

On the RTAI-Lab implementation of an observer-based controller for a DC motor

**R. Salas-Cabrera, J. C. Mayo-Maldonado, E. Y. Rendon-Fraga,
E. N. Salas-Cabrera, H. Cisneros-Villegas**

Instituto Tecnológico de Ciudad Madero

Departamento de Ingeniería Electrónica

Maestría en Ingeniería Eléctrica

Av. 1o. de Mayo S/N, Madero, Mexico

salascabrera@aol.com, jcarlos_mayo@hotmail.com, erendonfraga@hotmail.com,

nacu_salas@hotmail.com, hcvillegas@yahoo.com.mx

Abstract

This work deals with the experimental implementation of a state space controller for a DC motor that has separate winding excitation. The output to be controlled is the rotor position. Linux and RTAI-Lab are the open source tools that we use in this implementation. The control law is derived by using a model that consists in a set of ordinary differential equations. Additionally, in order to avoid several issues regarding the measurement of the state variables, we implement an experimental discrete-time state observer to estimate the transient values of the rotor speed, armature current and even the load torque. The real time program calculates the discrete-time state feedback including an integrator, a full order discrete-time state observer and a rotor position sensing algorithm. Custom-made digital, analog and power electronics designs are fundamental components of the hardware used in this closed loop implementation.

1 Introduction

This paper presents the implementation and the experimental results of an observer-based control law. RTAI was used in order to perform the experimental implementation of a state space controller for a DC motor. The field and armature windings of the DC machine were fed from independent sources, [1]. The output to be controlled is the rotor position.

An interesting contribution is presented in [2], the author uses RTAI-Lab, Scilab and a data acquisition hardware for implementing a transfer function-based controller for a DC motor. The main contribution in [2] is related to testing several open source real time tools (RTAI, Comedi, Scilab, xrtailab). Experimental results regarding high gain observers are presented in [3] where RTAI-Lab and Scicos/Scilab are used for the real time implementation. The system to be tested in [3] was a series DC motor. Valuable information about a problem associated with the analog outputs of the acquisition card was pre-

sented in [3].

Linux and RTAI-Lab are the open source real time platform that we employed for this implementation. In particular a version of knoppix 5.0 was used for this work, this version was developed by Gianluca Palli and it can be downloaded from his home page [4].

The real time program that we implemented consists of a discrete-time state feedback including an integrator, a full order discrete-time state observer and a rotor position sensing algorithm. In this control law implementation the only state variable that is measured is the rotor position of the motor. In our case, a set of five difference equations (fifth-order discrete time dynamical system) is computed every 0.2 milliseconds (5 kHz). Important components of the experimental system are: a personal computer, a National Instruments PCI-6024E data acquisition card [5], comedi driver library for Linux [6], a free open source real time platform [7], a custom-made

power electronics converter, an incremental encoder and signal conditioning circuits for measuring the rotor position.

2 DC Motor Model

The equations that establish the DC motor behavior are obtained by using fundamental electrical and mechanical laws [1], this is

$$V_f = r_f i_f + L_{ff} \frac{d}{dt} i_f \quad (1)$$

$$V_a = r_a i_a + L_{aa} \frac{d}{dt} i_a + L_{af} i_f \omega_r \quad (2)$$

$$J \frac{d}{dt} \omega_r + T_L = L_{af} i_f i_a \quad (3)$$

$$\frac{d}{dt} \theta_r = \omega_r \quad (4)$$

In order to improve the performance of the closed-loop system, a new linear state equation for the load torque is proposed, this is

$$\frac{d}{dt} T_L = K_0 \omega_r + K_1 T_L \quad (5)$$

Several transient and steady state tests were performed in order to obtain the nominal parameters of the DC motor. Since the field winding is fed with an independent constant voltage source, the equation (1) can be omitted and the field current is considered as a constant parameter. The notation of the variables and the calculated values of parameters of equations (2)-(5) are shown in the following table:

θ_r - Rotor Position
ω_r - Rotor Speed
i_a - Armature Current
i_f (0.46 Amps) - Field Current
V_f - Field Voltage
V_a - Armature Voltage
r_a (6.615 Ohms) - Armature Resistance
L_{aa} (0.0645 H) - Armature Inductance
L_{af} (1.7686 H) - Mutual Inductance
J (0.0038 kg/m ²) - Inertia
T_L - Load Torque
K_0 (0.20907) - Coefficient of Load Torque
K_1 (-9.8297) - Coefficient of Load Torque

A nominal dynamic model is determined by substituting the numerical parameters in the equations (2)-(5). Then, employing 5 kHz as a sample rate and following the procedure presented in [8], it is possible

to obtain the discrete time dynamic model. Also it is possible to obtain the discrete model by using the Scilab comand *dscr*.

The nominal linear time-invariant discrete time dynamic model obtained is

$$\begin{bmatrix} \theta_r(k+1) \\ \omega_r(k+1) \\ i_a(k+1) \\ T_L(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.000200 & 0.000004 & -0.000005 \\ 0 & 0.999945 & 0.042383 & -0.052578 \\ 0 & -0.002497 & 0.979644 & 0.000065 \\ 0 & 0.000041 & 0.9 \times 10^{-6} & 0.998035 \end{bmatrix} \begin{bmatrix} \theta_r(k) \\ \omega_r(k) \\ i_a(k) \\ T_L(k) \end{bmatrix} + \begin{bmatrix} 4.4 \times 10^{-9} \\ 0.0000659 \\ 0.0030691 \\ 9.201 \times 10^{-10} \end{bmatrix} V_a(k) \quad (6)$$

In addition, parametric uncertainty will be addressed by using an integrator which is defined in equation (7), where x_I is the integrator state variable, r is the setpoint and $y = \theta_r$ is the output to be controlled.

$$x_I(k+1) = x_I(k) + e(k) = x_I(k) + y(k) - r(k) \quad (7)$$

3 Controller

The armature voltage defined by the controller is

$$V_a(k) = - \begin{bmatrix} K_I & K_1 & K_2 & K_3 & K_4 \end{bmatrix} \begin{bmatrix} x_I(k) \\ \theta_r(k) \\ \omega_r(k) \\ i_a(k) \\ T_L(k) \end{bmatrix} \quad (8)$$

The gains K_I, K_1, K_2, K_3 and K_4 are calculated by using the pole placement technique explained in [8]. Also it is possible to calculate the gains by establishing new poles for the closed-loop system and using the scilab command *ppol*. In this case, the gains are

$$\begin{aligned} [K_I, K] &= [0.0006168, 1.2288494, -0.6467532, \\ &\quad -4.021708, -2.4009488] \end{aligned}$$

The main Scicos program of the implemented controller is shown in figure 1. The main program contains the previously calculated gains K_I, K_1, K_2, K_3 and K_4 which multiply the state variables for defining the armature voltage. In figure 1,

the integrator of equation (7) is implemented by using the $1/z$ block of the linear palette [9]. The gain A in figure 1 scales the numerical value of the calculated armature voltage. The scaled voltage is the numerical representation of the actual voltage applied to the power converter. The COMEDI A/D block provides access to the data acquisition card which is connected to the power electronics converter.

There are two super blocks in the main program shown in figure 1, one of them is a state observer which estimates the speed, the armature current and the load torque. The other super block is a position measurement algorithm used for conditioning the data of the analog and digital inputs.

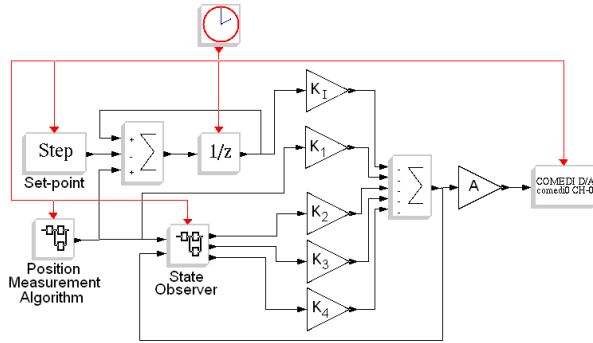


FIGURE 1: *Position Control Program in Scicos*

4 State Observer

Since the rotor position is the only measured variable, it is necessary to include a state observer for the state feedback. The implemented observer is based on the following equation

$$\tilde{x}(k+1) = G\tilde{x}(k) + Hu(k) + K_e[y(k) - C\tilde{x}(k)] \quad (9)$$

Clearly, G and H are known, they were previously defined in equation (6). The observer gain K_e is a 4×1 constant vector associated with the output error $y(k) - C\tilde{x}(k) = \theta_r(k) - \hat{\theta}_r(k)$ and is calculated by employing the scilab command *ppol* [9]. In this case the observer gain is

$$K_e = \begin{bmatrix} 0.0015523 \\ 0.1544085 \\ -0.0392419 \\ -0.0014389 \end{bmatrix} \quad (10)$$

The Scicos program related to the state observer

is shown in figure 2. This program is basically the expression (9) made of scicos blocks.

5 Rotor Position Measurement

A design for measuring the rotor position is implemented. There are several components of this hardware/software design. One of them is the Pepperl+Fuchs encoder that is physical attached to the motor shaft. The second one is a ATmega8535 microcontroller-based signal conditioning circuit and the third one is the real time software that calculates the rotor position from the digital signals provided by the microcontroller. The ATmega8535 microcontroller is programmed for being used as a 16-bit binary UP/DOWN counter. A signal conditioning circuit is necessary to determine the shaft direction, which is implemented by using a flip-flop integrated circuit. The flip-flop uses the pulses of channels A and B of the encoder and provides a binary 1 or 0 depending on the shaft direction. Channel A (or B) of the encoder is then connected to a micro-controller terminal without modifying the voltage as they both the encoder and the microcontroller have 5 volts as a nominal voltage. The microcontroller is in charge of counting the pulses provided by the encoder. In this case, the encoder generates 1024 pulses per revolution. The micro-controller count goes up or down depending on the flip-flop binary signal. The complete electronics diagram of the rotor position measurement setup is shown in figure 3.

The micro-controller provides a 16-bit binary count that represents the rotor position. On the other hand, the National Instruments PCI-6024E data acquisition card has only 8 digital inputs [5]. In order to be able to read the 16-bit binary data, 8 additional digital inputs are obtained by employing 8 analog inputs of the acquisition card and using them as digital inputs. The algorithm to interpret those analog inputs as digital inputs is coded in the Scilab/Scicos source program. The routine that reads binary data by using digital inputs is shown in figure 4. The algorithm consists of multiplying each bit (1 or 0) by its corresponding value in the decimal system, then the results are added. In order to read the data of digital input channels, the COMEDI DI block was used. The same algorithm is employed to convert bits that were obtained by using the analog inputs.

In addition, a signal conditioning routine is employed to use those analog inputs as digital inputs, see figure 5. A comparator determines the binary value (1 or 0) of the signal that was read at the

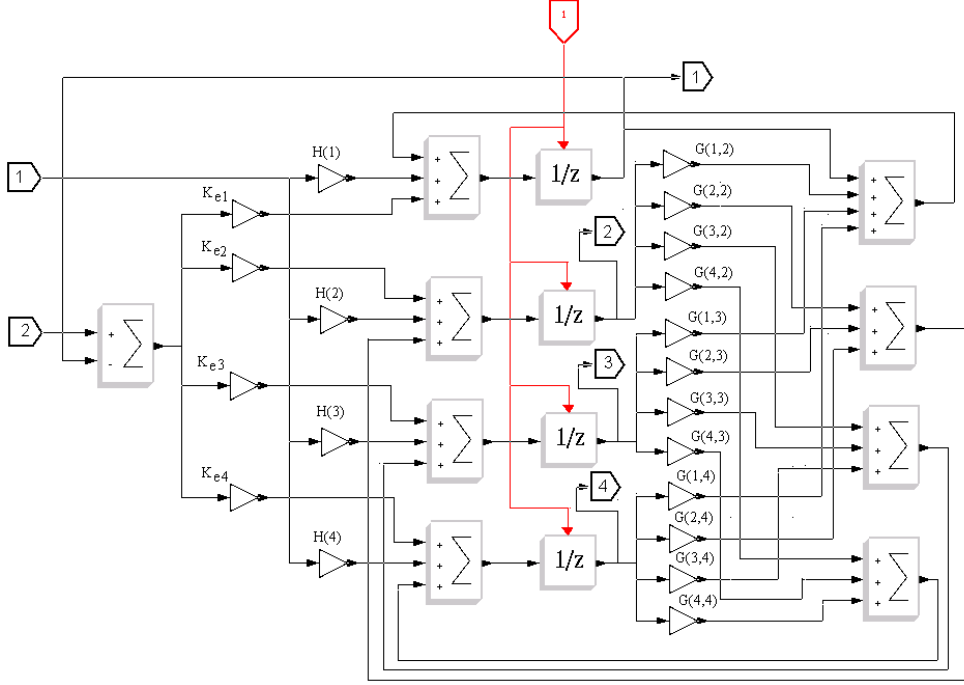


FIGURE 2: *State Observer Program in Scicos*

terminals of the analog input. This routine is implemented for each one of the 8 signals obtained by using the analog inputs. Combining the results provided by these routines a decimal measurement of the rotor position is obtained.

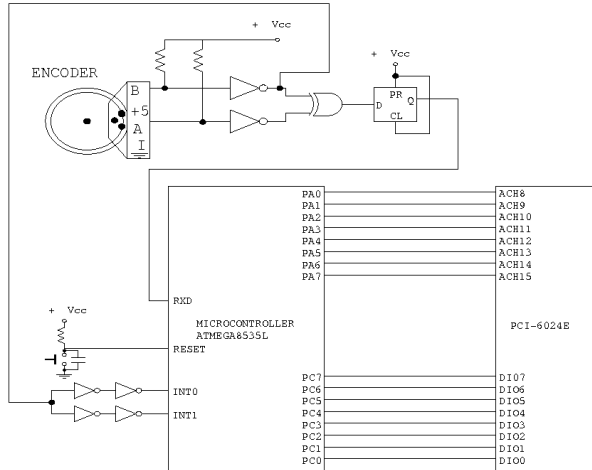


FIGURE 3: *Rotor position measurement setup*

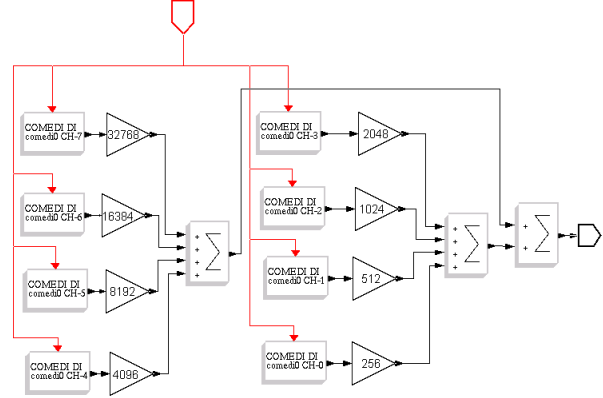


FIGURE 4: *Subroutine associated with the digital inputs of the position measurement algorithm*

A final super block called Position Measurement Algorithm (see figure 1) was created to include the described code, the program contained in this super block is shown in figure 6. This program adds the data of the previous routines. Then, the obtained value is converted into *radians*.

It is important to mention that the rotor position setup is able to measure positive and negative values. In order to accomplish this feature, the microcontroller was programmed to have an initial count that

is located at the middle of the 16-bit count range. Due to the lack of space, reader is referred to [10], where information regarding the ATmega8535 micro-controller assembler code can be found.

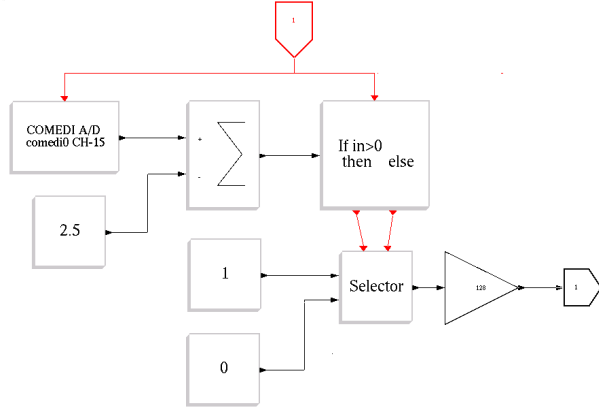


FIGURE 5: *Software-based signal conditioning for one of the eight analog inputs used as digital inputs*

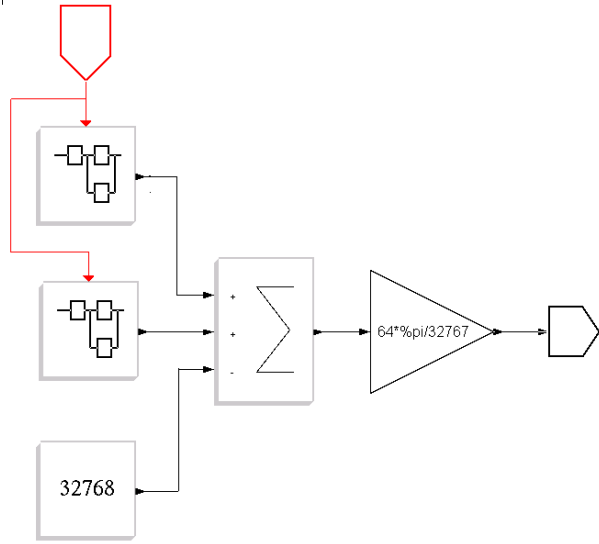


FIGURE 6: *Routine employed for converting measured rotor position values into radians*

6 Power Converter

A Pulse Width Modulation-based MOSFET H-Bridge converter was designed and implemented. This type of power electronics device is commonly used to drive DC motors when bidirectional speed/position control is needed [11]. The numerical value of the armature voltage is defined by the

state feedback and calculated by the computer. This value is written by the real time software to one of the analog output channels of the data acquisition card. The power electronics converter is shown in figure 7.

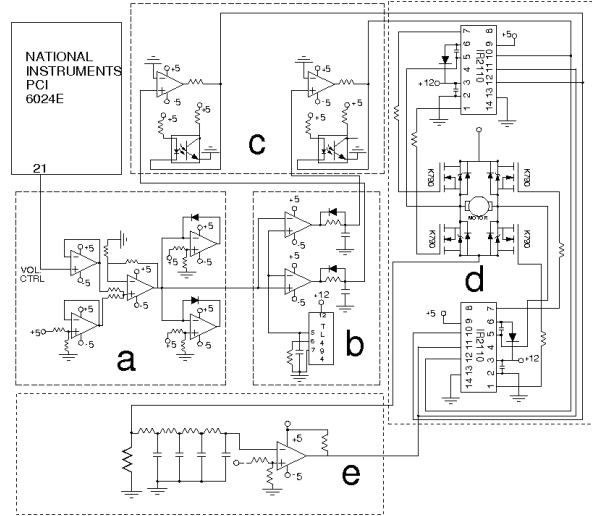


FIGURE 7: *Power Converter. a) Signal Conditioning. b) PWM and switching delay. c) Isolation. d) H Bridge and DC motor. e) Short Circuit Protection*

7 Experimental Results

The final program is compiled by using the RTAICodegen tool in Scicos, this procedure is explained in [7]. The rotor position and the experimental observer-based variables were saved by using FIFOs. In this case, a FIFOin block was employed for each variable and the data were saved by following the procedure indicated in [7].

The experimental trace shown in figure 8 illustrates the dynamic characteristic of the rotor position following a 25.1328 radian (4 revolution) reference command. The simulated model-based trace of the rotor position is also shown in figure 8. It is clear that these signals (simulated and measured) are similar to each other. Initially, the rotor position was at zero radians. The position begins to increase immediately until the position error is close to zero, which occurs approximately at 1.2 sec. Figure 9 shows the experimental observer-based rotor speed. Figure 10 shows the experimental observer-based armature current. The armature voltage computed by the controller is shown in figure 11.

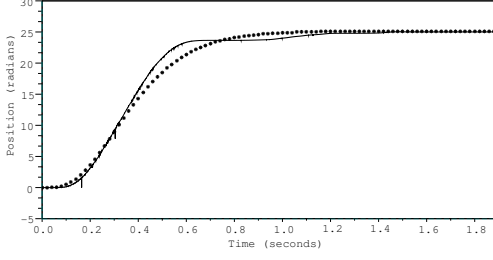


FIGURE 8: *Rotor Position: Experimental measured, ** Simulated model-based*

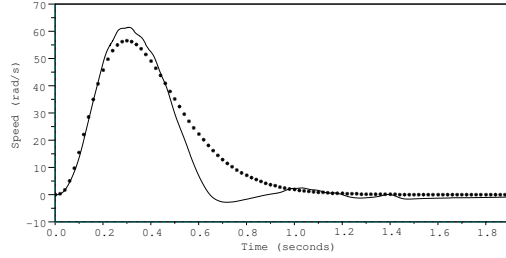


FIGURE 9: *Rotor speed: Experimental observer-based, ** Simulated model-based*

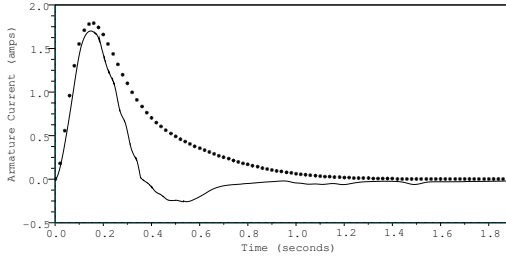


FIGURE 10: *Armature Current: Experimental observer-based, ** Simulated model-based*

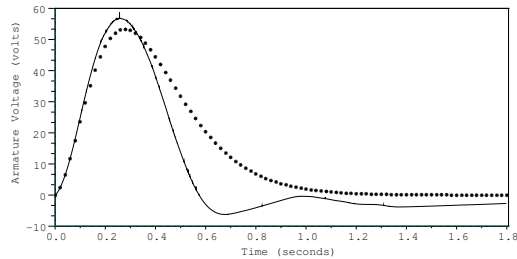


FIGURE 11: *Armature Voltage: Experimental Computed voltage, ** Simulated model-based*

The STEP block is employed in the main program (see figure 1). The parameters of the STEP block can be modified by using the xrtailab interface while the executable program is running [7]. In this case, this feature is used for selecting a set point value. Figure 12 shows the xrtailab interface and 3 radians as set point. Figure 13 illustrates the closed-loop system behavior with different set point values, 6π , 12π and 18π (3, 6 and 9 revolutions) respectively.

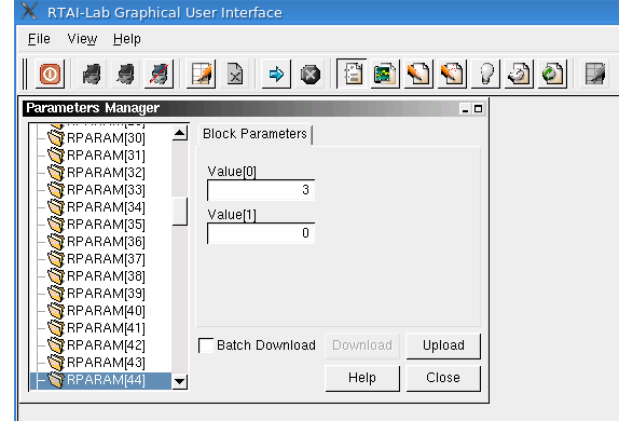


FIGURE 12: *Set point selection in the xrtailab interface*

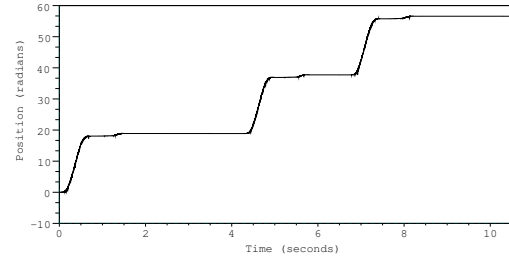


FIGURE 13: *Closed-loop system behavior with different set points*

8 Conclusions

The real time platform employed in this work provides ideal tools for this type of applications. The experimental results are similar to the simulation. The real time software is able to accomplish the control algorithm under strict time requirements. A future work might be related to the idea of implementing a new nonlinear control scheme, keeping the hardware platform and changing just the source program running under the same real time software platform.

9 Acknowledgments

Authors wish to thank Michael Griffin and Carlos A. Cruz-Villar for their valuable suggestions during the early stage of this work. This work was supported in part by the Instituto Tecnológico de Cd. Madero and the Fondo Mixto de Fomento a la Investigación Científica y Tecnológica COTACyT-Gobierno del Estado de Tamaulipas.

References

- [1] *Krause, P. C., Wasynczuk, O., y S. D. Sudhoff (2004). ANALYSIS OF ELECTRICAL MACHINERY AND DRIVE SYSTEMS. John Wiley and Sons, IEEE Press Power Engineering.*
- [2] *J. Jugo (2004). Real-time control of a DC motor using Scilab and RTAI. SCILAB INRIAROCQUENCOURT*
- [3] *Boizot, N., Busvelle, E. y J. Sachau (2007). High-gain observers and Kalman filtering in hard-time. RTL 9TH WORKSHOP*
- [4] *G. Palli. Home Page: <http://www-lar.deis.unibo.it/people/gpalli/>*
- [5] National Instruments (2006). PCI-6024E NATIONAL INSTRUMENTS MANUAL.
- [6] *D. Schleef, F. Hess, H. Bruyninckx (2003) THE CONTROL AND MEASUREMENT DEVICE INTERFACE HANDBOOK.*
- [7] Bucher, R., Mannori, S. y T. Netter (2008). RTAI-LAB TUTORIAL: SCILAB, COMEDI AND REAL-TIME CONTROL.
- [8] Franklin, G. F., Powell, J.D., y M. Workman (1998). DIGITAL CONTROL OF DYNAMIC SYSTEMS. Addison Wesley. Menlo Park, CA.
- [9] *Campbell, S. L., Chancellier, J. P. y R. Nikoukhah (2006). MODELING AND SIMULATION IN SCILAB/SCICOS. Springer.*
- [10] *E. Y. Rendon-Fraga. ROTOR POSITION CONTROL OF A DC MOTOR USING A REAL TIME PLATFORM, MSc. THESIS, IN SPANISH. Instituto Tecnológico de Cd. Madero. Cd. Madero, Mexico; 2009.*
- [11] *H. Rashid Muhammed (2004). POWER ELECTRONICS: CIRCUITS, DEVICES AND APPLICATIONS. Prentice Hall.*