A Low-Cost Real-Time Simulator Of Fuzzy Logic Control For Brushed Dc Motor Drives

Son T. NGUYEN[1] , *Anh Hoang*[1] *Tu M. PHAM*[1] *Tuan V. PHAM*[2]

[1]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
[2]Faculty of Electrical Engineering, Vinh University of Technology Education, Vinh, Vietnam

son.nguyenthanh@hust.edu.vn,anh.hoang@hust.edu.vn,tu.phamminh@hust.edu.vn,tuanvp.bk@gmail.com

Abstract. *This article describes the development of a low-cost, real-time simulator of several control strategies for brushed DC motor drives. An educational brushed DC motor drive test bench and an Arduino Mega 2560 microcontroller board were used to build the simulation system. The system can serve as a platform for laboratory studies in which various control strategies helpful for instruction and research purposes. The interaction between MATLAB Simulink and the Arduino board can be managed through the exploration of the MATLAB Support Package for Arduino. Different speed control algorithms for brushed DC motors can be easily deployed, monitored, and analysed using the Simulink multiform library. Three different types of controllers have been used in this study: a standard PI controller, a fuzzy logic controller, and a fuzzy PI controller. Finally, the system is particularly helpful in comparing various control approaches for many other controlled objects in addition to drives for brushed DC motors.*

Keywords

Brushed DC motor, real-time simulation, MATLAB Support Package for Arduino, fuzzy logic control.

1. Introduction.

Electric drive systems can be only seen as "high-performance" drive systems if they can accurately follow a specified trajectory regardless of unknown load variation and uncertainties of parameters [1]. An electric drive system is basically formed by an electrical motor, a power electronic converter, and a controller integrated to perform precise mechanical maneuvers. As being controlled easily over a wide speed range, brushed DC motors are still widely used in various industrial and commercial applications including electric vehicles, robotic manipulators, and precise mechanisms. In addition, brushed DC motors can be known as the most common controlled object for testing and evaluating different single-input and single-output (SISO) control algorithms.

The accurate mathematical model and related parameters of a specific brushed DC motor are usually needed for designing an appropriate controller for the motor. Additionally, the exact model of the motor can allow the designer to predict the closed-loop dynamics of the motor control system. This work can be only facilitated if the motor parameters can be precisely estimated. Motor parameter estimation approaches can be divided into two categories: offline estimation methods [2, 3, 4, 5] and online estimation methods [6, 7, 8]. Offline methods require the use of test inputs and the measurement of corresponding parameters using data acquisition (DAQ) devices. Next, measurement-based determined coefficients are used to obtain unknown parameters by offline computer-based analysis. Meanwhile, online techniques sometimes include power converters and high-speed DAQ devices. Online motor parameter estimations can be performed by using observers, Kalman filters [3], and artificial neural networks [6].

Conventional controller designs for brushed DC motor drives often involve the use of constant gains for controllers like proportional-integral-derivative (PID) controllers, which are only useful for the control of a narrow range of the motor speed [9, 10, 11].

Recently, the brushed DC motor drive control has seen widespread application of non-linear control approaches, such as sliding mode controllers [12, 13], fuzzy logic controllers [14, 15, 16, 17], and artificial neural network controllers [18, 19, 20].

High-performance brushed DC motor drive controllers frequently need to be adaptable, so various research have developed adaptive control methods [21, 22, 23], in which the coefficients of controllers can be adaptive utilizing techniques for intelligent inference such fuzzy logic and artificial neural networks. Several studies have suggested techniques of estimating the speed of brushed DC motors [24, 25, 26, 27, 28, 29] because the use of speed sensors like tachometers or pulse encoders could degrade system reliability and necessitate periodic maintenance. Most speed estimation techniques, however, have been developed using a precise mathematical model of the motor.

The rapid development of real-time simulation and hardware-in-the-loop (HIL) simulation techniques can allow researchers, lecturers, and students to conveniently develop laboratory setups for learning and testing control systems [30, 31].

Low-cost embedded systems for controlling brushed DC drives can be formed by using typical microcontrollers with the inclusion of ADC inputs and PWM outputs such as Arduino Uno board [32], Arduino Mega 2560 board [33] and STM32 microcontroller [34, 35]. In these systems. The use of Arduino boards can be seen as a cost-effective solution but still have high performances based on a lot of available libraries. One of the famous libraries for Arduino boards is the MATLAB Support Package for Arduino [36]. This MATLAB package can provide a powerful library of Simulink blocks for performing analogue/digital inputs and outputs of Arduino boards such as Arduino Uno and Arduino Mega 2560 boards.

As Simulink diagrams with the use of MATLAB Support Package for Arduino can be utilised to easily obtain real-time simulation environments for many control systems, the study is undertaken to contribute a detailed procedure of developing and evaluating real-time advanced control algorithms for DC motor drives including PI controller, fuzzy logic controller and fuzzy logic PI controller. Based on details stated in this paper, readers can conveniently simulate and develop their own control systems.

The rest of this paper is divided into the subsequent sections. The mathematical representation of a chopper-fed brushed DC motor is discussed in Section 2. , along with a straightforward but efficient technique for estimating the parameters of brushed DC motors. The conventional PI controller, fuzzy logic controller, and fuzzy PI controller are the three types of brushed DC motor drives controllers that are presented in Sec-

tion 3. . In Section 4. , a real-time simulation system based on three different controller types and a laboratory test bench for the brushed DC motor drive are discussed. Finally, Section 5. is the conclusions of this research.

2. Brushed DC Motors

The speed of a brushed DC motor can be easily controlled by using a DC chopper converting a fixed DC voltage to variable DC voltage applied to the armature circuit of the motor. For decreasing the ripple of the motor current and avoiding the discontinuous conduction, the DC chopper is usually required to operate at a high frequency. In this study, a separately excited DC motor is controlled by an IGBT based DC chopper.

2.1. Motor Mathematical Model

For brushed DC motors, a DC voltage can be directly applied to the terminal of the armature circuit. Using Kirchhoff's law for the armature circuit gives:

$$v_t(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K\omega(t). \qquad (1)$$

Based on Newton's law, the torque equation can be derived as follows:

$$Ki_a(t) = J\frac{d\omega(t)}{dt} + D\omega(t) + T_L(t), \qquad (2)$$

where K, R_a, L_a, J and D are the back-EMF constant, the armature resistance, the armature inductance, the rotor mass moment of inertia, and the damping constant, respectively. $\omega(t)$, $i_a(t)$, $v_t(t)$, and $T_L(t)$ denote the rotor angular velocity, the armature current, the terminal voltage, and the load torque, respectively.

At the steady-state, equation (1) yields:

$$\omega = \frac{V_t}{K} - \frac{R_a I_a}{K} = \omega_0 - \Delta\omega, \qquad (3)$$

in which

$$\omega_0 = \frac{V_t}{K}, \qquad (4)$$

$$\Delta\omega = \frac{R_a I_a}{K}. \qquad (5)$$

In equations (4) and (5), is the no-load rotor angular velocity and $\Delta\omega$ is the drop of the rotor angular velocity when a load torque is applied to the motor. Obviously, the back-EMF constant can be determined through the terminal voltage and the no-load rotor angular velocity as follows:

$$K = \frac{V_t}{\omega_0}. \qquad (6)$$

When a load torque is applied to the motor shaft, the measured armature current can be used to calculate the armature resistance as follows:

$$R_a = \frac{V_t - K\omega}{I_a}. \tag{7}$$

According to equation (7), if V_t is kept to be a constant, ω can be easily measured using a tachometer or an encoder, and I_a can be measured using a DC ammeter, then the value of R_a can be conveniently computed.

The Laplace transforms of equations (1) and (2) have the following forms:

$$V_t(s) = R_a I_a(s) + L_a s I_a(s) + K\omega(s), \tag{8}$$
$$Js\omega(s) = KI_a(s) - D\omega(s) - T_L(s), \tag{9}$$

where s is the Laplace operator. $V_t(s)$, $I_a(s)$, $\omega(s)$ and $T_L(s)$ are the Laplace transforms of the terminal voltage, the armature current, the rotor angular velocity, and the load torque, respectively.

2.2. Motor Parameter Estimation

The motor parameter estimation is useful for designing model-based controllers. This section describes the procedure for obtaining parameters of a brushed DC motor based on experimental tasks in a laboratory.

From equations (8) and (9), the relationship between the rotor angular velocity (the system output) and the terminal voltage, and the load torque (the system inputs) can be expressed as follows:

$$\omega(s) = \frac{KV_t(s) - T_L(s)(sL_a + R_a)}{s^2 L_a J + s(R_a J + L_a D) + (R_a D + K^2)}. \tag{10}$$

At the no-load regime ($T_L(s) = 0$), equation (10) yields:

$$\omega(s) = \frac{K}{s^2 L_a J + s(R_a J + L_a D) + (R_a D + K^2)} V_t(s). \tag{11}$$

Equation (11) indicates a second-order system and can be shorten as follows:

$$\frac{\omega(s)}{V_t(s)} = \frac{a}{s^2 + sb + c}, \tag{12}$$

in which:

$$a = \frac{K}{L_a J}, b = \frac{R_a J + L_a D}{L_a J}, c = \frac{R_a D + K^2}{L_a J}. \tag{13}$$

The coefficients a, b, and c of equation (12) can be obtained by acquiring the start-up motor speed with respect to the step terminal voltage and the MATLAB System Identification Toolbox [37]. According to equation (13), K and R_a can be determined using equations (6) and (7). Three remaining parameters of the motor can be computed as follows:

$$D = \frac{1}{R_a}\left(\frac{cK}{a} - K^2\right), \tag{14}$$

$$L_a = \frac{bK - \sqrt{(bK)^2 - 4aDKR_a}}{2aD}, \tag{15}$$

$$J = \frac{K}{aL_a}. \tag{16}$$

3. Controller Design

3.1. Conventional PI Controller

The output of a conventional PI controller is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{,}dt \tag{17}$$

$$e(t) = r(t) - y(t), \tag{18}$$

where $r(t)$, $y(t)$, and $e(t)$ are the reference rotor angular velocity, the real rotor angular velocity and the rotor angular velocity error, respectively. K_p, K_t, and K_d are the proportional constant, the integral constant, and the derivative constant, respectively. If only PI controller is considered, equation (17) yields:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt. \tag{19}$$

The Laplace transform of equation (19) is given by:

$$U(s) = K_p E(s) + K_i \frac{1}{s} E(s), \tag{20}$$

where s is the Laplace operator. $U(s)$ and $E(s)$ are the Laplace transforms of the controller output, and the error of the rotor angular velocity, respectively.

The z-transform of equation (20) has the following form:

$$U(z) = K_p E(z) + K_i T_s \frac{z}{z-1} E(z), \tag{21}$$

where z is the z operator and T_s is the sampling period.

Equation (21) can be re-written as follows:

$$U(z) = K_p E(z) + P(z), \tag{22}$$

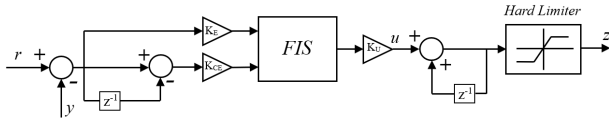In which:

$$P(z) = K_i T_s \frac{1}{1 - z^1} E(z). \tag{23}$$

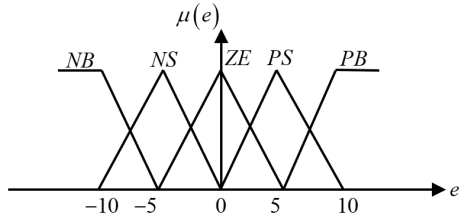**Fig. 1:** The structure of a typical fuzzy logic controller.

Fig. 2: MFs of the error.

Fig. 3: MFs of the change of the error.

Fig. 4: MFs of the output.

The discrete-time form of equation (22) is given by:

$$u(k) = K_p e(k) + p(k), \tag{24}$$

$$p(k) = p(k-1) + K_i T_s e(k), \tag{25}$$

where k labels the k-the sampling period. Equation (25) is used to implement digital PI controllers using microcontrollers.

3.2. Fuzzy Logic Controller

Fuzzy logic control is a heuristic technique that imitates human thought processes. The theory of fuzzy sets served as the foundation for the initial development of fuzzy logic controllers. The mathematical model of the controlled item is not necessary for the process of creating a fuzzy logic controller. Fuzzy logic controllers can therefore function while managing nonlinearity and imprecise inputs. In systems that are complex, nonlinear, or poorly specified, fuzzy logic controllers typically outperform other controllers. Figure 1 illustrates the structure of a fuzzy logic controller for closed loop control systems.

Tab. 1: Table of fuzzy rules for FLCs.

e \ ce	NB	NS	ZE	PS	PB
PB	ZE	PS	PS	PB	PB
PS	NS	ZE	PS	PS	PB
ZE	NS	NS	ZE	PS	PS
NS	NB	NS	NS	ZE	PS
NB	NB	NB	NS	NS	ZE

The main part of a fuzzy logic controller is the fuzzy inference system (FIS), which has two inputs including the error $e(t)$ and the change of the error $ce(t)$ calculated as follows:

$$e(t) = r(t) - y(t), \tag{26}$$
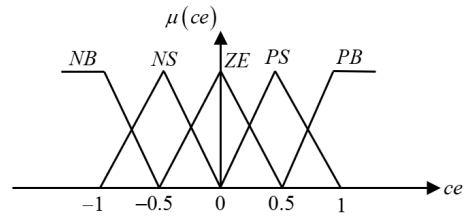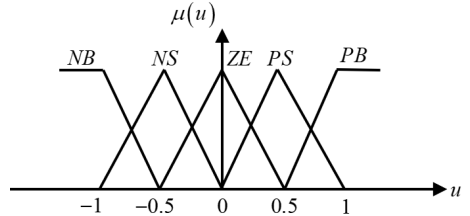
$$ce(t) = \frac{de(t)}{dt}, \tag{27}$$

where $r(t)$ is the reference and $y(t)$ is the output of the controlled object. The discrete-time forms of equation (26) and (27) are expressed as follows:

$$e(k) = r(k) - y(k), \tag{28}$$

$$ce(k) = \frac{e(k) - e(k-1)}{T_s}. \tag{29}$$

The inputs and the output of the FIS must be normalized to be between $[0,1]$ and $[-1,1]$ intervals by adding the scaling factors K_E, and K_{CE} for the inputs and the scaling factor K_U for the output. The discrete-time forms of the inputs of the FIS at the k-th sampling period are expressed as follows:

$$x_1(k) = K_E e(k), \tag{30}$$

$$x_2(k) = K_{CE}[e_k(k) - e(k-1)]. \tag{31}$$

If z_k is the output of the FIS, then discrete-time form of the output of the fuzzy logic controller at the k-th sampling period is given by:

$$z(k) = z(k-1) + K_U u(k). \tag{32}$$

The implementation of the fuzzy logic controller for brushed DC motor requires the following steps:

- Identification of input and output variables: The first input is the speed error, and the second input is the change of the speed error. The output variable is the duty cycle of pulse width modulation (PWM) signal generated to control the motor speed.

- Fuzzification: The universe of information spanned by each variable is divided into several fuzzy subsets and each subset is assigned a linguistic variable such as NB (Negative Big), NS (Negative Small), Zero (ZE), PS (Positive Small), and
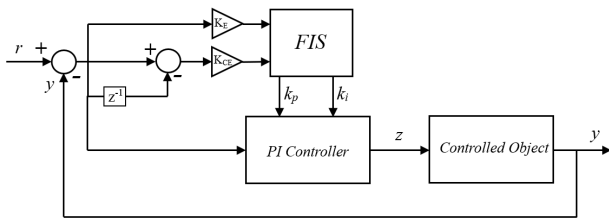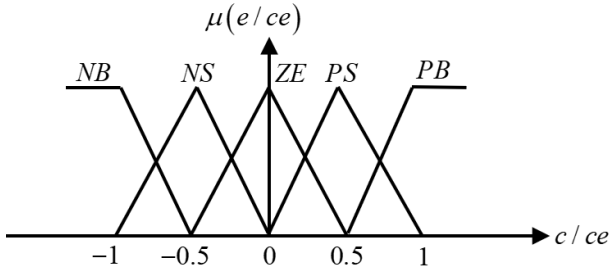
Fig. 5: The structure of a fuzzy PI controller.

Fig. 6: MFs of the error and the change of the error.

Fig. 7: MFs of the proportional gain.

Tab. 2: Table of fuzzy rules for fuzzy PI controller.

e \ ce	NB	NS	ZE	PS	PB
NB	VB/S	VB/M	S/M	S/M	M/S
NS	VB/B	B/B	S/B	S/B	B/B
ZE	B/B	M/B	S/VB	S/VB	VB/VB
PS	M/B	S/B	S/B	M/B	VB/B
PB	S/S	S/S	S/M	B/M	VB/S

PB (Positive Big). This step needs to establish membership functions (MFs) for fuzzy subsets.

- Configuration of the fuzzy rule base: Formulate a fuzzy rule base by assigning several different relationships between the fuzzy inputs and output.

- Determination of normalizing and scaling factors: Appropriate scaling factors for input and output variables must be chosen to normalize variables between [0, 1] and [-1, 1] intervals.

- Defuzzification: obtain a single number from the output of the aggregated fuzzy set. This step is used to transfer fuzzy inference results into a crisp output.

1) Fuzzification

In this step, MFs are formed to obtain features of fuzzy sets. A MF is a curve used to map each point in the input space to a membership value in a closed unit interval [0, 1]. Simple MFs widely used include trapezoidal MFs and triangle MFs. Figures 2 and 3 show MFs of the inputs and Figure 4 shows MFs of the output of the FIS.

2) Fuzzy Rules

Fuzzy rules are established to provide a mapping between the inputs and the output based on linguistic variables. Fuzzy rules work in an analogous way to human intuition. Table 1 is the table of fuzzy rules which can be applied to the problem of brushed DC motor drive control.
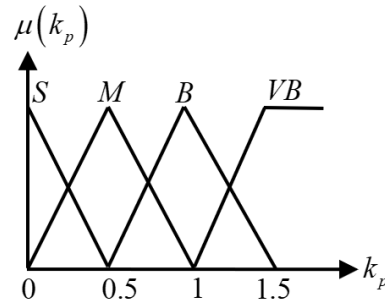
According to Table 1, some fuzzy rules can be expressed as follows:

- If e is PB and ce is NB, then u is ZE.

- If e is PS and ce is NB, then u is NS.

- If e is ZE and ce is NB, then u is NS.

- $\cdots$

3) Defuzzification

Defuzzification is the process of getting a single number from the output of the combined fuzzy set. It is used to convert the findings of fuzzy inference into a crisp output. The center of gravity (COG) approach of defuzzification is the most popular technique. The following formula is used to calculate the defuzzied value:

$$u_{COG}^* = \frac{\int u\mu(u)\,du}{\int \mu(u)\,du}. \tag{33}$$

3.3. Fuzzy Logic PI Controller

Fuzzy PI controllers can be seen as a generalization of the conventional PI controller. However, different from the conventional PI controllers with the fixed proportional and integral constants, fuzzy PI controllers can be designed to adaptively derive proportional and integral gains using fuzzy logic. The structure of a fuzzy PI controller is shown in Figure 5.

A fuzzy PI controller has two inputs (the error and the change of the error) and two outputs corresponding to the adaptive proportional and integral constants
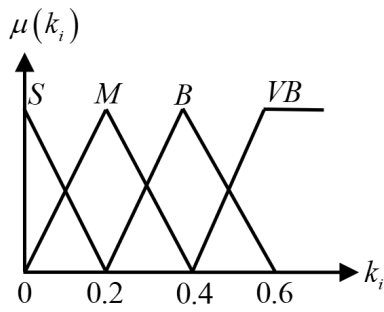
Fig. 8: MFs of the integral gain.

determined by a FIS. The implementation of fuzzy PI controllers requires the following steps:

- Identification of input and output variables: The first input is the speed error, and the second input is the change of the speed error. The first output variable is the proportional gain, and the second output is the integral gain.

- Fuzzification: For the inputs, fuzzy subsets are given the linguistic variables NB (Negative Big), NS (Negative Small), Zero (ZE), PS (Positive Small), and PB (Positive Big). Fuzzy subsets are given the linguistic variables S (Small), M (Medium), B (Big), and VB (Very Big) as inputs for the outputs. For the fuzzy subsets displayed in Figures 6, 7, and 8, membership functions (MFs) must be created in this stage.

- Configuring the fuzzy rule base: Create fuzzy rules by designating various connections between the fuzzy inputs and fuzzy outputs, as illustrated in Table 2.

- Determination of normalizing and scaling factors: To normalize variables between [0, 1] and [-1, 1] intervals, suitable scaling factors for input and output variables must be selected.

- Defuzzification: take two numbers from the combined fuzzy set's outputs. This process converts the outcomes of fuzzy inference into crisp outputs.

In defuzzification, the fuzzy inference results are transferred into the crisp outputs. Like fuzzy logic controllers, the center of gravity (COG) method is also used to obtain crisp outputs as follows:

$$k_p^* = \frac{\int k_p \mu(k_p) \, dk_p}{\int \mu(k_p) \, dk_p}, \qquad (34)$$

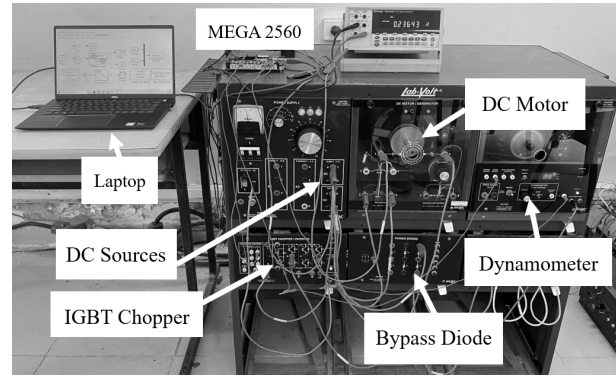$$k_i^* = \frac{\int k_i \mu(k_i) \, dk_i}{\int \mu(k_i) \, dk_i}. \qquad (35)$$

Fig. 9: The experimental setup.

4. Experimental Results

This section describes a procedure of developing a real-time simulation system for different control algorithms of a brushed DC motor drive in laboratory as shown in Figure 9. The system has been constructed by the following specifications:

- A 175W separately excited DC motor manufactured by Lab-Volt company.

- Two separated DC voltage sources: a fixed DC voltage source of 220V supplied to the excitation circuit and an adjustable DC voltage source ranging from 0 to 220V applied to a DC chopper connected the armature of the motor in series.

- A dynamometer working as a changeable load torque.

- An Arduino Mega 2560 board with a default PWM frequency of 980 Hz for a PWM output pin used to control the speed of the motor.

- A PC/laptop with the pre-installed MATLAB Support Package for Arduino.

$R_a(\Omega)$	$L_a(H)$	$K(V/rad/s)$	$J(kg.m^2)$	$D(N.m.s)$
24.2674	1.1752	1.8884	0.0383	7.6639×10^{-4}

Tab. 3: Estimated parameters of the motor

4.1. System Identification

Figure 10 shows the Simulink model for applying a step terminal voltage to the armature circuit and measuring the rotor speed. The Simulink model is configured for the real-time simulation of different controllers. The response of the rotor angular velocity with respect to the step terminal voltage at the no-load scheme as shown in Figure 11 can serve as the input information in the
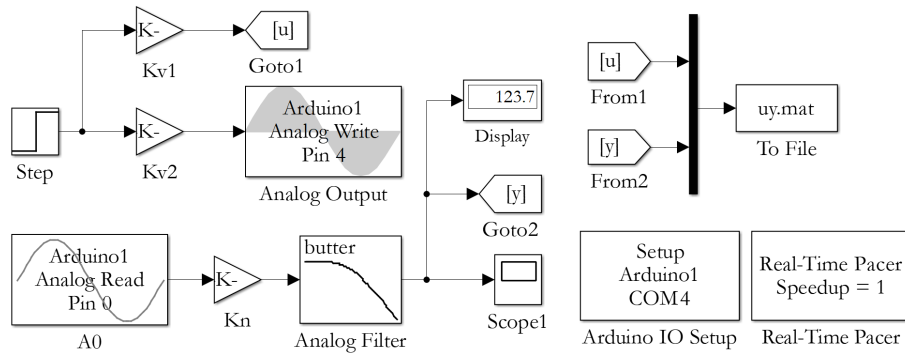
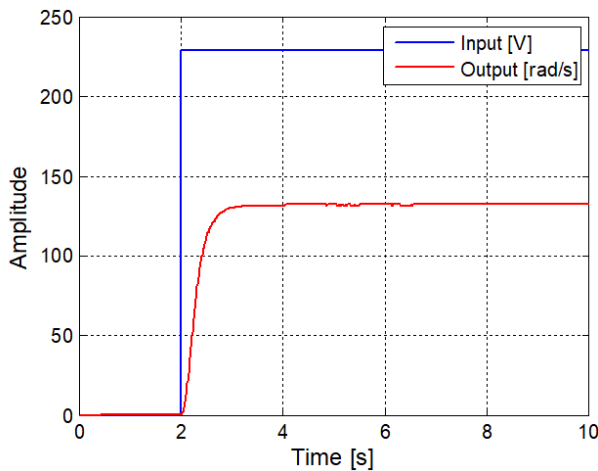**Fig. 10:** Simulink model to obtain the step response of the motor.

Fig. 11: Response of the rotor angular velocity with respect to the step terminal voltage at the no-load regime.

MATLAB System Identification Toolbox to derive the transfer function of the motor. To eliminate noises in the measured speed, a Butterworth filter is used.

Based on the speed response in Figure 11 and the MATLAB System Identification Toolbox, the DC motor transfer function at the no-load scheme is obtained as follows:

$$G(s) = \frac{\omega(s)}{U(s)} = \frac{a}{s^2 + bs + c} = \frac{41.97}{s^2 + 20.67s + 79.67}. \tag{36}$$

The terminal voltage was kept being $220(V)$ and the steady-state rotor angular velocity is $116.5(rad/s)$. Then, the back-EMF coefficient is given by:

$$K = \frac{V_t}{\omega_0} = \frac{220}{116.5} = 1.8884\,(V/rad/s). \tag{37}$$

With the load torque of $1.5(N.m)$, the measured armature current $I_a = 1.105(A)$ and the rotor angular velocity $\omega = 102.3(rad/s)$. R_a and D can be computed as follows:

$$R_a = \frac{V_a - K\omega}{I_a} = 24.2674\,(\Omega). \tag{38}$$

Three remaining parameters of the motor are determined as follows:

$$D = \frac{1}{R_a}\left(\frac{cK}{a} - K^2\right) = 7.6639 \times 10^{-4}\,(N.m.s), \tag{39}$$

$$L_a = \frac{bK - \sqrt{(bK)^2 - 4aDKR_a}}{2aD} = 1.1752\,(H), \tag{40}$$

$$J = \frac{K}{aL_a} = 0.0383\,(kg.m^2). \tag{41}$$

The estimated parameters of the motor are shown in Table 1.

```
num = 41.97;
den = [1 20.67 79.67];
G = tf(num, den)
pidtune(G,'pi')
```

Tab. 4: MATLAB script to determine Kp and Ki of the PI controller

4.2. Conventional PI Controller Simulation

Figure 12 is the Simulink diagram of a conventional PI controller. Based on the function of tuning gains for conventional PI controllers in MATLAB for the transfer function of the system at the no-load mode as shown in Table 4, the proportional constant K_p and the integral constant K_i are determined to be 3.06 and 17.89, respectively.

The increase response of the motor speed is depicted in Figure 13. Figure 14 is the decrease speed response of the motor.

4.3. Fuzzy Logic Controller Simulation

Figure 15 is the Simulink diagram of the fuzzy logic controller. After several trials, the scaling factors for
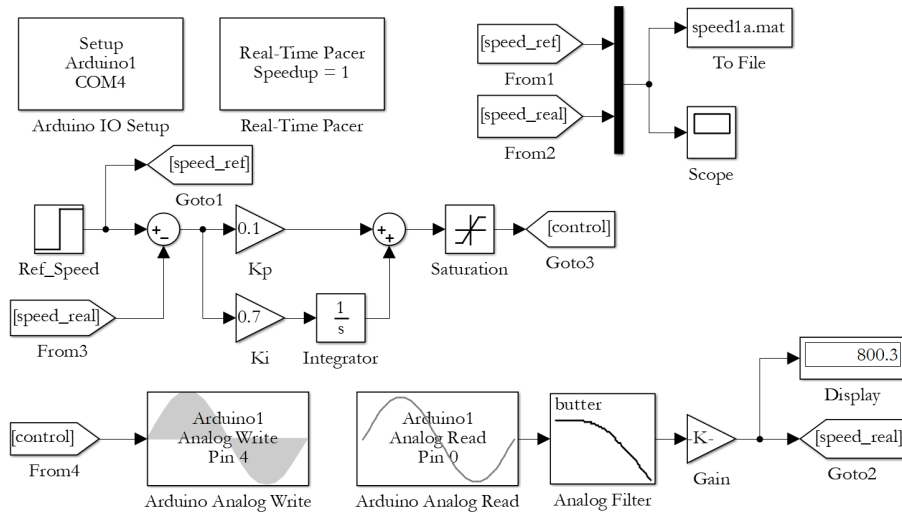
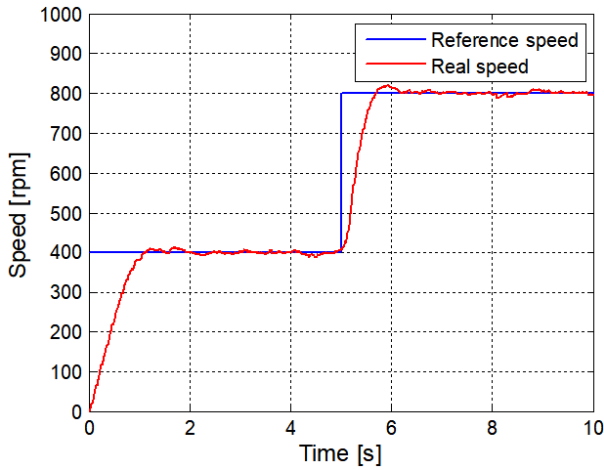**Fig. 12:** The Simulink model of the conventional PI controller.

Fig. 13: Increased speed response with the conventional PI controller.
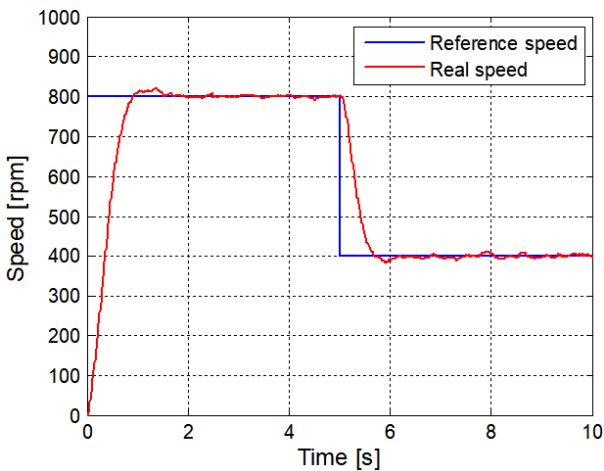

Fig. 14: Decreased speed response with the conventional PI controller.

both inputs are chosen to be 10-4, and the scaling factor for the output is selected to be 350. The increase speed response of the motor is shown in Figure 16 and the decrease speed response of the motor is illustrated in Figure 17.

4.4.   Fuzzy PI Controller Simulation

The Simulink model of fuzzy PI controller is shown in Figure 18. The scaling factors for both inputs are chosen to be 10^{-3}. The scaling factor for the first output is selected to be 0.9 and the scaling factor for the second output is set to be 1.35. The increase speed response of the motor is shown in Figure 19 and the decrease speed response of the motor is depicted in Figure 20.

According to Table 5, the PI controller can result in the lowest overshoot in the low-speed range. The PI controller can also give the shortest rise time and settling time. As shown in Table 6, in the high-speed range, the fuzzy PI controller can give the smallest overshoot and the PI controller can also result in the smallest rise time and settling time.

Controller type	Rise time (s)	Settling time (s)	Overshoot (%)
PI controller	0.7897	4.7868	1.6300
Fuzzy controller	1.0238	4.9308	2.7733
Fuzzy PI controller	0.9778	4.8648	6.4945

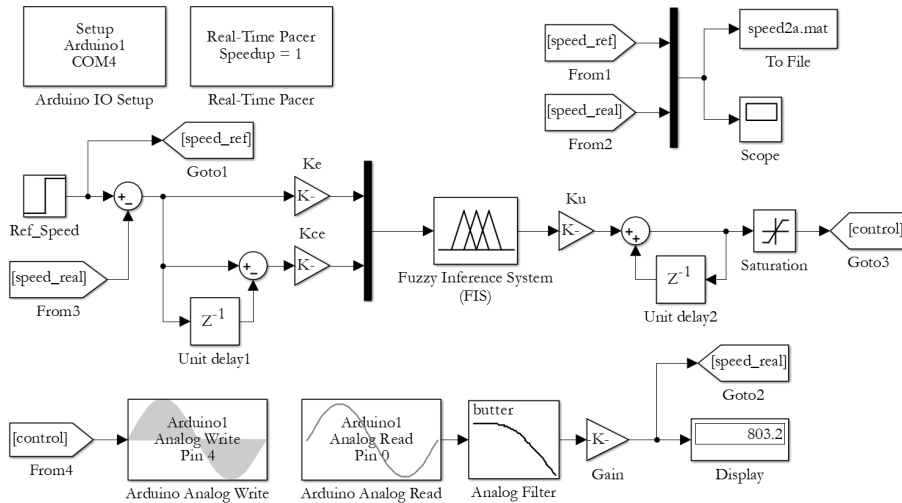**Tab. 5:** Comparison of performances of different controllers at the low-speed range (400 rpm).

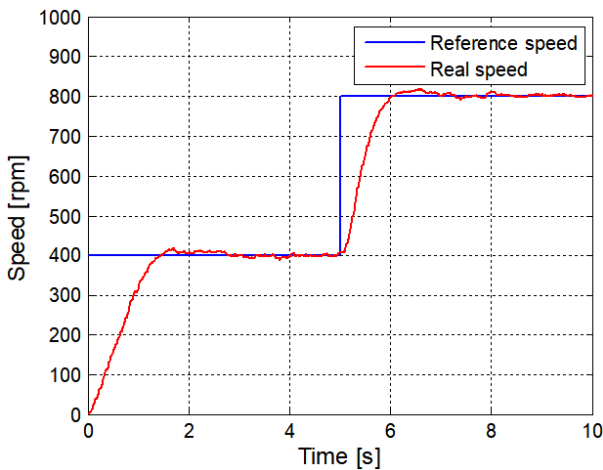**Fig. 15:** The Simulink model of the fuzzy logic controller.

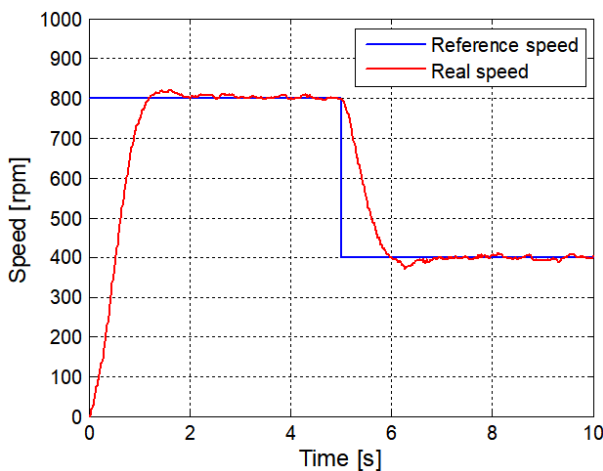**Fig. 16:** Increased speed response with the fuzzy logic controller.

Controller type	Rise time (s)	Settling time (s)	Overshoot (%)
PI controller	0.5933	1.4300	2.3866
Fuzzy controller	0.7649	1.7222	2.5216
Fuzzy PI controller	1.1220	1.7311	0.7434

Tab. 6: Comparison of performances of different controllers at the high-speed range (800 rpm).

5. Conclusions

This research performs the implementation and evaluation of the conventional PI controller, the fuzzy logic controller, and the fuzzy PI controller for tracking the speed trajectory of a laboratory brushed DC motor drive based on the combination of a widely used Arduino Mega 2560 and the MATLAB Package for Arduino. Based on an experimental system, all the parameters of the motor can be precisely estimated using the no-load motor speed response and the MATLAB System Identification Toolbox. In addition, the deployment of Simulink-based control algorithms can facilitate researchers and students to quickly analyze control strategies for the brushed DC motor drive. From analyses in the previous section, the PI controller can be used to obtain a high performance of the system if the motor parameters can be exactly determined. If the estimation of the motor model is difficult or impossible, the fuzzy PI controller should be used for the high reference speed. If the reference speed varies in a relatively wide range, then the fuzzy controller could be an optimal choice.
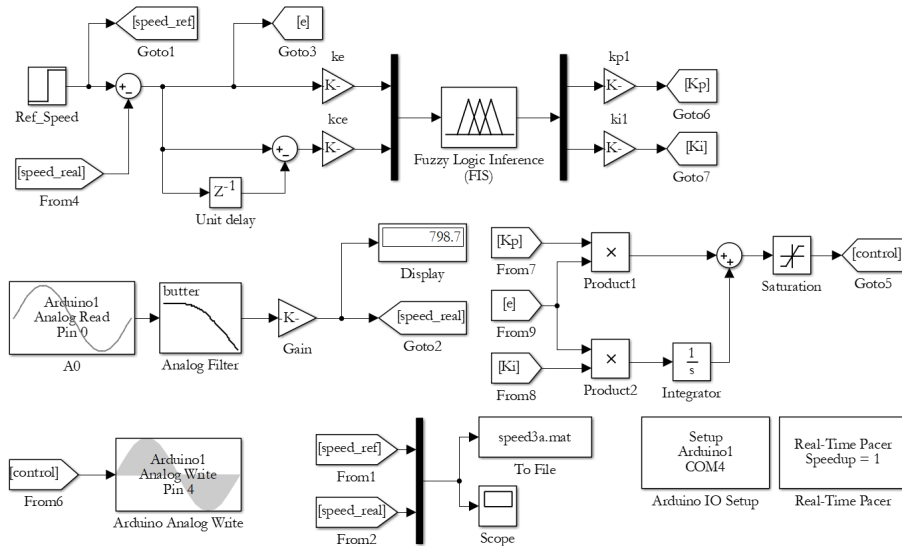

Fig. 17: Decreased speed response with the fuzzy logic controller.

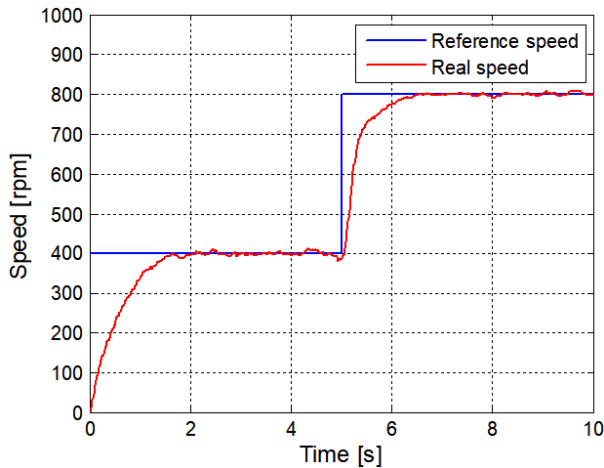**Fig. 18:** The Simulink model of the fuzzy PI controller.

Fig. 19: Increased speed response with the fuzzy PI controller.

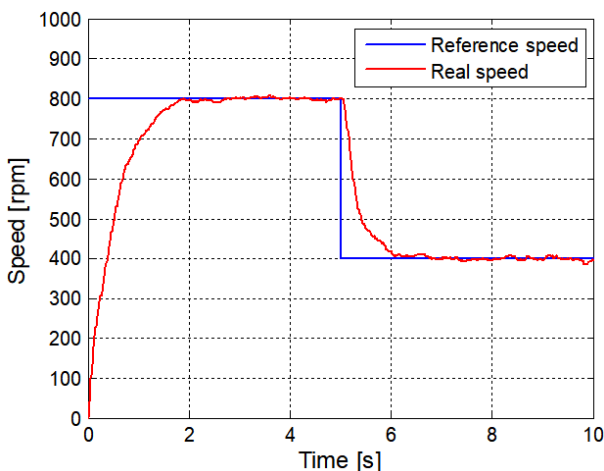**Fig. 20:** Decreased speed response with the fuzzy PI controller.

Acknowledgment

Author Contributions

Son T. Nguyen was responsible for establishing a detailed framework for the study. Anh Hoang, Tu M. Pham, and Tuan V. Pham were responsible for deploying the experimental system and finalised the manuscript.

References

[1] S. WEERASOORIYA, M.A. EL-SHARKAWI, Identification and control of a DC motor using back-propagation neural networks, *IEEE Transactions on Energy Conversion*, vol. 6, no. 4, pp. 663–669, Dec. 1991. DOI: 10.1109/60.103639.

[2] D. PUANGDOWNREONG, S. HLUNGNAMTIP, C. THAMMARAT, A. NAWIKAVATAN, Application of flower pollination algorithm to parameter identification of DC motor model, 2017 *International Electrical Engineering Congress (iEECON)*, 2017, DOI: 10.1109/IEECON.2017.8075889.

[3] S.S. SAAB, R.A. KAED-BEY, Parameter identification of a DC motor: an experimental approach, 8th *IEEE International Conference on Electronics, Circuits and Systems*, 2001, pp. 981–984, DOI: 10.1109/ICECS.2001.957638.

[4] ROBERT BELOIU, Dynamic determination of DC motor parameters - Simulation and testing, Proceedings of the 2014 6th *International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2014, pp. 13–17, DOI: 10.1109/ECAI.2014.7090191.

[5] FRANZ HAMANN, GUSTAVO MESONES, Development of a hybrid system for automatic identification of brushed direct current motors, *2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2020, DOI: 10.1109/INTERCON50315.2020.9220265.

[6] RUBAAI, R. KOTARU, Online identification and control of a DC motor using learning adaptation of neural networks, *IEEE Trans. Industry Applications*, vol. 36, no. 3, pp. 935–942, Jun. 2000. DOI: 10.1109/28.845075

[7] RAJAGOPAL. K.P JISHNU, T. ANANTHAN, Online System Identification of DC Motor Using LabVIEW-myRIO, *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018, pp. 528–531, DOI: 10.1109/ICIRCA.2018.8597260.

[8] MEIDONG LI, YANJIAO MA, Parameter Identification of DC Motor based on Compound Least Square Method, *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020, pp. 1108–1111, DOI: 10.1109/ITOEC49072.2020.9141652.

[9] ERSIN YOLACAN, SERKAN AYDIN, H. METIN ERTUNC, Real time DSP based PID and state feedback control of a brushed DC motor, *2011 XXIII International Symposium on Information, Communication and Automation Technologies*, 2011, DOI: 10.1109/ICAT.2011.6102119.

[10] HONGFU ZHOU, DC Servo Motor PID Control in Mobile Robots with Embedded DSP, *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 2008, pp. 332–336, DOI: 10.1109/ICICTA.2008.426.

[11] J.J. MUOZ-CESAR, E.A. MERCHAN-CRUZ, L.H. HERNANDEZ-GOMEZ, E. GUERRERO-GUADARRAMA, A. JIMENEZ-LEDES, Speed Control of a DC Brush Motor with Conventional PID and Fuzzy PI Controllers, *2008 Electronics, Robotics and Automotive Mechanics Conference (CERMA '08)*, pp. 344 - 349, Oct. 2008. DOI: 10.1109/CERMA.2008.67.

[12] MOHD SALIM QURESHI, PANKAJ SWARNKAR, SUSHMA GUPTA, Assessment of DC servo motor with sliding mode control approach, *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, 2016, pp. 351–355, DOI: 10.1109/CMI.2016.7413769.

[13] HAI-PENG REN, REN ZHOU, JIE LI, Adaptive backstepping sliding mode tracking control for DC motor servo system, *2017 Chinese Automation Congress (CAC)*, 2017, pp. 5090–5095, DOI: 10.1109/CAC.2017.8243683.

[14] MEEI-LING HUNG, HER-TERNG YAU, PI-YUN CHEN, YUAN-HUNG SU, Intelligent Control Design and Implementation of DC Servo Motor, *2010 International Symposium on Computer, Communication, Control and Automation*, pp. 369–372, DOI: 10.1109/3CA.2010.5533445.

[15] J.J. MUOZ-CESAR, E.A. MERCHAN-CRUZ, L.H. HERNANDEZ-GOMEZ, E. GUERRERO-GUADARRAMA, A. JIMENEZ-LEDESMA, Speed Control of a DC Brush Motor with Conventional PID and Fuzzy PI Controllers, *2008 Electronics, Robotics and Automotive Mechanics Conference (CERMA '08)*, 2008, pp. 344–349, DOI: 10.1109/CERMA.2008.67.

[16] MUNADI, M. AMIRULLAH AKBAR, Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink, *2014 International Conference on Intelligent Autonomous Agents, Networks and Systems*, 2014, pp. 42–46, DOI: 10.1109/INAGENTSYS.2014.7005723.

[17] JANG, A deadzone compensator of a DC motor system using fuzzy logic control, *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol: 31, no, 1, pp. 42–48, Feb. 2001, DOI: 10.1109/5326.923267.

[18] M.A. RAHMAN, M.A. HOQUE, On-Line Self-Tuning ANN-Based Speed Control of a PM DC Motor, *IEEE Trans. Mechatronics*, vol. 2, no. 3, pp. 169–178, Sep.1997, DOI: 10.1109/TMECH.1997.653053.

[19] Zhou Runjing, Yuan Weiting, Zhangfei, "Application of NN-PI Controller in Direct Current Motor Servo System," *2007 8th International Conference on Electronic Measurement and Instruments*, 2007, pp. 468–471, DOI: 10.1109/ICEMI.2007.4351185.

[20] JIRAPUN PONGFAI, WUDHICHAI ASSAWIN-CHAICHOTE, Self-tuning PID parameters using NN-GA for brush DC motor control system, *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017, pp. 111–114, DOI: 10.1109/ECTI-Con.2017.8096185.

[21] DIMITRI MAHAYANA, SABAT ANWARI, Brushed DC motor controller using robust indirect adaptive control, *2009 International Conference on Electrical Engineering and Informatics*, 2009, pp. 253–257, DOI: 10.1109/ICEEI.2009.5254780.

[22] SUWANDI DWI SAHPUTRO, FAHMI FADIL-LAH, NANDA AVIANTO WICAKSONO, FERI YUSIVAR, Design and implementation of adaptive PID controller for speed control of DC motor, *2017 15th International Conference on Quality in Research (QiR) : International Symposium on Electrical and Computer Engineering*, 2017, pp. 179–183, DOI: 10.1109/QIR.2017.8168478.

[23] ABDULLAH Y. AL-MALIKI, KAMRAN IQBAL, FLC-based PID controller tuning for sensorless speed control of DC motor, *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018, pp. 169–174, DOI: 10.1109/ICIT.2018.8352171.

[24] E. AFJEI, A. NADIAN GHOMSHEH, A. KARAMI, Sensorless speed/position control of brushed DC motor, *2007 International Aegean Conference on Electrical Machines and Power Electronics*, 2007, DOI: 10.1109/ACEMP.2007.4510598.

[25] YOSHIYUKI KAMBARA, SEIJI UOZUMI, KOUHEI OHNISHI, Disturbance suppression method for position-sensorless motion control of DC brushed motor, *2015 IEEE International Conference on Mechatronics (ICM)*, 2015, pp. 195–199, DOI: 10.1109/ICMECH.2015.7083973.

[26] ROSHAHLIZA M. RAMLI, NAOKI MIKAMI, HISASHI TAKAHASHI, Adaptive filters for rotational speed estimation of a sensorless DC motor with brushes, *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, 2010, pp. 562–565, DOI: 10.1109/ISSPA.2010.5605587.

[27] DANIEL ERTL, LUKE WEBER, A Method for Real-Time Sensorless Speed Control of Brushed DC Motors in Cost Constrained Systems, *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, 2020, pp. 152–157, DOI: 10.1109/GPECOM49333.2020.9248712.

[28] SON NGUYEN THANH, CONG NGUYEN THE, HOA HA XUAN, "Improved Performance of a Sensorless DC Motor Control using Fuzzy Logic," *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2014, DOI: 10.1109/ICIAS.2014.6869506.

[29] A. FARROKH PAYAM, B. MIRZAEIAN DEHKORDI, Nonlinear Sliding-Mode Controller for Sensorless Speed control of DC servo Motor Using Adaptive Backstepping Observer, *2006 International Conference on Power Electronic, Drives and Energy Systems*, 2006, DOI: 10.1109/PEDES.2006.344347.

[30] YUJIA HUO, GIAMBATTISTA GRUOSSO, LUIGI PIEGARI, Power hardware in the loop simulator of photovoltaic plant for smart grid interation analysis, *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe*, Jun. 2017. DOI: 10.1109/EEEIC.2017.7977629.

[31] SHUVRA PROKASH BISWAS, MD. KAMAL HOSAIN, MD. WALIUR RAHMAN, Real-time arduino based simulator enabled hardware-in-the-loop electric DC machine drive system, *2017 IEEE Region 10 Humanitarian Technology Conference*, pp. 823-826, Dec. 2017. DOI: 10.1109/R10-HTC.2017.8289082.

[32] ALFIAN MA'ARIF, ISWANTO, NIA MAHARANI RAHARJA, PHISCA ADITYA ROSYADY, AHMAD RADITYA CAHYA BASWARA, ANINDITYA ANGGARI NURY-ONO, Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation, *2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE)*, 2020, pp. 74 – 78, DOI: 10.1109/ICIEE49813.2020.9277258.

[33] ZARIFI ADEL, AIT ABBAS HAMOU, SEGHIOUR ABDELLATIF, Design of Real-time PID tracking controller using Arduino Mega 2560 for a permanent magnet DC motor under real disturbances, *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, 28-31 Oct 2018, DOI: 10.1109/CISTEM.2018.8613560.

[34] CHONG ZHAO, ZHAOXI HUA, Design of Motor Speed Control System Based on STM32 Microcontroller, *2022 International Conference on Computation, Big-Data and Engineering (IC-CBE)*, 27-29 May 2022, DOI: 10.1109/IC-CBE56101.2022.9888225.

[35] SUWANDI DWI SAHPUTRO, FAHMI FADIL-LAH, NANDA AVIANTO WICAKSONO, FERI YUSIVAR, Design and implementation of adaptive PID controller for speed control of DC motor, *2017 15th International Conference on Quality in Research (QiR) : International Symposium on Electrical and Computer Engineering*, 24-27 Jul 2017, pp. 179 -183, DOI: 10.1109/QIR.2017.8168478.

[36] Legacy MATLAB and Simulink Support for Arduino:https://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink\-support-for-arduino

[37] MATLAB System Identification Toolbox: https://www.mathworks.com/products/sysid.html.

About Authors

Son T. NGUYEN (corresponding author) received the B.Eng. and the M.S degrees in electrical engineering from Hanoi University of Science and Technology (HUST), Vietnam, in 1997 and 1999, and the Ph.D. degree in electrical engineering from University of Technology, Sydney (UTS), Australia in 2007. He worked as a postdoctoral researcher in computer science at the University of Kent (United Kingdom) in 2012. He holds an Australian patent for an effective hands-free control system used by severely disabled people based on head direction and electroencephalography. He is currently a Main Lecturer at the School of Electrical and Electronic Engineering, HUST. His research interests include electrical machines, power electronics, artificial intelligence-based fault diagnosis of electrical equipment, assistive technologies for disabled people. He can be contacted at email: son.nguyenthanh@hust.edu.vn.

Anh HOANG is currently a lecturer at Hanoi university of science and technology. He received his Ph.D. in Sustainable Industrial System from Lorraine University (France). He is currently focusing on designing and installing electrical system, sustainability, and renewable energy system. He was also active in the energy/emission auditing programs and built-up road map for energy efficiency standard. His current research interests include prognostic, energy audit, maintenance plan and energy management.

Tu M. PHAM received the B.Eng. and the M.S. degrees in electrical engineering from Hanoi University of Science and Technology (HUST), Vietnam, in 2009 and in 2012, respectively. He is currently pursuing a Ph.D. degree with the Faculty of Electrical Engineering, School of Electrical and Electronic Engineering, HUST. He has worked as a lecturer in School of Electrical and Electronic Engineering, HUST. His research interests include high efficiency electrical machines, power reactors, reactive power compensation technology, renewable energy. He can be contacted at email: tu.phamminh@hust.edu.vn.

Tuan V. PHAM received the B.Eng., the M.S degrees, and PhD. degree in electrical engineering from Hanoi University of Science and Technology (HUST), Vietnam in 2008, 2012, and 2019, respectively. He has worked as a lecturer in Faculty of Electrical Engineering, Vinh University of Technology Education, Vinh, Vietnam. His research interests include electric machines and drives, power electronics, electrical motor parameter estimation, artificial Intelligence, and machine learning. He can be contacted at email: tuanvp.bk@gmail.com.