Engineering Journal issn: 0125-8281

Article

A Nonlinear Model for Online Identifying a High-Speed Bidirectional DC Motor

Ayad Mahmood Kwad[1,2,a,*], **Dirman Hanafi**[1,b], **Rosli Omar**[1,c], **and Hisyam Abdul Rahman**[1,d]

1 Faculty of Electrical and Electronic Engineering (FKEE), Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, Malaysia
2 Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq
E-mail: [a]ayad_m_k@yahoo.com (Corresponding author), [b]dirman@uthm.edu.my, [c]roslio@uthm.edu.my, [d]arhisyam@uthm.edu.my

Abstract. The modeling system is a process to define the real physical system mathematically, and the input/output data are responsible for configuring the relation between them as a mathematical model. Most of the actual systems have nonlinear performance, and this nonlinear behavior is the inherent feature for those systems; Mechatronic systems are not an exception. Transforming the electrical energy to mechanical one or vice versa has not been done entirely. There are usually losses as heat, or due to reverse mechanical, electrical, or magnetic energy, takes irregular shapes, and they are concerned as the significant resource of that nonlinear behavior. The article introduces a nonlinear online Identification of a high-speed bidirectional DC motor with dead zone and Coulomb friction effect, which represent a primary nonlinear source, as well as viscosity forces. The Wiener block-oriented nonlinear system with neural networks are implemented to identify the nonlinear dynamic, mechatronic system. Online identification is adopted using the recursive weighted least squares (RWLS) method, which depends on the current and (to some extent) previous data. The identification fitness is found for various configurations with different polynomial orders, and the best model fitness is obtained about 98% according to normalized root mean square criterion for a third order polynomial.

Keywords: Coulomb friction, DC motor, dead zone, nonlinear online identification, neural networks, recursive weighted least squares, Wiener model.

ENGINEERING JOURNAL Volume 24 Issue 5
Received 22 April 2020
Accepted 3 August 2020
Published 30 September 2020
Online at https://engj.org/
DOI:10.4186/ej.2020.24.5.245

DOI:10.4186/ej.2020.24.5.245

1. Introduction

DC motors are the central unit of most manufacturing activities and functions. These motors are used in a wide-ranging field of implementations, for instance, mechanized movements, automated controls, robotic vehicles, servo structures, high-speed motors for drones, driving fluctuating loads, and functions that demand a controlled speed and positioning. The ease control and driving (for speed and torque), high precision, small size, superior power, and low cost are factors that make the DC motor be classified as the most convenient solution for industrial processes in comparison with other types of electrical motors [1, 2]. Precise modeling of DC motors is a necessity for efficient feedback model in many useful operations. The current hypotheses and models, which are frequently utilized in the books and by academics, do not consider all the existent effects. Likewise, the friction rates are not typically recognized before. It is essential to quantify the friction rates when the traditional modeling methods utilized or employ more advanced modeling procedures that are built on identifying dynamic-system to obtain an accurate model for a DC motor [3, 4].

Modeling a dynamic system is achieved by developing a mathematical model that determines interactions among the real input and output amounts that can be observed empirically. Numerous approaches have been formulated and improved in various applied fields concerning the modeling. In the Mechatronic engineering branch, the methods are identified beneath the name system identification [5, 6, 7]. The modeling system often starts with the dynamics of the system then tries to find parameters for that model, which generates estimations that minimize the error function, which is defined as a cost function, in a compromise interrelationship [8]. The mathematical model for a specific dynamic system can forecast the dynamic behavior of the real system as a response for a given input stimulus [9]. Most modeling processes have been carried out after acquiring the data altogether. Still, there is an alternative type of modeling done in real-time to assess the parameters of an individual system, is called online modeling system [10, 11, 12]. The online modeling system started with measured input (stimulation) and output signals in real-time. Some signals need to be filtered before processing them, which demands to throw from a low pass filter. The prepared data are passing through a recursive process that estimates the model parameters and error at that moment. Many engineering systems have a nonlinear behavior; they need a nonlinear modeling process. The systems nonlinearity comes from some effective coefficients that affect the response of the system to input signals, are unknown in system analysis, or ignored to simplify the calculation. In these systems, precise modeling is almost not essential, and the linearity that suggested is somewhat adequate. In the real system, many reasons motivate the researchers to build a nonlinear system to be more realistic, two of the crucial motivations are; the accurate control and right prediction for system performance [13, 14].

This paper presents a developed nonlinear model to identify the DC motor. The numerical methods are used to simplify the differential equations of the real system, which supports the discrete mathematics and time series. The nonlinear model is developed using the block-oriented Wiener model with neural networks to represent the nonlinear behavior in the DC motor. In the static nonlinear block of the Wiener model, the nonlinear neural function will be replaced with a high-order polynomial to be convenient with real-time estimation. RWLS method is used to train online the multi-layer neural networks (MLNN) with one shot for each time that the input/output data are available on the input ports of the controller come from the sensors.

2. Deriving a Nonlinear Dynamical Model

Many types of researches in the modeling of dynamic system motion have been written by using a linear approximation. Although the assumption of linearity does well with some of those systems, especially some types of the DC motor, it restricts the generality as a consequence, and it does not work with many systems [15].

2.1. Essential Linear Model

A piece of advanced knowledge about the system helps to represent the physical model with mathematical equations. In the DC motor case, to get a mathematical model that describes the system nonlinear behavior (which is near to real one), it is necessary to start with a fundamental configuration. It can be derived with some considerations that represent the boundary condition of the system, starting from Kirchhoff's voltage law, as shown in Fig. 1 below:

$$u = iR + L\frac{di}{dt} + bemf \tag{1}$$

and from Newton's second law for rotating parts:

$$T_m - T_L = J\frac{d\omega_m}{dt} \tag{2}$$

Where;

$$bemf = K\omega_m \tag{3}$$

and,

$$T_m = Ki \tag{4}$$

By substituting Eq. (3) and Eq. (4) in Eq. (1) and Eq. (2) respectively, and rearranging the last two equations, they will be:

$$\frac{di}{dt} = \frac{u}{L} - \frac{iR}{L} - \frac{K\omega_m}{L} \tag{5}$$

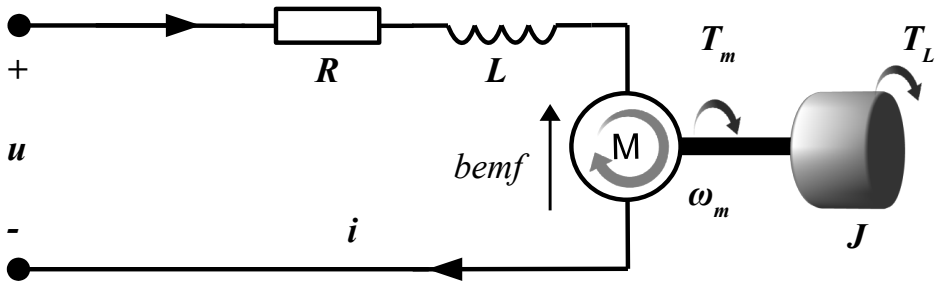

Fig. 1. Representative drawing of the DC motor and the driven load.

and,

$$\frac{d\omega_m}{dt} = \frac{Ki}{J} - \frac{T_L}{J} \tag{6}$$

Using a conventional notation and setting $x_1 = \omega_m$, $x_2 = i$, Eq. (6) and Eq. (5) will be:

$$\dot{x_1} = \frac{K}{J}x_2 - \frac{T_L}{J} \tag{7}$$

$$\dot{x_2} = -\frac{1}{L}(Kx_1 + Rx_2) + \frac{1}{L}u \tag{8}$$

Where the meaning of physical symbols as following:

u	input voltage (V)
i	armature current of motor (A)
R	total armature and field resistor (Ω)
L	total armature and field inductance (H)
$bemf$	back electromotive force (V)
T_m	the generated motor torque (N.m)
T_L	load torque (N.m)
J	associated moment of inertia for motor and load (Kg.m)
ω_m	angular speed for motor (rad/sec)
K	motor constant (N.m/A) or (V/rad.s^{-1})

2.2. Friction Effects and Non-linearity

Friction is one of the phenomena that enables human beings to live on the earth and do many of their activities such as standing, walking, swimming, using various transportation methods, and many manufacturing processes, etc. But on the other hand, friction has a role in decreasing net mechanical energy generally, machine wearing along duty cycle, and noisy hindering force in mechanical moving parts. One of the important effects is changing machine behavior, which affects system response to the external stimulant enormously, this effect can be regarded from zero effect (neglected) to a brutal level which is, in the main, against movement direction of machine parts.

In general, friction is a complex nonlinear motion resistance that occurs among surfaces or between two surfaces (in either lubricated or dry cases) that are in contact with each other when there is a relative motion among those surfaces. It could happen between two (or more) solid surfaces, or solid with the fluid environment due to friction of the solid part with molecules of that environment, or even among the molecules of the same substance. The resulting impedance force is named friction force, which has a contrary action line to the original motion direction [16].

Demonstrating the friction pattern is not an easy task in mechatronic machines. Friction's effect can have a tough influence on the system dynamics and can cause a complicated dynamic action. The effect of friction is especially important in dynamic system identification and deriving mathematical model issues. The applications subjected to online system identification and control often demand high accuracy and fast operation at the same time [3]. Non-linearity effect of friction is often ignored in DC motor modeling by most researchers, where accuracy is not a major concern. However, adding friction components is more appropriate to achieve a precision of dynamic system identification, and to give a general, real, mathematical formula.

The previous equations are derived by ignoring any effect for nonlinear forces that act on the mechanical parts of the DC motor. Still, when friction and viscosity of air gap between stator and armature, lubrication in bearings, and aerodynamic drag have been calculated, the non-linearity behavior will be revealed.

From Eq. (2):

$$T_m - T_f - T_v - T_L = J\frac{d\omega_m}{dt} \tag{9}$$

Where;

$$T_v = \nu\omega_m \tag{10}$$

and;

$$T_f = \mu\text{sgn}(\omega_m) \tag{11}$$

Where;

ν	viscosity coefficient (N.m/rad.s^{-1})
μ	friction anti-torque coeficient (N.m)
T_v	viscous friction (N.m)
T_f	Coulomb friction (N.m)

As the angular velocity increases, the T_v increases, and vice versa. While T_f is not affected with ω_m, but the sign of T_f changes with it. Figure 2 describes the relation between friction torque and angular velocity [1, 17].

DOI:10.4186/ej.2020.24.5.245

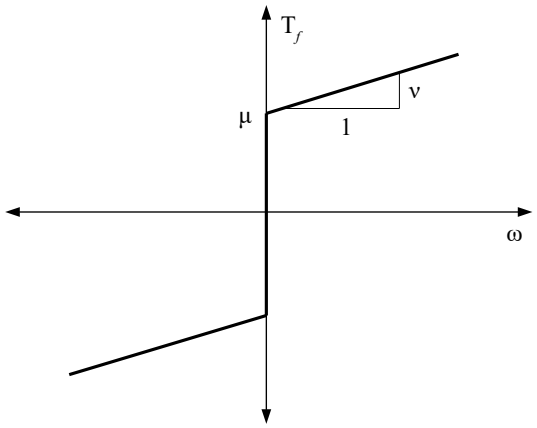

Fig. 2. The relation between friction torque and angular velocity.

The coefficient of viscosity is lower than the coefficient of Coulomb friction μ (μ is positive constant). The sgn (signum function) is identified as :

$$sgn(\omega_m) = \begin{cases} 1 & \omega_m > 0 \\ 0 & \omega_m = 0 \\ -1 & \omega_m < 0 \end{cases} \tag{12}$$

Now, it is possible to rewrite Eq. (1) and Eq. (9) to be:

$$u = iR + L\frac{di}{dt} + K\omega_m \tag{13}$$

$$J\frac{d\omega_m}{dt} = Ki - \mu\ sgn(\omega_m) - \upsilon\omega_m - T_L \tag{14}$$

In discrete-time, it can use a numerical method to evaluate the derivative terms in Eq. (13) and Eq. (14). Here, back difference method is used, as followed:

$$\frac{di}{dt} = \frac{i(t) - i(t-\tau)}{\tau} \tag{15}$$

and;

$$\frac{d\omega_m}{dt} = \frac{\omega_m(t) - \omega_m(t-\tau)}{\tau} \tag{16}$$

Where τ is a sampling time. Substituting the equivalent value to the current derivative from Eq. (15) into Eq. (13):

$$u(t) = Ri(t) + L\frac{(i(t) - i(t-\tau))}{\tau} + K\omega_m(t) \tag{17}$$

$$(R + \frac{L}{\tau})i(t) = u(t) + \frac{L}{\tau}i(t-\tau) - K\omega_m(t) \tag{18}$$

Assuming $L/\tau = \rho$ and $R + \rho = z$;

$$i(t) = \frac{1}{z}u(t) + \frac{\rho}{z}i(t-\tau) - \frac{K}{z}\omega_m(t) \tag{19}$$

Substituting Eq. (19) and Eq. (16) in Eq. (14), and let T_L be constant, the result will be:

$$\begin{aligned} &\frac{K}{z}u(t) + \frac{K\rho}{z}i(t-\tau) - \frac{K^2}{z}\omega_m(t) \\ &-\mu\ sgn(\omega_m) - \nu\ \omega_m(t) - T_L \\ &= \frac{J}{\tau}(\omega_m(t) - \omega_m(t-\tau)) \end{aligned} \tag{20}$$

Let the $\tau/J = \xi$;

$$\begin{aligned} &\frac{\xi K}{z}u(t) + \frac{\xi K\rho}{z}i(t-\tau) - \frac{\xi K^2}{z}\omega_m(t) \\ &- \xi\mu\ sgn(\omega_m) - \xi\nu\ \omega_m(t) - \xi T_L \\ &= \omega_m(t) - \omega_m(t-\tau) \end{aligned} \tag{21}$$

then;

$$\begin{aligned} &\frac{\xi K}{z}u(t) + \frac{\xi K\rho}{z}i(t-\tau) \\ &- \xi\mu sgn(\omega_m) - T_L + \omega_m(t-\tau) \\ &= (1 + \frac{\xi K^2}{z} + \xi\nu)\omega_m(t) \end{aligned} \tag{22}$$

Let $(1 + \frac{\xi K^2}{z} + \xi\nu) = r$, then substitute in Eq. (22).

$$\begin{aligned} \omega_m(t) = &\frac{\xi K}{rz}u(t) + \frac{\xi K\rho}{rz}i(t-\tau) \\ &- \frac{\xi\mu}{r}sgn(\omega_m) - \frac{T_L}{r} + \frac{1}{r}\omega_m(t-\tau) \end{aligned} \tag{23}$$

The $\omega_m(t)$ is a current output, the $u(t)$ is a current input, the $i(t-\tau)$ is a previous intermediate variable (here, it is current), The $sgn(\omega_m)$ is a nonlinear function, T_L is a load torque (here, it is considered constant), and $\omega_m(t-\tau)$ is a previous output. From the Eq. (23), it is possible to find the output equation before a period τ of time.

$$\begin{aligned} \omega_m(t-\tau) = &\frac{\xi K}{rz}u(t-\tau) + \frac{\xi K\rho}{rz}i(t-2\tau) \\ &- \frac{\xi\mu}{r}sgn(\omega_m) - \frac{T_L}{r} + \frac{1}{r}\omega_m(t-2\tau) \end{aligned} \tag{24}$$

Then, the output before a period 2τ of time will be:

$$\begin{aligned} \omega_m(t-2\tau) = &\frac{\xi K}{rz}u(t-2\tau) + \frac{\xi K\rho}{rz}i(t-3\tau) \\ &- \frac{\xi\mu}{r}sgn(\omega_m) - \frac{T_L}{r} + \frac{1}{r}\omega_m(t-3\tau) \end{aligned} \tag{25}$$

and so on, for s time before:

$$\begin{aligned} \omega_m(t-s\tau) = &\frac{\xi K}{rz}u(t-s\tau) + \frac{\xi K\rho}{rz}i(t-(s+1)\tau) \\ &- \frac{\xi\mu}{r}sgn(\omega_m) - \frac{T_L}{r} + \frac{1}{r}\omega_m(t-(s+1)\tau) \end{aligned} \tag{26}$$

Where $s \in \mathbb{N}$. Each equation is substituted in the previous one and the term $\omega_m(t-(s+1)\tau))$, which is the previous output for $(t-s\tau)$ moment because it will be far from the current instant and almost has no effect, is ignored. Then, it could assume the value of τ equals to one, and it can avoid

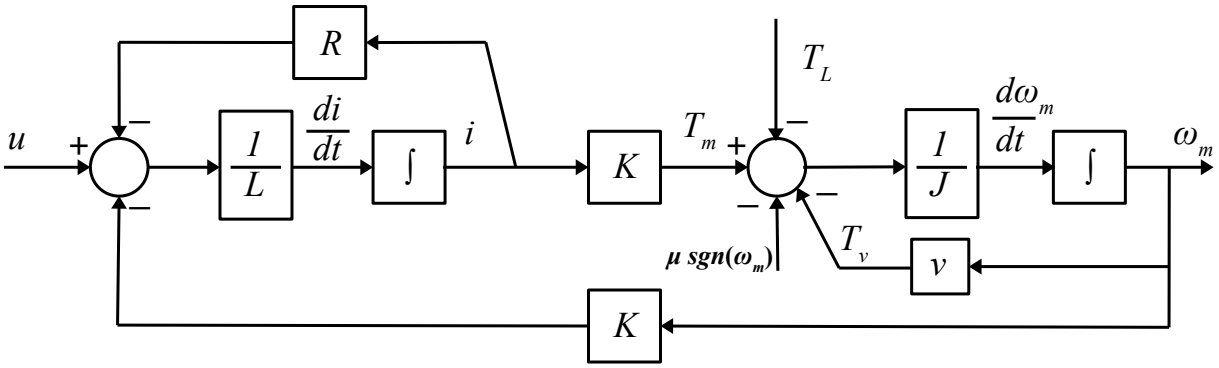

Fig. 3. Block-diagram of DC motor.

the present term of input because it does not get into the system practically. So the Eq. (23) can be written as:

$$\omega_m(t) = \sum_{h=1}^{s} b_h u(t-h) + \sum_{j=1}^{s+1} a_j i(t-j) - f(sgn(\omega_m)|T_L) \tag{27}$$

The j is greater or equal to h. The $f(.)$ is a nonlinear function, depends on sgn function and load torque, which is constant here. b_h is the factor of $u(t-h)$ term calculated such as:

$$b_h = \frac{\xi K}{z r^{1+h}}; \quad h = 1, 2, 3, ..., s \tag{28}$$

a_j is the factor of $x(t-h)$ term is calculated as:

$$a_j = \frac{\xi K \rho}{z r^{j}}; \quad j = 1, 2, 3, ..., s+1 \tag{29}$$

The i variable (direct current) can be changed with x as intermediate variable and the ω_m with the y variable as in traditional references.

$$y(t) = \sum_{h=1}^{s} b_h u(t-h) + \sum_{j=1}^{s+1} a_j x(t-j) - nonlinear \tag{30}$$

3. Non-linear Wiener Model

In general, most non-linear systems can be described by the Wiener model [18], Hammerstein model [19], or Hammerstein-Wiener combined model [20]. The Wiener model, as shown in Fig. 4, can be described by two sequential blocks; the first is a dynamic linear block (that means in addition to present values of input/output, it relies on the previous amount of the input/output as well), and the second is a static nonlinear block.

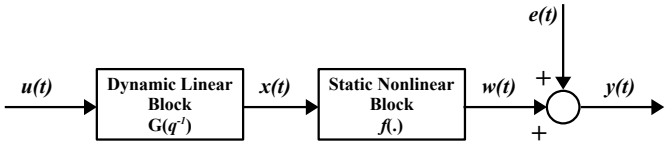

Fig. 4. Wiener model.

For the single-input single-output system, Wiener dynamic block can be defined as:

$$x(t) = G(q^{-1})u(t) = \frac{B(q^{-1})}{A(q^{-1})}u(t) \tag{31}$$

Where;

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{ma} q^{-ma}$$
$$B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3} + \cdots + b_{mb} q^{-mb}$$

The A and B are polynomials of delay (q^{-1}), ma and mb are the order of the polynomial, respectively. In general ($ma \geq mb$).

$$x(t) = -a_1 x(t-1) - \cdots - a_{ma} x(t-ma) + b_1 u(t-1) + \cdots + b_{mb} u(t-mb) \tag{32}$$

The Eq. (32) demonstrates the dynamic relationship between the input and the intermediate signals. The nonlinear static block will be:

$$y(t) = w(t) + e(t) = f(x(t)) + e(t) \tag{33}$$

$y(t)$, $u(t)$, and $x(t)$ are output, input and intermediate variables severally. $x(t)$ has no physical meaning. The error $e(t)$ is regarded as white noise (uncorrelated sequential samples of error) with a zero-mean random sequence of finite variance, and independent of $x(t)$.

$$E(e(t)) = 0\ ; \quad \text{for variance } \sigma^2 < \infty \tag{34}$$

So, according to this assumption, Fig. 4 could be compacted to be:

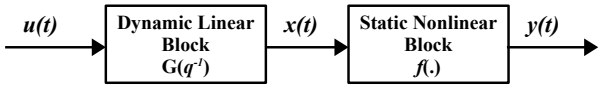

Fig. 5. Wiener model with zero-mean error assumption.

The output accordingly will be changed to be:

$$y(t) = f(x(t)) \tag{35}$$

There are some approaches to represent the nonlinear function in the static nonlinear block using; neural nonlinear activation functions, piece-wise linear method, or polynomial representation. Besides, many methods have been developed to evaluate the whole parameters of the Wiener model, such as the basic correlation methods, linear and nonlinear optimization methods [21].

Some nonlinear representations are unsuitable for online nonlinear system identification that has unknown (part or all) parameters of that system.

Using the nonlinear activation function has a limitation because the output of these functions is limited to +1 as maximum and -1 as minimum value, $tanh$, for example. So, it is necessary for adding another layer to the neural network and pure linear function with its bias and weights to get real numbers larger than maximum or lease than minimum limits [22]. As a result, the time of calculation will increase, and the weights will be correlated with each other in a non-separable form.

The piece-wise linear method is used for following a polygonal route separated by hinge points. The essential weak point of this method is required the input and output data to be available, or at least, the hinge points are determined [23]. The data availability discords with online system identification essence, which depends on prediction.

The polynomial fitting method is used to forge a nonlinear relationship between the independent input and dependent output variables as nth degree multi-term of input. It has the flexibility to deal with many physical systems and in prediction models by least-squares method help or one of its family [24].

4. Wiener Neural Network

Generally, in neural networks, the network has no physical meaning or real description for the original system or process, and the parameters of network or weights have no physical explanation. It is near to the black box structure. Relationships between the Eq. (32) and Eq. (35) configure the shape and properties of the Wiener neural network. It contains two nodes in Three layers. A dynamic linear part with two attached delay series one for input and another for intermediate value as input for this node and the output of this node represents the variable value of the polynomial that represents a static non-linear part with k lines as shown in the Fig. 6. Each term of the polynomial is multiplied by the corresponding weight.

So, the hidden layer can be expressed as:

$$x(t) = -\sum_{i=1}^{ma} a_i x(t-i) + \sum_{j=1}^{mb} b_j u(t-j) \tag{36}$$

It is used to be represented in terms of parameters vector and input-intermediate delay variables.

$$x(t) = \boldsymbol{\theta_h^T \varphi_h(t)} \tag{37}$$

Where;

$$\boldsymbol{\theta_h^T} = (a_1 \cdots a_{ma} \;\; b_1 \cdots b_{mb})$$

And,

$$\boldsymbol{\varphi_h^T(t)} = \left[-x(t-1) \cdots -x(t-ma) \;\; u(t-1) \cdots u(t-mb)\right]$$

Where $\boldsymbol{\theta_h}$ and $\boldsymbol{\varphi_h}$ are weights and input variable vectors of hidden layer respectively, and the subscription h refers to the hidden layer. The output layer will be:

$$\hat{y}(t) = \sum_{r=1}^{k} c_r x^r(t) \tag{38}$$

it means;

$$\begin{aligned} \hat{y}(t) = c_1 x(t) + c_2 x^2(t) + \cdots \\ + c_{k-1} x^{k-1}(t) + c_k x^k(t) \end{aligned} \tag{39}$$

it could represent it with vector term, to be:

$$\hat{y}(t) = \boldsymbol{\theta_o^T \varphi_o(t)} \tag{40}$$

Where;

$$\boldsymbol{\theta_o^T} = (c_1 \; c_2 \cdots c_{k-1} \; c_k)$$

And,

$$\boldsymbol{\varphi_o^T(t)} = \left(x^1(t) \; x^2(t) \cdots x^{k-1}(t) \; x^k(t)\right)$$

Where $\boldsymbol{\theta_o}$ and $\boldsymbol{\varphi_o}$ are weights of output layer and hidden-output variable vectors respectively, and the subscription o refers to the output layer. In this case, the training algorithm must be implemented through two steps, as in [25, 26], the first step is estimating the weights of output layer Eq. (40), then estimate the weights of hidden layer Eq. (37).

To speed up the training process, and make it one-shot teaching for the whole network at a time. There is a step that must be taken in this way—referring to Fig. 6, it is possible to consider the weigh of the first polynomial term, c_1, equal to unity; however, it will not lose the feature of a multi-layer network. New form for output, after substituting Eq. (32) in Eq. (39), will be:

$$\begin{aligned} \hat{y}(t) = -a_1 x(t-1) - \cdots - a_{ma}(t-ma) + b_1 u(t-1) \\ + \cdots + b_{ma} u(t-mb) + c_2 x^2(t) + \cdots + c_k x^k(t) \end{aligned} \tag{41}$$

$$\hat{y}(t) = \boldsymbol{\theta^T \varphi(t)} \tag{42}$$

Where;

$$\boldsymbol{\theta^T} = (a_1 \cdots a_{ma} \; b_1 \cdots b_{mb} \;\; c_2 \cdots \; c_k) \tag{43}$$

And,

$$\begin{aligned} \boldsymbol{\varphi^T(t)} = \left[-x(t-1) \cdots - x(t-ma)\right. \\ \left. u(t-1) \cdots u(t-mb) \;\; x^2(t) \cdots \; x^k(t)\right] \end{aligned} \tag{44}$$

Where $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ are parameters, and variable vectors respectively for the whole network. The Eq. (41) is extremely similar to the Eq. (30) which has been derived previously.

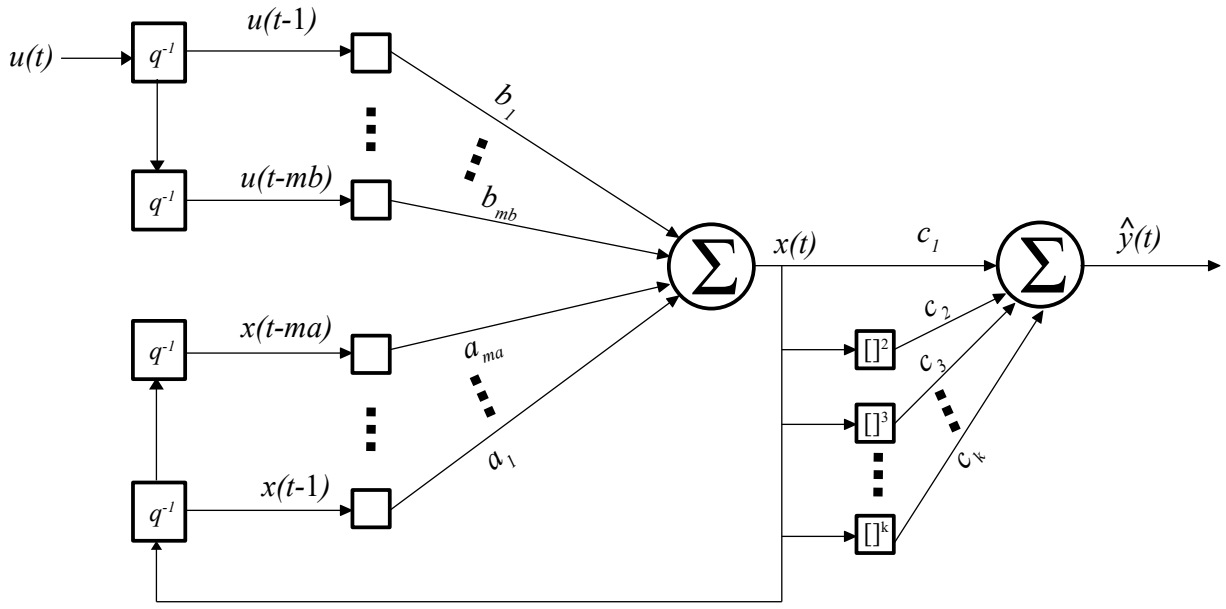

Fig. 6. Wiener neural network.

5. Teaching Algorithm

In this work, an algorithm of the teaching process is based on the weighted least squares method. The basic idea of this process built on minimizing-error, which is the essence of the least-squares error method, as shown:

$$F_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \hat{y}(t)]^2 \tag{45}$$

Where F_N is the least-squares function for N points (readings), $y(t)$ is the desired or real output, and $\hat{y}(t)$ is the estimated output and equal to $\boldsymbol{\varphi^T\theta}$. Each error point of the N points in the Eq. (45) has the same weight of effect. Whereas the weighted-least squares (WLS) method gives for each error point a different weight according to its impact; where points are near to a regression line have a bigger weight than those which are far away from the regression line (or the sensitive noise readings) [27], as shown in the Eq. (46).

$$F_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\alpha_t[y(t) - \boldsymbol{\varphi^T(t)\theta}]^2 \tag{46}$$

Which α a scalar value varies for each single error point. It is feasible to rewrite the Eq. (46) in discrete time symbols, as shown in the Eq. (47).

$$F_N(\theta) = (Y_N - \Phi_N\theta)^T Q_N (Y_N - \Phi_N\theta) \tag{47}$$

To estimate the parameters at the smallest error (minimum error). First, deriving the right side of Eq. (47) relative to θ (parameter vector), and then solving for θ after equaling the left side to zero. That is leading to Eq. (48).

$$\hat{\theta}_N = [\Phi_N^T Q_N \Phi_N]^{-1}\Phi_N^T Q_N Y_N \tag{48}$$

It is possible to make the (WLS) deals with streaming data recursively to be meet the online modeling requirements, by using recursive weighted least squares (RWLS) [28, 29]. In the beginning, it is suitable to rewrite the Eq. (46) in discrete-time symbolic terms (k) and present time (t), as in Eq. (49).

$$\hat{\theta}_t = \arg_\theta \min \sum_{k=1}^{t}\beta(t,k)[y(k) - \varphi^T(k)\theta]^2 \tag{49}$$

Which means, deriving the equation relative to θ and equating it to zero to get the minimum value. β is the weight of a specific point at k time to the present time [27, 30]. The Eq. (49) will be:

$$\hat{\theta} = \Big[\sum_{k=1}^{t}\beta(t,k)\varphi(k)\varphi^T(k)\Big]^{-1}\sum_{k=1}^{t}\beta(t,k)\varphi(k)y(k) \tag{50}$$

It is more convenient to minimize the terms, to be:

$$R(t) = \sum_{k=1}^{t}\beta(t,k)\varphi(k)\varphi^T(k) \tag{51}$$

$$f(t) = \sum_{k=1}^{t}\beta(t,k)\varphi(k)y(k) \tag{52}$$

So, the Eq. (50) will be:

$$\hat{\theta}_t = R^{-1}(t)f(t) \tag{53}$$

Now, suppose that the successive weight has the following property:

$$\begin{aligned}&\beta(t,k) = \lambda(t)\beta(t-1,k), \quad 0 \le k \le t-1\\ &\beta(t,t) = 1\end{aligned} \tag{54}$$

That leads to:

$$\beta(t,k) = \prod_{k+1}^{t} \lambda(i) \tag{55}$$

If the Eq. (54) was substituted in the Eq. (51) and Eq. (52), the result will be:

$$R(t) = \lambda(t)R(t-1) + \varphi(t)\varphi^T(t) \tag{56}$$

$$f(t) = \lambda(t)f(t-1) + \varphi(t)y(t) \tag{57}$$

That leads to:

$$\begin{aligned}\hat{\theta}_t &= R^{-1}(t)f(t) = R^{-1}(t)\big[\lambda(t)f(t-1) + \varphi(t)y(t)\big] \\ &= R^{-1}(t)\big[\lambda(t)R(t-1)\hat{\theta}_{t-1} + \varphi(t)y(t)\big] \\ &= R^{-1}(t)\Big\{\big[R(t) - \varphi(t)\varphi^T(t)\big]\hat{\theta}_{t-1} + \varphi(t)y(t)\Big\}\end{aligned}$$

So, the result is:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + R^{-1}(t)\varphi(t)\big[y(t) - \varphi^T(t)\hat{\theta}_{t-1}\big] \tag{58}$$

Which is a recursive equation. To avoid finding the inverse of $R(t)$ recursively, it is suitable to use:

$$P(t) = R^{-1}(t) \tag{59}$$

Then implementing the matrix inversion lemma.

$$[A+BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \tag{60}$$

Substituting in Eq. (56), by considering $A = \lambda(t)R(t-1)$, $B = D^T = \varphi(t)$, and $C = 1$, this leads to:

$$P(t) = \frac{1}{\lambda(t)}\left[P(t-1) - \frac{P(t-1)\phi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{61}$$

Furthermore, the inverting matrix multiplied by variables vector will be:

$$\begin{aligned}R^{-1}(t)\varphi(t) &= \frac{1}{\lambda(t)}P(t-1)\varphi(t) \\ &\quad - \frac{1}{\lambda(t)}\frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \\ &= \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\end{aligned} \tag{62}$$

Summarizing of the whole recursive algorithm in the sequence is as follows:

$$P(t) = \frac{1}{\lambda(t)}\left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{63}$$

$$L(t) = R^{-1}(t)\varphi(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \tag{64}$$

$$\hat{\theta}_t = \hat{\theta}(t-1) + L(t)\big[y(t) - \varphi^T(t)\hat{\theta}(t-1)\big] \tag{65}$$

6. Online Modeling Steps

The following steps explain how to implement the recursive method to estimate the model in real-time:

1. Initialize the covariance matrix P for the whole network.
2. Initialize the estimated weight-matrix or parameter-matrix $\hat{\theta}$ for each layer in the network.
3. Select the weighting factor in the range $0 < \lambda < 1$.
4. Read the input u and calculating the intermediate variable x.
5. Read the output y and calculating the estimated output $\hat{y}$.
6. Form the variable matrix φ for the present time t as in Eq. (44).
7. Calculate the error square e^2.
8. Calculate the matrix of the estimated parameters $\hat{\theta}$ through calculating the Eq. (63) — Eq. (65).
9. Update a covariance matrix P and the estimated weight-matrix $\hat{\theta}$.

Step 4 — step 9 are executed recursively.

7. Implementation of Model

The high-speed DC motor physical model is built with the load by the Simscape™ software, which is an extension for MATLAB® language. The input is a noisy sinusoidal signal. The output is high speed in (rpm) for relatively small input in (voltage). The motor is constructed with minimal Coulomb friction torque about 0.02×10^{-3} N.m with no viscous friction coefficient. The weighting factor λ is taken equal to 0.90; this is the best factor consists of two digits after the decimal period for this system and some other systems [31, 32, 33]. Figure 7 demonstrates the input voltage and the output speed of the DC motor in (rpm).

8. Experiment Methodology

Most experiments have appeared that the model converges from the real system after 230 ms. The sampling time for reading input/output data is 10 ms, which is about 4.3 % of convergence time. The input voltage varies from 7 to −7 volts around zero volts, to examine the non-linearity of the dead zone and Coulomb friction in the bidirectional DC motor and their impact on the modeled system. The experiments have been done to determine the number of terms, that represent the best model, in each u, x, and x^n polynomial in the Eq. (42), Eq. (43), and Eq. (44). Where u is the number of dynamic input terms, for example $u = 2$;

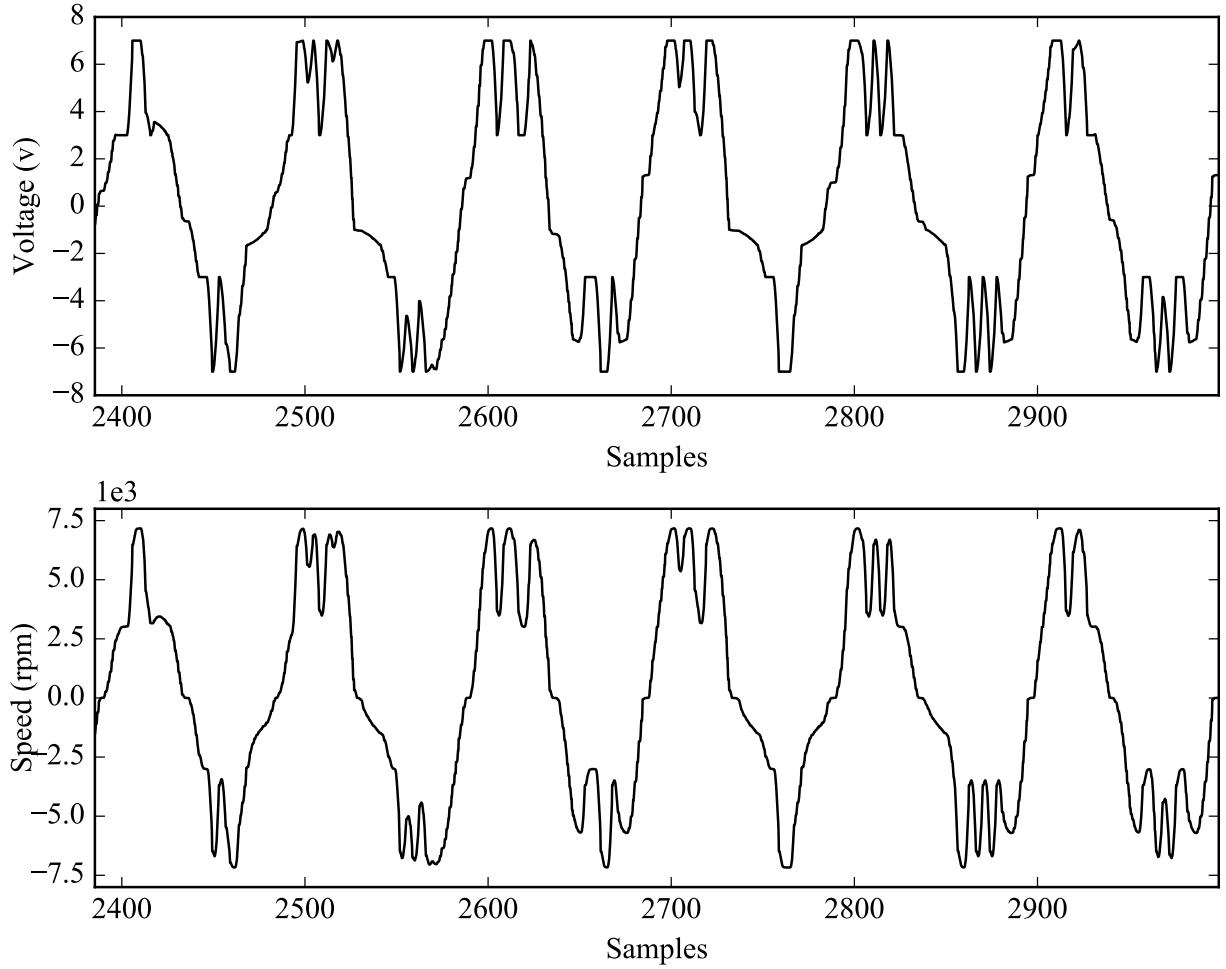

Fig. 7. Input voltage and motor speed.

it means $b_1u(t-1)+b_2u(t-2)$), x is the number of dynamic feedback of first stage, for example $x=3$; it is means $a_1x(t-1)+a_2x(t-2)+a_3x(t-3)$, and x^n is the order polynomial; for example $x^n=3$; it means $c_2x^2(t)+c_3x^3+c_4x^4$, the x^n=1 is a trivial case [2], where the set of x^n is $\{x^2,\ x^3,\ x^4,\dots\}$ starts from x^2. The number and best distribution for terms on the u, x, and x^n variables are determined through experiments using 15 terms as a maximum for all variables. In comparison, the number of the maximum terms for a specific variable does not exceed 11 terms. More than forty combinations have been examined as samples, which is more than most other researches [34, 22], the minimum number of terms is six terms two terms for each variable. The x^n series specifies the degree of non-linearity, and it is responsible for determining the order of the model.

8.1. Result and Model Evaluation

There are some methods to measure the validity of the model in the modeling system field. Most two methods used frequently are based on mean square error (MSE), and Normalized root-mean-square error (NRMSE) [2, 35]; the MSE is appropriate for small scale signals, but NRMSE is used for various scales including large scale signals [35]. Here, the goodness of fit (gof) is used built on the NRMSE criterion as a cost function, as shown in Eq. (66).

$$gof=\left[1-\frac{\sqrt{\sum_{t=1}^{N}(y(t)-\hat{y}(t))^2}}{\sqrt{\sum_{t=1}^{N}(y(t)-\bar{y}(t))^2}}\right]\times 100\% \quad (66)$$

Where the $\hat{y}(t)$ is the estimated output from the model and $\bar{y}(t)$ is the mean value of the original output of the motor $y(t)$. Table 1 clarifies the gof, which is depicted graphically by Fig. 8, for forty-one samples and the relation with the number of terms for u, x, and x^n variables. Where the best model has two terms of dynamic inputs, five dynamic feedback terms, and two x^n terms as shown in Eq. (67); the order polynomial = x^n+1; means number of terms + 1, here is (3). The NRMSE criterion was about 98% as shown in Table 1.

DOI:10.4186/ej.2020.24.5.245

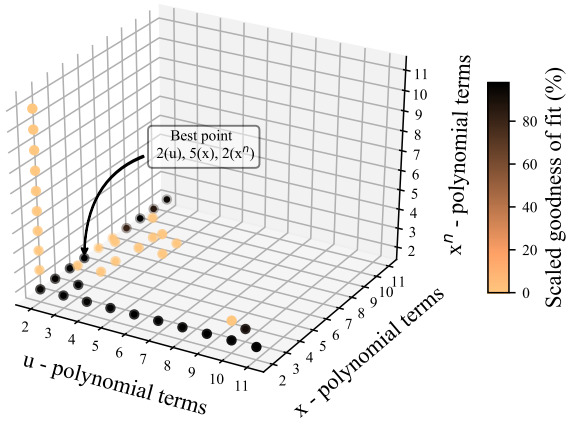

Fig. 8. Goodness of fit (gof) values for different combinations of u, x, and x^n polynomials.

Figure 9 and Fig. 10 show the difference between the real output for DC motor and the estimated output for the model, and display the tracking of the suggested model to the real system. Figure 11 depicts the error of estimated output relative to the real system $e = y - \hat{y}$ along the working period.

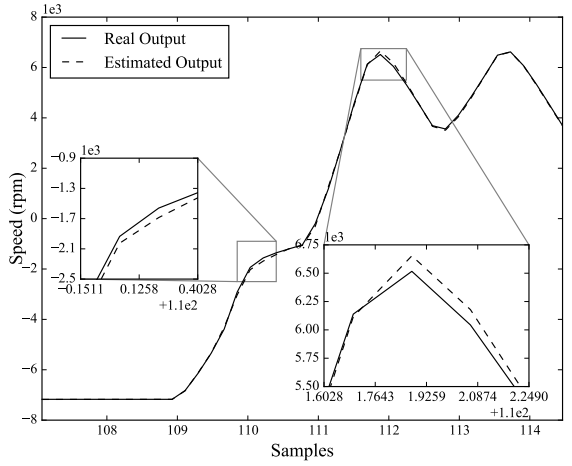

Fig. 9. Difference between real system and the model of the system.

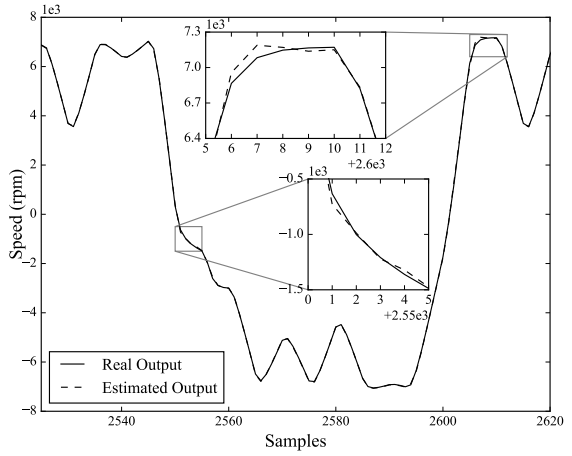

Fig. 10. Difference between real system and the model of the system.

The closest model for the DC motor, as shown in Table 1 and Fig. 8, is:

$$\begin{aligned}\hat{y}(t) = {} & a_1x(t-1) + a_2x(t-2) \\ & + a_3x(t-3) + a_4x(t-4) \\ & + a_5x(t-5) + b_1u(t-1) \\ & + b_2u(t-2) + c_2x^2(t) + c_3x^3(t)\end{aligned} \tag{67}$$

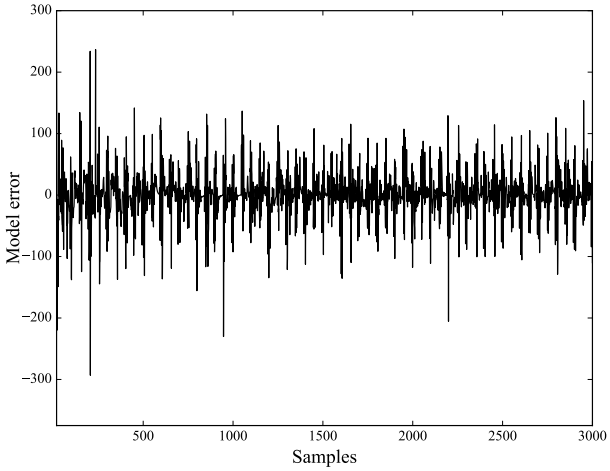

Fig. 11. The error of the estimated model.

According to the Eq. (41) and as it was clarified by Fig. 6, the x^n polynomial is starting with c_2x^2, so the two terms of this polynomial will be $c_2x^2 + c_3x^3$ at the specific time t. The figures from Fig. 12 to Fig. 18 depict the estimated parameters for online identification. After substituting the values of parameters in the Eq. (67), it will be as follows in Eq. (68):

$$\begin{aligned}\hat{y}(t) = {} & 0.2268x(t-1) + 0.0284x(t-2) \\ & - 0.0133x(t-3) + 0.0047x(t-4) \\ & + 0.0011x(t-5) + 487.3654u(t-1) \\ & + 380.7769u(t-2) - 2.8936e-7x^2(t) \\ & + 2.1245e-10x^3(t)\end{aligned} \tag{68}$$

The Eq. (68) shows the significant effect of the preceding input terms, $u(t-1)$ and $u(t-2)$, over the rest terms of the polynomial, and the small effect of the nonlinear terms of polynomial because of the minimal value of Coulomb friction torque that used in building the physical model of DC motor in the Simscape/MATLAB®.

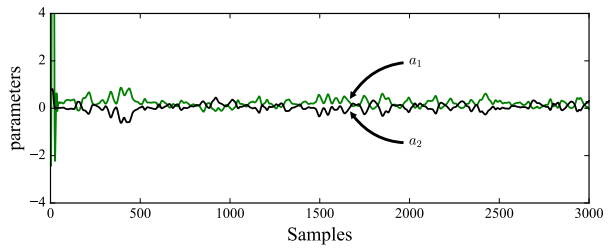

Fig. 12. Estimated a_1 and a_2 parameters.

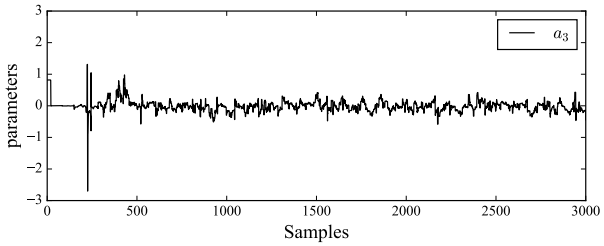

Fig. 13. Estimated a_3 parameter.

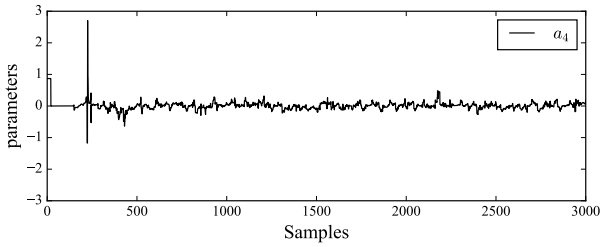

Fig. 14. Estimated a_4 parameter.

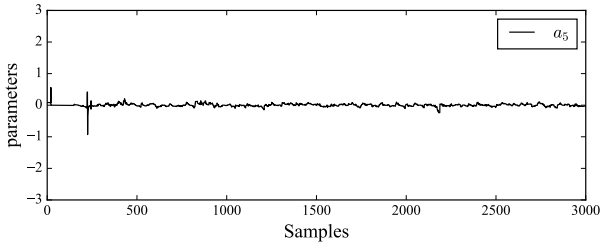

Fig. 15. Estimated a_5 parameter.

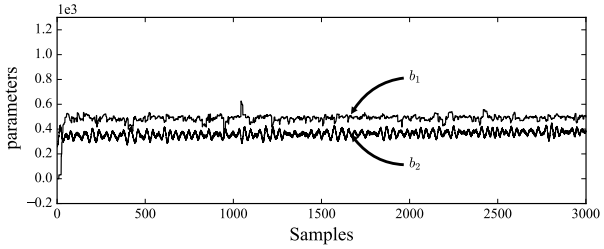

Fig. 16. Estimated b_1 and b_2 parameters.

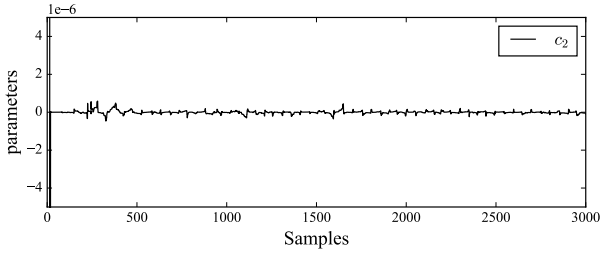

Fig. 17. Estimated c_2 parameter.

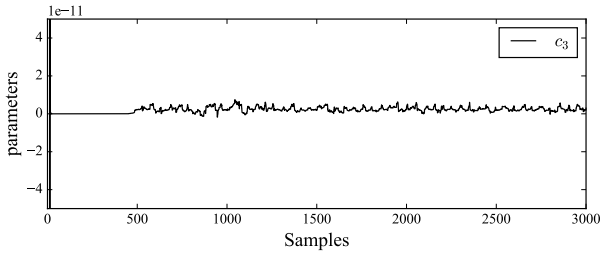

Fig. 18. Estimated c_3 parameter.

9. Conclusion

The high-speed bidirectional DC motor has been modeled as a nonlinear online model, taking into account the nonlinear effects of the Coulomb friction, dead zone, and the viscous torque. The model developed from the principle equation using the neural networks with time-shifting and the Wiener block-oriented model. The RWLS method is used to update the weights of the net. The physical model is built by Simscape/MATLAB® and examined online. The best model was third order polynomial with fitness 98.03 % using NRMSE as a criterion function. This research and its methodology focus on setting general rules for dealing with modeling mechatronic devices and implementing various techniques for identifying nonlinear systems. The experiments show promising results that can be applied to identify other systems in the real world in real-time.

Table 1. The goodness of fit (gof) values for different combinations of u, x, and x^n polynomials.

u	2	2	2	2	2	2	2	2	
x	2	2	2	2	2	2	2	2	
x^n	2	3	4	5	6	7	8	9	
gof	95.8596	-inf	-inf	-inf	-inf	-inf	-inf	-inf	
u	2	2	2	2	**2**	2	2	2	
x	2	2	3	4	**5**	6	7	8	
x^n	10	11	2	2	**2**	2	2	2	
gof	-inf	-inf	96.37867	97.9527	**98.0311**	-474.896	1.8404	80.2049	
u	2	2	2	3	3	3	4	4	
x	9	10	11	2	3	3	2	3	
x^n	2	2	2	2	2	3	2	3	
gof	97.7863	85.6256	93.9708	97.9429	97.7765	-inf	95.5151	-inf	
u	4	4	5	5	5	5	6	6	
x	4	4	2	4	5	5	2	4	
x^n	3	4	2	4	4	5	2	4	
gof	-inf	-inf	97.8157	-inf	-inf	-inf	97.9821	-inf	
u	6	6	7	8	9	10	10	10	11
x	4	5	2	2	2	2	2	3	2
x^n	5	4	2	2	2	2	3	2	2
gof	-inf	-inf	97.9625	97.4642	97.9331	97.9136	-inf	-inf	-inf

References

[1] I. Virgala and M. Kelemen, "Experimental friction identification of a dc motor," *International journal of mechanics and applications*, vol. 3, no. 1, pp. 26–30, 2013.

[2] T. Kara and I. Eker, "Nonlinear modeling and identification of a dc motor for bidirectional operation with real time experiments," *Energy Conversion and Management*, vol. 45, no. 7-8, pp. 1087–1106, 2004.

[3] C. Budai, "Friction effects in mechanical system dynamics and control," 2017.

[4] A. Milovanovic, M. Bjekić, and S. Antic, "Permanent magnet dc motor friction measurement and analysis of friction's impact," *International Review of Electrical Engineering*, vol. 6, pp. 2261–2269, 10 2011.

[5] A. M. Kwad, D. Hanafi, R. Omar, and H. Abdul Rahman, "Development of system identification from traditional concepts to real-time soft computing based," *IOP Conference Series: Materials Science and Engineering*, vol. 767, p. 012050, mar 2020.

[6] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of hammerstein–wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[7] D. Hanafi, M. S. Huq, M. S. Suid, and M. F. Rahmat, "A quarter car arx model identification based on real car test data," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 2-5, pp. 135–138, 2017.

[8] T. Sands, "Nonlinear-adaptive mathematical system identification," *Computation*, vol. 5, no. 4, p. 47, 2017.

[9] C. M. Pappalardo and D. Guida, "System identification algorithm for computing the modal parameters of linear mechanical systems," *Machines*, vol. 6, no. 2, p. 12, 2018.

[10] W. Plaetinck, D. Pool, M. Van Paassen, and M. Mulder, "Online identification of pilot adaptation to sudden degradations in vehicle stability," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 347–352, 2019.

[11] E. Y. Hong, T. K. Meng, and M. Chitre, "Online system identification of the dynamics of an autonomous underwater vehicle," in *Underwater Technology Symposium (UT), 2013 IEEE International*. IEEE, 2013, pp. 1–10.

[12] Q. Yang, X. Ren, W. Zhao, Y. Lv, and S. Wang, "Optimal adaptive parameter estimation-based tracking control of motor turntable servo system," in *Modelling, Identification and Control (ICMIC), 2017 9th International Conference on*. IEEE, 2017, pp. 139–144.

[13] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[14] J. Jayachandran and D. Ashok, "Neural network based approach for the generation of road feel in a steer-by-wire system," *Engineering Journal*, vol. 20, no. 5, pp. 49–66, 2016.

[15] S. E. Lyshevski, "Nonlinear control of mechatronic systems with permanent-magnet dc motors," *Mechatronics*, vol. 9, no. 5, pp. 539–552, 1999.

[16] Y. Berthier, "Handbook of materials behavior models," in *Background on friction and wear*. Lemaître Academic Press, 2001, pp. 676–699.

[17] G. Phanomchoeng and R. Chancharoen, "Hybrid motor system for high precision position control of a heavy load plant," *Engineering Journal*, vol. 23, no. 6, pp. 161–173, 2019.

[18] T. R. Khalifa, A. M. El-Nagar, M. A. El-Brawany, E. A. El-Araby, and M. El-Bardini, "A novel fuzzy wiener-based nonlinear modelling for engineering applications," *ISA transactions*, vol. 97, pp. 130–142, 2020.

[19] S. Khankalantary, S. Rafatnia, and H. Mohammadkhani, "An adaptive constrained type-2 fuzzy hammerstein neural network data fusion scheme for low-cost sins/gnss navigation system," *Applied Soft Computing*, vol. 86, p. 105917, 2020.

[20] V. Cerone, V. Razza, and D. Regruto, "One-shot set-membership identification of generalized hammerstein–wiener systems," *Automatica*, vol. 118, p. 109028, 2020.

[21] J. Peng, R. Dubay, J. M. Hernandez, and M. Abu-Ayyad, "A wiener neural network-based identification and adaptive generalized predictive control for nonlinear siso systems," *Industrial & Engineering Chemistry Research*, vol. 50, no. 12, pp. 7388–7397, 2011.

[22] A. Tavakolpour-Saleh, S. Nasib, A. Sepasyan, and S. Hashemi, "Parametric and nonparametric system identification of an experimental turbojet engine," *Aerospace Science and Technology*, vol. 43, pp. 21–29, 2015.

[23] E. L. Allgower and K. Georg, "Piecewise linear methods for nonlinear equations and optimization," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 245–261, 2000.

[24] J. Gergonne, "The application of the method of least squares to the interpolation of sequences," *Historia Mathematica*, vol. 1, no. 4, pp. 439–447, 1974.

[25] J. Peng and R. Dubay, "Identification and adaptive neural network control of a dc motor system with dead-zone characteristics," *ISA transactions*, vol. 50, no. 4, pp. 588–598, 2011.

[26] K. M. Passino, *Biomimicry for optimization, control, and automation*. Springer Science & Business Media, 2005.

[27] L. Ljung, Ed., *System Identification (2Nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[28] Y. Yu, H. Zhao, R. C. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1566–1581, 2019.

[29] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină, "Recursive least-squares algorithms for the identification of low-rank systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 5, pp. 903–918, 2019.

[30] R. Isermann and M. Münchhof, *Identification of dynamic systems: an introduction with applications*. Springer Science & Business Media, 2010.

[31] S. Song, J.-S. Lim, S. Baek, and K.-M. Sung, "Gauss newton variable forgetting factor recursive least squares for time varying parameter tracking," *Electronics letters*, vol. 36, no. 11, pp. 988–990, 2000.

[32] P. Khunabut, S. Kunaruttanapruk, P. Tansongcharoen, and S. Jitapunkul, "Rls channel estimation with forgetting factor adaptation for the downlink of mc-cdma system," in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 2. IEEE, 2004, pp. 1160–1164.

[33] Y. Lu, Q. Li, Z. Pan, and S. Y. Liang, "Prognosis of bearing degradation using gradient variable forgetting factor rls combined with time series model," *IEEE Access*, vol. 6, pp. 10 986–10 995, 2018.

[34] D. Aryani, L. Wang, and T. Patikirikorala, "On identification of hammerstein and wiener model with application to virtualised software system," *International Journal of Systems Science*, vol. 48, no. 6, pp. 1146–1161, 2017.

[35] H. Muroi and S. Adachi, "Model validation criteria for system identification in time domain," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 86–91, 2015.

DOI:10.4186/ej.2020.24.5.245

Ayad Mahmood Kwad was born in Al-Rusafa, Baghdad, Iraq in 1978. He received the B.S. and M.S. degrees in Mechatronic engineering from the University of Baghdad, Engineering college in 2001 and Al-Khawarizmi Engineering college 2010 successively, and he started the Ph.D. degree in Mechatronic, electrical engineering department in University of Tun Hussein Onn Malaysia (UTHM), Malaysia in 2017. From 2013 to 2017, he was a Lecturer Assistant in engineering college, Al-Iraqia University, Baghdad, Iraq. He is the author of *Two-axes sun tracking system the theory and design* book in 2013. His research interests include control of mechatronic systems, embedded systems and microcontrollers, system identification development and applications.

Dirman Hanafi was born in Agam, Indonesia, in 1967. He received the B.Eng. degree in electrical engineering from the University of Bung Hatta, Padang, Indonesia, in 1994, and the M. Eng. degree in Control and Instrumentation Program sandwich program between Bandung Institute of Technology (ITB) Indonesia and FH Karlsruhe, Germany, in 1998. And Ph.D. degree in Mechatronic Engineering (specialized in Control and Instrumentation Engineering) from the Department of Control and Robotic Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM), Malaysia in 2006. In 1998, he joined the Department of Electrical Engineering, University of Bung Hatta, as a Lecturer. Since May 2008, he has been with the Department of Mechatronic and Robotic Engineering, University Tun Hussein Onn Malaysia, an Assistant Professor, became an Associate Professor in 2015. July 2019 he has been with the Department of Electrical Engineering, University Tun Hussein Onn Malaysia an Associate Professor. His current research interests include in the field of system identification and Intelligent Control for applications in vehicle suspension systems, robotics, and electrical power plants.

Rosli Omar currently is an Associate Professor and Dean at the Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia. He received his PhD in engineering from the University of Leicester, the United Kingdom in 2012. He is a highly motivated academician with more than 15 years of teaching and research experience. His research interests are in robotic engineering, autonomous systems and system identification.

Hisyam Abdul Rahman received the Ph.D. degree in Mechatronic Engineering (specialized in Biomedical Engineering) from the Department of Control and Robotic Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM), Malaysia in 2016. Currently, he is working at University Tun Hussein Onn Malaysia (UTHM) as a lecturer and researcher. His research interest is in robotic and rehabilitation after stroke injury.