Delft University of Technology

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

Technical report bds:99-10a

Model predictive control for max-plus-linear discrete-event systems: Extended report & Addendum*

B. De Schutter and T. van den Boom

November 2000

A short version of this report has been published in *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 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`

*This report can also be downloaded via `http://pub.deschutter.info/abs/99_10a.html`

Model predictive control for max-plus-linear discrete-event systems: Extended report & Addendum*

Bart De Schutter and Ton van den Boom

Control Laboratory, Faculty of Information Technology and Systems
Delft University of Technology, P.O.Box 5031, 2600 GA Delft, The Netherlands
fax: +31-15-278.66.79, email: {b.deschutter,t.j.j.vandenboom}@its.tudelft.nl

Abstract

Model predictive control (MPC) is a very popular controller design method in the process industry. A key advantage of MPC is that it can accommodate constraints on the inputs and outputs. Usually MPC uses linear discrete-time models. In this report we extend MPC to a class of discrete-event systems that can be described by models that are "linear" in the max-plus algebra, which has maximization and addition as basic operations. In general the resulting optimization problem are nonlinear and non-convex. However, if the control objective and the constraints depend monotonically on the outputs of the system, the model predictive control problem can be recast as problem with a convex feasible set. If in addition the objective function is convex, this leads to a convex optimization problem, which can be solved very efficiently.

1 Introduction

Process industry is characterized by always tighter product quality specifications, increasing productivity demands, new environmental regulations and fast changes in the economical market. In the last decades Model Predictive Control (MPC), has shown to respond in an effective way to these demands in many practical process control applications and is therefore widely accepted in the process industry. Control design techniques such as pole placement, LQG, H_2, H_∞, etc. yield optimal controllers or control input sequences for the entire future evolution of the system. However, extending these methods to include additional constraints on the inputs and outputs is not easy. An important advantage of MPC is that the use of a finite horizon allows the inclusion of such additional constraints.

Predictive control was pioneered simultaneously by Richalet *et al.* (Richalet *et al.*, 1978), and Cutler and Ramaker (Cutler and Ramaker, 1979). There are several reasons why currently MPC is probably the most applied advanced control technique in the process industry:

*This report is an extended version of the paper "Model predictive control for max-plus-linear discrete-event systems" by B. De Schutter and T. van den Boom, *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001. Equation numbers and other labels that have an asterisk as a superscript or that are preceded by an A belong to the extended report and the addendum respectively and do not appear in the published journal paper.

- MPC is a model based controller design procedure that can easily handle processes with large time-delays, non-minimum phase processes and unstable processes.
- It is an easy-to-tune method: in principle there are only three parameters to be tuned.
- Industrial processes have their limitations due to technological requirements and are supposed to deliver output products with some pre-specified quality specifications. Furthermore, in many control applications signal constraints are present, caused by limited capacity of liquid buffers, valves, saturation of actuators, etc. MPC can handle these constraints in a systematic way during the design and implementation of the controller.
- Finally, MPC can handle structural changes, such as sensor or actuator failures and changes in system parameters or system structure, by adapting the model.

Traditionally MPC uses linear discrete-time models for the process to be controlled. In this report we extend and adapt the MPC framework to a class of discrete-event systems. 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 (e.g. machines, communications channels, or processors) that are shared by several users (e.g. product types, information packets, or jobs) all of which contribute to the achievement of some common goal (e.g. the assembly of products, the end-to-end transmission of a set of information packets, or a parallel computation) (Baccelli *et al.*, 1992).

There are many modeling techniques for discrete-event systems, such as (extended) state machines, max-plus algebra, formal languages, automata, temporal logic, generalized semi-Markov processes, Petri nets, computer simulation models and so on (see (Baccelli *et al.*, 1992; Cassandras *et al.*, 1995; Ho, 1989; Ho, 1992) and the references cited therein). 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 (Baccelli *et al.*, 1992; Cuninghame-Green, 1979). 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 this report we will develop an MPC framework for max-plus-linear discrete-event systems. Several other authors have already developed methods to compute optimal control sequences for max-plus-linear discrete-event systems (Baccelli *et al.*, 1992; Boimond and Ferrier, 1996; Cuninghame-Green, 1979; Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*a*; Menguy *et al.*, 1998*b*). The main advantage of our approach is that it allows to include general linear inequality constraints on the inputs and outputs of the system.

The report is organized as follows. In Section 2 we give a brief introduction to conventional MPC for linear discrete-time systems. In Section 3 we present the max-plus algebra and max-plus-linear discrete-event systems. Next we extend the MPC framework to max-plus-linear systems. In Section 5 we discuss some methods to solve the max-plus-algebraic MPC problem. We conclude with an illustrative example.

2 Model predictive control

In this section we give a short introduction to MPC. Since we will only consider the deterministic, i.e. noiseless, case for max-plus-linear systems (cf. Remark 1), we will also omit the noise terms in this brief introduction to MPC for linear systems. More extensive information on MPC can be found in (Camacho and Bordons, 1995; Clarke *et al.*, 1987; García *et al.*, 1989). and the references therein.

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

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

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

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 (1) − (2) from the max-plus-linear systems that will be considered later on, a system that can be modeled by (1) − (2) 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}):

$$\begin{aligned} J &= J_{\text{out}} + \lambda J_{\text{in}} \\ &= \sum_{j=1}^{N_{\text{p}}} \big(\hat{y}(k+j|k) - r(k+j)\big)^T \big(\hat{y}(k+j|k) - r(k+j)\big) \ + \\ &\qquad \lambda \sum_{j=1}^{N_{\text{p}}} u^T(k+j-1)u(k+j-1) \end{aligned} \tag{3}$$

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: $u(k+j) = u(k+N_{\text{c}}-1)$ for $j = N_{\text{c}}, \ldots, N_{\text{p}}-1$ 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. At time step k the future control sequence $u(k)$, …, $u(k+N_{\text{c}}-1)$ is determined such that the cost criterion is minimized subject to the constraints. At time step k the first element of the optimal sequence ($u(k)$) is applied to the process. At the next time instant the horizon is shifted, 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 (1) in (2), estimates of the future values of the state and the output can be computed (Camacho and Bordons, 1995):

$$\hat{x}(k+j|k) = A^j x(k) + \sum_{i=0}^{j-1} A^{j-i} B u(k+i)$$

$$\hat{y}(k+j|k) = CA^j x(k) + \sum_{i=0}^{j-1} CA^{j-i} Bu(k+i)$$

for $j = 1, 2, \ldots$, where $\hat{x}(k+j|k)$ is the estimate of the state at time step $k+j$ based on the information available at time step k. In matrix notation we obtain:

$$\tilde{y}(k) = H\tilde{u}(k) + g(k)$$

for

$$\tilde{y}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_\mathrm{p}|k) \end{bmatrix}, \ \tilde{r}(k) = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N_\mathrm{p}) \end{bmatrix}, \ \tilde{u}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_\mathrm{p}-1) \end{bmatrix} \tag{4}$$

and

$$H = \begin{bmatrix} CB & 0 & \ldots & 0 \\ CAB & CB & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_\mathrm{p}-1} & CA^{N_\mathrm{p}-2}B & \ldots & CB \end{bmatrix}, \quad g(k) = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_\mathrm{p}} \end{bmatrix} x(k) \ ,$$

where H is called the predictor matrix and $g(k)$ the free-run output signal. The cost criterion is now equal to

$$\begin{aligned} J &= \big(\tilde{y}(k) - \tilde{r}(k)\big)^T \big(\tilde{y}(k) - \tilde{r}(k)\big) + \lambda \tilde{u}^T(k)\tilde{u}(k) \\ &= \tilde{u}^T(k)(H^T H + \lambda I)\tilde{u}(k) + 2(g(k) - \tilde{r}(k))^T H\tilde{u}(k) + \\ &\quad (g(k) - \tilde{r}(k))^T (g(k) - \tilde{r}(k)) \ . \end{aligned}$$

So J is a quadratic function of $\tilde{u}(k)$.

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

> Find the input sequence $u(k), \ldots, u(k+N_\mathrm{c}-1)$ that minimizes the performance index J subject to the linear constraint
>
> $$E(k)\tilde{u}(k) + F(k)\tilde{y} \leqslant h(k) \tag{5}$$
>
> with $E(k) \in \mathbb{R}^{p \times mN_\mathrm{p}}$, $F(k) \in \mathbb{R}^{p \times lN_\mathrm{p}}$, $h(k) \in \mathbb{R}^p$ for some integer p, and where the inequality holds componentwise, and subject to the control horizon constraint
>
> $$u(k+j) = u(k+N_\mathrm{c}-1) \qquad \text{for } j = N_\mathrm{c}, N_\mathrm{c}+1, \ldots \tag{6}$$

Note that minimizing J subject to the linear constraints (5) and (6), boils down to a convex quadratic programming problem, which can be solved very efficiently.

The parameters N_p, N_c and λ are the three basic tuning parameters of the MPC algorithm:

- The prediction horizon N_p is related to the length of the step response of the process, and 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. An important effect of a small control horizon is the smoothing of the control signal. The control signal is then rapidly forced towards its steady-state value, which is important for stability properties. Another important consequence of decreasing N_c is the reduction in computational effort, because the number of optimization parameters is reduced.
- The parameter λ makes a trade-off between the tracking error and the control effort, and 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.

3 Max-plus algebra and max-plus-linear systems

3.1 Max-plus algebra

The basic operations of the max-plus algebra 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 structure $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ is called the max-plus algebra (Baccelli *et al.*, 1992; Cuninghame-Green, 1979). 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 (Baccelli *et al.*, 1992; Cuninghame-Green, 1979)). The max-plus-algebraic summation of a finite sequence of numbers $a_1, a_2, \ldots, a_m \in \mathbb{R}_\varepsilon$ is defined by

$$\bigoplus_{k=1}^{m} a_k = a_1 \oplus a_2 \oplus \ldots \oplus a_m = \max_{k=1,\ldots,m} a_k \ .$$

Let $k \in \mathbb{N}$. The kth max-plus-algebraic power of $x \in \mathbb{R}$ is denoted by $x^{\otimes^k}$ and corresponds to kx in conventional algebra. If $k > 0$ then $\varepsilon^{\otimes^k} = \varepsilon$. We have $\varepsilon^{\otimes^0} = 0$ by definition. 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 matrix $\mathcal{E}_{m\times n}$ is 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: $(E_n)_{ii} = 0$ for all i and $(E_n)_{ij} = \varepsilon$ for all i, j with $i \neq j$. 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

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

for all i, j. Note the analogy with the definitions of matrix sum and matrix product in conventional linear algebra. 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 $k = 1, 2, \ldots$

3.2 Max-plus-linear systems

In (Baccelli *et al.*, 1992; Cohen *et al.*, 1985; Cuninghame-Green, 1979) 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:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k) \quad (7)$$

$$y(k) = C \otimes x(k) \quad (8)$$

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 (7) – (8) with the state space model (1) – (2) for PTL systems. This analogy is another reason why the symbols $\oplus$ and $\otimes$ are used to denoted max and +. However, an important difference with the description (1) – (2) is that now the components of the input, the output and the state are event times, and that the counter k in (7) – (8) is an event counter (and event occurrence instants are in general not equidistant), whereas in (1) – (2) k increases each clock cycle.

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. A discrete-event system that can be modeled by (7) – (8) will be called a max-plus-linear time-invariant discrete-event system or *max-plus-linear* (MPL) system for short. For discrete event systems we usually do not have a direct feed-through from input to output. That is why there is no term of the form $D \otimes u(k)$ in (8).

Remark 1 For PTL systems the influence of noise is usually modeled by adding an extra noise term to the state equation and/or the output equation. For MPL models the entries of the system matrices correspond to production times, waiting times, transportation times, and so on. So instead of modeling the noise or the nondeterministic effects, i.e. the variation in the processing times, by adding an extra max-plus-algebraic term in (7) or (8), the variation in the processing times should rather be modeled as an additive term to these system matrices. However, this would not lead to a nice model structure. Moreover, for many manufacturing or transportation systems, the processing times are usually rather constant over time (or otherwise we could consider the nominal behavior of the system).
In the next section we will use the deterministic model (7) – (8) 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 (7) – (8). $\diamond$

4 Model predictive control for MPL systems

In this section we extend and adapt the MPC framework from PTL systems to MPL systems. If possible we use analog constraints and cost criteria for both types of systems. However, as we shall see, in some cases different constraints and cost criteria are more appropriate. We use the max-plus-linear model (7) – (8) 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 measurements[1]. 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 $u(k), \ldots, u(k+N_\mathrm{p}-1)$ using the model (7) – (8) as follows:

$$\hat{x}(k+j|k) = A \otimes \hat{x}(k+j-1|k) \oplus B \otimes u(k+j-1) \tag{8*a}$$

$$\hat{y}(k+j|k) = C \otimes \hat{x}(k+j|k) \tag{8*b}$$

for $j = 1, 2, \ldots, N_\mathrm{p}$. Note that $\hat{x}(k-1|k) = x(k-1)$. Just as in MPC for linear systems, we can eliminate the state estimates from (8*a) – (8*b) in order to reduce the number of variables in the resulting optimization problem, leading to faster computation of the optimal MPC policy. We have

$$\hat{y}(k+j|k) = C \otimes A^{\otimes^{j}} \otimes x(k) \;\oplus\; \bigoplus_{i=0}^{j-1} C \otimes A^{\otimes^{j-i}} \otimes B \otimes u(k+i) \;. \tag{8*c}$$

So if we define

$$H = \begin{bmatrix} C \otimes B & \mathcal{E} & \ldots & \mathcal{E} \\ C \otimes A \otimes B & C \otimes B & \ldots & \mathcal{E} \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}-1}} \otimes B & C \otimes A^{\otimes^{N_\mathrm{p}-2}} \otimes B & \ldots & C \otimes B \end{bmatrix},$$

$$g(k) = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes^{2}} \\ \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}}} \end{bmatrix} \otimes x(k)\,,$$

then we obtain

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$$

where $\tilde{y}(k)$ and $\tilde{u}(k)$ are defined by (4). Note the analogy between these expressions and the corresponding expressions for PTL systems.

4.2 Cost criterion

Recall that the MPC cost criterion for PTL systems can be written as $J = J_\mathrm{out} + \lambda J_\mathrm{in}$, where J_out is related to the tracking error and J_in is related to the control effort. Now we discuss some possible choices for J_out and J_in in MPC for MPL systems.

[1] Since 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.

4.2.1 Tracking error or output cost criterion J_{out}

A straightforward translation of the tracking error cost criterion used in MPC for PTL systems would yield

$$\begin{aligned} J_{\text{out}} &= \big(\tilde{y}(k) - \tilde{r}(k)\big)^T \otimes \big(\tilde{y}(k) - \tilde{r}(k)\big) \\ &= \bigoplus_{j=1}^{N_{\text{p}}} \big(\hat{y}(k+j|k) - r(k+j)\big)^T \otimes \big(\hat{y}(k+j|k) - r(k+j)\big) \\ &= \bigoplus_{j=1}^{N_{\text{p}}} \bigoplus_{i=1}^{l} \big(\hat{y}_i(k+j|k) - r_i(k+j)\big)^{\otimes^2} \\ &= 2 \bigoplus_{j=1}^{N_{\text{p}}} \bigoplus_{i=1}^{l} \big(\hat{y}_i(k+j|k) - r_i(k+j)\big) \ . \end{aligned} \tag{8*d}$$

This objective function does not force the difference between $\hat{y}(k+j|k)$ and $r(k+j)$ to be small since there is no absolute value in (8*d). Therefore, it is not very useful in practice.

If the due dates r for the finished products are known and if we have to pay a penalty for every delay, a better suited cost criterion is the tardiness:

$$J_{\text{out},1} = \sum_{j=1}^{N_{\text{p}}} \sum_{i=1}^{l} \max(\hat{y}_i(k+j|k) - r_i(k+j), 0) \ . \tag{9}$$

If we have perishable goods, then we could want to minimize the differences between the due dates and the actual output time instants. This leads to

$$J_{\text{out},2} = \sum_{j=1}^{N_{\text{p}}} \sum_{i=1}^{l} |\hat{y}_i(k+j|k) - r_i(k+j)| \ . \tag{10}$$

If we want to balance the output rates, we could consider the following cost criterion:

$$J_{\text{out},3} = \sum_{j=2}^{N_{\text{p}}} \sum_{i=1}^{l} |\Delta^2 \hat{y}_i(k+j|k)| \tag{11}$$

where $\Delta^2 s(k) = \Delta s(k) - \Delta s(k-1) = s(k) - 2s(k-1) + s(k-2)$ for a signal $s(\cdot)$.

The dynamics of the MPL system force all events to occur as soon as possible, i.e. as soon as all prerequisite conditions are fulfilled. Furthermore, the max-plus-algebraic matrix product is monotonically nondecreasing, i.e. if $u \leqslant \bar{u}$ then $A \otimes u \leqslant A \otimes \bar{u}$. So we could say that minimizing an expression of $\tilde{x}(k)$ corresponds to minimizing a modified expression of $\tilde{u}(k)$. Therefore, a cost criterion based on $\tilde{x}(k)$ is not as relevant as for MPL systems as it is for PTL systems.

4.2.2 Input cost criterion J_{in}

A straightforward translation of the input cost criterion $\tilde{u}^T(k)\tilde{u}(k)$ would lead to a minimization of the input time instants. Since this could result in input buffer overflows, a better objective is to *maximize* the input time instants. 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. Note that this also leads to a notion of stability if we let instability for the manufacturing system correspond to internal buffer overflows. So for MPL systems an appropriate cost criterion is

$$J_{\mathrm{in},0} = -\tilde{u}^T(k)\tilde{u}(k) \ .$$

Note that this is exactly the opposite of the input effort cost criterion for PTL systems. Another objective function that leads to a maximization of the input time instants is

$$J_{\mathrm{in},1} = -\sum_{j=1}^{N_\mathrm{p}} \sum_{i=1}^{m} u_i(k+j-1) \ . \tag{12}$$

If we want to balance the input rates we could take

$$J_{\mathrm{in},2} = \sum_{j=1}^{N_\mathrm{p}-1} \sum_{i=1}^{l} |\Delta^2 u_i(ik+j| \ . \tag{13}$$

Note that for the input cost criteria defined above we could replace the upper summation index N_p by N_c (for $J_{\mathrm{in},1}$ and $J_{\mathrm{in},2}$) or redefine $\tilde{u}(k)$ accordingly (for $J_{\mathrm{in},0}$).

We could replace also the summations in (9) – (13) by max-plus-algebraic summations:

$$\begin{aligned} J_{\mathrm{out},11} &= \bigoplus_{j=1}^{N_\mathrm{p}} \bigoplus_{i=1}^{l} \max(\hat{y}_i(k+j|k) - r_i(k+j), 0) \\ J_{\mathrm{out},21} &= \bigoplus_{j=1}^{N_\mathrm{p}} \bigoplus_{i=1}^{l} |\hat{y}_i(k+j|k) - r_i(k+j)| \end{aligned}$$

and so on. Moreover, we could also add some weight factors to the terms of the cost criterion, or consider weighted mixtures of several cost criteria.

4.3 Constraints

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

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

Furthermore, it is easy to verify that typical constraints for discrete-event systems are minimum or maximum separation between input and output events:

$$a_1(k+j) \leqslant \Delta u(k+j-1) \leqslant b_1(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{c} \tag{15}$$

$$a_2(k+j) \leqslant \Delta \hat{y}(k+j|k) \leqslant b_2(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{p}\,, \tag{16}$$

or maximum due dates for the output events:

$$\hat{y}(k+j|k) \leqslant r(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{p}\,, \tag{17}$$

can also be recast as a linear constraint of the form (14).

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[2]

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

to guarantee that the input sequences are nondecreasing. If the input sequences are nondecreasing and if the finite entries of the system matrices are nonnegative (which may be assumed to be the case for physical systems since these entries correspond to execution or transportation times), then it follows from (8*c) that the output sequences are also nondecreasing.

4.4 The evolution of the input beyond the control horizon

Recall that in MPC for PTL systems we introduced a control horizon to reduce the number of variables in the MPC optimization problem. For the same reason we also use a control horizon in MPC for MPL systems.

A straightforward translation of the conventional control horizon constraint would imply that the input should stay constant from event step $k + N_{\mathrm{c}}$ on, which is not very useful for MPL systems since there the input sequences should normally be increasing. Therefore, we change this condition as follows: the feeding rate should stay constant beyond event step $k + N_{\mathrm{c}}$, i.e.

$$\Delta u(k+j) = \Delta u(k + N_{\mathrm{c}} - 1) \qquad \text{for } j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}} - 1\,, \tag{18}$$

or equivalently $\Delta^2 u(k+j) = 0$ for $j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}} - 1$. Note that this condition introduces regularity in the input sequence. Furthermore, it prevents the buffer overflow problems that could arise when all resources are fed to the system at the same time instant as would be implied by the conventional control horizon constraint (6).

Recall that MPC uses a receding horizon strategy. This implies that, although (18) will hold for the optimal input sequence at each event step k, in closed loop the feeding rate will not necessarily become constant after N_{c} steps.

4.5 The standard MPC problem for MPL systems

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

$$\min_{\tilde{u}(k)} J = \min_{\tilde{u}(k)} J_{\mathrm{out},p_1} + \lambda J_{\mathrm{in},p_2} \tag{19}$$

for some $J_{\mathrm{out},p_1}, J_{\mathrm{in},p_2}$ subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{20}$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) \tag{21}$$

$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_{\mathrm{p}} - 1 \tag{22}$$

$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}} - 1 \tag{23}$$

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

[2] Note that in fact the upper range limit $N_{\mathrm{p}} - 1$ can be replaced by $N_{\mathrm{c}} - 1$ because of (18).

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

Other design control design methods for MPL systems are discussed in (Baccelli *et al.*, 1992; Boimond and Ferrier, 1996; Cuninghame-Green, 1979; Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*a*; Menguy *et al.*, 1998*b*). Note that in contrast to the MPL-MPC method, these methods do not allow the inclusion of general linear constraints of the form (21) or even more simple constraints of the form (15) or (16).

5 Algorithms to solve the MPL-MPC problem

5.1 Nonlinear optimization

In general the problem (19) – (23) is a nonlinear non-convex optimization problem: although the constraints (21) – (23) are convex in $\tilde{u}$ and $\tilde{y}$, the constraint (20) is in general not convex. So we could use standard multi-start nonlinear non-convex local optimization methods to compute the optimal control policy.

The feasibility of the MPC-MPL problem can be verified by solving the system of (in)equalities (20) – (23)[3]. If the problem is found to be infeasible we can use the same techniques as in conventional MPC and use constraint relaxation (Camacho and Bordons, 1995). Additional information on these topics can be found in the Addendum.

5.2 The ELCP approach

Now we discuss an alternative approach which is based on the Extended Linear Complementarity problem (ELCP) (De Schutter and De Moor, 1995). Consider the ith row of (20) and define $\mathcal{J}_i = \{ j \mid h_{ij} \neq \varepsilon \}$. We have

$$\tilde{y}_i(k) = \max_{j \in \mathcal{J}_i} \left(h_{ij} + \tilde{u}_j(k), g_i(k) \right)$$

or equivalently

$$\begin{aligned} \tilde{y}_i(k) &\geqslant h_{ij} + \tilde{u}_j(k) \qquad \text{for } j \in \mathcal{J}_i \\ \tilde{y}_i(k) &\geqslant g_i(k) \end{aligned}$$

with the extra condition that at least one inequality should hold with equality (i.e. at least one residue should be equal to 0)[4]:

$$\left(\tilde{y}_i(k) - g_i(k) \right) \cdot \prod_{j \in \mathcal{J}_i} \left(\tilde{y}_i(k) - h_{ij} - \tilde{u}_j(k) \right) = 0 \ . \tag{24}$$

Hence, equation (20) can be rewritten as a system of equations of the following form:

$$A_e \tilde{y}(k) + B_e \tilde{u}(k) + c_e(k) \geqslant 0 \tag{25}$$

[3] In general this is a nonlinear system of equations but if the constraints depend monotonically on the output, the feasibility problem can be recast as a linear programming problem. (cf. Theorem 2).

[4] If $g_i(k) = \varepsilon$ then the inequality $\tilde{y}_i(k) \geqslant g_i$ and the corresponding factor $\tilde{y}_i(k) - g_i(k)$ in (24) may be omitted. A similar remark holds if $h_{ij} = \varepsilon$.

$$\prod_{j\in\phi_i} (A_{\rm e}\tilde{y}(k) + B_{\rm e}\tilde{u}(k) + c_{\rm e}(k))_j = 0 \qquad \text{for } i = 1,\ldots,lN_{\rm p} \tag{26}$$

for appropriately defined matrices and vectors $A_{\rm e}, B_{\rm e}, c_{\rm e}$ and index sets ϕ_i. We can rewrite the linear constraints (21) – (23) as

$$D_{\rm e}(k)\tilde{y}(k) + E_{\rm e}(k)\tilde{u}(k) + f_{\rm e}(k) \geqslant 0 \tag{27}$$

$$G_{\rm e}\tilde{u}(k) + h_{\rm e} = 0 \ . \tag{28}$$

So the feasible set of the MPC problem (i.e. the set of feasible system trajectories) coincides with the set of solutions of the system (25) – (28), which is a special case of an Extended Linear Complementarity Problem (ELCP) (De Schutter and De Moor, 1995).

In order to determine the optimal MPC policy we have to minimize the objective function J over the solution set of the ELCP (25) – (28) as follows. The solution set of the system (25) – (28) is characterized by a set of vectors

$$V = \left\{ \left[\begin{array}{c} \tilde{u}^i(k) \\ \tilde{y}^i(k) \end{array} \right] \middle| \, i = 1,\ldots,r \right\}$$

and a set $\Lambda = \{\psi_j \,|\, j = 1,\ldots,p\}$ such that for any j any convex combination of the form

$$\sum_{i\in\psi_j} \nu_i \left[\begin{array}{c} \tilde{u}^i(k) \\ \tilde{y}^i(k) \end{array} \right]$$

with $\nu_i \geqslant 0$ and $\sum_{i\in\phi_j} \nu_i = 1$, is a solution of (25) – (28). The vectors of V correspond to vertices of the polyhedron P defined by the system (25) (27) (28), and each index set ψ_j corresponds to a face of this polyhedron. Now we could use a nonlinear optimization algorithm to determine for each index set ψ_j the combination of the ν_i's for which the objective function J reaches a global minimum and afterwards select the overall minimum.

The algorithm of (De Schutter and De Moor, 1995) to compute the solution set of a general ELCP requires exponential execution times. This implies that the ELCP approach sketched above is not feasible if $N_{\rm c}$ is large.

5.3 Monotonically nondecreasing objective functions

Now we consider the *relaxed* MPC problem which is also defined by (19) – (23) but with the =-sign in (20) replaced by a $\geqslant$-sign. Note that whereas in the original problem $\tilde{u}(k)$ is the only independent variable since $\tilde{y}(k)$ can be eliminated using (20), the relaxed problem has both $\tilde{u}(k)$ and $\tilde{y}(k)$ as independent variables. It is easy to verify that the set of feasible solutions of the relaxed problem coincides with the set of solutions of the system of linear inequalities (25) (27) (28). So the set of feasible solutions of the relaxed MPC problem is convex. As a consequence, the relaxed problem is much easier to solve numerically. Also note that due to condition (26) the feasible set of the original problem consists of points lying on the border of the feasible set of the relaxed problem.

We say that a function $F : y \rightarrow F(y)$ is a monotonically nondecreasing function of y if $\bar{y} \leqslant \check{y}$ implies that $F(\bar{y}) \leqslant F(\check{y})$. Now we show that if the objective function J and the linear constraints are monotonically nondecreasing as a function of $\tilde{y}$ (this is the case for $J = J_{\rm out,1}$,

$J_{\text{out},11}$, $J_{\text{out},12}$, $J_{\text{in},0}$, $J_{\text{in},1}$, $J_{\text{in},2}$, $J_{\text{in},21}$ or $J_{\text{in},22}$, and e.g. $F_{ij} \geqslant 0$ for all i, j), then the optimal solution of the relaxed problem can be transformed into an optimal solution of the original MPC problem. So in that case the optimal MPC policy can be computed very efficiently. If in addition the objective function is convex (e.g. $J = J_{\text{out},1}$, $J_{\text{out},11}$, $J_{\text{out},12}$ or $J_{\text{in},1}$), we finally get a convex optimization problem. Note that $J_{\text{in},1}$ is a linear function. So for $J = J_{\text{in},1}$ the problem even reduces to a linear programming problem, which can be solved very efficiently[5].

Theorem 2 *Let the objective function J and mapping $\tilde{y} \to F(k)\tilde{y}$ be monotonically nondecreasing functions of $\tilde{y}$. Let $(\tilde{u}^*, \tilde{y}^*)$ be an optimal solution of the relaxed MPC problem. If we define $\tilde{y}^\sharp = H \otimes \tilde{u}^* \oplus g(k)$ then $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MPC problem.*

Proof: First we show that $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the original problem. Clearly, $(\tilde{u}^*, \tilde{y}^\sharp)$ satisfies the constraints (20), (22) and (23). Since $\tilde{y}^* \geqslant H \otimes \tilde{u}^* \oplus g(k) = \tilde{y}^\sharp$ and since the mapping $\tilde{y} \to F(k)\tilde{y}$ is monotonically nondecreasing, we have

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

So $(\tilde{u}^*, \tilde{y}^\sharp)$ also satisfies the constraint (21). Hence, $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the original problem. Since the set of feasible solutions of the original problem is a subset of the set of feasible solutions of the relaxed problem, we have $J(\tilde{u}, \tilde{y}) \geqslant J(\tilde{u}^*, \tilde{y}^*)$ for any feasible solution $(\tilde{u}, \tilde{y})$ of the original problem. Hence, $J(\tilde{u}^*, \tilde{y}^\sharp) \geqslant J(\tilde{u}^*, \tilde{y}^*)$. On the other hand, we have $J(\tilde{u}^*, \tilde{y}^\sharp) \leqslant J(\tilde{u}^*, \tilde{y}^*)$ since $\tilde{y}^\sharp \leqslant \tilde{y}^*$ and since J is a monotonically nondecreasing function of $\tilde{y}$. As a consequence, we have $J(\tilde{u}^*, \tilde{y}^\sharp) = J(\tilde{u}^*, \tilde{y}^*)$, which implies that $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MPC problem. □

Note that we can always obtain an objective function that is a monotonically nondecreasing function of $\tilde{y}(k)$ by eliminating $\tilde{y}(k)$ from the expression for J using (20) before relaxing the problem. However, some of the properties (such as convexity or linearity) of the original objective function may be lost in that way. Furthermore, if the objective function is not monotonically nondecreasing as a function of $\tilde{y}(k)$, then it is still possible to use the solution of the relaxed problem (with another objective function that is monotonically nondecreasing) as an efficient way to determine a good initial solution for an iterative optimization algorithm for the original (non-convex) MPC problem.

6 Example

Consider the production system of Figure 1. This manufacturing system consists of three processing units: P_1, P_2 and P_3, and works in batches (one batch for each finished product). Raw material is fed to P_1 and P_2, processed and sent to P_3 where assembly takes place. Note that each input batch of raw material is split into two parts: one part of the batch goes to P_1 and the other part goes to P_2.

The processing times for P_1, P_2 and P_3 are respectively $d_1 = 11$, $d_2 = 12$ and $d_3 = 7$ time units. We assume that it takes $t_1 = 2$ time units for the raw material to get from the input source to P_1, and $t_3 = 1$ time unit for a finished product of P_1 to get to P_3. The other transportation times (t_2, t_4 and t_5) and the set-up times are assumed to be negligible. At the

[5]It easy to verify that for $J = J_{\text{out},1}$, $J_{\text{out},11}$, $J_{\text{out},12}$ the problem can also be reduced to a linear programming problem by introducing some additional dummy variables.

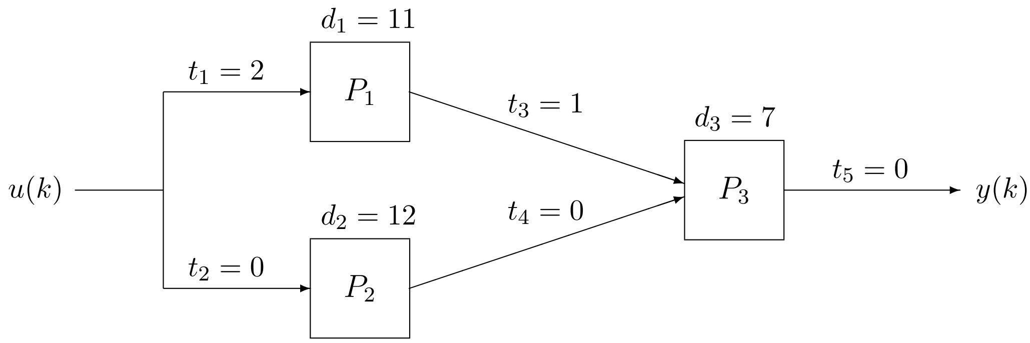

Figure 1: A simple manufacturing system.

input of the system and between the processing units there are buffers with a capacity that is large enough to ensure that no buffer overflow will occur. A processing unit can only start working on a new product if it has finished processing the previous product. We assume that each processing unit starts working as soon as all parts are available.

The evolution of the manufacturing system is described by the following state space model (see Addendum):

$$x(k+1) = \begin{bmatrix} 11 & \varepsilon & \varepsilon \\ \varepsilon & 12 & \varepsilon \\ 23 & 24 & 7 \end{bmatrix} \otimes x(k) \oplus \begin{bmatrix} 2 \\ 0 \\ 14 \end{bmatrix} \otimes u(k) \tag{29}$$

$$y(k) = \begin{bmatrix} \varepsilon & \varepsilon & 7 \end{bmatrix} \otimes x(k) \tag{30}$$

with $u(k)$ the time instant at which a batch of raw material is fed to the system for the $(k+1)$th time, $x_i(k)$ the time instant at which P_i starts working for the kth time, and $y(k)$ the time instant at which the kth finished product leaves the system.

Let us now compare the efficiency of the methods discussed in Section 5 when solving one step of the MPC problem for the objective function $J = J_{\mathrm{out},1} + J_{\mathrm{in},1}$ (so $\lambda = 1$) with the additional constraints $2 \leqslant \Delta u(k+j) \leqslant 12$ for $j = 0, \ldots, N_{\mathrm{c}} - 1$. We take $N_{\mathrm{c}} = 5$ and $N_{\mathrm{p}} = 8$. Assume that $k = 0$, $x(0) = [\, 0 \;\; 0 \;\; 10 \,]^T$, $u(-1) = 0$, and $\tilde{r}(k) = [\, 40 \;\; 45 \;\; 55 \;\; 66 \;\; 75 \;\; 85 \;\; 90 \;\; 100 \,]^T$.

The objective function J and the linear constraints are monotonically nondecreasing as a function of $\tilde{y}$ so that we can apply Theorem 2. We have computed a solution $\tilde{u}_{\mathrm{elcp}}$ obtained using the ELCP method and the ELCP algorithm of (De Schutter and De Moor, 1995), a solution $\tilde{u}_{\mathrm{nlcon}}$ using nonlinear constrained optimization, a solution $\tilde{u}_{\mathrm{penalty}}$ using linearly constrained optimization with a penalty function for the nonlinear constraints, a solution $\tilde{u}_{\mathrm{relaxed}}$ for the relaxed MPC problem, and a linear programming solution $\tilde{u}_{\mathrm{linear}}$ (cf. footnote 5). For the nonlinear constrained optimization we have used a sequential quadratic programming algorithm, and for the linear optimization a variant of the simplex algorithm. All these methods result in the same optimal input sequence:

$$\{u_{\mathrm{opt}}\}_{k=0}^{7} = 12, 24, 35, 46, 58, 70, 82, 94.$$

The corresponding output sequence is $\{y_{\mathrm{opt}}(k)\}_{k=1}^{8} = 33, 45, 56, 67, 79, 91, 103, 115$ and the corresponding value of the objective function is $J = -381$.

In Table 1 we have listed the CPU time needed to compute the various input sequence vectors $\tilde{u}$ for $N_{\mathrm{c}} = 4, 5, 6, 7$ (with all other variables having the same values as above) on a

Table 1: The CPU time needed to compute the optimal input sequence vectors for the example of Section 6 for $N_c = 4, 5, 6, 7$. For $N_c = 7$ we have not computed the ELCP solution since it requires too much CPU time.

$\tilde{u}_{\text{opt}}$	CPU time			
	$N_c = 4$	$N_c = 5$	$N_c = 6$	$N_c = 7$
$\tilde{u}_{\text{elcp}}$	5.525	106.3	287789	—
$\tilde{u}_{\text{nlcon}}$	0.870	1.056	1.319	1.470
$\tilde{u}_{\text{penalty}}$	0.826	0.988	1.264	1.352
$\tilde{u}_{\text{relaxed}}$	0.431	0.500	0.562	0.634
$\tilde{u}_{\text{linear}}$	0.029	0.030	0.031	0.032

Pentium II 300 MHz PC running Linux with the optimization routines called from MATLAB and implemented in C. The CPU times are average values over 10 experiments. For the input sequence vectors that have to be determined using a nonlinear optimization algorithm selecting different (feasible) initial points always leads to the same numerical value of the final objective function (within a certain tolerance). Therefore, we have only performed one run with a random feasible initial point for each of these cases.

The CPU time needed to compute the optimal switching interval vector using the ELCP algorithm of (De Schutter and De Moor, 1995) increases exponentially as the number of variables increases (see also Table 1). So in practice the ELCP approach cannot be used for on-line computations if the control horizon or the number of inputs or outputs are large. In that case one of the other methods should be used instead. If we look at Table 1 then we see that the $\tilde{u}_{\text{linear}}$ solution — which is based on Theorem 2 and which uses a linear programming approach — is clearly the most interesting.

Let us now compare the MPC-MPL method with the other control design methods mentioned in Section 4.5.

In the Addendum we will use results from (Baccelli *et al.*, 1992; Cuninghame-Green, 1979) to derive an analytic solution for two special cases of the MPL-MPC problem[6]. If we use these analytic solutions we obtain

$$\begin{aligned}
\{u^1(k)\}_{k=0}^{7} &= -3, 9, 21, 33, 45, 57, 68, 79 \\
\{y^1(k)\}_{k=1}^{8} &= 31, 43, 55, 67, 79, 91, 103, 115 \\
\{u^2(k)\}_{k=0}^{7} &= 8, 20, 32, 44, 56, 68, 79, 90 \\
\{y^2(k)\}_{k=1}^{8} &= 31, 43, 55, 67, 79, 91, 103, 115.
\end{aligned}$$

The first solution is not feasible since for this solution we have $u(0) = -3 < 0 = u(-1)$. This infeasibility is caused by the fact that the solution aims to fulfill the constraint $\tilde{y}(k) \leqslant$

[6]Case 1: $\min J_{\text{out},11}$ subject to $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$ and $\tilde{y}(k) \leqslant \tilde{r}(k)$; Case 2: $\min J_{\text{out},21}$ subject to $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$. In both cases we have $N_p = N_c$ and the input constraint (22) is *not* taken into account.

$\tilde{r}(k)$, which cannot be met using a nondecreasing input sequence. So other control design methods that also include this constraint such as (Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*b*) would also yield a nondecreasing — and thus infeasible — input sequence. The second solution $\{u^2(k)\}_{k=0}^{7}$ is feasible and the corresponding value of the objective function is $J = -358$.
The control design method of (Boimond and Ferrier, 1996) leads to

$$\begin{aligned} \{u^3(k)\}_{k=0}^{7} &= 19, 30, 41, 53, 65, 77, 89, 101 \\ \{y^3(k)\}_{k=1}^{8} &= 40, 51, 62, 74, 86, 98, 110, 122. \end{aligned}$$

but this input sequence does not satisfy the constraints since $\Delta u(0) = u(0) - u(-1) = 19 \not\leqslant 12$.
The method of (Menguy *et al.*, 1998*a*) results in

$$\begin{aligned} \{u^4(k)\}_{k=0}^{7} &= 12, 23, 34, 46, 58, 70, 82, 94 \\ \{y^4(k)\}_{k=1}^{8} &= 33, 44, 55, 67, 79, 91, 103, 115 \end{aligned}$$

with $J = -380$.

So for this particular case the MPC method and the method of (Menguy *et al.*, 1998*a*) outperform the other methods to compute (optimal) input time sequences for MPL systems that have been considered in this report. However, the method of (Menguy *et al.*, 1998*a*) does not take the input constraint $2 \leqslant \Delta u(k + j) \leqslant 12$ for $j = 0, 1, \ldots, 9$ into account so that in general this method will not always lead to a feasible solution. So in general the MPC-MPL method is the only method among the methods considered in this report that is guaranteed to yield a feasible solution (provided that one exists).

7 Conclusions

We have extended the popular MPC framework from linear discrete-time systems to max-plus-linear discrete event systems. The reason for using an MPC approach for max-plus-linear systems is the same as for conventional linear systems: MPC allows the inclusion of constraints on the inputs and outputs, it is an easy-to-tune method, and it is flexible for structure changes (since the optimal strategy is recomputed every time or event step so that model changes can be taken into account as soon as they are identified). Note that although in general the optimization may be complex and time-consuming and should be performed each event step, the inter-event times are usually sufficiently long so that the calculation can be performed on-line (especially if the objective function and the constraints are monotonically nondecreasing functions of the output and if the objective function is convex, since then the resulting (relaxed) optimization problem is convex).

We have discussed the analogies and differences between the objective functions and constraints in the conventional MPC problem and in the max-plus-algebraic MPC problem. We have also presented some methods to solve the max-plus-algebraic MPC problem. In general this leads to a nonlinear non-convex optimization problem. If the objective function and the constraints are nondecreasing functions of the output, then we can relax the MPC problem to problem with a convex set of feasible solutions. If in addition the objective function is convex, this leads to a problem that can be solved very efficiently.

Topics for future research include: investigation of issues (such as prediction) that arise when we consider MPC for nondeterministic max-plus-algebraic systems, investigation of the effects of the tuning parameters on the stability of the controlled systems and the smoothness of the input control sequence, and determination of rules of thumb for the selection of appropriate values for the tuning parameters.

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

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

Boimond, J.L. and J.L. Ferrier (1996). Internal model control and max-algebra: Controller design. *IEEE Transactions on Automatic Control* **41**(3), 457–461.

Camacho, E.F. and C. Bordons (1995). *Model Predictive Control in the Process Industry.* Springer-Verlag. Berlin, Germany.

Cassandras, C.G., S. Lafortune and G.J. Olsder (1995). Introduction to the modelling, control and optimization of discrete event systems. In: *Trends in Control: A European Perspective* (A. Isidori, Ed.). pp. 217–291. Springer-Verlag. Berlin, Germany.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalized predictive control – Part I. The basic algorithm. *Automatica* **23**(2), 137–148.

Cohen, G., D. Dubois, J.P. Quadrat and M. Viot (1985). 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.

Cuninghame-Green, R.A. (1979). *Minimax Algebra.* Vol. 166 of *Lecture Notes in Economics and Mathematical Systems.* Springer-Verlag. Berlin, Germany.

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

De Schutter, B. and B. De Moor (1995). The extended linear complementarity problem. *Mathematical Programming* **71**(3), 289–325.

García, C.E., D.M. Prett and M. Morari (1989). Model predictive control: Theory and practice — A survey. *Automatica* **25**(3), 335–348.

Ho, Y.C., Ed.) (1989). *Proceedings of the IEEE.* Vol. 77. Special Issue on Dynamics of Discrete Event Systems.

Ho, Y.C., Ed.) (1992). *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World.* IEEE Press. Piscataway, New Jersey.

Libeaut, L. and J.J. Loiseau (1995). Admissible initial conditions and control of timed event graphs. In: *Proceedings of the 34th IEEE Conference on Decision and Control*. New Orleans, Louisiana. pp. 2011–2016.

Menguy, E., J.L. Boimond and L. Hardouin (1997). A feedback control in max-algebra. In: *Proceedings of the European Control Conference (ECC'97)*. Brussels, Belgium, paper 487.

Menguy, E., J.L. Boimond and L. Hardouin (1998*a*). Adaptive control for linear systems in max-algebra. In: *Proceedings of the International Workshop on Discrete Event Systems (WODES'98)*. Cagliari, Italy. pp. 481–488.

Menguy, E., J.L. Boimond and L. Hardouin (1998*b*). Optimal control of discrete event systems in case of updated reference input. In: *Proceedings of the IFAC Conference on System Structure and Control (SSC'98)*. Nantes, France. pp. 601–607.

Olsder, G.J. and B. De Schutter (1999). The minimal realization problem in the max-plus algebra. In: *Open Problems in Mathematical Systems and Control Theory* (V.D. Blondel, E.D. Sontag, M. Vidyasagar and J.C. Willems, Eds.). Chap. 32, pp. 157–162. Springer-Verlag. London.

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

Model predictive control for max-plus-linear discrete event systems: Addendum

This addendum is organized as follows. In Section A we discuss how the feasibility of the MPL-MPC problem can be determined and what can be done if the problem is found to be infeasible. In Section B we derive an analytic expression for the solution of a max-plus-algebraic optimization problem. In Section C we consider two special cases of the MPL-MPC problem for which analytic solutions can be formulated. In the last section we give some additional information in connection with the worked example of Section 6.

A Feasibility and constraint relaxation

The existence of a solution of the MPL-MPC problem at event step k can be verified by solving the system of (in)equalities (20) – (23), which describes the feasible set of the problem. Since (20) is a nonlinear equation, in general this requires solving a system of nonlinear equations subject to linear constraints. However, if the entries of F are nonnegative, then we can use the result of Theorem 2 and replace (20) by the relaxed equation $\tilde{y}(k) \geqslant H \otimes \tilde{u}(k) \oplus g(k)$. So in that case the MPL-MPC problem is feasible if and only if the system of linear inequalities

$$
\begin{aligned}
&\tilde{y}_i(k) \geqslant h_{ij} + \tilde{u}_j(k) && \text{for } i = 1,\ldots,lN_\mathrm{p} \text{ and } j = 1,\ldots,mN_\mathrm{p}\\
&\tilde{y}_i(k) \geqslant g_i(k) && \text{for } i = 1,\ldots,lN_\mathrm{p}\\
&E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) &&\\
&\Delta u(k+j) \geqslant 0 && \text{for } j = 0,\ldots,N_\mathrm{p}-1\\
&\Delta^2 u(k+j) = 0 && \text{for } j = N_\mathrm{c},\ldots,N_\mathrm{p}-1
\end{aligned}
$$

is feasible. Indeed, the feasibility of the problem is independent of the choice of the objective function. So if we consider a dummy monotonic objective function (e.g. $J = J_{\mathrm{out},1}$ or $J_{\mathrm{in},1}$), then we can use Theorem 2 to recast the problem as a linear programming problem. Note that there exist very efficient algorithms to determine the feasibility of a system of linear inequalities.

If the MPC problem is found to be infeasible then we could use constraint relaxation (see (Camacho and Bordons, 1995) and the references therein), i.e. we relax some of the constraints in such a way that we obtain a feasible problem. This can be done as follows. The constraints $\tilde{y} = H \otimes \tilde{u}(k) \oplus g(k)$ and $\Delta u(k+j) \geqslant 0$ for $j = 0,\ldots,N_\mathrm{p}-1$ should always be satisfied because of their physical meaning. Furthermore, the constraint $\Delta^2 u(k+j) = 0$ for $j = N_\mathrm{c},\ldots,N_\mathrm{p}-1$ 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 $E(k)$. If J_orig is the original objective function of the (infeasible) MPL-MPC problem, then we introduce a vector $\nu \in \mathbb{R}^{n_E}$ of dummy variables and we solve the problem

$$\min_{\tilde{u}(k),\nu} \; J_{\text{orig}} + \sum_{i=1}^{n_E} \nu_i \tag{A.1}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{A.2}$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) + R\nu \tag{A.3}$$

$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_{\text{p}} - 1 \tag{A.4}$$

$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_{\text{c}}, \ldots, N_{\text{p}} - 1 \tag{A.5}$$

$$\nu \geqslant 0 \; . \tag{A.6}$$

This problem is feasible since the constraints can always met by making the components of the vector ν sufficiently large. Also note that inclusion of the term $\nu_1 + \cdots + \nu_{n_E}$ 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 2 (i.e. J_{orig} and the mapping $\tilde{y} \to F(k)\tilde{y}$ are monotonically nondecreasing functions of $\tilde{y}$) then the problem (A.1) – (A.6) also satisfies these conditions so that Theorem 2 still applies. Moreover, if the original objective function J_{orig} is convex or linear, then the same holds for the new objective function 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 non-convex. But if the constraints are monotonically nondecreasing as a function of $\tilde{y}(k)$ and if the objective function is strictly convex, then Theorem 2 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.

B The analytic solution of a special max-plus-algebraic optimization problem

In this section we determine an analytic solution of the following problem:

$$\text{Given } A \in \mathbb{R}_\varepsilon^{m \times n} \text{ and } b \in \mathbb{R}_\varepsilon^m, \text{ find a vector } x \in \mathbb{R}_\varepsilon^n \text{ that minimizes } \|b - A \otimes x\|_\infty \; , \tag{A.7}$$

where $\|x\|_\infty = \max_i |x_i|$. This problem and its solution will be used in Section C.2 to provide an analytic solution to a special case of the MPL-MPC problem. Now we will show that the following theorem holds:

Theorem A-1 *An optimal solution of Problem (A.7) is given by $x^\sharp = x^* \otimes \frac{\delta}{2}$, with x^* the greatest subsolution of $A \otimes x = b$ and $\delta = \|b - A \otimes x^*\|_\infty$. We have $\|b - A \otimes x^\sharp\|_\infty = \frac{\delta}{2}$.*

Before we prove this theorem we first make some remarks, and we state a lemma that will be used in the proof.

Note that $x^\sharp_j = \left(x^* \otimes \frac{\delta}{2}\right)_j = x^*_j + \frac{\delta}{2}$. The solution $x^\sharp = x^* \otimes \frac{\delta}{2}$ for Problem (A.7) is mentioned on p. 165 of (Cuninghame-Green, 1979) but not proven explicitly. Furthermore, it is easy to verify that in general the solution of Problem (A.7) is not unique since "non-critical" components of x^* do not have to be augmented by $\frac{\delta}{2}$.
Since $A \otimes x^* \leqslant b$ we have $\delta = \|b - A \otimes x^*\|_\infty = \max_i |(b - A \otimes x^*)_i| = \max_i \big((b - A \otimes x^*)_i\big)$, i.e., the absolute values are redundant.
The greatest subsolution x^* of $A \otimes x = b$ is given by (Baccelli *et al.*, 1992; Cuninghame-Green, 1979)

$$x^*_j = \min_{i=1,\ldots,m} (b_i - a_{ij}) \qquad \text{for } j = 1, \ldots, n\,.$$

Recall that the greatest subsolution of $A \otimes x = b$ is the solution of $A \otimes x \leqslant b$ for which all components are as large as possible without violating the constraint $A \otimes x \leqslant b$. Since the function $x \to A \otimes x$ is a monotonically nondecreasing function of x, this implies that the following lemma holds:

Lemma A-2 *Consider $A \in \mathbb{R}_\varepsilon^{m\times n}$ and $b \in \mathbb{R}_\varepsilon^m$, where A has a finite entry in each column. Let x^* be the greatest subsolution of $A \otimes x = b$. Then for each index $r \in \{1, 2, \ldots, n\}$ there exists an index $i_r \in \{1, 2, \ldots, m\}$ such that $a_{i_r r} + x^*_r = b_{i_r}$.*

Proof of Theorem A-1 : Clearly, δ is always nonnegative. Furthermore, if $\delta = 0$ then $x^\sharp = x^*$ and $\|b - A \otimes x^\sharp\|_\infty = 0$. Hence, $x^\sharp = x^*$ is an optimal solution of Problem (A.7) if $\delta = 0$. So from now on we assume that $\delta > 0$. Furthermore, we may assume without loss of generality that all columns of A that contain only ε entries are removed, because changes to the corresponding components of the x vector do not change the value of $A \otimes x$.

In the first step of the proof we set $x = x^* \otimes \alpha$ for some real number α, and we prove that $\|b - A \otimes (x^* \otimes \alpha)\|_\infty$ is minimized by $\alpha = \frac{\delta}{2}$. In the second step we prove that it is impossible to find another x that yields a smaller value of the objective function $\|b - (A \otimes x)\|_\infty$ than $x = x^\sharp = x^* \otimes \frac{\delta}{2}$.

Step 1

Define $x(\alpha) = x^* \otimes \alpha$ with $\alpha \in \mathbb{R}_\varepsilon$. Then $A \otimes x(\alpha) = A \otimes (x^* \otimes \alpha) = (A \otimes x^*) \otimes \alpha$.
We can partition $\{1, \ldots, m\}$ into three pairwise disjunct subsets $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$ such that:

$$\begin{aligned}
&(A \otimes x^*)_i = b_i && \text{for all } i \in \mathcal{I}\\
&(A \otimes x^*)_i = b_i - \delta && \text{for all } i \in \mathcal{J}\\
&(A \otimes x^*)_i = b_i - \xi_i\delta && \text{for all } i \in \mathcal{K} \text{ with } 0 < \xi_i < 1\,.
\end{aligned}$$

From Lemma A-2 it follows that $\mathcal{I}$ is non-empty. Since $\delta > 0$, the set $\mathcal{J}$ is also non-empty. Since $A \otimes x(\alpha) = (A \otimes x^*) \otimes \alpha$, we have

$$(A \otimes x(\alpha))_i - b_i = \begin{cases} \alpha & \text{if } i \in \mathcal{I} \\ \alpha - \delta & \text{if } i \in \mathcal{J} \\ \alpha - \xi_i\delta & \text{if } i \in \mathcal{K}\,. \end{cases}$$

Since $\mathcal{I}$ and $\mathcal{J}$ are not empty, and since $0 < \xi_i < 1$ for all $i \in \mathcal{K}$, this implies that

$$\|b - (A \otimes x(\alpha))\|_\infty = \max_i |b_i - (A \otimes x(\alpha))_i| = \max(|\alpha|, |\alpha - \delta|)\ .$$

Now it is easy to verify that $d(\alpha) \stackrel{\text{def}}{=} \max(|\alpha|, |\alpha - \delta|)$ is minimized for $\alpha = \frac{\delta}{2}$ and that $d(\frac{\delta}{2}) = \frac{\delta}{2}$. So for $x^\sharp = x(\frac{\delta}{2}) = x^* \otimes \frac{\delta}{2}$, we have $\|b - (A \otimes x^\sharp)\|_\infty = \frac{\delta}{2}$.

In the next step of the proof we will show that we cannot find another x for which $\|b - (A \otimes x)\|_\infty$ is less than $\frac{\delta}{2}$. This will be done by contradiction.

Step 2

Suppose that there exists a vector $\tilde{x}$ such that

$$\|b - (A \otimes \tilde{x})\|_\infty < \frac{\delta}{2} \ . \tag{A.8}$$

Define $\beta = \tilde{x} - x^*$. Note that β is a vector and not a number as α was in Step 1 of the proof. Hence, $A \otimes \tilde{x} = A \otimes (x^* + \beta)$. Consider an arbitrary $r \in \{1, 2, \ldots, n\}$. From Lemma A-2 it follows that exists an index i_r such that $a_{i_r r} + x_r^* = b_{i_r}$. Since $(A \otimes \tilde{x})_{i_r} \geqslant a_{i_r r} + x_r^* + \beta_r$, we have $(A \otimes \tilde{x})_{i_r} \geqslant b_{i_r} + \beta_r$. Hence, $\beta_r < \frac{\delta}{2}$ because of (A.8). Since r is an arbitrary index, this implies that

$$\beta_r < \frac{\delta}{2} \qquad \text{for each } r \in \{1, 2, \ldots, n\} \ . \tag{A.9}$$

From the definition of δ it follows that there exists an index $s \in \{1, 2, \ldots, m\}$ such that $(A \otimes x^*)_s = b_s - \delta$. Hence,

$$a_{sj} + x_j^* \leqslant b_s - \delta \qquad \text{for each } j \in \{1, 2, \ldots, n\} \ . \tag{A.10}$$

As a consequence, we have

$$\begin{aligned}
(A \otimes \tilde{x})_s &= \max_{j=1,\ldots,n} (a_{sj} + x_j^* + \beta_j) \\
&\leqslant \max_{j=1,\ldots,n} (b_s - \delta + \beta_j) && \text{(by (A.10))} \\
&\leqslant b_s - \delta + \max_{j=1,\ldots,n} \beta_j \\
&< b_s - \delta + \frac{\delta}{2} && \text{(by (A.9))} \\
&< b_s - \frac{\delta}{2} \ .
\end{aligned}$$

But this would imply that $\|b - (A \otimes \tilde{x})\|_\infty \geqslant \frac{\delta}{2}$, which is in contradiction with our initial assumption that $\|b - (A \otimes \tilde{x})\|_\infty < \frac{\delta}{2}$.

Hence, $x^\sharp$ is an optimal solution of Problem (A.7) and the corresponding value of the objective function is $\|b - (A \otimes x^\sharp)\|_\infty = \frac{\delta}{2}$. □

C Analytic solutions for two special cases of the MPL-MPC problem

Now we consider two problems that can be considered as special cases of the MPL-MPC problem and for which analytic solutions exist.

C.1 Special case 1

Consider the following special MPL-MPC problem:

$$\min_{\tilde{u}(k)} J_{\text{out},11} = \bigoplus_{j=1}^{N_\text{p}} \bigoplus_{i=1}^{l} \max(\hat{y}_i(k+j|k) - r_i(k+j), 0) \tag{A.11}$$

$$\tag{A.12}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{A.13}$$

$$\tilde{y}(k) \leqslant \tilde{r}(k) \tag{A.14}$$

with $N_\text{c} = N_\text{p}$. Note that the input constraint (22) is *not* taken into account[7].

An optimal input sequence vector $\tilde{u}^*$ for this MPC is given by (Cuninghame-Green, 1979; Baccelli *et al.*, 1992)

$$\tilde{u}_j^* = \min_{i=1,\ldots,lN_\text{p}} (\tilde{r}_i(k) - h_{ij}) \qquad \text{for } j = 1, \ldots, mN_\text{c}\,, \tag{A.15}$$

provided that $g(k) \leqslant \tilde{r}(k)$ since otherwise there would be no feasible solution. The solution (A.15) corresponds to the "principal solution" (Cuninghame-Green, 1979) or "greatest subsolution" (Baccelli *et al.*, 1992) of the system $H \otimes \tilde{u} = \tilde{r}(k)$, i.e. the solution of $H \otimes \tilde{u} \leqslant \tilde{r}(k)$ for which all components of $\tilde{u}$ are as large as possible without violating the constraint. So this solution in fact minimizes the objective function $J_{\text{in},1}$ (or $J_{\text{in},0}$) over all *optimal* solutions of the problem (A.11)–(A.14).

An alternative way (that also directly yields the evolution of the state of the system) to compute the optimal input sequence that corresponds to $\tilde{u}^*$ is to use the backward equations of the system (Baccelli *et al.*, 1992, Section 5.6).

Remark A-3 In fact (Baccelli *et al.*, 1992) and (Cuninghame-Green, 1979) do not consider the term $g(k)$ in the evolution equation $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$. Nevertheless, it is easy to verify that the result also holds if $g(k)$ is present since the function $\tilde{u} \to H \otimes \tilde{u} \oplus g(k)$ is a monotonically nondecreasing function of $\tilde{u}$. So if $\tilde{u}^*$ is the greatest subsolution of $H \otimes \tilde{u} = \tilde{r}(k)$, then any increase in one of the components of $\tilde{u}^*$ will cause the constraint $H \otimes \tilde{u} \leqslant \tilde{r}(k)$ and thus also $H \otimes \tilde{u} \oplus g(k) \leqslant \tilde{r}(k)$ to be violated[8]. ◇

C.2 Special case 2

Now we consider another special MPC-MPL problem:

$$\min_{\tilde{u}(k)} J_{\text{out},21} = \bigoplus_{j=1}^{N_\text{p}} \bigoplus_{i=1}^{l} |\hat{y}_i(k+j|k) - r_i(k+j)| \tag{A.16}$$

subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{A.17}$$

[7] Also note that the constraint (23) is void since $N_\text{c} = N_\text{p}$.

[8] Recall that all components of the greatest subsolution $\tilde{u}^*$ are as large as possible without violating the constraint $H \otimes \tilde{u} \leqslant \tilde{r}(k)$.

with $N_c = N_p$ and no other constraints. So in this case we to minimize the maximal differences between the due dates and the actual output time instants, which could be useful if we have perishable goods.

If $H \otimes \tilde{u}^* \geqslant g(k)$ with $\tilde{u}^*$ defined by (A.15), then it follows from Theorem A-1 that the optimal solution of the MPC problem (A.16) – (A.17) is given by

$$\tilde{u}^\sharp = \tilde{u}^* \otimes \frac{\delta}{2} \qquad \text{with } \delta = \max_{i=1,\ldots,mN_p} \left(\tilde{r}_i(k) - (H \otimes \tilde{u}^*)_i\right) \ . \tag{A.18}$$

Remark A-4 The analytic solutions for Special Cases 1 and 2 can also be used as initial solutions for iterative optimization methods for the general MPC problem. Note however that the analytic solutions do not take into account all of the MPC constraints (21) and (22) so that they will not always result in a feasible solution (see also Section 6). ◇

D Additional information and results for the example of Section 6

D.1 Derivation of the MPL state space model

Now we derive the max-plus-linear state space model of the production system of Section 6. For readers familiar with Petri nets, we have presented the Petri net or timed event graph that corresponds to this production system in Figure A-1.

First we determine the time instant at which processing unit P_1 starts working for the $(k+1)$th time. If we feed raw material to the system for the $(k+1)$th time, then this raw material is available at the input of processing unit P_1 at time $t = u(k) + 2$. However, P_1 can only start working on the new batch of raw material as soon as it has finished processing the current, i.e. the kth batch. Since the processing time on P_1 is $d_1 = 11$ time units, the kth intermediate product will leave P_1 at time $t = x_1(k) + 11$. Since P_1 starts working on a batch of raw material as soon as the raw material is available and the current batch has left the processing unit, this implies that we have

$$x_1(k+1) = \max(\, x_1(k) + 11,\, u(k) + 2\,) \ . \tag{A.19}$$

Using a similar reasoning we find the following expressions for the time instants at which P_2 and P_3 start working for the $(k+1)$st time and for the time instant at which the kth finished product leaves the system:

$$x_2(k+1) = \max(\, x_2(k) + 12,\, u(k) + 0\,) \tag{A.20}$$

$$x_3(k+1) = \max(\, x_1(k+1) + 11 + 1,\, x_2(k+1) + 12 + 0,\, x_3(k) + 7\,) \tag{A.21}$$

$$= \max(\, x_1(k) + 23,\, x_2(k) + 24,\, x_3(k) + 7,\, u(k) + 14\,) \tag{A.22}$$

$$y(k) = x_3(k) + 7 + 0 \ . \tag{A.23}$$

Let us now rewrite the evolution equations of the production system using the symbols $\oplus$ and $\otimes$. It is easy to verify that (A.19) can be rewritten as

$$x_1(k+1) \;=\; 11 \otimes x_1(k) \;\oplus\; 2 \otimes u(k) \ .$$

Equations (A.20) – (A.23) result in

$$\begin{aligned} x_2(k+1) &= 12 \otimes x_2(k) \ \oplus \ u(k) \\ x_3(k+1) &= 23 \otimes x_1(k) \ \oplus \ 24 \otimes x_2(k) \ \oplus \ 7 \otimes x_3(k) \ \oplus \ 14 \otimes u(k) \\ y(k) &= 7 \otimes x_3(k) \ . \end{aligned}$$

If we rewrite these evolution equations in max-algebraic matrix notation, we obtain the description (29) – (30).

D.2 A further comparison of some control design methods for MPL systems

In this section we compare the results of several control design methods when applied to the MPC problem of Section 6 but instead of considering one MPC step we now look at the evolution over 10 MPC steps.

We take $N_{\mathrm{c}} = 3$ and $N_{\mathrm{p}} = 5$ for the control horizon and the prediction horizon. Note that these values of N_{c} and N_{p} lie closer to the heuristic values that would by used in MPC for PTL systems than the values selected in the example in Section 6 There we have used the values $N_{\mathrm{c}} = 5$ and $N_{\mathrm{p}} = 8$. That choice has mainly been inspired by the need to have enough parameters in the optimization so that the computational requirements of the various methods can be effectively compared. Note however that the behavior of the simple system considered in this example is not sufficiently rich to warrant such large values of N_{c} and N_{p} in practice: it can be shown that the minimal system order of this system is 2 (see (Olsder and De Schutter, 1999) for an overview of techniques to determine minimal system orders and (partial) minimal state space realizations for MPL systems). Furthermore, the impulse response of the system is given by the sequence 21, 32, 43, 55, 67, 79, ... So the steady state behavior, which is characterized by an increment rate of 12 time units, is already reached after 3 steps.

We define

$$\{r(k)\}_{k=1}^{10} = 40, 45, 55, 66, 75, 85, 90, 100, 110, 118.$$

For the 6th up to the 10th MPC step we extend the sequence $\{r(k)\}_{k=1}^{10}$ by assuming that the system operates in steady state with an increment rate of 12 time units[9]. So $r(k+1) = r(k)+12$ for $k \geqslant 10$.

This leads to the following (actual) MPC-based input sequence:

$$\{u_{\mathrm{mpc}}(k)\}_{k=0}^{9} = 12, 24, 35, 46, 58, 70, 82, 94, 106, 118. \tag{A.24}$$

The corresponding output sequence is

$$\{y_{\mathrm{mpc}}(k)\}_{k=1}^{10} = 33, 45, 56, 67, 79, 91, 103, 115, 127, 139,$$

and the corresponding value of the objective function defined by

$$J_{\mathrm{act}} = \sum_{k=1}^{10} \max\bigl(y(k) - r(k), 0\bigr) \ - \ \sum_{k=1}^{10} u(k-1)$$

[9] Note that this corresponds to the steady state increment rate of the impulse response of the system.

is -567. The exact optimal solution, which can be determined by solving the MPC problem for $k = 0$ with $N_\mathrm{c} = N_\mathrm{p} = 10$, is the same as solution (A.24).

The analytic solution given by (A.15) is physically infeasible since it is not nondecreasing.

The solution defined by (A.18) results in the following input sequence:

$$\{u^2(k)\}_{k=0}^{9} = 5, 17, 29, 41, 53, 65, 77, 89, 100, 111 \ .$$

The corresponding output sequence is

$$\{y^2(k)\}_{k=1}^{10} = 31, 43, 55, 67, 79, 91, 103, 115, 127, 139 \ ,$$

and the corresponding value of the objective function J_act is -510.
The control design method of (Menguy *et al.*, 1998*a*) results in

$$\{u^4(k)\}_{k=0}^{9} = 12, 23, 34, 46, 58, 70, 82, 94, 106, 118$$

with $\{y^4(k)\}_{k=1}^{10} = 33, 44, 55, 67, 79, 91, 103, 115, 127, 139$ and $J_\mathrm{act} = -566$.
The other control design methods mentioned at the end of Section 4.5 also lead to input sequences that do not satisfy the constraints.

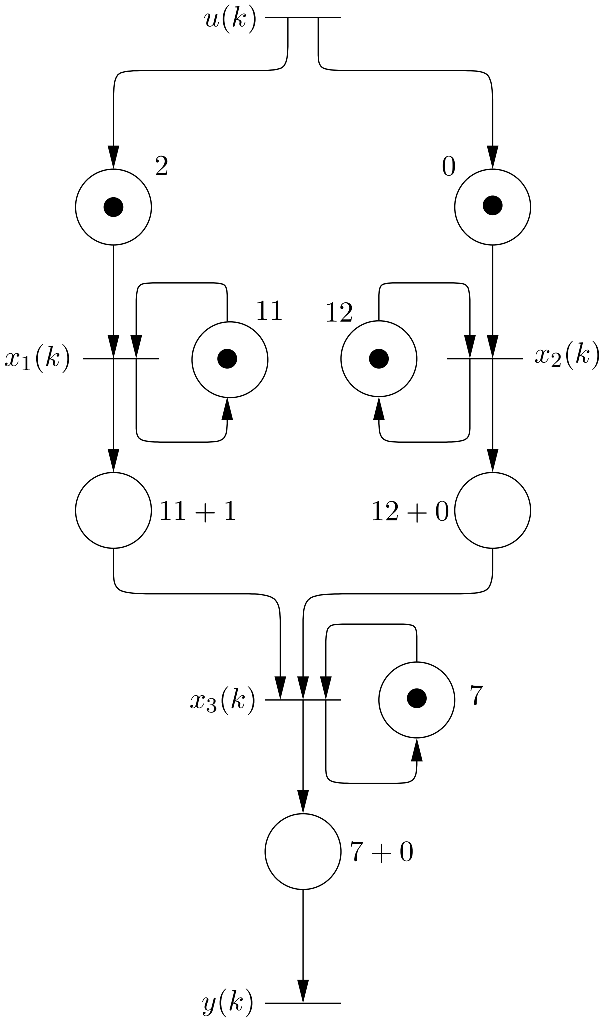

Figure A-1: The Petri net or timed event graph that corresponds to the production system of Figure 1. The numbers next to each place correspond to the holding times of the tokens in that place.