Interpolating Optimizing Process Control

Bjarne A. Foss*
Department of Engineering Cybernetics
The University of Trondheim - NTH
Norway
email: baf@itk.unit.no

S. Joe Qin
Department of Chemical Engineering
The University of Texas at Austin
qin@che.utexas.edu

August 9, 1996

Keywords: Model-based control, optimization, convexity, model predictive control, nonlinear systems, robustness, interpolation

Abstract

In this paper a new model-based optimizing controller for a set of nonlinear systems is proposed. The nonlinear model set is based on a convex combination of two bounding linear models. An optimal control sequence is computed for each of the two bounding models. The proposed control algorithm is based on a convex combination of the two control sequences. A novel feature in these two optimizations is an added constraint related to the feasibility of the 'other' bounding model. The control algorithm can for example be used in model predictive control. We provide robust feasibility guarantees and an upper bound on the optimal criterion if the bounding models are linear FIR models. Further, simulation examples demonstrate significant feasibility improvements in the case where the bounding models are general linear state-space models. The proposed method guarantees robust feasibility for a 1-step ahead prediction in the general case. This can be of interest in MPC applications.

1 Introduction

The combined use of dynamic models and optimization for process control offers a concept in which process knowledge can be linked to operational goals formulated by some optimization criterion. This concept has seen widespread use, particularly through the applications of model predictive control (MPC). MPC refers to a class of algorithms where an optimization problem is solved repetitively, at every new time-instant. Only the first part of the computed control sequence is applied to the system since a new optimal control sequence is computed and

*On sabbatical leave at The Department of Chemical Engineering, The University of Texas at Austin

applied at the next time-step. Several reviews of MPC technology exist, see for example Lee [4], Rawlings et al. [8] and Qin and Badgwell [7], the latter emphasizing industrial use of the technology. The interaction between control and optimization is discussed in an illuminating way in Mayne [6]. In this paper the divide between convex and non-convex problems is emphasized. In particular the problems arise by using nonlinear models since the optimization problem in these cases in general becomes non-convex.

State-of-the-art optimizing control is based on linear dynamic models and linear constraints on the control inputs and system outputs. Nonlinear optimizing control has been studied by Rawlings et al. [8] and Genceli and Nikalaou [3]. Further, some approaches were described by Bequette [1] in a somewhat earlier paper. These control strategies normally result in a nonconvex optimization problem.

In this paper we explore optimizing control based on nonlinear models. In particular, the goal is to derive an approach which can offer a smooth transition from linear optimizing control to nonlinear and robust optimizing control. By smooth transition we mean an approach which does not invoke the full mathematical 'machinery' of nonlinear optimizing control in the general case. This is particularly important from an industrial viewpoint as it simplifies the transition from the application of linear to nonlinear optimizing control. The smooth transition is accomplished by constraining the nonlinear optimization problem along three axes. First, the set of nonlinear models is limited. Second, constraints are added to the optimization problem to enhance feasibility of an optimal solution for a *set* of bounding linear models. Third, the control sequence is computed as a convex combination of control sequences computed on the basis of the bounding linear models.

The remainder of this paper is structured as follows: In the next section we formulate the problem. Following this, the theoretical foundation for the proposed control algorithm is developed. In section 4 three simulation examples are investigated to explore the theoretical findings, and to study the proposed method beyond the assumptions given by the theory. The proposed method is after this investigated in closed loop control. Finally, a comprehensive discussion is provided before the conclusions finalizes the paper.

2 Problem definition

2.1 Background

Assume an optimality criterion given by

$$\phi(\pi, \chi) = \sum_{i \in I_N} l(x_{i+1}, u_i) \tag{1}$$

where

$$\begin{aligned}
\pi &= \{u_0^T, \ldots, u_{N-1}^T\}, \quad u_i \in U \subseteq R^{m_u}, \quad \Pi = U \times \ldots \times U \\
\chi &= \{x_1^T, \ldots, x_N^T\}, \quad x_i \in X \subseteq R^{m_x}, \quad \mathcal{X} = X \times \ldots \times X \\
U, X &- \text{convex set}
\end{aligned}$$

$$\begin{aligned} l &: \quad R^{m_u \times m_y} \to R^{+}, \qquad l- \text{ convex function, } i \in I_N \\ I_N &= \{0, \ldots, N-1\} \end{aligned}$$

The optimality criterion is defined on a time horizon 0 to N. u_i denotes the control input to a system. The control input u_i is constant during the time span $[i, i+1)$. x_i denotes the system states at time i. Since U and X are convex sets, Π and $\mathcal{X}$ are convex sets. Further, ϕ is convex since l is convex.

We define two linear state-space models (Σ_1, Σ_0) and a set of nonlinear models (Σ_w) based on the interpolation of the two linear models.

$$\Sigma_1 : \quad h_1(x_{i+1}, x_i, u_i) = x_{i+1} - f_1(x_i, u_i) = x_{i+1} - A_1 x_i - B_1 u_i = 0, \quad i \in I_N \tag{2}$$

$$\Sigma_0 : \quad h_0(x_{i+1}, x_i, u_i) = x_{i+1} - f_0(x_i, u_i) = x_{i+1} - A_0 x_i - B_0 u_i = 0, \quad i \in I_N \tag{3}$$

$$\begin{aligned} \Sigma_w = \{h_w : h_w(x_{i+1}, x_i, u_i) &= w(x_i, u_i) h_1(x_{i+1}, x_i, u_i) + (1 - w(x_i, u_i)) h_0(x_{i+1}, x_i, u_i) = 0 \\ & \forall i \in I_N, \quad w \in W, \quad W = \{w \in \mathcal{C} : w(x_i, u_i) \in [0,1] \ \forall x_i \in X \ \forall u_i \in U\}\} \end{aligned} \tag{4}$$

h_w is a nonlinear function, constructed as convex combinations of h_1 and h_0. It should be noted that the continuous function w in general will depend on the states and the control inputs. We further observe that the interpolated model $h_w \in \Sigma_w$ is not generally a convex function since w may be any continuous function bounded by $[0, 1]$.

Finally, we define initial conditions.

$$x_0 - \text{ given} \tag{5}$$

The problem we want to address is to minimize (1) with respect to π based on the different constraints discussed above, hence we want to solve

$$\pi^o = arg\ min_{\pi \in \Pi}\ \phi(\pi, \chi) \tag{6}$$

subject to the constraints $\chi \in \mathcal{X}$, (5) using one of the models (2), (3) or (4).

The optimization problem above is generally nonconvex when a nonlinear model within Σ_w is used. This can cause problems, especially in on-line control where computation time can be an issue.

In MPC the minimization problem (6) is solved repetitively, at each time-step, with new initial conditions. Only the first control value of the π sequence is actually applied to the process. Furthermore, it is typical to parameterize the control sequence as follows:

$$\pi = \{u_0^T, \ldots, u_M^T, \ldots, u_M^T\}, \ M < N - 1 \tag{7}$$

This means that the control input is constant during the last part of the control input sequence.

The criterion function (1) does not cover all possible criteria, penalizing changes in the control input is for example not included. This type of change does not influence the results in this paper as long as the criterion function remains convex.

2.2 Interpolating control

We will in the following describe the proposed controller. First, we define two control sequences $\pi_1 \in \Pi$ and $\pi_0 \in \Pi$ and a set of interpolating controllers.

$$\Pi_\alpha = \{\pi_\alpha : \pi_\alpha = \alpha\pi_1 + (1-\alpha)\pi_0,\ \forall\alpha \in [0,1]\} \tag{8}$$

The control sequence π_α forms the basis for the controller. It should be noted that π_α is feasible since it is based on interpolation within a convex set, ie. $\Pi_\alpha \subseteq \Pi$. α may in general vary from one time-instant to another.

For later convenience we pursue with some definitions. χ_1 and χ_0 are the state sequences on the horizon $\{1, \ldots, N\}$ obtained by applying some control sequence $\pi \in \Pi$ to Σ_1 and Σ_0, respectively. $\chi_k^l = \{x_{k1}^l, \ldots, x_{kN}^l\}$ is the state sequence obtained by applying the control sequence π_l, $l \in \{1,0\}$ to the system Σ_k, $k \in \{1,0\}$. Further, we define the set of states associated with the model set Σ_w and the control sequence set Π_α.

$$\mathcal{X}_w^\alpha = \{\chi_w^\alpha : \chi_w^\alpha = \{x_{w1}^\alpha, \ldots, x_{wN}^\alpha\},\ \forall\, h_w \in \Sigma_w,\ \alpha \in [0,1]\} \tag{9}$$

Later we will in particular study the interpolated controller set Π_α where π_1 and π_0 are computed on the basis of the following two optimizing problems.

$$\pi_1^o = arg\ min_{\pi\in\Pi}\ \phi(\pi, \chi_1) \tag{10}$$

subject to the initial conditions (5), constraints $\chi_1 \in \mathcal{X}$, and Σ_1 given by (2).

$$\pi_0^o = arg\ min_{\pi\in\Pi}\ \phi(\pi, \chi_0) \tag{11}$$

subject to the initial conditions (5), constraints $\chi_0 \in \mathcal{X}$, and Σ_0 given by (3).

If we choose $\pi_1 = \pi_1^o$ and $\pi_0 = \pi_0^o$ the state sequences χ_1^1 and χ_0^0 are the optimal sequences for Σ_1 and Σ_0, respectively. Further, following the above notation ϕ_k^l is the criterion value obtained by applying the control sequence π_l^o, $l \in \{1,0\}$ to system Σ_k, $k \in \{1,0\}$. This means that ϕ_1^1 and ϕ_0^0 are the optimal criterion values for Σ_1 and Σ_0.

Note that we assume that the state sequences discussed above are based on the initial value (5).

3 Analysis

We will in this section analyze the use of the interpolating controller on the model set Σ_w.

3.1 Guaranteeing feasibility

There is no guarantee that the state constraints are satisfied on the horizon $\{1,\ldots,N\}$ when the control sequence $\pi_\alpha \in \Pi_\alpha$ is applied to the system $h_w \in \Sigma_w$. We will in the following include restrictions on the choice of π_1 and π_0, and on Σ_w so as to ensure that $\mathcal{X}_w^\alpha \subseteq \mathcal{X}$.

Before we do this we state the well-known connection between FIR-models and state space models.

Proposition 1 *An FIR-model of this form*

$$\begin{aligned} y_{i+1} &= C_{k0}u_i + \ldots + C_{kL}u_{i-L} = \tilde{C}_k\tilde{u}_i, \qquad k \in \{1,0,W\},\ i \in I_N \\ y_i &\in Y \subseteq R^{m_y} \end{aligned} \tag{12}$$

is equivalent to a state-space model on the form (2), (3) or (4) if

$$A_k = \begin{pmatrix} & \tilde{I} & & 0 \\ C_{k1} & \ldots & C_{kL} & 0 \end{pmatrix} \qquad \tilde{I} = \begin{pmatrix} 0 & \ldots & \ldots & 0 \\ I & 0 & \ldots & 0 \\ \vdots & \ddots & & 0 \\ 0 & \ldots & I & 0 \end{pmatrix} \qquad B_k = \begin{pmatrix} I \\ 0 \\ \vdots \\ 0 \\ C_{k0} \end{pmatrix}, \qquad k \in \{1,0,W\}$$

and $y_i = (0,\ldots,0,I)^T x_i$.

We note that C_{wj}, or $\tilde{C}_w$ ($w \in W$), and correspondingly A_k and B_k in general depend on the control inputs and states.

If we limit the bounding linear models to be FIR-models, and include feasibility constraints on the choice of π_1 and π_0 it is possible to guarantee feasibility in the sense that $\mathcal{X}_w^\alpha \subseteq \mathcal{X}$. This is shown below.

Theorem 1 *Assume two control sequences π_1 and π_0 such that $\chi_k^l \in \mathcal{X}\ \forall l \in \{0,1\}, k \in \{0,1\}$ for the initial conditions (5). Further, define a control input sequence set Π_α by (8).*

If the linear models Σ_1 and Σ_0 are FIR-models, and $\pi_\alpha \in \Pi_\alpha$, then $\chi_w^\alpha \in \mathcal{X}$. χ_w^α is defined in (9).

Proof:

We first use Proposition 1. Since the bounding models (2) and (3) are FIR-models, we transform them into the equivalent input-output model structure (12). The model set Σ_w contains nonlinear FIR-model and may, hence, also be transformed into a FIR-structure.

Let z_k^l denote the j'th element of the predicted output vector y_{i+1} of the FIR-model (12) at some $i \in I_N$ if we apply the control sequence π_l, $l \in \{1,0\}$ to the system Σ_k, $k \in \{1,0\}$. The assumptions on π_1 and π_0 guarantee that $z_k^l \in X_j$ where $X = X_1 \times \ldots \times X_{m_x}$.

Let z_w^α denote the j'th element of output vector y_{i+1} of the FIR-model (12) by applying a control sequence $\pi_\alpha \in \Pi_\alpha$ to the system $h_w \in \Sigma_w$.

Let c_1, c_0 and c_w denote the j'th row vector of $\tilde{C}_1$, $\tilde{C}_0$ and $\tilde{C}_w$, $w \in W$, respectively.

We first show that $c_1^T \tilde{u}_{\alpha i} \in X_j$.

$$c_1^T \tilde{u}_{\alpha i} = c_1^T(\alpha \tilde{u}_{1i} + (1-\alpha)\tilde{u}_{0i}) = \alpha c_1^T \tilde{u}_{1i} + (1-\alpha) c_1^T \tilde{u}_{0i} = \alpha z_1^1 + (1-\alpha) z_1^0 \tag{13}$$

$c_1^T \tilde{u}_{\alpha i}$ lies in X_j since it is a convex combination of two elements in the convex set X_j. An identical argument may be formulated for $c_0^T \tilde{u}_{\alpha i}$.

We now choose some element $\alpha \in [0,1]$ and show that $c_w^T \tilde{u}_{\alpha^i} \in X_j$ for any $w(x_i, u_i) \in [0,1]$.

$$c_w^T \tilde{u}_{\alpha i} = w(x_i, u_i) c_1^T \tilde{u}_{\alpha i} + (1 - w(x_i, u_i)) c_0^T \tilde{u}_{\alpha i} \tag{14}$$

We have above shown that $c_1^T \tilde{u}_{\alpha^i} \in X_j$ and $c_0^T \tilde{u}_{\alpha i} \in X_j$. Since X_j is a convex set, the convex combination $c_w^T \tilde{u}_{\alpha i}$ must lie in X_j.

The above result cannot be generalized to arbitrary A_1 and A_0 matrices. Further, it is critically dependent on the assumption that each of the control sequences π_1 and π_0 applied to either of the models Σ_1 or Σ_0 obey the state constraint on the horizon $\{1, \ldots, N\}$.

This theorem will form the basis for several corollaries. First, we use it in the case where π_1 and π_0 are based on the solution of some optimization problem.

Corollary 1 *Given the following control sequences*

$$\pi_1^o = arg\ min_{\pi \in \Pi}\ \phi(\pi, \chi_1) \tag{15}$$

subject to the initial conditions (5), constraints $\chi_1^1 \in \mathcal{X}$ (χ_1^1 is the state sequence obtained by applying $\pi_1 \in \Pi$ to Σ_1), and $\chi_0^1 \in \mathcal{X}$.

$$\pi_0^o = arg\ min_{\pi \in \Pi}\ \phi(\pi, \chi_0) \tag{16}$$

subject to the initial conditions (5), constraints $\chi_0^0 \in \mathcal{X}$, and $\chi_1^0 \in \mathcal{X}$.

If the linear models Σ_1 and Σ_0 are FIR-models, and $\pi_\alpha \in \Pi_\alpha$, then $\chi_w^\alpha \in \mathcal{X}$.

Proof: Since the control sequences defined in (15) and (16) satisfy the state constraints $\chi_k^l \in \mathcal{X}\ \forall l \in \{0,1\}, k \in \{0,1\}$ in Theorem 1, the Corollary follows directly from Theorem 1.

Remark: This corollary states the important result that an arbitrity control sequence $\pi_\alpha \in \Pi_\alpha$ applied to an arbitrary model $h_w \in \Sigma_w$ will satisfy the state constraints on the horizon $\{1, \ldots, N\}$ provided *additional hard constraints* are included in the optimization, ie. including $\chi_0 \in \mathcal{X}$ for computing π_1^o, and $\chi_1 \in \mathcal{X}$ for π_0^o.

If we limit our attention to 1-step ahead prediction, however, the result can be extended.

Corollary 2 *Given the control sequences π_1^o, π_0^o and π_α as defined in (15), (16) and (8). The state sequence which arises by applying π_α to $h_w \in \Sigma_w$ is denoted $\chi_w^\alpha = \{x_{w1}^\alpha, \ldots, x_{wN}^\alpha\}$, cf. (9).*

Then $x_{w1}^\alpha \in X$

Proof:

Let z_k^l denote element j of state vector x_{k1}^l which arises by applying the control input u_{l0}, $l \in \{1, 0\}$ to the system Σ_k, $k \in \{1, 0\}$, and z_w^α denote element j of state vector x_{k1}^α which arises by applying the control sequence $\pi_\alpha \in \Pi_\alpha$ to the system $h_w \in \Sigma_w$.

Let a_1, a_0 and a_w denote the j'th row vector of A_1, A_0 and A_w, and b_1, b_0 and b_w denote the j'th row vector of B_1, B_0 and B_w. Further, u_{10} and u_{00} denote the control input of π_1 and π_0 at time 0.

We need to show that $z_w^\alpha \in X_j$, where $X = X_1 \times \ldots \times X_{m_x}$.

We first show that $a_1^T x_0 + b_1^T u_{\alpha i} \in X_j$.

$$\begin{aligned} a_1^T x_0 + b_1^T u_{\alpha 0} &= a_1^T(\alpha x_0 + (1-\alpha)x_0) + b_1^T(\alpha u_{10} + (1-\alpha)u_{00}) \\ &= \alpha(a_1^T x_0 + b_1^T u_{10}) + (1-\alpha)(a_1^T x_0 + b_1^T u_{00}) \\ &= \alpha z_1^1 + (1-\alpha) z_1^0 \end{aligned} \tag{17}$$

$a_1^T x_0 + b_1^T u_{\alpha 0}$ lies in X_j since it is a convex combination of two elements in the convex set X_j. An identical argument may be formulated for $a_0^T x_0 + b_0^T u_{\alpha 0}$.

We now choose some element $\alpha^\in [0,1]$ and show that $a_w^T x_0 + b_1^T u_{\alpha^0} \in X_j$ for any $w(x_i, u_i) \in [0,1]$.

$$a_w^T x_0 + b_w^T u_{\alpha^0} = w(x_i,u_i)(a_1^T x_0 + b_1^T u_{\alpha^0}) + (1 - w(x_i,u_i))(a_0^T x_0 + b_0^T u_{\alpha^0}) \tag{18}$$

Since $a_1^T x_0 + b_1^T u_{\alpha^0} \in X_j$ and $a_0^T x_0 + b_0^T u_{\alpha^0} \in X_j$, and X_j is a convex set, the convex combination $a_w^T x_0 + b_w^T u_{\alpha^0}$ must lie in X_j.

Remark: This corollary shows that the 1-step ahead prediction will satisfy the state constraints for *any linear models* Σ_1 and Σ_0 if π_1^o and π_0^o are computed using the *additional constraints* as in (15) and (16). This implies that an MPC controller based on π_α and (15) and (16) always will be feasible even though the constraints may be violated on the optimization horizon. The reason is that the MPC controller only applies the the first control input before recalculating the control input at the next time-step.

3.2 Robust performance

Theorem 1 provides a possibility to obtain an upper bound on the following worst case scenario. The upper bound is defined by:

$$\bar{\phi} = \sup_{h_w \in \Sigma_w,\ \pi_\alpha \in \Pi_\alpha} \phi(\pi_\alpha, \chi_w^\alpha) \tag{19}$$

This result is formulated by the following lemma.

Corollary 3 *Suppose the control sequence $\pi_\alpha \in \Pi_\alpha$ is applied to the system $h_w \in \Sigma_w$ with initial conditions (5). Σ_1 and Σ_0 are FIR-models.*

Further, suppose the optimality criterion function l is given by some separable norm function, ie. $l(x_{i+1}, u_i) = \|y_{i+1}\|_Q^2 + \|u_i\|_R^2$, where y_i is the output of the FIR-model formulation (12).

Then an upper bound for $\bar{\phi}$ in (19) is given by

$$\bar{\phi} \leq \sum_{i \in I_N} \overline{Cu_i} + \bar{u}_i \tag{20}$$

where $\overline{Cu_i} = max_{k,l\in\{1,0\}} \|\tilde{C}_k \tilde{u}_{li}\|_Q^2$ and $\bar{u}_i = max_{l\in\{1,0\}} \|u_{li}\|_R^2$.

Proof:

We first compute a upper bound for $\|y_{i+1}\|_Q^2$.

$$\begin{aligned}
\|y_{i+1}\|_Q^2 &= \|\tilde{C}_w \tilde{u}_{\alpha i}\|_Q^2 = \|(w\tilde{C}_1 + (1-w)\tilde{C}_0)\tilde{u}_{\alpha i}\|_Q^2 \\
&\leq [w\|\tilde{C}_1 \tilde{u}_{\alpha i}\|_Q + (1-w)\|\tilde{C}_0 \tilde{u}_{\alpha i}\|_Q]^2 \\
&\leq max_{k\in\{1,0\}} \|\tilde{C}_k \tilde{u}_{\alpha i}\|_Q^2 \\
&= max_{k\in\{1,0\}} \|\tilde{C}_k [\alpha \tilde{u}_{1i} + (1-\alpha)\tilde{u}_{0i}]\|_Q^2 \\
&\leq max_{k\in\{1,0\}} [\alpha \|\tilde{C}_k \tilde{u}_{1i}\|_Q + (1-\alpha)\|\tilde{C}_k \tilde{u}_{0i}\|_Q]^2 \\
&\leq max_{k,l\in\{1,0\}} \|\tilde{C}_k \tilde{u}_{li}\|_Q^2 \\
&= \overline{Cu_i}
\end{aligned} \tag{21}$$

A upper bound on $\|u_i\|_R^2$ is easily computed.

$$\begin{aligned}
\|u_i\|_R^2 &= \|u_{\alpha i}\|_R^2 = \|(\alpha u_{1i} + (1-\alpha)u_{0i}\|_R^2 \\
&\leq [\alpha\|u_{1i}\|_R + (1-\alpha)\|u_{0i}\|_R]^2 \\
&\leq max_{l\in\{1,0\}} \|u_{li}\|_R^2 \\
&= \bar{u}_i
\end{aligned} \tag{22}$$

By these expressions the corollary is proved.

The important consequence of the above lemma is the fact that an easily computed upper bound on robust performance can be found. This can hence be used as a measure for robust performance.

To compute the bound we use the output y_i instead of x_i. This is no important limitation since y_i forms the output part of the states of a FIR-model. A weighting of y_i is, hence, the obvious choice for the criterion.

4 Examples

We will in this section further explore the findings in the above sections through three numerical examples. All the models, except for Σ_0 in example 3, are typical models found in the process control problems. In the first example we will investigate the method on a nonlinear FIR-model. In examples 2 and 3 we will study the method beyond the assumptions given by the theory. In particular, we investigate situations where there is no theoretical guarantee for a feasible solution.

4.1 Example 1: FIR models

We assume that a single-input single-output system is given by some model $h_w \in \Sigma_w$, cf. (4), with bounding FIR-models given by

$$\begin{aligned} y_{i+1} &= c_{k0}u_i + \ldots + c_{k4}u_{i-4} = \tilde{c}_k\tilde{u}_i, \qquad k \in \{1, 0\} \qquad (23) \\ \tilde{c}_1 &= \{-0.10, 0.30, 0.40, 0.30, 0.10\} \\ \tilde{c}_0 &= \{0.20, 0.20, 0.15, 0.10, 0.05\} \end{aligned}$$

Proposition 1 gives the equivalent state-space formulation.

The two bounding models resemble a system with an inverse response and a stationary gain of 1.0, and a 1st order response and a stationary gain of 0.7, respectively.

We use the well-known quadratic optimality criterion, add hard constraints on the system output and control input, and assign initial conditions.

$$\begin{aligned} \phi(\pi, \chi) &= 0.5\sum_{i=1}^{N} y_i^2 + ru_{i-1}^2 \\ N &= 7 \\ r &= 0.01 \\ U &= [-1, 1] \\ X &= [-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [0.00, 1.01] \\ x_0 &= (1, 1, 1, 1, 0)^T \end{aligned}$$

The state-space model has 5 states. The first 4 states are delayed control inputs while the last state is the system output, equivalent to y_i in (23). Hence, the state constraints X limits the system output to $[0, 1.01]$. Note that the system output does not depend on the last element of x_0 since it does not have any auto-regressive terms.

We will first study the feasibility of the proposed controller, (8), by studying the output trajectories. This is done in three steps.

1. The control sequences $\pi_1^o = (0.90, 0.60, -0.70, 0.33, 0.06, -0.11, 0.11)^T$ and $\pi_0^o = (0.90, 0.60, -0.70, 0.13, 0.28, -0.17, 0.08)^T$ are computed using (15), (16), ie. with the additional constraints in place. 6 interpolating control sequences π_α are generated by choosing $\alpha = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

2. We choose 6 models within Σ_w by choosing constant interpolation weights $w = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. The 6 models are simulated using the 6 interpolating control sequences as inputs. This gives a total of $6 \times 6 = 36$ simulation runs. The system is simulated from 0 to 7, ie. equal to the optimization horizon.

3. The system outputs for each simulation run is checked against the constraints, ie. $[0.00, 1.01]$. The result is shown in the matrix in Table 1. The 36 columns relate to each of the simulation runs, while the 8 lines are related to time instants $\{0, 1, \ldots, 7\}$.

   0 indicates no constraint violation, while 1 or 2 signifies a violation of the minimum or maximum allowable output, respectively. Table 1 shows that there is no violation of output constraints. This is in accordance with Theorem 1.

Corollary 3 gives an upper bound on the optimality criterion, for this example we compute $\bar{\phi} = 1.93$. The criterion values for the 36 simulation runs in Table 1 lie between 0.47 and 1.61. This indicates that the upper bound need not be very conservative.

To study the impact of the additional constraint, ie. $\chi_0 \in \mathcal{X}$ to compute π_1^o and $\chi_1 \in \mathcal{X}$ to compute π_0^o, we remove these constraints and perform the computations in item 1-3. This means computing π_1^o and π_0^o according to (10) and (11). The results are shown in Table 2. There are in these cases many violations of the output constraints.

In the latter case the control sequences are $\pi_1^o = (0.90, 0.60, -0.70, 0.33, 0.06, -0.12, 0.09)^T$ and $\pi_0^o = (-1.00, -0.34, 0.34, 0.17, 0.00, -0.17, 0.00)^T$. They are very different from each other. This is not the case in item 1 above, in that case the control sequences are identical except for the last control input. The reason for this are the added constraints which limit the possible choices of control inputs.

4.2 Example 2: Bounding 2nd order models

We again assume that a model set Σ_w defines a single-input single-output system. In this example the bounding linear models are given by

$$A_1 = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ 0.00 & -0.79 & 1.78 \end{pmatrix} \qquad B_1 = \begin{pmatrix} 1.00 \\ 0.00 \\ 0.01 \end{pmatrix}$$

$$A_0 = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ 0.01 & -0.855 & 1.85 \end{pmatrix} \qquad B_0 = \begin{pmatrix} 1.00 \\ 0.00 \\ 0.00 \end{pmatrix}$$

The 3rd state is the system output, hence $y_i = (0,0,1)^T x_i$. Σ_1 is a 2nd order model with two time constants of about 6 and 15 time-steps, no time-delay and a stationary gain of 1.0. Σ_0 is a 2nd order model with two time-constants of about 10 and 20, a time-delay of 1 and a stationary gain of 2.0.

To elaborate on the model set Σ_w, let $w(x_i, u_i) \in [0,1]$ be a constant. Hence, the model set Σ_w consists of only linear models. The stationary gain, time-delay and time-constants for this model set are shown in Fig.1. We observe that there is a close to linear change in all the characterizing variables.

The optimality criterion, hard constraints on the system output and control inputs, and initial conditions are given below.

$$\begin{aligned} \phi(\pi,\chi) &= 0.5 \sum_{i=1}^{N} y_i^2 + r u_{i-1}^2 \qquad (24) \\ N &= 6 \\ r &= 0.01 \\ U &= [-10, 10] \\ X &= (-\infty,\infty) \times (-\infty,\infty) \times [0.00, 1.01] \\ x_0 &= (1,1,1)^T \end{aligned}$$

The state constraints X limits the system output to $[0, 1.01]$.

The control sequences $\pi_1^o = (-7.48, -5.14, -3.28, -1.87, -0.89, -0.28)^T$ and $\pi_0^o = (-7.49, -4.84, -2.80, -1.35, -0.43, 0.00)^T$ are computed using (15), (16), ie. with the additional constraints in place. We compute 6 interpolating control sequences π_α by choosing $\alpha = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, and choose 6 models within Σ_w by selecting constant interpolation weights $w = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Again we simulate a total of $6 \times 6 = 36$ runs and check the system outputs against the constraints. There is no violation at any time-step in any of the 36 simulation runs.

We also compute the control sequences omitting the additional constraints, ie. using (10), (11). In this case we obtain identical control sequences as above. Hence, in this case the additional constraints make no difference on the computed control sequences.

4.3 Example 3: Bounding 2nd order models

To make the problem more challenging we will in this example study a set of systems where the bounding models have different types of behaviour. Σ_1 is equal to Σ_1 in the above example, while Σ_0 is a 2nd order model with oscillatory modes. The 'time-constant' of the oscillatory modes is 29 time-steps and the oscillation period is $5-6$ time-steps. The stationary gain of Σ_0 is 1.0 and the time-delay is 1. The bounding models are described by:

$$A_1 = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ 0.00 & -0.79 & 1.78 \end{pmatrix} \qquad B_1 = \begin{pmatrix} 1.00 \\ 0.00 \\ 0.01 \end{pmatrix}$$

$$A_0 = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \\ 0.62 & -0.32 & 0.70 \end{pmatrix} \qquad B_0 = \begin{pmatrix} 1.00 \\ 0.00 \\ 0.00 \end{pmatrix}$$

To get a feel for the model set Σ_w we simulate a step response for 6 models with constant w-values, cf. Fig.2. The stationary gain does not change , the dynamics do, however, change a lot. The dynamics are particularly sensitive to w close to 1.

The optimality criterion, constraints and initial conditions are equal to those in (24). The output is again equal to $y_i = (0,0,1)^T x_i$.

The control sequences $\pi_1^o = (-0.61, 0.52, 0.00, 0.00, 0.00, -0.84)^T$ and $\pi_0^o = (-0.59, 0.50, 0.01, 0.00, 0.00, 0.00)^T$ are computed using (15), (16). Again we simulate a total of $6 \times 6 = 36$ runs and check the system outputs against the constraints. The result is shown in Table 3. We observe that in many of the late time-steps (4 - 6) there are violations of the minimum value of the system output.

To study the impact of the additional constraints, we remove these constraints. This gives the control sequences $\pi_1^o = (-7.48, -5.14, -3.28, -1.87, -0.89, -0.28)^T$ and $\pi_0^o = (-0.59, 0.50, 0.01, 0.00, 0.00, 0.00)^T$. The results are shown in Table 4, the number of violations is substantially larger than in Table 3. The numbers behind these tables accentuates this difference. The minimum output constraint, ie. 0, is only marginally violated in Table 3. Most of the violating outputs lie in the range $[-0.10, -0.05]$ with a minimum value of -0.16. In Table 4 most of the violating outputs lie in the range $[-3, -2]$ with a minimum value of -5.92.

To further illuminate the impact of the additional constraints we show the criterion values for the 36 simulation runs in the cases with and without the additional constraints in Tables 5 and 6. The $(1,1)$ and $(6,6)$ -entries show the nominal cases for Σ_1 and Σ_0, ie. ϕ_1^1 and ϕ_0^0. We observe, as should be expected, that the optimal criterion value in these nominal cases may increase due to the additional constraints. More interesting, however, is the fact that the added constraints seem to have a significant robustifying effect. The worst case criterion values show a dramatic improvement, 48.70 to 2.57. The improvements are especially pronounced for high α-values, ie. by comparing the lower left part of Tables 5 and 6.

5 Closed loop behaviour

As discussed earlier the proposed method can be used for closed loop control. In this section we will investigate this by extending the simulation examples and discuss issues related to feasibility, closed loop stability and robustness.

5.1 Example 1

We extend Example 1 by using our method in MPC. The system, optimization criterion, constraints and initial values are unchanged. Fig. 3 shows some typical results. In this simulation the system is given by Σ_1, ie. $w \in 1$, while the control input is computed by choosing $\alpha = 0$, cf. (8), ie. the control input is based on Σ_0. We observe that the output constraints are satisfied when the added constraints are used. This is, however, not the case if these added constratins are omitted.

The above might lead to the conclusion that an initial feasible solution implies a feasible solution at any time. This cannot be guaranteed. This is seen by performing the same simulation as above but increasing the gain of the simulated system. We increase the gain of Σ_1 by a factor of 5. Fig.4a shows a violation of the output constraints at certain time instances. The reason for this is that there exists some $x \in X$ where no feasible π can be found, ie. no $\pi \in \Pi$ such that the predicted states satisfy the state constraints for *both* bounding linear models. It should be noted that this shows a rather extreme situation as the 1st coefficient of the FIR-model can vary between -0.50 and 0.20, cf. (23) and remember that the gain of Σ_1 is multiplied by 5.

Comparing Fig.4a and b indicates that the added constraints have a robustifying effect even though the constraints are violated in both cases.

5.2 Example 2

To further gain insight we used our method on the system in Example 2. The system, optimization criterion, constraints and initial values are unchanged. Fig. 5 shows some typical results. In this simulation the system is given by Σ_0, ie. $w \in 0$, while the control input is computed by choosing $\alpha = 1$, cf. (8), ie. the control input is based on Σ_1. We can make two observations from the results. First, the output constraints are satisfied when the added constraints are invoked. This is in accordance with Corollary 2 since a feasible solution can be found at all the simulated time steps.

Second, as in Example 1 the results indicate that the added constraints have a robustifying effect on the closed loop controller.

6 Discussion

The proposed method is an attempt to offer a smooth transition from linear to nonlinear optimizing control as for example nonlinear MPC. In our opinion a smooth transition is important to promote industrial applications of nonlinear model-based optimizing control. The features that underline the smooth transition are: (i) The control algorithm uses well-known optimization based on linear models as its building-blocks. (ii) A control sequence is computed by smooth interpolation, a commonly applied engineering principle.

The proposed controller can be seen as an extension of linear optimizing control from both a theoretical as well as an applicational perspective. First, we elaborate on the theoretical results.

- A key and novel feature of the proposed controller is the added constraints included for the two minimization problems, ie. constraint $\chi_0^1 \in \mathcal{X}$ in (15), and $\chi_1^0 \in \mathcal{X}$ in (16). This makes it possible to guarantee robust feasibility on some given optimization horizon when the bounding models are linear FIR-models provided that there exists a feasible control input sequence for each of the two bounding linear models. It should be noted that the model set Σ_w may have autoregressive terms even if the bounding models are FIR-models, since the weighting function may depend on x_i.

- An easily computable upper bound can be found for the optimality criterion in the case of nonlinear FIR-models. This provides a tool to investigate robust performance of the proposed controller scheme for this model class.

- Corollary 2 shows robust feasibility for the 1-step ahead prediction for any type of bounding linear models if π_1^o and π_0^o are computed using the additional constraints. This implies that an MPC controller based on the proposed approach always will be feasible if there exists a feasible control input sequence for each of the two bounding linear models at each time-step. The reason for this is that the optimal control sequence is recomputed at every time-step. It will be, however, neither optimal nor necessarily stable if the constraints are violated on the optimization horizon beyond the 1st prediction step.

- The theoretical results apply whether the underlying system $h_w \in \Sigma_w$ is linear or nonlinear. The richness of the nonlinear model class covered by Σ_w and the relevance of this model class for process control problems need further investigation.

Second, we discuss the simulation results.

- Example 1 with bounding FIR-models support the theoretical findings. It also clearly demonstrates, cf. Tables 1 and 2, the importance of the added constraints. Further, the computed upper bound on the optimality criterion seems to be reasonable in the sense that it is not overly conservative.

- Examples 2 and 3 investigate the results beyond the bounds of the theory. This is of interest since practical use of controllers usually violates theoretical assumptions at some stage. In Example 2 the bounding models are quite similar in the sense that they both are 2nd order damped systems, while the model characteristics are more divided in Example 3. The findings in Example 2 indicate that constraint handling need not be difficult when the bounding model are similar, in this example control computation without the added constraints gave only feasible solutions. This picture changed significantly in Example 3, the positive effect of the added constraints was very pronounced here, even though there is no feasibility guarantee in this case.

- Corollary 1, ie. the result on feasibility of the 1-step ahead prediction, is supported by the simulations. There were no constraint violation at time $i = 1$ with the added constraints, see Tables 1 and 3. This was not necessarily the case if these constraints were removed as can be seen in Table 2.

- The closed loop simulations substantiates the open loop findings. The added constraints do seem to improve robust stability and performance. It is important to note that the feasibility guarantee depends on the existence of a feasible control input sequence for each of the two bounding linear models. This is demonstrated in Fig.4a.

Finally, we elaborate on some additional issues.

- An open question on the proposed approach is the richness of the model set Σ_w. It might be necessary to include more than two bounding linear model, ie. let the h_w be a convex combination of several, say M, linear models. In this case the model set can comprise a large class of nonlinear systems as shown in Johansen and Foss [5]. The problem that arises is the complexity increase in our algorithm since the number of added constraints to compute each of the optimal control sequence increases. In addition, the control sequence π_α will be a convex combination of M control sequences.
- There exists methods and tools for identifying models on a format like (4), see eg. [2].
- An alternative approach to the one pursued in this paper is to compute a control sequence based on one nominal model with added constraints related to one or more bounding models.
- This paper does not provide a closed loop stability proof for the proposed method. The findings do, however, provide a basis for seeking such a result.
- Nonlinear optimizing control provides a nonconvex optimization problem. By this, there is no guarantee that the global optimum can be found. Hence, an alternative approach is to search for a suboptimal solution with some performance guarantee. This can be done by limiting the search to the set Π_α proposed in this paper. $\alpha \in [0,1]$ might be computed in one of three ways. (1) A gain-scheduling approach can be applied if it is possible to schedule α on some process variables. (2) α is assumed to be time-invariant, and it is computed using an off-line or on-line estimation scheme based on some performance measure. (3) α is time-varying and is computed using a recursive on-line algorithm. This may pave the way for a reliable adaptive MPC controller, since only one time-varying parameter is required. In practice, robustness of adaptive control is difficult to obtain when many parameters are estimated on-line.

7 Conclusions

A new model-based optimizing controller for a set of nonlinear systems is proposed. Robust feasibility with respect to hard state and/or output constraints is guaranteed on the optimization horizon for a set of nonlinear FIR-models, while significant feasibility improvements are encountered in the case where the bounding models are general linear state-space models. Finally, the method guarantees robust feasibility for a 1-step ahead prediction in the general case. This can be of interest in MPC-applications.

References

[1] B.W.Bequette. Nonlinear control of chemical processes: A review. *Ind.Eng.Chem.Res.*, 30:1391–1413, 1991.

[2] B. A. Foss and T. A. Johansen. Local modelling as a tool for semi-empirical or semi-mechanistic process modeling. In A. Bulsari, editor, *Neural Networks for Chemical Engineers*, pages 297–334. Elsevier, Amsterdam, 1995.

[3] H. Genceli and M. Nikalaou. Design of robust constrained model-predictive controllers with volterra series. *AIChE Journal*, 41:2098–2107, 1995.

[4] J.H.Lee. Recent advances in model predictive control and other related areas. In *Preprints CPC-V, Lake Tahoe, USA*, 1996.

[5] T. A. Johansen and B. A. Foss. Constructing NARMAX models using ARMAX models. *Int. J. Control*, 58:1125–1153, 1993.

[6] D. Q. Mayne. Optimization in model based control. In *Preprints IFAC Symposium DYCORD, Helsingor, Denmark*, 1995.

[7] S. Joe Qin and T.A. Badgwell. An overview of industrial model predictive control technology. In *Preprints CPC-V, Lake Tahoe, USA*, 1996.

[8] J. B. Rawlings, E. S. Meadows, and K. R. Muske. Nonlinear model predictive control: A tutorial and survey. In *Preprints IFAC Symposium ADCHEM, Kyoto, Japan*, pages 203–214, 1994.

Time	Model 1	$w = 0.8$	$w = 0.6$	$w = 0.4$	$w = 0.2$	Model 0
	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$
i=0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=1	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=2	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=3	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=4	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=5	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=6	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=7	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0

Table 1: Example 1 - Bounding FIR-models. Each column represents a simulation run using the control sequence π_α on a model $h_w \in \Sigma_w$. The system outputs of each simulation run are checked against the constraints at each time-step. 0 indicates no output constraint violation. 1 or 2 signifies a violation of the minimum or maximum allowable output, respectively.

Time	Model 1	$w = 0.8$	$w = 0.6$	$w = 0.4$	$w = 0.2$	Model 0
	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$
i=0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=1	0 2 2 2 2 2	0 0 0 0 0 2	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=2	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=3	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=4	0 0 0 1 1 1	0 0 0 0 1 1	0 0 0 0 1 1	0 0 0 0 1 1	0 0 0 0 1 1	0 0 0 0 0 0
i=5	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 0 0 0
i=6	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=7	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0

Table 2: Example 1 - Bounding FIR-models. Each column represents a simulation run using the control sequence π_α on a model $h_w \in \Sigma_w$. The system outputs of each simulation run are checked against the constraints at each time-step. 0 indicates no output constraint violation. 1 or 2 signifies a violation of the minimum or maximum allowable output, respectively.

Time	Model 1	$w = 0.8$	$w = 0.6$	$w = 0.4$	$w = 0.2$	Model 0
	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$
i=0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=1	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=2	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=3	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=4	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1 1	1 1 1 1 1 1	0 0 0 0 0 0
i=5	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	0 0 0 0 0 0
i=6	0 0 0 0 0 0	0 0 0 0 0 0	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	0 0 0 0 0 0

Table 3: Example 3 - Bounding 2nd order models. Each column represents a simulation run using the control sequence π_α on a model $h_w \in \Sigma_w$. The system outputs of each simulation run are checked against the constraints at each time-step. 0 indicates no output constraint violation. 1 or 2 signifies a violation of the minimum or maximum allowable output, respectively.

Time	Model 1	$w = 0.8$	$w = 0.6$	$w = 0.4$	$w = 0.2$	Model 0
	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$	$\pi_1^o, \ldots, \pi_0^o$
i=0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=1	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0
i=2	0 0 0 0 0 0	1 1 0 0 0 0	1 1 1 1 0 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0
i=3	0 0 0 0 0 0	1 1 1 1 0 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0
i=4	0 0 0 0 0 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 0
i=5	0 0 0 0 0 0	1 1 1 1 1 0	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 0
i=6	0 0 0 0 0 0	1 1 1 1 1 0	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 0

Table 4: Example 3 - Bounding 2nd order models. Each column represents a simulation run using the control sequence π_α on model a model $h_w \in \Sigma_w$. The system outputs of each simulation run are checked against the constraints at each time-step. 0 indicates no output constraint violation. 1 or 2 signifies a violation of the minimum or maximum allowable output, respectively.

ϕ	π_1^o	$\alpha = 0.8$	$\alpha = 0.6$	$\alpha = 0.4$	$\alpha = 0.2$	π_0^o
Model 1	2.57	2.57	2.57	2.57	2.57	2.57
$w = 0.8$	1.06	1.06	1.06	1.06	1.06	1.06
$w = 0.6$	0.74	0.74	0.74	0.74	0.74	0.74
$w = 0.4$	0.60	0.60	0.60	0.60	0.60	0.60
$w = 0.2$	0.53	0.53	0.53	0.53	0.53	0.53
Model 0	0.51	0.51	0.50	0.50	0.50	0.50

Table 5: Example 3 - Bounding 2nd order models. The criterion values using the control sequence π_α on model $wh_1 + (1-w)h_0$ for different values of α and w. The extra constraints are included.

ϕ	π_1^o	$\alpha = 0.8$	$\alpha = 0.6$	$\alpha = 0.4$	$\alpha = 0.2$	π_0^o
Model 1	1.67	1.71	1.81	2.00	2.25	2.57
$w = 0.8$	19.28	12.05	6.61	2.97	1.12	1.06
$w = 0.6$	38.52	24.64	13.93	6.37	1.98	0.74
$w = 0.4$	45.94	29.52	16.78	7.71	2.32	0.60
$w = 0.2$	48.08	30.92	17.59	8.08	2.39	0.53
Model 0	48.70	31.34	17.84	8.20	2.42	0.50

Table 6: Example 3 - Bounding 2nd order models. The criterion values using the control sequence π_α on model $wh_1 + (1-w)h_0$ for different values of α and w. The extra constraints are not included.

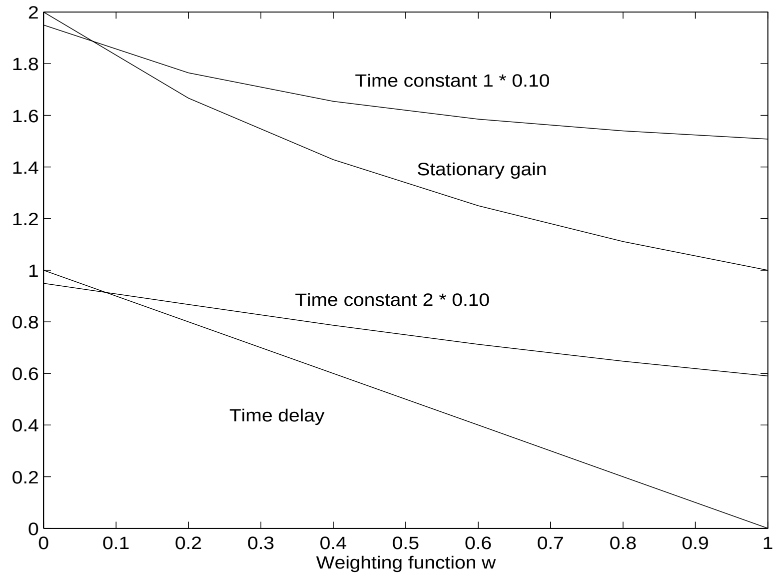

Figure 1: Example 2 - Bounding linear models. The gain, time-delay and time-constants of the models with constant w-values are depicted.

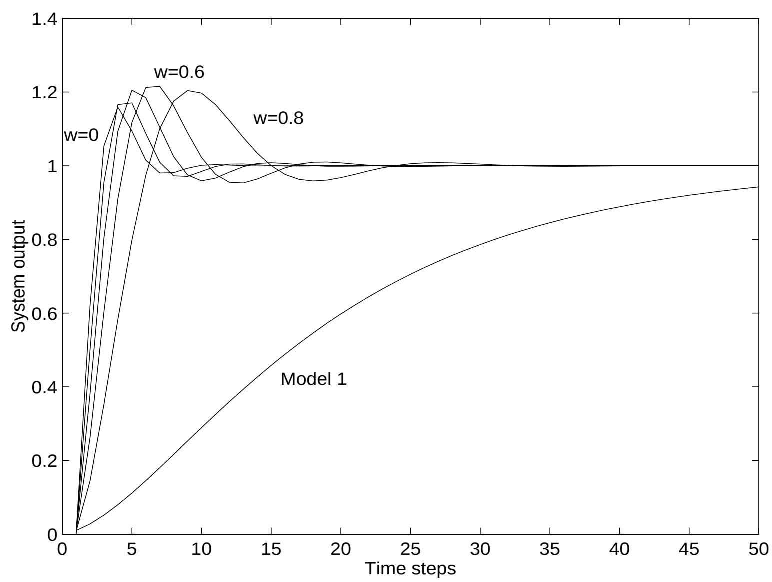

Figure 2: Example 3 - Bounding linear models. There is a step in u (u goes from 0 to 1) at time 1. The initial conditions are $x = [0,0,0]^T$. Step responses are shown for the 6 models used in the computations in Example 3, ie. $w = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

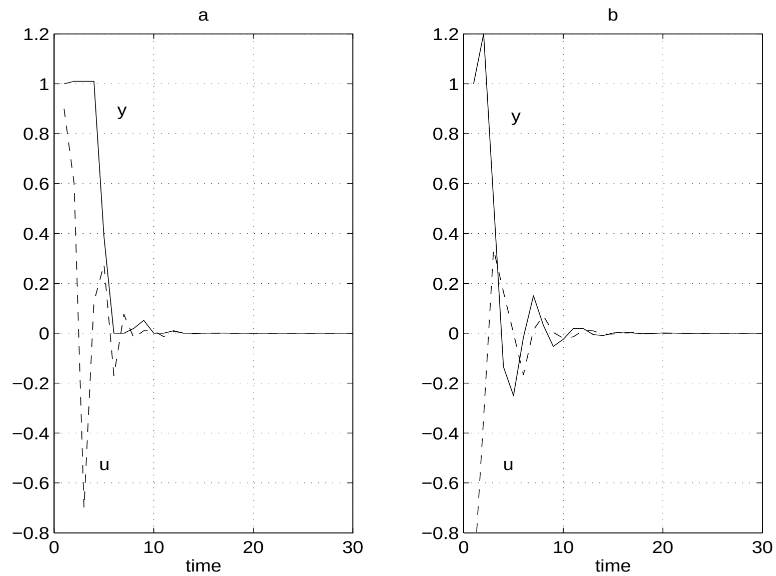

Figure 3: Example 1 - MPC. The simulated system is given by Σ_1, ie. $w \in 1$, while the control input is computed by choosing $\alpha = 0$ (8), ie. the control input is based on Σ_0. In a the added constraints are invoked, while these are omitted in b. The figures show the system output and the control input.

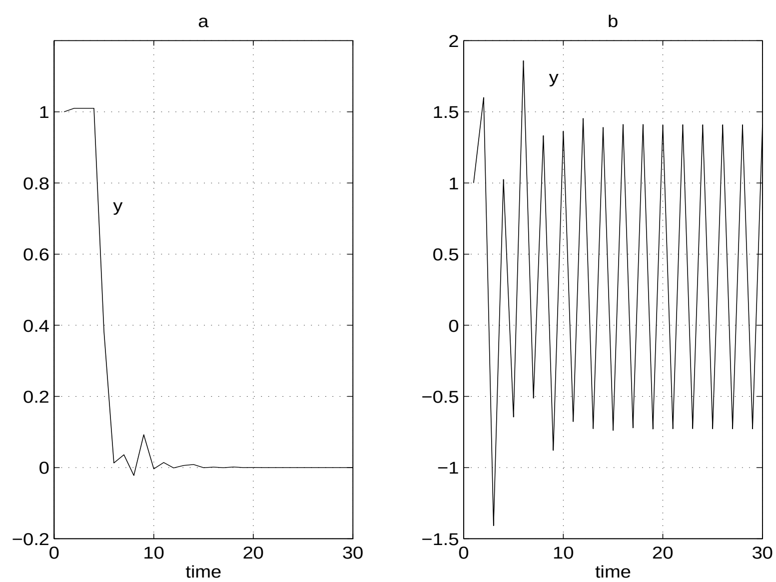

Figure 4: Example 1 - MPC. Same simulation as in Fig.3, but the gain of the simulated system is increased by a factor of 5. In a the added constraints are invoked, while these are omitted in b. The figures show the system output and the control input (note the axis).

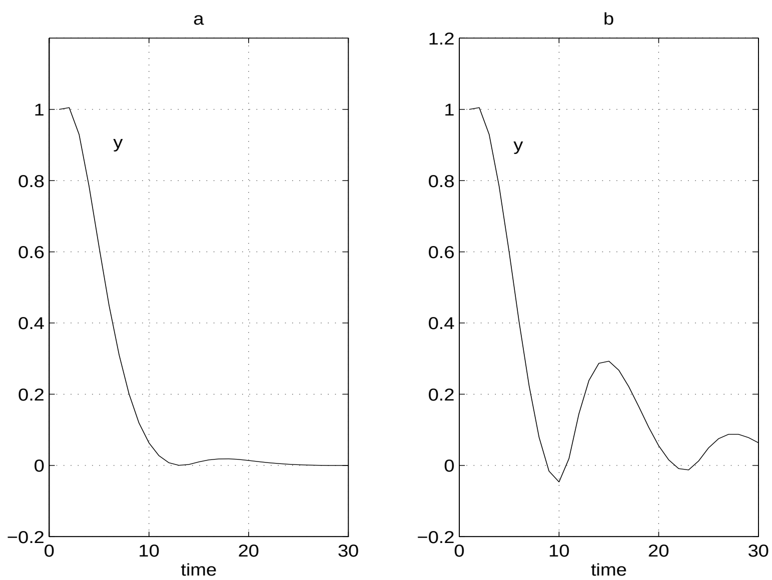

Figure 5: Example 2 - MPC. The system is given by Σ_0, ie. $w \in 0$, while the control input is computed by choosing $\alpha = 1$ (8), ie. the control input is based on Σ_1.In a the added constraints are invoked, while these are omitted in b. The figures show the system output.