LMIs for constrained polynomial interpolation with application in trajectory planning

Didier Henrion[1,2,3], Jean-Bernard Lasserre[2]

July 16, 2003

Abstract

We consider an open-loop trajectory planning problem for linear systems with bound constraints originating from saturations or physical limitations. Using an algebraic approach and results on positive polynomials, we show that this control problem can be cast into a constrained polynomial interpolation problem admitting a convex linear matrix inequality (LMI) formulation.

Keywords

Linear systems, Algebraic/polynomial approach, Trajectory planning, Polynomial interpolation, Positive polynomials, LMI, Computer-aided control system design

1 Introduction

In the algebraic setting described by Kučera in his textbook [K79], fractions of polynomials and polynomial matrices are widely used to model linear control systems. In this framework, system stabilization amounts to solving polynomial or rational Bézout identities, yielding a parametrization of all stabilizing controllers. Pole placement design is equivalent to solving polynomial Diophantine equations, and more advanced techniques such as H_2 or H_∞ optimal control rely additionally on solving a quadratic polynomial equation, a problem generally referred to as spectral factorization.

[1]Corresponding author. FAX: +33 5 61 33 69 69. E-mail: henrion@laas.fr

[2]Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, 31 077 Toulouse, cedex 4, France.

[3]Also with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Praha, Czech Republic.

In this paper we apply polynomial/algebraic techniques to deal with a trajectory planning problem, motivated by the recent work [LN03]. Given a linear system, an open-loop control law must be computed so that system signals (input, output and their derivatives) satisfy given algebraic constraints:

- linear equality constraints, e.g. to ensure that system trajectories go through given points, or that they are smooth enough;
- bound constraints, e.g. on the input signal due to actuator saturation, or on the output signal due to physical limitations.

Modelling the system with polynomial matrix fraction descriptions, we can derive all the system signals from an auxiliary variable called internal state [K80] or flat output [FM00]. As explained in [K79], internal state equations are obtained by solving polynomial matrix Bézout identities. Algebraic constraints on system signals are then translated onto the internal state. Finding the appropriate open-loop control stategy then amounts to solving a polynomial interpolation problem with additional bound constraints.

Using results on positive polynomials recently exposed in [N00] (based on conic duality), [P00] (based on sum-of-squares decompositions and algebraic geometry results around Hilbert's 17th problem) and [L01] (based on the theory of moments), it turns out that this constrained polynomial interpolation problem can be easily formulated as a convex optimization problem over linear matrix inequalities (LMI) for which efficient semidefinite programming software are available.

2 Problem statement

We are given a multivariable linear system described by a left coprime polynomial matrix fraction

$$y(s) = A_l^{-1}(s)B_l(s)u(s) \tag{1}$$

in the Laplace domain, where $u(s)$ and $y(s)$ are the input and output signals, respectively, and $A_l(s)$ is non-singular [K79].

We seek a control law $u(t)$ in the time-domain $t \in [t_q, t_r]$ such that system input $u(t)$ and output $y(t)$ and their respectives derivatives $u^{(k)}(t)$ and $y^{(k)}(t)$ satisfy

- linear constraints
$$\begin{array}{rcl} u^{(k_i)}(t)\big|_{t=t_i} &=& u_i \\ y^{(k_i)}(t)\big|_{t=t_i} &=& y_i, \quad i = 1, 2, \ldots \end{array} \tag{2}$$
  where $k_i \geq 0$ are given integers, $t_i \in [t_q, t_r]$ are given real numbers, and u_i, y_i are given real numbers;

- bound constraints

$$\begin{array}{rcccl} u_i^{\rm low} & \geq & u^{(k_i)}(t) & \geq & u_i^{\rm up} \\ y_i^{\rm low} & \geq & y^{(k_i)}(t) & \geq & y_i^{\rm up}, \quad t \in [t_q, t_r], \quad i = 1, 2, \ldots \end{array} \tag{3}$$

  where $k_i \geq 0$ are given integers, and $u_i^{\rm low}$, $u_i^{\rm up}$, $y_i^{\rm low}$, $y_i^{\rm up}$ are given real vectors.

3 Algebraic formulation

Linear system (1) can be represented by a right coprime polynomial matrix fraction

$$A_l^{-1}(s)B_l(s) = B_r(s)A_r^{-1}(s)$$

where $A_r(s)$ is non-singular. Under the coprimeness assumption on the pair $(A_r(s),\, B_r(s))$ there exists a polynomial matrix solution pair $(X_l(s),\, Y_l(s))$ to the Bézout identity

$$X_l(s)A_r(s) + Y_l(s)B_r(s) = I.$$

Defining vector

$$x(s) = X_l(s)u(s) + Y_l(s)y(s)$$

as the internal state, or flat output of system (1), it follows that all system signals can be represented as linear combinations of signal $x(t)$ and its derivatives. For example, by virtue of the above Bézout identity, the input and output signals can be obtained as follows:

$$\begin{array}{rcccl} u(s) & = & A_r(s)x(s) & = & \left(\sum_k A_k s^k\right) x(s) \\ y(s) & = & B_r(s)x(s) & = & \left(\sum_k B_k s^k\right) x(s). \end{array}$$

As a consequence, algebraic constraints on vectors y and u can be translated into algebraic constraints on vector x.

Now assuming that $x(t)$ is a vector polynomial

$$x(t) = \sum_k x_k t^k$$

of given degree, interpolation constraints (2) can be written as:

$$\begin{array}{rcl} \sum_k A_k x^{(k+k_i)}(t)\big|_{t=t_i} & = & u_i \\ \sum_k B_k x^{(k+k_i)}(t)\big|_{t=t_i} & = & y_i, \quad i = 1, 2, \ldots \end{array}$$

which are linear constraints on coefficients x_k of polynomial $x(t)$, that we denote for convenience

$$Fx = f \tag{4}$$

where F is a given matrix, f is a given vector, and x denotes the column vector obtained by stacking column vector coefficients x_k. Note that we use the symbol x indifferently to denote time signal $x(t)$, its Laplace transform $x(s)$, and the vector of coefficients x_k.

Similarly, bound constraints (3) can be written as:

$$\begin{array}{ccccc} u_i^{\text{low}} & \leq & \sum_k A_k x^{(k+k_i)}(t) & \leq & u_i^{\text{up}} \\ y_i^{\text{low}} & \leq & \sum_k B_k x^{(k+k_i)}(t) & \leq & y_i^{\text{up}}, \quad t \in [t_q, t_r], \quad i = 1, 2, \ldots \end{array}$$

which can be formulated entrywise as non-negativity constraints

$$g_i(t) \geq 0, \quad t \in [t_q, t_r], \quad i = 1, 2, \ldots \tag{5}$$

on a set of scalar polynomials $g_i(t)$ whose coefficient vectors g_i depend linearly on coefficient vector x, i.e.

$$G_i x = g_i, \quad i = 1, 2, \ldots \tag{6}$$

for given matrices G_i.

Our original trajectory planning problem with bound constraints can therefore be cast into the following equivalent constrained polynomial interpolation problem.

Problem 1 *Given matrices F, G_i and vector f, find polynomial coefficient vector x satisfying polynomial positivity constraints (5), as well as linear constraints (4) and (6).*

4 Constrained polynomial interpolation

Lemma 1 *A given scalar polynomial $g_i(t)$ of degree n is non-negative along the interval $t \in [t_q, t_r]$ if and only if there exist polynomials $q_{ij}(t)$, $r_{ij}(t)$ of degree $m = n/2$ (n even) or $m = (n-1)/2$ (n odd) such that*

$$g_i(t) = (t - t_q) \sum_j q_{ij}^2(t) + (t_r - t) \sum_j r_{ij}^2(t). \tag{7}$$

Proof: When $n = 2m + 1$, the result follows readily from the Markov-Lukacs theorem, see e.g. [N00, §3.3]. When $n = 2m$, in virtue of the Markov-Lukacs theorem if $g_i(t)$ is positive along the interval $[t_q, t_r]$ there exist polynomials $\bar{q}_{ij}(t)$, $\bar{r}_{ij}(t)$ of degree m and $m - 1$ respectively, such that

$$g_i(t) = \sum_j \bar{q}_{ij}^2(t) + (t - t_q)(t_r - t) \sum_j \bar{r}_{ij}^2(t)$$

so that

$$(t - t_q)g_i(t) + (t_r - t)g_i(t) = (t_r - t_q)g_i(t) = (t - t_q)(\textstyle\sum_j \bar{q}_{ij}^2(t) + (t_r - t)^2 \sum_j \bar{r}_{ij}^2(t)) + \\ (t_r - t)(\textstyle\sum_j \bar{q}_{ij}^2(t) + (t - t_q)^2 \sum_j \bar{r}_{ij}^2(t)).$$

Hence denote

$$\sum_j q_{ij}^2(t) = (\sum_j \bar{q}_{ij}^2(t) + (t_r - t)^2 \sum_j \bar{r}_{ij}^2(t))/(t_r - t_q)$$

and

$$\sum_j r_{ij}^2(t) = (\sum_j \bar{q}_{ij}^2(t) + (t - t_q)^2 \sum_j \bar{r}_{ij}^2(t))/(t_r - t_q)$$

and expression (7) follows with polynomials of appropriate degrees. Conversely, if expression (7) is satisfied, then $g_i(t)$ is obviously positive along $[t_q, t_r]$. □

It turns out that the problem of finding polynomials satisfying the sum-of-squares decomposition (7) can be formulated as an LMI optimization problem. More specifically, define H_k as the Hankel matrix of dimension $m+1$ with ones along the $(k+1)$-th anti-diagonal, i.e.

$$H_0 = \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ & & & \ddots \end{bmatrix}, \quad H_1 = \begin{bmatrix} 0 & 1 & 0 & \\ 1 & 0 & 0 & \\ 0 & 0 & 0 & \\ & & & \ddots \end{bmatrix}, \quad H_2 = \begin{bmatrix} 0 & 0 & 1 & \\ 0 & 1 & 0 & \\ 1 & 0 & 0 & \\ & & & \ddots \end{bmatrix} \cdots$$

If $k < 0$ or $k > 2m$ then H_k is the zero matrix of dimension $m+1$. Denoting

$$g_i(t) = g_{i0} + g_{i1}t + g_{i2}t^2 + \cdots$$

it can be shown that relation (7) holds if and only if there exists positive semidefinite symmetric matrices Q_i, R_i solving the LMI problem

$$g_{ik} = \operatorname{trace} Q_i(H_{k-1} - t_q H_k) + \operatorname{trace} R_i(t_r H_k - H_{k-1}) \tag{8}$$

see e.g. [N00, §3.3].

Denoting for convenience the set of linear equations (8) for $k = 1, 2, \ldots$ as

$$g_i = H(Q_i, R_i)$$

where g_i is the coefficient vector of polynomial $g_i(t)$, we can now state the main result of this paper.

Theorem 1 *Coefficient vector x solves problem 1 if and only if there exist positive semidefinite symmetric matrices Q_i, R_i solving the LMI problem*

$$\begin{array}{rcl} G_i x & = & H(Q_i, R_i) \\ F x & = & f. \end{array} \tag{9}$$

5 Numerical example

The numerical example was treated with the help of Matlab 6.5 running under SunOS release 5.8 on a SunBlade 100 workstation. Operations on polynomials were performed with the Polynomial Toolbox 2.5 [P00]. The LMI problems were solved with SeDuMi 1.05 [S99] with default tuning parameters.

We consider the motorized base-stage high-precision positioning system described in [LN03]. System polynomial matrices in description (1) are given by:

$$A_l(s) = \begin{bmatrix} 25s^2 & 0 \\ 0 & 450s^2 + 1.1875 \cdot 10^4 s + 6.3955 \cdot 10^5 \end{bmatrix} \quad B_l(s) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

where the input $u(t)$ is the force applied to the stage, the first output $y_m(t)$ is the relative position of the center of mass of the stage with respect to a coordinate frame attached to the base, whose origin is the second output $y_b(t)$, the position of the center of mass of the base in a fixed coordinate frame related to the ground [LN03].

We want to generate displacements of the stage from one steady state ($t_r = 0$, $y_m(t_r) = 0$) to another one ($t_q = 1$, $y_m(t_q) = 0.02$) with the base also in steady state at the stage's final position ($y_b(t_r) = y_b(t_q) = 0$). Moreover the trajectory of the stage must be smooth enough ($\dot{y}_m(t_r) = \dot{y}_m(t_q) = 0$). So we enforce six interpolation constraints of the type (2).

Using macro `lmf2rmf` (left to right polynomial matrix fraction) of the Polynomial Toolbox we obtain:

$$\begin{array}{rcl} A_r(s) & = & 0.99903s^2 + 0.18550 \cdot 10^{-1}s^3 + 0.70294 \cdot 10^{-3}s^4 \\ B_r(s) & = & \begin{bmatrix} 0.39961 \cdot 10^{-1} + 0.74200 \cdot 10^{-3}s + 0.28117 \cdot 10^{-4}s^2 \\ -0.15621 \cdot 10^{-5}s^2 \end{bmatrix}. \end{array}$$

Given these matrices, interpolation data in (4) are as follows:

$$F = \begin{bmatrix} 0 & 0 & -0.31242 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & -0.31242 \cdot 10^{-5} & -0.93725 \cdot 10^{-5} & -0.18745 \cdot 10^{-4} & -0.31242 \cdot 10^{-4} \\ 0.39961 \cdot 10^{-1} & 0.74200 \cdot 10^{-3} & 0.56235 \cdot 10^{-4} & 0 & 0 & 0 \\ 0.39961 \cdot 10^{-1} & 0.40703 \cdot 10^{-1} & 0.41501 \cdot 10^{-1} & 0.42356 \cdot 10^{-1} & 0.43267 \cdot 10^{-1} & 0.44233 \cdot 10^{-1} \\ 0 & 0.39961 \cdot 10^{-1} & 0.14840 \cdot 10^{-2} & 0.16870 \cdot 10^{-3} & 0 & 0 \\ 0 & 0.39961 \cdot 10^{-1} & 0.81406 \cdot 10^{-1} & 0.12450 & 0.16942 & 0.21633 \end{bmatrix}$$

$$f = \begin{bmatrix} 0 & 0 & 0 & 0.20000 \cdot 10^{-1} & 0 & 0 \end{bmatrix}^T$$

A fifth-degree interpolating polynomial $x(t)$ is readily obtained by solving non-singular linear system (2):

$$x(t) = 0.40962 \cdot 10^{-3} - 0.22060 \cdot 10^{-1}t + 5.2255t^3 - 7.8382t^4 + 3.1353t^5.$$

The resulting output trajectories are shown in figure 1.

Now assume that for physical reasons, base position $y_b(t)$ cannot exceed some given upper bound, e.g.

$$y_b(t) \leq 3 \cdot 10^{-6}.$$

Note that we do not enforce any lower bound on signal $y_b(t)$. In order to cope with the additional constraint, suppose we seek a polynomial of degree seven. Applying theorem 1, SeDuMi returns after 0.4 seconds of CPU time the following polynomial

$$x(t) = 0.12205 \cdot 10^{-2} - 0.65731 \cdot 10^{-1}t + 15.570t^3 - 48.999t^4 + 65.207t^5 - 41.780t^6 + 10.567t^7$$

solving LMI problem (9). The resulting output trajectories are shown in figure 2. We can see that the bound constraint is indeed satisfied, at the cost of a higher degree of polynomial $x(t)$ and a larger displacement of the base in the other direction

6 Conclusion

In this paper we applied recent results on the LMI formulation of polynomial interpolation with additional constraints to solve a trajectory planning control problem for linear systems in a pure algebraic/polynomial framework. Note that an alternative technique was proposed in [FM00, Ex. 4.2] to cope with saturation constraints: an appropriate time-scaling is used there to reduce the control amplitude, by widening the time interval during which the control law is applied. Our approach is different, since we assume that the time interval is kept unchanged.

Our work can be extended in various directions:

- Since the general flatness framework has been originally developped for nonlinear systems [FM00], and LMI techniques can also be applied to solve stability and stabilization problems for nonlinear (polynomial) systems [P00], it may be worth studying potential extensions of this work to nonlinear systems;
- As pointed out in [N00, §4.1], besides bound constraints, general semi-algebraic constraints can be enforced in the polynomial interpolation problem without loss of the convex LMI formulation. For example, system trajectories can be upper- and lower-bounded by polynomial curves;
- Based on results in [L01], the approach can be extended to two-dimensional (time and space) systems, or general multi-dimensional systems. The multi-dimensional constrained polynomial interpolation problem can then be solved with a hierarchy of LMI optimization problems of increasing size, with theoretical guarantee of convergence;
- Finally, we are currently investigating connections with polynomial techniques for the control of input constrained systems [HTK01], with the objective of designing a closed-loop feedback law ensuring stabilization (or robust stabilization) in the presence of input saturation.

Acknowledgment

Support of the Grant Agency of the Czech Republic under Project No. 102/02/0709 is acknowledged.

References

[FM00] M. Fliess, R. Marquez. Continuous-time linear predictive control and flatness: a module-theoretic setting with examples. *International Journal of Control*, Vol. 73, No. 7, pp. 606–623, 2000.

[HTK01] D. Henrion, S. Tarbouriech, V. Kučera. Control of Linear Systems subject to Input Constraints: a Polynomial Approach. *Automatica*, Vol. 37, No. 4, pp. 597-604, 2001.

[K80] T. Kailath. Linear Systems. *Prentice Hall*, Englewood Cliffs, NJ, 1980.

[K79] V. Kučera. Discrete Linear Control: The Polynomial Approach. *John Wiley and Sons*, Chichester, UK, 1979.

[L01] J. B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, Vol. 11, No. 3, pp. 796–817, 2001.

[LN03] J. Lévine, D. V. Nguyen. Flat output characterization for linear systems using polynomial matrices. *Systems and Control Letters*, Vol. 48, pp. 69–75, 2003.

[N00] Yu. Nesterov. Squared Functional Systems and Optimization Problems. Chapter 17, pp. 405–440 in H. Frenk, K. Roos, T. Terlaky and S. Zhang (Editors). High Performance Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[P00] P. A. Parrilo. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. *PhD Thesis*, California Institute of Technology, Pasadena, California, 2000.

[P00] PolyX, Ltd. The Polynomial Toolbox for Matlab. Prague, Czech Republic. Version 2.5 released in 2000. See `www.polyx.cz`

[S99] J. F. Sturm. Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, Vol. 11-12, pp. 625–653, 1999. See `fewcal.kub.nl/sturm`

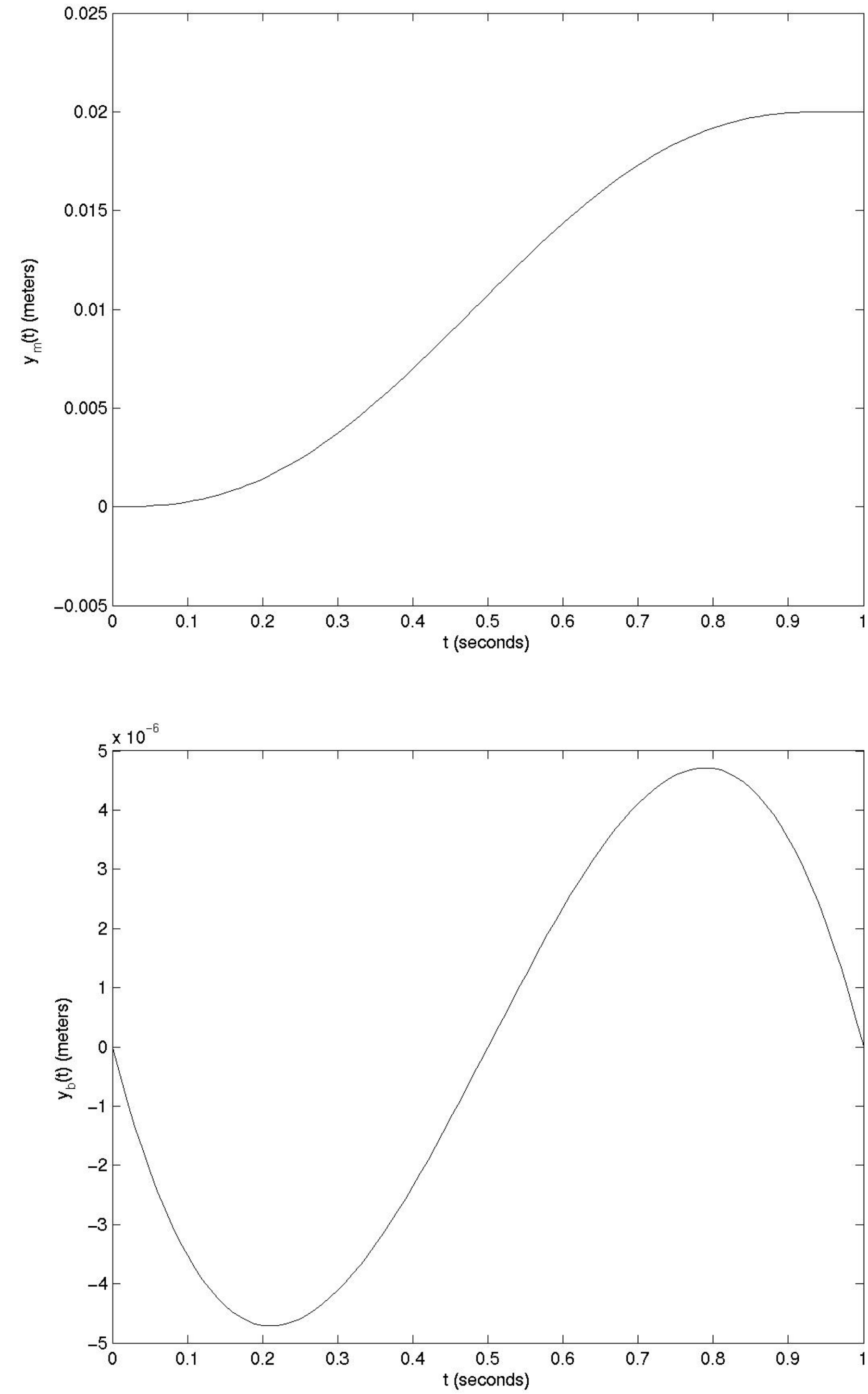

Figure 1: System outputs $y_m(t)$ (top) and $y_b(t)$ (bottom) without bound constraint.

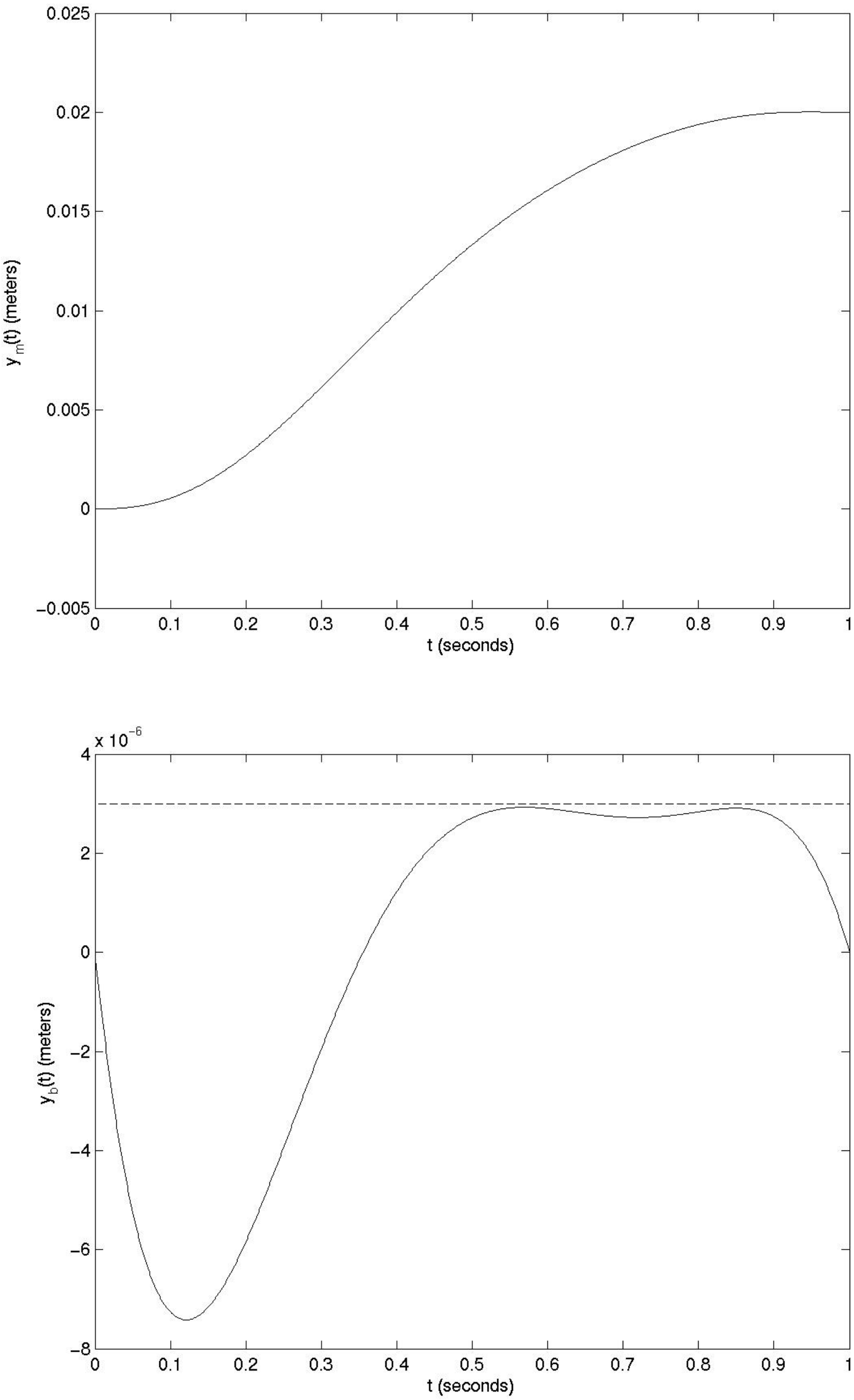

Figure 2: System outputs $y_m(t)$ (top) and $y_b(t)$ (bottom) with bound constraint.