Linear Control of Time-Domain Constrained Systems

W.H.T.M. Aangenent, W.P.M.H. Heemels, M.J.G. van de Molengraft, Didier Henrion, Maarten Steinbuch

Linear Control of Time-Domain Constrained Systems

W.H.T.M. Aangenent[1], W.P.M.H. Heemels[1], M.J.G. van de Molengraft[1], D. Henrion[2,3,4], M. Steinbuch[1]

October 25, 2011

Abstract

This paper presents a general framework for the design of linear controllers for linear systems subject to time-domain constraints. The design framework exploits sums-of-squares techniques to incorporate the time-domain constraints on closed-loop signals and leads to conditions in terms of linear matrix inequalities (LMIs). This control design framework offers, in addition to constraint satisfaction, also the possibility of including an optimization objective that can be used to minimize steady state (tracking) errors, to decrease the settling time, to reduce overshoot and so on. The effectiveness of the framework is shown via a numerical example.

Keywords: Linear Systems; Constrained systems; Polynomial methods; LMIs; Motion systems.

1 Introduction

The transient response to reference commands or disturbance inputs is an important performance qualifier in many control systems. Unfortunately, most control design strategies cannot cope directly with requirements on time-domain signals such as actuator amplitude or rate limits, no output signal overshoot or undershoot, trajectory planning constraints and so on. Especially in the continuous-time case, there are hardly any systematic controller design methods to enforce time-domain constraints on e.g. tracking errors and control inputs.

In the discrete-time case, model predictive control (MPC) (see e.g. the surveys [11,28,31]) is a widely used technique to cope with constraints on inputs and states. In MPC a control

[1] Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[2] CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse; France

[3] Université de Toulouse; UPS, INSA, INP, ISAE; UT1, UTM, LAAS; F-31077 Toulouse; France

[4] Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, CZ-16626 Prague, Czech Republic

action is prescribed that is obtained by solving a finite or infinite horizon optimization problem that can incorporate input, state and output constraints in a direct manner. A drawback of predictive control concepts and online optimization-based methods in general is that they require a high computational effort with the consequence that they cannot be implemented on fast motion systems where high sampling rates are required, typically in the order of several kHz. Explicit MPC [4–6, 37] might offer an appealing solution as it precomputes a piecewise affine state feedback for discrete-time systems off-line. Still, the explicit control law often leads to a complex description consisting of many affine feedbacks, which also cannot realize the high sampling rates typically needed for motion systems of considerable size, although recent research is focussed on decreasing the implementation complexity of MPC, see for instance [15,16,20,23,26] and the references therein. An alternative approach with strong ties to MPC is based on so-called reference governors, see e.g. [3,12] and the references therein. A reference governor is a nonlinear device that is added to a primal controller, which functions well in the absence of constraints. The reference governor modifies the reference signal supplied to the primal controller in order to enforce the input and state constraints. This approach suffers from the mentioned drawbacks in MPC to some extent as well, but has the advantage that the reference modifications are often needed at a lower sampling frequency than the updates of the primal control loop. A major difference with the method presented in this paper is that the overall control systems in case of reference governors become nonlinear devices modifying the supplied reference, while the method in this paper aims at designing *linear* controllers that satisfy the time-domain constraints without any modification of the references or disturbances.

Besides these predictive control methods, that are typically suited for a discrete-time context, there are only a few methods available in the literature that can directly synthesize controllers incorporating time-domain constraints in the continuous-time setting. For instance in the case of input constraints, [13, 17] consider the linear quadratic regulator problem with positivity constraints on the input, while various control problems with amplitude and rate constraints on the input signal are solved in [33]. The latter line of work has also been extended to stabilization and output regulation problems with amplitude and rate constraints on certain output variables, see, e.g., [32]. Other methods exist that actually allow the control output to saturate such as, for instance, the usage of anti-windup schemes [10, 35, 36] or LQR/LQG control methods [7, 14]. These methods, however, do not enforce constraint satisfaction but rather guarantee stability or recover performance despite the saturation nonlinearity in the loop. The above mentioned techniques cannot handle *time-varying* constraints, and, except for [32], state or output constraints are not considered either. In addition, all the above mentioned techniques result in general in nonlinear controllers.

As already briefly mentioned, in this paper the objective is to derive a design method for *linear* controllers that incorporate possibly *time-varying* time-domain constraints on all closed-loop signals (inputs, states and outputs). Within this context, a commonly used method to capture the essence of time-domain specifications is the reformulation into frequency domain requirements [8]. Unfortunately, such reformulations are in general either approximate, conservative or both.

A methodology to enforce time-domain constraints on the input and output of a continuous-time linear control system is presented recently in [19], where linear matrix inequality (LMI) techniques are used to synthesize a fixed order linear controller that satisfies the constraints. This is done in a polynomial setting in the sense that a controller is designed according to the well-known pole placement method using the Diophantine equation. This method allows the design of a controller that results in a closed-loop transfer function with prescribed pole locations, either exact, or within an admissible region of the complex plane. In [19], all controllers with the prescribed pole locations are characterized using the Youla-Kučera parametrization [21]. Next the degrees of freedom of the Youla-Kučera parametrization are used to enforce certain time-domain constraints, such as bounds on the input amplitude and output overshoot, exploiting sums-of-squares techniques. Unfortunately, the approach in [19] is limited to the assignment of distinct strictly negative *real* closed-loop poles, which is a severe restriction in the case of many practical situations such as, for instance, lightly damped systems. As a consequence, there is a strong need for a general framework encompassing arbitrary closed-loop pole placement. The development of such a framework is the main purpose of this paper.

In particular, we propose an extension to the method in [19], which leads to a general design framework based on sums-of-squares LMI techniques and we show indeed that the resulting linear controller satisfies the time-domain constraints on closed-loop signals, even when complex conjugate poles are assigned. This framework is based on two relaxations. One of these relaxations, of which a preliminary version was presented by the authors in [1], can solve the constrained control problem at hand with arbitrary accuracy and still lead to LMIs. In addition to constraint satisfaction, we will also include an objective function in the convex programming problem that can be used to minimize steady state (tracking) errors, to decrease the settling time, to reduce the overshoot and so on. As a consequence, the ideas presented in this paper will lead to a general design framework for optimized linear controllers with guarantees regarding constraint satisfaction.

The organization of the paper is as follows. The proposed methodology from [19] is briefly reviewed in Section 2. The extension to complex conjugate poles is treated in Section 3, which includes the main results. Section 4 discusses the proposed control design method, and Section 5 provides an illustrative example. Finally, the conclusions are stated in Section 6.

2 Methodology involving real poles

In [19] a method is presented to incorporate time-domain constraints on input and output signals of a linear system. It is shown that finding a controller of fixed order that satisfies these constraints boils down to solving a set of LMIs. In this section, we shortly review this procedure for completeness and self-containedness.

2.1 Youla-Kučera parametrization

Consider the control system depicted in Fig. 1 with a linear single-input-single-output plant P given by the strictly proper transfer function

$$P(s) = \frac{b(s)}{a(s)}, \tag{1}$$

where $a(s)$ and $b(s)$ are polynomials in the Laplace variable s. The controller C, which is to be designed, is described accordingly by

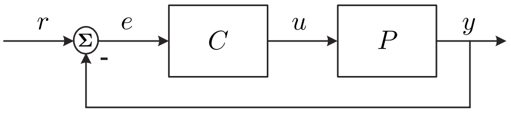

Figure 1: Block diagram of the closed-loop system with controller C, plant P, and reference signal r, control output signal u, and output signal y.

$$C(s) = \frac{d(s)}{c(s)}, \tag{2}$$

resulting in the complementary sensitivity given by

$$T(s) = \frac{y(s)}{r(s)} = \frac{b(s)d(s)}{a(s)c(s) + b(s)d(s)}. \tag{3}$$

If $a(s)$ and $b(s)$ are coprime (i.e., their greatest common divisor is 1), then arbitrary pole placement can be achieved by designing the corresponding controller polynomials. This is done by solving the polynomial Diophantine equation

$$a(s)c(s) + b(s)d(s) = z(s), \tag{4}$$

where $z(s) = (s + p_1)(s + p_2) \dots (s + p_n)$ is the polynomial with given roots $-p_1, \dots, -p_n$, which are the desired poles of the closed-loop system. There are infinitely many solutions $(c(s), d(s))$ to (4), but there is a unique solution pair $(c_0(s), d_0(s))$ such that $\deg d_0(s) < \deg a(s)$. In this case we have that $d_0(s)$ is of minimal degree and as such, $(c_0(s), d_0(s))$ is called the d-minimal solution pair. All possible solutions to the Diophantine equation can then be written as

$$\begin{aligned} c(s) &= c_0(s) + b(s)q(s), \\ d(s) &= d_0(s) - a(s)q(s), \end{aligned} \tag{5}$$

where $q(s)$ is an arbitrary polynomial such that $c_0(s) + b(s)q(s)$ is non-zero. This polynomial, called the Youla-Kučera parameter [9], creates extra freedom in the design of the controller. While the closed-loop poles are invariant for any choice of the Youla-Kučera parameter, the Youla-Kučera parameter enables placement of closed-loop zeros to alter the response. Only *proper* controllers are considered and therefore there is a degree constraint on $q(s)$. Since the plant was assumed to be strictly proper, and under the additional assumption that $\deg z(s) \geq 2 \deg a(s) - 1$ (to enable arbitrary pole placement with proper controllers), this constraint is given as in [22] by

$$\deg q(s) \leq \deg z(s) - 2 \deg a(s). \tag{6}$$

The extra freedom in the control design parameterized by $q(s)$ satisfying (6) can now be used to satisfy additional time-domain constraints as will be explained in the next section.

2.2 A positive polynomial formulation of time-domain constraints

We will explain the procedure in [19] using the typical example of constraints on the step response. Hence, we consider the response y to a step input ($r(s) = \frac{1}{s}$). The Laplace transform of the closed-loop system's output (assuming zero initial conditions) is then given by

$$y(s) = \frac{1}{s}\frac{b(s)d(s)}{z(s)} = \frac{1}{s}\frac{b(s)d_0(s)}{z(s)} - \frac{1}{s}\frac{a(s)b(s)}{z(s)}q(s). \tag{7}$$

At this point of the control design a *restrictive assumption* was made [19], namely

Assumption 2.1 *All the assigned poles $-p_1, \ldots, -p_n$ are distinct strictly negative rational numbers.*

Using this assumption and $z(s) = \prod_{i=1}^{n}(s+p_i)$ the partial fractional decomposition of (7) leads to

$$y(s,q) = \sum_{i=0}^{n} \frac{y_i(q)}{s+p_i}, \tag{8}$$

where $p_0 = 0$ and $y_i(q)$, $i = 1, \ldots, n$ are appropriate coefficients following from the decomposition, which are influenced by the choice of the design parameter $q(s) = \sum_{i=0}^{d_q} q_i s^i$. The coefficients $y_i(q)$ depend in an *affine* manner on the parameter $q = (q_0, q_1, \ldots, q_{d_q})$ in the sense that there exist matrices $A \in \mathbb{R}^{(n+1)\times(n+1)}$, $B \in \mathbb{R}^{(n+1)\times(q_d)}$ and a vector $b \in \mathbb{R}^{n+1}$ such that

$$A \begin{pmatrix} y_0(q) \\ y_1(q) \\ \vdots \\ y_{n+1}(q) \end{pmatrix} = Bq^T + b. \tag{9}$$

This follows directly by comparing (7) and (8), and equating the coefficients of the powers of s in the resulting numerator polynomials (see also (46) below for an example). The corresponding time-domain signal is given by

$$y(t,q) = \sum_{i=0}^{n} y_i(q) e^{-p_i t}. \tag{10}$$

Let $p_i = \frac{n_i}{d_i}$ be the ratios of the integers n_i and d_i, and let m denote the smallest positive number[1] of the denominators such that $p_i = \frac{\bar{p}_i}{m}$ for some positive integers $\bar{p}_i$, $i = 0, 1, \ldots, n$.

[1] In principle m can be chosen to be any positive number that results in integer values of $\bar{p}_i$. However, it turns out that by choosing m as the smallest possible positive number the order of the resulting polynomial optimization problem is the lowest.

This means that the time-domain output signal at time $t \in \mathbb{R}_+ := [0, \infty)$ can now be expressed as the polynomial

$$y(\lambda, q) = \sum_{i=0}^{n} y_i(q)\lambda^{\bar{p}_i} \tag{11}$$

in the indeterminate $\lambda = e^{-t/m}$. Obviously, λ lies in the interval $[0, 1]$ as $t \in \mathbb{R}_+$. Suppose that the output $y(t, q)$ of the system needs to be bounded according to

$$y_{\min} \leq y(t, q) \leq y_{\max} \quad \forall\, t \in \mathbb{R}_+. \tag{12}$$

Formulation (12) is equivalent to enforcing the polynomial bound constraints

$$\begin{cases} P_1(q, \lambda) := y(\lambda, q) - y_{\min} \geq 0 \\ P_2(q, \lambda) := y_{\max} - y(\lambda, q) \geq 0 \end{cases} \quad \forall\, \lambda \in [0, 1], \tag{13}$$

where P_1 and P_2 are polynomials in both λ and q. This problem is a special case of the following more general problem of minimizing a polynomial with polynomial constraints over a basic semialgebraic set.

Definition 2.1 *A set $\mathcal{D}$ is called a basic semialgebraic set if it can be described as*

$$\begin{aligned} \mathcal{D} = \{x \in \mathbb{R}^n \mid e_i(x) \geq 0,\ i = 1, \ldots, M_e \text{ and} \\ f_j(x) = 0,\ j = 1, \ldots, M_f\} \end{aligned} \tag{14}$$

for certain polynomials $e_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, M_e$ and $f_j : \mathbb{R}^{n_x} \to \mathbb{R}$, $j = 1, \ldots, M_f$.

Problem 2.1 (Polynomial optimization problem) *Consider two variables $z \in \mathbb{R}^{n_z}$ and $x \in \mathbb{R}^{n_x}$ and let polynomials $g_i : \mathbb{R}^{n_z} \times \mathbb{R}^{n_x} \to \mathbb{R}$, $i = 1, \ldots, M_g$, and $p : \mathbb{R}^{n_z} \to \mathbb{R}$ be given. Moreover, let a collection of basic semialgebraic sets $\mathcal{D}_l \subseteq \mathbb{R}^{n_x}$, $l = 0, \ldots, N$ be given. A (robust) polynomial optimization problem according to this data is given by*

$$\begin{aligned} &\min_{z} \quad p(z) \\ &s.t. \quad g_i(z, x) \geq 0, \quad i = 1, \ldots, M_g \quad \forall\, x \in \bigcup_{l=0}^{N} \mathcal{D}_l. \end{aligned} \tag{15}$$

Indeed, (13) can now be written in the form of Problem 2.1 by taking $z = q$, $x = \lambda$, $M_g = 2$, $N = 0$, $p(z) = 0$, $g_1(z, x) = P_1(q, \lambda)$, $g_2(z, x) = P_2(q, \lambda)$, and $\mathcal{D}_0 = \{\lambda \in \mathbb{R} \mid 0 \leq \lambda \leq 1\}$. Although the bounds $y_{\min}$ and $y_{\max}$ in (13) are chosen to be constants for illustrating purposes, they can also be selected as polynomials in λ, i.e., in the form $y_{\min}(\lambda)$ and $y_{\max}(\lambda)$ without any complications. In this case the bounds in (12) become *time-varying.* Univariate positive polynomial constraints (meaning polynomials in only one variable), such as (13) with $\lambda \in \mathcal{D}_0 = [0, 1] \subseteq \mathbb{R}$, can be transformed into LMI conditions, see [19] for the details. Once we transformed the design problem into a polynomial optimization problem as formulated in Problem 2.1, there are appropriate tools available for solving the problem. Therefore, we restrict ourselves to the transformation of the constrained control problems at hand into manifestations of Problem 2.1.

The approach discussed in this section is not restricted to bounding only the output of a system. Indeed, by using the appropriate transfer functions, any signal in the loop can be constrained. The control output u, for example, can be bounded using

$$u(s) = r(s)\frac{a(s)d_0(s)}{z(s)} - r(s)\frac{a^2(s)}{z(s)}q(s) \tag{16}$$

in addition to, or instead of (7). Also, other Laplace transformable inputs or disturbances can be used as long as the poles of the Laplace transform of the corresponding signal are distinct strictly negative rational numbers and differ from the closed-loop poles p_i. In case the disturbance signal is a (filtered) random process, the method cannot be applied as is. However, a possible extension can be to bound the infinity- or 2-norm of the suitably weighted process sensitivity (or other relevant transfer functions) via e.g.

$$||V(s)\frac{b(s)c(s)}{a(s)c(s)+b(s)d(s)}W(S)||_{2/\infty} \leq 1 \tag{17}$$

This way the knowledge of stochastic disturbances can be used to shape the relevant sensitivity functions to achieve desired disturbance reduction.

A combination of requirements on different reference signals can easily be handled at the cost of increasing the size of the set of LMIs. As for LMIs there are efficient solvers available, e.g. [34], transforming the problem at hand into Problem 2.1 provides an effective solution. The problem derived in this section was only a feasibility problem (the cost criterion $p(z)$ was chosen to be 0). In Section 4 below, we will also provide relevant choices for the cost criterion that next to satisfaction of the time-domain constraints also provides additional desirable properties of the constructed controller.

3 Problem formulation: the complex poles case

The polynomial representation (10), as derived in [19], of the time response of a linear system to a Laplace transformable input is unfortunately only possible when strictly negative rational closed-loop poles are assigned (see Assumption 2.1). However, in many cases the assignment of purely real poles can be undesirable, especially in lightly damped systems such as most motion systems. Furthermore, many reference signals have Laplace transforms with complex poles. If, for instance, a sinusoid is used as the reference signal instead of a step, the Laplace transform is given by $r(s) = \frac{\omega}{s^2+\omega^2}$ resulting in complex poles in the system's response. Therefore, such reference signals cannot be handled by the approach from [19]. The main objective of this paper is to present a solution to the linear control design problem for time-domain constrained systems of which the Laplace transforms of the closed-loop responses may contain complex roots.

When we allow both distinct real and complex poles to be present in the closed-loop transfer function $T(s)$ and/or the Laplace transform of the reference signal $r(s)$, the Laplace transform of the system's output can be decomposed as the partial fractional decomposition

$$\begin{aligned} y(s) = & \sum_{i=0}^{n_r} \frac{y_i}{s+p_i} \\ & + \sum_{i=n_r+1}^{n_r+n_c/2+1} \frac{y_i}{s+\alpha_i+j\beta_i} + \frac{y_i^*}{s+\alpha_i-j\beta_i}, \end{aligned} \tag{18}$$

where n_r and n_c denote the number of real and complex poles, respectively, $-p_i$, $i = 1, \ldots, n_r$ are the locations of the real poles, $-\alpha_i \pm j\beta_i$, $i = n_r + n_c/2 + 1$ are the locations of complex conjugate pairs of poles, and y_i are the possibly complex coefficients (with complex conjugate y_i^*) that affinely depend on the design parameter q, see (9) (we omitted this dependence on q for ease of exposition). To enforce stability, we again assume that the assigned closed-loop poles have strictly negative real part. The corresponding time-domain signal is then described by

$$y(t) = \sum_{i=0}^{n_r} y_i e^{-p_i t} \quad + \sum_{i=n_r+1}^{n_r+n_c/2+1} (y_i e^{-j\beta_i t} + y_i^* e^{j\beta_i t}) e^{-\alpha_i t}. \tag{19}$$

As before, we use the following assumption

Assumption 3.1 *p_i, α_i, and β_i are rational numbers.*

We denote $p_i = \frac{\bar{p}_i}{m}$, $\alpha_i = \frac{\bar{\alpha}_i}{m}$, $\tau = \frac{t}{m}$ where m is the smallest positive number (not necessarily an integer) of p_i and α_i such that $\bar{p}_i$ and $\bar{\alpha}_i$ can be taken as integers. We denote $\beta_i = \frac{\theta\bar{\beta}_i}{m}$ for a number θ (not necessarily an integer) such that $\bar{\beta}_i$ can be taken as integer as well. For guidelines how to choose θ, see Remark 3.1 below. Furthermore, let

$$\lambda = e^{-\tau}. \tag{20}$$

Using Euler's formula $e^{j\phi} = \cos(\phi) + j\sin(\phi)$ and decomposing the complex coefficients as $y_i = a_i + jb_i$, $y_i^* = a_i - jb_i$, yields

$$\begin{aligned} y(t) = & \sum_{i=0}^{n_r} y_i \lambda^{\overline{p}_i} \\ & + \sum_{i=n_r+1}^{n_r+n_c/2+1} \left(a_i 2\cos(\overline{\beta}_i\theta\tau) + b_i 2\sin(\overline{\beta}_i\theta\tau)\right) \lambda^{\overline{\alpha}_i}. \end{aligned} \tag{21}$$

Obviously, the terms involving the complex poles are non-polynomial in the indeterminate λ because of the presence of $\cos(\overline{\beta}_i\theta\tau)$ and $\sin(\overline{\beta}_i\theta\tau)$, which make it impossible to directly use the positive polynomial approach in Section 2 to bound the output as in (12). Although the parameters $\overline{\alpha}_i$, $\overline{\beta}_i$ and $\overline{p}_i$ are fixed as a result of the pole placement (and the choice of m and θ), there still is freedom in the choice for the coefficients a_i, b_i, which depend on the coefficients $q = (q_0, \ldots, q_{d_q})$ in the Youla-Kučera parameter $q(s)$. We propose two relaxations to determine the values y_i, a_i, b_i via polynomial optimization problems to shape the time response $y(t)$, thereby overcoming the limitations in [19].

The first approach is based on an exponential bound relaxation that results in univariate polynomials. This method has the advantage that it results in a simple polynomial optimization problem, but introduces some conservatism. The second method proposes a multivariate polynomial relaxation that leads to polynomial problems as in Problem 2.1, while the conservatism can be made arbitrarily small. Both methods result in polynomial optimization problems of the type as in Problem 2.1 that can be solved using LMIs.

3.1 Exponential bounds relaxation

To resolve the problem induced by the presence of the products $\cos(\overline{\beta}_i\theta\tau)\lambda^{\overline{\alpha}_i}$ and $\sin(\overline{\beta}_i\theta\tau)\lambda^{\overline{\alpha}_i}$ in (21) we relax the problem by using the fact that

$$\cos(\overline{\beta}_i\theta\tau),\ \sin(\overline{\beta}_i\theta\tau) \in [-1,1] \quad \forall\tau\in\mathbb{R}, \tag{22}$$

and instead of the exact time-response (21), we consider

$$\begin{aligned} y_{\text{upper}}(\lambda) &= \sum_{i=0}^{n_r} y_i\lambda^{\overline{p}_i} + \sum_{i=n_r+1}^{n_r+n_c/2+1} \left(2|a_i| + 2|b_i|\right)\lambda^{\overline{\alpha}_i},\\ y_{\text{lower}}(\lambda) &= \sum_{i=0}^{n_r} y_i\lambda^{\overline{p}_i} - \sum_{i=n_r+1}^{n_r+n_c/2+1} \left(2|a_i| + 2|b_i|\right)\lambda^{\overline{\alpha}_i}. \end{aligned} \tag{23}$$

In contrast to (21), these exponential bounds on the closed-loop time response are univariate polynomials in the indeterminate $\lambda = e^{-\tau}$ (if q is fixed) and can be bounded by specified polynomials $g_u(\lambda)$ and $g_l(\lambda)$ via the polynomial non-negativity constraints

$$\begin{array}{ll} P_3(q,\lambda) := g_u(\lambda) - y_{\text{upper}}(\lambda) & \geq 0 \\ P_4(q,\lambda) := y_{\text{lower}}(\lambda) - g_l(\lambda) & \geq 0 \end{array} \quad \forall\lambda\in[0,1], \tag{24}$$

where we included the explicit dependence of y_i, a_i and b_i on q again. The constraints (24) cannot straightforwardly be cast in the form Problem 2.1 because of the nonlinear operator $|\cdot|$, which is present in these equations. However, each of the two nonlinear inequality constraints P_3 and P_4 can be expressed as 2^{n_c+1} equivalent polynomial inequality constraints $\tilde{P}_3$ and $\tilde{P}_4$ (2 inequalities for each absolute value expression). Enforcing non-negativity of (24) on the interval $\lambda\in[0,1]$ is then again a special case of Problem 2.1 with $z=q$, $x=\lambda$, $M_g=2$, $M_h=0$, $N=0$, $p(z,x)=0$, $g_1(z,x)=\tilde{P}_3(q,\lambda)$, $g_2(z,x)=\tilde{P}_4(q,\lambda)$, and $\mathcal{D}_0 = \{\lambda\in\mathbb{R} \mid 0\leq\lambda\leq1\}$. Therefore, it is possible to determine the values $q = (q_0,\ldots,q_{d_q})$ such that the upper and lower bounds (23) of the closed-loop time response are bounded by $g_u(\lambda)$ (e.g. $g_u(\lambda) = y_{max}$) and $g_l(\lambda)$ (e.g. $g_u(\lambda) = y_{min}$) via a polynomial optimization problem. The exponential bounds relaxation does introduce some conservatism by using relaxation (23) instead of the exact time-response (21). The second method presented next offers the possibility to render this conservatism arbitrary small. In other words, the second method can approximate the original time-domain constraints with arbitrary accuracy and still lead to polynomial optimization problems.

3.2 Multivariate polynomial relaxation

The time response (21) is equivalent to

$$\begin{aligned} y(t) = \sum_{i=0}^{n_r} y_i \lambda^{\overline{p}_i} + \\ \sum_{i=n_r+1}^{n_r+n_c/2+1} \Big[(a_i + j b_i) \big(\cos(\overline{\beta}_i \theta\tau) - j \sin(\overline{\beta}_i \theta\tau)\big) \\ + (a_i - j b_i) \big(\cos(\overline{\beta}_i \theta\tau) + j \sin(\overline{\beta}_i \theta\tau)\big) \Big] \lambda^{\overline{\alpha}_i}. \end{aligned} \tag{25}$$

De Moivre's formula, which is closely related to Euler's formula and $(e^{j\phi})^n = e^{jn\phi}$, states that for any $\phi \in \mathbb{R}$ and any integer $n \in \mathbb{Z}$

$$(\cos(\phi) + j \sin(\phi))^n = \cos(n\phi) + j \sin(n\phi), \tag{26}$$

and hence (25) is equal to

$$\begin{aligned} y(t) = \sum_{i=0}^{n_r} y_i \lambda^{\overline{p}_i} + \\ \sum_{i=n_r+1}^{n_r+n_c/2+1} \Big((a_i + j b_i) \left[\cos(\theta\tau) - j \sin(\theta\tau)\right]^{\overline{\beta}_i} + \\ (a_i - j b_i) \left[\cos(\theta\tau) + j \sin(\theta\tau)\right]^{\overline{\beta}_i} \Big) \lambda^{\overline{\alpha}_i}. \end{aligned} \tag{27}$$

Appropriate polynomial functions $w_i : \mathbb{R}^2 \to \mathbb{R}$ and $r_i : \mathbb{R}^2 \to \mathbb{R}$, $i = n_r + 1, \ldots, n_r + n_c/2 + 1$ in two variables can now be defined such that (27), and thus the time response (21), can be written as

$$\begin{aligned} y(t) = \sum_{i=0}^{n_r} y_i \lambda^{\overline{p}_i} + \\ \sum_{i=n_r+1}^{n_r+n_c/2+1} (a_i 2 w_i(\cos(\theta\tau), \sin(\theta\tau)) \\ + b_i 2 r_i(\cos(\theta\tau), \sin(\theta\tau))) \lambda^{\overline{\alpha}_i}. \end{aligned} \tag{28}$$

This proves the following theorem.

Theorem 3.1 *Consider the closed-loop system* (3) *and let* y *be the response to a reference input* r *and assume that the Laplace transform* $y(s)$ *of* y *has only distinct poles such that* (18) *and Assumption 3.1 hold. Then we have that*

$$\{y(t) \mid t \in \mathbb{R}^+\} = \{y(u, v, \lambda) \mid (u, v, \lambda) \in \mathcal{F}_{original}\}, \tag{29}$$

where $y(u, v, \lambda)$ *is given by the multivariate polynomial*

$$\begin{aligned} y(u, v, \lambda) = \sum_{i=0}^{n_r} y_i \lambda^{\overline{p}_i} \\ + \sum_{i=n_r+1}^{n_r+n_c/2+1} (a_i 2 w_i(u, v) + b_i 2 r_i(u, v)) \lambda^{\overline{\alpha}_i} \end{aligned} \tag{30}$$

with $w_i : \mathbb{R}^2 \to \mathbb{R}$ *and* $r_i : \mathbb{R}^2 \to \mathbb{R}$*,* $i = n_r + 1, \ldots, n_r + n_c/2 + 1$ *polynomials as in* (28) *and*

$$\begin{aligned} \mathcal{F}_{original} := \big\{ (u, v, \lambda) \in \mathbb{R}^3 \mid u = \cos(\theta\tau),\ v = \sin(\theta\tau), \\ \lambda = e^{-\tau} \textit{ for some } \tau \in \mathbb{R}^+ \big\}. \end{aligned} \tag{31}$$

Proof: The reasoning before the formulation of the theorem revealed that $y(t)$ under the given assumptions is equal to (19), which can equivalently be written as (28), where $\lambda = e^{-\tau}$ and $\tau = \frac{t}{m}$. ■

Due to Theorem 3.1, bounding the output as in (12) to the interval $[y_{\min}, y_{\max}]$ is equivalent to enforcing the polynomial non-negativity constraints

$$\begin{aligned} P_5(q, u, v, \lambda) &:= y(u, v, \lambda) - y_{\min} \geq 0, \\ P_6(q, u, v, \lambda) &:= y_{\max} - y(u, v, \lambda) \geq 0 \end{aligned} \tag{32}$$

for all $(u, v, \lambda) \in \mathcal{F}_{\text{original}}$. Recall that $y(u, v, \lambda)$ depends on q via y_i, a_i and b_i. As we mentioned before, it is of interest to transform the linear constrained control problem into Problem 2.1. The conditions (32) are not in this form due to the fact that $\mathcal{F}_{\text{original}}$ is *not* a (finite union of) basic semialgebraic set(s) as in Definition 2.1. However, this set can be overapproximated by a finite union of basic semialgebraic sets in an arbitrarily close manner.

Definition 3.1 *We call a set* $\mathcal{F}_{approx}$ *an* ε-close overapproximation *of* $\mathcal{F}_{original}$ *for some* $\varepsilon > 0$*, if it satisfies the following three properties:*

1. $\mathcal{F}_{approx} = \bigcup_{l=0}^{N} \mathcal{F}_l$ *for a finite collection of basic semialgebraic sets* $\mathcal{F}_0, \ldots, \mathcal{F}_N$*;*
2. $\mathcal{F}_{original} \subseteq \mathcal{F}_{approx}$*;*
3. $\mathcal{F}_{approx} \subseteq \mathcal{F}_{original} + \mathbb{B}_\varepsilon$*, where* $\mathbb{B}_\varepsilon := \{(0, 0, z) \mid -\varepsilon \leq z \leq \varepsilon$*.*

Hence, an ε-close overapproximation of $\mathcal{F}_{\text{original}}$ contains the set $\mathcal{F}_{\text{original}}$ as drawn by the white line in Figure 2 (for $\theta = 1$), but it is ε-close in the sense of property 3. Hence, for small $\varepsilon > 0$, replacing $\mathcal{F}_{\text{original}}$ by $\mathcal{F}_{\text{approx}}$ only results in small errors and all guarantees on $\mathcal{F}_{\text{approx}}$ also apply to $\mathcal{F}_{\text{original}}$ due to property 2. Moreover, due to property 1 an ε-close overapproximation $\mathcal{F}_{\text{approx}}$ of $\mathcal{F}_{\text{original}}$ can be used to embed the polynomial constraints in (32) for all $(u, v, \lambda) \in \mathcal{F}_{\text{original}}$ into a version of the constraints in Problem 2.1, where $\mathcal{D}_l = \mathcal{F}_l,\ l = 0, \ldots, N$.

The following algorithm provides an algorithm that constructs for each desirable level of approximation ε an ε-close overapproximation of $\mathcal{F}_{\text{original}}$. The basic idea of the algorithm is to overapproximate the $\mathcal{F}_{\text{original}}$-set by the union of basic semialgebraic sets $\mathcal{F}_l$, which are obtained by splitting the set $\mathcal{F}_{\text{original}}$ in the τ-direction by considering intervals $I_l := [\tau_l, \tau_{l+1})$, $l = 0, \ldots, N$, where $0 = \tau_0 < \tau_1 < \ldots < \tau_{N+1} = \infty$. On each of these subintervals I_l we approximate $e^{-\tau}$ by $\psi_l(\cos(\theta\tau), \sin(\theta\tau))$ using Fourier series, where $\psi_l : \mathbb{R}^2 \to \mathbb{R}$ is a polynomial such that $|e^{-\tau} - \psi_l(\cos(\theta\tau), \sin(\theta\tau))| \leq \varepsilon$ for all $\tau \in I_l$. Next to ε, the algorithm uses another parameter $0 < T < \frac{2\pi}{\theta}$, which indicates the desired length of the intervals I_l, $l = 0, \ldots, N-1$ (although it will be modified such that all intervals have the same length).

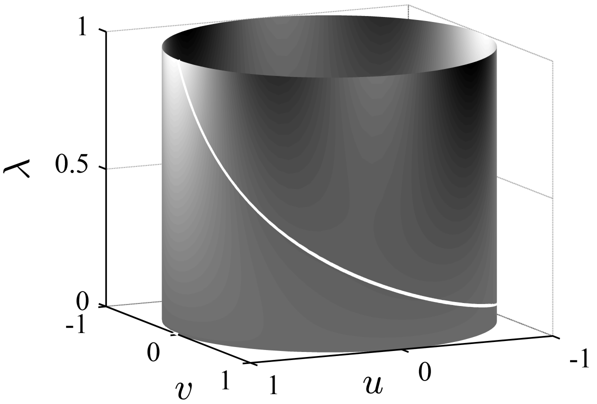

Figure 2: $\mathcal{F}_{\text{original}}$ (white line) drawn inside the cylinder given by $u^2+v^2=1$ and $0 \le \lambda \le 1$.

Algorithm 1 *Let $0<\varepsilon<1$ and $0<T<\frac{2\pi}{\theta}$ be given.*

Step 1: *Define $N:=\lceil\frac{-\ln\varepsilon}{T}\rceil$ and $\tau_N:=-\ln\varepsilon$ and $\tau_{N+1}:=\infty$.*

$$\mathcal{F}_N := \{(u,v,\lambda)\in\mathbb{R}^3 \mid u^2+v^2=1 \text{ and } 0\le\lambda\le\varepsilon\}. \tag{33}$$

Step 2: *Divide the remaining interval $[0,\tau_N)$ in N subintervals of length $\bar{T}:=\frac{\tau_N}{N}\le T<\frac{2\pi}{\theta}$. $I_l:=[\tau_l,\tau_{l+1})$ with $\tau_l=l\bar{T}$, $l=0,\dots,N-1$.*

Step 3: *For each $l=0,\dots,N-1$ define a function $\phi_l:\mathbb{R}\to\mathbb{R}$ that satisfies:*

- *ϕ_l is at least continuously differentiable, but preferably m times continuously differentiable (C^m) for $m\in\mathbb{N}$ large;*
- *ϕ_l is periodic with period $\frac{2\pi}{\theta}$;*
- *$\phi_l(\tau)=e^{-\tau}$ for all $\tau\in I_l$.*

Step 4: *For each $l=0,\dots,N-1$ compute the Fourier series approximation of ϕ_l of sufficiently high degree K_l such that*

$$|\phi_l(\tau)-\sum_{k=0}^{K_l}[a_k\cos(k\theta\tau)+b_k\sin(k\theta\tau)]|\le\varepsilon \text{ for all } \tau\in I_l, \tag{34}$$

where a_k,b_k, $k=0,\dots,K_l$ are the Fourier coefficients of ϕ_l.

Step 5: *For each $l=0,\dots,N-1$ use De Moivre's formula to rewrite $\sum_{k=0}^{K_l}[a_k\cos(k\theta\tau)+b_k\sin(k\theta\tau)]$ obtained in the previous step as*

$$\sum_{k=0}^{K_l}\sum_{i=0}^{k}c_{ki}(\cos(\theta\tau))^k(\sin(\theta\tau))^l =: \psi_l(\cos(\theta\tau),\sin(\theta\tau)),$$

where $\psi_l:\mathbb{R}^2\to\mathbb{R}$ is a polynomial of degree equal to the degree of the Fourier series.

Step 6: *For each $l=0,\dots,N-1$, define*

$$\begin{aligned}\mathcal{F}_l := \{(u,v,\lambda)\in\mathbb{R}^3 \mid -\varepsilon \le \lambda - \psi_l(u,v) \le \varepsilon \wedge \\ u^2+v^2=1 \wedge (S_l-S_{l+1})u + (C_{l+1}-C_l)v + \\ S_{l+1}C_l - C_{l+1}S_l \le 0\},\end{aligned} \quad (35)$$

where $C_l := \cos(\theta\tau_l)$, $S_l := \sin(\theta\tau_l)$.

Step 7: *Take $\mathcal{F}_{approx} = \bigcup_{l=0}^{N} \mathcal{F}_l$.*

Theorem 3.2 *For each $0 < \varepsilon < 1$ and $0 < T < \frac{2\pi}{\theta}$ Algorithm 1 produces an ε-close overapproximation $\mathcal{F}_{approx}$ of $\mathcal{F}_{original}$ in the sense of Definition 3.1.*

Proof: First of all, we write $\mathcal{F}_{\text{original}}$ as $\bigcup_{l=0}^{N} \mathcal{F}_{\text{original},l}$ with $\mathcal{F}_{\text{original},l} := \{(\cos\theta\tau, \sin\theta\tau, e^{-\tau}) \mid \tau \in I_l\}$ for $l = 0,1,\dots,N$ as $\bigcup_{l=0}^{N} I_l = [0,\infty)$. Step 1 considers the interval $I_N := [\tau_N, \tau_{N+1}) = [-\ln\varepsilon, \infty)$ for which it holds that $0 \le e^{-\tau} \le \varepsilon$. Hence, clearly $\mathcal{F}_{\text{original},N} \subseteq \mathcal{F}_N$ and $\mathcal{F}_N \subseteq \mathcal{F}_{\text{original},N} + \mathbb{B}_\varepsilon$. The construction of functions ϕ_l in step 3 is possible as $e^{-\tau}$ is continuous and the fact that $0 < \bar{T} \le T < \frac{2\pi}{\theta}$. Hence, step 3 can always be taken, while the function ϕ_l can still be made $\frac{2\pi}{\theta}$-periodic and continuously differentiable. Step 4 can be realized, because the Fourier series converges uniformly to a continuously differentiable periodic function, see, e.g., [2, 29]. Therefore, uniform convergence proves the existence of a finite K_l such that (34) holds. Note now that for $(u,v,\lambda) \in \mathcal{F}_{\text{original},l}$ it holds that $u^2+v^2=1$. Obviously, $(\cos(\theta\tau), \sin(\theta\tau))$ for $\tau \in I_l = [\tau_l, \tau_{l+1})$ lies in one of the half spaces generated by the straight line in $\mathbb{R}^2$ through the points $(\cos(\theta\tau_l), \sin(\theta\tau_l))$ and $(\cos(\theta\tau_{l+1}), \sin(\theta\tau_{l+1})$ given by

$$\begin{aligned}[\sin(\theta\tau_l) - \sin(\theta\tau_{l+1})]\underbrace{\cos(\theta\tau)}_{u} + [\cos(\theta\tau_{l+1}) - \cos(\theta\tau_l)]\underbrace{\sin(\theta\tau)}_{v} \\ + \sin(\theta\tau_{l+1})\cos(\theta\tau_l) - \cos(\theta\tau_{l+1})\sin(\theta\tau_l) = 0,\end{aligned} \quad (36)$$

see Figure 3. In particular, for all $(u,v,\lambda) \in \mathcal{F}_{\text{original},l}$ it holds that

$$(S_l - S_{l+1})u + (C_{l+1} - C_l)v + S_{l+1}C_l - C_{l+1}S_l \le 0 \quad (37)$$

where $C_l := \cos(\theta\tau_l)$, $S_l := \sin(\theta\tau_l)$. Due to (34), step 5, and using the above observations, it holds that $\mathcal{F}_{\text{original},l} \subseteq \mathcal{F}_l$. Moreover, similar reasoning using (34) shows that $\mathcal{F}_l \subseteq \mathcal{F}_{\text{original},l} + \mathbb{B}_\varepsilon$. Hence, by taking $\mathcal{F}_{\text{approx}}$ as the union of the resulting sets, i.e. $\mathcal{F}_{\text{approx}} = \bigcup_{l=0}^{N} \mathcal{F}_l$, an ε-close overapproximation $\mathcal{F}_{\text{approx}}$ of $\mathcal{F}_{\text{original}}$ is obtained. This completes the proof. ■

For instance, for $\theta=1, \varepsilon=e^{-1.5\pi}\approx 0.009$ and $T = \bar{T} = 0.75\pi$ the ε-close overapproximation $\mathcal{F}_{\text{approx}}$ of $\mathcal{F}_{\text{original}}$ can be generated with $\tau_0 = 0$, $\tau_1 = 0.75\pi$, $\tau_2 = 1.5\pi$ and $\tau_3 = \infty$ and polynomials

$$\begin{aligned}\psi_0(u,v) &= 0.398u - 0.971v + 0.616u^2 - 0.192uv + 1.179v^2 \\ &\quad - 0.015u^3 + 0.184u^2v, \\ \psi_1(u,v) &= 0.033u + 0.096v + 0.0760u^2 + 0.0534uv + 0.094v^2 \\ &\quad + 0.013uv^2 - 0.011v^3.\end{aligned} \quad (38)$$

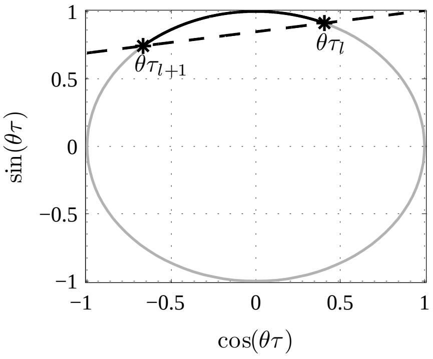

Figure 3: Top-view of the cylinder $\{(u,v,\lambda)\in\mathbb{R}^3 \mid u^2+v^2=1, 0\leq\lambda\leq 1\}$ (solid grey), the hyperplane $\{(u,v,\lambda)\in\mathbb{R}^3 \mid (36)\}$ (dashed), and the resulting part of the cylinder given by $\{(u,v,\lambda)\in\mathbb{R}^3 \mid u^2+v^2=1, (37), 0\leq\lambda\leq 1\}$ (solid black).

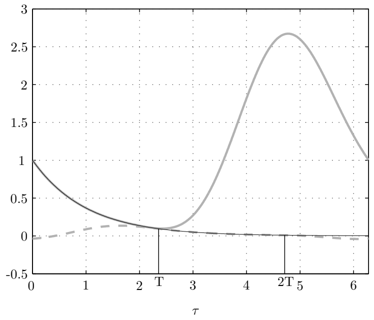

Figure 4: Functions $e^{-\tau}$ (solid black), ψ_0 (solid grey), and ψ_1 (dashed grey).

This overapproximation and the polynomials are illustrated in Figure 4. If one is satisfied with an overapproximation accuracy of $\varepsilon = e^{-1.5\pi} \approx 0.009$, then one can use this precomputed overapproximation (for the case $\theta = 1$). If it is desired to have a simpler overapproximation (with less regions and polynomials of lower degree) or an even tighter approximation with $\varepsilon < 0.009$, one can run Algorithm 1 to obtain it.

Remark 3.1 *A few comments are in order:*

- *The reason to take the functions ϕ_l in step 3 as smooth as possible (m in C^m as large as possible) is that a lower degree K_l is needed to satisfy* (34)*. Indeed, when $\phi_l \in C^m$ then the Fourier coefficients satisfy $k^m a_k \to 0$, $k^m b_k \to 0$ when $k\to\infty$ and thus the approximation error $|\sum_{k=0}^{K_l}[a_k\cos(k\theta\tau)+b_k\sin(k\theta\tau)]-e^{-\tau}| = |\sum_{k=K_l}^{\infty}[a_k\cos(k\theta\tau)+b_k\sin(k\theta\tau)]|$ on I_l is smaller than ε for smaller values of K_l.*
- *A sufficiently high degree of K_l such that* (34) *holds can be obtained by increasing K_l incrementally until* (34) *is satisfied. If one is satisfied with an overapproximation accuracy of $\varepsilon = e^{-1.5\pi}$, then the precomputed overapproximation* (38) *with $N = 2$ and $K_l = 3$ can be used in case $\theta=1$.*

- *Instead of selecting T a priori, we can also select a maximal degree K of the approximation functions ψ_l in the sense that $K_l \leq K$ for all $l = 0, 1, \dots, N-1$. Instead of increasing the degrees of the approximation functions ψ_l, one now can split the time interval $[0, -\ln\varepsilon)$ into smaller pieces until $|\psi_l(\cos\theta\tau, \sin\theta\tau) - e^{-\tau}| \leq \varepsilon$ for all $\tau \in I_l$ is satisfied for the fixed (low) degree K. This might lead to more regions (a larger N). This indicates that there is a trade-off between N (number of basic semialgebraic sets in the overapproximation $\mathcal{F}_{approx}$) and K (the maximal degree of the Fourier series approximation). The smaller N the higher K and vice versa. However, note that the example of the overapproximation with $N = 2$ and $K = 3$ already provides a very tight approximation of $\varepsilon = e^{-1.5\pi} \approx 0.009$ in case $\theta = 1$.*

- *The reason for splitting the set $\mathcal{F}_{original}$ in the $\tau = \frac{t}{m}$ direction is that the exponential function $\lambda = e^{-\tau}$ can generally not be ε-close approximated with the basis functions $\cos(\theta\tau)$ and $\sin(\theta\tau)$ in the interval $[0, -\ln\varepsilon]$ (unless ε is chosen relatively large or θ is very small and thus T can be selected larger). The additional scaling θ can be used to adjust the period of $\cos(\theta\tau)$ and $\sin(\theta\tau)$ thereby offering a trade-off between the required amount of intervals N and the order of the polynomial $y(u, v, \lambda)$ (provided the integer requirement on $\overline{\beta}_i$ is fulfilled). Indeed, $N = 1$ in Definition 3.1 can be obtained by choosing θ sufficiently small such that $e^{-\tau}$ can be ε-close approximated using $u = \cos(\theta\tau)$ and $v = \sin(\theta\tau)$ in the interval $[0, -\ln\varepsilon]$ for any ε. However, this implies that the polynomial order $\overline{\beta}_i$ (see (27)) increases due to the relation $\beta_i = \frac{\theta\bar{\beta}_i}{m}$, which is undesirable from a computational point of view. On the other hand, if θ is chosen as large as possible (while respecting the integer requirement on $\overline{\beta}_i$), the order of the polynomial $y(u, v, \lambda)$ in (27) is minimal but a large amount of regions N could be required.*

Using the ε-close overapproximation $\mathcal{F}_{\text{approx}}$, the conditions (32) again are a special case of Problem 2.1 with $z = q$, $x = (u, v, \lambda)$, $\ell = 2$, $m = 0$, $p(z, x) = 0$, $g_1(z, x) = P_5(q, u, v, \lambda)$, $g_2(z, x) = P_6(q, u, v, \lambda)$, and $\mathcal{D}_l = \mathcal{F}_l$, $l = 0, \dots, N$. However, since the polynomial positivity constraints (32) are now multivariate (meaning polynomials in more than one variable) instead of univariate, the equivalent LMI expression from [19] is not applicable. In general, checking positivity of a multivariate polynomial on a basic semialgebraic set is a hard problem, but it can often be approximated as closely as desired by a hierarchy of convex relaxations [24,25]. In this case, to make Problem 2.1 computationally tractable, the inequality conditions from (15) are replaced by stronger conditions in terms of primal moment and dual sums-of-squares (SOS) problems to formulate a hierarchy of upper bounds on the minimum in Problem 2.1 that converge in the limit to the real minimum. We refer the reader to [24,25] for the conversion techniques to obtain the LMIs and further details. There exist software packages, such as GloptiPoly [18], SOSTOOLS [30] or YALMIP [27], that automatically build up a hierarchy of LMI relaxations, whose associated monotone sequence of optimal values converges to the global optimum. Numerical certificates of optimality are also available, in terms of ranks of embedded moment matrices, see [24,25].

4 Control design procedure

To summarize the previous design setup, suppose that the d-minimal controller (2) for system (1) has been designed via the Diophantine equation (4) such that the desired closed-loop pole locations are achieved. Also suppose that more closed-loop poles are assigned than twice the number of poles of the plant. Then, according to (6) there is additional control design freedom parameterized in the form of the Youla-Kučera parameter, which can be used to shape the closed-loop time response. Time-domain constraints can be imposed using one of the proposed relaxations in Section 3, which yield the polynomial constraints in (15). Next to constraint satisfaction, (15) allows also the minimization of an objective function $p(z)$. This minimization can be exploited to obtain additional desired properties of the response in terms of the design parameters q and thus y_i, a_i and b_i. Consider the step response for example, which can be written as (18) with $p_0 = 0$ and where y_0 is the steady-state solution. Desirable properties of the unit step response are, for instance, a zero steady-state error, a small settling-time and small overshoot. These properties can be accommodated in $p(z)$ as follows

Small steady-state error: Set $p(z) = (1 - y_0)^2$ to minimize the steady-state error.

Short settling-time: Set $p(z) = a_i^2 + b_i^2$, where index i corresponds to a slow mode in (21), to minimize the contribution of this mode, which improves the settling time. Alternatively, exponentially decreasing constraints can be specified that directly impose a certain desired settling behavior.

Overshoot minimization: In case one is interested in constraining the response by a fixed constant, e.g. constraining the overshoot, the exponential bounds relaxation is not suitable. This is due to the fact that the peak values of the systems response often occur in the time interval where the exponential upper and lower bounds are still very far from the actual signal. As opposed to the exponential bounds relaxation, the multivariate polynomial relaxation from Section 3.2 is typically suited to minimize the overshoot of a step response by constraining the response (30) as $y(u, v, \lambda) \leq \gamma$ and specify $p(z) = \gamma$ to minimize the overshoot.

Of course, one can combine the above objectives in $p(z)$ using suitable weighting factors. Furthermore, the proposed relaxations enable the incorporation of the extensions that were mentioned in Section 2.2 (e.g. related to responses to disturbances) in case the poles of the Laplace transform of the corresponding signal are complex. Setting up the optimization problem (15) by including the time-domain constraints on closed-loop signals using one of the proposed relaxations and defining the objective function $p(z)$ provides a systematic manner for obtaining linear controllers with desirable properties. This design framework will be illustrated in the next section.

5 Numerical example

We start with a simple simulation example to illustrate the efficiency of the proposed design method. Consider the simple model given by

$$P(s) = \frac{y(s)}{u(s)} = \frac{1}{s+1}. \tag{39}$$

The control objective is to let y track a step reference from 0 to 1 as close as possible. Moreover, the controller (2) will be designed such that the assigned complex closed-loop poles are $p_{1,2} = -1 \pm 2j$, $p_{3,4} = -2 \pm 4j$. This is done by solving the Diophantine equation (4) leading to the d-minimal controller

$$C(s) = \frac{d_0(s)}{c_0(s)} = \frac{68}{s^3 + 5s^2 + 28s + 32}, \tag{40}$$

resulting in the closed-loop system given by the complementary sensitivity function

$$T(s) = \frac{68}{s^4 + 6s^3 + 33s^2 + 60s + 100}. \tag{41}$$

Using the Youla-Kučera parameter $q(s)$ and realizing that according to (6) we have $\deg q(s) \leq 2$, i.e., $q(s) = q_0 + q_1 s + q_2 s^2$, the set of allowable controllers assigning the specified closed-loop poles is parameterized as

$$\begin{aligned} C(s) &= \frac{d(s)}{c(s)} = \frac{d_0(s) - a(s)q(s)}{c_0(s) + b(s)q(s)} \\ &= \frac{68 - (q_0 + (q_0 + q_1)s + (q_1 + q_2)s^2 + q_2 s^3)}{s^3 + 5s^2 + 28s + 32 + (q_0 + q_1 s + q_2 s^2)}, \end{aligned} \tag{42}$$

resulting in the set of closed-loop transfer functions

$$T(s) = \frac{68 - (q_0 + (q_0 + q_1)s + (q_1 + q_2)s^2 + q_2 s^3)}{s^4 + 6s^3 + 33s^2 + 60s + 100}. \tag{43}$$

The Laplace transform of the time response of (43) to a step input is then parameterized as

$$\begin{aligned} y(s) = \frac{1}{s}T(s) = \\ \frac{68 - (q_0 + (q_0 + q_1)s + (q_1 + q_2)s^2 + q_2 s^3)}{s(s + 1 + 2j)(s + 1 - 2j)(s + 2 + 4j)(s + 2 - 4j)}. \end{aligned} \tag{44}$$

The corresponding partial fractional decomposition is equal to

$$\begin{aligned} y(s) =& \frac{y_0}{s} + \frac{a_1 + jb_1}{s + 1 + 2j} + \frac{a_1 - jb_1}{s + 1 - 2j} + \frac{a_2 + jb_2}{s + 2 + 4j} \\ &+ \frac{a_2 - jb_2}{s + 2 - 4j} \end{aligned} \tag{45}$$

where y_0, a_1, b_1, a_2, b_2 can be solved from the linear system of equations

$$\begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 60 & 40 & 80 & 20 & 40 \\ 33 & 48 & 16 & 18 & 16 \\ 6 & 10 & 4 & 8 & 8 \\ 1 & 2 & 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} y_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} 68 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix}, \tag{46}$$

where q_0, q_1, q_2 are the free variables in the Youla-Kučera parameter to shape the time response. The goal is to determine values of y_0, a_1, b_1, a_2, b_2 (via q_0, q_1, q_2) such that the closed-loop time response to the step input has a favorable shape. The LMI problems were modeled with YALMIP [27] and solved with SeDuMi [34].

5.1 Using the exponential bounds relaxation

The exponential bounds on the step response of the closed-loop system (43) are given by

$$\begin{aligned} \bar{y}_{\text{upper}}(t) &= y_0 + (2|a_1| + 2|b_1|)e^{-t} + (2|a_2| + 2|b_2|)e^{-2t}, \\ \bar{y}_{\text{lower}}(t) &= y_0 - (2|a_1| + 2|b_1|)e^{-t} - (2|a_2| + 2|b_2|)e^{-2t}, \end{aligned} \tag{47}$$

where y_0, a_1, b_1, a_2, b_2 are related to q_0, q_1, q_2 via (46). The goal of this relaxation is to determine q_0, q_1, q_2 such that

$$\begin{aligned} P_3(\lambda) &= g_u(\lambda) - y_{\text{upper}}(\lambda) &\geq 0, \\ P_4(\lambda) &= y_{\text{lower}}(\lambda) - g_l(\lambda) &\geq 0, \end{aligned} \tag{48}$$

for appropriately chosen *time-varying* bounds related to $g_u(\lambda)$ and $g_l(\lambda)$. Note that $\lambda = e^{-t}$ and

$$\begin{aligned} y_{\text{upper}}(\lambda) &= y_0 + (2|a_1| + 2|b_1|)\lambda + (2|a_2| + 2|b_2|)\lambda^2, \\ y_{\text{lower}}(\lambda) &= y_0 - (2|a_1| + 2|b_1|)\lambda - (2|a_2| + 2|b_2|)\lambda^2. \end{aligned} \tag{49}$$

To define the time-varying bounds, we first consider suitable upper and lower bounds on the step response that correspond to the d-minimal controller (40) (with $q = 0$). These are given by

$$\begin{aligned} g_{u_{\text{original}}}(\lambda) &= 0.68 + 1.58\lambda + 0.38\lambda^2, \\ g_{l_{\text{original}}}(\lambda) &= 0.68 - 1.58\lambda - 0.38\lambda^2, \end{aligned} \tag{50}$$

respectively. Based on these bounds, we now define tighter bounds $g_u(\lambda)$ and $g_l(\lambda)$ that are specified to be

$$\begin{aligned} g_u(\lambda) &= g_{u_{\text{original}}}(\lambda) + c_u = 1.01 + 1.58\lambda + 0.38\lambda^2 \\ g_l(\lambda) &= g_{l_{\text{original}}}(\lambda) + c_l = 0.99 - 1.58\lambda - 0.38\lambda^2. \end{aligned} \tag{51}$$

where $c_u = 0.33$ and $c_l = 0.31$ to guarantee a small steady-state error (smaller than 0.01). The dominant term in the upper and lower bound in (49) corresponds to the slow mode e^{-t} with coefficient $(2|a_1| + 2|b_1|)$. To minimize the contribution of this term and to bring the steady-state error close to zero, we specify the objective function according to Section 4 as

$$p(q_0, q_1, q_2) = 10(1 - y_0)^2 + 2(a_1^2 + b_1^2), \tag{52}$$

where y_0, a_1 and b_1 depend on q_0, q_1, q_2 as in (46). This results in the optimization problem

$$\begin{array}{cl} \min\limits_{q_0,q_1,q_2} & p(q_0, q_1, q_2) \\ \text{s.t.} & (46) \\ & (48) \quad \forall\ \lambda \in [0, 1]. \end{array} \tag{53}$$

The Youla-Kučera parameter resulting from the minimization problem (53) is

$$q(s) = -32.0 - 23.0s - 3.0s^2, \tag{54}$$

which yields the controller and closed-loop

$$C(s) = \frac{3s^3 + 26s^2 + 55s + 100}{s^3 + 2s^2 + 5s}, \tag{55}$$

$$T(s) = \frac{3s^3 + 26s^2 + 55s + 100}{s^4 + 6s^3 + 33s^2 + 60s + 100}, \tag{56}$$

together with the new bounds

$$\begin{aligned} g_{u_{\text{new}}}(t) &= 1.00 + 1.25e^{-2t}, \\ g_{l_{\text{new}}}(t) &= 1.00 - 1.25^{-2t}. \end{aligned} \tag{57}$$

This shows that the steady-state error is zero and the contribution of the slow mode has been completely eliminated. The step responses together with their bounds of the original closed-loop system (41) and of the new, optimized closed-loop system (56) are depicted in Fig. 5(a), while the Bode diagrams of the original controller (40) and the new controller (55) are shown in Fig. 5(b). From Fig. 5(a) it is obvious that the step response of the designed closed-loop system satisfies the specified bounds and additionally results in zero steady-state tracking error. Upon examination of Fig. 5(b) this can be explained by the fact that controller (55) exhibits an overall higher gain and hence results in a higher bandwidth of the closed-loop system resulting in a faster response, while it also implements integrating action providing the steady-state accuracy. Although there is some conservatism introduced by the fact that upper and lower bounds are used, this example demonstrates that a significant increase of the performance can be obtained using this method.

Remark 5.1 *In this example the ability of the method to minimize the contribution of the slow mode using an additional objective function $p(z)$ in Problem 2.1 was demonstrated. Because of the specific objective function* (52) *the slow mode was completely cancelled within the controller. Therefore, for this specific example another way to arrive at controller* (55) *is to specify only two closed-loop poles $p_{3,4} = -2 \pm 4j$ (without $p_{1,2} = -1 \pm 2j$), resulting in $C(s) = \frac{3s+20}{s}$, which indeed is the minimal form of* (55). *Interestingly, the optimization problem results in this controller in an automated and systematic manner.*

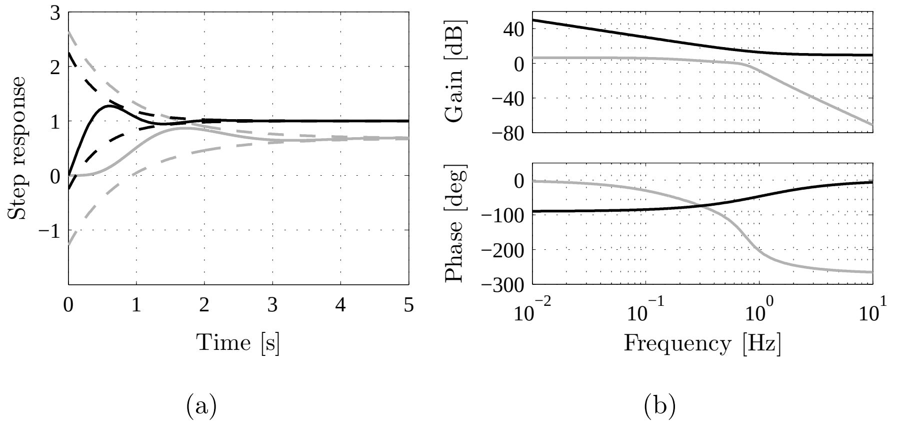

Figure 5: Results exponential bounds relaxation. (a) New (solid black) and original (solid grey) step responses, original bounds (dashed grey), and new bounds (dashed black), (b) Bode diagrams of the original (grey) and the new (black) controller.

5.2 Using the multivariate polynomial relaxation

In this section, the multivariate polynomial relaxation from Section 3.2 is used to obtain suitable values of q_0, q_1, q_2 to improve the step response of closed-loop system (43). We have that $m=1, n_r=0, n_c=2$, $\overline{\alpha}_1=-1$, $\overline{\alpha}_2=-2$, $\overline{\beta}_1=2$ and $\overline{\beta}_2=4$. Let $\theta=1$ such that $u=\cos(\tau)$ and $v=\sin(\tau)$ so that (27) yields

$$\begin{aligned} y(t) &= \left((a_1+jb_1)(u+jv)^2+(a_1-jb_1)(u+jv)^2\right)\lambda \\ &+ \left((a_2+jb_2)(u+jv)^4+(a_2-jb_2)(u+jv)^4\right)\lambda^2 \\ &= \left(2a_1\left(u^2-v^2\right)+2b_12uv\right)\lambda \\ &+ \left(2a_2\left(u^4+v^4-6u^2v^2\right)+2b_1\left(4vu^3-4uv^3\right)\right)\lambda^2. \end{aligned} \tag{58}$$

As a consequence, for this example we obtain

$$\begin{aligned} w_1(u,v) &= u^2-v^2, \quad r_1(u,v)=2uv \\ w_2(u,v) &= u^4+v^4-6u^2v^2, \quad r_2(u,v)=4vu^3-4uv^3. \end{aligned} \tag{59}$$

yielding the time response

$$\begin{aligned} y(u,v,\lambda) =& y_0+(2a_1(u^2-v^2)+4b_1uv)\lambda \\ &+(2a_2(u^4+v^4-6u^2v^2)+8b_2(vu^3-uv^3))\lambda^2, \end{aligned} \tag{60}$$

which is a multivariate polynomial with 3 independent variables (u,v,λ) and three decision variables (q_0,q_1,q_2). Note that $(u,v,\lambda)\in\mathcal{F}_{\text{original}}$. Since $\mathcal{F}_{\text{original}}$ is not the finite union of a basic semialgebraic set, we can use Algorithm 1 or use the precomputed overapproximation in Section 3.2 to obtain an ε-close overapproximation $\mathcal{F}_{\text{approx}}$ of $\mathcal{F}_{\text{original}}$. We use here the precomputed overapproximation $\mathcal{F}_{\text{approx}}$ with $\varepsilon=e^{-1.5\pi}\approx 0.009$. In accordance with Section 4, we formulate the problem as to find q such that the overshoot γ is small and that the steady-state error of the step response is minimized. Therefore, the problem is posed as

$$\begin{array}{cl} \min\limits_{q_0,q_1,q_2} & 10(1-y_0)^2+\gamma \\ \text{s.t.} & (46) \\ & \gamma - y(u,v,\lambda) \geq 0 \quad \forall(u,v,\lambda)\in\mathcal{F}_{\text{approx}}. \end{array} \tag{61}$$

Rewriting this optimization problem gives

$$\begin{array}{cl} \min\limits_{q_0,q_1,q_2} & 10(1-y_0)^2+\gamma \\ \text{s.t.} & (46) \\ & \gamma - y(u,v,\lambda) \geq 0 \quad \forall(u,v,\lambda)\in\mathcal{F}_0 \\ & \gamma - y(u,v,\lambda) \geq 0 \quad \forall(u,v,\lambda)\in\mathcal{F}_1 \\ & \gamma - y(u,v,\lambda) \geq 0 \quad \forall(u,v,\lambda)\in\mathcal{F}_2, \end{array} \tag{62}$$

with $\mathcal{F}_{0,1}$ as in (35) and $\mathcal{F}_2$ as in (33). This optimization problem is then solved with a hierarchy of LMI relaxations, as explained at the end of Section 3.2. The size of the resulting LMI problem depends on the order of the relaxation, and this can be used as a tuning knob to adjust the trade-off between the desired accuracy and the computational complexity, see Section 2. Although it is a priori not clear what is the order of the relaxation to arrive at the global minimum of γ, a heuristic method is to increase the order until not much improvement in the relaxed optimum $\tilde{\gamma}$ is observed anymore or until one is satisfied with the obtained value of $\tilde{\gamma}$. The obtained minimum values of $\tilde{\gamma}$ for various orders of relaxation are given in Table 1. We used this heuristic approach for illustration purposes only. It is recommended to use the more systematic approach that is implemented in GloptiPoly [18] to arrive at the global minimum, and certify it numerically. Based on the figures in this table we expect that the global minimum of γ is equal to 1.0718, representing an overshoot of 7.18%. The corresponding Youla-Kučera parameter is given by

	order of LMI relaxation					
	1	2	3	4	5	10
$\tilde{\gamma}$	297.170	1.235	1.235	1.0718	1.0718	1.0718

Table 1: Upper bound $\tilde{\gamma}$ for various orders of LMI relaxation.

$$q(s) = -32.0 - 17.0607s - 3.0227s^2, \tag{63}$$

which yields the controller

$$C(s) = \frac{3.0s^3+20.0s^2+49s+100}{s^3+2.0s^2+10.9s}. \tag{64}$$

The step responses of both the original closed loop with the d-minimal controller (40) and of the closed loop with controller (64) are depicted in Fig. 6, which shows a significant improvement as expected. The maximum of the step response $y(t)$ equals 1.0714 (i.e., 7.14% overshoot), which indeed is ε-close to $\tilde{\gamma}=1.0718$.

This example showed that after controller design by pole placement it is possible to shape the transient time response of the system by assigning zeros to the closed-loop system through a suitable extension of the controller under pole invariance.

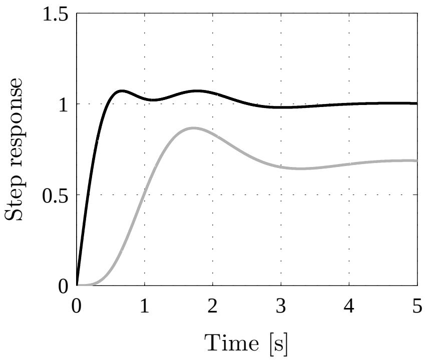

Figure 6: New (black) and original (grey) step responses.

6 Conclusions

In this paper we provided a generally applicable design framework to obtain linear controllers for linear systems subject to time-domain constraints. In order to arrive at this design framework we extended recent results in [19] that applied only in case of *real* closed-loop poles and external signals (e.g. references or disturbances) having Laplace transforms with *real* poles. The design method is based on synthesizing linear controller via a closed-loop pole placement method in which the additional design freedom in terms of the Youla-Kučera parameter is used to satisfy time-domain constraints on the closed-loop signals based on sum-of-square techniques. In order to extend the method to the practically relevant case where both inputs and closed-loop systems with complex conjugate poles are allowed, we proposed two relaxations.

The first relaxation, called the exponential bounds relaxation, exploited exponential upper and lower bounds on the response to any Laplace transformable input. Although this gives rise to potential conservatism, an example showed that by prescribing polynomial time-domain bounds, the system's performance to a step input can be improved with respect to the settling-time and the steady-state error. The second relaxation, the multivariate polynomial relaxation, removed the potential conservatism completely as we formally proved that it can approximate the original problem with arbitrary accuracy. Using these relaxations, we indicated how a polynomial optimization problem could be set up, in which next to constraint satisfaction, we could also optimize certain important closed-loop properties such as overshoot, settling-time and steady-state error. The resulting optimization problem can be solved using sum-of-squares and convex programming methods. As a consequence, the provided design framework is systematic in nature, as was also illustrated using a numerical example.

Ackwnoledgements

D. Henrion acknowledges support by project No. 103/10/0628 of the Grant Agency of the Czech Republic.

References

[1] W.H.T.M. Aangenent, W.P.M.H. Heemels, M.J.G. van de Molengraft, and M. Steinbuch. Linear control of time-domain constrained systems. In *Proceedings of the Conference on Decision and Control*, pages 5339–5344, Shanghai, China, 2009.

[2] N.H. Asmar. *Partial Differential Equations with Fourier Series and Boundary Value Problems.* 2nd edition, Pearson Prentice Hall, 2005.

[3] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Trans. Automat. Contr.*, 42(3):340–349, 1997.

[4] A. Bemporad, W.P.M.H. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. *IEEE Trans. Automat. Contr.*, 47(5):863–869, 2002.

[5] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[6] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences.* Springer-Verlag, 2003.

[7] S. Ching, P.T. Kabamba, and S.M. Meerkov. Simultaneous design of controllers and instrumentation: ILQR/ILQG. *IEEE Trans. Automat. Contr.*, 55:217–221, 2010.

[8] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback control theory.* MacMillan, New York, 1992.

[9] B.A. Francis. *A course in H-infinity control theory*, volume 88 of *Lecture Notes in Control and Information Science.* Springer, 1987.

[10] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback control of dynamic systems.* Prentice Hall, New Jersey, 2002.

[11] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.

[12] E. Gilbert and I. Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: A generalized reference governor. *Automatica*, 38:2063–2073, 2002.

[13] R. Goebel and M. Subbotin. Continuous time constrained linear quadratic regulator – convex duality approach. *IEEE Trans. Automat. Contr.*, 52(5):886–892, 2007.

[14] C. Gokcek, P.T. Kabamba, and S.M. Meerkov. An LQR/LQG theory for systems with saturating inputs. *IEEE Trans. Automat. Contr.*, 46(10):1529–1542, 2001.

[15] A. Grancharova, T.A. Johansen, and P. Tøndel. Computational aspects of approximate explicit model predictive control. In *International workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Freudenstadt-Lauterbad, Germany, 2005.

[16] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari. Stabilizing low complexity feedback control of constrained piecewise affine system. *Automatica*, 41(10):1683–1694, 2005.

[17] W.P.M.H. Heemels, S.J.L. van Eijndhoven, and A.A. Stoorvogel. Linear quadratic regulator problem with positive controls. *Int. Journal of Control*, 70(4):551–578, 1998.

[18] D. Henrion and J.-B. Lasserre. GloptiPoly: global optimization over polynomials with Matlab and SeDuMi. *ACM Transactions on Mathematical Software*, 29(2):165–194, 2003.

[19] D. Henrion, S. Tarbouriech, and V. Kučera. Control of linear systems subject to time-domain constraints with polynomial pole placement and LMIs. *IEEE Trans. Automat. Contr.*, 50(9):1360–1364, 2005.

[20] T.A. Johansen, W. Jackson, R. Schreiber, and P. Tøndel. Hardware synthesis of explicit model predictive controllers. *IEEE Trans. Contr. Syst. Technol.*, 2006.

[21] V. Kučera. The pole placement equation - a survey. *Kybernetica*, 30:578–584, 1994.

[22] V. Kučera and P. Zagalak. Proper solutions of polynomial equations. In *Proceedings of the 14th IFAC World Congress*, pages 357–362, Beijing, China, 1999.

[23] M. Kvasnica, F.J. Christophersen, M. Herceg, and M. Fikar. Polynomial approximation of closed-form MPC for piecewise affine systems. In *Proceedings of the 17th IFAC World Congress*, pages 3887–3882, Seoul, Korea, 2008.

[24] J.-B. Lasserre. *Moments, Positive Polynomials and Their Applications.* Imperial College Press, 2009.

[25] M. Laurent. Sums of suqares, moment matrices and optimization over polynomials. In M. Putinar and S. Sullivant, editors, *Emerging Applications of Algebraic Geometry*, volume 149 of *Volumes in Mathematics and its applications*, pages 157–270. Springer, 2009.

[26] M. Lazar, W.P.M.H. Heemels, B.J.P. Roset, H. Nijmeijer, and P.P.J.v.d. Bosch. Input-to-state stabilizing sub-optimal nmpc with an application to dc-dc converters. *Int. Journal of Robust and Nonlinear Control*, 18(8):890–904, 2008.

[27] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD Conference*, Taipei, Taiwan, 2004.

[28] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[29] D.L. Powers. *Boundary value problems and partial differential equations.* 5th edition, Elsevier, 2006.

[30] S. Prajna, A. Papachristodoulou, and P.A. Parrilo. Introducing sostools: a general purpose sum of squares programming solver. In *Proceedings of the Conference on Decision and Control*, volume 1, pages 741–746, 2002.

[31] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[32] A. Saberi, J. Han, and A.A. Stoorvogel. Constrained stabilization problems for linear plants. *Automatica*, 38(4):639–654, 2002.

[33] A. Saberi, A.A. Stoorvogel, and P. Sannuti. *Control of linear systems with regulation and input constraints*. Springer Verlag Ltd., London, 2000.

[34] J. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11-12:625–653, 1999.

[35] S. Tarbouriech, G. Garcia, and A.H. Glattfelder. *Advanced strategies in control systems with input and output constraints*, volume 346 of *LNCIS*. Springer-Verlag, 2007.

[36] S. Tarbouriech and M. Turner. Anti-windup design: an overview of some recent advances and open problems. *IET Control theory & Applications*, 3(1):1–19, 2009.

[37] P. Tøndel, T.A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions,. *Automatica*, 39(3):489–497, 2003.