Delft University of Technology

Fac. of Information Technology and Systems **Control Systems Engineering**

Technical report bds:00-19

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 2001 American Control Conference*, Arlington, Virginia, pp. 325–330, June 2001.

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

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

Ton van den Boom and Bart De Schutter

Control Laboratory, Faculty of Information Technology and Systems,
Delft University of Technology, P.O.Box 5031, 2600 GA Delft, The Netherlands,
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 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.* [9], and Cutler and Ramaker [4]. 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, like systematic constraint handling, applicability to multivariable systems, good signal tracking and disturbance rejection properties. Finally, it is an easy-to-tune method. Basically three tuning parameters have to be chosen and adequate tuning rules are available.

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 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]. Discrete event systems with synchronization but no concurrency can be described by models that are "linear" in the max-plus algebra [1, 3], and are denoted as max-plus-linear systems.

In [6, 5] we have extended the MPC framework to max-plus-linear systems and focused on efficient solution techniques for a single step in the max-plus-linear MPC algorithm. 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 introduce max-plus-linear discrete event systems and we briefly recapitulate the MPC methodology for max-plus-linear systems. In Section 3 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 The MPC problem for max-plus-linear systems

In [1, 2, 3] 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:

$$x(k+1) = A \otimes x(k) \;\oplus\; B \otimes u(k) \tag{1}$$
$$y(k) = C \otimes x(k) \tag{2}$$

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. The operations $\oplus$ and $\otimes$ denote max-plus-algebraic addition and max-plus-algebraic multiplication 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 $\varepsilon = -\infty$. 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 to:

$$\begin{aligned}(A \oplus B)_{ij} &= a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})\ , \quad \forall i,j \\ (A \otimes C)_{ij} &= \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_k (a_{ik} + c_{kj})\ , \quad \forall i,j\end{aligned}$$

In [6] we show that prediction of future values of $y(k)$ 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 $\tilde{D}$ only depend on A,B,C and $\tilde{u}$, $\tilde{y}$ 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 time k and N_p is the prediction horizon. The MPC problem for max-plus-linear (MPL) systems is formulated as follows:

$$\begin{aligned}&\min_{\tilde{u}(k),\tilde{y}(k)} J(\tilde{u}(k), \tilde{y}(k)) = \min_{\tilde{u}(k),\tilde{y}(k)} J_\mathrm{out}(\tilde{y}(k)) + J_\mathrm{in}(\tilde{u}(k)) \\ &= \min_{\tilde{u}(k),\tilde{y}(k)} \sum_{i=1}^{mN_\mathrm{p}} \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0) - \lambda \sum_{i=1}^{lN_\mathrm{p}} \tilde{u}_i(k) \qquad (3)\end{aligned}$$

subject to

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

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

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

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

J_out is measuring the tracking error of the system, which is equal to the delay between the output date $\tilde{y}_i(k)$ and due date $\tilde{r}_i(k)$ if $(\tilde{y}_i(k) - \tilde{r}_i(k)) > 0$, and zero otherwise. J_in maximizes the input dates $\tilde{u}_i(k)$. Equation (5) reflects constraints on the input and output event separation times or maximum due dates for the output events, equation (6) guarantees a nondecreasing input signal and equation (7) is due to the control horizon N_c (see Section 4). The above problem will be called the MPL-MPC problem[1] for event step k. In [5] we showed that, if the linear constraints are monotonically nondecreasing as a function of $\tilde{y}(k)$, the MPL-MPC problem can be recast as a linear programming problem.

[1] Also other cost criteria are possible. See [6, 5].

MPC uses a receding horizon strategy. So after computation of the optimal control sequence $u(k), \ldots, u(k+N_\mathrm{c}-1)$, 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 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 multivariable case.

3.1 Closed-loop expression

In conventional MPC theory, in the absence of inequality constraints, the closed loop consisting of the (conventional) LTI process with the MPC controller, is again a (conventional) LTI system. Unfortunately, it seems that there is no analogous property for MPL systems. In general, the closed loop of process with MPC controller will not be a MPL system.

3.2 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 $x_i(k), x_i(k+1), \ldots$ should always be nondecreasing, 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 [8].

Definition 3.1 *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, 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

$$y(k) - r(k) \leqslant M_{yr} \quad (8)$$

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

$$y(k) - u(k) \leqslant M_{yu} \quad (10)$$

Condition (8) means that the delay between the actual output date $y(k)$ and the due date $r(k)$ remains bounded. Condition (9) means that the number of parts in the output buffer will remain bounded. Finally, condition (10) 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 date of the system. Or more precisely, it depends on the asymptotic slope of the input and due dates. We will elaborate on this in the next section.

3.3 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. An empty solution set can be caused by conflicting constraints in (5)–(7). Note that in the absence of (5) 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 (4) and (6) should always be satisfied because of their physical meaning. Furthermore, the constraint (7) 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

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

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

$$\min_{\tilde{u}(k),\tilde{y}(k),\nu} J_{\mathrm{out}}(\tilde{y}) + \lambda J_{\mathrm{in}}(\tilde{u}) + \sum_{i=1}^{n_E} \nu_i \quad (11)$$

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

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) + R\nu \quad (12)$$

$$\nu \geqslant 0 \ . \quad (13)$$

This problem is feasible since the constraints can always met by making the components of the vector ν sufficiently large. Furthermore, if the original (infeasible) MPL-MPC problem satisfies the convexity conditions (i.e. the mapping $\tilde{y} \to F(k)\tilde{y}$ is a monotonically nondecreasing function of $\tilde{y}$) then the problem (11) – (13) also satisfies these conditions. Note that the relaxed problem is still a linear programming problem.

4 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 appropriate due dates $r(k)$. Again we assume that we are dealing with a SISO system (so $l = m = 1$). Furthermore, we will assume irreducibility of the system[2]. In many applications, for example in manufacturing systems, this assumption is not restrictive [2].

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 (3) with additional conditions (4), (6) and (7). For now we will not consider constraints of the form (5).

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:

[2] 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 $i \neq j$.

- The parameter λ is usually chosen as the smallest non-negative value that still results in a stabilizing controller.
- The prediction horizon $N_{\rm p}$ is related to the length of the step response of the system: the time interval $(1, N_{\rm p})$ should contain the crucial dynamics of the process.
- The control horizon $N_{\rm c} \leqslant N_{\rm p}$ 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 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 a max-plus-algebraic unit impulse to an MPL system is called the impulse response of the system[3]. 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 4.1 ([1, 2]) *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 $\rho \in \mathbb{R}$ such that*

$$G(k) = c\,\rho + G(k-c) \qquad \text{for all } k \geqslant k_0. \tag{14}$$

An impulse response that exhibits the behavior (14) 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 (14) holds.

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

Instability will occur if the slope of the due date sequence is not steep enough. Even if $u(k+j) = u(k-1)$ for $j = 0, 1, \ldots$ (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). The maximum production rate of the system is given by $1/\rho$. The slope of the due date must therefore be such that the average production rate is lower than $1/\rho$. For a feasible solution we need a due date sequence $r(k)$ for which there exist a $\rho_r > \rho$ and an $r_0 \in \mathbb{R}$, such that $r(k) \geqslant r_0 + k\,\rho_r$ for all k.

[3] 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.

4.2 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 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 $\lambda J_{\rm in}$ 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 $\Delta J_{\rm out} = -\Delta J_{\rm in}$. For $\lambda = 1$, this causes non-uniqueness of the solution.

For $\lambda > 1$ the input cost criterion $J_{\rm in}$ 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 $\Delta u_{\max}$ on the input increment. In the bounded case we the increment of the input signal is maximal: $\Delta u(k) = \Delta u_{\max}$. 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 5), without causing instability or numerical problems in the optimization.

4.3 Tuning of the parameter $N_{\rm p}$

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

4.4 Tuning of the parameter $N_{\rm c}$

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:

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

The parameter $N_{\rm c}$, called control horizon, can be chosen between 1 and $N_{\rm p}$. We usually take it equal to the upper bound of the minimal system order, which is easy to compute [3, 7]. Choosing $N_{\rm c}$ larger than the system order could be interesting when the constraints are stringent. On the other hand, one may expect that a small $N_{\rm c}$ will lead to a more robust control law in the case of modeling error. The choice $N_{\rm c} = 1$ 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.

5 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 due date $r(k)$. For the analysis we use two systems.
The first system is described by equations (1)–(2) with system matrices:

$$A = \begin{bmatrix} 3 & 5 & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 & 4 \\ 6 & 1 & 3 & 2 \\ 3 & 0 & 4 & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} 6 \\ 6 \\ 1 \\ 4 \end{bmatrix}, \qquad (15)$$

$$C = \begin{bmatrix} 6 & 2 & 1 & 2 \end{bmatrix}$$

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:

$$A = \begin{bmatrix} 2 & 5 & 5 & 5 \\ \varepsilon & 2 & 1 & \varepsilon \\ 4 & 2 & 1 & \varepsilon \\ 3 & 0 & \varepsilon & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \qquad (16)$$

$$C = \begin{bmatrix} 6 & 3 & 1 & 1 \end{bmatrix}$$

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 three due date sequences, defined by

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

Figure (1)-(5) 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 1 the influence of N_p on the closed-loop of system (15) with an MPC controller is displayed. It

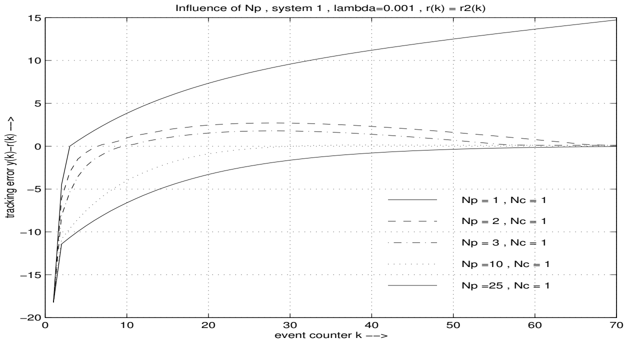

Figure 1: Influence of N_p for $N_\mathrm{c} = 1$.

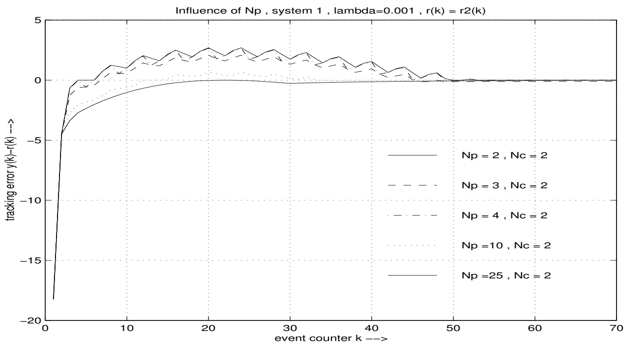

Figure 2: Influence of N_p for $N_\mathrm{c} = 2$.

is clear that $N_\mathrm{p} = 1$ 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 2 the same is done for $N_\mathrm{c} = 2$ and $N_\mathrm{p} = 2, 3, 4, 10, 25$.

Example 2 (Influence of N_c):
Figure 3 reveals the influence of N_c on the closed-loop behavior of system (16) with an MPC controller. $N_\mathrm{c} = 1$ leads to an unstable behavior, $N_\mathrm{c} = 2$ gives a sluggish output response, and for $N_\mathrm{c} \geqslant 3$ (= the 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. ($\lambda = 0$ and $\lambda = 1$ lead to uniqueness problems and are not computed).

Example 4 (Influence of $r(k)$):
Figure 5 shows the tracking error of system (15) for the input signals $r_1(k)$, $r_2(k)$ and $r_3(k)$. Note that $r_1(k)$ has an asymptotic slope $\rho_r = 4.5 < \rho = 4.75$. The due date-schedule is too tight and the delays grow unbounded. The asymptotic slopes of $r_2(k)$ and $r_3(k)$

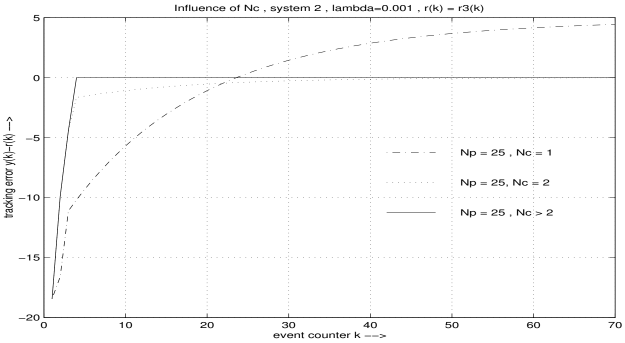

Figure 3: Influence of $N_{\rm c}$.

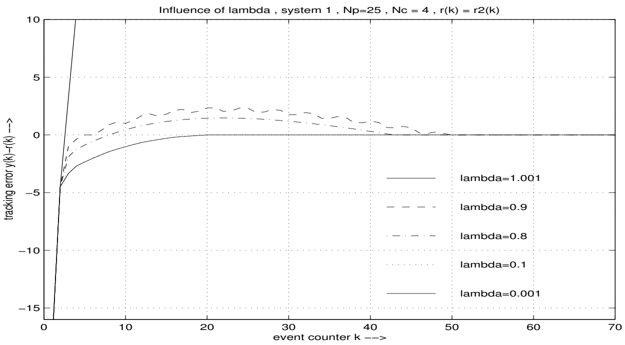

Figure 4: Influence of λ.

are larger than 4.75. We see that the tracking error $|y(k) - r(k)|$ becomes zero for large k.

6 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.

Initial settings for the parameters ($N_{\rm p}$,$N_{\rm c}$,λ) were given and the influence of the due date $r(k)$) was studied. Appropriate choices result in a stabilizing and effective MPC-control law. In practical industrial situations, the 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 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 LTI MPC). The issue of stability has been discussed and we considered the relation between buffer overflow and the settings of the parameters. Based on closed-loop aspects and some illustrative examples, we have derived some guidelines for the settings of tuning parameters and due dates.

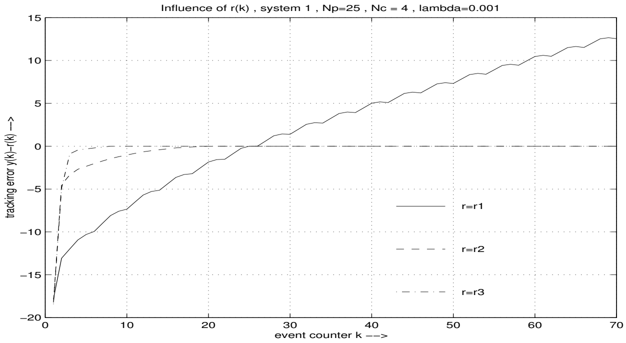

Figure 5: Influence of the due date $r(k)$.

References

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

[2] 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.

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

[4] 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.

[5] B. De Schutter and T. van den Boom, “Model predictive control for max-plus-linear discrete event systems,” Tech. rep. bds:99-10, Control Laboratory, Fac. of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, June 2000. Revised version. Accepted for publication in *Automatica*, vol. 37, no. 7, July 2001.

[6] 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.

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

[8] K.M. Passino and K.L. Burgess, *Stability Analysis of Discrete Event Systems.* New York: John Wiley & Sons, 1998.

[9] 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.