Design and construction of a microcontrolled system to simultaneously control the servo motors of an educational robotic manipulator

Armando Aguiar de Souza Cruz Neto, armandoeng@yahoo.com.br
Renato Zanetti, renatoznt@yahoo.com.br
Antônio Augusto Torres Maia, aamaia@demec.ufmg.br
Universidade Federal de Minas Gerais – Escola de Engenharia

Abstract. *Industrial Robotics is a vast field with a high growth. There are many enthusiasts, students and teachers who build simple robotic manipulators using servo motors, due to its low cost and ease of use. These systems usually are microcontrolled and one of various types of microcontrollers used is the PIC. A robotic manipulator with five degrees of freedom can have five or more servo motors units, which can be controlled using a microcontroller. The microcontroller, however, has limitations to control a very large amount of servo motors efficiently, which may lead to use of a greater number of microcontrollers, making the system more costly and complex. The aim of this study is to design and build an electronic circuit capable to control simultaneously up to eight servo motors using only one microcontroller and three programmable timers. The number of servo motors to be controlled only depends on the number of programmable timers used. A Graphical User Interface (GUI) was created to make easier the interaction between the mircrontroller and the user. The developed system has been tested on an educational robotic manipulator with five degrees of freedom and seven servo motors. The results indicated that the proposed system is able to control the servo motors of the manipulator efficiently.*

Keywords*: Robotic Manipulator, PIC, Microcontroller, Embedded System*

1. INTRODUCTION

The industrial robotics is a vast field and it had a high growth in the last decades (Craig, 2005). Many are their enthusiasts, students and teachers who build robotic manipulators using servo motors and microcontrollers (µC). The servo motors are widely used due to its low weight, cost, size and good torque provided (Iovine, 2004). The servo motor has an internal control system that receives a pulse train (Fig. 1) in a frequency between 50 Hz and 60 Hz, where the pulse width ‘t’ indicates the position in which the servo should be. Then, this signal is compared with their current position, driving the servo according to that difference. For a good positioning precision, this signal must be sent continually. This artifice causes the servo to remain energized and capable to maintain in the position which was sent. Additionally, it can withstand a heavier load than normally would when turned off.

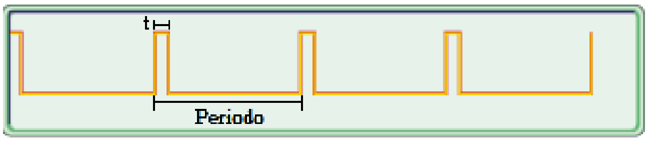

Figure 1 – Pulse Train example to be sent to the servo

Microcontrollers are often used to control the servo motors, which can be purchased with DIP package, allowing the assembly of prototypes in protoboards, as well its low cost and wide availability of documentation. An efficient way to control the servo motors would be the use of the internal timers of a µC (Sandhu, 2009), but that is limited by the number of timers available. Could be used more than one µC, but not without adding some complexity in terms of both software and hardware, plus an extra cost by the µCs. A less efficient solution would be dividing the timers between the servos, without controlling them continuously and simultaneously.

Within this context, the objective of this study was to design and build a circuit that allows simultaneous and continuous control of the servo motors of an educational robotic manipulator, using only a µC. The proposed circuit demands only a minimum computational time from the µC. In the Figure 2 is shown the educational robotic manipulator used to test the circuit, which was developed in the Automation and Control Laboratory of the Mechanical Engineering Department of UFMG (Bustamante et al, 2009).

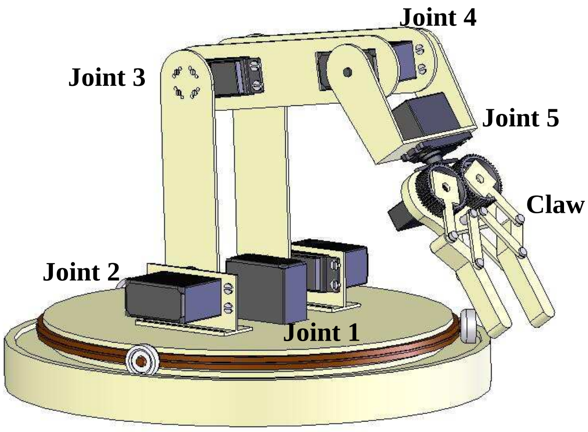

Figure 2 – Draw of the educational robotic manipulator used in this work

2. CONTROL CIRCUIT

The control circuit developed to control the servo motors can be divided into three areas identified by letters A, B and C (Fig.3): the µC circuit and their auxiliary circuit are in area A, the integrated circuit (IC) 8253 in area B and the auxiliary circuits in area C. The red tracks are the upper side of the circuit and the blue tracks are the lower side of the circuit. This study used the PIC18F4550 µC, but any other microcontroller could be used.

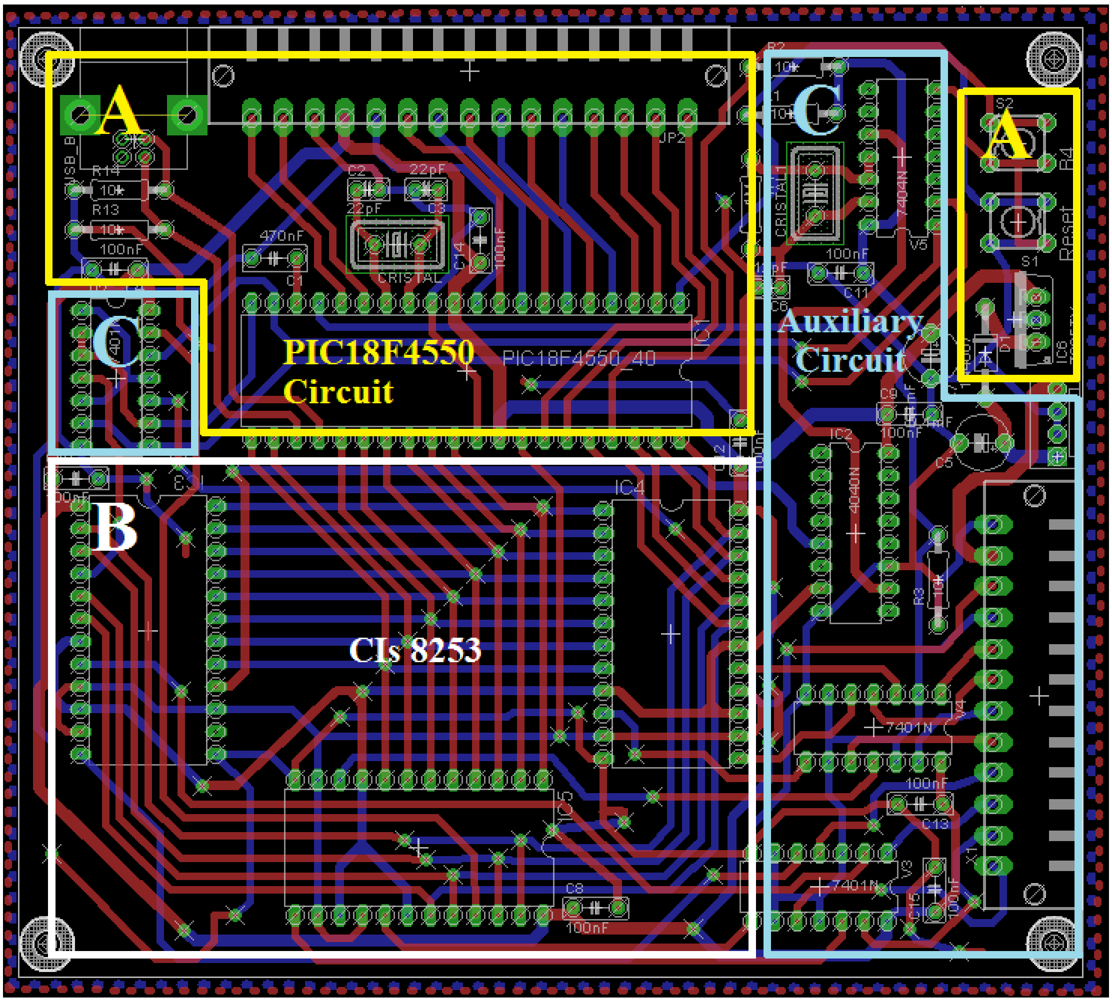

Figure 3 – Printed Circuit Board Layout

The IC 8253 is a programmable timer/counter, often used in motherboard chipsets, which has three independent internal timers (Sharp-MZ, 2002). In this study, the timers are configured to operate in the mode 1 and 3 only. The mode 1 changes the output logic level to high every time that the timer ends the counting, repeating the procedure after detecting each trigger rising edge, and is responsible for generating the pulses with a pulse width equal to the time counted. The mode 3 generates a clock signal with the period value to be counted, as long that the trigger pin remains at the level logic high. This mode is responsible for the external interrupt on the PIC and the trigger signal used by the other timers that work in mode 1. There is another IC called 8254, which has all the same functionalities of the 8253, except that can work with higher frequencies. Each internal timer has a trigger pin, a clock pin and an output pin, as shown in Fig. 4.

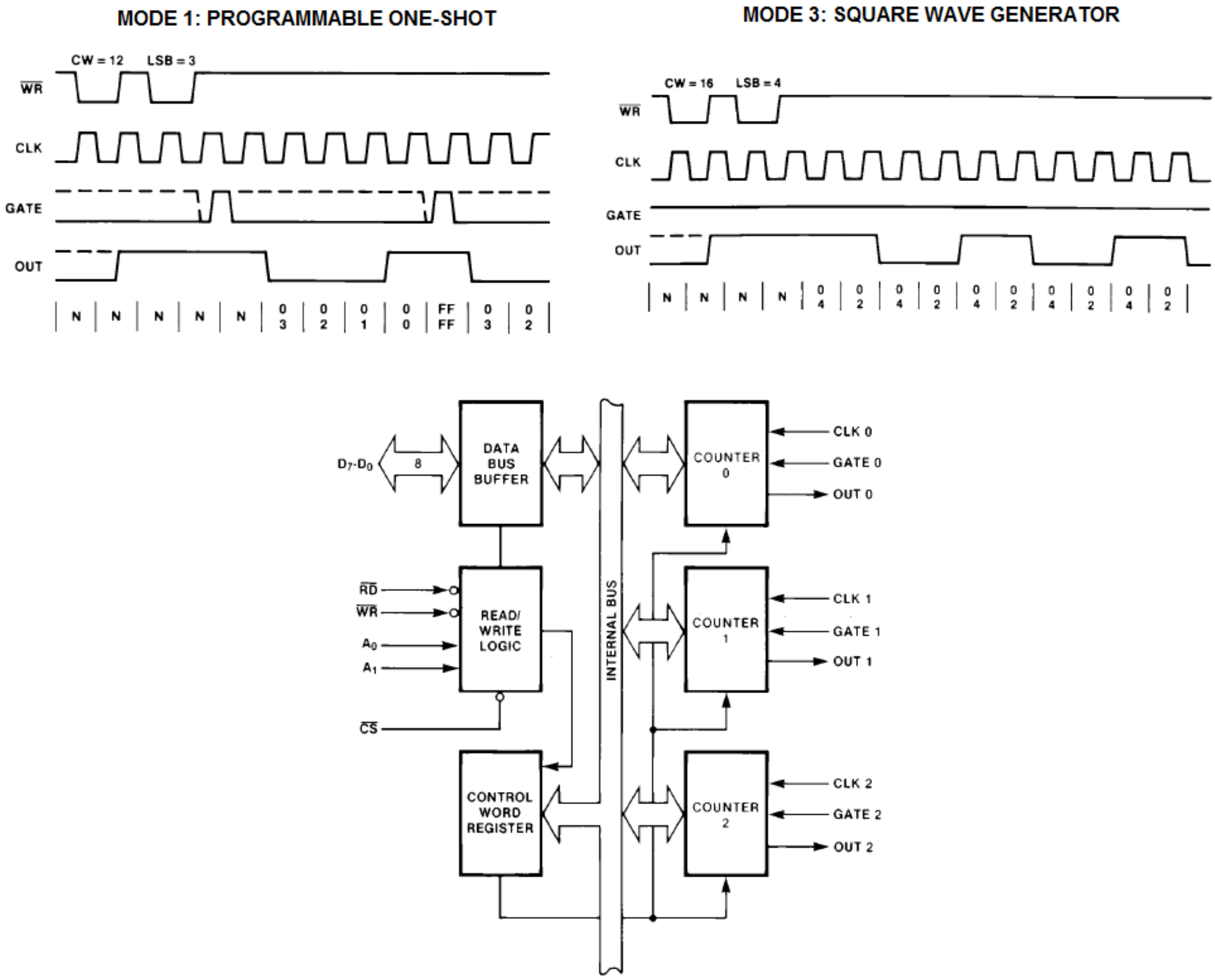

Figure 4 – Two modes of operation and the Block Diagram of 8253/54 IC

The 8253 can use a clock signal with the frequency up to 2.6 MHz and the 8254 up to 12 MHz. The internal timers operating at mode 1 are linked in their trigger input with the signal out of the mode 3. The pulses generated by the mode 1 are inverted in relation to the signals required by the servo motors, and that is corrected by using a logical NOT in the internal timers output.

The 8253 advantage is its ability to continually generate control signals, where each internal timer is responsible for one servo, without the PIC direct intervention. This continuity in the control signal allows greater precision in the servo desired position, as well as the capacity to sustain a greater load without relying only on the inertia of the servo motor gearbox.

The auxiliary circuit executes the details for the correct operation of 8253, performing three basic tasks, namely: choosing one of the three 8253 ICs to be accessed during the configuration process, generating the clock signal for each IC and inverting the output signal of the internal timers. The clock is generated by dividing the clock signal of a 4 MHz crystal by 4 and 16, resulting in a clock of 1MHz and 250 kHz, respectively.

Because the 1 MHz frequency used, the PIC is able to configure the 8253 timers with a resolution of up to 1 µs. One of the improvements to the system could be to use only one source of 1 MHz clock, saving tracks and ICs in the circuit and thus making it simpler and easier to manufacture. Another improvement would be to use a 2 MHz clock, improving the resolution to 1/2 µs. It is noteworthy that such changes do not bring any difference to the control signals sent to servos, thus not harming the results of this study.

The circuit was designed to allow the use of the microcontroller USB port and to access the unused microcontroller pins. The circuit has a 5 V voltage regulator, a reset button and a button connected to RB4 port which allows to record a new program using the PIC18 bootloader.

The PIC primary role in the system is to configure the 8253 internal timers at every two cycles of 50 Hz (or every 40 ms), thus allowing the velocity and acceleration control of the servo motors. The configuration acts basically on the internal timers that work in mode 1, setting the time to be counted or the width ‘t’ of the pulse. In order to avoid any interference in the pulse generated by the internal timers , the configuration process must occur in the falling edge of the 50 Hz clock.

Figure 5 shows the fabricated circuit, with the PIC and the 8253 ICS, working to generate the output signals necessary to control the servo motors. As can be seen in the oscilloscope, the pulses have a frequency of 50 Hz (20 ms).

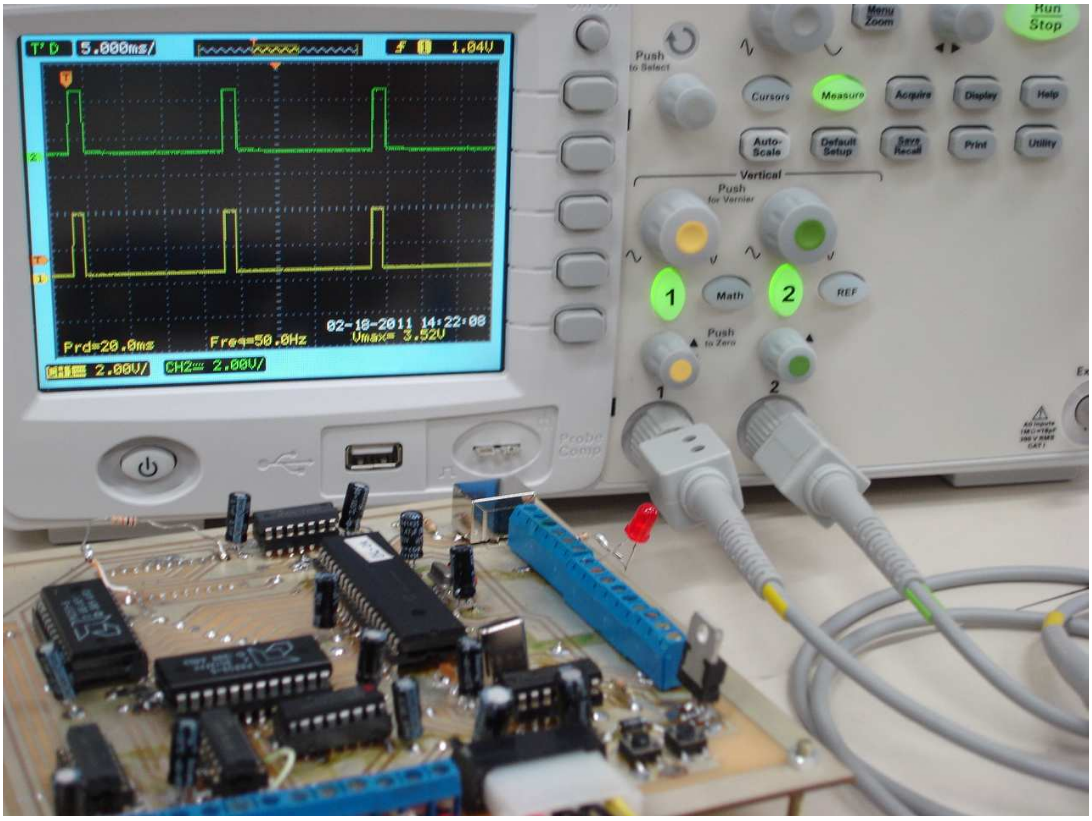

Figure 5 - Fabricated circuit and their signal output

2.1. Velocity Control

The purpose of the speed control is to make the servo motor obey a speed profile similar to the one shown in Figure 6. After setting the desired speed, the motor must accelerate to achieve it and then decelerate (Fig. 6a). In the case of reaching the halfway path without obtaining the desired speed, the servo will be obligated to begin the process of deceleration (Fig. 6b).

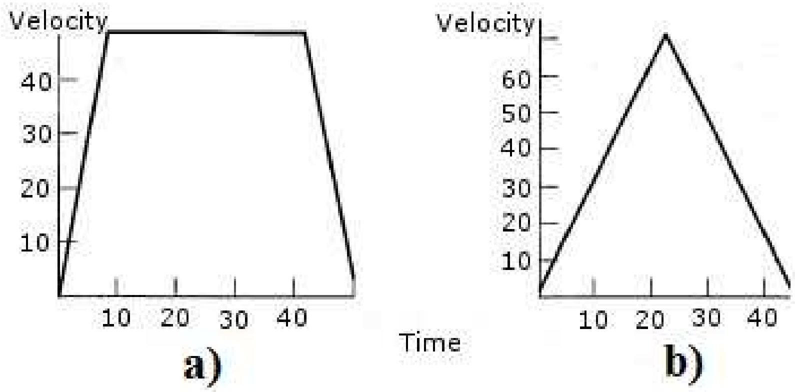

Figure 6 – An ideal servo motor velocity curve

The servo speed control is obtained by controlling the servo position in time. That is made by controlling the pulse width sent to the servo, increasing the width in a gradual and constant way, until it reaches the desired value. In that way, the velocity will be proportional to the increment of the pulse width. Acceleration is controlled in a similar manner, where the pulse width increment (velocity) is constantly increased until it reaches the desired value.

Figure 7a shows the speed and acceleration control, where the velocity is constantly increased until a determined value and then reduced to zero. Figure 7b shows the pulse width (in ms) of the control signal sent to the servo motor, as well its changing in time as an effect of its velocity. As explained, each pulse has a width 't', where the value of 't' indicates the position that the servo should move. The 8253 IC is responsible to generate these pulses by counting a time 't' in their internal timers, which was previously configured by the PIC, and repeating these countings at every 20 ms.

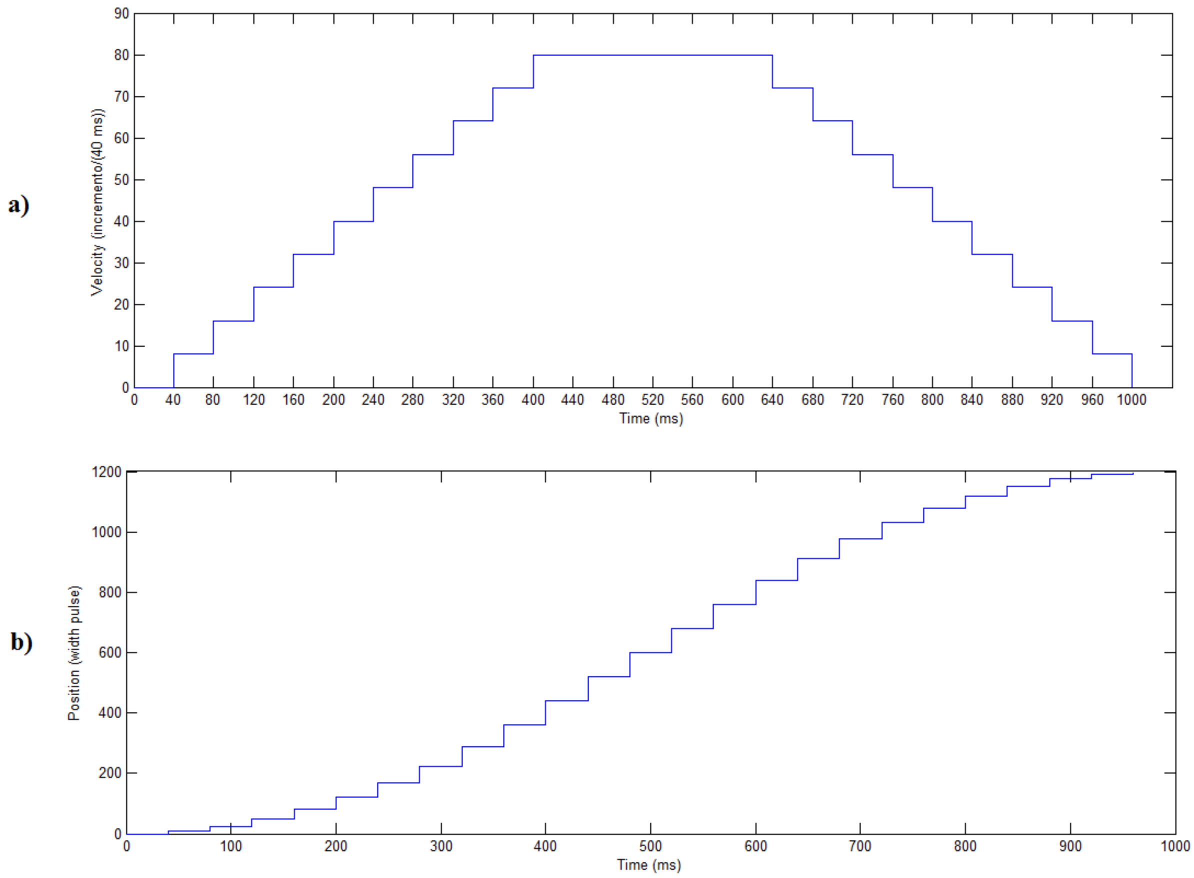

Figure 7 – Curve of velocity (a) and position (b) of the output signals sent to control the servo motors

The results were collected by eye observation, because the lack of sensors that could measure the servo motor position. The servo motors movements was alike this graph when they have little or no load. Significant errors that a human eye could notice occur when the servos need a higher torque, showing a delay at the start of their movement and consequent increase in speed at the end with an abrupt deceleration. This could be corrected by using different values of acceleration for different loads or making a closed loop.

2.2. Graphical User Interface

A graphical user interface (Fig. 8) was developed using MatLab, Mathworks Inc., which allowed an easy control of the robotic manipulator. The GUI communicates with the PIC using the HID (Human Interface Device) protocol. The advantage of this protocol is the ease of identifying the device to which you want to connect without the need for user intervention, as would occur in a serial communication.

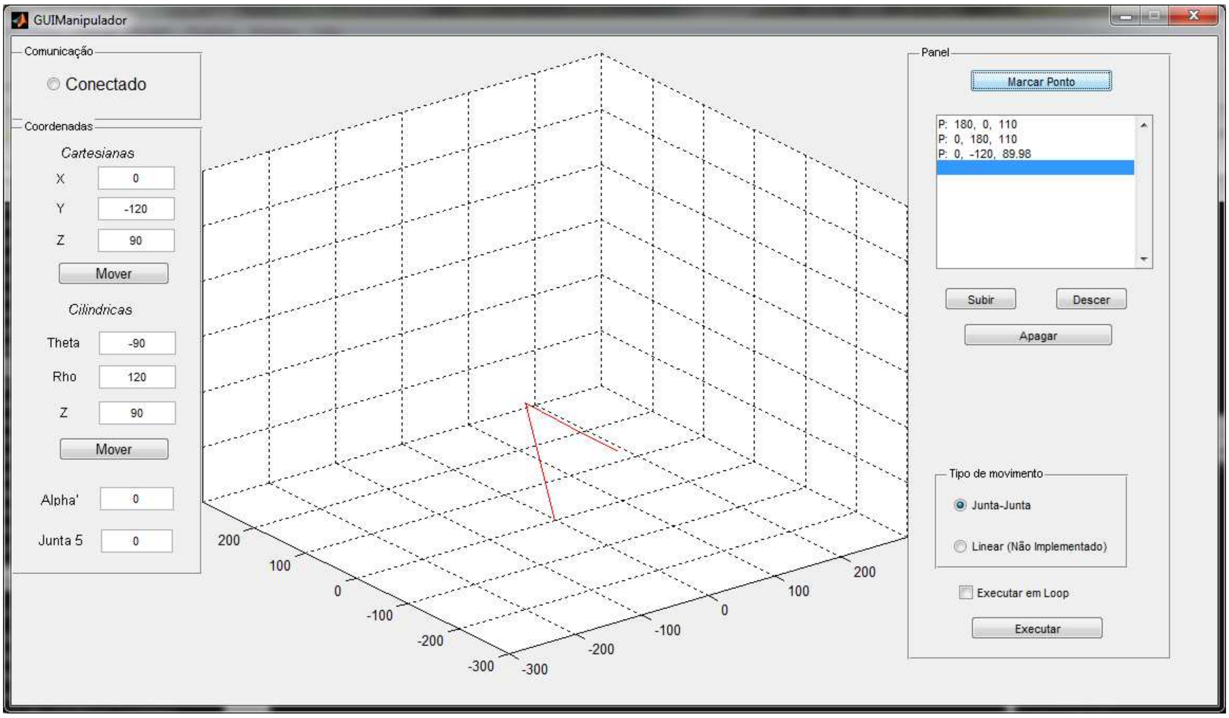

Figure 8 – Graphical User Interface developed on MatLab

The GUI is responsible for calculating the position at which the servos must be, given a point and its orientation in space, and to calculate the movement at Joint-Space Schemes of the manipulator, where the servos must start and finish the movement together (Craig, 2005). For this type of movement, the algorithm calculates the speed of the servos based on the speed of the servo that should travel the longest route. The result is sent to the PIC through USB port, that in turn will run the speed and acceleration control to reach the new position.

3. CONCLUSION

With the use of the programmable timer, the simultaneous and continuous control of the servo motors became computationally simple and efficient. The 8253 IC is easily found with an accessible cost. The only problem is the tracks complexity that forces the use of a circuit board with two sides. Now, the educational robotic manipulator built in the Automation and Control Laboratory gets closer to an industrial robotic manipulator on its movements, making it easier to understand and learn, and that is a contribution to the work done previously by Bustamante et al (2009). The built circuit can be used for any situation in which one of the five modes of the programmable 8253 timer is useful with any other microcontroller than the PIC18F4550.

4. REFERENCES

Bustamante, P. C., Maia, A. A. T., 2009, “Projeto e Construção de um Manipulador Robótico Educacional Feito em Material Compósito”, 5 Congresso Brasileiro de Engenharia de Fabricação, 2009, Belo Horizonte.
Craig, John J., 2005, “Introduction to Robotics – Mechanics and Control”, Third Edition, Pearson Prentice Hall
Iovine, John, 2004, “PIC Robotics – A Beginner’s Guide to Robotics Projects Using the PICmicro”, McGraw-Hill
Sandhu, Harprit S., 2009, “Running Small Motors with PIC Microcontrollers”, McGraw-Hill
Sharp-MZ, 2002, “Overview of the 8253 functions”, 27/03/2011, <http://www.sharpmz.org/mz-700/8253ovview.htm>

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.