Implementation of a FIR Filter and PID controllers into 8-bit Microcontrollers

C. Hernandez-Rosales, G. Quiroz, R. Femat
Instituto Potosino de Investigación Científica y Tecnológica IPICYT
Camino a la Presa san Jose 2055, lomas 4^{a}. sección
heros, gquiroz, rfemat@ipicyt.edu.mx, teléfono: (52)-444-8342000

Abstract—In this paper, we show the implementation of a digital filter and classical discrete-time PID controllers on a low-cost eight-bit microcontrollers (8-MCs) [1],[2]. These MCs are limited in comparison with 16-bit and 32 digital signal processors (DSPs). However, we show that with aid of special routines which are also contribution of this work, these can be exploited in servo control applications. One advantage of this proposal is that these algorithms can be programmed into 8-MCs with less than 128 bytes of RAM and 4 Kbytes of EPROM, such chips still dominate the market because of small size, low-cost and simple programming. By other hand, in this article the instrumentation is detailed with the purpose of show other alternative in the use of these 8-MCs (which had been used only in the control states) in closed-loop configurations; such as, (i) Control of a valve to handle flow-rates in tubes, (ii) Delivery of insulin bolus in the therapy of type 1 diabetes mellitus [3] or (iii) DC motor control in robot manipulators.

Keywords: Embedded Systems, 8-bit microcontrollers, FIR filters, discrete-time PID controllers.

I. Introducción

Until few years ago, micro-positioning and servo-control had been addressed using 16-bit and 32-bit DSPs, because of such devices are faster and have higher precision than the 8-bits MCs. Nevertheless, some applications with low dynamic don't require high precision and processing or, the cost of a DSP is not justified. In this case, using a 8-MCs is a better option, which some times could diminish the costs of the embedded system.

On the other hand, the success of the classical control theory in the past century was that it made possible the implementation of servomechanisms and the automatic control of process. Nevertheless, almost always the realization of control laws like PID as associated to computers or sophisticated processors; in which case the cost of the control systems was increased. Nowadays and with the developed of the micro electronic the processor and consequently the 8-MCs have been improved even triplicate the storage and processing capacity, making possible their use to realize more complex tasks. Unfortunately, in the most of cases this is ignored by the designer by wasting these new features [4, 5]. For this reason and with educational purposes we describe a novel method that is being utilized to exploit this kind of MCs using them in closed-loop applications aided with the well-known classical control theory.

II. Servomechanisms

The principal feature of a servomechanism or closed loop system is that always is necessary to feedback some variables of the system to be controlled. Nevertheless, in industrial applications with noisy environment, most of sensors or transducers used in measurement system's variables provide data with additional information that pollutes the actual signal. This is undesirable mainly as measured signal are used to take a decision about process. One alternative to remove these signals is to use filters on line and once with the filtered signal, this can be safely used in the feedback scheme to perform the control law. In order to illustrate some examples, following sections show the implementation of a digital filter and a PID controller into an 8-bit MC.

III. Implementing a fir filter

A filter is defined as an electronic circuit that "let to pass" information of a signal and rejects those are not; the *embedded systems* by other hand, allow a wider software-implementation of the filters beyond an electronic circuit in which case are known as *digital filters.* The digital filters can be grouped in two types: The Finite Impulse Response (FIR) and the Infinite Impulse Response (IIR). The difference in the responses (and here its name) is because in the second there is feedback in the filter and the first not; nevertheless the two structures the filter's output depends upon each sample of the previous output and of the previous and/or present data [6].

a) FIR versus IIR filters

FIR Filters are usually implemented using a fixed-point processor with a finite number of N-taps to compute the filter's output; in contrast IIR Filters are implemented using a floating point digital processor. Here, the number of taps indicates the amount of memory required to implement the filter, the number of calculations, and the amount of "filtering" that the filter can do; i.e., more taps means more stopband attenuation, less ripple, narrower filters, etc. The disadvantage of FIR Filters compared to IIR filters is that they require more memory and/or calculation to achieve a given filter response; moreover and the most important, certain responses cannot be implemented with FIR filters, even so the FIR filters are more widely used in digital signal processing applications by their easy design.

b) Design and implementation of the low-pass FIR filter.

This filter is designed using MatLab®, and its performance is tested by filtering a signal provided by a precision potentiometer (see Figure 1). This sensor is supplied to +/-5V and supplies an analog signal of 500 milli-volts per each revolution given by the motor's shaft. The angular motion of the motor is restricted by software at ten turns in both directions. Figure 1, also shows the schematic diagram of the electronic circuit used for the implementation of the FIR filter. In this circuit, the 8-MC drives the Analog/Digital conversion by means of the control bus in port 2. Once a cycle of A/D conversion is executed, the micro enables the data bus to read the digital signal with a 12 bit resolution (D_0-D_{11}). Thereafter, the filter's output is sent to PC to plot it.

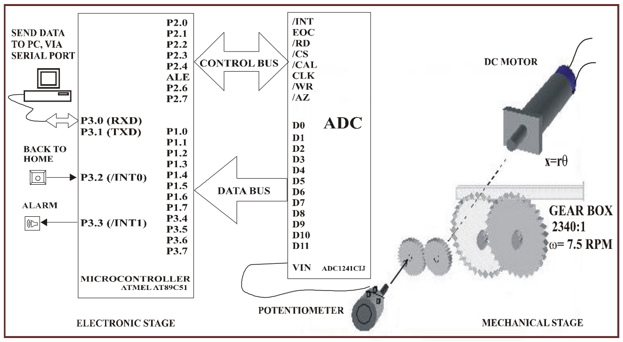

Figure 1 a) Mechanical stage used to measure the angular position of a 12 Watts DC Motor. b) Schematic diagram of the electronic circuit.

b) Filter design by numerical simulations

Figure 2a shows the signal provided by the potentiometer after that is turned to a digital signal by means of the DAC model ADC1241CIJ [7]. This position-time curve represents a displacement of the motor's shaft of 6 turns in counter clockwise. The initial position of the motor corresponds to 1.4 turns (0.7 Volts) of home position, note that these *data* are mixed with signals at different frequency. The frequency contents of a signal and their power spectrum density (PSD) can be obtained through the FFT command of MatLab™. In this application, the PSD of the data is obtained with the purpose of quantifying the frequency range of the data and to select the cut-off frequency for the filter. Figure 2b displays the PSD of the signal showed in Figure 2a. Note that, the signal of interest is within a range minor than 10^1 Hz. Thus a low pass filter was selected with a cutoff frequency $f_c = 3.18$ Hz, recording that $\omega = 2\pi f_c$, we obtain a $\omega = 20$ rad/seg.

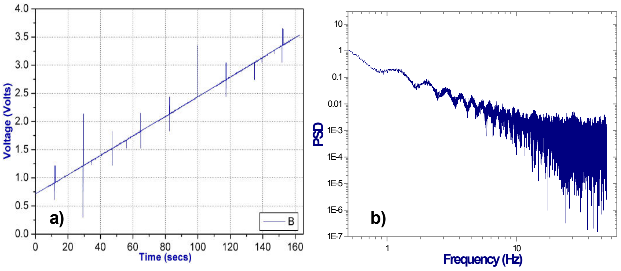

Figure 2. a) Signal provided by the potentiometer, which it is given in Volts. **b)** Power spectrum density of the signal showed in Figure 2a.

Once with the cutoff frequency, we must select the order of the filter that shows better performance to remove the noise but keeping up the amplitude of the signal. From numerical simulations, we found that a second order filter is enough to attenuate most of the noise. Thereafter, we proceed to prove the most common mathematical structures of filters, taking into count the same order and cutoff frequency of the filter.

Table 1 shows each transfer function of the designed low pass filters, where for Bessel filter the Rp = 0.2 dB, and Rs= 40 dB decibels are of peak-to-peak ripple and the stopband attenuation, respectively.

Table 1. Transfer functions of the second order filters, n = order of the filter, w_n = cutoff frequency in rad/secs, 's' because is an analog filter.

Command (n, wn, 's')	Transfer function
[A,B]=butter(2, 20, 's')	$Butt = \dfrac{400}{s^2 + 28.28s + 400}$
[B,A] = besself(2,20)	$Bess = \dfrac{400}{s^2 + 34.64 + 400}$
[B,A] = chevy1(2,.1,20,'s')	$cheb = \dfrac{1310}{s^2 + 47.45s + 1326}$
[A,B]=ellip(2,.2,40,20,'s')	$Ellip = \dfrac{0.01005s^2 + 923.2}{s^2 + 38.34s + 944.8}$

Once transfer functions are obtained, the performances of the filters are proved in simulations before their implementation. For this, the transfer functions are converted to discrete time with the *c2d* command of Matlab and with the *filter* command it is possible filtering the real data. Figure 3a, shows a comparison in numerical simulations of the performance of the four filters designed. Figure 3b, displays a zoom of Figure 3a, where is easy notice that the response of Bessel (black line) and Butterworth (green line) is more acceptable that others. However, the Butterworth filter shows the better performance because it is the nearest to the real data; in other words, it removes most of the noise without changing the amplitude of the signal, so then this filter was selected for its implementation.

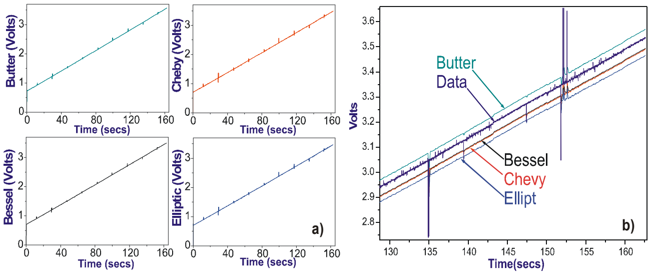

Figure 3: a) Comparison in the performance of the four filters. **b)** Zoom of the Figure 3**a**, in the range from 2.5 to 4.0 Volts.

Equation (1) shows the transfer function of the Butterworth filter selected,

$$\frac{x(s)}{x_P(s)} = \frac{400}{s^2 + 28.28s + 400} \qquad (1)$$

where $x(s)$ is the output of the filter and $x_p(s)$ is the signal provided by the potentiometer. Regrouping terms in equation 1, we have

$$\left(\left[s^2x(s)\right]+28.28\left[sx(s)\right]+400\left[x(s)\right]\right)=400\left[x_P(s)\right] \qquad (2)$$

Next, applying the Laplace inverse transform in equation 2, we obtain the continuous-time representation of the filter.

$$\ddot{x}=-28.28\dot{x}-400x+400x_P \qquad (3)$$

Later, using backward finite differences with a sampling-time Δt = 10 milliseconds (ms) = 0.01 seconds $\ddot{x}\cong\frac{x_k-2x_{k-1}+x_{k-2}}{(\Delta t)^2};\dot{x}\cong\frac{x_k-x_{k-1}}{\Delta t}$, and substituting in (3) we obtain the difference equation representation of the filter:

$$x_k=\left[A_1x_{k-1}-A_2x_{k-2}+A_3x_{Pk}\right]\left[\frac{1}{A_4}\right] \qquad (4)$$

where A_1 = 6BDH, A_2 = 2F4H, A_3 =1EH, A_4 = 3E8H are the filter's coefficients in hexadecimal base. Figure 4a, shows the experimental response of the digital filter programmed into MC and where is easy to notice that its performance is better than the response obtained in numerical simulations, see Figure 4a.

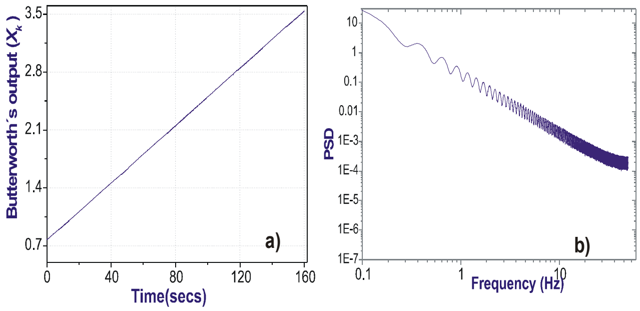

Figure 4. Performances of the Butterworth filter. **a)** Experimental responses. **b)** Power Spectrum density of the filter's output (X_k.).

Figure 4b displays the power spectrum density of the filtered signal x_k and in comparison with Figure 2b, corroborates that the noise has been removed. Here to obtain the output x_k of the filter the 8-MC should performs arithmetical operations with 16 and 32 bits in fixed point. Nevertheless, in spite of these calculations, the micro performs the filter's output in less 1 ms; thus this program can be used to filter signals in applications with low dynamic.

IV. Implementing a PI controller

In this section, we show how an 8-MCs can be used to control the angular motion of a DC-Gearmotor based on the measure of the angular position of shaft's motor.

The Figure 5, describe of the sample-data system, with the control scheme and the stages that conform our design. Here, $r(kT)$ denotes the reference signal, u(t) is the control voltage, $y(t)$ stands for the output signal, $\hat{y}(kT)$ represents the filtered potentiometer's output, $\hat{e}(kT)$ is the error defined by $r(kT)$-$\hat{y}(kT)$, $d(t)$ designates a disturbance input to the plant, and $n(t)$ describes the noise in the sensor [8]. In this nomenclature (kT) is used to represent a discrete signal with a sample-time given by T.

The analog signal $y_m(t)$ provided by the potentiometer, is converted to a digital signal by means of the ADC device. In this stage the sensor generates electrical noise in the measured signal, thus a FIR filter is implemented before to feedback the signal.

The control scheme is a classical PI programmed into the MC which calculates the error $\hat{e}(kT)$ and performs the controller action in order to provide the output $u(kT)$. Such an output is then converted to the analog form, $u(t)$, by the DAC stage [9].

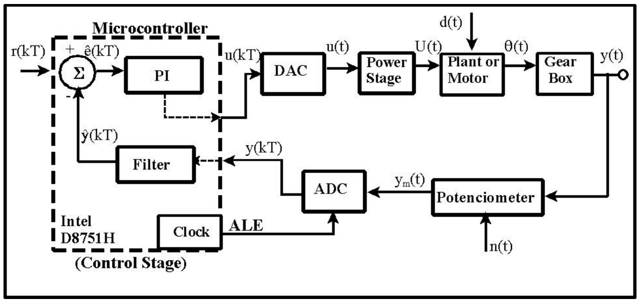

Figure 5: Block diagram of the position's control scheme

In this way, the control stage contains the control unit, and comprises: (i) the control algorithm, (ii) the FIR filter, and (iii) the libraries designed to perform arithmetic operations like sum, subtraction, multiplication and division with 16 and 32 bit in fixed point.

The digital to analog stage is carries out by an 8-bit converter model DAC0800 [9]. The control input is given by u(t) = 10*[(-255/256) + (2(u(kT))/256)], where u(kT) is the input calculated by the PI algorithm and u(t) is the analog voltage supplied to the motor, this converter is used in bipolar output with a +/- 10 Volts range.

The power stage was implemented with power PNP and NPN Darlington transistors, model TIP-125 and TIP-122 with an operation ranging from 0 to +/-10 Volts. This stage amplifies the output current of the DAC from 10 mA to 4 Amperes to provide the necessary current to the motor.

The plant is a 7.5 Watt DC Gearmotor; Model 317A248-1, at 12 Volts, here the input to the plant is a DC voltage and the measured plant output is the angular displacement, θ(t).

The measurement stage is carries out by a precision potentiometer; model S-7286, with a 5 KΩ resistance and 20 maximum turns, where the angular position of the motor is calculated through the following relationship: 1.38 mV = 1º = 0.017 radians. For the analog to digital stage, a 12-bit plus sign analog-to-digital converter model ADC1241CIJ is used. The minimal and maximal converted voltage is ą +/-1.22 mV signal and +/- 5 Volts respectively.

a) *PI-Controller Synthesis*

The angular position of a DC motor can be represented by a third order transfer function [10] (see also reference [11], page 54, chapter 2),

$$\frac{\theta(s)}{u(s)}=\frac{K_T}{s\left[(Js+B)(Ls+R)+K_TK_E\right]} \qquad (5)$$

where θ(s) is the angular position in rpms, $u(s)$ is the control input in Volts.

For this Gearmotor, the follow parameters are known and the units are in the English system [12]: K_T is the torque constant = 1.74 oz.-in./Amp, K_E is Back EMF or electrical constant = 1.289 e^{-3} Volts/RPM, J is the rotor inertia = 5.2 e^{-5} oz.-in.-sec^2, B is the viscous friction coefficient of the rotor in oz.-in./sec (unknown), R is the electrical resistance = 67.9 ohms, and L is the electrical inductance L = 2.564 mH. On the other hand and due to the high value of the Gearmotor torque (1250 oz·-in = 8.8 Nw-m), the friction present in the gearbox and the viscous friction coefficient in the motor are assumed negligible and therefore, from equation 5, we obtain the following plant for the motor, G_m:

$$G_m = \frac{\theta(s)}{u(s)} = \frac{775.8}{s\left(5.94e^{-5}s^2 + 1.57s + 1\right)} \qquad (6)$$

The PI parameters are found by means of the root locus method design in continuous-time [13, 14]. These parameters are chosen such that the closed-loop system is stable and satisfying some performance requirements. Thus, the PI parameters are chosen such that the closed-loop contains a pair of dominant complex poles. In this case, we place the poles such that the damping ratio is 0.9 and the settling time is 6 seconds. Thus, the parameters of the PI-controller were K_c = 250 and τ_i = 0.89. and the poles in closed-loop with these parameters are: P_1 = [-21.1898], P_2 = [-18.4851], P_3 = [-6.0692], P_4 = [-1.6261 + 0.9572j], P_5 = [-1.6261 - 0.9572j], P_6 = [-0.0036] and the zero is Z_1 = [-0.0036].

b) Digital Implementation

Once the controller in continuous-time is designed, the stability of the closed-loop system in discrete-time should be proved, of this analysis was found that the poles and zeros of this transfer function lying in the unit circle, which guarantees stability in discrete time. At this point, we proceed with the implementation of the PI-controller and digital filter. Thus using the typical representation in discrete-time for the PI-controller,

$$U(z) = \left[K_p + \frac{K_i}{(1-z^{-1})}\right]e(z) \qquad (7)$$

and using the finite difference approach, substituting z by k, and z^{-1} by k-1. We have: $U_k = U_{k-1} + K_p e_k - K_p e_{k-1} + K_i e_k$, where $K_p = K_c - (K_i/2)$ and $K_i = K_c T/\tau_i$, which gives K_p = 248.6 and K_i = 2.8 with a sample-time of T = 10 ms. We obtain:

$$U_k = (U_{k-1}) + \left[(K_{INT})e_k - (K_{PROP})e_{k-1}\right]\left[\frac{1}{CoefPI}\right] \qquad (8)$$

where $K_{PROP} = K_p$ = *9B6H*, $K_{INT} = K_p + K_i$ = *9D2H, and CoefPI = 64H*, in the hexadecimal base.

c) Experimental Results.

Figure 6(a) shows the closed-loop response of the plant for three equal changes in the reference signal, each 78 mV = 0.960 radians. The magnitude of the steady state errors is 3.8 %, equivalent to 0.036 radians in the angular position, this error fulfills the performance's requirements.

Figure 6(b) shows the voltage calculated by the PI eq.(8) for each change in the reference signal (0.960 radians). Here $u(t)$ is the voltage supplied to the motor in analog form and amplified in current by the power stage (see Figure 5).

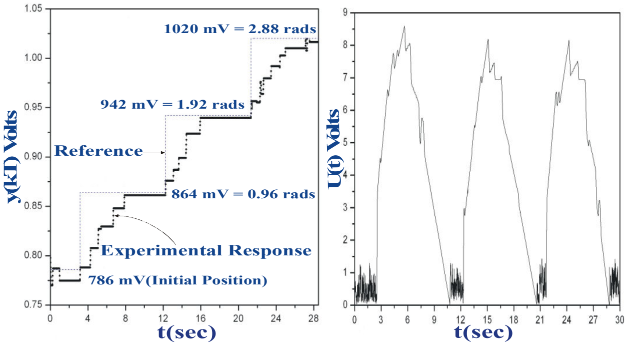

Figure 6. a) Closed-loop response of the motor for a reference signal 0.960 radians. b) Control voltage calculated by the PI-controller and supplied to the motor for each change in the reference signal.

The maximum current required by the motor in this application is 300 mA. Note in Figure 6(b), that there exist noisy measurements in u(t). However, the noise is a high frequency signal and does not affect the motor's angular position; this can be corroborating in Figure 6(a).

It is worth mentioning, that the time spent for the MC to obtain u_k and $\hat{y}_k$ for each sample-time was of 8 milliseconds (in the worst case) and never exceeds the sample time of the embedded system. To avoid mistakes, the sample time is calculated for an internal timer of the MC, which it actives an overflow flag every 10 ms, and resume the loop control until that ê(kT) is zero, when this had happened the loop is stopped and the MC waiting for a new reference.

V. Implementing a PID controller

In this section, the problem control consists in controlling the angular velocity of a DC-gearmotor by using a PID controller. Figure 2 depicts the block diagram of the sample-data system. The nomenclature is similar to the showed in Figure 5, except that in this case the ADC stage and the filter are not used, because the motor used has already an optical encoder attached [15].

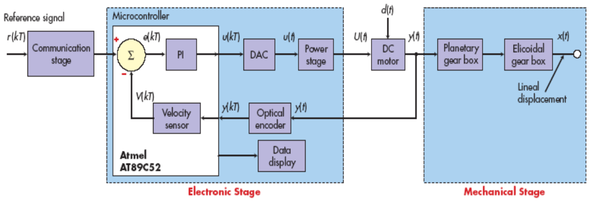

Figure 7. Block diagram of the servo control scheme.

For clarity in the presentation, the velocity control algorithm is briefly described. Firstly a reference signal is given to the MC via serial port. In that moment the MC enables the "*VS*" to count during two ms the pulses per revolution provided by the optical encoder, with this signal, the MC calculates the error signal $e(kT)$ and the voltage $u(kT)$ by mean of the PID controller. Such signal is then converted into analog form, $u(t)$, by mean of the DAC [16]. Thereafter, this control input $u(t)$ is amplified in current by the power stage and applied to the motor. This loop is made as many times as necessary until that the error signal $\hat{e}(kT)$ is near to zero, as the control objective is reached as the loop is stopped and the MC waits for a new reference

a) Description of the instrumentation

The control unit is also based in an 8-MCs specifically the model AT89C52 by ATMEL®, and contains programmed i) the control algorithm PID, and ii) the velocity sensor.

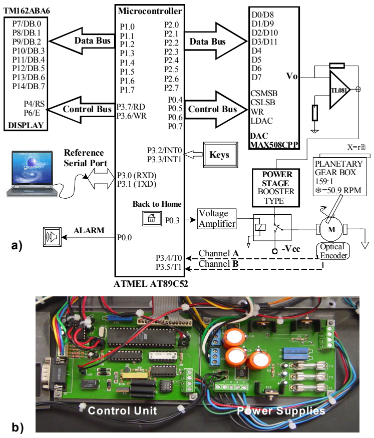

Figure 8. a) Schematic diagram of the control system, b) Electronic boards designed for the embedded system.

Figure 8(a) depicts the schematic diagram of the electronic stage. Figure 8(b), shows the electronic boards designed to control of the DC gear-motor. The left card, contains the AT89C52-based control unit, and the right card, the power supplies that can supply 3 Amperes.

Digital-to-analog stage (DAC): In this stage a MAX508 by MAXIM© performs the digital-to-analog conversion with a 12-bits resolution and an output voltage range, 0V to +10V. The resolution voltage is given by u(t) = 5*[(u(kT))/2^{11})], where, u(kT), is the input code calculated by the PID and u(t) the analog voltage supplied to the motor [16].

Power stage: The power stage is implemented using a TIP-41C (NPN power transistor) in a "BOOSTER" current amplifier configuration, with an operation range from 0 to +10V. In this stage, the DAC's output current is amplified from 10 mA to 3 A to provide the current requested by the motor. This stage also contains an auxiliary circuitry to supply a negative voltage of -10.47V to return the motor at home position.

Measurement stage: The motor velocity is measured by mean of the *VS* programmed into the MC and the optical encoder HEDS5540G, which provides 360 pulses per revolution (PPR). The *VS* works as follows: The optical encoder is connected to the external timer/counter T/C0 of the MC and with the timer 2 a capture time of 2 ms is established, over which the T/C 0 is incremented for each digital pulse provided for the encoder, thus the units of the *VS* are pulses per ms.

Plant (Gearmotor): The motor used is a 12W DC gearmotor Model 2342S012CR, with a nominal voltage of 12 Volts. Its foremost characteristics are: i) 637 oz.-in. at normal load ii) gear reduction radio 159:1, iii) 5.3 RPM at 12V with load, iv) maximum current consumption at load 75 mA. The control input is a DC voltage and the measured output is the angular velocity ω(t) in RPMs [15]. The motor model was obtained by identification input-output; where steps in volts were applied to its terminals and the angular velocity was measured. The nominal model found was

$$G_M = \frac{c(s)}{u(s)} = \frac{611.26}{12.085ms+1} \tag{9}$$

where the *time-constant* is given in milliseconds, for more details about of the identification of this model, see reference [1]. The Figure 9 shows a comparison between the model response (red line) and the actual motor's response (black line). From this graph we can deduce that the model is capable of reproducing the motor dynamic and therefore, can be used to tune the PI parameters. It is important to appoint that the inputs given to the model were similar in magnitude and time to the voltage applied to the motor in the experimental identification.

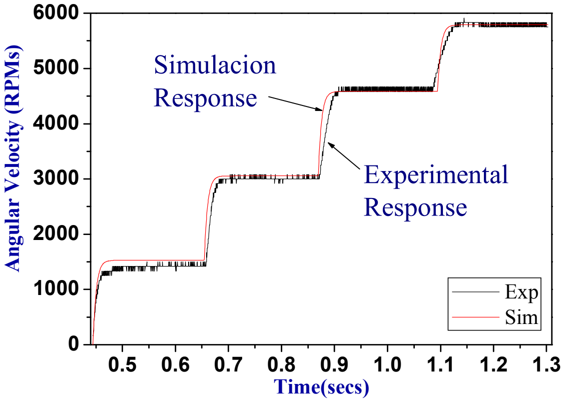

Figure 9. Comparison between the nominal model's response, versus the actual motor response.

The selection of the PID parameters is also based on *root locus* design using the control system toolbox of MatLab®. Figure 10 shows in numerical simulations the closed-loop motor response for different values of K_p and using the model given by equation 9, with a τ_i = 50 ms.

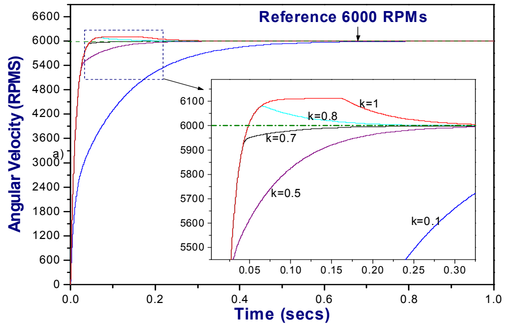

Figure 10. Numerical simulations of closed-loop motor response, considering different values for the proportional gain, K_p.

Note in this graph, that, for $K_p > 1$, the system's response is faster than for $K_p < 1$. However, for $K_p > 1$ the motor exhibits oscillations, which are undesirable in positioning applications. To avoid this, $K_p = 0.5$ was selected. Following section shows the experimental motor response with the gains selected.

b) Experimental Results

Equation 10 shows the expression of an PID controller in continuous time.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t) \qquad (10)$$

Taking the derivative of (4), we obtain,

$$\dot{u}(t) = K_p \dot{e}(t) + K_i e(t) + K_d \ddot{e}(t)$$

This expression is difficult to implement into MC, so then we can approximates the derivative by mean of the forward finite difference approach, where ΔT is the sample time.

$$\dot{u}(t) = \frac{u_{k+1} - u_k}{\Delta t}, \quad \dot{e}(t) = \frac{e_{k+1} - e_k}{\Delta t}, \quad \ddot{e}(t) = \frac{e_{k+1} + e_{k-1} - 2e_k}{(\Delta t)^2}$$

Substituting the PID parameters selected, $K_p = 0.5$, $\tau_i = 50$ ms and $\tau_d = 2$ ms, with a sample time $\Delta T = 10$ ms, the discrete-time representation of the PI controller is obtained

$$u_{k+1} = u_k - 0.6e_k + 0.55e_{k+1} + 0.1e_{k-1} \qquad (11)$$

The Figure 11a shows the experimental motor response using equation 11. In this example, the reference given to the MC was of 6000 RPMs. In practice, the reference is reached by the motor in 0.01 min (600 ms) and without overshoot, nevertheless note in the zoom the Figure 11a that there is a steady-state error, although this magnitude represents the 1.3% of the steady-state value, which is acceptable in practice design. Figure 11b shows the control input computes by the PI and applied to the motor for reach the reference imposed.

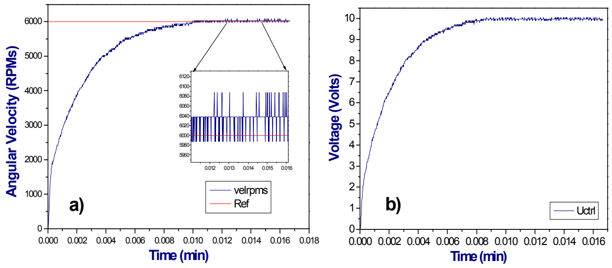

Figure 11. Motor closed-loop control **a)** Experimental motor response for a reference of 6000 rpms b) Control voltage calculates by the PID for reach the reference.

VI. Conclusiones

In section III, an educative example of the design and implementation of a digital filter using 8-MCs has been presented. In this procedure the difficulty could be the filter's design; however in those cases where the designer have already the transfer function of the analog filter, the example of implementation can be useful. Other advantage of this configuration is that thanks that the assembly code of the program is small, it can be used in closed-loop control applications where could be necessary filtering signals before they can be feed backed. For more details in this kind of applications, see reference [2].

By the other hand, in section IV and V, we showed that a feedback control scheme can be implemented in a low cost 8 bit MC. In the case of the velocity control depicted in section IV, the performance of the embedded system was better, thanks to an ADC stage and the filter is not required. Thus, the time spent by the MC to carries the motor to the maximum reference 9100 RPMs, was (in the worst case) 866 ms. So we believe that this configuration can be an alternative to control some mechanical/electrical actuators as valves, or for small DC motors in mechatronics devices with low current consumption.

For the case of the position control where the MC must to performs 16 and 32 bits operations to obtain the filter's output and executes the PI controller, the time spent for that the motor to reach the reference imposed was of 8 seconds in the worst case, even so, this configuration can be useful in the control of systems with slow dynamic such as heat process. Moreover, selecting an 8-MC of 40 or 60 MHZ this time can be reduced.

VII. References

[1] Hernández-Rosales C., Femat R. and Quiroz G., (2006), Make a Discrete-Time PI controller on an 8 bit Microcontroller, Embedded Systems Design, Vol. **19, No.1**, 28-43.

[2] Hernández-Rosales C., Femat R., Ruiz-Velázquez E., Solís-Perales G., (2005), A Standard Microcontroller Based Discrete-Time PI for Controlling the motion of a DC-Gearmotor, Journal of Applied Research and Technology, Vol. **3, No. 1**, 44-52,.

[3] Hernández-Rosales C., Femat R., (2004), *Presente y Futuro en el Desarrollo de una Bomba para el Suministro de Insulina*, Diabetes Hoy para el médico y el profesional de la salud (Revista de la Federación Mexicana de Diabetes), Vol. 5, No. 5, Pp.1290-1293,.

[4] Maurice B., (1998), *ST62 Microcontrollers drive home appliance motor technology*, AN885/1196, Application Note, ST microelectronics companies, www.st.com.

[5] Katausky J., Horder I., Smith L., *Analog/Digital Processing with Microcontrollers*, AR-526 Applications Engineers Intel Corporation, www.intel.com.

[6] (1999), An introduction to digital filters, Aplication note AN9603.2, Intersil, www.intersil.com

[7] Data Sheet, November (1994), *ADC1241CIJ Self-Calibrating 12-Bit Plus Sign μP-Compatible A/D Converter with Sample-and-Hold*, National Semiconductor Corporation.

[8] Franklin G. F., Powell J. D., (1990), Digital Control of Dynamic Systems, 2d Ed., Addison-Wesley.

[9] Data Sheet: (1999), DAC0800/DAC0802 8-Bit Digital-to-Analog Converters, National Semiconductors Corporation,

[10] Franklin G. F., Powell J. D., (2002), Feedback Control of Dynamic Systems, 4d Ed., Prentice-Hall.

[11] De Vegte J. V., (1994), *Feedback control system*, third ed., Prentice-Hall, New Jersey 07632.

[12] Data Sheet: DC Permanent Magnet Planetary Gearmotors series CMM & CLL, E-2030 Pages 10-13. http://www.globe-motors.com/cmmgr_in.pdf

[13] Dorf R. C., Bishop R. H., (1995), Modern Control Systems, 7d Ed. Addison-Wesley.

[14] Ogata K., (1998), Ingeniería de Control Moderna, 3d Ed., Prentice-Hall.

[15] Data Sheet, *12 Watts DC-Micromotors Graphite commutation*, series 2342…CR, MicroMo Electronics Inc., www.micromo.com.

[16] Data Sheet, (1991), *MAX507/MAX0508, Voltage-Output 12-Bit DACs with Internal Reference*, 19-4338: Rev A: 9/91, Maxim Integrated Products.