Low Cost Closed loop Identification of a DC Motor

Alberto Soria, Rubén Garrido y Antonio Concha

Abstract—this paper presents the closed loop identification of a velocity-controlled servomechanism using a data acquisition system based on a microcontroller. The low cost and simplicity of the system enables undergraduate students to build it and to program an identification algorithm. Experiments in a laboratory prototype allow assessing the performance of the system.

Key words: closed loop parameter identification, velocity control, DC motor, microcontroller.

I. INTRODUCTION

The role of laboratories in engineering education is to show to the students how some of the theoretical issues studied at the classroom apply in practical situations. In the case of Automatic Control, Robotics and Mechatronics model-based controllers require knowledge of the model of the device to be controlled; this knowledge includes the structure and the model parameters. In most instances, numerical simulations precede laboratory experiments; in the former case, the student knows the model and estimating the model parameters is not an issue. However, in the case of real-time experiments, frequently the students known beforehand the structure of the model but they face the problem of estimating the model parameters. Another key issue is the stability of the device to be controlled. If it is open loop stable there exists identification methods that work in this case [1], [2]. However, if the device is not open loop stable, or for security reasons the device cannot operate in open loop, then parameter identification must resort on closed loop methods.

System identification courses frequently include computer-aided control system design software packages (CACSD) such as Matlab/Simulink, virtual laboratories, remote or in-situ laboratories. CACSD packages allow implementing thorough numerical simulations the identification algorithms, either manually or using existing tools as the System Identification Toolbox with the advantage that only the CACSD is necessary. Virtual laboratories [3] incorporate visual representations of the experiments. The main disadvantage of these approaches is that the students do not necessarily acquire experience with real world data experiments. Remote laboratories [4], [5], [6] permit the use of real prototypes accessible via a standard web browser interface. These facilities allow students to work with real world data; however, the experiments will be limited by the available interface, and, for security reasons, the operating points of the prototypes are restricted in order to produce a safe behaviour.

Commercial laboratory prototypes are available from a number of vendors [7], [8], [9], [10]. In most cases, these prototypes are sold as turnkey systems; however, they are often expensive, mask implementation details and could be inadequate for a particular type of experiment. On the other hand, custom-made platforms are more flexible and allow a significant reduction in cost since their design fulfils a specific application. With the development of powerful Peripheral Interface Controllers (PIC) [11], [12], it has become easier to implement low cost experimental platforms since the PIC incorporates a number of integrated peripheral such as analog to digital converter (AD), quadrature encoder interfaces (QEI), pulse width modulation (PWM), and serial ports or timers among others.

Graduate and undergraduate control systems engineering programs currently offer system identification laboratories for example [13], [14], [15], [16]. In these programs, discrete or continuous models combined with least squares algorithms allow performing parameter identification; in most cases, the plant under identification works in open loop. The use of a brushed DC motor is commonplace because simple linear models reasonably explain its behaviour and its low cost. Moreover, simple power circuits allow driving this kind of motors.

It is interesting to review works related to system identification in engineering education. Reference [17] presents a real-time laboratory for system modelling, control analysis and design. This reference also mentions the parameter estimation for a dc servo motor but it does not give details on the identification procedure. In [18] the Authors propose a state-space parameter identification methodology. The plant consists of two armature-controlled dc motors connected by their shafts; one of the motor works as a generator. A Least Squares algorithm permits identifying the plant parameters. Reference [19] uses the same experimental setup than that used in [18]. Nevertheless, a Least Squares fit allows estimating the plant static gain and a procedure based on the step response and a Bode plot permit estimating the plant time constant; the implementation is performed using operational amplifiers. Reference [20] shows a laboratory for undergraduate courses consisting on a simple temperature process; here, two transistors heat a small plate cooled by a fan. Inputs are the commands to the transistors and the fan while outputs are the temperatures of the plate and the transistors. Students identify the transfer functions by hand, i.e., selecting the model parameters so that the step response of the model fits the measured one. The Authors of [21] present a temperature control kit employing a microcontroller, a temperature

The authors are with the Departamento de Control Automático, CINVESTAV-IPN, Av. IPN 2508 Col. San Pedro Zacatencos, Mexico D.F. Corresponding author: betosoria@gmail.com.
sensor and a NMosfet transistor. The input is the voltage applied to the control to the NMosfet gate and the output is the temperature. Parameter identification of a model is performed using the reaction curve method and then a Ziegler-Nichols tuning method is subsequently used to tune a PID controller. Reference [22] describes an undergraduate prototype using a DSP board, an optical encoder and a DC motor. Identification is carried out in open loop using a Least Squares algorithm.

In this paper we present the parameter identification of a velocity controlled servomechanism using a low cost system based on a microcontroller. A closed loop identification method allows estimating a first order linear model of the servomechanism. A proportional controller closes the loop around the servomechanism and its identified model closed using a proportional controller. The error between the velocity of the motor and the velocity generated by the model feeds a gradient-like algorithm. The parameter estimates are compared with the parameters computed using the servomechanism technical data. The low cost and simplicity of the system enables undergraduate students to build it; programming of the identification algorithm is not complicated because of its low complexity. The experimental prototype includes a DC motor, an optical incremental encoder, an H-Bridge integrated circuit, and a microcontroller with dedicated Pulse Width Modulation and serial modules. Velocity measurements are obtained using a filter from the pulse wave generated by the optical encoder. The identified parameters are transmitted to a PC using a serial interface.

The paper layout is as follows. Section II describes the DC motor model and the identification algorithm. Section III shows servomechanism parameters obtained from its technical data. Section IV describes the PIC microcontroller, its programming and the PC monitoring software. Section V presents the experimental results. The paper ends with some concluding remarks.

II. DC SERVOMECHANISM MODEL

The following equations describe a linear model of a DC motor [23]:

\[ M_{v} = Ri + L \frac{di}{dt} + K_{emf} v \]

\[ J \frac{dv}{dt} + Bv = \tau = K_{m} i \]

with

\[ M_{v} : \text{Motor input voltage (V)} \]
\[ R : \text{Motor armature resistance (\( \Omega \))} \]
\[ J : \text{Motor and load inertia (Kg m}^{2}/\text{m}^{2}) \]
\[ L : \text{Motor inductance (H)} \]
\[ K_{emf} : \text{Back emf voltage constant (V/rev/s)} \]
\[ B : \text{viscous-friction coefficient of motor and load (N m s)} \]
\[ \tau : \text{motor torque (N m)} \]

The PIC PWM interface drives the integrated circuit LM-18200 H-Bridge in order to move the DC motor. A Proportional Integral (PI) controller regulates the motor current and the corresponding gains are \( K_{p}, K_{i} \). This controller is implemented within the PIC microcontroller and uses the AD converter to convert the current flowing in the H-Bridge and computes the control signal given to the PWM. In the PI implementation we directly use the units handled by the PWM and AD converter, thus the units for constants \( K_{C} \) and \( K_{u} \) are given in ADC counts. The current loop allows controlling the current and decreases the electrical time constant making it much faster that the motor mechanical time constant. The gain \( K_{H} \) of the H-bridge is related to the way the PWM interface of the PIC is programmed for a given resolution and frequency.

Figure 1 depicts a block-diagram of the servomechanism including the H-Bridge and the current loop. Since the pole \( K_{H} \) is far away from the origin than the mechanical pole \( \frac{B}{J} \), at low frequencies, this fact allows obtaining the diagram shown in Figure 2; the corresponding transfer function \( \rho(s) \) is:

![Figure 1. Block diagram of the DC Servomechanism.](image1)

![Figure 2. Reduced block diagram.](image2)
\[
\omega_n^2 \left( \frac{K_p}{K_i} s + 1 \right) \left( \frac{1}{s + \frac{1}{J}} \right) = \frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2} \left( \frac{K_m}{K_C} \right) \left( \frac{1}{J s + B} \right)
\]

(2)

with \( \omega_n = \sqrt{\frac{K_H K_i}{L}} \) and \( 2\xi\omega_n = \frac{R + K_m K_p}{L} \). High values of the integral gain increases the undamped frequency \( \omega_n \) thus allowing to ignore this dynamics. Therefore, at low frequencies, transfer function (2) simplifies to:

\[
\frac{K_m (J s + B)}{K_C}
\]

(3)

This reduced model permits obtaining the diagram shown in Figure 3 and the following transfer function:

\[
\frac{K_m K_C}{J + \frac{K_m K_{emf}}{K_i} s + B}
\]

(4)

Since \( \frac{K_m K_{emf}}{K_i} \) is much smaller than \( J \), transfer function (4) becomes:

\[
\frac{B}{s + a}
\]

(5)

with \( b = \frac{K_m K_C}{J} \) and \( a = \frac{B}{J} \).

III. IDENTIFICATION METHOD

In the closed loop identification method [24], a proportional controller regulates both the model and the plant (cf. Figure 4):

\[
u(t) = k_p e(t)
\]

(6)

\[
u_e(t) = k_p e(t)
\]

(7)

with the velocity error \( e(t) = v_d - v(t) \) and \( v_d \) the desired velocity. The plant and the model are given by:

\[
v(t) + av(t) = bu(t)
\]

(8)

\[
v_e(t) + \hat{a}v_e(t) = \hat{b}u_e(t)
\]

(9)

where \( u(t) \) and \( u_e(t) \) are the control signals for the motor and the model respectively. Model output \( v_e(t) \) calculation employs the 4th order Runge-Kutta method with \( h \) is the sampling period:

\[
v_e(t) = f(t, y), \ y(t_0) = y_0
\]

\[
y_{n+1} = y_n + \frac{h}{6} \left( k_1 + 2k_2 + 2k_3 + k_4 \right)
\]

(10)

\[
k_1 = f(t_n, y_n)
\]

\[
k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)
\]

\[
k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)
\]

\[
k_4 = f(t_n + h, y_n + h k_3)
\]

Applying the proportional control laws (1) and (2) to both systems yields:

\[
v(t) = -av(t) + bK_p e(t)
\]

(11)

\[
v_e(t) = -\hat{a}v_e(t) + \hat{b}K_p e(t)
\]

Define the identification error between the plant and the model \( \varepsilon = v(t) - v_e(t) \). The following equations defines its first time derivative:

\[
\dot{\varepsilon}(t) = -(a + bK_p)\varepsilon + (\hat{a} - a)v_e(t) - (\hat{b} - b)u_e(t)
\]

(12)

This last expression can be written alternatively as:

\[
\dot{\varepsilon}(t) = -c\varepsilon(t) + \hat{\Phi}^T \dot{\phi}
\]

(13)

where \( c = a + bK_p \) and \( a > 0, b > 0, K_p > 0 \), and vectors \( \hat{\Theta} \) and \( \phi \) are defined as:

\[
\hat{\Theta} = \hat{\Phi} - \theta = \begin{bmatrix} \hat{a} - a \\ \hat{b} - b \end{bmatrix}
\]

(14)

The stability proof for (12) follows along the ideas presented in [24] using the following Lyapunov function candidate:
The time derivative of (7) is:

\[ V(\varepsilon, \Theta) = \frac{1}{2} \varepsilon^2(t) + \Theta^T \Gamma^{-1} \Theta \]  

(15)

The time derivative of (7) is:

\[ \dot{V} = -c \varepsilon^2 + \Theta^T \left[ \varepsilon \phi + \Gamma^{-1} \dot{\Theta} \right] \]  

(16)

If an adaptation law is chosen such that:

\[ \dot{\Theta} = -\Gamma \Phi \varepsilon = \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix} \begin{bmatrix} \varepsilon_c(t) \\ u_c(t) \end{bmatrix} \varepsilon \]  

(17)

Then:

\[ \dot{V} = -c \varepsilon^2 \]  

(18)

\[ V \] is negative semi-definite since \( c > 0 \); thus \( \varepsilon \) and \( \Theta \) are bounded. Using Barbalat’s lemma [25] it is possible to show that the error \( \varepsilon = q(t) - \hat{q}(t) \), converges towards cero. In addition, if signal \( \phi \) fulfills the persistent excitation conditions [25], then the parametric error \( \Theta \) also converges to cero.

The update laws for each of the parameter estimates is given by:

\[ \dot{\hat{a}} = -\gamma_1 q_e(t) \varepsilon \]  

(19a)

\[ \dot{\hat{b}} = \gamma_2 u_e(t) \varepsilon \]  

(19b)

The Euler’s method allows implementing (18) in the PIC microcontroller, i.e. the time derivative is approximated as follows:

\[ y \approx \frac{1}{h} \left( y(t+1) - y(t) \right) \]  

(20)

where \( h \) is the sampling period. Consequently, applying Euler’s approximation to the adaptation parameters (13) we have:

\[ \begin{align*}  
\dot{\hat{a}} & \approx \frac{1}{h} \left[ \hat{a}(t+1) - \hat{a}(t) \right] = -\gamma_1 q_e(t) \varepsilon(t) \\
\dot{\hat{b}} & \approx \frac{1}{h} \left[ \hat{b}(t+1) - \hat{b}(t) \right] = \gamma_2 u_e(t) \varepsilon(t) 
\end{align*} \]  

(21)

If we chose sampling times \( t_k = 0, 1, 2, \ldots, n \), the discrete adaptation laws become:

\[ \begin{align*}  
\hat{a}(t_{k+1}) &= \left[ -\gamma_1 q_e(t_k) \varepsilon(t_k) \right] h + \hat{a}(t_k) \\
\hat{b}(t_{k+1}) &= \left[ \gamma_2 u_e(t_k) \varepsilon(t_k) \right] h + \hat{b}(t_k) 
\end{align*} \]  

(22)

IV. THE PIC MICROCONTROLLER, HARDWARE AND SOFTWARE

Figure 11 shows a picture of the system while figure 10 shows the system interconnection in detail. The PIC used in the experiments has the number DsPIC 30f4012. This PIC is targeted mainly for motor control applications and has two modules to interface a motor, namely the Quadrature Encoder (QE) and the Pulse Width Modulation (PWM) modules. The QE allows estimating the DC motor speed while the PWM module together with the LM18200 H-bridge with a carrier frequency of at 20KHz comprise the power amplifier stage controlling the power delivered to the motor. The analog to digital converter (ADC) 10 bit module is used to measure the current in the H-bridge. The timer module permits handling the sampling periods of the excitation signal, the identification routine, speed measurement and model following routine. The UART serial communication module is used to communicate the
experimental data to a PC computer. The PIC software was organized in eight parts:

i) Main routine comprising the current loop, speed measurement and the main execution loop.

ii) Initialisation and configuration routines for the QEI, PWM, ADC, timers and UART modules.

iii) Timers and UART interrupt service routines.

iv) Identification routine.

v) Model following routine.

vi) Delay routines for the PIC.

vii) General function declarations and definitions.

viii) DsPIC 30F4012 configuration bits definitions.

The highest priority routines are those called by interrupt scheme: QEI, ADC, timers, and UART modules. The main routine performs module initialisation and calls the first identification and then the model following once the identification is completed.

The optical incremental encoder is the US-Digital E-62500 model allowing counts up to 10000 pulses per revolution. We calculate velocity from encoder positions with the filter:

\[
\frac{39.64 z^2 + 39.64}{z^2 + 1.40z + 0.4811}
\]  

(23)

The motor used is a dual shaft Dynetic Systems brush DC motor 509051. The price for the main components including the quadrature incremental encoder, the motor, the H-Bridge and the DsPIC is around $115 USD. This price is low-cost if we consider for example the NI-ElvisII system [26] having a price around 40 times higher. Therefore, the proposed system implementation is a reasonable low-cost alternative.

V. EXPERIMENTAL RESULTS

The motor shaft speed \( v \) was measured by counting the number of pulses generated every 1ms by the incremental optical encoder. The excitation signal \( v_e \) used for the identification is a pseudorandom binary sequence with amplitude of 2 units plus a constant value of 2. This signal is then filtered using the following low pass filter:

\[
\frac{10}{s + 10}
\]  

(24)

This filter was discretized using the Euler method. The sampling period for the excitation signal \( v_e \) was 100ms. The identification loop was sampled at 1ms. The adaptation gains used are \( \gamma_1 = 5 \) and \( \gamma_2 = 2000 \). The controller gain is \( K_p = 3 \). A serial interface at 38400 was used in order to read the data from the PIC. Figure 5 shows the time evolution for parameter \( \hat{a} \) and figure 6 shows the time evolution for \( \hat{b} \). In order to verify the soundness of the identified parameters the a model following control scheme presented in figure 7 is proposed. The control law for this scheme is:

\[
u(t) = \frac{\alpha e_m + \dot{v}_m(t) + \dot{v}(t)}{\hat{b}}
\]  

(25)

with \( \dot{v}_m(t) = -\dot{v}_m(t) + r(t) \) and \( e_m(t) = v_m(t) - v(t) \) where \( r(t) \) is the reference signal. Figure 8 shows the reference and motor speed signals. Figure 9 shows the tracking error when performing the model following.

VI. CONCLUSIONS

This paper presents the identification of a DC motor using a simple closed loop identification method implemented in a low cost system based on a PIC microcontroller. The estimated parameters given by the identification method where tested using a model following scheme between the identified model and the real plant. These experimental results show the feasibility of the approach.

VII. ACKNOWLEDGEMENTS

Authors would like to thank Jesús Meza and Gerardo Castro for the technical assistance for the paper.
REFERENCES


