Adaptive receding horizon control for constrained MIMO systems ⋆

Marko Tanaskovic [a], Lorenzo Fagiano [a,b], Roy Smith [a], Manfred Morari [a]

[a] *Automatic Control Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland.*

[b] *ABB Switzerland Ltd., Corporate Research, Baden-Daettwil, Switzerland.*

Abstract

An adaptive control algorithm for open-loop stable, constrained, linear, multiple input multiple output systems is presented. The proposed approach can deal with both input and output constraints, as well as measurement noise and output disturbances. The adaptive controller consists of an iterative set membership identification algorithm that provides a set of candidate plant models at each time step and a model predictive controller that enforces input and output constraints for all the plants inside the model set. The algorithm relies only on the solution of standard convex optimization problems that are guaranteed to be recursively feasible. The experimental results obtained by applying the proposed controller to a quad-tank testbed are presented.

Key words: Adaptive control; Self tuning control; Learning control; Set membership identification; Model predictive control; Control of constrained systems; Impulse response.

1 Introduction and motivation

The idea of adaptive control is to carry out real time controller adjustments, on the basis of input-output data collected on-line. Adaptive strategies may be used to control time invariant systems for which the model identification experiments are complex and expensive, and/or when the control algorithm needs to be applied to many copies of the system, which are affected by uncertainty due to production variability. In these cases an adaptive control approach can reduce or eliminate the need for time-consuming tuning of each produced unit, hence decreasing the production costs. In addition to uncertainty, most real world systems have control variables (voltage, current, flow, etc.) that are constrained due to physical limitations of the actuators. Moreover, many systems also require that the controlled variables satisfy certain constraints due to safety reasons or due to physical limitations. However, to derive an adaptive control approach for systems with constraints is a non-trivial task that requires the integration of on-line system identification and constrained control. Crucial aspects of both fields interact and give rise to challenging issues, like the need to ensure constraint satisfaction while the model of the plant dynamics is being updated. Finally, despite the fact that a well established theory for adaptive control has been developed [2], there are few results on adaptive control of constrained multiple input multiple output (MIMO) systems [18]. Hence, further research in adaptive control of MIMO systems subject to constraints could broaden the range of the practical applications in which adaptive control may be used.

Model predictive control (MPC) is a powerful technique for controlling constrained MIMO systems [11]. While the topic of MPC in the presence of constraints and fixed model uncertainty (i.e. robust MPC) has received considerable research attention (see e.g. [5] and [19]), the topic of adaptive MPC for constrained systems has received little attention due to difficulties in guarantee-

⋆ This research has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. PIOF-GA-2009-252284 - Marie Curie project "Innovative Control, Identification and Estimation Methodologies for Sustainable Energy Technologies". Part of this paper was presented at 11^{th} IFAC International Workshop on Adaptation and Learning in Control and Signal Processing, July 3 – 5, Caen, France. Corresponding author M. Tanaskovic. Tel. +41 44 632 0403. Fax +41 44 632 1211.

Email addresses: tmarko@control.ee.ethz.ch (Marko Tanaskovic), lorenzo.fagiano@ch.abb.com (Lorenzo Fagiano), rsmith@control.ee.ethz.ch (Roy Smith), morari@control.ee.ethz.ch (Manfred Morari).

ing constraint satisfaction and recursive feasibility under adaptation [15]. Adaptive MPC for input-constrained MIMO systems was considered in [21]. In [17] an adaptive MPC algorithm based on model switching was proposed. However, these adaptive control approaches can not deal with hard output constraints. In [16], an adaptive MPC algorithm for a class of single input multiple output linear systems, based on modified recursive least squares identification and tube-like robust MPC, was proposed. The algorithm is capable of handling both input and output constraints and it guarantees stability and recursive feasibility, but the state space structure of the plant needs to be known and noise free measurements of the plant states are required. In [3], a learning based MPC algorithm was introduced. It uses statistical learning techniques in order to improve the control performance over time and robustly enforces input and output constraints by using an approximate model of the plant together with a bound on its uncertainty. The approximate model and the uncertainty bounds are not updated over time and therefore, a relatively good initial knowledge of the plant model is required. Nonlinear adaptive MPC for a specific class of systems was considered in [1]. The idea of using parameter bounding and predictive control in an adaptive context was presented in [30], where a set of plant models is identified in order to formulate a min-max finite horizon predictive control problem at each time step. However, only input constrained SISO systems are considered and the proposed control algorithm is very computationally intensive and can be applied to system models of low dimension only. More recently, set membership (SM) identification was used for adaptive MPC of SISO systems subject to both input and output constraints in [25], where an explicit MPC law is repeatedly re-calculated in an off-line manner, when new information on the controlled plant becomes available.

We propose here a new adaptive model predictive control approach for open-loop stable, linear MIMO systems. In contrast to the previously mentioned contributions, our approach is capable of dealing with stable MIMO systems in the presence of both input and hard output constraints, as well as output disturbances and measurement noise. Moreover, the proposed algorithm is computationally tractable and suitable for on-line application, as it only requires the solution of standard convex optimization problems. The required initial information on the system is given by some (eventually very loose) bounds on its impulse response coefficients and by bounds on the magnitudes of the output disturbance and measurement noise. Unlike the methods in [16] and [3], the proposed approach can not be used to control open-loop unstable systems but, on the other hand, it can handle systems with multiple inputs and measurement noise and requires a smaller amount of initial knowledge on the plant. It has to be noted that the requirement for open-loop stability is quite common in the context of system identification and adaptive control. The main idea underlying the control algorithm is to recursively identify the set of all the plant models (”model set”), that are consistent with the initial available information on the plant and with the input-output data collected during operation. Then, an MPC approach is used to guarantee satisfaction of input and output constraints for all the plants inside the model set and hence also for the real system. In addition to robust constraint satisfaction, the approach guarantees recursive feasibility and, under a convergence assumption, it exhibits integral action which enables offset-free tracking. Although not rigorously proved here, the mentioned convergence property is met in practice with long enough prediction horizons, as we show through an experimental testbed. In addition, we comment on possible extensions of the formulation to use a model parametrization with basis functions in order to reduce the computational complexity of the algorithm, and to include an exploring strategy, which aims to select the input that yields a higher amount of information in order to speed-up the identification of the plant dynamics. In addition to the adaptive approach and its theoretical properties, we present the experimental results obtained by applying the method to a non-minimum phase quad-tank testbed.

2 Problem Statement

We consider a MIMO, discrete time, strictly proper, linear time invariant (LTI) system with n_u inputs and n_y outputs. The system is known to be stable, but the exact system's dynamics are not known. We denote the vector of control inputs at time step $t \in \mathbb{Z}$ by $u(t) = [u_1(t), \dots, u_{n_u}(t)]^T$, where $u_i(t) \in \mathbb{R}$, $i = 1, \dots, n_u$ are the individual plant inputs at the time step t and ^{T} stands for the matrix transpose operator. In addition, we denote the vector of plant outputs by $y(t) = [y_1(t), \dots, y_{n_y}(t)]^T$, where $y_j(t) \in \mathbb{R}$, $j = 1, \dots, n_y$ are the individual plant outputs, and we denote the vector of output disturbances by $d(t) = [d_1(t), \dots, d_{n_y}(t)]^T$, where $d_j(t) \in \mathbb{R}$, $j = 1, \dots, n_y$ denotes the contribution of the disturbances to the output j at time step t. The dynamic relation between each input and output can, in general, be described by an infinite impulse response. In order to have a tractable model, we use finite impulse responses (FIR) of length m, i.e. the influence of the input i to the output j can be described by the FIR coefficients $h_{ji}(k)$, $k = 1, \dots, m$. The approximation error induced by the truncated impulse response coefficients is embedded into the output disturbance vector $d(t)$ (see also Remark 2.2).

Remark 2.1 *Note that the same FIR length is assumed here for all the input-output pairs in order to simplify the notation. All the results can easily be extended to the case when different FIR lengths are used.*

In the described setting, each of the plant outputs $y_j(t)$, $j = 1, \dots, n_y$, is given by:

$$\begin{aligned} y_j(t) &= \sum_{i=1}^{n_u} \sum_{k=1}^{m} u_i(t-k) h_{ji}(k) + d_j(t) \\ &= H_j^T \varphi(t) + d_j(t),\ j = 1, \dots, n_y, \end{aligned} \tag{1}$$

where $\varphi(t) \in \mathbb{R}^{n_u m}$ is the regressor vector: $\varphi(t) \doteq [u_1(t-1), \dots, u_1(t-m), \dots, u_{n_u}(t-1), \dots, u_{n_u}(t-m)]^T$, and each of the vectors $H_j \in \mathbb{R}^{n_u m}$, $j = 1, \dots, n_y$ contains the impulse response coefficients needed to describe the influence of all the control inputs on the plant output y_j: $H_j \doteq [h_{j1}(1), \dots, h_{j1}(m), \dots, h_{jn_u}(1), \dots, h_{jn_u}(m)]^T$. By defining the matrix $H \in \mathbb{R}^{n_y \times n_u m}$ as $H \doteq \left[H_1, \dots, H_{n_y}\right]^T$, the dependence of the plant output on the regressor and the disturbance vectors can be written as:

$$y(t) = H\varphi(t) + d(t). \tag{2}$$

The measured output available for feedback control is corrupted by noise. In particular, the vector of measured plant outputs is denoted by:

$$\tilde{y}(t) = y(t) + v(t),$$

where $v(t) = [v_1(t), \dots, v_{n_y}(t)]^T$ and $v_j(t)$, $j = 1, \dots, n_y$ are the individual measurement noise terms that affect each of the measured plant outputs.

Assumption 1 *(Prior assumption on disturbance and noise) d and v are bounded as:*

$$\begin{aligned} |d_j(t)| &\le \epsilon_{d_j} \\ |v_j(t)| &\le \epsilon_{v_j} \end{aligned},\ \forall t \in \mathbb{Z},\ \forall j = 1, \dots, n_y, \tag{3}$$

where ϵ_{d_j} and ϵ_{v_j} are positive scalars.

We also use the noise and disturbance magnitude bounds in the vector notation as $\epsilon_d = [\epsilon_{d_1}, \dots, \epsilon_{d_{n_y}}]^T$ and $\epsilon_v = [\epsilon_{v_1}, \dots, \epsilon_{v_{n_y}}]^T$.

Assumption 2 *(Prior assumption on the system) The plant belongs to the following model set: $H \in \mathcal{F}(0)$, with*

$$\mathcal{F}(0) \doteq \left\{H \in \mathbb{R}^{n_y \times n_u m} : A_j(0) H_j \le b_j(0), j = 1, \dots, n_y\right\}, \tag{4}$$

where the inequalities in (4) should be interpreted as element-wise inequalities and each matrix $A_j(0) \in \mathbb{R}^{r_j(0) \times n_u m}$ and vector $b_j(0) \in \mathbb{R}^{r_j(0)}$ define a closed and convex set, i.e. a polytope with $r_j(0)$ faces.

According to Assumption 2, the initial knowledge about the impulse response coefficients is that the vectors H_j, $j = 1, \dots, n_y$, which form the rows of the matrix H, belong to polytopic sets. Note that we initialized the set $\mathcal{F}(0)$ in (4) at $t = 0$ without loss of generality, just to indicate that this is the information available before any measured data is available. Hence, these initial polytopes have to be defined a priori on the basis of the prior information on the system. If the only available information is the fact that the system is stable, a possible approach to define the set $\mathcal{F}(0)$ is to impose a maximum magnitude and an exponential decay rate on the FIR coefficients. This can be done by selecting the following three variables for each input i and output j: $L_{ji} \in \mathbb{R}$, $L_{ji} \ge 0$, $\rho_{ji} \in \mathbb{R}$, $\rho_{ji} \in (0, 1)$ and $\mu_{ji} \in \mathbb{N}$, $\mu_{ji} \le m$, and then defining the upper and the lower bounds on the impulse response coefficients for the input-output pair defined by i and j as:

$$\begin{aligned} |h_{ji}(k)| &\le L_{ji} && \text{if } k \in [1, \mu_{ji}] \\ |h_{ji}(k)| &\le L_{ji}\rho_{ji}^{k-\mu_{ji}} && \text{if } k \in [\mu_{ji}+1, m]. \end{aligned} \tag{5}$$

If additional initial knowledge about the system is available, e.g. the sign of the steady state gain for some of the input-output pairs, this can be incorporated as additional inequalities defining the set $\mathcal{F}(0)$.

Under Assumptions 1 and 2, the goal is to control the plant in order to track a desired output reference and reject disturbances from $t = 0$ up to some finite time step T, where the time horizon T can be very large ($T \gg m$). Moreover, the controller shall enforce input and output constraints. Assuming that the control inputs $u(l)$, $l = -m+1, \dots, -1$ are known, such a control objective can be formalized by the following optimization problem:

$$\begin{aligned} \min_{u(0),\dots,u(T)} \sum_{t=0}^{T} &\left(y(t) - y_{\text{des}}(t)\right)^T Q \left(y(t) - y_{\text{des}}(t)\right) \\ &+ u(t)^T S u(t) + \Delta u(t)^T R \Delta u(t) \end{aligned} \tag{6}$$

$$\begin{aligned} &\text{Subject to, } \forall t \in [0, T] \\ &Cu(t) \le g \\ &L\Delta u(t) \le f \\ &Ey(t) \le p \end{aligned} \tag{7}$$

where $y_{\text{des}}(t) \in \mathbb{R}^{n_y}$ is the desired output reference, $Q \in \mathbb{R}^{n_y \times n_y}$, $S \in \mathbb{R}^{n_u \times n_u}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are positive semi-definite weighting matrices selected by the control designer, and $\Delta u(t) = u(t) - u(t-1)$ is the rate of change of the control input. The element-wise inequalities in (7) define convex sets through the matrices $C \in \mathbb{R}^{n_i \times n_u}$, $L \in \mathbb{R}^{n_{\Delta_u} \times n_u}$, $E \in \mathbb{R}^{n_o \times n_y}$ and the vectors $g \in \mathbb{R}^{n_i}$, $f \in \mathbb{R}^{n_{\Delta_u}}$, $p \in \mathbb{R}^{n_o}$, where n_i, n_{Δ_u} and n_o are the number of linear constraints on the inputs, input rates and outputs, respectively. We assume that the set defining the constraints on $\Delta u(t)$ contains the origin and that the constraint set of $u(t)$ is compact. This assumption is satisfied in most practical problems.

Remark 2.2 *The facts that the control inputs are bounded and that the system is stable can be exploited to calculate bounds on the magnitude of the contribution of the truncated part of the impulse response for each of the plant outputs. Such a contribution can be considered as part of the additive output disturbance. For completeness, the way to calculate these bounds is given in the Appendix A.*

3 Adaptive control algorithm

Since the true plant is not exactly known and its outputs are subject to unknown output disturbances, the optimal control problem (6) can not be solved a priori and a suboptimal approach has to be sought. Therefore, in order to approximately optimize the given control objective, while guaranteeing satisfaction of the constraints (7), we propose the use of a receding horizon approach, combined with an adaptive control scheme that aims to improve the knowledge on the system's dynamics over time. In this setting, at each time step a sequence of future control inputs is calculated and only the first element of this sequence is applied to the plant. In particular, to guarantee output constraint satisfaction, we aim to identify, at each time step, the set of all the models that are consistent with the initial assumptions on the real plant and the input-output measurements collected up to that time step (model set). If the prior assumptions are valid, this set is guaranteed to contain also the true plant's dynamics. Then, the control computation is carried out in such a way to ensure that the constraints are satisfied for all the models inside this set, hence also for the actual plant.

In order to accomplish the model set identification and the robust control computation, we rely on a recursive SM identification algorithm, and an MPC controller. The identification algorithm is such that the model set can be refined with each new output measurement. In addition to the model set, the identification algorithm also provides a nominal model of the plant at each time step. The control input is calculated by solving an optimal control problem that minimizes a weighted quadratic cost penalizing the tracking error of the nominal model over a finite horizon, while at the same time satisfying robustly the constraints (7). Algorithm 1 sumarizes the proposed adaptive control scheme.

In the subsections that follow, each of the components of the proposed adaptive control algorithm is described in detail.

3.1 Real-time set membership identification

We denote the sequence of the input-output data collected up to time step t as:

$$\{\varphi(l),\ \tilde{y}(l)\}_{l=0}^{t}, \tag{8}$$

Algorithm 1 Adaptive MPC algorithm

```
1) At time step $t$, obtain $\tilde{y}(t)$ and update the model set
   based on the past applied control inputs and measured plant outputs;
2) Select a nominal model of the plant inside the model set;
3) Calculate a sequence of possible future control inputs by solving
   a finite horizon optimal control problem (FHOCP) that minimizes a
   weighted quadratic cost involving the tracking error of the nominal
   model and enforces input and output constraints for all the models
   inside the model set;
4) Apply the first element of the calculated input sequence,
   set $t = t + 1$, go to 1).
```

where $\varphi(l) \in \mathbb{R}^{n_u m}$ is the regressor vector formed by the control inputs applied from time $l - m$ up to time $l - 1$, and $\tilde{y}(l) \in \mathbb{R}^{n_y}$ is the corresponding measured plant output. Then, at a given time step t, we define the model set $\mathcal{F}(t)$ as the set containing all the matrices H that are consistent with the Assumptions 1 and 2 and the collected input-output data (8):

$$\mathcal{F}(t) \doteq \left\{ \begin{array}{c} H \in \mathcal{F}(0) : -\epsilon_d - \epsilon_v \leq \tilde{y}(l) - H\varphi(l) \leq \epsilon_d + \epsilon_v, \\ \forall l \in [0, t] \end{array} \right\}. \tag{9}$$

Each one of the element-wise inequalities in (9) comes from the fact that the discrepancy between the measured and the predicted values of the output can not exceed the disturbance and noise bounds (3). Since the initial model set $\mathcal{F}(0)$ is defined by polytopic constraints on each row H_j^T of the matrix H, and the constraints in (9) are linear, the model set $\mathcal{F}(t)$ is still defined by polytopic constraints on H_j^T, $j = 1, \dots n_y$. Each of these polytopes can be uniquely described by a set of non-redundant inequalities. Therefore, at a generic time step t, the model set $\mathcal{F}(t)$ can be represented as:

$$\mathcal{F}(t) = \left\{ H \in \mathbb{R}^{n_y \times n_u m} : A_j(t) H_j \leq b_j(t),\ j = 1, \dots, n_y \right\},$$

where $A_j(t) \in \mathbb{R}^{r_j(t) \times n_u m}$, $b_j(t) \in \mathbb{R}^{r_j(t)}$, and $r_j(t)$ is the number of non-redundant inequalities pertaining to the j^{th} row of the matrix H.

The matrices $A_j(t)$ and the vectors $b_j(t)$ have to be updated at each time step in order to account for the new measurements. To this end, let us consider the following polytopes:

$$\mathcal{F}_j(t) = \{H_j \in \mathbb{R}^{n_u m} : A_j(t) H_j \leq b_j(t)\},\ j = 1 \dots n_y.$$

We note that for each j, the polytope $\mathcal{F}_j(t)$ can be calculated recursively in time as the intersection of the polytope $\mathcal{F}_j(t-1)$ and the two half spaces defined by the newly measured plant output, $\tilde{y}_j(t)$:

$$\begin{aligned}\mathcal{F}_j(t) =& \mathcal{F}_j(t-1) \\ & \cap \{H_j \in \mathbb{R}^{n_u m} : \varphi(t)^T H_j \leq \tilde{y}_j(t) + \epsilon_{d_j} + \epsilon_{v_j}\} \\ & \cap \{H_j \in \mathbb{R}^{n_u m} : -\varphi(t)^T H_j \leq -\tilde{y}_j(t) + \epsilon_{d_j} + \epsilon_{v_j}\}.\end{aligned} \tag{10}$$

The matrix $A_j(t)$ and the vector $b_j(t)$ can then be calculated by removing any redundant faces of the polytope $F_j(t)$. This can be done by solving an LP for each face of the polytope, in order to determine whether it is redundant or not [24]. A problem of the described recursive update is that the number of faces of $\mathcal{F}_j(t)$, $r_j(t)$, can become arbitrarily large, as in general it grows linearly with time, and hence the memory needed to store $A_j(t)$ and $b_j(t)$ can become impractical. In order to overcome this problem, we employ a polytope update algorithm with bounded complexity, similar to the one proposed in [31]. In this approach, the polytope $\mathcal{F}_j(t)$ is updated by using (10) as long as the number of its faces is smaller than a predefined maximum limit M_1. Once this limit is reached, each new face that is added to the polytope is parallel to a plane that belongs to a predefined set of M_2 planes, which makes the total number of faces bounded by $M_1 + M_2$.

In particular, a set D containing a finite number M_2 of $n_u m$-dimensional vectors with the same magnitude, that will determine the shape of the resulting polytope, has to be defined. Based on this set, the update of the polytope $\mathcal{F}_j(t)$ is given by the following intersection:

$$\begin{aligned}\mathcal{F}_j(t) =& \mathcal{F}_j(t-1) \\ & \cap \{H_j \in \mathbb{R}^{n_u m} : \varphi^+(t)^T H_j \leq \tilde{y}(t) + \delta_j^+(t)\} \\ & \cap \{H_j \in \mathbb{R}^{n_u m} : \varphi^-(t)^T H_j \leq -\tilde{y}(t) + \delta_j^-(t)\},\end{aligned} \tag{11}$$

where the vectors $\varphi^+(t)$ and $\varphi^-(t)$ are taken as elements of D that are "closest", in the inner product sense, to the vectors $\varphi(t)$ and $-\varphi(t)$:

$$\begin{aligned}\varphi^+(t) &= \arg\max_{v \in D} \varphi(t)^T v \\ \varphi^-(t) &= \arg\max_{v \in D} -\varphi(t)^T v,\end{aligned} \tag{12}$$

and the scalars $\delta_j^+(t)$ and $\delta_j^-(t)$ are selected such that the bounded complexity polytope includes the polytope that would be obtained by a normal update (as per (10)). Hence, the values of $\delta_j^+(t)$ and $\delta_j^-(t)$ can be calculated by solving the following linear program (LP):

$$\begin{aligned}&\delta_j^+(t) = \max_\theta \varphi^+(t)^T \theta - \tilde{y}_j(t) \\ &\delta_j^-(t) = \max_\theta \varphi^-(t)^T \theta + \tilde{y}_j(t) \\ &\text{Subject to:} \\ &A_j(t-1)\theta \leq b_j(t-1) \\ &\quad \varphi(t)^T \theta \leq \tilde{y}_j(t) + \epsilon_{d_j} + \epsilon_{v_j} \\ &\quad -\varphi(t)^T \theta \leq -\tilde{y}_j(t) + \epsilon_{d_j} + \epsilon_{v_j}.\end{aligned} \tag{13}$$

The set D is a fixed set of vectors that have to be chosen beforehand; one possible way to construct D is to take regularly distributed vectors on the unit circle (see e. g. [23]). Algorithm 2 summarizes the above procedure for recursive updating of the model set $\mathcal{F}(t)$ in our adaptive control scheme (see step 1) of Algorithm 1).

Algorithm 2 Bounded complexity model set update

```
1) At time step $t$, compute the regressors vector $\varphi(t)$ and measure the plant output $\tilde{y}(t)$;
2) For $j = 1, \dots, n_y$, if $r_j(t-1) \leq M_1 - 2$, update $\mathcal{F}_j(t)$ by using (10), otherwise calculate $\varphi^+(t)$ and $\varphi^-(t)$ as in (12), find the values of $\delta^+(t)$ and $\delta^-(t)$ by solving the LP (13) and update $\mathcal{F}_j(t)$ according to (11);
3) For $j = 1, \dots, n_y$, calculate $A_j(t)$ and $b_j(t)$ by removing any redundant faces from $\mathcal{F}_j(t)$.
```

The set $\mathcal{F}(t)$ obtained by using the Algorithm 2 is an outer approximation of the set defined in (9): by increasing M_1 and M_2, the tightness of such approximation can be increased, at the cost of higher complexity. The algorithm guarantees that $\mathcal{F}_j(t) \subseteq \mathcal{F}_j(t-1)$, $j = 1, \dots, n_y$ and hence $\mathcal{F}(t) \subseteq \mathcal{F}(t-1)$, a property that is needed to obtain recursive feasibility and output constraint satisfaction, as we show in section 4.

In SM identification problems, an important issue that has to be dealt with is whether the considered prior assumptions are invalidated by the data or not.

Lemma 3.1 *Let the Assumptions 1 and 2 hold, then the set $\mathcal{F}(t)$ obtained by using the Algorithm 2 has the following properties: $\mathcal{F}(t) \neq \emptyset$ and $H \in \mathcal{F}(t)$, $\forall t$, i.e. the model set is never empty and is guaranteed to contain the true plant's coefficients.*

Proof 3.1 *We prove the Lemma by induction. From the Assumption 2 it holds that $H \in \mathcal{F}(0)$. Assume now that at a generic time step t, it holds that $\mathcal{F}(t) \neq \emptyset$ and $H \in \mathcal{F}(t)$. This implies that $H_j \in \mathcal{F}_j(t)$, $j = 1, \dots, n_y$. Moreover, from Assumption 1 we have that:*

$$\begin{array}{r} \varphi(t+1)^T H_j \leq \tilde{y}_j(t+1) + \epsilon_{d_j} + \epsilon_{v_j} \\ -\varphi(t+1)^T H_j \leq -\tilde{y}_j(t+1) + \epsilon_{d_j} + \epsilon_{v_j} \end{array}, \forall j = 1, \dots, n_y.$$

Therefore, if the polytope $\mathcal{F}_j(t)$ is updated according to (10), we have that $H_j \in \mathcal{F}_j(t+1)$. The same holds true also if the bounded complexity polytopic update is used, since the solution of the LP (13) guarantees that the faces that are added to the polytope are such that the bounded complexity polytope contains the one that would be obtained by (10). Therefore, the set $\mathcal{F}(t+1)$ obtained by using the Algorithm 2 is guaranteed to satisfy $H \in \mathcal{F}(t+1)$ and hence $\mathcal{F}(t+1) \neq 0$. The lemma is then proved by applying this result recursively from $t = 0$ to any $t \geq 0$.■

The converse of Lemma 3.1 provides a necessary condition for Assumptions 1 and 2 to hold: if $\mathcal{F}(t) \neq \emptyset$, then the prior information on $\mathcal{F}(0)$, ϵ_d and ϵ_v is "not invalidated" by the collected measurements (8). In practice, the initial set and the disturbance and noise bounds have to be chosen by the designer in order to have non-empty model set $\mathcal{F}(t)$, while at the same time avoiding excessive conservativeness.

In addition to the model set, the proposed SM identification algorithm also provides a nominal model of the plant (step 2) of Algorithm 1). The latter is given by a matrix $H_c(t) \in \mathbb{R}^{n_y \times n_u m}$, $H_c = [H_{c,1}, \dots H_{c,n_y}]^T$, where $H_{c,j}(t) \in \mathbb{R}^{n_u m}$, $j = 1, \dots, n_y$ are computed as the centers of the maximum volume 2-norm balls inscribed in the polytopes $\mathcal{F}_j(t)$. This can be done by solving an LP, however the solution is not unique in general. Therefore, we introduce a regularization term, that penalizes the deviation of the new nominal model from the previous one, giving rise to the following LP:

$$\max_{\xi_j(t), H_{c,j}(t)} \sum_{j=1}^{n_y} \xi_j(t) - \alpha \|H_{c,j}(t-1) - H_{c,j}(t)\|_1$$

Subject to:

$$a_{ji}(t) H_{c,j}(t) + \xi_j(t) \|a_{ji}(t)\|_2 \leq b_{ji}(t), \quad \begin{array}{l} \forall j = 1, \dots, n_y \\ \forall i = 1, \dots, n_u \end{array}, \tag{14}$$

where $\xi_j(t) \in \mathbb{R}$ is the radius of the maximum volume ball inscribed in $\mathcal{F}_j(t)$, $\alpha > 0$ is a design variable, and $a_{ji}(t)$ and $b_{ji}(t)$ stand for the i^{th} row of the matrix $A_j(t)$ and the vector $b_j(t)$. Initially, at time step $t = 0$, the matrix $H_c(0)$ can be taken as an arbitrary nonzero point inside the set $\mathcal{F}(0)$.

3.2 Constrained predictive controller

Let $u(k|t)$, $k \in [t, t+N-1]$, $N \geq m$, be the candidate future control moves, where the notation $k|t$ indicates the prediction at step $k \geq t$ given the information at the current step t. For brevity, we collect these decision variables in vector $U \doteq [u(t|t)^T \dots u(t+N-1|t)^T]^T$. We also define the vectors of future input increments $\Delta u(k|t)$, $k \in [t, t+N-1]$ as:

$$\Delta u(k|t) = \begin{cases} u(t|t) - u(t-1) & \text{if } k = t \\ u(k|t) - u(k-1|t) & \text{if } t+1 \leq k \leq t+N-1 \end{cases}$$

Moreover, we define the future regressor vectors $\varphi(k|t) \in \mathbb{R}^{n_u m}$, $k \in [t+1, t+N]$ as:

$$\varphi(k|t) = \begin{cases} W\varphi(t) + Zu(t|t) & \text{if } k = t+1 \\ W\varphi(k-1|t) + Zu(k-1|t) & \text{if } t+2 \leq k \leq t+N, \end{cases} \tag{15}$$

where $W \in \mathbb{R}^{n_u m \times n_u m}$ and $Z \in \mathbb{R}^{n_u m \times n_u}$ are suitable matrices, given in the Appendix B for the sake of completeness. In addition, we define the current prediction error $\hat{d}(t) \in \mathbb{R}^{n_y}$ as the difference between the measured plant output and the one predicted by the nominal model at time step t:

$$\hat{d}(t) \doteq \tilde{y}(t) - H_c(t)\varphi(t). \tag{16}$$

Then, we consider the following cost function:

$$\begin{aligned} & J(U, \tilde{y}(t), \varphi(t)) \doteq \\ & \sum_{k=t}^{t+N-1} (\hat{y}(k+1|t) - y_{\text{des}}(k+1|t))^T Q (\hat{y}(k+1|t) \\ & - y_{\text{des}}(k+1|t)) + u(k|t)^T S u(k|t) + \Delta u(k|t)^T R \Delta u(k|t), \end{aligned} \tag{17}$$

where:

$$\hat{y}(k+1|t) = H_c(t)\varphi(k+1|t) + \hat{d}(t). \tag{18}$$

In (17), $\tilde{y}(t)$ and $\varphi(t)$ are known parameters and $y_{\text{des}}(k|t)$, $k \in [t+1, t+N]$, are the predicted values of the desired output. The introduction of the disturbance estimate $\hat{d}(t)$ in the cost function enables offset free tracking under certain conditions (see section 4).

Satisfaction of input constraints can be enforced by the following set of inequalities:

$$\begin{array}{rl} Cu(k|t) \leq & g \\ L\Delta u(k|t) \leq & f \end{array} \quad \forall k \in [t, t+N-1]. \tag{19}$$

And the robust satisfaction of the output constraints can be achieved by enforcing them for all the plants inside the model set $\mathcal{F}(t)$ and for all disturbance realizations:

$$EH\varphi(k|t) + \overline{d} \leq p, \; \forall H \in \mathcal{F}(t), \; \forall k \in [t+1, t+N], \tag{20}$$

where $\overline{d} = [\overline{d}_1, \dots, \overline{d}_{n_o}]^T$, and $\overline{d}_l \in \mathbb{R}$, $l = 1, \dots, n_o$ are given as:

$$\overline{d}_l = \sum_{j=1}^{n_y} |e_{lj}| \epsilon_{d_j},$$

where e_{lj} stands for the element of the l^{th} row and j^{th} column of the matrix E. However, using the constraints (20) would result in an infinite dimensional optimization problem, that is in general hard to solve. The following result shows how (20) can be equivalently written in the form of linear constraints. Before stating the result, let us introduce the vector of auxiliary decision variables $\Lambda \doteq \left[\Lambda_1^T, \dots, \Lambda_{n_o}^T\right]^T \in \mathbb{R}^{n_o N r(t)}$, where $\Lambda_l \doteq \left[\lambda_l(t+1|t)^T, \dots, \lambda_l(t+N|t)^T\right]^T$, $l = 1, \dots, n_o$, and for each $k = t+1, \dots, t+N$, $\lambda_l(k|t) \in \mathbb{R}^{r(t)}$ and $r(t) = \sum_{j=1}^{n_y} r_j(t)$.

Lemma 3.2 *The constraints (20) are satisfied if and only if there exist $\varphi(k|t)$, $k \in [t+1, t+N]$ and Λ such that the following set of inequalities is feasible:*

$$\left.\begin{aligned} &A(t)^T \lambda_l(k|t) = \begin{bmatrix} e_{l1}\varphi(k|t) \\ \vdots \\ e_{ln_y}\varphi(k|t) \end{bmatrix} \\ &b(t)^T \lambda_l(k|t) \le p_l - \overline{d}_l \\ &\lambda_l(k|t) \ge \mathbf{0} \end{aligned}\right\} \begin{array}{l} \forall l = 1, \dots, n_o \\ \forall k \in [t+1, t+N] \end{array} \tag{21}$$

with

$$A(t) = \begin{bmatrix} A_1(t) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & A_2(t) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & A_{n_y}(t) \end{bmatrix}$$
$$b(t) = \begin{bmatrix} b_1(t) \\ \vdots \\ b_{n_y}(t) \end{bmatrix},$$

where $\mathbf{0}$ represents zero matrices of appropriate dimensions and p_l is the l^{th} element of the vector p.

Proof 3.2 *We first note that, from the definition of the set $\mathcal{F}(t)$, it follows that constraints* (20) *are satisfied if and only if the following set of inequalities is satisfied:*

$$\gamma_l(k) \le p_l - \overline{d}_l, \quad \begin{array}{l} \forall l = 1, \dots, n_o \\ \forall k \in [t+1, t+N] \end{array} \tag{22}$$

where

$$\gamma_l(k) = \max_{A_j(t)H_j \le b_j(t)} \sum_{j=1}^{n_y} e_{lj}\varphi(k|t)^T H_j. \tag{23}$$

For each value of l and k and for fixed values of the vectors $\varphi(k|t)$, $k \in [t+1, t+N]$, by using the fact that the inequalities $A_j(t)H_j \le b_j(t)$, $j = 1, \dots, n_y$ form nonempty (from Lemma 3.1), closed and bounded convex sets (i.e. polytopes), we can write the dual of the LP (23) *as:*

$$\tilde{\gamma}_l(k) = \min_{\lambda_l(k|t)} b(t)^T \lambda_l(k|t) \tag{24}$$

Subject to:

$$A(t)^T \lambda_l(k|t) = \begin{bmatrix} e_{l1}\varphi(k|t) \\ \vdots \\ e_{ln_y}\varphi(k|t) \end{bmatrix} \tag{25}$$

$$\lambda_l(k|t) \ge \mathbf{0}. \tag{26}$$

According to the strong duality theorem for LPs [6], it holds that: $\gamma_l(k) = \tilde{\gamma}_l(k)$. Therefore, for any $\lambda_l(k|t)$ that satisfies the constraints (25) and (26)*, it holds that $\gamma_l(k) \le b(t)^T \lambda_l(k|t)$. Hence the existence of U and Λ that satisfy the set of constraints (21) guarantees that the constraints (22) are also satisfied, which implies the satisfaction of the original constraints* (20)*. On the other hand if the constraints* (20) *are satisfied, then there exists $\gamma_l(k)$ satisfying* (22)*. Then, by the strong duality theorem for LP, $\tilde{\gamma}_l(k) = \gamma_l(k)$ exists and hence the constraints* (25) *and* (26) *have to be feasible, which implies the feasibility of* (21). ■

In order to be able to recursively satisfy the input and output constraints (see e.g. Theorem 4.1 below), we introduce an additional constraint on the terminal stage:

$$\varphi(t+N|t) = W\varphi(t+N|t) + Zu(t+N-1|t). \tag{27}$$

This means that we require the terminal regressor to correspond to a steady state (i.e. a constant control input is kept for the last m predicted steps).

For fixed values of N, Q, S and R, we can now define the finite horizon optimal control problem (FHOCP) to be solved at each time step t (see step 3) of Algorithm 1):

$$\begin{aligned} &\min_{U,\Lambda} J(U, \tilde{y}(t), \varphi(t)) \\ &\quad \text{Subject to: (19), (21), (27),} \end{aligned} \tag{28}$$

which is a quadratic program (QP), that can be efficiently solved in general. The number of decision variables and constrains of the QP (28) depends on the chosen prediction horizon N and the number of faces describing the polytopes of the model set $\mathcal{F}(t)$. Therefore, the computational complexity of (28) can be decreased by reducing the bound $M_1 + M_2$ on the faces' number, at the cost of higher conservativeness as discussed in section 3.1.

4 Properties of the proposed adaptive control algorithm

The described control algorithm guarantees recursive feasibility and robust satisfaction of both input and output constraints, as shown by the following result.

Theorem 4.1 *Let the Assumptions 1–2 hold, and assume that the problem* (28)*, solved under the Algorithm 1, is feasible at time* $t = 0$*. Then the problem* (28) *is recursively feasible and the closed-loop system obtained by applying the Algorithm 1 is guaranteed to satisfy input and output constraints* $\forall t \geq 0$*.*

Proof 4.1 *We first prove that if the Assumptions 1–2 hold and the problem* (28) *is feasible at time* $t = 0$*, then the problem* (28) *solved under the Algorithm 1 remains feasible* $\forall t \geq 0$*. To this end we use induction.*

The problem (28) *is feasible for* $t = 0$ *by assumption. Assume that the problem* (28) *is feasible at a generic time step* t *and let the optimal control sequence be* $U^*(t) = [u^*(t|t)^T, \dots, u^*(t+N-1|t)^T]^T$*, and its corresponding sequence of predicted regressor vectors be* $\varphi^*(k|t)$*,* $k = t+1, \dots, t+N$*. Then, a possible feasible control sequence at* $t+1$ *is* $U(t+1) = [u^*(t+1|t)^T, \dots, u^*(t+N-1|t)^T, u^*(t+N-1|t)^T]^T$*. This sequence satisfies constraints* (19) *and* (27)*. In addition, we note that the predicted regressor vectors* $\varphi(k|t+1)$*,* $k = t+2, \dots, t+N+1$ *that correspond to the input sequence* $U(t+1)$*, by construction satisfy the equalities* $\varphi(k|t+1) = \varphi^*(k|t)$*, for* $k \in [t+2, t+N]$ *and that from* (27) *it follows that* $\varphi(t+N+1|t+1) = \varphi^*(t+N|t)$*. Since Algorithm 2 guarantees that* $\mathcal{F}(t+1) \subseteq \mathcal{F}(t)$*, the sequence of inputs* $U(t+1)$ *satisfies the constraints* (20)*. Lemma 3.2 then implies that also the constraints* (21) *have a feasible solution and hence the problem* (28) *is feasible at* $t+1$*. From this result and Lemma 3.1, the other claim of the Theorem follows directly.* ■

In practice, feasibility at time $t = 0$ means that the initial assumptions are such that there exists a nonzero (eventually very small) input sequence that does not violate the input and output constraints for all the plants in the initial model set $\mathcal{F}(0)$, which is a reasonable condition. Note that the key property that is needed for guaranteeing the recursive feasibility is that $\mathcal{F}(t) \subseteq \mathcal{F}(t-1)$. However, in practice, due to faults and outliers it may happen that noise and/or output disturbances temporarily violate the bounds (3), which may cause the actual plant model to be outside the model set, or the model set to be empty (see e.g. Lemma 3.1). In such cases, normal operation could be recovered by removing the outliers with techniques such as the ones described in [26] and [28]. Removing the outliers might cause the model set to expand, which may lead to infeasibility of (28). In such cases, feasibility could be recovered by temporarily softening the output constraints.

The next result is concerned with offset-free tracking in the presence of constant disturbance and zero measurement noise. Before stating the result, the following technical assumption is needed:

Assumption 3 *The steady state gain matrix of the nominal model obtained by solving* (14)*, that we denote by:*

$$H_{c,ss} = \begin{bmatrix} \sum\limits_{l=1}^{m} h_{c,11}(l) & \dots & \sum\limits_{l=1}^{m} h_{c,1n_u}(l) \\ \vdots & \ddots & \vdots \\ \sum\limits_{l=1}^{m} h_{c,n_y1}(l) & \dots & \sum\limits_{l=1}^{m} h_{c,n_yn_u}(l) \end{bmatrix},$$

satisfies the condition: rank$(H_{c,ss}) = n_y$*, at each time step.*

Remark 4.1 *Note that if the steady state gain matrix of the actual system has rank* n_y*, then (from Lemma 3.1) there is always a model inside the model set* $\mathcal{F}(t)$ *with the same property. Although the algorithm for selecting the nominal model does not guarantee that the latter has always this property, this does not represent a problem for the control computation itself. However, the technical Assumption 3 is needed to prove the offset-free property. In order to formally satisfy this requirement, the algorithm could be modified such that the rank of the steady state gain matrix of the nominal model is checked, and if it were different from* n_y*, the nominal model could be slightly perturbed in order to satisfy the rank condition of the steady state gain matrix.*

Lemma 4.1 *Let Assumption 3 hold and suppose that the tuning matrices in the cost function* (17) *are selected such that* $S = 0$ *and* Q *and* R *are positive definite. In addition, assume that the vector of the output references is constant:* $y_{des}(t) = y_{des}, \forall t$*, that the output disturbances are constant :* $d(t) = d$*,* $\forall t$*, and that there is no measurement noise:* $v(t) = 0, \forall t$*. Then, if the input and the output of the closed loop system converge to constant vectors,* u_{ss} *and* y_{ss} *respectively, for which no constraints are active, it holds that* $y_{ss} = y_{des}$ *.*

Proof 4.2 *First, we show that if the control input and the output of the closed loop system converge to constant vectors, the output disturbances are constant and the measurement noise is equal to zero, the model set* $\mathcal{F}(t)$ *becomes constant in time and so does the FHOCP* (28)*. To this end, we note that since the control input vector converges to a constant value, the applied regressor vector will also converge to a fixed value, that we denote by* φ_{ss}*. Since the output disturbances are constant and the measurement noise is assumed to be zero, the measured plant outputs correspond to the actual ones and by assumption they have constant values:* $\tilde{y}_{ss} = y_{ss} = [y_{ss,1} \dots y_{ss,n_y}]^T$*. Now, recall that the update of the polytopes* $\mathcal{F}_j(t)$*,* $j =$

$1,\dots,n_y$ under the Algorithm 2 has in general the following form (see section 3.1):

$$\begin{aligned}\mathcal{F}_j(t) =&\mathcal{F}_j(t-1)\\ &\cap \{H_j{\in}\mathbb{R}^{n_u m}{:}\,\varphi'(t)^T H_j \le y_{ss,j}{+}\delta'(t)\}\\ &\cap \{H_j{\in}\mathbb{R}^{n_u m}{:}\,\varphi''(t)^T H_j \le -y_{ss,j}{+}\delta''(t)\}.\end{aligned}$$

If the number of faces of the polytope $\mathcal{F}_j(t-1)$ is smaller than $M_1 - 2$, then the polytope is updated according to (10) and therefore $\varphi'(t) = \varphi_{ss}$, $\varphi''(t) = -\varphi_{ss}$ and $\delta'(t) = \delta''(t) = \epsilon_{d_j} + \epsilon_{v_j}$. In this case, the faces that are added to the polytope $\mathcal{F}_j(t)$ are the same at each time step, and therefore the polytope remains unchanged. If the number of faces of the polytope $\mathcal{F}_j(t-1)$ is larger than $M_1 - 2$, then $\varphi'(t)$ and $\varphi''(t)$ are calculated as (see Algorithm 2):

$$\begin{aligned}\varphi'(t) &= \arg\max_{v\in D} \varphi_{ss}^T v\\ \varphi''(t) &= \arg\max_{v\in D} -\varphi_{ss}^T v,\end{aligned}$$

which means that also in this case the directions of the faces that are added to the polytope $\mathcal{F}_j(t), j = 1,\dots,n_y$ remain the same. The solutions of the LP (13) are constant over time, too, resulting in $\delta'(t-1) = \delta'(t) = \delta'$ and $\delta''(t-1) = \delta''(t) = \delta''$. Therefore, the faces added to the polytopes $\mathcal{F}_j(t), j = 1,\dots,n_y$ under the Algorithm 2 are the same at each time step, resulting in a constant model set $\mathcal{F}(t)$. Thus, the nominal model of the plant obtained by solving (14) converges to a fixed value, that we denote by H_c, and the FHOCP (28) becomes time invariant.

Next, we show that the FIR plant model with constant prediction error term that is used to formulate the FHOCP (28) is equivalent to a velocity form state space model. To this end, we introduce a differential form of the predicted regressor vectors as: $\Delta\varphi(k|t) = \varphi(k|t) - \varphi(k-1|t)$, $k = t+2,\dots,t+N$ and $\Delta\varphi(t+1|t) = \varphi(t+1|t) - \varphi(t)$. From (16) and (18) it holds that:

$$\hat{y}(t+1|t) = H_c\Delta\varphi(t+1|t) + \tilde{y}(t). \tag{29}$$

Moreover, from (18) it follows that:

$$\hat{y}(k{+}1|t) = H_c\Delta\varphi(k{+}1|t) + \hat{y}(k|t),\ k = t{+}1,\dots,t{+}N{-}1. \tag{30}$$

In addition, if we define the predicted tracking error as $e(k|t) \doteq \hat{y}(k|t) - y_{des}$, $k \in [t+1, t+N]$ and $e(t|t) \doteq \tilde{y}(t) - y_{des}$, from (18) and (29) it holds that:

$$e(k{+}1|t) = H_c\Delta\varphi(k{+}1|t) + e(k|t),\ k \in [t, t{+}N{-}1]. \tag{31}$$

From (31) it follows that the cost function (17) is equivalent to:

$$\begin{aligned}&J(U, \tilde{y}(t), \Delta\varphi(t)) \doteq\\ &\sum_{k=t}^{t+N-1} e(k+1|t)^T Q e(k+1|t) + \Delta u(k|t)^T R \Delta u(k|t),\\ &\text{Subject to:}\\ &\begin{bmatrix}\Delta\varphi(k{+}1|t)\\ e(k{+}1|t)\end{bmatrix} = \begin{bmatrix} W & \mathbf{0}\\ H_c W & I\end{bmatrix}\begin{bmatrix}\Delta\varphi(k|t)\\ e(k|t)\end{bmatrix} + \begin{bmatrix} Z\\ H_c Z\end{bmatrix}\Delta u(k|t)\\ &e(k|t) = \begin{bmatrix}\mathbf{0} & I\end{bmatrix}\begin{bmatrix}\Delta\varphi(k|t)\\ e(k|t)\end{bmatrix},\end{aligned} \tag{32}$$

where $\mathbf{0}$ and I denote the matrix of all zeros and an identity matrix of appropriate dimensions. The form (32) corresponds to a velocity form state space model as in [27].

From Assumption 3, it follows that the linear system in (32) is controllable [7]. Since, by assumption, control inputs and outputs converge to constant values and no constraints are active in the steady state, we can consider the solution of the optimization problem given by the cost function (32) with the underlying velocity state space model and the constraint (27), without considering the input and output constraints. In this case, the MPC control law can be derived explicitly, and it has the form: $\Delta u(t) = Ke(t|t)$, where $K \in \mathbb{R}^{n_u\times n_y}$ has rank equal to n_y, since the system in (32) is controllable and Q and R are positive definite [34]. Therefore, since by assumption the system reaches a steady state, we have $\Delta u(t) = 0$, which implies that the tracking error satisfies $e(t|t) = y_{ss} - y_{des} = 0$. ■

Note that the condition for having a zero tracking error is that the reference y_{des} is feasible with respect to input and output constraints. If this is not the case, a closest possible feasible reference could be calculated by solving a QP. Another possibility would be to follow the approach in [20] and reformulate the cost function of the problem (28) such that the output reference value would be an additional optimization parameter.

5 Possible extensions of the algorithm

In this section, we comment on possible extensions of the described adaptive control approach. These include the possibility of using basis functions instead of an FIR parametrization of the underlying plant model, in order to reduce the computational requirements of the algorithm, and the possibility of introducing an "exploring" property in order to speed up the identification of the plant dynamics, while retaining the guarantees for satisfaction of input and output constraints. For the sake of brevity, we will describe only the main concepts here, referring to other documents for the technical details.

5.1 Model parametrization with orthonormal basis functions

FIR models have the advantage of being simple and straightforward to use, however, depending on the specific application, the required number of coefficients can be quite large, which can make the proposed adaptive control algorithm computationally demanding. It is reasonable to expect that if some additional information on the system to be controlled is available, such as the approximate location of the dominant poles, the number of coefficients that are needed to model the system can be significantly reduced. This kind of information can be captured well by using model representations given by orthonormal basis functions, like Laguerre [32], Kautz [33] or generalized orthonormal basis functions [8].
By using the basis function model, each of the plant outputs $y_j(t)$, $j = 1, \dots, n_y$ can be written as:

$$\begin{aligned} y_j(t) &= \sum_{i=1}^{n_u} \sum_{k=1}^{m} h_{ji}(k) \zeta\left(\mathcal{L}_k(a,q), u_i(t)\right) + d_j(t) \\ &= H_j^T \varphi(t) + d_j(t), \end{aligned}$$

where $\mathcal{L}_k(a,q)$ are the basis transfer functions defined by parameter a that is selected by the control designer, q is the time shift operator ($qu(t) \doteq u(t+1)$) and the operator $\zeta\left(\mathcal{L}_k(a,q), u_i(t)\right)$ denotes the output of the linear system represented by the transfer function $\mathcal{L}_k(a,q)$ at time step t, when the signal u is applied as its input. The value of a should be chosen such that it captures the dominant dynamic behavior of the plant [32,33], while the functional form of the (stable and strictly proper) functions $\mathcal{L}_k$, $k = 1, \dots, m$ depend on the chosen basis function family. Hence, in this case the dependence of the plant output on the regressor vector and the disturbance can still be written as in (2), the only difference being that the regressor vector now contains the outputs of the individual basis transfer functions and is given by: $\varphi(t) \doteq [\zeta\left(\mathcal{L}_1(a,q)u_1(t)\right), \dots, \zeta\left(\mathcal{L}_m(a,q)u_1(t)\right), \dots, \zeta\left(\mathcal{L}_1(a,q)u_{n_u}(t)\right), \dots, \zeta\left(\mathcal{L}_m(a,q)u_{n_u}(t)\right)]^T$. The evolution of the regressor vector can be described by the following recursive equation:

$$\varphi(t+1) = W_{\mathcal{L}} \varphi(t) + Z_{\mathcal{L}} u(t), \tag{33}$$

where the matrices $W_{\mathcal{L}} \in \mathbb{R}^{n_u m \times n_u m}$ and $Z_{\mathcal{L}} \in \mathbb{R}^{n_u m \times n_u}$ depend on the selected type of the basis functions and the parameter a.

In this framework, by using equation (33) to update the regressor vector $\varphi(t)$, the already described SM identification algorithm can be used with minor modifications to recursively update the model set $\mathcal{F}(t)$ and calculate the nominal model of the plant $H_c(t)$. The control computation part of the adaptive algorithm has the same form as before, with the only difference that $W_{\mathcal{L}}$ and $Z_{\mathcal{L}}$ are used instead of W and Z in (15) and (27). All the results presented in the section 4 still hold.

The main challenge when using a basis function parametrization is the computation of the initial model set $\mathcal{F}(0)$ and of the bounds on the contribution of the truncated part of the basis function sequence to the plant output. To this end we propose a tractable approach in section 3 of [29].

5.2 Adding an exploring property to the control algorithm

The proposed adaptive control algorithm relies on the idea that the discrepancy between the nominal and the actual models of the plant results in control inputs that are informative, such that over time the collected input-output data will reduce the size of the model set $\mathcal{F}(t)$ and therefore improve the accuracy of the identified plant model. Formally, the approach does not require a persistence of excitation assumption to avoid numerical problems, unlike other approaches based on least squares [12]. Nevertheless, in order to achieve good performance, the applied control inputs should be informative enough such that the model set $\mathcal{F}(t)$ becomes small as quickly as possible.

Following the idea of dual control [9] that a good balance between identification and control should be achieved by an adaptive controller, we propose a method to add an exploring property to the proposed algorithm. In the context of MPC, the need of enforcing the persistence of excitation has been addressed by introducing additional constraints in the MPC formulation (see e.g. [22,10]). Our approach is different as it relies on splitting the calculation of the control input in two stages. In the first stage, the FHOCP (28) is solved as usual. The computed optimal input and output sequences and the knowledge of the model set $\mathcal{F}(t)$ are then used to calculate the upper bounds, along the chosen prediction horizon, on the absolute difference between all the possible future outputs of the plant and the nominal optimal output trajectory. In the second stage, by allowing these bounds to be inflated by a factor selected by the control designer, the sequence of control inputs can be recalculated in order to improve the reduction in size of the model set while at the same time enforcing input and output constraints.

With this approach, the relative importance of reference tracking and identification is automatically linked to the amount of information available on the system, which is represented by the size of the model set. In fact, if the model set $\mathcal{F}(t)$ is large, the input and output trajectories computed at the second stage will be allowed to significantly deviate from the ones calculated in the first stage, in order to generate a control input that is informative and reduces the size of the model set $\mathcal{F}(t)$. On the other hand, if the uncertainty is small, the future plant output will be allowed to change only slightly from the first to the second stage.

To be more specific, let us consider the solution of the FHOCP (28), which constitutes the first stage of the described approach. We denote the predicted regressor vectors and plant outputs obtained by solving (28) by $\varphi^*(k|t)$ and $\hat{y}^*(k|t)$, $k \in [t{+}1, t{+}N]$. Then for the second stage, we compute the following quantities:

$$\begin{aligned}\bar{\epsilon}_j(k|t) = &\max\left\{\overline{y}_j(k|t) - \hat{y}_j^*(k|t),\ \hat{y}_j^*(k|t) - \underline{y}_j(k|t)\right\},\\ &k \in [t+1, t+N],\ j = 1, \dots, n_y,\end{aligned} \tag{34}$$

where $\bar{\epsilon}(k|t) = [\bar{\epsilon}_1(k|t), \dots, \bar{\epsilon}_{n_y}(k|t)]^T$, $\bar{\epsilon}_j(k|t) \geq 0$ denotes the maximal possible difference between the future output of the plant and the predicted output of the nominal model at time step k, and

$$\begin{aligned}\overline{y}_j(k|t) &= \max_{A_j(t)H_j \leq b_j(t)} H_j^T \varphi^*(k|t)\\ \underline{y}_j(k|t) &= \min_{A_j(t)H_j \leq b_j(t)} H_j^T \varphi^*(k|t).\end{aligned}$$

In addition, we define the matrix $\Phi(t{+}1|t) \in \mathbb{R}^{n_u m \times n_u m}$ that depends on the $n_u m$ past regressor vectors and the first future regressor vector as:

$$\Phi(t+1|t) = \Big[\, \varphi(t - n_u m + 1)\ \dots\ \varphi(t)\ \ \varphi(t+1|t) \,\Big].$$

This matrix can be indirectly related to the size of the polytopes that will form the model set at the next time step $\mathcal{F}(t+1)$, as indicated by the following result.

Lemma 5.1 *(Theorem 3.2 in [4]) Each of the polytopes $\mathcal{F}_j(t+1)$, $j = 1, \dots, n_y$ obtained by using the polytopic update of the form* (10) *for a sequence of control input vectors that define the regressor vectors forming the matrix $\Phi(t+1|t)$ is guaranteed to have volume smaller than* $\frac{\left(2(\epsilon_{d_j} + \epsilon_{v_j})\right)^{n_u m}}{|\det(\Phi(t+1|t))|}$.

The input to be applied to the plant is then selected within the set of all the control inputs that satisfy input and output constraints and that keep all of the possible predicted output trajectories inside an interval obtained by scaling up the values of $\bar{\epsilon}(k|t)$, centered at the trajectory $\hat{y}^*(k|t)$, $k \in [t+1, t+N]$. In order to improve the knowledge on the system, we need a suitable criterion that is linked to the size of the model set. Considering Lemma 5.1, we choose to use $|\det(\Phi(t+1|t))|$ as an indicator and we compute an input aimed at increasing its value (hence decreasing the volume of the model set). Therefore, the optimization problem to be solved at the second stage of the control input calculation is given as:

$$\begin{aligned}&\max_{U} |\det \Phi(t+1|t)|\\ &\text{Subject to:}\\ &(19),\ (27),\\ &\left.\begin{aligned} H\varphi(k|t) &\leq \hat{y}^*(k|t) + r\bar{\epsilon}(k|t)\\ H\varphi(k|t) &\geq \hat{y}^*(k|t) - r\bar{\epsilon}(k|t)\\ EH\varphi(k|t) + \bar{d} &\leq p \end{aligned}\right\} \begin{aligned}&\forall H \in \mathcal{F}(t)\\ &\forall k \in [t{+}1, t{+}N]\end{aligned}\end{aligned} \tag{35}$$

where $r \in \mathbb{R}$, $r \geq 1$ is a design parameter that indicates by how much the bounds (34) are allowed to be inflated. Problem (35) is a non convex, infinite dimensional program that is in general difficult to solve. However, in this specific case the problem can be solved by solving two additional LPs. For more details, the interested reader is referred to [29] (section 5).

6 Experimental results

The performance of the proposed adaptive control algorithm is illustrated by experiments on a quad-tank testbed.

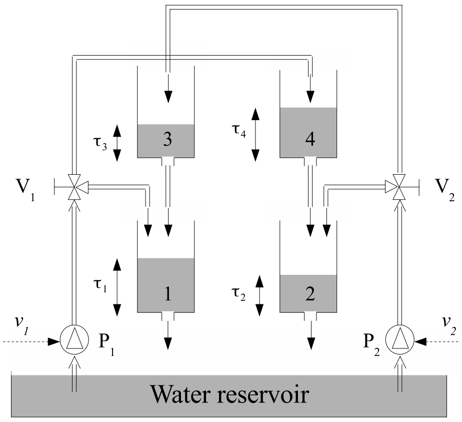

Fig. 1. Quad tank system consisting of 4 water tanks that are mutually connected by a network of pipes and valves. Water is injected into the system from a reservoir by two pumps.

The experimental setup consists of four identical, mutually connected water tanks as in Fig. 1. The water level of each of the tanks is denoted by τ_i, $i = 1, \dots, 4$ and can be measured by a pressure sensor located at the bottom of the tank. Each tank has a water inlet on top and an outlet at the bottom. We denote the cross section of the tanks by A^a and the cross section of the outlets by A^o. Water from the reservoir is injected into the tanks by a system of pumps, valves and pipes as in Fig. 1. The water flows generated by the pumps P_1 and P_2 are proportional to the voltages v_1 and v_2 applied to the pumps

Table 1
Physical parameters of the quad-tank system.

A^a [cm^2]	A^o [cm^2]	k_p [cm^3/sV]	γ_1	γ_2
15.52	0.178	3.3	0.3	0.32

with a constant factor that we denote by k_p. The valves V_1 and V_2 distribute the water injected by the pumps to the upper and the lower tanks. They can be modeled by introducing constants γ_1 and γ_2 that denote the ratio of the flow that is directed into the lower tanks. The numerical values of the quad-tank physical parameters are listed in Table 1

The described plant is a nonlinear system. However, we regulate it in proximity of a steady state, whose corresponding water levels are denoted by $\overline{\tau}_i$, $i = 1, \ldots, 4$. As plant outputs, we take the differences between the water levels of the two lower tanks with respect to their steady state values: $y(t) = [\tau_1(t) - \overline{\tau}_1, \tau_2(t) - \overline{\tau}_2]^T$. Similarly, the control inputs are defined as the differences of the pump voltage levels from the steady state values, denoted by $\overline{v}_1$ and $\overline{v}_2$: $u(t) = [v_1(t) - \overline{v}_1, v_2(t) - \overline{v}_2]^T$. In all the experiments the operating point was the one that corresponds to the pump steady state voltages $\overline{v}_1 = 8$ V and $\overline{v}_2 = 7.25$ V. Note that for the selected values of γ_1 and γ_2, the plant is MIMO non-minimum phase (see e.g. [14]). Therefore, the plant that is being controlled exhibits both nonlinear and non-minimum phase behaviors, which makes it a challenging testbed for an adaptive control algorithm.

A sampling time of 8 seconds was used to control the system. In order to comply with the physical limitations imposed by the water pumps, the input amplitude is required to satisfy the following constraints:

$$\begin{bmatrix} -\overline{u} \\ -\overline{u} \end{bmatrix} \leq \quad u(t) \leq \quad \begin{bmatrix} \overline{u} \\ \overline{u} \end{bmatrix}, \forall t,$$

where $\overline{u} = 2.5\,\text{V}$. In addition, in order to prevent any damage to the setup that may be caused by overflowing the tanks, the following output constraints are required to be satisfied during the control experiment:

$$y(t) \leq \begin{bmatrix} \overline{y} \\ \overline{y} \end{bmatrix}, \forall t,$$

where $\overline{y} = 5.2\,\text{cm}$.

The initial model set $\mathcal{F}(0)$ is formed by assuming an upper and a lower bound on each impulse response coefficient. From the structure of the system model, it is reasonable to assume the lower bound on each of the impulse response coefficients to be 0. The upper bounds are constructed by selecting the same values of the parameters L, μ and ρ for each input-output pair (see section 2). The bounds on the measurement noise and output disturbance are selected to be equal for both outputs. Table 2 lists the values of the chosen design parameters. The employed values are identical for all the input-output pairs. The weighting matrix R is selected as an identity matrix, Q is selected as 2 times the identity matrix and S is taken to be a zero matrix. The ratio between the weights in Q and R reflects the balance between aggressiveness of the control action, and saving of input energy, as increasing Q results in a more aggressive control. By increasing the length of the prediction horizon N, control performance can often be improved at the cost of higher computational complexity. The bound on the measurement noise ϵ_v was determined from sensor readings for constant tank water levels. Selection of the output disturbance bound ϵ_d is not so straightforward, however, unless taken too small or overly conservative, it was not observed to have a significant influence on the controller performance. The initial plant model was formed by selecting a random nonzero point inside the set $\mathcal{F}(0)$.

Table 2
Design parameters of the controller.

ϵ_v [cm]	ϵ_d [cm]	L	μ	ρ	m	α	N
0.25	0.35	1.8	2	0.78	12	0.01	18

The model set $\mathcal{F}(t)$ is updated according to Algorithm 2, where the face number limits where chosen as $M_1 = 200$ and $M_2 = 48$. The set of predefined face directions D was constructed by the vectors that form an ∞-norm ball. The adaptive control algorithm was implemented in Matlab and run on a laptop with Intel i7-36667U processor. The solver Gurobi [13] was used in order to solve LPs and the QP required by the algorithm. With this configuration, the maximal execution time of the algorithm, at each sampling instant, was not greater than 4 seconds.

The set $\mathcal{F}(0)$ was selected quite conservatively, which is illustrated in Fig. 2 that compares the initial model set with the impulse response coefficient values of the nominal model at the end of a typical experiment. Despite this, the experimental results of Fig. 3 show that good reference tracking is obtained. As predicted by the theoretical analysis, output constraints are satisfied, also during the adaptation transient. In addition, the control performance improves over time as more information is gathered and the model set is reduced. The contraction of the model set over time is illustrated by the fact that the gap between the maximal and minimal values of the outputs, computed by considering all possible plants in the model set, decreases over time (see e.g. Fig. 3, gray lines).

In order to illustrate the effectiveness of the proposed approach in satisfying the output constraints, we compare

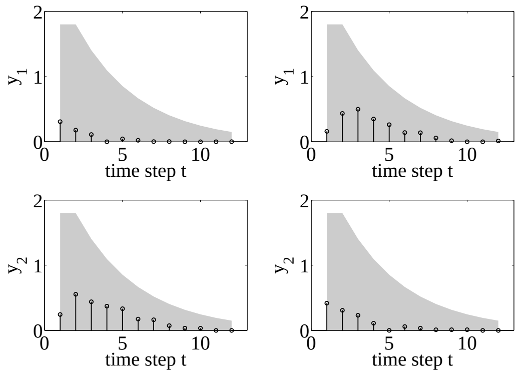

Fig. 2. Initial model set $\mathcal{F}(0)$ (grey area) compared with the impulse response coefficients of the nominal model at the end of the experiment. The top left plot shows the transfer function from u_1 to y_1, the top right plot from u_2 to y_1, the bottom left from u_1 to y_2 and the bottom right from u_2 to y_2.

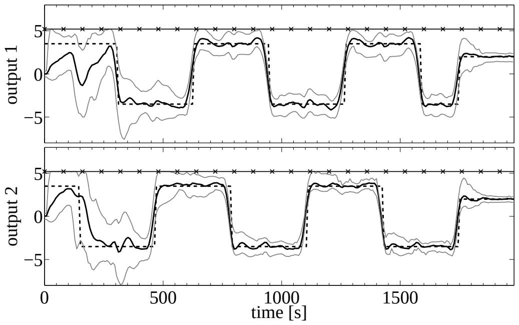

Fig. 3. Experimental results obtained by applying the proposed adaptive control algorithm to the quad-tank testbed. The desired output references y_{des} (dashed black lines) are compared with the measured plant outputs $\tilde{y}(t)$ (solid black lines), for output y_1 (upper plot) and y_2 (lower plot). The uncertainty intervals of the outputs for all the plants in the model set (solid gray lines) are also shown, as well as the output constraints (solid black lines with ×).

the performance of the proposed controller with that of a certainty equivalence adaptive controller, which uses recursive least squares to identify the nominal model of the plant and a receding horizon optimal control algorithm similar to (28) in order to control the system outputs. Since this modified algorithm uses only a point estimate of the plant model, instead of robustly enforcing output constraints, soft output constraints are used in order to avoid feasibility problems. Apart from this difference, the same tuning parameters and initial guess for the plant model were used. The experimental results obtained with this adaptive controller are shown in Fig. 4. As it can be seen, the output constraints are violated during the adaptation transient in this case. However, due to a more aggressive control action in the initial phase, which results in a more informative collection of input/output data, the adaptation is faster. On the other hand, in order to robustly satisfy the output constraints, the newly proposed adaptive control algorithm introduces conservativeness during the adaptation transient, since it makes sure that the outputs of all the plants inside the model set satisfy output constraints. This results in quite cautious control at the beginning, when the model uncertainty is large. However, as the uncertainty is reduced, the tracking performance of the controller improves, as it can be seen in Fig. 3.

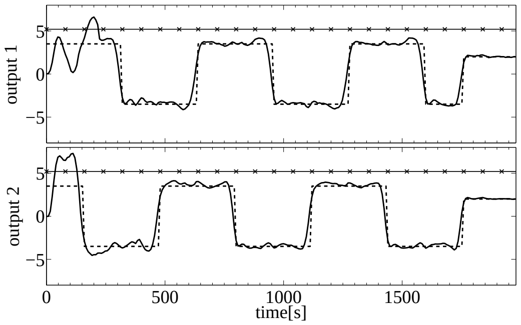

Fig. 4. Experiment results obtained by applying a controller that uses a recursive least squares point estimate of the plant model to the quad-tank testbed. The desired output references y_{des} (dashed black lines) are compared to the measured plant output $\tilde{y}(t)$ (solid black lines) and the maximal allowed values of the plant outputs (solid black lines with ×), for output y_1 (upper plot) and y_2 (lower plot).

7 Conclusion

We proposed an adaptive model predictive control algorithm for open-loop stable, linear, time invariant MIMO systems subject to both input and output constraints. The method relies on real-time SM identification to provide guaranteed bounds on any linear combination of predicted system outputs. These bounds are used to design a receding horizon controller able to robustly satisfy output constraints. The proposed adaptive control algorithm exhibits integral action. It requires the solution of standard convex programs (LPs and QPs) which are guaranteed to be recursively feasible. We also described the possibility to consider various model parameterizations and to add an active exploring capability. We reported the experimental results obtained by testing the approach on a non-minimum phase quad-tank system.

References

[1] V. Adetola, D. DeHan, and M. Guay. Robust adaptive MPC for constrained uncertain nonlinear systems. *International Journal of Adaptive Control and Signal Processing*, 25:155–168, 2011.

[2] K. J. Åström and B. Wittenmark. *Adaptive control, second edition.* Addison-Wesley, 1995.

[3] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49:1216–1226, 2013.

[4] E-W Bai, R. Tempo, and H. Cho. Membership set estimators: size, optimal inputs, complexity and relations with least squares. *IEE Transactions on circuits and systems -I: Fundamental theory and applications*, 42:266–277, 1995.

[5] A. Bemporad and M. Morari. Robust model predictive control: A survey. *Robustness in Identification and Control*, 245:207–226, 1999.

[6] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, 1997.

[7] G. Betti, M. Farina, and R. Scattolini. A robust MPC algorithm for offset-free tracking of constant reference signals. *IEEE Transactions on Automatic Control*, 58:2394–2400, 2013.

[8] P. M. J. Van den Hof, P. S. C. Heuberger, and J. Bokor. System identification with generalized orthonormal basis functions. *Automatica*, 31:1821–1834, 1995.

[9] A. A. Feldbaum. Dual control theory, Parts I and II. *Automation and Remote control*, 21:874–880, 1033–1039, 1961.

[10] H. Genceli and M. Nikolaou. New approach to constrained predictive control with simultaneous model identification. *AIChE Journal*, 42:2857–2868, 1998.

[11] G. C. Goodwin, M. M. Seron, and J. A. De Don. *Constrained control and estimation: an optimisation approach*. Springer, London, 2005.

[12] G. C. Goodwin and K. S. Sin. *Adaptive filtering, prediction and control*. Prentice-Hall INC., 1984.

[13] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2013. http://www.gurobi.com.

[14] K. H. Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Automatic Control*, 8:456–465, 2000.

[15] J.-S. Kim. Recent advances in adaptive MPC. *International conference on control, automation and systems (ICCAS), Kintex, Gyeonggi-do, Korea*, pages 218–222, 2010.

[16] T.-K. Kim and T. Sugie. Adaptive receding horizon predictive control for constrained discrete-time systems with parameter uncertainties. *International Journal of Control*, 81:62–73, 2008.

[17] J.-S. Kimn, T.-W. Yoon, H. Shim, and J.H. Seo. Switching adaptive output feedback model predictive control for a class of input-constrained linear plants. *IET Control Theory & Applications*, 2:573–582, 2008.

[18] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi. *Adaptive control: algorithms, analysis and applications*. Springer, New York, 2011.

[19] Y. I . Lee and B. Kouvaritakis. A linear programming approach to constrained robust predictive control. *IEEE Transactions on Automatic Control*, 45:1765–1770, 2000.

[20] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44:2382–2387, 2008.

[21] V. M. Maniar, S. L. Shah, D. G. Fischer, and R. K. Mutha. Multivariable constrained adaptive GPC: theory and experimental evalation. *International Journal of Adaptive Control and Signal Processing*, 11:343–365, 1997.

[22] G. Marafioti, R. R. Bitmead, and M. Hovd. Persistently exciting model predictive control. *International Journal of Adaptive Control and Signal Processing*, 28:536–552, 2014.

[23] S. Maraoui and H. Messaoud. Design and comparative study of limited complexity bounding error identification algorithms. *IFAC symposium on system structure and control, Prague-Czech Republique*, 2001.

[24] T. H. Mattheiss. An algorithm for determining irrelevant constraints and all vertices in system of linear inequalities. *Operations Research*, 21:247–260, 1973.

[25] G. Nikolakopoulos, L. Dritsas, A. Tzes, and J. Lygeros. Adaptive constrained control of uncertain ARMA-systems based on set membership identification. *Mediterranean conference on control and automation, Ancona , Italy*, 2006.

[26] J. P. Norton and S. M. Veres. Outliers in bound-based state estimation and identification. *Circuits and Systems*, 1:790–793, 1993.

[27] D. M. Prett and C. E. Garcia. *Fundamental process control*. Butterworths series in chemical engineering, 1988.

[28] L. Pronzato and E. Walter. Robustness to outliers of bounded-error estimators and consequences on experiment design. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding approaches to system identification*, pages 199–212. Plenum Press, New York, 1993.

[29] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari. Adaptive model predictive control with exploring property for constrained linear systems that uses basis function model parametrization, 2013. arXiv:1309.5304 (available online).

[30] S. M. Veres and J. P. Norton. Predictive self-tuning control by parameter bounding and worst-case design. *Automatica*, 29:911–928, 1993.

[31] S. M. Veres, H. Messa Oud, and J. P. Norton. Limited-complexity model-unfalsifying adaptive tracking-control. *International Journal of Control*, 72:1417–1426, 1999.

[32] B. Wahlberg. System identification using Laguerre models. *IEEE Transactions on Automatic Control*, 36:551–562, 1991.

[33] B. Wahlberg. System identification using Kautz models. *IEEE Transactions on Automatic Control*, 39:1276–1282, 1994.

[34] L. Wang. A tutorial on model predictive control: Using a linear velocity-form model. *Developments in Chemical engineering and Mineral Processing*, 12:573–614, 2004.

A Bounds on the unmodeled dynamics

We denote the contribution of the unmodeled dynamics to each of the outputs j by $\eta_j(t)$, $j = 1, \dots, n_y$, where:

$$\eta_j(t) = \sum_{i=1}^{n_u} \sum_{k=m+1}^{\infty} h_{ji}(k) u_i(t-k).$$

Since the constraint set of the control input magnitudes is bounded, the upper bound and the lower bound on each of the control inputs $i = 1, \dots n_u$ is given by:

$$\begin{array}{l} \overline{u}_i = \max\limits_{Cu \leq g} u_i \\ \underline{u}_i = \min\limits_{Cu \leq g} u_i \end{array}, \; i = 1, \dots, n_u.$$

If, in addition we assume that the bounds on the impulse response coefficients, given by (5), can be extended to the case of an infinite impulse response, then the upper bound on the contribution of the truncated part of the impulse response, $|\eta_j(t)| \leq \overline{\eta}_j$, $\forall t$, is given by:

$$\overline{\eta}_j = \sum_{i=1}^{n_u} \max(|\underline{u}_i|, |\overline{u}_i|) L_{ji} \rho_{ji}^{m-\mu_{ji}} \frac{\rho_{ji}}{1-\rho_{ji}},\; j = 1, \ldots, n_y.$$

This formula can be used to calculate the joint bounds on the contribution of the output disturbances and the unmodeled dynamics to the plant outputs in (3).

B Definition of the matrices in (15)

We first define the matrix $w \in \mathbb{R}^{n_u \times n_u}$ with the following structure:

$$w = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

Based on this, the matrix W is given by:

$$W = \begin{bmatrix} w & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & w & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & w \end{bmatrix} \in \mathbb{R}^{n_u m \times n_u m},$$

where $\mathbf{0}$ denotes the matrix of all zeros with appropriate dimension. In addition, let $z = [1, 0, \ldots, 0]^T \in \mathbb{R}^m$, the matrix Z is given as:

$$Z = \begin{bmatrix} z & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & z & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & z \end{bmatrix} \in \mathbb{R}^{n_u m \times n_u}.$$