Cuckoo Search for Modelling of a Flexible Single-Link Manipulator

ALI ABDULHUSSAIN MADHLOOM AL-KHAFAJI 1 AND INTAN Z. MAT DARUS 2
1Department of Applied Mechanics and Design, Fakulti Kejuruteraan Mekanikal Universiti Teknologi Malaysia 81310, Johor Bahru MALAYSIA
ali1977abdalhusain@gmail.com; intan@fkm.utm.my

Abstract—This study presents the development of approaches with Cuckoo Search algorithm and Particle Swarm Optimization technique for modelling of a flexible single-link manipulator. Finite element method and Lagrangian approaches were used to acquire the input-output data of the system. A bang-bang input torque was applied and the dynamic response of the system was investigated. Next, a suitable model structure was selected and optimized using an intelligent parametric identification technique of Cuckoo Search algorithm and Particle Swarm Optimization techniques. Mean squared error, One Step Ahead prediction, correlation tests have been performed for validation and verification of the obtained model in characterizing the manipulator system. A comparative assessment of the two models in characterizing the manipulator system is presented in time and frequency domains. Results demonstrate the advantages of Cuckoo Search algorithm over Particle Swarm Optimization in parametric modeling of the flexible single-link manipulator.

Key-Words: - Dynamic modelling, Cuckoo Search algorithm, Particle Swarm Optimization, Flexible single-link Manipulator, finite element method

1 Introduction
Because of the extraordinary request for automation and safety, robotic manipulators are currently used in many applications to accomplishment difficult jobs at dangerous places. Maximum stiffness rigid robotic manipulators leads to large robot weight to payload ratio, increased energy consumption, increased size of actuator, low speed and increased overall cost. To cover these problems, the design of flexible links manipulators with lightweight has been motivated. The flexible link manipulator system, comparing with rigid link manipulator system offers Significant advantages for their rigid counterparts, include low inertia, light weight, few powerful actuators, cheaper construction, fast in response, safer operation, higher payload carrying capacity, more compact design and longer reach. Moreover, they can quickly be adapted to changing situations and product design variations. For such advantages, modern flexible link manipulator systems are widely used in many areas such the space vehicles, medical, defense and automation industries. The flexible manipulators vibration, due to structural flexibility is a major ongoing and an unwanted feature. Thus, to achievement the benefits of flexible manipulators, accurate models and efficient control methods have to be established to control the vibration problems [1],[2],[3].

To develop an effective control mechanism for a dynamic system such as flexible link manipulator system, it is required to model and predict the behaviors of the system based on given input-output data. An appropriate system model is a significant first step in control system design and analysis [2]. In present decade, the potential candidate for many control application to obtain an appropriate model of the system are System Identification techniques [4],[5],[6]. many parametric and nonparametric estimation techniques have been employed as optimization tools in identification of flexible link manipulator system 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],[13], bacterial foraging algorithms (BFA) [14]. It can be noted from the literature that the use of CS for modelling flexible manipulators has not been reported yet.

2 The Flexible Manipulator System
For the mathematical modeling of a flexible single-link manipulator, a finite element method and Lagrangian approach is utilized [15],[16],[17]. The development of the algorithm can be divided into three main steps:

1. FEM analysis
2. State space representation
3. Obtaining the result

The overall approach involves treating the link of the manipulator as an assemblage of \( n \) elements of length
For each of these elements the kinetic energy \( T_i \) and potential energy \( V_i \) are computed in terms of a suitably selected system of \( n \) generalized variables \( q \) and their rate of change \( \dot{q} \). In this study, a flexible single-link manipulator system is considered as 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 \( I_H \), where the input torque \( \tau(t) \) is applied at the hub by a motor and payload mass \( M_p \) is attached at the free end. \( E, I \) and \( \rho \) represent the Young Modulus, second moment of inertia and mass density per unit length of the flexible manipulator respectively. \( XY \) axis and \( X_0Y_0 \) axis represent the stationary and moving coordinate respectively. Both axes lie in a horizontal plane and all rotation occurs about a vertical axis. For a small angular displacement \( \theta(t) \) and small elastic deflection, \( w(x,t) \) the overall displacement \( y(x,t) \) of a point along the link at a distance \( X \) from the hub can be defined as a function of both the rigid rotation angle \( \theta(t) \) of the hub and flexible displacement \( w(x,t) \) of the beam measured from the line \( o \) thus the total displacement \( y(x,t) \) is:

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

using the standard finite element method to solve dynamic problems, leads to the well-known equation:

\[
w_n(x,t) = N_n(x)Q_n(t)
\]

Where \( N_n(x) \) and \( Q_n(x) \) represent the shape function and nodal displacement respectively.

Define, \( s = x - \sum_{i=1}^{N}l_i \), where \( l_i \) is the length of the \( i \)th element. the kinetic energy of an element can be expressed as follows:

\[
T_i = \frac{1}{2}[Q_n]_i'[M_n][Q_n]_i, \quad i = 1,2,\ldots,N\]

and the potential energy due to the elasticity of the FE can be obtained as:

\[
P_i = \frac{1}{2}\sum_{i=1}^{N}[Q_n]_i'[K_n][Q_n]_i
\]

The total kinetic and potential energy can be written as:

\[
T = T_n + \sum_{i=1}^{N}T_i + P_i = \frac{1}{2}\dot{Q}_n'\dot{M}_n\dot{Q}_n
\]

Similarly, the potential energy due to the elasticity of the FE can be obtained as:

\[
P = \sum_{i=1}^{N}P_i = \frac{1}{2}\dot{Q}_n'\dot{K}_n\dot{Q}_n
\]

**Dynamic equation of the single link manipulator**

The Lagrangian of the link can be derived as followed:

\[
L = T - P = \frac{1}{2}\dot{Q}_n'\dot{M}_n\dot{Q}_n - \frac{1}{2}\dot{Q}_n'\dot{K}_n\dot{Q}_n
\]

A further detail on dynamic modeling can be found in[15],[16],[17]. By using Lagrange equation, the dynamic equations of a flexible manipulator can be derived utilizing the equation as followed:

\[
M\ddot{Q} + D'\dot{Q}' + K'Q' = b'\tau
\]

where \( M', K', Q' \) are mass matrix, rigidity matrix and general coordinates when substituting boundary conditions, respectively. \( b' = [1 \ 0 \ \cdots \ 0]^T \) \( \tau \) is input torque. \( D' \) is damping matrix and we choose the linear type Rayleigh damping and \( Q(t) \) is the nodal displacement given as:

\[
Q(t) = [\theta_1 w_1 \theta_2 \cdots w_n \theta_n]
\]

The \( M', D', K' \) matrices are of size \( m_1 \times m_1 \) and \( b' = [1 \ 0 \ \cdots \ 0]^T \) \( \tau \) has \( m_1 \times 1 \) size and \( m_1 = 2n + 1 \). For simplicity \( D' = 0 \) and \( Q(0) = 0 \)

The state-space form of the equation of motion is:

\[
\dot{v} = Av + Bu \quad (3) \quad \theta = Cv + Du \quad (4)
\]

Where

\[
A = \begin{bmatrix}
0_{m_1} & \cdots & I_{m_1} \\
\cdots & \cdots & \cdots \\
-M^{-1}K & \cdots & -M^{-1}D
\end{bmatrix}, \quad B = \begin{bmatrix}
0_{m_1m^1} \\
\cdots \\
M^{-1}e
\end{bmatrix}, \quad C = \begin{bmatrix}
0_{m_1} & \cdots & I_{m_1}
\end{bmatrix}, \quad D = \begin{bmatrix}
0_{m^1m^1}
\end{bmatrix}
\]

\( 0_{m_1} \) is an \( m_1 \times m_1 \) null matrix, \( I_{m_1} \) is an \( m_1 \times m_1 \) identity matrix, \( 0_{m^1m^1} \) is an \( m_1 \times 1 \) null vector, and the vector \( e \) is the first column of the identity matrix.

\[ u = [\tau \ \ 0 \ \ \cdots \ \ 0]^T \]

\[ v = [\theta \ u_1 \ \ \cdots \ u_n \ \dot{\theta}_1 \ \cdots \ \dot{\theta}_n]^T \]

3 Cuckoo Search Algorithm
Cuckoo search algorithm is a search algorithm developed by Xin-she Yang and Suash Deb 2009. The algorithm inspired by the breeding behavior of cuckoos. Cuckoo birds lay their eggs in other birds’ nests and rely on those birds for hosting the egg. Some of the other host birds discover that an egg is not their own, it might throw out the alien egg or just move to new locations elsewhere. A cuckoo might emulate the shape, color and size of the host eggs to protect their egg from being discovered. To increase the hatching probability of cuckoo birds own eggs, some of them might throw out other native eggs from the host nest. On the other hand, a hatched cuckoo chick will also throw other eggs away from nest to improve its feeding share [18], [19].

The following three idealized rules used to describe the Cuckoo Search in simple way:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2. The best nests with high quality of eggs (solutions) will carry over to the next generations;
3. The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a$ [0, 1]. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location. The procedures for application of Cuckoo Search is outlined as:

1. **Start**
2. **Initial Setup**
   - Number of dimensions (d)
   - Upper and lower bounds $[ub, lb]$
   - Number of nests (n)
   - Maximum Iteration
   - Define hypothesis function $h(\theta)$
   - Define the cost function $J(\theta)$
   - Parameters of Levy flights $\alpha$
   - Discovery rate of alien eggs ($p_a$)

3. **Initial Population**
   - Create random (n) nests of dimension (d) without violating $[ub, lb]$
   - Set initial minimum cost to a very high value
4. **Update Population**
   - Loop from 1: (Max Generation) or (stop criterion)
   - Create proposed nests using current nests values and Levy flights with $\beta$, with constraints that the best nest is not altered all solutions within bounds
   - Evaluate cost of the proposed nests using $J(\theta)$
   - Perform elitism
   - Create new nests by mutating current nests randomly based on $p_a$
   - Evaluate fitness of the proposed nests using $J(\theta)$
   - Perform elitism

5. **Display Output Results**
6. **End**

4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based, stochastic optimization technique introduced by Kennedy and Eberhart [20]. The operating procedure of a PSO can be described through the stages shown as:

1. **Start**
2. **Initial Setup**
   - Number of dimensions (d)
   - Upper and lower bounds $[ub, lb]$
   - Number of particles (n)
   - Maximum Iteration
   - Define hypothesis function $h(\theta)$
   - Define the cost function $J(\theta)$

3. **Initial Population**
   - Create position vectors of random (n) particles of dimension (d) without violating $[ub, lb]$

4. **Update Population**
   - Loop from 1: (Max Generation) or (stop criterion)
   - Evaluate cost of each particle using $J(\theta)$
   - Update the pbest and gbest positions and values
   - Calculates the new velocity and positions for all swarm’s elements

5. **Display Output Results**
6. **End**

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 memorizes it. 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 pbestand gbest, particle is then accelerated toward those two best values by updating the particle position and velocity using the following set of equations:

$$v_{id}(t) = \nu v_{id}(t-1) + c_1 r_1 (p_{id} - x_{id}(t-1)) + c_2 r_2 (p_{ged} - x_{id}(t-1))$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t)$$

$v_{id}(t)$ 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 the random number between 0 and 1. \( \text{rand} \) is the inertia serves as memory of the previous direction, preventing the particle from drastically changing direction. High value of promote global exploration and exploitation, while low value of \( w \) leads to a local search. The common approach is to provide balance between global and local search by linearly decrease during the search process. Decreases the inertia over time can be expressed as:

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

where \( w_{\text{start}} \) and \( w_{\text{end}} \) are the starting and end point of inertia weight defined as linearly decrease from 0.9 to 0.25 and \( T_{\text{max}} \) is the maximum number of time step the swarm is allowed to search. Start \( w_{\text{end}} \) end \( w_{\text{max}} T\)

5 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 done above. A thin aluminum alloy with the specifications shown in Table 1 is considered [3].

<table>
<thead>
<tr>
<th>Components</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length (l)</strong></td>
<td>960 mm</td>
</tr>
<tr>
<td><strong>Width (w)</strong></td>
<td>19.008 mm</td>
</tr>
<tr>
<td><strong>Thickness (h)</strong></td>
<td>3.2004 mm</td>
</tr>
<tr>
<td><strong>Mass density per unit volume ((\rho))</strong></td>
<td>2710 kg m(^{-3})</td>
</tr>
<tr>
<td><strong>The second moment of inertia ((I))</strong></td>
<td>5.1924 \times 10(^{-11}) m(^4)</td>
</tr>
<tr>
<td><strong>The young modulus ((E))</strong></td>
<td>71 \times 10^9 N m(^{-2})</td>
</tr>
<tr>
<td><strong>The hub inertia ((I_h))</strong></td>
<td>5.86 \times 10^-4 kg m(^2)</td>
</tr>
</tbody>
</table>

For simplicity purposes, the effect of mass payload is neglected. Throughout this simulation, a bang-bang torque input with an amplitude \( \pm 0.3 \) Nm was applied at the hub of the manipulator as shown in Fig 2. 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 Fig 3(a),(b) and Fig 4(a),(b) in both time and frequency domain respectively. The first three resonant mode captured for end-point residual and hub angle response is at 11.59 Hz, 31.88 Hz and 58.93 Hz is compared with the experimental first three modes of vibration of the manipulator which obtained at at 11.67 Hz, 36.96 Hz and 64.22 Hz respectively (14). These, as noted, closely match with the corresponding experimental value with small percentage error that 0.6% for first mode, 13.7% for second mode and 8.2% for third mode. The flexible single-link manipulator system considered in this study is a single-input multiple-output (SIMO) system, with one input, the torque of the motor, and two outputs, namely hub-angle, end-point residual. Two single-input single-output (SISO) models are developed representing the system behavior from input torque to hub-angle output and end point residual output.
5.1 Modeling of Hub Angle Using CS

A MATLAB program has been created based on the CS algorithm, and was used to estimate the parameters of ARX model based on the torque input and hub angle output data obtained from the simulation study of the flexible manipulator system, as described in the earlier sections of this 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 CS based on One Step-Ahead prediction. The best result was achieved with model order = 4, that mean, $n_u = 2$, $n_y = 2$ for 4000 data length, which was used to find the parameters. From the work carried out, it was found that satisfactory results were achieved with the following set of parameters:

- Population Initial Range: $[-2; 2]$
- Number of Generations: 3000
- 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 whilst the second set was used to validate the model. Both output and estimated outputs of the hub angle in time and frequency domains are plotted in Figs.5 and 6 respectively. The error between actual and predicted CS output are plotted in Fig 7. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Fig 5 and Fig 7. The best mean square error of CS algorithm is $2.5 \times 10^{-4}$. Using the proposed identification procedure, the parameters of the model were estimated as follows:

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$-1.854273472263909$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_2$</td>
<td>$0.855152038545899$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>$0.006416925485563$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$-0.006312863003462$</td>
</tr>
</tbody>
</table>

The output and estimated outputs of the hub angle in time and frequency domains and error show that the model was able to mimic the measured output well. The pole-zero diagram and correlation tests are depicted in Figs 8 and 9 respectively. It is noticed that the model was stable, 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 CS-based model. The results were found to be within 95% confidence level thus confirmed the accuracy of the results.

Fig. 8. Pole-zero diagram of hub angle using C

Fig. 9. Correlation tests for hub angle using CS

5.2 Modeling End-point residual Using CS

A MATLAB program has been created based on the CS algorithm, and was used to estimate the parameters of ARX model based on the torque input and end-point residual output data obtained from the simulation study of the flexible manipulator system, as described in the earlier sections of this 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 CS based on One Step-Ahead prediction. The best result was achieved with model order = 6, that mean, \( n_u = 3 \), \( n_y = 3 \) for 4000 data length, which was used to find the parameters. For the best model order, the CS was designed with 10 individuals in each iteration with maximum number of iterations was set to 9000. 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 compute the model parameters whilst the second set was used to validate the model. In addition to that, the best mean square error of CS algorithm is \( 1.5 \times 10^{-6} \). Using the proposed identification procedure, the parameters of the model were estimated as follows:

\[
\begin{align*}
a_1 & \approx -2.987703017371025 \\
a_2 & \approx 2.97966125781491 \\
a_3 & \approx -0.991958223745479 \\
b_1 & \approx -0.00056698498788 \\
b_2 & \approx 0.000016630048664 \\
b_3 & \approx 0.000547426954055
\end{align*}
\]

Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs 10 and 11 respectively and the error between actual and predicted CS output are plotted in Fig 12. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Fig 10 and 12. The pole-zero diagrams and correlation tests are depicted in Fig. 13 and Fig. 14 respectively, indicating that all models were stable with non-minimum phase behavior. The correlation test functions results confirm that the models are acceptable.

Fig. 10. Output and estimated outputs of end-point residual in time domain using CS

Fig. 11. Output and estimated outputs of end-point residual in frequency domain using CS
Investigations were then carried out with the PSO algorithm to modelling the hub angle using the same input-output data and same model order achieved for hub angle using CS. The PSO was designed with 50 individuals in each iteration with maximum number of iterations was set to 30000. The best mean square error of PSO algorithm is 6.43645235839840. Using the proposed identification procedure, the parameters of the model were estimated as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>0.643645235839840</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0.003884513014574</td>
</tr>
<tr>
<td>$b_2$</td>
<td>-0.003815956600461</td>
</tr>
</tbody>
</table>

Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs.15 and 16 respectively and the error between actual and predicted CS output are plotted in Fig 17. The division between the trained data and the unseen data is indicated as a vertical line located at point 4000 as shown in Fig 15 and Fig 17. The pole-zero diagrams and correlation tests are depicted in Fig. 18 and Fig. 19 respectively, indicating that all models were stable with non-minimum phase behavior. The correlation test functions results confirm that the models are acceptable.
5.4 Modeling End-point residual Using PSO

Investigations were then carried out with the PSO algorithm to modelling the end-point residual using the same input-output data and same model order achieved for end-point residual that used in CS. The PSO was designed with 150 individuals in each iteration with maximum number of iterations was 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:

<table>
<thead>
<tr>
<th>$a_i$</th>
<th>$b_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>-0.448655649457146</td>
</tr>
<tr>
<td>$a_2$</td>
<td>-2.375043245413949</td>
</tr>
<tr>
<td>$a_3$</td>
<td>1.828526069760020</td>
</tr>
<tr>
<td>$b_1$</td>
<td>-0.63599378695153660</td>
</tr>
<tr>
<td>$b_2$</td>
<td>0.341808251288217</td>
</tr>
<tr>
<td>$b_3$</td>
<td>-0.303859485671289</td>
</tr>
</tbody>
</table>

Both output and estimated output of end-point residual in time and frequency domains are plotted in Figs. 20 and 21 respectively and the error between actual and predicted CS output are plotted in Fig 22. The division between the trained data and the unseen data is indicated as a vertical line located at point 60000 as shown in Figs. 20 and 22. The pole-zero diagrams and correlation tests are depicted in Figs. 23 and 24 respectively, indicating that the model is unstable. The correlation test functions results showed that the model is biased.
6 Comparative Assessments
The overall comparative performance of CS and PSO modelling approaches in terms of the mean-squared error, stability and correlation tests is summarised in Table 2.

From Table 2, it can be seen that the performance of CS modelling technique gives better approximation to the system response compared to PSO modelling technique. It is also noted that the stability and the correlation test results by CS is better than PSO especially for the end point residual modelling. However, a major advantage of the PSO is that the algorithm is simple. Table 2 also shows that the performances of CS in terms of mean-squared error are 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 CS based identification.

Table 2: Overall comparative assessment

<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>CS</td>
<td>1.8 × 10⁻⁶</td>
<td>stable</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>1.61 × 10⁻⁵</td>
<td>unstable</td>
</tr>
<tr>
<td>Hub angle</td>
<td>CS</td>
<td>2.50 × 10⁻⁴</td>
<td>stable</td>
</tr>
<tr>
<td></td>
<td>PSO</td>
<td>2.52 × 10⁻⁴</td>
<td>stable</td>
</tr>
</tbody>
</table>

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

Acknowledgement
The authors wish to thank the Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for providing the research grant and facilities. This research is supported using UTM Research University grant, Vote No. 04H17.

References:


