Chapter 1
On the Experimental Control of a Separately Excited DC Motor

Rubén Salas-Cabrera, Jonathan C. Mayo-Maldonado, Erika Y. Rendón-Fraga, E. Nacú Salas-Cabrera, and Aaron González-Rodríguez

Abstract This paper presents the experimental implementation of a controller for the rotor position of a separately-excited DC Motor. The control law was designed by using a dynamic model for the motor and a dynamic model for the load torque. The torque model was proposed in order to improve the performance of the closed loop system. Discrete state feedback, discrete integrator and a discrete state observer are implemented and calculated by employing a real time control tool that works under a Linux operating system. Custom-made digital, analog and power electronics designs are the fundamental components of the closed loop implementation.

Keywords Real time · data acquisition · DC motor · electronics

1.1 Introduction

This work deals with the experimental implementation of a state space controller for a DC motor that has separate winding excitation. In other words, the field and armature windings of the electric machine are fed from independent sources, Krause et al. [7]. A brief review of the literature follows. An interesting contribution is presented in [12]. It is related to the speed control of a series DC motor. Zhao and Zhang [12] use a nonlinear dynamic model for deriving a backstepping-based control law. The emphasis is on the analysis of the controller. Jugo [5] uses RTAI-Lab and Scilab for implementing a transfer function-based controller of a DC motor. The main contribution in [5] is related to testing several open source real time tools (RTAI, RTAI-Lab, Comedi, Scilab, xrtailab). Experimental results regarding high gain observers are shown in [1]. RTAI-Lab and Scicos/Scilab are used for the real

R. Salas-Cabrera (✉), J.C. Mayo-Maldonado, E.Y. Rendón-Fraga, E.N. Salas-Cabrera, and A. González-Rodríguez
1501-1ro. de Mayo Avenue Pte., Cd. Madero, Tams., Mexico
e-mail: salascabrera@aol.com; jcarlos_mayo@hotmail.com; erika.yuridia.rendon@hotmail.com; nacu_salas@hotmail.com; aaronglzrod@yahoo.com.mx

S.-I. Ao and L. Gelman (eds.), *Electronic Engineering and Computing Technology*, Lecture Notes in Electrical Engineering 60, DOI 10.1007/978-90-481-8776-8_1,

time implementation. The system to be tested in [1] was a series DC motor. The contribution of this work is associated with the real time experimental implementation of the closed loop system. In other words, several hardware and software components were designed, i.e. some analog, digital and power electronics circuits were implemented. Important components of the experimental system are: a personal computer, a PCI-6024E data acquisition card [9], comedi driver library for Linux [11], a free open source real time platform [2], a custom-made power electronics converter, an incremental encoder and signal conditioning circuits for measuring the rotor position.

1.2 Modeling

The equations that establish the DC motor behavior are obtained by using fundamental electrical and mechanical laws [7], this is

$$V_f = r_f i_f + L_{ff} \frac{d}{dt} i_f \tag{1.1}$$

$$V_a = r_a i_a + L_{aa} \frac{d}{dt} i_a + L_{af} i_f \omega_r \tag{1.2}$$

$$J \frac{d}{dt} \omega_r + T_L = L_{af} i_f i_a \tag{1.3}$$

$$\frac{d}{dt} \theta_r = \omega_r \tag{1.4}$$

Notation for parameters and variables is given in Table 1.1. Parameters involved in (1.1)–(1.4) were calculated once we analyzed the experimental results of several transient and steady state tests.

Table 1.1 Notation for variables and parameters

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_f (309.52 Ohms)	Field resistance
r_a (6.615 Ohms)	Armature resistance
L_{aa} (0.0645 H)	Armature inductance
L_{ff} (14.215 H)	Field inductance
L_{af} (1.7686 H)	Mutual inductance
J (0.0038 kg/m^2)	Inertia
T_L	Load torque
K_0 (0.20907), K_1(−9.8297)	Coefficients of load torque

1.2.1 Load Torque Model

For this implementation, an experimental-based load torque dynamic model is proposed. The idea of proposing this model is to improve the performance of the closed loop system. Let us propose a state equation for the load torque, this is

$$\frac{d}{dt}T_L = K_0\omega_r + K_1 T_L \tag{1.5}$$

In order to obtain parameters in (1.5), we rewrite Eqs. (1.3) and (1.5) as

$$\frac{d}{dt}\begin{bmatrix}\omega_r \\ T_L\end{bmatrix} = \begin{bmatrix}0 & -\frac{1}{J} \\ K_0 & K_1\end{bmatrix}\begin{bmatrix}\omega_r \\ T_L\end{bmatrix} + \begin{bmatrix}\frac{L_{af} i_f}{J} \\ 0\end{bmatrix} i_a \tag{1.6}$$

where the armature current i_a and rotor speed w_r are interpreted as the input and output of this subsystem, respectively.

To identify parameters in (1.5) an experimental test was carried out. During this test, the field winding was fed with a constant current at $i_f = 0.46$ amps. It is clear that under these conditions state equation in (1.6) becomes a linear time-invariant description. Basically, the test consists of applying a random sequence to the armature terminals. In order to obtain the experimental transient variables, a tool called FIFO that is included in the real time platform was used [2, 3]. The experimental setup used for identifying parameters in state equation (1.6) consists of an ATmega8 microcontroller for defining the random sequence, a Nana Electronics SHR-100 hall effect sensor for measuring the armature current, a Pepperl+Fuchs incremental encoder for measuring the rotor speed, a National Instruments data acquisition card, RTAI-Lab real time platform and a custom-made PWM H-bridge converter. Diagrams of the experimental setup for parametric identification can be found in [10]. Additionally, reference [10] contains the source program of the microcontroller for defining the random sequence. Once we obtained the transient experimental data, we utilized Matlab for the off-line processing of the data. In particular, prediction-error approach is used to calculate the numerical version of the state equation in (1.6). By comparing (1.6) and its corresponding numerical version, load torque parameters K_0 and K_1 are defined.

1.2.2 State Space Representation

Mostly of the physical systems presents non-linear dynamic behavior. However, under some conditions they may be considered to have a linear time-invariant representation. This is the case of a DC motor with a field winding that is fed from a constant source. Substituting the numerical parameters specified in Table 1.1 into

(1.2)–(1.5) and employing 5 kHz as a sample rate, it is possible to obtain the nominal linear time-invariant discrete time dynamic model. Thus we have

$$\begin{bmatrix} \theta_r(k+1) \\ \omega_r(k+1) \\ i_a(k+1) \\ T_L(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.000200 & 0.000004 & -0.000005 \\ 0 & 0.000045 & 0.042383 & -0.052578 \\ 0 & -0.002497 & 0.979644 & 0.000065 \\ 0 & 0.000041 & 0.9 \times 10^{-6} & 0.998035 \end{bmatrix} \begin{bmatrix} \theta_r(k) \\ \omega_r(k) \\ i_a(k) \\ T_L(k) \end{bmatrix} + \begin{bmatrix} 4.4 \times 10^{-9} \\ 0.0000659 \\ 0.0030691 \\ 9.201 \times 10^{-10} \end{bmatrix} V_a(k) \tag{1.7}$$

$$\theta_r(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_r(k) & \omega_r(k) & i_a(k) & T_L(k) \end{bmatrix}^T \tag{1.8}$$

where (1.7) is the discrete time state equation and (1.8) is the discrete time output equation.

It is clear the following definition for the state $x = \begin{bmatrix} \theta_r & \omega_r & i_a & T_L \end{bmatrix}^T$, the output to be controlled $y = \theta_r$ and the input of the system $u = V_a$. State equation (1.7) assumes that parameters are linear time-invariant. Similarly, the field winding is supposed to be fed by a voltage source that keeps constant the field current at 0.46 amps. This is the reason for not including the state equation of the field winding. Experimental results in Section 4 will show that the nominal model in (1.7)–(1.8) contains the fundamental dynamic characteristics of the system necessary to design a successful experimental control law. Parametric uncertainty will be addressed by using an integrator, i.e.

$$x_I(k+1) = x_I(k) + e(k) = x_I(k) + y(k) - r(k) \tag{1.9}$$

where x_I is the integrator state variable, r is the setpoint and $y = \theta_r$ is the output to be controlled.

1.2.3 State Observer

On the other hand, since the experimental setup contains a PWM H-bridge converter some switching noise occurs. In addition, some problems regarding constructive imperfections of the incremental encoders make the measurement of the rotor speed unreliable [8]. In order to avoid these issues, we implement an experimen-

tal discrete-time state observer to estimate the rotor speed, armature current and even the load torque [4]. The state space equation of a standard linear time-invariant discrete time observer can be written as

$$\tilde{x}(k+1) = G\tilde{x}(k) + Hu(k) + K_e[y(k) - C\tilde{x}(k)] \tag{1.10}$$

The observer gain K_e is a 4x1 constant vector associated with the output error $y(k) - C\tilde{x}(k) = \theta_r(k) - \tilde{\theta}_r(k)$. As it is explained in the next subsection, desired poles are basically computed by specifying the desired time constant. Once the set of desired poles are obtained, a standard procedure for calculating the gain K_e is applied. This gain can also be computed by using the Scilab/Scicos *ppol* command. In this particular case, gains in (1.10) were chosen such that the desired observer poles become

$$z_{1,2,3,4} = 0.994017964, 0.994017963, 0.994017962, 0.994017961$$

this is

$$K_e = \begin{bmatrix} 0.0015523 \\ 0.1544085 \\ -0.0392419 \\ -0.0014389 \end{bmatrix} \tag{1.11}$$

It is important to emphasize that for the purpose of implementing the real time closed loop system, the rotor position is the only state variable to be measured. The state feedback uses the state variables (including the load torque) provided by the experimental discrete-time state observer. Armature current and rotor speed were measured just for the off-line parametric identification of the load torque model.

1.2.4 Controller

Pole placement technique is used to calculate the gains associated with the state feedback. Armature voltage $u(k) = V_a(k)$ is now defined by the following standard state feedback

$$u(k) = -\begin{bmatrix} K_I & K \end{bmatrix} \begin{bmatrix} x_I(k) \\ x(k) \end{bmatrix} \tag{1.12}$$

where K_I is the gain corresponding to the integral state variable, K is the gain vector associated with the state variable of the original system x. These gains are calculated following a standard procedure [4]. Gains can also be computed by using the Matlab command *place*. On the other hand, it is well known that the relationship between

a s-plane pole having a particular time constant and the corresponding z-plane pole is given by $z = e^{-\frac{1}{\tau}T}$, where τ is the time constant and T is the sampling period. If $\tau = 0.1$ and $T = 0.0002$ s, thus the corresponding pole in the z-plane becomes $z_1 = e^{-\frac{1}{0.1}(0.0002)} = 0.998001998$. Similarly, the rest of desired controller poles can be obtained

$$z_{1,2,3,4,5} = 0.998001998, 0.998001997, 0.998001996, 0.998001995, 0.998001994$$

thus the corresponding controller gains become

$$[K_I, K] = [0.0006168, 1.2288494, -0.6467532, -4.021708, -2.4009488] \tag{1.13}$$

1.3 Experimental Setup

1.3.1 Real-Time Platform

Linux and RTAI-Lab are the open source tools for real time tasks that we employed to solve the control algorithm [2, 6]. The resulting executable program is able to provide correctness of the calculated values and to accomplish strict time requirements. In this case, the control algorithm was computed every 0.0002 s (5 kHz). The real time program consists of a discrete-time state feedback including an integrator, a full order discrete-time state observer and a rotor position sensing algorithm. This program is able to access the analog/digital input/output terminals of the data acquisition card. In this particular work, a National Instruments PCI-6024E data acquisition card is used. The source program is compiled by using the RTAIcodegen tool which is included in the RTAI-Lab package [2]. The Knoppix 5.0 Linux distribution contains RTAI-Lab. Scilab/Scicos is a graphical environment that includes a library of blocks that can be employed to simulate the closed loop system [3]. Scilab/Scicos are used to simulate the closed-loop system as a first step of the implementation. Then, the Scilab/Scicos simulation blocks are switched to the corresponding RTAI-Lib real-time blocks. The main source program is shown in Fig. 1.1. In order to be able to show it in just one figure we used several subroutines that are called super blocks. They are a tool for organizing the structure of the program. A super block is similar to any other block, however its operation is defined by a custom-made Scicos code. The main program can be divided in three parts. The first part is the super block that includes the position measurement algorithm. The second part is the super block that computes the state observer. The last part calculates the state feedback and solves the integral state space equation. Closed

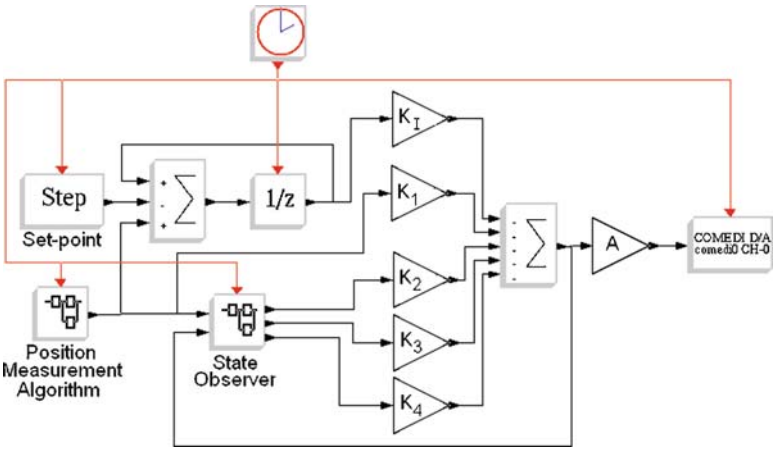

Fig. 1.1 Main Scicos program including several super blocks (subroutines)

loop gains K_I, K_1, K_2, K_3 and K_4 are defined in (1.13). In addition, the program includes the set-point block, the Comedi block (DAC) and a gain denoted by A. This gain scales the numerical value of the calculated armature voltage.

1.3.2 Rotor Position Measurement Design

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 an 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. In this case, the encoder generates 1,024 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 Fig. 1.2. The micro-controller provides a 16-bit binary count that represents the rotor position. The algorithm to interpret the 16-bit binary count is coded in the Scilab/Scicos source program. A part of this algorithm is shown in Fig. 1.3. The algorithm consists of multiplying each bit (1 or 0) by its corresponding value in the decimal system. Once the results are added a decimal measurement of the rotor position is obtained.

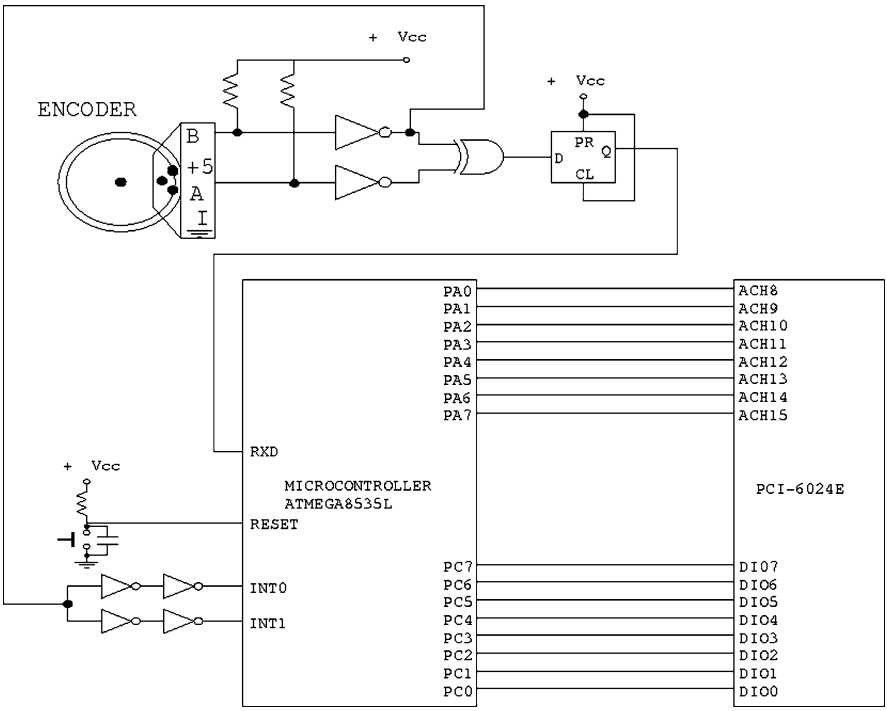

Fig. 1.2 Rotor position measurement setup

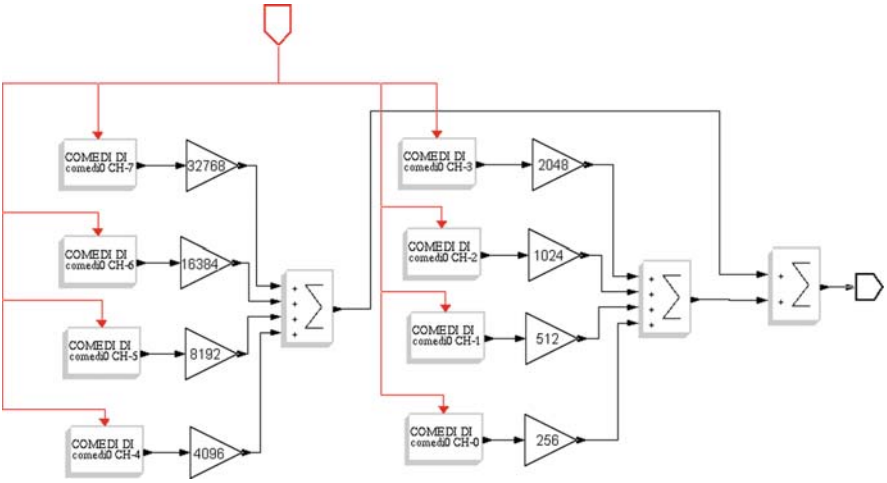

Fig. 1.3 Subroutine associated with the digital inputs of the position measurement algorithm

A final super block called *Position Measurement Algorithm* (see Fig. 1.1) was created to include the described code. 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.

1.3.3 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. 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 program to one of the analog output channels of the data acquisition card. The power electronics converter and motor are shown in Fig. 1.4.

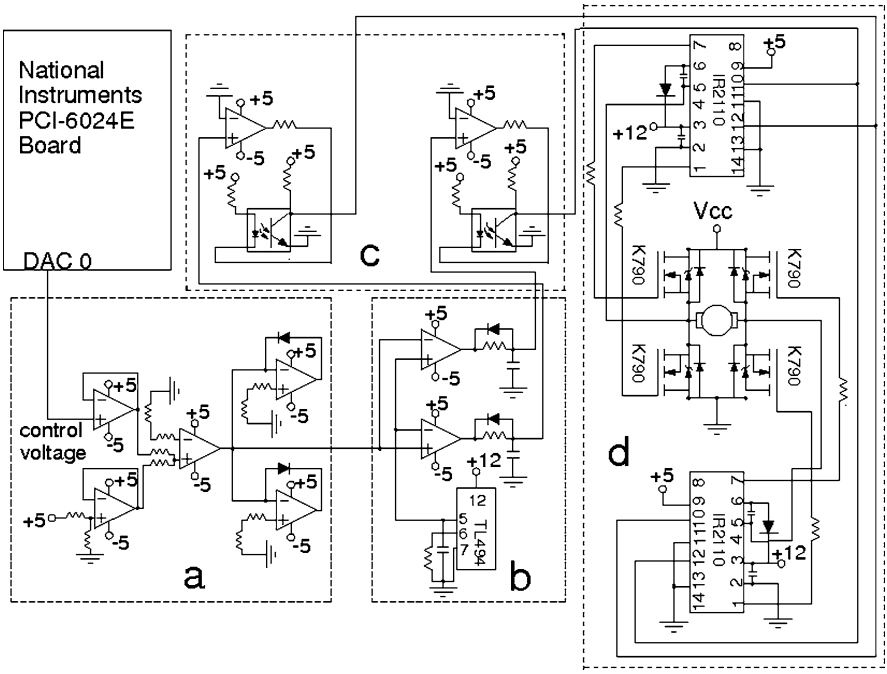

Fig. 1.4 Power converter and DC motor. (**a**) Signal conditioning. (**b**) PWM and switching delay. (**c**) Isolation. (**d**) H bridge and DC motor

1.4 Experimental Results

This section presents the experimental transient characteristics of the closed loop system. The armature voltage is computed as a function of estimated states excepting the rotor position, which is the only state variable that is measured. Estimated states, including the load torque, are obtained by solving the full order discrete-time state observer in (1.10). Initial conditions of the integrator and observer state variables were set numerically equal to zero. The experimental trace shown in Fig. 1.5 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 Fig. 1.5. 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 s. In order to save these transient data, a real-time block called FIFO was used [2,3]. Experimental observer-based variables were also saved by using FIFO. One of these variables is shown in Fig. 1.6. The armature voltage computed by the real time program is depicted in Fig. 1.7.

For the purpose of testing the experimental closed loop system in a more demanding operating condition, we installed an inertial disk. The original value of the inertia was 0.0038 kg/m^2, see Table 1.1. The new inertia is 0.0398 kg/m^2. In other words, the new parameter (inertia) is about 10.5 times the original value. It means a significant increase of a particular parameter of the mechanical subsystem. It is clearly connected to the output to be controlled (rotor position). In addition, the inertial disk was intentionally slightly loose to the shaft, therefore the parameter varied during the test around the non-original value. The original gains that were calculated by using the nominal parameters in Table 1.1 were not changed during

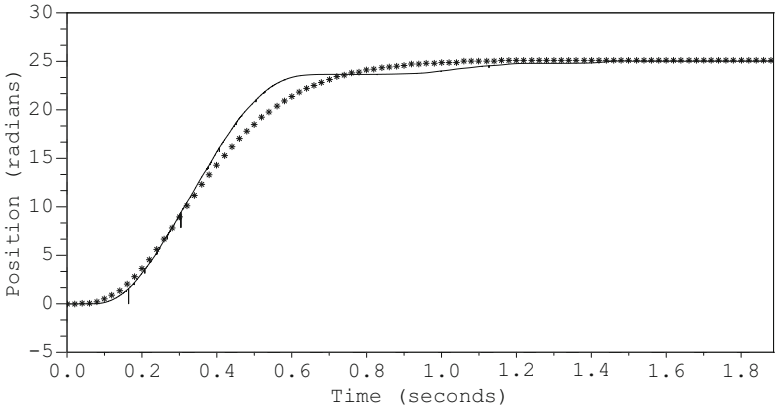

Fig. 1.5 Rotor position: – Experimental measured, ** Simulated model-based

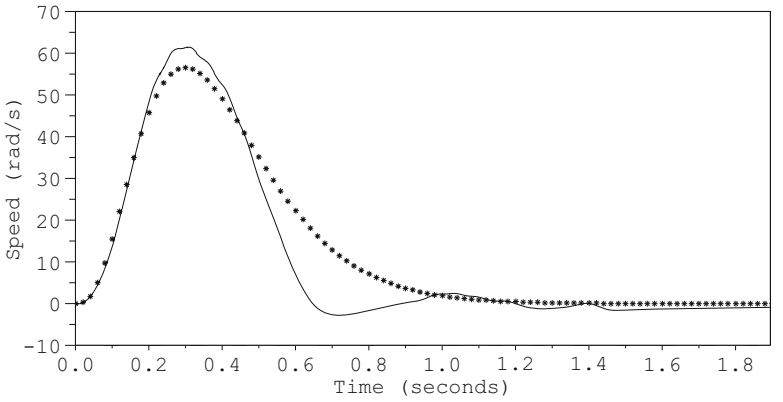

Fig. 1.6 Rotor speed: – Experimental observer-based, ** Simulated model-based

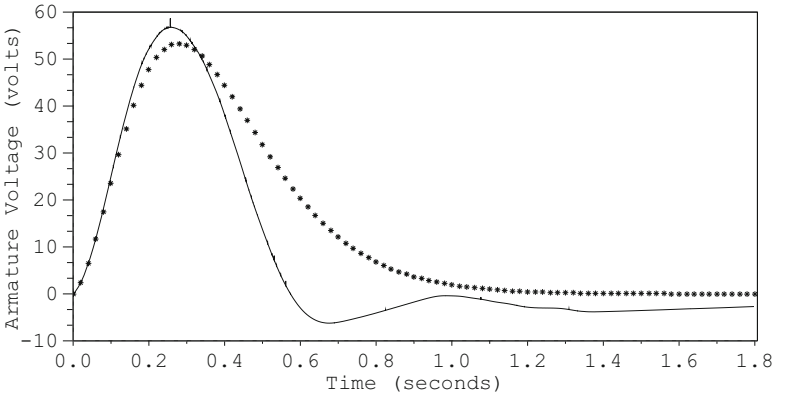

Fig. 1.7 Armature voltage: – Experimental, ** Simulated model-based

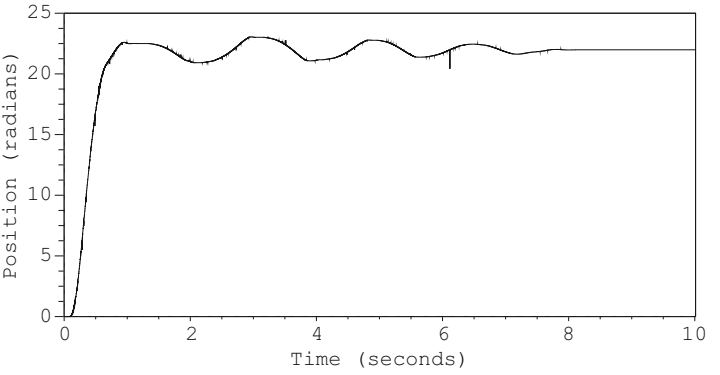

Fig. 1.8 Experimental measured rotor position when a significant change of a parameter occurs (10.5 times the original value approx.)

this test. Even under that demanding condition the experimental closed loop system was able to follow a 21.99 *radian* (3.5 *revolution*) reference command, see Fig. 1.8.

References

1. Boizot, N., Busvelle, E., Sachau, J.: High-gain observers and Kalman filtering in hard real-time. RTL 9th Workshop, 2007
2. Bucher, R., Mannori, S., Netter, T.: RTAI-Lab tutorial: Scilab, comedi and real-time control. http://www.rtai.org. 2008
3. Campbell, S.L., Chancellier, J.P., Nikoukhah, R.: Modeling and simulation in Scilab/Scicos. Springer, New York (2006)
4. Franklin, G.F., Powell, J.D., Workman, M.: Digital control of dynamic systems. Addison Wesley, Menlo Park, CA (1998)
5. Jugo, J.: Real-time control of a DC motor using Scilab and RTAI. Scilab INRIA Rocquencourt, 2004
6. Knoppix: Open source Linux distribution. Knoppix 5.0, 2004
7. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of electrical machinery and drive systems. Wiley, IEEE Press Power Engineering (2004)
8. Merry, R., van de Molengraft, R., Steinbuch, M.: Error modeling and improved position estimation for optical incremental encoders by means of time stamping. American Control Conference, 2007
9. National-Instruments: PCI-6024E National Instruments Manual (2006)
10. Rendon-Fraga, E.Y.: Rotor position control of a DC motor using a real time platform. M.Sc. thesis (in spanish). Instituto Tecnologico de Cd. Madero. Cd. Madero, Mexico (2009)
11. Schleef, D., Hess, F., Bruyninckx, H.: The control and measurement device interface handbook. http://www.comedi.org, 2003
12. Zhao, D., Zhang, N.: An improved nonlinear speed controller for series DC motors. IFAC 17th World Congress, 2008