IMPLEMENTATION OF AN INTERPRETABLE INVERSE FUZZY CONTROL ON MICROCONTROLLER (DC MOTOR CASE STUDY)

Juan Contreras[1, 2], Alcides Ramos[1], Shirley Urdaneta[2] and Luis Murillo[3]
[1]Faculty of Naval Engineering, Escuela Naval Almirante Padilla, Colombia
[2]Institución Universitaria ITSA, Colombia
[3]Faculty of Electronic Engineering, Universidad Antonio Nariño, Colombia
E-Mail: epcontrerasj@ieee.org

ABSTRACT

This paper describes the design and implementation of an interpretable inverse fuzzy control for the purpose of controlling permanent magnet DC (PMDC) motor speed. The design of fuzzy controller has been done using MATLAB and Simulink and implemented on an Arduino Uno board. The antecedent fuzzy sets are defined by triangular membership function with 0.5 interpolations avoiding the presence of complex overlapping that happens in other methods. The real-world applicability of the proposed approach is demonstrated by application to control the speed of the motor at a desired value with the possibility of changing it.

Keywords: arduino, interpretability, inverse fuzzy control, DC motor speed.

1. INTRODUCTION

Automation is used to improve productivity and obtain high-quality products. The following are some advantages of implementing automation in large, medium and small companies: reduction of the production process duration, reduction of costs, facilitation of procedures, increase in production capacity, reduction of the needed space and others [1]. Automation enables a level of accuracy and productivity that a human could not achieve [2].

A Proportional-Integral-Derivative (PID) controller is a control loop feedback mechanism widely used in industrial control systems. They are used in most automatic process control applications in industry [3]. However, fuzzy control is gaining extensively more acceptance within industry [4] because it is applicable to plants that are mathematically poorly understood and where the experienced human operators are available [5]. When a system is too complex or too poorly understood to be described in precise mathematical terms, fuzzy modelling provides the ability to linguistically specify approximate relationships between the input and the desired output [6].

The brushless DC motors are widely used for industrial applications such as automotive, medical, robotics, etc. based on inherent advantages: high start torque characteristic, high response performance, high efficiency and can be easily controlled in a wide speed range [7]. PID Control has proven to be good enough to handle control tasks on system control but its implementation relies on an exact mathematical model of the plant to be controlled and not simple mathematical operations. An alternative method for the speed control of a DC motor is based on fuzzy controller, which is based on heuristic knowledge instead of using a complex and high accuracy system model [5].

The construction of fuzzy models demands the selection and tuning of many parameters as the shape and distribution of membership functions of input variables, rule base, logic operators, shape and distribution of consequents, etc. The great number of parameters has obstructed the development of a unique technique for the optimal tuning of the parameters of fuzzy controllers, especially in the case of fuzzy identification from experimental input and output data [8].

The fuzzy identification algorithm employed in this work uses triangular membership function with 0.5 interpolations for antecedent partition avoiding the presence of complex overlapping that happens in other methods [9]. This approach guarantees the completeness of the rule base and avoids the exponential growth of the rule base as the number of inputs increases that occurs with classical methods. It is because the proposed method generates n x j rules, where n is the number of input variables and j is the number of membership functions for each input variable. Each fuzzy region is covered with a fuzzy rule and there is no redundancy in the rule base.

The outline of this paper is as follows: In section 2 the structure of the fuzzy model proposed is presented. Section 3 describes a brief overview about the process to be controlled. Section 4 presents the fuzzy model obtained from experimental data. A brief description about the method employed for obtaining interpretable fuzzy models from data is included. Section 5 presents the steps for obtaining the inverse fuzzy controller ir order to control the speed of a dc motor. Section 6 shows the experimental results. Finally, some conclusions are drawn.

2. FUZZY MODEL STRUCTURE

2.1 Shape and distribution of input and output membership functions

For each input variable an uniform partition is built using normalized triangular sets with an overlap of 0.5 between two successive fuzzy sets and there will be two triangular membership functions with their modal

VOL. 13, NO. 9, MAY 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

values placed, respectively, in the minimum and the maximum of the universe of discourse. This fuzzy partition is considered a Strong Fuzzy Partition (SFP) because it satisfies the following semantic constraints [10]: distinguishability; overlapping in 0.5; coverage; normality; convexity and the number of fuzzy set is no upper than 9.

The main advantage of using triangular membership functions is that they allow the reconstruction of the linguistic value at the same numeric value after a defuzzyfication method has been applied [11].

For each triangular membership function of each input variable a singleton consequent is generated. The singleton fuzzy terms are estimated from data using recursive least-squares techniques. There will be so many rules as singleton consequents.

2.2 Operations on input membership functions

No calculations of fuzzy intersection between input membership functions are needed in this method.

2.3 Inference method

There is a rule for each input triangular membership function in the form:

$$If\ u_i\ is\ A_i^j\ Then\ y_{ij}\ is\ {}_{ij} \quad (1)$$

where

A_i^j represents the jth linguistic value of the linguistic variable u_i defined over the universe of discourse U_i and ${}_{ij}$ is the singleton associated to the linguistic value A_i^j.

The output of the fuzzy system is given by

$$Y = W\ {}_{ij} \quad (2)$$

where W is represented by the following vector

$$W = [u_{A_1^1}(x_1^k)\, u_{A_1^2}(x_1^k) \ldots u_{A_1^j}(x_1^k)$$
$$u_{A_2^1}(x_2^k)\, u_{A_2^2}(x_2^k) \ldots u_{A_2^j}(x_2^k) \ldots \quad (3)$$
$$u_{A_n^1}(x_n^k)\, u_{A_n^2}(x_n^k) \ldots u_{A_n^j}(x_n^k)]$$

x_n^k is the n-dimensional input array $x_1^k, x_2^k, \ldots, x_n^k$, n represents the n-th input variable and j represents the number of membership function for each input variable. The singleton consequent parameters ${}_{ij}$ are obtained from data using recursive least-squares technique.

3. SYSTEM DESCRIPTION

A permanent magnet DC motor coupled with a tachometer, flywheel, and a load motor is utilized.

Figure-1. DC motor plant.

The hardware has been implemented on an Arduino Uno board in order to gather input and output data pairs for system identification (Figure-2). The input data is the voltaje applied to the dc motor and the output data is the tachometer voltaje correspondig to the speed of the dc motor. Data was acquired for the unloaded spin up of the motor.

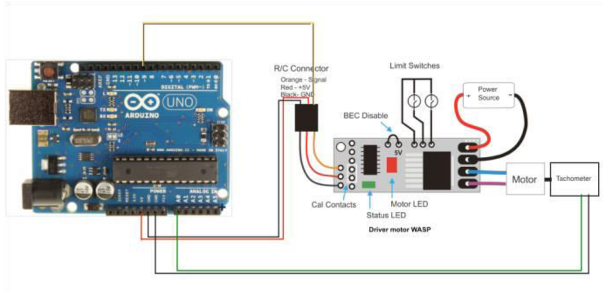

Figure-2. Hardware.

The first step is to find the working región (and the linearity of the system). It means the largest range of the input signal that ensures a valid linear approximation. The center of the working región is called the working point. In order to find the working region several step response experiments has been performed. The Arduino uses a 10-bit ADC. It means that the ADC assumes 5 dc volts is 1023.

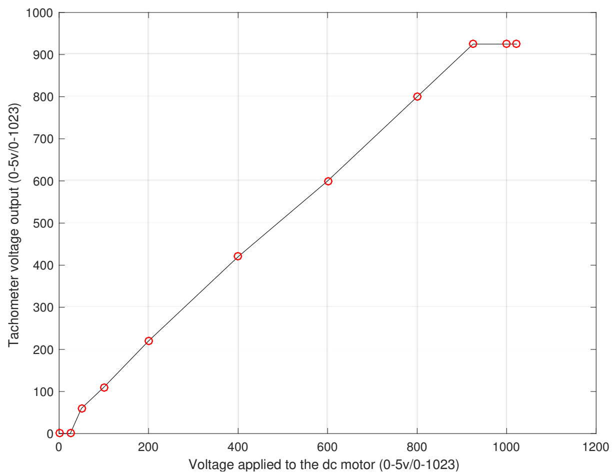

Figure-3. Linearity test: Step response test

The process is approximately linear for input signals within the interval [500, 900]. This is the working región.

VOL. 13, NO. 9, MAY 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

4. IDENTIFICATION OF FUZZY MODEL

For the fuzzy identification process, the following input and output data were used (Figure-4)

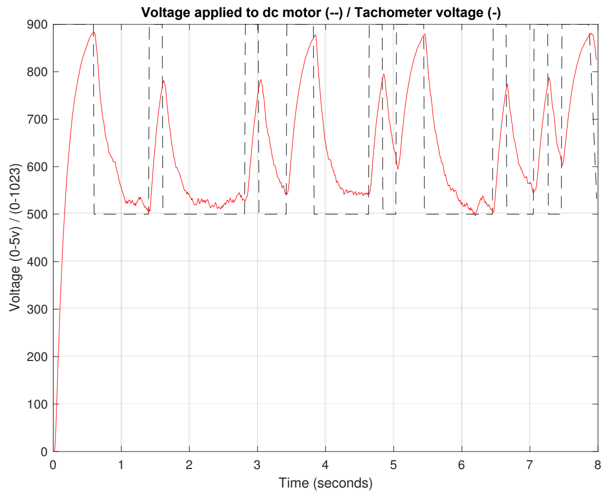

Figure-4. Data for identification process.

The fuzzy model obtained uses three input variables: the past outputs y(k - 1), y(k - 2) and the current input u(k) to predict the output y(k). The membership functions for each input variable are shown in Figure-5.

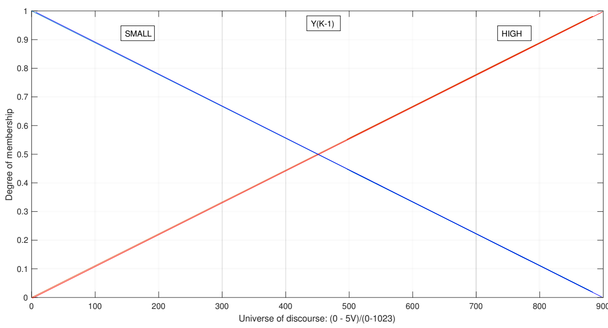

Figure-5(a). Membership functions for y(k-1)

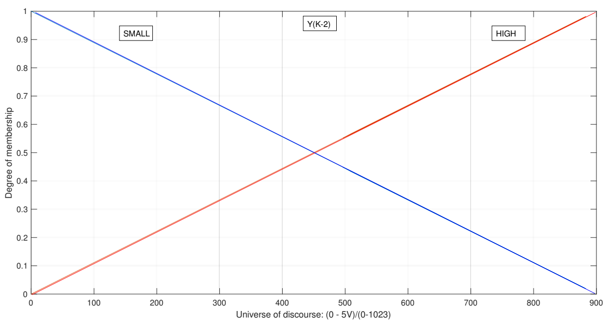

Figure-5(b). Membership functions for y(k-2).

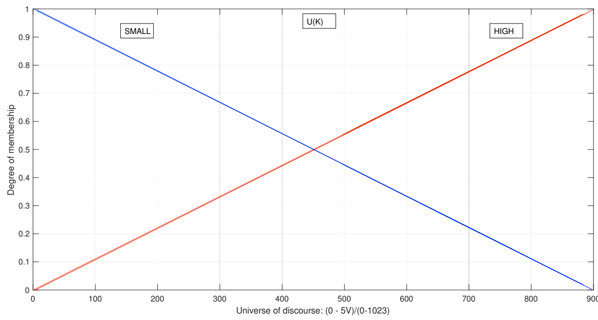

Figure-5(c). Membership functions for u(k)

Then

$$W=[u_1(y(k\ \ 1))\,u_2(y(k\ \ 1))\,u_1(y(k\ \ 2))\,u_2(y(k\ \ 2))\quad u_1(x(k))\,u_2(x(k))] \qquad (4)$$

$$= \quad {}_1 \quad {}_2 \quad {}_3 \quad {}_4 \quad {}_5 \quad {}_6 \quad {}^T \qquad (5)$$

The consequent parameter vector is obtained by least square technique

$$(k) = \ (k\ \ 1) + P(k)W(k)e(k)] \ (6)$$

where

$$e(k) = y(k)\ \ f(x(k)) = y(k)\ \ w(k)\ \ (k) (7)$$

$$P(k) = \frac{P(k\ \ 1)}{}\ I\ \ \frac{w(k)w(k)^T P(k\ \ 1)}{w(k)^T P(k\ \ 1)w(k)} \ (8)$$

At the beginning, the covariance matrix P(0) = identity matrix.   is the forgetting factor (0< <1).

The consequent parameter thus obtained is given by

$$= \begin{matrix} -84.8672 \\ 384.1563 \\ -42.1926 \\ 341.4817 \\ 121.9386 \\ 177.3505 \end{matrix}$$

The rule base can be expressed in the form:

IF u(y(k 1)) *is Small THEN Output is* -84.8672
IF u(y(k 1)) *is High THEN Output is* 384.1563
IF u(y(k 2)) *is Small THEN Output is* -42.1926
IF u(y(k 2)) *is High THEN Output is* 341.4817
IF u(x(k)) *is Small THEN Output is* 121.9386
IF u(x(k)) *is High THEN Output is* 177.3505

Once the fuzzy model receives input the rule base is evaluated using equation (2).

ARPN Journal of Engineering and Applied Sciences

The results of identification and validation process are shown in Figures 6 and 7, with a normalized root mean square error (NRMSE) of 0.000471 and 0.0006104 respectively.

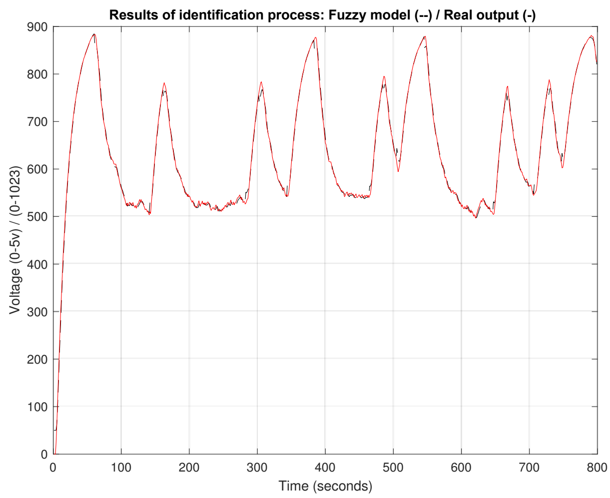

Figure-6. Results of identification process.

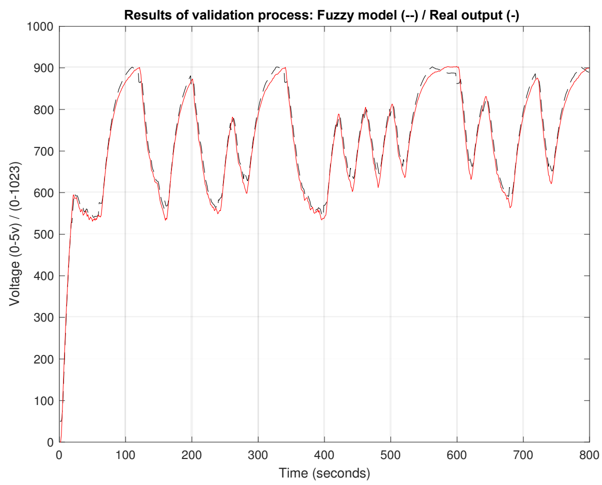

Figure-7. Results of validation process.

5. INVERSE FUZZY CONTROL

Yr is the reference model and can be expressed as

$$Y_r = W_1 \;{}_1 + W_2 \;{}_2 + W_3 \;{}_3 \quad (4)$$

where

$$W_1 = u_1(y(k\ 1))\, u_2(y(k\ 1))\ ; \quad {}_1 = {}_1 \quad {}_2^{\ T}$$
$$W_2 = u_1(y(k\ 2))\, u_2(y(k\ 2))\ ; \quad {}_2 = {}_3 \quad {}_4^{\ T}$$
$$W_3 = u_1(x(k))\, u_2(x(k))\ ; \quad {}_3 = {}_5 \quad {}_6^{\ T} \quad (5)$$

The vectors $u_1(y(k\ 1))$, $u_2(y(k\ 1))$, $u_1(y(k\ 2))$, $u_2(y(k\ 2))$, $u_1(x(k))$ y $u_2(x(k))$ contain the membership degree of the partitions of each variable: y(k-1), y(k-2) y u(k) respectively. The first four vectors and Y_r are known. The objective is to find the input signal that produce Y_r. It is obtained from

$$W_3 \;{}_3 = Y_r \quad W_1 \;{}_1 + W_2 \;{}_2 \quad (6)$$

$$u_1(x(k)) = \frac{Yr \quad u_1(y(k\ 1))\, u_2(y(k\ 1))\, u_1(y(k\ 2))\, u_2(y(k\ 2)) \quad {}_{1\ 2\ 3\ 4}\ {}_6}{{}_5 \quad {}_6} \quad (7)$$

$$u_2(x(k)) = 1 \quad u_1(x(k)) \quad (8)$$

6. RESULTS

The inverse fuzzy controller was implemented on Arduino Uno. A server program is installed on Arduino board for passing information between the hardware and the host computer running a Simulink/MATLAB model (inverse fuzzy controller) as shown in Figure-8. The inverse fuzzy controller is implemented using the function blocks supported in Simulink.

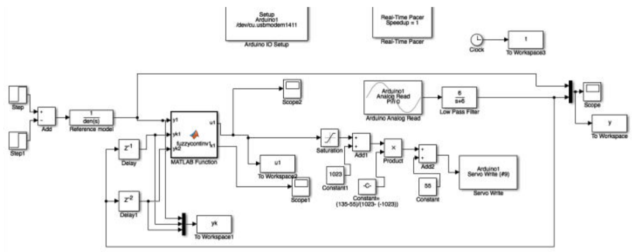

Figure-8. Inverse fuzzy controller running in Simulink.

In the first experiment the set point (step) was moved from 0 to 500 and from 500 to 650 and from 650 to 800 (measure in volts using adc converter of 10 bits). The response of the process variable is shown in Figure-9.

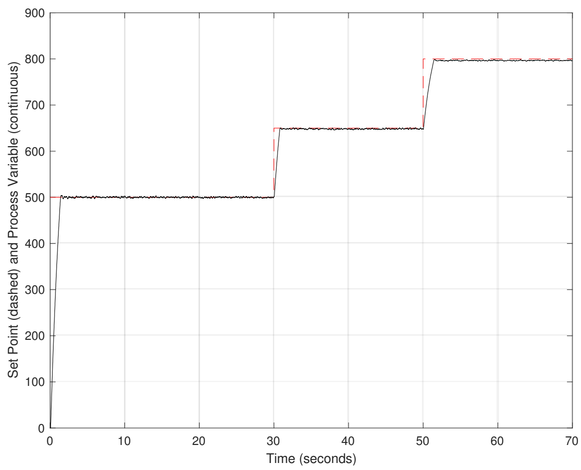

Figure-9. Response of the system. Experiment 1.

VOL. 13, NO. 9, MAY 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

In the second experiment the set point (step) was moved from 0 to 500 and from 500 to 750 and from 750 to 600. The response of the process variable is shown in Figure-10.

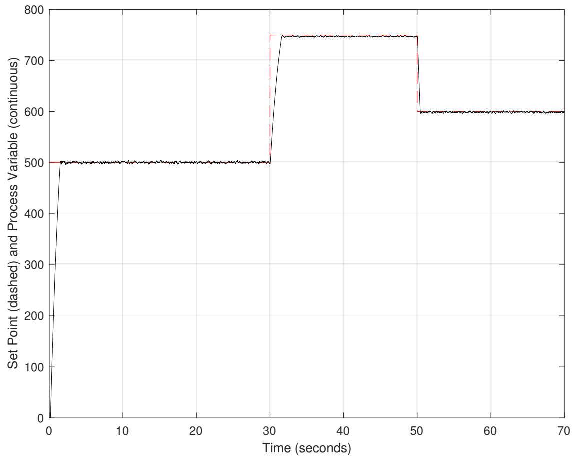

Figure-10. Response of the system. Experiment 2.

Finally, in the third experiment a Transfer Funciton Block is included in Simulink in order to modify the shape of the set point (see Figure-8). The response of the process variable is shown in Figure-11.

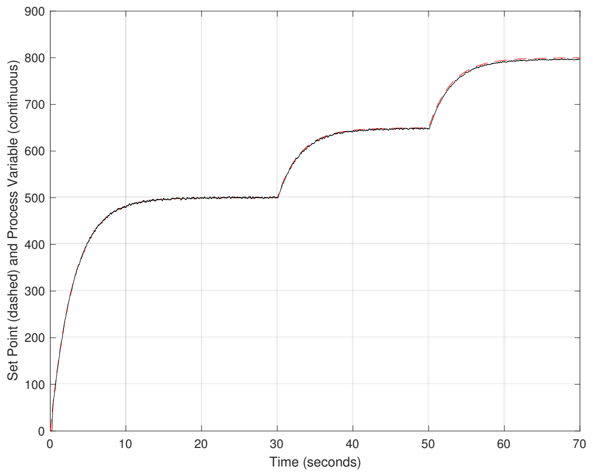

Figure-11. Response of the system. Experiment 3.

The experimental results demonstrated that the inverse fuzzy controller provides a good transient response without overshoot.

7. CONCLUSIONS

This paper describes how to design and implement an inverse fuzzy controller on microcontroller.

A fuzzy inverse controller has been employed for speed control of dc motor by Simulink/Matlab interfaced with Arduino Uno. The expermental results under varying reference speed show a good performance in both transient and steady state response, without overshoot.

Experiments have shown that Arduino is quite flexible for programming a control algorithms using Simulink/Matlab.

REFERENCES

[1] Jahan S.H.M., Najafi S.E., Kashan A.H., Bahadoran N. 2013. Proposing a Model for Production Agility Measurement, Using Industrial Automation Effectiveness, with Case Study in Amol School Facilities Company. International Journal of Application or Innovation in Engineering & Management. 2(3): 242-250.

[2] Taylor G. S., Reinerman-Jones L. E., Szalma J. L., Mouloua M. & Hancock P. A. 2013. What to Automate: Addressing the Multidimensionality of Cognitive Resources Through System Design. Journal of Cognitive Engineering and Decision Making. 7(4): 311-329.

[3] Torres W.L., Araujo I. B. Q., Menezes Filho J. B., Costa Junior A. G. 2017. Mathematical Modeling and PID Controller Parameter Tuning in a Didactic Thermal Plant. IEEE Latin America Transactions. 15(7): 1250-1256.

[4] Farana R., Walek B., Janosek M., Zacek J. 2015. Advantages of fuzzy control application in fast and sensitive technological processes. International Journal of Electrical and Computer Engineering. 9(11): 1270-1274.

[5] Ozturk N. 2012. Speed Control for DC Motor Drive based on Fuzzy and Genetic PI Controller - A Comparative Study. Elektronika ir Elektrotechnika. 123(7): 43-48.

[6] Sudkamp T., Knap A., Knap J. 2003. Model Generation by Domain Refinement and Rule Reduction. IEEE Transactions System, Man and Cybernetics. 33(1): 45-55.

[7] Agrawal P, Dubey S.P., Bharti S. 2014. Comparative Study of Fuzzy Logic Based Speed Control of Multilevel Inverter fed Brushless DC Motor Drive. International Journal of Power Electronics and Drive System. 4(1): 70-80.

[8] Naba A., Miyashita K. 2005. Tuning Fuzzy Controller Using Approximated Evaluation Function. Soft Computing as Transdisciplinary Science and Technology. Advances in Soft Computing. 29: 113-122. Springer, Berlin, Heidelberg.

[9] Ramos A., Contreras J. 2015. Adaptive Fuzzy Autopilot for a Scale Ship Model. Applied Mechanics and Materials. 798: 249-255.

[10] Alonso J.M. 2007. Interpretable Fuzzy Systems Modeling with Cooperation between Expert and Induced Knowledge, PhD Thesis. Universidad Politécnica de Madrid, Spain.

[11] Paiva R.P., Dourado A. 2004. Interpretability and Learning in Neuro-Fuzzy Systems, Fuzzy Sets and System. 147(1): 17-38.