Particle Swarm Optimization for NARX Structure Selection – Application on DC Motor Model

Mohd I. Abdullah

[#]*Faculty of Electrical Engineering, Universiti Teknologi MARA*
40450, Shah Alam, MALAYSIA
mohdikhwan_abdullah@yahoo.com

Abstract **- This paper presents the nonlinear identification of a DC motor using Binary Particle Swarm Optimization (BPSO) algorithm, as a model structure selection method, replacing the typical Orthogonal Least Squares (OLS) used in system identification. The BPSO algorithm is an evolutionary computing technique put forward by (Kennedy and Eberhart, 1997). By representing its particles technique as probabilities of change (bit flip) of a binary string, the binary string was then used to select a set of regressors as the model structure, and the parameter estimated using QR decomposition. The DC motor dataset was simulated to test the performance of the new model structure selection approach. The findings indicate that the BPSO-based selection method has the potential to become an excellent and effective method to determine parsimonious NARX model structure in the system identification model.**

Keywords **- System identification, Non-linear Autoregressive Model with Exogenous Inputs (NARX), DC motor.**

I. INTRODUCTION

With the advancement of technology, the importance of non-linear modelling in control engineering has been growing. Since most of practical system have inherently nonlinear characteristic such as saturation, the development of accurate nonlinear system identification algorithm is a key problem for precise analysis, prediction or control design. System identification refers to the general process of extracting information about a system from measured input-output data. It also can be defined as the task of inferring a mathematical modelling of dynamic systems based on a series of measurements collected data from the system [1, 2] to resolve practical problems [2]. For systems that have complex dynamics, it is a well and the best established technique for modelling which are not well understood or difficult to model [3].

System identification consists of two types of modelling techniques which usually employed either as linear or non-linear. The linear modelling assumes that the relationship between the input and output is linear. The model then tries to match and compared with the actual system as closely as possible (within certain parameter and computational constraints) during identification , then treats the nonlinearity as part of the uncertainty [4, 5]. Despite being relatively the linear models have certain limitations in describing various important dynamics of the actual system, since all systems are inherently nonlinear [5-10]. Since nonlinear dynamics are also incorporated in the model, the nonlinear modelling techniques are not bound by these limitations. However, the nonlinear dynamics require increase the complexity of the model structure, thus requiring an efficient model structure selection method to ensure a parsimonious model (where the system dynamics are explained in the least possible number of regressors while maintaining a good model fit).

Generally, in time series modelling the Nonlinear Autoregressive Model with Exogenous Inputs (NARX) model is a nonlinear autoregressive model which has exogenous inputs and convenient system identification model which it can describe any non-linear system well [11]. This means that the model relates the present value of the time series to both either the past values of the same series or present and past value of the driving (exogenous) series. In additional, the model contains an "error" or "residual" term which relates to the fact that knowledge of the other terms will not enable the present value of the time series to be predicted exactly.

NARX is a simplification of the well-established Nonlinear Autoregressive Moving Average with Exogenous Inputs (NARMAX) model, where the residual terms are ignored. Normally, to perform model structure selection the Orthogonal Least Squares (OLS) method is used and guided by certain information criterion the parameters for the linear least squares identification problem was simultaneously estimated [1].

In this paper, the Binary Particle Swarm Optimization algorithm (BPSO) is presented as a novel method for model structure selection of a NARX DC motor model by [1, 12, 13]. Particle Swarm Optimization (PSO) is a population-based stochastic optimization method. The (PSO) algorithm is base on the simulation social behavior of prediction by birds and the thought of swam intelligent [14, 15]. Compared to other evolutionary population-based methods, it is a powerful algorithm and highly efficient and also showing significant advantages in speed and convergence [16-18]. It is also flexible enough to apply for diverse problems [13, 16, 19, 20].

This paper is organized as follows: Section II presents the theoretical background of the DC motor model is presented first, followed by NARX, BPSO and proposed application to the model structure selection problem Section II. In Section III, the experimental setup is presented. In Section IV, the results and discussions are presented. Finally, conclusions are presented in Section V.

II. Theoretical Background

In this section, the theoretical background related to the proposed approach will be presented. In section II.A the description of the DC motor model used to generate the dataset is presented. Section II.B presents an overview of the NARX model. Lastly in section II.C the description of the BPSO algorithm and its application to the model structure selection problem is presented.

A. Derivation of DC Motor Model

Modeling and identification of mechanical system constitute an essential stage in practical control design and application. One of the components in an electromechanical system is the DC motor. The DC motor model can be constructed and developed based on its mechanical and electrical characteristics [21].

For the DC motor, V_a is the voltage source. The armature coil elements can be described with a resistance (Ω) - inductance (L) series and an induced back electromotive force (EMF) voltage. V_{emf} is generated by the rotation of the electrical coil through the flux lines of the permanent magnets inside the motor.

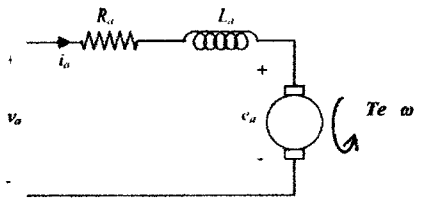

Fig 1. A schematic diagram of the permanent magnet DC motor

By consider the equation below, the electrical characteristics of the motor can be describing by given:

$$V_a - iR - L\frac{di}{dt} - V_{emf} = 0 \quad (1)$$

where:

V_a = voltage across the armature coil.
I_a = armature current.
R_a = resistance of armature coil. be
L_a = inductance of armature coil.
V_{emf} = EMF voltage
K_v = velocity constant.
ω = rotational velocity.

According to Ohm's law, the voltage across the resistor can be represented as;

$$V_{R_a} = i_a R_a \quad (2)$$

Where i_a is the armature current. The voltage across the inductor is proportional to the change of current through the coil with respect to time and can be written as

$$V_{L_a} = L_a \frac{d}{dt} i_a \quad (3)$$

Where L_a is the inductance of the armature coil. Finally, the back emf can be written as

$$V_{emf} = k_v \omega_a \quad (4)$$

Where k_v is the velocity constant determined by the flux density of the permanent magnets, the reluctance of the iron core of the armature, and the number of turns of the armature winding. ω_a is the rotational velocity of the armature.

Substituting equations. (2), (3), and (4) into equation. (1) gives the following differential equation:

$$V_a - i_a R_a - L_a \frac{d}{dt} i_a - k_v \omega_a = 0 \quad (5)$$

(i) Mechanical Characteristic

From the application of energy-balance principle [21], the mechanical characteristics of the motor are derived. The sum of torques acting during the motor's operation must be equal to zero by according to this principle. Therefore:

$$T_e - T_{\omega'} - T_\omega - T_L = 0 \quad (6)$$

Where:

T_e = electromagnetic torque = $k_f i$.
$T_{\omega'} = J\frac{d\omega}{dt}$ = torque produced by rotor's acceleration.
T_ω = torque produced by the rotor's velocity = $\beta\omega$.
T_L = torque of the mechanical load.
k_f = torque constant.
J = rotor inertia (equivalent to the mechanical load).
β = damping coefficient of the motor's rotation mechanism.

The electromagnetic torque is proportional to the current through the armature winding and can be written as;

$$T_e = k_t i_a \quad (7)$$

Where k_t is the torque constant and like the velocity constant is dependent on the flux density of the fixed magnets, the reluctance of the iron core, and the number of turns in the armature winding. $T_{\omega'}$ can be written as

$$T_{w'} = J\frac{d}{dt}\omega_a \quad (8)$$

Where, J is the inertia of the rotor and the equivalent mechanical load. The torque associated with the velocity is written as;

$$T_\omega = B\,\omega_a \quad (9)$$

Where, B is the damping coefficient associated with the mechanical rotational system of the machine.

Substituting equations. (7), (8), and (9) into equation. (6) gives the following differential equation in (10):

$$k_f i_a - J\frac{d}{dt}\omega_a - B\omega_a - T_L = 0 \quad (10)$$

(ii) State Space Representation

Based on (1) and (2), the following differential of the equations given in equations. (5) and (10) for the armature current and the angular velocity can be written as equations can be derived:

$$\frac{d}{dt}i_a = -\frac{R_a}{L_a}i_a - \frac{k_v}{L_a}\omega_a + \frac{V_a}{L_a}$$

$$\frac{d_i}{d_t} = -\frac{iR - k_v\omega - V_a}{L} \quad (11)$$

$$\frac{d}{dt}\omega_a = \frac{k_f}{J}i_a - \frac{B}{J}\omega_a - \frac{T_L}{J}$$

$$\frac{d_\omega}{d_t} = \frac{ik_f - \beta\omega + T_L}{J} \quad (12)$$

Which describe the dc motor system. Putting the differential equations into state space form gives

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_v}{L_a} \\ \frac{k_f}{J} & -\frac{B}{J} \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}\begin{bmatrix} V_a \\ T_L \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_a \\ T_L \end{bmatrix} \quad (14)$$

This is expressed symbolically as

$$\frac{d}{dt}x = Ax + Bu \quad (15)$$

$$y = Cx + Du \quad (16)$$

Where x is the state vector, u is the input vector, and y is the output vector.

(iii) Transfer Function Block Diagram

A block diagram for the system can be developed from the differential equations given in equations. (11) and (12). Taking the Laplace transform of each equation gives. The corresponding Laplace transforms for (3) and (4) are:

$$sI(s) - i(0) = -\frac{R}{L}I(s) - \frac{k_v}{L}\Omega(s) + \frac{1}{L}V_a \quad (17)$$

$$s\Omega(s) - \omega(0) = -\frac{k_f}{J}I(s) - \frac{\beta}{J}\Omega(s) + \frac{1}{J}T_L \quad (18)$$

The initial conditions will become zero and all the variables change around a reference state [21], If perturbations around the steady-state value are considered. This causes (5) and (6) to become:

$$I(s) = \frac{-k_v\Omega(s) + V_a(s)}{L_s + R} \quad (19)$$

$$\Omega(s) = \frac{-k_f I(s) + T_L(s)}{J_s + \beta} \quad (20)$$

Finally, by assuming the load torque is constant and considering only the angular velocity as the observation of interest, the block diagram for (19) and (20) can be represented as Fig. 2. [21].

$$G(s) = \frac{G_1(s)}{1 + G_1(s)H(s)} \quad (21)$$

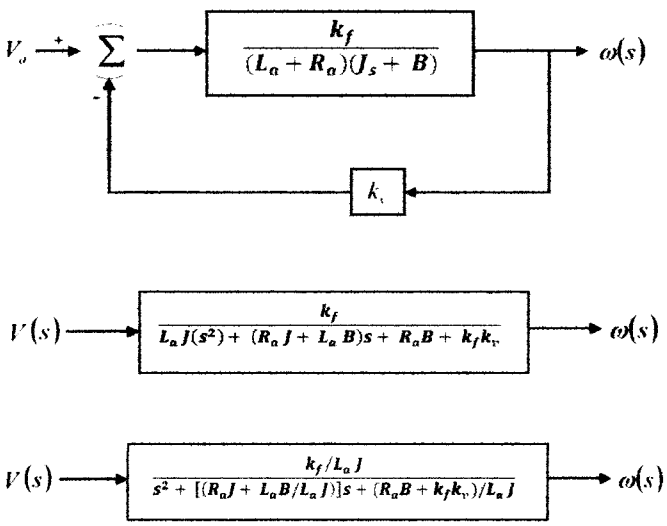

Figure 2. Overall transfer function for the DC Motor

(iv) Control System Modeling

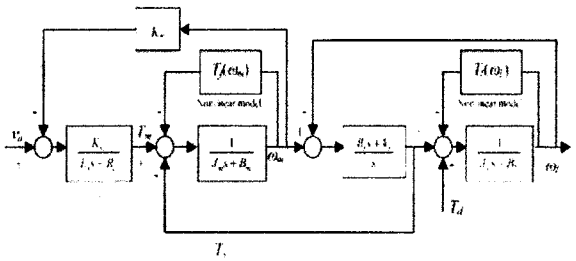

Figure 3. Nonlinear System Identification for DC Motor (control system)

Finally, by assuming the load torque is constant and considering only the angular velocity as the observation of interest, the block diagram for (7) and (8) can be represented as Fig.3 [21].

B. Description of NARX

The Nonlinear Autoregressive Model with Exogenous Inputs (NARX) model is a general and convenient system identification model. Consider the class of discrete-time nonlinier system that can be represent by the following NARX structure [11]. The NARX model is defined as:

$$y(t) = \begin{matrix} f^d[y(t-1), y(t-2), \ldots, y(t-n_y), \\ u(t-1), u(t-2), \ldots, u(t-n_u)] + \varepsilon(t) \end{matrix} \quad (22)$$

Where, f^d are the coefficients, $y(t-1), y(t-2), \ldots y(t-n_y)$ are lagged output terms, $u(t-1), u(t-2), \ldots u(t-n_u)$ are lagged input terms, and $\varepsilon(t)$ are the white noise residuals.

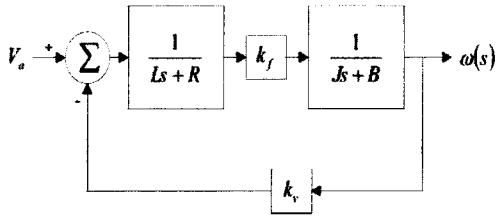

Fig.4. Block representation of DC motor model.

Where y and u is the system input an output variable. The selection of model structure involves selecting which lagged y and u terms and estimating the parameters that can explain the future values of y.

C. Particle Swarm Optimization (PSO)

The PSO algorithm is base on population two socio-metric principles or also know as stochastic optimization technique. It taking by looking into animal swarming behavior in nature [22] which bird are the optimization corresponds in search space or called particle. By taking advantage of the cooperative and competitive behavior of particles, PSO iteratively searches for solutions in the problem space.

The particle swarm optimization with constriction factor (PSO_{CF}) was chosen due to its convergence properties compared to the inertia weight variant will considered for implemented in this paper [1]. To decrease the iteration progress of particle velocities, the particle moving into the nearest the optimum are localized by the PSO_{CF} method updated the original velocity equation.

The modified equation for PSO_{CF} adds the constriction factor parameter, χ, to the velocity update equation:

$$V_{id} = \chi[V_{id} + C_1(pBest - X_{id}) \times rand_1 + C_2(gBest - X_{id}) \times rand_2] \quad (23)$$

Where:

V_{id} = particle velocity.
X_{id} = particle position.
$pBest$ = particle's best fitness so far.
$gBest$ = best solution achieved by the swarm so far.
C_1 = cognition learning rate
C_2 = social learning rate.
$rand_1$, $rand_2$ = random numbers between 0 and 1.

χ is calculated using:

$$\chi = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \quad (24)$$

where:

$$\varphi = C_1 + C_2, \varphi > 4 \quad (25)$$

After new velocities have been calculated, X_{id} is updated:

$$X_{id} = X_{id} + V_{id} \quad (26)$$

Parameter $vMax$ is the maximum velocity required and it functioning as a constraint to prevent velocity explosion [19, 23]. Generally, $vMax$ is set as the dynamic range of values [24].

By using parameters of $xMin$ and $xMax$, the value of X_{id} may be bounded to ignore solutions outside an acceptable range [17]. Whenever X_{id} violates $xMin$ or $xMax$, they are artificially brought back to their nearest side constraint. Additionally, to discourage any more searches in that direction [17], the velocity will set to 0 each time this occurs.

D. BPSO for Model Structure Selection

Models of binary decisions [25-27] to solve binary problems the PSO algorithm represents particle positions as "probabilities of change" rather than the actual solution. In other meaning is the probability that an individual's decision will be yes or no, true or false, or some other binary decision is a function of personal and social factor. To make the statement more clearly, it can be transform into the values of the particles indicated. The probability of the bit is given a binary string of x bits, it to change from its current state to another (bit flip). For example the indicated value, the bit will change from its current state (from 0 to 1 or 1 to 0) when a particle value higher than 0.5. Else, the bit will maintain its current state if the particle value is less than 0.5.

The application of the BPSO for the system identification problem is described – The system identification problem can be defined as a linear least squares problem:

$$P\theta + \varepsilon = y \quad (27)$$

Where, P is a $n \times m$ regressor matrix, θ is a $m \times 1$ coefficient vector, and y is the $n \times 1$ vector of actual observations. P is arranged such that the columns represent the m lagged regressors. ε is the white noise residuals.

Based on Eq. (22), the regressor matrix P, is a matrix of combinations of lagged input and output terms. A binary string of length $1 \times m$ is defined, so that each regressor column has a bit assigned to it. The initial value of the binary string can be randomly defined during initialization. A value of 1 indicates that the column will be considered in the construction of the reduced regressor matrix, while the value of 0 indicates that the regressor column is ignored.

In the swarm, the particles each carry a $1 \times m$ vector if solutions x_{id}. This vector contains the "probabilities of change" defined earlier. These values change during optimization and alter which regressor column is selected.

The linear least squares solution for the reduced regressor matrix, $P_{reduced}$ can then be estimated based on a by-product of the error reduction ratio (ERR) method described in [11]. For the selected regressors, the coefficients (θ) was estimated by using the Householder-based QR decomposition.

The QR decomposition of matrix is:

$$P_{reduced} = QR \tag{28}$$

Let:

$$g = Q^T y \tag{29}$$

Such that:

$$R\theta = g \tag{30}$$

Solving equation in (30) yields the solutions for the linear least squares problem in Eq. (27).

III. METHODOLOGY

E. Hardware Description

To make all the experimental testing, it was run by using the Personal Computer (PC) computer: Acer Aspire model 5551 with process by Intel Centrino Duo Central Processing Unit (CPU) (running at 1.66 GHz) with 2 GB of Random Access Memory (RAM). Microsoft Windows 7 Home Premium (64-bit) was installed as the operating system. The all program were implemented and simulated using MATLAB version 7.8.0.347 (R2009a).

1) Dataset Description

The NARX model was created based on time series data of a Matlab Simulink model based on the transfer function described in Section II.A, and 5,000 test cases were simulated. The dataset was then divided equally into training and testing set. The training and validation sets are shown in Fig. 5. Prior to training the NARX model the dataset was rescaled so that the input and output data resides between 0 and 1.

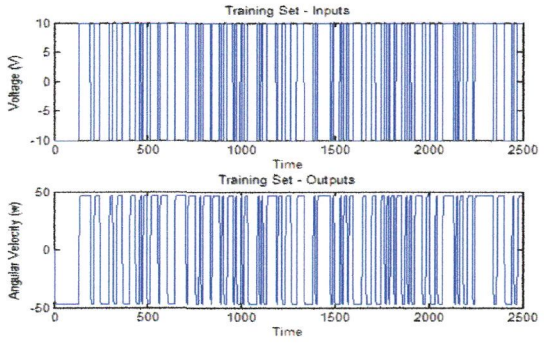

Fig.5 Inputs and outputs for training set.

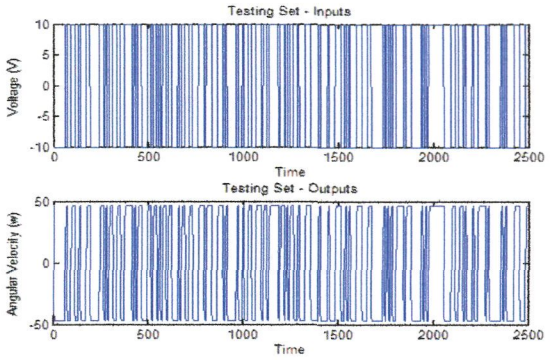

Fig. 6 Inputs and outputs for testing set.

F. PSO Parameters Description

The parameter settings that used for PSO_{CF} will explain details in this section which the main objective of PSO_{CF} is to select which one of regressors to include in the regression matrix until the Normalized Sum Squared Error (NSSE) value reaches 0. This NSSE fitness function and objective were chosen similar to the work by [21].

During the search, *xMin* and *xMax* are used to constrain the movement of particles and it was described in Section II.C. Respectively, the values of *xMin* and *xMax* were set to 0 and 1. For each of iteration it was controlled by using the values of *vMin* and *vMax* and it for the maximum movement of the particles. Usually, these values are set to the dynamic range of the variable [24]. Therefore, the dynamic range of *vMin* and *vMax* were respectively set to -1 (when V_{id} moves from 1 to 0) and 1 (when V_{id} moves from 0 to 1), since it *xMin* and *xMax* was set to 0 and 1, respectively.

Additionally, they were artificially brought back to the side constraints (*xMin* and *xMax*) if at any time X_{id} violates the *xMin* and *xMax* constraints. To discourage any more searches in that direction , V_{id} was set to 0 [17] if this happens. From MATLAB's pseudo-random number generator, the random parameters (*rand1* and *rand2*) were generated using the Mersenne-Twister algorithm (MTA) [28].

The values of C_1 and C_2 differ according to the particle swarm optimization algorithm. For PSO_{CF}, the values of C_1 and C_2 were both set to 2.05 [26] (to avoid violating the rule in [15]).

The value of χ is a function of C_1 and C_2 for PSO_{CF} (see [15]). The value of χ is 0.7290 throughout the optimization course and it was based on the values of C_1 and C_2.

G. NARX Parameters Description

At this section, the parameters used in the NARX model throughout the experiments will describe. Usually, the NARX model is similar to the model used. A short description is presented here. The "number of delayed signals used as regressors" will called the NARX model lags [29].

In [21], by using Lipschitz Analysis (LA), the lag spaces for the inputs and outputs were both determined to be 2. Therefore, the maximum lag space was set to be 2 for input and output data.

The information criterion used to guide the structure selection process is the Minimum Descriptor Length (MDL) criterion [1]. The MDL is given by:

$$V_{MDL} = \left(1 + \log(N)\frac{d}{N}\right) V_N(1, Z^N) \quad (31)$$

Where;

V_{MDL}= MDL information extension
N = Length of data
d = number of regressors selected
$V_N\ (1, 2^N)$= mean squared error for residuals

H. Swarm Initialization using MTA

PSO convergence is sensitive to its initial particles values. The selecting of optimal number of particles may be difficult since the convergence is unpredictable and the particles are initialized randomly before optimization. To test the effectiveness of the proposed method, each experiment was repeated 5 times with different initialization values for each particle by using a pseudo-random number generator called the Mersenne-Twister algorithm (MTA).

In MTA, the sequence of random numbers generated is determined by the internal state of the generator. Setting the generator to different states leads to unique computations and outcomes for each state. The unique computations result in the generation of unique series of random numbers based on the state. To ensure repeatability of the experiments, the generator state is set to some fixed value each time the optimization executes to ensure that the same set of random numbers are generated. The initial seed from each test is shown in Table I.

TABLE I
STARTING PSEUDO-RANDOM SEEDS USED FOR EACH PARTICLE TEST

Test	Initial Seed
1	0
2	50,000
3	100,000
4	150,000
5	200,000

The convergence of the BPSO algorithm was tested over several different initial states (Table 1). The initial MTA state (for the first test) starts from 0 and increased by 50,000 for the next test. For example, in the first test, the initial MTA state was set to 0. For the next test, the initial MTA state was set to 50,000, and so on, until the 5th test. This was done to measure the convergence with different initial particle values. The swarm size was set to 50, which was sufficient based on preliminary tests. The maximum iteration for the BPSO algorithm was 100, while all bits in the initial binary string were set to zero.

IV. RESULTS & DISCUSSION

The output of our MATLAB program is shown in Fig. 7

```
Model: narx
Dataset: asmadi
Original Model: DC Motor
Criterion: mdl
Orthogonalization: house
Norm (Training Set): 1.3541
SSE (Training Set): 1.8337
MSE (Training Set): 0.00073495
Norm (Testing Set): 1.3342
SSE (Testing Set): 1.78
MSE (Testing Set): 0.00071341

Regressors      Value
u - 1           2.3292
u - 2           0.038906
y -1            0.49114
```

Output of identification program.

The NARX identification result is in Eq. (32).

$$y(t) = 2.3292u(t-1) + 0.0389u(t-2) + 0.4911y(t-1) + 1.0616 \times 10^{-5}u(t-1)u(t-2) \quad (32)$$

The indicating of a good model fit is by an accurate simulation result coupled with small, uncorrelated random residuals. In Fig. 8 and Fig. 9, it has shown the one-step-ahead prediction results. The model fit is very good by indicated the closeness of the identified signal and the output signal for both the training and testing sets and all it will can be seen from the results. The sum-squared error (SSE) and mean-squared error (MSE) is shown in Table II. Both the values of SSE and MSE were very small, indicating a good fit.

TABLE II
STARTING PSEUDO-RANDOM SEEDS USED FOR EACH PARTICLE TEST

Test	Training Set	Testing Set
SSE	1.8309	1.7761
MSE	7.3382×10^{-4}	7.1185×10^{-4}

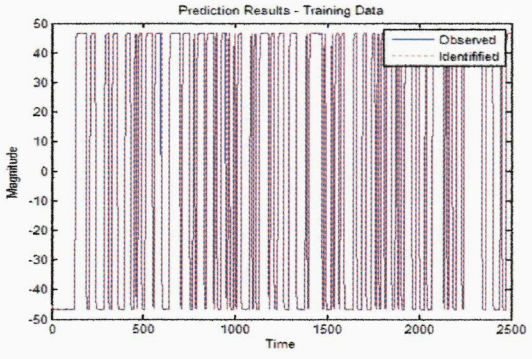

Fig. 8 Identification result for training set. Solid line indicates actual (observed) output; dotted line indicates NARX model (predicted) output.

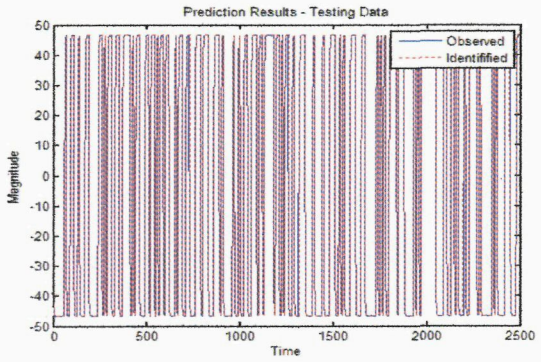

Fig.9 Identification result for testing set. Solid line indicates actual (observed) output; dotted line indicates NARX model (predicted) output.

The residual plots for the training and validation sets are shown in Fig. 10 and Fig. 11. The residuals for both cases were very small, indicating very small difference between the predicted and actual observations.

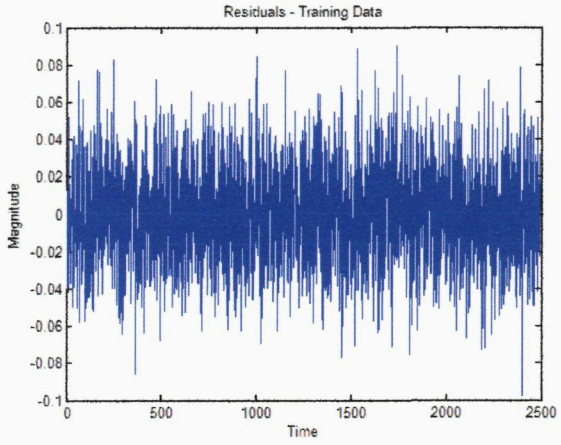

Fig.10. Residual plot for training set.

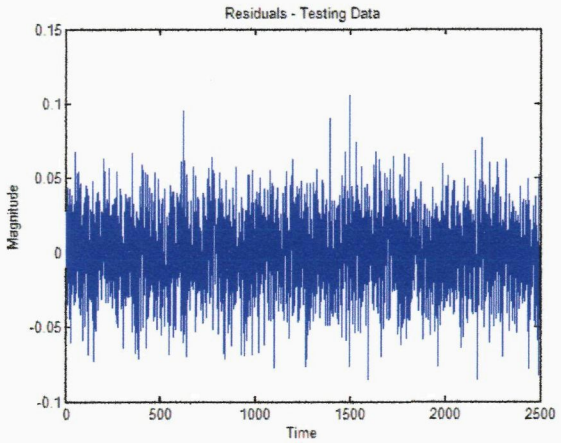

Fig.11. Residual plot for testing set.

A good model fit would leave only white-noise residuals behind and it will determine from Eq. (32). The autocorrelation test on the residuals was performed to test the fit in terms of the residuals, and cross-correlation between the residuals and the inputs. In Fig. 12 to Fig. 13, the results of the correlation tests are shown:

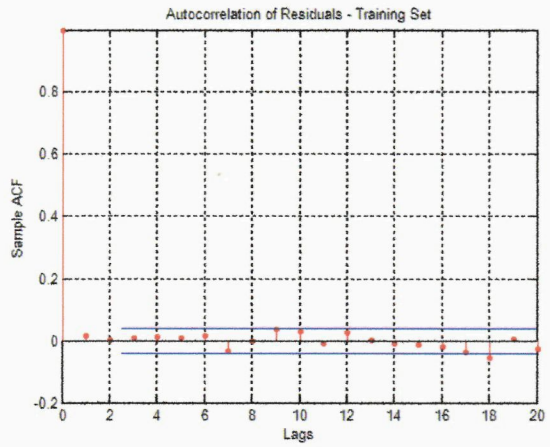

Fig.12. Auto-correlation of residuals for training set, solid line indicates 95% confidence limits.

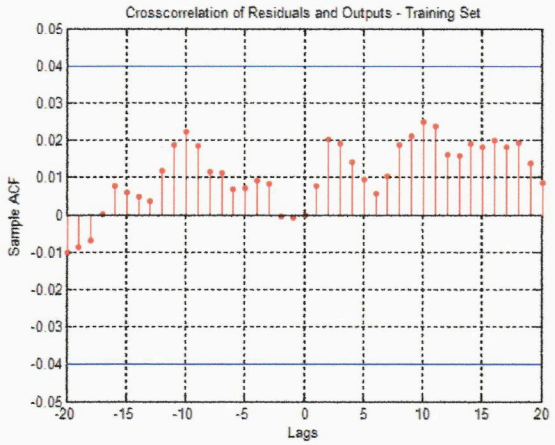

Fig.13. Cross-correlation between residuals and input of training set, solid line indicates 95% confidence limits.

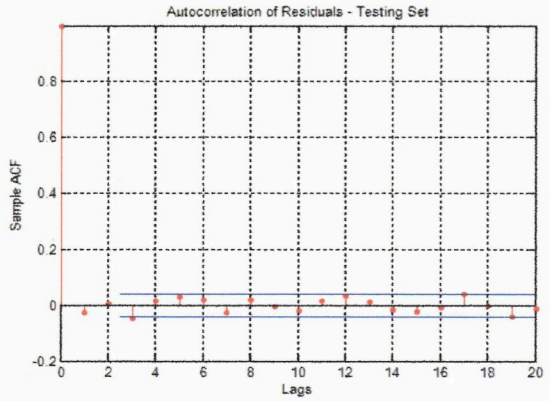

Fig.14. Auto-correlation of residuals for testing set, solid line indicates 95% confidence limits.

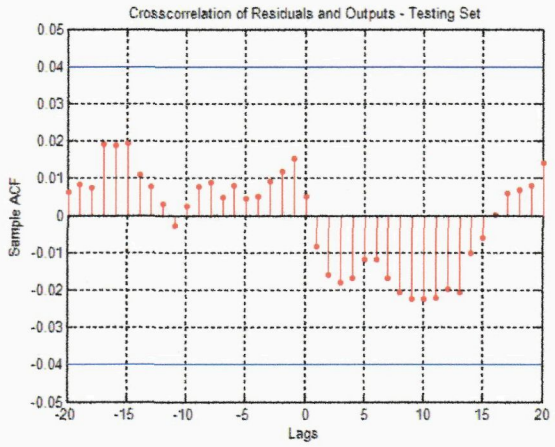

Fig. 15. Cross-correlation between residuals and input of testing set, solid line indicates 95% confidence limits.

By distribution of Gaussian, white noise also has a distinct characteristic. In Fig. 14 and Fig. 15 the residual histograms are shown and clearly show the Gaussian distribution.

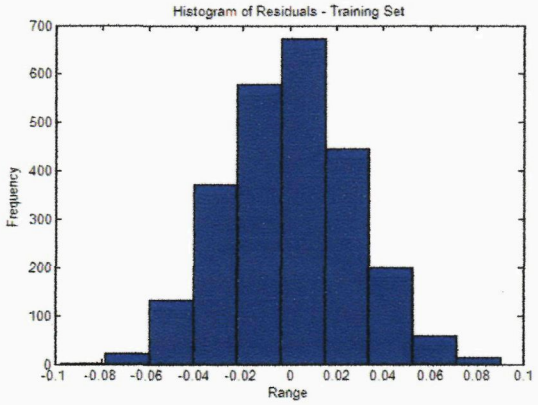

Fig. 16. Histogram of residuals for training set.

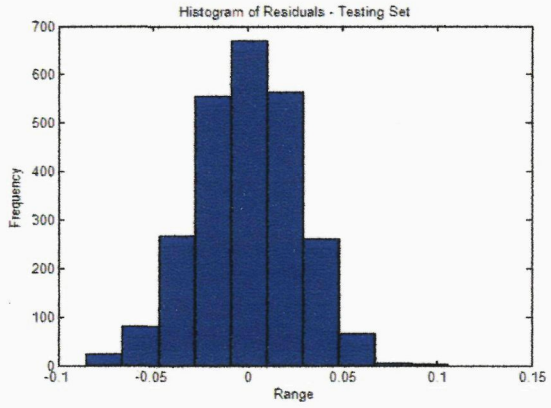

Fig.17. Histogram of residuals for testing set.

V. Conclusions

A DC motor model by [13] has been identified using the NARX method. The model structure selection process was done using the binary PSO. One step ahead and residual tests indicate that the model is valid, and the suitability of the PSO algoritm to perform model structure selection in NARX models.

Acknowledgment

This project was implemented under final year project for the final year student, Faculty of Electrical Engineering UiTM. The author would also like to hank Universiti Teknologi Mara for their equipment support to complete the project.

References

[1] M. N. Taib, "Series Modelling and Prediction using Neural Networks," M. S. Thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield,UK, 1993.

[2] X. Deng, "System identification Base on Particle Swarm Optimization Algorithm," presented at the International Conferenceon Computational Intelligence and Security, Guangzhou, Guangdong, 510006,PR China, 2009.

[3] Z. Chen, *et al.*, "Dynamic modeling using cascade-correlation RBF networks for tilt rotor aircraft platform," presented at the Proc. Int. Conf. on Neural Networks and Brain, 2005.

[4] L. K. Chen and A. G. Ulsoy, "Identification of a nonlinear driver model via NARMAX modeling," presented at the Proc. American Control Conf., Chicago, IL, 2000.

[5] M. Solomou and D. Rees, "System modeling in the presence of nonlinear distortions," presented at the Instrumentation and Measurement Technology Conf., Lake Como, Italy, 2004.

[6] L. d. S. Coelho and R. A. Krohling, "Nonlinear system identification based on B-spline neural network and modified particle swarm optimization," presented at the Int. Joint Conf. on Neural Networks, Vancouver, Canada, 2006.

[7] N. Chiras, *et al.*, "Nonlinear gas turbine modeling using NARMAX structures," *IEEE Trans. Instrumentation and Measurement*, vol. 50(4), pp. 893-898, 2001.

[8] Y. Bi, *et al.*, "A new nonlinear system identification method using gene expression programming," presented at the Proc. of IEEE Int. Conf. on Mechatronics and Automation, Harbin, China, 2007.

[9] N. R. Butt, *et al.*, "An adaptive root-solving controller for tracking of nonlinear dynamic plants," *Proc. Int. Conf. on Industrial Electronics and Applications*, pp. 6-6, 2005.

[10] C. A. Sierakowski, *et al.*, "Particle swarm optimization approach for multi-step ahead prediction using radial basis function neural network," presented at the Proc. 15th IFAC World Congress, Praga, Portugal, 2005.

[11] B. A. Amisigo, *et al.*, "Monthly streamflow prediction in the Volta Basin of West Africa: A SISO NARMAX polynomial modelling," *Physics and Chemistry of the Earth*, vol. 33(1-2), pp. 141-150, 2007.

[12] C. R.-L. R. Salas-Cabrera, E. N. Salas-Cabrera, M. Gomez-Garcia, C. Garcia-Guendulain & H. E.Sanjuan-Garcia., "Identification of a Nonlinear Model For a DC Motor," Department of Electrical and Electronic Egineering, Instituto Tecnologico de Cd. Madero.

[13] N. A. Rahim, "The NARMAX Model for DC Motor using MLP Neural Network," Electronic and Biomedical Intelligent System Group (EBItS), School of Mechatronic Engineering Kolej University Kejuruteraan Utara Malaysia, Arau, Perlis, Malaysia.

[14] M. Settles, "AnIntroduction to Particle Swarm Optimization," presented at the Idaho, Department of Computer Science, University of Idaho,Moscow,Idaho U.S.A 2005.

[15] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, 1998, pp. 69-73.

[16] W. H. Slade, *et al.*, "Inversion of ocean color obsevations using particle swarm optimization," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 42, pp. 1915-1923, 2004.

[17] R. Hassan, *et al.*, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," presented at the 1st AIAA Multidisciplinary Design Optimization Specialist Conf., Austin , TX, 2005.

[18] S. Kannan, *et al.*, "Application and comparison of metaheuristic techniques to generation expansion planning problem," *IEEE Trans. on Power Systems*, vol. 20, pp. 466-475, 2005.

[19] M. G. H. Omran, *et al.*, "Dynamic clustering using particle swarm optimization with application in unsupervised image classification," *Trans. Engineering, Computing and Technology*, vol. 9, pp. 199-204, 2005.

[20] X. Cui, *et al.*, "Document clustering using particle swarm optimization," presented at the Proc. IEEE Swarm Intelligence Symposium, Pasadena, CA, 2005.

[21] N. A. Rahim, "The design of a non-linear autoregressive moving average with exegenous input (NARMAX) for a DC motor," MSc, Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, 2004.

[22] X. Hu, *et al.*, "Recent advances in particle swarm," in *IEEE Congress on Evolutionary Computation*, Portland, Oregon, USA, 2004, pp. 90-97.

[23] L. Khriji and K. El-Metwally, "Rational-based particle swarm optimization for digital image interpolation " *Int. J. Intelligent Technology* vol. 1, pp. 223-227, 2006.

[24] X. Hu and R. C. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," in *Sixth World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, USA, 2002.

[25] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive swarm optimization," presented at the In Congress on Evolutionary Computation (CEC99), 1999.

[26] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," presented at the Proc. Congress of Evolutionary Computation, San Diego, CA, 2000.

[27] j. K. a. R.C.Eberhart, "A discreet binary version of the particle swarm algorithm," 1997.

[28] M. M. a. T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8(1):3, pp. 3-30, 1998.

[29] M. Norgaard, *et al.*, *Neural networks for modeling and control of dynamic systems: A practitioner's handbook*. London: Springer, 2000.