Design of a Control and Acquisition System in an FPGA Platform for a Single Degree of Freedom of an Articulated Robot Via Wifi

Javier Ruiseco Lopez, Sergio Vergara Limón, M. A. Vargas Treviño, Fernando Reyes Cortés[1] y Amparo Palomino Merino, G. Humberto Bezares Colmenares

Benemérita Universidad Autónoma de Puebla, Maestría en Ciencias de la Electrónica opción Automatización, Puebla, México
javierruiseco_88@hotmail.com, svergara@ece.buap.mx, mavargas@ece.buap.mx, recf62@gmail.com, palomino@ece.buap.mx, theboy115@hotmail.com

Abstract. The Maestría en Ciencias de la Electrónica, opción en Automatización (MCEA) has been developing systems and control techniques for ROTRADI robots which it has built. With the purpose of improving this systems and techniques, a Fuzzy controller and WiFi communication is being implemented. Initially, ROTRADI robots were controlled by desktop computers via PCI cards and then a Field Programmable Gate Array platform was introduced. The advantage of an FPGA platform is that it allows building an embedded system that can manage all the needs of controlling a ROTRADI, discarding the use of a computer. A hyperbolic tangent controller was developed for the FPGA, but due to calculation timings and usage of logical elements a Fuzzy Controller was proposed in an attempt to reduce these aspects. On the other hand, WiFi was proposed to replace PCI communication in order to make use of a more innovative technique to manipulate the robot.

Keywords: Fuzzy control, robot, data acquisition system, WiFi.

1 Introduction

MCEA has managed to develop robot arm manipulators which are known as ROTRADI [1] (see figure 1) which initially were controlled using desktop computers running Windows 98 disk operating system in order to guarantee a real-time control. MCEA has been improving its control systems and Field Programmable Gate Array (FPGA) [2] control platforms were introduced, in which computers are no longer required because all control requirements have been implemented on an embedded system that has an FPGA as core of operation. The advantage of having an FPGA platform is to guarantee real-time control and simplify the elements that constitute the whole system. Giving this, a hyperbolic tangent controller was implemented in the FPGA. Due to the FPGA's logical elements limitations, the control of a single degree of freedom was proposed. In order to continue to improve the control system, a Fuzzy

Advances in Computer Science and Engineering
Research in Computing Science 67, 2013 pp. 93–103
Paper Recived 20-09-2013 and Acepted 21-10-2013

controller [3] is being developed in an attempt to reduce the use of FPGA's logical elements and reduce the clock cycles that hyperbolic tangent's calculations require. The objective of this improvement is to be able to add more capabilities in the FPGA in the future. On the other hand, current PCI data communication is intended to be replaced by IEEE802.11 [4] wireless protocol known as WiFi, whereby the user will be able to control the robot and analyze the controller's performance wirelessly. It is important to mention that this work is currently under development.

Fig. 1. ROTRADI robot.

Networking technologies have been applied in the area, for example with sensors [5]. This is due to the advances in microprocessors, memory, power consumption and cheaper technology in which networking with wireless technologies is becoming cheaper with and increasing market. On the other hand, robot related technologies have a broad area of application, from designing robots and control laws, through designing robots that are self-configurable [6] and therefore, can change their own shape.

2 System description

The system in general is constituted by software, firmware and hardware.

Software: The software that will be developed outside of the embedded system, whose function is to serve as a means of interaction between the user and the card controlling the robot.

Firmware: The FPGA programming, in which the control law resides, particularly a fuzzy controller.

Hardware: The electronics developed to implement the control and communication between the user and ROTRADI robot.

The system in development has he following block structure:

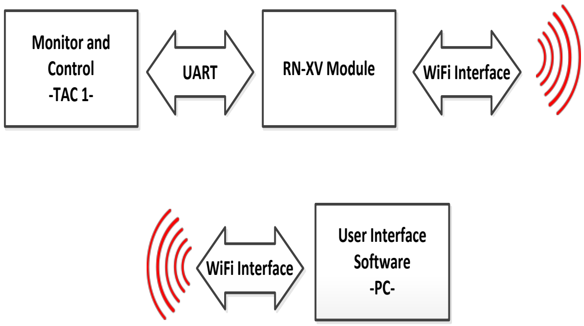

Fig. 2. General block diagram.

The proposed architecture for the system aims to create the embedded system which autonomously controls the robot, given the configurations of the user. The description for the block diagram is as follows.

User Interface: Is the software by which the user can perform the relevant configurations, such as the desired position of the degree of freedom, and to carry out the task of monitoring data from the robot. Particularly, the software will be developed in National Instruments Labview [7] which will also manage the Transport Control Protocol/Internet Protocol (TCP/IP) [8] communication to establish a WiFi connection.

RN-XV module [9]**:** Is by itself an embedded system which serves as a WiFi to serial converter. It will be integrated into the Tarjeta de Adquisición para Control 1(TAC1) to send and receive data wirelessly via the protocol IEE 802.11g, known as WiFi, "g" generation.

TAC1: Is the circuit board developed by the FCE MCEA in which the electronics regarding tasks of controlling the robot, reading motor encoders and controlling motor drivers reside.

ROTRADI robot: The ROTRADI robot (see figure 1) is an articulated robot with three degrees of freedom which first characteristic is that it is of direct transmission. This means that the motors of the joints are directly assembled to robot aluminum links, and therefore, no gearboxes are required. The motors have a linear response and they produce the necessary force to move the joints.

3 Acquisition system

3.1 TAC1 board

The main electronics hardware is the circuit board knows as TAC1 (see figure 3) developed by the MCEA. Initially the TAC1 was designed to function as data acquisition and send/receive data to the ROTRADI via the PCI bus. The TAC1 has been modified as a standalone embedded system which no longer requires a PC to control the ROTRADI robot therefore, the PCI bus is only used to apply supply voltages to the card by an external power supply.

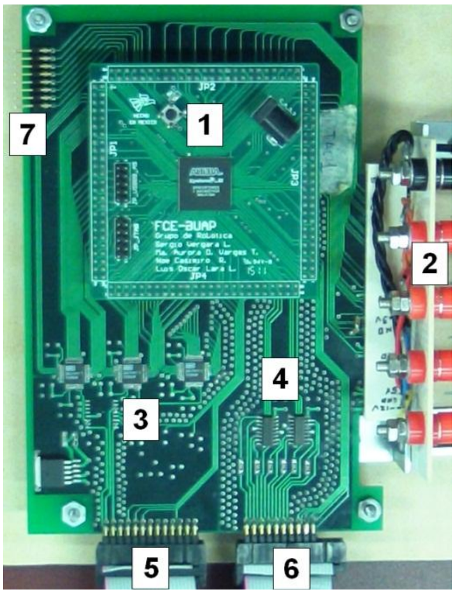

Fig. 3. TAC1 board.

1) FPGA Cyclone II EP2C8Q208C7: Is the core of the TAC1, in it resides the control law as well as the required firmware to acquire signals from the encoders and send the control data to the drivers.

2) Power supply bus: Voltages of 12V, 5V, 3.3V and -12V are applied.

3) Texas Instruments 7741Y DAC's: They translate the control law from the FPGA to voltage that serves as input to the motor drivers.

4) Differential filters SN65LVDT32B: They serve as filters for the incoming encoder signals.

5) Driver Bus: The voltage from the DAC's is sent to the driver via this bus.

6) Encoder Bus: The signals from the motors encoders are received via this bus.

7) H3 connector: A general purpose I/O bus with 3.3v and GND that has direct access to the FPGA.

3.2 RN-XV WiFi module

An important contribution of the present work is adding WiFi capabilities to the current TAC1 board. WiFi was chosen because of its increasing popularity and the innovation that a wireless robot control system represents. The core element of the WiFi interface is the RN-XV which will be integrated to the TAC1. The RN-XV module is based on the RN-171 module in which resides a firmware with the IEEE 802.11b/g TCP/IP protocol stack as well as the required electronics for the wireless transmission. It is a WiFi to serial converter and this is done via the serial protocol Universal Asynchronous Receiver Transmitter [10] [11]. The TCP/IP stack protocol [4] is a suite of protocols that has a layered architecture (see figure 4). It is formed by four layers; each one of them has its own protocols that work in order to establish a communication between TCP/IP. Particularly, WiFi resides in the bottom most layer, known as network access layer. Here resides the circuitry required to interpret the electric signals of the received/sent messages. The RN-XV has implemented in its firmware a mode of operation known as Access Point which is the one that will be used for this project. An access point is an infrastructure network that the user can connect to it and therefore control the robot. There are two protocols to use when sending data over TCP/IP, UDP and TCP. Due to the fact that TCP is a more reliable protocol to send data because it integrates means to determine if there was an error on the transmission, and on the other side UDP integrates rudimentary means to determine errors.

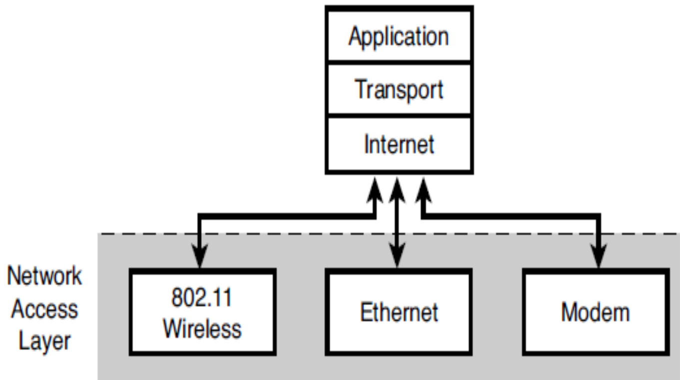

Fig. 4. TCP/IP stack.

When the module enters in Access Point mode in automatically broadcasts it's SSID which is the network's name and it will be ready to receive incoming connection on IP 1.2.3.4 and port 2000. The data is being sent over TCP with Labview, particularly in string type. As mentioned before, the module serializes the data received from WiFi via UART. In order to interface with the RN-XV's UART, a firmware in the FPGA has been implemented in order to establish communication between the FPGA and the WiFi module. The UART baudrate, network SSID, and other configurations where performed via a Telnet client, due to the fact that the module has implemented a Telnet server and therefore it can be configured wirelessly.

3.3 UART

The serial format of UART data may vary depending on whether it uses parity bits for error control. For this case, we will use 10 bits: 1 start bit + 8 data bits + 1 stop bit.

The signal is idle on logic "1" and activates at logic "0". In Figure 5 shows the format of the bit stream.

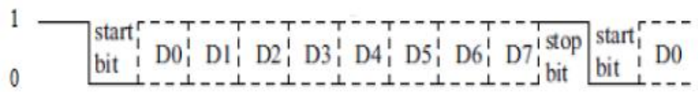

Fig. 5. UART bit transmission format.

The speed of bits is known as baudrate which is how many bits per second are being transmitted. The order of the data is from least significant to most significant. In the nomenclature of the UART, TX is for transition and RX is for receiving. It is important to establish a bit sampling at the middle of each bit transition, therefore, the RX sample rate is faster than the TX rate. The UART FPGA firmware was implemented in Altera Hardware Design Language (AHDL) based on state machines as can be seen in figure 6.

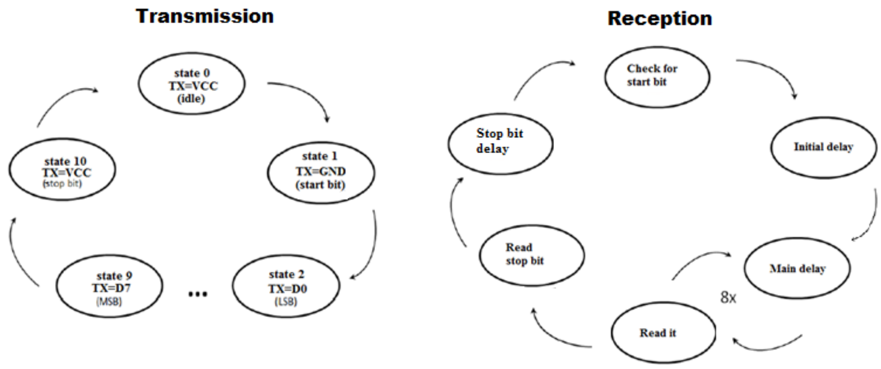

Fig. 6. UART Transmission and Reception state machines.

4 Control

The FPGA firmware currently has a hyperbolic tangent controller [1] with mechanical brake.

$$\tau = K_p \tanh \tilde{q} - K_v \tanh \dot{q} \quad (1)$$

Where $\tilde{q}$ is the position error, $\dot{q}$ is the velocity, K_p is the proportional gain and K_v is the derivative gain. The derivative section of the equation represents a brake to the control law. Since this controller uses floating-point units, the use of FPGA capacity is higher, at about 76% to control one degree of freedom. The reason that this work attempts to control one degree of freedom is due to the restriction of the amount of logical elements available on the FPGA, in addition, there must be logical elements left for the WiFi intercommunication, monitoring and user interface. In order to reduce the number of logic elements in use, and improve the clock cycles that hyperbolic tangent calculations require, it was proposed to implement a fuzzy controller as an alternative control law. The Fuzzy Controller has as input fuzzy variable the position error and as output variable the torque, which establishes the membership functions of the position error which range varies between $\pm 360^{\circ}$ (see figure 7), due to the fact that the robot joints are rotational. The nomenclature of the membership functions is **E** for error, **T** for torque, **N** for negative, **P** for positive, **G** for large, **M** for medium, **EP** for small error and **MG** to very large.

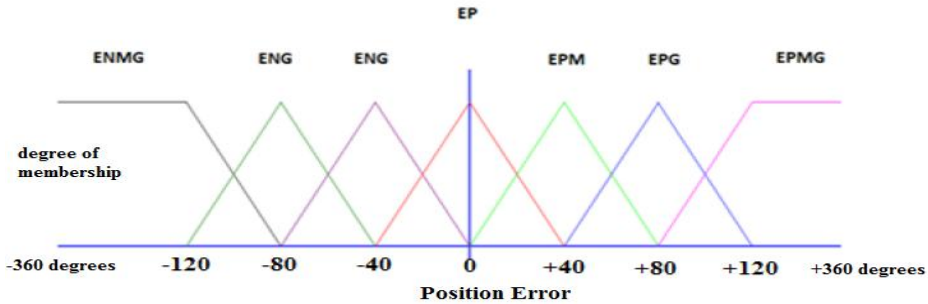

Fig. 7. Position Error membership function.

The rules were defined to be proportional to the error. Given a position error which magnitude is big, the output torque should be big. In contrast, if a position error is small, the output torque should be small. Having this, the fuzzy controller rules where defined as follows:

$$IF\ \mu ENMG, then\ \mu TNMG = \mu ENMG$$
$$IF\ \mu ENG, then\ \mu TNG = \mu ENG$$
$$IF\ \mu ENM, then\ \mu TNM = \mu ENM$$
$$IF\ \mu EP, then\ \mu TP = \mu EP$$
$$IF\ \mu EPM, then\ \mu TPM = \mu EPM$$
$$IF\ \mu EPG, then\ \mu TPG = \mu EPG$$
$$IF\ \mu EPMG, then\ \mu TPMG = \mu EPMG$$

Due to the fact that the output torque signal to the motor driver is a 16 bit number that represent a voltage range of $\pm 10\ volts$, the fuzzy output variable was defined as can be seen in figure 8.

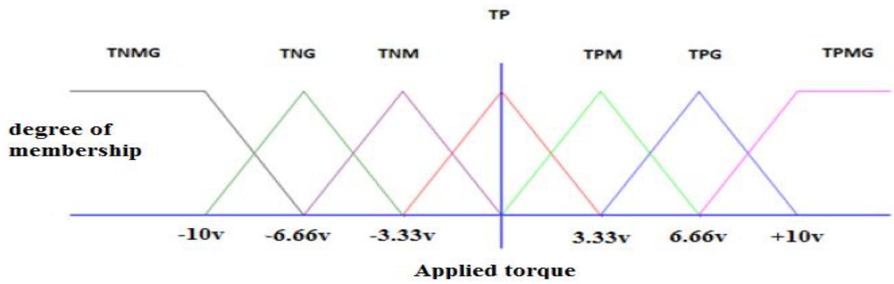

Fig. 8. Torque membership function.

The main state machines which constitute the fuzzy controller firmware are shown in figure 9, which also represent the fuzzy controller algorithm in use.

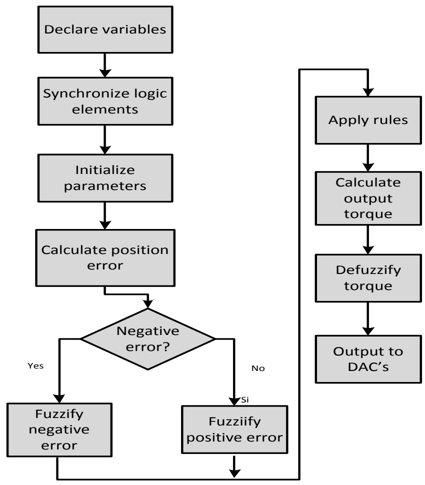

Fig. 9. Fuzzy controller algorithm.

5 Results

This work is in development and these are only preliminary results. The fuzzy controller implemented in AHDL code was able to reduce the number of logic elements in the FPGA compared with hyperbolic tangent. The fuzzy controller occupies about a ~ 62 % of logic elements in comparison to ~ 76 % of hyperbolic tangent. This is due largely to the operations with bit shifts. It is proposed as future work to optimize the firmware, particularly rethinking the architecture as the current one is optimized for speed and not for space. The response of the fuzzy controller is relatively fast, about 20 clock cycles to produce the result, which is about 0. 00000020 because the FPGA is clocked at 100Mhz. Due to the fact that an output gain adjusts were added to adjust the maximum torque that can be applied which the user can configure. Since the fuzzy controller delivers faster response than the hyperbolic tangent controller, in which a single Floating point operation takes about 11 to 14 clock cycles, it is intended that in the future the same fuzzy controller block originally designed to control one degree of freedom, can be used to control the three degrees of freedom of the ROTRADI robot by multiplexing data, since calculations are relatively fast. In order to ensure the proper functioning of the fuzzy controller firmware, a simulation of the firmware was performed (see fig. 10). First the theory of operation will be explained and then the results. In the first instance, the signals with which the motor operates will be explained. The controller takes the position of the encoder signals from the robot motor, which has a resolution of 0 to 655360 pulses per revolution, which equals the 0 to 360 degrees. As to the applied torque, the engine has a driver which converts a signal of ±10 volts into torque. The controller has 16-bit, value that is converted to a voltage with the DAC's of the TAC1; -10 volts equals a value of 0, 0 volts to a value of 32767 and +10 volts to a value of 65535 binary.

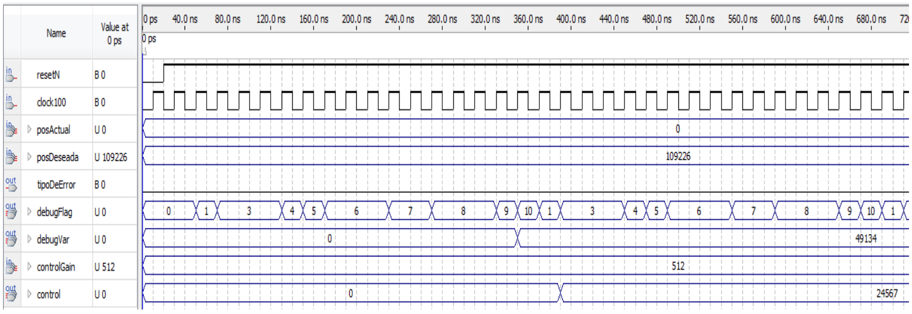

Fig. 10. Fuzzy Controller simulation.

Knowing these facts, the simulation will be then explained. Clock100 signal is the clock signal for the controller block , which operates at 100Mhz . For testing purposes, we assumed a current position of 0 degrees (home position) and a desired position of 60 degrees, which equals the 109226 if you were reading the encoder. It has an error signal "tipoDeError", which has a logical value of zero, if the error was positive and one if the error was negative. Since the error for the test values above is positive, "tipoDeError" has a logical value of zero. TorqueOut signal represents the output voltage to the motor driver (which will be converted from voltage to torque by the motor driver). TorqueOut has the value of 49134, a value that corresponds to 4.994 volts. This value was expected according to the rules applied, as it is expected to be applied half of the maximum torque by the controller in accordance with the output variable torque. Finally, there is a variable called debugFlag indicating the state machine in which is the FPGA firmware in order to monitor the proper functioning and the time it takes to process the information in each firmware's state machine.

As for WiFi communication, it was achieved to establish a bidirectional communication between Labview and the FPGA. Figure 11 shows values from a logic analyzer from an experiment where a data transfer over WiFi is being performed. In the RX (reception) test, the RN-XV module receives the data over WiFi and these are sent via UART to the FPGA. In the picture, number one indicates the pulses of when that data is read and number two indicates the actual bit being read. For this example the byte to read is AF hexadecimal. On the TX (transmission) test, transmission from the FPGA to the RN-XV module is shown in the following figure, where number one indicates the hexadecimal number 55 serialized. The RN-XV was configured to send data over WiFi each time is has a byte of data on its buffer.

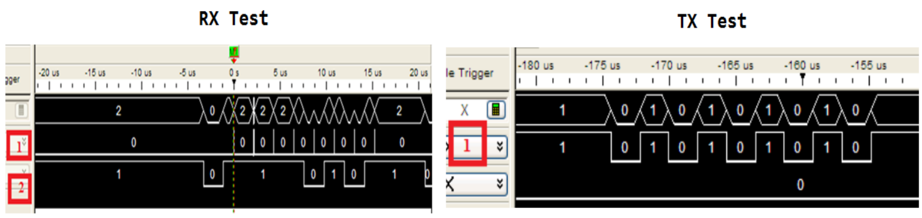

Fig. 11. Send/Receive data test.

6 Conclusions

Given the fact that the Fuzzy Controller has a faster response in comparison to the hyperbolic tangent, it is possible that in the future the same controller block can be used to control three degrees of freedom. The hyperbolic tangent uses floating point operations in which each ones requires an average of 11 clock cycles, and calculating the hyperbolic tangent requires considerable operations. In contrast, due to the simplicity of the Fuzzy controller and replacing floating point operation with bit shifts. Regarding the WiFi communication, sending commands to the FPGA and receiving data has been achieved.

References

1. F. Reyes Cortes, Robótica: Control de Robots Manipuladores, Primera Ed. Alfaomega, México D.F. Págs. 315,319. (2012).
2. I. Grout, Digital Systems Design with FPGAs and CPLDs, First Ed. Newnes, USA, Pág. 28. (2008)
3. V. Limón, Curso Lógica Difusa en la Vida Diaria, Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, (2012).
4. J. Casad, Teach Yourself TCP/IP in 24 Hours, Fifth Ed. Sams, Págs. (20120).
5. Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw., 11(1):2–16, (2003).
6. Distributed Control Diffusion: Towards a Flexible Programming Paradigm for Modular Robots. Ulrik P. Schultz Maersk Institute University of Southern Denmark.
7. National Instruments Labview, available at: http://www.ni.com/labview, [23 Septiembre 2012].
8. D. E. Comer, Internetworking with TCP/IP Vol I: Principles,Protocols and Architecture, Fourth Ed. Prentice Hall, Department of Computer Sciences, Purdue University, West Lafayette. (2000).
9. WiFly Command Reference, Advanced Features & Applications User's Guide. Roving Networks. USA. 2013. www.rovingnetworks.com. [Enero 9 del 2013].
10. R. Reese, Microprocessors from assembly language to C using the PIC18Fxx2, First Ed. Da Vinci Engineering Press, Págs. 259-263. (2005).
11. N. Laddha, A. Thakare, Implementation of serial communication using UART with configurable baud rate. International Journal on Recent and Innovation Trends in Computing and Communication. Volume: 1. Issue: 4. Pags. 263-268. (2013).