Model predictive control for switching max-plus-linear systems with random and deterministic switching

Ton van den Boom* Bart De Schutter*,**

* Delft Center for Systems and Control,
** Maritime & Transport Technology,
Delft University of Technology,
Mekelweg 2, 2628 CD Delft, The Netherlands
a.j.j.vandenboom@tudelft.nl, bdeschutter.info,
http://dcsc.tudelft.nl

Abstract: Switching max-plus-linear (SMPL) systems are discrete event systems that can switch between different modes of operation. In each mode the system is described by a max-plus-linear state equation and a max-plus-linear output equation, with different system matrices for each mode. The switching may depend on input and state, or it may be a stochastic process. We derive a stabilizing model predictive controller for SMPL systems with both deterministic and stochastic switching. In general, the optimization in the MPC algorithm boils down to a nonlinear optimization problem, where the cost criterion is piecewise polynomial on polyhedral sets and the inequality constraints are linear.

1. INTRODUCTION

The class of discrete event systems (DES) essentially consists of man-made systems that contain a finite number of resources (such as machines, communications channels, or processors) that are shared by several users (such as product types, information packets, or jobs) all of which contribute to the achievement of some common goal (the assembly of products, the end-to-end transmission of a set of information packets, or a parallel computation) (Baccelli et al., 1992).

In this paper we will consider switching max-plus-linear (SMPL) systems, discrete event systems that can switch between different modes of operation, in which the mode switching depends on a stochastic sequence or depends on the input and the previous state. In each mode the system is described by a max-plus-linear state equation and a max-plus-linear output equation, with different system matrices for each mode. In van den Boom and De Schutter (2006) we have discussed SMPL systems with deterministic switching, and in van den Boom and De Schutter (2007) we have discussed SMPL systems with random switching. In this paper we will give a design procedure for stabilizing controllers of SMPL systems with both types of switching procedures. This means that we will introduce an auxiliary integer-valued input $v(k)$ for deterministic switching and the optimization becomes more complicated compared to van den Boom and De Schutter (2007).

The class of SMPL systems contains discrete event systems with synchronization but no concurrency, in which the order of synchronization of the event steps may vary randomly, or is determined by input signals or the previous state. Typical examples of SMPL systems are flexible manufacturing systems, telecommunication networks, logistics networks, and signal controlled urban traffic networks.

Mode switching depending on input signals allows us to model a change in the structure of the system, such as breaking a synchronization or changing the order of events. Mode switching depending on the state may be due to concurrency between various events (see van den Boom and De Schutter (2006)). Random mode switching between may be due to e.g. (randomly) changing production recipes, varying customer demands or traffic demands, or failures in production units, transmission lines or traffic links.

The paper is organized as follows. In Section 2 we introduce the max-plus algebra and the concept of SMPL systems. Section 3 reviews some conditions for a stabilizing controller, and in Section 4 we derive a stabilizing model predictive controller for SMPL systems. In Section 5 we present a worked example.

2. MAX-PLUS ALGEBRA AND SMPL SYSTEMS

Max-plus algebra

In this section we give the basic definition of the max-plus algebra (Baccelli et al., 1992; Cuninghame-Green, 1979).

Define $\varepsilon = -\infty$ and $\mathbb{R} = \mathbb{R} \cup \{\varepsilon\}$. The max-plus-algebraic addition ($\oplus$) and multiplication ($\otimes$) are defined as follows:

$$x \oplus y = \max(x, y), \quad x \otimes y = x + y$$

$x \oplus y$ is also denoted by $x \vee y$, and $x \otimes y$ is also denoted by $x \wedge y$. $\mathbb{R}$ is a semiring under the operations $\oplus$ and $\otimes$.
for numbers $x, y \in \mathbb{R}_\varepsilon$ and
\[ A \oplus B \]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})
\[ A \otimes C \]_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \ldots \]
for matrices $A, B \in \mathbb{R}^{m \times n}_\varepsilon$ and $C \in \mathbb{R}^{n \times p}_\varepsilon$. The matrix $\mathbb{E}$ is the max-plus-algebraic zero matrix: $[\mathbb{E}]_{ij} = \varepsilon$ for all $i, j$.

A max-plus diagonal matrix $S = \text{diag}(s_1, \ldots, s_n)$ has elements $s_i = \varepsilon$ for $i \neq j$ and diagonal elements $s_i$ for $i = 1, \ldots, n$. If all diagonal elements $s_i$ are finite we find that the max-plus inverse of $S$ is equal to $S^{-1} = \text{diag}(s_1, \ldots, s_n)$. There holds $S \otimes S^{-1} = S^{-1} \otimes S = E$, where $E = \text{diag}(0, \ldots, 0)$ is the max-plus identity matrix.

**SMPL systems**

Switching Max-Plus-Linear (SMPL) systems are discrete event systems that can switch between different modes of operation (van den Boom and De Schutter, 2006). In each mode $\ell = 1, \ldots, L$, the system is described by a max-plus-linear state equation and a max-plus-linear output equation:

\[ x(k) = A^{(\ell(k))} \otimes x(k-1) \oplus B^{(\ell(k))} \otimes u(k) \] (1)
\[ y(k) = C^{(\ell(k))} \otimes x(k) \] (2)
in which the matrices $A^{(\ell)} \in \mathbb{R}^{n_{x} \times n_{x}}_\varepsilon$, $B^{(\ell)} \in \mathbb{R}^{n_{y} \times n_{x}}_\varepsilon$, $C^{(\ell)} \in \mathbb{R}^{2 \times n_{x}}_\varepsilon$ are the system matrices for the $\ell$-th mode $1$. The index $k$ is called the event counter. For discrete event systems the state $x(k)$ typically contains the time instants at which the internal events occur for the $k$th time, the input $u(k)$ contains the time instants at which the input events occur for the $k$th time, and the output $y(k)$ contains the time instants at which the output events occur for the $k$th time $2$.

In van den Boom and De Schutter (2006) we have considered deterministic switching, which was a function of the previous state or an input signal. In van den Boom and De Schutter (2007) we introduced random switching, i.e. the mode switching depended on a stochastic process. In this paper we combine both switching types. For the SMPL system (1)-(2), the mode switching variable $\ell(k)$ depends on both stochastic variables as well as deterministic variables (state and input).

The switching times are determined by a switching mechanism. For the SMPL system (1)-(2), the mode switching variable $\ell(k)$ is a stochastic process, which depends on the previous mode $\ell(k-1)$, the previous state $x(k-1)$, the input variable $u(k)$, and an (additional) control variable $v(k)$. For a system with $L$ possible modes, we assume the probability of switching to mode $\ell(k)$ given $\ell(k-1), x(k-1), u(k), v(k)$ is denoted by $P(\ell(k)|\ell(k-1), x(k-1), u(k), v(k))$. We assume that for all $\ell(k), \ell(k-1) \in \{1, \ldots, L\}$, the probability $P$ is piecewise affine on polyhedral sets in the variables $x(k-1), u(k), v(k)$. $P$ is a probability, so obviously

\[ 0 \leq P(\ell(k)|\ell(k-1), x(k-1), u(k), v(k)) \leq 1 \]

and

\[ \sum_{i=1}^{L} P(\ell(k)|\ell(k-1), x(k-1), u(k), v(k)) = 1 \]

**Example 1 (deterministic switching):**

Let $v(k)$ be a control variable that decides the switching from mode 1 to mode 2 (for $v(k) < 0.5$) or to mode 3 (for $v(k) \geq 0.5$).

\[ v(k) < 0.5 \]
\[ \text{mode 2} \]
\[ v(k) \geq 0.5 \]
\[ \text{mode 3} \]

We achieve this by defining the probability functions

\[ P(2,1,x,u,v) = \begin{cases} 1 & \text{for } v < 0.5, \forall x,u \\ 0 & \text{for } v \geq 0.5, \forall x,u \end{cases} \]
\[ P(3,1,x,u,v) = \begin{cases} 0 & \text{for } v < 0.5, \forall x,u \\ 1 & \text{for } v \geq 0.5, \forall x,u \end{cases} \]

So for deterministic switching, the probability functions are piecewise constant with values either 0 or 1.

**Example 2 (stochastic switching with fixed probability):**

In this case of stochastic switching we assume the probability to switch from mode 1 to mode 2 is equal to $\beta$, and the probability to switch from mode 1 to mode 3 is equal to $1-\beta$, where $\beta \in [0,1]$ is constant.

\[ P(2,1,x,u,v) = \beta, \forall x,u \]
\[ P(3,1,x,u,v) = 1-\beta, \forall x,u \]

We can achieve this by defining the probability functions

\[ P(2,1,x,u,v) = \beta, \forall x,u \]
\[ P(3,1,x,u,v) = 1-\beta, \forall x,u \]

So for stochastic switching with fixed probability, the probability functions are piecewise constant with values between 0 and 1.

**Example 3 (stochastic switching with a probability depending on the state or the input):**

In this example we are able to control the stochastic properties of the switching. Consider a system with state $x(k)$ and let $x_1(k)$ be the first entry of the state. We assume the system switches from mode 1 to a mode 2 for $x_1(k) < 0$, from mode 1 to a mode 3 for $x_1(k) > 1$, and for $x_1(k) \in [0,1]$ we have a probability equal to $x_1(k)$ to switch from mode 1 to mode 2, and a probability equal to $1-x_1(k)$ to switch from mode 1 to mode 3.

1 Note that if we consider a SMPL system with only one mode, we have a special subclass, namely the class of max-plus-linear systems, which describe discrete event systems in which there is synchronization but no concurrency (Baccelli et al., 1992; Cuninghame-Green, 1979).

2 More specifically, for a manufacturing system, $x(k)$ contains the time instants at which the processing units start working for the $k$th time, $u(k)$ the time instants at which the $k$th batch of raw material is fed to the system, and $y(k)$ the time instants at which the $k$th batch of finished product leaves the system.
mode switching and due-date signal (3), and a maximum growth rate $\lambda$. Define the matrices $A_p^{(l)}$ with $[A_p^{(l)}]_{ij} = [A^{(l)}]_{ij} - \rho$. Further assume $C^{(l)}$ to be row-finite. Now if

\begin{equation}
(1) \quad \rho < \lambda,
\end{equation}

\begin{equation}
(2) \quad \text{the system is controllable},
\end{equation}

then any input signal

\begin{equation}
u(k) = \rho k + \mu(k), \quad \text{where } \mu_{\min} \leq \mu(k) \leq \mu_{\max}, \forall i,
\end{equation}

and $\mu_{\min}$ and $\mu_{\max}$ are finite, will stabilize the SMPL system.

\begin{remark}
For a max-plus-linear system (so $L = 1$), condition (8) is equivalent to the condition that the production rate $\rho$ should be larger than the max-plus-linear eigenvalue $\lambda$ of the matrix $A^{(1)}$.
\end{remark}

4. A STABILIZING MODEL PREDICTIVE CONTROLLER

Model predictive control (MPC) (Maciejowski, 2002) is a model-based predictive control approach that has its origins in the process industry and that has mainly been developed for linear or nonlinear time-driven systems. Its main ingredients are: a prediction model, a performance criterion to be optimized over a given horizon, constraints on inputs and outputs, and a receding horizon approach. In van den Boom and De Schutter (2006, 2007) we have extended this approach to MPL systems and deterministic or purely stochastic switching MPL systems and shown that the resulting optimization problem can be solved efficiently. In this section we study the MPC optimization problem for systems with both random and deterministic switching.

In MPC we use predictions of future signals based on the SMPL model. Define the prediction vectors

\begin{align*}
\tilde{y}(k) &= \begin{bmatrix} \tilde{y}(k) \\ \tilde{y}(k+N_p-2) \\ \vdots \\ \tilde{y}(k+N_p-1) \end{bmatrix}, \\
\tilde{u}(k) &= \begin{bmatrix} u(k) \\ u(k+N_p-2) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}, \\
\tilde{\ell}(k) &= \begin{bmatrix} \ell(k) \\ \ell(k+N_p-2) \\ \vdots \\ \ell(k+N_p-1) \end{bmatrix}, \\
\tilde{r}(k) &= \begin{bmatrix} r(k) \\ r(k+N_p-2) \\ \vdots \\ r(k+N_p-1) \end{bmatrix},
\end{align*}

where $\tilde{y}(k+j)$ denotes the prediction of $y(k+j)$ based on knowledge at event step $k$, $u(k+j)$ denotes the future input, $\ell(k+j)$ denotes the future mode, $r(k+j)$ denotes the future due date, and $N_p$ is the prediction horizon (so it determines how many cycles we look ahead in our control law design).

Define

\begin{align*}
\tilde{A}_m(\ell(k)) &= A^{(\ell(k+m-1))} \otimes \cdots \otimes A^{(\ell(k))}, \\
\tilde{B}_{mn}(\ell(k)) &= \begin{cases} A^{(\ell(k+m-1))} \otimes \cdots \otimes A^{(\ell(k+m-1))} & \text{if } m > n, \\ B^{(\ell(k+m-1))} & \text{if } m = n, \\ \mathcal{E} & \text{if } m < n \end{cases}.
\end{align*}
and
\[ C_m(\tilde{\ell}(k)) = C(\ell(k+m-1)) \otimes \tilde{A}_m(\tilde{\ell}(k)), \]
\[ D_{mn}(\tilde{\ell}(k)) = C(\ell(k+m-1)) \otimes \tilde{B}_{mn}(\tilde{\ell}(k)). \]
For any mode sequence \( \tilde{\ell}(k) \) the prediction model for (1)–(2) is now given by:
\[ \tilde{y}(k) = \tilde{C}(\tilde{\ell}(k)) \otimes x(k-1) + \tilde{D}(\tilde{\ell}(k)) \oplus \tilde{u}(k) \]
(10)
in which \( \tilde{C}(\tilde{\ell}(k)) \) and \( \tilde{D}(\tilde{\ell}(k)) \) are given by
\[ \tilde{C}(\tilde{\ell}(k)) = \begin{bmatrix} \tilde{C}_1(\tilde{\ell}(k)) \\ \vdots \\ \tilde{C}_{N_\ell}(\tilde{\ell}(k)) \end{bmatrix}, \]
\[ \tilde{D}(\tilde{\ell}(k)) = \begin{bmatrix} \tilde{D}_{11}(\tilde{\ell}(k)) \cdots \tilde{D}_{1N_p}(\tilde{\ell}(k)) \\ \vdots \\ \tilde{D}_{N_\ell,1}(\tilde{\ell}(k)) \cdots \tilde{D}_{N_\ell,N_p}(\tilde{\ell}(k)) \end{bmatrix}. \]
Further we can write
\[ x(k+j) = \tilde{A}_j(\tilde{\ell}(k)) \otimes x(k-1) + \tilde{B}_j(\tilde{\ell}(k)) \otimes \tilde{u}(k), \]
(11)
where
\[ \tilde{B}_j(\tilde{\ell}(k)) = \begin{bmatrix} \tilde{B}_{j1}(\tilde{\ell}(k)) \\ \vdots \\ \tilde{B}_{j,N_p}(\tilde{\ell}(k)) \end{bmatrix}. \]
With (11) the probability of switching to mode \( \ell(k+j) \) given \( x(k+j-1), \ell(k+j-1), u(k+j), v(k+j) \) can be written as
\[ P(\ell(k+j)|x(k+j-1), \ell(k+j-1), u(k+j), v(k+j)) = P(\ell(k+j)|\tilde{A}_j(\tilde{\ell}(k)) \otimes x(k-1) + \tilde{B}_j(\tilde{\ell}(k)) \otimes \tilde{u}(k), \ell(k+j-1), u(k+j), v(k+j)) \]
where \( P \) denotes the switching probability (see Section 2). Note that from (11) we find that for a fixed \( \tilde{\ell}(k) \) the state \( x(k+j) \) is piecewise affine on polyhedral sets in the variables \( x(k-1) \) and \( \tilde{u}(k) \). From that we can conclude that for a fixed \( \tilde{\ell}(k), x(k-1) \) and \( \ell(k-1) \) the probability \( P \) is piecewise affine on polyhedral sets in the variables \( \tilde{u}(k) \) and \( \tilde{v}(k) \). The probability for the switching sequence \( \tilde{\ell}(k) \in L_{N_\ell} \), given \( \ell(k-1), x(k-1), \tilde{u}(k), \tilde{v}(k) \), is computed as
\[ P(\tilde{\ell}(k)|x(k-1), \ell(k-1), \tilde{u}(k), \tilde{v}(k)) = P(\tilde{\ell}(k)|x(k-1), \ell(k-1), u(k), v(k)) \cdot P(\ell(k+1)|x(k), \ell(k), u(k+1), v(k+1)) \cdot \ldots \]
\[ P(\ell(k+N_p-1)|x(k+N_p-2), \ell(k+N_p-2), u(k+N_p-1), v(k+N_p-1)) \]
The probability function \( P \) is a multiplication of piecewise affine functions \( P \), and will therefore be a piecewise polynomial function on polyhedral sets in the variables \( \tilde{u}(k), \tilde{v}(k) \) (for a given \( \tilde{\ell}(k) \), \( x(k-1) \) and \( \ell(k-1) \)).

In MPC we aim at computing the optimal \( \tilde{u}(k), \tilde{v}(k) \) that minimize the expectation of a cost criterion \( J(k) \), subject to linear constraints on the inputs. The cost criterion reflects the input and output cost functions \( J_{in} \) and \( J_{out} \), respectively) in the event period \([k, k+N_p-1]\):
\[ J(k) = J_{out}(k) + \beta J_{in}(k), \]
(12)
where \( \beta \geq 0 \) is a tuning parameter, chosen by the user. The output cost function is defined by
\[ J_{out}(k) = \mathbb{E} \left\{ \sum_{j=0}^{N_p-1} \max(y_i(k+j), -r_i(k+j), 0) \right\} \]
\[ = \mathbb{E} \left\{ \sum_{i=1}^{n_y N_p} \max(\tilde{y}_i(k), -\tilde{r}_i(k), 0) \right\} \]
\[ = \mathbb{E} \left\{ \sum_{i=1}^{n_y N_p} [ (\tilde{y}(k) - \tilde{r}(k)) \oplus \tilde{0}]_i \right\} \]
\[ = \mathbb{E} \left\{ \sum_{i=1}^{n_y N_p} \left[ (\tilde{C}(\tilde{\ell}(k)) \otimes x(k-1) + \tilde{D}(\tilde{\ell}(k)) \otimes \tilde{u}(k) - \tilde{r}(k)) \oplus \tilde{0} \right]_i \right\} \]
where \( \oplus \tilde{0} \) stands for the expectation over all possible switching sequences, and is a zero column vector. The output cost function \( J_{out} \) measures the tracking error or tardiness of the system, which is equal to the delay between the output dates \( \tilde{y}_i(k) \) and due dates \( \tilde{r}_i(k) \) if \( \tilde{y}_i(k) - \tilde{r}_i(k) > 0 \), and zero otherwise.

The input cost function is chosen as
\[ J_{in,u}(k) = - \sum_{j=0}^{N_p-1} \sum_{i=1}^{n_y} u_i(k+j) + \sum_{j=0}^{N_p-1} \sum_{i=1}^{n_y} \alpha_{ij} v_i(k+j) \]
\[ = - \sum_{i=1}^{n_y N_p} [\tilde{u}(k)]_i + \sum_{i=1}^{n_y N_p} \alpha_i [\tilde{v}(k)]_i. \]
(14)
where \( \alpha = [\alpha_{11} \alpha_{21} \ldots \alpha_{n_y(N_p-1)}]^T \geq 0 \) is a weighting vector. The first term in the input cost function \( J_{in} \) maximizes the input dates \( \tilde{u}_i(k) \), the second term can be used to (possibly) penalize specific actions of the variable \( \tilde{v}_i(k) \).

The MPC problem for SMPL systems with due date signal (3) can be defined at event step \( k \) as
\[ \min_{\tilde{u}(k), \tilde{v}(k)} \quad J(k) \]
(15)
subject to
\[ u(k+j) - u(k+j-1) \geq 0, \quad j=0, \ldots, N_p-1 \]
(16)
\[ \mu_{\min} \leq u_i(k) - \rho k \leq \mu_{\max}, \quad i = 1, \ldots, n_u, \]
(17)
\[ 0 \leq \tilde{v}(k) - \tilde{r} \leq \tilde{r}_{\max} \]
(18)
where (16) guarantees a non-decreasing input sequence, (17) guarantees stability (cf. Theorem 3), and (18) defines the set for the auxiliary input variable \( \tilde{v}(k) \).

MPC uses a receding horizon strategy. So after computation of the optimal control sequences \( \tilde{u}^*(k) \), only the first control sample \( u(k) = \tilde{u}^*(k) \) will be implemented,
subsequently the horizon is shifted and the model and the initial state estimate are updated if new measurements are available, then the new MPC problem is solved, etc.

So the optimization in the MPC algorithm boils down to a nonlinear optimization problem, where the cost criterion is piecewise polynomial and the inequality constraints are linear. This problem can be solved in several ways. Let $\mathcal{P} = \{P_1, \ldots, P_K\}$ be the set of polyhedral regions formed by the intersection of linear constraints (16)–(18) and the regions on which the piecewise polynomial functions expressing $J$ are defined. If the number $K$ of polyhedral regions in $\mathcal{P}$ is small, one could apply for each region $P_i$ a multi-start optimization method for smooth, linearly constrained functions such as steepest descent with gradient projection or sequential quadratic programming (Pardalos and Resende, 2002), and afterwards take the minimum over all regions $P_i$. If $K$ is larger global optimization methods like tabu search (Glover and Laguna, 1997), genetic algorithms (Davis, 1991), simulated annealing (Eglese, 1990), or (multi-start) pattern search (Audet and Dennis Jr., 2005) could be applied.

Note that in the special case where each probability $P$ is a piecewise constant function, $J$ will be a piecewise affine function, and then it can be shown (using an approach similar to the one used in (Bemporad and Morari, 1999)) that the optimization problem reduces to a mixed-integer linear programming problem, for which reliable algorithms are available (Fletcher and Leyffer, 1998; Atamtürk and Savelsbergh, 2005).

Finally we consider the timing issue. Note that $k$ is the event counter and is therefore not directly related to a specific time. We use the assumption that at event step $k$ the state $x(k)$ is available. That means that for an optimization at time $t$ the present event $k$ is defined as

$$k = \arg \max \{ l : x_i(l) \leq t \ \forall i \in \{1, 2, \ldots, n\} \}$$

Hence, the state $x(k)$ is completely known at time $t$ and thus $u(k-1)$ is also available. Note that at time $t$ some components of the forthcoming states and of the forthcoming inputs might be known (so $x_i(k+l) \leq t$ and $u_i(k+l-1) \leq t$ for some $l > 0$). Due to causality, these states are completely determined by the known forthcoming inputs. During the optimization at time $t$ the known values of the input have to be fixed by equality constraints. Due to the information at time $t$ it might be possible to conclude that certain forthcoming modes ($l(k+l)$ for $l > 0$) are not possible anymore. In that case we can set the switching probabilities for this mode to zero, and normalize the switching probabilities of the other modes. With these new probabilities we can do the optimization at time $t$.

If some of the control variables are integer-valued, we get a mixed-integer nonlinear programming problem, which could be solved using branch-and-bound methods (Leyffer, 2001).

5. EXAMPLE: A PRODUCTION SYSTEM

Consider the production system of Figure 1. This system consists of three machines $M_1$, $M_2$, and $M_3$. Two products (A,B) can be made with this system, both with its own recipe, meaning that the order in the production sequence is different for every product. For product A the production order is $M_1\cdot M_3\cdot M_3$, which means that the raw material is fed to machine $M_1$ where it is processed. The intermediate product is sent to machine $M_2$ for further processing, and finally the product is finished in machine $M_3$. For product B two processing orders are allowed, namely $M_2\cdot M_1\cdot M_2$ (denoted as $B_1$) or $M_1\cdot M_2\cdot M_2$ (denoted as $B_2$).

We assume that the type of the $k$th product (A or B) only becomes available at the start of the production, so that we do not know $t(k)$ when computing $u(k)$.

Each machine starts working as soon as possible on each batch, i.e., as soon as the raw material or the required intermediate products are available, and as soon as the machine is idle (i.e., the previous batch has been finished and has left the machine). We define $u(k)$ as the time instant at which the system is fed for the $k$th time, $x_i(k)$ as the time instant at which machine $i$ starts for the $k$th time, and $y(k)$ as time instant at which the $k$th product leaves the system. We assume that all the internal buffers are large enough, and no overflow will occur.

We assume the transportation times between the machines to be negligible, and the processing time of the machines $M_1$, $M_2$ and $M_3$ are given by $d_1 = 1$, $d_2 = 2$ and $d_3 = 3$, respectively. The system equations for $x_1$, $x_2$ and $x_3$ for recipe A are given by

\[
x_1(k) = \max(x_1(k-1) + 1, u(k)),
\]

\[
x_2(k) = \max(x_1(k) + d_1, x_2(k-1) + d_2),
\]

\[
x_3(k) = \max(x_3(k) + d_2, x_3(k-1) + d_3).
\]

leading to the systems matrices for recipe A:

\[
A^{(1)} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\
2d_1 & d_2 & \varepsilon \\
2d_1 + d_2 & 2d_2 & d_3 \end{bmatrix}, \quad B^{(1)} = \begin{bmatrix} 0 \\
d_1 \\
d_1 + d_2 \end{bmatrix}
\]

\[
C^{(1)} = [\varepsilon \ \varepsilon \ \varepsilon].
\]

Similarly we derive for recipe B1:

\[
A^{(2)} = \begin{bmatrix} d_1 & 2d_2 & \varepsilon \\
\varepsilon & d_2 & \varepsilon \\
2d_1 & d_1 + d_2 & d_3 \end{bmatrix}, \quad B^{(2)} = \begin{bmatrix} d_2 \\
0 \\
d_1 + d_2 \end{bmatrix}
\]

\[
C^{(2)} = [\varepsilon \ \varepsilon \ \varepsilon].
\]

and for recipe B2:

\[
A^{(3)} = \begin{bmatrix} d_1 & \varepsilon & \varepsilon \\
2d_1 + d_3 & d_2 & \varepsilon \\
2d_1 & \varepsilon & d_3 \end{bmatrix}, \quad B^{(3)} = \begin{bmatrix} 0 \\
0 \\
d_1 + d_3 \end{bmatrix}
\]

\[
C^{(3)} = [\varepsilon \ \varepsilon \ \varepsilon].
\]
$C^{(3)} = \begin{bmatrix} \varepsilon & d_2 & \varepsilon \end{bmatrix}$.

Note that the matrices $\Gamma^I_1(\tilde{\ell}) = B^{(I)}$, $\ell \in \{1, 2, 3\}$ are all row-finite, and so the SMPL system is controllable.

The demand mechanism for the recipe type is such that if we have a specific recipe in cycle $k$, then the probability of having the same recipe for cycle $k+1$ is 65%, and the probability of a switching to any other recipe is 35%.

At this point we introduce an auxiliary binary control variable $v(k) \in \{0, 1\}$ that can be used to choose between processing order $B_1$ and $B_2$. The switching probability from one recipe to the next one is now given by:

$P(1|1, v) = 0.65$, $P(1|3, v) = 0.35$, $P(2|v) = 0.65(v(k))$, $P(3|1, v) = 0.35(1-v(k))$, $P(3|3, v) = 0.65(1-v(k))$.

The maximum growth rate of the system is equal to $\lambda = 11$. We therefore choose a due date signal given by $r(k) = \rho \cdot k$, where $\rho = 12.1 > \lambda$. The initial state is equal to $x(0) = [4 \ 4 \ 4]^T$, and $J$ is given by (12) for $N_p = 3$, and $\beta = 10^{-5}$. In the experiment, the true switching sequence is simulated for a random sequence with the above given switching probability. The optimization turns out to be a mixed-integer linear programming problem. Figure 2-a gives the due date error between the due date signal $r(k)$ and the output signal $y(k)$, with the corresponding control variable $v(k)$ (see Figure 2-b) for a switching sequence given in Figure 2-c, when the system is in closed-loop with the receding horizon model predictive controller. It can be observed that $y(k)-r(k)$ is initially larger than zero, which is due to the initial state. The error decreases very rapidly and for $k \geq 6$ the error is always equal to zero, which means that the product is always delivered in time. It can clearly be seen that recipe $B_1$ is chosen when $v(k) = 0$ and recipe $B_2$ for $v(k) = 1$.

In this paper we have considered the control of switching max-plus-linear systems, a subclass of discrete event systems, in which we can switch between different modes of operation. In each mode the system is described by max-plus-linear equations with different system matrices for each mode. The switching between the modes can be both deterministic and stochastic.

We have derived a stabilizing model predictive controller for switching max-plus-linear systems. The resulting optimization problem is nonlinear with a piecewise polynomial cost criterion and linear inequality constraints.

6. DISCUSSION

REFERENCES


