Evolutionary Algorithm for Identification of a Flexible Single-link Manipulator System

ALI A. M. AL-KHAFAJI, NIK M.R. SHAHARUDDIN, INTAN Z. M. DARUS  
Department of Applied Mechanics & Design, 
Faculty of Mechanical Engineering, 
Universiti Teknologi Malaysia, 
81310, Johor Bahru 
MALAYSIA  
ali1977abdalhusain@gmail.com, nmridzuan2@hotmail.my, intan@fkm.utm.my

Abstract: - This study presents an investigation into dynamic modelling of a flexible single-link manipulator system using differential evolutionary technique (DE) and particle swarm optimization technique (PSO). Details of simulation study, Model Selection, optimization and result analysis are given in this study. The input-output data of the system were first acquired through the simulation study using finite element method (FDM) based on Lagrangian approach. A bang-bang torque was applied as an input and the dynamic responses of the system were investigated. Next, an appropriate model structure was chosen and optimized using DE and PSO. One Step Ahead (OSA) prediction, correlation tests and mean squared error (MSE) have been performed for validation and verification of the obtained model in characterizing the manipulator system. Furthermore, an unseen data was used to observe the prediction ability of the model. A comparative assessment of the two models in characterizing the manipulator system is presented in time and frequency domains. Results demonstrate the advantages of DE over PSO in parametric modeling of the flexible manipulator system used in this study.

Key-Words: - Differential evolutionary algorithm, particle swarm optimization, flexible single-link manipulator, system identification, finite element method, dynamic modelling.

1 Introduction

Modern flexible link manipulator systems are widely used in many areas such as space vehicles, medical, defense and automation industries. The flexible link manipulator system, compared with rigid link manipulator system offers many advantages including low inertia, light weight, few powerful actuators, cheaper construction, fast response, safer operation, higher payload carrying capacity and longer reach. In general, the flexibility in modern robot manipulators system is an unwanted feature since it causes a serious of vibration problem when subjected to disturbance forces. Therefore the control of the flexible link manipulator robot becomes much more complex than rigid link robots. Accordingly, there is a growing need to develop suitable control strategies for such kind of systems [1], [2].

To control the vibration of a flexible link system effectively, it is often required to obtain an accurate or approximate model of the system structure, which will result in good control. Thus, finding an appropriate model of a dynamic system, such as a flexible system, would be the key to design an effective controller to suppress the vibration in the flexible link manipulator system [1].

The process of constructing a structure that characterizes the dynamics of a system is called system modelling. Different modeling techniques for analysis the flexible manipulators are studied by a number of researchers [1]. Mathematical method and numerical analysis have been proposed, developed and used by researchers. Through mathematical analysis, the dynamic model of the flexible manipulator dynamics is represented in the form of a partial deferential equation (PDE). The exact solution of such systems is not feasible practically and the infinite dimensional model imposes severe constraints on the design of controllers as well. Hence, the numerical analysis method was used to represent the flexible manipulator dynamics by solving the PDE using assumed modes, finite elements or lumped parameter methods [1], [2], [3].

To avoid solving the dynamics of flexible single-link manipulator, it can be modelled using system identification approaches [4]. System identification is the process of developing an accurate or
approximate dynamic model of a physical system using a set of input-output data pairs collected at input-output terminals of the system using experimental or simulation study. Modeling from first principles requires an in-depth knowledge of the system. System identification techniques can represent several system dynamics without knowledge of the actual physics of the system. This is one of the advantages of using the system identification method to represent the model of physical systems.

System identification consists of parametric and nonparametric techniques. Parametric modelling constitutes building a linear mathematical function of a dynamic system based on measured data in form of partial differential/difference equation or transfer function. Non parametric modelling is an attempt to represent plant with an input-output behavioral box, not as an explicit mathematical function.

There are two stages for these methods. The first stage is known as characterization, in which the assembly of the model such as type and order of the differential/difference equation that relate the input to the output is defined. The second stage is called identification, in which the numerical values of the structural parameters are determined. Construction of a model based on input-output data pair consist of four main components [5]:

1. Collection of a set of input-output data from experiment or simulation. The accuracy of the model is highly dependent on the accuracy of the collected data.
2. Determination of a suitable model structure from a set of candidate models.
3. Model estimation, which means using an assessment mechanism to determine the best model. The estimated model must have similar properties to that of the true one, it should be able to simulate the dynamic system, and predict the future values of the output.
4. Model validation, which means verifying whether the model is sufficient and if it has fulfilled the necessary requirements in representing the system.

Several parametric and nonparametric estimation methods can be found in the literature. Researchers and engineers have been developing and applying hard computing methods and soft computing methods [6]. One of the parametric soft computing methodologies is differential evolutionary algorithm.

Many parametric and nonparametric estimation techniques have been employed as optimization tools in identification of flexible link manipulator such as least mean square (LMS) and recursive least squares (RLS) [7], [8]; genetic algorithm (GA) [7], [9]; particle swarm optimization (PSO) [10], [11]; neural networks (NNs) [12], bacterial foraging algorithms (PFA) [13]. It can be noted from the literature that the use of DE for modelling manipulators has not been reported yet.

In this study, DE is compared to PSO in term of identification of a single-link flexible manipulator. First, simulation environment characterizing the dynamic behavior of the flexible manipulator structure is developed using FEM based on Lagrangian approach. Then, DE and PSO based identification are carried out using the input-output data acquired. The results are obtained in both time and frequency domains. The validity of the obtained model also will be investigated using OSA prediction, correlation tests and MSE.

2 Parametric system identification
Parametric system identification constitutes building mathematical model of a dynamic system based on measured data. So, the first step is to obtain the input-output data pairs through simulation or experimental work. Then, define a suitable model structure that could represent the system. After determining the model structure, the main task of identification is to estimate the model parameters, which are usually determined on the basis of a global minimum criterion function. The final step is to verify whether the model is adequate or not using validation tests.

2.1 Flexible Manipulator System
The flexible single-link manipulator system considered in this study is shown in Fig. 1. The link has been modelled as a pinned-free flexible beam. The pinned end of the flexible beam of length \( L \) is attached to the hub with inertia \( I_z \), where the input torque \( \tau(t) \) is applied at the hub by a motor, and a payload mass \( M_p \) is attached at the free end. \( e \), \( i \) and \( \rho \) represent the Young’s modulus, second moment of inertia and mass density per unit length of the flexible manipulator respectively. \( X_0Y_0 \) axis and \( XY \) axis represent the stationary and moving coordinate respectively. Both axes lie in a horizontal plane and all rotation occurs about a vertical axis passing through \( o \).
The flexible link, viewed as an Euler-Bernoulli beam, is modeled based on the following assumptions and constraints:

- The flexible link is viewed as a pinned-free flexible beam.
- The flexible link is assumed to be moving in the horizontal plane. Thus, perpendicular deformation is neglected as the gravitational term is not considered.
- Cross-section area and material properties remain constant in each segment.
- The effect of rotary inertia and shear deformation are ignored.

The mathematical modeling of a flexible single-link manipulator was derived using a finite element method based on Lagrangian approach [14], [15], [16]. The main step in a finite element analysis can be divided into six sections as follows:

1. Decompose the structure into finite elements, which are assumed to be interconnected at certain points. Number of elements determines the accuracy of analysis.
2. For each element, select an approximating polynomials function describing the behavior of the element using an approximation technique to interpolate the result.
3. Formulate the element characteristic matrices and vectors. The equations can be derived from the properties of the material and kinetic and potential energies.
4. Assemble the element matrices and vectors and derive the system equation. This equation describes the dynamic behavior of the system.
5. Incorporate the boundary condition. These conditions will prevent the structure from unlimited body motion.
6. Solve the system equations with the inclusion of the boundary condition.

In this manner, the overall approach involves treating the link of the manipulator as an assemblage of \( N \) elements of equal length \( l = \frac{L}{N} \). For each of these elements the kinetic energy \( T \) and potential energy \( U \) are computed in terms of a suitably selected system of five generalized variables \( Q \) and their rate of change \( \dot{Q} \). The development of the algorithm can be divided into three main steps: finite element method and Lagrangian approach analysis, state space representation and obtaining the result. An outline of this process is given below:

For a small angular displacement \( \theta(t) \) and small elastic deflection \( v(x,t) \), the overall displacement \( y(x,t) \) of a point along the link at a distance \( x \) from the hub can be described as a function of both the rigid body motion \( \theta(t) \) and elastic deflection \( v(x,t) \) measured from the line passing through \( O \):

\[
y(x,t) = x\theta(t) + v(x,t)
\]  

Using the standard FEM to solve dynamic problems, the flexible displacement \( v_i(x,t) \) can be approximately expressed as:

\[
v_i(x,t) = N_s(s) Q_i(t)
\]  

where \( s \) represent the local variable, \( N_s(s) \) and \( Q_i(t) \) represent the shape function and nodal displacement respectively. As a consequence of using the Euler-Bernoulli beam theory, the finite element method requires each the nodes through which we divide the beam into elements to possess two degrees of freedom, a transverse deflection and rotation at the same time when the beam is rotating. The displacements of two nodes of the ith element are denoted by \( v_n \) and \( v_{n+1} \), and the rotations of the nodes by \( \theta_n \) and \( \theta_{n+1} \). Because there are \( N \) elements in total, the number of generalized variables \( Q \) equal \( 2N + 2 \). These necessitate the use of Hermite cubic basis functions as the element shape function. Hence, the shape function can be obtained as:

\[
N_n(s) = [\phi_1(s) \, \phi_2(s) \, \phi_3(s) \, \phi_4(s)]
\]  

where,

\[
\phi_1(s) = 1 - \frac{(3s^2)}{l^2} + \frac{(2s^3)}{3l^3}
\]

\[
\phi_2(s) = -s + \frac{(2s^2)}{l}
\]

\[
\phi_3(s) = \frac{(3s^2)}{l} - \frac{(2s^3)}{3l^2}
\]

\[
\phi_4(s) = \frac{(s^3)}{l^2}
\]

and the generalized variables

\[
Q_i(t) = [v_i(t) \, \theta_i(t) \, v_{n+1}(t) \, \theta_{n+1}(t)]^T
\]

Substituting equation (2) into (1) gives...
\[ y(x,t) = x\theta(t) + \sum_{i=1}^{n} N_i(s)Q_i(t) \] (3)

where,
\[ N_i(s) = [x N_i(s)] \] (4)
\[ Q_i(t) = [\theta(t) \ Q_i(t)]^T \] (5)

The new shape function \( N_i(s) \) in equation (4) and new nodal displacement vector \( Q_i(t) \) in equation (5) incorporate local and global variables. Among these, the angle \( \theta(t) \) and the distance \( x \) are global variables while \( N_i(s) \) and \( Q_i(t) \) are local variables.

Define, \( s = x - \sum_{i=1}^{n} l_i \) as a local variable of the \( i \)th element, where \( l_i \) is the length of the \( i \)th element. The equations (4) and (5) can be expressed as:
\[ N_i(s) = [s + l(N-1)] N_i(s) \] (6)
\[ y(s,t) = N_i(s)Q_i(t) \] (7)

According to the basic finite element method and energy principle, the kinetic energy and potential energy of \( i \)th element of the link can be acquired according to equation (8) and (10) as follows:
\[ T_e = \frac{1}{2} \int_{0}^{1} \rho A \left( \frac{\partial y(s,t)}{\partial t} \right)^2 ds = \frac{1}{2} \int_{0}^{1} \rho A \dot{y}^T \dot{y} ds \]
\[ = \frac{1}{2} \dot{Q}_i^T \int_{0}^{l} \rho A (N_i^T N_i)ds \dot{Q}_i \] (8)

where
\[ M_i = \int_{0}^{l} \rho A (N_i^T N_i)ds \]

is element mass matrix
\[ P = \frac{1}{2} \int_{0}^{1} EI \left[ \frac{\partial^2 y(s,t)}{\partial s^2} \right]^2 ds \]
\[ = \frac{1}{2} \int_{0}^{l} EI (B_i \dot{Q}_i)^T (B_i \dot{Q}_i) ds \] (9)

\[ = \frac{1}{2} \dot{Q}_i^T \int_{0}^{l} EI (B_i^T B_i) ds \dot{Q}_i \]
\[ = \frac{1}{2} \dot{Q}_i^T [K_i] \dot{Q}_i \]

where
\[ K_i = \int_{0}^{l} EI (B_i^T B_i) ds \]

is element stiffness matrix
\[ B_i = \frac{d\ N_i(s)}{ds} \]

For a flexible single-link manipulator system, the mass matrix consists of three terms; mass matrix due to the structural mass of the manipulator, mass matrix due to hub inertia and mass matrix due to tip payload mass. A brief outline of the mass matrices is given below. The kinetic energy of the \( i \)th element due to structural mass of the manipulator can be obtained using equation (8). Thus,
\[ T_e = \frac{1}{2} \dot{Q}_i^T [M_i] \dot{Q}_i \]

(12)

the element structural mass matrix yields:
\[ [M_i] = \frac{\rho A l_i}{420} \]

(13)

where \( A_i \) is the cross section area of the manipulator, and \( \rho \) is the density of the material of the manipulator and
\[ m_{11} = 140l_i(3N_i^2 - 3N_i + 1) \]
\[ m_{12} = m_{13} = 2l_i(10N_i - 7) \]
\[ m_{14} = m_{15} = 7l_i(5N_i - 3) \]
\[ m_{16} = m_{17} = -7l_i(5N_i - 2) \]

and the generalized variables are \( Q_i(t) = [\theta(t) \ \dot{\theta}(t) \ v_i(t) \ \dot{v}_i(t) \ \theta_{i1}(t) \ \theta_{i2}(t)] \). The kinetic energy of tip payload mass can be found using equation (14):
\[ T_p = \frac{1}{2} M_{p} \dot{\theta}_{i}^T \dot{\theta}_{i} \]
\[ = \frac{1}{2} \dot{Q}_i^T [M_{p}] \dot{Q}_i \]

(14)

where \( [M_{p}] = M_{p} N_i^T N_i \). Hence, the inertia matrix of tip payload mass \( M_{p} \) is:
\[ [M_{p}] = \begin{bmatrix} L^2 M_p & 0 & 0 & LM_p & 0 \\ 0 & 0 & 0 & 0 & 0 \\ LM_p & 0 & 0 & M_p & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

(15)

and the generalized variables are \( Q_i(t) = [\theta \ \dot{\theta} \ v_i \ \dot{v}_i \ \theta_{i1} \ \theta_{i2}] \). Rotational kinetic energy of driving end can be found using equation (16):
\[ T_h = \frac{1}{2} I_h \left( \frac{\partial \theta(t)}{\partial t} \right)^2 = \frac{1}{2} I_h \dot{\theta}_{i}^T \dot{\theta}_{i} \]

(16)

The inertia matrix of driving end is expressed as:

E-ISSN: 2224-2856

Volume 10, 2015
The dynamic equations of motion of the flexible manipulator can be derived utilizing the Lagrange equation based on general coordinates:

$$\frac{d}{dt} \left[ \begin{array}{c} \frac{\partial L}{\partial \dot{q}} \\ \frac{\partial L}{\partial \ddot{q}} \end{array} \right] = F$$

(23)

where \( L = T - P \) is the Lagrangian and \( F \) is a vector of external forces. Considering the damping, the desired dynamic equations of motion of the system can be obtained as:

$$M^* \ddot{Q} + D^* \dot{Q} + K^* Q = b^T \tau$$

(24)

where \( M^* \) is the global mass matrix which consist of the structural mass, added mass of fluid, payload mass and hub mass. \( K^* \) is global rigidity matrix and \( \dot{Q} \) is the general coordinates when substituting boundary conditions. \( b^T \) is the nodal displacement given as:

$$b^T = [\theta \ w_2 \ \theta_2 \ \cdots \ w_N \theta_N]$$

\( D^* \) is structural damping matrix due to structural material. For the flexible manipulator under consideration, the global mass matrix can be represented as:

$$M^* = \begin{bmatrix} M_{\omega\omega} & M_{\omega\theta} \\ M_{\theta\omega} & M_{\theta\theta} \end{bmatrix}$$

where \( M_{\omega\omega} \) is associated with the elastic degrees of freedom (residual motion), \( M_{\theta\theta} \) represents the coupling between these elastic degrees of freedom and the hub angle \( \theta \) and \( M_{\omega\theta} \) is associated with the inertia of the system about the motor axis. Similarly, the global stiffness matrix can be written as:

$$K^* = \begin{bmatrix} 0 & 0 \\ 0 & K_{\theta\theta} \end{bmatrix}$$

where \( K_{\theta\theta} \) is associated with the elastic degrees of freedom (residual motion). It can be shown that the elastic degrees of freedom do not couple with the hub angle through the stiffness matrix. The global damping matrix \( D^* \) in equation (24) can be represented as:

$$D^* = \begin{bmatrix} 0 & 0 \\ 0 & D_{\theta\theta} \end{bmatrix}$$

where \( D_{\theta\theta} \) denotes the sub-matrix associated with the material damping. The matrix is obtained by assuming that the beam exhibits the characteristics of Rayleigh damping. This proportional damping model has been assumed because it allows experimentally determined damping ratios of
individual modes to be used directly in forming the 
global matrix. It also allows assignment of 
individual damping ratios to individual modes, such 
that the total beam damping is the sum of the 
damping in the modes. Using this assumption, the 
damping can be obtained as
\[
\begin{bmatrix} v_v \end{bmatrix} = \begin{bmatrix} \alpha \end{bmatrix} \begin{bmatrix} M \end{bmatrix} + \begin{bmatrix} \beta \end{bmatrix} \begin{bmatrix} K \end{bmatrix}
\]
(25)
where \( \alpha \) and \( \beta \) are the Rayleigh damping 
coefficients related to the modal damping and 
frequency of the manipulator by:
\[
\begin{bmatrix} \omega_i \end{bmatrix}^2 = \begin{bmatrix} \alpha_i \end{bmatrix} \begin{bmatrix} \zeta_i \end{bmatrix} \begin{bmatrix} \omega_i \end{bmatrix}
\]
(26)
where \( \omega \) and \( \zeta \) are the modal frequency and 
damping ratio and the subscripts ‘1’ and ‘2’ are the lower and upper bounds of the frequency region in interest, respectively.
The \( M^s, D^s, K^s \) matrices are of size \( m_i \times m_i \) and 
\( b^s = [0 \ 0 \ \ldots \ 0]^T \) has \( m_i \times 1 \) size and \( m_i = 2n + 1 \). For simplicity \( d^s = 0 \) and \( Q(0) = 0 \).

Now, the equation of motion is expressed in state-space form, so that it can be solved using control system approaches.
The state-space form of the equation of motion is:
\[
\begin{bmatrix} \dot{v} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}
\]
where
\[
A = \begin{bmatrix} 0_{m_i} & I_{m_i} \\ \vdots & \vdots & \vdots \\ -M^{-1}K & -M^{-1}D \end{bmatrix}
\]
\[
B = \begin{bmatrix} 0_{m_i}^T \\ \vdots \\ M^{-1}e \\ M^{-1}e \end{bmatrix}
\]
\[
C = \begin{bmatrix} 0_{m_i} \\ I_{m_i} \end{bmatrix}
\]
\[
D = \begin{bmatrix} 0_{m_i}^T \\ 0_{m_i} \end{bmatrix}
\]
\( 0_{m_i} \) is an \( m_i \times m_i \) null matrix, \( I_{m_i} \) is an \( m_i \times m_i \) identity matrix, \( 0_{2n+1} \) is an \( m_i \times 1 \) null vector, and the vector \( e \) is the first column of the identity matrix.
\( u = [\theta \ \bar{u} \ \theta \ \bar{u} \ \theta \ \bar{u} \ \theta \ \bar{u}]^T \)
\( v = [\theta \ \bar{u} \ \theta \ \bar{u} \ \theta \ \bar{u} \ \theta \ \bar{u}]^T \)

2.2 Model Structure
A variety of model structures are available to assist in modelling a system. The choice of model structure is based upon an understanding of the system identification method and insight and understanding into the system undergoing identification. The autoregressive moving average model with exogenous inputs (ARMAX) model is one of the most popular linear models [17], [18]. The ARMAX Model structure provides a complete model with disturbance properties modelled separately from system dynamics. If the model is good enough to identify the system without incorporating the noise term or considering the noise as additive at the output, the model can be represented in the ARX form. The ARX model is the simplest model incorporating the stimulus signal. The estimation of the ARX model is the most efficient of the polynomial estimation methods because it is the result of solving linear regression in analytic form. Moreover, the solution is unique. In other words, the solution always satisfies the global minimum of the loss function. The ARX model therefore is preferable [18].

As the simulation data would be collected by the sampling process from the simulation procedure, it is more straightforward to relate the observed data to a discrete time model. The schematic of ARX model is shown in discrete domain as in Fig. 2.

Mathematically the ARX model is given by the following equation:
where
\[ A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n} \]
\[ B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n} \]
are polynomials with associated parameters. \( x(k) \) is the input data, \( y(k) \) is the output data, \( \xi(k) \) is the zero mean white noise. \( z^{-1} \) is a back-shift operator, \( n \) is order of the model and \( a_1 \) until \( a_n \) and \( b_1 \) until \( b_n \) are the model parameters. The main objective of system identification is thus to estimate the model parameters as best as possible.

### 2.3 Parameters estimation

The main task of identification after determining the model structure is to estimate the model parameters. Optimization methodologies are techniques to estimate parameters in a given model structure essentially by finding (by numerical search) those numerical values of the parameters that give the best agreement between the simulated or predicted output of the model and the measured one. Fig. 3 shows the principle of system identification via optimization methodologies. As noted, optimization methodology uses error function to estimate the model parameters.

2.3.1 Differential Evolution Algorithm

Evolutionary algorithms are used effectively for solving difficult optimization problems such as nonlinear global optimization problems subjected to multiple nonlinear constraints [20]. The differential evolution algorithm (DE) proposed by Storn and Price is a population-based stochastic optimization algorithm developed to solve nonlinear and multimodal global optimization problems. DE is an efficient, robust, and effective algorithm. It uses mutation and crossover operations as searching mechanisms and selection operations to direct the search to the most promising regions of the search space. New candidate individuals are made by combining the target individual and several other individuals by randomly choosing from the same population, where the candidate with a better fitness value replaces the target individual [21]. The DE can be outlined as shown in Fig. 4.

The basic operations of differential evolution can be described as:

**Initialization:** All the DE optimization parameters required for optimization process are listed below:
- Problem dimension, \( D \).
- Population size, \( N \).
- Number of generation /stopping condition, \( G \).
- Boundary constraints, \( L-H \).

The upper and lower bounds of each parameter is set based on the parameters range such that
\[ x^i_{j-1} \leq x^i_j \leq x^i_{j+1} \]
where \( j = 1, 2, \ldots, D \) and \( i = 1, 2, \ldots, N \). After that, the target (parent) vector is generated by a random number assigned to each parameter of every vector within the prescribed
range \([X^L_j, X^U_j]\). Equation (28) shows how a target (parent) vector is created.

\[
X_{j,(i)} = \{X_{1,(i)}, X_{2,(i)}, \ldots, X_{(i),0}\} \\
X_{j,(i)} = \text{rand} \cdot (0,1) \cdot (X^U_j - X^L_j) + X^L_j \\
(28)
\]

After that, evaluation of fitness function of each vector is implemented. At the first generation \(X_{j,(i)} = X_{j,(i)}\).

**Mutation**: Mutation can be viewed as an operation that generate Mutant individuals from the target vectors such that the weighted difference of two randomly selected individuals from the target vectors are multiplied by a constant factor and then added with a randomly selected individual from the target vectors. Equation (29) shows how a mutant (donor) vector is created.

\[
V_{j,(i)} = \{V_{1,(i)}, V_{2,(i)}, \ldots, V_{(i),0}\} \\
V_{j,(i)} = V_{j,(i)} + F \cdot (V_{j,(i)} - V_{j,(i-1)}) \\
(29)
\]

The mutation factor (or scale factor), \(F\) is a positive real constant that controls the rate of the differential variation between two individuals and its value lies between 0 and 2. This type of mutation is called \(de/rand/1\). There are other forms of creating the mutant vector frequently used in the literature.

**Crossover**: crossover is introduced to increase the diversity of the perturbed parameters for each individual in the population. The aim of applying the crossover is to achieve trial vectors by replacing a certain individual in the target vectors with corresponding individual in the donor vectors such that an individual is selected randomly from the mutant individuals when \(\text{rand} (D)\) is less than the value of the crossover probability. Equation (30) shows how a trail vector is created.

\[
U_{j,(i)} = \{U_{1,(i)}, U_{2,(i)}, \ldots, U_{(i),0}\} \\
U_{j,(i)} = \begin{cases} V_{j,(i)} & \text{if } \text{rand} (D) \leq CR \\ X_{j,(i)} & \text{otherwise} \end{cases} \\
(30)
\]

The crossover ratio controls the fraction of parameter values that are to be copied from the mutant vector. If the value of a first random number
is less than the chosen CR then the corresponding element of mutant vector is passed on to the target vector otherwise it is copied from the trial vector. This process is repeated for all elements and for the entire population.

**Selection:** If the objective function value of the individual in the trial vector \( U_{j(G_{t+1})} \) has an equal or lower corresponding individual in the target vectors \( X_{j(G_t)} \), it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one generation. Equation (31) shows how a mutant (donor) vector is created.

\[
U_{j(G_{t+1})} = \begin{cases} U_{j(G_{t+1})} & \text{if } f(U_{j(G_{t+1})}) \leq f(X_{j(G_t)}) \\ X_{j(G_t)} & \text{otherwise} \end{cases} \quad (31)
\]

### 2.3.2 Particle Swarm Optimization

Particle swarm optimization is a population based, stochastic optimization technique introduced by Kennedy and Eberhart [22]. The operating procedure of a PSO can be described through the stages shown in Fig. 5.

![Flowchart of PSO algorithm](image)

**Fig. 5:** Flowchart of PSO algorithm: (a) PSO procedure, (b) Initial setup procedure of PSO, (c) Initial population procedure of PSO, (d) Update population procedure of PSO
PSO is initialized with a group of random particles, ‘fly’ in the search space of an optimization problem. Particles are updated with two ‘best’ values every iteration. The first one is called pbest, which is the best position a particle has visited so far and is memorized. Another ‘best’ value is the global best or gbest, obtained so far by any particle in the population. Using their memories of the best positions of pbest and gbest, particle is then accelerated toward those two best values by updating the particle position and velocity using the following set of equations:

\[
v_{id}(t) = w_{id}(t-1) + c_1 \text{rand} \cdot (p_{id}(t) - x_{id}(t-1)) + c_2 \text{rand} \cdot (p_{gd}(t) - x_{id}(t-1))
\]

\[
x_{id}(t) = x_{id}(t-1) + v_{id}(t)
\]

where \(v_{id}(t)\) and \(x_{id}(t)\) are the current velocity and position vector of the \(i\)-th particle in the \(d\)-dimensional search space respectively. \(C_1\) and \(C_2\) are acceleration coefficients usually \(C_1 = C_2 = 2\) and \text{rand} is a random number between 0 and 1. \(w\) is the inertia which serves as memory of the previous direction, preventing the particle from drastically changing direction. High values of \(w\) promote global exploration and exploitation while low values of \(w\) lead to a local search. The common approach is to provide balance between global and local search by linearly decrease \(w\) during the search process. Decreases the inertia over time can be expressed as:

\[
w(t) = w_{start} - \frac{w_{start} - w_{end}}{T_{max}} \\
\]

where \(w_{start}\) and \(w_{end}\) are the starting and end point of inertia weight set as 0.9 to 0.25 respectively and \(T_{max}\) is the maximum number of time step the swarm is allowed to search.

### 2.4 Model Validation

Model validity tests are procedures designed to detect the adequacy of a fitted model. Once a model of the system is obtained, it is required to validate the model whether it is good enough to represent the system or not. Many model validation tests are available in the literature. In this study, mean squared error, one step-ahead prediction and correlation tests were used to validate the model. Furthermore, an unseen data used to observe the prediction ability of the model and pole-zero plots test was used to check the stability of the obtained models.

#### 2.4.1 One Step-Ahead Prediction (OSA)

A common measure of predictive accuracy used in control and system identification is to compute the OSA prediction of the system output. This is expressed as:

\[
y(k) = -a_1y(k-n_1) \ldots -a_2y(k-l) + b_1y(k-n_1) \ldots + b_2y(k-1)
\]

Often \(y(k)\) will be a relatively good prediction of \(y(k)\) over the estimation set, even if the model is biased, because the model was estimated by minimizing the prediction errors.

#### 2.4.2 Mean Squared Error

Mean squared error is one of the most common variables used in the validations. The mean-squared error is difference between actual output of the system and the predicted output produced from the input to the system and the optimized parameters.

#### 2.4.3 Correlation Tests

Correlation Tests is a more convincing method to be used in model validation. Correlation test is a statistical test that shows the degrees of the relationship between two variables. There are two types of correlation test:

1. Auto correlation test is representing as a vector.
2. Cross correlation test is representing as matrix.

In the theory of linear systems, the usual statistical approach to validate identified linear and nonlinear models consists of computing the autocorrelation test examines the correlation among the residuals themselves while the cross-correlation test surveys the correlation between the residuals and past inputs. It has been shown that acceptable predictions over different data sets are produced only if the model is unbiased. If the model structure and the estimated parameters are correct then the prediction error sequence \(e(t)\) should be uncorrelated with all linear and nonlinear combinations of past inputs and outputs (unbiased) and this will hold if and only if the following conditions are satisfied:

\[
\phi_{e(e)}(\tau) = E[e(t-\tau) e(t)] = \delta(\tau)
\]

\[
\phi_{u(e)}(\tau) = E[u(t-\tau) e(t)] = 0, \quad \forall \tau
\]

\[
\phi_{u^2(e)}(\tau) = E[u^2(t-\tau) e^2(t)] = 0, \quad \forall \tau
\]

\[
\phi_{e(e)(u)}(\tau) = E[e(t) e(t-1-\tau) u(t-1-\tau)] = 0 \quad \tau \geq 0
\]
In practice, the correlation will never be exactly zero for all lags but the model is considered as adequate if the correlation tests lay within 95% confidence limits, defined as $1.96/\sqrt{N}$, where N is the data length. Autocorrelation of the residual will also never be an ideal delta function but will be considered as adequate if the autocorrelation plot enters the 95% confidence limits before lag one.

Ideally the model validity test should detect all the deficiencies in algorithm performance including bias due to internal noise. Consequently the full five tests defined by equation should be satisfied. For linear system only first two conditions must be satisfied.

### 3. Implementation and Results

To study the dynamic behavior of the flexible manipulator system, a computer program was written within Matlab environment to simulate the state space matrices derived from the mathematical modeling (referred to section 2.1). A thin aluminium alloy with the specifications shown in Table 1 is considered [3].

For simplicity purposes, the effect of mass payload is neglected. Throughout this simulation, a bang-bang torque input with an amplitude ±0.3 Nm was applied at the hub of the manipulator as shown in Fig. 6. The response of the flexible manipulator at the hub angle and end-point residual is monitored for duration of 3.0 seconds with sampling time 0.37 ms and is observed and recorded as shown in Figs. 7 (a), (b) and 8 (a), (b) in both time and frequency domain respectively. Previous researches demonstrated the validity of FEM in modeling flexible single link manipulator system [1]. Therefore, the same FEM is adopted in this study.

The flexible manipulator is a single-input multiple-output (SIMO) system. The input, is the torque of the motor and the outputs are the hub-angle and end-point residual. In this study, two single-input single-output (SISO) models are developed. The first model represents system behavior from input torque to hub-angle output and the second model represents system behavior from input torque to the end-point residual output.

### Table 1

<table>
<thead>
<tr>
<th>Components</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length (l)</td>
<td>960 mm</td>
</tr>
<tr>
<td>Width (w)</td>
<td>19.008 mm</td>
</tr>
<tr>
<td>Thickness (h)</td>
<td>3.2004 mm</td>
</tr>
<tr>
<td>Mass density (ρ)</td>
<td>2710 kg m$^{-3}$</td>
</tr>
<tr>
<td>second moment of inertia (I)</td>
<td>$5.19 \times 10^{-11}$ m$^4$</td>
</tr>
<tr>
<td>Young’s modulus (E)</td>
<td>$71 \times 10^9$ N m$^{-2}$</td>
</tr>
<tr>
<td>Hub inertia (I$_h$)</td>
<td>$5.86 \times 10^{-4}$ kg m$^2$</td>
</tr>
</tbody>
</table>
3.1 Modeling of Hub Angle Using DE

A Matlab program has been created based on DE algorithm and used to estimate the parameters of ARX model using torque input and hub angle output data obtained from the simulation study. Since there was no a priori knowledge regarding the suitable order of the model, the structure realization was performed by a trial-and-error method. Randomly selected parameters were optimized for different, arbitrarily chosen order to fit into the system by applying the working mechanism of DE based on One Step-Ahead prediction. Table 2 shows the work that carried out to select the best model order.

It can be noted from Table 2 that the best result was achieved with model order = 4 for 4000 data length. From the work carried out for the model order = 4, the satisfactory results were achieved with the following set of parameters:

- Population Initial Range: [-3; 3]
- Number of Generations: 2800
- Population Size: 10

The data set, comprising 7986 data points was divided into two sets of 4000 and 3986 data points respectively. The first set was used to estimate the model parameters while the second set was used to validate the model. Both output and estimated output of the hub angle in time and frequency domains are plotted in Figs. 9 and 10 respectively. The error between actual and predicted output are plotted in Fig. 11. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Figs. 9 and 11. The best mean square error of DE algorithm is $2.504 \times 10^{-4}$. Using the proposed identification procedure, the parameters of the model were estimated as follows:

$$a_1 = -1.998052$$
$$a_2 = 0.998075$$
$$b_1 = 0.000156$$
$$b_2 = -0.000172$$

Figs. 9, 10 and 10 show that the model was able to mimic the measured output well. The pole-zero diagram and correlation tests are depicted in Figs. 12 and 13 respectively. It is noticed from Fig. 12 that the model was stable and the poles of the transfer function were inside the unit circle and the zero was outside the unit circle indicating non-minimum phase behavior. The correlation functions were carried out for 1000 samples to determine the effectiveness of the DE-based model. The results were found to be within 95% confidence level thus confirming the accuracy of the results.

<table>
<thead>
<tr>
<th>Model Order</th>
<th>MSE</th>
<th>Correlation Test</th>
<th>Time</th>
<th>No. of Generation</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>four</td>
<td>$2.5 \times 10^{-4}$</td>
<td>unbiased</td>
<td>25.05</td>
<td>2800</td>
<td>stable</td>
</tr>
<tr>
<td>six</td>
<td>$3.75 \times 10^{-4}$</td>
<td>unbiased</td>
<td>57.8</td>
<td>6000</td>
<td>stable</td>
</tr>
<tr>
<td>eight</td>
<td>$5.02 \times 10^{-4}$</td>
<td>unbiased</td>
<td>133.4</td>
<td>12800</td>
<td>stable</td>
</tr>
<tr>
<td>ten</td>
<td>$6.32 \times 10^{-4}$</td>
<td>unbiased</td>
<td>262.4</td>
<td>23000</td>
<td>unstable</td>
</tr>
<tr>
<td>twelve</td>
<td>$7.61 \times 10^{-4}$</td>
<td>unbiased</td>
<td>323</td>
<td>36000</td>
<td>unstable</td>
</tr>
<tr>
<td>fourteen</td>
<td>$8.83 \times 10^{-4}$</td>
<td>unbiased</td>
<td>424.8</td>
<td>42000</td>
<td>unstable</td>
</tr>
<tr>
<td>Sixteen</td>
<td>0.0010</td>
<td>unbiased</td>
<td>447.5</td>
<td>48000</td>
<td>unstable</td>
</tr>
</tbody>
</table>
3.2 Modeling End-point residual Using DE

A Matlab program has been created based on the DE algorithm, and used to estimate the parameters of ARX model based on the torque input and end-point residual output data obtained from the simulation study. Since there was no a priori knowledge about the suitable model order of the flexible manipulator system, the structure realization was performed using a heuristic method. Randomly selected parameters were optimized for different, arbitrarily chosen order to fit into the system by applying the working mechanism of DE based on OSA prediction. Table 3 shows the work carried out to select the best model order.

It can be noted from Table 3 the best result was achieved with model order = 6, for 4000 data length. For the best model order, DE was designed with 10 individuals with maximum number of iterations set to 6000. The data set, comprising 7986 data points divided into two sets of 4000 and 3986 data points respectively. The first set was used to compute the model parameters whilst the second set was used to validate the model. In addition, the best mean square error of DE algorithm is $1.587 \times 10^{-6}$. Using the proposed identification procedure, the parameters of the model were estimated as follows:

$$a_1 = -2.990312$$
Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs. 14 and 15 respectively and the error between actual and predicted DE output are plotted in Fig. 16. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Figs. 14 and 16. The pole-zero diagrams and correlation tests are depicted in Figs. 17 and 18 respectively. Fig. 17 indicated that the model is stable. The correlation test functions results confirm that the model is acceptable.

### Table 3
Comparative assessment for end-point residual best order

<table>
<thead>
<tr>
<th>Order</th>
<th>MSE</th>
<th>Correlation test</th>
<th>Time</th>
<th>No. of generation</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>four</td>
<td>$1.8 \times 10^{-6}$</td>
<td>biased</td>
<td>15.5</td>
<td>4000</td>
<td>stable</td>
</tr>
<tr>
<td>six</td>
<td>$2.3 \times 10^{-6}$</td>
<td>unbiased</td>
<td>55.9</td>
<td>6000</td>
<td>stable</td>
</tr>
<tr>
<td>eight</td>
<td>$2.1 \times 10^{-6}$</td>
<td>unbiased</td>
<td>105.5</td>
<td>10400</td>
<td>stable</td>
</tr>
<tr>
<td>ten</td>
<td>$6.4 \times 10^{-6}$</td>
<td>biased</td>
<td>330</td>
<td>32000</td>
<td>unstable</td>
</tr>
<tr>
<td>twelve</td>
<td>$4.5 \times 10^{-6}$</td>
<td>biased</td>
<td>353.0</td>
<td>36000</td>
<td>unstable</td>
</tr>
<tr>
<td>fourteen</td>
<td>$7.2 \times 10^{-6}$</td>
<td>biased</td>
<td>495.5</td>
<td>42000</td>
<td>unstable</td>
</tr>
<tr>
<td>Sixteen</td>
<td>$6.9 \times 10^{-6}$</td>
<td>biased</td>
<td>523.3</td>
<td>48000</td>
<td>unstable</td>
</tr>
</tbody>
</table>

Fig. 14: Actual and estimated output of end-point residual in time domain using DE

Fig. 15: Actual and estimated output of end-point residual in frequency domain using DE

Fig. 16: Error between actual and estimated output for end-point residual using DE

Fig. 17: Pole-zero diagram of end-point residual using DE
behavior. The correlation test functions results confirm that the model is acceptable.

![Fig.18: Correlation tests of end-point residual using DE](image1)

![Fig.19: Actual and estimated output of the hub angle in time domain using PSO](image2)

![Fig.20: Actual and estimated output of the hub angle in frequency domain using PSO](image3)

![Fig.21: Error between actual and estimated output for hub angle using PSO](image4)

![Fig.22: Pole-zero diagram for hub angle using PSO](image5)

### 3.3 Modeling Hub Angle Using PSO

Investigations were then carried out with PSO algorithm to model the hub angle using the same input-output data and same model order achieved with DE. PSO was designed with 50 individuals with maximum number of iterations set to 30000. The best mean square error of PSO is $2.519 \times 10^{-4}$. Using the proposed identification procedure, the parameters of the model were estimated as follows:

$$
\begin{align*}
    a_1 & = -1.643623 \\
    a_2 & = 0.643645 \\
    b_1 & = 0.003884 \\
    b_2 & = -0.003815
\end{align*}
$$

Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs. 19 and 20 respectively and the error between actual and predicted PSO output are plotted in Fig. 21. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Figs. 19 and 21. The pole-zero diagrams and correlation tests are depicted in Figs. 22 and 23 respectively. Fig. 22 indicating that the model is stable with non-minimum phase
3.4 Modeling End-point residual Using PSO

Investigations were then carried out with the PSO algorithm to model the end-point residual using the same input-output data and same model order achieved for end-point residual used in DE. PSO was designed with 150 individuals with maximum number of iterations set to 30000. The best mean square error of PSO algorithm is $1.014 \times 10^{-4}$.

Using the proposed identification procedure, the parameters of the model were estimated as follows:

\[
\begin{align*}
\alpha_1 & = -0.448655 \\
\alpha_2 & = -2.375043 \\
\alpha_3 & = 1.828526 \\
b_1 & = -0.036951 \\
b_2 & = 0.341808 \\
b_3 & = -0.305859
\end{align*}
\]

Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs. 24 and 25 respectively and the error between actual and predicted PSO output are plotted in Fig. 26. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Figs. 24 and 26. The pole-zero diagrams and correlation tests are depicted in Figs. 27 and 28 respectively. Fig. 27 indicated that the model is unstable. The correlation test functions results showed that the model is biased.
5 Comparative Assessment

The overall comparative performance of DE and PSO modelling approaches in terms of mean squared error, stability and correlation tests is summarised in Table 4.

From Table 4, it can be seen that the stability and the correlation test results by DE are better than PSO especially for the end-point residual modelling. However, a major advantage of the PSO is that the algorithm is simple. Table 4 also shows that the performance of DE in terms of mean-squared error is better than the PSO with the same model structure. Therefore, high performance computing power could provide better solutions in the real time implementation of the DE based identification.

6 Conclusion

In this work, DE and PSO have been adopted for modelling a single-link flexible manipulator system. Input-output data pairs have been collected from a simulation study and used in developing linear models of the system from input torque to hub-angle and end-point acceleration. The performances of DE-ARX and PSO-ARX models have been assessed through many validation tests. It has been demonstrated that the DE and PSO modelling technique has performed well in approximating the system response and DE is better than PSO in modelling a single-link flexible manipulator system.

This simulation platform forms the basis to implement different control structures to control a single-link flexible manipulator system. Moreover, this study can be extended to use system identification techniques for modeling a single-link flexible manipulator system using a real plant to implement different control structures before online control can be implemented.

Acknowledgment

The authors would like to express their gratitude to Minister of Education Malaysia (MOE) and Universiti Teknologi Malaysia (UTM) for funding and providing facilities to conduct this research. This research is supported using FRGS Vote No. 4F395 and UTM Research University grant, Vote No. 05H71.

Table 4

<table>
<thead>
<tr>
<th>Modelling domain</th>
<th>MSE</th>
<th>Stability</th>
<th>Correlation test</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-point residual</td>
<td>DE: $1.58 \times 10^{-6}$</td>
<td>stable</td>
<td>unbiased</td>
</tr>
<tr>
<td></td>
<td>PSO: $1.01 \times 10^{-4}$</td>
<td>unstable</td>
<td>biased</td>
</tr>
<tr>
<td>Hub angle</td>
<td>DE: $2.50 \times 10^{-4}$</td>
<td>stable</td>
<td>unbiased</td>
</tr>
<tr>
<td></td>
<td>PSO: $2.52 \times 10^{-4}$</td>
<td>stable</td>
<td>unbiased</td>
</tr>
</tbody>
</table>

References


[3] H. Supriyono, Novel Bacterial Foraging Optimisation Algorithms with Application to...


[22] M. Settles, An Introduction to Particle Swarm Optimization. Department of Computer Science, University of Idaho, 2005