Journal of ELECTRICAL ENGINEERING, VOL. 59, NO. 5, 2008, 254–259

EXPERIMENTAL COMPARISON OF SATURATED VELOCITY CONTROLLERS FOR DC MOTORS

Javier Moreno–Valenzuela *

This paper concerns the velocity control of direct current (DC) motors by considering that voltage input is saturated. A new control scheme to solve that problem is proposed. By using real-time experiments conducted in an industrial DC motor, we discuss the performance of the PI control in cascade with a saturation function, the PI control equipped with anti-windup and the new controller.

K e y w o r d s: velocity control; DC motor; voltage saturation; PI control; anti–windup

1 INTRODUCTION

Velocity regulation and velocity control of electrical motors have many applications in industrial processes and machine tools. The objective of velocity regulation is to maintain the velocity of the motor shaft to a given constant desired reference. In contrast, the goal of velocity control is to track accurately a time-varying desired velocity, see, *eg*, [7], [2], [13], and references therein. Let us notice that the design of stable velocity controllers is important in industrial robots, since they use DC motor endowed with an inner velocity feedback loop as the core of the control system [6], [5].

The proportional-integral (PI) control is the most used algorithm to regulate the velocity of motor drives. However, even when a tuning design to satisfy some desired performance, the physical limitations of the motor should be taken into account, since the current command is limited to a prescribed maximum value due to the converter protection, magnetic saturation, and motor overheating [4].

Actuator saturation can cause undesirable effects such as excessive overshoots, long settling times, and in some cases instability. In PI controllers, saturation can cause the integrator to drift to undesirable values so that the integrator will produce even larger control signals. In other words, actuator saturation and integrator may produce *windup*, which can lock the system in saturation. In process control, integrator windup is often referred to as reset windup since integral control is called reset control [3], [9].

Control techniques designed to reduce the effects from integrator windup are known as *anti–windup*. For PI controllers, a well-known technique is the back-calculation anti-windup scheme. As pointed out in [11], that method was first described by Fertik and Ross [8] and is mentioned in most introductory literature on the subject, *eg*, Åström and Wittenmark [2], Franklin *et al* [9] or Åström and Hägglund [1].

The purpose of this paper is to present a study on velocity control of DC motors subject to input voltage constraints.

Our theoretical contribution is:

- A new controller, which consists in using hyperbolic tangent functions in the proportional and integral part of the controller.

In addition, as a contribution in the field of the electrical engineering practice, we present:

- An experimental evaluation carried out in an industrial DC motor, which shows the performance of two well-known PI control structures and the new one.

This paper is organized as follows. Section 2 is devoted to the DC motor model and control objective. Sections 3 and 4 concerns the PI controller and PI plus anti-windup algorithm, respectively. In Section 5, the new controller is introduced. The experimental evaluation is described in Section 6, while some concluding remarks are drawn in Section 7.

2 MODEL AND CONTROL GOAL

Many industrial motor drives are equipped with a servo amplifier, which can operate in current mode, that is, with a current control loop implemented inside the servo amplifier, which makes that motor input current equal to the desired current. If the control gains of the current loop are tuned with high gains, the electrical dynamics becomes less dominant than the mechanical one.

* Centro de Investigación y Desarrollo de Tecnología Digital del IPN, CITEDI–IPN, Ave. del Parque 1310, Mesa de Otay, Tijuana, B.C., 22510, Mexico; moreno@citedi.mx

Work partially supported by CONACyT and SIP–IPN, Mexico.

ISSN 1335-3632

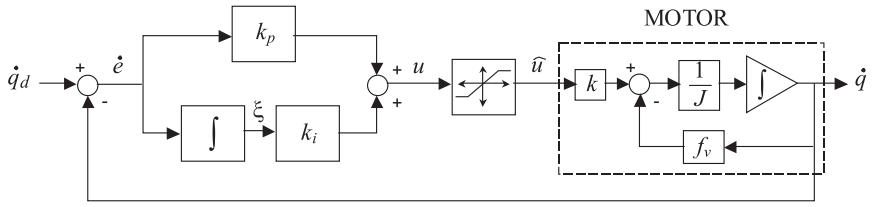

Fig. 1. Block diagram of the PI velocity control in cascade with a saturation function

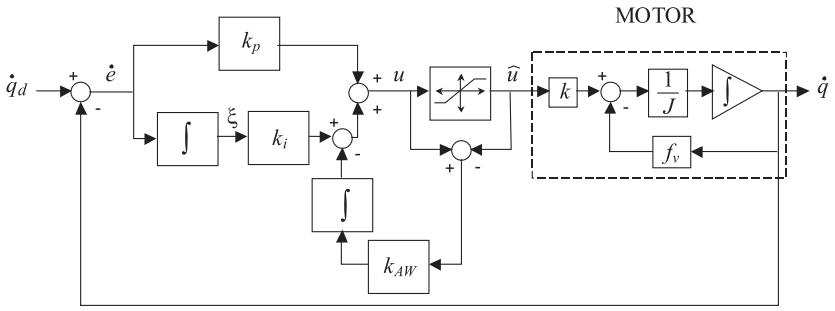

Fig. 2. Block diagram of the PI velocity controller equipped with the back-calculation anti-windup

Let us consider the dynamics of an armature-controlled DC motor given by [12, 16, 7, 15],

$$J\ddot{q} + f_v\dot{q} = k_m\, i\,, \tag{1}$$

$$L\frac{\mathrm{d}}{\mathrm{d}t}i + Ri7 = v - k_f\dot{q}\,, \tag{2}$$

where $\dot{q}(t)$ denotes the motor shaft velocity, J is the motor shaft inertia, f_v is the viscous friction coefficient, $i(t)$ is the armature current, L is the inductance, R is the resistance, k_m is the motor constant, k_f is the back electromotive force constant and $v(t)$ is the motor voltage input, which is provided by the servo amplifier usually containing a high-gain PI current controller, *ie*,

$$v(t) = k_p^*[i_d - i(t)] + k_i^*\int_0^t [i_d - i(t)]\mathrm{d}t\,, \tag{3}$$

where

$$i_d = k_{sa}\hat{u} \tag{4}$$

is the desired current, k_{sa} the amplifier gain, $\hat{u}$ to be defined later, $i(t)$ the actual current, k_p^* and k_i^* are the proportional and integral gains, respectively. Assuming that the current loop is implemented with high gains k_p^* and k_i^*, we can assume that the actual current $i(t)$ converges to the desired current i_d fast enough, so that

$$i(t) = i_d = k_{sa}\hat{u}\,. \tag{5}$$

Since practical current loops in motor drives are implemented with high gains, the equation (5) is valid, and the motor model (1)-(2), operated with a servo amplifier using an embedded high gain PI current loop collapses into the model

$$J\ddot{q} + f_v\dot{q} = k\hat{u}\,, \tag{6}$$

where

$$k = k_m k_{sa}\,,$$

is interpreted as the so-called motor constant, and $\hat{u}$ is new the voltage control input.

In order to prevent excessive warming and a failure situation, the new voltage input $\hat{u}$ is defined in terms of the set $\mathcal{V}$ defined as

$$\mathcal{V} = \{u \in \mathbb{R}\colon u^{\min} \le \hat{u} \le u^{\max}\}\,, \tag{7}$$

with $u^{\min} < 0$ the minimum voltage input and $u^{\max} > 0$ the maximum voltage input.

The velocity control problem consists in designing a voltage control input $\hat{u} \in \mathcal{V}$ so that the velocity error

$$\dot{e}(t) = \dot{q}_d - \dot{q}(t),$$

where $\dot{q}_d$ is a constant that specifies the desired velocity and $\dot{q}(t)$ is the motor shaft velocity, achieves the limit

$$\lim_{t\to\infty}\dot{e}(t) = \lim_{t\to\infty}[\dot{q}_d - \dot{q}(t)] = 0\,. \tag{8}$$

Let us define the saturation function

$$sat(u) = \begin{cases} u^{\max} & \text{for} \quad u > u^{\max}, \\ u & \text{for } u^{\min} \le u \le u^{\max}, \\ u^{\min} & \text{for} \quad u < u^{\min}, \end{cases} \tag{9}$$

which will be useful in the coming discussions.

3 PI CONTROL

The proportional-integral (PI) velocity controller is defined as [12, 16, 7, 15]

$$u = k_p\dot{e} + k_i\xi\,, \tag{10}$$

$$\dot{\xi} = \dot{e}\,, \tag{11}$$

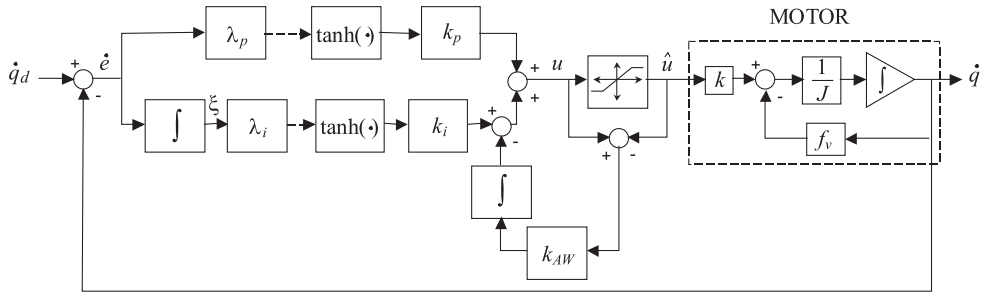

Fig. 3. Block diagram of the new saturated PI velocity controller

where $k_p > 0$ and $k_i > 0$ are the integral and proportional gains of the controller, respectively. In practice, it is common to saturate the PI control action generated by (10)-(11). Hence the effective control action $\hat{u}$ applied to the DC motor is given by

$$\hat{u} = sat(u)\,. \tag{12}$$

Figure 1 shows a block diagram of the the PI controller in cascade with a saturation function (10)-(12).

In presence of noise and load disturbances and if large integral gain is selected, the saturated PI controller (10)-(12) may have a performance with large settling time, overshoot, and eventually, the undesirable windup, *ie*, bouncing of the control signal $\hat{u}$ between its limits several times. In order to overcome those disadvantages, the addition of an anti-windup algorithm is frequently used.

4 PI CONTROL EQUIPPED WITH ANTI–WINDUP

Consider the PI velocity controller equipped with the back–calculation anti–windup [8] given by

$$u = k_p\dot{e} + k_i\xi - k_{AW}\rho\,, \tag{13}$$

$$\dot{\xi} = \dot{e}\,, \tag{14}$$

$$\dot{\rho} = u - \hat{u}\,, \tag{15}$$

with $k_{AW} > 0$, and

$$\hat{u} = sat(u)\,. \tag{16}$$

A block diagram of the controller (13)–(16) is presented in Fig. 2, being the saturation output $\hat{u}$ the effective control action applied to the DC motor.

To avoid windup, the integral of the difference between the control output u and the plant input $\hat{u}$ is fed back to the control output u through an adjustable gain k_{AW}. This will make the controller $u(t)$ tracks the mode where the control output $u(t)$ equals the $\hat{u}(t)$. The idea is that the control signal $u(t)$ be recalculated to a new value so that it stays into the voltage admissible limits. The gain k_{AW} governs the rate at which the performance of the linear PI controller is recovered [1]. Further discussion on the use and stability analysis of controllers equipped with back-calculation anti-windup is found in [11].

5 satPI CONTROL EQUIPPED WITH ANTI—WINDUP

We have discussed two solutions for the velocity regulation by using saturated voltage input. A new velocity controller is introduced as follows

$$u = k_p\tanh(\lambda_p\dot{e}) + k_i\tanh(\lambda_i\xi) - k_{AW}\rho\,, \tag{17}$$

$$\dot{\xi} = \dot{e}\,, \tag{18}$$

$$\dot{\rho} = u - \hat{u}\,, \tag{19}$$

where λ_p, λ_i, k_p, k_i and $k_{AW} > 0$, and

$$\hat{u} = sat(u)\,, \tag{20}$$

the effective control action. Figure 3 depicts a block diagram of the implementation of the controller (17)-(18), which in further reference will denoted as satPI control plus anti-windup.

The controller (17)–(20) incorporates the back-calculation anti-windup term. By using similar arguments that in [1], it is possible to show that the constant k_{AW} in the controller (17)-(18) governs the rate at which the controller operates in the admissible voltage input, *ie*,

$$\hat{u} = k_p\tanh(\lambda_p\dot{e}) + k_i\tanh(\lambda_i\xi)\,, \tag{21}$$

$$\dot{\xi} = \dot{e}\,, \tag{22}$$

which in closed-loop with the DC motor model (6) yields a stable system in the Lyapunov sense. A Lyapunov function is

$$V(\xi,\dot{e}) = \frac{J}{2}\dot{e}^2 + \Big[\frac{kk_i}{\lambda_i}\ln(\cosh(\lambda_i\xi)) - f_v\dot{q}_d\xi\Big] + C\,,$$

with some constant C. It follows to use the LaSalle's Theorem [10] to complete the proof. Since the aim of this paper is focused in the experimental evaluation rather than theoretical aspects, we leave out the details of the proof of convergence of the velocity error $\dot{e}(t)$.

It is worthwhile to notice that the use of saturation functions in the proportional and derivative parts of position motion controllers has been used previously. This kind of controllers have several advantages, like robustness to measurement noise and less consumption of energy than linear controllers. For a recent reference, see [14], where the trajectory tracking control of robot manipulators is studied.

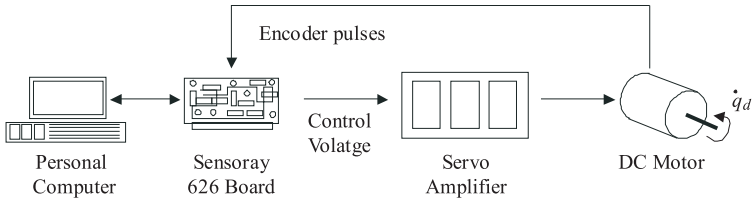

Fig. 4. Block diagram of the experimental system

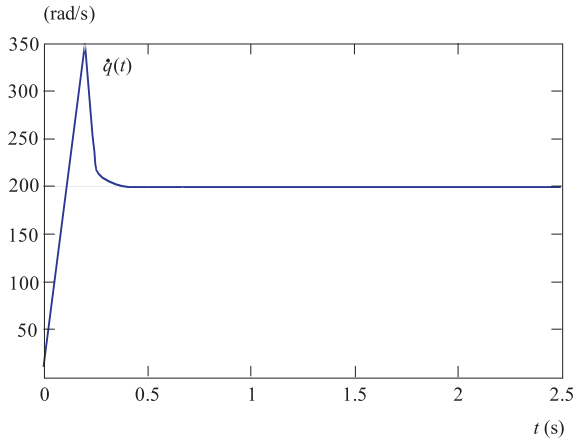

Fig. 5. PI control: velocity $\dot{q}(t)$

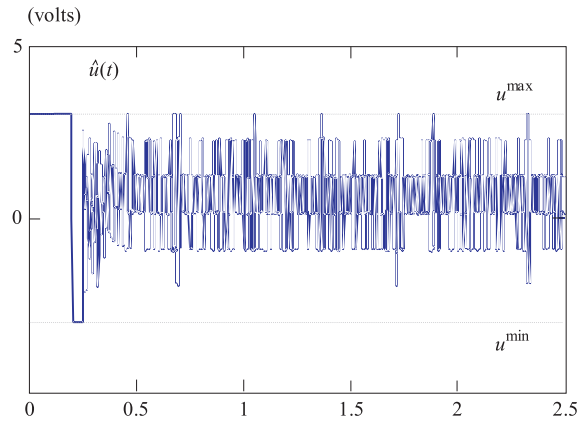

Fig. 6. PI control: effective control action $\hat{u}(t)$

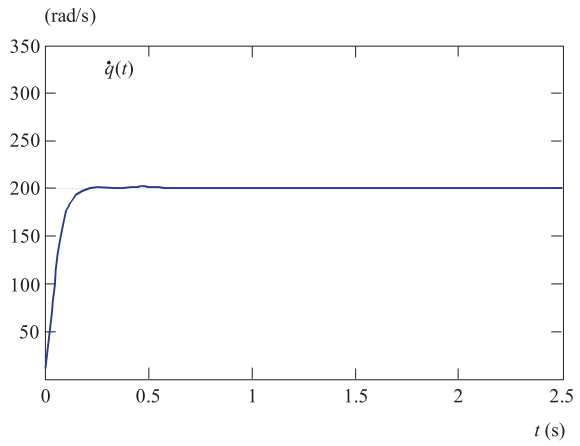

Fig. 7. PI control + AW: velocity $\dot{q}(t)$.

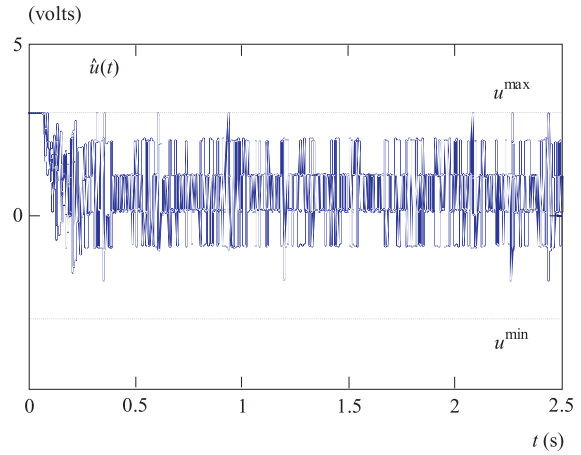

Fig. 8. PI control + AW: effective control action $\hat{u}(t)$

6 EXPERIMENTAL RESULTS

In order to assess the studied controllers, we have carried out real–time experiments in a DC motor *Pittman* 9236S009. To execute real-time experiments, the motor is operated through a PC, a servo amplifier *Advanced Motion Controls* model 16A20AC, and a data acquisition board *Sensoray* 626, which is used to read the optical quadrature encoder signal and transfer the control signal to the servo amplifier. Figure 4 depicts a block diagram of the experimental system.

Algorithms are performed with a sampling frequency of 1 kHz on the operative system *Windows XP*, *Matlab*, *Simulink* and *Real-Time Windows Target*. Concerning the proposed structure of real-time computer-aided control system, the following remarks are in order:

- *Matlab*: This is the primary software for our control system and can be described as an interactive environment for algorithms design and testing tool with graphical interface capabilities. This is the specific purpose software in which the proposed control system is designed for implementing software.
- *Simulink*: Simulink is a Matlab toolbox for the construction and simulation of block models to represent system dynamics, control architectures, etc. This programming environment was used to develop the controller model.

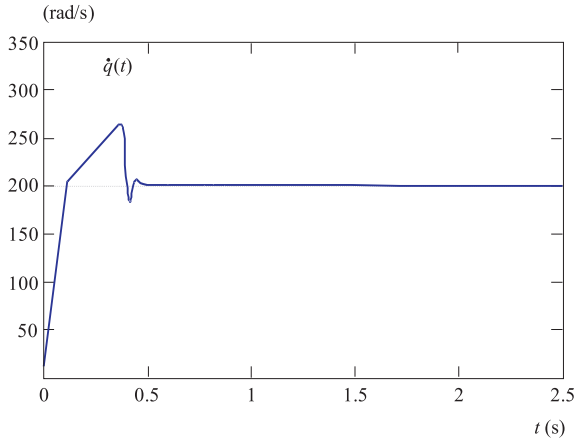

Fig. 9. satPI control + AW: velocity $\dot{q}(t)$

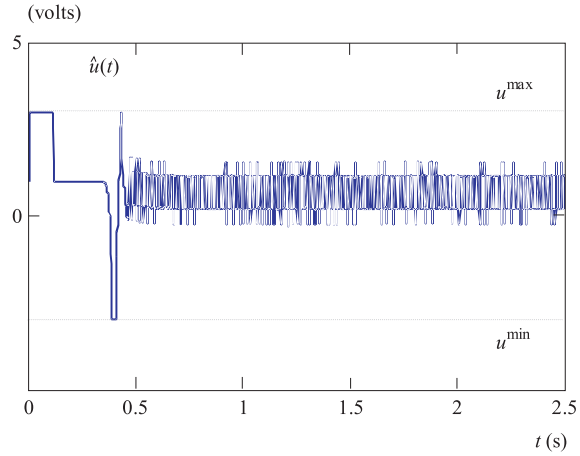

Fig. 10. satPI control + AW: effective control action $\hat{u}(t)$

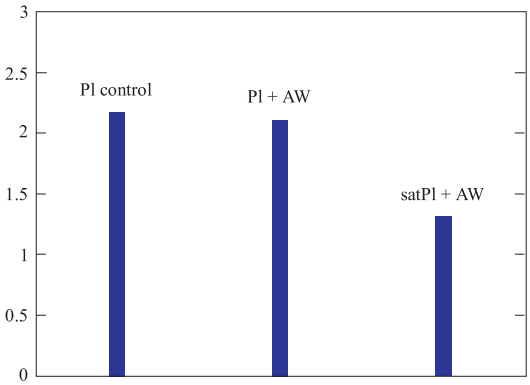

Fig. 11. Bar chart of $E = \int_{0.5}^{2.5} \hat{u}(\sigma)^2 d\sigma$ computed for the three tested controllers.

Table 1. Performance of the three controllers.

Index	PI	PI+AW	satPI+AW
Percent overshoot	74 %	0 %	32 %
$E = \int_{0.5}^{2.5} \hat{u}(\sigma)^2 d\sigma$	2.173	2.111	1.308

- *Real-Time Workshop*: This is a Simulink toolbox used to enable real-time execution. The built model must be executed in external mode of Simulink. When using this tool, the constructed model is compiled and converted into an executable file to be run in real-time by the operating system kernel.

Velocity information is obtained with the optical quadrature encoder signal by using the numerical differentiation, *ie*,

$$\dot{q}(kT) = \frac{q(kT) - q([k-1]T)}{T},$$

where $k = 1, 2, 3,$ *etc*, is the discrete time, $T = 0.001$ s is the sampling period, $q(kT)$ is the actual motor shaft angular position and $q([k-1]T)$ the previous one.

6.1 PI control

The first experiment consist in implementing the PI controller (10)–(12) by using the velocity reference

$$\dot{q}_d = 200 \text{ rad/s}, \tag{23}$$

the gains

$$\begin{aligned} k_p &= 1 \ \text{ Vs/rad}, \\ k_i &= 30 \text{ Vs/rad}, \end{aligned} \tag{24}$$

and the saturation limits

$$\begin{aligned} u_{\max} &= 3 \quad \text{V}, \\ u_{\min} &= -3 \text{ V}. \end{aligned} \tag{25}$$

The time history of the velocity response $\dot{q}(t)$ is shown in Fig. 5, while the effective control action $\hat{u}(t)$ is depicted in Fig. 6.

In Fig. 5, saturation of the control action $\hat{u}(t)$ is observed, which implies that the integrator accumulates a large numerical value. In consequence a large overshoot in the velocity response $\dot{q}(t)$ is obtained, as seen in Fig. 5.

6.2 PI control + anti–windup

In order to keep a fair comparison with respect to the PI controller (10)–(12), we have implemented the PI controller plus anti-windup (13)-(14) using the desired velocity $\dot{q}_d$ in (17), k_p and k_i in (24), the saturation limits (25), and

$$k_{AW} = 500 \text{ 1/s}. \tag{26}$$

The results are given in Figs. 7 and 8, which show the velocity $\dot{q}(t)$ and the effective control action $\hat{u}(t)$, respectively. Note that a nice response is observer for the velocity $\dot{q}(t)$ which converges rapidly to the desired value $\dot{q}_d$, while the $\hat{u}(t)$ remains saturated during a short time.

6.3 satPI control + anti–windup

The implementation of the controller (17)–(20) has been carried out using $\lambda_p = \lambda_i = 1$, the desired velocity $\dot{q}_d$ in (23), k_p and k_i in (24), the saturation limits

(25), and k_{AW} in (26). Figure 9 describes the time history of the velocity $\dot{q}(t)$ and Fig. 10 shows the effective control action $\hat{u}(t)$. After a short time, the velocity $\dot{q}(t)$ converges to the desired value $\dot{q}_d$, although a relatively large overshoot was presented.

6.4 Discussions

In the three experiments, the velocity response $\dot{q}(t)$ tends clearly to the desired velocity $\dot{q}_d$ after a short time (less than 0.5 [seconds]). However, different transients are observed. The percent overshoot has been computed has been computed, see Table 1. The controller PI plus anti-windup (13)-(14) presented the best performance.

The implementations showed that the effective control action $\hat{u}(t)$ in Figs. 6 and 8, obtained with the PI control (10)–(12) and PI plus anti–windup control (13)–(16), respectively, is noisier than $\hat{u}(t)$ in Fig. 10 obtained with the satPI plus anti–windup control (17)–(20). We have computed the energy–like index of the effective control action $\hat{u}(t)$, *ie*,

$$E = \int_{0.5}^{2.5} \hat{u}(\sigma)^2 d\sigma.$$

Note that the computing of E was done in steady state regimen, *ie*, approximately, after 0.5 [seconds]. The results are given in Table 1 and in Figure 11, which depicts the bar chart of E. With respect to this index, the best performance in obtained for the satPI plus anti-windup control (17)–(20).

The reason the new controller (21)-(22) presented a lower amplitude in the control action with respect to the other implemented controller is that new controllers incorporates saturation functions in the proportional and integral parts. In other words, it seems to be that the incorporation of those saturation functions has the effect in limiting the measurement noise in the velocity signal, which was confirmed with some numerical simulation studies.

Due to discretization, quantization errors, and high frequency PWM switching of the servo amplifier, high frequency components appear in the control signal. These components can be either increased or decreased with the control gains; in particular we noticed that were sensitive to the value of the proportional gain. But in spite of the high frequency contents in the voltage control signal, we did not observe negative effects in the performance of the motor, such as mechanical vibrations.

7 CONCLUSIONS

In this paper, a study on velocity regulation of DC motors subject to saturated control voltage has been presented. A new controller was introduced, which is based on the hyperbolic tangent function and an anti-windup loop. Through real-time experiments in a DC motor, we evaluated its performance. The results indicated that the accuracy of the new controller is similar to the one obtained with conventional techniques. The main advantage of the new method is that in steady state regimen consumes less energy.

References

[1] ÅSTRÖM, K.—HÄGGLUND, T.: PID Controllers: Theory, Design, and Tuning, 2nd ed., ISA–The Instrumentation, Systems, and Automation Society, Research Triangle Park, 1995.

[2] ÅSTRÖM, K. J.—WITTENMARK, B.: Computer Controlled Systems, Theory and Practice, Prentice Hall, Upper Sadle River, 1990.

[3] BAK, M.: Control of Systems with Contraints, PhD dissertation, Technical University of Denmark, 2000.

[4] BOLDEA, I.—NASAR, S.A.: Vector Control of AC Drives, CRC Press, Boca Raton, 1992.

[5] CORKE, P.I.: The Unimation PUMA Servo System, report MTM-226, CSIRO Division of Manofacturing Technology, 1994.

[6] DAGGETT, K.E.—ONAGA, E.M.—CASLER Jr., R.J.: Position and Velocity Feedback System for a Digital Robot Control, United States Patent No. 4876494, October 24, 1989.

[7] DORF, R.—BISHOP R.: Modern Control Systems, 8th ed, Addison–Wesley, Reasing, 1998.

[8] FERTIK, H.—ROSS, C.: Direct Digital Control Algorithms with Anti-Windup Feature, ISA Transactions **6** (1967), 317-328.

[9] FRANKLIN, G.—POWELL, J.—EMANI-NAINI, A.: Feedback Control of Dynamic Systems, Addison-Wesley, Reading, 1991.

[10] KHALIL, H.: Nonlinear Systems, Prentice Hall, Upper Saddle River, 1996.

[11] KHOTARE, M.V.—CAMPO, P.J.—MORARI, M.—NETT, C.N.: A Unified Framework for the Study of Anti-Windup Designs, Automatica **30** (1994), 1869–1883.

[12] KUO, B.: Automatic Control Systems, 7th ed, John Wiley & Sons, New York, 1995.

[13] MORENO, J.—KELLY, R.: On Motor Velocity Control by Using only Position Measurements: Two Case Studies, International Journal of Electrical Engineering Education **39** (2002), 118–127.

[14] MORENO–VALENZUELA, J.—SANTIBÁÑEZ V.—CAMPA, R.: A Class of OFT Controllers for Torque-Saturated Robot Manipulators: Lyapunov Stability and Experimental Evaluation, Journal of Intelligent and Robotic Systems (2008), 65-88.

[15] OGATA, K.: Modern Control Engineering, 4th ed, Prentice Hall, Upper Saddle River, 2002.

[16] SPONG, M.—VIDYASAGAR, M.: Robot Dynamics and Control, John Wiley & Sons, New York, 1989.

Received 10 February 2008

Javier Moreno–Valenzuela received the BSc degree in Electronics Engineering from the Instituto Tecnológico de Culiacán, Mexico, in 1997, and the PhD degree in Automatic Control from CICESE Research Center, Ensenada, Mexico, in 2002. He was a Postdoctoral Fellow at the Université de Liège, Belgium, from 2004 to 2005. Since 2002, he has been with CITEDI-IPN Research Center. His main research interests are control of electro-mechanical systems.