - 1)$ for $j = 0, 1, \ldots$ (i.e., all tasks are started as soon as possible). So an input rate and a due date rate that are too fast may result in unstable phenomena. We will elaborate on this in Section 5.

4.3 Feasibility

The existence of a solution of MPL-MPC at event step k problem can be verified by solving the system of (in)equalities (4)–(7), which describes the feasible set of the problem. Now, feasibility in the MPL-MPC problem is comparable to feasibility in conventional MPC. Infeasibility occurs when solving $\tilde{u}(k)$ from (4)–(7) results in a solution set that is empty. In that case, some constraints could be released. The constraints (4) and (6) should always be satisfied because of their physical meaning. The control horizon constraint (7) is used to reduce the number of variables in the optimization. By increasing N_c the degrees of freedom increases and the optimization may become feasible for a larger N_c. However, an increase of N_c may give a dramatic increase of computational burden and may also lead to instability in the case of modeling errors. So for constraint relaxation, we therefore concentrate on the constraint (5).

If for a certain step k the problem is not feasible, so if the set described by the constraints (4)–(7) is empty, then constraint (5) can be relaxed as follows. First, we choose a diagonal matrix $R \in \mathbb{R}^{n_E \times n_E}$ with non-negative diagonal entries, where n_E is the number of rows of $E(k)$. Now we introduce a vector $\nu(k) \in \mathbb{R}^{n_E}$ of dummy variables and we solve the problem

$$\min_{\tilde{u}(k), \tilde{y}(k), \nu(k)} \quad J_{\text{out}}(\tilde{y}) + \lambda J_{\text{in}}(\tilde{u}) + \sum_{i=1}^{n_E} \nu_i(k) \tag{12}$$

subject to (4), (6), (7) and

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) + R\nu(k) \tag{13}$$

$$\nu(k) \geqslant 0 \ . \tag{14}$$

The entries of diagonal matrix R give a measure on the violation degree of the corresponding constraints.

For $R > 0$ optimization problem (12)-(14) is feasible since the constraints can always be met by making the components of $\nu(k)$ sufficiently large. Also note that inclusion of the term $\nu_1(k) + \cdots + \nu_{n_E}(k)$ in the objective function makes the constraint violations w.r.t. the original infeasible problem as small as possible. Furthermore, if the original (infeasible) MPL-MPC problem satisfies the conditions of Theorem 2.1 (i.e. the mapping $\tilde{y} \to F(k)\tilde{y}$ is a monotonically non-decreasing function of $\tilde{y}$) then the problem (12)–(14) also satisfies these conditions so that Theorem 2.1 still applies. Moreover, the new objective function is also convex since the relaxation term is linear.

5 Tuning

In this section we will give some guidelines to find suitable choices of the three tuning parameters (N_p, N_c, λ) and to select an appropriate due date sequence $r(k)$. The selection of appropriate parameters has to lead to a stabilizing and effective control law. Again we assume that we are dealing with a SISO system (so $l = m = 1$). Furthermore, we will assume irreducibility of the system[7]. In many applications, e.g., in manufacturing systems, this assumption is not restrictive [6].

The parameters N_p, N_c and λ are the three basic tuning parameters of the MPC algorithm. However, as we have already pointed out in the previous section, a closer look at the due dates is necessary for stability reasons. As will be become clear in this section, the conventional MPC rules of thumb for tuning of N_p, N_c and λ can be applied to MPC for MPL systems as well, with only minor changes. In conventional MPC the following rules of thumb for selecting N_p, N_c and λ are used [23]:

- The parameter λ is usually chosen as small as possible, 0 in most cases. In many cases (e.g. for non-minimum phase systems), the choice $\lambda = 0$ will lead to stability problems and so λ should be chosen as the smallest positive value that still results in a stabilizing controller.

- The prediction horizon N_p is related to the length of the step response of the system: the time interval $(1, N_\mathrm{p})$ should contain the crucial dynamics of the process.

- The control horizon $N_\mathrm{c} \leqslant N_\mathrm{p}$ is usually taken equal to the system order.

Before we discuss the MPL-MPC tuning rules, we first need to consider some properties of the impulse response of an MPL system. The sequence $\{e(k)\}_{k=0}^{\infty}$ with $e(0) = 0$ and $e(k) = \varepsilon$ for $k \neq 0$ is the max-plus-algebraic unit impulse. The output sequence that results from applying a max-plus-algebraic unit impulse to an MPL system is called the impulse response of the system[8]. It is easy to verify that the impulse response of an MPL system with system matrices A, B, C is given by $\{G(k)\}_{k=0}^{\infty}$ with $G(k) = C \otimes A^{\otimes^k} \otimes B$.

Proposition 5.1 ([1, 6]) *Let $\{G(k)\}_{k=0}^{\infty}$ be the impulse response of a SISO MPL system with an irreducible system matrix A. Then there exist constants $c, k_0 \in \mathbb{N} \backslash \{0\}$ and $\rho_0 \in \mathbb{R}$ such that*

$$G(k) = c\,\rho_0 + G(k - c) \qquad \text{for all } k \geqslant k_0. \tag{15}$$

An impulse response that exhibits the behavior (15) is called *ultimately periodic* with *cycle period* c. The variable ρ_0 gives the average duration of a cycle and is equal to the max-plus-algebraic eigenvalue of system matrix A. The length of the impulse response is defined as the minimal value k_0 for which (15) holds.

---

[7]An MPL system with system matrix $A \in \mathbb{R}_{\varepsilon}^{n \times n}$ is said to be irreducible if $(A \oplus A^{\otimes^2} \oplus \ldots \oplus A^{\otimes^{n-1}})_{ij} \neq \varepsilon$ for all i, j with $i \neq j$.

[8]If we consider a production system then we can give the following physical interpretation to the impulse response. At event counter $k = 0$ all the internal buffers of the system are empty. Then we start feeding raw material to the input buffer and we keep on feeding raw material at such a rate that the input buffer never becomes empty. The time instants at which finished products leave the system correspond to the terms of the impulse response.

The minimal system order of an MPL system is of importance in the selection of the control horizon N_c. The state space representation of the input-output behavior of a given MPL system by a triple of system matrices A, B, C is not unique. Just as in conventional system theory we define the minimal system order of an MPL system as the minimal dimension of the system matrix A over all possible state space realizations of the given system. In conventional system theory for linear discrete-time systems the minimal system order is given by the rank of the semi-infinite Hankel matrix $H_{\infty,\infty}$ defined by

$$
H_{\infty,\infty} = \left[\begin{array}{cccc} G(0) & G(1) & G(2) & \dots \\ G(1) & G(2) & G(3) & \dots \\ G(2) & G(3) & G(4) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{array} \right] .
$$

with $G(k)$ the kth Markov parameter. In contrast to linear algebra the different notions of rank (like column rank, row rank, minor rank, ...) are in general not equivalent in the max-plus algebra. Nevertheless, a characterization of the minimal system order of an MPL system can be found in [14]. Unfortunately, computing the minimal system order of an MPL system is not a trivial task and it is often computationally very intensive. However, upper and lower bounds for the minimal system order of an MPL system can be determined as follows. The so-called max-plus-algebraic minor rank and Schein rank of H provide lower bounds [14, 15]. At present, there are no efficient (i.e., polynomial time) algorithms to compute the max-plus-algebraic minor rank or the Schein rank of a matrix. The max-plus-algebraic weak column rank of H provides an upper bound [14, 15]. Efficient methods to compute this rank are described in [7, 14].

5.1 Selection of the due date sequence $r(k)$

Let ρ_0 be the average cycle duration of the system. Then the maximum production rate of the system is given by $1/\rho_0$. The slope of due date sequence must therefore be such that the average production rate is lower than $1/\rho_0$. For a stable (and — if (5) is present — feasible) solution we need a due date sequence $r(k)$ for which there exist a $\rho_r > \rho_0$ and an $r_0 \in \mathbb{R}$, such that $r(k) \geqslant r_0 + k\, \rho_r$ for all k.

5.2 Tuning of the parameter λ

The parameter λ makes a trade-off between minimization of the tracking error and the control effort needed. For $\lambda = 0$, the input sequence is not taken into account in the objective function, and we may not have a unique solution. Any input value $u(k)$ that guarantees $\tilde{y}(k) \leqslant \tilde{r}(k)$ will do, and so we may set $u(k) = u(k-1)$ for all k. This will result in an input buffer overflow for k large.

Input buffer overflow will also appear when $\lambda < 0$, because λJ_{in} will become infinitely small. Therefore, the parameter λ should be chosen larger than zero.

A small change of $\tilde{u}$ in the neighborhood of the optimum may cause a similar change in $\tilde{y}$, such that $\Delta J_{\text{out}} = -\Delta J_{\text{in}}$. For $\lambda = 1$, this causes non-uniqueness of the solution.

For $\lambda > 1$ the input cost criterion J_{in} will be dominant in the optimization, which results in a maximization of the control input. The input will become unbounded in the absence of an upper bound Δu_{max} on the input increment. In the bounded case the increment of the

input signal will be maximal: $\Delta u(k) = \Delta u_{\text{max}}$. In the receding horizon implementation this leads to a unbounded output delay $y(k) - r(k)$ and the system will become unstable.

Resuming, the parameter λ should satisfy

$$0 < \lambda < 1 \ ,$$

and it is usually chosen as small as possible without causing instability or numerical problems in the optimization (see Example 3 in Section 6).

5.3   Tuning of the parameter N_{p}

The time interval $[1, N_{\text{p}}]$ should contain the crucial dynamics of the process, and important information of the due date sequence. To be sure that all crucial dynamics are situated in the prediction interval, a good lower bound for the prediction horizon N_{p} is the length of the impulse response of the system (k_0) (see Example 1). A closer look to the reference signal may become important if the due dates are gathered in batches (see Example 5).

5.4   Tuning of the parameter N_{c}

The real power of the MPC approach lies in the assumption made about future control actions. Instead of allowing them to be "free", the increments of $u(k)$ are assumed to be zero:

$$\Delta^2 u(k + j - 1) = 0 \qquad \text{for } j > N_{\text{c}}.$$

The parameter N_{c}, called control horizon, can be chosen between 1 and N_{p}. We usually take it equal to the upper bound of the minimal system order, which is easy to compute [7, 14]. Choosing N_{c} larger than the system order could be interesting when the constraints are stringent. On the other hand, one may expect that a small N_{c} will lead to a more robust control law in the case of modeling error. The choice $N_{\text{c}} = 1$ often leads to an unstable or a degraded closed-loop behavior, because of a lack on degrees of freedom (see Examples 1 and 2). In many cases, the optimal input signal will be asymptotically equal to $u(k) = u_0 + k \, \Delta u_0$, where u_0 and Δu_0 are appropriate constants. We need at least two degrees of freedom to be able to reach this asymptotic behavior.

6   Examples

The MPC algorithm for MPL systems was simulated for some examples using MATLAB. The objective is to study the effect of changes in the tuning parameters λ, N_{p} and N_{c} and the choice of due date sequence $r(k)$. For the analysis we use two systems.

System 1 is described by the model (1)–(2) with system matrices:

$$A = \begin{bmatrix} 3 & 5 & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 & 4 \\ 6 & 1 & 3 & 2 \\ 3 & 0 & 4 & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} 6 \\ 6 \\ 1 \\ 4 \end{bmatrix}, \quad C = \begin{bmatrix} 6 & 2 & 1 & 2 \end{bmatrix} . \tag{16}$$

This system has a minimal system order $n = 4$, cycle period $c = 4$, cycle duration $\rho_0 = 4.75$ and impulse response length $k_0 = 5$.
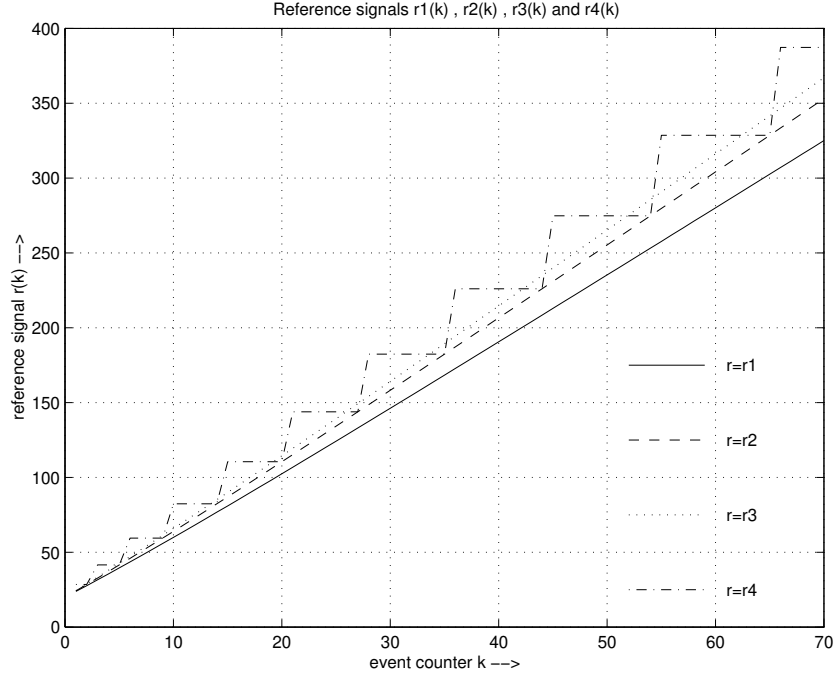
Figure 1: The due date sequences used in the examples.

System 2 is described by (1)–(2) with system matrices:

$$A = \begin{bmatrix} 2 & 5 & 5 & 5 \\ \varepsilon & 2 & 1 & \varepsilon \\ 4 & 2 & 1 & \varepsilon \\ 3 & 0 & \varepsilon & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 6 & 3 & 1 & 1 \end{bmatrix}, \quad (17)$$

and has a minimal system order $n = 3$, cycle period $c = 1$, cycle duration $\rho_0 = 5$ and impulse response length $k_0 = 10$.

We choose the following due date sequences:

$$r_1(k) = 10 + 4.5\,k + 10\,e^{-0.07\,k}$$
$$r_2(k) = 10 + 4.9\,k + 10\,e^{-0.07\,k}$$
$$r_3(k) = 10 + 5.1\,k + 10\,e^{-0.07\,k}\,,$$

and fourth due date sequence signal $r_4(k)$, which is a staircase signal with batches, which increase in length and satisfy: $r_4(k) \geqslant r_2(k)$ for all k (see Figure 1).

Figures 2–7 display the tracking error $y(k) - r(k)$ over 70 simulation samples for various settings of the control parameters.

Example 1 (Influence of N_{p}):

In Figure 2 the influence of N_{p} on the closed-loop of System 1 with an MPC controller is displayed for $N_{\mathrm{c}} = 1$. We have selected $r(k) = r_2(k)$ and fixed $\lambda = 0.001$. It is clear that $N_{\mathrm{p}} = 1$ gives an unstable closed loop behavior, because of the unbounded growth of the
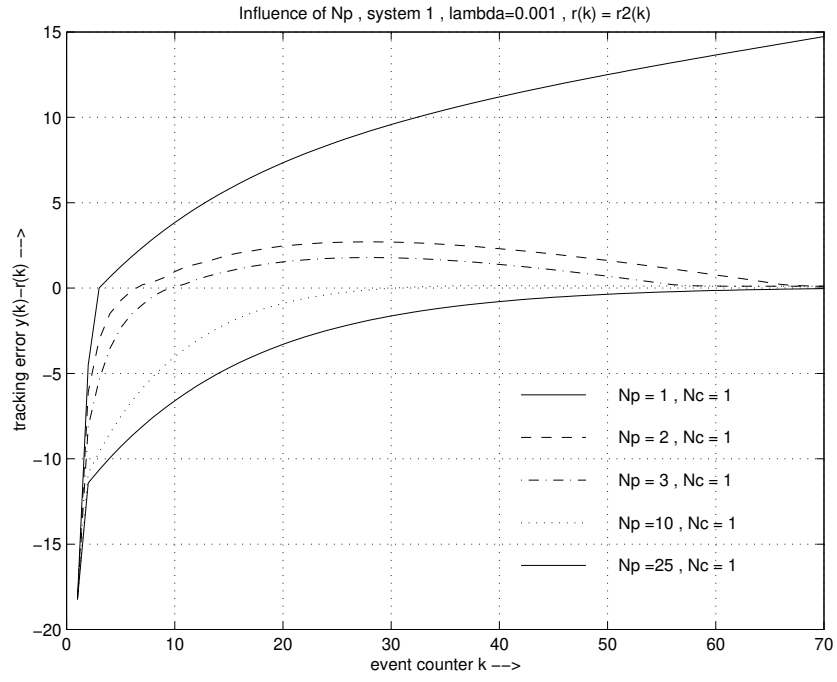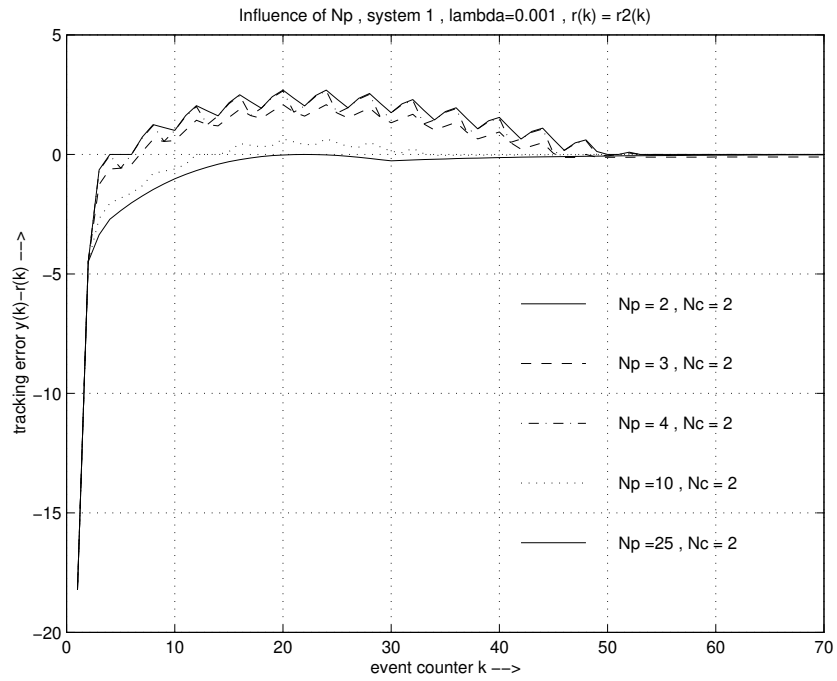
11

Figure 2: Influence of N_{p} for $N_{\mathrm{c}} = 1$.

Figure 3: Influence of N_{p} for $N_{\mathrm{c}} = 2$.
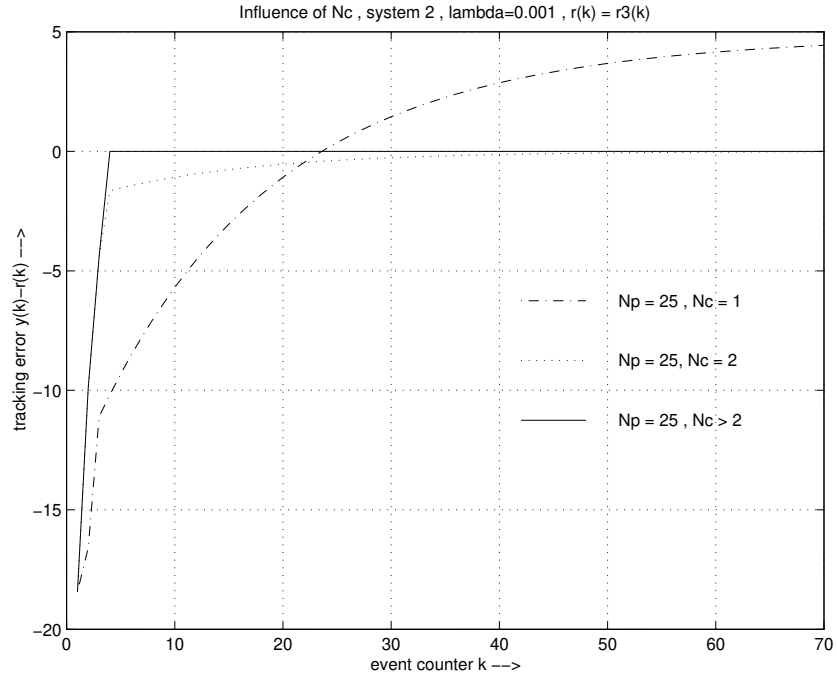
Figure 4: Influence of N_{c}.
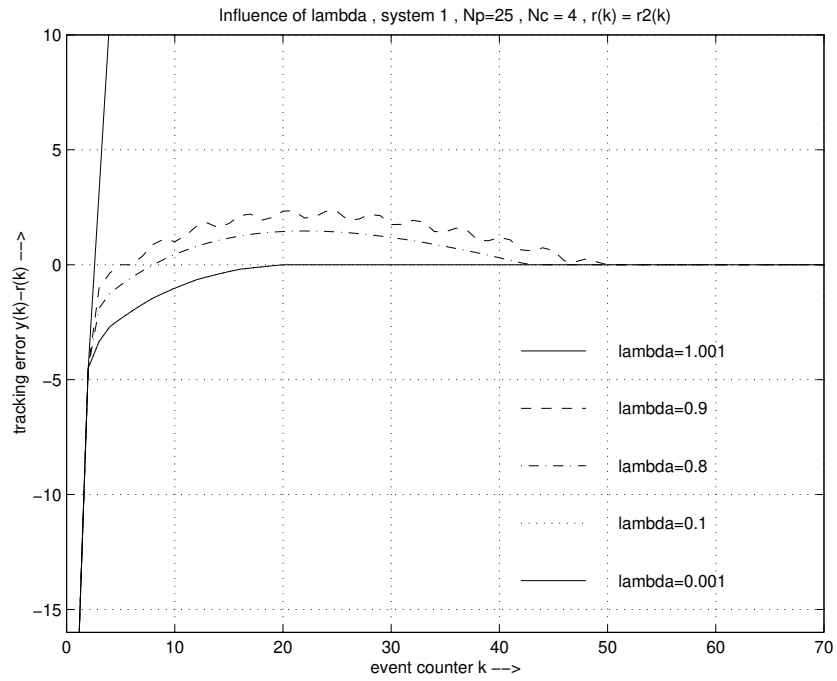

Figure 5: Influence of λ.

13

output delay. Increasing N_p from 2 to 25 leads to a decrease in delays. In Figure 2 we have repeated the above simulations for $N_\mathrm{c} = 2$.

Example 2 (Influence of N_c):

Figure 3 reveals the influence of N_c on the closed-loop behavior of System 2 with an MPC controller. Taking $N_\mathrm{c} = 1$ leads to an unstable behavior, $N_\mathrm{c} = 2$ gives a sluggish output response, and for $N_\mathrm{c} \geqslant 3$ (= the minimal system order) the tracking error is minimal.

Example 3 (Influence of λ):

Figure 4 shows that $\lambda > 1$ leads to an unstable MPC-control law. For $0 < \lambda < 1$ the control law is stabilizing, with a better tracking behavior for λ closer to zero (the choices $\lambda = 0$ and $\lambda = 1$ lead to uniqueness problems and are not displayed).

Example 4 (Influence of $r(k)$):

Figure 5 shows the tracking error of System 1 for the input signals $r_1(k)$, $r_2(k)$ and $r_3(k)$. Note that $r_1(k)$ has an asymptotic slope $\rho_r = 4.5 < \rho_0 = 4.75$. The due date-schedule is too tight and the delays grow unbounded. The asymptotic slopes of $r_2(k)$ and $r_3(k)$ are larger than 4.75. We see that the tracking error $|y(k) - r(k)|$ becomes zero for large k.

Next, we choose the staircase due date sequence $r_4(k)$ in which the due dates are gathered in batches. For this due date sequence we study the influence of N_p. The batch-length ℓ_b increases by 1 for every next batch. If the prediction horizon is fixed to N_p, we can only look N_p "orders" ahead. If a batch length is larger than the prediction horizon, the $(N_\mathrm{p} + 1)$th
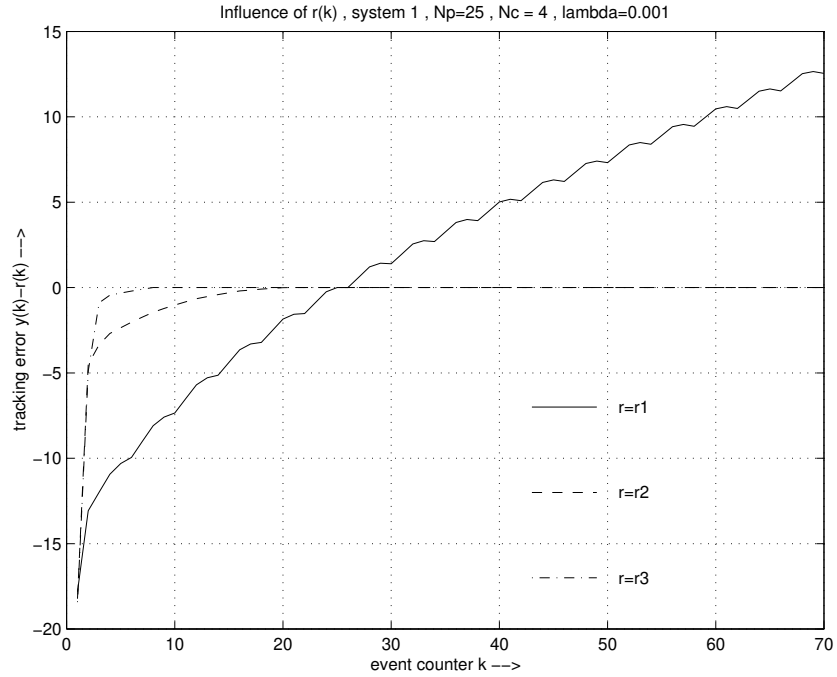

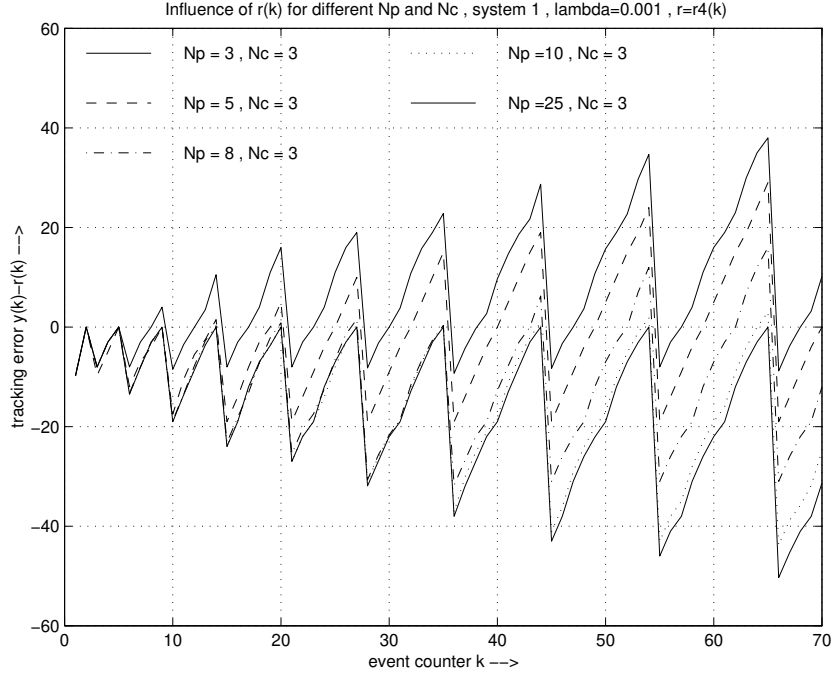Figure 6: Influence of the due date sequence $r(k)$.

Figure 7: Influence of N_p for a staircase due date sequence.

order is not considered in the optimization, and the production of this order is started too late. The result is a delay for $\ell_\mathrm{b} > N_\mathrm{p}$. Consider the dash-dot line in Figure 7, corresponding to $N_\mathrm{p} = 8$. For the batch corresponding to $k = 28, \ldots, 35$, we have $\ell_\mathrm{b} = 8$. This batch can still be produced in time, without delay. However, the next batch, which corresponds to $k = 36, \ldots, 44$ (so $\ell_\mathrm{b} = 9$), results in a delay for $k = 44$.

7 Discussion

Model predictive control (MPC) for max-plus-linear (MPL) systems is a practical approach to design optimal input sequences for a specific class of discrete event systems without concurrency or choice and in which only synchronization plays a role. Typical examples of discrete event systems that fall into this class are serial production lines, queuing systems, railway networks, and parallel computing systems. In this paper we have considered several properties and implementation aspects of MPL-MPC. The timing of MPC for MPL systems is not as straightforward as in conventional MPC. Therefore, we have presented a method to tackle the timing issues of MPL-MPC. In conventional MPC we can guarantee closed-loop stability by the introduction of an end-point constraint [19]. Unfortunately, for MPL systems such an end-point constraint is not yet derived. However, making appropriate choices for the controller parameters and selecting an adequate due date sequence, results in a stabilizing and effective MPC control law. Initial settings for the parameters $(N_\mathrm{p}, N_\mathrm{c}, \lambda)$ were given and the influence of the due date sequence $r(k)$ was studied. In practical industrial situations, these initial parameter settings have to be fine-tuned to obtain the desired closed-loop behavior. Because of the receding horizon strategy used in MPC, and the properties of the max-plus algebra, it is not so straightforward to study closed-loop behavior and an analytic closed-loop

expression is hard to find (contrary to conventional MPC for linear time-invariant discrete-time systems). However, it can be shown that the closed-loop MPL-MPC system is piecewise affine. The issue of stability has been discussed and we have considered the relation between buffer overflow and the values of the tuning parameters and the due date signal. Based on closed-loop aspects and some illustrative examples, we have derived some guidelines for the settings of tuning parameters and the due date sequence.

MPC for max-plus-linear systems is closely related to MPC for mixed-logical-dynamical (MLD) systems [2, 5]. To be more precise: the set of max-plus-linear systems is a subset of the class of max-min-plus-scaling systems, which is shown to be equivalent to the above mentioned set of MLD systems [3, 4]. The disadvantage of MPC using MLD models is the fact that this approach needs integer variables and therefore will lead to a mixed integer linear programming problem, which is known to be NP-hard. The advantage of exploiting the special structure of MPL systems is that the final optimization in the MPC algorithm boils down to a linear programming problem with *real-valued* variables, which is polynomially solvable [17] and for which efficient algorithms are available.

An important topic for future research will be to include noise and uncertainty in the timing and tuning rules, based on the MPC algorithms for perturbed MPL systems developed in [26, 25, 24]. Furthermore, we will study stability in more detail and put our focus on a MPL-MPC equivalent of the conventional MPC end-point constraint.

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. John Wiley & Sons, New York, 1992.

[2] A. Bemporad, F. Borrelli, and M. Morari. Optimal controllers for hybrid systems: Stability and piecewise linear explicit form. In *Conference on Decision and Control' 2000, Sydney, Australia*, pages 1190–1194, December 12–15, 2000.

[3] A. Bemporad, W.P.M.H. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 1645–1650, Orlando, Florida, December 2001.

[4] A. Bemporad, W.P.M.H. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. Technical report, January 2002. Accepted for publication in *IEEE Transactions on Automatic Control*.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[6] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control*, 30(3):210–220, March 1985.

[7] R.A. Cuninghame-Green. *Minimax Algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 1979.

[8] C.R. Cutler and B.L. Ramaker. Dynamic matrix control – a computer control algorithm. In *Proceedings of the 86th AIChE National Meeting*, Houston, Texas, April 1979.

[9] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete-event systems: Extended report & Addendum. Technical report, Control Laboratory, Fac. of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, November 2000. This is an extended version of [11].

[10] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear systems. In *Proceedings of the 2000 American Control Conference*, pages 4046–4050, Chicago, Illinois, June 2000.

[11] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[12] B. De Schutter and T.J.J. van den Boom. Model predictive control for max-min-plus-scaling systems. In *Proceedings of the 2001 American Control Conference*, pages 319–324, Arlington, Virginia, June 2001.

[13] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.

[14] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, France, July 1992.

[15] S. Gaubert. On rational series in one variable over certain dioids. Technical Report 2162, INRIA, Le Chesnay, France, January 1994.

[16] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[17] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, January–February 1979.

[18] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, England, 2002.

[19] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(7):789–814, June 2000.

[20] K.M. Passino and K.L. Burgess. *Stability analysis of discrete event systems*. John Wiley & Sons, Inc., New York, USA, 1998.

[21] S. Qin and T. Badgewell. An overview of industrial model predictive control technology, 1997.

[22] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, September 1978.

[23] A.R.M. Soeterboek. *Predictive control - A unified approach*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[24] T.J.J. van den Boom and B. De Schutter. Model predictive control for perturbed max-plus-linear systems: A stochastic approach. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 4535–4540, Orlando, Florida, December 2001.

[25] T.J.J. van den Boom and B. De Schutter. MPC for perturbed max-plus-linear systems. In *Proceedings of the European Control Conference 2001 (ECC'01)*, pages 3783–3788, Porto, Portugal, September 2001.

[26] T.J.J. van den Boom and B. De Schutter. Model predictive control for perturbed max-plus-linear systems. *Systems & Control Letters*, 45(1):21–33, January 2002.