International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 6, No. 3 (May-2017)

http://dx.doi.org/10.12785/ijcds/060301

NeuroFuzzy Controller Design and FPGA-Based Embedded System for ACROBOT Model

Yesid E.CastroCaicedo[1,2], Walter de Britto Vidal Filho[2], Edward David Moreno[3] Leandro dos Santos Coelho[4] and Carlos H. Llanos[2]

[1] *Centro de PesquisasAvançadasWherner Von Braun, Brasil,* [2] *Universidade de Brasília, Brasil*
[3]*Departamento de Computação da Universidade Federal de Sergipe (DCOMP/UFS), Brasil*
[4]*Industrial and Systems Engineering Graduate Program, Pontifical Catholic University of Parana, Brazil*

Received 7 Mar. 2017, Revised 1 Apr. 2017, Accepted 20 Apr. 2017, Published 1 May 2017

Abstract: This work presents the modeling and control system design for a robot that rides a bicycle using the well-known Acrobot model for slow speeds, which was finally implemented in an FPGA-based embedded system. In this work the implementation of the controller was achieved for the Acrobot option following two ways. The first one implementation was developed based on modern control theories, involving:(a) the states feedback controller issues based on the appropriated poles allocation, guarantying stability and (b) the designs of a LQR (Linear Quadratic Regulator) type controller that minimizes the criterion of quadratic cost. The second one implementation was in turn achieved by means of some intelligent control methods, involving: (a) artificial neural networks, (c) tuning a neurofuzzy system, which joints the capacity of learning of the neural networks with the power of linguistic interpretation of the fuzzy inference systems (neurofuzzy system ANFIS, **A**daptive **N**euro**F**uzzy **S**ystems) and (c) fuzzy logic. For testing the designed controller, a simulator has been also developed and connected to the controller embedded in FPGAs.

Keywords: Acrobot, FPGAs, Embedded Systems, Intelligent Control Methods, NeuroFuzzy Systems

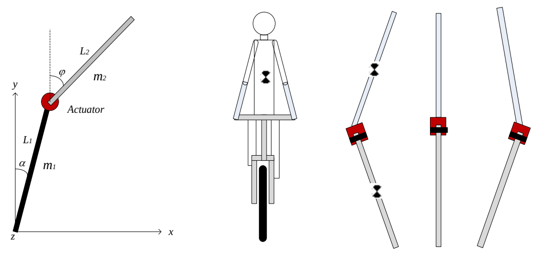
1. Introduction

The modeling and controller design for a robot that rides a bicycle using the well-known Acrobot model can be addressed using either two approaches: (a) a dedicated configuration where the robot is the bicycle itself and (b) the robot having human characteristics. In the first one, the stability of the robot can be achieved by means of the movement of the handlebar of the bicycle, with limitation of speeds. Therefore, in low speeds the centrifugal force generated by the circular or elliptical movement as a response of the handlebar movement is not enough to keep the robot balanced [1]. On the other hand, the second option allows the robot to be controlled for low speed, even for null speed. In this case, the modeling of the robot cyclist has similarity to the problem of the under actuated inverted double pendulum (Acrobot)[2,3,4] (see Fig. 1). In this case, a definitive or absolute model has not been found for this complex problem so far. The controller designs for this system have been developed, using several techniques and the stability of the robot can be achieved in terms of the movement of the handlebar of the bicycle, with a limitation due to the fact that for critical speeds is not possible to keep the bicycle posture [1,5].

The study of systems based on Acrobot and inverted pendulum are still current research [6,7,8], although the Acrobot problem have been proposed since the 90s [2]. Otherwise, FPGA applications on prototypes such as Acrobot [6] and inverted penduluns [7] have shown an increased interest.

In the context of the controller design development, two elements are necessary: (a) a mathematical model for simulating the dynamic behavior system in the computer, which has been developed and simulated in Matlab software, and (b) the implementation of a efficient fuzzy controller in hardware (using a FPGA - Field Programmable Gate Array), which allows a dynamic interaction with the plant as a first approach for the real prototyped system.

E-mail address: yesid.caicedo@vonbraunlabs.com.br, yesid@unb.br, wbritto@unb.br, edwdavid@gmail.com,leandro.coelho@pucpr.br, llanos@unb.br



(a) (b) (c)

Figure 1.Approximation model of Robot Cyclist.

In this context, the logic fuzzy systems are widely used, being capable to deal with the imprecision of the models and additionally offering faster and straightforward solutions with respect to the conventional control approaches [9,10]. For instance, natural language uncertainty and the approximated reasoning mechanism of the human brain can be modeled through the use of fuzzy logic. The knowledge base of a fuzzy system is described by a set of rules (based on membership functions and several logic and arithmetic operations) such as in an expert system. However, the rule computation is performed numerically in a digital system. The symbolic description and analytic processing aspects mean that fuzzy logic is useful for modeling those systems in which it is difficult to achieve a mathematical behavior.

In the fuzzy algorithms implementation issues there are several proposals for software implementation (using standard processors) and hardware ones mainly through of reconfigurable hardware (for instance, based on FPGAs).In the software implementation of a fuzzy algorithm there are several limitations related to the von Neumann model, mainly due to the sequential execution of the instructions in the processors and the memory bottleneck, which impose serious speed limitations in the system [11]. These limitations address the controller system design to the use of hardware/software co-design techniques, taking advantage of the inherent parallelism in the fuzzy inference algorithms and executing multi-operand operations. These considerations justify the necessity of the use of hardware structures in order to accelerate the fuzzy algorithm executions [12,13].

For application purposes, the FLCs (Fuzzy Logic Controllers) can be divided into two classes: (a) general-purpose fuzzy processor with specialized fuzzy computations and (b) dedicated fuzzy hardware for specific applications. The general-purpose fuzzy processor has been implemented on various platforms, such as computers (PCs), microprocessors, microcontrollers, digital signal processor (DSP) and transputers (pioneer processor in parallel processing).

The relatively simple architecture and the processing algorithm of the FLC naturally lead to straightforward implementations in dedicated hardware. The dedicated fuzzy hardware has been implemented by different technologies, digital and analog ones [10,14].

The design methodology used in this paper is based on the interconnection of the Microblaze microprocessor (from Xilinx) and a fuzzy system developed in hardware, using the Xfuzzy tool [15,16,17,18]. The proposed design methodology involves firstly the fuzzy controller design in software for simulation and testing issues, taking into account the model of the plant. The controller was synthesized to the hardware description language VHDL, using the fuzzy environment Xfuzzy tool [19,20]. For improving the performance of the fuzzy controller and its suitability for implementing in the FPGA, the same was divided into 2 modules, each one producing a torque control variable. The first module receives the position and angular speed of the first link (L_1, see Fig. 1) of the Acrobot system. The second module receives the position and angular speed of the second link (L_2, see Fig. 1). The final torque variable is calculated in the Microblaze, subtracting the two outputs, over which respective gains are previously applied. In this case, each gain represents a priority that depends of the current state variables. These gains were experimentally calculated through several simulations executed in Matlab.

Additionally, the fuzzy controller partitioning was also developed due to the fact that Xfuzzy 2.0 does not allow the designer to synthesize VHDL modules with more than one output. Additionally, dividing the fuzzy controller into two subsystems the number of knowledge rules base were reduced from 89 to 18 rules.

2. MATHEMATICAL MODEL OF THE ACROBOT

The mathematical model of the system allows the use of numerical methods (method RungeKutta of 4^{ta} order, method of Euler) for the solution of the system's dynamic equations in order to develop the simulator of the plant in the computer. Moreover, the joined equations become linear around of equilibrium point that allows finding the required matrices for the development of the suitable controllers. To accomplish this, the method of Energy of Lagrange was used [21]. The equations that govern the system are shown in Eq. (1), the damping air was not considered because of the relatively narrow condition of the bodies used.

$$\begin{bmatrix} \left(\frac{m_1}{3}+m_2\right){L_1}^2 & \frac{m_2}{2}L_1L_2\cos(\varphi-\alpha) \\ \frac{m_2}{2}L_1L_2\cos(\varphi-\alpha) & \frac{m_2}{3}L_2^2 \end{bmatrix}\begin{bmatrix}\ddot{\alpha}\\ \ddot{\varphi}\end{bmatrix}+$$

$$\begin{bmatrix} 0 & -\frac{m_2}{2}L_1L_2\sin(\varphi-\alpha)\dot{\varphi} \\ \frac{m_2}{2}L_1L_2\sin(\varphi-\alpha)\dot{\alpha} & 0 \end{bmatrix}\begin{bmatrix}\dot{\alpha}\\ \dot{\varphi}\end{bmatrix}+$$

$$\begin{bmatrix} (\frac{m_1}{2}+m_2)gL_1\sin(\alpha) \\ \frac{m_2}{2}gL_2\sin(\varphi) \end{bmatrix}=\begin{bmatrix}0\\ \tau\end{bmatrix} \qquad (1)$$

It is easy to conclude through the Eq. (1) that the system is nonlinear. Therefore, it is possible to get information about a non-linear system by approximating the system by a linear one, close to the equilibrium point (the vertical invert position), using the state variables shows in Eq. (2). The representation of system in the state space is usually used in multivariable or complex systems. The main reason for choosing this type of representation in this work is because of the presence of four output variables.

$$\begin{Bmatrix} x_1 = \alpha \\ x_2 = \dot{\alpha} \\ x_3 = \varphi \\ x_4 = \dot{\varphi} \end{Bmatrix} \qquad (2)$$

Now is possible to rewrite the system equations given in Eq. (1) like a 1st order differential equations system according to Eq. (3).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-d_4 d_5}{d_5 d_1 - d_2^2} & 0 & \frac{d_2 d_6}{d_5 d_1 - d_2^2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-d_4 d_2}{d_2^2 - d_5 d_1} & 0 & \frac{d_1 d_6}{d_2^2 - d_5 d_1} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -d_2 \\ 0 \\ -d_1 \end{bmatrix} \tau \qquad (3)$$

Where

$d_1 = \left(\frac{m_1}{3} + m_2\right) L_1^2$, $d_2 = \frac{m_2}{2} L_1 L_2$, $d_4 = \left(\frac{m_1}{2} + m_2\right) g L_1$, $d_5 = \frac{m_2}{3} L_2^2$, $d_6 = m_2 g \frac{L_2}{2}$.

3. THEORETICAL FUNDAMENTS AND THE IMPLEMENTED CONTROLLERS

A. Modern Control based Controllers

The classical control theory is based on the input-output relations or transfer functions. A previous control model was developed by the authors in [22]. On the other hand, the theory of modern control expresses the equations of any system in terms of 1st order differential equations that can also be represented as a 1st order differential equation matrix [23], as can be observed in Eq. (4). The concept of states space is related to the concept of state variable. The state of the system is a set of variables such that the knowledge of these variables, the input functions and the dynamic equations supply to the future state for the system [21]. For the development of the controllers a system (whichis controllable and time invariant) has been supposed such as shown in Eq. (1), which was obtained around the equilibrium point, namely $\{x_1, x_2, x_3, x_4\}=\{0, 0, 0, 0\}$. The states equation matrix is depicted in Eq. (4), where the matrices A and B are found by means of a comparison between Eq. (4) and Eq. (3).

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \quad (4) \qquad y = C\boldsymbol{x} + D$$

where the matrixes C e D are shown as following:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},\ D = 0.$$

The control law is achieved in the Eq. (5), considering that the reference is equal to zero, which indicates that the inverted double pendulum is in the vertical line.

$$\boldsymbol{u} = -K\boldsymbol{x} \qquad (5)$$

The vector K represents the controller gains, and substituting Eq. (5) in Eq. (4) the close loop system is obtained in Eq. (6).

$$\dot{\boldsymbol{x}} = (A - BK)\boldsymbol{x} \qquad (6)$$

1) Linear Quadratic Regulator (LQR)

Optimal control theory (specifically the Linear Quadratic Regulator) uses a performance index in which is possible to optimize physical greatnesses. The goal of optimal control is defined by gain matrix K of the optimal control vector obtained in Eq. (5), in order to minimize the index of performance as shown in Eq. (7).

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (7)$$

Where Q and R are real Hermitian matrices, which are symmetrical positive defined. The relative importance of error and of the energy consumption is determinate by these matrices. If the elements of the matrix K were previously determinate in order to minimize the performance index shown in Eq. (7), then the control law shown in Eq. (5) is optimal for any initial state $x(0)$.

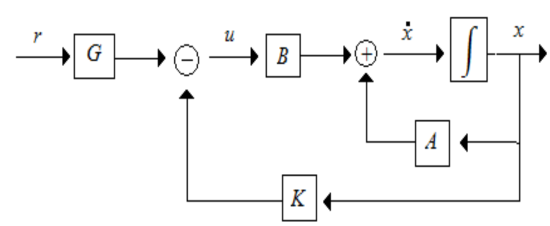


Figure 2.The block diagram of the state feedback controller.

2) The State Feedback Controller (SFC)

In this work the use of a controller with poles allocation is achieved by means of the state feedback, in order to perform the regulation of desired outputs for the Acrobot systems (namely are the position and angular speed).Moreover, the controller design is based on the linear system, and the same is applied over the original nonlinear equations shown in Eq. (1).

In this case the feedback system showed in Eq. (6) can be represented by means of the block diagram in Fig. 2.

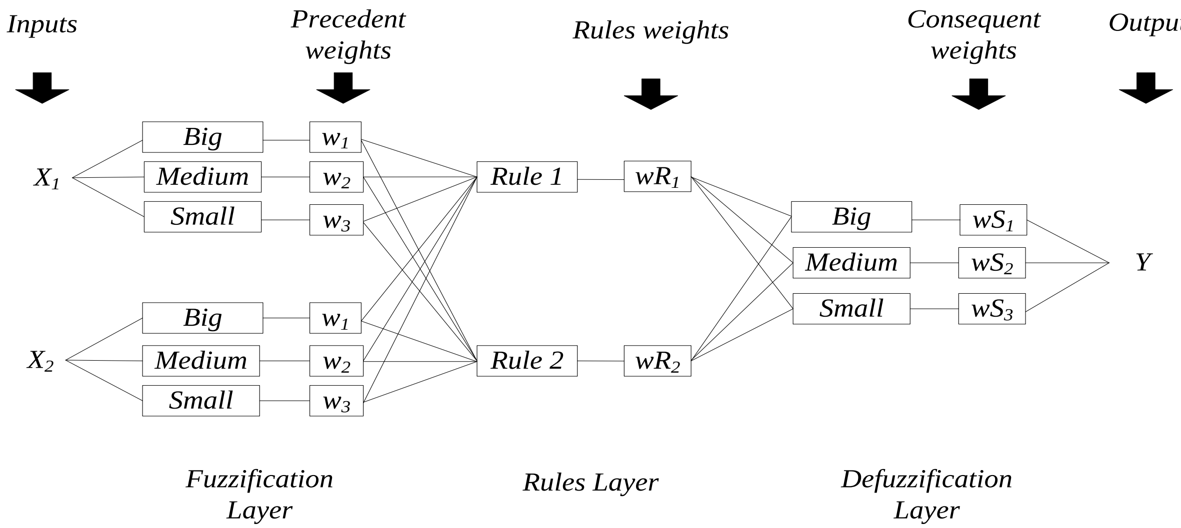


Figure 3.ANFIS Architecture

B. The Intelligent Control based on Neurofuzzy

The project of intelligent controllers has been widely used such as an alternative to either complex mathematical models, or those where anyone model is available. For developing a controller with the system characteristics a qualitative description of the performance system (linguistic variable) can be considered jointly with a quantitative description based on experimental data collections (data sets).

The neurofuzzy-ANFIS proposal consists of a set of learning of fuzzy rules and the member's functions adjust, using neural networks (backpropagation algorithm and the squared minimums method). A wide variety of algorithms have been proposed to this problem (Fsom, Nefclass, Fuzzytech, etc.), and the work developed in [24] was the seminal work. The traditional neurofuzzy models have a limited capacity of adjustment in their structure, and due to the rule number explosion problem these models are generally used in the applications with restrict input number [25]. Moreover, the major part of these neurofuzzy systems are appropriate for supervised training.

The data set used for member functions training of neural-fuzzy systems is achieved from the developed state feedback controller, due to absence of the real system. The intelligent control learns the behavior of the system under action of state feedback controller. Fig. 3 shows the architecture of the ANFIS system for the fuzzy system learning process, which was generated by the toolbox of Matlab.

To accomplish the controller three member functions have been chosen for each input of the neurofuzzy system (see Fig. 3). All the member functions are Gaussian type functions and the fuzzy controller is a Takagi-Sugeno type [26]. The system ANFIS generated 81 singletons outputs, and for the network training an hybrid algorithm (the back-propagation plus squared minimums methods). In the training process a set of 8000 data and 100 epochs were spent.

C. The Intelligent Control based on Fuzzy logic

1) The fuzzy logic aspects

The FLC, typically, consists of four principal units, namely: (1) the fuzzifier which converts a crisp input into a fuzzy term set, (2) the fuzzy rule base which stores fuzzy rules describing how the fuzzy system performs, (3) the fuzzy inference engine that performs approximate reasoning by associating input variables with fuzzy rules and (4) the defuzzifier which converts the FLC's fuzzy output to a crisp value for the actual system input over the target. The control performance is more or less influenced by the selection of fuzzy sets of linguistic variables, the shapes of membership functions, the fuzzy rule base, the inference mechanism, and the defuzzification method [14, 27, 28].

In this case the Xfuzzy tool has been used[15,16], in which the circuit generator of the membership functions (MFCs) generates the fuzzy sets in antecedent rules.

For each input values the MFC provides the levels, the activation values and the overlap degree of the system. For each cycle counter the membership degrees are combined by means of MIN or product operators for calculating the membership degree the each rule, whereas the level antecedents addresses the rule memory positions that contain the respective consequent for rule. Finally, the stage of defuzzyfication calculates the output system as the media consequents of pondered rule by the activation degrees [19, 27]. The MFCs can be implemented by means of vectorial or arithmetic techniques [32]. In this work the vectorial approach was used because this allows the definition of membership functions arbitraries but it can to limit the circuit implementation when the number of inputs is incremented or her precision [19].

2) An approach for a fuzzy logic design based on partitioning

In this approach the fuzzy controller (see Fig 4.a) was divided into 2 subsystems given that complex systems contain knowledge in which the fuzzy rules generation is exponentially increased by the input number (see Fig 4.b). For control multivariable problems (Multiple Inputs – Multiple Outputs, Multiple Inputs – Simple Output) is recommended to split the system for decreasing the complexity [29].

The first controller depends of the input variable for the link L_1 (position and angular speed) and the second controller depend of corresponding input variables (for the link L_2). The output values of each controller are multiplied by the gain values, which have been found experimentally. Their results are subtracted for producing the final control torque variable as shown in the Fig. 4. Initially, the controllers were developed using the Matlab fuzzy tool and tested on a plant simulator environment, which has been developed in Matlab.

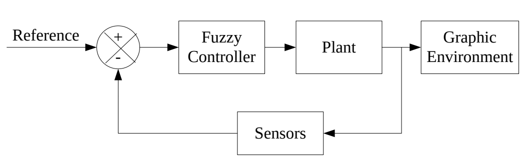


Figure 4a.Arquitecture Fuzzy Controller

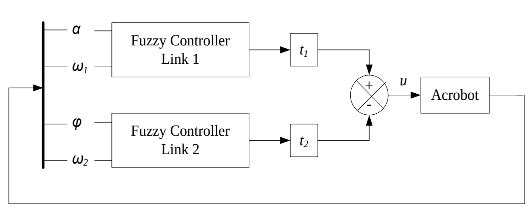


Figure 4b.FuzzyController

The controller design is a Takagi Sugeno type, which comprises three membership functions (see Fig. 5) for each input variable. Triangular functions have been used, namely: *N* (*negative*), *M* (*medium*) and *P* (*positive*). This means that the knowledge base for each controller is defined by 9 fuzzy rules. The first controller rules are depicted in Fig. 6.

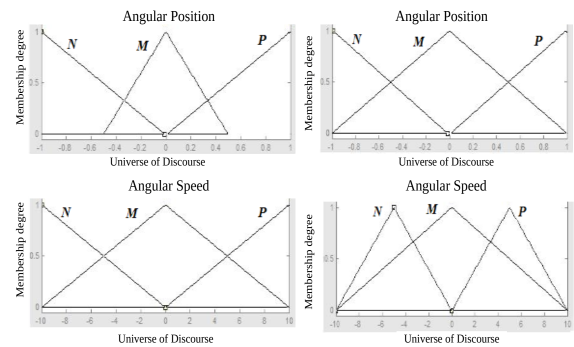


Figure 5. Membership Functions for Fuzzy Controllers

1. If (α is N) and (w is N) then (torque is ten)
2. If (α is N) and (w is M) then (torque is eight)
3. If (α is N) and (w is P) then (torque is six)
4. If (α is M) and (w is N) then (torque is four)
5. If (α is M) and (w is M) then (torque is zero)
6. If (α is M) and (w is P) then (torque is nfour)
7. If (α is P) and (w is N) then (torque is nsix)
8. If (α is P) and (w is M) then (torque is neight)
9. If (α is P) and (w is P) then (torque is nten)

Figure 6. Fuzzy Rules for link1 fuzzy controller

4. RESULTS

A. Results of the LQR Controller

For the design of the LQR (Linear Quadratic Regulator) controller is necessary to define both the Q and R matrices, which were experimentally chosen by observing to system response. The parameters used for numerical simulation and controllers test are $L_1=L_2=0.4$ Kg, $M_1=M_2=1$ Kg, $g=9.8$ m/s^2 and the inertia moments are considered in the middle of links as $I_1=I_2=(1/12)M_{1,2}*L_{1,2}^2$.

The R matrix represents the energy consumption of the control signals, yielding a reduction/enlargement of the requirements of input torque in the cost criterion [39]. On the other hand, the Q matrix is related both to the overshoot and stabilization time of the system. In this project different values were tried for the matrices and the better response was finding using the values: R=7000 and Q = [1000 0 0 0; 0 0 0 0; 0 0 1000; 0 0 0 0].

The results are shown in Fig. 7 and Fig. 8 where Fig. 7 shows the LQR controller response with initial conditions $\{x_1, x_2, x_3, x_4\}=\{15, 0, -15, 0\}$. It can be observed that the stabilization time of system is less than three seconds. Additionally, the required variations of torque are also shown, and it can be observed that for these conditions is not overtook a torque of 1.5 Nm.

Otherwise, Fig. 8 shows the system response when applied a disturbance of 1.5 Nm in a impulse form of 0.05 sec. Additionally, in the same figure is shown how the system is stabilized again in less than one second.

The disturbance is applied in the 5 second time and it is shown the torque variations. Finally, Fig. 7 depicts that the controller performance is satisfactory.

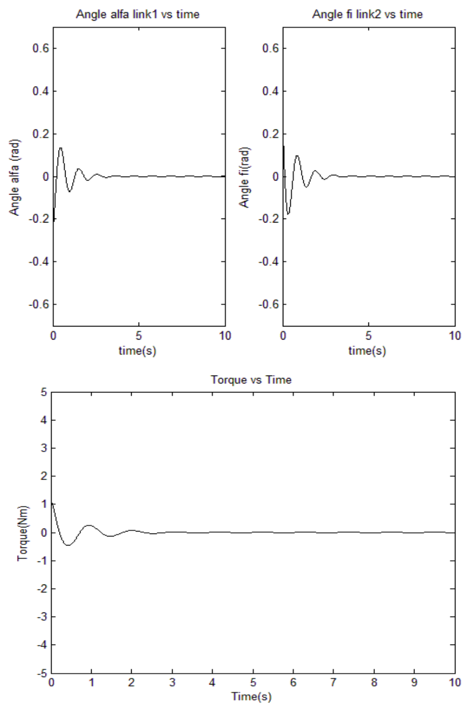


Figure 7.LQR without disturbance

B. Results of the State Feedback Controller

For the state feedback method is assumed that all state variables are measurable and available for feedback. If the system is completely state controllable, then the closed-loop poles of system can be placed in any one desired position by means of the state feedback, using appropriate matrix gains.

To find the matrix gains (that leads the system for the desired positions) the root locus technique is used in order to observe dominants poles of the system. In this case four states variables were used for the system representation, and the gains vector is composed of the four elements corresponding to the gain for each state variable.

The idea is to choose a pair of dominant poles for the system, and the others ones are placed to the left of these.

The allocation of the poles positions can be chosen for different values, but some positions have the best behavior.

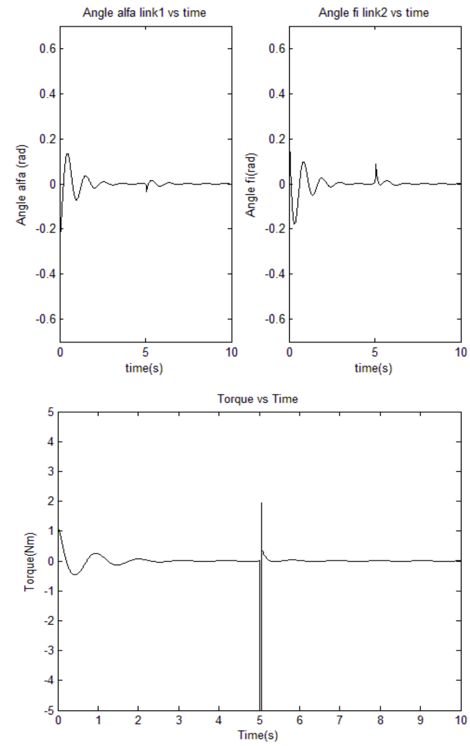


Figure 8.LQR with disturbance

The chosen poles are {-0.2 $\pm$ 11.5274i, -2, -8} and with these poles jointly with the *A* and *B* state matrices different methods (namely, Ackerman, Bass-Gura, Mayne-Murdoch) can be used to find the matrix gain. The Ackerman method was chosen and a Matlab function receives as input arguments the state matrices, and the chosen poles, returning the values from the feedback gains (satisfying the block diagram shown in Fig. 2).

Fig. 9 and Fig 10 show the system response using the control law showed in Eq. (5) and the matrix gains *K* ={67.2601, -61.8928, -34.0390, -23.6652}. The initial conditions are $\{x_1, x_2, x_3, x_4\} = \{15, 0, -15, 0\}$, and Fig. 9 shows the stabilization behavior. In this case the time is less than 2.5 sec, and it does not present overshoot if compared with the LQR controller (see Fig. 7 and Fig. 8).

The Fig. 10.shows the system response when applied a 1.5 Nm disturbance in form of impulse with a duration of 0.05 sec. The system goes quickly for the invert position and this in turn leads the pendulum for the inverted position again, preventing the system from knocks down. It can be observed that the control effort of state feedback approach is larger than the LQR controller. In contrast, the state feedback system converges more quickly.

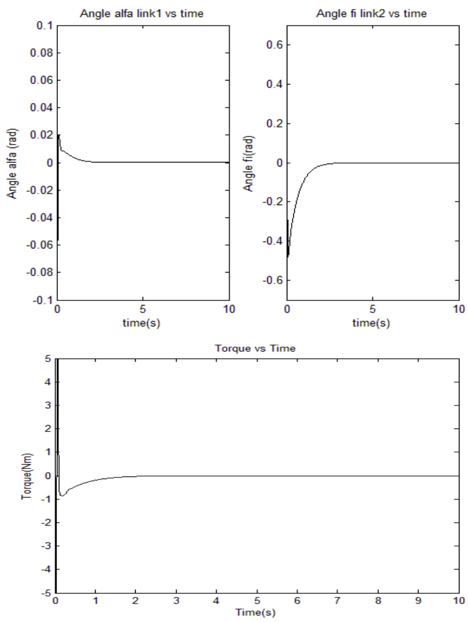


Figure9.State feedback without disturbance

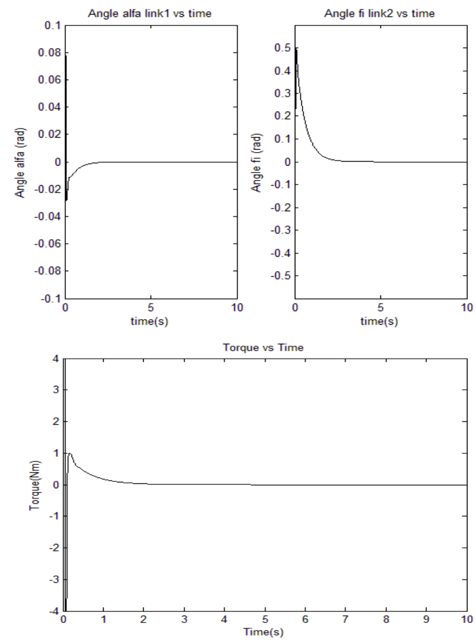


Figure11.Neurofuzzy without disturbance

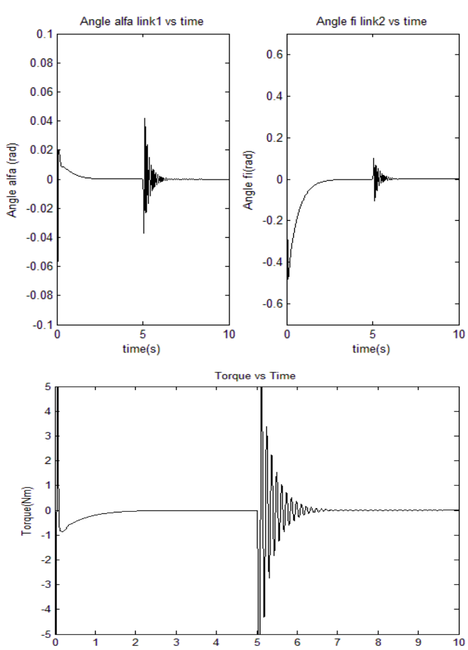


Figure 10. State feedback with disturbance

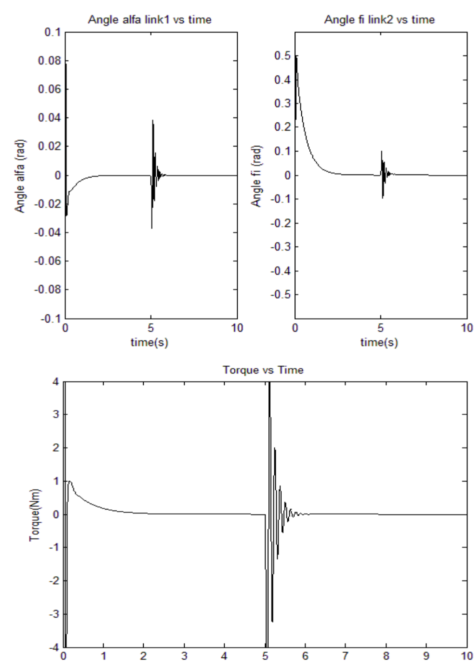


Figure 12. Neurofuzzy with disturbance

C. Results of the Neurofuzzy controller

The Fig. 11 and Fig. 12 show the system behavior with the fuzzy controller optimized by ANFIS with initial conditions, namely: $\{x_1, x_2, x_3, x_4\} = \{-15, 0, 15, 0\}$.

Fig.11 shows the system response without disturbance, and Fig. 12 shows the system response with the same disturbance used for the others controllers.

The system learned from the state feedback controller, but the system response with intelligent control is a little bit different.

Comparing Fig. 9 and Fig. 10 against Fig. 11 and 12, it can be observed that they are very similar. Otherwise, Fig. 12 shows that the intelligent controller is more robust than state feedback controller and it stabilizes the system faster than the state feedback one.

TABLE I. PERFORMANCE INDEX OF THE CONTROLLERS

Controllers	J_1		J_2	J_1		J_2
	Without Disturbance			With Disturbance		
	α	φ	u	α	φ	u
LQR	0.7055E-3	0.7925E-3	0.0012	0.7133E-3	0.8207E-3	0.5007
S-F	0.0001	0.0041	0.0745	0.0001	0.0042	0.0906
Anfis	0.0002	0.0069	0.0866	0.0002	0.0070	0.1129

Two performance indexes [30] for the developed controllers have been applied in order to have quantitative points for comparison in terms of the variations of the control signals and the behavior of the error throughout the simulation run time. This can indicate how the controller is fast for reaching the vertical position and how much effort is made by the actuator. The performance indexes are the sum of squared error and the sum of the quadratic control increment, as shown in the Eq. (8) and Eq. (9), respectively.

$$J_1 = \frac{1}{N}\sum_{k=1}^{N}[e(k)]^2 \quad (8)$$

$$J_2 = \frac{1}{N}\sum_{k=1}^{N}[u(k) - u(k-1)]^2 \quad (9)$$

Table I shows the obtained values for the performance indexes in the simulation time. Otherwise, the error values of the angles links with the J_1 index (with and without disturbance for each controller) are also analyzed, as well as the signals variation of the controllers in relation to the J_2 index. In the case of the test without disturbance, the LQR controller presents a higher value of J_1 for the case of the two angles than the state feedback and ANFIS controllers. This means that the two links have a larger movement around the vertical axis, generating higher error values, which can be conferred comparing the Fig. 7, Fig. 8 and Fig. 9 Otherwise, it can be observed that the state feedback one has J_1 index with lower values, by comparing the LQR against the state feedback controllers. This means that the system is stabilized a little faster than the LQR controller. The J_2 index indicates that the LQR controller demands less actuator effort than the state feedback and ANFIS ones.

When the systems is submitted to a disturbance, the state feedback and ANFIS controllers continues showing less values for J_1 index whether compared with the LQR controller. The values J_1 almost not modifies, which means that they are more robust than the LQR one, in relation to the disturbance, and the J_2 index indicates that the LQR controller requires an actuator effort larger.

5. RESULTS OF THE FUZZY LOGIC BASED CONTROLLER AND ITS FPGA IMPLEMENTATION

A. The developed FPGA Architecture

The fuzzy embedded controller comprises: (a) the respective control tasks, (b) the gathering of the state variable of the Acrobot and (c) the generation and transmission of the control variable to the simulation environment.

The specification and implementation of the two fuzzy systems like hardware peripherals was accomplished by using the XPS 10.1 wizard and connected to the MicroBlaze through the FSL interface, such as presented by the authors in [31]. For each peripheral two FSL buses are generated (one for transmission another for reception, see Fig. 13). The serial interface RS232 is used to send and receiving data to/from the graphical simulation environment. The speed transmission is 120 kbps in the RS232_DCE peripheral of Spartan 3E starter kit. Additionally, the *LEDs_8bit* peripheral was used for monitoring the controller states (see Fig. 13).

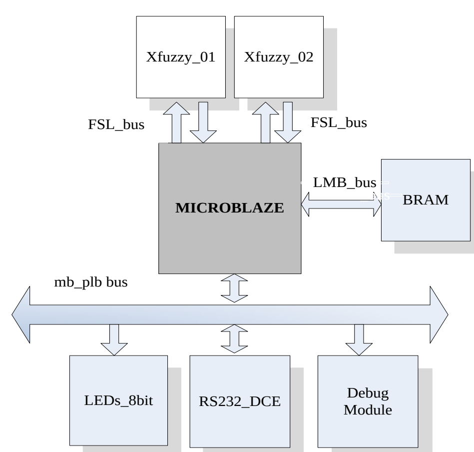
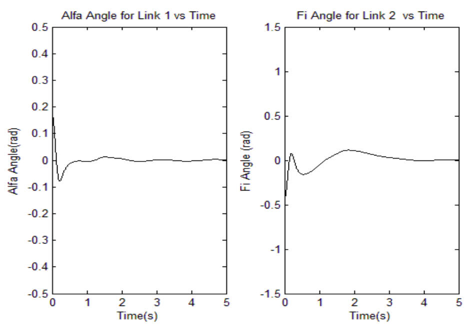


Figure 13. Embedded System and Hardware Peripherals Architecture

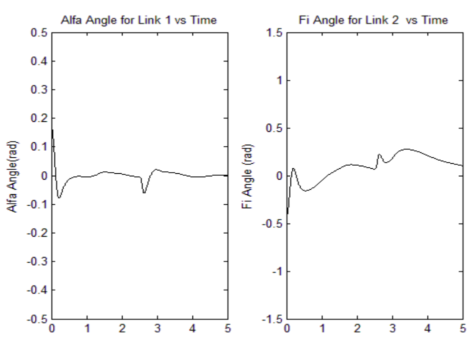
B. Implementing the Fuzzy controller in the FPGA

Several initial conditions were tested for the system. Figure 14 shows the controller response with initial conditions $\{\alpha,\omega_1,\varphi,\omega_2\}$ = {10,-25,-12,18}. The fuzzy controller was capable to hold of Acrobot in the equilibrium positions. The system was submitted to disturbance torques in form of impulse. The controller gains experimentally obtained were t_1 = 1.2 and t_2=0.3. The simulation time was configured for 5 seconds. It can be observed that the link L_1 reaches the zero grade position in a time smaller than simulation time. Note that the link L_1 reaches the zero grade position in the time smaller than one second (Fig. 14.a).

The link L_2 has a balance larger than the link L_1 around to zero grade and it guarantees that the link L_1 stays in the position inverted whereas the link L_2 reaches slowly the invert position 2. Thus, the stability of the system is reached and the link L_2 does not show more than two over-signal and the stabilization is reached in the 7 seconds, approximately.



a. Link 1 and Link 2 Angles without disturbance



b. Link 1 and Link 2 Angles with disturbance

Figure 14. System response with fuzzy controller

An impulse disturbance was applied with an intensity of 1.5 Nm through 0.05 seconds (see Fig. 14.b). The fuzzy controller had a good response for this disturbance. In the simulation time the system leads the required position. The link L_2 returns to the required position in approximately 10 seconds. This disturbance leads the system to a different position, but the controller is capable to return the system for the specified position again.

The proposed methodology for obtaining the output set (*singletons*) values for the controller consists of testing the different torque values for several initial conditions of the plant. In this case, the torque values were chosen in such a way that they lead the system to the inverted position, with the minimum energy possible (minimal energy criterion). The torque output depends of both the interpolation of these values and the membership degree in the fuzzy sets.

Table II shows the synthesis results of the fuzzy controller where it can be observed the cost and performance of each fuzzy controller.

TABLE II : SYNTHESIS RESULTS FOR THE SPARTAN3E (XC3S500E-4FG320C) IMPLEMENTATION

Fuzzy Controller	4 input LUT *Max: 9,312*	SliceFFs *Max: 9,312*	IOBs *Max: 232*	RAMB16s *Max: 20*
ControllerXfuzzy1	188	148	40	8
ControllerXfuzzy2	156	148	40	8

6. THE SIMULATION ENVIRONMENT

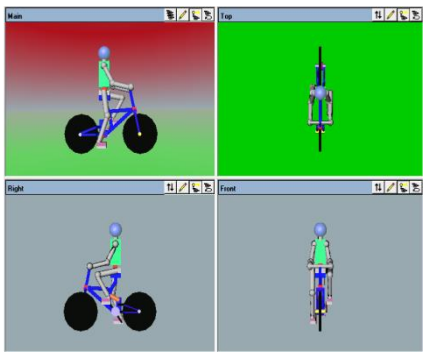
A virtual three dimensional prototype was developed for the robot cyclist using the programming language Virtual Realm Builder 2.0, such a way to supplies a graphic interface to the user.

Additionally, an object in Matlab was created to have access to prototype instances from the main program. Figure 15.a. shows the project of the cyclist robot form different point of views and each part of robot cyclist is an object that has relative coordinates with relation to its proper geometric center and absolute coordinates of the virtual space. This coordinate system is different than the Matlab coordinates, for which the programmer must define. In general 13 objects had been used for the upper´s rider and for the lower´s rider and the bicycle 45 objects were used.

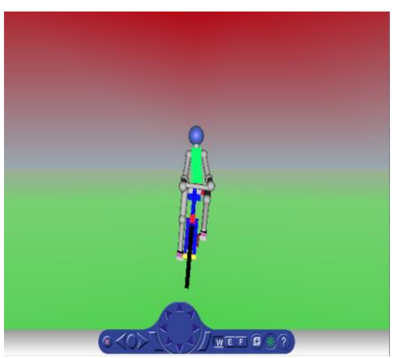
The prototype achieved in Virtual Realm 2.0 Builder and Matlab implement the Acrobat behavior, assuming that the front part of bicycle is in the same plan that the bicycle frame. The lean of the bicycle with relation to the vertical line is in agreement with the lean of the first link. Additionally, the trunk of the rider is in agreement with the lean of the second link (Fig. 15.b. and Fig. 15.c.). This is possible through the actuator action that substitutes the hip of the rider. The state variable is delivered for the virtual prototype at each time. Notice that the direction of the angles is inverted due to the fact that there is a different coordinate system between the Matlab and the Virtual Realm Builder simulator.

Fig.16 shows the overall system which comprises both the simulation environment (which was developed in Virtual Reality [33]) and the embedded FPGA controller. The embedded controller integrates: (a) the respective control tasks, (b) the gathering of the state variable of the Acrobot and (c) the generation and transmission of the control variable to the simulation environment.

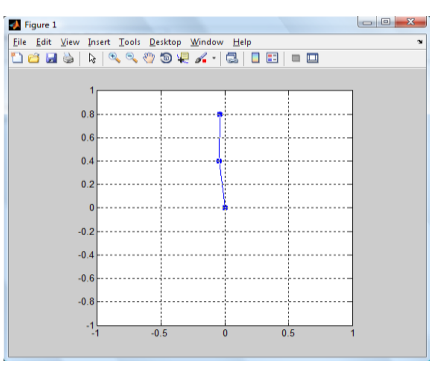
The embedded controller integrates: (a) the respective control tasks, (b) the gathering of the state variable of the Acrobot and (c) the generation and transmission of the control variable to the simulation environment.



(a) Different viewpoints



(b)Viewer WRL



(c) Acrobot in Matlab

Figure 15. Three dimensional representation of robot cyclist using Virtual Realm Builder 2.0

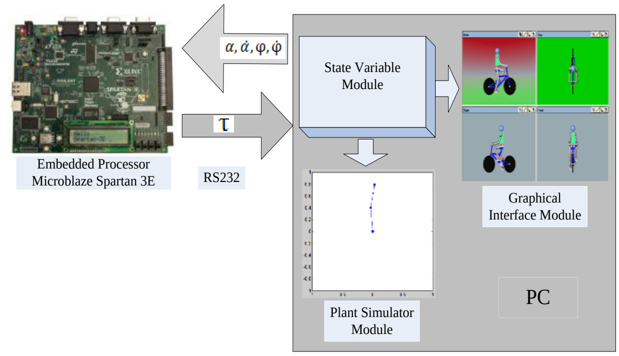


Figure 16.Full System Implementation

7. CONCLUSIONS

This paper presents the problem of equilibrium of a robot cyclist considering very low speeds including the zero speed. The dynamic system model of the cyclist robot was approximated to a model of inter-actuated double pendulum control, known as Acrobot.

This research allowed the application and comparison of control strategies for the proposed problem. In this context, four controllers were designed using conventional control methods and artificial intelligence techniques. Both modern control strategies and strategies based on artificial intelligence proved to be robust to maintain the equilibrium of the pendulum around the operating point, showing divergence only in the stabilization velocity and the effort to maintain the system in the vertical direction. This research shows the implementation of the controllers, the designing requirements for the controllers and its dependence on the mathematical model.

The intelligent controllers can be designed following an observational behavior and empirical measures of the plant. The classic controllers allow a deeper insight into the nature of the phenomenon to be controlled. The pole diagram permitted to observe that the system show reversed phase, i.e., the system tends to fall in the opposite direction of the control force, making it more difficult to stabilize.

Fuzzy controllers with multiple inputs complicate the rules generation and setting which form the knowledge base of the controller, thus the number of rules grows exponentially with the number of entries and sets defined for each entry causing conflicts between them. In this research it was observed the utility of the neuro-fuzzy techniques to handle multiple systems inputs, generating the controller from a set of training data, thus increasing the computational effort but leaving invisible the generation of the knowledge base for the user.

The models were evaluated using simulations and the fuzzy controller was chosen to be implemented in FPGA. FPGA are very useful devices for its ability to incorporate systems which design together software-hardware, so it was possible to use the tool XFuzzy 2.0 to generate the VHDL code of the fuzzy controller designed which was divided into two modules. The modules were connected in Microblaze embedded processor as peripherals acting in parallel and increasing the processing speed when compared to the general purpose processor of the computer. Evaluation criteria were adopted based on performance indicators. The use of these indicators proved to be a simple and practical technique for comparison. The mathematical model of the system allowed the plant simulation using the 4th order RungeKutta algorithm, considering a tolerance of 10% between the parameters provided by the system for validation of the plant.

System model validation was developed in a three dimensional environment using the Virtual Realm Builder 2.0 tool connected by Matlab. Tuning and optimization techniques can help adjust more systematically the parameters of controllers, increasing its performance in future researches.

REFERENCES

[1] J. P. Meijaard, J. M. Papadopoulos, A. Ruina and A. L. Schwab, "Linearized Dynamics Equations for the Balance and Steer of a Bicycle: a Benchmark and Review," Roy. Soc. London, Vol. 463, pp. 1955-1982, Dec 2006.

[2] M. W. Spong, "The Swing up Control Problem for the Acrobot," Proc. IEEE Control Systems Magazine, Vol. 15, No. 1, Feb 1995, pp 49 – 55.

[3] N. Kobori, S. Kenji, H. Pitoyo and H. Shuji, "Learning to Control a Joint Driven Double Inverted Pendulum Using Nested Actor Critic Algorithm," Proc. 9th Int. Conference on Neural Information Processing (ICONIP'O2), Vol. 5, Nov. 2002, pp 2610 – 2614.

[4] S. C. Brown and K. M. Passino, "Intelligent Control for an Acrobot," Intelligent and Robotic Systems, Vol. 18, Issue 3, Nov. 1996, pp 209 – 248.

[5] K. J. Åström, R. E. Klein, and A. Lennartsson, "Bicycle Dynamics and Control: Adapted Bicycles for Education and Research," Proc. IEEE Control Systems Magazine, Vol.25, Issue 4, pp. 26-47, 2005

[6] Hao Jianbao, Huang Xinbing."Mechanism design and dynamics analyses for an under-actuated three-joint acrobot",IEEE International Conference onConsumer Electronics, Communications and Networks (CECNet), XianNing, China, 2011.

[7] Yu-Sheng Lu, Hua-Hsu Chiu, and Shu-Fen Lien. "Practical Design and Control of a Novel Inverted Pendulum", International Conference on Modelling, Identification and Control, Shanghai, China, 2011.

[8] Cheng-Hao Huang, Wen-June Wang and Chih-Hui Chiu, "Design and Implementation of Fuzzy Control on a Two-Wheel Inverted Pendulum", IEE Transactions on Industrial electronics,Vol. 58, No 7, july, 2011.

[9] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II," Proc. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, March/April 1993, pp. 419 – 435.

[10] D. Hung, W. Zajac, "Implementing a Fuzzy Inference Engine Using FPGA," Proc. Sixth Annual IEEE International ASIC Conference and Exhibit, Sep 1993, pp. 349-352, doi: 10.1109/ASIC.1993.410736.

[11] R. Hartenstein, "Basics of Reconfigurable Computing," Embedded Computing – A Low Power Perspective, Chapter 20, Springer-Verlag, 2007.

[12] M. Brox and S. Sánchez-Solano, "Development of IP Modules of Fuzzy Controllers for the Design of Embedded Systems on FPGAs,"Proc. 16th Intl. Conference on Field Programmable Logic and Applications (FPL2006), Aug. 2006, Madrid**.**

[13] J. D. Newcomb, "A Scalable Approach to Multi-Core Prototyping," Thesis submitted to the Faculty of the Virginia Polythecnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electric Engineering, 2008.

[14] D. Kim, "An implementation of Fuzzy Logic Controller on the Reconfigurable FPGA Systems," Proc. IEEE Transactions on Industrial Electrónics, Vol 47, No. 3, Jun 2000, pp. 703 – 715.

[15] D. López, C. Jiménez, I. Baturone, A. Barriga and S. Sánchez, "Xfuzzy: A Design Environment for Fuzzy Systems," Proc. 5th IEEE Int. Conf. on Electronics, Circuits and Systems, Vol. 1, May 1998, pp. 431-434, doi: 10.1109/FUZZY.1998.686265.

[16] F. Moreno, I. Baturone, S. Sánchez and A. Barriga, "XFUZZY 3.0: A Developement Environment for Fuzzy Systems," Proc. International Conference in Fuzzy Logic and Technology (EUSFLAT'2001), Sep 2001, pp. 93-96.

[17] I. Gonzalez and F. Gomez, "Chipering algorithms in MicroBlaze-based Embedded Systems," Proc. IEE - Comput. Digit. Tech., Vol. 153, No. 2, Sep 2005, pp. 87 – 92.

[18] I. Gonzalez, , F. Gomez and S. Lopez, "Hardware Accelerated SSH on Self Reconfigurable Systems," Proc. IEEE Intl. Conference on Field-Programmable Technology,p 289 – 290, Dec. 2005.

[19] S. Sánchez, A. Cabrera, M. Brox and A. González, "Controladores Difusos Adaptativos como Módulos de Propriedade Intelectual para FPGAs," Proc. XII Workshop IBERCHIP (IWS-2006), Costa Rica, Mar. 2006.

[20] S. Sanchez, A. Cabrera, I. Baturone, J. Moreno and M. Brox, "FPGA Implementation of Embedded Fuzzy Controllers for Robotics Applications," Proc. Transactions on Industrial Electronics, Vol. 54, No. 4, 2007, pp. 1937 – 1945.

[21] Hung, V. V., Esfandiari, R. S. (1997). "Dynamic Systems: Modeling and Analysis", McGraw-Hill, Singapore, pp. 108-155.

[22] Y.E. Castro, C. H. Llanos, W. Britto, L. S. Coelho "A Study of three Control Approaches forthe Ciclist Robot Problem". COBEM 2009 - 20th International Congress of Mechanical Engineering. pp. 1-10.

[23] Chen, C.T. (1999). "Linear System Theory and Design", The Oxford Series in Electrical and Computer Engineering, Oxford, New York, USA, 3rd edition.

[24] Jang, J. S. (1993). A0FIS: "Adaptive-Network-Based Fuzzy Inference System". In: The IEEE Transaction on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665-685.

[25] Kelly, D. J., Burton, P. D., Rahman, M. A. (1994). "The Application of a Neural-Fuzzy Logic Controller to Process Control". In: First International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference, San Antonio, TX, USA, pp. 235-236.

[26] Coelho, L. S., Almeida O. M. e Coelho A. A. R. (2003). "Projeto e Estudo de Caso da Implementação de um Sistema de Controle Nebuloso". Revista SBA Controle & Automação, Vol. 14, No. 1, pp. 20-29.

[27] A. Mahmoud, Manzoul and D. Jayabharathi, "FPGA for Fuzzy Controller," Proc. IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 1, Jan. 1995, pp. 213-216.

[28] H. Rosinger, "Connecting Customized IP to the Microblaze Soft Processor Using the Fast Simplex Link (FSL) Channel", Proc. Application Note: Microblaze, XAPP529 (v1.3), Xilinx 2004, pp. 1-12.

[29] Y. Zong-Mu, L. Kuei–Hsiang, “A systematic approach for designing multistage fuzzy control systems,” Proc. Science Direct Fuzzy Sets and Systems Elsevier, Vol. 143, Issue 2, pp. 251-273.

[30] Copetti, C. T., Coelho L. S. e Coelho A. A. R. (2007). “Controle Nebuloso Adaptativo porModelo de Referência: Projeto e Aplicação em Sistemas não Lineares”. RevistaControle & Automação, Vol. 18, No. 4, pp. 479-489.

[31] Y. E. Castro, C. H. Llanos, W. Britto, L. S. Coelho, “Fuzzy Control for Cyclist Robot Stability using FPGAs”. In: ReConFig 09-International Conference on Reconfigurable Computing and FPGAs, 2009.pp. 410-415. doi: 10.1109/ReConFig.2009.53.

[32] Xfuzzy home page: http://www.imse.cnm.es/Xfuzzy, accessedonMarch 30, 2017.

[33] V-Realm™ Builder, 2009, “User's Guide and Reference”, 2009, <http://metalab.uniten.edu.my/~farrukh/vrml/user_guide.pdf>.

Yesid Enrique Castro Caicedo. He is graduated in Physical Engineering from UNICAUCA (University of Cauca, Colombia), (2006), and MSc in Mechatronics Systems (2009) from UnB (University of Brasilia, Brazil). He has experience in Mobile Robotics, Mechatronics, Reconfigurable Systems, and AutomationSystems.

Walter de Britto Vidal Filho. He is graduated in Mechanical Engineering (1995) from UFPE –, Brazil, and MSc in Mechanical Engineering from PUC-RJ, (1998), and PhD in Mechanical Engineering from USP, Brazil, (2003). He is professor in UnB – University of Brasilia and has experience in Mechanical Engineering, mechatronics.

Edward David Moreno. Bachelor in Electrical Engineering at UNIVALLE (Colombia, 1991), MSc in Electrical Engineering – USP, Brazil (1994) and PhD in Electrical Engineering – USP, Brazil (1998). He is professor at UFS – Federal University of Sergipe, Brazil. He has experience in Computer Science, focusing on Computer Architecture, parallel processing, high performance computing, Hardware security, Digital design using FPGAs, and embedded systems.

Leandro dos Santos Coelho. Associate professor of Mechatronics Engineering, Pontifical Catholic University of Parana (PUCPR). He has experience in Evolutionary algorithms, Neural networks, Swarm intelligence, Quantum computing and Nonlinear identification.

Carlos H. Llanos Quintero. He is graduated in Electrical Engineering from UNIVALE (Colombia, 1983), MSc in Computer Science from UFMG, Brazil (1990) and PhD in Electrical Engineering from University Sao Paulo (USP, 1998), Brazil. He has experience in Computer Science, focusing on Computer Systems Architecture, acting on the following subjects: FPGAs, mechatronics, automation and control, and reconfigurable systems..