

Self-tuning Active Vibration Controller using Particle Swarm Optimization for Flexible Manipulator System

HANIM MOHD YATIM and INTAN Z. MAT DARUS

Department of Applied Mechanics

Universiti Teknologi Malaysia

81310 Skudai, Johor Bahru

MALAYSIA

hanim.yatim@gmail.com; intan@fkm.utm.my

Abstract: - This paper presents the development of an optimal self-tuning PID controller for vibration suppression of flexible manipulator structures using particle swarm optimization (PSO). A simulation environment characterizing the dynamic behavior of the flexible manipulator system was first developed using finite difference (FD) approach to acquire the input-output data of the system. The global search technique of PSO is used to estimate the model transfer function through parametric identification in comparison to the conventional recursive least square (RLS) method. Next, the control structure comprises conventional PID controller and an intelligent PID incorporated PSO controller for position and vibration control respectively acted on flexible manipulator model system. Behavior of system response including hub angle and end-point displacement are recorded and assessed. The validation of the algorithm is presented in time and frequency domains. It was demonstrated that proposed controller is effective to move the flexible link to the desired position with suppression of vibration at end-point of a flexible manipulator structure.

Key-Words: - Active vibration control; single-link flexible manipulator; particle swarm optimization, PID controller

1 Introduction

Unwanted vibration reduction of flexible manipulator has received substantial attention in these recent years. The increased utilization of flexible manipulator has been motivated by the requirement of industrial applications since its offer several advantages over rigid manipulator including light weight, lower energy consumption, safer operation due to reduced inertia, smaller actuator requirement, less bulky design, more transportable and maneuverable [1,2,3]. Due to flexibility, manipulator systems will demonstrate vibration when subjected to disturbance forces [4]. The manipulator will vibrate during maneuver and become more severe when maneuver becomes faster [5]. This unwanted vibration will bring difficulty in accurate positioning of flexible manipulator and may reduce the system effectiveness. This highly non-linear dynamics of the system make control of flexible manipulators become complicated. If advantages of flexible manipulator are not to be sacrificed, thus an accurate model and efficient controls system need to be studied and developed. It is important to control this non-linear dynamics of the system as the flexible manipulator need to have

reasonable end-point accuracy in response of input commands.

Generally, the purpose of vibration control in flexible manipulator is to suppress the unwanted vibration with satisfactory end-point tracking. A number of approaches have been proposed and developed for control of flexible manipulator. However, PID control is still one of the most widely used in industries even with the introduction of many control theories and approaches. PID control is popular due to low cost, simple in control structure, as well as easy to maintain and implement. This demonstrated by Mahamood and Pedro [6] which implemented PD-PID controller for control of two link flexible manipulator. Ahmad *et al.* [7] investigated a hybrid controller incorporated collocated PD controller and a feedforward controller based on input shaping for vibration suppression and end-point trajectory tracking of flexible robot manipulator. The effectiveness of the proposed controller for corresponding system was shown.

In PID control, finding optimum PID gains is very important in order to give a satisfactory system response. Improper PID gains optimization may lead to poor robustness and slow recovery [8].

Several investigations have applied intelligent techniques for optimization of PID parameters such in [8] where Genetic Algorithm (GA) and Differential Evolution (DE) were applied to optimize PID controller for high order system, system with time delay and non-minimum phase system. The performance of the PID control systems was also compared with Ziegler-Nichols method. Results indicate the effectiveness of GA and DE in tuning PID parameters compared to conventional method.

Md Zain *et al.* [9] proposed GA for tuning PID controller for flexible manipulator in vertical motion. The advantages of the proposed control schemes were shown. However, this paper focused on utilization of global search method of PSO for active PID control of single-link flexible manipulator constrained in horizontal motion.

PSO is a global, stochastic optimization technique inspired by social behavior of bird flocking or fish schooling [10,11]. PSO implementing an information sharing approach within the search space in order to find an optimum solution. Advantages of PSO are it has only a few parameters to adjust with exclusion of crossover and mutation operator which possesses the properties of easy implementation and fast convergence. PSO has been found to be robust and widely applied in continuous and discrete optimization problems and promisingly suitable for both scientific and engineering applications [12,13,14].

Applications of PSO in control system can be seen in many research papers. Jaafar *et al.* [15] utilized PSO-tuned PID controller for nonlinear gantry crane system. Their results indicated an effective controller to move the trolley as fast as possible to the desired position with low payload oscillation. Selvan *et al.* [16] presented a new insight in PSO algorithm for tuning PID parameters. PID controller was tested on several different plant transfer functions conformed to the theoretical predictions. The result was compared with the conventional Ziegler-Nichols method and it was found that the latter method provides better results and faster convergence.

In this paper, global search of PSO is utilized to optimize the PID controller gains. Prior to this, a simulation environment characterize dynamic behavior of the flexible manipulator structure is developed using FD method. Then, PSO based identification is employed to obtain the transfer function of the system responses in comparison to conventional RLS method.

Subsequently a control structure with two PID controllers is proposed for position tracking and end-point vibration reduction respectively. The proposed controller is implemented in Matlab simulation environment and its performance is assessed.

2 Simulation of Flexible Manipulator Structure

Fig. 1 shows a single-link flexible manipulator that can be 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 an input control 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 ρ represent the Young Modulus, second moment of inertia and mass density per unit length of the flexible manipulator respectively.

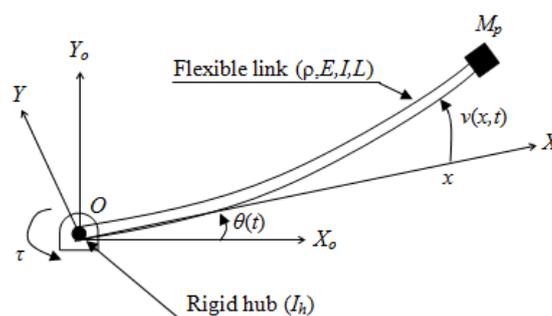


Fig. 1: Schematic diagram of the flexible manipulator system

Let $X_O Y_O$ and XOY be the stationary and moving coordinates respectively. For an angular displacement $\theta(t)$ and an elastic deflection $v(x,t)$, the total (net) displacement $w(x,t)$ of a point along the manipulator at a distance x from the hub can be described as a function of both rigid body motion and elastic deflection measured along the line OX .

(1)

$$w(x,t) = x\theta(t) + v(x,t)$$

The well-known governing equation of a flexible manipulator can be formulated as [17,18]

$$EI \frac{\partial^4 w(x,t)}{\partial x^4} + \rho \frac{\partial^2 w(x,t)}{\partial t^2} = \tau(t) \quad (2)$$

The corresponding boundary and initials conditions at the pinned and free ends of the manipulator are obtained as

$$\begin{aligned} w(0,t) &= 0 \\ I_h \frac{\partial^3 w(0,t)}{\partial x \partial t^2} - EI \frac{\partial^2 w(0,t)}{\partial x^2} &= \tau(t) \\ M_p \frac{\partial^2 w(L,t)}{\partial t^2} - EI \frac{\partial^3 w(L,t)}{\partial x^3} &= 0 \\ EI \frac{\partial^2 w(L,t)}{\partial x^2} = 0, w(x,0) = 0, \frac{\partial w(x,0)}{\partial x} &= 0 \end{aligned} \quad (3)$$

The fourth-order PDE in (2) represents the dynamic equation describing the motion of the flexible manipulator. Note that the model in (2) does not incorporate damping.

A number of approaches have been proposed and developed for dynamic modeling of flexible manipulator. Finite difference (FD) method has been widely used to solve differential equations. It has been reported that the method is simple in mathematical terms, easy to program and more convenient for uniform manipulator structure [2].

A simulation algorithm characterizing the dynamic behavior of the flexible manipulator structure is thus developed using FD method to obtain the numerical solution of the PDE in (2). In the case of a flexible manipulator, a two dimensional coordinate system is considered. This involves dividing the manipulator into n sections each of length Δx and deflection of each section at sample times Δt . Thus, a solution of the PDE is obtained using the central difference approximations for the partial derivative terms of the response $w(x,t)$ of the manipulator at points $x = i\Delta x$ and $t = j\Delta t$.

A solution of the PDE in (2) can be obtained as,

$$\begin{aligned} w_{i,j+1} &= -p[w_{i+2,j} + w_{i-2,j}] + q[w_{i+1,j} + w_{i-1,j}] \\ &+ r w_{i,j} - w_{i,j-1} + \frac{\Delta t^2}{\rho} \tau(i,j) \end{aligned} \quad (4)$$

where,

$$r = 2 - \frac{6EI\Delta t^2}{\rho\Delta x^4}; \quad q = \frac{4EI\Delta t^2}{\rho\Delta x^4}; \quad p = \frac{EI\Delta t^2}{\rho\Delta x^4}$$

Eqn. (4) gives the displacement $w_{i,j+1}$ of grid points $i=1,2,\dots,n$ of the manipulator at time step $j+1$. Note that in evaluating the displacement at $w_{n-1,j+1}$ and $w_{n,j+1}$, the fictitious displacement at $w_{n+2,j}$, $w_{n+1,j}$ and $w_{n+1,j-1}$ are required. These can be obtained by utilized the boundary conditions in (3)

and thus the discrete form of the corresponding boundary conditions are

$$\begin{aligned} w_{0,j} &= 0 \\ w_{-1,j} &= w_{1,j} + \frac{\Delta x I_h}{EI\Delta t^2} [w_{1,j+1} - 2w_{1,j} + w_{1,j-1}] \\ &+ \frac{\Delta x^2}{EI} \tau(j) \\ w_{n+2,j} &= 2w_{n+1,j} - 2w_{n-1,j} + w_{n-2,j} \\ &+ \frac{2M_p \Delta x^3}{EI\Delta t^2} [w_{n,j+1} - 2w_{n,j} + w_{n,j-1}] \\ w_{n+1,j} &= 2w_{n,j} - w_{n-1,j} \end{aligned} \quad (5)$$

The above equation (4) can be written in a matrix form as:

$$\mathbf{W}_{i,j+1} = \mathbf{A}\mathbf{W}_{i,j} + \mathbf{B}\mathbf{W}_{i,j-1} + \mathbf{C}\mathbf{F} \quad (6)$$

where $\mathbf{W}_{i,j+1}$ is the displacement of grid points $i = 1,2,\dots,n$ of the manipulator at time step $j+1$. $\mathbf{W}_{i,j}$ and $\mathbf{W}_{i,j-1}$ are the corresponding displacements at time steps j and $j-1$ respectively. \mathbf{A} and \mathbf{B} are constant matrix $n \times n$ matrices where depends on flexible manipulator specifications and the number of sections the manipulator is divided into, \mathbf{C} is a constant matrix related to input torque and \mathbf{F} is an $n \times 1$ matrix related to time step Δt and mass per unit length of the flexible manipulator.

To study the dynamic behavior of the flexible manipulator system, a thin aluminium alloy of flexible manipulator with dimensions $0.96 \times 0.01923 \times 0.0032004$ m³, mass density per area, ρ , 2710 kg/m², Young's Modulus, E , 7.11×10^{10} N/m² and second moment of inertia, I , 5.1924×10^{-11} m² is considered. The response of the flexible manipulator is monitored for duration of 3.0 seconds with sampling time 0.2 ms and the system behaviour is observed and recorded at the hub and end point of the manipulator.

The first three modes of vibration of the manipulator calculated based on equation (7)[19] are obtained at 12.60 Hz, 40.83 Hz and 85.19 Hz where first mode being dominant mode.

$$f_i = \frac{\lambda_i^2}{2\pi L^2} \left(\frac{EI}{m} \right)^{1/2} \quad i = 1, 2, 3 \quad (7)$$

Validation of the developed finite difference model was carried out by comparing the simulation result with the calculated results at resonant modes as given in Table 1. The algorithm was implemented

on the basis of varying number of sections along the link manipulator.

Table 1: Modes of vibration with varying number of FD sections

No. of Sections	Mode 1 (Hz)		Mode 2 (Hz)		Mode 3 (Hz)	
	12.60	% Error	40.83	% Error	85.19	% Error
5	6.65	47.22	25.00	38.77	45.00	47.18
10	10.00	20.63	31.64	22.51	64.95	23.76
15	11.65	7.54	35.00	14.28	71.60	15.95
20	11.65	7.54	36.63	10.29	74.95	12.02

It is noted that the frequency parameter corresponding to the first mode of vibration converges to a reasonably stable value with the algorithm using at least 15 sections. It was also appeared that the error converges to minimum as the number of sections increases. It can be concluded that 20 sections will be reasonable to use for the simulation algorithm characterizing the dynamic behaviour of the flexible manipulator system to have a reasonable accuracy.

Using the validated FD method, the manipulator is divided into 20 equal length sections. For simplicity purposes, the effects of hub inertia and mass payload are neglected. Throughout this simulation, a bang-bang torque input with an amplitude ± 0.3 Nm and duration of 0.5 seconds is applied at the hub of the manipulator as shown in Fig. 2. This type of input is chosen so that the manipulator will accelerate, decelerate and stop accordingly at the desired positions.

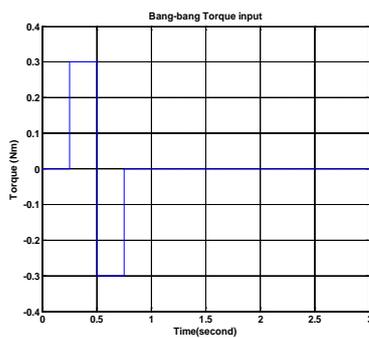


Fig. 2: Input torque bang-bang

Figs. 3 and 4 show the dynamic response of hub angle and end-point displacement of the flexible manipulator respectively in time and frequency domain. The first resonant mode captured by both responses is at 11.65 Hz.

3 Recursive Least Square

Recursive least squares (RLS) is a well-known method uses an iterative refinement technique to continuously tune estimated parameters using knowledge of some existing parameters as well as previous history (e.g., inputs, outputs) [20].

The RLS algorithm is based on mathematical weighted LS and creates continuous estimation for a set of unknown system parameters. The RLS algorithm is described by the following set of equations [20]:

$$\hat{\beta}(i) = \hat{\beta}(i-1) + K(i)E(i) \tag{8}$$

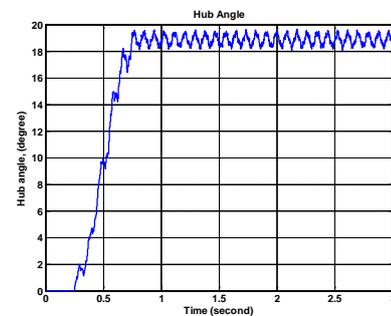
$$K(i) = \frac{\lambda^{-1}P(i-1)x(i)}{1 + \lambda^{-1}x(i)^T P(i-1)x(i)} \tag{9}$$

$$P(i) = \lambda^{-1}P(i-1) - \lambda^{-1}K(i)x(i)^T P(i-1) \tag{10}$$

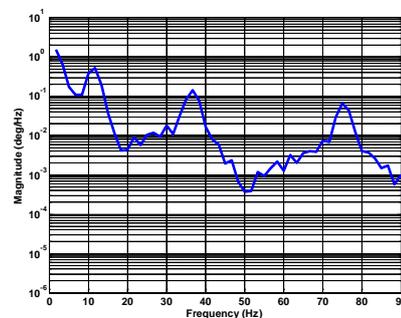
where the error is,

$$E(i) = y(i) - x(i)^T \hat{\beta}(i-1) \tag{11}$$

$\hat{\beta}(i)$ is the unknown parameter vector, $x(i)$ is the regression vector and $y(i)$ is the system output vector. λ is defined as the forgetting factor or weighting factor which reduces the influence of old data usually taking the form, $0 < \lambda < 1$.



(a) Time domain



(b) Spectral density

Fig. 3: Hub angle

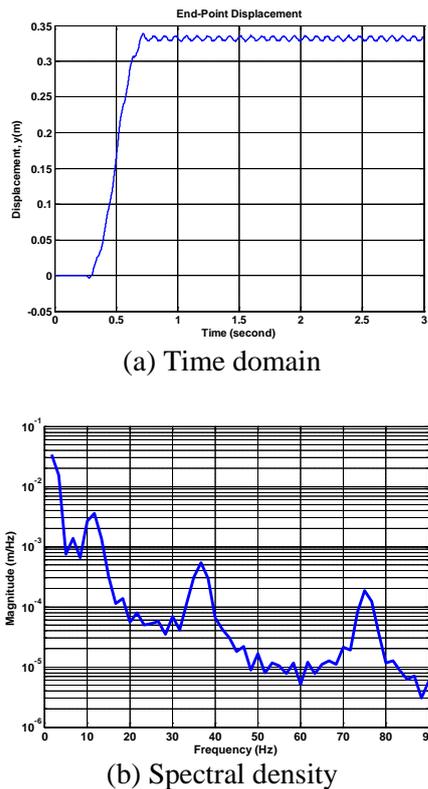


Fig. 4: End-point displacement

4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population based, global, stochastic optimization technique introduced by Kennedy and Eberhart [10,11]. The search process of PSO can be described through the stages shown in Fig.5.

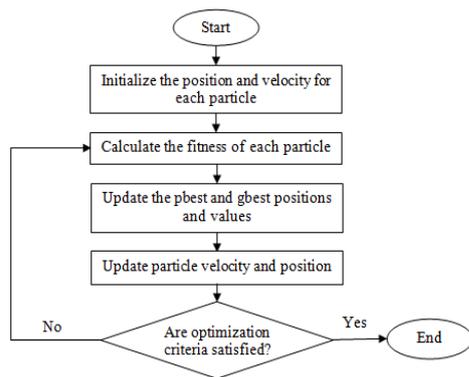


Fig. 5: Flowchart of PSO algorithm

PSO is initialized with a group of random particles, move in the search space of an optimization problem. Each particle is associated with a position and a velocity. The particles evaluate their positions based on a global fitness value and have a memory to remember their best position in the search space.

At each iteration, particles are updated with two ‘best’ values. The first one is called *pbest* (P_{id}), which is the best position a particle has visited so far. Another ‘best’ value is the global best or *gbest* (P_{gd}), obtained so far by any particle in the population. After finding the *pbest* and *gbest*, particle is then accelerated toward those two best values by updating the particle position and velocity for the next iteration using the following set of equations [21]:

$$v_{id}(t) = kv_{id}(t-1) + C_1 \cdot rand \cdot (P_{id} - x_{id}(t-1)) + C_2 \cdot rand \cdot (P_{gd} - x_{id}(t-1)) \quad (12)$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t) \quad (13)$$

$v_{id}(t)$ and $x_{id}(t)$ are the current velocity and position vector of the i -th particle in the d -dimensional search space respectively. The cognitive part C_1 encourage the particles to move towards their own best position so far and the social component of C_2 represents the collaborative effect of the particles in finding the global optimum solution [11]. Usually $C_1 = C_2 = 2$ and *rand* is the random number between 0 and 1.

k is the inertia serves as memory of the previous direction, preventing the particle from drastically changing direction. High value of k promote global exploration and exploitation, while low value of k leads to a local search. The common approach is to provide balance between global and local search by linearly decrease k during the search process. Decreases the inertia over time can be expressed as

$$k(t) = k_{start} - \frac{(k_{start} - k_{end})}{T_{max}} \cdot t \quad (14)$$

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

5 Parametric Identification

Relationship between input and output of the system is required to develop a suitable model of the system [20]. For this purpose, it is essential to determine an appropriate order and parameters for the model that best fits that relation. The most basic relationship between the input and output of a system is given by ARX model described as

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})}u(t) + \frac{\xi(t)}{A(z^{-1})} \quad (15)$$

where $A(z^{-1})$ and $B(z^{-1})$ expressed as

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_n z^{-n} \quad (16)$$

$$B(z^{-1}) = b_o + b_1 z^{-1} + \dots + b_n z^{-(n-1)} \quad (17)$$

z^{-1} is defined by a backshift operator, white noise, $\xi(t) = 0$, n is the orders of the model and $[a_1, \dots, a_n, b_1, \dots, b_n]$ are estimated model parameters. In this study, $u(t)$ and $y(t)$ are the system input and output vector respectively which substituted by the torque bang-bang (Fig.2) as input and the system response of hub angle and end-point displacement (Figs. 3 and 4) as measured output of the system. Therefore, the corresponding transfer function of the system model $H(z^{-1})$ can be represented as follows

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_o + b_1 z^{-1} + \dots + b_n z^{-(n-1)}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (18)$$

Thus, the main objective of system identification is to optimize the model parameters as best as possible. The performance of optimization formula is evaluated by setting their objective function as mean square error (MSE) defined as

$$MSE = \frac{1}{S} \sum_{t=1}^S (e(t))^2 \quad (19)$$

where, S represents the number of samples and $e(t)$ is the predicted error between the desired output and estimated model output.

6 Control Schemes

In this work, the proposed control structure for vibration control of the flexible manipulator is introduced. Initially, a collocated PID control is developed for rigid body motion. Subsequently an intelligent PID tuning by PSO controller is implemented for flexible body motion. These two loops are combined together to give control inputs to the flexible manipulator system. A block diagram of the control scheme is shown in Fig. 6.

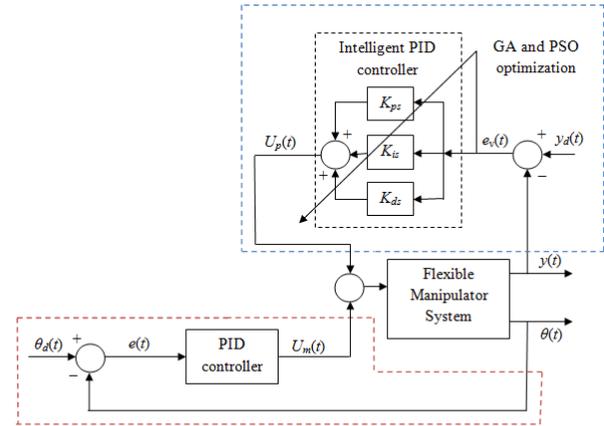


Fig. 6: Block diagram of proposed controller

6.1 Collocated PID control

For rigid body motion control, $\theta_d(t)$, $\theta(t)$ and $e(t)$ represents reference hub angle, hub angle and error of the system respectively as shown in Fig. 7. Essentially, PID control system is adopted to position the flexible link to the desired angle of demand. Here the motor gain, A_c and sensor gain, G are set as a linear gains. C and H are the transfer function of controller and flexible manipulator respectively. The hub angle signal is fed back and used for control purpose.

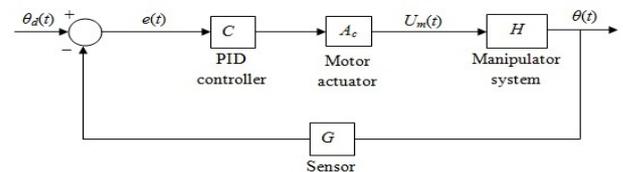


Fig. 7: PID control structure

For the closed loop signal in Fig. 7, the control signal $U_m(s)$ can be written as

$$U_m(s) = A_c \left[\left(K_p + \frac{K_i}{s} + K_d s \right) e(s) \right] \quad (20)$$

where, s is the Laplace variable. Error function of the system $e(s)$ defined as

$$e(s) = \theta_d(s) - G\theta(s) \quad (21)$$

The closed loop transfer function is therefore obtained as

$$\frac{\theta(s)}{\theta_d(s)} = \frac{\left(K_p + \frac{K_i}{s} + K_d s \right) A_c H(s)}{1 + \left(K_p + \frac{K_i}{s} + K_d s \right) A_c G H(s)} \quad (22)$$

where $H(s)$ is the open loop transfer function from the input torque to the hub angle obtained through the parametric identification process. In this study, PID parameter is determined from the auto-tuning function in Matlab software.

6.2 Intelligent PID-PSO control

An intelligent PID control is utilized to suppress an unwanted vibration at the end-point of flexible link. For flexible body motion control, K_{ps} , K_{is} and K_{ds} are proportional, integral and derivative gains respectively. From Fig. 8, $y(t)$ and $e_v(t)$ represents end-point displacement and error of the system respectively. $y_d(t)$ represents reference end-point displacement which is set to zero as the control objective to have zero vibration.

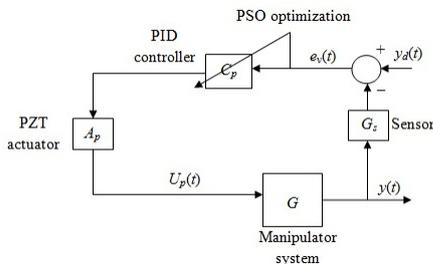


Fig. 8: Intelligent PID-PSO control structure

The end-point displacement signal is fed back and thus controls input $U_p(s)$ can be defined as

$$U_p(s) = A_p \left[\left(K_{ps} + \frac{K_{is}}{s} + K_{ds}s \right) e_v(s) \right] \quad (23)$$

Since the reference end-point displacement was set to zero signal, the error function $e_v(t)$ is defined as

$$e_v(s) = |0 - G_s y(s)| \quad (24)$$

The closed loop transfer function is therefore obtained as

$$\frac{y(s)}{y_d(s)} = \frac{\left(K_{ps} + \frac{K_{is}}{s} + K_{ds}s \right) A_p G(s)}{1 + \left(K_{ps} + \frac{K_{is}}{s} + K_{ds}s \right) A_p G_s G(s)} \quad (25)$$

where $G(s)$ is the open loop transfer function from the input torque to the end-point displacement obtained through the parametric identification process. To design a particular control loop, the K_{ps} , K_{is} and K_{ds} controller parameters were tuned using a global optimization method of PSO so that control input $U_p(s)$ provide acceptable performance of

flexible manipulator system. In PID-PSO tuning procedure, PID parameters are optimized by assigning three swarms in search space where each swarms representing each parameter of K_{ps} , K_{is} and K_{ds} . The minimization of MSE is performed by PSO to tuning PID gains and set as objective function for optimization formula.

7 Implementation and results

Hub angle and end-point displacement response of flexible manipulator structure was modeled with the parametric identification approaches of RLS and PSO algorithm using an ARX model structure. The obtained system model is then used in the control structure of flexible manipulator includes collocated PID control for rigid body motion and intelligent PID-PSO control for flexible motion control.

7.1 RLS modeling

The flexible manipulator was modeled with RLS algorithm using measured input-output data from dynamic modeling of the system (Section 2).

The model was observed with different orders and the best result was achieved with an order 3. Forgetting factor for end-point displacement response is determined as 0.9 while for hub angle response is assigned to 0.1. The simulated output of hub angle and end-point displacement in both time and frequency domains are shown in Figs. 9 and 10. It was noted that correlation tests for RLS model were within 95% confidence interval. The best results of RLS algorithm achieved are presented in Table 2.

7.2 PSO modeling

Investigations were then carried out for modelling with PSO algorithm using the input-output data obtained from dynamic simulation of flexible manipulator system. The model achieved the best result with an order of 2. The PSO was designed with 50 individuals in each iteration with maximum number of iterations was set to 100. The best MSE results of PSO algorithm achieved with their transfer function obtained are depicted in Table 2.

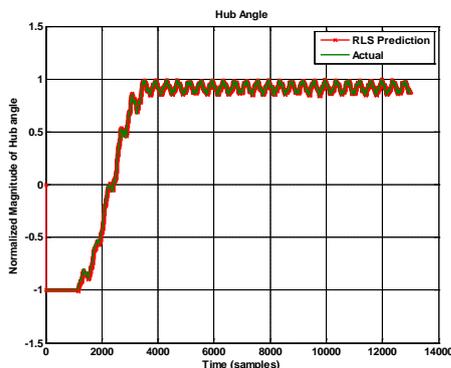
The corresponding correlation tests for identification using PSO were found to be within 95% confidence interval indicating adequate model fit. Figs. 11 and 12 show the simulated output of hub angle and end-point displacement in time and frequency domains.

Table 2: Parametric Identification

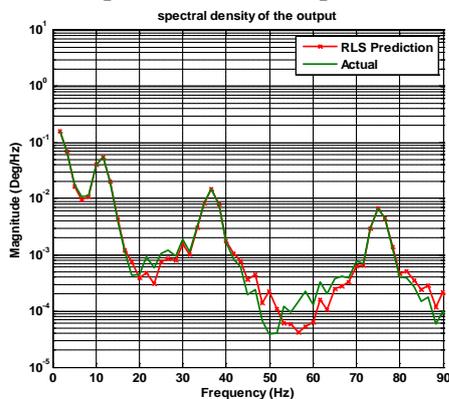
	Modeling domain	MSE	Transfer function
Hub angle	RLS	1.9614×10^{-4}	$\frac{0.06324s^2 - 0.6357s + 77}{s^3 + 13.86s^2 + 1035s + 1.835}$
	PSO	1.5725×10^{-4}	$\frac{0.02205s + 0.00097}{s^2 - 4.978s + 0.001696}$
End-point displacement	RLS	1.5015×10^{-4}	$\frac{0.01168s^2 + 0.06364s + 12.84}{s^3 + 13.86s^2 + 1035s - 0.3067}$
	PSO	4.6084×10^{-5}	$\frac{-3.714s - 0.04183}{s^2 + 3.522s - 0.009484}$

From Table 2, corresponding MSE results reveals that the identification using PSO has performed far better than that with RLS for both hub angle and end-point displacement. It is demonstrated that PSO are successfully find an optimum global solution for identification where the conventional method fail. Due to its effectiveness, PSO algorithm is adopted in subsequent investigations as an optimization tools for vibration control of flexible manipulator.

From Figs. 9, 10, 11 and 12, it is noted that the simulated output using RLS and PSO match the actual output very well for the time domains mapping of both response. This is further demonstrated by the frequency domain plots which show that the model has successfully characterized the system dynamics especially the first three modes of vibration. The first vibration mode captured by both RLS and PSO is recorded at 11.65 Hz, which is very close to the theoretical vibration mode with percentage of error 7.46%

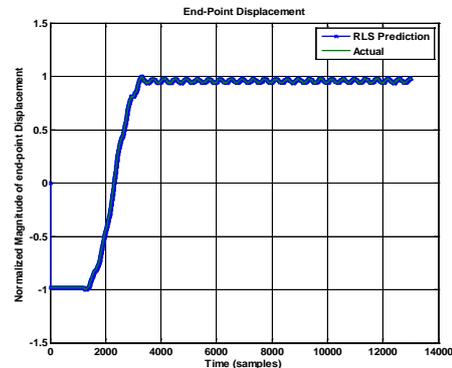


(a) Actual and predicted RLS output in time domain

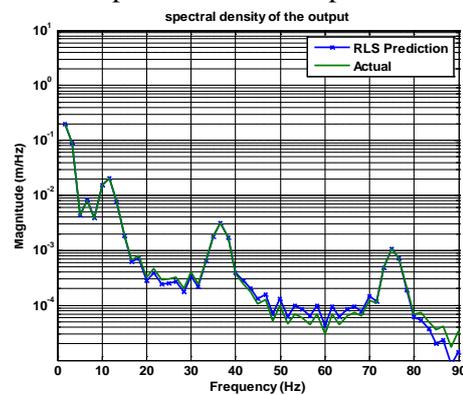


(b) Spectral density of output

Fig. 9: Hub angle modeling using RLS

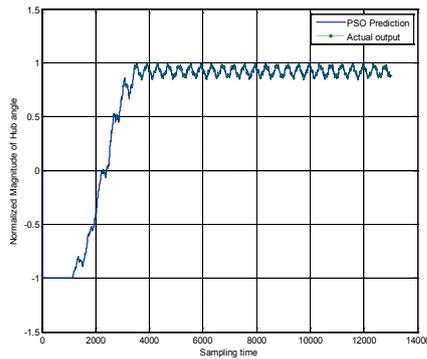


(a) Actual and predicted RLS output in time domain

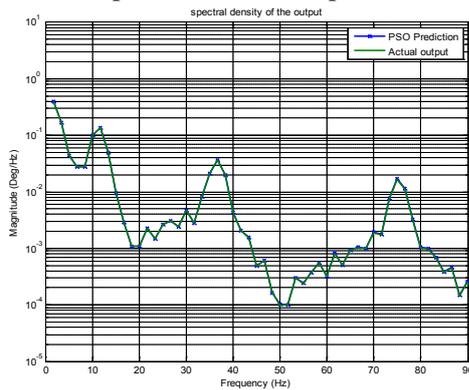


(b) Spectral density of output

Fig. 10: End-point displacement modeling using RLS

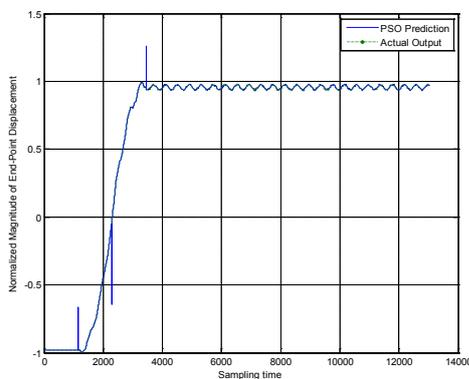


(a) Actual and predicted PSO output in time domain

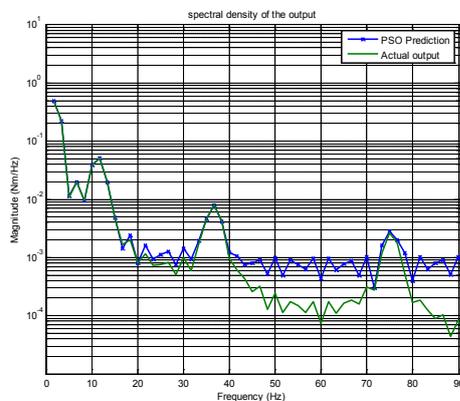


(b) Spectral density of output

Fig. 11: Hub angle modeling using PSO



(a) Actual and predicted PSO output in time domain



(b) Spectral density of output

Fig. 12: End-point displacement modeling using PSO

7.3 Control Results

Proposed control schemes are implemented and tested within Matlab environment of the flexible manipulator. Hub angle and end-point displacement responses of the system is measured and analyzed to study the performance of the proposed controllers. The performances are assessed in terms of input tracking and vibration suppression achieved with the controllers.

7.3.1 Rigid motion control

Rigid motion control of flexible manipulator has been controlled by collocated PID controller (Section 6.1). The flexible link of the manipulator is required to follow a step input of 19°. The PID parameters are deduced using auto-tuning function in Matlab software and the transfer function is provided from the open loop response of input torque to hub angle obtained using RLS and PSO algorithm. The controller parameters obtained as shown in Table 3.

Table 3: Performances of input tracking for all models

Modeling domain	PID Parameter	Tracking capabilities		
		SSE	OS (%)	T_s (s)
RLS	$K_{pr}= 80.03$ $K_{ir}= 9.96 \times 10^{-2}$ $K_{dr}= 20.59$	0	0	1.039
PSO	$K_p= 1.96 \times 10^5$ $K_i= 5000.41$ $K_d= 100.39$	0	0	0.264

The corresponding hub angle responses of the manipulator using the PID control are shown in Fig. 13. It is clear that the response from PID control is almost indistinguishable and an acceptable hub angle response was achieved. The manipulator reached the desired angle with zero steady state error (SSE) and zero overshoot (OS) for both models but PSO model reached desired angle faster with settling time, T_s of 0.264 s.

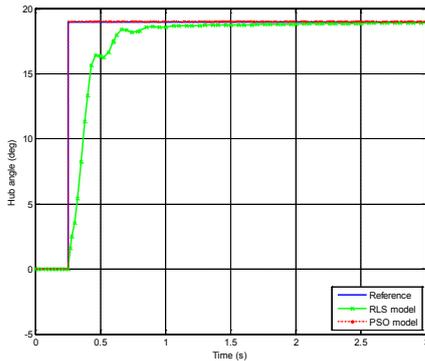


Fig. 13: Input tracking by PID controller

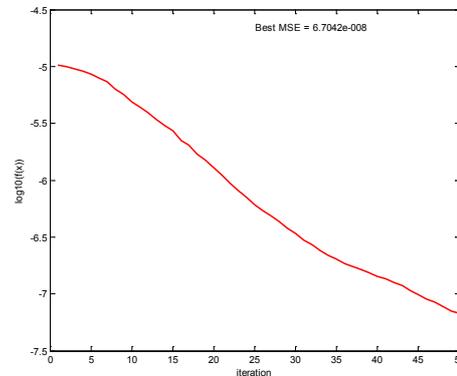
7.3.2 Flexible motion control

Intelligent PID controller was employed to actively suppress the unwanted vibration at the end-point of flexible manipulator. In this study, 30 particles are considered with 50 iterations for PSO algorithm to tune the PID controller. For purposes of comparison, the flexible body motion also control by PID controller where the parameters being tuned by autotuning function in Matlab software. Table 4 shows the optimal PID parameters achieved for RLS and PSO models.

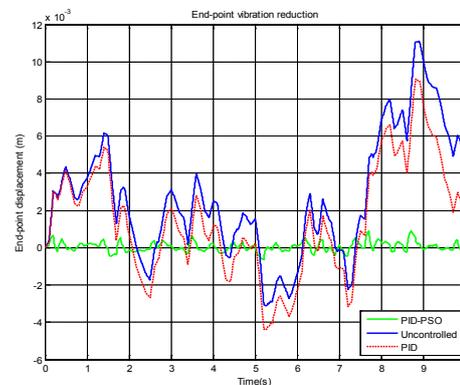
Figs. 14 and 15 shows the simulated output of vibration suppression of PID controller in time and frequency domains for RLS and PSO model respectively. The RLS algorithm achieved the best MSE level of 6.7042×10^{-8} while PSO algorithm achieved the best MSE level of 3.9334×10^{-8} in the 50th iteration as shown in Figs. 14(a) and 15(a). Figs. 14(b)-(c) and 15(b)-(c) show the control system performance in this case. It is noted from time domain mapping, PID controller actively suppress the unwanted vibration at end-point. This is further demonstrated by the frequency domain plots which show the significant amount of spectral attenuation at the resonant mode.

The spectral attenuation achieved by PID and PID-PSO controller for both models has been depicted in Table 4. It is reveals that PSO model has a better identify model where the attenuation achieved is bigger than RLS model. It is also demonstrated that the developed PSO for tuning PID controller has performed far better in the suppression of vibration of the flexible manipulator structure as compared with the conventional PID controller.

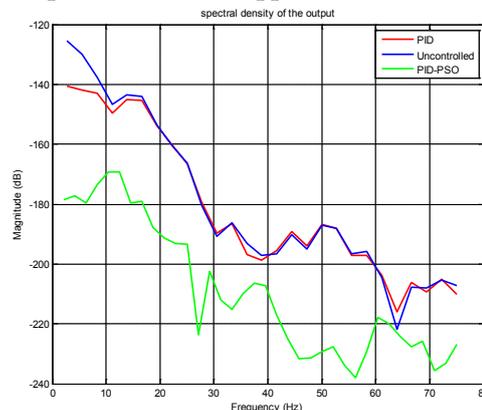
Therefore optimizations of PID parameters are crucial in order to improve overall performance of the system and PSO algorithm is effective as an optimization tools.



(a) MSE as a function of number of iteration for PID-PSO



(b) End-point vibration suppression in time domain

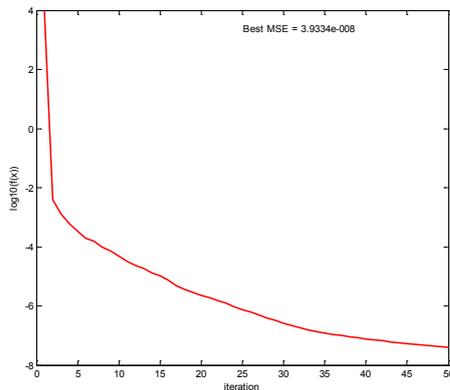


(c) Spectral density

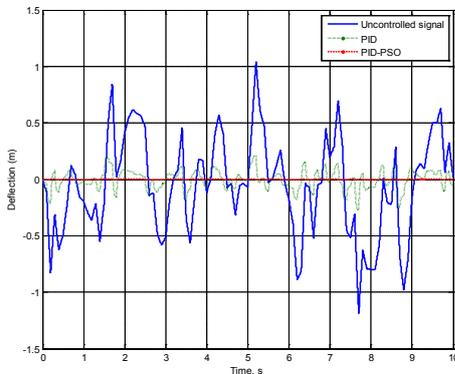
Fig. 14: Performance of proposed controller on RLS model

Table 4: Performances of end-point vibration for all models

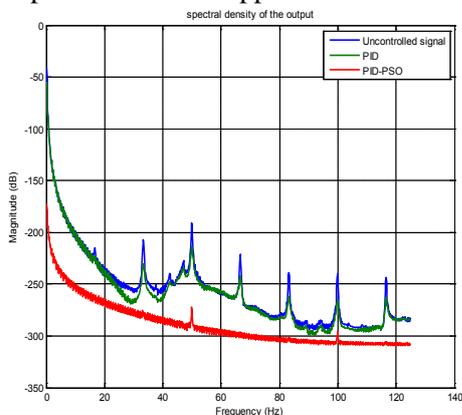
Modeling domain	Control schemes	PID Parameter	MSE	Attenuation (dB)		
				Mode 1	Mode 2	Mode 3
RLS	PID	$K_{pr2} = 10.00$ $K_{ir2} = 1.18$ $K_{dr2} = 410.76$	-	1.8	0.2	0.1
	PID-PSO	$K_{ps2} = 13.18$ $K_{is2} = 6.71$ $K_{ds2} = 3.89 \times 10^3$	6.7042×10^{-8}	25.7	28.9	42.3
PSO	PID	$K_{p2} = 10.00$ $K_{i2} = 0.13$ $K_{d2} = 0.00$	-	8.8	23.4	22.6
	PID-PSO	$K_{ps} = 5.25 \times 10^3$ $K_{is} = 3.95$ $K_{ds} = 29.90$	3.9334×10^{-8}	48.3	67	77.2



(a) MSE as a function of number of iteration for PID-PSO



(b) End-point vibration suppression in time domain



(c) Spectral density

Fig 15: Performance of proposed controller on PSO model

8 Conclusion

This paper has presented design of an optimum PID controller using PSO for control of a flexible manipulator system. A simulation environment characterizing the dynamic behavior of the flexible manipulator system was first developed using finite difference method. Subsequently, flexible manipulator system has been modelled using PSO modelling techniques in comparison with conventional RLS method in order to obtain the transfer function of the system response. Next, hybrid PID controller was employed for control of flexible manipulator. The optimum gains that acquired through global search of PSO technique has been tested on the control structure. System responses including input tracking and vibration suppression at the end-point has been evaluated. Simulation results reveals that proposed controller has successfully position the flexible link to the desired position with reduction of vibration at end-point.

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:

- [1] M.O. Tokhi, Z. Mohamed & A.K.M. Azad. (1996). *Finite Difference and Finite Element Simulation of a Flexible Manipulator*, Research Report no. 617, Department of Automatic Control & System Engineering, University of Sheffield, UK.

- [2] M.O. Tokhi, Z. Mohamed & M.H. Shaheed. (2001). Dynamic Characterization of a Flexible Manipulator System, *Robotica*, Vol 19, pp 571-580.
- [3] M.O.Tokhi, A.K.M. Azad, H. Poerwanto, S. Kourtis & M.J. Baxter. (1996). A Simulink Environment for Simulation & Control of Flexible Manipulator Systems. *UKACC International Conference on Control '96*, Vol 1, University of Sheffield, UK, 2-5 September 1996, pp. 210-215.
- [4] Noorfadzli bin Abdul Razak (2007). *Modeling of Single-link Flexible Manipulator with Flexible Joint*, Master Thesis, Universiti Teknologi Malaysia.
- [5] Mohammad Vakil (2008). *Dynamics and Control of Flexible Manipulators*, Doctor of Philosophy, University of Saskatchewan, Saskatoon, Canada.
- [6] R.M. Mahamood and J.O. Pedro (2011). Hybrid PD-PID with Iterative Learning Control for Two-Link Flexible Manipulator. *Lecture Notes in Engineering and Computer Science*, Vol.2194 (1).
- [7] M.A. Ahmad, A.N.K. Nasir, R.M.T. Raja Ismail and M.S. Ramli (2009). Comparison of Hybrid Control Schemes for Vibration Suppression of Flexible Robot Manipulator. *International Conference on Computer Modeling and Simulation 2009*, Macau, 20-22 Feb 2009.
- [8] Mohd S. Saad, H. Jamaluddin and Intan Z. Mat Darus. (2012). Implementation of PID controller Tuning using Differential Evolution and Genetic Algorithms. *International Journal of Innovative Computing Information and Control*, Vol.8 (11).
- [9] B.A. Md Zain, M.O. Tokhi and S.F. Toha. (2009). PID based Control of a Single Link Flexible Manipulator in Vertical Motion with Genetic Optimisation. *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, Athens, 25-27 Nov 2009.
- [10] M. Settles. (2005). *An Introduction to Particle Swarm Optimization*, Department of Computer Science, University of Idaho, Moscow:1-8.
- [11] S. Julai, M.O. Tokhi, M. Mohamad & I.A. Latiff. (2009). Control of a Flexible Plate Structure using Particle Swarm Optimization. *Evolutionary Computation, 2009, CEC'09*. Trondheim, 18-21 May 2009, pp 3183-3190.
- [12] Hanhong Zhu, Yi Wang, Kesheng Wang, Yun Chen. (2011). Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications*, Volume 38, Issue 8, August 2011, pp 10161-10169.
- [13] S.M. Salleh, M.O. Tokhi, S. Julai, M.Mohamad and I.A. Latiff. (2009). PSO-Based Parametric Modelling of a Thin Plate Structure, *2009 Third UKSim European symposium on Computer Modeling and Simulation*, pp.43-48.
- [14] S. Panda and N.P. Padhy. (2007). Comparison of Particle Swarm Optimization and Genetic Algorithm for TCSC-based Controller Design. *International Journal of Computer Science and Engineering*. 1(1).
- [15] H.I. Jaafar, Z. Mohamed, A. F. Z. Abidin and Z. Ab Ghani (2012). PSO-tuned PID controller for a nonlinear Gantry Crane System. *IEEE International Conference on Control System, Computing and Engineering*. Penang, Malaysia, 23-25 Nov.
- [16] S.E. Selvan, S. Subramanian and S.T Solomon (2003). Novel Technique for PID Tuning by Particle Swarm Optimization. *Seventh Annual Swarm Users/Researchers Conference*.
- [17] M.O. Tokhi & A.K.M. Azad. (2008). *Flexible Robot Manipulators: Modelling, Simulation & Control*. London, UK: Institution of Engineering & Technology
- [18] H. Mohd Yatim, I.Z. Mat Darus & M. Mohamad. (2012). Parametric Identification and Dynamic Characterisation of Flexible Manipulator System. *2012 IEEE Conference on Control, Systems & Industrial Informatics, ICCSII2012*. Bandung, Indonesia, 23-26 September 2012.
- [19] Blevins R.D. (1979). *Formulas for Natural Frequency & Mode Shape*. New York: Van Nostrand Reinhold
- [20] I.Z. Mat Darus, F.M. Aldebrez & M.O. Tokhi. (2004). Parametric Modeling of a Twin Rotor System using Genetic Algorithms. *Proceedings of the 1st IEEE International Symposium on Control, Communications and Signal Processing, ISCCSP 2004*, Hammamet, Tunisia, 21-24 March 2004.
- [21] R. Hassan, B. Cohaniam, O. De Weck, & G. Venter. (2005). A comparison of Particle Swarm Optimization and the Genetic Algorithm. *Proceedings of the 46th AIAA Structures, Structural Dynamics & Materials Conference*, 18-21 April 2005.