DC MOTOR POSITION CONTROL USING MODEL REFERENCE ADAPTIVE CONTROLLER (MRAC)

*Okafor, P.U.[1], Eneh, I.I.[2], Okoye, F.A.[3] and Eke, J.[4]

1,2,4 Department of Electrical/Electronic Engineering Enugu State University of Science and Technology, Enugu State, Nigeria
3 Department of Computer Engineering Enugu State University of Science and Technology, Enugu State, Nigeria

*Author for correspondence:Okafor, P.U.; Email: ptuch7777@yahoo.com

ABSTRACT

It may not be easily obvious that precise control of speed means accurate control of the position. Thus for accurate position to be achieved, the profile of position should be incorporated in the equation governing the precise control of speed by simple integration; this is same for all DC motors. The artificial neural network (ANN) inverse model (AIM) of the DC motor was designed as the plant in the neural model reference adaptive controller (MRAC). The input to the AIM is the DC motor position at four successive intervals. The plant model is identified first then; the controller is trained so that the plant output follows the reference model output. The idea is to bring down the tracking error to zero or minimal acceptable value. Simulation shows that using motor position as input ensures accurate position control which automatically brings about accurate speed tracking.

Keywords: Angular displacement, Angular velocity, Artificial Neural Network (ANN), DC motor

1.0 INTRODUCTION

Direct Current (DC) motors find numerous applications as the most prime movers in industry (Sheel et.al 2010). When compare with alternating current (AC) motors, DC motors are more expensive, because of their brushes and commutators, and variable-flux dc motors are suitable only for certain types of control applications. But in reality, AC motors are very difficult to control more especially in position control due to their nonlinear characteristics, which makes analytical task difficult (Kuo and Golnaraghi, 2002). Before the full introduction of permanent-magnet technology, the torque - per-unit volume or weight of a DC motor with a permanent-magnet (PM) field was far from desirable.

Today, with the development of the rare-earth magnet, it is possible to achieve very high torque-to-volume PM dc motors at reasonable cost. Also, the advances made in brush-and-commutator technology have made these wearable parts practically maintenance-free (Baldor Motors and Drives, 2009). The advancements made in power electronics have made brushless dc motors quite popular in high-performance control systems. Also low-time-constant properties have opened new applications for dc motors in computer peripheral equipment such as tape drives, printers, disk drives, and word processors, as well as in the automation and machine-tool industries.

Thus it is compulsory that an adaptive controller must be deployed in order to achieve this compensation (Pathak and Adhyaru, 2015). Artificial Neural Networks (ANN) is among the newest signal-processing technologies in the engineer’s toolbox (Narendra and Parthasarathy, 1990). An ANN is an adaptive, most often nonlinear system that learns to perform a function (an input/output map) from data. Being adaptive simply means that the system parameters are changed during operation, normally called the training phase. After the training phase the ANN parameters are fixed and the system is deployed to solve the problem at hand.

The ANN is built with a systematic step-by-step approach to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule. Accordingly the input/output training data are fundamental in neural network technology, because they convey the necessary information to discover the optimal operating point (Principe, 2000). The nonlinear nature of the neural network processing elements (PEs) provides the system with lots of flexibility to achieve practically any desired input/output map, this means that some ANN are universal mappers (Beale, Hagan and Demuth, 2015). Adaptive techniques are best suited when the parameters to be controlled are unknown and nonlinear. The neural model reference control (MRAC) technique was deployed because it has better performance in control applications where precision is paramount.

2.0 THEORY

Basically, the dc motor is a torque transducer that converts electric energy into mechanical energy. The torque developed on the motor shaft is directly proportional to the field flux and the armature current. Assume that a current-carrying conductor is established in a magnetic field with flux ϕ, and the conductor is located at a distance r from the center of rotation (Zhao and Yu, 2011).

The relationship among the developed torque, flux ϕ, and current i_a is given as:

$$T_m = k_m \phi i_a \quad (1)$$

Where;

T_m = the motor torque (in N-M)

ϕ, = the magnetic flux in (in webers)

i_a = the armature current (in amperes)

k_m = a proportional constant

In addition to the torque developed, when the conductor moves in the magnetic field, a voltage is generated across its terminals. This voltage is known as the *back emf*, which is proportional to the shaft velocity, and tends to oppose the current flow. The relationship between the back emf and the shaft velocity is:

$$e_b = k_m \phi w_m \quad (2)$$

Where;

e_b = the back emf (in volts)

w_m = the shaft velocity of the motor (in rad/sec)

Equations 1 and 2 form the fundamentals of the dc-motor operation.

Generally, the magnetic field of a dc motor can be produced by field windings or permanent magnets (Brown, 2002). For the purpose of this work, focus will be on PM dc motors in control system applications. PM dc motors can be classified according to commutation scheme and armature design. Conventional dc motors have mechanical brushes and commutators. However, an important type of dc motors in which the commutation is done electronically is called brushless dc**.** In accordance with the armature construction, the PM dc motor can be broken down into three types based on type of armature design: *iron-core, surface-wound,* and *moving-coil motors.*

2.1 Artificial Neural Network Controller

An input to the ANN controller is made from the DC motor with a corresponding desired or target response set at the output. Error is computed from the difference between the desired response and the system output. This error information is fed back to the system and adjusts the system parameters in a systematic fashion (the learning rule). The process is repeated until the performance is acceptable. It is clear from this description that the performance rests heavily on the data. This operating approach should be contrasted with the traditional engineering design, made of exhaustive subsystem specifications and intercommunication protocols.

2.2 The Model Reference Adaptive Controller:

MRAC is one of the neural network architectures for prediction and control implemented in Neural Network Toolbox software. The model reference architecture requires that a separate neural network controller be trained offline, in addition to the neural network plant model. The controller training is computationally expensive, because it requires the use of dynamic back propagation. The MRAC architecture uses two neural networks: a controller network and a plant model network. The plant model is identified first, and then the controller is trained so that the plant output follows the reference model output (Beale, Hagan and Demuth, 2015).

3.0 MODELING THE PM DC MOTOR

DC motors are used extensively in control systems especially in industrial actuators so, it is paramount to establish mathematical model for analytical purposes for efficient control application of dc motors. The DC motor takes in single input in the form of an input voltage and generates a single output parameter in the form of output speed. It is a single-input, single-output system (SISO). Fig 1 shows the electrical representation of a DC motor.

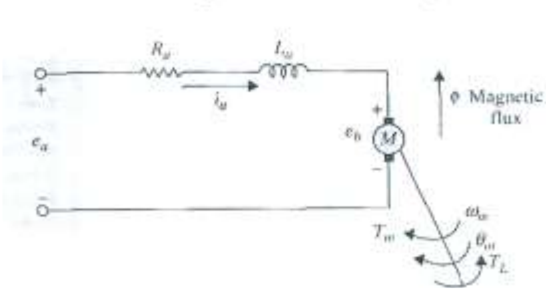

Fig 1: *Electrical Model of DC Motor (Kuo and Golnaraghi, 2002)*

The armature is modeled as a circuit with resistance R_a connected in series with an inductance L_a , and a voltage source e_b representing the back electromotive force (emf) in the armature when the rotor rotates. Looking at the diagram of fig 1, it can be seen that the control of the dc motor is applied at the armature terminals in the form of applied voltage e_a (t). It can be deduced that the torque developed in the motor is proportional to the air-gap flux and the armature current. The equations that described the DC servomotor behavior are giving below (Kuo and Golnaraghi, 2002):

$$T_m(t) = K_m(t)\,\Phi\, i_a(t) \qquad (3)$$

Since Φ is constant, equation 3 is in form

$$T_m(t) = K_i\, i_a(t)$$

$$K_i\, i_a(t) = i_m\omega_m + b\omega_m + T_L \qquad (4)$$

Where:

$T_m(t)$ =motor torque. $i_a(t)$ = armature current. $T_L(t)$ = load torque. ϕ = magnetic flux in the air gap. k_m = proportionality constant. K_i = torque constant in N-m/A. $\omega_m(t)$=rotor angular velocity i_m = equivalent moment of inertia reflected at the motor shaft. Putting the control input voltage e_a (t) into consideration, the cause and effect equations for the motor circuit in same fig 4 are:

$$\frac{di_a(t)}{dt} = \frac{1}{L_a}e_a(t) - \frac{R_a}{L_a}i_a(t) - \frac{1}{L_a}e_b(t) \qquad (5)$$

$$e_b(t) = k_b\frac{d\theta_m(t)}{dt} = K_b\omega_m(t) \qquad (6)$$

$$\frac{d^2\theta_m(t)}{dt^2} = \frac{1}{J_m}T_m(t) - \frac{1}{J_m}T_L(t) - \frac{B_m}{J_m}\frac{d\theta_m(t)}{dt} \qquad (7)$$

Where:

$l_a(t)$ = armature inductance

R_a = armature resistance. $e_a(t)$ = applied voltage

$e_b(t)$ = back emf. K_b = back emf constant

$\omega_m(t)$= rotor angular velocity. B_m = viscous-friction coefficient. θ_m (t)= rotor displacement

J_m=rotor inertia. From equations 3 through 6, the applied voltage $e_a(t)$ is considered as the cause and Equation 5 considers that $\frac{di_a(t)}{dt}$ the immediate effect due to the applied voltage. From Equation 3, armature current $i_a(t)$ causes the motor torque $T_m(t)$, while in Equation 6 the back emf $e_b(t)$ was defined. It can be seen also from Equation 7 that the motor torque produced causes the angular velocity $\omega_m(t)$ and displacement θ_m (t) of the rotor respectively.

The state variables are of the system can be define as

- Armature current = $i_a(t)$
- Rotor angular velocity = $\omega_m(t)$
- Rotor angular displacement =θ_m (t)

It is possible to eliminate all the non-state variables from Equation 3 through 7 by direct substitution then present the dc state equation in vector-matrix form as follows:

$$\begin{bmatrix} \frac{di_a}{dt}(t) \\ \frac{d\omega_m(t)}{dt} \\ \frac{d\theta_m(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} & 0 \\ \frac{K_i}{J_m} & -\frac{B_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega_m(t) \\ \theta_m(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} e_a(t) + \begin{bmatrix} 0 \\ -\frac{1}{J_m} \\ 0 \end{bmatrix} T_L(t) \quad (8)$$

Note that in the case of Equation 8 above, that $T_L(t)$ is handled as a second input to the state equations. The transfer function between the motor displacement and the input voltage is obtained as thus;

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{k_i}{L_aJ_ms^3 + (R_aJ_m + B_mL_a)s^2 + (K_bK_i + R_aB_m)s} \quad (9)$$

Note that T_L has been set to zero in Equation 9. Fig 2 shows a block diagram of the dc motor system for speed control. From the diagram, one can see clearly how the transfer function is related to each block. It can be seen from Equation 9 that *s* can be factored out of the denominator and the significance of the transfer function $\frac{\Theta_m(s)}{E_a(s)}$ is that the dc motor is an integrating device between these two variables. $\Theta_m(s)$ is the rotor angular displacement Laplace transfer function, $E_a(s)$ is the input voltage Laplace transfer function and $\Omega_m(s)$ is the transform of angular velocity respectively. From fig 2 also, it can be seen that the motor has a built-in feedback loop caused by the back emf Eb.

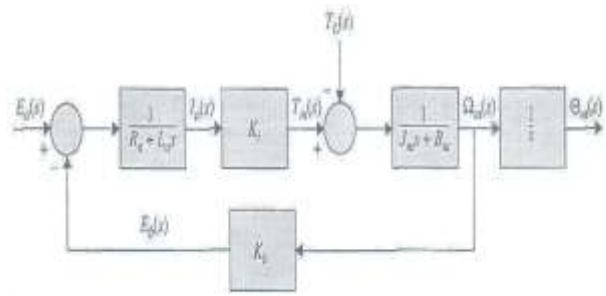

Fig 2: *Block Diagram of DC Servomotor in terms of speed*

The back-emf physical represents the feedback of a signal that is proportional to the negative of the speed of the motor. From equation 9, it can be noted that back emf constant K_b represents an added term to the resistance R_a and the viscous-friction coefficient B_m . Effectively, the back-emf effect is equivalent to an electric friction which tends to improve the stability of the motor and apparently the stability of the system.

3.1. DC Motor Equivalent Circuit in Discrete Form:

Recall that artificial neural network (ANN) is the modeling tool. So simulation can be performed on the control of Dc motor using ANN model, there is need to construct an equivalent DC motor to a discrete time model. Effectively, the load torque is assumed as

$$T_L = \mu\omega^2 m(t)[sgn(\omega_m(t))] \quad (10)$$

Where μ = a constant

It is obvious that from equation (10) that load torque is always opposes the direction of motion. Note that the choice of load torque here is arbitral because considering load torque as one of the functions of a DC motor; it is a

common characteristic for most propeller driven loads.
Alternatively,
Direct substitution and substitute for position in equations (4), (5) and (10) i.e. rewriting the angular velocity $\omega_m(k)$, which is the speed in terms of angular displacement θ_m which is the position as shown in figure 3.

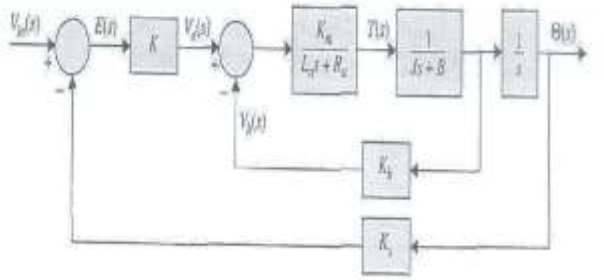

Fig 3: *Block Diagram of DC Servomotor in terms of speed and position*

Deploying direct substitution and substitute for position in equations (4), (5) and (10) i.e. rewriting the angular velocity $\omega_m(k)$, which is the speed in terms of angular displacement θ_m which is the position as shown in figure 3. Then the equations yields;

$$k_e\theta_m = -R_a i_a - L_a\frac{di_a}{dt} + v_a \tag{11}$$

$$k_e i_a = I_m\theta_m + b\theta_m + T_L \tag{12}$$

$$T_L = \mu\left(\frac{d\theta_m}{dt}\right)^2[sgn(\theta_m)] \tag{13}$$

Next is to estimate the derivatives of position and current in discrete form using a sampling interval of ΔT and forward difference.

$$\frac{d\theta_m(k)}{dt} = \frac{\theta_m(k+1) - \theta_m(k)}{\Delta T} \tag{14}$$

$$\frac{d^2\theta_m(k)}{dt^2} = \frac{\frac{d\theta_m(k+1)}{dt} - \frac{d\theta_m(k)}{dt}}{\Delta T} \tag{15}$$

$$\frac{d^3\theta_m(k)}{dt^3} = \frac{\frac{d^2\theta_m(k+1)}{dt^2} - \frac{d^2\theta_m(k)}{dt^2}}{\Delta T} \tag{16}$$

$$\frac{di_a}{dt} = \frac{i_a(k+1) - i_a(k)}{\Delta T} \tag{17}$$

$$T_L(k) = \mu\left(\frac{d\theta_m(k)}{dt}\right)^2[sgn(\theta_m(k)]$$

(18)

$$\frac{dT_L(k)}{dt} = \frac{T_L(k+1) - T_L(k)}{\Delta T} \tag{19}$$

Next is to evaluate the armature current i_a in terms angular displacement θ_m which is the position here. Then substituting i_a in terms of θ_m using equations (11) into the following equation $\frac{K_e{}^2(\Delta T)^2 + R_a(b\Delta)^2}{(L_a I_m + R_a I_m \Delta T)}$ from the work of Weerasooriya and El-Sharkawi (1991), to determine the function governing the speed control of a DC motor, gives

$$k_e\dot{\theta}_m = -\frac{R_a}{k_e}\left[I_m\ddot{\theta}_m + b\dot{\theta}_m + T_L\right] - \frac{L_a}{k_e}\left[I_m\dddot{\theta}_m + b\ddot{\theta}_m + T_L\right] + v_a \tag{20}$$

Or

$$v_a = \frac{L_a I_m}{k_e}\dddot{\theta}_m + \left[\frac{R_a I_m}{k_e} + \frac{L_a b}{k_e}\right]\ddot{\theta}_m + \left[\frac{R_a b}{k_e} + k_e\right]\dot{\theta}_m + \left[\frac{R_a}{k_e}T_L + \frac{L_a}{k_e}T_L\right] \tag{21}$$

Integrating equations (12), (13), (14), (15), (16) and (17) into equation (19) then the input voltage in equation (19) can be written as a function of:

$\theta_m(k+1)$.

$\theta_m(k)$.

$\theta_m(k-1)$.

$\theta_m(k-2)$.

$$v_a(k) = g[\theta_m(k+1), \theta_m(k), \theta_m(k-1), \theta_m(k-2)] \tag{22}$$

Effectively, equation (22) forms the relationship between the input voltage v_a and the motor position θ_m at four successive sampling instances. Assume that the term $\theta_m(k+1)$ is replaced in equation (22) with desired reference motor position at next instance as $\theta_d(k+1)$, and compute the control voltage (the input voltage) $v_a(k)$ with the following equation, then

$$v_a(k) = g[\theta_d(k+1), \theta_m(k), \theta_m(k-1), \theta_m(k-2)] \qquad (23)$$

So, if the computed voltage $v_a(k)$ at sampling instance k is applied, then the resulting motor position at instant $(k+1)$ will be equal to:

$\theta_m(k+1) = \theta_d(k+1)$, i.e. the desired motor position, it effectively takes the following forms of input to the ANN

$\theta_d(k+1)$. = the reference position

$\theta_m(k)$. = Position at first instance

$\theta_m(k-1)$. = Position at second instance

$\theta_m(k-2)$. Position at third instance

4.0 STRUCTURE OF THE ANN CONTROLLER

The non-linear controller for this work is the Artificial Neural Network (ANN) Controller. Here, the design of ANN incorporated a feed forward neural network (FFNN), which is made up of one input layer and one hidden layers with an output layer. The layers respectively consist of number of neurons. Each neuron has two functions as:

- Summing up all the outputs from the previous layers multiply by the corresponding weights
- Performing the nonlinear sigmoidal or linear function on the sum

During training, errors are back propagated and also minimized using least mean square algorithm. The basis for weights connection between the input and hidden layers are based on the fact that errors in the output determine the measures of the hidden layer output errors. This adjustment of weights between the layers and recalculating the output in an iterative process is continued till the error falls below a tolerable level.

4.1 Mapping:

The input vectors are mapped to the output vectors by the feed-forward back propagation network. Before the training begins, the first thing to be done is to choose pairs of input and desired output vectors that will be used in training the network. As soon as the training is completed and the weights set, this time the system has learnt, then the network is used to determine or find outputs for new inputs. It could be noted that the dimension of the input vector determines the number of neurons in the input layer, and the dimension of the outputs determines the number neurons in the output layer respectively (Arnari, 1990).

4.2 Layout:

This is based on the same fact mentioned above that the dimensions of the input and out patterns determine the number of neurons in the input and output layer. The network has three fields of neurons:

- One field for the input layer (Field A)
- Another for the hidden processing element (Field B)
- And the last for the output neurons (Field C)

The layers are connected and the connections are for feed-forward activity. They are connected in a manner that every neuron in A connects to B and those from B connect to C. also it should be noted that there are two sets of weights:

- Ones responsible for neuron activation in the hidden layer and
- Those responsible for neuron activation in the output layer.

4.3 Training the ANN Controller:

The feed-forward back propagation undergoes supervised learning with a finite number of patterns consisting of an input pattern and a desired output pattern (Valluru, 1995). At the input layer, the input pattern is presented. Then, the input layer neurons pass the activations to the next layer neurons i.e. those in the hidden layer. The outputs of the hidden layer neurons are got by introducing a bias, and also a threshold function, with activations determined by the weights and the inputs Zurada, (1992). The output from the hidden layer becomes the input to the output neurons, which process the input using an optional bias and a threshold function. Then the final out of the network is determined by the activations from the output layer.

The input and output of this network is guided by some basic equations, the net input of the jth neuron of the hidden layer at the time instant n is given as follows (Haykins, 1999):

$$S_j^h(n) = \sum_{I=1}^{N} W_{ij}^h(n) I_i(n) \qquad (24)$$

Where W_{ij}^h is the connecting weight between the i^{th} neuron at the input layer and the j^{th} neuron at the hidden layer. The I_i is the i^{th} input, and N is the number of inputs. Then, the output from the j^{th} neuron from the hidden layer at n^{th} instant is given by:

$$O_j^h(n) = f^h[S_j^h(n) + B_j^h(n)] \qquad (25)$$

From equation 25, B_j^h is the bias of the j^{th} neuron and f^h is the activation function acting on each neuron at the hidden layer. The activation function can be tan sigmoidal, log sigmoidal or linear. The functions are described as follows (Beale, Hagan and Demuth, 2015):

$$tansig(x) = \frac{1-e^{-2x}}{1+e^{-x}} \qquad (26)$$

$$logsig(x) = \frac{1}{1+e^{-x}} \qquad (27)$$

$$linear(x) = x \qquad (28)$$

In the above equations x represents the input to the activation function. It follows that the net input of the k^{th} neuron of the output layer at time instant n is given by:

$$S_k^o(n) = \sum_{j=1}^{M} w_{jk}^o(n)\, O_j^h(n) \qquad (29)$$

Where M is the number of neurons in the hidden layer and $w_{jk}^o(n)$ is the weight between the j^{th} neuron at the hidden layer and k^{th} neuron at the output layer respectively. It therefore also followed that output from the k^{th} neuron at the output layer at time instant n can be presented in the form:

$$O_k^o(n) = f^o[S_k^o(n) + B_k^o(n)] \qquad (30)$$

Where f^o = the activation function of the output layer and

$B_k^o(n)$ = the bias of the k^{th} neuron at the output layer.

It is also important to put into consideration how the weight is updated at various levels during the network training. To do that, there is a basic equation that describes the updating of the weight through the error signal at the output of the neuron *k* at the iteration and it is given as follow:

$$e_k(n) = d_k(n) - O_k^o(n) \qquad (31)$$

Where $d_k(n)$ represents the desired output for neuron *k*.

5.0 ANN MODEL OF DC MOTOR

Before ANN can be used to control the operation of the DC motor, the DC motor being the plant must also be modeled using ANN tool. From equation 23 where $v_a(k)$ is a function of speed at successive time intervals k+1, k and k-1 for any required trajectory, what happens is that the ANN Inverse Model (AIM) generates an output that is proportional to the voltage required at the input of the DC motor to produce these speed at the time intervals. Here, the output-input mapping is many to one perhaps, disturbances and other uncertainties may lead to the input-output mapping to become one-to-many leading to degradation in the control performance. Though, the AIM relies on the accuracy of the model used for the controller design so, the research will not worry about the degradation. The block diagram of the speed based AIM is shown in fig 4.

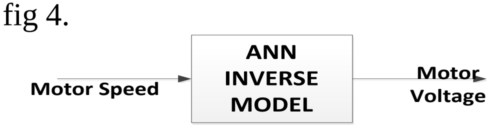

Fig 4: *Block diagram of the AIM*

5.1 Structure of the AIM:

The AIM for this research work is made up of three inputs and a single output structure for the three successive speed instances. Based on equation 23, the three inputs are $\omega_m(k+1)$;

Speed at first instance $\omega_m(k)$; speed at second instance, $\omega_m(k-1)$;speed at third instance and the output is the $V_a(k)$ which is the motor terminal output voltage $V_t(k)$ from fig 5.

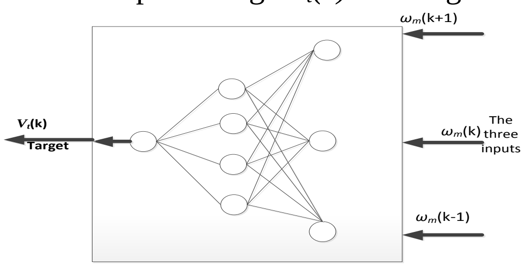

Fig 5: *The structure of AIM*

So based on the same equation 23, the nonlinear function (f.) can be presented in the following form:

$$f(\omega_m(k+1), \omega_m(k), \omega_m(k-1) = \frac{(\omega_m(k+1) - \alpha\omega_m(k) - \beta\omega_m(k-1) - \gamma\, sgn(\omega_m(k))\omega_m^2(k) - \delta sgn(\omega_m(k-1))\omega_m^2(k-1)}{\zeta} \quad (32)$$

The values of $(\omega_m(k+1), \omega_m(k), and\ \omega_m(k-1)$ apparently form the independent inputs of the ANN and the corresponding output as well is generated from equation 28.

5.2 Evaluating the performance of the AIM:

The generated $(\omega_m(k+1), \omega_m(k), \omega_m(k-1)$ inputs and the corresponding targets $V_a(k)$ are used for offline training of the AIM to represent any DC servomotor with unknown parameters. From figure 6, it could be seen that the performance error is represented by $e_i(k)$. In evaluating the AIM performance, the value of $[e_i(k)]^2$ for all kT that are elements of time from 0 to t_f is minimized, that is;

$$[e_i(k)]^2 \forall kT \in [0, t_f] \quad (33)$$

Where T is the sampling period and t_f is the time for which simulation is performed. The terminal voltage (estimated) is given by:

$$\hat{V}_t = N[\omega_m(k), \omega_m(k-1), \omega_m(k-2)] \quad (34)$$

Once the DC motor is excited by an input signal, the output from the DC motor which is speed in this context is fed into the AIM as an input. The terminal voltage i.e. $\hat{V}_t$(k-1) is compared with the actual motor output $e_i(k)$ for a common excitation signal. Then the mean square value of the error $e_i(k)$ between the actual motor input and the estimated output voltage yields the performance error of the AIM.

5.3 Structure of the Position-based AIM:

From equation 23, the input variables at four successive instances form the input to the ANN while the terminal $V_T(k)$ corresponds to the desired output of the controller, which is what is fed to the motor. To obtain the training sets, sequence of voltage signals capable of exciting the motor are applied and the motor position at successive sampling instants recorded then, the training set can be generated from those recorded input and output data. So, each training set is made up of the four successive motor positions and the applied voltage. The four successive instances are $\theta_d(k+1)$; the reference position, $\theta_m(k)$; Position at first instance, $\theta_m(k-1)$; Position at second instance, $\theta_m(k-2)$; Position at third instance and the applied voltage $V_a(k)$.

5.4. Performance Evaluation of the position-based AIM:

To evaluate the system performance for this position-based AIM, the parameter that is most important to put into consideration is the modeling error. This error is recorded in the form of;

$$[e_i(k)]^2 \forall kT \in [0, t_f] \quad (35)$$

Where T is the sampling period and t_f is the time for which simulation is performed. The terminal voltage (estimated) is given by:

$$\hat{V}_t(k-1) = N[\theta_m(k), \theta_m(k-1), \theta_m(k-2)] \quad (36)$$

So, those randomly generated input variables at four successive instants and the corresponding output are the parameters that are used in offline training of the ANN. The block diagram for the performance evaluation of the AIM with position as input is shown in figure 6.

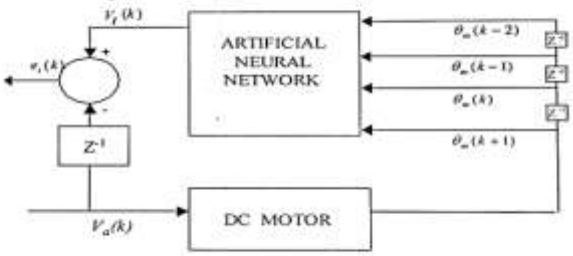

Fig 6: *Performance Evaluation of the AIM with position as input*

The target value is the voltage at the instant k. Only one hidden layer is chosen as it was found to work well in order to arrive at the desired accuracy. In the simulation carried out, a sampling frequency of 0.01 sec was used. In order to test the performance of the AIM, the model was excited with the two different reference input voltage signals $v_I(t)$ and $v_{II}(t)$ as given in Eqs. (37a) and (37b) respectively:

$$v_I(t) = 10\sin(0.75u) + 10\cos(0.5u) \qquad (37a)$$

$$v_{II}(t) = 10\cos(0.5u) + 10\cos(0.25u) + 10\exp(-0.05u) \qquad (37b).$$

Figure 7a illustrates the error between the AIM output and the reference voltage signal $v_I(t)$.

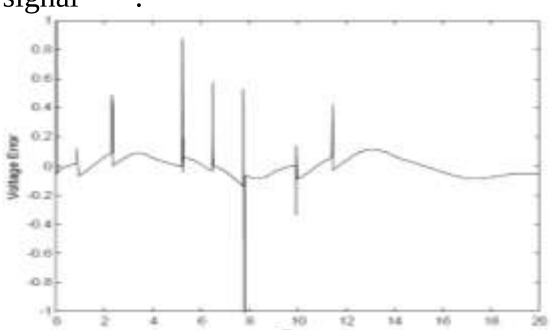

Fig 7a: *Error Between The Reference Signal $v_1(t)$ and Actual AIM Output.*
Figure 7b *shows the error between the actual*

AIM output and reference signal $v_{II}(t)$. Both graphs: 7a and 7b respectively shows that the error is within ±0.2V which is within 1% variation from the input (reference) profile.

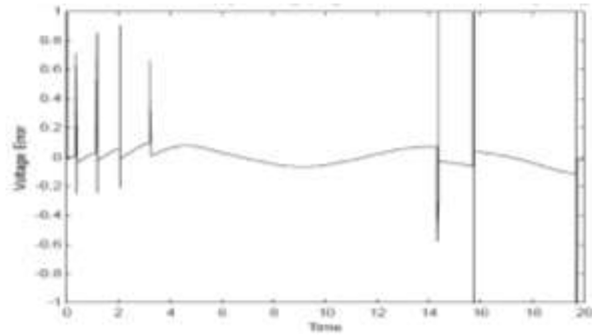

Fig 7b: *Error Between The Actual AIM Output and Reference Signal* $v_{II}(t)$

6.0 POSITION CONTROL OF DC MOTOR

The essence of establishing the evaluation performance of the AIM is to minimize modeling error and once that is done, the position control can be accomplish by incorporating the AIM controller with the Model Reference Adaptive Control (MRAC) mechanism. The work of this MRAC system is to generate a reference position $\theta_d(k)$ that will serves as a reference point to the applied voltage $V_t(k)$ generated by the AIM controller to drive the motor position to $\theta_m(k)$ as shown in figure 8.

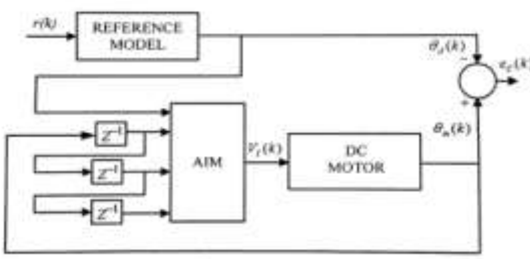

Fig 8: *Position Control of a DC Motor*

As in equation 35, the terminal voltage is calculated as well to bring down the error to a minimal such that

$$[e_c(k)]^2 \; \forall kT \in [0, t_f] \qquad (38)$$

is minimized. The reference position generated by the MRAC should be within the physical dynamic capabilities of the motor. Second-order reference model is deployed so that the dynamic parameters of the motor can be varied. From figure 13, the *r(k)*, serves as the input to the system and the second-order reference model takes the form of

$$\theta_d(k+1) = 0.6\theta_d + 0.2\theta_d(k-1) + r(k) \quad (39)$$

To achieve a successful training of the network, foremost it will be more realistic if the past values of $\theta_d(k)$ from equation 39 be replaced with past values of the actual current output $\theta_m(k)$ in order to avoid arriving at unpredictable results. By so doing, the new estimate for the desired output could then be $\hat{\theta}_m$ *(k+1)*. Therefore, the terminal voltage from the AIM controller will be generated by using the positions of the motor at four successive instants with the estimated output in the form

$$\hat{V}_t(k-1) = N[\hat{\theta}_m(k), \theta_m(k-1), \theta_m(k-2)] \quad (40)$$

For ranges of input *r(k)* to the system, $\theta_m(k)$, $\theta_m(k-1)$ and *r(k)* are used to generate the desired reference position from the MRAC at instant (k+1) using equation 38. Consequently, the desired reference positions and the motor positions at the k, k-1, k-2 instants will form the input to the AIM as shown in figure 9. So it is the duty of the AIM to evaluate the voltage at the instant (k+1), which is the voltage expected to drive the motor in accordance with the reference position.

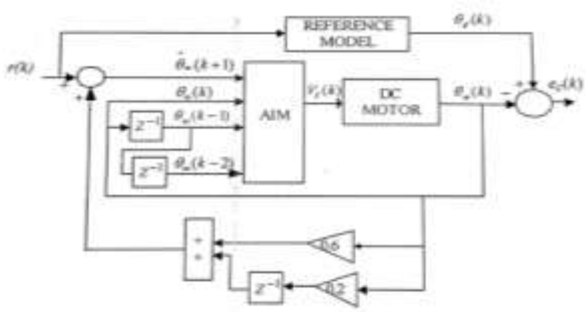

Fig 9: *Position Control System for DC Motor*

7. CONCLUSION

The model was developed using the motor position as input to the controller, which is the artificial inverse model (AIM) and that was established in equation 23. The motor positions at four successive instances were used as inputs to the AIM controller. Thus, from previous research works, using motor speed as inputs to the AIM improves accurate speed control but not accurate position control. Perhaps, using positions as inputs ensures accurate position control because, accurate position control automatically brings about accurate speed tracking.

8. REFERENCES

Amari, S. I. 1990. "Mathematical Foundations of Neurocomputing," *IEEE Proc.* 78(9): 1443-1463.

Baldor Motors and Drives, 2009).Servo Control Facts. Online at: www. Baldormotorsanddrives.com. Last Accessed 27th -05-2014.

Beale M.H, Hagan M.T. and Demuth H.B. (2015) *Neural Network Toolbox User'*

Benjamin C.Kuo and Farid Golnaraghi (2002). *Automatic Control Systems*, Eight Edition; John Wiley & Sons, Inc. USA.

Brown W. (2002). *Brushless DC Motor Control Made Easy.*Microchip AN857

Haykins S. (1999) "Neural Networks– A comprehensive foundation", Second Edition, Prentice Hall, 1999.

Narendra K. S. and Parthasarathy K. (1990). "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, 1, (1), pp. 4-27

Pathak K.B. and Adhyaru D.M, (2015). *Performance analysis of neural network based MRAC.* Conference on: Electrical, Electronics, Signals, Communication and Optimization (EESCO):IEEE.

Principe, J.C.(2000) "Artificial Neural Networks"*The Electrical Engineering*

Handbook Ed. Richard C. Dorf Boca Raton: CRC Press LLC, 2000.

Sheel S., Chandkishor R., and Gupta O. (2010). *Speed Control of DC Drives Using MRAC Technique.*International Conference on Mechanical and Electrical Technology:*EEE.*

Valluru B.R. (1995). *C++ Neural Networks and Fuzzy Logic*. MTBooks, IDG Books Worldwide, Inc. ISBN:n1558515526.

Weerasooriya S. and El-Sharkawi M.A. (1991) “Identification and control of a DC motor using Back-propagation Neural Networks”, IEEE Transactions on Energy Conversion, Vol.6, No.4, pp. 663-669.

Zhao J. and Yu Y. (2011) *Brushless DC Motor Fundamentals Application Note*.

Zurada, Jacek M (1992).*Introduction to Artificial Neural Systems*. Includes index. ISBN 0-3 14-93391 -3.