# Rapid Prototyping using HIL, PID and Fuzzy Logic Controller

**Daw M. Alzentani**
.[1] (*Department of Electrical and Electronic Engineering*)
*Alzawia University,* Libya.
*daw_425@yahoo.com* .

**Almokhtar M. Alzhari**
.[2] (*Department of Computer Engineering*)
*Alzawia University,* Libya.
*ali441@yahoo.com* .
.

## Abstract

Rapid Prototyping of embedded hardware/software systems is important, because it shortens the path from specification to the final product. Prototypes play a major role in decision making, concept and design validation, feature and limit exploration, as well as design verification in every phase of the product development cycle, including product planning, requirement engineering, and product development. Rapid prototyping of embedded systems can emulate different kind of processes, through the mathematical modelling that represent their dynamic characteristics. This make easier to emulate different control strategies, which can interact with the real signals of the embedded process, making a better approach of the real response than a simple simulation. This paper presents the results of the emulation of the dynamic behaviour in the study case to work, in order to validate control strategies like PID and Fuzzy, using the concepts of rapid prototyping and hardware-in-the-loop (HIL). To achieve this objective, two embedded systems were employed, the first one to emulate the dynamics of the process, and the second one to implement the control strategy. Both systems were interconnected using the Controller Area Network protocol (CAN). The principal contribution of this work is the methodological development in the application of the control strategies through HIL

**Keywords:** HIL, CAN bus, Fuzzy controller, PID controller.

## i. INTRODUCTION

Nowadays, one of the most useful strategies for such a technological product development is the communication platform between each devices and components of the whole system [1].

CAN protocol technology, uses in the interconnection task of the embedded system, allowing the distribution of the control process through different stages using a modular structure [3].

The CAN protocol facilitates the development of control systems based in the concept of rapid prototyping, which can emulate processes or control methods through the use of Hardware-in-the-loop techniques, as show in [4]. This help the product development, reducing the time and production costs, giving the possibility to experiment with the dynamic signals in an embedded system and test different control strategies, in order to obtain a final desired system that interacts through a communication protocol [5, 6].

As support of the HIL concept, exist tools of rapid prototyping which allow to generate the code to be implemented in embedded systems since process simulation [7]. An example of this kind of tools is the Waijung® Blockset developed by Math works [8].

This tool creates code for ARM® processors based in a MatLab-Simulink Package. An important work about rapid prototyping is presented in [9], where is shown the application of modern procedures to implement control methods to regulate climatic behaviours into a vehicle, which are complemented with HIL to emulate the control strategies using models of heating, ventilation and air conditioning elements, which finally provide an efficient tool to verify, validate and optimize the control algorithms in real time, that are implemented in interconnected platforms through communication interfaces, reducing costs and production times in the process.

An implementation of hardware in the loop to design a flexible architecture to control an electric hybrid vehicle platform, presented in [10]. is taking into account methods that guarantee the emulation in real time. Methods based on multi-thread technology, and implemented in development platforms which use the high speed CAN communication protocol, have as result the emulation of hardware and the control strategies which validate the process, removing errors automatically.

Hence, this paper describes a product development methodology using the rapid prototyping and HIL concepts, and based on the implementation of the Waijung® tool, to describe the dynamic process in the embedded systems, communicated through CAN protocol with the simulated models in MatLab-Simulink®. According to this, are obtained the results to apply this method to operation stages in a process with different control strategies in discrete time.

This document is organized as follows: In section I, the preliminary and introductory theoretical basis about the developed work is presented. In section II, is described the principal documentation of the software tool to use, as well as the main features of the CAN protocol and its implementation characteristics in the embedded system. In section III, is summarized the analysis of results. And finally, the conclusions of the work are presented.

ISSN 2356-5608

### ii. CAN protocol characteristics

For the development of this work, is proposed the CAN protocol to communicate two embedded systems of the ARM® family, to interconnect the signals between the embedded process in one board with the control system emulated in the second board. As control strategies a PID controller and a Fuzzy Control were tested. The plant to control consists of a PITTMAN direct current motor. In Figure 1, a diagram of this application is presented.

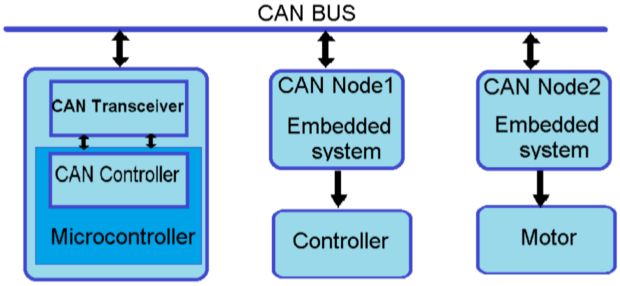

**Figure 1:** Connection CAN bus diagram

### iii. Mathematical Model of the DC Motor

To calculate the mathematical model of the DC motor, as a first step, must be defined if the motor is controlled by armature or by field, in this case, the motor is controlled by armature, so the electro-mechanical diagram of the motor is defined as is depicted in the Figure 2.

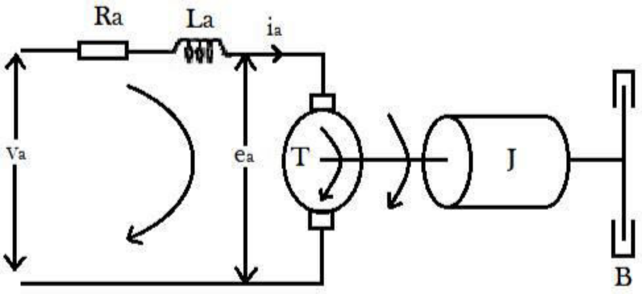

**Figure 2:** DC Motor armature Current controlled

From schematic Diagram figure 2, the dynamic behavior of the electrical part is presented [11].

$$v_a(t) = R_a * i_a(t) + L_a * \frac{di_a(t)}{dt} + e_a(t) \tag{1}$$

In the equation (2), is shown the mathematical model of the mechanical part [12].

$$T(t) = J * \frac{dw(t)}{dt} + B * w(t) \tag{2}$$

The relation between the mechanical and electrical part is given in the equation (3) and (4) respectively.

$$T(t) - T_L = K_m * i_a(t) \ \Rightarrow \ T_L = 0 \tag{3}$$

$$e_a(t) = K_e * w(t) \tag{4}$$

Substituting the equation (4) in (1) is obtained the equation (5).

$$v_a(t) = R_a * i_a(t) + L_a * \frac{di_a(t)}{dt} + K_e * w(t) \tag{5}$$

Then it is replaced the equation (3) in the equation (2), where is obtained the equation (6).

$$K_m * i_a(t) = J * \frac{dw(t)}{dt} + B * w(t)$$
$$i_a(t) = \frac{J}{K_m} * \frac{dw(t)}{dt} + \frac{B}{K_m} * w(t) \tag{6}$$

Finally, the equation (6) is replaced in the equation 5, where is obtained the equation (7) that represents the system behaviour.

$$v_a(t) = \frac{R_a * J}{K_m} * \frac{dw(t)}{dt} + \frac{R_a * B}{K_m} * w(t) + \frac{L_a * J}{K_m} * \frac{d^2w(t)}{dt} + \frac{L_a * B}{K_m} * \frac{dw(t)}{dt} + K_e * w(t) \tag{7}$$

Applying Laplace in equation (7) is obtained the equation (8)

$$v_a(s) = \left(\frac{R_a * J}{K_m} * s + \frac{R_a * B}{K_m} + \frac{L_a * J}{K_m} * s^2 + \frac{L_a * B}{K_m} * s + K_e\right) * w(s) \tag{8}$$

Resolving the equation (8), should be obtained the transfer function of the system which is shown in the equation (9).

$$\frac{w(s)}{v_a(s)} = \frac{\frac{K_m}{L_a J}}{s^2 + \left(\frac{R_a}{L_a} + \frac{B}{J}\right) s + \left(\frac{R_a B}{L_a J} + \frac{K_e K_m}{L_a J}\right)} \tag{9}$$

The variables values that characterize the 24 volts PITTMAN motor DC054B-2 (14202), are presented in table 1 [13], and are used to simulate the system.

**Table 1:** Features motor data [13]

| Parameters | Symbol | Value |
|---|---|---|
| Armour Resistance | $R_a$ | $1.73\,\Omega$ |
| Armour Inductance | $L_a$ | $2.54 * 10^{-3} H$ |
| Friction Coefficient | $B$ | $0.09\ Nm - s/rad$ |
| Electrical Constant | $K_e$ | $0.055\ Vs/rad$ |
| Mechanical Constant | $K_m$ | $0.042\ N - m/A$ |
| Inertial Moment | $J$ | $0.0023\ Kgm^2$ |

The values in table 1 should be replaced in the equation 9 to obtain the final equation (10), which describes the system behaviour.

$$\frac{\omega(s)}{V_a(s)} = \frac{7189}{s^2 + 720.2s + 27050} = G(s)$$

( 10)

The DC Motor's transfer function is obtained, and then we have to calculate the PID and the Fuzzy controller.

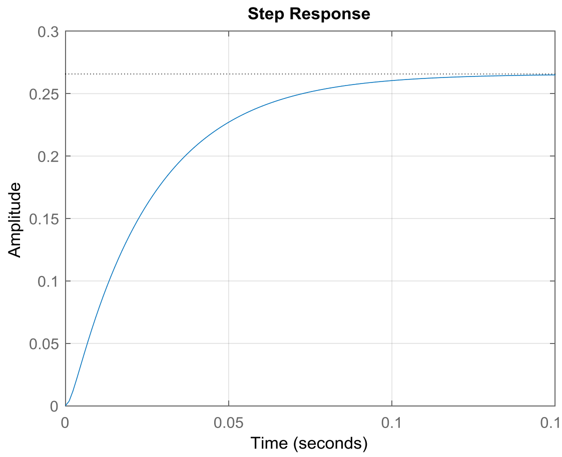

**Figure 3** DC Motor Time Response

### iv. Calculation of PID Controller

To calculate the PID controller, first a system discretization stage must be performed. For this case the discretization method to use, is the ZOH (Zero Order Hold), which executes the Z transformation using partial fractions of the continuous control signal [14].

$$G(z) = Z\{G_{ZOH}(s) * G(s)\} = (1 - z^{-1})\left\{\frac{G(s)}{s}\right\}$$

(11)

Then, T is defined as the sample time, used to perform the Discretization of the plant, which for this case is:

$$T = 0.0012789$$

Finally the discretized plant transfer function is represented by the equation (12).

$$G(z) = \frac{0.0044083z^{-1} + 0.0032477z^{-2}}{1 - 1.3693z^{-1} + 0.3981z^{-2}}$$

(12)

The discretized plant's transfer function obtained, then we have to calculate PID parameters, controller with filter in discrete time is conducted using the characteristic transfer function shown in equation (13) [15].

$$PID(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 + (so - 1)z^{-1} - soz^{-2}}$$

(13)

With the definition of the characteristic equation of a PID controller with filter, the process is continued with the definition of the characteristic and the desired system polynomial. The characteristic polynomial is a representation that shows the current plant behaviour, and has as variables the four factors defined by equation (13). the desired polynomial represents the wanted behavior for the plant. Keeping this in mind, will be defined the representative equation. The equation (14) defines the structure of the desire polynomial.

PC(z) = 1+

$$+ \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 + (so - 1)z^{-1} - soz^{-2}} \frac{0.0044083z^{-1} + 0.0032477z^{-2}}{1 - 1.3693z^{-1} + 0.3981z^{-2}}$$

14)

To complete the definition of the desire polynomial has to be Set the sample time value, in this case is the same value of Time used to discretize the plant. Also, the parameters that define the plant behaviour after the control application have to be set. So, is proposed the characteristics in table 2.

**Table 2:** Design parameters [13]

| Parameters | Value |
|---|---|
| Error | 2 % |
| Ts / open loop | 0.012789 |
| Desired Ts | 95 % \| open loop |
| Number of Samples | 10 samples |
| Overshoot | 10 % |

With the overshoot (Mp), is defined the value of zita $(\xi)$ to be used in the calculation of the controller. Using the equation (15) is obtained this value.

$$\xi = \sqrt{\frac{1}{1 + \frac{\pi^2}{\ln (Mp)^2}}}$$

(15)

Resolving the equation, the value of zita corresponds to:

$\xi = 0.591155$

The next variable to use is the natural system frequency, represented by equation (16).

$$Wn = \frac{-\ln(error)}{\xi * 0.95\, ts|_{open\ loop}}$$

(16)

$$Wn = 544.6780077\frac{rad}{s}$$

Obtaining the value of Wn, the next variable to calculate is Wd, for this is used the equations(17) and (18):

$$if\ 0 < \xi < 1\ then\ Wd = Wn\sqrt{1 + \xi^2}$$

(17)

$$if\ \xi \geq 1\ then\ Wd = \sqrt{Wn^2 + \xi^2}$$

(18)

$$Wd = 632.733064\frac{rad}{s}$$

When obtained these values, should be proceeded to determinate the magnitude and angle of the system in the Z domain, the values, are calculated using equation (19) and (20) respectively.

$$|Z| = e^{-T*\xi*Wn}$$

(19)

$$\angle Z = T * Wd$$

(20)

The desired polynomial is represented by the equation (21) and is calculated using the real and imaginary values obtained in (19) and (20).

$$Pd(k+t) = (1 - re + jim) \otimes (1 - re - jim) \otimes \left(1 - e^{-T*\xi*Wn}\right) \otimes \left(1 - e^{-T*\xi*Wn}\right) \quad (21)$$

Where$\left(1_e_{-T*\xi*Wn}\right)$, is a non-dominant Z pole.

The next step is the equalization of the characteristic and the desired polynomial to finally determinate the values in the equation (13) which represent the PID controller with filter.

$$PID(z) = \frac{177.2707 - 190.3629 * z^{-1} + 53.23637 * z^{-2}}{1 - 0.56599 * z^{-1} - 0.43401 * z^{-2}}$$

### v. Calculation of the Fuzzy Controller

Nowadays a useful alternative to design the control of a system is using the fuzzy logic concepts, which allows through the expert understanding of the process to generate a knowledge base for the system, and provide it the ability to
Make decisions about certain actions to rule its operation [16]. To implementing a fuzzy control, is necessary to define the fuzzy logic of the process, this allows to the systems to work with no accurate information and with qualitative values, such as low error, decrement control, among other variables, which are defined as linguistic values [17].

To develop a fuzzy controller, first starts defining the output variables of the system and the structure of the input variables. The control output in this case is given directly by the DC motor operation. To define the input, should be considered what kind of variable controls the output, thus is defined as input variable the error, calculated from the difference between the system reference and the feedback plant value. This process is depicted in the Figure 4.

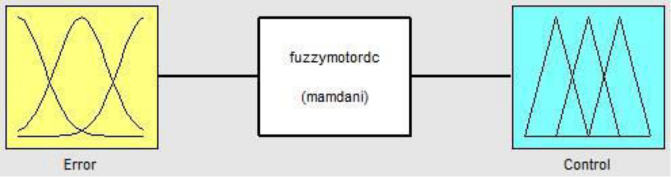

**Figure 4:** fuzzy control structure

Considering the fuzzy sets, then is proceeded with the clusters structure that set up the input and output variables. For the input variable (Error), are defined five clusters, four of them are trapezoidal and one triangular, each one obeys to the error linguistic variables defined as: very low, low, none, high and very high. The shapes for the clusters were chosen in judgment of the developers. Finally that set includes a range of values from-4 to 4, because the error system values vary within this range, as it can be seen at Figure 5.

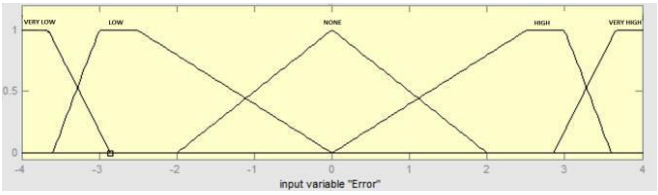

**Figure 5:** Input Fuzzy Control

In the output control variable case, it has five clusters, four of them are trapezoidal and the last one is triangular, each one corresponds to a linguistic control value defined as: Decrease, few decreases, none, few increase, and increase. In this case, the sets have a range of-300 to 300, to generate the sub-damped desire response, as shown in Figure 6.

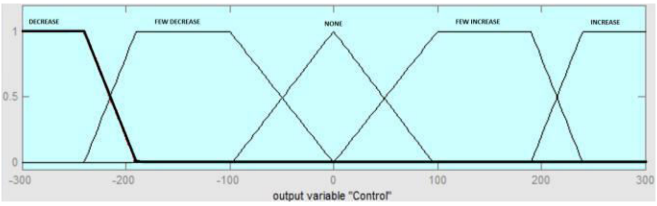

**Figure 6*:* Output Fuzzy Control**

To set the fuzzy rules that define the control operation, in this case is proposed that for an error input given as very low, the control must have a decrease value and for a very high error the control must increment, thus, are proposed intermediate cases to complete the set of rules to make the control work, as shown in table 3.

**Table 3:** Control Rules

| Rule | Case |
|---|---|
| 1. | If (Error is very low) then (Control is Decrease) |
| 2. | If (Error is none) then (Control is none) |
| 3. | If (Error is very high) then (Control is Increase) |
| 4. | If (Error is low) then (Control is Few Decrease) |
| 5. | If (Error is high) then (Control is Few Increase) |

After to define the input and output fuzzy sets, and the stipulated fuzzy rules, is proceeded with the inference process as shows in Figure 7, for this case was chosen the Mamdani process. This method is based on the calculation of membership values for different clusters in each input space

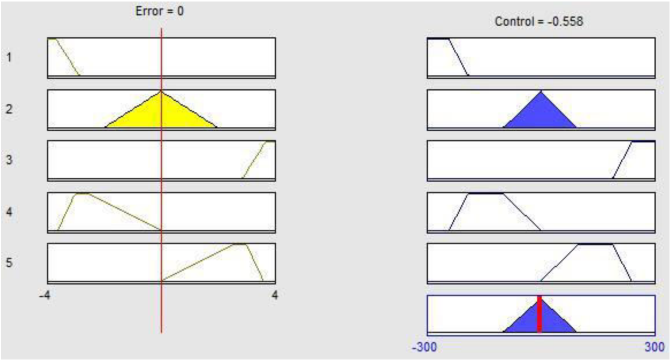

**Figure 7:** Inference Process to Zero Input Error

Finally, must be performed the defuzzification process, which for this case is used the centroid method, and consists in the calculation of the gravity centre from the resulting sets after the implementation of the controller rules. This defuzzification method is defined by the equation (22).

$$y_{centroide} = \frac{\sum_{x\in X} x\mu_A(x)}{\sum_{x\in X} \mu_A(x)} \qquad (22)$$

### vi. Embedded System Program

To program the embedded controllers, is used the Matlab® Simulink® tool, which allows to design the feedback control system. The first step is the implementation of the discretized plant, for this, is required the basic program configuration of the development board to use in the block Simulink Target Setup, then are configured the CAN and USB VCP communication protocols. The CAN protocol allows to perform the communication between the embedded control board and the discrete plant board. To configure this protocol PB8 and PB9 pins are set up to receive and transmit information respectively, also the baud rate is configured to a rate of 125Kb, and the line communication which the data transfer is performed, corresponds to CAN LOW.

Moreover the USB VCP protocol is used to display the system response from a pulse train input. The visualization can be performed in the setup of the reception blocks as shown in Figure 8.

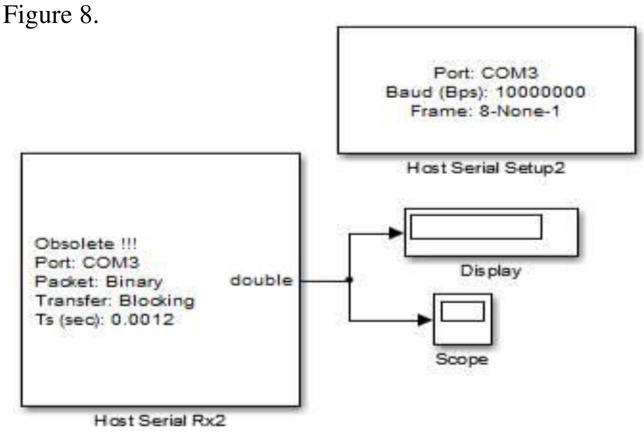

**Figure 8:** Block diagram for displaying the Plant Response

The program also offers the possibility to observe the system response in an oscilloscope through a DAC (Digital-to-Analog converter) output, programmed in the development board.

Continuing with the board configuration, is proceeded with the implementation of the plant to work and the integration of the previously mentioned blocks, to complete the programming task of the board, and emulate the behavior of a controlled by armature DC motor, as shown in the Figure 9.

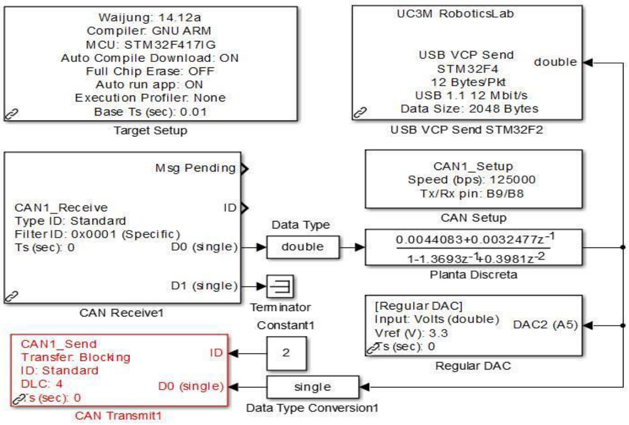

**Figure 9:** Block Diagram for Data Transfer and Emulation DC Motor

In Figure 9 can be observed, the data is received through the CAN module, then is converted to double data type and connected with the plant, next the output data is returned using the same CAN protocol line in order to feedback the plant response, and calculate the error to introduce it as control parameter and determinate the proper action for the plant to obtain the desire behavior.

### vii. Embedded Control Programming

After implement the plant in the embedded system, is continued with the programming of the control methods to handle the desired response of the system. The Figure 10 shows the block diagram of the programming of the development board that is responsible of the system emulation controller. There is shown some blocks are surrounded by coloured squares which will be explained in next section of this work

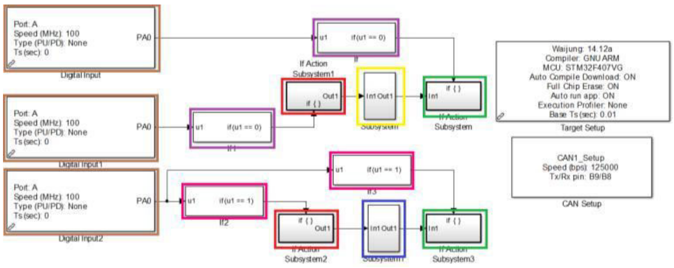

**Figure 10:** Block Diagram for Configuring the PID and Fuzzy Controllers

First, the program gets as parameter a value of zero or one given by the user, which determinates if the controller to use is PID or Fuzzy, this signal is acquired through the PA0 pin (brown Colored Square). If this signal is zero, the purple block is activated and allows performing the functions in the red, yellow and green subsystems within a loop until the signal gets the value of one. For this case the PID controller blocks are activated.

The red subsystem is responsible to receive the plant values from the CAN protocol this is used for both controllers to work. It consists of a reception CAN block that filters the data received with ID 2, with a sample time of 0.0008 seconds, then the data is sent to the following subsystems in the control line. The data flow is shown in Figure 11.

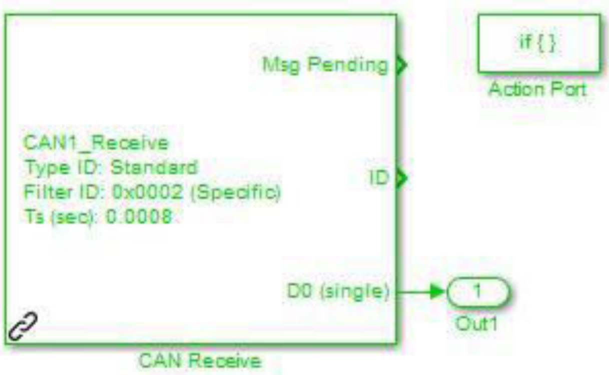

**Figure 11:** Reception Module Configuration through CAN Protocol

The yellow subsystem contains the PID controller structure, which gets the data from the red subsystem to be processed. The received data comes from the plant feedback given and enters to a sub-process where is performed a subtraction with the reference signal. The result, enters to the controller which calculates the values that must be set on the plant to change and generate the desired behaviour. This subsystem is depicted in Figure 12, which shows how the error is calculated to enter in the green subsystem.

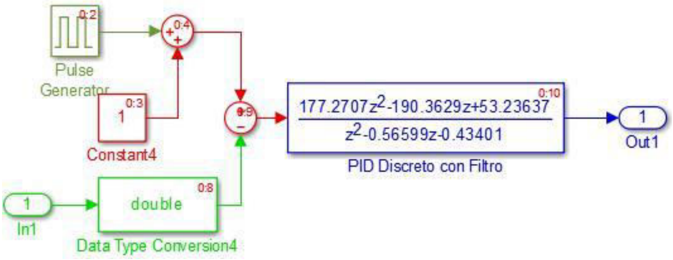

**Figure 12:** PID Controller Setup

From the yellow subsystem to the green subsystem is transferred the variable through CAN protocol, to change the system behaviour. In this case the CAN transmission module is configured with a sample time of 0.012 seconds to send the signal, because the sample time of the controller and the plant have the same value that is shown in Figure 13.

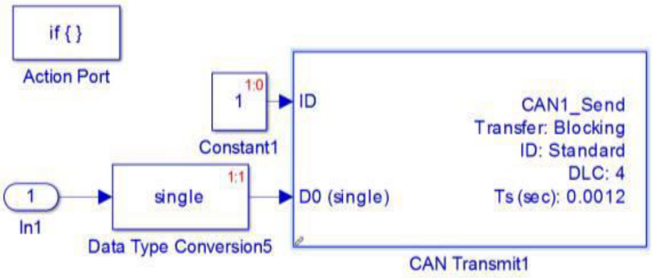

**Figure 13:** CAN Transmission module Setup

Now is analysed the behaviour of the program when the signal has a constant value of one and is entered to the brown block of the Figure 9. This action performs the activation of the pink blocks and deactivating the purple blocks.

When the pink blocks are activated the signal enters as parameter for the red subsystem, which as previously mentioned has the same function of the reception block subsystem for the PID controller.

Then the feedback value enters to the blue subsystem, where the subtraction between the reference system and the value generated by the plant is performed. Is observed that the value generated by the plant have a compensation of 0.955 to delete peaks in the control signals that cause a system destabilization, however as consequence the plant does not reach the desired reference. The used controller in this stage is a fuzzy controller whose input is the error and the output is the control signal, as shown in Figure 14.

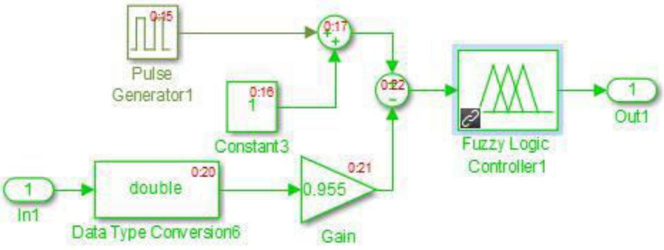

**Figure 14:** Fuzzy Controller Setup

This subsystem throws an output variable that enters to green subsystem, which transmits it by CAN protocol to the plant and achieve the desired effects in the system. As it is shows in this case, the sample time of the protocol handles a value of 0.0008, which mark a difference between the values of transmission between PID controller and the Fuzzy Controller as it is shown in Figure 15.

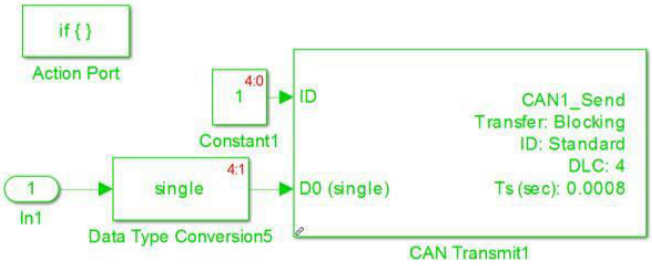

**Figure 15:** CAN Transmission Module Setup

## viii. Analysis of Results and Discussion

After assembling the system embedded controllers, in the Figure 16 the response of the system is obtained. There is shown that for stabilize adequately the system, has a strong control signal to keep the system in the desire point after a change in the reference, in this case the control generate a peak in the signal with a big overshoot (overshoot reached in the second step). This kind of behaviour is observed when the system have a settling time longer and it must force the plant to stabilize in short times.

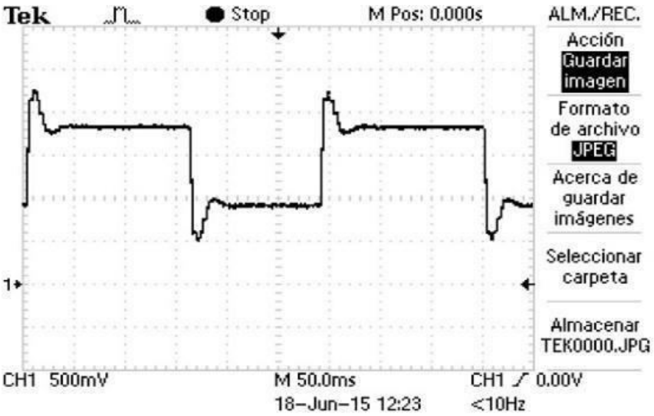

**Figure 16:** PID Controller Response

In Figure17 is depicted the signal taken from the CAN module.

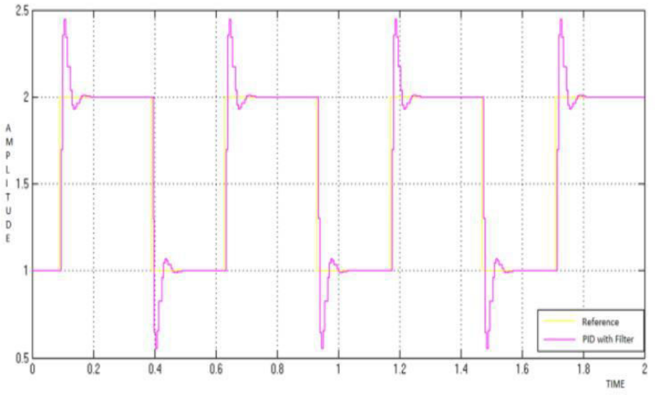

**Figure 17:** PID Controller Response Taken From HOST

Regarding to the stabilization of the system, is presented a reduced settling time and to achieve it are sufficient few samples.

For a Fuzzy controller is necessary more samples to stabilize the system in a desired reference, because is complicated to get a robust fuzzy control, for this case is due to the generation of the control rules which do not take into account settlings times or overshoots. So it generates a strong control signal if the error is maximum and a weak control signal if the error is minimum. Hence, the obtained control over the system is observed in the Figure 18.

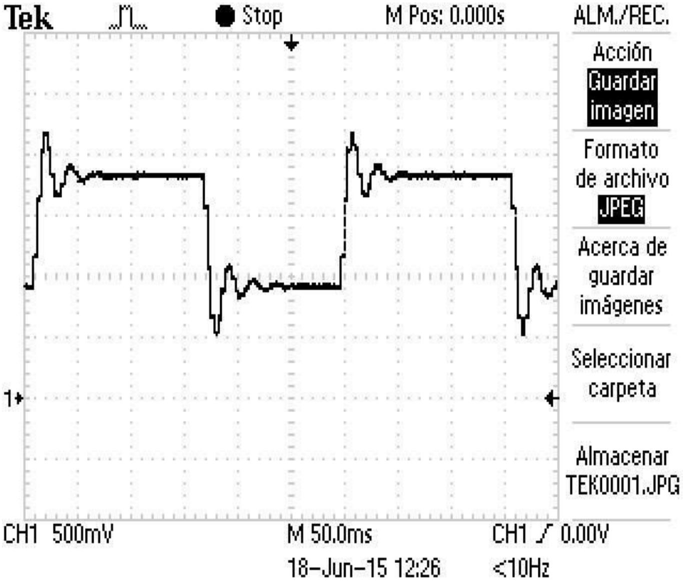

**Figure 18:** Fuzzy Controller Response

In Figure 19 is shown the signal that taken by the CAN module for the fuzzy controller.

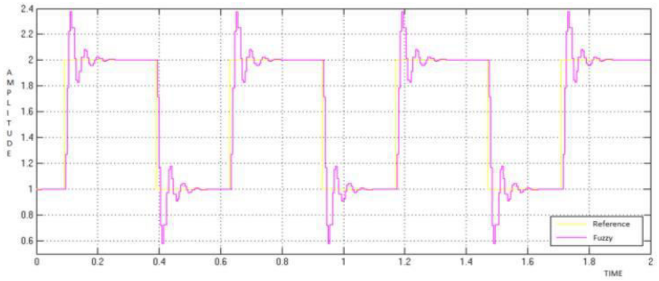

**Figure 19:** Fuzzy Controller Response Taken From HOST

Applying these two control methodologies, is necessary keep in mind that although the PID controller had a better response and stabilization with fewer samples than the fuzzy controller, can be observed that the last controller provides more robustness to system disturbances, or unwanted signals, due the rules and conditions of the clusters.

Another point to analyse into the development of this work, is the fact to have several transmission and reception CAN modules, instead of one for transmission and one for reception. This is because as shown previously, the reception and transmission has to be done with different sample times for each process. If the same sample times would have taken the control signal would distorted, having as a consequence an unstable system.

## ix. CONCLUSION

During the development of the work. It was observed that the sensitivity of the controllers in discrete time increases when are performed slight changes of the sample time, so is important to keep in mind that to perform the design of the control, first is necessary to define the time values to avoid interfere in the operation of the system when exist modifications by different control strategies.

Even though a conclusion may review the main results or contributions of the paper, do not duplicate the abstract or the introduction. For a conclusion, you might elaborate on the importance of the work or suggest the potential applications and extensions.

Moreover, the development of this work counted as advantage with the implementation of tools as Matlab® and its toolboxes, which allow the designer to focus in the knowledge on how to provide solutions to certain works, leaving as a second item, generating lines of C code, implemented in this type of development boards that intend emulate the behaviours of everyday life processes.

As future work, is proposed to perform the emulation of different parts of a vehicle and interconnect them through CAN communication protocol. As complement is planned to build a graphic interface to observe the behaviour of the emulated systems and apply possible control methods.

## REFERENCES

[1] P.S. Kedareswar and V. Krishnamoorthy, "A CAN protocol based embedded system to avoid rear-end collision of vehicles," in Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on, Feb 2015, pp. 1-5.

[2] Xiaoyuan Z., Hui Z., Dongpu C., Zongde F., “Robust control of integrated motor-transmission powertrain system over controller area network for automotive Applications” Mechanical Systems and Signal Processing, Volumes 58-59, June 2015, Pages 15-28.

[3] T.R. Jena, A.K. Swain, and K. Mahapatra, "A novel bit stuffing technique for Controller Area Network (CAN) protocol," in Advances in Energy Conversion Technologies (ICAECT), 2014 International Conference on, Jan 2014, pp. 113-117.

[4] A.Devi, G. Gnanavel, and G. Antoni Gracy, "MCS-51 microcontroller based industrial automation and control system using CAN protocol," in Communications and Signal Processing (ICCSP), 2014 International Conference on, April 2014, pp. 61-65.

[5] Zhibin S.; Hui Z.; Junmin W.; Jianqiu L.; Minggao O.; "Combined AFS and DYC Control of Four-Wheel-Independent-Drive Electric Vehicles over CAN Network with Time-Varying Delays," in Vehicular Technology, IEEE Transactions on, vol.63, no.2, pp.591-602, Feb. 2014. doi: 10.1109/TVT.2013.2279843

[6] Richardson, P.; Elkateeb, A.; Sieh, L., "An adaptive real-time intravehicle network protocol for intelligent vehicle systems," in Vehicular Technology, IEEE Transactions on, vol.53, no.5, pp.1594-1606, Sept. 2004. doi: 10.1109/TVT.2004.833616.

[7] R. Subramanian, P. Venhovens, and B.P. Keane, "Accelerated design and optimization of battery management systems using HIL simulation and Rapid Control Prototyping," in Electric Vehicle Conference (IEVC), 2012 IEEE International, March 2012, pp. 1-5.

[8] Caballero A., Copaci D., Blanco M. Moreno L. Herrán J., Fernández I., Ochoteco E., Cabañero G., Grande H.; “Innovative Pressure Sensor Platform and Its Integration with an End-User Application”; Sensors 2014, Vol. 14 pp. 10273-10291; doi:10.3390/s140610273.

[9] D. Michalek, C. Gehsat, R. Trapp, and T. Bertram, "Hardware-in-the-loop-simulation of a vehicle climate controller with a combined HVAC and passenger compartment model," in Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on, July 2005, pp. 1065-1070.

[10] Heike B.; Florian R.; Toralf B.: Development and hardware-in-the-loop test of a guidance,navigation and control system for on-orbit servicing; Acta Astronautica; Vol. 44; 2014; pp. 67-80.

[11] Youngki K.; Ashwin S.; Jason B.; Zoran F.; Anna S.; Tulga r.; ” Hardware-in-the-loop validation of a power management strategy for hybrid powertrains”; Control Engineering Practice; V 9; 2014; pp. 277-286.

[12] Zeina B.; Samih J.; Imad K.; “Modeling and Simulation of Series DC Motors in Electric Car”; Energy Procedia; Vol. 50; 2014; pp. 460-470.

[13] PITTMAN. (2015) PITTMAN DC054B-2 (14202) Brush DC Motor. [Online]. http://www.pittman-motors.com/Brush-DC-Motors/14202-Brush-DC-Motor.aspx

[14] Xie L.; Shieh L.; Tsai J.; Wu C.; Park J.; “Digital and analog integer delayed modeling and control for multivariable systems with multiple time delays in states, inputs and outputs”; Applied Mathematical Modelling; 2015. DOI: 10.1016/j.apm.2015.08.012.

[15] K. Benjamin C, Sistema de Control Automatico, cap 10, pag 708-714: Prentice Hall, Septima Edición.

[16] Cigánek J.; Filip N.; Štefan K.; "Modeling and Control of Mechatronic Systems Using Fuzzy Logic"; international review of automatic control; Vol 7, No 1, 2014. DOI: http://dx.doi.org/10.15866/ireaco.v7i1.1291.

[17] Ibrahim H., Mahmoud E.; "A Comparative Study of PID, Fuzzy, Fuzzy-PID, PSO-PID, PSO-Fuzzy, and PSO-Fuzzy-PID Controllers for Speed Control of DC Motor Drive"; International Review of Automatic Control; Vol 6, No 4, 2013.