LMI-Based Nonlinear Control of the Furuta Pendulum ⋆

David Vázquez * **Juan Carlos Arceo** * **Raymundo Márquez** *
Víctor Estrada-Manzo * **Miguel Bernal** *

* *Department of Electrical and Electronics Engineering, Sonora Institute of Technology, Ciudad Obregón, Sonora, Mexico (e-mail: raymundo.marquez@itson.edu.mx)*

Abstract: This paper is concerned with nonlinear control of an underactuated rotatory system better known as the Furuta pendulum. Instead of the traditional approaches which are based on linearization or involved nonlinear control techniques, the proposed controller design is based on exact nonlinear convex representations of the plant which lead to linear matrix inequalities when combined with the direct Lyapunov method. It is shown that robustness as well as actuator bounds can be easily incorporated in the methodology, which preserves its nonlinear qualities while making it numerically tractable. Simulation and real-time results allow the reader to appreciate the effectiveness of the proposed technique.

Keywords: Lyapunov stability, Convex optimisation, Linear matrix inequalities.

1. INTRODUCTION

Highly nonlinear plants have been traditionally employed to illustrate the effectiveness of a wide variety of control techniques. Among these systems, the different pendulum setups such as the inverted pendulum on a cart (Angeli, 2001), the Pendubot (Begovich et al., 2002), the triple one (Farwig et al., 1990), on an inclined rail (Furuta et al., 1980), with an inertia wheel (Spong et al., 2001), and the rotatory one (Furuta et al., 1992), have been traditionally used due to their rich dynamics, underactuated characteristics, and highly nonlinear behaviour. This paper is concerned with the latter one, also known as the Furuta pendulum, which is an underactuated mechanism with 2 degrees of freedom (DOF), 2 beams, and 2 rotational joints. The actuator only rotates an horizontal beam, based on which the second beam should be stabilized in its upright position.

Any number of control techniques have been employed for the class of inverted pendulum systems. The simplest are based on the linearization of the nonlinear model and a more or less straightforward application of linear techniques such as pole placement (Ogata, 2001). Needless to say, these approaches lack a proper treatment of the nonlinearities, which limits their applicability to a neighbourhood of the operating point. The latter disadvantage is inherited when multiple linear controllers are mixed via some gain scheduling approach, since it also relies on approximations (Khalil, 2002). On the other hand, using exact nonlinear models may prove to be highly complex and little rewarding, with most of the nonlinear control techniques being very involved and requiring a deep knowledge of specialized areas such as geometric control (Isidori, 1995), sliding modes (Utkin, 1992), passivity (Ortega et al., 1998), or feedback linearization/backstepping (Khalil, 2014).

In recent years, nonlinear control design has shifted from purely analytical solutions as those mentioned above to systematic numerically computable techniques such as those based on linear matrix inequalities (LMIs) (Boyd et al., 1994). LMIs have become very popular as an adequate tool for controller design because they belong to the P-class of computational problems, i.e., their feasibility can be optimally decided in polynomial time (Scherer, 2004). Moreover, they arise when exact convex rewriting of nonlinear expressions such as the sector nonlinearity approach (Taniguchi et al., 2001) is combined with the direct Lyapunov method, producing sufficient conditions for genuine systematic nonlinear controller design (Tanaka and Wang, 2001). Input/output saturation limits, decay rate, and other performance measures can be easily translated into LMIs even in a nonlinear context via convex representations (Duan and Yu, 2013).

This work is concerned with real-time implementation of an LMI-based nonlinear controller in the Furuta pendulum. The plant model is described in section 2, where the nonlinear expressions are exactly rewritten as convex sums. Section 3 shows how a nonlinear model in a convex form can be analyzed and controlled via LMI conditions, which are derived by means of a quadratic Lyapunov function. Section 4 presents the simulation and real-time implementation results of the designed controller along with a discussion on related issues. Concluding remarks are given in section 5.

2. THE FURUTA PENDULUM

The Furuta pendulum is shown in Fig.1; it consists on two beams: an horizontal one at the base which is

⋆ The authors have been sponsored by the CONACYT scholarships No. 419080 and No. 415715 and by the PROFAPI Project 2016-0081.

rotationally driven by a fixed DC motor, and a vertical one which is articulated by a rotary joint with the first one. Obviously, controlling the second beam as to keep it in its upright position via the DC motor is an underactuated task. Via Euler-Lagrange modelling, a state-space representation of this plant can be found (Quanser, 2006); it is given in (1). The horizontal beam describes an angle x_1 with respect to a fixed position; its angular velocity is given by x_2. The vertical beam describes an angle x_3 with respect to the vertical upright axis; x_4 is the corresponding angular velocity. Our task is to keep the second beam in its upright position, which coincides with driving the system to $x_3 = 0$.

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{(\beta+\gamma)\left(\delta x_4^2 \sin x_3 - 2\beta x_2 x_4 \cos x_3 \sin x_3 + u\right)}{((\beta+\gamma)\beta+\delta^2)\sin^2 x_3 + (\beta+\gamma)\alpha - \delta^2} \\
&\quad - \frac{\delta \cos x_3 \left(\beta x_2^2 \cos x_3 \sin x_3 + \sigma g \sin x_3\right)}{((\beta+\gamma)\beta+\delta^2)\sin^2 x_3 + (\beta+\gamma)\alpha - \delta^2} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{\left(\beta \sin^2 x_3 + \alpha\right)\left(\beta x_2^2 \cos x_3 \sin x_3 + \sigma g \sin x_3\right)}{((\beta+\gamma)\beta+\delta^2)\sin^2 x_3 + (\beta+\gamma)\alpha - \delta^2} \\
&\quad - \frac{\delta \cos x_3 \left(\delta x_4^2 \sin x_3 - 2\beta x_2 x_4 \cos x_3 \sin x_3 + u\right)}{((\beta+\gamma)\beta+\delta^2)\sin^2 x_3 + (\beta+\gamma)\alpha - \delta^2}
\end{aligned} \tag{1}$$

with $\alpha = \left(J_0 + m_1 L_0^2\right)/T_c$, $\beta = \left(m_1 l_1^2\right)/T_c$, $\gamma = J_1/T_c$, $\delta = \left(m_1 L_0 l_1\right)/T_c$, $\sigma = \left(m_1 l_1\right)/T_c$, $L_0 = 0.068$m and $J_0 = 6.9885e^{-5}$kg·m^2 are length and total moment of inertia of the horizontal link, respectively, $m_1 = 0.02366$kg, $l_1 = 0.08$m, and $J_1 = 1.7590e^{-4}$kg·m^2 are the mass, center of mass, and total moment of inertia of the vertical beam, respectively, $g = 9.81$m/s^2 and $T_c = 0.0049431$ N·m/V are the gravitational constant and the torque constant, respectively.

Each nonlinearity in (1) can be *algebraically rewritten* as a convex sum; the first step towards this goal consists on choosing these expressions. For instance, if the chosen nonlinearities are $z_1 = (\beta+\gamma)\,\delta x_4^2 - 2\beta\,(\beta+\gamma)\,x_2 x_4 \cos x_3 - \beta\delta x_2^2 \cos^2 x_3 - \delta\sigma g \cos x_3$, $z_2 = (\sin x_3)/x_3$, $z_3 = \left(\beta \sin^2 x_3 + \alpha\right)\left(\beta x_2^2 \cos x_3 + \sigma g\right) - \delta \cos x_3 \left(\delta x_4^2 - 2\beta x_2 x_4 \cos x_3\right)$, $z_4 = \cos x_3$, and $z_5 = 1/\left(\left(\beta^2 + \gamma\beta + \delta^2\right)\sin^2 x_3 + \alpha\beta + \alpha\gamma - \delta^2\right)$, then, (1) is equivalent to:

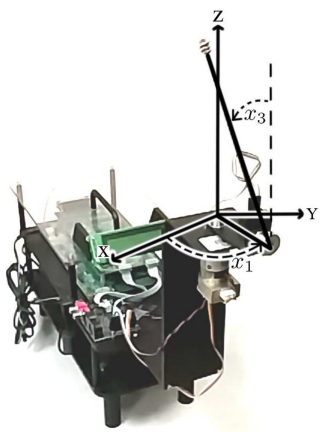

Fig. 1. Furuta pendulum

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & z_1 z_2 z_5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & z_2 z_3 z_5 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ (\beta+\gamma) z_5 \\ 0 \\ -\delta z_4 z_5 \end{bmatrix} u. \tag{2}$$

Since the Furuta pendulum is not supposed to operate for any value of the model states, we can restrict our operation regime to $|x_2| \leq 10$ rad/s, $|x_3| \leq 15$ (in degrees), and $|x_4| \leq 3$ rad/s (note that x_1 is not at the right-hand side of the model). Taking into account these bounds, the variables in (2) can also be bounded as $z_1 \in [-0.2833, -0.0324]$, $z_2 \in [0.9886, 1.0000]$, $z_3 \in [0.1251, 0.2961]$, $z_4 \in [0.9659, 1.0000]$, and $z_5 \in [524.0978, 579.0187]$. Note that for any nonlinearity $z_i \in [z_i^0, z_i^1]$, the following is *an algebraic identity*:

$$z_i = \underbrace{\frac{z_i^1 - z_i}{z_i^1 - z_i^0}}_{w_0^i}\left(z_i^0\right) + \underbrace{\frac{z_i - z_i^0}{z_i^1 - z_i^0}}_{w_1^i}\left(z_i^1\right), \quad \begin{array}{l} w_0^i + w_1^i = 1, \\ 0 \leq w_j^i \leq 1. \end{array} \tag{3}$$

Therefore, since $w_0^i + w_1^i = 1$, each z_i, $i \in \{1, 2, \ldots, 5\}$ in (2) can be rewritten as a convex sum. Convex sums can be stacked together because if the expression they involve is not concerned with their index, it is equivalent to multiply by 1:

$$\dot{x} = \sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\cdots\sum_{i_5=0}^{1} w_{i_1}^1 w_{i_2}^2 \cdots w_{i_5}^5 \left(\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & z_1^{i_1} z_2^{i_2} z_5^{i_5} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & z_2^{i_2} z_3^{i_3} z_5^{i_5} & 0 \end{bmatrix}}_{A_i} x + \underbrace{\begin{bmatrix} 0 \\ (\beta+\gamma) z_5^{i_5} \\ 0 \\ -\delta z_4^{i_4} z_5^{i_5} \end{bmatrix}}_{B_i} u \right) = \sum_{i=1}^{r} h_i(x)\left(A_i x + B_i u\right), \tag{4}$$

where $h_i = w_{i_1}^1 w_{i_2}^2 \cdots w_{i_5}^5$, $[i_1 i_2 \cdots i_5]$ is the 5-digit binary representation of $(i-1)$, $i \in \{1, 2, \ldots, r\}$, $r = 2^5 = 32$. Despite its appearance, system (4) is *not linear* nor an approximation of (1): its construction proves that it is *algebraically equivalent* to the original one.

As a way of illustration, some of the 32 pairs (A_i, B_i) are given below; they correspond to the cases described in table 1, where z_i, $i \in \{1, 2, \ldots, 5\}$ have taken some of their corresponding extreme values.

Table 1. Some values for (2)

i	z_1	z_2	z_3	z_4	z_5
1	-0.2833	0.9886	0.1251	0.9659	524.0978
8	-0.2833	0.9886	0.2961	1.0000	579.0187
16	-0.2833	1.0000	0.2961	1.0000	579.0187
32	-0.0324	1.0000	0.2961	1.0000	579.0187

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -146.77 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 64.84 & 0 \end{bmatrix}, \; A_8 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -162.15 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 169.48 & 0 \end{bmatrix},$$

$$A_{16} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -164.02 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 171.43 & 0 \end{bmatrix}, A_{32} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -18.77 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 171.43 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 34.695 \\ 0 \\ -13.162 \end{bmatrix}, B_8 = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -15.055 \end{bmatrix},$$

$$B_{16} = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -15.055 \end{bmatrix}, B_{32} = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -15.055 \end{bmatrix}.$$

Naturally, the choice of nonlinearities is not unique. For instance, if 6 nonlinearities $\zeta_1 = z_2$, $\zeta_2 = z_4$, $\zeta_3 = z_5$, $\zeta_4 = \sin^2 x_3 \in [0, 0.0670]$, $\zeta_5 = x_2 \sin x_3 \in [-2.5882, 2.5882]$, and $\zeta_6 = x_4 \sin x_3 \in [-0.7765, 0.7765]$ are picked up, then, following the same procedure, (1) can be alteratively written as:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \delta(\zeta_4 - 1)\beta\zeta_5\zeta_3 & -\delta\sigma g\zeta_1\zeta_2\zeta_3 & \delta\zeta_6\zeta_3(\beta+\gamma) \\ 0 & -2\beta(\beta+\gamma)\zeta_2\zeta_3\zeta_6 & 0 & 1 \\ 0 & (\beta\zeta_4 + \alpha)\,\beta\zeta_5\zeta_2\zeta_3 & \beta\sigma g\zeta_1\zeta_3\zeta_4 & -\delta^2\zeta_2\zeta_3\zeta_6 \\ & +2\beta\delta(1-\zeta_4)\zeta_6\zeta_3 & +\alpha\sigma g\zeta_1\zeta_3 & \end{bmatrix} x + \begin{bmatrix} 0 \\ (\beta+\gamma)\zeta_3 \\ 0 \\ -\delta\zeta_2\zeta_3 \end{bmatrix} u, \quad (5)$$

from which a convex model of the form (4) is found with $r = 2^6 = 64$ new functions $h_i(x)$ and new pairs (A_i, B_i).

Again, some of the 64 pairs (A_i, B_i) are given below; their extreme values are taken from table 2.

Table 2. Some values for (5)

i	ζ_1	ζ_2	ζ_3	ζ_4	ζ_5	ζ_6
1	0.989	0.966	524.0988	0.000	-2.588	-0.777
16	0.989	0.966	579.019	0.067	2.588	0.777
32	0.989	1.000	579.019	0.067	2.588	0.777
64	1.000	1.000	579.019	0.067	2.588	0.777

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 2.67 & -48.88 & -0.70 \\ 0 & 0 & 0 & 1 \\ 0 & -2.10 & 70.65 & 0.27 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 34.695 \\ 0 \\ -13.162 \end{bmatrix},$$

$$A_{16} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.87 & -54 & 0.77 \\ 0 & 0 & 0 & 1 \\ 0 & 2.37 & 82.46 & -0.29 \end{bmatrix}, B_{16} = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -14.541 \end{bmatrix},$$

$$A_{32} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.93 & -55.90 & 0.77 \\ 0 & 0 & 0 & 1 \\ 0 & 2.43 & 82.46 & -0.30 \end{bmatrix}, B_{32} = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -15.055 \end{bmatrix},$$

$$A_{64} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2.93 & -56.55 & 0.77 \\ 0 & 0 & 0 & 1 \\ 0 & 2.43 & 83.41 & -0.30 \end{bmatrix}, B_{64} = \begin{bmatrix} 0 \\ 38.331 \\ 0 \\ -15.055 \end{bmatrix}.$$

3. CONTROLLER DESIGN VIA LMIS

What is the convenience of writing the nonlinear model (1) in a convex form (4)? As it will be shown in this section, convexity allows using a quadratic Lyapunov function of the form $V(x) = x^T P x$ to perform controller design via sufficient LMI conditions. The proposed control law, known as *parallel distributed compensation* (PDC), is a nonlinear generalization of the ordinary state feedback:

$$u(t) = \sum_{i=1}^{r} h_i(x) F_i x(t), \quad (6)$$

where $h_i(x)$ are the same nonlinear convex functions of the convex model (2) or (5), depending on which $r = 32$ or $r = 64$. Notice that if a common gain F is used, (6) reduces to ordinary state feedback because $\sum_{i=1}^{r} h_i = 1$.

Once (6) is substituted in (4) we have the following closed-loop system:

$$\dot{x}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(x) h_j(x) \left(A_i + B_i F_j\right) x(t), \quad (7)$$

where convexity of the sums have been used to stack them at the left.

Now, consider a quadratic Lyapunov function candidate $V(x) = x^T P x$, $P = P^T > 0$. Taking its time derivative and omitting arguments when convenient, leads to

$$\begin{aligned} \dot{V} =& x^T P \dot{x} + \dot{x}^T P x, \\ =& x^T P \sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j \left(A_i + B_i F_j\right) x \\ &+ \sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j x^T \left(A_i + B_i F_j\right)^T P x \\ =& \sum_{i=1}^{r}\sum_{j=1}^{r} h_i h_j x^T \left(P(A_i + B_i F_j) + (A_i + B_i F_j)^T P\right) x. \end{aligned}$$

Thus, due to convexity of functions h_i and h_j, $\dot{V} < 0$ is guaranteed if for $(i, j) \in \{1, 2, \ldots, r\}^2$ the following holds

$$\begin{aligned} & P A_i + P B_i F_j + A_i^T P + F_j^T B_i^T P < 0 \\ & \iff A_i X + B_i M_j + X A_i^T + M_j^T B_i^T < 0, \end{aligned} \quad (8)$$

provided $X = P^{-1}$ and $M_j = F_j P^{-1}$. Thus, the origin of the nonlinear system (1) under the PDC control law (6) is asymptotically stable if $\exists X = X^T > 0$ such that LMIs (8) hold. Once these LMIs are solved via any convex programming tool such as the LMI Toolbox (Gahinet et al., 1995) or SeDuMi (Sturm, 1999), the gains in the PDC control law (6) are given by $F_i = M_j X^{-1}$ and the Lyapunov matrix is $P = X^{-1}$.

Due to the fact that $h_i h_j = h_j h_i$ in double convex sums as those found above, it is possible to find less conservative LMI conditions to guarantee $\dot{V} < 0$. One popular option which does not require additional decision variables, is programming (Tuan et al., 2001):

$$\begin{aligned} & \frac{2}{r-1}\left(A_i X + B_i M_i + X A_i^T + M_i^T B_i^T\right) \\ & \quad + \left(A_i X + B_i M_j + X A_i^T + M_j^T B_i^T\right) \\ & \quad + \left(A_j X + B_j M_i + X A_j^T + M_i^T B_j^T\right) < 0 \end{aligned} \quad (9)$$

instead of (8), for each $(i, j) \in \{1, 2, \ldots, r\}^2$.

In this report, two PDC controllers of the form (6) were designed: one is based on the convex model with 5 nonlinearities (2); the other one on the model with 6 nonlinearities (5). Recall that, regardless of the convex representation, the nonlinear model is *fully taken into account with no approximations involved*.

Moreover, the LMI framework allows easily incorporating restrictions such as decay rate $\eta > 0$, which increases the rate of convergence while augmenting the magnitude of the control signal (Tanaka and Wang, 2001):

$$\dot{V} \leq -2\eta V \Leftarrow A_i X + B_i M_j + X A_i^T + M_j^T B_i^T + 2\eta X < 0. \quad (10)$$

To compensate a possible excess on the magnitude of the input signal $|u(t)|$, the following LMIs can be added in order to guarantee that actuator limits are not surpassed (Tanaka and Wang, 2001):

$$|u(t)| \leq \mu \Leftarrow \begin{bmatrix} 1 & x^T(0) \\ x(0) & X \end{bmatrix} \geq 0, \begin{bmatrix} X & M_j^T \\ M_j & \mu^2 \end{bmatrix} \geq 0, \quad (11)$$

where $x(0)$ is the initial condition.

As happens with any set of inequalities, LMI conditions corresponding to different objectives (stabilization, decay rate, input constraints, etc.) can be programmed together without further adaptations; if feasible, all the objectives involved will be met. For the performance conditions above, sum relaxations as those in (9) can also be adopted.

Since the LMI conditions (10) and (11) are only sufficient, an important issue is the computational complexity of the LMI problem. Such complexity can be established in terms of the number of scalar decision variables $\mathcal{N}_v$ and the number of LMI rows $\mathcal{N}_L$. Considering conditions (10) and (11), computational complexity indexes are given by

$$\mathcal{N}_v = n_x (n_x + 1) + r n_x n_u$$
$$\mathcal{N}_L = r^2 n_x + (n_x + 1)(r + 1) + 2,$$

where n_x is the number of states, n_u is the number of inputs, and r is the number of rules.

4. SIMULATION AND REAL-TIME RESULTS

The LMI problems have been solved by using SeDuMi (Sturm, 1999) and the YALMIP toolbox (Lofberg, 2004) within a MATLAB R2009b platform. Simulations and implementations have been performed using the files developed by Quanser which are based on Simulink/MATLAB and Wincon 5.2 (Quanser, 2006).

The PDC control laws obtained via the LMI methods above have been applied to the original nonlinear model for simulation; the provider simulation files were first used (Quanser, 2006) to guarantee a safe real-time implementation. As it is usually the case, an adequate behaviour in simulation may not straightforwardly translate into the same performance when applied in real-time to the plant. Whenever the control performance in real-time conditions was unsatisfactory, controller redesign was made: if not enough actuator energy was involved, decay rate was maximized; if the actuator signal was over the permitted values, input constraints were employed. Simulation and real-time results below correspond to a

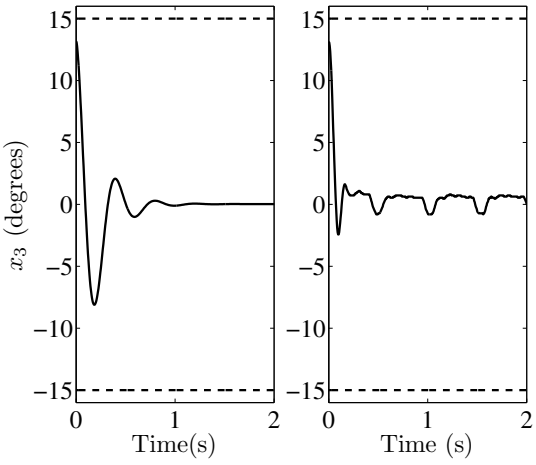

Fig. 2. Angle x_3 (solid-line) and its upper- and lower-bound (dashed-lines) of non-perturbed system (5-nonlinearity case): simulation (left), real-time (right)

decay rate of $\eta = 0.25$ and input constraint of $\mu = 22$. The closed-loop behaviour has been tested under two situations:

i) without perturbations, and
ii) with a systematic perturbation occurring at $t = 5$ seconds, which consists on adding 4 volts to the control input $u(t)$.

4.1 The 5-nonlinearity case

The nonlinear model (2) where 5 nonlinearities were rewritten in a convex form is employed in this section. The fact that only 5 nonlinearities were used implies that the number of pairs (A_i, B_i) is $2^5 = 32$, which may suppose a numerical advantage over the 6-nonlinearity case. This is indeed true, but it must be taken into account that a reduced number of nonlinearities produce convex representations whose model pairs (A_i, B_i) (the only ones involved in the LMIs) are "further" amongst them. In other words, though both representations are *algebraically exact and equivalent* to the original nonlinear system, the "corners" of the convex representation (i.e., the pairs (A_i, B_i)) are not the same. Therefore, there is a tradeoff between the size of the numerical problem and its feasibility (Boyd et al., 1994).

The controller design based on LMIs (8), (10), and (11), through the Tuan relaxation (9) yielded a feasible solution. Due to space limitations, only the Lyapunov matrix and some of the controller gains are given:

$$P = \begin{bmatrix} 0.0032 & 0.0078 & 0.1325 & 0.0244 \\ 0.0078 & 0.0272 & 0.4561 & 0.0844 \\ 0.1325 & 0.4561 & 14.3922 & 1.6289 \\ 0.0244 & 0.0844 & 1.6289 & 0.2818 \end{bmatrix}, \; F_1 = \begin{bmatrix} 0.6350 \\ 2.2342 \\ 78.3384 \\ 8.0987 \end{bmatrix}^T,$$

$$F_8 = \begin{bmatrix} 0.3510 \\ 1.2466 \\ 58.6170 \\ 4.7811 \end{bmatrix}^T, \; F_{16} = \begin{bmatrix} 0.3479 \\ 1.2357 \\ 58.4687 \\ 4.7449 \end{bmatrix}^T, \; F_{32} = \begin{bmatrix} 0.1991 \\ 0.7203 \\ 45.3534 \\ 3.0389 \end{bmatrix}^T.$$

The stabilization of x_3 (angle of the vertical beam w.r.t. the upright position) and the control signal $u(t)$ are shown in Figs. 2 and 3, respectively. In each plot, simula-

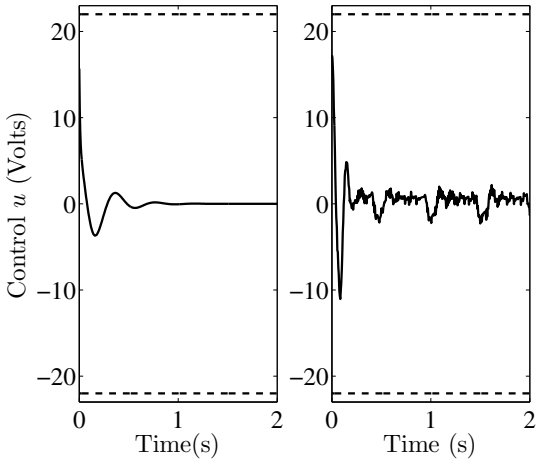

Fig. 3. Control law u (solid-line) and its upper- and lower-bound (dashed-lines) of non-perturbed system (5-nonlinearity case): simulation (left), real-time (right)

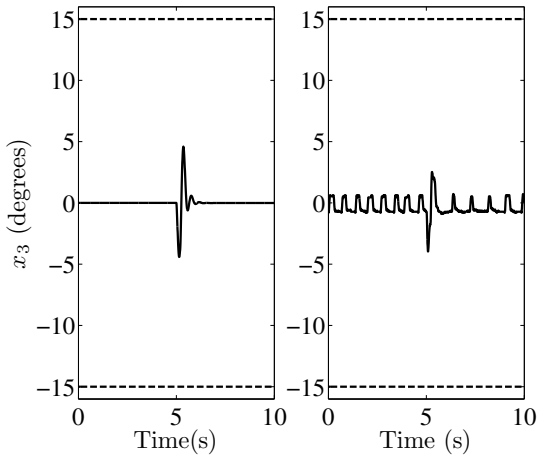

Fig. 4. Angle x_3 (solid-line) and its upper- and lower-bound (dashed-lines) of perturbed system (5-nonlinearity case): simulation (left), real-time (right)

tion results are shown at the left side whereas real-time ones are displayed at the right. The initial conditions have been set to $x(0) = [0\ 0\ 12.5^\circ\ 0]^T$. Note that the bounds employed for the convex representation (operating regime) as well as the control saturation of $u(t)$ are not exceeded, even in real-time. These simulations and real-time implementations were undisturbed whereas the results in Fig. 4 correspond to the perturbed case; clearly, the control scheme is able to successfully deal with the latter.

4.2 The 6-nonlinearity case

When the model (5) is employed (which, again, we insist, is the *same* as the original nonlinear model), $2^6 = 64$ pairs (A_i, B_i) arise. With these pairs, LMIs (8), (10), and (11), altogether with the Tuan relaxation (9) were found feasible. Again, only the Lyapunov matrix and some of the controller gains are given for brevity:

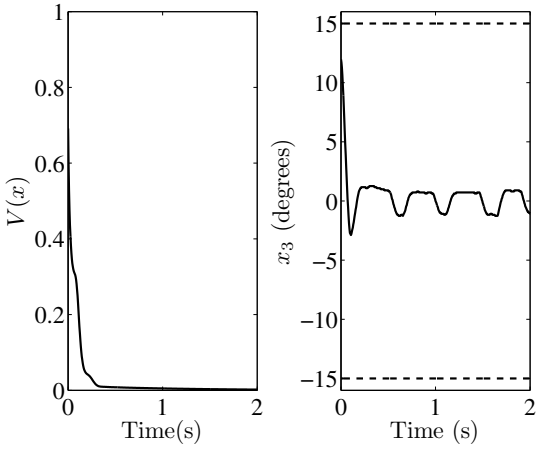

Fig. 5. Lyapunov function $V(x)$ in simulation (left) and angle x_3 (solid-line) and its upper- and lower-bound (dashed-lines) of non-perturbed system in real-time (right): 6-nonlinearity case

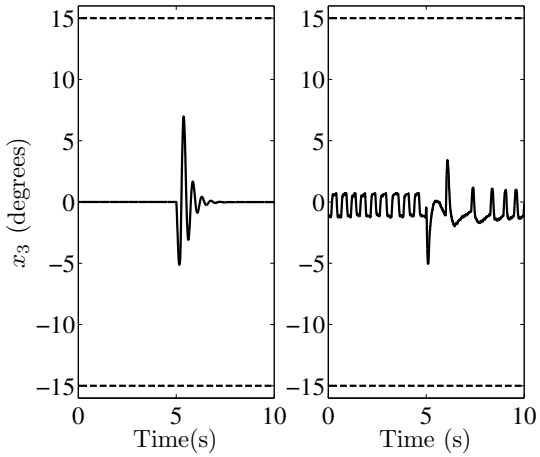

Fig. 6. Angle x_3 (solid-line) and its upper- and lower-bound (dashed-lines) of perturbed system (6-nonlinearity case): simulation (left), real-time (right)

$$P = \begin{bmatrix} 0.0178 & 0.0116 & 0.2760 & 0.0376 \\ 0.0116 & 0.0227 & 0.5284 & 0.0745 \\ 0.2760 & 0.5284 & 14.5259 & 1.8392 \\ 0.0376 & 0.0745 & 1.8392 & 0.2680 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 1.3979 \\ 2.9469 \\ 82.2753 \\ 10.7664 \end{bmatrix}^T,$$

$$F_{16} = \begin{bmatrix} 0.1703 \\ 0.6919 \\ 22.1566 \\ 2.0755 \end{bmatrix}^T, \quad F_{32} = \begin{bmatrix} 0.1539 \\ 0.6635 \\ 21.4156 \\ 1.9684 \end{bmatrix}^T, \quad F_{64} = \begin{bmatrix} 0.1561 \\ 0.6678 \\ 21.5449 \\ 1.9837 \end{bmatrix}^T.$$

As mentioned before, this system has been tested with and without perturbations: for the former the Lyapunov function as well as the controlled angle x_3 are shown in Fig. 5; for the latter, the plant is perturbed at $t = 5$ seconds and initialized at $x(0) = 0$. Figures 6 and 7 display the state x_3 as well as the control input $u(t)$ behaviour, respectively. It can be noticed that the state x_3 leaves the origin when the perturbation appears, nevertheless it is driven back to equilibrium point $x = 0$. Moreover, the bounds $|x_3| = 15^\circ$ and $|u(t)| = 22$ are not surpassed.

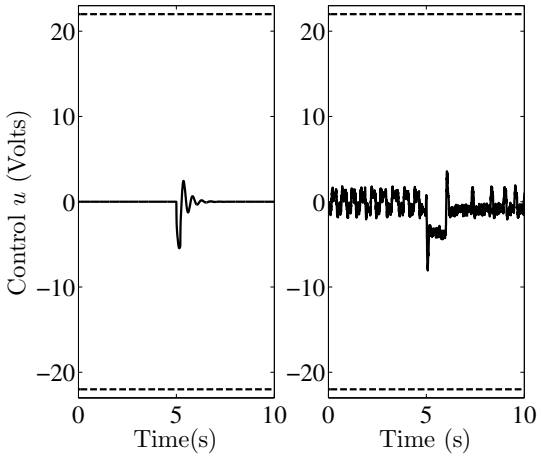

Fig. 7. Control law u (solid-line) and its upper- and lower-bound (dashed-lines) of perturbed system (6-nonlinearity case): simulation (left), real-time (right)

Note that, the 5-nonlinearity case overcomes the 6-nonlinearity one, that is: a) in the presence of perturbations, the amplitude of the angle x_3 is lower than the 6-nonlinearity case; b) the angle x_3 returns faster to the origin (see figures 4 and 6); c) the computational complexity of the 5-nonlinearity case is $\mathcal{N}_v = 138$, $\mathcal{N}_L = 4263$, while for the 6-nonlinearity case is $\mathcal{N}_v = 266$ and $\mathcal{N}_L = 16711$.

5. CONCLUSIONS

An LMI-based nonlinear control methodology has been presented and applied to the Furuta pendulum, both in simulation and real-time. Two cases have been addressed: 5 and 6 nonlinearities. The results show that considering 5 nonlinearities instead of 6 provides a better performance. Also, the use of 6 nonlinearities requires more computational effort, which may lead to numerical problems. The proposed approach has been successfully tested under perturbations; it is based on a variety of exact convex representations of the model nonlinearities which leads to LMIs when combined with the direct Lyapunov method. It has been shown that LMI framework allows performance measures to be straightforwardly incorporated in the controller design.

REFERENCES

Angeli, D. (2001). Almost global stabilization of the inverted pendulum via continuous state feedback. *Automatica*, 37(7), 1103–1108.

Begovich, O., Sanchez, E.N., and Maldonado, M. (2002). Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an underactuated robot. *IEEE Transactions on Control Systems Technology*, 10(1), 14–20.

Boyd, S., Ghaoui, L.E., Feron, E., and Belakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15. SIAM: Studies In Applied Mathematics, Philadelphia, USA.

Duan, G. and Yu, H. (2013). *LMIs in Control Systems: Analysis, Design and Applications*. CRC Press, Boca Raton, Florida.

Farwig, M., Zu, H., and Unbehauen, H. (1990). Discrete computer control of a triple-inverted pendulum. *Optimal Control Applications and Methods*, 11(2), 157–171.

Furuta, K., Kajiwara, H., and Kosuge, K. (1980). Digital control of a double inverted pendulum on an inclined rail. *International Journal of control*, 32(5), 907–924.

Furuta, K., Yamakita, M., and Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4), 263–269.

Gahinet, P., Nemirovski, A., Laub, A.J., and Chilali, M. (1995). *LMI Control Toolbox*. Math Works, Natick, USA.

Isidori, A. (1995). *Nonlinear Control Systems*. Springer, London, 3 edition.

Khalil, H. (2002). *Nonlinear Systems*. Prentice Hall, New Jersey, USA, 3 edition.

Khalil, H. (2014). *Nonlinear Control*. Prentice Hall, New Jersey, USA.

Lofberg, J. (2004). YALMIP : a toolbox for modeling and optimization in matlab. In *2004 IEEE International Symposium on Computer Aided Control Systems Design*, 284–289.

Ogata, K. (2001). *Modern control engineering*. Prentice Hall PTR, NJ, USA.

Ortega, R., Loría, J., Nicklasson, P., and Sira-Ramirez, H. (1998). *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*. Springer, London.

Quanser, I. (2006). *Mechatronics Control Kit User's Manual (Instructor)*. Mathworks, inc, Natick, MA.

Scherer, C. (2004). *Linear Matrix Inequalities in Control Theory*. Delf University, Delf, The Netherlands.

Spong, M.W., Corke, P., and Lozano, R. (2001). Nonlinear control of the reaction wheel pendulum. *Automatica*, 37(11), 1845–1851.

Sturm, J. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12, 625–653.

Tanaka, K. and Wang, H. (2001). *Fuzzy Control Systems Design and Analysis: A linear matrix inequality approach*. John Wiley & Sons, New York.

Taniguchi, T., Tanaka, K., and Wang, H. (2001). Model construction, rule reduction and robust compensation for generalized form of Takagi-Sugeno fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 9(2), 525–537.

Tuan, H., Apkarian, P., Narikiyo, T., and Yamamoto, Y. (2001). Parameterized linear matrix inequality techniques in fuzzy control system design. *IEEE Transactions on Fuzzy Systems*, 9(2), 324–332.

Utkin, V. (1992). *Sliding Modes in Control and Optimization*, volume 116. Springer, Berlin.