Model-based Predictive Control for Hammerstein systems [1]

H. H. J. Bloemen, T. J. J. van den Boom, H. B. Verbruggen
Department of Information Technology and Systems
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
Tel: (+31) 15 2782087 Fax: (+31) 15 2786679 Email: H.Bloemen@its.tudelft.nl

Abstract

Hammerstein systems are a class of systems represented by a static nonlinearity at the input followed by a linear dynamic block. In this paper the static input nonlinearity is transformed into a polytopic description. The remaining uncertain linear model is used in a MPC algorithm of which the optimization problem involves minimization of a linear objective function subject to Linear Matrix Inequalities (LMIs), which is a convex problem. A procedure is presented to remove a number of LMIs from the optimization problem, prior to solving it. By means of an iterative procedure the conservatism of the polytopic description can be reduced. Nominal closed loop stability of this Hammerstein MPC algorithm is guaranteed. A comparison is presented between the proposed algorithm and an algorithm which removes the nonlinearity from the control problem via an inversion.

Keywords: Predictive control, Hammerstein systems, Linear matrix inequalities, Stability, Actuator nonlinearities

1 Introduction

In Model-based Predictive Control (MPC) the input is calculated by an on-line minimization of a performance index based on predictions by the model, subject to constraints. When the model is nonlinear, the optimization problem is non convex in general, due to the nonlinear model which acts as an equality constraint. Optimizing a non convex optimization problem is not attractive because it is computationally expensive and the algorithm may get stuck in a local minimum. A Hammerstein model consists of a series connection of a static input nonlinearity and a linear dynamic block, and therefore represents a subclass of nonlinear systems. Using a Hammerstein model within a MPC algorithm, in a straightforward way, will result in above problems. However, the structure of Hammerstein models can be exploited such that a convex optimization problem is retained. Although Hammerstein models only represent a small subclass of general nonlinear models, they have been used successfully in the identification of nonlinear systems [5]. Moreover, the availability of black box identification algorithms, e.g. [5, 10], makes the class of Hammerstein models a suitable candidate for modeling, in cases where linear models are poor.

A popular way of handling Hammerstein models in a MPC algorithm is to remove the nonlinearity from the control problem via an inversion [6, 9]. The performance index for the remaining linear dynamic part of the Hammerstein model is convex. However, in the original design goal the nonlinearity is present since it usually involves both the output and the physical input. Removing the nonlinearity from the control problem therefore may lead to a decreased performance.

In this paper a MPC algorithm is presented which does take the effect of the static nonlinearity into account, while the optimization problem remains convex. This is achieved by transforming the nonlinearity into a polytopic description. The remaining linear model with polytopic uncertainty is then used in a robust linear MPC strategy based on the one presented in [2]. The optimization problem involves a minimization of a linear objective function subject to Linear Matrix Inequalities (LMIs) and is presented in section 2. In section 3 the basic concept of the algorithms presented in [6, 9] is outlined. A simulation example is given in section 4, which demonstrates the benefits of the algorithm presented in this paper, compared to an algorithm which removes the nonlinearity from the control problem. Finally the conclusions are given in section 5.

2 The Hammerstein MPC algorithm

In state space a Hammerstein model is represented by:

$$w(k) = h(u(k)) \tag{1}$$

$$LTI \begin{cases} x(k+1) = Ax(k) + Bw(k) \\ y(k) = Cx(k) + Dw(k) \end{cases} \tag{2}$$

where $u \in \mathbb{R}^m$ is the physical input to the plant, which is passed through the nonlinearity h to give the input $w \in \mathbb{R}^m$ to the linear time invariant (LTI) dynamic

[1] Supported by the Dutch Technology Foundation (STW) under project number DEL55.3891

block, $x \in \mathbb{R}^n$ is the state of the linear block, $y \in \mathbb{R}^p$ is the output of the model. The static nonlinearity h is assumed to be invertible. A, B, C, D are the system matrices (of conformal dimensions) of the linear dynamic block. Assume that the desired steady state $(u_{ss}, w_{ss}, x_{ss}, y_{ss}) = (0, 0, 0, 0)$. When this is not the case the system can be shifted such that the origin of the shifted system corresponds to the desired steady state. Then the goal is to minimize the following quadratic performance index (notation: $||b||_\Psi^2 = b^T \Psi b$):

$$J_\infty(k) = \sum_{i=0}^{\infty} ||y(k+i)||_Q^2 + ||u(k+i)||_{R_u}^2 \qquad (3)$$

subject to the model equations (1), (2) and possible input, state and output constraints, where $u(k+i)$ are the degrees of freedom. Q and R_u are positive definite weighting matrices. Because of the input nonlinearity h this optimization problem is non-convex in general. The infinite horizons provide the property of nominal closed loop stability, however this is computationally intractable since this results in an infinite dimensional optimization problem. To avoid an infinite dimensional optimization problem while retaining the property of nominal closed loop stability the performance index is restated:

$$\begin{aligned} J(k) &= J_1(k) + J_2(k) \qquad (4) \\ J_1(k) &= \sum_{i=0}^{H_s-1} ||y(k+i)||_Q^2 + ||u(k+i)||_{R_u}^2 \\ J_2(k) &= ||x(k+H_s)||_P^2 \end{aligned}$$

and an end-point inequality constraint is added requiring the state $x(k+H_s)$ to lie in a region in which a stabilizing controller exists. The weighting matrix P in $J_2(k)$ is designed such that $J_2(k)$ is an upperbound for the tail of the infinite horizon performance index, for which the inputs are provided by the stabilizing controller. Therefore $J(k) \geq J_\infty(k)$. Since at time $k+H_s$ the parameterization of the input changes, H_s will be referred to as the switching horizon. The design of the stabilizing controller, the end-point inequality constraint and P can be done off-line, e.g. [4], but this can also be incorporated into the on-line optimization problem as presented in [2]. Although the on-line computational load will increase when doing all the computations on-line, the performance of the controller will be better, especially for short switching horizons [2]. In the following subsections the MPC algorithm for Hammerstein models is stated as an optimization problem subject to linear matrix inequalities (LMIs), where the number of LMIs increases rapidly with increasing H_s in general. This motivates the use of a small H_s and as a result, the on-line design of the stabilizing controller, the end-point inequality constraint and P, i.e. an extension of the algorithm presented in [2] towards Hammerstein models.

To avoid a non-convex optimization problem, the nonlinearity of the Hammerstein model is transformed into a polytopic description. For reasons discussed in subsection 2.3 the original nonlinearity, equation (1), is inverted:

$$u(k) = h^{-1}(w(k)) = g(w(k)) \qquad (5)$$

where the nonlinear function g is the inverse of h. Equation (5) can be represented as:

$$u(k) = g(w(k)) = G(w(k)) \cdot w(k) \qquad (6)$$

because the desired steady state corresponds to the origin. The elements of the matrix $G(w(k))$ are nonlinear functions of $w(k)$. The operating region for w is restricted due to the facts that: a) physical constraints on u will result in constraints on w, b) the model is only valid in the operating region in which it is identified. As the operating region for w is restricted, the elements of $G(w(k))$ are bounded for all w within this operating region. In this way the nonlinearity can be captured by a polytopic uncertainty description (Co refers to the convex hull):

$$\left\{ \begin{aligned} u(k) &= G(w(k)) \cdot w(k) \\ G(w(k)) &\in \Omega = Co\{\mathcal{G}_1 \dots \mathcal{G}_N\} \end{aligned} \right. \qquad (7)$$

which means that the nonlinear matrix $G(w(k))$ can be expressed as some convex combination of all constant vertices $\mathcal{G}_1 \dots \mathcal{G}_N$ for all values of $w(k)$ within the valid operating region. The accuracy of the polytope (i.e. how well the polytope fits around the nonlinearity) can usually be increased by increasing the number of vertices. A common approach is to use a hyper rectangle for the polytope, however, better choices may exist. How to make this choice is beyond the scope of this paper, more information on this topic can be found in [1] for example. Some practical aspects regarding the type of nonlinearity and elimination of vertices can be found in subsection 2.3.

Using above procedure the Hammerstein model is transformed into a linear model, equations (2), with a polytopic uncertainty description for the physical input u, equations (7). This enables to use a robust linear MPC algorithm to control the Hammerstein system. The MPC algorithm used in this paper is based on the one presented in [2], which is extended to handle models represented by equations (2) and (7). Deriving the LMIs connected to the minimization of the performance index, equation (4), consists of a part corresponding to $J_1(k)$ and of a part corresponding to $J_2(k)$.

2.1 Minimizing $J_1(k)$

The prediction of y up to time $k + H_s - 1$ can be represented as, (see [7] for example):

$$\tilde{y} = \tilde{C}\tilde{A}x(k) + (\tilde{C}\tilde{B} + \tilde{D})\tilde{w}$$

with $\tilde{w}$ and $\tilde{y}$ vectors of stacked $w(k+i)$ and $y(k+i)$ for $i = 0 \dots H_s - 1$ respectively, and $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ prediction matrices of conformal dimensions. Then $J_1(k)$ is given by:

$$\begin{aligned} J_1(k) &= ||\tilde{C}\tilde{A}x(k) + (\tilde{C}\tilde{B} + \tilde{D})\tilde{w}||^2_{\tilde{Q}} \\ &\quad + ||\tilde{w}||^2_{\tilde{G}(\tilde{w})^T \tilde{R}_u \tilde{G}(\tilde{w})} \end{aligned} \tag{8}$$

with $\tilde{Q}$, $\tilde{R}_u$ and $\tilde{G}(\tilde{w})$ representing the block diagonal matrices $diag(Q,\dots,Q)$, $diag(R_u,\dots,R_u)$ and $diag(G(w),\dots,G(w))$ respectively. Minimizing $J_1(k)$ is equivalent to minimizing η_1 under the constraint $J_1(k) \leq \eta_1$. Using Schur complements, see [8], this constraint can be tranformed into a nonlinear matrix inequality which is affine in the nonlinearity $\tilde{G}(\tilde{w})$ and in the degrees of freedom $\tilde{w}$ and η_1. Similar to the polytopic description for one time step, equations (7), a polytopic description for $\tilde{G}(\tilde{w})$ is given by:

$$\begin{cases} \tilde{u} &= \tilde{G}(\tilde{w}) \cdot \tilde{w} \\ \tilde{G}(\tilde{w}) &\in \tilde{\Omega} = Co\{\tilde{\mathcal{G}}_1 \dots \tilde{\mathcal{G}}_{\tilde{N}}\} \end{cases} \tag{9}$$

Based on the structure of $\tilde{G}(\tilde{w})$, a straightforward choice for generating the vertices $\tilde{\mathcal{G}}_i$ is to build block diagonal matrices based on all possible combinations of vertices of Ω on the main diagonal. This leads to $\tilde{N} = N^{H_s}$ vertices for $\tilde{\Omega}$, i.e. an exponential increase as function of H_s, which motivates the use of a small H_s, and as a consequence, the choice of the MPC strategy, see the first part of section 2.

Now any valid $\tilde{G}(\tilde{w})$ can be represented by some convex combination of the elements of $\tilde{\Omega}$. This implies that if the obtained nonlinear matrix inequality holds for every vertex of $\tilde{\Omega}$, every convex combination of these inequalities will hold, and thus that the original nonlinear inequality is also satisfied. This transforms the nonlinear optimization problem into a convex optimization problem given a LMI for every vertex of the polytopic description $\tilde{\Omega}$. Although this optimization problem is convex, it may require too many LMIs, especially in case of large H_s, for the on-line optimization to be computationally tractable. In subsection 2.3 a procedure is given to eliminate a number of LMIs from the optimization problem, prior to solving it.

Besides the LMIs for all the vertices of $\tilde{\Omega}$ additional LMIs are needed to specify the constraints for $\tilde{w}$, which were used to calculate the polytopic description for $\tilde{G}(\tilde{w})$. When these constraints are linear they are easily converted into LMIs. The same holds for linear constraints on $\tilde{x}$ and $\tilde{y}$. Constraints on $\tilde{u}$ can be implemented using the uncertainty description $\tilde{\Omega}$, but they are preferably approximated by (or converted into) linear constraints on $\tilde{w}$, which will be assumed in the sequel. The motivation for this is the fact that the polytopic description will give a conservative constraint handling of input constraints on $\tilde{u}$, which will decrease the region of attraction of the controller. The conservatism of the polytopic description will not affect the feasibility for inequality $J_1(k) \leq \eta_1$ as the nonlinearity can basically be interpreted as a nonlinear weighting, see equation (8). Therefore conservatism will only change the minimum of $J_1(k)$ given by η_1.

2.2 Minimizing $J_2(k)$

In this part of the optimization problem $J_2(k)$ is minimized under the constraint that $J_2(k)$ is an upper bound for the tail of J_∞, i.e. equation (3) $\forall\ i \geq H_s$. This is accomplished by imposing the following constraint, [2, 8]:

$$\begin{aligned} &||x(k+i)||^2_P - ||x(k+i+1)||^2_P \geq \\ &\qquad ||y(k+i)||^2_Q + ||u(k+i)||^2_{R_u}, \quad \forall\ i \geq H_s \end{aligned} \tag{10}$$

which is easily demonstrated by summing inequality (10) for $i = H_s \dots \infty$. To avoid an infinite number of degrees of freedom the input is parameterized as:

$$w(k+i) = Fx(k+i), \quad \forall\ i \geq H_s \tag{11}$$

Minimizing $J_2(k)$, equation (4), is equivalent to minimizing η_2 under the constraint $J_2(k) \leq \eta_2$. After a variable transformation $P = \eta_2 S^{-1}$ and division by η_2, this constraint yields:

$$x(k+H_s)^T S^{-1} x(k+H_s) \leq 1 \tag{12}$$

which, by using Schur complements [8], can be transformed into an LMI in $\tilde{w}$ and S (note that $x(k+H_s)$ is an affine function of $\tilde{w}$). Using the feedback, equation (11), and the model, equations (2) and (7), inequality (10) yields:

$$\begin{aligned} &x(k+i)^T\{P - ||A+BF||^2_P - ||C+DF||^2_Q \\ &-||F||^2_{G(w)^T R_u G(w)}\}x(k+i) \geq 0, \quad \forall\ i \geq H_s \end{aligned} \tag{13}$$

which holds $\forall\ i \geq H_s$ if the matrix expression between the curly brackets is positive semi-definite. Substituting $P = \eta_2 S^{-1}$ and $F = YS^{-1}$, followed by pre and post multiplication by S, division by η_2, and using Schur complements, this matrix inequality can be written as a nonlinear matrix inequality which is affine in the variables (Y, S and η_2) and is affine in the nonlinearity $G(w)$, see [8] for more details concerning this procedure. Using a similar reasoning as in subsection 2.1 this nonlinear matrix inequality will hold if all LMIs resulting from every vertex of the polytopic description Ω, equation (7), will hold. In this way the nonlinear matrix inequality is replaced by N corresponding LMIs.

Input, state and output constraints should still be incorporated into the optimization problem, $\forall\ i \geq H_s$. Suppose the constraints on w, which are always present since they are used to calculate the polytope Ω, are given by:

$$\Pi_w w(k+i) \leq \theta_w \quad \forall\ i \geq H_s \tag{14}$$

Using the state feedback, the j-th constraint on w, the j-th row of equation (14), is satisfied if:

$$\max_{i \geq H_s} |\Pi_{w,j} Y S^{-1} x(k+i)|^2 \leq \theta_{w,j}^2 \tag{15}$$

Because $\varepsilon = \{z | z^T S^{-1} z \leq 1\}$, equivalent to inequality (12), is an invariant ellipsoid for the predicted states due to inequality (10), $x(k+H_s) \in \varepsilon$ implies $x(k+i) \in \varepsilon \ \forall \ i \geq H_s$. This means inequality (15) is satisfied if (because $\max_{i \geq H_s} |\Pi_{w,j} w(k+i)|^2 \leq \max_{z \in \varepsilon} |\Pi_{w,j} Y S^{-1} z|^2$):

$$\max_{z \in \varepsilon} |\Pi_{w,j} Y S^{-1} z|^2 \leq \theta_{w,j}^2$$

which is satisfied if (using the invariant ellipsoid ε):

$$\max_{z \in \varepsilon} z^T S^{-1} Y^T \Pi_{w,j}^T \frac{1}{\theta_{w,j}^2} \Pi_{w,j} Y S^{-1} z \leq z^T S^{-1} z$$

Taking z outside brackets, the matrix expression within the brackets yields an LMI after pre and post multiplication by S and using Schur complements. Constraints on the state and output can be expressed as LMIs similarly.

2.3 Removing LMIs from the optimization problem

An unfavorable property of the polytopic description to account for the nonlinearity of the Hammerstein model is the rapid increase in the number of LMIs for increasing number of inputs and increasing H_s. A partial remedy for this is based on the observation that the nonlinearity basically enters as a nonlinear weighting in the performance index, see equations (8) and (13). This implies that the active LMIs for minimizing $J_1(k)$ and $J_2(k)$ will correspond to those vertices of Ω and $\tilde{\Omega}$ that yield the largest performance index. Thus the vertices $\mathcal{G}_j$ of Ω which will never lead to active LMIs (connected to the performance index) are characterized by:

$$\mathcal{G}_j^T R_u \mathcal{G}_j \leq \mathcal{G}_i^T R_u \mathcal{G}_i, \quad \text{for any} \quad i \neq j \tag{16}$$

and the corresponding LMIs can be eliminated from the optimization problem prior to solving it, thus decreasing the computational complexity. A similar reasoning holds for the vertices of $\tilde{\Omega}$. In the special case when the nonlinearity consists of m parallel SISO nonlinearities, $G(w)$ will be diagonal. If then R_u is chosen diagonal as well, there will only be one active vertex $\mathcal{G}_{lim}$ characterized by:

$$|\mathcal{G}_{lim}| \geq |\mathcal{G}_i|, \quad \forall \quad \mathcal{G}_i \in \Omega$$

This reduces the number of LMIs connected to $J_1(k)$ to one, i.e. $\tilde{G}(\tilde{w})$ substituted by $\tilde{\mathcal{G}}_{lim}$, and the number of LMIs connected to $J_2(k)$ to two, i.e. $G(w)$ substituted by $\mathcal{G}_{lim}$ plus the LMI resulting from equation (12).

This procedure to remove LMIs from the optimization problem is the main motivation for firstly inverting the nonlinearity, equation (5), leading to the polytopic description (7), instead of directly generating a polytopic description for equation (1):

$$\begin{cases} w(k) &= H(u(k)) \cdot u(k) \\ H(u(k)) &\in \Omega_h = Co\{\mathcal{H}_1 \ldots \mathcal{H}_N\} \end{cases} \tag{17}$$

In the latter case, for $J_1(k)$ for example this yields:

$$J_1(k) \quad = \quad ||\tilde{C}\tilde{A}x(k) + (\tilde{C}\tilde{B} + \tilde{D})\tilde{H}(\tilde{u})\tilde{u}||_{\tilde{Q}}^2 + ||\tilde{u}||_{\tilde{R}_u}^2$$

where the nonlinearity can no longer be interpreted as a nonlinear weighting matrix, and consequently LMIs can not be removed from the optimization problem according to a condition similar to inequality (16).

Total optimization problem: *After using the procedure to remove LMIs from the optimization problem, the optimization problem of the MPC is a combination of the optimization problems stated in subsections 2.1 and 2.2:*

$$\min_{\tilde{w}, Y, S, \eta_1, \eta_2} \eta_1 + \eta_2 \tag{18}$$

subject to the remaining LMIs.

2.4 Iteratively solving the optimization problem

Given the valid operating region for the model (expressed as linear constraints for w) the polytopic description of the nonlinearity may be very conservative. The performance index minimized by the controller is an upperbound for performance index (4) because it over-estimates the original weighting on w, see equations (8) and (13), $G(w(k+i))^T R_u G(w(k+i))$, by $\mathcal{G}_{lim}(k+i)^T R_u \mathcal{G}_{lim}(k+i)$. When the difference between these two is large the control action is likely to be very conservative. To improve the control action of the proposed MPC algorithm the optimization problem stated in equation (18) is solved iteratively such that this difference decreases. Firstly the optimization problem is solved using a polytopic description based on the entire valid operating region of the model. This yields $\tilde{w}^*$, Y^* and S^* (the star denotes the optimal value). Then "artificial" constraints around $\tilde{w}^*$ and $w(k+H_s)^*$ can be specified. Given these constraints a new polytopic description for the nonlinearity is calculated for every time instant k till $k+H_s$. Because the operating region is reduced, the uncertainty and thus conservatism will be reduced, compared to the previous iteration. This means the upperbound of the performance index that is minimized by the controller is closer to the original performance index, thus yielding a better control action. Note that because the new uncertainty description is based on a narrow operating region around the solution from the previous optimization, it varies for every time instant k till $k+H_s$. This can be interpreted as some kind of automatic gain scheduling. Every new optimization problem will always be feasible because $\tilde{w}^*$, Y^* and S^* resulting from the previous optimization is always a feasible solution. This procedure can be repeated several times.

The tightening of the operating region for w should be performed gradually, to avoid the "artificial" constraints from becoming active in an early stage. The constraints for $w(k+i)\ \forall i \geq H_s$, θ_w in inequality (14), are symmetrical around the origin due to the way they are handled in inequality (15), so $\theta_{w,new}$ should be the largest absolute value of the new artifical constraints at time $k+H_s$. Moreover, because the constraint handling for this part of the optimization problem is conservative, $\theta_{w,new}$ should be tightened conservatively.

Theorem (Nominal closed loop stability)
Under the assumption that R_u and Q are positive definite, nominal closed loop stability of the MPC algorithm presented in this section is guaranteed for a controllable and observable plant once a feasible solution of the MPC problem is found.
proof: Due to space limitations a detailed proof is omitted. Stability can be proven by showing that the performance index is a Lyapunov function, which can be done along the same lines as in [3].

3 Removing the nonlinearity via an inversion

Via an inversion, equation (5), the nonlinearity can be removed from the control problem, by minimizing an alternative performance index [6, 9]:

$$J_{LTI}(k) = \sum_{i=0}^{\infty} ||y(k+i)||_Q^2 + ||w(k+i)||_{R_w}^2 \qquad (19)$$

subject to the LTI part of the model equations (2) and possible input, state and output constraints. R_w is a positive definite weighting matrix. $J_{LTI}(k)$ can be split into two parts similarly as $J(k)$, equation (4). If the transformed input constraints are convex, or can be approximated by convex ones, the minimization of (19) is a convex optimization problem. This optimization problem provides a solution $w(k)$, and via equation (5) the physical input to the plant $u(k)$ is obtained. Although the calculated $w(k)$ is optimal with respect to $J_{LTI}(k)$, the resulting $u(k)$ may be far from optimal with respect to the original performance index $J_\infty(k)$, especially when there is a strong nonlinearity. This is caused by the fact that the influence of the nonlinearity on the input-output behavior of the plant is not taken into account in $J_{LTI}(k)$. Note the difference between $J(k)$ and $J_{LTI}(k)$. In $J(k)$ the effect of the nonlinearity is still taken into account by the controller via an upperbound of the nonlinear weighting matrix $\tilde{G}(\tilde{w})^T \tilde{R}_u \tilde{G}(\tilde{w})$, which can be fine tuned as presented in subsection 2.4, whereas in $J_{LTI}(k)$ a constant weighting matrix R_w is used.

4 Simulation results and discussion

In this section the performance of the algorithms presented in section 2 and 3 are compared using the following model.

$$w = \frac{e^u - 1}{e^u + 1}$$
$$A = \begin{bmatrix} 2.6 & -1.1 & 0.7 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}$$
$$C = \begin{bmatrix} -0.4 & 0 & -0.1 \end{bmatrix}, D = \begin{bmatrix} -0.2 \end{bmatrix}$$

The input nonlinearity represents a smooth saturation. Since black box identification algorithms like presented in [10] may provide a non-zero D (for example to approximate a delay which is small with respect to the sample time) this justifies the use of a non-zero D in the MPC algorithm. The weighting matrices of J_∞ are $Q = 1$ and $R_u = 0.5$. H_s is set to 5. The initial condition used is $x = [0, 0, 0]^T$. The setpoint for y was 0.5, 0, 1, 0, 2, 0 for blocks of 25 time steps respectively. The operating region for u was limited to $|u| \leq 5$. These constraints can be transformed into linear constraints on w. Two extra iterations, as sketched in subsection 2.4, were performed for the algorithm of section 2. The operating region for w is reduced by adding the constraints:

$$w_{j,old}(k+i) - w_b \leq w_j(k+i) \leq w_{j,old}(k+i) + w_b$$

where $w_{j,old}$ is the solution from the previous iteration and w_b specifies the reduced operating region around the solution given by $w_{j,old}$. Every extra iteration w_b is reduced by $w_{b,new} = 0.05 + 0.5 \cdot w_{b,old}$. The first extra optimization w_b was set to 0.5, the value of w_b converges to 0.1. Every time the setpoint changes all "artificial" constraints are deleted and the constraints are reset to the original operating region.

The MPC algorithm of section 3 is focused on minimizing $J_{LTI}(k)$, which involves the weighting R_w. Since the input nonlinearity is linear close to the origin, the MPC algorithms of sections 2 and 3 will give a similar performance in this region when:

$$R_w = \left(\left. \frac{\partial u}{\partial w} \right|_{w=0} \right)^T R_u \left. \frac{\partial u}{\partial w} \right|_{w=0} = 2 \cdot R_u \cdot 2$$

In figure 1 the simulation results are plotted for both algorithms. Until $t = 75$ the input remains within the linear part of the nonlinearity and both algorithms show a similar behavior. This is also visible in figure 2 because the simulation cost:

$$J_s(k) = \sum_{i=0}^{i=k} ||y(i) - y_{ref}(i)||_Q^2 + ||(u(i) - u_{ref}(i))||_{R_u}^2$$

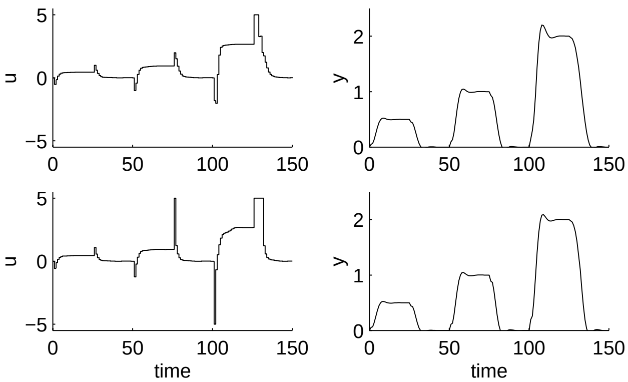

Figure 1: Simulation using the MPC algorithm of section 2 (upper plots) and section 3 (lower plots)

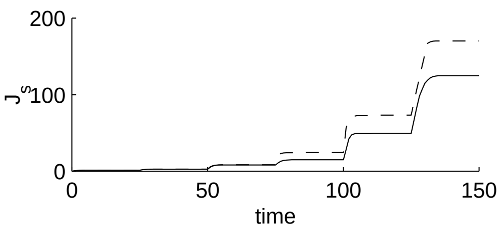

Figure 2: Simulation cost for the MPC algorithm of section 2 (solid line) and section 3 (dashed line)

is equivalent for both algorithms during this part of the simulation. Note that J_s corresponds to J_∞. When the setpoint for the output returns to zero at $t = 76$ and $t = 126$, the MPC algorithm of section 3 returns the maximal value for w. It does not take into account the influence of the nonlinearity, which in this case causes a more than proportional increase in u, i.e. u is steered into the saturated region. Therefore it is beneficial to use a smaller input w if possible, resulting in a much smaller u, at the cost of only a slightly slower response of the output. This is exactly what the MPC algorithm of section 2 is doing, which thus generates a smaller simulation cost, see figure 2. When the setpoint for the output goes from zero to two, at $t = 101$, a similar effect is observed. The smaller input w calculated by the MPC algorithm of section 2, leads to a slightly larger overshoot in the output, but because the associated control effort in u decreases more than proportional, this again will lead to a better simulation cost, see figure 2. The output responses for both algorithms are nearly the same, but since the algorithm of section 2 generates a much better control input, it outperforms the algorithm of section 3, demonstrating that taking the effect of the nonlinearity into account via a polytopic description offers an advantage over removing the nonlinearity from the control problem.

5 Conclusions

In this paper a new MPC algorithm for Hammerstein systems is presented. This algorithm transforms the static input nonlinearity into a polytopic description, which enables to use a robust linear MPC algorithm, consisting of a convex optimization problem. In a simulation example it is demonstrated that taking the nonlinearity into account via a polytopic description offers better control performance compared to an algorithm in which the nonlinearity is removed from the control problem. Nominal closed loop stability of the proposed MPC algorithm is guaranteed.

References

[1] F. Amato, F. Garofalo, L. Glielmo, and A. Pironti. Robust and quadratic stability via polytopic set covering. *International Journal of Robust and Nonlinear Control*, 5(8):745–756, 1995.

[2] H. H. J. Bloemen and T. J. J. van den Boom. Constrained linear Model-Based Predictive Control with an infinite control and prediction horizon. In *14-th Triennial World Congress, Beijing, China*, volume C, pages 229–234. IFAC, July 5–9 1999.

[3] H. H. J. Bloemen and T. J. J. van den Boom. MPC for Wiener systems. In *38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA*, pages 4595–4600, December 1999.

[4] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[5] E. Eskinat, S. H. Johnson, and W. L. Luyben. Use of Hammerstein models in identification of nonlinear systems. *AIChE Journal*, 37(2):255–268, 1991.

[6] K. P. Fruzzetti, A. Palazoglu, and K. A. McDonald. Nonlinear model predictive control using Hammerstein models. *Journal of Process Control*, 7(1):31–41, 1997.

[7] M. Kinnaert. Adaptive generalized predictive controller for MIMO systems. *International Journal of Control*, 50(1):161–172, 1989.

[8] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[9] R. S. Patwardhan, S. Lakshminarayanan, and S. L. Shah. Constrained nonlinear MPC using Hammerstein and Wiener models: PLS framework. *AIChE Journal*, 44(7):1611–1622, 1998.

[10] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63(2):331–349, 1996.