

Technical report bds:00-11

MPC for max-plus-linear systems: Closed-loop behavior and tuning*

T. van den Boom and B. De Schutter

If you want to cite this report, please use the following reference instead:

T. van den Boom and B. De Schutter, "MPC for max-plus-linear systems: Closed-loop behavior and tuning," *Proceedings of the Workshop on Systems with Time-Domain Constraints*, Eindhoven, The Netherlands, 18 pp., Aug.–Sept. 2000.

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
Current URL: <http://www.dcsc.tudelft.nl>

*This report can also be downloaded via http://pub.deschutter.info/abs/00_11.html

MPC for max-plus-linear systems: Closed-loop behavior and tuning

Ton van den Boom* and Bart De Schutter*

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 the closed-loop behavior and on the tuning aspects of MPC for max-plus-linear discrete event systems.

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

1 Introduction

Model predictive control (MPC) was pioneered simultaneously by Richalet *et al.* [14], and Cutler and Ramaker [6]. Since then, MPC has become probably the most applied advanced control technique in the process industry and many papers report successful applications. 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.
- It is applicable to multi-variable systems.
- It is capable of tracking pre-scheduled reference signals, using the concept of making predictions based on a process model.
- It is an easy-to-tune method. Basically three parameters have to be chosen and adequate tuning rules are available.

Traditionally MPC uses linear discrete-time models for the process to be controlled. Recently, we have extended the MPC framework to max-plus-linear systems [9, 8], which is a special class of discrete event systems that can be described by a model that is “linear” in the max-plus algebra.

Typical examples of discrete event systems are flexible manufacturing systems, telecommunication networks, parallel processing systems, traffic control systems and logistic systems. The class of discrete event systems essentially consists of man-made systems that contain a

*Control Laboratory, 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

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]. In general, models that describe the behavior of a discrete event system are nonlinear in conventional algebra. However, there is a class of discrete event systems – the max-plus-linear discrete event systems – that can be described by a model that is “linear” in the max-plus algebra [1, 5]. The max-plus-linear discrete event systems can be characterized as the class of discrete event systems in which only synchronization and no concurrency or choice occurs. So typical examples are serial production lines, production systems with a fixed routing schedule, and railway networks.

In [9, 8] we have only introduced the MPC framework for max-plus-linear systems and focused on efficient solution techniques for a single MPC step. In this paper we investigate 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 also 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 present the max-plus algebra and max-plus-linear discrete event systems. In Section 3 we give a brief introduction to conventional MPC for linear discrete-time systems. In Section 4 we present MPC for max-plus-linear systems. In Section 5 we discuss the closed-loop properties of max-plus-algebraic MPC. Next we discuss the tuning of the parameters in MPC for max-plus-linear systems. We conclude with some illustrative examples.

2 Max-plus-linear discrete event systems

The basic operations of the max-plus algebra [1, 5] are maximization and addition, which will be represented by \oplus and \otimes respectively:

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

for $x, y \in \mathbb{R}_\varepsilon \stackrel{\text{def}}{=} \mathbb{R} \cup \{-\infty\}$. Define $\varepsilon = -\infty$. 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, 5]). The rules for the order of evaluation of the max-plus-algebraic operators are similar to those of conventional algebra. So max-plus-algebraic power has the highest priority, and max-plus-algebraic multiplication has a higher priority than max-plus-algebraic addition.

The basic max-plus-algebraic operations are extended to matrices as follows. If $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ and $C \in \mathbb{R}_\varepsilon^{n \times p}$ then

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}) \quad \text{and} \quad (A \otimes C)_{ij} = \bigoplus_{k=1}^n a_{ik} \otimes c_{kj} = \max_k(a_{ik} + c_{kj})$$

for all i, j . Note the analogy with the definitions of matrix sum and matrix product in conventional linear algebra. The matrix $\mathcal{E}_{m \times n}$ is the $m \times n$ max-plus-algebraic zero matrix:

$(E_{m \times n})_{ij} = \varepsilon$ for all i, j . The matrix E_n is the $n \times n$ max-plus-algebraic identity matrix: $(E_n)_{ii} = 0$ 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 $k = 1, 2, \dots$

In [1, 4, 5] 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:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k) \quad (1)$$

$$y(k) = C \otimes x(k) \quad (2)$$

with $A \in \mathbb{R}_\varepsilon^{n \times n}$, $B \in \mathbb{R}_\varepsilon^{n \times m}$ and $C \in \mathbb{R}_\varepsilon^{l \times n}$ where m is the number of inputs and l the number of outputs. The vector x represents the state, u is the input vector and y is the output vector of the system. Note the analogy of the description (1)–(2) with the state space model for linear discrete-time systems (see also (3)–(4)). This analogy is another reason why the symbols \oplus and \otimes are used to denote max and $+$. However, an important difference with the linear discrete-time description, where the index k is a sample counter that increases each clock cycle, is that in the model (1)–(2) the components of the input, the output and the state are event times, and that the counter k is an event counter (and event occurrence instants are in general not equidistant).

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 k th batch of intermediate products, and $y(k)$ the time instants at which the k th batch of finished products leaves the system. A discrete event system that can be modeled by (1)–(2) will be called a max-plus-linear time-invariant discrete event system or *max-plus-linear* (MPL) system for short.

Remark 2.1 In the Section 4 we will use the deterministic model (1)–(2) as an approximation of a discrete event system with uncertainty and/or modeling errors. Therefore, and since we do not want to make our exposition on the extension of the MPC framework to MPL systems overly complicated, we have not included any noise terms in the model (1)–(2). \diamond

3 Model predictive control

In this section we give a short introduction to MPC for linear discrete-time systems. Since we will only consider the deterministic, i.e. noiseless, case for MPL systems, we will also omit the noise terms in this introduction to MPC. More extensive information on MPC for discrete-time systems can be found in [2, 3, 10] and the references therein.

Consider a plant that can be modeled by a state space description of the form

$$x(k+1) = Ax(k) + Bu(k) \quad (3)$$

$$y(k) = Cx(k) \quad (4)$$

The vector x represents the state, u the input vector, and y the output vector. If the system has m inputs and l outputs and if the dimension of the state is n , then we have $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$. In order to distinguish systems that can be described by a model of the form (3)–(4) from max-plus-linear systems, a system that can be modeled by (3)–(4) will be called a *plus-times-linear* (PTL) system.

In MPC a performance index or cost criterion J is formulated that reflects the reference tracking error (J_{out}) and the control effort (J_{in}):

$$J = J_{\text{out}} + \lambda J_{\text{in}} = \sum_{j=1}^{N_p} \left(\hat{y}(k+j|k) - r(k+j) \right)^T \left(\hat{y}(k+j|k) - r(k+j) \right) + \lambda \sum_{j=1}^{N_p} u^T(k+j-1)u(k+j-1) \quad (5)$$

where $\hat{y}(k+j|k)$ is the estimate of the output at time step $k+j$ based on the information available at time step k , r is a reference signal, λ is a nonnegative scalar, and N_p is the prediction horizon. Note that J_{out} and J_{in} depend on the output and the input of the system respectively.

In MPC the input is taken to be constant from a certain point on: $u(k+j) = u(k+N_c-1)$ for $j = N_c, \dots, N_p - 1$ where N_c is the control horizon. The use of a control horizon leads to a reduction of the number of optimization variables. This results in a decrease of the computational burden, a smoother controller signal (because of the emphasis on the average behavior rather than on aggressive noise reduction), and a stabilizing effect (since the output signal is forced to its steady-state value).

MPC uses a receding horizon principle: after computation of the optimal control sequence $u(k), \dots, u(k+N_c-1)$, only the first control sample $u(k)$ will be implemented, subsequently the horizon is shifted one sample, the model is updated with new information of the measurements, and a new optimization at time step $k+1$ is performed.

By successive substitution of (3) in (4), estimates of the future values of the state and the output can be computed [2]:

$$\begin{aligned} \hat{x}(k+j|k) &= A^j x(k) + \sum_{i=0}^{j-1} A^{j-i} B u(k+i) \\ \hat{y}(k+j|k) &= C A^j x(k) + \sum_{i=0}^{j-1} C A^{j-i} B u(k+i) \end{aligned}$$

for $j = 1, 2, \dots$. In matrix notation we obtain:

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

for

$$\tilde{y}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix}, \quad \tilde{r}(k) = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N_p) \end{bmatrix}, \quad \tilde{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} \quad (6)$$

and

$$\tilde{D} = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1} & CA^{N_p-2}B & \dots & CB \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix},$$

where \tilde{D} is called the predictor matrix and $\tilde{C}x(k)$ the free-run output signal. The cost criterion is now equal to

$$\begin{aligned} J &= (\tilde{y}(k) - \tilde{r}(k))^T (\tilde{y}(k) - \tilde{r}(k)) + \lambda \tilde{u}^T(k) \tilde{u}(k) \\ &= \tilde{u}^T(k) (\tilde{D}^T \tilde{D} + \lambda I) \tilde{u}(k) + 2(\tilde{C}x(k) - \tilde{r}(k))^T \tilde{D} \tilde{u}(k) + (\tilde{C}x(k) - \tilde{r}(k))^T (\tilde{C}x(k) - \tilde{r}(k)) . \end{aligned}$$

The MPC problem at time step k for PTL systems is defined as follows:

Find the input sequence $u(k), \dots, u(k + N_c - 1)$ that minimizes the performance index J subject to the linear constraint

$$E(k)\tilde{u}(k) + F(k)\tilde{y} \leq h(k) \quad (7)$$

with $E(k) \in \mathbb{R}^{p \times m N_p}$, $F(k) \in \mathbb{R}^{p \times l N_p}$, $h(k) \in \mathbb{R}^p$ for some integer p , and where the inequality holds componentwise, and subject to the control horizon constraint

$$u(k + j) = u(k + N_c - 1) \quad \text{for } j = N_c, N_c + 1, \dots \quad (8)$$

Note that minimizing J subject to the linear constraints (7) and (8), boils down to a convex quadratic programming problem, which can be solved very efficiently.

4 MPC for max-plus-linear discrete event systems

In this section we briefly repeat the results of [9, 8] where we have extended and adapted the MPC framework from PTL systems to MPL systems.

We use the max-plus-linear model (1)–(2) as an approximation of a discrete event system with uncertainty and/or modeling errors. This also motivates the use of a receding horizon strategy when we define MPC for MPL systems, since then we can regularly update our model of the system as new information and measurements become available. Other reasons for using a finite horizon in MPC for MPL systems are that it allows the inclusion of general linear constraints on the inputs and outputs, and that it reduces the computational complexity.

4.1 Evolution of the system

We assume that $x(k)$, the state at event step k , can be measured or estimated using previous measurements¹. If we know the state of the system $x(k)$ at event step k then we can estimate the evolution of the output of the system for a given input sequence $u(k), \dots, u(k + N_p - 1)$ using the model (1)–(2). as follows:

$$\hat{x}(k + j|k) = A \otimes \hat{x}(k + j - 1|k) \oplus B \otimes u(k + j - 1) \quad (9)$$

$$\hat{y}(k + j|k) = C \otimes \hat{x}(k + j|k) \quad (10)$$

for $j = 1, 2, \dots, N_p$. Note that $\hat{x}(k - 1|k) = x(k - 1)$.

Just as in MPC for linear systems, we can eliminate the state estimates from (9)–(10) in

¹Since the components of $x(k)$ correspond to event times, they are in general easy to measure. Also note that measuring occurrence times of events is in general not as susceptible to noise as measuring continuous-time signals involving variables such as temperature, speed, pressure, etc.

order to reduce the number of variables in the resulting optimization problem, leading to faster computation of the optimal MPC policy. We have

$$\hat{y}(k+j|k) = C \otimes A^{\otimes j} \otimes x(k) \oplus \bigoplus_{i=0}^{j-1} C \otimes A^{\otimes j-i} \otimes B \otimes u(k+i) . \quad (11)$$

So if we define

$$\tilde{D} = \begin{bmatrix} C \otimes B & \mathcal{E} & \dots & \mathcal{E} \\ C \otimes A \otimes B & C \otimes B & \dots & \mathcal{E} \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes N_p-1} \otimes B & C \otimes A^{\otimes N_p-2} \otimes B & \dots & C \otimes B \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes 2} \\ \vdots \\ C \otimes A^{\otimes N_p} \end{bmatrix},$$

then we obtain

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

where $\tilde{y}(k)$ and $\tilde{u}(k)$ are defined by (6). Note the analogy between these expressions and the corresponding expressions for PTL systems.

4.2 Cost criterion

In MPC for MPL systems the cost criterion is also defined as $J = J_{\text{out}} + \lambda J_{\text{in}}$, where J_{out} is related to the tracking error and J_{in} is related to the control effort. In [9] we have discuss several possible choices for J_{out} and J_{in} for MPL systems. In this paper we will limit ourselves to one particular choice for the input and output cost criterion.

For the output cost criterion we select

$$J_{\text{out},1} = \sum_{i=1}^{lN_p} \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0) , \quad (12)$$

where the reference vector $\tilde{r}(k)$ contains the due dates for the finished products. The cost criterion $J_{\text{out},1}$ is also called the ‘‘tardiness’’. Note that it penalizes all delays with respect to the dues dates.

For the input cost criterion we take

$$J_{\text{in},1} = - \sum_{j=1}^{mN_p} \tilde{u}_i(k+j-1) . \quad (13)$$

Note that — in contrast to MPC for PTL systems where the input energy is minimized — the criterion $J_{\text{in},1}$ *maximizes* the input time instants. This criterion is used to prevent input buffer overflows. For a manufacturing system, this would correspond to a scheme in which raw material is fed to the system as late as possible. As a consequence, the internal buffer levels are kept as low as possible.

4.3 Constraints

Just as in MPC for PTL systems we consider the linear constraint

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) . \quad (14)$$

Note that this constraint encompasses constraints on the input and output event separation times such as

$$a_1(k+j) \leq \Delta u(k+j-1) \leq b_1(k+j) \quad \text{for } j = 1, \dots, N_c \quad (15)$$

$$a_2(k+j) \leq \Delta \hat{y}(k+j|k) \leq b_2(k+j) \quad \text{for } j = 1, \dots, N_p, \quad (16)$$

or maximum due dates for the output events:

$$\tilde{y}(k) \leq \tilde{r}(k) . \quad (17)$$

Since for MPL systems the input and output sequences correspond to occurrence times of consecutive events, they should be nondecreasing. Therefore, we should always add the condition

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (18)$$

to guarantee that the input sequences are nondecreasing.

Finally, we use the condition that feeding rate should stay constant beyond event step $k + N_c$:

$$\Delta u(k+j) = \Delta u(k + N_c - 1) \quad \text{for } j = N_c, \dots, N_p - 1, \quad (19)$$

as control horizon constraint. Note that just as in MPC for PTL systems, this constraint is used to reduce the number of variables in the MPC optimization problem.

4.4 The standard MPC problem for MPL systems

If we combine the material of previous subsections, we finally obtain the following problem:

$$\min_{\tilde{u}(k)} J = \min_{\tilde{u}(k)} J_{\text{out},1} + \lambda J_{\text{in},1} \quad (20)$$

subject to

$$\tilde{y}(k) = \tilde{C} \otimes x(k) \oplus \tilde{D} \otimes \tilde{u}(k) \quad (21)$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \quad (22)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (23)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1 \quad (24)$$

This problem will be called the MPL-MPC problem for event step k .

Recall that MPC uses a receding horizon principle. So after computation of the optimal control sequence $u(k), \dots, u(k+N_c-1)$, 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.

4.5 Algorithms to solve the MPL-MPC problem

In general the problem (20)–(24) is a nonlinear non-convex optimization problem. However, in some cases the problem can be recast as a convex optimization problem, which can be solved very efficiently.

First we introduce the relaxed MPC problem which is also defined by (20)–(24) but with the =-sign in (21) replaced by a \geq -sign:

$$\min_{\tilde{u}(k)} J = \min_{\tilde{u}(k)} J_{\text{out},1} + \lambda J_{\text{in},1} \quad (25)$$

subject to

$$\tilde{y}(k) \geq \tilde{C} \otimes x(k) \oplus \tilde{D} \otimes \tilde{u}(k) \quad (26)$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \quad (27)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (28)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1. \quad (29)$$

This problem will be called the *relaxed* MPL-MPC problem corresponding to the original MPL-MPC problem (20)–(24).

It is easy to verify that set of feasible solutions of the relaxed MPC problem is convex. As a consequence, the relaxed problem is much easier to solve numerically. Note that whereas in the original problem $\tilde{u}(k)$ is the only independent variable since $\tilde{y}(k)$ can be eliminated using (21), the relaxed problem has both $\tilde{u}(k)$ and $\tilde{y}(k)$ as independent variables.

If the linear constraints are monotonically nondecreasing as a function of $\tilde{y}(k)$ (e.g. $F_{ij} \geq 0$ for all i, j), then the optimal solution of the relaxed problem can be transformed into an optimal solution of the original MPC problem:

Theorem 4.1 *Let the mapping $\tilde{y} \rightarrow F(k)\tilde{y}$ be a monotonically nondecreasing function of \tilde{y} . Let $(\tilde{u}^*, \tilde{y}^*)$ be an optimal solution of the relaxed MPL-MPC problem. If we define $\tilde{y}^\# = \tilde{C} \otimes x(k) \oplus \tilde{D} \otimes \tilde{u}^*$ then $(\tilde{u}^*, \tilde{y}^\#)$ is an optimal solution of the original MPL-MPC problem.*

Proof: See [9]. □

So if the linear constraints are monotonically nondecreasing as a function of $\tilde{y}(k)$, the MPL-MPC problem can be recast as a convex problem. Moreover, by introducing some additional dummy variables the problem can even be reduced to a linear programming problem (see [9]).

5 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 the previous section. We will only consider SISO systems, but most of the properties can be directly interpreted in the multi-variable case. The three topics we will elaborate on are:

- Closed-loop expression in the absence of inequality constraints.
- Concept of stability.
- Problem of infeasibility.

5.1 Closed-loop expression

In conventional MPC theory, in the absence of inequality constraints, the closed loop consisting of the PTL process with the MPC controller, is again a PTL system. Unfortunately, it seems that there is no analogous property for MPL systems. One of the indications for this is that there are two special cases for which we can derive a closed analytic expression for the optimal MPL-MPC input sequence [7]². 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. This suggests that in the general case the closed loop system (consisting of the PTL process with the MPL-MPC controller) will probably not be an MPL system.

5.2 Stability

Stability in conventional system theory is concerned with boundedness of the states. In MPL systems however, the variable k is an event counter and $x_i(k)$ refers to the occurrence time of an event. So the sequence $x_i(k), x_i(k+1), \dots$ should always be nondecreasing, and for $k \rightarrow \infty$ the event time $x_i(k)$ will usually grow unbounded. We therefore adopt the following notion of stability for discrete event systems [13].

Definition 5.1 *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 conditions for stability that all delays between due date and actual output date remain bounded as well. If there are no internal buffers that are not (indirectly) coupled to the output of the system, then it is easy to verify that the buffer levels are bounded if the dwelling times of the parts or batches in the system remain bounded. This implies that closed-loop stability is achieved for a SISO system if there exist finite constants M_{yr} , M_{ry} and M_{yu} such that

$$y(k) - r(k) \leq M_{yr} \quad (30)$$

$$r(k) - y(k) \leq M_{ry} \quad (31)$$

$$y(k) - u(k) \leq M_{yu} \quad (32)$$

Condition (30) means that the delay between the actual output date $y(k)$ and the due date $r(k)$ remains bounded. Condition (31) means that the number of parts in the output buffer will remain bounded. Finally, condition (32) 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 reference signal (i.e., the due dates) of the system. Or more precisely, it depends on the asymptotic slope of the input and reference signals. Above a certain rate (defined by the inverse of the asymptotic slope of the impulse response), the

²These 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 6.2.

system, driven by an input which is too fast, “blows up”, and the delays become infinite [1]. The same effect will occur if the slope of the reference signal is not steep enough. Even if $u(k+j) = u(k-1)$ for $j = 0, 1, \dots$ (all tasks are started as soon as possible), the system cannot complete the tasks in time (i.e. $y(k) \gg r(k)$ for large k). We will elaborate on this in the next section.

5.3 Feasibility

The existence of a solution of MPL-MPC at event step k problem can be verified by solving the system of (in)equalities (21)–(24), 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 (21)–(24) results in a solution set that is either empty or non-unique. Non-uniqueness can usually be solved by choosing $\lambda \neq 0$, as will be discussed in the next section. An empty solution set can be caused by conflicting constraints in (22)–(24). Note that in the absence of (22) a feasible solution can always be reached. Specific constraints have to be removed or relaxed if no feasible solution is found. A selection algorithm has to be designed which organizes the constraints in a hierarchical way and removes or adapts the least critical ones first.

Constraint relaxation can be done as follows. The constraints (21) and (23) should always be satisfied because of their physical meaning. Furthermore, the constraint (24) is used to reduce the number of variables. Therefore, we will not relax it. So the only “soft” constraint in the problem is the constraint

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) .$$

This constraint is relaxed as follows. First we choose a diagonal matrix $R \in \mathbb{R}^{n_E \times n_E}$ with positive diagonal entries that determine the relative weights of the constraints (i.e. if satisfying constraint i is more important than satisfying constraint j then we select r_{ii} and r_{jj} such that r_{ii} is much smaller than r_{jj}) where n_E is the number of rows of $E(k)$. Now we introduce a vector $\nu \in \mathbb{R}^{n_E}$ of dummy variables and we solve the problem

$$\min_{\tilde{u}(k), \nu} J_{\text{out},1} + \lambda J_{\text{in},1} + \sum_{i=1}^{n_E} \nu_i \quad (33)$$

subject to

$$\tilde{y}(k) = \tilde{C}(k) \otimes x(k) \oplus \tilde{D}(k) \otimes \tilde{u}(k) \quad (34)$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) + R\nu \quad (35)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (36)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1 \quad (37)$$

$$\nu \geq 0 . \quad (38)$$

This problem is feasible since the constraints can always be met by making the components of the vector ν sufficiently large. Also note that inclusion of the term $\nu_1 + \dots + \nu_{n_E}$ 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 4.1 (i.e. the mapping $\tilde{y} \rightarrow F(k)\tilde{y}$ is a monotonically nondecreasing

function of \tilde{y}) then the problem (33)–(38) also satisfies these conditions so that Theorem 4.1 still applies. Moreover, the new objective function is also convex since the relaxation term is linear.

In general the solution of the MPL-MPC problem is not necessarily unique since the general MPC problem for MPL systems is nonlinear and nonconvex. But if the constraints are monotonically nondecreasing as a function of $\tilde{y}(k)$, then Theorem 4.1 applies and then we have a strictly convex problem that has a unique solution. Note however that in practice the uniqueness issue is not really important since as soon as we have an optimal solution that satisfies all constraints, we are satisfied and we can use that solution.

6 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 reference signal $r(k)$. Again we assume that we are dealing with a SISO system (so $l = m = 1$). Furthermore, we will assume irreducibility of the system³. In many applications, for example in manufacturing systems, this assumption is not restrictive [4].

The selection of appropriate parameters has to lead to a stabilizing and effective control law. The MPC algorithm computes the vector of controls using optimization of the cost criterion

$$J(\tilde{y}, \tilde{u}) = J_{\text{out},1} + J_{\text{in},1} = \sum_{i=1}^{N_p} \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0) - \lambda \sum_{i=1}^{N_p} \tilde{u}_i(k)$$

with additional conditions

$$\begin{aligned} \Delta u(k+j) &\geq 0 && \text{for } j = 0, \dots, N_p - 1, \\ \Delta^2 u(k+j) &= 0 && \text{for } j = N_c, \dots, N_p - 1. \end{aligned}$$

For now we will not consider constraints of the form (22).

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 reference signal is necessary for stability reasons. As will 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:

- The parameter λ is usually chosen as small as possible, 0 in most cases. In many cases (e.g. for non-minimum phase PTL 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 $(1, N_p)$ should contain the crucial dynamics of the process.
- The control horizon $N_c \leq N_p$ is usually taken equal to the system order.

³An MPL system with system matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$ is said to be irreducible if $(A \oplus A^{\otimes 2} \oplus \dots \oplus A^{\otimes n-1})_{ij} \neq \varepsilon$ for all i, j with $i \neq j$.

Before we discuss the MPL-MPC tuning rules, we first need to consider some properties of the impulse response of a MPL systems. 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 max-plus-algebraic unit impulse to an MPL system is called the impulse response of the system⁴. It is easy to verify that the impulse response of an MPL 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 6.1 ([1, 4]) *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 $c, k_0 \in \mathbb{N} \setminus \{0\}$ and $\rho \in \mathbb{R}$ such that*

$$G(k) = c\rho + G(k - c) \quad \text{for all } k \geq k_0. \quad (39)$$

An impulse response that exhibits the behavior (39) is called *ultimately periodic* with *cycle period* c . The variable ρ 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 now defined as the minimal value k_0 for which (39) holds.

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 realization of the given system. A characterization of the system order of an MPL system can be found in [11].

Define the Hankel matrix $H(\alpha, \beta)$ as

$$H(\alpha, \beta) = \begin{bmatrix} G(0) & G(1) & \dots & G(\beta) \\ G(1) & G(2) & \dots & G(\beta + 1) \\ \vdots & \vdots & \ddots & \vdots \\ G(\alpha) & G(\alpha + 1) & \dots & G(\alpha + \beta - 1) \end{bmatrix}.$$

In conventional system theory for linear discrete-time systems the minimal system order is given by the rank of the Hankel matrix $H(\infty, \infty)$. However, 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 and can only be used to obtain upper and lower bounds for the minimal system order of an MPL system. The so-called max-plus-algebraic minor rank and Schein rank of H provide lower bounds [11, 12]. 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 [11, 12]. Efficient methods to compute this rank are described in [5, 11]. The minimal system order will become of importance in the selection of the control horizon N_c .

6.1 Selection of the reference signal $r(k)$

The maximum production rate of the system is given by $1/\rho$. The slope of reference signal must therefore be such that the average production rate is lower than $1/\rho$. For a feasible

⁴If 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.

solution we need a reference signal $r(k)$ for which there exist a $\rho_r > \rho$ and an $r_0 \in \mathbb{R}$, such that $r(k) \geq r_0 + k \rho_r$ for all k .

6.2 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 measured and we do not have a unique solution. Any input value $u(k)$ that guarantees $\tilde{y}(k) \leq \tilde{r}(k)$ 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. The same happens when $\lambda < 0$, because then $\lambda J_{\text{in},1}$ will become infinitely small. Therefore, the parameter λ should be chosen larger than zero.

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

$$0 < \lambda < 1$$

and is usually chosen as small as possible (see Example 3 in Section 7).

6.3 Tuning of the parameter N_p

The time interval $(1, N_p)$ should contain the crucial dynamics of the process, and important information of the reference signal. To be sure that all crucial dynamics is 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 can become important, if the due dates are gathered in batches (see Example 5).

6.4 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:

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

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 [5, 11]. 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 $N_c = 1$ 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.

7 Examples

The MPC algorithm for MPL systems was simulated in 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 reference signal $r(k)$. For the analysis we use two systems.

The first system is described by equations (1)–(2) with system matrices:

$$A = \begin{bmatrix} 3 & 5 & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 & 4 \\ 6 & 1 & 3 & 2 \\ 3 & 0 & 4 & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} 6 \\ 6 \\ 1 \\ 4 \end{bmatrix}, \quad C = [6 \quad 2 \quad 1 \quad 2]. \quad (40)$$

This system has a system order $n = 4$, cycle period $c = 4$, cycle duration $\rho = 4.75$ and impulse response length $k_0 = 5$.

The second system is described by equations (1)–(2) with system matrices:

$$A = \begin{bmatrix} 2 & 5 & 5 & 5 \\ \varepsilon & 2 & 1 & \varepsilon \\ 4 & 2 & 1 & \varepsilon \\ 3 & 0 & \varepsilon & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \quad C = [6 \quad 3 \quad 1 \quad 1]. \quad (41)$$

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

We choose reference signals defined by

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

and fourth reference signal $r_4(k)$, which is a staircase signal with batches, which increase in length and satisfy: $r_4(k) \geq r_3(k)$ for all k (see Figure 1).

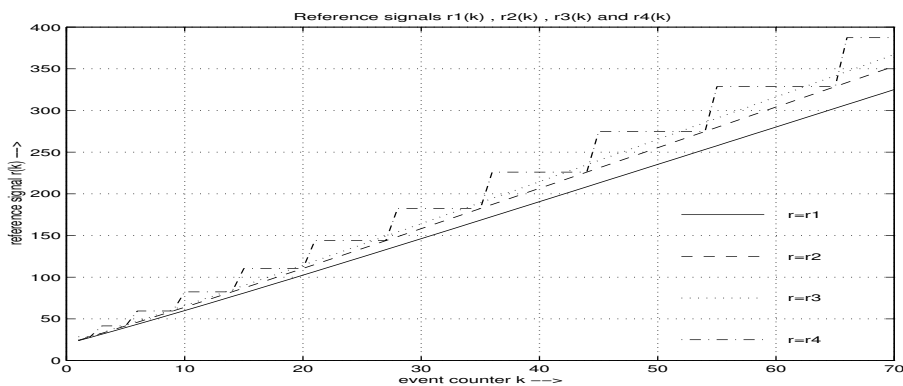


Figure 1: Reference signals.

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

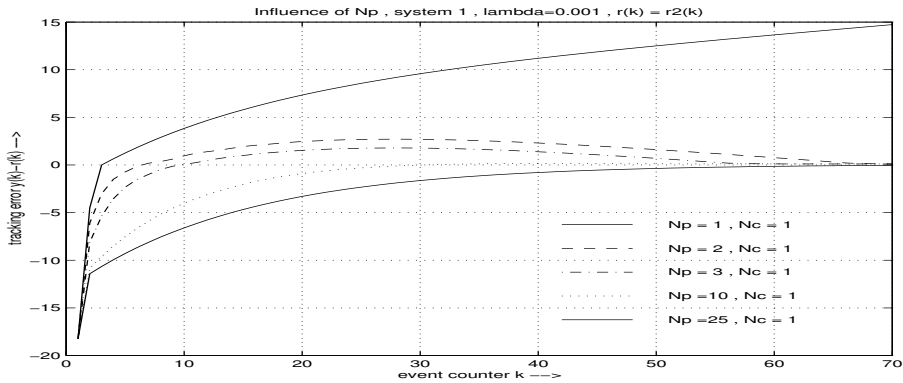


Figure 2: Influence of N_p for $N_c = 1$.

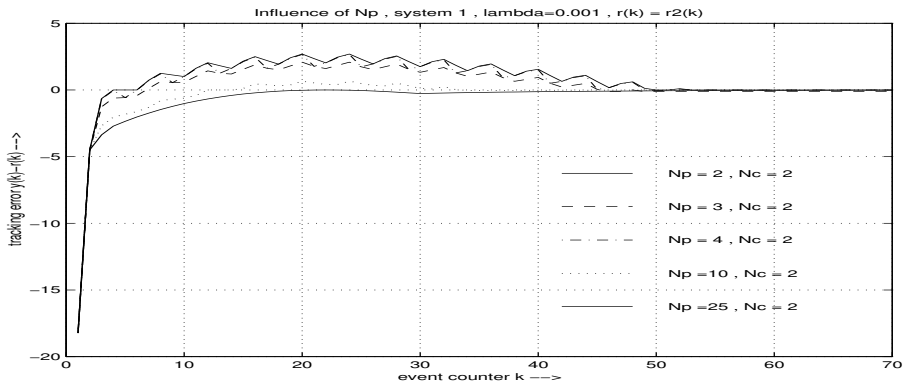


Figure 3: Influence of N_p for $N_c = 2$.

Example 1 (Influence of N_p):

In Figure 2 the influence of N_p on the closed-loop of system (40) with an MPC controller is displayed. It is clear that $N_p = 1$ gives an unstable closed loop behavior, because of the unbounded growth of the output delay. Increasing N_p from 2 to 25 leads to a decrease in delays. We have selected $r(k) = r_2(k)$ and fixed $\lambda = 0.001$.

In Figure 3 the same is done for $N_c = 2$ and $N_p = 2, 3, 4, 10, 25$.

Example 2 (Influence of N_c):

Figure 4 reveals the influence of N_c on the closed-loop behavior of system (41) with an MPC controller. $N_c = 1$ leads to an unstable behavior, $N_c = 2$ gives a sluggish output response, and for $N_c \geq 3$ (= the system order), the tracking error is minimal.

Example 3 (Influence of λ):

Figure 5 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. ($\lambda = 0$ and $\lambda = 1$ lead to uniqueness problems and are not computed).

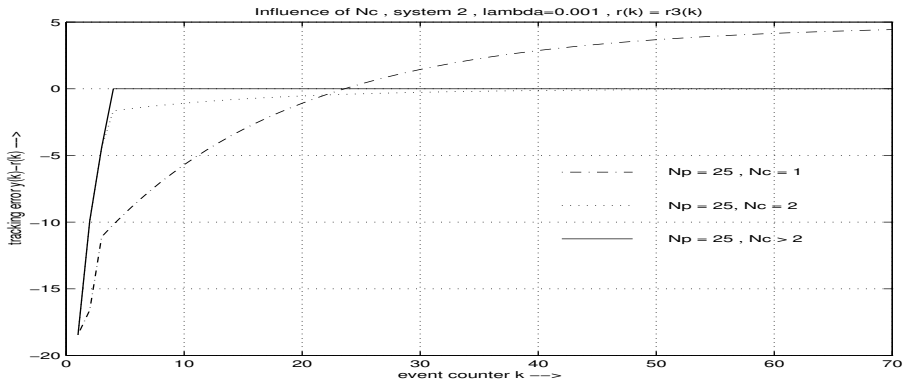


Figure 4: Influence of N_c .

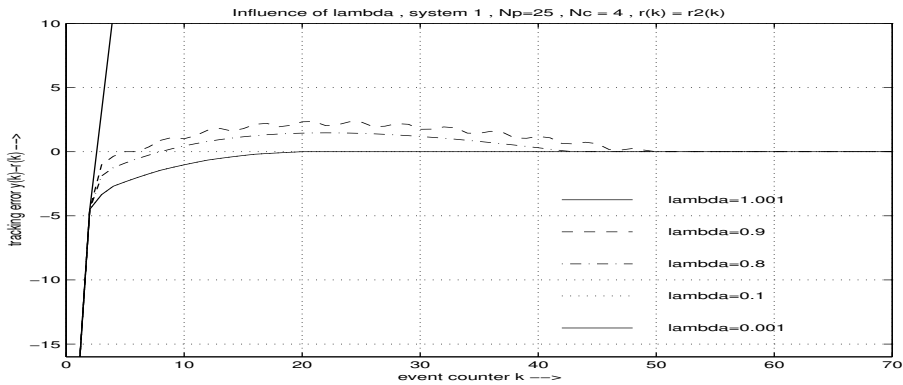


Figure 5: Influence of λ .

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

Figure 6 shows the tracking error of system (40) for the input signals $r_1(k)$, $r_2(k)$ and $r_4(k)$. Note that $r_2(k)$ has a asymptotic slope $\rho_r = 4.5 < \rho = 4.75$. The reference-schedule is too tight and the delays grow unbounded. The asymptotic slope of $r_1(k)$ is 4.9 which is larger than 4.75. We see that the tracking error $|y(k) - r(k)|$ becomes zero for large k . If we choose the staircase reference signal $r_4(k)$ in which the due dates are gathered in batches, we see an increase of the output buffer level through every period. If the length of the batches will continue to increase, the output buffer level will continue to grow as well.

For the staircase reference signal $r(k) = r_4(k)$ 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 $(N_p + 1)$ -th order is not considered in the optimization, and the production of this order is started too late. The result is a delay for $\ell_b > N_p$. Consider the dash-dotted line in Figure 7, corresponding to $N_p = 8$. For the batch corresponding to $k = 28, \dots, 35$, we have $\ell_b = 8$. This batch can still be produced in time, without delay. However, the next batch, which corresponds to $k = 36, \dots, 44$ (so $\ell_b = 9$), results in a delay for $k = 44$.

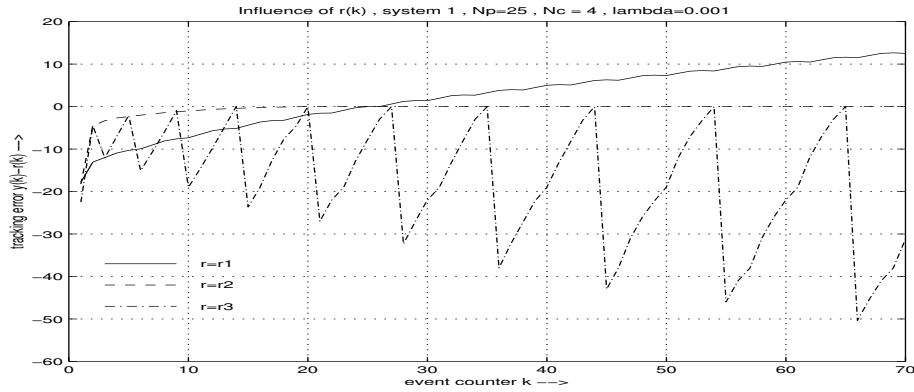


Figure 6: Influence of the reference signal $r(k)$.

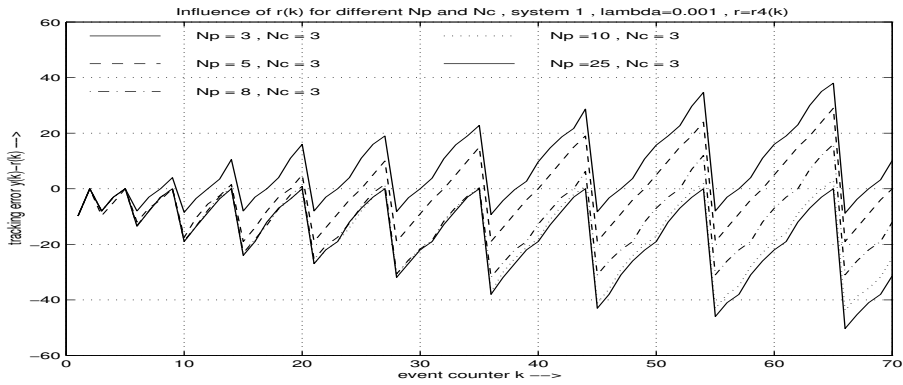


Figure 7: Influence of N_p for a staircase reference signal.

8 Discussion

Model predictive control for max-plus linear systems is a practical approach to design optimal input sequences for a specific class of discrete event systems in which only synchronization and no concurrency or choice plays a role. As shown in this paper, a disadvantage of MPC methods (for both MPL and PTL systems) is that no a priori guarantee on stability or performance can be given. However, making appropriate choices for the controller parameters and selecting an adequate reference signal, results in a stabilizing and effective MPC-control law. Initial settings for the parameters (N_p, N_c, λ) were given and the influence of the reference signal $r(k)$ was studied. In practical industrial situations, these initial parameter settings have to be fine-tuned to obtain the desired closed-loop behavior.

Acknowledgments

This research was partially sponsored by the TMR project ALAPEDES (Algebraic Approach to Performance Evaluation of Discrete Event Systems) of the European Community Training and Mobility of Researchers Program (network contract ERBFMRXCT960074), and by the FWO (Fund for Scientific Research–Flanders) Research Community ICCoS (Identification and Control of Complex Systems).

References

- [1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity*. New York: John Wiley & Sons, 1992.
- [2] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.
- [3] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, “Generalized predictive control – Part I. The basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987.
- [4] 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*, vol. 30, no. 3, pp. 210–220, Mar. 1985.
- [5] R.A. Cuninghame-Green, *Minimax Algebra*, vol. 166 of *Lecture Notes in Economics and Mathematical Systems*. Berlin, Germany: Springer-Verlag, 1979.
- [6] C.R. Cutler and B.L. Ramaker, “Dynamic matrix control – A computer control algorithm,” in *Proceedings of the 86th AIChE National Meeting*, Houston, Texas, Apr. 1979.
- [7] B. De Schutter and T. van den Boom, “Model predictive control for max-plus-linear discrete-event systems: Extended report & Addendum,” Tech. rep. bds:99-10a, Control Systems Engineering, Fac. of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, Nov. 2000. A short version of this report has been published in *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001.
- [8] B. De Schutter and T. van den Boom, “Model predictive control for max-plus-linear systems,” in *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, pp. 4046–4050, June 2000.
- [9] B. De Schutter and T. van den Boom, “Model predictive control for max-plus-linear discrete event systems,” *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001.
- [10] C.E. García, D.M. Prett, and M. Morari, “Model predictive control: Theory and practice – A survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.
- [11] S. Gaubert, *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, France, July 1992.
- [12] S. Gaubert, “On rational series in one variable over certain dioids,” Tech. rep. 2162, INRIA, Le Chesnay, France, Jan. 1994.
- [13] K.M. Passino and K.L. Burgess, *Stability Analysis of Discrete Event Systems*. New York: John Wiley & Sons, 1998.
- [14] J. Richalet, A. Rault, J.L. Testud, and J. Papon, “Model predictive heuristic control: Applications to industrial processes,” *Automatica*, vol. 14, no. 5, pp. 413–428, Sept. 1978.