Embedded two level direct adaptive fuzzy controller for DC motor speed control

Ahmad M. Zaki\textsuperscript{a,}\textsuperscript{*}, Mohammad El-Bardini\textsuperscript{b}, F.A.S. Soliman\textsuperscript{a}, Mohammed Mabrouk Sharaf\textsuperscript{b}

\textsuperscript{a}Department of Electronics and Computers Engineering, Nuclear Materials Authority, El-Maadi, 530, Cairo, Egypt
\textsuperscript{b}Department of Industrial Electronics and Control Engineering, Faculty of Electronic Engineering, Menofia University, Menouf 32852, Egypt

Received 23 May 2015; revised 3 September 2015; accepted 12 October 2015

KEYWORDS
Embedded systems; Direct adaptive fuzzy controller; DC motor; Speed control; T–S fuzzy controller

Abstract
This paper presents a proposed approach based on an adaptive fuzzy logic controller for precise control of the DC motor speed. In this concern, the proposed Direct Adaptive Fuzzy Logic Controller (DAFLC) is estimated from two levels, where the lower level uses a Mamdani fuzzy controller and the upper level is an inverse model based on a Takagi–Sugeno (T–S) method in which its output is used to adapt the parameters of the fuzzy controller in the lower level. The proposed controller is implemented using an Arduino DUE kit. From the practical results, it is proved that the proposed adaptive controller improves, successfully both the performance response and the disturbance due to the load in the speed control of the DC motor.

\section{1. Introduction}

The permanent magnet direct current (PMDC) motor is an example of electromechanical systems with electrical and mechanical components. In this concern, this type of motors is commonly used in many industrial applications such as robot manipulators, home applications and sun trackers. Noting that, there are many classical and intelligent control techniques [1–5], such as PID, FLC, artificial neural networks, and many methods of AFLC are applied to control the speed of DC motor for achieving high performance. Classical mathematics and conventional control theory are very limited and difficult in modeling and controlling complex nonlinear dynamical systems [6,7]. On the other hand, fuzzy logic controller (FLC) is an alternative tool for PID controller [8,9], where the motivation for using fuzzy logic technology in control systems stems from the fact that it allows control designers to build a controller even when their understanding of the system is still in a vague and incomplete [10,11]. It provides a good tool for the control of nonlinear systems that are difficult in modeling [12,13]. But, the design of the FLC is not the optimum, where numerous difficulties appear to choose the controller parameters. Also, the presence of noise or any

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Diagram of the proposed DAFLC system.}
\end{figure}
changes in the plant parameters, the controller may not be able to achieve adequate performance level. Finally, an adaptive controller is one of the controllers that adjust itself to reach adequate performance [14–16]. In this concern, the main contribution of the present study is the success of the proposed controller to be implemented on the Arduino DUE board to control the speed of DC motor using a proposed DAFLC which uses Takagi–Sugeno (T–S) fuzzy system in the second level to minimize the effects of the system’s load disturbance.

In the present paper, two-level fuzzy controller (TLFC) is presented to control and improve the speed of DC motor. In general, it is considered as an adaptive fuzzy control, where DAFLC can be adjusted directly with a reference model. This design is implemented on Arduino DUE kit, applying Arduino 1.5.5-r2 software. Finally, the organization of the paper is as follows: Section 2 describes the structure and the design of the fuzzy PI controller, and the DAFLC is explained in Section 3. The Arduino implementation is explained in Section 4. Results and discussions are presented in Section 5. Finally, conclusion is in the last section.

2. Fuzzy PI controller design

FLC enables control designers to design and build the controller by forming IF–THEN rules which are in the form of statements [12,17]. The structure of FLC contains four main parts [14,15] as shown in Fig. 1: Fuzzification, inference mechanism, rule base and defuzzification, where fuzzification part is used for converting real input to fuzzy input. The rule-base part contains the expert knowledge in the form of a set of rules. The inference mechanism part evaluates which control rules are relevant at the current time and then decides what the input to process should be and defuzzification part is opposite fuzzification, where it is used for converting fuzzy output to real output. There are two inputs and one output, where the first input is error (e) which is the difference between reference speed of the motor (W_{ref}) and its actual speed of motor (W_{act}). The second input is the change of error (Ce) which is defined as the difference between the present error (e(k)) and the previous error (e(k-1)) given by Eqs. (1) and (2), respectively, where, k is the sampling instance. On the other hand, the output is the change of control signal (ΔU). For getting control signal (u), Eq. (3) is applied, as shown in Fig. 2 as follows:

\[ e(k) = W_{ref} - W_{act} \]  
\[ Ce(k) = e(k) - e(k - 1) \]  
\[ u(k) = u(k - 1) + \Delta U(k) \]

For each of the two inputs and output, there are seven fuzzy sets on universes with linguistic values; namely: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Big (PB). The motor range of speed is very large, so the error and change of error ranges are very high, as it is difficult to handle with large values. The universe of discourse for input and output is normalized value from –6 to 6, where we can use gains G1, G2 and G3 for two inputs and output universe of discourse respectively [18] as shown in Fig. 2. The designers can choose many different shapes based on their preference and experience [17]. These are characterized by the Gaussian membership and are shown in Fig. 3 where, in general the mathematical expression for Gaussian function is

\[ \mu(x) = \exp \left( -\frac{1}{2} \left( \frac{x-c}{\sigma} \right)^2 \right) \]  

where 
\[ c: \text{the center of membership function and} \]  
\[ \sigma > 0: \text{determines the spread or width of the function.} \]

Mamdani type rule base structure is used in this controller; there are 49 rule bases, as given in Table 1. To evaluate the value of the rule antecedents, one should use the AND operator (minimum). But, if a given fuzzy rule has multiple consequent, the OR fuzzy operator (maximum) is used to obtain a single number that represents the result of the consequent evaluation [19]. Finally, many defuzzification methods can be used for leading defuzzification [20]. Finally, weighted average method (Eq. (5)), is proposed to be applied throughout the present work. The matter is due to that it is characterized by its simple calculations and easy for implementation on Arduino (kit/board). From which, and after compensation of the ΔU value into Eq. (3), one can easily get the control signal u (duty cycle). Hence, this result is transferred to the DC motor.

\[ \Delta U = \frac{\sum \mu_l(\Delta U_l) \cdot c_l}{\sum \mu_l(\Delta U_l)} \]  

where 
\[ \mu_l(\Delta U_l): \text{values of membership function (MF) for output, and} \]  
\[ c_l: \text{values of output MF centers.} \]

3. Direct Adaptive Fuzzy Logic Controller (DAFLC)

DAFLC is called a fuzzy model reference learning controller (FMRLC). The functional block diagram for the FMRLC [14,21] is shown in Fig. 4. Basically, it consists of four main parts: the plant, fuzzy controller to be tuned, the reference model, and the learning mechanism block (an adaptation mechanism). Mainly, the fuzzy control system loop (the lower level of Fig. 4) operates to make "W_{act}(kT)" track "W_{ref}(kT)" by manipulating u(kT), while the upper level adaptation control loop which uses Takagi-Sugeno (T–S) fuzzy controller as an inverse model (the upper level of Fig. 4) seeks to make the output of the plant W_{act}(kT) track the output of the reference model "W_m(kT)" by manipulating the fuzzy controller parameters. FLC is explained in Section 2 and we will describe the other parts of DAFLC.
3.1. The reference model

One of the priorities is to choose from the reference model that specifies the desired performance for the overall system. It may be discrete or continuous time, linear or nonlinear, time-invariant or time-varying, and so on. The second order dynamical reference model was used to specify the desired performance for DC motor. The continuous time and discrete time models are given in Eqs. (6) and (7), respectively:

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]  

(6)

\[ \text{Wm}(k) = d_1m \times \text{Wm}(k-1) - d_2m \times \text{Wm}(k-2) + d_3m \times \text{Wref}(k) \]  

(7)

where

- \( \omega_n \): the natural frequency,
- \( \zeta \): damping ratio,
- \( d_1m, d_2m \) and \( d_3m \): discrete model parameters and functions in \( \omega_n \) and \( \zeta \).

3.2. Learning mechanism

The FLC parameters are tuned by the learning mechanism that consists of two main parts: a “fuzzy inverse model” and a “knowledge-base modifier”. The fuzzy inverse model is proposed as T–S fuzzy controller, which is called a “functional fuzzy system” where, the consequent part is just a mathematical function of the input variables [22–24]. The general format of the rule was given in Eq. (8):

\[ R_i: \text{IF } B_{i1}(x_1), B_{i2}(x_2), \ldots, B_{in}(x_n) \text{ THEN } P = f(x_1, x_2, \ldots, x_n). \]  

(8)

where

- \( x_1, x_2, \ldots, x_n \): input variables,
- \( B_{ij}(x_j) (j = 1,2,\ldots,n) \): a fuzzy set on \( X_j \).

In this concern, the antecedent part is processed in exactly the same way as the Mamdani method, and then the consequent part is processed where, \( P \) is calculated as the function of real inputs [18]:

\[ P = a_0 + a_1 \times x_1 + a_2 \times x_2 + \cdots + a_n \times x_n \]  

(9)

In this paper the T–S fuzzy inverse model has two inputs \( (x_1 \text{ and } x_2) \), 25 rule base and one output \( (P \) is a function of two real inputs), where, each of the two inputs membership is triangular, and there are five fuzzy sets on universes with linguistic values; namely: Negative (N), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive (P). The first input \( (x_1) \) is error \( (\text{em}(k)) \) which is the difference between the output speed of the motor, \( \text{Wm}(k) \), and model output speed of the motor \( \text{Wref}(k) \). The second input \( (x_2) \) is change of error \( (\text{Cem}(k)) \) which is defined as the difference between the present error \( \text{em}(k) \) and the previous error \( \text{em}(k-1) \) and its output is \( P(k) \) (adaptation factor) which leads \( \text{em}(k) \) to be zero. The proposed algorithm can be summarized as follows:

- Initialize the FLC parameters.
- Calculate the control signal value \( (u(k)) \), as illustrated in Eq. (3).
Measure the speed value \( W_{\text{act}}(kT) \) and calculate the two inputs of the T–S fuzzy inverse model (error \( \text{em}(kT) \) and change of error \( \text{Cem}(kT) \)).

Calculate the adaptation factor \( P(kT) \).

Find all the rules in the fuzzy controller who the “active set” of the rules at time \( kT - T \).

\[ \mu_i(\varepsilon(kT - T), \text{Ce}(kT - T)) > \varepsilon \]  

where \( 0 \leq \varepsilon < 1 \), because Gaussian membership functions (MFs) are used in FLC and we will not have to modify all the output centers at each time step.

The knowledge-base modifier adjusts the centers and the gains of the output MFs of the FLC as illustrated in Eqs. (11) and (12) where, \( C_i(kT) \) denote the center of the \( i \)th output membership function at time \( kT \) and \( G_j(kT) \) denote the gains of the two inputs and output where, \( j = 1, 2, 3 \).

\[ C_i(k) = C_i(k-1) + P(k) \]  

\[ G_j(k) = G_j(k-1) + P(k) \]

4. Arduino implementation

Arduino [25] is an open-source electronics platform based on easy-to-use hardware and software. The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU Fig. 5, and it is the first Arduino board based on a 32-bit ARM core microcontroller (that allows operations on 4 bytes wide data within a single CPU clock). It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a

Please cite this article in press as: Zaki AM et al., Embedded two level direct adaptive fuzzy controller for DC motor speed control, Ain Shams Eng J (2015), http://dx.doi.org/10.1016/j.asej.2015.10.003
Figure 6  Real system of the DC motor speed control using Arduino DUE.

![Real system of the DC motor speed control using Arduino DUE.](image)

**Figure 7**  (a) Simulation speed response of FLC with DAFLC under 30% load and (b) control signals.

![Simulation speed response of FLC with DAFLC under 30% load](image)

**Figure 8**  (a) Simulation speed response of FLC with DAFLC under 60% load and (b) control signals.

![Simulation speed response of FLC with DAFLC under 60% load](image)
84 MHz clock, a USB capable connection, 2 DAC (digital to analog), 96 kBytes of SRAM and 512 kBytes of Flash memory for the program code. In the proposed Arduino-based speed control system, FLC and DAFLC will be implemented applying Arduino hardware using Arduino 1.5.5-r2 software code. In this concern, Gaussian fuzzifier, Gaussian MF, max of min inference rule and Weighted average method defuzzifier method, will be investigated. Finally, the transfer function of the reference model is chosen by a second order system with the natural frequency of 2.1 rad/s and the damping ratio of 1. The parameters of the discrete equation (Eq. (7)) are given as follows: $d_1m = 1.2269$, $d_2m = 0.3763$, and $d_3m = 0.1494$. Considering the motor specification is illustrated in Table 2 where, the motor current can be measured by ACS712 which has sensitivity of 66 mV/A. The structure of the laboratory experiment is limited to the maximum 60% load. Its speed is measured by optical encoder sharp J3 GPIOA30R, which is considered as a digital input of the Arduino (Fig. 5). Hence, this value was incorporated into the controller algorithm (FLC or DAFLC), resulting-in the control signal $u(k)$ in the form of PWM. Following that, it is sent to the drive circuit MD10C where, one of the features of MD10C is supporting

![Simulation results of RMSE at 30% load](image1)

![Simulation results of RMSE at 60% load](image2)

Figure 9 Simulation of RMSE for FLC and DAFLC under 30% load.

Figure 10 Simulation of RMSE for FLC and DAFLC under 60% load.

![Simulation speed response with two directions and control signal](image3)

Figure 11 (a) Simulation speed response with two directions and (b) control signal.
Figure 12  (a) Step response of the DC motor for DAFLC with two positions, (b) control signal and (c) current waveform.

Figure 13  (a) Open loop response of the DC motor, FLC with DAFLC response under 30% load, (b) control signals and (c) current waveform.
both the sign-magnitude and locked anti-phase PWM signal, the matter which means that one can control motor in two different ways [26]. Real system of the DC motor speed control using Arduino DUE is shown in Fig. 6.

5. Results and discussion

The proposed controller is implemented in Arduino DUE for practical speed controller of the DC motor. The dynamic performance of the DC motor for FLC and DAFLC is applied. In order to clear the improvement of the proposed DAFLC controller, the FLC controller also is implemented for comparison purposes using the same number of the membership functions, a number of rules, the same universe of discourse and the same scaling factors at initial of DAFLC. In this concern, to show the visual indications of the control performance, an objective measure of an error performance was made using the root mean square error (RMSE) and the mean absolute error (MAE) criteria [27,28]. The RMSE and the MAE are defined in Eqs. (13) and (14), respectively:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |e(t)|$$  \hspace{1cm} (13)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (e(t))^2}$$  \hspace{1cm} (14)

5.1. Task 1: simulation results

5.1.1. Disturbance with 30% and 60% load

During the present part of the work, simulation results for the FLC and DAFLC responses, under 30% and 60% loads, on the DC motor at \( k = 500 \) were presented, as shown in Figs. 7 and 8. From this, the RMSE values for the proposed controller and other controllers are shown in Figs. 9 and 10. Besides, the simulation was extended to include the effect of changing the set point of the speed response Fig. 11. It is clearly shown that the DAFLC has better rise time and the priority to respond when applying load rather than FLC.

5.2. Task 2: experimental results

5.2.1. Square wave input for the DC motor

In addition to the simulation, experimental investigation was carried out Fig. 12, where, it is shown that the step response uses a reference speed square wave with positive and negative values (600 rpm and −600 rpm), and it is clear that the proposed DAFLC is better than FLC in the rising time.
5.2.2. External disturbance with 30% and 60% load

The open loop response of the DC motor, FLC with DAFLC response under 30% and 60% load, reference speed (600 rpm), its control signals and current waveform are illustrated as shown in Figs. 13 and 14, where the centers for DAFLC are adapted as shown in Fig. 15. After the load is applied at instant (7.5 s) the updates of gains values are listed in Table 3. The RMSE values for the proposed controller with FLC at 30% load and 60% load are shown in Figs. 16 and 17. The RMSE and MAE values for FLC and DAFLC are illustrated in Table 4. The presented controllers are implemented in different Arduino boards for computing the computation time and implemented size memory; this is presented in Tables 5 and 6. From experimental results it becomes clear that the proposed DAFLC is better than FLC in terms of performance and the priority to respond when applying the load.

![Figure 15](image1.png)

Figure 15  Updates of centers for the DAFLC under 60% load: (a) at initial, (b) at the load and (c) at final.

<table>
<thead>
<tr>
<th>Table 3 Updates of gains for DAFLC.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>At initial</td>
</tr>
<tr>
<td>At load</td>
</tr>
<tr>
<td>At final</td>
</tr>
</tbody>
</table>

![Figure 16](image2.png)

Figure 16  RMSE of FLC with DAFLC at 30% load.

![Figure 17](image3.png)

Figure 17  RMSE of FLC with DAFLC at 60% load.

Table 4  MAE and RMSE for FLC and DAFLC.

<table>
<thead>
<tr>
<th>At 30% load</th>
<th>At 60% load</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FLC</td>
<td>DAFLC with centers and gains</td>
</tr>
<tr>
<td>RMSE</td>
<td>6.653</td>
<td>3.4336</td>
</tr>
<tr>
<td>MAE</td>
<td>0.2477</td>
<td>0.1691</td>
</tr>
<tr>
<td></td>
<td>FLC</td>
<td>DAFLC with centers and gains</td>
</tr>
<tr>
<td>RMSE</td>
<td>7.2808</td>
<td>3.2047</td>
</tr>
<tr>
<td>MAE</td>
<td>0.2698</td>
<td>0.1792</td>
</tr>
</tbody>
</table>
Table 5 Memory size for FLC and DAFLC with different Arduino boards.

<table>
<thead>
<tr>
<th></th>
<th>Arduino Uno</th>
<th>Arduino MEGA</th>
<th>Arduino DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum memory</td>
<td>32,256 bytes</td>
<td>258,048 bytes</td>
<td>524,288 bytes</td>
</tr>
<tr>
<td>FLC</td>
<td>53% (17,164 byte)</td>
<td>7% (19,824 byte)</td>
<td>7% (38,300 byte)</td>
</tr>
<tr>
<td>DAFLC</td>
<td>100% (32,460 byte) over size of memory cannot be implemented</td>
<td>13% (34,352 byte)</td>
<td>8% (45,084 byte)</td>
</tr>
</tbody>
</table>

Table 6 Time of computation for FLC and DAFLC with different Arduino boards.

<table>
<thead>
<tr>
<th></th>
<th>Arduino Uno</th>
<th>Arduino MEGA</th>
<th>Arduino DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLC</td>
<td>2252 µs</td>
<td>2352 µs</td>
<td>989 µs</td>
</tr>
<tr>
<td>DAFLC</td>
<td>Over size of memory cannot be implemented</td>
<td>3824 µs</td>
<td>1246 µs</td>
</tr>
</tbody>
</table>

6. Conclusions

From the experimental work, results, simulation and discussions, it is proved that the design and implementation of both the FLC and the DAFLC were successfully approached, where precise control of the DC motor speed was reported. The DAFLC was tested under two different load conditions, while monitoring its ability to control the speed in both directions. Both FLC and DAFLC are implemented on Arduino DUE hardware using Arduino 1.5.5-r2 software, which are designed to achieve as mentioned high performance and to eliminate the disturbance load. The simulation and practical results show that the proposed controller is able to respond to the disturbance, and it is clear that the proposed DAFLC had better performance than FLC.

References

Ahmad M. Zaki received the B.Sc. degree in automatic control engineering from Faculty of Electronic Engineering, Menoufa University, Menoufa, Egypt. He is currently working toward the M.Sc. degree in automatic control engineering. He is currently an Engineer with Department of Electronics and Computers Engineering, Nuclear Materials Authority, Cairo, Egypt. His current research interests include Fuzzy Logic Systems and adaptive fuzzy control.

Mohammed Mabrouk Sharaf received Ph.D. degree in Control Engineering in 1981 from Department of Electrical Engineering and Electronics, Liverpool University, UK, Lecturer in 1981, Associate professor, 1986, and Professor, 1990, in Department of Industrial Electronics and Control Engineering, Faculty of Electronic Engineering, Menofia University, Egypt. In Oct. 1983–Aug. 1984 he was a postdoctoral research fellow in the Department of Electrical and Electronics Engineering, University of Auckland, Auckland, New Zealand, for processing and modeling of brain data, a member and senior member in the IEEE since 1978, then a life member in 2012. From 1995 till 2001 he was the vice dean of the faculty of electronics engineering, then a head of the dept. during 2003–2004. During 2004–2007 he was the dean of the faculty of electronic engineering, Menofia Univ., Egypt. He is the author of a book on “industrial control system technology” (810 pages in Arabic), and more than fifty research papers. His research interests include mathematical modeling and simulation of dynamic systems, control engineering, adaptive systems, networked control systems and biomedical systems.

Fouad A. S. Soliman is currently a Professor with the Department of Electronics and Computer Science, Nuclear Materials Authority. He is Editor Board Member of Progress in Photovoltaics’ “Research and Applications”, John Wiley and Sons, UK, since 1993, Periodicals of the Association for the Advancement of Modeling and Simulation Techniques, AMSE Lune, France, and International Journal of Computer Science and Engineering Applications (IJCSEA). He is a member of American Association for Advancement of Sciences, NY, USA and New York Academy of Sciences, New York, USA. He is Chosen for who’s in the World, A.N. Marquis, NJ, USA, Outstanding People of the 20th Century and International Biographical Center of Cambridge, England.