International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 02– Issue 03, May 2013

Fuzzy Control System for Robot Using Proprioceptive Algorithms

Elmo Batista de Faria
Universidade Federal de Mato Grosso
IARA: Intelligence Artificial, Robotics and Automation
Cuiabá, Brazil

Mauricio Fernando L. Pereira
Universidade Federal de Mato Grosso
CAP: Computing Precision Agriculture
Cuiabá, Brazil

Abstract—**The proposal of this work is to develop hardware and algorithms completely integrated with the sensorial systems of robots. The intention is adjust its movements in sensorial environments. In this context, the algorithms are called Artificial Proprioceptive Systems (APS), where these algorithms detail how to configure controllers for to provide the movement of robot-arm.This work initially introduces some concepts of control and intelligent controllers, approach an introduction on sensors and its more varied types and finally he considers a diagram to integrate the systems of movements that are used in robots. A considered experiment is the simulation of imperfections in joints of robots and to use the Artificial Proprioceptive Algorithms, to identify the imperfection and to provide a modification in the action of control to compensate this effect and guarantee that the system continues functioning normally.**

Keywords:Fuzzy Controller, robots, proprioceptive algorithms

I. INTRODUCTION

***"Proprioception** from Latin proprius, meaning "one's own" and perception, is the sense of the relative position of neighbouring parts of the body. Unlike the exteroceptive senses, by which we perceive the outside world, and interoceptive senses, by which we perceive the pain and movement of internal organs, proprioception is a third distinct sensory modality that provides feedback solely on the status of the body internally. It is the sense that indicates whether the body is moving with the required effort, as well as where the various parts of the body are located in relation to each other."[5]*

The position-movement sensation was originally described in 1557 by Julius Caesar Scaliger as a *'sense of locomotion'*. Much later, in 1826, Charles Bell expounded the idea of a *'muscle sense'* and this is credited with being one of the first physiologic feedback mechanisms. Bell's idea was that commands were being carried from the brain to the muscles, and that reports on the muscle's condition would be sent in the reverse direction. Later, in 1880, Henry Charlton Bastian suggested *'kinaesthesia'* instead of *'muscle sense'* on the basis that some of the afferent information (back to the brain) was coming from other structures including tendons, joints, and skin. In 1889, Alfred Goldscheider suggested a classification of kinaesthesia into 3 types: muscle, tendon, and articular sensitivity.

In 1906, Charles Scott Sherrington published a landmark work that introduced the terms *'proprioception'*, *'interoception'*, and *'exteroception'*. The *'exteroceptors'* are the organs responsible for information from outside the body such as the eyes, ears, mouth, and skin. The interoceptors then give information about the internal organs, while *'proprioception'* is awareness of movement derived from muscular, tendon, and articular sources. Such a system of classification has kept physiologists and anatomists searching for specialized nerve endings that transmit data on joint capsule and muscle tension (such as muscle spindles and Pacini corpuscles)[5].

In many robots is common the use of controllers, sensors, complex mechanics and intelligent programs. All this for create machines capable of moving like humans. These robots can develop tasks important and to substitute the action human in activities of risk and repetitive effort. These complex robotic systems moving with the use of links and innumerable other subsystems how communication, control movement and intelligent programs.

One difficulty consists on development of systems capable to integrate all these elements and still to promote the interaction between them. These systems have information of the environment and the system robotic to take decisions and to provide information for interaction with real world[1][2].

These robots systems initially can be composites of three great groups of systems involving the hardware and software, is they: Systems of security; Systems of movement and controlled action; Proprioceptive and sensorial systems. All the systems need the perception with the use sensors of innumerable classrooms and categories. Since simplest sensor of position and speed until the most sophisticated and complex as the sensors of torque and gyroscopes. The first group of systems acts in the measurement of destined sensors to keep the system in functioning. It among others consists of temperature sensors, positioning, that determine the stoppage so that yours hardware is not damaged seriously. These systems also guarantee the limits for the movements and correct functioning of engines and mechanical components of the robots[3][4].

The second group of systems is composed of programs to simulate common human tasks. It is a module programmed for the user for interaction and to execute the predetermined tasks. They consist in algorithms of movement, perception and execution of tasks.

The third basic group of algorithms is composed for the proprioceptive algorithms. They are the algorithms that monitor the signals produced in the two previous processes, beyond using specific sensors, to identify to anomalies and situations where the algorithms of tasks must be adjusted.

This process of adjustment consists mainly of modifying the action of control on the movement engines. Producing modifications in the tasks that are executed but guaranteeing that they occur normally. For this controllers, must be intelligent systems and that provides the possibility of adaptation on the actions of control. In this work used Fuzzy controllers, therefore he allows to the adjustment of the actions based on rules of the fuzzy logic. The Proprioceptive algorithms also are created using base of rules fuzzy.

The Proprioceptive algorithms have as input the signals of the controllers and the values measured for sensors. The all signals of components are also evaluated, as the engines, sensors, actuators and values of its physical and mechanical measures. The development of these algorithms can allow that the robots have one better perception of its proper metallic body, knowing the dimensions of the arms, height, width among others information that before were processed for the system of execution of tasks. The tasks are the programs executed normally for the robots in controlled environment, and that they are many times successfully executed. But if some unexpected situations happen, an inherent characteristic of dynamic environments, the Proprioceptive system can act and through adjustments to provide that task is executed. The robots have difficulties to decide simple situations, as the simple fact to suffer a push, where the same ones tumble and fall. When this happens need to identify this new situation and to use another algorithm to be arisen. The proposal of the Proprioceptive algorithms is that when perceiving that they are being pushed, the Proprioceptive system detects this uncommon situation and acts on controllers to prevent the fall. Preventing this unexpected situation.

II. 2-Controllers

The control systems are in practically all the modern equipment since simple bikes until sophisticated equipment like automobiles. In this brief summary will be presented enters some used techniques of control, initiating with classic control and techniques of intelligent control.

A control system can be defined as [18] and [19]:

"A Control System is a device, or a collection of devices that manage the behavior of other devices. Some devices are not controllable. A control system is an interconnection of components connected or related in such a manner as to command, direct, or regulate itself or another system."[10]

A. Mathematical Modeling

For SISO systems we have the following partial list of typical classical performance specifications. Linear models such as the one in Equation (1.1) have been used extensively in the past and the control theory for linear systems is quite mature [10],[13] and [16].

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du \qquad (1.1)$$

In this case u is the m-dimensional input; x is the n-dimensional state ($\dot{x}$ =dx(t)/dt); y is the p dimensional output; and A, B, C, and D are matrices of appropriate dimension. Such models, or transfer functions (G(s) = C(sI − A)−1B + D where s is the Laplace variable), are appropriate for use with frequency domain design techniques (e.g., Bode plots and Nyquist plots), the root-locus method, state-space methods, and so on. Sometimes it is assumed that the parameters of the linear model are constant but unknown, or can be perturbed from their nominal values.

B. Fuzzy Control System Design

Basically, the difficult task of modeling and simulating complex real-world systems for control systems development, especially when implementation issues are considered, is well documented. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity). Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system[11]. The fuzzy controller block diagram is given in Figure 1, where we show a fuzzy controller embedded in a closed-loop control system. The plant outputs are denoted by y(t), its inputs are denoted by u(t), and the reference input to the fuzzy controller is denoted by r(t), where P is the process or "*plant*".

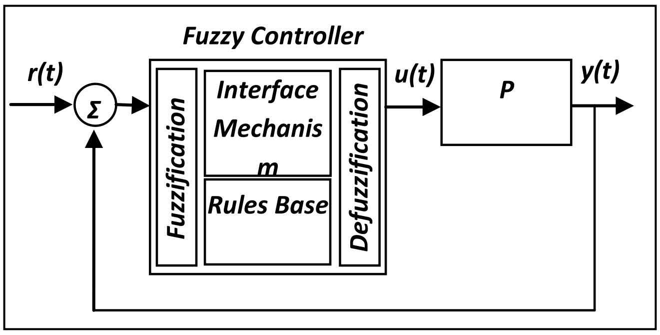


Figure 1 - Fuzzy Control System

The fuzzy controller has four main components: (1) the "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system. (2) The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be. (3) The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base. And (4) the defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant. Basically, you should view the fuzzy controller as an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data y(t), compares it to the reference input r(t), and then decides what the plant input u(t) should be to ensure that the performance objectives will be met, figure 1.

C. Fuzzy Controller for Robot Join

Consider the simple robot manipulator illustrated in Figure 2 , the equations of motion for this system are quite simple to derive, and take the form of the standard "manipulator equations" (1.2):

$$H(q)\ddot{q} + C(q, \dot{q})\,\dot{q} + G(q) = B(q)u \qquad (1.2)$$

According to Newton, the dynamics of mechanical systems are second order (F=ma). Their state is given by a vector of positions, q, and a vector of velocities $\dot{q}$ and (possibly) time. The general form for a second-order controllable dynamical system is:

$$\ddot{q} = f\,(q, \dot{q}, u, t); \qquad (1.3)$$

where u is the control vector. As we will see, the forward dynamics for many of the robots that we care about turn out to be affine in commanded torque, so let's consider a slightly constrained form:

$$\ddot{q} = f1(q, \dot{q}, t) + f2(q, \dot{q}, t)u \qquad (1.4)$$

It is well known that the inertial matrix, H(q) is (always) uniformly symmetric and positive definite, and is therefore invertible. Putting the system into the form of equation 1.4 yields:

$$\ddot{q} = H\text{-}1(q)\,[C(q, \dot{q})\,\dot{q} + G(q)] + H\text{-}1(q)B(q)u \qquad (1.5)$$

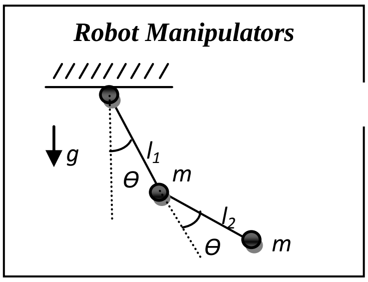
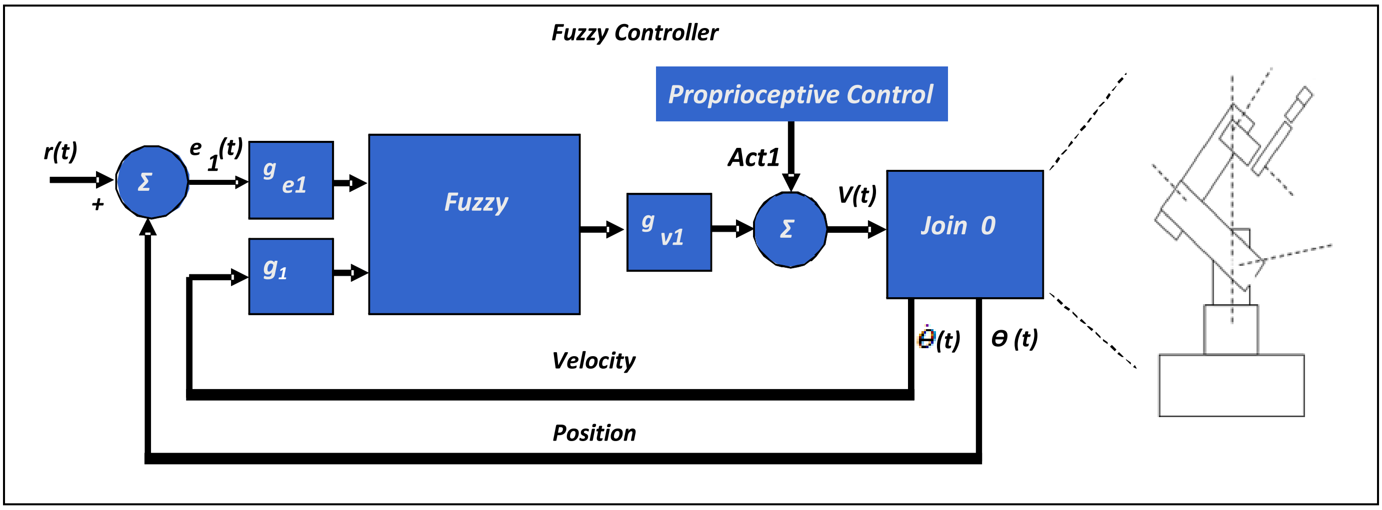


Figure 3 – Single controller

Figure 2 - Simple double pendulum

Because $H^{-1}(q)$ is always full rank, we find that a system described by the manipulator equations is fully-actuated if and only if B(q) is full row rank.

For this particular example, $q = [\Theta 1, \Theta 2]^T$ and $u = [\tau_1, \tau_2]^T$, and $B(q) = I_{2x2}$. The system is fully actuated. Now imagine the somewhat bizarre case that we have a motor to provide torque at the elbow, but no motor at the shoulder. In this case, we have $u = \tau_2$, and $B(q) = [0, 1]^T$. This system is clearly underactuated, for both cases we can consider the following fuzzy controller in figure 3. Where r(t) is the reference input, e_1 the reference error, g is the gain of velocity, position and V(t), last one is the control variable.



The use of fuzzy controllers to this procedure can be done in the following steps.

- The first step consists on execute the normal program of position the manipulator.

- In the second phase, simulate a fail in join 1 (interrupt the energy in the servo-motor 1). In this time the Proprioceptive System Fuzzy Control implement the action on the local controllers. This action was be detected by current sensor of motor and send de Proprioceptive System.

III. LAB IMPLEMENTATION

The proprioceptive algorithms are implemented with the use of Fuzzy controllers. The structure presented in the figure 3. General diagram of the robotic system and controllers is presented in figure 4.

A. Robotic System

A typical arrangement used for control the Robots Manipulator has been used to illustrate the approach proposed in this work. (Figure 5)

It consists of an robot witch 4 DOF driven by D.C. motors. A potentiometer is used to measure the position the each

Figure 5 – System for fuzzy controller

joints. The system is excited by values and the input-output data is used by Proprioceptive System of Control. A figure 5 illustrates the real system of proposal.

The figure 6 illustrates in details the pressure sensor your position and the assembler in the grip. This sensor is used to generate de fuzzy signal from de proprioceptive systems. In this case the force control is applied to move joins and capture of the objects. The pressure sensor is a little piezoelectric sensor[9].

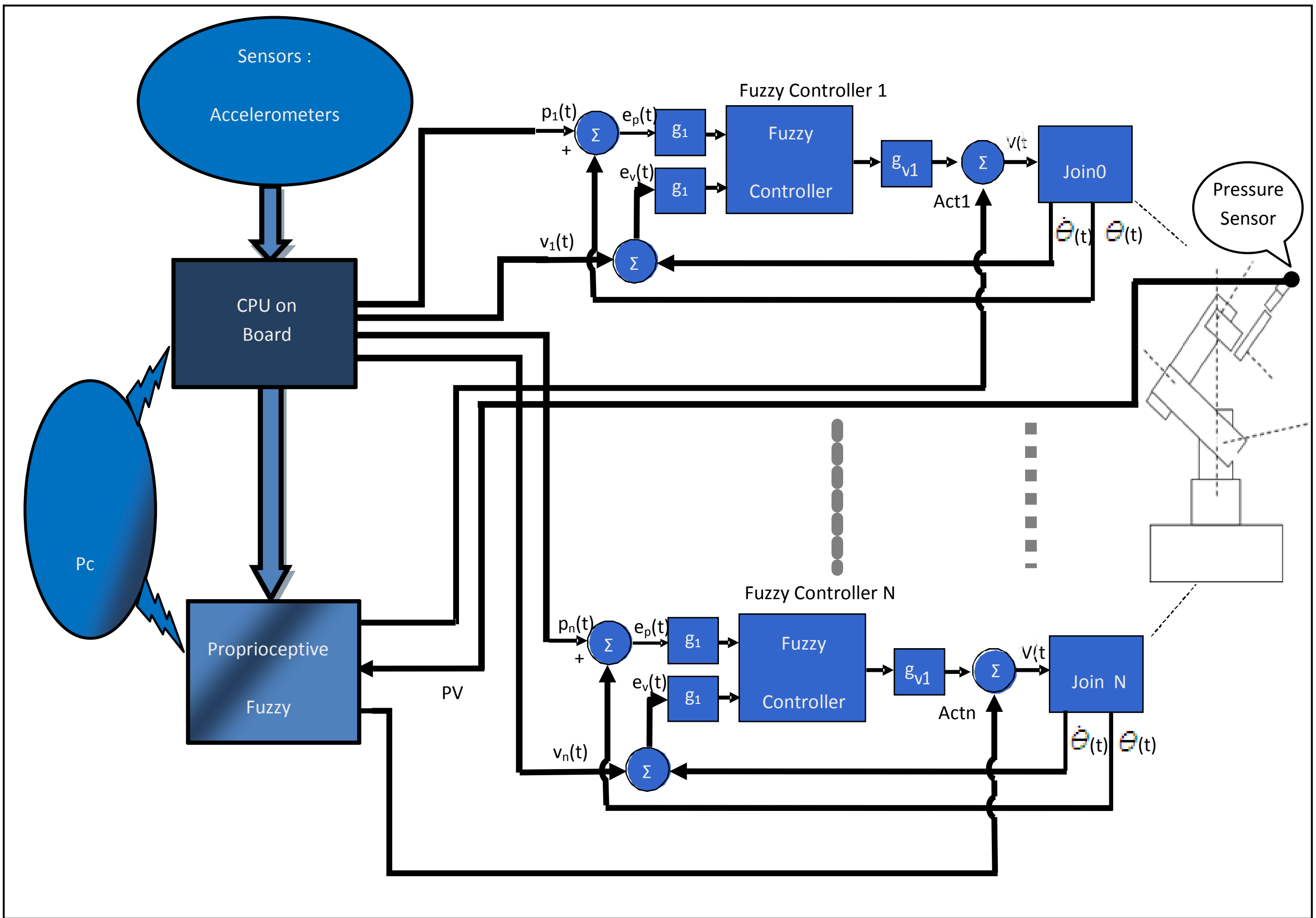


Figure 4- Fuzzy Control from Position and Velocity

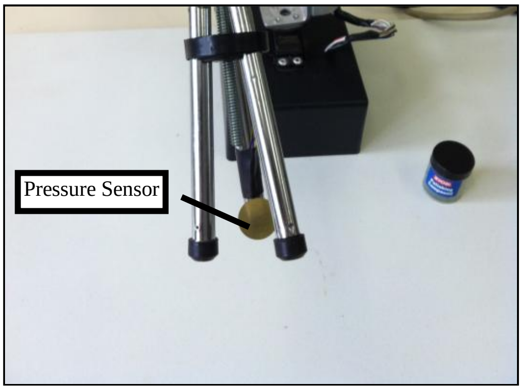


Figure 6 – Pressure Sensor detail

B. 3.3-Fuzzy Cntrol

The control Board use ARM 7 Microcontroller, equipped with protocol TCP/IP, digital outputs, PWM, converters AD/DA, protocol for memory card and pendrive, real time clock and innumerable digital communication port, figure 8.

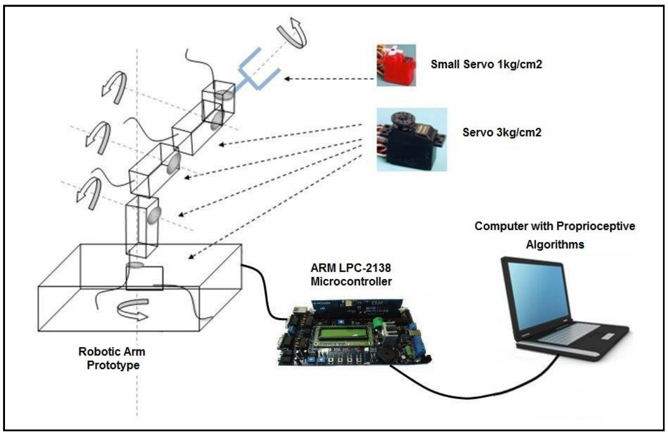


Figure 8 – Communications Diagram

Fuzzy rules for controller are printed in figure 9 and 10, where column is the error € and line is de variation of error (Δe). The inference rule is max/min like the Takagi and Sugeno fuzzy controller.

```
//NB NM NS Z PS PM PB
//0  1  2  3  4  5  6
{ 0, 0, 1, 1, 2, 2, 3 }, //NB - 0
{ 0, 1, 1, 2, 2, 3, 4 }, //NM - 1
{ 1, 1, 1, 2, 3, 4, 4 }, //NS - 2
{ 1, 2, 2, 3, 4, 4, 5 }, //Z  - 3
{ 2, 2, 3, 4, 5, 5, 5 }, //PS - 4
{ 2, 3, 4, 4, 5, 5, 6 }, //PM - 5
{ 3, 4, 4, 5, 5, 6, 6 }  //PB - 6
```

Figure 9 – Fuzzy Rules

An example for Fuzzy rules applied on pressure sensor is list below:

If Pressure is not PM then actn is Z

In table 1 is printed de fuzzy sets and cardinally of the system.

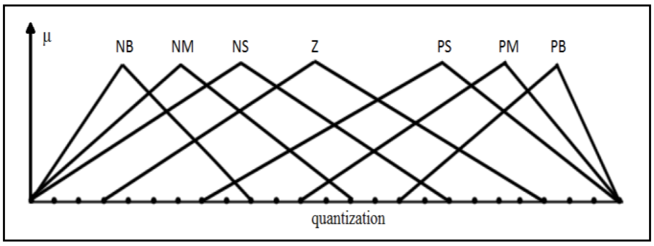


Figure 10 – Fuzzy Sets

TABLE 1- FUZZY SETS

Fuzzy Sets	Description	Numeric Definition
NB	Negative Big	0
NM	Negative Medium	1
NS	Negative Small	2
Z	Zero	3
PS	Positive Small	4
PM	Positive Medium	5
PB	Positive Big	6

C. 3.4-Computer Interface

In the figure 11 is showed the block diagram that represent the programs S1 and S2. The S1 program is the binary codes include in there arm microprocessor, development in height level language and the yagarto library form LPC2138 processor.

The S2 program two tests were made using the java language and C++ language both presented results satisfactory for different applications.

D. 3.5-Results

The next graphics present the result for trajectory from joins for fuzzy control in many situations. The figures 12, 13 and 14 present the trajectory for one basic task. In this task de robot arm use the proprioceptive algorithm for get objects of different types.

The sensor piezoelectric is used for determine the load capacity for get different objects figure 15.

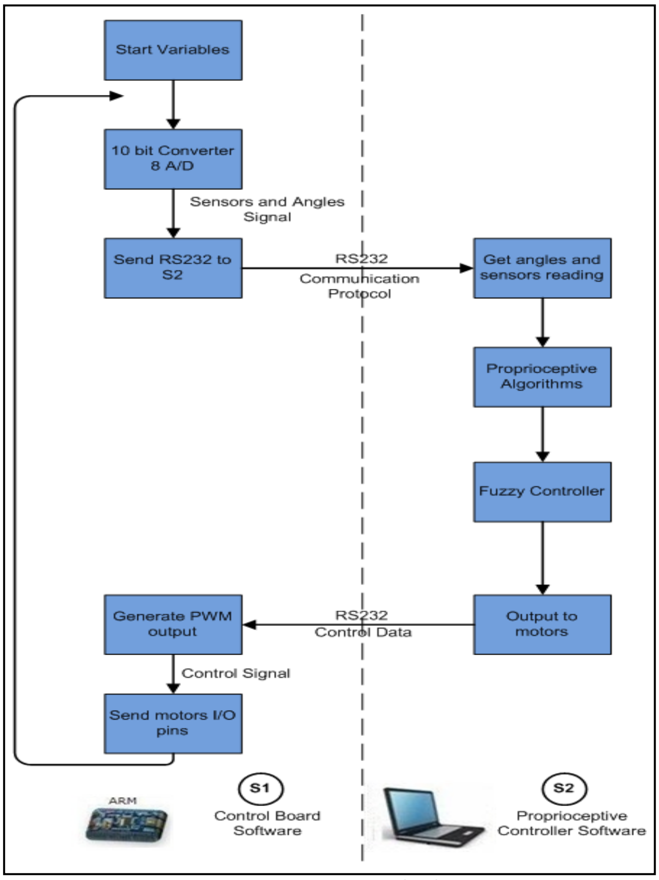


Figure 11 – Architecture of software

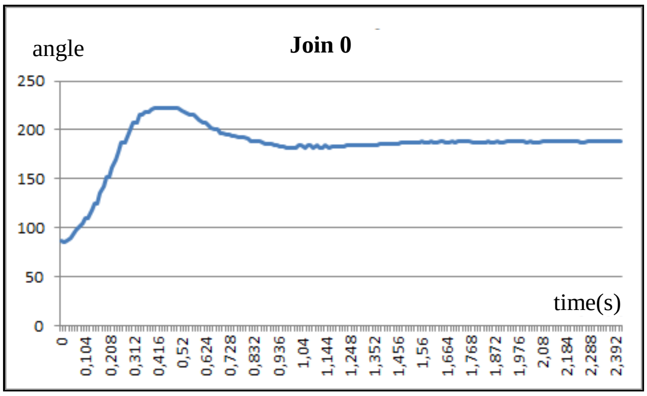


Figure 12 – Trajectory for Join 0

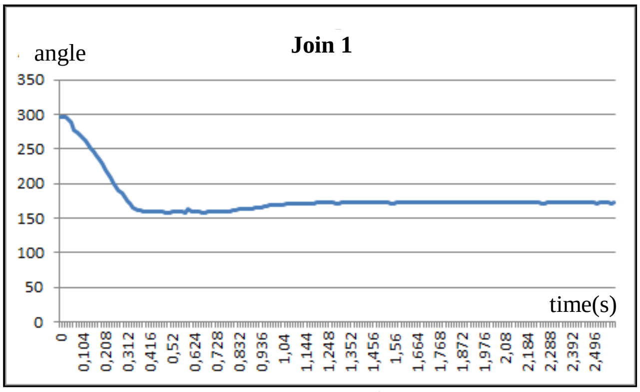


Figure 13 – Trajectory for Join 1

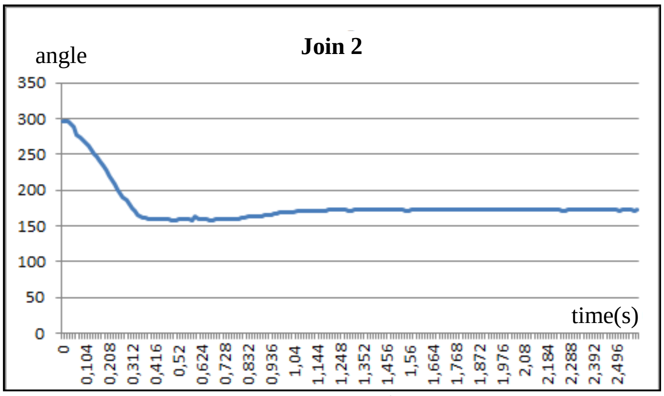


Figure 14 – Trajectory for Join 2

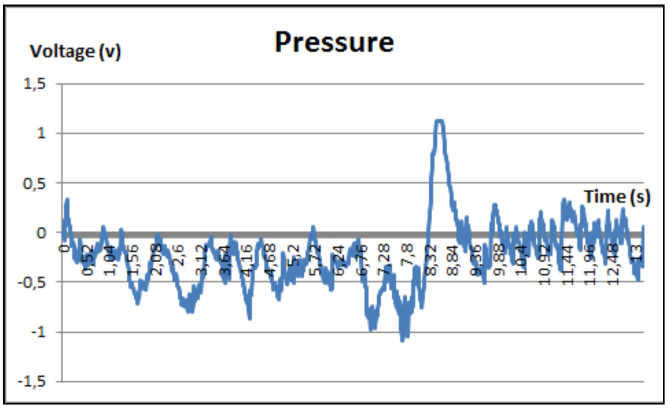


Figure 15 – Pressure Sensor

IV. CONCLUSIONS

The results obtained from the lab experiments show that robot system using de fuzzy control for get object. The fuzzy control sends to Proprioceptive software and the robot system dealing the best trajectory for the robot.

All circuits and hardware developed for this project have been showed efficient. The result obtained for enhancement using fuzzy logic techniques, show us effective, very speed and of the simple implementation.

In figure 16 the robot start the process of movement, in this moment the algorithm compute all nodes of links and search for a best trajectory.

A new stage in the robotic system must avoid obstacles perceived by sensors attached to the arm, in order to avoid those obstacles and accomplish their tasks.

V. BIBLIOGRAPHY AND REFERENCES

[1] Okko H. Bosgra, Huibert Kwakernaak and Gjerrit Meinsma, **"Design Methods for Control Systems"**, 2009.

[2] Hunt, R., Su, R. Meyer, G. - **"Global Transfomation of Nonlinear Systems"**, IEEE Trans. Aut. Control, AC-28, Nº.1, pp 24-31, 1983.

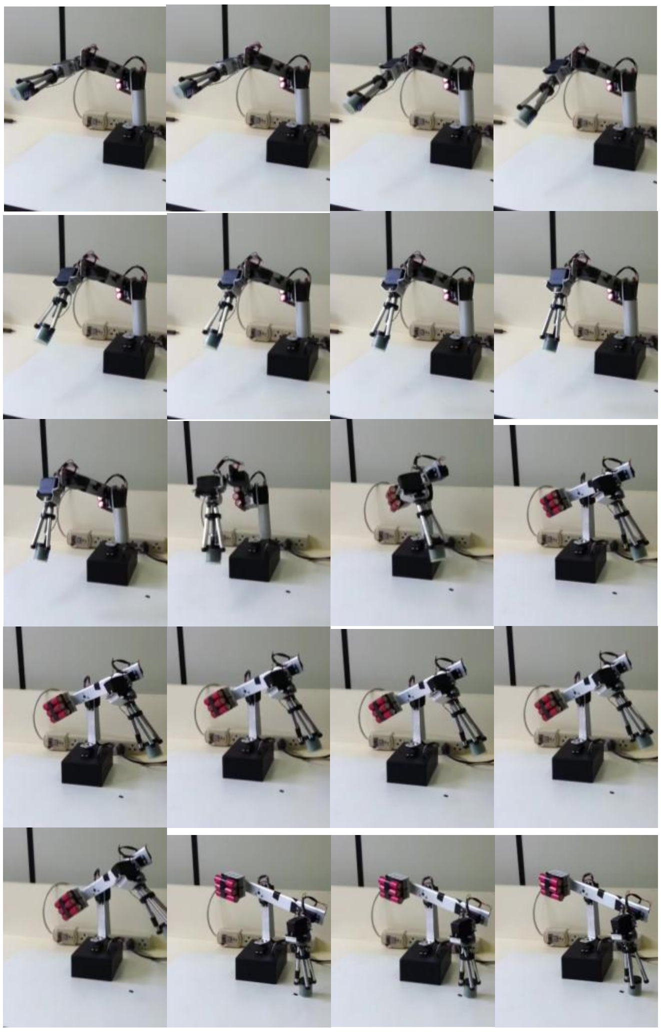


Figure 16- Movement of robot

[3] Markus Eich, Felix Grimminger, Frank Kirchner DFKI Bremen **"Proprioceptive Control of a Hybrid Legged-Wheeled Robot (German Research Center For Artificial Intelligence) "**, 28359 Bremen, Germany

[4] Dmitry Kaynov, **"TESIS DOCTORAL Open motion control architecture for humanoid robots"** Leganés, octubre 2008

[5] http://en.wikipedia.org/wiki/Proprioception

[6] Jakubczyk, B. **"Feedback Linearization of Discrete Time Systems"** , Systems And Control Letters, Nº. 9, pp 411-416, 1987a.

[7] G. Benet∗ , F. Blanes, J.E. Simó, P. Pérez, **"Using infrared sensors for distance measurement in mobile robots"** Robotics and Autonomous Systems 40 (2002) 255–266.

[8] Meyer, G., Cicolani, L. - **"Application for Nonlinear System Inverses to Automatic Flight Control Design-System Concepts and Flight Evaluation"**, Theory and Apllications of optiomal Control in Aerospace systems, P Kent. Ed. Nato, 1980.

[9] Matej Andrejasi, **"MEMS ACCELEROMETERS"**, University of Ljubljana - Faculty for mathematics and physics - Department of physics (seminar)

[10] Andrew Whitworth ,**"Control Systems And Control Systems Engineering With Classical and Modern Techniques And Advanced Concepts"**, http://upload.wikimedia.org/wikipedia/commons/e/e4/Control_Systems.pdf

[10] Elmo B. Faria, Ghilson R. Corrêa & Edilberto P. Teixeira, " **Implementation of Neural Networks controller form Robot Manipulates ",** Proceedings of the IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais pp. 350-355, July 20-22, 1999 - ITA, São José dos Campos - SP – Brazil.

[11] K.S. Narendra, and K. Parthasarathy, **"Identification and Control of Dynamical Systems Using Neural Networks,"** *IEEE Transactions on Neural Networks*, vol. 1, pp. 4-27, 1990.

[12] K.S. Narendra and B. Balakrishnan, **"Improving Transient Response of Adaptive Control Systems Using Multiple Models and Switching,"** *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1861-1866, 1994.

[13] R.S. Sutton, and A.G. Barto**, "*Introduction to Reinforcement Learning",*** Cambridge, Mass.: MIT Press, 1998.

[14] E.P. Nahas, M.A. Henso and D.E. Seborg, **"Nonlinear Internal Model Control Strategy for Neural Models"**, *Computers and Chemical Engineering*, vol. 16, pp. 1039-1057, 1992.

[15] L.G. Kraft and D.P. Campagna, **"A Comparison between CMAC Neural Network Control and Two Traditional Control Systems,"** *IEEE Control Systems Magazine*, vol. 10, no. 2, pp. 36-43, 1990.

[16] W.T. Miller, **"Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm,"** *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 157-165, 1987.

[17] E. B. Faria, **"How Linear Is A System Feedback Linearized By A Neural Networks. International Conference On Neural Networks**", ICNN -IEEE\1997, 1997.

[18] Michael B. Histand & david G. Alciatore, **"Introduction to Mechatronics and Measurement Systems"**, McGraw-Hil,1998

[19] Sastry, S. & Isidori, A.- **"Adaptative Control of Linearizable Systems"**, IEEE Trans. On Autom. Control, Vol.34, Nº. 11, pp.1123-1131, November 1989.