Autotunning of P/PI Fuzzy Gains for a Rapid Control of a Brushless Motor in an Hybrid System

Geovanny Robles-Escobar*, Luis López-Pérez*, Carlos Ríos-Rojas*, Arnulfo Rosales-Quintero*,
Rafael Mota-Grajales*, Horacio Solís-Cisneros*
*Instituto Tecnológico de Tuxtla Gutiérrez, Tuxtla Gutiérrez, Chiapas, México
iqgjre@gmail.com, lpladrian_84@gmail.com, crios@ittg.edu.mx,
arnol122@gmail.com, rafael.mota.grajales.13@gmail.com, sochoracio@gmail.com

***Abstract*—In this paper the experimental modeling of the transfer function of an DC voltage three-phase brushless synchronous electric motor (Brushless DC Electric Motor) for a hybrid system is presented, the bench system consists of an architecture in parallel using devices for power division called electromagnetic clutches, that consist in a double canal pulley, a coil and a drag plate, these clutches were connected to the motor′s arrow in this hybrid system, allowing an individual control of both. The identified transfer function is used to design and to simulate a control of angular speed being used the Root-Locus technique and Fuzzy Logic to autotune the proportional integral gains. The primary objective is to synchronize the engine speeds to make efficient the fuel use in the combustion motor. The simulation of the control system in closed loop is made in the Simulink tool and is designed with Siso tool from the mathematical software Matlab. The system identification is conducted through the System Identification Toolbox of Matlab with data acquisition of angular speed, voltage and sampling time.**

Keywords*—brushless motor, electromagnetic clutches, velocity, feedback control, fuzzy logic.

I. INTRODUCTION

Nowadays the conventional control techniques can not fulfill the most important needs to optimize dynamic systems, electronic circuits, industrial processes, among others. During the last couple of years the control action for manipulated variables within a feedback architecture control has been applied altogether with artificial intelligence with the purpose of providing an efficient solution to the control of output variables of superordinated systems that not necessarily present a linear correlation with an input signal of the system. The fuzzy control offers an advantage to use linguistic terms like: enough, almost little, much, something, among others. This makes raise the problema statement in the minimum terms, that would make an highly trained person in the area [1].

Previous work by [2], [3], [4], consider the necessity to implement effective controllers in the digital control of brushless motors, from which the most common are the PID controller, controller by fuzzy logic and the combination between these: neuro-fuzzy networks, fuzzy-genetic algorithms, without forgetting that the main requirement for the application of this type of controller is the stability and reliability. Investigations related to the PID control application on a motor and [5], [6], have reported that is very common to exclude the derivational part from a classic control system since more advanced electronic devices are physically needed and of higher costs, which provide speed to levels of nanoseconds so that the derivational action provides speed to the integral part, this is to correct the error in the steady state of the proportional gain in the control system.

According to the previously mentioned investigations there is a need in the development of tuning methods that can be incorporated in hardware modules, this paper is focused in finding the technology or methodology for the parameters tuning [7]. The necessity to tune gains on time capable to adapt to different forcing functions has taken many researchers to apply the fuzzy logic to design controllers able to tune gains in real time of a classic control, due to their versatility and effectiveness when approaching nonlinear uncertain systems in their output response [8].

Architectures such as [9] implement a speed control for a brushless motor using genetic algorithms based on fuzzy logic to evaluate the linguistic rules and to tune the parameters of the controller on line, using a Digital Signal Processor (DSP) for speed data acquisition, improving performance and the robustness in the fuzzy genetic control. An optimal PID speed control is proposed by [10] using a quadratic linear regulator for a dc brushless motor with a dynamic model of second order, the controller only receives the signal of the error and it doesn′t need to feedback the complete states for the speed.

In this paper the implementation of a classic fuzzy controller able to tuning in real time the proportional and integral gains, using a gain factor range, obtained through the root-locus plot of the polynomial characteristic of a transfer function in open loop, and the dynamic modeling of the brushless motor is conducted through the sampling of data correlated with the test data collected, in addition a comparative study of the transitory reponses is performed regarding to conventional PI control and a PI with fuzzy logic. In the control system, the control variable is the angular speed of the brushless motor and a manipulated variable is the voltage sent towards a trapezial speed shifter. The angular speed control is the main objective of this investigation because it helps to synchronize the speed in the hybrid system.

II. EXPERIMENTAL DESIGN

For the control system design is necessary to define firstly the main components of the individual motors manipulation, secondly the classic control system, and lastly the architecture of the experimental data acquisition. The efficiency in the system depends on the positioning of the most important components for the speed transmission towards a mandrel of pulleys, consisting in two rowlocks that allow the angular bearing, generated by the connection of A-49 bands, towards the electrical motor and as well towards the combustin engine, the clutches are driven with a stage of power for the engineering specifications of the clutches, and controlled as well with an on/off control philosophy, being manipulated by an interface conduced in simulink of Matlab.(Fig. 1)

A. Electromagnetic clutchs

An electromagnetic clutches were used for power division by means of a Sanden SD 508 clutch, this clutch has double pulley (Fig. 2a) one of them is used to join the brushless motor with a mandrel pulley by means of a rubber band.

B. The architecture of the hibrid system

The efficiency in the system depends on the positioning of the most important components for the speed transmission towards an mandrel of pulleys, consisting in two rowlocks that allow the angular bearing, generated by the connection of A-49 bands, towards the electrical motor and as well towards the combustion engine, the clutches are driven with a stage of power for the clutches engineering specifications, and controlled as well with an on/off control philosophy (Fig. 1), being manipulated by an interface conduced in simulink of Matlab.

Fig. 1: Hybrid bench system with parallel architecture.

C. Calculation for angular speed

The feedback control of the speed (Fig. 3), was designed to analyzing the type of disturbance that generates an instability to the system, modifying the response to a setpoint and making a nonlinear correlation between input and output variables, as well as increasing the number of poles and zeros in the dynamic modeling of the system. An incremental encoder was used for the sampling the angular velocity, this encoder has 3 photo transmissive sensors that were adapted to the arrow of the electrical motor allowing to calculate the time (Fig. 2b), according on the number of slots in the encoder, the optical disk of encoder has 480 slots (Fig. 2c). Thus angular speed was calculated in the following way, equation (1).

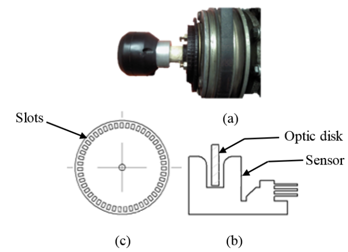

Fig. 2: Incremental encoder with optic disk.

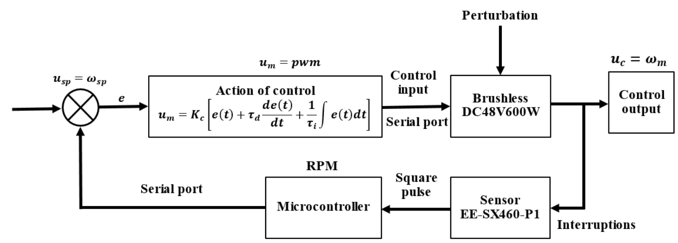

Fig. 3: Experimental setup for the angular velocity feedback control of the electric motor.

$$w_s = (c/480)/T \tag{1}$$

Where w_s is the angular speed (rpm), T is the time necessary for the encoder sampling each slot (min) and c is the number of detected slots.

D. Experimental setup

In order to apply the control action that allows to correct the error, a feedback control was implemented by an interface in Matlab Simulink tool environment, and a PID control action was used, equation (2).

$$u_m = K_c[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i}\int e(t)dt] \tag{2}$$

Where u_m is the manipulated variable in our case the electric motor voltage, K_c is the proportional controller gain, T_d is the derivative controller gain, T_i is the integral controller gain, $e(t)$ represents the signal error; that is the difference between the output variable and the setpoint. The proportional control gives an offset between the output variable and the setpoint.The integral action corrects the deviation in steady state of the actual error, this control action stays integrating until the error tends to zero. The derivative control action which allows to anticipate to the process output, therefore the PID control action has a rapidily change for error between the measure angular speed and the setpoint of the angular speed.

E. Data acquisition

In order to identify the transfer function that describes the dynamic behaviour of the brushless motor, a microcontroller′s interface of serial communication through reception of text chains in ASCII code was implemented. The sampling of the angular velocity of the motor arrow, the time sampled by the incremental encoder and the voltage of the speed variator. The interface was designed in the tool Matlab′s Simulink, using a Query Intrument block. The block doesn′t have input ports. The block has an output port corresponding to the data received from the acquisition instrument and receives the data in text chains in floating number of 64 bits.

F. Identification of the dynamic system

12 runs were performed to different inputs with values from 0 to 5 volts with the purpose of modeling the dynamic response of the motor. 1499 angular speed data were obtained. The system′s identification were made in the System Identification toolbox of Matlab, the modeling of the second order transfer function was evaluating by increasing the number of zeros in the modeled transfer functions at different samples, an experimental design by randomized complete blocks was developed and it was analyzed by applying ANOVA′s statistical analisys with a significance level of α = 0.05, this enables us to select the transfer function that bettter describes the behavior to the different input voltage to the system.

G. Root-Locus location of the characteristic equation

The root-locus plot were obtained using the Siso tool of Matlab, considering that when finding the transfer function that describes the motor′s dynamic behaviour, there is a time delay due to voltage transport between the microcontroller and the speed driver, therefore the response to the angular velocity is not instant. This speed driver sends a trapezial pulses to the electrical motor, alternating the current phases, through sensors of hall effect. The time′s delay was modeled using the Pade′s aproximation. The delay was modeled with a transfer function of second order for the system in open loop. Increasing the number of zeros and poles to the system. The gain factor asymptotes, from the Siso tool, and the root-locus branches were obtained from transfer function in open loop a criterium of angle and magnitude is consider:

$$GH = \frac{KN(p_1)}{D(p_1)} = -1 \tag{3}$$

$$GH = ArgGH(p_1) = 180 + 360l = (2l+1)\pi radians \tag{4}$$

Where $l = 0, 1, 2, ,.$ and $ArgGH$ is the argument of the transfer function in open loop. N and D = finite polynomials in complex variable s and K = gain factor

$$|GH(p_1)| = 1 \tag{5}$$

$$Arg\frac{N(p_1)}{D(p_1)} = \begin{cases} (2l+1)\pi radians, & k > 0 \\ 2l\pi radians, & k < 0 \end{cases} \tag{6}$$

Besides of this, the roots of the characteristic polynomial of the transfer function in open loop σ_c according to the formula:

$$\sigma_c = \frac{\sum_{i=1}^{n} p_i - \sum_{j=1}^{m} z_i}{n-m} \tag{7}$$

Where $-p_i$ are the poles, $-z_i$ are the zeros, n is the number of poles and m the number of zeros of the transfer function in open loop.

The angles between asintotes β are calculated using:

$$\beta = \begin{cases} \frac{(2l+1)180degrees}{n-m}, & k > 0 \\ \frac{(2l)180degrees}{n-m}, & k < 0 \end{cases} \tag{8}$$

for l= 0, 1, 2, ..., $n - m - 1$

And breaking point σ_b is calculated using:

$$\sum_{i=1}^{n} \frac{1}{\sigma_b + p_i} = \sum_{i=1}^{n} \frac{1}{\sigma_b + z_i} \tag{9}$$

The control architecture in this design is by feedback of the output signal (Fig. 4), through a gain modification we turn the numerical value of the output revolutions, to a voltage range, as well the control action that adjust the output system to the setpoint. No filter is used for the input voltage data.

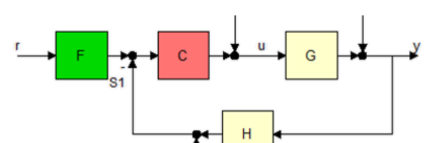

Fig. 4: Architecture of the control system by feedback, r=input, y= output, C=compensator, G=plant, H= feedback gain, u=compensator output, S1=adder.

H. Ziegler and Nichols Tuning Techniques

A Matlab script was programmed to observe the transitory response of the modeled transfer function in the closed loop mode, for this was necessary to know the inflection point of the control variable as a function of time (Fig. 5), with the aim of finding the values of L and T, that are the delay time and the time constant respectively, (Table I).

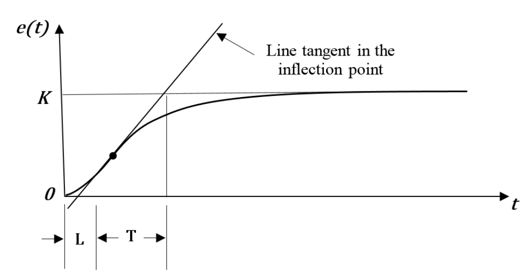

Fig. 5: Response curve in s form (Adaptated from [11]).

TABLE I: Ziegler-Nichols tuning rule based on the response of the plant′s step (Adaptated from [11])

TIPE OF CONTROLLER	K_p	T_i	T_d
P	T/L	∞	0
PI	0.9 * T/L	L/0.3	0
PID	1.2 * T/L	2*L	0.5 * L

I. Fuzzy Controller Description

The fuzzy controllers were designed using Publishing FIS of MatLab, with an Mamdani type architecture. The error signal, input to the fuzzy controllers, works on the basis of a reference voltage (setpoint). From the error signal the controller increases or diminishes the gain value, output from the fuzzy controllers. In the other hand fuzzification were made by the use of triangular membership functions.

J. Definition of the inputs, outputs and rules

The input linguistic variable, includes an universe of discourse in the range of -3 to 5 volts which inside of the error signal interval, and is composed by five membership functions (Fig. 6). Also is important to mention that the error signal were used a input for the fuzzy control and classical control $i.e.$ Ziegler-Nichols and Root-Locus.

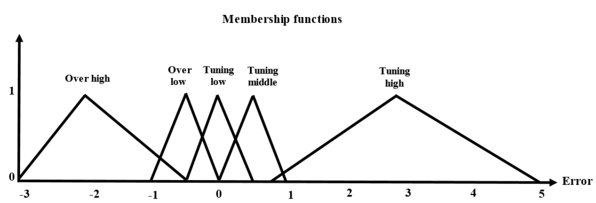

Fig. 6: Input linguistic variable.

The set of terms of the output linguistic variable applied to the root-locus method is constituted by four membership functions all of them distributed in the universe of discourse that goes from 0 to 4.499 of the gain value, (Fig. 7).

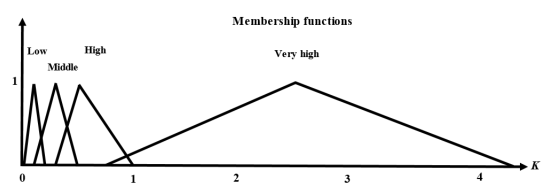

Fig. 7: Output linguistic variable.

The set of terms of the output linguistic variable applied to the Ziegler-Nichols method is constituted by four membership functions all of them distributed in other universe of discourse from 0 to 1, (Fig. 8).

When linguistic variables have been defined we can write the rules of the system, these rules are the fundamental part of the fuzzy control and they contain all the information necessary to find gain K. The control was conducted with 5 rules,

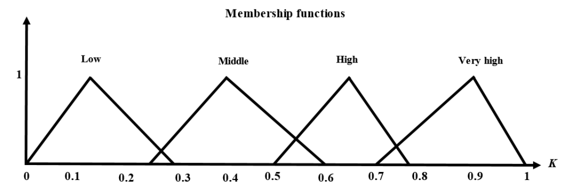

Fig. 8: output linguistic variable.

that were the same in both cases (Ziegler-Nichols and Roots-Locus). These rules relate the premises of the linguistic input variable to the premises of the linguistic output variable (Table II).

TABLE II: Base of rules

No.	IF	IS	THEN	IS
1	Input	Over high	Output	Low
2	Input	Over low	Output	Middle
3	Input	Tuning low	Output	Middle
4	Input	Tuning middle	Output	High
5	Input	Tuning High	Output	Very high

K. Block diagram of the fuzzy controller architecture

The block diagram implemented in simulink of the control system $i.e.$ classic controller, fuzzy-proportional and fuzzy-proportional-integral (Fig. 9). The fuzzy gain is autotuned using the methods previuosly mentioned.

III. RESULTS

The discretization was carried out by obtaining a reference voltage. The corresponding calibration of the sensor was carried out using an optical tachometer, to verify that if the obtained value fits is the output data with the calibrated tachometer, this was made simultaneously with the data acquisition of angular speed. The identification of the dynamic system was carried out for the 12 run tests in the brushless motor by using a completely randomized experimental design. The acquired data from each one test were simulated by fitting a transfer function for each one, then a set of new simulated data were generated for each test. Thus an ANOVA statistical analysis for all fitted functions were made searching the function that better correlates the experimental data. The statistical analysis was applied to the speed angular and the input voltage of the speed driver from brushless motor.

The eighth test experiment has a correlation of 99% in regard to the experimental data, because of this, it was decided to model a transfer function with the data of that test. The transfer function identified, equation (10), shows a good fit with respect to the linear regression of 96.41% and has a delay time of 0.05 seconds.

$$G(s) = \frac{rpm}{voltage} = e^{-0.05s} * \frac{-9.321s^2 + 649.7s + 711.5}{s^2 + 8.076s + 6.439} \quad (10)$$

The stability in close loop was determined from the poles′s place in open loop from complex plane s. If any of these

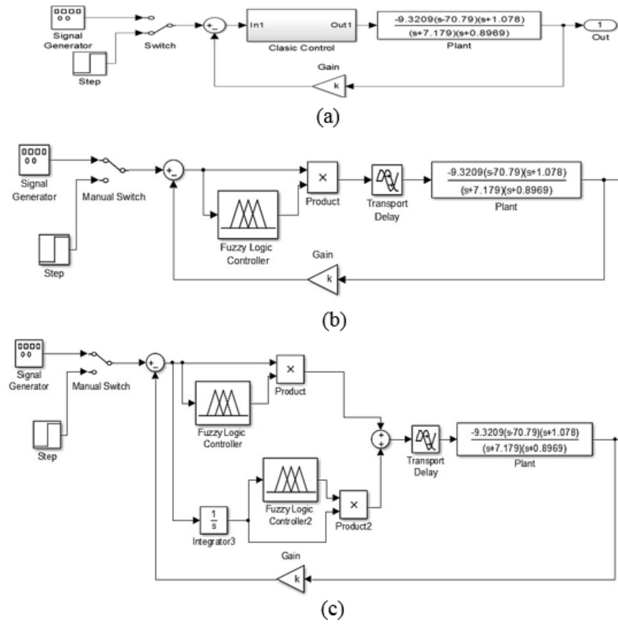

Fig. 9: Block diagrame: (a) classic controller, (b) fuzzy-proportional controller, (c) fuzzy-proportional-integral controller.

poles of the characteristic equation were in the right semiplane of the complex plane and they will be dominant, then the transitory response in time will increase in monotonic form or will oscillate with an increasing amplitude, leading system to an unstable regin. Nevertheless this transfer function, equation (11), the poles of the characteristic equation are placed in the left semiplane.

Pade′s aproximation for the time delay of order 2 was modeled for the system in close loop, and the root-locus plot of this new characteristic equation was obtained acording to equation (11).

$$G'(s) = \frac{-9.3209(s-70.79)(s+1.078)(s^2-120s+4800)}{(s-7.179)(s+0.8969)(s^2-120s+4800)} \tag{11}$$

A. *Design of the proportional control using Roots-Locus technique*

In the proportional control design was made by trail and error, searching for the gain values were the system doesn′t show an unstable transitory response to a disturbance of step type. The roots-locus plot in open loop for a compensator (gain factor) =0.9286 is shown in Fig. 10, using an Pade′s aproximation of 2 order for the time delay.

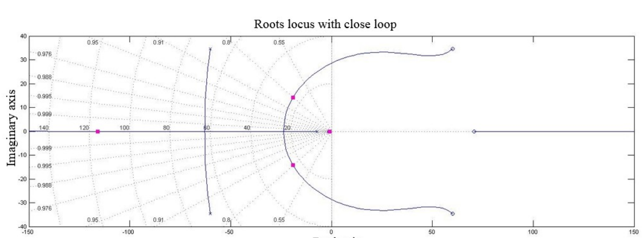

Fig. 10: Root-locus plot of the transfer function in open loop, proportional control.

The system response to the proportional control in steady state shows an offset and the output signal, angular velocity, doesn′t reach the setpoint, nevertheless the response doesn′t show an overshoot and has a rise time of 0.128 seconds and a setting time of 1.44 seconds in a simulation time of 10 seconds.

B. *Design of the Integral Proportional Controller using Roots-Locus technique*

The PI controller is an alternative to correct the error in steady state arised by the proportional action control. In this case a proportional controller tuned through the roots-locus plot placing the gain in a stable point within the left semiplane of complex plane s, as it is shown in Fig. 11. The roots-locus plot in open loop for a compensator (factor of gain) = 1.073 and an integral gain = 1.073 had a satisfactory result to fit the signal from the output to the input signal.

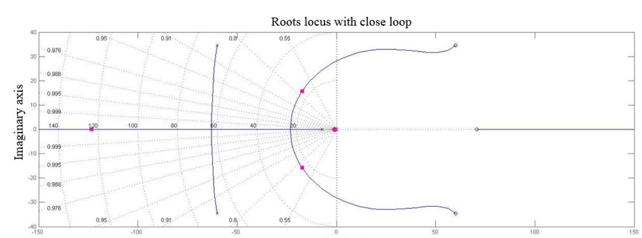

Fig. 11: Root-locus plot of the transfer function in open loop, proportional-integral control.

The system response shows a rise time of 2.68 seconds and a setting time of 5.23 seconds, higher values than those of a proportional controller.

C. Design of the feedback controller system using Ziegler and Nichols's techniques for proportional and integral tuning of gains

Using the Ziegler-Nichols rules the proportional and integral gains were tuned (Fig. 12), founding a values of L =0.0633 and T =0.1574 in open loop mode.

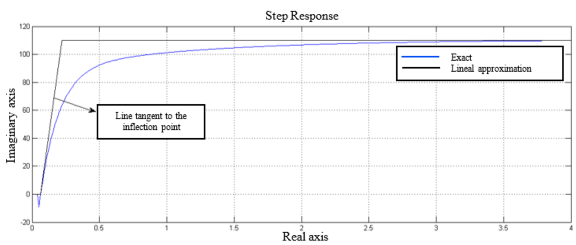

Fig. 12: Line tangent to the inflection point.

Using the rules of Ziegler and Nichols gives an proportional gain of:

$$K_p = \frac{T}{L} = \frac{0.1574}{0.0633} = 2.4865 \tag{12}$$

and a integral and derivative gains of:

$$K_p = 0.9 * \frac{T}{L} = 0.9 * \frac{0.1574}{0.0633} = 2.2381 \tag{13}$$

$$K_i = \frac{K_p}{L/0.3} = \frac{K_p}{0.0633/0.3} = 10.6053 \tag{14}$$

D. Tuning of speed and comparison of the classic control system with fuzzy classic

Two comparative simulations between the method of Ziegler-Nichols and the Root-Locus were carried out, the obtained response are in Fig. 13. and Fig. 14. we can observe that a better fit to the set point tan those obtained from the fuzzy gains K.

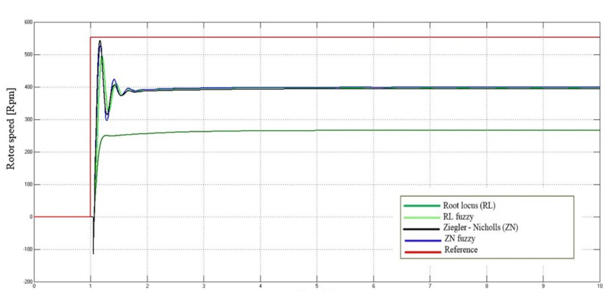

Fig. 13: Autotuning of the proportional control.

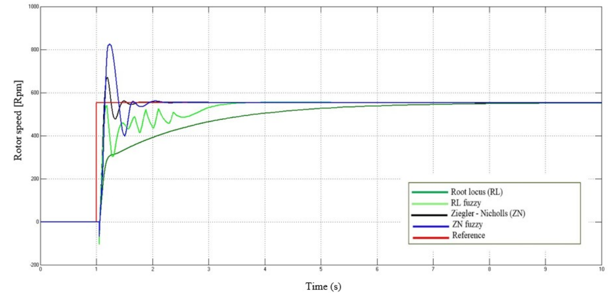

Fig. 14: Autotuning of the proportional-integral control.

IV. CONCLUSIONS

Results shows that a fuzzy controller coupled with a classic PI controller applied to control the angular speed of the hybrid system has the advantage to autotune in real time the gains for the control action, rise and setting time are diminished compared with the sole use of Root-Locus and Ziegler-Nichols methodologies. The proportional control action in the feedback system has a deviation from the setpoint, thus we choose a PI controller action to correct the error in steady state to obtain a stable transient response. As a result of the implementation of coupled control methodology, the brushless motor acted efficiently to increase performance and control the angular velocity.

This investigation is a first step in the coordination between fuzzy control and PI control strategies using a single universe of discourse from the linguistic variables, in order to obtain a better prediction in the control variable. Also is necessary test another modes of the dynamic behavior of the bench hybrid system.

REFERENCES

[1] Sivanandam, Sumathi, & Deepa., *Introduction to Fuzzy Logic in MATLAB*. Springer, First edition, 2007.

[2] Wang, H., "Design and Implementation of Brushless DC Motor Drive and Control System." *Procedia Engineering*, pp. 2219—2224, 05, 2012.

[3] Cox, E., "Fuzzy Fundamentals." *Spectrum Magazine of IEEE*, vol. 106, pp. 58–61, 1992.

[4] Gunes, M., & Dogru, N., "Fuzzy Control of Brushless Excitation System for Steam Turbogenerators" *IEEE Transactions on Energy Conversion,*, pp. 844–852, 08, 2010.

[5] K. H. , A., Chong, G., & Li, Y., "PID control system analysis, design, and technology" *IEEE Trans. Control Syst. Technol*,pp. 5559–5576, 2005.

[6] Rubaai, A., Castro-Sitiriche, M. J., & R. Ofoli, A., "Design and Implementation of Parallel Fuzzy PID Controller for High-Performance Brushless Motor Drives: An Integrated Environment for Rapid Control Prototyping.," *IEEE Transactions on Industry applications*, pp. 1090–1098, 2008.

[7] Li, Y., Ang, K., & Chong, G. , "Patents, software, and hardware for PID control" *IEEE Control Syst. Mag*, pp. 42–54, 2006,2013.

[8] Weirong, J., Haibo, H., & Jianping, L. "Simulation and Design of Integral Separation Adaptive Fuzzy Control System for Brushless DC Motor" *International Conference on Computational and Information sciences*, pp. 1194–1197,2013.

[9] Xia, C., Guo, P., Shi, T., & Wang, M. "Speed Control of Brushless DC Motor Using Genetic Algorithm Based Fuzzy Controller" *International Conference on Intelligent Mechatronics and Automation IEEE*, pp. 460–464, 2004.

[10] Ruey Yu, G., & Che Hwang, R. "Optimal PID Speed Control of Brushless DC Motors Using LQR Approach" *International Conference on Systems, Man and Cibernetics*, pp. 473–478, 2004.

[11] K.Ogata, *Modern Control Engineering*. Pearson Education, MA: Prentice Hall, Fifth edition, 2010.