Delft University of Technology

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

Technical report bds:00-04

On model predictive control for max-min-plus-scaling discrete event systems*

B. De Schutter and T. van den Boom

June 2000

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

On model predictive control for max-min-plus-scaling discrete event systems

Bart De Schutter* and Ton van den Boom*

June 2000

Abstract

We extend the model predictive control framework, which is very popular in the process industry due to its ability to handle constraints on inputs and outputs, to a class of discrete event systems that can be modeled using the operations maximization, minimization, addition and scalar multiplication, and that we call max-min-plus-scaling systems. We show that this class encompasses several other classes of discrete event systems such as max-plus-linear systems, bilinear max-plus systems, polynomial max-plus systems, separated max-min-plus systems and regular max-min-plus systems. In general the model predictive control problem for max-min-plus-scaling systems leads to a nonlinear non-convex optimization problem, that can also be solved using extended linear complementarity problems. We show that under certain conditions the optimization problem reduces to a convex programming problem, which can be solved very efficiently.

Keywords: discrete event systems, model predictive control, nonlinear systems

1 Introduction

1.1 Overview

Model predictive control (MPC) is a very popular controller design method in the process industry. If we look at the deployment of controllers in the process industry then conventional PID controllers take up about 95 % of the installed base and of the remaining 5 % the majority are MPC controllers, making MPC currently the most widely used *advanced* control design method in the process industry. MPC is a proven technology for solving industrial problems with a good economic pay-back. MPC provides many attractive features:

- It can handle constraints in a systematic way and it can keep the system behavior as close as possible to the constraints without violating them.
- It is applicable to multivariable systems.
- It is capable of tracking pre-scheduled reference signals, using the concept of making predictions based on a process model.

*Control Lab, Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands, phone: +31-15-278.51.13, email: {b.deschutter,t.j.j.vandenboom}@its.tudelft.nl

- It is an easy-to-tune method. Basically three parameters have to be chosen and adequate tuning rules are available.

Furthermore, variance reduction and constraint pushing can be achieved using MPC. This leads to an improved product quality, a faster adaptation to different working conditions, a decrease of pollution effluents, and a reduction in the workload for the human operators.

Usually MPC uses discrete-time models. In this paper we extend MPC to a class of discrete event systems that can be described using the operations maximization, minimization, addition and scalar multiplication, and that will be called max-min-plus-scaling systems. We also show that this class encompasses several other classes of discrete event systems. Typical examples of max-min-plus-scaling systems are digital circuits, computer networks, telecommunication networks, and manufacturing plants.

A key advantage of conventional MPC is that it allows the inclusion of constraints on the inputs and outputs. This is also one of the main reasons why we introduce MPC for max-min-plus-scaling systems. Furthermore, MPC uses a receding horizon strategy which allows us to regularly update the model of the system and/or the estimates of the current state.

In order to compute an MPC controller for a max-min-plus-scaling system we have to solve a nonlinear non-convex optimization problem. This can be solved using a nonlinear optimization algorithm. An alternative approach consists in using an extended linear complementarity problem to describe the possible trajectories of the system in a compact way and then to minimize the cost criterion over the solution set of the extended linear complementarity problem. However, for systems with many inputs or outputs or if the control horizon is large, these approaches are not tractable in practice. Therefore, we also investigate under which conditions the problem can be recast as a problem with a convex feasible set, or as a convex or even a linear programming problem.

To the authors' best knowledge there are currently no other general control design methods for max-min-plus-scaling systems. Nevertheless, several authors have developed control design methods for some specific subclasses of max-min-plus-scaling systems such as max-plus-linear systems (see e.g. [2, 4, 16, 17] and the references therein). However, in contrast to the MPC method proposed in this paper, these methods do not allow the inclusion of general linear or even more complex constraints on the inputs and the outputs of the system.

This paper is organized as follows. First we give a brief introduction to discrete event systems. Next we introduce the class of max-min-plus-scaling systems. Then we describe the conventional model predictive control framework for discrete-time systems. Next we extend the MPC framework to max-min-plus-scaling systems. We also show that under certain conditions the resulting optimization problem reduces to a convex programming problem, which can be solved very efficiently. We conclude with a worked example.

1.2 Discrete event systems

Typical examples of discrete event systems are flexible manufacturing systems, telecommunication networks, parallel processing systems, traffic control systems and logistic systems. In general, we could say that the class of discrete event systems 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) [2]. 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 [2, 6, 14, 13] and the references cited therein). When selecting the most appropriate model for a discrete event system, an important trade-off that has to be taken into account is that of modeling power versus decision power, i.e. modeling frameworks that can describe large and more general classes of discrete event systems in general lend themselves less easily to mathematical analysis. Therefore, several researchers have focused on special subclasses of discrete event systems that are more amenable to mathematical analysis. One such class for which several analytic results are available is the class of the max-plus-linear systems. Loosely speaking we could say that this class corresponds to the class of discrete event systems in which there is synchronization but no concurrency. Such systems can be modeled using the operations maximization (corresponding to synchronization[1]) and addition (corresponding to durations[2]). This leads to a description that is linear in the max-plus algebra [2, 8] (see also Section 3.1). The operation minimization appears in discrete event descriptions when there is concurrency (e.g. if several machines can process a part, then the part could be sent to the first machine that is available). That is why in the next section we will consider discrete event systems that can be described by models in which the operations maximization, minimization, addition and scalar multiplication appear.

2 Max-min-plus-scaling systems

Definition 2.1 (Max-min-plus-scaling (MMPS) expression)
A max-min-plus-scaling expression f of the variables $x_1, x_2, \ldots, x_n$ is defined by the grammar[3]

$$f := x_i | \alpha | \max(f_k, f_l) | \min(f_k, f_l) | f_k + f_l | \beta f_k \enspace ,$$

with $i \in \{1, 2, \ldots, n\}$, $\alpha \in \mathbb{R}$, and f_k and f_l are again max-min-plus-scaling expressions.

Some examples of MMPS expressions of the variables x_1, x_2, x_3 are $x_1 + 8x_2 - 5$, $\min(\max(x_1 - 3, x_2 + x_3), x_2 - 7x_3)$, or $x_1 - x_2 + 3\max(x_1 - x_2 + 2x_3, x_1 + \min(x_1 - x_2 + x_3, x_2 - x_3, \max(x_1, x_2 - x_3 - 3)))$.

Now we consider discrete event systems that can be described by state space equations of the following form:

$$x(k) = \mathcal{M}_x(x(k-1), u(k)) \tag{1}$$

$$y(k) = \mathcal{M}_y(x(k), u(k)) \tag{2}$$

where $\mathcal{M}_x$ and $\mathcal{M}_y$ are MMPS expressions. For a discrete event system $x(k)$ would typically contain the time instants at which the internal events occur for the kth time, $u(k)$ would contain the time instants at which the input events occur for the kth time, and $y(k)$ would contain the time instants at which the output events occur for the kth time. Systems the

[1] A new operation starts as soon as all preceding operations have been finished.

[2] The finishing time of an operation equals the starting time plus the duration.

[3] The symbol | stands for "or". Also note that the definition is recursive. So an MMPS expression is a variable or a constant, or the maximum or minimum or sum of two MMPS expressions, or a scalar multiple of an MMPS expression.

behavior of which can be described by a model of the form (1)–(2) will be called *MMPS systems*.

Let l, m and n be respectively the number of outputs, inputs and states of the MMPS system.

Remark 2.2 In general the functions $\mathcal{M}_x$ and $\mathcal{M}_y$ in (1)–(2) may even depend on k. For sake of simplicity we will only consider time-invariant MMPS descriptions here. Note however that all the results obtained in this paper also hold for time-varying MMPS systems. $\diamond$

3 MMPS systems and other classes of discrete event systems

In this section we will show that the model (1)–(2) can be considered as a generalized framework that encompasses several subclasses of discrete event systems such as

- max-plus-linear systems
- max-plus-bilinear systems
- max-plus-polynomial systems
- separated max-min-plus systems
- max-min-plus systems

3.1 Max-plus-linear systems

Max-plus-linear systems [2, 8] are discrete event systems that can be described by a state space model of the following form:

$$x(k) = A \otimes x(k-1) \ \oplus \ B \otimes u(k) \tag{3}$$

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

where the operations $\oplus$ and $\otimes$ are defined by

$$(U \oplus V)_{ij} = u_{ij} \oplus v_{ij} = \max(u_{ij}, v_{ij}) \tag{5}$$

$$(U \otimes W)_{ij} = \bigoplus_{k=1}^{q} u_{ik} \otimes w_{kj} = \max_{k=1,\ldots,q} (u_{ik} + w_{kj}) \tag{6}$$

for matrices $U, V \in \mathbb{R}_{-\infty}^{p\times q}$, and $W \in \mathbb{R}_{-\infty}^{q\times r}$ with $\mathbb{R}_{-\infty} = \mathbb{R} \cup \{-\infty\}$. Regarding the order of evaluation, the operation $\otimes$ has precedence over $\oplus$. The reason for selecting the symbols $\oplus$ and $\otimes$ to represent max and $+$ is that there is a remarkable analogy between the operations $\oplus$ and $\otimes$ on the one hand and the operations $+$ and $\times$ on the other hand [2, 8]. Therefore, $\oplus$ is called the max-plus-algebraic addition and $\otimes$ the max-plus-algebraic multiplication. This also explains why the model (3)–(4) is called max-plus-linear, i.e. linear in the max-plus-algebraic sense.

The model (3)–(4) can be rewritten as

$$x_i(k) = \max\big(\max_j(a_{ij} + x_j(k-1)), \max_j(b_{ij} + u_j(k))\big) \qquad \text{for } i = 1, 2, \ldots, n,$$

$$y_i(k) = \max\big(\max_j(c_{ij} + x_j(k)), \max_j(d_{ij} + u_j(k))\big) \qquad \text{for } i = 1, 2, \ldots, l,$$

which is clearly a special case of an MMPS system.

3.2 Max-plus-bilinear systems

Max-plus-bilinear systems are discrete event systems that can be described by a state space model of the following form:

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \oplus \bigoplus_{p=1}^{m} N_p \otimes u_p(k) \otimes x(k-1) \tag{7}$$

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

with $N_p \in \mathbb{R}_{-\infty}^{n \times n}$ for $p = 1, 2, \ldots, m$. This description is the max-plus-algebraic equivalent of conventional bilinear discrete-time systems. Max-plus-bilinear systems arise when some of the inputs of a max-plus-linear system of the form (3)–(4) are used as a switch to control the entries of the system matrix A, i.e. the constant system matrix A is replaced by the input-dependent system matrix $A \oplus N_1 u_1(k) \oplus \ldots \oplus N_m u_m(k)$.

Since the model (7)–(8) can be rewritten as

$$x_i(k) = \max\big(\max_j(a_{ij} + x_j(k-1)), \max_j(b_{ij} + u_j(k)), \max_{p,j}((N_p)_{ij} + u_p(k) + x_j(k-1))\big) \quad \text{for } i = 1, 2, \ldots, n,$$

$$y_i(k) = \max\big(\max_j(c_{ij} + x_j(k)), \max_j(d_{ij} + u_j(k))\big) \quad \text{for } i = 1, 2, \ldots, l,$$

max-plus-bilinear systems are also a subclass of the MMPS systems.

3.3 Max-plus-polynomial systems

The rth max-plus-algebraic power of the scalar variable v is defined by $v^{\otimes^r} = rv$. A max-plus-polynomial p of the scalar variables $v_1, v_2, \ldots, v_n$ can be written as

$$p(v_1, v_2, \ldots, v_n) = \bigoplus_{i=1}^{q} c_i \otimes {v_1}^{\otimes^{r_{i,1}}} \otimes {v_2}^{\otimes^{r_{i,2}}} \otimes \ldots \otimes {v_n}^{\otimes^{r_{i,n}}} \tag{9}$$

where c_i and $r_{i,j}$ are scalars.

Max-plus-polynomial systems are a further extension of max-plus-linear and max-plus-bilinear discrete event systems. They can be described by a state space model of the following form:

$$x(k) = p_x(x(k-1), u(k)) \tag{10}$$

$$y(k) = p_y(x(k), u(k)) \tag{11}$$

where p_x and p_y are max-plus-polynomials. In [23] a subclass of max-plus-polynomial systems[4] has been used in the design of traffic signal switching schemes.

Since (9) can be rewritten as

$$p(v_1, v_2, \ldots, v_n) = \max_{i=1,\ldots,q} (c_i + r_{i,1}v_1 + r_{i,2}v_2 + \cdots + r_{i,n}v_n) \tag{12}$$

which is an MMPS expression, the system (10)–(11) is also an MMPS system.

[4] The systems in considered [23] have a state equation of the form $x(k) = \max(A_1 x(k-1), A_2 x(k-1), \ldots, A_N x(k-1))$, which can be considered as a max-plus-polynomial equation with the entries of the system matrices A_1, A_2, $\ldots$, A_N as exponents (cf. (12) and (9)).

3.4 Separated max-min-plus systems

Separated max-min-plus systems [15, 19, 22] are described by a model of the form

$$x(k) = A \otimes x(k-1) \;\oplus\; B \otimes \tilde{x}(k-1) \tag{13}$$

$$\tilde{x}(k) = C \otimes' x(k-1) \;\oplus'\; D \otimes' \tilde{x}(k-1) \tag{14}$$

where $\oplus$ and $\otimes$ are defined as in (5)–(6), and where $\oplus'$ and $\otimes'$ are the min-plus-algebraic equivalents of $\oplus$ and $\otimes$, i.e.

$$(U \oplus' V)_{ij} = u_{ij} \oplus' v_{ij} = \min(u_{ij}, v_{ij})$$

$$(U \otimes' W)_{ij} = \bigoplus_{k=1}^{q}{}' \, u_{ik} \otimes' w_{kj} = \min_{k=1,\ldots,q} (u_{ik} + w_{kj})$$

for matrices $U, V \in \mathbb{R}_{\infty}^{p \times q}$ and $W \in \mathbb{R}_{\infty}^{q \times r}$ with $\mathbb{R}_{\infty} = \mathbb{R} \cup \{+\infty\}$.

In conventional notation (13)–(14) becomes

$$x_i(k) = \max\big(\max_j(a_{ij} + x_j(k-1)), \max_j(b_{ij} + \tilde{x}_j(k-1))\big) \qquad \text{for } i = 1, 2, \ldots, n,$$

$$\tilde{x}_i(k) = \min\big(\min_j(c_{ij} + x_j(k-1)), \min_j(d_{ij} + \tilde{x}_j(k-1))\big) \qquad \text{for } i = 1, 2, \ldots, m,$$

which is clearly a special case of an MMPS model.

Remark 3.1 The separated max-min-plus descriptor systems considered in [18] correspond to constrained MMPS systems, i.e. MMPS systems of the form (1)–(2) but with an additional MMPS constraint of the form $\mathcal{M}_{\mathrm{c}}(x(k-1), u(k), y(k)) = 0$ where $\mathcal{M}_{\mathrm{c}}$ is an MMPS expression. ◇

3.5 Max-min-plus systems

Max-min-plus systems [12, 20] are described by the model

$$x(k) = \mathcal{M}_{\mathrm{mm}}(x(k-1)) \tag{15}$$

where $\mathcal{M}_{\mathrm{mm}}$ is a max-min expression, i.e. an expression defined by the grammar

$$f := x_i | f_k + \alpha | \max(f_k, f_l) | \min(f_k, f_l)$$

where α is a scalar, and f_k and f_l are again max-min expressions. So max-min expressions are special cases of MMPS expressions. This implies that max-min-plus systems are also a subclass of the MMPS systems.

Note that the class of MMPS systems is a non-trivial superset of the max-min-plus systems described by (15) and considered in [12, 20] since in contrast to the max-min expressions of [12, 20] we also allow the addition of two MMPS expressions and the multiplication by a scalar in the definition of MMPS expressions. Furthermore, the max-min-plus systems considered in [12, 20] are autonomous[5] (i.e. only the state is considered, and there is no input) whereas in the model (1)–(2) we have included inputs and outputs.

So the MMPS systems can be considered as a generalized framework for several classes of discrete event systems.

[5]Note that in fact the separated max-min-plus system (13)–(14) considered in [15, 19, 22] are also autonomous systems with state $[\, x^T(k) \;\; \tilde{x}^T(k) \,]^T$.

4 Model predictive control

In this section we give a short introduction to MPC for deterministic nonlinear discrete-time systems. Since we will only consider the deterministic, i.e. noiseless, case for MMPS systems, we will also omit the noise terms in this introduction to MPC. More extensive information on MPC for (linear and nonlinear) discrete-time systems can be found in [1, 3, 5, 7, 11] and the references therein.

Consider a plant with m inputs and l outputs that can be modeled by a nonlinear discrete-time state space description of the following form:

$$x(k) = f(x(k-1), u(k)) \tag{16}$$

$$y(k) = h(x(k), u(k)) \tag{17}$$

where f and h are smooth functions of x and u.

In MPC we consider the future evolution of the system over a given prediction horizon N_p. For the system (16)–(17) we can make an estimate $\hat{y}(k+j|k)$ for the output at sample step $k+j$ based on the state at step k and the future inputs $u(k+i)$, $i = 0, 1, \ldots, j$. Using successive substitution, we obtain an expression of the following form:

$$\hat{y}(k+j|k) = F_j(x(k-1), u(k), u(k+1), \ldots, u(k+j))$$

for $j = 0, 1, \ldots, N_\mathrm{p} - 1$. If we define the vectors

$$\begin{aligned}\tilde{u}(k) &= \begin{bmatrix} u^T(k) & u^T(k+1) & \ldots & u^T(k+N_\mathrm{p}-1) \end{bmatrix}^T \\ \tilde{y}(k) &= \begin{bmatrix} \hat{y}^T(k|k) & \hat{y}^T(k+1|k) & \ldots & \hat{y}^T(k+N_\mathrm{p}-1|k) \end{bmatrix}^T \ ,\end{aligned}$$

we can derive the expression

$$\tilde{y}(k) = \tilde{F}(x(k-1), \tilde{u}(k)) \ ,$$

which characterizes the estimated future evolution of the output of the system at sample step k over the prediction horizon N_p for the input sequence $u(k), u(k+1), \ldots, u(k+N_\mathrm{p}-1)$.

The MPC cost criterion $J(k)$ measures the reference tracking error $J_\mathrm{out}(k)$ and the control effort $J_\mathrm{in}(k)$ in the interval $[k, k+N_\mathrm{p}-1]$:

$$\begin{aligned} J(k) &= J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k) \\ &= \sum_{j=0}^{N_\mathrm{p}-1} \big(\hat{y}(k+j|k) - r(k+j)\big)^T \big(\hat{y}(k+j|k) - r(k+j)\big) + \lambda \sum_{j=0}^{N_\mathrm{p}-1} u^T(k+j) u(k+j) \\ &= \big(\tilde{y}(k) - \tilde{r}(k)\big)^T \big(\tilde{y}(k) - \tilde{r}(k)\big) + \lambda \tilde{u}^T(k) \tilde{u}(k) \end{aligned}$$

where λ is a nonnegative integer, the signal $r(k)$ is the reference signal for $y(k)$, and $\tilde{r}(k)$ is defined similarly to $\tilde{y}(k)$. In practical situations, there will be constraints on the input and output signals. This is reflected in the nonlinear constraint function

$$C_\mathrm{c}(k, \tilde{u}(k), \tilde{y}(k)) \leqslant 0 \ .$$

Often a linear function may reflect all the desired constraints, and we obtain

$$A_\mathrm{c}(k)\tilde{u}(k) + B_\mathrm{c}(k)\tilde{y}(k) - c_\mathrm{c}(k) \leqslant 0 \ . \tag{18}$$

The MPC problem at sample step k consists in minimizing $J(k)$ over all possible future input sequence vectors $\tilde{u}(k)$ subject to $C_{\mathrm{c}}(k, \tilde{u}(k), \tilde{y}(k)) \leqslant 0$. This is usually a non-convex optimization problem. To reduce the complexity of the optimization problem a control horizon N_{c} is introduced in MPC, which means that the input is taken to be constant beyond sample step $k + N_{\mathrm{c}}$:

$$u(k+j) = u(k+N_{\mathrm{c}}-1) \qquad \text{for } j = N_{\mathrm{c}}, N_{\mathrm{c}}+1, \ldots, N_{\mathrm{p}}-1 \, . \tag{19}$$

In addition to a decrease in the number of optimization parameters and thus also the computational burden, a smaller control horizon N_{c} also gives a smoother control signal, which is often desired in practical situations.

MPC uses a receding horizon principle. This means that after computation of the optimal control sequence $u(k), u(k+1), \ldots, u(k+N_{\mathrm{c}}-1)$, only the first element of the optimal sequence $(u(k))$ is applied to the system. Next the horizon is shifted and a new MPC optimization is performed for sample step $k+1$. Note that the receding horizon approach implies that the MPC optimization problem has to be solved at each sample step k. The computational burden depends on the choices of the horizons N_{p} and N_{c}, and on the complexity of the prediction function $\tilde{F}$ and the constraint function C_{c}. More information on model predictive control of nonlinear systems can be found in [1, 3].

5 Model predictive control for MMPS systems

In this section we extend and adapt the MPC framework from discrete-time systems to MMPS 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. Also note that the counter k in the MMPS model (1)–(2) is an *event counter* (and event occurrence instants are in general not equidistant), whereas in the discrete-time model (16)–(17) k is a *sample counter*, which increases each clock cycle.

We will use the deterministic model (1)–(2) as an approximation of a discrete event system with modeling errors or uncertainty. We also assume that at event step k the current state $x(k)$ can be measured, estimated or predicted using previous measurements. Since MPC uses a receding finite horizon approach, we can regularly update the model and the state estimate as new information and measurements become available.

5.1 Cost criterion

Just as in MPC for discrete-time systems, we define the MPC cost criterion for MMPS systems at event step k as $J(k) = J_{\mathrm{out}}(k) + \lambda J_{\mathrm{in}}(k)$, where $J_{\mathrm{out}}(k)$ is related to the output and $J_{\mathrm{in}}(k)$ is related to the input. Now we discuss some possible choices for $J_{\mathrm{out}}(k)$ and $J_{\mathrm{in}}(k)$.

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

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

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

$$J_{\text{out},2}(k) = \sum_{j=0}^{N_\text{p}-1} \sum_{i=1}^{l} |\hat{y}_i(k+j|k) - r_i(k+j)| \ .$$

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

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

where $\Delta^2 \hat{y}_i(k+j|k) = \hat{y}_i(k+j|k) - 2\hat{y}_i(k+j-1|k) + \hat{y}_i(k+j-2|k)$.

The conventional MPC 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 internal buffer overflows, a better objective is to *maximize* the input time instants. For a manufacturing system, this would correspond to a production 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[6]. So for MMPS systems a more appropriate cost criterion is

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

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

$$J_{\text{in},1}(k) = -\sum_{j=0}^{N_\text{p}-1} \sum_{i=1}^{m} u_i(k+j) \ .$$

If we want to balance the input rates we could use the following cost criterion:

$$J_{\text{in},2}(k) = \sum_{j=1}^{N_\text{p}-1} \sum_{i=1}^{l} |\Delta^2 u_i(k+j)| \ .$$

Remark 5.1 Sometimes we want to design a schedule that is robust against unexpected internal delays in the system. In that case we should minimize the input time instants, which leads to an input cost criterion $J_\text{in}(k) = -J_{\text{in},0}(k)$ or $J_\text{in}(k) = -J_{\text{in},1}(k)$. ◇

Note that for the input cost criteria defined above we could replace the upper summation index N_p by N_c or redefine $\tilde{u}(k)$ accordingly. We could also replace both summations (or only the second) in the definitions of the input and output cost criteria given above by maximizations, add some weight factors to the terms of the cost criterion, or consider weighted mixtures of several cost criteria.

[6]This also leads to a notion of stability if we let instability for the manufacturing system correspond to internal buffer overflows. This is related to the "internal stability" as defined in [2]. Note that in general several definitions of stability are possible for discrete event systems (see e.g. [2, 21]).

5.2 Constraints

In the context of discrete event systems typical constraints are:

$$a_1(k+j) \leqslant \Delta u(k+j) \leqslant b_1(k+j) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{c} - 1,$$
$$a_2(k+j) \leqslant \Delta \hat{y}(k+j|k) \leqslant b_2(k+j) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p} - 1,$$
$$\hat{y}(k+j|k) \leqslant r(k+j) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p} - 1,$$

where $\Delta u(k+j) = u(k+j) - u(k+j-1)$. It is easy to verify that all these constraints can also be recast as a linear constraint of the form (18), which is used in conventional MPC.

Since for MMPS systems the input sequence usually corresponds to occurrence times of consecutive events, it should always be nondecreasing. Therefore, we also have to add the condition

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

This is also a constraint of the form (18).

In general we could consider constraints of the form

$$\mathcal{M}_\mathrm{c}(k, \tilde{u}(k), \tilde{y}(k), x(k-1)) \leqslant 0$$

where $\mathcal{M}_\mathrm{c}$ is an MMPS expression.

5.3 The evolution of the input beyond the control horizon

In MPC for discrete-time systems the condition that from step $k + N_\mathrm{c}$ on the input should stay constant, helps to reduce the number of variables in the MPC optimization problem. Therefore, we also introduce a control horizon constraint in MPC for MMPS systems.

A straightforward application of that the input should stay constant from step $k + N_\mathrm{c}$ on is not very useful for MMPS systems since the input sequences should normally be increasing. Therefore, we change this condition as follows: the feeding rate should stay constant beyond step $k + N_\mathrm{c}$, i.e. $\Delta u(k+j) = \Delta u(k + N_\mathrm{c} - 1)$ for $j = N_\mathrm{c}, N_\mathrm{c} + 1, \ldots, N_\mathrm{p} - 1$, or equivalently

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

This condition introduces regularity in the input sequence. In addition 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 (19).

5.4 The standard MPC problem for MMPS systems

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

Find the input sequence vector $\tilde{u}(k)$ that minimizes the cost criterion $J(k)$ subject to

$$\hat{x}(k+j|k) = \mathcal{M}_x(\hat{x}(k+j-1|k), u(k+j)) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p} - 1, \tag{20}$$
$$\hat{y}(k+j|k) = \mathcal{M}_y(\hat{x}(k+j-1|k), u(k+j)) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p} - 1, \tag{21}$$
$$\mathcal{M}_\mathrm{c}(k, \tilde{u}(k), \tilde{y}(k), x(k-1)) \leqslant 0 \tag{22}$$
$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p} - 1, \tag{23}$$
$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_\mathrm{c}, N_\mathrm{c} + 1, \ldots, N_\mathrm{p} - 1, \tag{24}$$

with $\hat{x}(k-1|k) = x(k-1)$.

This problem will be called the MMPS-MPC problem for event step k. Note that in this case we also use a receding horizon approach in which in each step we effectively apply only the first input sample.

6 Algorithms to solve the MMPS-MPC problem

6.1 Nonlinear optimization

The MMPS-MPC problem is a nonlinear nonconvex optimization problem. So we could use multistart local constrained optimization algorithms to compute the optimal input sequence. However, this will not always yield the global optimum. In addition the required computation time may be too large to use this approach in practice. Therefore, we will now discuss two other approaches, one based on the Extended Linear Complementarity Problem, which can always be applied, and a much more efficient approach that can however only be used in some special cases.

6.2 The Extended Linear Complementarity Problem

The Extended Linear Complementarity Problem (ELCP) is defined as follows [9]:

> Given $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{q\times n}$, $c \in \mathbb{R}^p$, $d \in \mathbb{R}^q$ and m subsets $\phi_1, \ldots, \phi_m$ of $\{1, \ldots, p\}$, find $z \in \mathbb{R}^n$ such that
> $$\prod_{i\in\phi_j} (Az - c)_i = 0 \quad \text{for } j = 1, \ldots, m, \tag{25}$$
> subject to $Az \geqslant c$ and $Bz = d$, or show that no such z exists.

Equation (25) represents the complementarity condition of the ELCP. One possible interpretation of this condition is the following: each set ϕ_j corresponds to a group of inequalities of $Az \geqslant c$ and in each group at least one inequality should hold with equality, i.e. the corresponding residue should be equal to 0. So for each j there should exist an index $i \in \phi_j$ such that $(Az - c)_i = 0$.

In general, the solution set of the ELCP defined above consists of the union of faces of the polyhedron defined by the system of linear equations and inequalities ($Az \geqslant c$ and $Bz = d$) of the ELCP. In [9] we have developed an algorithm to compute the complete solution set of an ELCP. This algorithm yields a description of the solution set of an ELCP by vertices, extreme rays and a basis of the linear subspace corresponding to the largest affine subspace of the solution set.

Let us now show how the MMPS-MPC problem can be solved using the ELCP. This will be done by showing that each of the 6 basic constructors for an MMPS expression fits the ELCP framework:

- Expressions of the form $f = x_i$, $f = \alpha$, $f = f_k + f_l$ and $f = \beta f_k$ (or their combinations) result in linear equations of the form $Bz = d$ where z contains the variables[7] f, x_i, f_k and f_l.

[7] In this case f, f_k and f_l are dummy variables.

- An expression of the form $f = \max(f_k, f_l)$ can be rewritten as

$$\begin{aligned} &f \geqslant f_k \\ &f \geqslant f_l \\ &f = f_k \ \text{ or } \ f = f_l \end{aligned}$$

  or equivalently

$$\begin{aligned} &f - f_k \geqslant 0 \\ &f - f_l \geqslant 0 \\ &(f - f_k)(f - f_l) = 0 \ , \end{aligned}$$

  which is an ELCP.

- In a similar way an expression of the form $f = \min(f_k, f_l)$ can be rewritten as

$$\begin{aligned} &f_k - f \geqslant 0 \\ &f_l - f \geqslant 0 \\ &(f_k - f)(f_l - f) = 0 \ , \end{aligned}$$

  which is also an ELCP.

This implies that by introducing additional dummy variables if necessary, any MMPS expression can be recast as an ELCP. Furthermore, two or more ELCPs can be combined into one large ELCP. The constraints (23)–(24) just yield additional linear (in)equalities. So the system (20)–(24), which defines the feasible set of the MMPS-MPC problem, can be rewritten as an ELCP. We can compute a compact parametric description of the solution set of an ELCP using the algorithm of [9]. In order to determine the optimal MPC policy we then have to determine for which values of the parameters the objective function $J(k)$ over the solution set of the ELCP that corresponds to (20)–(24). The algorithm of [9] 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_c, m or l are large. However, in the next section we will show that under certain conditions the MMPS-MPC problem leads to a convex optimization problem, which can be solved very efficiently.

6.3 Monotonic objective functions and constraints

We say that a function F is a monotonically nondecreasing (nonincreasing) function of y if $y^* \leqslant y^\sharp$ implies that $F(y^*) \leqslant F(y^\sharp)$ $\big(F(y^*) \geqslant F(y^\sharp)\big)$.

Definition 6.1 (Max-plus-positive-scaling (MaxPPS) expression)
A max-plus-positive-scaling expression f of the variables $x_1, \ldots, x_n$ is defined by the grammar

$$f := \alpha x_i \,|\, \beta \,|\, \max(f_k, f_l) \,|\, f_k + f_l \,|\, \rho f_k$$

with $i \in \{1, \ldots, n\}$, $\alpha, \beta, \rho \in \mathbb{R}$, $\rho > 0$ and where f_k and f_l are again max-plus-positive-scaling expressions.

Definition 6.2 (Min-plus-positive-scaling (MinPPS) expression)
A min-plus-positive-scaling expression f of the variables $x_1, \ldots, x_n$ is defined by the grammar

$$f := \alpha x_i \,|\, \beta \,|\, \min(f_k, f_l) \,|\, f_k + f_l \,|\, \rho f_k$$

with $i \in \{1, \ldots, n\}$, $\alpha, \beta, \rho \in \mathbb{R}$, $\rho > 0$ and where f_k and f_l are again min-plus-positive-scaling expressions.

Now it is easy to verify that the following statements hold:

Statement 6.3 *A MaxPPS function is a convex function of its arguments.*

Statement 6.4 *A MinPPS function is a concave function of its arguments.*

Statement 6.5 *A MaxPPS function is a monotonically nondecreasing function of its arguments.*

Statement 6.6 *A MinPPS function is a monotonically nonincreasing function of its arguments.*

In the remainder of this section we consider the MPC problem for a subclass of MMPS systems that can be described by the following state space model:

$$x_{\max}(k) = \mathcal{M}_x^{\max}(x_{\max}(k-1), u(k)) \tag{26}$$
$$x_{\min}(k) = \mathcal{M}_x^{\min}(x_{\min}(k-1), u(k)) \tag{27}$$
$$y_{\max}(k) = \mathcal{M}_y^{\max}(x_{\max}(k), u(k)) \tag{28}$$
$$y_{\min}(k) = \mathcal{M}_y^{\min}(x_{\min}(k), u(k)) \tag{29}$$

where $\mathcal{M}_x^{\max}$ and $\mathcal{M}_y^{\max}$ are MaxPPS expressions, and $\mathcal{M}_x^{\min}$ and $\mathcal{M}_y^{\min}$ are MinPPS expressions. The vector $x(k) = [\, x_{\max}^T(k) \;\; x_{\min}^T(k) \,]^T$ is the state of the system at event step k, and $y(k) = [\, y_{\max}^T(k) \;\; y_{\min}^T(k) \,]^T$ is the output of the system at event step k. Furthermore, we consider a linear constraint instead of the general MMPS constraint (22). So the MPC constraints are

$$A_{\mathrm{c}}(k)\, \tilde{u}(k) + B_{\mathrm{c}}(k)\, \tilde{y}(k) \leqslant c_{\mathrm{c}}(k) \tag{30}$$
$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_{\mathrm{p}} - 1, \tag{31}$$
$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}} - 1. \tag{32}$$

Remark 6.7 It is easy to verify that any MaxPPS expression f of $x_1, \ldots, x_n$ can be written in a conjunctive normal form:

$$f = \max(f_1, f_2, \ldots, f_n)$$

where $f_1, \ldots, f_n$ are affine expressions of the variables $x_1, \ldots, x_n$. In a similar way any MinPPS expression f of $x_1, \ldots, x_n$ can be written in a disjunctive normal form:

$$f = \min(f_1, f_2, \ldots, f_n)$$

where $f_1, \dots, f_n$ are affine expressions of the variables $x_1, \dots, x_n$.
This implies that the linear constraint (30) can also be considered to encompass MaxPPS and MinPPS constraints of the form

$$\mathcal{M}_{\mathrm{c}}^{\max}(k, \tilde{u}(k), \tilde{y}(k), x(k-1)) \leqslant 0$$
$$\mathcal{M}_{\mathrm{c}}^{\min}(k, \tilde{u}(k), \tilde{y}(k), x(k-1)) \leqslant 0$$

where $\mathcal{M}_{\mathrm{c}}^{\max}$ is a MaxPPS expression and $\mathcal{M}_{\mathrm{c}}^{\max}$ is a MinPPS expression. $\diamond$

Now we consider the *relaxed MMPS-MPC problem* for the system described by (26)–(29). This problem is defined by the evolution equations (26)–(29) and the constraints (30)–(32) but with the =-sign in (26) and (28) replaced by a $\geqslant$-sign, and the =-sign in (27) and (29) replaced by a $\leqslant$-sign:

$$x_{\max}(k) \geqslant \mathcal{M}_x^{\max}(x_{\max}(k-1), u(k)) \tag{33}$$
$$x_{\min}(k) \leqslant \mathcal{M}_x^{\min}(x_{\min}(k-1), u(k)) \tag{34}$$
$$y_{\max}(k) \geqslant \mathcal{M}_y^{\max}(x_{\max}(k), u(k)) \tag{35}$$
$$y_{\min}(k) \leqslant \mathcal{M}_y^{\min}(x_{\min}(k), u(k)) \ . \tag{36}$$

The constraints (30)–(32) are linear and thus convex. Furthermore, from Statements 6.3 and 6.4 it follows that the constraints (33)–(36) are also convex. As a consequence, the set of feasible solutions of the relaxed MMPS-MPC problem is convex. Hence, the relaxed problem is much easier to solve numerically.

Let $\tilde{y}_{\max}(k)$ and $\tilde{y}_{\min}(k)$ be defined in a similar way as $\tilde{y}(k)$. Using a reasoning that is an extension of the one used in [10] for the max-plus-linear MPC, we will now show that if the objective function $J(k)$ and the linear constraints are monotonically nondecreasing functions of $x_{\max}(k)$ and $\tilde{y}_{\max}(k)$ and monotonically nonincreasing functions of $x_{\min}(k)$ and $\tilde{y}_{\min}(k)$, then the optimal solution of the relaxed MMPS-MPC problem can be transformed into a solution of the original MPC problem:

Theorem 6.8 *Consider an MMPS system that can be modeled by (26)–(29). Let the objective function $J(k)$ and the mapping $\tilde{y}(k) \to B_{\mathrm{c}}(k)\tilde{y}(k)$ be monotonically nondecreasing functions of $\tilde{y}_{\max}(k)$ (and $x_{\max}(k)$) and monotonically nonincreasing functions of $\tilde{y}_{\min}(k)$ (and $x_{\min}(k)$), Let $(\tilde{u}^*(k), \tilde{y}^*(k))$ be an optimal solution of the relaxed MMPS-MPC problem. If we define $\tilde{y}^\sharp(k)$ by*

$$x_{\max}^\sharp(k+j|k) = \mathcal{M}_x^{\max}\big(x_{\max}^\sharp(k+j-1|k), u^*(k+j)\big)$$
$$x_{\min}^\sharp(k+j|k) = \mathcal{M}_x^{\min}\big(x_{\min}^\sharp(k+j-1|k), u^*(k+j)\big)$$
$$y_{\max}^\sharp(k+j|k) = \mathcal{M}_y^{\max}\big(x_{\max}^\sharp(k+j|k), u^*(k+j)\big)$$
$$y_{\min}^\sharp(k+j|k) = \mathcal{M}_y^{\min}\big(x_{\min}^\sharp(k+j|k), u^*(k+j)\big)$$

for $j = 0, 1, \dots, N_{\mathrm{p}}-1$ and with $x_{\max}^\sharp(k-1|k) = x_{\max}(k-1)$ and $x_{\min}^\sharp(k-1|k) = x_{\min}(k-1)$, then $(\tilde{u}^*(k), \tilde{y}^\sharp(k))$ is an optimal solution of the original MMPS-MPC problem.

Proof: For ease of notation we define $\tilde{u}^* \stackrel{\text{def}}{=} \tilde{u}^*(k)$, $\tilde{y}^* \stackrel{\text{def}}{=} \tilde{y}^*(k)$, $\tilde{y}^\sharp \stackrel{\text{def}}{=} \tilde{y}^\sharp(k)$.
Clearly, $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the original MMPS-MPC problem.

Now we first show by induction that

$$x^*_{\max}(k+j|k) \geqslant x^\sharp_{\max}(k+j|k) \qquad \text{for } j = 0, 1, \ldots, N_\mathrm{p}-1. \tag{37}$$

By definition (37) holds for $j = 0$ since $x^*_{\max}(k-1|k) = x^\sharp_{\max}(k-1|k) = x_{\max}(k-1)$. Now we assume that (37) holds for $j = M$ with $M \in \{0, 1, \ldots, N_\mathrm{p}-2\}$ and we show that it also holds for $j = M+1$. Since by Statement 6.5 $\mathcal{M}^{\max}_x$ is a monotonically nondecreasing function of its arguments, the induction assumption implies that

$$\begin{aligned} x^*_{\max}(k+M|k) &= \mathcal{M}^{\max}_x\big(x^*_{\max}(k+M-1|k), u^*(k+M)\big) \\ &\geqslant \mathcal{M}^{\max}_x\big(x^\sharp_{\max}(k+M-1|k), u^*(k+M)\big) \\ &\geqslant x^\sharp_{\max}(k+M|k) \ . \end{aligned}$$

Hence, (37) holds for all values of j.
In a similar way we can show that $\tilde{y}^*_{\max}(k) \geqslant \tilde{y}^\sharp_{\max}(k)$, and — since MinPPS functions are monotonically nonincreasing by Statement 6.6 — that $x^*_{\min}(k+j|k) \leqslant x^\sharp_{\min}(k+j|k)$ for $j = 0, 1, \ldots, N_\mathrm{p}-1$ and thus $\tilde{y}^*_{\min}(k) \leqslant \tilde{y}^\sharp_{\min}(k)$. Since the objective function $J(k)$ and the mapping $\tilde{y}(k) \to B_\mathrm{c}(k)\tilde{y}(k)$ are monotonically nondecreasing functions of $\tilde{y}_{\max}(k)$ and $x_{\max}(k)$ and monotonically nonincreasing functions of $\tilde{y}_{\min}(k)$ and $x_{\min}(k)$, this implies that

(a) $J(\tilde{u}^*, \tilde{y}^\sharp) \leqslant J(\tilde{u}^*, \tilde{y}^*)$

(b) $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the relaxed MMPS-MPC problem.

Since $(\tilde{u}^*, \tilde{y}^*)$ is an optimal solution of the relaxed MMPS-MPC problem, (a) and (b) imply that $J(\tilde{u}^*, \tilde{y}^\sharp) = J(\tilde{u}^*, \tilde{y}^*)$. So $(\tilde{u}^*, \tilde{y}^\sharp)$ is also an optimal solution of the relaxed MMPS-MPC problem. Since in addition it is also a feasible solution of the original MMPS-MPC problem and since the set of feasible solutions of the relaxed MMPS-MPC problem is a superset of the set of feasible solutions of the original MMPS-MPC problem, this implies that $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MMPS-MPC problem. □

Note that we can always obtain an objective function that is a monotonically nondecreasing function of function of $\tilde{y}_{\max}(k)$ and a monotonically nonincreasing function of $\tilde{y}_{\min}(k)$ by eliminating $\tilde{y}(k)$ from the expression for $J(k)$ using the evolution equations (26)–(29) 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.

Recall that the relaxed MMPS-MPC problem has a convex feasible set. So if Theorem 6.8 applies the optimal MPC policy can be computed much more efficiently than in the general case. If in addition the objective function is convex (e.g. if $J(k)$ equals $-J_{\mathrm{in},0}(k)$, $\pm J_{\mathrm{in},1}(k)$, $J_{\mathrm{out},1}(k)$ or a weighted combination of these objective functions), we finally get a convex optimization problem, which can be solved efficiently using, e.g. an interior point method. Since $J_{\mathrm{in},1}(k)$ is a linear function, the problem even reduces to a linear programming problem for $J(k) = \pm J_{\mathrm{in},1}(k)$, which can be solved very efficiently. Furthermore, it easy to verify that for $J(k) = J_{\mathrm{out},1}(k)$ the problem can also be reduced to a linear programming problem by introducing some additional dummy variables (see also Footnote 8).

7 Example

Consider the MMPS system that can be described by the following state space model

$$\begin{aligned}
x_1(k) &= \max(0.5x_1(k-1) + x_2(k-1) + 0.8u(k), x_1(k-1) + 0.6x_2(k-1)) \\
x_2(k) &= \max(x_1(k-1) + 0.4x_2(k-1) + u(k), 0.5x_1(k-1) + x_2(k-1)) \\
x_3(k) &= \min(0.5x_3(k-1) + x_4(k-1) + u(k), x_3(k-1) + 2x_4(k-1)) \\
x_4(k) &= \min(x_3(k-1) + 0.4x_4(k-1) + 0.8u(k), 1.5x_3(k-1) + x_4(k-1)) \\
y_1(k) &= \max(x_1(k), x_2(k)) \\
y_2(k) &= \min(x_3(k), x_4(k)) \ .
\end{aligned}$$

Let $x_0 = [\,1\ 2\ 3\ 1\,]^T$.

We will solve the MMPS-MPC problem for this system with $N_\mathrm{c} = 4$, $N_\mathrm{p} = 6$, with the following constraints:

$$\begin{aligned}
1 \leqslant \Delta u(k) \leqslant 5 \qquad & \text{for } k = 1, 2, \ldots, 6, \\
y_2(k) \geqslant r_2(k) \qquad & \text{for } k = 1, 2, \ldots, 6,
\end{aligned}$$

where $\{r_1(k)\}_{k=1}^6 = 5, 10, 15, 30, 55, 90$, $\{r_2(k)\}_{k=1}^6 = 4, 10, 20, 35, 60, 100$, and with the objective function

$$J(k) = \sum_{k=1}^{6} \max(y_1(k) - r_1(k), 0) \ + \ 0.02\, \tilde{u}^T(k)\tilde{u}(k) \ ,$$

which makes a trade-off between the tardiness of the output y_1 w.r.t. the reference signal r_1, and the input, which is minimized so that raw material is fed to the system as early as possible to guarantee robustness against possible unexpected internal delays. Since the model of the system is a model of the form (26)–(29), and since the objective function and the constraints are monotonically nondecreasing as a function of $\tilde{y}_1(k)$ and monotonically nonincreasing as a function of $\tilde{y}_2(k)$, Theorem 6.8 applies. Hence, we can consider the MMPS-MPC relaxed problem when we want to compute the optimal input sequence. In addition, the objective function $J(k)$ is convex so that we can solve the relaxed problem using a convex optimization algorithm.

In order to compare the efficiency of the different MMPS-MPC algorithms discussed in Section 6, we have solved one step of the MMPS-MPC problem for $k = 1$ using the ELCP method, using nonlinear constrained optimization, and using the relaxed problem with a quadratic programming algorithm[8]. All methods yield the same optimal input sequence[9]:

$$\{u(k)\}_{k=1}^6 = 2.50, 3.98, 4.98, 7.72, 10.46, 13.20$$

with $J(k) = 30.64$ as the optimal value of the objective function. However, for the method that uses nonlinear constrained optimization several runs with different initial starting points where necessary to find the global optimum[10], whereas for the two other methods starting

[8] If we introduce a dummy variable t and additional constraints $0 \leqslant t$, $y_1(k) - r_1(k) \leqslant t$ for $k = 1, 2, \ldots, 6$ and if we consider the objective function $\tilde{J} = t + 0.02\tilde{u}^T(k)\tilde{u}(k)$, we get a quadratic programming problem that is equivalent to the relaxed MMPS-MPC problem.

[9] All numerical results in this section will be specified up to 2 decimal places.

[10] In an experiment with 20 random starting points 7 runs did not result in a feasible final solution. The average value of the final objective function value for the 14 runs that resulted in a feasible final solution was 30.97 with a standard deviation of 0.87.

Method	CPU time
ELCP	7568.390
nonlinear optimization	0.679
quadratic programming	0.027

Table 1: CPU time (in seconds, on a 1 GHz PC with 256 MB RAM, up to 3 decimal places) needed to solve one step of the MMPS-MPC problem of Section 7. Each indicated value is the average over 20 runs with random starting points.

from different initial points always yields almost the same result[11]. The output sequences that correspond to the optimal input sequence are

$$\{y_1(k)\}_{k=1}^{6} = 4.50, 10.20, 19.05, 34.49, 59.90, 99.19$$
$$\{y_2(k)\}_{k=1}^{6} = 5.00, 10.34, 20.00, 35.44, 60.90, 100.00 \ .$$

The CPU times required to compute the optimal input sequences are listed in Table 1. Clearly, the approach using the relaxed problem is much more efficient than the other approaches.

Note that in this example we have only considered one MPC step. In practice the computation of the optimal input sequence should be repeated every event step[12]. Nevertheless, the results given above also hold if we consider several consecutive MPC steps: the approach using the relaxed problem will stay much more efficient than the other approaches.

8 Conclusions and future research

In this paper we have further extended the popular model predictive control (MPC) framework from discrete-time systems to max-min-plus-scaling (MMPS) discrete event systems. The reason for using an MPC approach for MMPS 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 step or event step so that model changes can be taken into account as soon as they are identified).

We have also presented some methods to solve the MMPS-MPC problem. In general this leads to a nonlinear non-convex optimization problem. If the state and output equations can be split in max-plus-positive-scaling and min-plus-positive-scaling parts that are decoupled, and if the objective function and the constraints are monotonic functions of the states and the outputs, then we can relax the MMPS-MPC problem to problem with a convex set of feasible solutions. If in addition the objective function is convex or linear, this leads to a problem that can be solved very efficiently.

In this paper we have only considered the noiseless case, which we have used as an approximation of a discrete event system with modeling errors or uncertainty. The extension of the current MPC framework to nondeterministic MMPS models will be a topic for future research. We will also develop efficient methods to solve this extended problem.

[11] In an experiment with 20 random starting points the first 8 decimal places of the final objective function always had the same value.

[12] This implies that we can use a shifted version of the current optimal input sequence as an initial starting point for the next optimization.

Another topic for further research is the investigation of the effects of the three tuning parameters (the input cost weight λ, the prediction horizon N_p and the control horizon N_c) and the determination of guidelines for selecting appropriate values for these tuning parameters. If we take $J_\text{in}(k)$ equal to $J_{\text{in},0}(k)$ or $J_{\text{in},1}(k)$ then the parameter λ influences stability in the sense that internal buffer overflows are prevented by maximizing the input time instants. However, in contrast to discrete-time systems — where the components of the input sequence are minimized — we now want to maximize the components of the input sequence. Nevertheless, just as in conventional MPC we have to find a balance between the output error and control effort. It is still an open question how this can be translated into rules or guidelines for the selection of appropriate values of the parameter λ. Furthermore, it is obvious that for the MPC problem for MMPS systems the prediction horizon N_p is also related to the speed of the dynamics of the system. For linear discrete-time systems N_c is usually set equal to the system order. For MMPS systems there do not yet exist (efficient) algorithms for the computation of the system order. Just as for discrete-time systems, an important consequence of decreasing N_c in MPC for MMPS systems is the reduction in computational effort, especially if we use the ELCP approach to solve the problem. So a more elaborate determination of the influences of the tuning parameters λ, N_p and N_c will be a topic for further research.

References

[1] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright, "Nonlinear predictive control and moving horizon estimation – An introductory overview," in *Advances in Control: Highlights of ECC '99* (P.M. Frank, ed.), pp. 391–449, London, UK: Springer, 1999.

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

[3] L. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear Model Predictive Control* (F. Allgöwer and A. Zheng, eds.), vol. 26 of *Progress in Systems and Control Theory*, Basel, Switzerland: Birkhäuser Verlag, 2000.

[4] J.L. Boimond and J.L. Ferrier, "Internal model control and max-algebra: Controller design," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 457–461, Mar. 1996.

[5] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.

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

[7] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized predictive control – Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987.

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

[9] B. De Schutter and B. De Moor, "The extended linear complementarity problem," *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.

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

[11] C.E. García, D.M. Prett, and M. Morari, "Model predictive control: Theory and practice – A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[12] J. Gunawardena, "Cycle times and fixed points of min-max functions," in *Proceedings of the 11th International Conference on Analysis and Optimization of Systems* (Sophia-Antipolis, France, June 1994) (G. Cohen and J.P. Quadrat, eds.), vol. 199 of *Lecture Notes in Control and Information Sciences*, pp. 266–272, London: Springer-Verlag, 1994.

[13] Y.C. Ho, ed., *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*. Piscataway, New Jersey: IEEE Press, 1992.

[14] Y.C. Ho, ed., "Special issue on dynamics of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, Jan. 1989.

[15] A. Jean-Marie and G.J. Olsder, "Analysis of stochastic min-max systems: Results and conjectures," *Mathematical and Computer Modelling*, vol. 23, no. 11/12, pp. 175–189, 1996.

[16] E. Menguy, J.L. Boimond, and L. Hardouin, "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, Aug. 1998.

[17] E. Menguy, J.L. Boimond, and L. Hardouin, "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, July 1998.

[18] G.J. Olsder, "Descriptor systems in the max-min algebra," in *Proceedings of the 1st European Control Conference*, Grenoble, France, pp. 1825–1830, July 1991.

[19] G.J. Olsder, "Eigenvalues of dynamic max-min systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 1, no. 2, pp. 177–207, Sept. 1991.

[20] G.J. Olsder, "On structural properties of min-max systems," in *Proceedings of the 11th International Conference on Analysis and Optimization of Systems* (Sophia-Antipolis, France, June 1994) (G. Cohen and J.P. Quadrat, eds.), vol. 199 of *Lecture Notes in Control and Information Sciences*, pp. 237–246, London: Springer-Verlag, 1994.

[21] T.I. Seidman and L.E. Holloway, "Stability of a 'signal kanban' manufacturing system," in *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico, pp. 590–594, June 1997.

[22] Subiono and G.J. Olsder, "On bipartite min-max-plus systems," in *Proceedings of the European Control Conference (ECC'97)*, Brussels, Belgium, paper 207, July 1997.

[23] R.J. van Egmond, G.J. Olsder, and H.J. van Zuylen, "A new way to optimise network traffic control using maxplus algebra," in *Proceedings of the 78th Annual Meeting of the Transportation Research Board*, Washington, DC, Jan. 1999. Paper 660.