Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 05-199

Two Key Ingredients Enabling Efficient Long-Horizon Robust MPC [1]

B. Pluymers[2] , J.A. Rossiter[3] , J.A.K. Suykens[2] and B. De Moor[2]

September 2005

Submitted for publication

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.be* in the directory *pub/sista/pluymers/reports/acc05_longhorizon.pdf*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD-SISTA, Kasteelpark 10, 3001 Leuven, Belgium, Tel. 32/16/32 10 35, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.be/scd*. E-mail: {*bert.pluymers,johan.suykens,bart.demoor*}*@esat.kuleuven.ac.be*.

Research supported by Research Council KUL: GOA-Mefisto 666, GOA-Ambiorics, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99 (multilinear algebra), G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0452.04 (QC), G.0499.04 (robust SVM), research communities (ICCoS, ANMMM, MLDM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, GBOU (McKnow) Belgian Federal Government: Belgian Federal Science Policy Office: IUAP V-22 (Dynamical Systems and Control: Computation, Identification and Modelling, 2002-2006), PODO-II (CP/01/40: TMS and Sustainibility); EU: FP5-Quprodis; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard; Bert Pluymers is a research assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry). Dr. Johan Suykens is an associate professor at the Katholieke Universiteit Leuven. Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium.

[3]University of Sheffield, Department of Automatic Control and Systems Engineering, Mappin Street, Sheffield S1 3JD, United Kingdom, E-mail: *j.a.rossiter@sheffield.ac.uk*

Abstract

This paper identifies and combines two main ingredients that are necessary to formulate an efficient robust MPC algorithm. Efficiency is interpreted as non-conservative constraint handling and the ability to handle long horizon lengths (e.g. >20) and large-dimensional systems (e.g. >10) in a computationally tractable way. The first ingredient, *within-horizon feedback*, eliminates the spreading of the predictions due to model uncertainties or disturbances. The specific form of feedback used in this paper additionally allows the constraints of the MPC algorithm to be calculated as an invariant set of an augmented system. The second ingredient is the ability to construct *reduced complexity polyhedral invariant sets* for the augmented system. The main contributions of this paper are **a)** the combination of both ingredients into an efficient robust MPC algorithm that enables the use of significantly larger horizon lengths and significantly larger dimensional models than previously computationally feasible and **b)** the assessment of the computational complexity and the control performance using the model of a real-world example.

Two Key Ingredients Enabling Efficient Long-Horizon Robust MPC

B. Pluymers†, J.A. Rossiter‡, J.A.K. Suykens†, B. De Moor†

†Katholieke Universiteit Leuven
Department of Electrical Engineering, ESAT-SCD-SISTA
E-Mail : {bert.pluymers, johan.suykens, bart.demoor}@esat.kuleuven.be
Internet : http://www.esat.kuleuven.be/scd/

‡University of Sheffield
Department of Automatic Control and Systems Engineering
E-Mail : j.a.rossiter@sheffield.ac.uk
Internet : http://www.shef.ac.uk/acse/

***Abstract*— This paper identifies and combines two main ingredients that are necessary to formulate an efficient robust MPC algorithm. Efficiency is interpreted as non-conservative constraint handling and the ability to handle long horizon lengths (e.g. >20) and large-dimensional systems (e.g. >10) in a computationally tractable way. The first ingredient, *within-horizon feedback*, eliminates the spreading of the predictions due to model uncertainties or disturbances. The specific form of feedback used in this paper additionally allows the constraints of the MPC algorithm to be calculated as an invariant set of an augmented system. The second ingredient is the ability to construct *reduced complexity polyhedral invariant sets* for the augmented system. The main contributions of this paper are a) the combination of both ingredients into an efficient robust MPC algorithm that enables the use of significantly larger horizon lengths and significantly larger dimensional models than previously computationally feasible and b) the assessment of the computational complexity and the control performance using the model of a real-world example.**

I. INTRODUCTION

Modelbased Predictive Control (MPC, [4], [21]) has become the standard advanced process control technique in recent decennia [20] and has reached a high degree of maturity in its linear variant [12]. Current research mainly focusses on nonlinear and robust MPC and related computational aspects. However, current methods typically represent a trade-off between optimal control performance, non-conservative constraint handling and efficient on-line optimization. This paper discusses (and provides) the two key methods needed to eliminate this compromise and combine these three properties in one robust MPC algorithm.

Traditionally two different robustness aspects have been assessed in MPC literature. The first is the inherent robustness that is present in MPC schemes that are not specifically designed for robust stability and/or performance [6], [11]. The second is the explicit inclusion of robustness requirements into the design of the control law [8], [10], [13], [23]. This second approach is followed in this article.

Traditionally robust MPC requires the solution of min-max optimization problem, where an optimization over all possible control moves is performed in order to minimize a worst-case (over all possible uncertainty realizations) cost function. Furthermore, constraint satisfaction also has to be guaranteed for all possible future trajectories. It was shown in [23] that one has to consider feedback within the horizon in order to be able to prove stability of the resulting algorithm. This means that one has to optimize over a sequence of control strategies rather than a sequence of fixed control moves, which drastically increases the number of optimization variables. All these elements contribute to the on-line computational complexity which makes robust MPC often intractable.

Several alternative robust MPC algorithms with a reduced computational complexity can be identified. The classical result by Kothare [8], which can be seen as a robust, constrained LQR algorithm, has computational advantages in the sense that it does not use a finite-horizon with the corresponding exponential number of optimization variables, but rather optimizes over a linear feedback policy. The drawback, however, is conservative constraint handling and the use of on-line LMI-based optimization, that can still be computationally intractable in some cases.

Another approach is the use of tube-based methods like the ones introduced in [18] and [13]. The first method has linear scaling properties as a function of the horizon length and the state dimension, but exhibits conservative constraint handling due to the use of ellipsoidal invariant sets and still has a large computational complexity due to the use of on-line LMI-based optimization. The latter method is based on Quadratic Programming (QP) and exhibits the most similarities with the algorithm proposed in this paper. However, both algorithms are markedly different, since the algorithm proposed here makes use of the maximal robust positive invariant set, whereas the algorithm proposed in [13] makes use of minimal robust positive invariant sets.

Recently, a new approach was proposed in [16], [17], extending the algorithm that was introduced in [22] to the robust case. The algorithm makes use of a fixed linear feedback law that is superimposed with a finite number of input perturbations over which the on-line optimization takes place. In this way the idea of including within-horizon feedback introduced in [23] is combined with the linear

number of control moves (as a function of horizon length) of nominal MPC. Furthermore, the worst-case objective function can be solved off-line by means of an LMI-based optimization problem reducing the on-line optimization to a QP. However, the number of constraints of this optimization typically still increases exponentially as a function of the horizon length. This paper improves upon this algorithm in order to obtain an algorithm that is computationally tractable to implement, without significantly impairing control performance or constraint handling.

This paper is organized as follows. Section II gives the necessary background and introduces the notations used in the rest of the paper. Section III then explains the importance of within-horizon feedback and the parametrization of the input sequence after which Section IV then explains the main principles of set invariance and summarizes a recent result allowing the construction of reduced complexity polyhedral invariant sets. Section V then states the new algorithm based on the previous two sections. Section VI finally gives some examples illustrating the efficiency of the new algorithm following by conclusions and future research directions in Section VII.

Throughout this paper MATLAB notation is used to denote rows, columns and elements of matrices, i.e. $A_{[i,:]}$ denotes the i-th row of matrix A and $A_{[i,j]}$ denotes the element at row i and column j of matrix A. The notation $[A_1;\ A_2]$ will be used to denote stacked matrices $[A_1^{\mathrm{T}}\ A_2^{\mathrm{T}}]^{\mathrm{T}}$.

II. BACKGROUND

This section explains the notations and the basic concepts used in the rest of the paper, i.e. Modelbased Predictive Control (MPC) and Set Invariance.

A. Notation

In this paper we consider Linear Parameter-Varying (LPV) systems

$$x_{k+1} = A(k)x_k + B(k)u_k, \qquad k = 0, \dots, \infty, \tag{1a}$$

with polytopic uncertainty description (see [3])

$$[A(k)\ B(k)] \in \Omega \triangleq \mathrm{Co}\left\{[A_1\ B_1], \dots, [A_L\ B_L]\right\}, \tag{1b}$$

where L is a positive integer denoting the number of vertices describing the convex hull Ω and $A_i \in \mathbb{R}^{n_x \times n_x}, B_i \in \mathbb{R}^{n_x \times n_u}$ denote the dynamic behavior at the vertices of this polytope. n_x and n_u respectively denote the number of system states and inputs. $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ respectively denote the system state and input at discrete time k. The system is subject to input and state constraints of the form

$$x_k \in \mathcal{X} \triangleq \{x | A_x x \leq 1_{\mathrm{v}}\}, \qquad \forall k, \tag{2a}$$

$$u_k \in \mathcal{U} \triangleq \{x | A_u u \leq 1_{\mathrm{v}}\}, \qquad \forall k, \tag{2b}$$

where 1_{v} in each equation denotes a vector of appropriate dimensions containing only 1's.

B. Nominal MPC

Before moving to robust MPC in the next section we first give a brief description of nominal linear MPC, i.e. MPC for linear systems without model uncertainty or disturbances.

Assume a nominal model is given by (1) with $L = 1$ and denote $[A\ B] = [A_1\ B_1]$. The nominal MPC can be defined as a control scheme that for all $k > 0$, given a current state x_k, calculates an optimal input sequence $\mathbf{u}_k = [u_k \dots u_{k+n_c-1}]$ with corresponding state sequence $\mathbf{x}_k = [x_{k+1} \dots x_{k+n_c}]$ by solving the following optimization problem :

$$\min_{\mathbf{u}_k} \sum_{i=0}^{n_c-1} \left(x_{k+i}^{\mathrm{T}} Q x_{k+i} + u_{k+i}^{\mathrm{T}} R u_{k+i}\right) + x_{k+n_c}^{\mathrm{T}} Q_{\mathrm{f}} x_{k+n_c}, \tag{3a}$$

subject to

$$x_{k+i} \in \mathcal{X}, \qquad i = 0, \dots, n_c - 1, \tag{3b}$$

$$u_{k+1} \in \mathcal{U}, \qquad i = 0, \dots, n_c - 1, \tag{3c}$$

$$x_{k+n_c} \in \mathcal{X}_{\mathrm{f}}, \tag{3d}$$

after which the optimal input u_k^{o} is applied to the system. This procedure is repeated at every time instant.

The integer $n_c > 0$ denotes the length of the prediction horizon, $Q = Q^{\mathrm{T}} > 0 \in \mathbb{R}^{n_x \times n_x}$ and $R = R^{\mathrm{T}} > 0 \in \mathbb{R}^{n_u \times n_u}$ denote positive definite state and input weighting matrices. Q_{f} and $\mathcal{X}_{\mathrm{f}}$ respectively denote the terminal cost and terminal constraint used to guarantee closed-loop stability [12]. Closed-loop stability of this algorithm is guaranteed if there is no model-plant mismatch and there are no disturbances acting on the system. We refer to [12] for details.

C. Set Invariance

The following two definitions summarize the concept of robust positive set invariance :

Definition 1 (Robust Positive Invariance): *Given an autonomous LPV system*

$$x_{k+1} = \Phi(k)x_k, \quad \Phi(k) \in \Omega_{\mathrm{aut}} \triangleq \mathrm{Co}\{\Phi_1, \dots, \Phi_L\}, \tag{4}$$

then $\mathcal{S} \in \mathbb{R}^{n_x}$ is a robust positive invariant set if

$$\Phi x \in \mathcal{S}, \quad \forall x \in \mathcal{S}, \forall \Phi \in \Omega_{\mathrm{aut}}. \tag{5}$$

Definition 2 (Feasibility): *An invariant set $\mathcal{S}$ for a system* (4) *is* feasible *with respect to constraints* (2a) *if*

$$\mathcal{S} \subseteq \mathcal{X}. \tag{6}$$

The *Maximal Admissable Set* (MAS, [7]) is defined as the largest feasible positive invariant set for a given system and constraints in the sense that no other feasible positive invariant set can contain points outside the MAS. This set can be shown to be uniquely defined and is equal to the union of all possible feasible positive invariant sets. In the rest of this paper we will refer to a *feasible robust positive invariant set* by the shorter term *feasible invariant set*. We refer to [2] for an overview on set invariance.

III. ROBUST FEEDBACK MPC

This section clarifies the need for **Ingredient 1 :** within-horizon feedback in order to obtain an efficient robust MPC algorithm.

A. Open-Loop Min-Max MPC

A straightforward modification of the nominal MPC algorithm introduced in the previous section consists of replacing the minimization with a min-max optimization in order to capture the worst-case behavior of the system over all possible uncertainty realizations [24] :

$$\min_{\mathbf{u}_k} \max_{\substack{j_i\in\{1,\ldots,L\}\\ i=0,\ldots,n_c-1}} x_k^{\mathrm{T}}Qx_k + u_k^{\mathrm{T}}Ru_k + \sum_{i=1}^{n_c-1}\left(x_{k+i,(j_0,\ldots,j_{i-1})}^{\mathrm{T}}Qx_{k+i,(j_0,\ldots,j_{i-1})} + u_{k+i}^{\mathrm{T}}Ru_{k+i}\right) + x_{k+n_c,(j_0,\ldots,j_{n_c-1})}^{\mathrm{T}}Q_{\mathrm{f}}x_{k+n_c,(j_0,\ldots,j_{n_c-1})},$$

subject to

$$\begin{aligned} x_{k+i,(j_0,\ldots,j_{i-1})} &\in \mathcal{X}, & i &= 1,\ldots,n_c-1,\\ & & j_{0,\ldots,i-1} &= 1,\ldots,L,\\ u_{k+i} &\in \mathcal{U}, & i &= 0,\ldots,n_c-1,\\ x_{k+n_c,(j_0,\ldots,j_{n_c-1})} &\in \mathcal{X}_{\mathrm{f}}, & j_{0,\ldots,n_c-1} &= 1,\ldots,L, \end{aligned}$$

with

$$\begin{aligned} x_{k+1,(j_0)} &= A_{j_0}x_k + B_{j_0}u_k, \qquad j_0 = 1,\ldots,L,\\ x_{k+i+1,(j_0,\ldots,j_i)} &= A_{j_i}x_{k+i,(j_0,\ldots,j_{i-1})} + B_{j_i}u_{k+i},\\ & \qquad\qquad j_i = 1,\ldots,L. \end{aligned}$$

In this way the worst-case objective function over all possible (time-dependent) uncertainty realizations is minimized and constraints are imposed on all possible future trajectories. It is clear that the number of constraints increases exponentially as a function of n_c, making the algorithm practically intractable to implement. Recursive feasibility is not guaranteed in the presence of state constraints and hence stability can only be proven if no state constraints are present. Furthermore, since the optimization is performed over a single open-loop input profile, for larger prediction horizons the predicted state trajectories can spread out severely due to the model uncertainty, which can lead to significant feasibility problems.

B. Closed-Loop Min-Max MPC

The problem of the spread of the predicted state trajectories and the lack of a recursive feasibility guarantee in the presence of state constraints was overcome in [23] by means of introducing within-horizon feedback, i.e. by making the input trajectory state-dependent :

$$\min_{\mathbf{u}_k} \max_{\substack{j_i\in\{1,\ldots,L\}\\ i=0,\ldots,n_c-1}} x_k^{\mathrm{T}}Qx_k + u_k^{\mathrm{T}}Ru_k + \sum_{i=1}^{n_c-1}\left(x_{k+i,(j_0,\ldots,j_{i-1})}^{\mathrm{T}}Qx_{k+i,(j_0,\ldots,j_{i-1})} + u_{k+i,(j_0,\ldots,j_{i-1})}^{\mathrm{T}}Ru_{k+i,(j_0,\ldots,j_{i-1})}\right) + x_{k+n_c,(j_0,\ldots,j_{n_c-1})}^{\mathrm{T}}Q_{\mathrm{f}}x_{k+n_c,(j_0,\ldots,j_{n_c-1})},$$

subject to

$$\begin{aligned} & u_k \in \mathcal{U},\\ & x_{k+i,(j_0,\ldots,j_{i-1})} \in \mathcal{X}, \qquad \begin{cases} i = 1,\ldots,n_c-1,\\ j_{0,\ldots,i-1} = 1,\ldots,L, \end{cases}\\ & u_{k+i,(j_0,\ldots,j_{i-1})} \in \mathcal{U},\\ & x_{k+n_c,(j_0,\ldots,j_{n_c-1})} \in \mathcal{X}_{\mathrm{f}}, \qquad j_{0,\ldots,n_c-1} = 1,\ldots,L, \end{aligned}$$

with

$$\begin{aligned} x_{k+1,(j_0)} &= A_{j_0}x_k + B_{j_0}u_k, \qquad j_0 = 1,\ldots,L,\\ x_{k+i+1,(j_0,\ldots,j_i)} &= A_{j_i}x_{k+i,(j_0,\ldots,j_{i-1})} + B_{j_i}u_{k+i,(j_0,\ldots,j_{i-1})},\\ & \qquad\qquad j_i = 1,\ldots,L, \end{aligned}$$

where $\mathbf{u}_k$ now contains all possible input vectors $u_k, u_{k+i,(j_0,\ldots,j_{i-1})}, i = 1,\ldots,n_c-1, j_{0,\ldots,i-1} = 1,\ldots,L$. The state-dependence of the inputs allows to keep the spread of the predicted state trajectories small and allows the formulation of a recursive feasibility and stability proof in the presence of state constraints. However, both the number of constraints and the number of optimization variables now increase exponentially as a function of the prediction horizon n_c, which further prohibits a practical implementation.

C. Closed-Loop Min-Max MPC with Parameterized Inputs

The exponential number of optimization variables obtained in the previous subsection can be reduced to a number of variables that depends linearly on the prediction horizon, by parameterizing the state-dependent input sequence as follows :

$$\begin{aligned} u_{k+i} &= -Kx_{k+i} + Ec_{k+i}, \qquad i = 0,\ldots,n_c-1,\\ u_{k+i} &= -Kx_{k+1}, \qquad\qquad i \geq n_c, \end{aligned}$$

where the vectors $c_{k+i} \in \mathbb{R}^{n_p}, i = 0,\ldots,n_c-1$ are the variables over which the on-line optimization is performed and $E \in \mathbb{R}^{n_u\times n_p}$ is a user-defined real matrix. This input parametrization is slightly more general than the one used in [9], [17], [22] and comes down to the following state-dependent input parametrization :

$$\begin{aligned} u_k &= -Kx_k + Ec_k,\\ u_{k+i,(j_0,\ldots,j_{i-1})} &= -Kx_{k+i,(j_0,\ldots,j_{i-1})} + Ec_{k+i},\\ & \qquad i = 1,\ldots,n_c-1, j_i = 1,\ldots,L. \end{aligned}$$

This shows that, while the number of optimization variables is drastically reduced, the closed-loop character of the algorithm is maintained. A further advantage is that this

parametrization allows the maximization of the on-line min-max optimization to be performed off-line by constructing an autonomous augmented system

$$z_{k+1} = \Phi(k) z_k, \qquad \Phi(k) \in \Omega_{\text{aug}}, \tag{7a}$$

with $z_k = [x_k;\ c_k;\ \dots\ c_{k+n_c-1}]$ and the vertices of Ω_{aug} defined by the matrices $\Phi_{\text{aug},i} = 1, \dots, L$:

$$\Phi_{\text{aug},i} = \begin{bmatrix} A_i - B_i K & [B_i E\ 0\ 0\ \dots\ 0] \\ 0 & S_{n_c} \end{bmatrix}, \quad S_{n_c} = \begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & & I \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{7b}$$

subject to constraints

$$\begin{bmatrix} A_x & 0 & 0 \\ -A_u K & A_u E & 0 \end{bmatrix} z_k \leq 1_{\text{v}}. \tag{8}$$

A worst-case cost function $z_k^{\text{T}} P z_k, P = P^{\text{T}} > 0$ can be computed off-line as

$$\min_{P = P^{\text{T}} > 0} \quad \text{trace}(P), \tag{9a}$$

subject to

$$P - \Phi_{\text{aug},i}^{\text{T}} P \Phi_{\text{aug},i} > \tilde{Q} + \tilde{R}, \qquad i = 1, \dots, L, \tag{9b}$$

with

$$\tilde{Q} = \text{diag}(Q, 0, \dots, 0), \tag{9c}$$

$$\tilde{R} = [-K\ E\ 0\ \dots\ 0]^{\text{T}} R [-K\ E\ 0\ \dots\ 0], \tag{9d}$$

while the on-line optimization occurs over all z_k that, given the current state measurement x_k, lie within an off-line constructed set $\mathcal{S}$ that is invariant with respect to (7b) and feasible with respect to (8). Recursive feasibility and asymptotic stability can be proven as will be clarified in Section V. It is clear that the number of optimization variables of this formulation is linear as a function of the horizon length. The number and nature of the optimization constraints depend on the description of the set $\mathcal{S}$, which is clarified in the next section.

IV. REDUCED COMPLEXITY INVARIANT SETS

This section clarifies **Ingredient 2:** the construction of reduced complexity polyhedral invariant sets for LPV systems. The first subsection shows how to construct polyhedral invariant sets for LPV systems, which allows the algorithm of the previous section to be formulated as a Quadratic Program (QP), while the second subsection shows how to reduce the number of constraints describing the polyhedral invariant set. In this way polyhedral invariant sets are obtained that can be described by a number of constraints that typically scales linearly as a function of the state dimension of the system for which the set is calculated.

A. Polyhedral invariant sets for LPV systems

Recently an algorithm was introduced [17] that allows the construction of polyhedral invariant sets for LPV systems with polytopic uncertainty description. The algorithm iteratively adds constraints to the imposed constraints set $\mathcal{X}$ until robust invariance is obtained. The algorithm can be summarized as follows :

Algorithm 1 (Invariant Sets for LPV systems): Given an autonomous system (4) subject to constraints (2a), construct an invariant set $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq 1_{\text{v}}\}$ using the following steps :

- Set $A_{\mathcal{S}} := T_0 = A_x$ and $i := 1$.
- Iterate until all constraints $T_{i-1}\Phi_{\text{aut},1} \leq 1_{\text{v}}, \dots, T_{i-1}\Phi_{\text{aut},L} \leq 1_{\text{v}}$ are redundant with respect to $A_{\mathcal{S}} x \leq 1_{\text{v}}$:
  - Set $T_i = [T_{i-1}\Phi_{\text{aut},1}; \dots; T_{i-1}\Phi_{\text{aut},L}]$.
  - Remove all constraints from T_i that are redundant with respect to $[A_{\mathcal{S}}; T_i]$.
  - Set $A_{\mathcal{S}} = [A_{\mathcal{S}}; T_i]$.
  - Remove all redundant constraints from $A_{\mathcal{S}}$.

Return $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq 1_{\text{v}}\}$.

This algorithm converges in a finite number of iterations if (4) is quadratically stable and the resulting set can be proven to be the MAS for the given system. Due to the removal of redundant constraints every iteration, a significant reduction in computational time is obtained compared to a more naive implementation. However, it can be seen that in the worst-case the number of constraints can still increase with a factor L in every iteration. Due to the specific structure of (7b) this worst-case scenario is what typically is obtained when constructing a polyhedral invariant set. Therefore, the number of constraints describing $\mathcal{S}$ typically increases exponentially as a function of the horizon length. The next section describes an algorithm that is able to reduce this number significantly.

B. Reduced Complexity Polyhedral invariant sets for LPV systems

In this section we describe the *pruning* method explained in [19]. This method strategically tightens certain constraints of the invariant set in order to increase the number of redundant constraints. In this way at each iteration a reduced number of constraints is added to the set. The algorithm can be summarized as follows :

Algorithm 2 (Reduced complexity invariant sets): Consider an autonomous system (4) subject to constraints (2a). Furthermore, consider user-defined scalars $d_1 > \max_j \|(A_x)_{[j,:]}\|,\ d_2 \in [\max_j \|\Phi_j\|, 1)$ and $\gamma > 0$. Construct an invariant set $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq 1_{\text{v}}\}$ using the following steps :

- Set $A_{\mathcal{S}} := T_0 = A_x$ and $i := 1$.
- Iterate until all constraints $T_{i-1}\Phi_{\text{aut},1} \leq 1_{\text{v}}, \dots, T_{i-1}\Phi_{\text{aut},L} \leq 1_{\text{v}}$ are redundant with respect to $A_{\mathcal{S}} x \leq 1_{\text{v}}$:
  - Set $T_i = \Gamma[T_{i-1}\Phi_{\text{aut},1}; \dots; T_{i-1}\Phi_{\text{aut},L}]$, with $I \leq \Gamma \leq (1+\gamma)I$. The coefficients of Γ are calculated

such that **a)** each coefficient that is strictly larger than 1 results in at least 1 more redundant constraint in the next step and **b)** $\max_j \|(T_i)_{[j,:]}\| \leq d_1 \cdot d_2^i$.
- Remove all constraints from T_i that are redundant with respect to $[A_{\mathcal{S}}; T_i]$.
- Set $A_{\mathcal{S}} = [A_{\mathcal{S}}; T_i]$.
- Remove all redundant constraints from $A_{\mathcal{S}}$.

Return $\mathcal{S} = \{x | A_{\mathcal{S}} x \leq 1_{\mathrm{v}}\}$.

Convergence of the algorithm and invariance of the resulting set are proven in [19]. The proofs are omitted here due to space constraints. The main achievement of the algorithm is that the typically exponential number of constraints is reduced to a linear number of constraints as a function of the state dimension of the system. Figure 1 illustrates this significant decrease by means of a numerical example.

V. EFFICIENT ROBUST MPC ALGORITHM

The two ingredients described in the previous two sections can now be combined in the following algorithm :

Algorithm 3 (Efficient Robust MPC Algorithm): Consider a system (1) subject to constraints (2a)-(2b) and a feedback controller $u_k = -Kx_k$ resulting in a closed-loop system that is quadratically stable. Consider state and input weighting matrices $Q = Q^{\mathrm{T}} > 0 \in \mathbb{R}^{n_x \times n_x}, R = R^{\mathrm{T}} > 0 \in \mathbb{R}^{n_u \times n_u}$ defining control optimality.

Off-line :

- Choose $n_p \in \{1, \ldots, n_u\}$ and $E \in \mathbb{R}^{n_u \times n_p}$ (e.g. $n_p = n_u, E = I$ if all inputs should be optimized independently or $n_p < n_u, E = [I; 0]$ if additional d.o.f. are only necessary for the first n_p inputs).
- Construct an objective function $z^{\mathrm{T}} P z$ by solving the optimization problem (9).
- Construct an invariant set $\mathcal{S} = \{z | A_{\mathcal{S}} z \leq 1_{\mathrm{v}}\}$ for the augmented system (7) subject to constraints (8) using Algorithm 2.

On-line : At each time instant, given the current state x_k solve the following QP

$$\mathbf{c}_k^{\mathrm{o}} = \underset{\mathbf{c}_k}{\operatorname{argmin}} \quad z_k^{\mathrm{T}} P z_k, \tag{10a}$$

subject to

$$A_{\mathcal{S}} z_k \leq 1_{\mathrm{v}}, \tag{10b}$$

with $\mathbf{c}_k = [c_k; \ldots; c_{k+n_c-1}]$, $\mathbf{c}_k^{\mathrm{o}} = [c_k^{\mathrm{o}}; \ldots; c_{k+n_c-1}^{\mathrm{o}}]$ and apply $u_k = -Kx_k + Ec_k^{\mathrm{o}}$ to the system.

Lemma 1 (stability): If algorithm 3 is feasible at time $k = 0$ then it robustly asymptotically stabilizes the system (1) without violating constraints (2a)-(2b).

Proof : First we prove that the algorithm is recursively feasible, which means that, given a feasible solution c_k^{o} to (10) at time k, it is possible to construct a feasible solution c_{k+1}^{f} to (10) at time $k+1$. For the state $x_{k+1} = (A(k) - B(k)K)x_k + B(k)Ec_k^{\mathrm{o}}$ at time $k+1$ it is possible to construct a feasible solution to (10) as $c_{k+1}^{\mathrm{f}} = [c_{k+1}^{\mathrm{o}}; \ldots; c_{k+n_c-1}^{\mathrm{o}}; 0]$. It can algebraically be verified that

$$z_{k+1}^{\mathrm{f}} = \Phi_{\mathrm{aug}}(k) z_k^{\mathrm{o}}, \tag{11}$$

with $z_{k+1}^{\mathrm{f}} = [x_{k+1}; c_{k+1}^{\mathrm{f}}]$ and $z_k^{\mathrm{o}} = [x_k; c_k^0]$. Since $\mathcal{S}$ is robust invariant with respect to (7), it is clear that since c_k^{o} is feasible at time k, c_{k+1}^{f} is also feasible at time $k+1$. Furthermore, due to the construction of (8), it can be verified that, if $z_k \in \mathcal{S}$, state and input constraints (2a)-(2b) are satisfied as well. This shows that Algorithm 3 guarantees robust constraint satisfaction. Furthermore, defining $V_k^{\mathrm{o}} = z_k^{\mathrm{o},\mathrm{T}} P z_k^{\mathrm{o}}$ and $V_{k+1}^{\mathrm{f}} = z_{k+1}^{\mathrm{f},\mathrm{T}} P z_{k+1}^{\mathrm{f}}$, it is straightforward to verify that $V_{k+1}^{\mathrm{f}} < V_k^{\mathrm{o}}$ due to the fact that P satisfies (9b). As a result it is also guaranteed that $V_{k+1}^{\mathrm{o}} < V_k^{\mathrm{o}}$, which shows that the algorithm robustly quadratically stabilizes (1). □

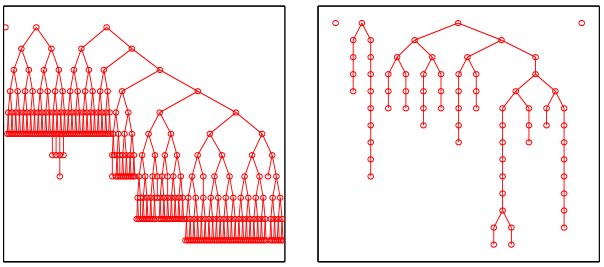


Fig. 1. Depiction of the tree structure of polyhedral invariant sets for (7)-(8) with system dynamics $A_1 = [1\ 0.1; 0\ 1], A_2 = [1\ 0.2; 0\ 1], B_1 = [0\ 1], B_2 = [0\ 1.5]$, constraints $A_x = [0.2I; -0.1I], A_u = [1; -2]$, feedback gain $K = [0.5\ 0.3]$ and horizon length $N = 5$. Each node represents a constraint. **Left :** the MAS constructed using Algorithm 1 (306 constraints). **Right :** reduced complexity invariant set constructed using Algorithm 2 with parameters $d_1 = 10 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|, \gamma = 0.2$, (71 constraints).

VI. EXAMPLE

A. Problem formulation

In this section the model of a continuously stirred tank reactor (CSTR) for a copolymerization process is considered [5]. The system consists of a tank to which different reagents are continuously fed, while the reaction product are simultaneously drained from the tank. The inputs of the system are the feed rates of the 6 different reagents (monomer A, monomer B, initiator, inhibitor, ...), while the outputs are the average molar mass of the polymer, the average mass fraction of monomer A in the polymer, the polymer production rate and the reactor temperature. The system is described by a nonlinear discrete-time model with 12 states.

The system is operated at the working point described in [15]. In order to obtain a realistic but still transparant example, two linearized models $[A1\ B1], [A2\ B2]$ at two different operating points (90% of the nominal monomer A concentration and 110% of the nominal monomer A concentration) are combined into an LPV model which is then used for both the controller design and the simulations. The model is shifted and normalized with respect to the steady state values. Control performance is measured as

n_c	# states in (7)	# constr. (Alg. 2)	# constr. (Alg. 1)
0	12	58	74
5	22	177	771
10	32	299	>2000
15	42	417	/
20	52	538	/
25	62	642	/

TABLE I

DIMENSIONALITY OF THE AUGMENTED SYSTEM (7) AND THE NUMBER OF CONSTRAINTS INVARIANT SETS FOR (7) FOR DIFFERENT PREDICTION HORIZONS n_c.

$Q = 10 \cdot C^{\mathrm{T}}C, R = 0.001I$. The inputs are constrained to the interval $[-0.2, 0.1]$, the states are constrained to the interval $[-1, 1]$ and the outputs are constrained to the interval $[-0.5, 0.5]$.

The case-study discussed in this section is that of stabilizing the system to the steady state after an initial disturbance in the monomer A or monomer B concentration in the reactor.

B. Controller design

A local feedback controller $u_k = -Kx_k$ is designed as the LQR-optimal for the given weights for a model linearized around the steady-state, which is also verified to be quadratically stabilizing for the LPV system. The feasible region for this linear feedback controller is shown in Figure 2. It is clear that a linear feedback controller has a very restricted feasibility region so adding additional degrees of freedom to the controller is preferable, i.e using Algorithm 3 with $n_c > 0$. The feed rates of monomer A and monomer B are the inputs that have the largest impact on the corresponding tank concentrations and therefore $n_p = 2$ and $E = [I; 0]$ are chosen. Invariant sets for the augmented system (7) with

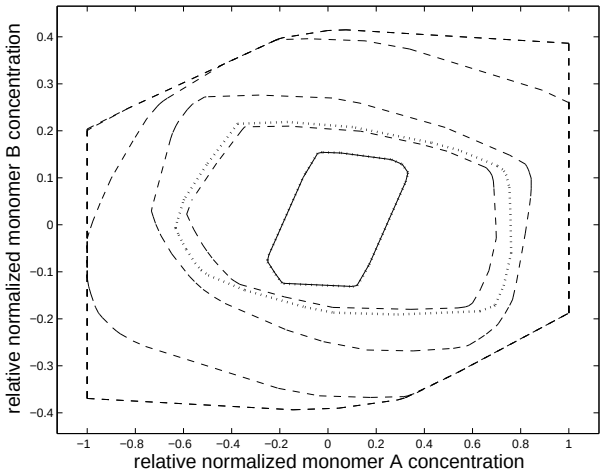


Fig. 2. Projections onto the two relevant state dimensions of the feasible regions for $n_c = 0$ (solid) and $n_c = 5, \ldots, 25$ (dashed) corresponding to invariant sets for (7) constructed using Algorithm 2. Feasible regions corresponding to invariant sets for (7) using Algorithm 1 for $n_c = 0, 5$ are also depicted (dotted). The two different feasible regions for $n_c = 0$ are only marginally different, as well as the two feasible regions for $n_c = 20, 25$ using Algorithm 2.

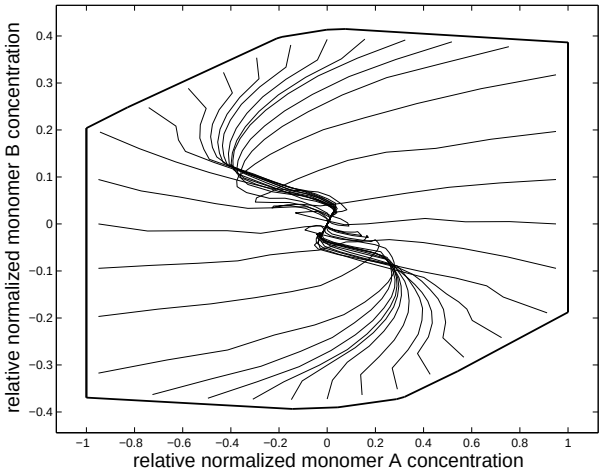


Fig. 3. State trajectories (projected onto the first two state dimensions) resulting from Algorithm 3 starting from initial states near the feasibility boundary.

$n_c = 5, 10, \ldots, 25$ are computed using Algorithm 2 with parameters $d_1 = 10 \max_i \|(A_x)_{[i,:]}\|, d_2 = \max_j \|\Phi_j\|, \gamma = 0.15$.

C. Results

1) **Feasible region:** The feasible regions for $n_c = 0, 5, \ldots, 25$ are depicted in Figure 2. It is clear that increasing n_c significantly improves the feasible region of the algorithm. Using reduced-complexity invariant sets instead of the MAS slightly reduces the size of the feasible region, as can also be seen in Figure 2 for $n_c = 5$, but this is more than compensated for by the ability to use larger n_c. Table I shows that using the MAS with $n_c = 5$ results in a higher complexity invariant set than using a reduced-complexity invariant set with $n_c = 25$, while the latter leads to a significantly larger feasible region. Figure 5 shows the feasible region when using ellipsoidal invariant sets, which

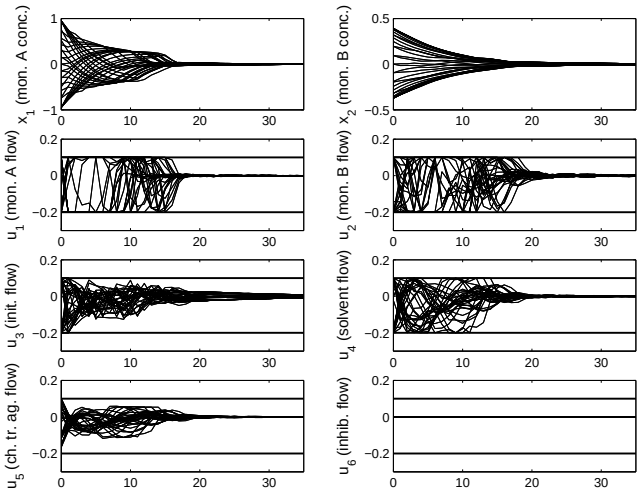


Fig. 4. State (first 2 states only) and input sequences corresponding to the trajectories depicted in Fig. 3.

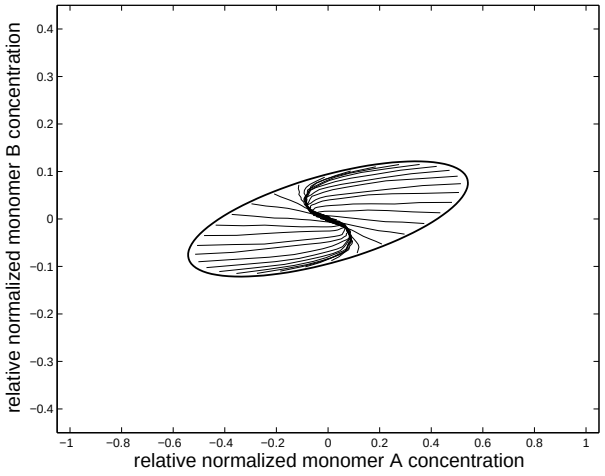


Fig. 5. State trajectories (projected onto the first two state dimensions) resulting from Algorithm 3 when applying ellipsoidal invariant sets (similar to [9]) instead of reduced-complexity polyhedral sets, with initial states near the feasibility boundary.

is obviously significantly smaller, indicating the ellipsoidal invariant sets become increasingly conservative when moving to larger state dimensions and longer prediction horizons.

2) **Control performance:** Trajectories for initial states near the border of the feasibility region corresponding to $n_c = 25$ are depicted in Figures 3 and 5 for reduced-complexity polyhedral invariant sets and ellipsoidal invariant sets respectively indicating that both variants are robustly stabilizing. $[A(k)\ B(k)]$ where chosen time-varying within Ω. Figures 4 and 6 show the corresponding state and input sequences. It is clear that Algorithm 3 (i.e. using reduced-complexity invariant sets) has non-conservative constraint handling and efficiently drives the two relevant states to their steady state values. Figure 5 shows that the corresponding algorithm using ellipsoidal invariant sets never actually reaches the input constraints, although the initial states were chosen

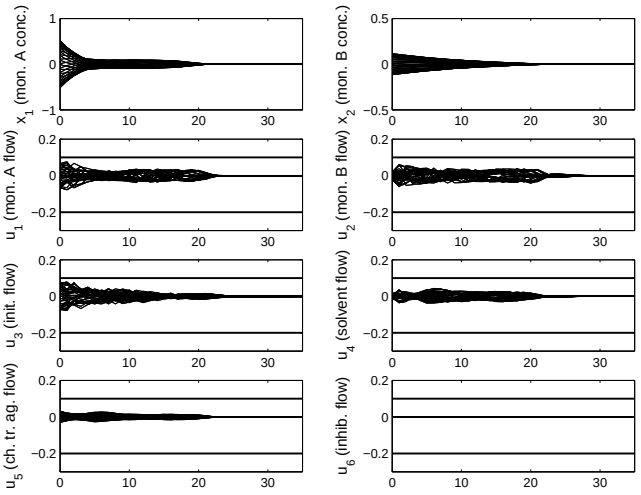


Fig. 6. State (first 2 states only) and input sequences corresponding to the trajectories depicted in Fig. 5.

near the border of the feasible region. As a consequence the settling-time of the closed loop system is slightly larger than when using polyhedral invariant sets, although the initial states lie closer to the origin.

3) **Computational complexity:** This case study confirms that the number of constraints describing a reduced-complexity invariant set typically increases as a linear function of the state dimension (in this case the state dimension of the augmented system (7)). Therefore the number of constraints involved in the on-line optimization increases as a linear function of the horizon length, similar to nominal linear MPC. Furthermore, the on-line optimization consists of solving a QP, whereas the corresponding algorithm using ellipsoidal invariant sets involves solving an SDP (or an SOCP after reformulation). The number of optimization variables is identical for both algorithms. In this case study the typical computation time per iteration was $<$ 1s when using reduced-complexity invariant sets and $\sim$ 10s when using ellipsoidal invariant sets.

The off-line computation time was $\sim$ 3.5h for the polyhedral case and $\sim$ 1h for the ellipsoidal case.

It should be noted that computation times are strongly dependent on the actual implementation of the optimization algorithm used. Also, numerical precision becomes a critical issue for larger-dimensional systems and longer prediction horizons, both in the on-line as in the off-line calculations. SeDuMi [1] was found to be a very reliable and reasonably efficient solver in both respects.

VII. CONCLUSION

This paper discusses a new, efficient robust MPC algorithm based on two key ingredients : within-horizon feedback with a reduced number of parameters and reduced complexity polyhedral invariant sets. The resulting algorithm is based on QP, has non-conservative constraint handling, can be applied to significantl larger-dimensional systems than previously possible and can use significantly longer prediction horizons than previously possible. These claims are confirmed by a case study based on the model of a real-life system. Extensions to include robustness with respect to bounded disturbances are straightforward.

Future research aims to further explore this research direction and to provide upper bounds on the off-line and on-line computational complexity.

Acknowledgments. Research supported by KUL: GOA-Mefisto 666, GOA-AmbioRics; FWO: G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0800.01, G.0452.04, G.0499.04, G.0211.05, G.0080.01, G.0226.06, research communities (ICCoS, ANMMM); IWT: PhD Grants, BFSPO: IUAP P5/22; PODO-II (CP/40: TMS and Sustainability); EU: FP5-CAGE; FP5-Quprodis; ERNSI; FP6-BioPattern; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB; Bert Pluymers is a research assistant with the I.W.T. at the Katholieke Universiteit Leuven, Belgium. Dr. Johan Suykens is an associate professor and Dr. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium.

REFERENCES

[1] Sedumi 1.1 MATLAB SDP optimization toolbox. http://sedumi.mcmaster.ca/.

[2] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Publications, 1994.

[4] E.F. Camacho and C. Bordóns. *Model Predictive Control in the Process Industry*. Springer-Verlag, 1995.

[5] J. P. Congalidis, J. R. Richards, and W. H. Ray. Feedforward and feedback controller of a solution copolymerization reactor. *AIChE Journal*, 35 (6), 1989.

[6] G. De Nicolao, L. Magni, and R. Scattolini. On the robustness of receding horizon control with terminal constraints. *IEEE Transactions on Automatic Control*, 41:451–453, 1996.

[7] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints : The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

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

[9] B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

[10] Y. I. Lee and B. Kouvaritakis. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 36:1497–1504, 2000.

[11] L. Magni and R. Sepulchre. Stability margins of nonlinear receding-horizon control via inverse optimality. *Systems & Control Letters*, 32(4):241–245, 1997.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[13] D.Q. Mayne, M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41:219–224, 2005.

[14] B. Pluymers, L. Roobrouck, J. Buijs, J. A. K. Suykens, and B. De Moor. Model-predictive control with time-varying terminal cost using convex combinations. *Automatica*, 41:831–837, 2005.

[15] B. Pluymers, L. Roobrouck, J. Buijs, J. A. K. Suykens, and B. De Moor. Model-predictive control with time-varying terminal cost using convex combinations. *Internal Report 04-028, ESAT-SISTA, K. U. Leuven (Leuven, Belgium)*, 2005, (http://www.esat.kuleuven.be/˜sistawww/cgi-bin/pub.pl).

[16] B. Pluymers, J.A. Rossiter, J.A.K. Suykens, and B. De Moor. A simple algorithm for robust MPC. In *Proceedings of the IFAC World Congress 2005, Prague, Czech Republic*, 2005.

[17] B. Pluymers, J.A. Rossiter, J.A.K. Suykens, and B. De Moor. Efficient computation of polyhedral invariant sets for LPV systems and application to robust MPC. *Submitted for publication*, 2005, (http://www.esat.kuleuven.be/˜sistawww/cgi-bin/pub.pl).

[18] B. Pluymers, J.A.K. Suykens, and B. De Moor. Robust finite-horizon MPC using optimal worst-case closed-loop predictions. In *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[19] B. Pluymers, J.A.K. Suykens, and B. De Moor. Construction of reduced complexity polyhedral invariant sets for LPV systems using linear programming. *Submitted for publication*, 2005, (http://www.esat.kuleuven.be/˜sistawww/cgi-bin/pub.pl).

[20] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. *AIChE Symposium Series 316*, 93:232–256, 1996.

[21] J.A. Rossiter. *Model Based Predictive Control*. CRC Press, 2003.

[22] J.A. Rossiter, B. Kouvaritakis, and M.J. Rice. A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 34:65–73, 1998.

[23] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[24] Zheng Z.Q. and Morari M. Robust stability of constrained model predictive control. In *Proceedings of the American Control Conference*, pages 379–383, San Francisco, 1993.