SELF-TUNING DC MOTOR DESIGN BASED ON RADIAL BASIS FUNCTION NEURAL NETWORK

Gaber Elsady¹, Abd-el Meged Mohamed², Ashraf Hemeda³, Asmaa Fawzy³

¹Faculty of Engineering, Assiut University, Egypt
²Faculty of Engineering, Aswan University, Egypt
³Faculty of Energy Engineering, Aswan University, Egypt

This paper introduces the inverse control design using neural network based self tuning regulator (STR) for control the dc motor. The controller is the radial basis function neural network (RBFNN) and acts as inverse of the dc motor. The dc motor parameters are estimated online using the system identification method where uses the Auto-regressive moving average (ARX) model which depends on the input and output values of the dc motor. The difference between the output of the dc motor and the reference signal is used to adjust coefficients of ARX model. These coefficients of ARX are used to update the weights of the RBFNN. The weight update equations are derived based on the least mean squares principle. The speed output tracks the reference trajectory though the self tuning regulator (STR) structure exposed to different types of disturbances for wide range of operating conditions. Then compared its result with the result of using proportional-plus-integral feedback (PI) self tuning regulator.

Keywords: STR, RBFNN, ARX, LMS, dc motor

1. INTRODUCTION

Adaptive control schemes are used for the control of plants, where the parameters of the plant are not known exactly or slowly time varying. Some reasons for using Adaptive control such as variations in process dynamics and variations in the character of the disturbances [1, 2]. Enzeng and others [3] present a neural network based self tuning PID controller for autonomous underwater vehicle, the control system consists of neural network identifier and neural network controller, and the weights of neural networks are trained by using Davidon least square method, also[4].

Neural network (NN) is a good structure for control the nonlinear plants and has many types [5, 6]. Kumar [7] used neural network for modeling the retention process and as controller. In this paper, we used the RBFNN as a controller. This type is faster one and uses least number of neurons at hidden layer [8,9]. The inverse control means that the controller (RBFNN) acts the inverse of the plant (dc motor) so the output tracks the reference input [10].

The DC motors have been extensively used in control systems. The main advantages of dc motors are easy speed or position control and wide adjustable range Therefore, DC motors are often used in a variety of industrial applications such as robotic manipulator, where a wide range of motions are required to follow a predetermined speed or position trajectory under variable load [11,12,13]. Sabahi [14] used a new adaptive and nonlinear control based on
neural network approaches, this method has been named feedback error learning (FEL) approaches, that classical controller is used for training of neural network feedforward controller.

Pal [15] proposed a simple self-tuning scheme for PI-type fuzzy logic controllers (FLCs) for a real time water pressure control system. This scheme is improved performance of the system even at load change and set point variations. Kota [16] used PID controller and fuzzy logic controller for control separately excited dc motor. Fuzzy self-tuning PID has better dynamic response curve, shorter response time, small overshoot, and small steady state error compared to the conventional PID controller. Fawaz [17] presented a simulation and hardware implementation of a closed loop control of a separately excited dc motor using a self-tuning PID controller. It gives very acceptable results in the reduction of overshoot, stability time and the steady-state transient response of the controlled plant. Saad [18] proved that the proposed Neural Network (NN) self-tuning PID controller is more efficient to control the robot manipulator to follow the desired trajectory compared to classical tuning method of PID controller. Alfonso [19] introduced a new self-tuning algorithm is developed for determining the Fourier Series Controller coefficients with the aim of reducing the torque ripple in a Permanent Magnet Synchronous Motor (PMSM), thus allowing for a smoother operation. This algorithm adjusts the controller parameters based on the component's harmonic distortion in time domain of the compensation signal.

In this paper a new technique is proposed that gives a good control for the dc motor. An online control algorithm is structured using the radial basis function neural network (RBFNN). The dc motor parameters are estimated on line and are used to update the weights of the RBFNN. The weight update equations are derived based on the least mean squares principle. The RBFNN virtually models the inverse of the dc motor and thus the output tracks the reference trajectory. This scheme is exposed to several types of disturbances for wide range of operating conditions. The armature resistance and rotor inertia for the dc motor are varied due to the temperature deviation and also, take in account the changing of disturbance torque. The self tuning regulator (STR) meets the aforementioned disturbances separately and simultaneously.

2. PROPOSED STRUCTURE

Figure (1) is the proposed structure. Autoregressive with exogenous input (ARX) is used to identify the dc motor and found the model. The model coefficients are updated online depending on dc motor parameters variation. These coefficients are fed the weight update block which trains the controller whether RBFNN or PI controller using the least mean square LMS algorithm.

![Figure 1. Proposed self-tuning dc motor regulator structure](image)
2.1 ARX Model

The process is modeled by an ARX model [20], whose output is given by

\[ y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{j=1}^{m} b_j x(t-j) \]  

Or in terms of \( q^{-1} \) operator

\[ y(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} x(t) \]  

2.2 Radial Basis Functions Neural Networks

A single input single output radial basis function neural network (SISO RBFNN) is shown in Figure (2). It consists of an input node \( r(t) \), a hidden layer with \( n \) neurons and an output node \( x(t) \). Each of the input nodes is connected to all the nodes in the hidden layer through unity weights (direct connection). While each of the hidden layer nodes is connected to the output node through some weights: \( w_1, \ldots, w_n \).

Each neuron finds the distance \( d \) of the input and its center and passes the resulting scalar through nonlinearity. So the output of the hidden neuron is given by [8, 20]

\[ \phi(d) = \exp(-\frac{1}{2} d^2) = \exp(-\frac{1}{2} \|r(t) - c_i\|^2_\Sigma) \]  

\[ d = \left( \frac{r(t) - c_1}{\beta_1} \right)^2 + \cdots + \left( \frac{r(t) - c_n}{\beta_2} \right)^2 \]  

\( c_i \) is the center of \( i^{th} \) hidden layer node where \( i = 1, \ldots, n \), \( \Sigma \) is the norm matrix and \( \phi(\cdot) \) is the nonlinear basis function. Normally this function is taken as a Gaussian function of width \( \beta \). The output \( x(t) \) is a weighted sum of the outputs of the hidden layer, given by

\[ x(t) = \sum_{i=1}^{n} w_i \phi(\|r(t) - c_i\|^2_\Sigma) \]  

As we see the radial basis function (RBF) network utilized a radial construction mechanism. This gives the hidden layer parameters of RBF networks a better interpretation than for the multilayer perceptron network MLP, and therefore allows new, faster training methods.

![Figure 2. A general RBF network](image-url)
2.3 Single-phase Full-converter Drive

The dc motor dynamic are given by the following equations:

\[
K_b w(t) = -Ri_a(t) - L \frac{di_a(t)}{dt} + v_a(t) \tag{6}
\]

\[
K_m i_a(t) = J \frac{dw(t)}{dt} + bw(t) \tag{7}
\]

where \(w, v_a, i_a, R, L, b, J, K_m\) and \(K_b\) the rotor speed, terminal voltage, armature current, armature resistance, armature inductance, damping constant, rotor inertia, torque constant and back emf constant, respectively. Figure (3) describes the block diagram of the DC motor

![Figure 3. The block diagram of the dc motor](image)

If the armature circuit of a dc motor is connected to the output of a single-phase controlled rectifier, the armature voltage can be varied by varying the delay angle of the converter, \(\alpha\), as [21]:

\[
V_a = \frac{2V_m}{\pi} \cos \alpha \quad \text{for} \quad 0 \leq \alpha \leq \pi \tag{8}
\]

2.4 Parameters Estimation for Self-tuning of RBFNN/PI

The parameters of the dc motor model are estimated online and are used to update the coefficients of the controller (weights of the RBFNN / parameters of PI). The weight/coefficient update equations are derived based on a recursive scheme (least mean squares principle). This previous parameters are updated by minimizing the performance index \(I\) given by [9]

\[
I = \frac{1}{2} e^2(t) \tag{9}
\]

\[
e(t) = r(t) - w(t) \tag{10}
\]

where \(r(t)\) is the reference input signal and \(w(t)\) is the output speed of the DC motor model. The coefficients of the ARX model and the weights of the RBFNN/parameters of the PI are updated in the negative direction of the gradient as,

\[
\theta(K + 1) = \theta(K) - \mu \frac{\partial I}{\partial \theta(K)} \tag{11}
\]

and

\[
W(K + 1) = W(K) - \mu \frac{\partial I}{\partial W(K)} \tag{12}
\]
\[ PI(K+1) = PI(K) - \mu \frac{\partial I}{\partial PI(K)} \]  

where \( \theta = [q \ldots a_h b_1 \ldots b_m] \) is the parameter vector, \( W = [w_i w_j \ldots w_m] \) is the weight vector for RBFNN, \( PI = [k_p k_i] \) vector for the parameters proportional-integral (PI) and \( \mu \) is the learning parameter. The variable \( K \) is used to show the iteration number of training.

Keeping the regressions of the variables in the system in a regression vector \( \psi \) as \( \psi(t) = [u(t-1) \ldots u(t-n) V_a(t-d) \ldots V_a(t-m-d)] \) and finding partial derivatives.

\[
\frac{\partial I}{\partial \theta} = \frac{1}{2} \frac{\partial e^2(t)}{\partial \theta} 
= e(t) \frac{\partial}{\partial \theta} \left( r(t) - w(t) \right) 
= e(t) \frac{\partial}{\partial \theta} \left( r(t) - \left( a_i q^{-i} + \ldots a_n q^{-n} \right) w(t) - \left( b_i q^{-i} + \ldots + b_m q^{-m} \right) q^{-d} V_a(t) \right) 
\]

\[
\frac{\partial I}{\partial \theta} = -e(t) \psi(t) 
\]

The final parameter update equation will be,

\[
\theta(K+1) = \theta(K) + \mu e(t) \psi(t) 
\]

The partial derivatives for the weights are derived as follows,

\[
\frac{\partial I}{\partial W} = \frac{1}{2} \frac{\partial e^2(t)}{\partial W} 
= e(t) \frac{\partial}{\partial W} \left( r(t) - \frac{B(q^{-1})}{A(q^{-1})} q^{-d} V_a(t) \right) 
\]

\[
\frac{\partial I}{\partial W} = -e(t) B(q^{-1}) q^{-d} \phi(t) 
\]

The final weight update equation will be,

\[
W(K+1) = W(K) + \mu e(t) B(q^{-1}) q^{-d} \phi(t) 
\]

But the final coefficients update equation of PI will be,

\[
PI(K+1) = PI(K) + \mu e(t) B(q^{-1}) \left(-v_a(t-1) + (t_s + 1) v_a(t)\right) 
\]

\( t_s \) is the sample time.

**3. SIMULATION RESULTS**

The proposed self-tuning regulator (STR) structure exposed to different disturbances as dc motor does in life. The dc motor meets to changing in its parameters due to increased temperature. This paper studies these variations where the motor model parameters are estimated online and are used to update the weights of the RBFNN at the same instant.
Figure (4) shows the efficient response of the RBFNN to the square input and the error signal between the reference and the RBFNN output, in the case no disturbance. The changing of temperature has an impact upon the parameters of a dc motor as an armature resistance and rotor inertia. Figure (5) show the effect of variance for the armature resistance at maximum value $R=14.8\,\Omega$ at specified period $t \geq 300$, where the resistance has the range $2 \leq R \leq 14.8$. The output system takes the same trajectory until the armature resistance became $14.9\,\Omega$.

Figure (6) show the effect of variance for the rotor inertia at maximum value $J = 0.08\,\text{Kg.m}^2$ at specified period $t \geq 300$; the parameters evolution turn into large values when the rotor inertia increases, so the inertia has the range $0.02 \leq J \leq 0.08$.

In the reality, we must take account of the disturbance torque. Figure (7) illustrates the efficient output while this disturbance reach to maximum value $T_d = -1.5\,N.m$ at specified period $300 \leq t \leq 400$, where the disturbance torque has the range $T_d < 1.5$. The previous mentioned disturbances are applied synchronized. Figure (8) presents the output signal in this case, figure 8.a when $R=10\,\Omega$, $J = 0.05\,\text{Kg.m}^2$ at $t \geq 300$ and $T_d = -1\,N.m$ at $220 \leq t \leq 310$, figure 8.b when $R=10\,\Omega$, $J = 0.05\,\text{Kg.m}^2$ at $t \geq 300$ and $T_d = -1\,N.m$ at $300 \leq t \leq 400$. 

-220-
Figure 6. The effect of variance for the rotor inertia at $t \geq 300$ at using radial basis case (a) $J = 0.07\ Kg\ m^2$ (b) $J = 0.09\ Kg\ m^2$

Figure 7. The effect of variance for the torque disturbance at $300 < t < 400$ at using radial basis case (a) $Td = -1\ N.m$ (b) $Td = -1.6\ N.m$

Figure 8. The effect of disturbances synchronized on the speed of radial basis self tuning DC motor system
The proposed PI self-tuning regulator (STR) structure exposed to different disturbances as previous counterpart. Figure (9) shows the efficient response of the proportional-integral (PI) to the square input and the error signal between the reference and the PI self tuning output without disturbance.

Figure 9. Tracking trajectory for PI self tuning dc motor and the error signal

Figure 10. The effect of variance for the armature resistance at \( t \geq 300 \) at using PI case (a) \( R = 15\Omega \) (b) \( R = 16\Omega \)

Figure (10) shows the effect of the armature resistance fluctuation at maximum value \( R=15\Omega \) at specified period \( t \geq 300 \), where the resistance has the range \( 2 \leq R \leq 15 \). The output system mimics the trajectory exactly until the armature resistance became \( 15\Omega \). But the error between the response of the system and the reference signal is not smaller than radial basis regulator.
Figure 11. The effect of variance for the rotor inertia at $t \geq 300$ at using PI case (a) $J = 0.06 \text{ Kg.m}^2$ (b) $J = 1 \text{ Kg.m}^2$.

Figure (11) shows the effect of variance for the rotor inertia at specified period $t \geq 300$. The speed of the dc drive follows the excitation signal; however, the ARX model parameters value fluctuate sharply and this is not good.

Figure (12) illustrates the efficient output while this disturbance reach to maximum value $Td = -1.7 \text{ N.m}$ at specified period $300 \leq t \leq 400$, where the disturbance torque has the range $Td < 1.7$.

Figure 12. The effect of variance for the torque disturbance at $300 < t < 400$ at using PI case (a) $Td = -1 \text{ N.m}$ (b) $Td = -1.7 \text{ N.m}$

The previously mentioned disturbances are applied synchronized. Figure (13) presents the output signal in this case, figure 13.a when $R = 3 \text{ \Omega}$, $J = 0.03 \text{ Kg.m}^2$ at $t \geq 300$ and $Td = -0.3 \text{ N.m}$ at $300 \leq t \leq 400$, figure 13.b when $R = 4 \text{ \Omega}$, $J = 0.04 \text{ Kg.m}^2$ at $t \geq 300$ and $Td = -0.4 \text{ N.m}$ at $220 \leq t \leq 310$. The PI self tuning controller doesn’t work well with this kind of disturbance.
Figure 13. The effect of disturbances synchronized on the speed of PI self-tuning DC motor system

4. CONCLUSIONS

This paper introduces a very simple structure for control the dc motor that updates itself online. The exact model of the dc motor needs not to be known and just the estimates are enough to drive the RBFNN as the process inverse. The changing of temperature is affected on the armature resistance and rotor inertia for the dc motor; also we take in account the changing of disturbance torque. The proposed STR structure exposed to all pervious disturbances separately and simultaneously.

The adaptive self-tuning regulator introduces a good solution for control the dc motor even if the model meets a different individual disturbances or synchronous disturbances.

The RBFNN is a fast neural network compared with others type due to using least mean squares principle as training algorithm. Its structure has 2 neurons in hidden layer.

APPENDIX–I

The parameter of the DC motor

Armature resistance $R = 2.0$ Ohms
Armature inductance $L = 0.5$ Henrys
Torque constant $K_m = 0.015$
Emf constant $K_b = 0.015$
Friction coefficient = $0.2 \, N.m / rad / sec$
Moment of inertia $J = 0.02 \, Kg \cdot m^2$
REFERENCES


