

# Anti-windup control of nonlinear cascade systems with particle swarm optimization parameter tuning

Fernando Serrano

Central American Technical University (UNITEC)  
Zona Jacaleapa, Tegucigalpa, Honduras  
serranofer@eclipseo.eu

Josep M. Rossell

Dept. of Mathematics. Univ. Politècnica de Catalunya (UPC)  
Avda. Bases de Manresa 61-73, 08242, Manresa, Spain  
josep.maria.rossell@upc.edu

**Abstract**—Assuming that many physical models can be decoupled, an anti-windup control scheme for nonlinear cascade systems is proposed. Taking into account that saturation appears frequently, in order to overcome this difficulty, an efficient control approach is developed. The paper is divided into two parts. Firstly, the design of a cascade control system with dynamic controllers in the inner and outer loops, considering the closed-loop stability in the controllers design with a suitable anti-windup compensator. Secondly, a PID cascade controller design in the inner and outer loop is presented, when the parameter tuning in both control schemes is done by particle swarm optimization (PSO). However, in this case, the implementation of an anti-windup compensator is not needed. Apart from the theoretical background, two numerical examples are shown to corroborate the provided results.

## I. INTRODUCTION

Cascade control systems have been investigated since several decades ago. In the SISO linear case, as it is known, the controllers are tuned in sequence, first by tuning the inner loop and then the outer loop. Usually, the kind of controllers implemented are proportional-integral-derivative (PID). In recent years, the research about cascade control systems has been extended to the nonlinear case, considering that many physical systems such as mechanical, electrical, power systems and chemical systems can be controlled and stabilized by means of this approach. The design is possible because a decoupled system can be divided into an inner and an outer loop, improving the performance in comparison with single loop control techniques. In the literature, the research about this topic is limited but an example can be found in [1] where a cascade control system is designed for the stabilization of underactuated mechanical systems. Although the anti-windup control problem for cascade control systems has not been investigated extensively, there are interesting results in single loop anti-windup design. In [2], an anti-windup control design is developed for the control of Takagi-Sugeno systems and a reliable state feedback control of Takagi-Sugeno fuzzy systems with sensor faults can be seen in [3]. A control scheme for disturbance observer systems is provided in [4], dealing with the saturation torque. Other theoretical and applied studies have been presented in [5], where the results are implemented in single loop linear systems and the gain matrices are

computed by using linear matrix inequalities (LMI's). Based on a linear approach, an anti-windup control scheme for an underwater vehicle is given in [6] and an anti-windup approach for nonlinear systems can be found in [7]. Other interesting works related to this topic are given in [8], [9], [10].

In this paper, an anti-windup control scheme is proposed for the stabilization of cascade nonlinear systems, which is developed in two parts. The first one is a dynamic controller implemented in the inner and outer loop. The closed-loop stability of the system is based on the theory stability of Lyapunov [11]. An anti-windup compensator is designed in order to reduce the unwanted effects of windup such as a poor performance or even instability. The second part is done by implementing PID controllers in the inner and outer loop but now without anti-windup compensation. In the first and second part of this study the gain matrices are tuned by particle swarm optimization [12], [13], [14], [15].

The paper is organized as follows: In Section II, the design of an anti-windup control scheme for cascade control systems, implementing dynamic controllers in the inner and outer loop, is developed. In Section III, a PID cascade control system design is presented by considering the input saturation but without anti-windup compensator. In Section IV, a PSO algorithm is supplied in order to tune the gain matrices for both approaches. Two numerical examples are given in Section V and the conclusions can be found in Section VI.

## II. ANTI-WINDUP CASCADE DYNAMIC CONTROLLER DESIGN

This section is devoted to design an anti-windup controller for nonlinear cascade systems. This strategy implements dynamic controllers in the inner and outer loops with gain matrices that help to improve the system performance. The controllers are tuned, as explained in Section IV, by a particle swarm optimization algorithm to reduce the integral square error, i.e. the difference between the reference variable and the output of the outer system. The same applies to the inner system. The main idea of this first approach is to design an appropriate anti-windup compensator to deal with the unwanted effects when saturation appears in the inner loop. Even when the gain matrices are tuned by a PSO algorithm, the closed-loop stability of the inner and the overall systems

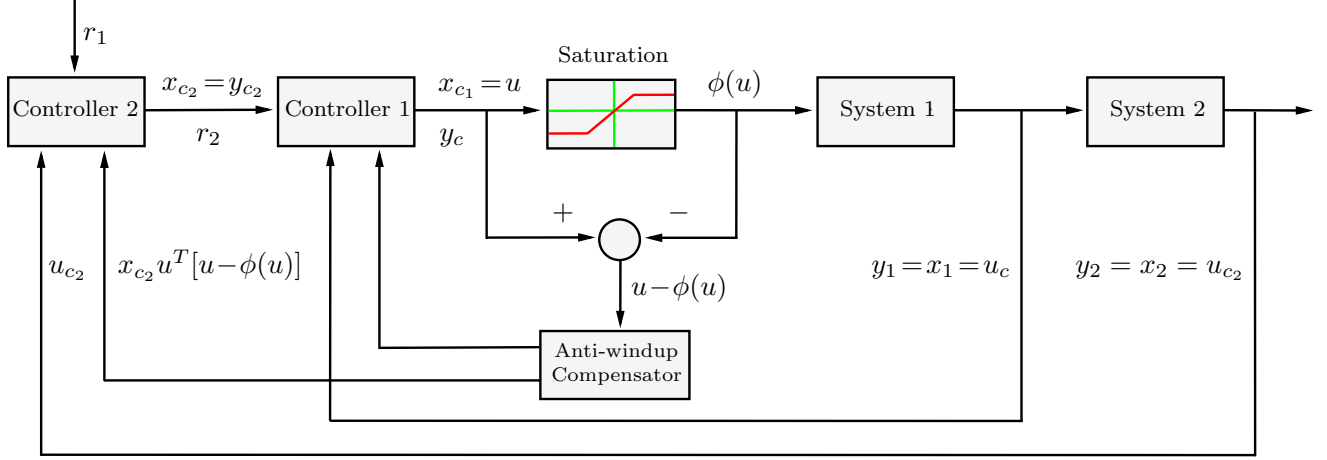


Fig. 1. Cascade anti-windup dynamic control scheme

are proved by the method of Lyapunov [11]. The obtained results are compared with the approaches given in [16], [17].

#### A. Inner loop dynamic controller design

Consider the cascade control dynamic scheme shown in Fig. 1. The inner loop system (system 1) is formed by

$$\begin{aligned} \dot{x}_1(t) &= -A x_1(t) + f_1(x_1(t)) + \phi(u(t)), \\ y_1(t) &= x_1(t) = u_c(t), \end{aligned} \quad (1)$$

where  $A \in \mathbb{R}^{n \times n}$  is an appropriate positive definite matrix due to the inner loop system is minimum phase in order to facilitate the particle swarm optimization parameter tuning;  $x_1(t) \in \mathbb{R}^n$  is the state vector;  $f_1(x_1(t))$  a nonlinear vector function;  $\phi(\cdot)$  the saturation nonlinearity;  $u(t) \in \mathbb{R}^n$  the input vector;  $y_1(t) \in \mathbb{R}^n$  the controller output and  $u_c(t) \in \mathbb{R}^n$  is the controller input vector. The inner loop controller is given by

$$\begin{aligned} \dot{x}_{c_1}(t) &= -K x_{c_1}(t) + r_2(t) - [u(t) - \phi(u(t))] + u_c(t), \\ y_c(t) &= x_{c_1}(t) = u(t), \end{aligned} \quad (2)$$

where  $K \in \mathbb{R}^{n \times n}$  is a positive definite gain matrix with a negative sign to make the closed-loop system of minimum phase type;  $x_{c_1}(t) \in \mathbb{R}^n$  is the controller state vector;  $r_2(t) \in \mathbb{R}^n$  the reference vector and  $u(t) - \phi(u(t))$  is the compensation term.

Before proving the closed-loop stability of the inner loop, the definition of sector condition is needed [1].

*Definition 1:* The sector condition for the saturation nonlinearity is given by

$$u^T(t)[u(t) - \phi(u(t))] > 0. \quad (3)$$

In the following theorem, the closed-loop stability of the inner loop is proved in order to obtain a stable controller by considering that the gain matrix  $K$  is tuned by means of a PSO algorithm, which is detailed in Section IV.

*Theorem 1:* There exists a gain matrix  $K$  for the controller (2) such that the closed-loop of the inner loop is stable.

*Proof:* Consider the following storage function [11]:

$$V(x_1(t), x_{c_1}(t)) = V_s(x_1(t)) + \sigma V_{sc}(x_{c_1}(t)), \quad (4)$$

with  $\sigma > 0$  and where

$$\begin{aligned} V_s(x_1(t)) &= \frac{1}{2} x_1^T(t) x_1(t), \\ V_{sc}(x_{c_1}(t)) &= \frac{1}{2} x_{c_1}^T(t) x_{c_1}(t) \end{aligned} \quad (5)$$

and the auxiliary input variable

$$v(t) = -f_1(x_1(t)) - \phi(u(t)). \quad (6)$$

Then, the system (1) becomes

$$\dot{x}_1(t) = -A x_1(t) - v(t) \quad (7)$$

and defining the input variable

$$w(t) = -r_2(t) - u_c(t), \quad (8)$$

the system (2) can be written as

$$\dot{x}_{c_1}(t) = -K x_{c_1}(t) - w(t) - [u(t) - \phi(u(t))]. \quad (9)$$

Now, obtaining the first derivative of (4) along (7) and (9), yields

$$\begin{aligned} \dot{V}(x_1(t), x_{c_1}(t)) &= -x_1^T(t) A x_1(t) - x_1^T(t) v(t) \\ &\quad - \sigma [x_{c_1}^T(t) K x_{c_1}(t) + x_{c_1}^T(t) w(t) + x_{c_1}^T(t) [u(t) - \phi(u(t))]]. \end{aligned} \quad (10)$$

From (1) and (2),  $x_1(t) = y_1(t)$  and  $x_{c_1}(t) = u(t)$ . Then, the equation (10) becomes in

$$\begin{aligned} \dot{V}(x_1(t), x_{c_1}(t)) &= -y_1^T(t) A y_1(t) - y_1^T(t) v(t) \\ &\quad - \sigma [y_c^T(t) K y_c(t) + y_c^T(t) w(t) + u^T(t) [u(t) - \phi(u(t))]] \end{aligned} \quad (11)$$

and using Definition 1 in (11) and considering that the system is zero state observable [7], i.e.  $y_i(t) = 0$  implies  $u_{c_i}(t) = 0$  and  $x_i(t) = 0$ , for  $i = 1, 2$ , we obtain

$$\dot{V}(x_1(t), x_{c_1}(t)) \leq 0 \quad (12)$$

and the theorem is proved [11].

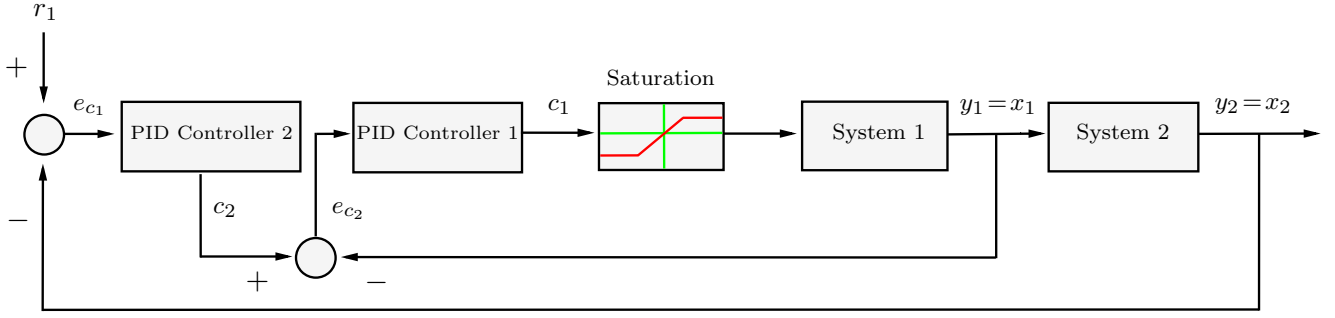


Fig. 2. Cascade PID control scheme

### B. Outer loop and overall dynamic controller design

Before deriving the outer loop controller and proving the overall closed-loop stability, it is necessary to make the following change of variables:

$$\begin{aligned} \bar{x}(t) &= [x_1^T(t), x_{c_1}^T(t)]^T, \\ \bar{u}(t) &= [u^T(t), \phi^T(u(t)), r_2^T(t), u_c^T(t)]^T. \end{aligned} \quad (13)$$

The equivalent closed-loop system is

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{f}(\bar{x}(t)) + \bar{B}\bar{u}(t), \quad (14)$$

with

$$\begin{aligned} \bar{A} &= \begin{bmatrix} -A & 0_n \\ 0_n & -K \end{bmatrix}, \quad \bar{f}(\bar{x}(t)) = \begin{bmatrix} f_1(x_1(t)) \\ 0 \end{bmatrix}, \\ \bar{B} &= \begin{bmatrix} 0_n & I_n & 0_n & 0_n \\ -I_n & I_n & I_n & I_n \end{bmatrix}, \end{aligned} \quad (15)$$

where  $I_n$  and  $0_n$  are the identity and the zero matrix of appropriate dimensions, respectively. Now, consider the outer loop dynamic equation (system 2) given by

$$\dot{x}_2(t) = -A_2 x_2(t) + f_2(x_2(t)) + u_2(t), \quad (16)$$

where  $A_2 \in \mathbb{R}^{m \times m}$  is a positive definite matrix;  $x_2(t) \in \mathbb{R}^m$  is the state vector;  $f_2(x_2(t))$  the nonlinearity vector and  $u_2(t) \in \mathbb{R}^m$  is the input vector. The overall closed-loop system is obtained by selecting the augmented vectors

$$\begin{aligned} \tilde{x}(t) &= [x_1^T(t), x_{c_1}^T(t), x_2^T(t)]^T, \\ \tilde{u}(t) &= [u^T(t), \phi^T(u(t)), r_2^T(t), u_c^T(t), u_2^T(t)]^T. \end{aligned} \quad (17)$$

Then,

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{f}(\tilde{x}(t)) + \tilde{B}\tilde{u}(t), \\ \tilde{y}(t) &= x_2(t), \end{aligned} \quad (18)$$

where the output is given by

$$\tilde{y}(t) = \begin{bmatrix} 0_n & 0_n & I_m \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_{c_1}(t) \\ x_2(t) \end{bmatrix} = x_2(t) \quad (19)$$

with  $x_2(t) = u_{c_2}(t) \in \mathbb{R}^m$  the controller input,

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} -A & 0_n & 0_m \\ 0_n & -K & 0_m \\ 0_n & 0_n & -A_2 \end{bmatrix}, \quad \tilde{f}(\tilde{x}(t)) = \begin{bmatrix} f_1(x_1(t)) \\ 0 \\ f_2(x_2(t)) \end{bmatrix}, \\ \tilde{B} &= \begin{bmatrix} 0_n & I_n & 0_n & 0_n & 0_m \\ -I_n & I_n & I_n & I_n & 0_m \\ 0_n & 0_n & 0_n & 0_n & I_m \end{bmatrix} \end{aligned} \quad (20)$$

and rewriting

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) - \tilde{v}_2(t) \quad (21)$$

with

$$\tilde{v}_2(t) = -\tilde{f}(\tilde{x}(t)) - \tilde{B}\tilde{u}(t). \quad (22)$$

Consider the following outer loop dynamic controller:

$$\begin{aligned} \dot{x}_{c_2}(t) &= -K_2 x_{c_2}(t) - r_1(t) - x_{c_2}(t) \tilde{z}^T(t) [u(t) - \phi(u(t))] - u_{c_2}(t) \\ &= -K_2 x_{c_2}(t) - v_2(t) - x_{c_2}(t) \tilde{z}^T(t) [u(t) - \phi(u(t))], \end{aligned} \quad (23)$$

where  $x_{c_2}(t) \in \mathbb{R}^m$  is the dynamic controller state vector,  $K_2 \in \mathbb{R}^{m \times m}$  the controller gain matrix,  $r_1(t) \in \mathbb{R}^m$  the reference vector and defining

$$\begin{aligned} v_2(t) &= r_1(t) + u_{c_2}(t), \\ y_{c_2}(t) &= x_{c_2}(t), \end{aligned} \quad (24)$$

where  $y_{c_2}(t) \in \mathbb{R}^m$  is the controller output, together with an extra output

$$\tilde{z}(t) = \begin{bmatrix} 0_n & I_n & 0_m \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_{c_1}(t) \\ x_2(t) \end{bmatrix} = x_{c_1}(t). \quad (25)$$

Then, the following theorem can be stated.

**Theorem 2:** There exists a gain matrix  $K_2$  for the controller (23) such that the closed-loop of the outer loop (overall system) is stable.

*Proof:* Consider the storage function for the overall closed-loop system

$$V(\tilde{x}(t), x_{c_2}(t)) = V_s(\tilde{x}(t)) + \sigma V_{sc}(x_{c_2}(t)), \quad (26)$$

with  $\sigma > 0$  and where

$$\begin{aligned} V_s(\tilde{x}(t)) &= \frac{1}{2} \tilde{x}^T(t) \tilde{x}(t), \\ V_{sc}(x_{c_2}(t)) &= \frac{1}{2} x_{c_2}^T(t) x_{c_2}(t). \end{aligned} \quad (27)$$

Now, deriving (26) along the trajectory given in (21) and (23), we obtain

$$\begin{aligned} \dot{V}(\tilde{x}(t), x_{c_2}(t)) &= \tilde{x}^T(t) \tilde{A} \tilde{x}(t) - \tilde{x}^T(t) \tilde{v}_2(t) \\ &- \sigma x_{c_2}^T(t) [K_2 x_{c_2}(t) + v_2(t) + x_{c_2}(t) u^T(t) [u(t) - \phi(u(t))]]. \end{aligned} \quad (28)$$

Then, from Definition 1, and considering that the system is zero state observable,

$$\dot{V}(\tilde{x}(t), x_{c_2}(t)) \leq 0 \quad (29)$$

is obtained and the closed-loop stability of the outer loop is ensured.

### III. PID CASCADE CONTROL SYSTEM DESIGN

A PID cascade control system is formed by two parts: An inner loop PID controller [13] and an outer loop PID controller (see Fig. 2). In this case, it is not necessary to implement an anti-windup compensator because the gain matrices are tuned by the particle swarm optimization routine shown in the next section.

The PID controllers for the inner and outer loop are given by the following equations:

$$c_1(t) = K_{p_1} e_{c_1}(t) + K_{i_1} \int_0^t e_{c_1}(t) dt + K_{d_1} \dot{e}_{c_1}(t), \quad (30)$$

$$c_2(t) = K_{p_2} e_{c_2}(t) + K_{i_2} \int_0^t e_{c_2}(t) dt + K_{d_2} \dot{e}_{c_2}(t),$$

where  $c_1(t) \in \mathfrak{R}^n$  is the output controller 1;  $c_2(t) \in \mathfrak{R}^m$  the output controller 2;  $e_{c_1}(t) = r_1(t) - y_2(t)$  the error variable for the controller 1;  $e_{c_2}(t) = c_2(t) - y_1(t)$  the error variable for the controller 2;  $K_{p_1}, K_{i_1}, K_{d_1} \in \mathfrak{R}^n$  are the proportional, integral and derivative gain matrices, respectively, for the controller 1 and  $K_{p_2}, K_{i_2}, K_{d_2} \in \mathfrak{R}^m$  for the controller 2.

### IV. PARTICLE SWARM OPTIMIZATION ROUTINE FOR CASCADE ANTI-WINDUP CONTROLLER DESIGN

Particle swarm optimization routines have been recently implemented for the parameter tuning for PID's and other controllers [12]–[15], [18]. In this study, the first step is to determine an anti-windup scheme for a cascade dynamic controller in order to find an appropriate controller and compensator ensuring the closed-loop overall stability. The gain matrices are found by using a PSO algorithm to minimize the integral squared error of the overall system [18], where

$$F_i(e_i) = \sum_{j=1}^m e_i^2(j) \Delta_j \quad (31)$$

is the objective function that minimizes the error  $e_i(n) = r_1(n) - y_2(n)$  with the time difference  $\Delta_j$ . The PSO algorithm that allows us to find the gain matrices  $K, K_2$  for the dynamic controller and  $K_{p_1}, K_{i_1}, K_{d_1}, K_{p_2}, K_{i_2}, K_{d_2}$  for the PID controller scheme is given by

while ( $gbest > r_1$  and  $gbest < r_2 \dots gbest < r_n$  and  $j < 100000$ )  
for ( $int\ i=0; i < paramnum; i++$ )

$V[i] = V[i] + c_1(rand())(pbest[i] - c_2(rand()))(gbest - X[i]);$

$X[i] = X[i] + V[i];$   
 $F = Objectivefunction(X)$   
if ( $F[i] \leq Fpbest[i]$ )  
 $pbest[i] = X[i];$   
 $Fpbest[i] = F[i];$   
if ( $F[1] \leq Fgbest$ )  
 $gbest = X[i];$   
 $Fgbest = F[1];$

where  $X$  is the particle position for the gain matrices component;  $V$  is the particle velocity;  $pbest$  and  $gbest$  are the best particle positions and  $Fgbest$  is the final result obtained by the objective function.

### V. NUMERICAL EXAMPLES

The following systems are implemented in two examples:

$$\begin{aligned} \dot{x}_{11}(t) &= -x_{11}(t) - x_{12}(t) + x_{11}^3(t) + x_{11}(t) x_{12}^2(t) + u_1(t), \\ \dot{x}_{12}(t) &= -x_{12}(t) + 2x_{11}(t) + x_{12}^4(t) x_{11}^2(t) + u_2(t), \end{aligned} \quad (32)$$

$$\begin{aligned} \dot{x}_{21}(t) &= x_{21}^3(t) + u_1(t), \\ \dot{x}_{22}(t) &= x_{22}^3(t) - x_{21}(t) x_{22}(t) + u_2(t), \end{aligned} \quad (33)$$

where (32) and (33) will be the systems (1) and (2) for the example 1, and

$$\begin{aligned} \dot{x}_{11}(t) &= -x_{11}(t) - 0.0001 x_{11}^2(t) + u_1(t), \\ \dot{x}_{12}(t) &= -x_{12}(t) - 0.0001 x_{12}^2(t) + u_2(t), \end{aligned} \quad (34)$$

$$\begin{aligned} \dot{x}_{21}(t) &= -x_{21}(t) - 0.0001 x_{21}^2(t) + u_1(t), \\ \dot{x}_{22}(t) &= -x_{22}(t) - 0.0001 x_{22}^2(t) + u_2(t), \end{aligned} \quad (35)$$

$$\phi(x) = \begin{cases} -19800 & \text{for } x \leq -19800 \\ x & \text{for } -19800 < x < 19800 \\ 19800 & \text{for } x \geq 19800 \end{cases} \quad (36)$$

where (34) and (35) will be the systems (1) and (2) for the example 2, with the saturation function given in (36).

#### A. Example 1: Dynamic controller experiment

In this subsection, a numerical example to test the dynamic controller design is shown. The obtained results are compared with the results presented in [16], [17], considering that the strategies evinced in both studies are used in a cascade controller configuration. Fig. 3 and Fig. 4 depict the trajectories obtained by the variables  $x_{11}(t)$  and  $x_{21}(t)$ , where the latest variable is the output of the overall closed-loop system when a reference is used to reach the origin or the equilibrium point. Note that these variables reach more efficiently the equilibrium point in comparison with the strategies given in [16] and [17], with less overshoot and faster response. The same occurs for the variable  $x_{22}(t)$  shown in Fig. 5. In Fig. 6 and Fig. 7 the controller inputs  $x_{c_{11}}(t)$  and  $x_{c_{22}}(t)$  are presented and the control effort generated by the control strategy is also smaller than the obtained in [16] and [17].

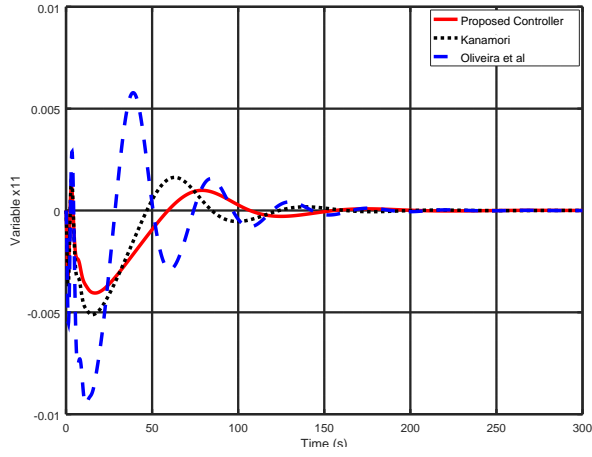


Fig. 3. Variable  $x_{11}(t)$

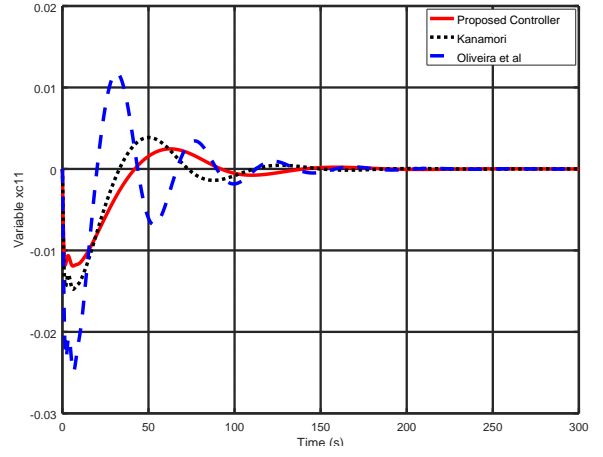


Fig. 6. Control input  $x_{c11}(t)$

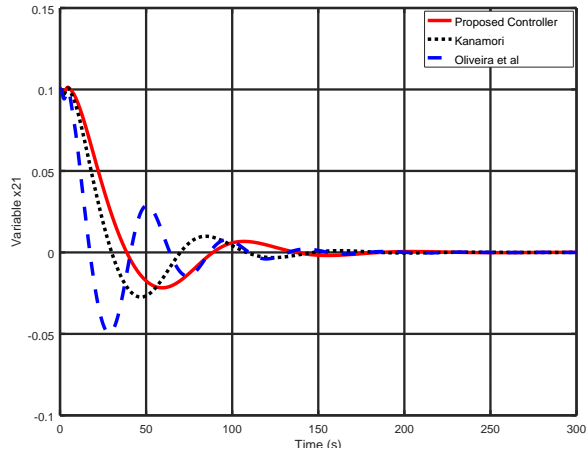


Fig. 4. Variable  $x_{21}(t)$

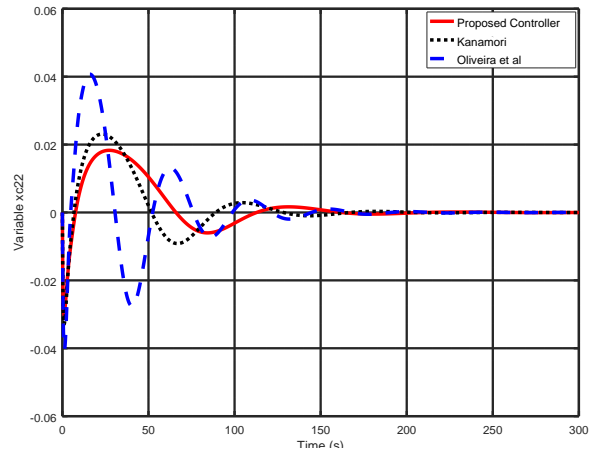


Fig. 7. Control input  $x_{c22}(t)$

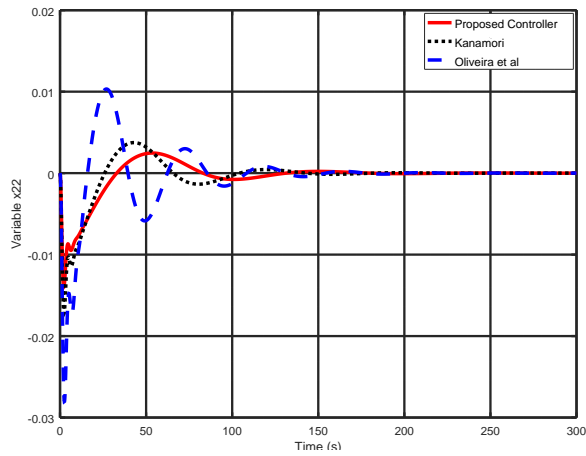


Fig. 5. Variable  $x_{22}(t)$

### B. Example 2: PID controller experiment

In this example, the PID controller gains are tuned by a PSO algorithm and the obtained results are compared with the approaches shown in [19], [20], with the origin as the equilibrium point. The control approaches given in [19], [20] have been modified to operate in a cascade closed-loop configuration. The variable  $x_{22}(t)$  is depicted in Fig. 8 and the desired final value has less oscillations and faster response in our case. Finally, in Fig. 9 and Fig. 10 the respective PID controller outputs  $c_{11}(t)$  and  $c_{22}(t)$  are shown and the control effort is smaller and with less oscillations, even when saturation is found in the input.

## VI. CONCLUSIONS

Two anti-windup schemes for nonlinear cascade systems have been proposed and, when input saturation appears, the system performance is improved. The results represent a contribution in some physical systems such as mechanical, aeronautical, electrical, power and energy systems, considering that saturation is a common phenomenon affecting them.

## REFERENCES

- [1] N. Mehdi, M. Rehan, F. M. Malik, A. I. Bhatti, and M. Tufail, "A novel anti-windup framework for cascade control systems: An application to underactuated mechanical systems," *ISA Transactions*, vol. 53, no. 3, pp. 802 – 815, 2014.
- [2] A. Nguyen, A. Dequidt, and M. Dambrine, "Anti-windup based dynamic output feedback controller design with performance consideration for constrained Takagi Sugeno systems," *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 76 – 83, 2015.
- [3] J. Dong and G-H. Yang, "Reliable state feedback control of T-S fuzzy systems with sensor faults," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 2, pp. 421 – 433, 2015.
- [4] X. Gao, S. Komada, and T. Hori, "A wind-up restraint control of disturbance observer system for saturation of actuator torque," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, 1999, pp. 84–88.
- [5] J. D. Silva and S.Tarbouriech, "Anti-windup design with guaranteed regions of stability: an LMI based approach," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 5, 2003, pp. 4451–4456.
- [6] J. P. Folcher, "LMI based anti-windup control for an underwater robot with propellers saturations," in *Proceedings of the IEEE International Conference on Control Applications*, vol. 1, 2004, pp. 32–37.
- [7] M. Z. Oliveira, J.M.G. Da Silva, D. Coutinho, and S. Tarbouriech, "Anti-windup design for a class of multivariable nonlinear control systems: An LMI based approach," in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4797–4802.
- [8] D. Zhai and L. An and J. Li and Q. Zhang, "Fault detection for stochastic parameter-varying Markovian jump systems with application to networked control systems," *Applied Mathematical Modelling*, vol. 40, no. 3, pp. 2368 – 2383, 2016.
- [9] D. Zhai and L. An and J. Li and Q. Zhang, "Adaptive fuzzy fault-tolerant control with guaranteed tracking performance for nonlinear strict-feedback systems," *Fuzzy Set and Systems*, vol. 302, pp. 80 – 100, 2016.
- [10] D. Zhai and A-Y. Lu and J-H. Li and Q-L. Zhang, "Simultaneous fault detection and control for switched linear systems with mode-dependent average dwell-time," *Applied Mathematics and Computation*, vol. 273, pp. 767 – 792, 2016.
- [11] W. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov Based Approach*. Princeton Press, 2008.
- [12] H. Gao and W. Xu, "Particle swarm algorithm with hybrid mutation strategy," *Applied Soft Computing*, vol. 11, no. 8, pp. 5129 – 5142, 2011.
- [13] M. I. Menhas, L. Wang, M. Fei, and H. Pan, "Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4390 – 4401, 2012.
- [14] L. Wang, X. Fu, Y. Mao, M.I. Menhas, and M. Fei, "A novel modified binary differential evolution algorithm and its applications," *Neurocomputing*, vol. 98, pp. 55 – 75, 2012.
- [15] Y. Wang, B. Li, T.Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4515 – 4538, 2011.
- [16] M. Kanamori, "Anti-windup adaptive law for Euler–Lagrange systems with actuator saturation," *IFAC Proceedings*, vol. 45, no. 22, pp. 875 – 880, 2012.
- [17] M. Z. Oliveira, J. M. Gomes da Silva, D. F. Coutinho and S. Tarbouriech, "Anti-windup design for a class of nonlinear control systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13432 – 13437, 2011.
- [18] F. E. Serrano and M.A. Flores, "C++ library for fuzzy type-2 controller design with particle swarm optimization tuning," *IEEE CONCAPAN 2015, Tegucigalpa, Honduras*, 2015.
- [19] A. H. Tahoun, "Anti-windup adaptive PID control design for a class of uncertain chaotic systems with input saturation," *ISA Transactions*, vol. 66, pp. 176 – 184, 2017.
- [20] C. Q. Huang, X. F. Peng and J. P. Wang, "Robust nonlinear PID controllers for anti-windup design of robot manipulators with an uncertain Jacobian matrix," *Acta Automatica Sinica*, vol. 34, no. 9, pp. 1113 – 1121, 2008.

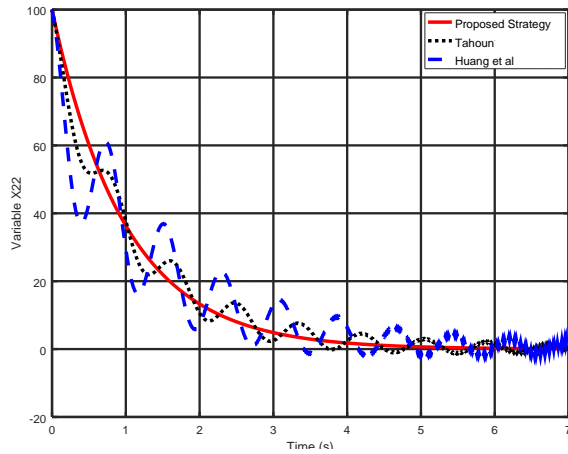


Fig. 8. Variable  $x_{22}(t)$

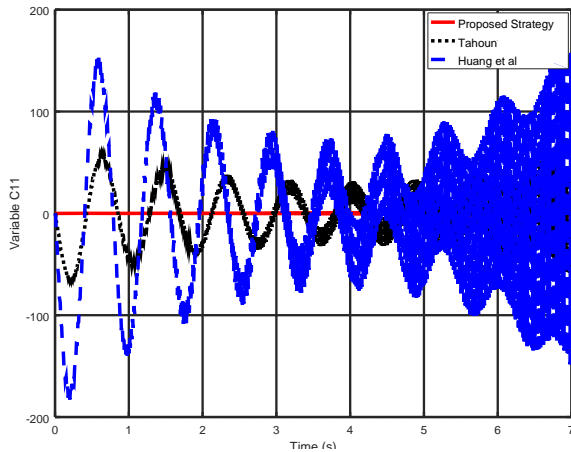


Fig. 9. Control variable  $c_{11}(t)$

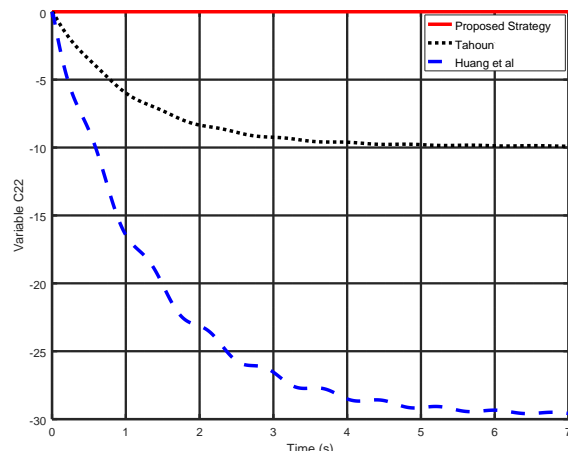


Fig. 10. Control variable  $c_{22}(t)$