Volume 58, Number 2, 2017

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

DEVELOPMENT OF A SPEED CONTROL SYSTEM FOR A DC MOTOR USING A FUZZY LOGIC CONTROLER

Gabriel OLTEAN, Laura-Nicoleta IVANCIU, Horea Alin BALEA,
Bases of Electronics Department, Technical University of Cluj-Napoca, Romania
26-28 G. Baritiu Street, Cluj-Napoca, Romania, phone:+40-264-401-471, laura.ivanciu@bel.utcluj.ro

Abstract: **The aim of this work is to implement an embedded system for the speed control of a DC motor, using a microcontroller-based development board. The control system is developed around an Arduino Mega board and involves a fuzzy logic system as controller. The implementation offers an easy understanding of the main concepts regarding closed loop systems and embedded systems, operating in real environment. The main achievements of this implementation can be summarized as follows: access for data in relevant points of the system, illustration of the use of some fundamental concepts in modern electronics (magnetic hall effect sensor for RPM measurement, C++ programming, implementation of a fuzzy logic controller), possibility to explore different operating scenarios. The experimental results prove the expected operation of our speed control system.**

Keywords: *fuzzy system, speed control, Arduino, embedded system*

I. INTRODUCTION

In the field of control system, fuzzy logic is a very popular solution, mainly because the process of fuzzy logic control is simply to put the realization of human control strategy, especially in the situations where conventional control heavily relays on appropriate mathematical model.

In the literature, there are a lot of approaches referring to the utilization of fuzzy logic system to implement control solutions in the framework of embedded systems. A system whose principal function is not computational but is controlled by a computational system embedded within it is referred to as an embedded system [1].

An adaptive fuzzy controller where the scaling factors of the fuzzy sets are adapted in real time based on a reinforcement Q-learning algorithm is implemented on an Arduino DUE board to control a DC motor with flexible shaft [2]. The design and implementation of a fuzzy logic controller operating in real time, for speed control of a dc motor, using an Arduino Due board is presented in [3]. Different temperature control systems implemented on microcontrollers are discussed in [4], [5], and [6].

The purpose of this implementation is to offer to students or novice designers a platform for understanding, experimenting and learning the main concepts of both embedded system and closed loop control system, in a real setup. The following aspects were achieved: access to read data in relevant points of the system, illustration of some fundamental devices and concepts in modern electronics (magnetic Hall effect sensor for RPM measurement, C++ programming, implementation of a fuzzy logic system, utilization of an H-Bridge, driving a DC Motor), and the possibility to create and test different operating scenarios (by changing the set point for the speed, by modifying the fuzzy system behavior - acting of the scaling factors, or by introducing a course adjustment of the control signal apart from the fuzzy logic system).

The paper is organized as follows: Section II gives the overview of the proposed system, Section III is dedicated to the presentation of the system implementation, Section IV presents and discuss some experimental results, while Section V concludes the paper.

II. SYSTEM OVERVIEW

Figure 1 presents the architecture of the proposed system. The controlled process consists of a brushed DC micro-metal 30:1 gear motor with long-life carbon brushes [7] whose speed (revolutions per minute) should follow a user specified speed profile RPM_{ref}. The current RPM is measured using a quadrature encoder, based on a magnetic disc and Hall effect sensors, which provides 12 counts per revolution of the motor shaft.

The DC motor is driven by a H-Bridge based on the dedicated L298 integrated circuit, which supplies the DC motor with a PWM voltage, whose magnitude is 9V in our design (battery). To its turn, the H-Bridge receives its control signal form the Arduino board.

The Arduino Mega board [8] is the “brain” of the entire system. It is responsible for the update of the digital control signal u, at every time instance. In each time instant, the current RPM value, RPM_k is read and the RPM error (err_k) and change of RPM error ($cerr_k$) are updated, as follows:

$$\begin{aligned} err_k &= RPM_k - RPM_{ref} \\ cerr_k &= err_k - err_{k-1} \end{aligned} \tag{1}$$

Manuscript received June 5, 2017; revised June 30, 2017

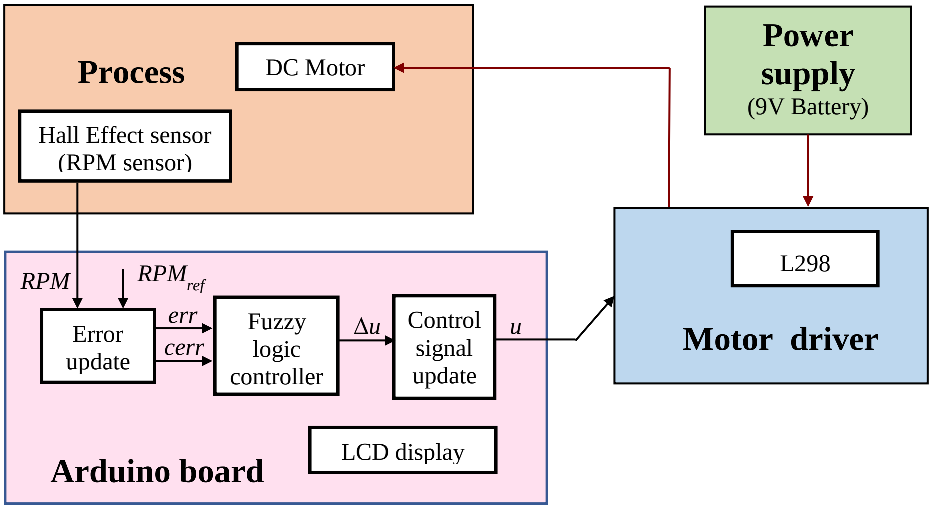

Figure 1. Block diagram of the proposed system.

where err_{k-1} is the *RPM* error in the previous time instance.

The central point of the entire system is the fuzzy logic controller, whose job is to infer the most appropriate modification in the control signal, in every time instance Δu_k. The digital control signal u_k is then updated using the relation:

$$u_k = u_{k-1} + \Delta u_k \qquad (2)$$

III. IMPLEMENTATION

Our system contains two main parts: the power circuit and the control circuit. The power circuit is represented by the motor driver and the brushed DC motor. On the other hand, the Arduino Mega board represents the control unit of the system.

The average voltage at the output of the H-Bridge is set by the Arduino board via the digital control signal *u* that lays in the [0; 255] range. The Arduino board transmits its control signal to the H-Bridge via a digital PWM output pin. The maximum value of the control signal (255) generates the maximum value of the duty cycle of the PWM signal leading to the maximum average voltage for the DC motor, namely maximum *RPM*. A smaller value of the control signal generates a smaller duty cycle, leading to a smaller *RPM*.

A. The Control Circuit

To measure the actual *RPM*, we are using the method based on a fixed time interval to count the revolutions of the main motor shaft. A counter is triggered at the initial time t_i and it counts the pulses received from the Hall effect sensor up to a final time t_f. The *RPM* is computed using the relation:

$$RPM = 60 \times \frac{C_f - C_i}{t_f - t_i}\,\frac{1}{C_r}\,\frac{1}{G_r} \qquad (3)$$

where:

- C_f - is the final value of the counter
- C_i - is the initial value of the counter
- $C_r = 12$, is the number of counts per revolution
- $G_r = 30$, is the gear ratio (30:1)

For the speed measurement, the quadrature encoder is mounted on the extended shaft motor, not on the gearbox output shaft.

The control circuit is presented in Fig. 2. It is developed around the fuzzy logic system with two inputs (*errFls* and *cerrFls*) and one output (Δ*uFls*). The range of value for all these three variables is the normalized one [-1; 1]. The detailed operation for the fuzzy logic system is presented in the next section.

The *RPM* error (*err*) and change in *RPM* error (*cerr*) are computed according to relation (1). To assure the flexibility of the control system, two scaling factors *Se* (for *err*) and *Sc* (for *cerr*) were introduced. By means of the scaling factor, the user can easily adjust the behavior of the control system, the fuzzy logic system being more or less sensitive to each input. To keep the input values of the fuzzy system in their ranges, a saturation block is used for each input having (+1) as the upper bound and (-1) as the

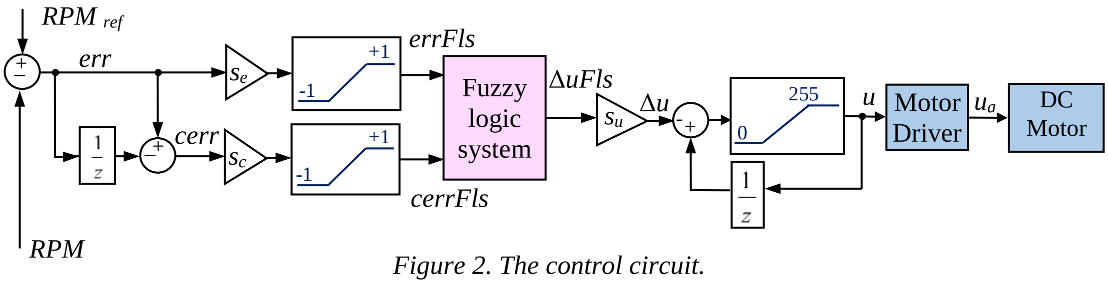

Figure 2. The control circuit.

lower bound. When the signal is within the accepted range it passes through unchanged. When the signal goes outside these bounds, it is clipped off.

The fuzzy logic system generates the $\Delta uFls$ signal that is further multiplied with the scaling factor *Su* to obtain the final modification of the control signal, Δu. The value of the control signal, *u* is then computed according with equation (2). The range of variation for the digital control signal is limited between 0 and 255 by the output saturation block.

B. The Fuzzy Logic System

The fuzzy logic system was implemented on the Arduino board using the open-source Arduino Integrated Development Environment (IDE) [9]. The fuzzy system is a Takagi-Sugeno one, with two inputs *errFls* and *cerrFls* and one output $\Delta uFls$. The universe of discourse is [-1; 1] for all variables. The fuzzy sets for both inputs are triangular (see Figure 3). For the output, there are three singleton fuzzy sets: *N* (with -1 as its support), *Z* (with 0 as its support), and *P* (with +1 as its support).

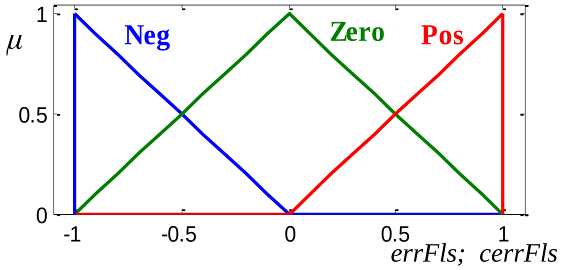

Figure 3. Fuzzy sets for inputs.

The rule base of the fuzzy system contains 9 fuzzy rules, presented in Table 1.

Table 1. The fuzzy rules table

cerrFls \ errFls	Neg	Zero	Pos
Neg	N	N	Z
Zero	N	Z	P
Pos	Z	P	P

The defuzzification method is the weighted average method. The control surface, that illustrate the operation of the fuzzy system is presented in Fig. 4.

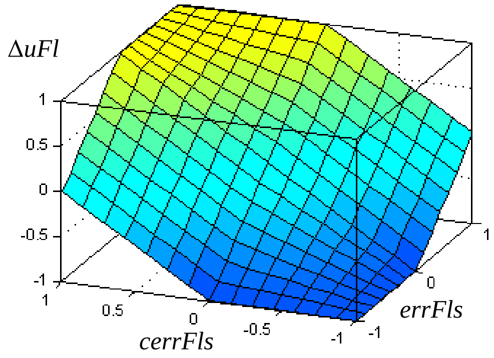

Figure 4. Control surface of the fuzzy system.

Despite of its simplicity (only the 9 standard rules) the fuzzy system performs very well, inferring the right output for any combination of inputs. To change the behavior of the control system, no intervention should be made on the fuzzy sets, but only on the scaling factors *Se*, *Sc*, or *Su*.

IV. RESULTS

The operation of the control system was tested on a real-time setup for a speed profile specified by the user.

Figure 5 presents the real response of the system (current speed) for a set point of 1000 rpm until t_1=20s, 500 rpm until t_2=40s, 750 rpm until T_3=80s, and 0 rpm afterwards. As expected, the system response is a typical one, presenting a critically damping response, that provides the quickest approach to the set point. The main parameters characterizing the system response are:

$t \in [0;\ 20\text{s}]$, speed from 0 to 1000 rpm
- rise time = 8.8 s;
- max. positive error = 5 rpm ;
- max. negative error = 5 rpm;

$t \in [20\text{s};\ 40\text{s}]$, speed from 1000 to 500 rpm
- fall time = 6.75 s;
- max. positive error = 6 rpm ;
- max. negative error = 9 rpm;

$t \in [40\text{s};\ 80\text{s}]$, speed from 500 to 750 rpm
- rise time = 4.75 s;
- max. positive error = 8 rpm ;
- max. negative error = 6 rpm;

$t > 80\text{s}$, speed from 750 to 0 rpm
- fall time = 6.5 s;

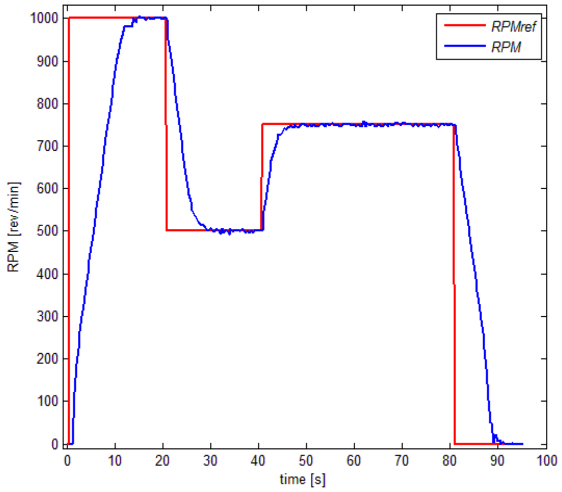

Figure 5. Current speed vs. the desired speed profile.

The quite large error (maximum 9 rpm) is mainly due to the limited accuracy of the speed measuring method (magnetic sensor). It can be easily see that the controller does its job, by continuously updating the control signal to keep the speed as close as possible to the reference value. Supplementary details can be observed on the intermediate signals in the control system presented in Figure 6. These intermediate signals provide the premises for a more in-depth understanding of the control system behavior. As an example, the modification of the control signal is large when a large variation of the motor speed is necessary: $\Delta u = -10$ when the speed should be change from 1000 rpm to 500 rpm, or $\Delta u = +10$ when the speed should be changed from 500 rpm to 750 rpm. As the current speed is getting closer to the reference speed, Δ*u* takes smaller positive or negative magnitude.

To drastically decrease the time response of the control system, the control strategy can be slightly modified. Because the control characteristic of the DC motor driven by the H-Bridge is almost liner, when a large variation of the motor speed is required (larger than 60 rpm), the control signal is not determined by the fuzzy logic system, but it is estimated by a simple linear interpolation, that acts as a course adjustment of the control signal. Then, the fuzzy logic system regains its role for the fine adjustment of the speed.

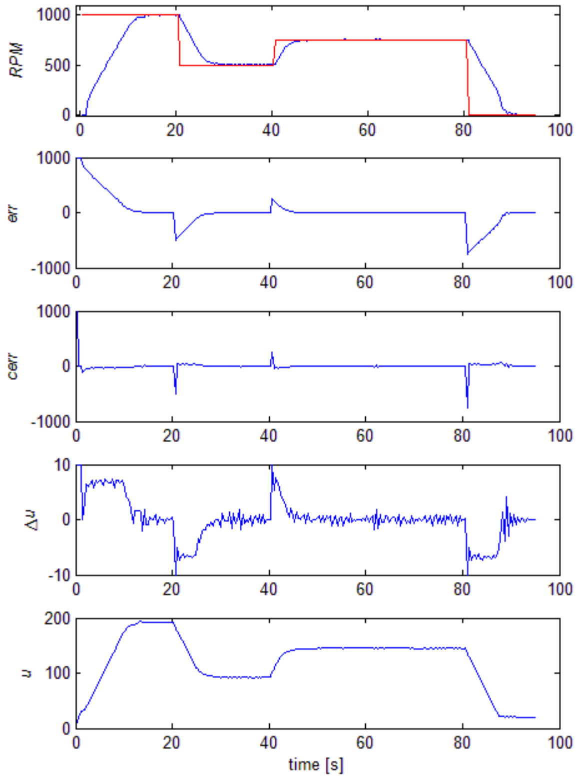

Figure 6. Intermediate variables in the control system.

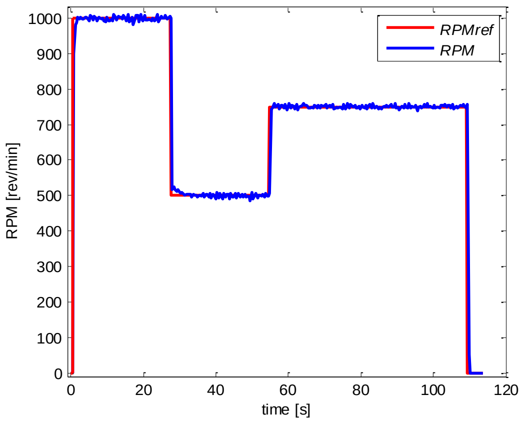

Figure 7. Current speed vs. the desired speed profile using a linear interpolation for course adjustment then the fuzzy logic system for fine adjustment.

The results obtained with this approach are presented in Figure 7 for the same speed profile as above. It is obvious that the control system reacts almost instantly to large changes in the reference speed, while maintain the same performance in the steady-state regime. Even if the control law of the DC motor is almost linear, a (fuzzy) controller is necessary to regulate the speed against load and/or dc supply voltage variation.

The plots presented in this section contains only experimental data. The data were collected during the motor operation, by means of the Arduino board and the software, and then transferred to Matlab for graphical representation.

V. CONCLUSIONS

An explicit implementation of an embedded, closed loop system, for speed tracking of a micro metal gear motor was presented in this paper. The main benefit of this implementation refers to the learning process, where understanding theoretical concepts is highly facilitated by experimenting the studied phenomena or process in their real environments. The embedded system is developed around an Arduino board and uses a fuzzy logic system as controller. Our approach can be seen as a knowledge integrator system, as long as it implies microcontroller-based development board, C++ programming, Hall effect sensor for speed measurement, H-Bridge for driving a DC motor, fuzzy logic system for the controller. By operating minor changes in the Arduino code, different operating scenarios can be investigated easily, so that the user acquires a thorough understanding of the operation of the entire control system.

REFERENCES

[1] A.M. Ibrahim, *Fuzzy Logic for Embedded System Applications*, Elsevier, 2004.

[2] A. A. Khater, M. El-Bardini, N. M. El-Rabaie, "Embedded Adaptive Fuzzy Controller Based on Reinforcement Learning for DC Motor with Flexible Shaft", Arab J. Sci. Eng., vol. 40, no. 8, pp. 2389–2406, 2015.

[3] H. R. Jayetileke, W. R. de Mei and H. U. W. Ratnayake, "Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due", 7th International Conference on Information and Automation for Sustainability, Colombo, 2014, pp. 1-6.

[4] A. Pimpalgaonkar, M. Jha, N. Shukla, K. Asthana, “A Precision Temperature Controller using Embedded System”, Intl. J. of Sci. and Res. Pub., vol. 3, no. 12, pp. 1-4, 2013.

[5] S.R.Sharma, P.B.Dahikar, "Embedded Design of Temperature Controller using PIC16F876A for Industries and Laboratories", Intl. J. of Innov. Res. in Comp. and Comm.Eng., vol. 1, no. 10, pp. 2414–2422, 2013.

[6] P. Singhala, D.N. Shah, B. Patel, “Temperature Control using Fuzzy Logic”, International Journal of Instrumentation and Control Systems (IJICS) Vol.4, No.1, January 2014, pp. 1- 10;

[7] *30:1 Micro Metal Gearmotor HPCB 6V with Extended Motor Shaft*. Available: https://www.pololu.com/product/3072 [Accessed: May, 2017].

[8] *Arduino Mega*, Available: https://www.arduino.cc/en/Main/arduinoBoardMega, [Accessed: May, 2017].

[9] *Arduino*, Available: https://www.arduino.cc/, [Accessed: May, 2017].