

Permanent magnet DC motor (PMDC) model identification and controller design

Ahmed Alkamachi *

System modeling is a set of mathematical equations that describe the dynamical behavior of a system. It is considered as a primary concern in determining a suitable controller to meet specific requirements. An autoregressive with exogenous terms (ARX) model for a PMDC motor is identified experimentally based on the recursive least square (RLS) method. Adaptive discrete pole placement controller (APPC) is proposed and designed aiming to control the motor revolving speed. For the comparison purpose, a discrete Proportional Integral (PI) controller is also considered in this work. The steady step response, transient response, and the mean squared error (MSE) is counted throughout the comparison. The effect of the uncertainties in the PMDC model is also investigated in this paper. The result shows a superiority in the performance of the proposed controller compared to that obtained using PI controller.

Key words: adaptive pole placement controller, ARX model, discrete proportional integral controller, parameter estimation, permanent magnet DC motor

1 Introduction

DC motor system is a typical electrodynamic actuator used in both industrial and conventional applications. It has been widely used due to its precise, simple, and continuous speed control characteristics [1].

PMDC motor came into view over other DC motors because it does not have a field circuit, so eliminating field circuit copper losses caused by the field current.

System identification has got a great attention in the control area, driven by the continuous updates in the control system theory and the recent technologies demands.

The design of a suitable controller to attain a particular response while maintaining the system stability requires appropriate knowledge of the dynamic model of any system. If given, the manufacturer data can be used to extract the system model. Otherwise, the model should be identified experimentally. According to the way it takes place, system identification can be carried out online or offline. In online identification, the update of the adaptive parameter model takes place using the online monitored data, while in offline identification process, the input/output system data set is collected, and a mathematical model is estimated after the entire identification experiment has concluded.

In this paper, a linear transfer function model that represents an input to output relevance of the PMDC motor and the load it drives is obtained. The linear modeling of a loaded PMDC motor is performed by neglecting the

model nonlinearities which normally appears when the motor runs with low speed or when it reverses its direction of rotation. This hypothesis is satisfactorily true as far as traditional control objectives are required. In other words, as long as the motor rotating speed does not pass the zero-speed point, the linear model can significantly be used in describe the system behavior for identification and controller design intents [2].

According to the researches overviewed, motor parameters have been identified by different methods. For example, the authors in [3, 4] used the RLS method to identify the DC motor parameters. G. Mamani *et al* proposed an online algebraic DC motor parameter estimation in [5]. In [6], B. Nayak *et al* used the whale optimization to estimate a separately excited DC motor parameters. In [2], the authors presented a nonlinear direct approach to identify loaded DC motor parameters in closed loop mode. Differential evolution method is proposed to estimate PMDC motor parameters and to control its speed in [7].

This work aims to contribute to the speed control of a PMDC motor. Firstly, an ARX model is constructed using the offline estimation of the motor parameters. Secondly, an adaptive discrete pole placement controller is nominated to control the PMDC motor speed considering the existence of model uncertainties due to the load variations, environmental effects, and other reasons. A PI controller is also designed to show the advantages and the supremacy of the APPC. Finally, the collected results are compared and discussed.

* *University of Baghdad, Al-Khwarizmi College of Engineering, Mechatronics Engineering Department, Baghdad, Iraq, amrk1978@gmail.com

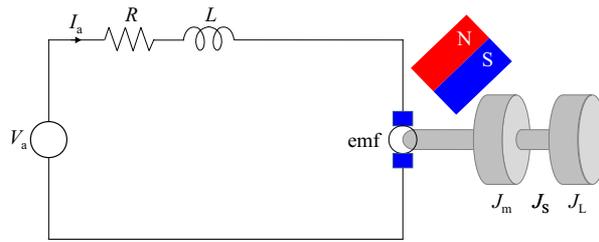


Fig. 1. PMDC motor electromechanical schematic drawing

2 Mathematical model description

The PMDC motors are found in a wide variety of low-power applications. Its permanent magnet presents several useful advantages such as there is no need for external excitations which lessen the required space and reduce the manufacturing costs. Due to their importance and wide range of applications, there is a continuous interest in the modeling and control of DC motors.

Obtaining a precise model represents an essential step in designing a proper controller. It is difficult to obtain an accurate model specially with the existence of motor uncertainties. For this purpose, the system parameters are determined using system identification process.

However, an appropriate model structure should be selected before the identification procedure can start [8]. The objective of modeling the PMDC motor is to obtain the differential equation that relates between the input voltage and output speed.

As mentioned earlier, it is assumed that the motor operates in one direction within the rated speed region so the non-linearity due to zero crossing will be neglected. In this case, the following Laplacian transfer function can be used to describe the PMDC model [9]

$$\omega_M(s) = \frac{K_M}{(sL + R)(sJ + B) + K_M^2} V_a(s) \quad (1)$$

where are: V_a – armature applied voltage (volt),

R – the armature resistance (Ω),

L – armature inductance (henry),

ω_M – rotational speeds of the motor (rad/s),

and moments of inertia (Kgm^2);

J_M – of the motor, J_L – of the load, J_S of the shaft;

B – the coefficient of the viscous friction ($\text{Nm}/(\text{rad/s})$),

K_M – the motor constant (Vs/rad).

The schematic drawing of electromechanical formation of a PMDC motor with load is shown in Fig. 1.

The Recursive Least Square (RLS) method is nominated for the model identification to determine the linear model parameters. It is recommended for its simplicity and applicability to the real system in addition to its advantage of lower computational complexity [8, 10]. A discrete time autoregressive model with exogenous input (ARX) along with recursive identification is used to build up the model structure and to design a suitable APPC. The ARX model is recommended in this work since it

is simple and easy to estimate because the estimation process is made by solving the linear regression equation analytically [11]. The following equation represents the noise free ARX model

$$A(q^{-1})y(t) = B(q^{-1})u(t) \quad (2)$$

where $A(q^{-1})$ is the numerator polynomial of order n_a

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{n_a}q^{-n_a}, \quad (3)$$

$B(q^{-1})$ is the denominator polynomial of order n_b

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \quad (4)$$

and q^{-1} represents the delay operator where the power of the symbol q is the number of delayed samples, and $y(t)$, $u(t)$ are the output and input signal respectively.

For the least square (LS) estimation method, (2) can be written as

$$y(t) = \mathbf{X}^T(t)\boldsymbol{\theta} \quad (5)$$

where $\boldsymbol{\theta}$ is the vector of the model parameters

$$\boldsymbol{\theta}^T = [-a_1, -a_2, \dots, -a_{n_a}, b_0, b_1, \dots, b_{n_b}] \quad (6)$$

and the vector $\mathbf{X}(t)$ is the input/output data set and it is also called the regressor vector

$$\mathbf{X}(t) = [y(t-1), y(t-2), \dots, y(t-n_a), u(t-1), u(t-2), \dots, u(t-n_b)]. \quad (7)$$

By means of the certainty equivalence principle, (5) becomes

$$y(t) = \mathbf{X}^T(t)\hat{\boldsymbol{\theta}} + \hat{\epsilon}(t) \quad (8)$$

where, $\hat{\boldsymbol{\theta}}$ are the estimated model parameters, and $\hat{\epsilon}(t)$ is the modeling error representing the difference between the real and estimated model.

For N input/output sample data set, (8) can be written in vector form as follows

$$\mathbf{Y} = \mathbf{X}^T\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\epsilon}}. \quad (9)$$

The LS method relies on minimizing the following cost function to identify the unknown model parameters

$$J = \hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}. \quad (10)$$

The above cost function can be minimized by equating its partial derivative with respect to $\hat{\boldsymbol{\theta}}$ by zero

$$\frac{\partial J}{\partial \hat{\boldsymbol{\theta}}} = 0 \quad (11)$$

which yields

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \quad (12)$$

For recursive least square RLS, it is assumed that one additional input/output data set is available, then it is

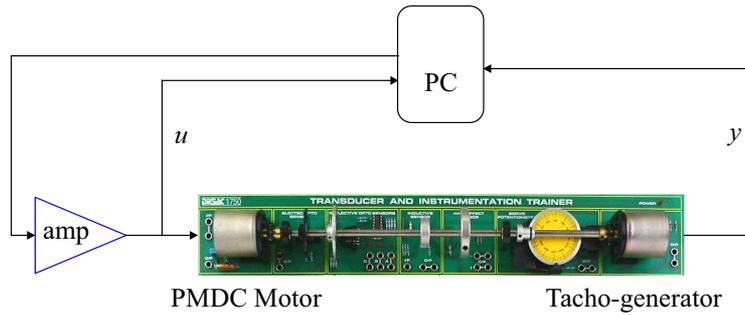


Fig. 2. Interconnection between PC and Digiac1750 kit

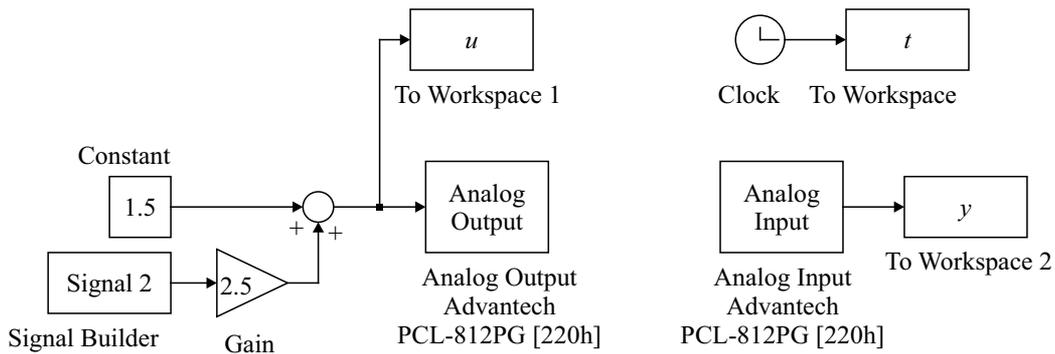


Fig. 3. Matlab/simulink model for the parameter estimation experiment

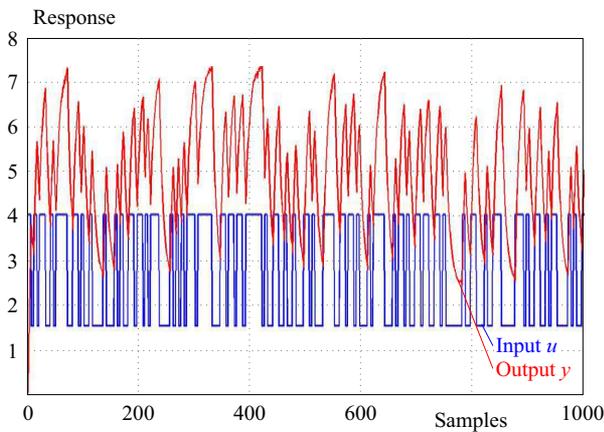


Fig. 4. Tachometer output versus motor input voltage

required to calculate the model parameters $\hat{\theta}(t+1)$ from $\hat{\theta}(t)$ and the additional observation $y(t+1)$, and $u(t+1)$. With this additional input/output data, $\mathbf{X}(t+1)$ and $\mathbf{Y}(t+1)$ become

$$\mathbf{X}(t+1) = \begin{bmatrix} \mathbf{X}(t) \\ \mathbf{x}^\top(t+1) \end{bmatrix}, \quad (13)$$

$$\mathbf{Y}(t+1) = \begin{bmatrix} \mathbf{Y}(t) \\ \mathbf{y}^\top(t+1) \end{bmatrix}. \quad (14)$$

Introducing a covariance matrix $\mathbf{P}(t)$ as

$$\mathbf{P}(t) = [\mathbf{X}^\top(t)\mathbf{X}(t)]^{-1} \quad (15)$$

$\mathbf{P}(t+1)$ becomes

$$\mathbf{P}(t+1) = [\mathbf{X}^\top(t+1)\mathbf{X}(t+1)]^{-1}. \quad (16)$$

Combining (12) and (16) and with some matrix multiplication and simplification, the following expressions can be obtained, and it will be used to tune the model parameters

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1)[y(t+1) - \mathbf{x}^\top(t+1)\hat{\theta}(t)], \quad (17)$$

$$K(t+1) = \frac{P(t)x(t+1)}{1 + M}, \quad (18)$$

$$P(t+1) = P(t) - \frac{P(t)x(t+1)x^\top(t+1)P(t)}{1 + M} \quad (19)$$

and $M = \mathbf{x}^\top(t+1)P(t)x(t+1)$.

Also, the following simplified expression can be deduced from the above equations

$$K(t+1) = P(t+1)x(t+1). \quad (20)$$

Returning to (1) and by looking at the Laplace to Z -transform table, it can be deduced that the ARX representation of the PMDC motor should be in the form

$$\frac{\omega_M(q)}{V_a(q)} = \frac{B(q)}{A(q)} = \frac{b_1q^{-1}}{1 + a_1q^{-1} + a_2q^{-2}}. \quad (21)$$

It means that $n_a = 2$ and $n_b = 1$ and model coefficients (a_1, a_2, b_0, b_1) should be determined.

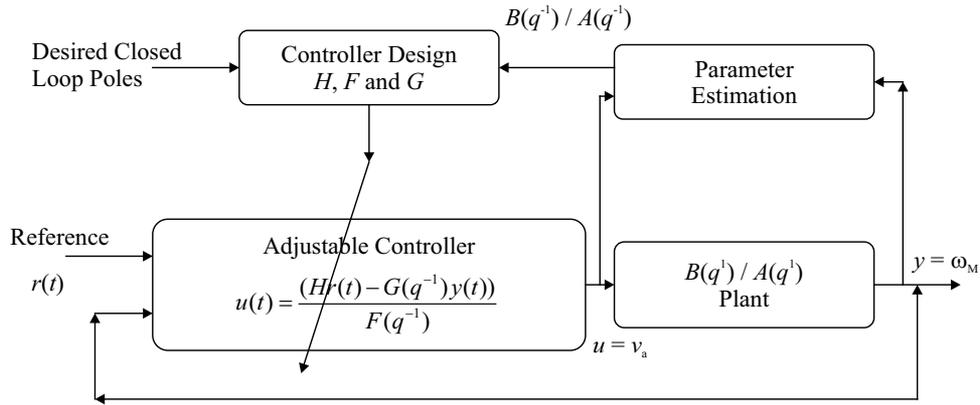


Fig. 5. APPC block diagram

3 Parameter estimation

An experiment has been setup to estimate the parameters of a PMDC motor that is used in Digiact1750 laboratory kit. A Pentium III with 256 MB of RAM is used to collect and process the data. A 12-bit PCL-812PG multi-function analog and digital I/O card is utilized as a data acquisition system to apply an input test signal and to collect the system response. In the following, the steps required to estimate the PMDC model parameter are illustrated.

Step 1: The Digiact1750 laboratory kit has been connected to the personal computer through the data acquisition cards as shown in Fig. 2.

Step 2: Matlab/Simulink environment has been used to send and receive signals to and from the data acquisition card as in Fig. 3.

Step 3: In this phase of the experiment, a test signal has been designed carefully. The input test signal spanned the range from 1 to 4 volt and it was random signal. The number of input samples was 1000 points at 50 Hz sampling rate.

Step 4: The Simulink model in Fig. 3 has been built up and executed. Figure 4 shows the input test signal against the motor speed (tachometer output voltage).

Step 5: After collecting the Input/Output data set, RLS Matlab program has been used to estimate the PMDC motor parameters.

RLS code has been written using Matlab and is used to estimate the required coefficients from the collected input/ output data set. The estimated parameters are $a_1 = -0.89$, $a_2 = 0.03$, $b_1 = 0.00$, $b_2 = 0.26$. These coefficients are used to represent the motor model which is the essential part through the pole placement controller design. Then, the overall PMDC motor transfer function is

$$\frac{\omega_M(q)}{V_a(q)} = \frac{0.26q^{-1}}{1 - 0.89q^{-1} + 0.03q^{-2}}. \quad (22)$$

4 Adaptive discrete pole placement controller

An adaptive pole placement controller is proposed to follow the desired reference input with maximum response speed, no oscillation, zero steady state error, and zero overshoot. Figure 5 represents a general block diagram to illustrate the control action procedure.

The control law of the ADPPC is obtained from the closed loop transfer function of the model and the controller and it is derived as

$$u(t) = \frac{H r(t) - G(q^{-1}) y(t)}{F(q^{-1})} \quad (23)$$

where $G(q^{-1})$ is a polynomial of order n_g ,

$$G(q^{-1}) = 1 + g_1q^{-1} + g_2q^{-2} + \dots + g_nq^{-n_g}, \quad (24)$$

$F(q^{-1})$ is a polynomial of order n_f

$$F(q^{-1}) = f_0 + f_1q^{-1} + f_2q^{-2} + \dots + f_{n_f}q^{-n_f} \quad (25)$$

and H represents the steady state compensator to overcome the pole placement high steady state error problem. It is also known as a feed forward compensator and it is considered an essential part in the pole placement controller construction [12]. Its value is calculated in each sample by summing the T vector coefficients divided by the sum of B vector coefficients. Then by definition,

$$\hat{H} \stackrel{\text{def}}{=} \left[\frac{T}{B} \right]_{SS} \quad (26)$$

$$H = \frac{1 + t_1 + t_2 + \dots}{b_0 + b_1 + b_2 + \dots} \quad (27)$$

where T is the placed poles polynomial of order n_p

$$T = 1 + P_1q^{-1} + P_2q^{-2} + \dots + P_{n_p}q^{-n_p}. \quad (28)$$

By combining (2) and (23), the closed loop transfer function becomes

$$\frac{y(t)}{r(t)} = \frac{H B}{A F + q^{-1} B G}. \quad (29)$$

The denominator of (29) represent the closed loop characteristic equation, so

$$T = AF + q^{-1}BG. \quad (30)$$

A separate Matlab function has been defined to calculate G and F by solving the Diophantine function in (30). With only one pole required to be placed, $n_p = 1$, then

$$\begin{aligned} n_f = n_b = 1, \quad n_g = n_a - 1 = 1, \\ n_t < (n_a + n_b - n_p) = (1) \xrightarrow{\text{yields}} n_t = 1. \end{aligned}$$

Expanding the Diophantine equation in (30) by replacing A , B , G , F , and T with its corresponding polynomial in (3), (4), (24), (25), and (28) respectively yields

$$\begin{aligned} 1 + t_1q^{-1} &= (1 + f_1q^{-1})(1 + a_1q^{-1} + a_2q^{-2}) + \\ & q^{-1}(b_0 + b_1q^{-1})(g_0 + g_1q^{-1}) = \\ & 1 + a_1q^{-1} + a_2q^{-2} + f_1q^{-1} + f_1a_1q^{-2} + f_1a_2q^{-3} + \\ & b_0g_0q^{-1} + b_0g_1q^{-2} + b_1g_0q^{-2} + b_1g_1q^{-3}. \quad (31) \end{aligned}$$

Equation (31) can be solved by equating coefficients that have the same delay operator as follows

$$a_1 + f_1 + b_0g_0 = t_1 \quad (32)$$

$$a_2 + f_1a_1 + b_0g_1 + b_1g_0 = 0 \quad (33)$$

$$f_1a_2 + b_1g_1 = 0 \quad (34)$$

Equivalently, (32), (33), and (34) can be arranged in a matrix form as follows

$$\underbrace{\begin{bmatrix} 1 & b_0 & 0 \\ a_1 & b_1 & b_0 \\ a_2 & 0 & b_1 \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} f_1 \\ g_0 \\ g_1 \end{bmatrix}}_{\eta} = \underbrace{\begin{bmatrix} t_1 - a_1 \\ -a_2 \\ 0 \end{bmatrix}}_{\alpha} \quad (35)$$

Ψ has a special structure (Sylvester matrix), and since it is invertible

$$\eta = \Psi^{-1}\alpha \quad (36)$$

where η is the coefficients vector of the controller polynomials.

5 Pole selection and controller coefficients calculations

The first step in the pole placement controller design is to select a proper closed loop pole. The selection is made to get a stable system with a desired response. To ensure the overall system stability, the selected discrete poles should be inside the unit circle. The closed loop pole is chosen to be $P = 0.75$, then $T = [1 \quad -0.75]$.

Applying the numerical values for selected closed loop pole and the model calculated parameters $A(q)$ and $B(q)$ in (35)

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.89 & 0.26 & 0 \\ 0.03 & 0 & 0.26 \end{bmatrix} \begin{bmatrix} f_1 \\ g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} -0.75 + 0.89 \\ -0.03 \\ 0 \end{bmatrix}.$$

Then, by applying (36), the coefficients vector of the controller polynomials can be calculated and they are: $\eta = [0.14, 0.3638, -0.0162]^T$, and the steady state compensator gain is $H = 0.9315$. The above calculated compensator and controller parameters are updated continuously at each sample to form a selftuning controller that can compensate any change in the system model.

6 Discrete proportional integral (PI) controller

For the comparison purpose, the paper also introduces the design and results of discrete PI controller. The following equation describes this controller transfer function

$$C_{PI}(z) = K_P \left(1 + K_I T_S \frac{1}{z-1} \right) \quad (37)$$

where, K_P , and K_I is the proportional and integral controller gain respectively, and T_S is the sampling time period.

Several values for K_P and K_I were applied and it is noticed that the best response can be obtained with $K_P = 0.3$ and $K_I = 12$.

The PI controller leaks to the automatic adjustment of its gain during system running. In other words, it cannot update its gain coefficients according to the system model and parameters variations. For this reason, the APPC gets on discrete PI controller as it will be seen in the results section.

7 Simulation results

Through the simulations process, simulation step time is chosen as 0.02 sec and the system runs for 10 sec in each test. The APPC and the PI controller will be examined in their response to a desired square wave input.

An algorithm has written to simulate the PMDC motor and to determine the APPC parameters. A subroutine is also used to solve the Diophantine equation and gets the controller coefficients back to the main program. Simulations were conducted for both APPC and discrete PI controller.

The characteristics that are used to compare between the two results are the settling time t_S , maximum peak overshoot M_P , and the mean squared error MSE criteria. These results are in Tab. (1).

Figure 6 and Tab. 1 indicate a better output tracking performance of the APPC as compared to the PI controller approach.

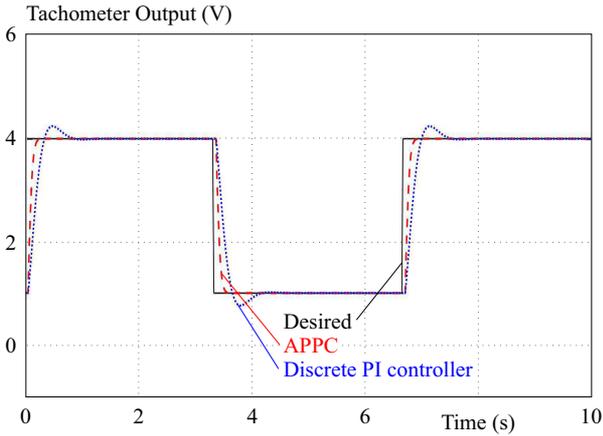


Fig. 6. PMDC motor response with APPC and discrete PI controller

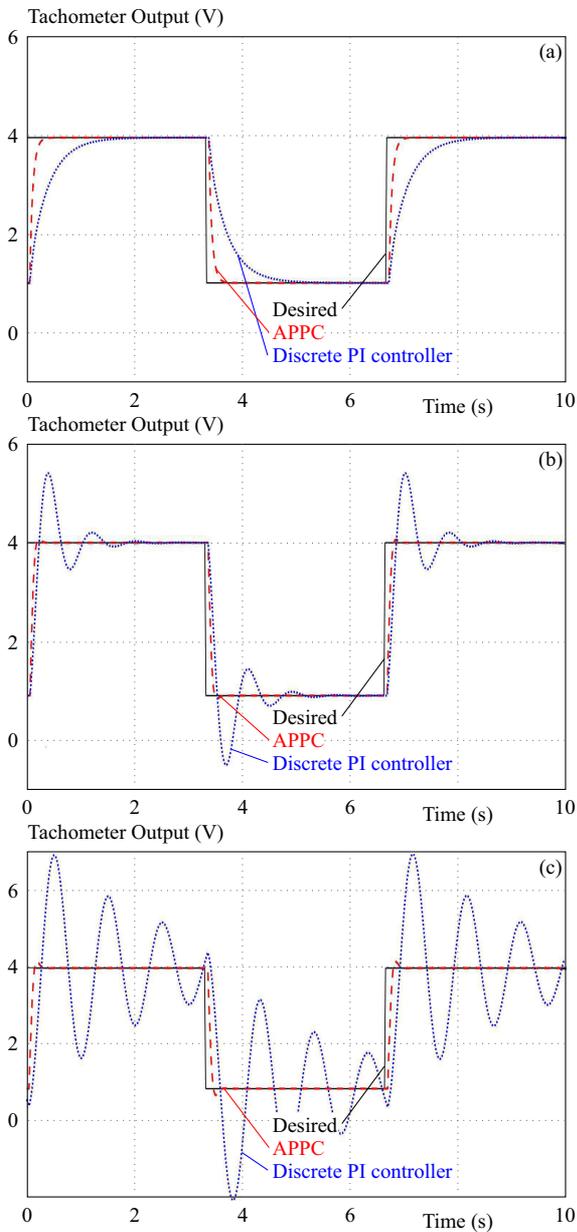


Fig. 7. The effect of model uncertainties of PMDC motor response. (a) – response to case 1, (b) – response to case 2, and (c) – response to case 3

Table 1. Simulation results of APPC versus discrete PI controller

Controller	t_s (s)	M_P (%)	MSE (V)
APPC	0.18	0	0.208
PI controller	0.7	6	0.349

Table 2. Simulation results with the existence of model uncertainties

Case	Controller	t_s (s)	M_P (%)	MSE (V)
1	APPC	0.25	0	0.215
	PI controller	1.26	0	0.547
2	APPC	0.15	1.5	0.207
	PI controller	1.34	34	0.432
3	APPC	0.24	4	0.206
	PI controller	non	70	2.041

To prove the adaptation property of the proposed controller, uncertainty terms were added to the obtained model. The following three arbitrary cases are considered:

Case 1: In the first case, it is assumed that b_1 , a_1 , and a_2 are decreased by 10%, 20%, 20% respectively.

Case 2: In the second case, it is assumed that b_1 , a_1 , and a_2 are increased by 10%, 10%, 20% respectively.

Case 3: For the last case, it is assumed that b_1 is decreased by 25%, a_1 is increased by 15%, and a_2 is decreased by 15%.

The model responses for the three cases are shown in Fig. 7, while the simulation results are described through Tab. 2.

A view at Fig. 7 and Tab. 2 is enough to conclude that the APPC has the adaptation property and it can adjust its coefficients according to the model change. In opposite to this, the PI controller cannot update its gain to suppress the model uncertainties.

8 Conclusions

The speed control of a PMDC motor is considered in this paper. First, an ARX model for a PMDC motor has been identified experimentally. Then, APPC is proposed and designed to control the motor rotating speed. Simulations were conducted for both APPC and discrete PI controller approaches. The experimental and simulation results were plotted and tabulated for the comparison purpose. The controller ability to suppress model uncertainties were also examined for three different cases of uncertain models. The results were compared through graphs and numerically. The overall performance of the two approaches were examined using settling time, maximum peak overshoot, and the MSE indices. The comparison lead to the fact that the APPC has much better performance than the discrete PI controller not only when used with nominal model but also when used with uncertain models.

REFERENCES

- [1] A. S. Al-araji, "Design of Nonlinear PID Neural Controller for the Speed Control of a Permanent Magnet DC Motor Model based on Optimization Algorithm", *Al-Khwarizmi Engineering Journal*, vol. 10, no. 1, pp. 72–82, 2014.
- [2] T. Kara and I. Eker, "Nonlinear Closed-Loop Direct Identification of a DC Motor with Load for Low Speed Two-Directional Operation", *Electrical Engineering*, vol. 68, no. 2, pp. 87–96, 2004.
- [3] R. Krneta, S. Antić, and D. Stojanović, "Recursive Least Squares Method in Parameters Identification of DC Motors Models", *Facta Universitatis-Series: Electronics Energetics*, vol. 18, no. 3, pp. 467–478, 2005.
- [4] M. Ruderman, J. Krettek, F. Hoffmann, and T. Bertram, "Optimal State Space Control of DC Motor", *IFAC Proceedings*, vol. 41, no. 2, pp. 5796–5801, 2008.
- [5] G. Mamani, J. Becedas, and V. Feliu-batlle, "On-Line Fast Algebraic Parameter State Estimation for a DC Motor Applied to Adaptive Control", *Proceedings of the World Congress on Engineering* vol. 2, 2008.
- [6] B. Nayak and S. Sahu, "Parameter Estimation of DC Motor through Whale Optimization Algorithm", *International Journal of Power Electronics Drive Systems*, vol. 10, no. 1, pp. 83–92, 2019.
- [7] M. C. Bosco, J. J. Guedes, M. F. Castoldi, A. Goedtel, E. R. P. Da Silva, and L. F. S. Buzachero, "Estimation of Parameters Tuning of a Speed PI of Permanent Magnet DC Motor using Differential Evolution", *IEEE International Electric Machines Drives Conference (IEMDC)*, pp. 1–6, 2017.
- [8] J. H. Horng, "Neural Adaptive Tracking Control of a DC Motor", *Information Sciences*, vol. 118, no. 1, pp. 1–13, 2009.
- [9] Z. Adel, A. A. Hamou, and S. Abdellatif, "Design of Real-Time PID Tracking Controller using Arduino Mega 2560 for a Permanent Magnet DC Motor under Real Disturbances", pp. 1–5, 7/10/1905.
- [10] X. L. Zhang, L. Cheng, S. Hao, W. Y. Gao, and Y. J. Lai, "Optimization Design of RBF-ARX Model Application Research on Flatness Control System", *Optimal Control Applications Methods*, vol. 38, no. 1, pp. 19–35, 2017.
- [11] D. Ramasubramanian, "Identification Control of DC motors", *Master's thesis Universitat Politecnica de Catalunya*, 2016.
- [12] A. Alkamachi, "Pole Placement Control of a Ball Beam System a Graphical User Interface (GUI) Approach", *Proceedings of the International Conference on Information Communication Technology*, ACM, pp. 184–189, 2019.

Received 6 July 2019

Ahmed Alkamachi was born in Baghdad in 1978. He graduated from the electrical engineering department in University of Baghdad, Iraq, in 2000. He got the MSc degree in electrical engineering/ computer and control division from University of Baghdad, Iraq, 2003. He received a PhD degree in control engineering from Gaziantep University, Turkey, 2017. Since 2004, he has been with Mechatronic engineering department in Al-Khwarizmi college of engineering in University of Baghdad where he is acting as a lecturer. His current research interests span robust control, adaptive control, UAV, and robotics.