Intelligent Control and Automation, **2011, 2, 430-449**
doi:10.4236/ica.2011.24050 Published Online November 2011 (http://www.SciRP.org/journal/ica)

Particle Swarm Optimization Algorithm vs Genetic Algorithm to Develop Integrated Scheme for Obtaining Optimal Mechanical Structure and Adaptive Controller of a Robot

Rega Rajendra, Dilip K. Pratihar
Soft Computing Laboratory, *Department of Mechanical Engineering*, *Indian Institute of Technology*, *Kharagpur*, *India*
E-mail: *regaraj@gmail.com*, *dkpra@mech.iitkgp.ernet.in*

Received July 20, 2011; *revised August* 11, 2011; *accepted September* 15, 2011

Abstract

The performances of Particle Swarm Optimization and Genetic Algorithm have been compared to develop a methodology for concurrent and integrated design of mechanical structure and controller of a 2-dof robotic manipulator solving tracking problems. The proposed design scheme optimizes various parameters belonging to different domains (that is, link geometry, mass distribution, moment of inertia, control gains) concurrently to design manipulator, which can track some given paths accurately with a minimum power consumption. The main strength of this study lies with the design of an integrated scheme to solve the above problem. Both real-coded Genetic Algorithm and Particle Swarm Optimization are used to solve this complex optimization problem. Four approaches have been developed and their performances are compared. Particle Swarm Optimization is found to perform better than the Genetic Algorithm, as the former carries out both global and local searches simultaneously, whereas the latter concentrates mainly on the global search. Controllers with adaptive gain values have shown better performance compared to the conventional ones, as expected.

Keywords: Manipulator, Optimal Structure, Adaptive Controller, Genetic Algorithm, Neural Networks, Particle Swarm Optimization

1. Introduction

In the age of high precision manufacturing of components and parts, there is a need to develop precisely controlled manipulator for handling micro-precision jobs and machining applications. Also, the manipulator has to traverse a given trajectory as accurately as possible to avoid variations in trajectory causing damage to other parts in assembly or distorted machining of geometric components or inaccurate welding of jobs. Thus, trajectory tracking is one of the most important tasks to be performed by the manipulator.

A user specify motions as the sequences of points through which a tool fixed to the end-effector of a manipulator has to pass. The effectiveness of such motion specifying mechanism is greatly increased, if the tool moves in a specified path between the user-specified points. Intermediate points are interpolated along the path at regular intervals of time during the motion, and the manipulator's kinematics equations are solved to produce the corresponding joint parameter values. The developed path interpolating function offers several advantages, including less computational cost and improved motion characteristics. A second method uses a motion planning phase to pre-compute enough intermediate points, so that the manipulator may be driven by interpolation of joint parameter values, while keeping the tool on an approximately pre-specified path. This technique allows a substantial reduction in real-time computation. The planning is done by an efficient recursive algorithm, which generates enough intermediate points to guarantee that the tool's deviation from the path to be tracked stays within a pre-specified error bounds.

Several attempts were made by various investigators to design and develop suitable controllers for the robots. The following model-based robot controllers had been used: computed torque control, non-adaptive Proportional Derivative (PD) control, PD control with feedback, and others. Some of those attempts are discussed below.

Qu [1] developed PD control scheme to solve trajectory tracking problem of a manipulator. He concluded that PD control with time-varying gains could guarantee global stability for the trajectory following problem of a manipulator. Homsup and Anderson [2] proposed a model-based PD control for a 2-dof planar manipulator. Its performance was measured using system performance ellipsoids drawn utilizing the information of control parameters, manipulator's Jacobian and inertia matrices. Thus, the performance was dependent not only on the robot's kinematics and dynamics, but also on the control algorithm. A correlation between trajectory tracking errors and workspace location was established. Kelly and Salgado [3] developed a design procedure for selecting gain values (that is, K_p and K_d) of PD control with computed feed-forward of robot dynamics and desired trajectory. The performance of their approach was tested through computer simulations on 2-dof manipulator. An evolutionary PD control strategy was used by Ouyang and Zhang [4] to improve trajectory tracking performance of a closed-loop robot manipulator. It could ensure good trajectory tracking performance without using the knowledge of robot dynamics. The performance of their strategy was tested through computer simulations and found to be better than conventional PD and non-linear PD control strategies in terms of trajectory tracking performance and fluctuation in the actuator's torques.

Ravichandran *et al.* [5] developed a scheme for simultaneous plant-controller design optimization for a 2-dof planar rigid manipulator and non-linear PD controller. During optimization, a heuristic search technique named evolution strategy and a weighted-sum problem formulation were adopted to take care of multiple objectives and generate a single Pareto-optimal solution. The performance of the developed scheme was demonstrated on computer simulations, and the potential of the scheme to yield desirable designs of the manipulator and controller was realized.

Soft computing-based tools [6] had also been used to design and develop suitable controller for the robots. Some of those studies are discussed here. Ozaki *et al.* [7] proposed a non-linear compensator using neural networks for trajectory control of a 2-dof manipulator. Its performance was compared with that of adaptive controller proposed by Craig [8] in compensating unstructured uncertainties of the manipulator. The neural network-based approach was found to be effective and efficient in learning manipulator dynamics, and consequently could track the trajectory accurately. Ghalia and Alouani [9] designed a fuzzy logic-based controller of a 2-dof manipulator to determine appropriate gain values of the compensator for tracking some trajectories accurately. The performance of their approach was tested on computer simulations. Rueda and Pedrycz [10] proposed a hierarchical fuzzy-neural-PD controller for N-dof robot manipulators to solve tracking problems accurately. In their approach, a coordinator was implemented as a fuzzy-neural network, whose purpose was to select activation levels for local regulators implemented as time-varying PD controllers.

Park and Asada [11] developed a concurrent design method of determining mechanical structure and suitable controller for a 2-dof planar non-rigid manipulator. An attempt was made to achieve high speed positioning by optimizing arm link geometry, actuator locations and feedback gains. In their study, optimal feed-back gains minimizing the settling time were obtained as the fluctuations of structural parameters, which were optimized using a non-linear programming technique. Based on the obtained optimal design, one prototype robot was built and an outstanding performance was observed.

The concept of PSO algorithm was introduced by Kennedy and Eberhart [12] in 1995. It is a population-based search algorithm, which is initialized with the population of random solutions, called particles, and the population is known as swarm. Several modifications in the PSO algorithm had been done by various researchers. Shi and Eberhert [13] introduced a new parameter called inertia weight into the original PSO algorithm, which played an important role in balancing the global and local searches. Clerc and Kennedy [14] analyzed how a particle carries out its search in a complex problem space and modified the original PSO on the basis of this analysis. Chen *et al.* [15] improved the PSO algorithm with adaptive inertia weight W and acceleration coefficients in order to maintain population diversity and sustain good convergence capacity to optimize back-propagation neural networks.

PSO is simple in concept, as it has a few parameters only to be adjusted. It has found applications in various areas like constrained optimization problems, min-max problems, multi-objective optimization problems and many more. In addition to these application areas, it has been applied to evolve weights and structure of some neural networks (NNs). Han and Jiang [16] proposed an endpoint prediction model of Basic Oxygen Furnace (BOF) steelmaking based on PSO-tuned radial basis function neural network. Braik *et al.* [17] developed a mechanism to improve performance of NN in modeling a chemical process through PSO. Abe and Komuro [18] utilized an NN, tuned by PSO to save energy of the flexible manipulator for point-to-point motion. Joint angles generated by the NN suppressed the residual vibration and hence, minimized the motor torques, which was kept as the objective function. The authors conducted numerical studies and verified by the same with experiments and concluded that PSO is an efficient optimizer.

Although a considerable amount of work has been carried out in this field of research, there is still a need for an integrated scheme for obtaining optimal mechanical structure and controllers with adaptive gain values of 2-dof manipulator solving tracking problems. The aim of this study is to obtain an optimal mechanical structure of the manipulator along with adaptive controller that enables high precision positioning. The links of the manipulator are treated as rigid bodies. To speed up operations, one needs powerful actuators and lightweight arm links. In positioning, however, the major issue is to minimize the settling time of the control system. The settling time depends on a broad range of design parameters including mass and stiffness properties of the links, gain values of the compensator, and others. These parameters are coupled to each other and have intricate interactions with respect to the robot's settling time. For instance, increasing the structural stiffness alone does not always decrease the settling time. All the design parameters must be considered in an integrated manner in order to optimize the performance. In the present paper, an integrated scheme for obtaining optimal mechanical structure and controller of a 2-dof manipulator solving path tracking problems, has been proposed. Four approaches are developed, and their performances have been compared on two path tracking problems.

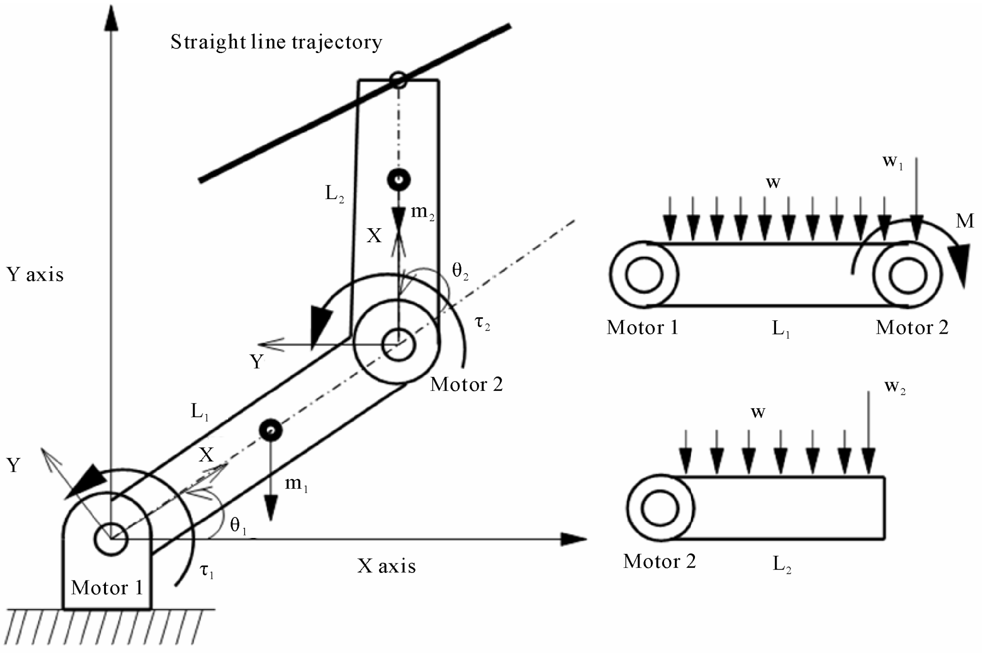
The remaining part of this paper has been organized as follows: Section 2 deals with mathematical formulation of the problem. Tools and techniques used in the present work are discussed and the proposed algorithms have been explained in Section 3. Results are stated and discussed in the Section 4. Some concluding remarks are made in Section 5 and the scope for future work is indicated in Section 6.

2. Mathematical Formulation of the Problem

This section deals with mathematical formulation of the problem. It has been posed as a constrained optimization problem

2.1. Trajectory Analysis

A two degrees of freedom serial manipulator with hollow circular cross-section links of lengths: L_1 and L_2 (where $L_1 > L_2$) has been considered, as shown in **Figure 1**. A photograph of the set-up has been displayed **Figure 2**. The links are connected using rotary joints. The end-effector of the manipulator is directed to track one straight and another circular paths separately, starting from initial position (X_i, Y_i) up to final position (X_f, Y_f) in time t. The forward kinematics equations of the manipulator can be written as follows:



(a) (b)

Figure 1. (a) A two degrees of freedom manipulator traversing a straight-line trajectory with torques applied at its joints; (b) load distributions on two links of the manipulator.

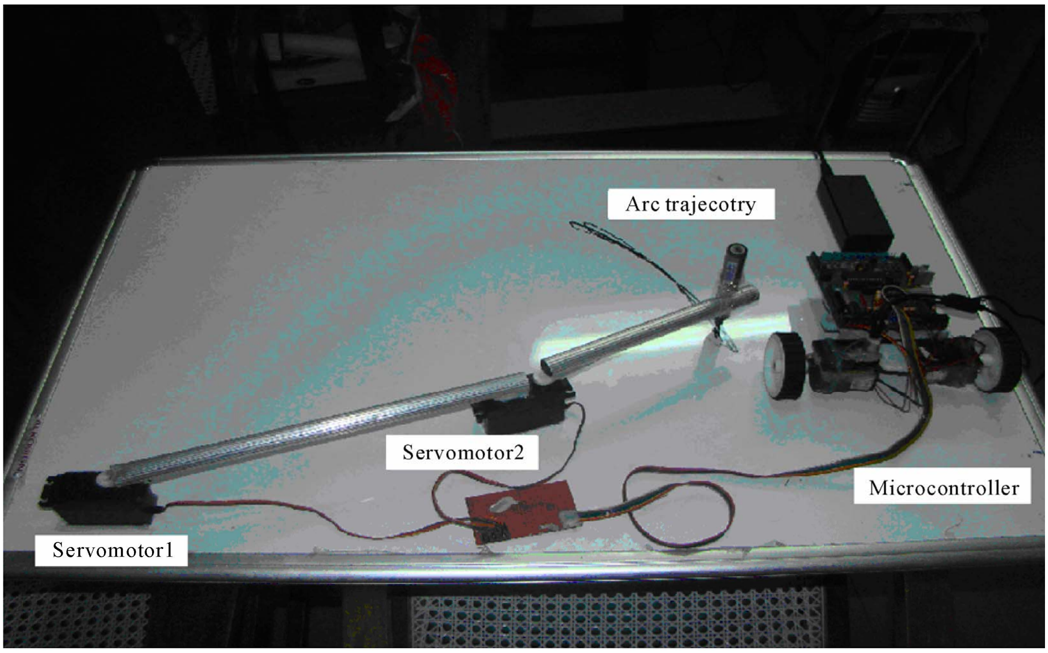


Figure 2. A set-up of the 2-dof manipulator.

$$\begin{aligned} X &= L_1 \cos\theta_1(t) + L_2 \cos\left(\theta_1(t)+\theta_2(t)\right) \\ Y &= L_1 \sin\theta_1(t) + L_2 \sin\left(\theta_1(t)+\theta_2(t)\right) \end{aligned} \quad (1)$$

The joint angles: $\theta_1(t)$ and $\theta_2(t)$ can be obtained by carrying out inverse kinematics, as given below:

$$\begin{aligned} \theta_2(t) &= \cos^{-1}\left(\frac{X^2+Y^2-L_1^2-L_2^2}{2L_1L_2}\right) \\ \theta_1(t) &= \tan^{-1}\left(\frac{Y}{X}\right) - \tan^{-1}\left(\frac{L_2\sin\theta_2(t)}{L_1+L_2\cos\theta_2(t)}\right) \end{aligned} \quad (2)$$

2.2. Determination of Power Consumption

Lagrange-Euler formulation [19] is used to determine energy consumption of the manipulator at its two joints. The manipulator's joints torques: $\tau_1(t)$ and $\tau_2(t)$ consist of inertia, centrifugal and Coriolis and gravity terms, as given below.

$$\tau(t) = D\left(\theta(t)\left(\ddot{\theta}\right)(t)\right) + h\left(\theta(t),\left(\dot{\theta}\right)(t)\right) + c\left(\theta(t)\right) \quad (3)$$

where D, h, and c represent inertia, centrifugal and Coriolis, and gravity terms, respectively. Now, torques required at two joints of the manipulator (refer to **Figure 1**) can be determined as follows:

$$\tau_1 = D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + h_1 + c_1,$$

$$\tau_2 = D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + h_2 + c_2,$$

where

$$D_{11} = \frac{1}{3}m_1L_1^2 + m_2L_1L_2\cos\theta_2(t) + m_2L_1^2 + \frac{1}{3}m_2L_2^2 + \frac{1}{4}(m_1+m_2)\left(R^2-r^2\right),$$

$$D_{12} = D_{21} = \frac{1}{4}m_2\left(R^2-r^2\right) + \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2\cos\theta_2(t),$$

$$D_{22} = \frac{1}{3}m_2L_2^2 + \frac{1}{4}m_2\left(R^2-r^2\right),$$

$$h_1 = -m_2L_1L_2\sin\theta_2(t)\dot{\theta}_1\dot{\theta}_2 - 0.5m_2L_1L_2\sin\theta_2(t)\dot{\theta}_2^2,$$

$$h_2 = 0.5m_2L_1L_2\sin\theta_2(t)\dot{\theta}_1^2,$$

$$c_1 = 0.5m_1gL_1\cos\theta_1(t) + 0.5m_2gL_2\cos\left(\theta_1(t)+\theta_2(t)\right), + m_2gL_1\cos\left(\theta_1(t)\right)$$

$$c_2 = 0.5m_2gL_2\cos\left(\theta_1(t)+\theta_2(t)\right).$$

Now, energy consumed by the manipulator (kW) in tracing the trajectory can be determined as follows:

$$E' = \frac{1}{100}\left[\sum_{i=1}^{100}\sum_{j=1}^{2}\left|\tau_{ij}\theta_j(t_i)\right| + 0.025\tau_{ij}^2\right], \quad (4)$$

where the second part of this equation indicates the energy dissipated by the motors connected at the joints [20]. It is important to mention that the cycle time has been assumed to be equal to 100 seconds and it is divided into 100 instants.

2.3. Structural Analysis: Bending of the Links

Structural analysis [21] is confined to Hooke's Law of analysis, for which the manipulator is used to verify its behavior while tracking the trajectory with minimum error. Due to the weight of the links and motors, there is a chance of bending and deflection of the links and thus, it may deviate from the desired trajectory. To prevent mechanical failure of the links, the developed stress $\sigma_{developed}$ should be less than the allowable stress $\sigma_{allowable}$, that is, $\sigma_{developed} < \sigma_{allowable}$.

For minimum consumption of energy by the manipulator, suitable hollow cross-section of the links possessing sufficient strength and rigidity to withstand the bending and deflection is to be determined. The links are made up of aluminum. **Figure 1(b)** displays load distribution, on two links of the manipulator, where W_1 indicates the weight of the second motor and second link, and load due to τ_1; W_2 denotes the load due to τ_2; and w represents uniformly distributed load on the links due to self-weight. Link1 can be considered as a beam of hollow circular cross-section, which is subjected to W_1, w and a fixed moment of $M = -0.5wL_2^2\cos\left(\theta_1(t)+\theta_2(t)\right)$. Thus, it is to be designed considering the effect of bending moment M_1, which can be determined as follows:

$$M_1 = -\left(W_{m2}L_1 + wL_1L_2 + 0.5wL_1^2 \pm \tau_1\right)\cos\theta_1(t) \\ -0.5wL_2^2\cos\left(\theta_1(t)+\theta_2(t)\right), \quad (5)$$

where W_{m2} represents the weight of second motor. As the two links are assumed to have same cross-section and link1 is more critical compared to the other one, an attention has been focused on its safe design. The developed bending stress on link1 can be determined as follows:

$$\sigma_{developed} = \left[\frac{4M_{\max}}{\pi\left(R^2-r^2\right)}\right], \quad (6)$$

where R and r represent the outer and inner radii of the hollow circular link and the maximum bending moment $M_{\max}$ can be determined as follows :

$$M_{\max} = -W_{m2}L_1 - wL_1L_2 \pm \tau_1 - 0.5w\left(L_1^2+L_2^2\right).$$

It is important to mention that linear relationships for elastic modulus (E) and yield strength (σ_y) of aluminum with its density (ρ) have been established according to [22-28], as given below.

$$E = 1.92554e9 + 2.56675\rho \quad (7)$$

$$\sigma_{allowable} = 7.52288E7 + 30638.91191\rho \quad (8)$$

It is to be noted that allowable stress $\sigma_{allowable}$ has been kept equal to σ_y.

2.4. Stability and Response Analysis in Trajectory Tracking

The control problem of a robotic manipulator is that of determining the time history of joint inputs required to cause its end-effectors to execute a commanded motion typically specified either as a sequence of end-effector's positions and orientations or as a continuous path. Depending on the controller design, the joints inputs may be in the form of joint forces, torques or inputs from actuators. A particular controller design has a significant impact on the performance of the manipulator and consequently, on the range of its possible applications (either point-to-point or continuous). In addition, mechanical design of the manipulator itself will influence the type of control scheme needed. It is to be noted that permanent magnet DC motors with gear reduction are commonly used in robots. The design objective is to choose the compensator (either PD or PID) in such a way that the plant output tracks or follows a desired path. For set points, tracking is a problem of constant or step reference command θ^d that arises in point-to-point motion [29], [30]. For a PD compensator, the control input $U(s)$ is given in the Laplace domain as follows:

$$U(s) = \left(K_p + K_d s\right)\left(\theta^d(s) - \Theta(s)\right) \quad (9)$$

The resulting closed-loop system response is given by the following expression:

$$\Theta(s) = \frac{K_p + K_d s}{\Omega(s)}\theta^d(s) - \frac{D(s)}{\Omega(s)}, \quad (10)$$

where $D(s)$ is the disturbance on the system (here, it has been assumed to be equal to zero), and $\Omega(s)$ is the closed-loop characteristic polynomial given by

$$\Omega(s) = Js^2 + \left(B + K_d\right)s + K_p \quad (11)$$

The closed-loop system will be stable for positive values of K_p, K_d and bounded disturbances, and the joint error is given by $\theta^d(s) - \Theta(s)$. For the PD compensator given in Equation (9), the step response is determined by the closed-loop natural frequency ω and damping ratio ζ. The compensator's gain values: K_p and K_d can be obtained as follows:

$$K_p = \omega^2 J, \\ K_d = 2\zeta\omega J - B, \quad (12)$$

where J is the inertia of the links, motor and gear, which can be determined from the relationship: $1/Js^2 = \theta/\tau$; B represents effective damping of the system [31]; ω indicates the natural frequency. Now, the joint errors can be calculated as follows:

$$\theta_{ijterror} = \theta_i^d(t) - \Theta_i(t), \text{ where } i = 1,2,\cdots \tag{13}$$

The trajectory equations are found to be as follows after considering error in motors and deflection of links.

$$\begin{aligned} X_{11} &= L_1\cos(\Theta_1(t)), \\ Y_{11} &= L_1\sin(\Theta_1(t)), \\ X_{22} &= X_{11} + L_2\cos(\Theta_1(t)+\Theta_2(t)), \\ Y_{22} &= Y_{11} + L_2\sin(\Theta_1(t)+\Theta_2(t)). \end{aligned} \tag{14}$$

Therefore, the trajectory error can be calculated as follows:

$$\begin{aligned} X_{error} &= X_2 - X_{22} \\ Y_{error} &= Y_2 - Y_{22} \end{aligned} \tag{15}$$

Total Trajectory error Err_{total} can be determined like the following.

$$Err_{total} = \sum_{i=1}^{100} X_{error} + \sum_{i=1}^{100} Y_{error}, \tag{16}$$

Now, average error (Err_{avg}) can be obtained as follows:

$$Err_{avg} = \frac{Err_{total}}{2\times 100}, \tag{17}$$

This problem can be posed as a constrained optimization as given below.

Minimize

$$E' + Err_{avg}, \tag{18}$$

subject to

$$\sigma_{developed} < \sigma_{allowable}$$

and

$$\begin{gathered} 0.001 < t' < 0.0029, \\ 2600.0 < \rho < 2800.00, \\ 5.0 < \omega_1 < 20.0, \\ 10.0 < \omega_2 < 15.0, \\ 1.0 < s_1 < 1.0, \\ 1.0 < s_2 < 1.0. \end{gathered}$$

3. Proposed Algorithm

The said constrained optimization problem has been solved using a real-coded Genetic Algorithm (GA) [32], as the variables are real in nature. Moreover, a real-coded GA has some advantages over the binary-coded GA, such as its ability to provide more precise solutions and there is no hamming cliff problem [6]. A GA is a population-based search and optimization technique, which works based on the mechanics of natural genetics and principle of natural selection [33,34]. It starts with a population of solutions created at random, which are further modified using the operators like reproduction, crossover and mutation. In the present study, tournament selection, polynomial mutation and simulated binary crossover have been used. Interested readers may refer to [6] for a detailed description of the algorithm.

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart in 1995 [12] is a stochastic population-based evolutionary computation technique, which has also been used to solve the said optimization problem. It can be linked to bird flocking, fish schooling or sociological behavior of a group of people. It has been used to solve a variety of optimization problems. In PSO, the population of solution is known as swarm. It uses a number of agents known as particles that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in an N-dimensional space, which adjusts its flying according to its own flying experience and that of other particles. Each particle keeps track of its coordinates in the solution space, which are associated with the best solution in terms of fitness that has been achieved so far by it. This value is called personal best, *Pbest*. Another best value that is tracked by the PSO is the best value obtained so far by any particle lying in its neighborhood. This value is called global best, *Gbest*. The basic concept of PSO lies in accelerating each particle toward its *Pbest* and the *Gbest* locations, with a random weighted acceleration at each time step. Four approaches have been developed as explained below.

3.1. Approach 1

Optimization using a real-coded GA only

Figure 3 shows the working cycle of a real-coded GA. A GA-solution carries real values of six variables, such as t', ρ, ω_1, ω_2, s_1, s_2. The constrained optimization problem has been solved using a penalty function approach [6]. As it is a minimization problem, a fixed positive penalty of P' = 100 has been added for each violated constraint (if any). Thus, the fitness (f) of the GA-solution can be represented as follows:

$$f = E' + Err_{avg} + P' \tag{19}$$

The GA tries to find optimal solution iteratively.

3.2. Approach 2

Optimization using a combined neural network and

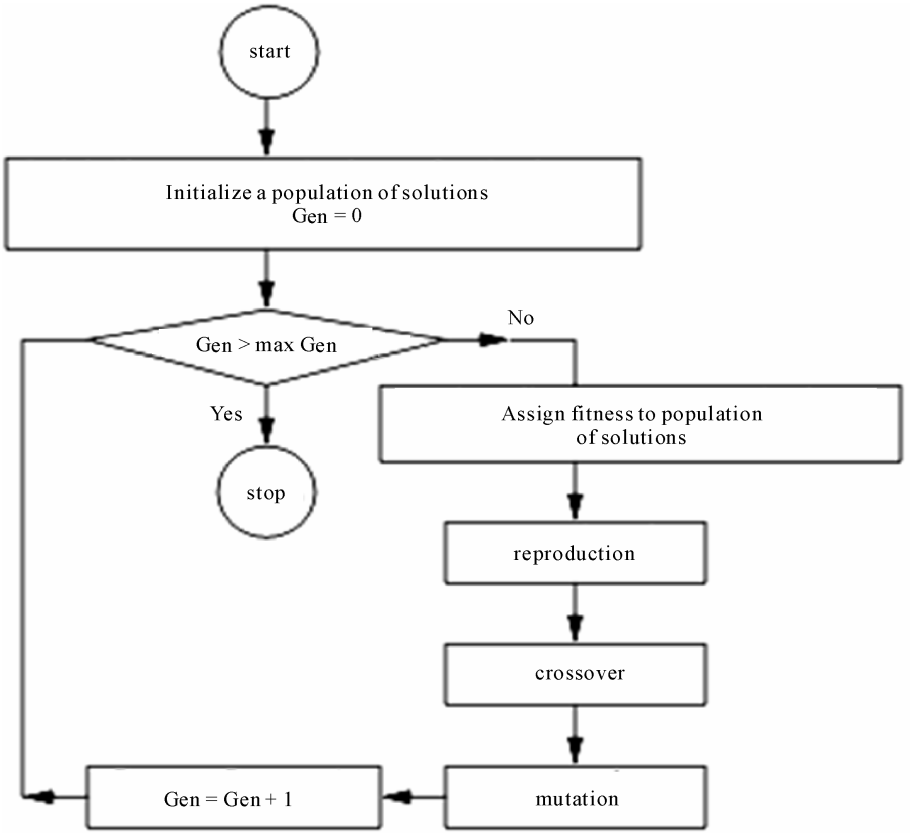

start
Initialize a population of solutions
Gen = 0
Gen > max Gen
No
Yes
stop
Assign fitness to population of solutions
reproduction
crossover
mutation
Gen = Gen + 1

Figure 3. A schematic view representing working cycle of real-coded GA (that is, approach 1).

real-coded GA

In approach 1, one GA-solution supplies one set of values of ω_1, ω_2, s_1, s_2 for the entire cycle time of 100 seconds. However, it may be required to adopt different sets of the said parameters for different instants of the cycle time to track the trajectory accurately. It has been tried in this approach. For each duration, X_{error} and Y_{error} are determined by comparing the calculated values of X and Y with their corresponding target values, and these are fed as inputs to a feed-forward neural network (refer to Appendix A), which consists of three layers, namely input, hidden and output layers. There are two and four neurons in the input and output layers of the network, as decided by the number of inputs and outputs, respectively. A thorough parametric study has been carried out to determine the numbers of hidden neurons, which has come to be equal to four. The first and third output neurons have log-sigmoid transfer function, and the second and fourth neurons are assumed to have tan-sigmoid transfer function. The neurons of input and hidden layers are assumed to have linear and tan-sigmoid transfer functions, respectively. In this approach, the GA carries information of thickness of the links (t'), density of the link material (ρ) and thirty real variables related to the neural network, such as eight and sixteen connecting weights between the input and hidden layers (that is, [V]) and hidden and output layers (that is, [W]), respectively; bias and coefficients of transfer function of hidden neurons; four different coefficients of transfer functions of the output neurons. The values of ω_1, ω_2, s_1 and s_2 are determined as the outputs of the neural network. **Figure 4** displays the flowchart of approach 2. The fitness of a solution has been calculated using Equation (19).

3.3. Approach 3

Optimization using Particle Swarm Optimization (PSO) only

Similar to other population-based optimization methods, such as GAs, the PSO starts with the random initialization of population particles in the search space. The PSO algorithm works based on the social behaviour of the particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual toward its own best location and toward the best particle of the entire swarm at each time step (generation) [12,14]. In PSO algorithm, the trajectory of each individual in the search space is adjusted by dynamically changing the velocity of each particle, accord-

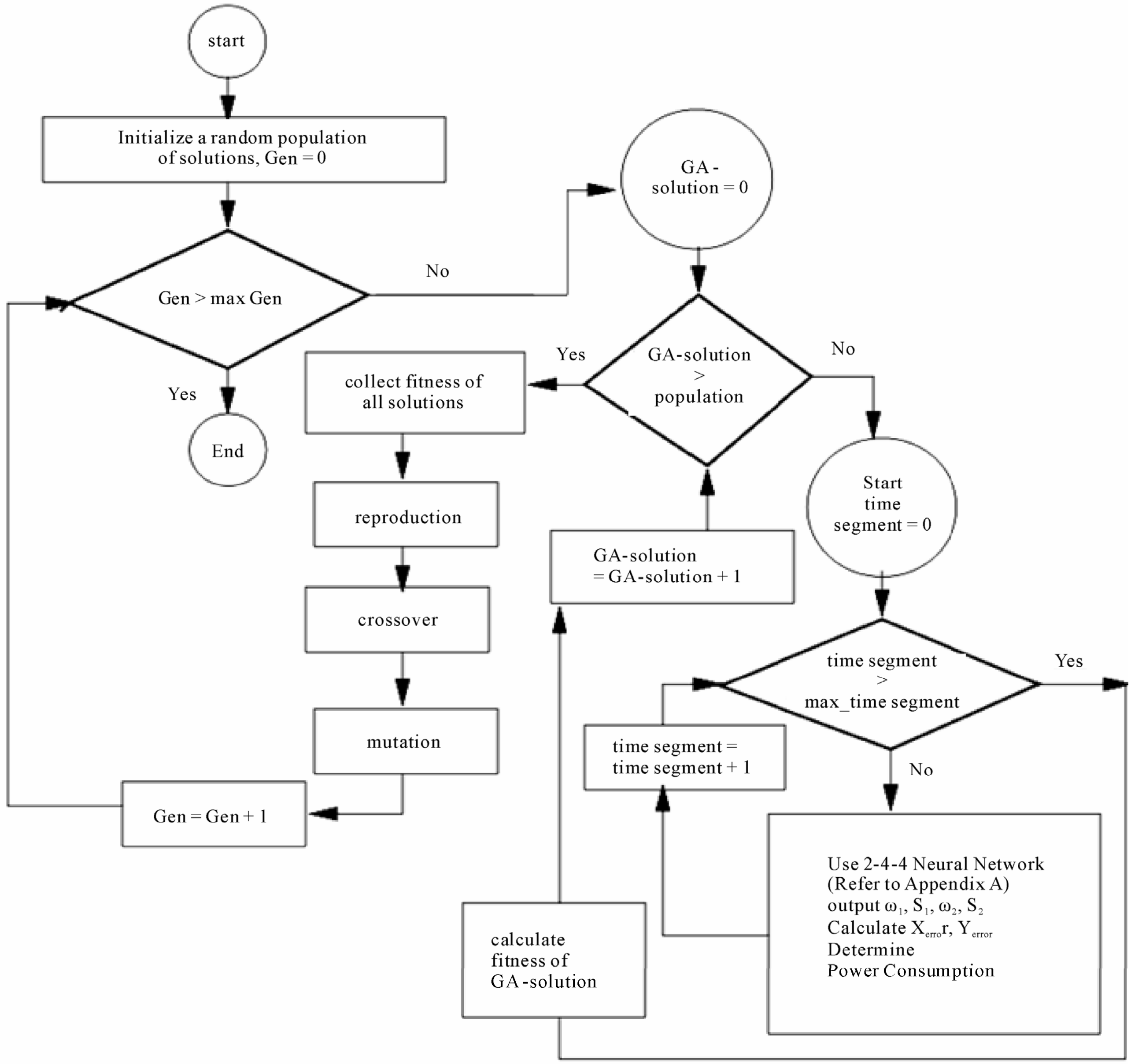

start
Initialize a random population of solutions, Gen = 0
Gen > max Gen
No
Yes
End
GA - solution = 0
GA-solution > population
Yes
No
collect fitness of all solutions
reproduction
crossover
mutation
Gen = Gen + 1
Start time segment = 0
GA-solution = GA-solution + 1
time segment > max_time segment
Yes
No
time segment = time segment + 1
Use 2-4-4 Neural Network (Refer to Appendix A) output ω1, S1, ω2, S2 Calculate Xerror, Yerror Determine Power Consumption
calculate fitness of GA-solution

Figure 4. Flowchart of approach 2.

ing to its own flying experience and that of other particles in the search space. The position and velocity vectors of i^{th} particle in the d-dimensional search space can be represented as $X_i = (x_{i1}, \cdots, x_{id})$ and $V_i = (v_{i1}, \cdots, v_{id})$, respectively. The value of V_i vector can be varied in the range of $[-v_{\max}, v_{\max}]$ to reduce the tendency of particles to leave the search space. The value of $v_{\max}$ is usually chosen to be equal to $k \times x_{\max}$, where $0.1 \le k \le 1.0$ [35]. According to a user defined fitness function, let us say that the best position of each particle (which corresponds to the best fitness value obtained by that particle at time *t*) is $P_i = (p_{i1}, \cdots, p_{id})$ and the fittest particle found so far at time *t* is $P_g = (p_{g1}, \cdots, p_{gd})$. The new velocities and positions of the particles for the next fitness evaluation are calculated using the following two equations:

$$v_{id}(t+1) = Wv_{id}(t) + b_1 Rand(\cdot)(p_{id} - x_{id}(t)) + b_2 Rand(\cdot)(p_{gd} - x_{id}(t)), \quad (20)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (21)$$

where v_{id} is the velocity of *d*th dimension of the *i*th particle, *W* is a constant known as inertia weight [36], b_1 and b_2 denote the acceleration coefficients, and $rand(\cdot)$ and $Rand(\cdot)$ are two separately generated uniformly distributed random numbers lying in the range of $[0,1]$. The first part of Equation (20) represents the previous velocity, which provides the necessary momentum for the particles to roam across the search space. The second part of Equation (20) is known as the cognitive component that represents the personal thinking of each particle. The cognitive component encourages the particles to move toward their own best positions found so far. The third part of Equation (20) is known as the social component, which indicates the collaborative effect of the particles in finding the global optimal solution. The social component always pulls the particles toward the global best particle found so far. The PSO is becoming very popular due to its simple architecture, ease of implementation and ability to quickly converge to a reasonably good solution. The flowchart of PSO algorithm is shown in **Figure 5**. The values of six variables, namely t', ρ, ω_1, ω_2, s_1 and s_2 are supplied by the PSO during optimization and the fitness of a solution

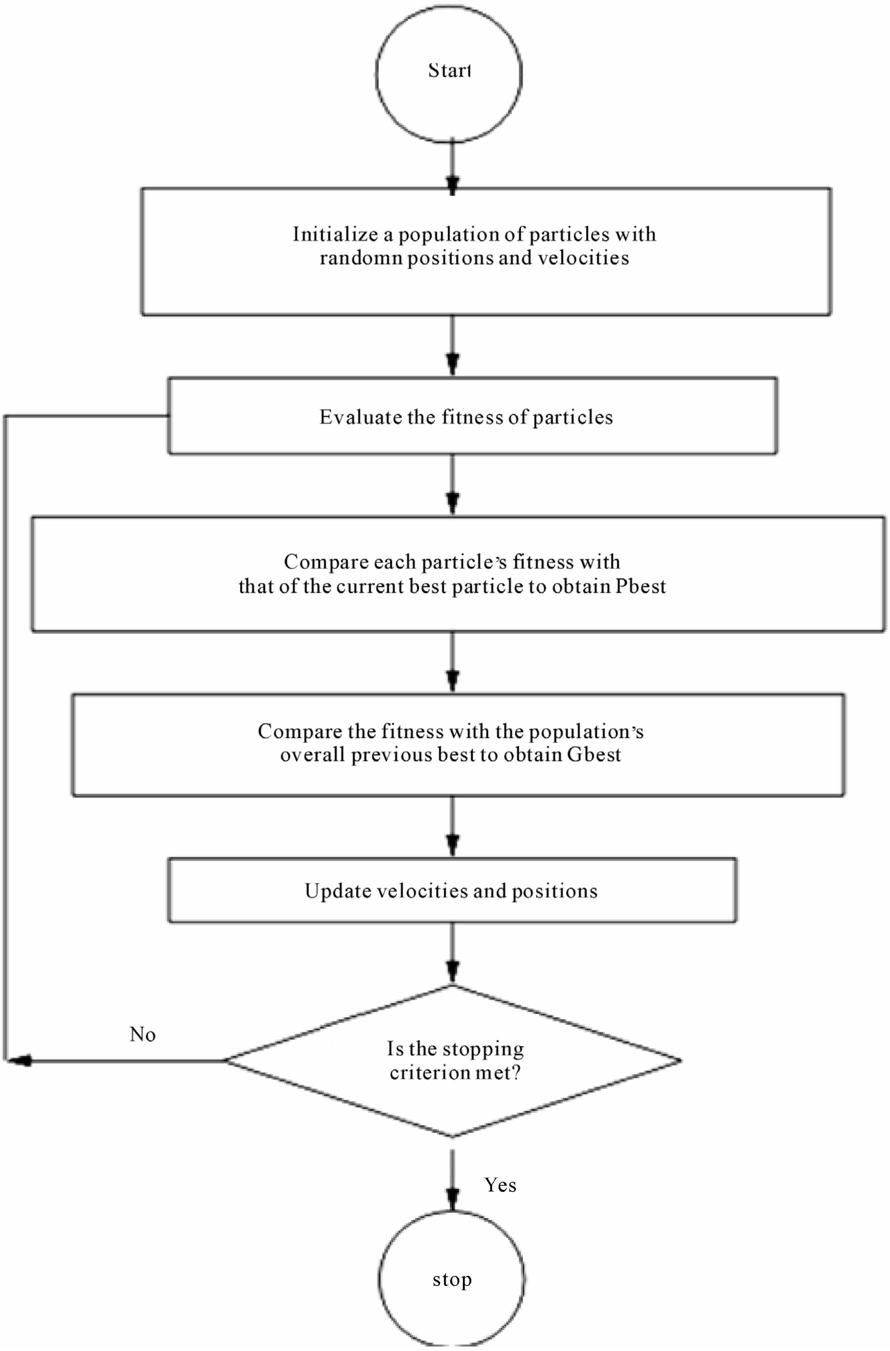

Start
Initialize a population of particles with randomn positions and velocities
Evaluate the fitness of particles
Compare each particle's fitness with that of the current best particle to obtain Pbest
Compare the fitness with the population's overall previous best to obtain Gbest
Update velocities and positions
Is the stopping criterion met?
No
Yes
stop

Figure 5. The flowchart of PSO algorithm.

has been determined using Equation (19).

3.4. Approach 4

Optimization using combined neural network and PSO algorithm

In approach 3, one PSO-solution gives one set of values of ω_1, ω_2, s_1, s_2 for the entire cycle time of 100 seconds. However, it may be required to choose separate sets of the said parameters for different instants of the cycle time to track the trajectory accurately. It has been attempted in this approach. For each duration, X_{error} and Y_{error} values are obtained by comparing the calculated values of X and Y with their corresponding target values, and these are fed as inputs to a feed-forward neural network (refer to Appendix A), which consists of three layers, namely input, hidden and output layers. The input and output layers of the network contain two and four neurons, respectively. A detailed parametric study has been carried out to decide the numbers of hidden neurons, which has turned out to be equal to four. The first and third output neurons have log-sigmoid transfer function, and the second and fourth neurons are assumed to have tan-sigmoid transfer function. The neurons of input and hidden layers are assumed to have linear and tan-sigmoid transfer functions, respectively. In this approach, the PSO solution carries information of the thickness of the links (*t*), density of the link material (*ρ*) and thirty real variables related to the neural network, such as eight and sixteen connecting weights between the input and hidden layers (that is, [*V*]) and hidden and output layers (that is, [*W*]), respectively; bias and coefficients of transfer function of hidden neurons; four different coefficients of transfer functions of the output neurons. The values of ω_1, ω_2, s_1, and s_2, are obtained as the outputs of the neural network. The fitness of a solution has been determined using Equation (19).

4. Results and Discussion

The performances of the developed four approaches have been tested through computer simulations. The lengths of two links: L_1 and L_2 are assumed to be equal to 0.3m and 0.2 m, respectively. The terms: *B* (effective damping) and ζ (damping ratio) [31] of Equation **(**12) have been set equal to 0.02 and 1.0, respectively. The outer radius of two circular links is considered as R = 0.03 m. The motor connected to the second link weighs 0.568 kg. Simulations are conducted on a P-IV PC. Results of four developed approaches are stated and discussed below for tracking of one straight path and another circular path separately.

4.1. Straight Path Tracking

The 2-dof manipulator will have to track a straight path as accurately as possible after consuming minimum power and ensuring enough mechanical strength. Inverse kinematics calculations have been carried out using Equation (2) to determine joint angles corresponding to the movement of the end-effector along a straight path in Cartesian coordinate system. A fourth-order polynomial is found to be suitable to represent the trajectory of $\theta_1(t)$ as follows: $\theta_1(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 + a_{14}t^4$, where the values of the coefficients have turned out to be as a_{10} = 0.183987352, a_{11} = −0.008541301
a_{12} = 0.000193741, a_{13} = −0.000001798 and
a_{14} = 0.000000007. Similarly, the trajectory function $\theta_2(t)$ is seen to be a cubic polynomial as follows:
$\theta_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3$, where
a_{20} = 2.279365512, a_{21} = −0.011757362,
a_{22} = 0.000026833, a_{23} = −0.000000209. Angular velocities and accelerations have been determined for both the joints by differentiating angular displacement with respect to time (*t*) for once and twice, respectively. Results of the developed four approaches are stated, discussed and compared below.

4.1.1. Results of Approach 1

As the performance of a GA depends on its parameters, a thorough parametric study is carried out to determine the values of optimal GA-parameters.

Figure 6 shows the results of the said parametric study, in which one parameter has been varied at a time keeping the others fixed.

Thus, the following GA-parameters are found to yield the best results: P_c = 0.76, P_m = 0.0055, Population size P = 150, maximum number of generations $G_{\max}$ = 150. The optimized values of the parameters: t', *ρ*, ω_1, s_1, ω_2 and s_2 are found to be equal to 0.001m, 2600.00 kg/m^3, 5.002, −0.02612, 10.00 and −0.0141, respectively. The obtained results will be discussed in **Figures 7** to **11**, at the end of this section.

4.1.2. Results of Approach 2

The set of optimal GA-parameters has been obtained using an approach explained earlier. The following GA-parameters are found to give the best results: P_c = 0.86, P_m = 0.0055, Population size P = 220, Maximum number of generations $G_{\max}$ = 240. During optimization, the ranges of t' and ρ have been kept the same with those of approach 1. The connecting weights: $[V]$, $[W]$ are varied in the range of −1.0 to 1.0. The coefficients of transfer functions of the hidden neurons (a_{1_hid}) and those of first through fourth neurons of output layers (that is, a_{2_out}, a_{3_out}, a_{4_out}, a_{5_out}**)** have

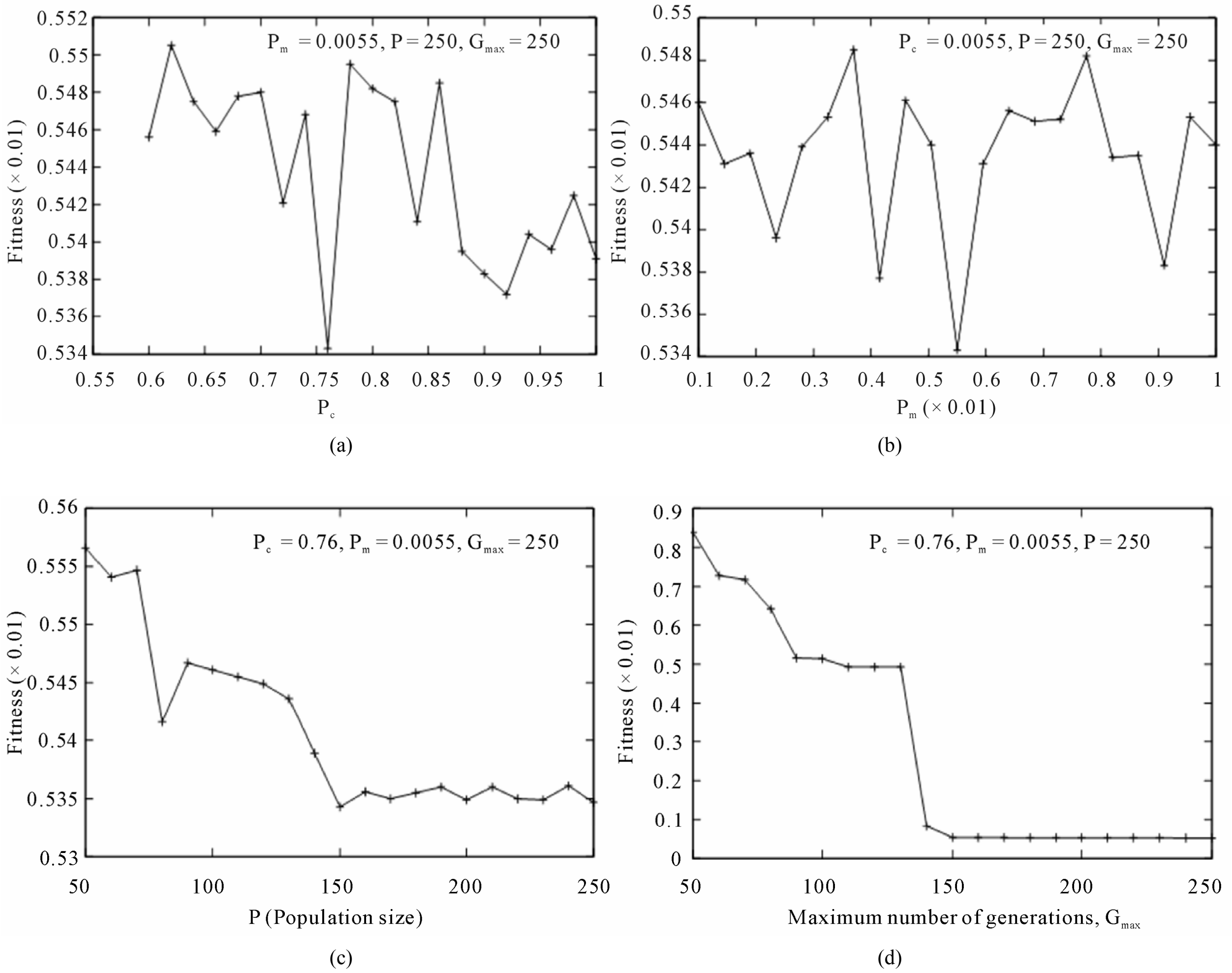


Figure 6. Results of GA-parametric study: (a) fitness vs Pc; (b) fitness vs P_m; (c) fitness vs population size P; (d) fitness vs maximum number of generations G_{max}.

been optimized in the ranges of (0.1, 1.0), (2.0, 3.0), (1.0, 2.0), (1.0, 2.0) and (1.0, 2.0), respectively. Moreover, the bias value: b has been varied from 0.000001 to 0.0001. The optimized values of the variables, which are obtained using this approach, are shown in **Table 1**. Results have been discussed and compared with those of other approaches with the help of **Figures 7** to **11**.

4.1.3. Results of Approach 3

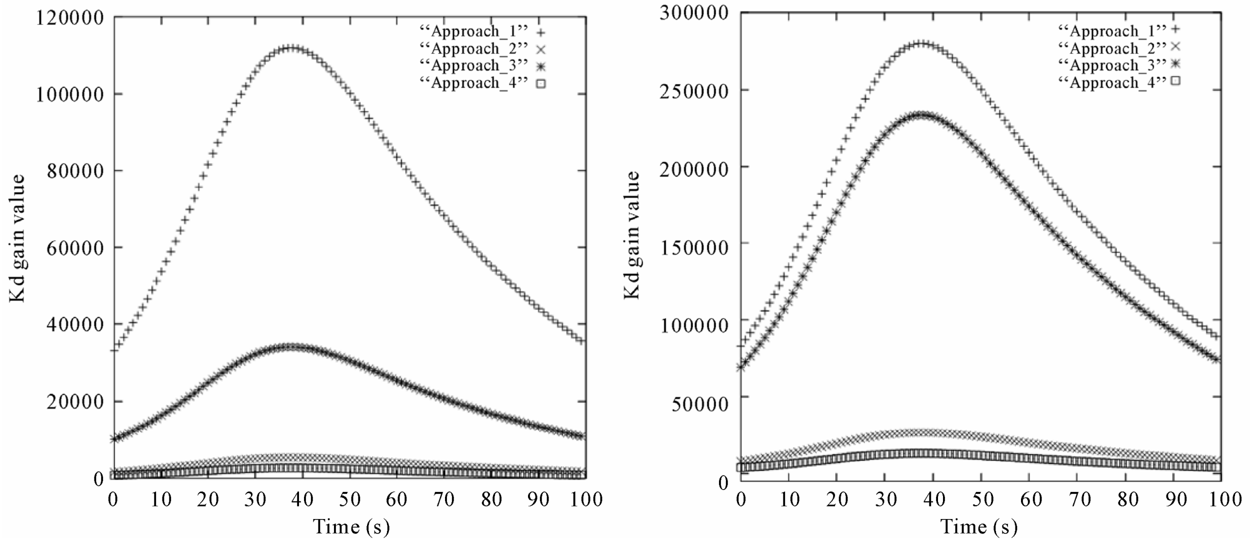
In this approach, the following PSO-parameters are found to yield the best results: number of particles interacting with each particle k = 3; dimension of the search space d = 6; number of runs =100; number of executions =1000. The optimized values of the parameters: t', ρ, ω_1, s_1, ω_2, and s_2 are found to be equal to 0.001m, 2600.00 kg/m^3, 13.715, 0.078, 14.271, and 0.00556, respectively. Results of this approach have been explained in detail with the help of **Figures 7** to **11**, at the end of this section.

4.1.4. Results of Approach 4

The following PSO-parameters are seen to give the best results: k = 3; d = 36; number of runs =100; number of executions = 1000. The optimal values of the parameters: t', ρ, ω_1, s_1, ω_2 and s_2 are found to be equal to 0.001 m, 2600.00 kg/m^3, 11.715, 0.078, 10.271 and 0.00551, respectively. **Table 2** shows the optimized values of other variables obtained by the PSO algorithm in approach 4. Results of this approach have been compared with that of other approaches in **Figures 7** to **11**.

4.1.5. Comparisons

The paths tracked and energy consumed by the manipulator using approaches 1, 2, 3 and 4 are compared here. The values of energy (kW) consumption, and average absolute percent deviation in trajectory tracking by four approaches are found to be equal to 0.241371, 0.241371, 0.238906, 0.23859, and 0.000000763, 0.000000009, 0.000000585, 0.00000000128, respectively, while trac-



(a) (b)

Figure 7. Variations of PD controller's gain values: (a) K_d; (b) K_p for Joint 1.

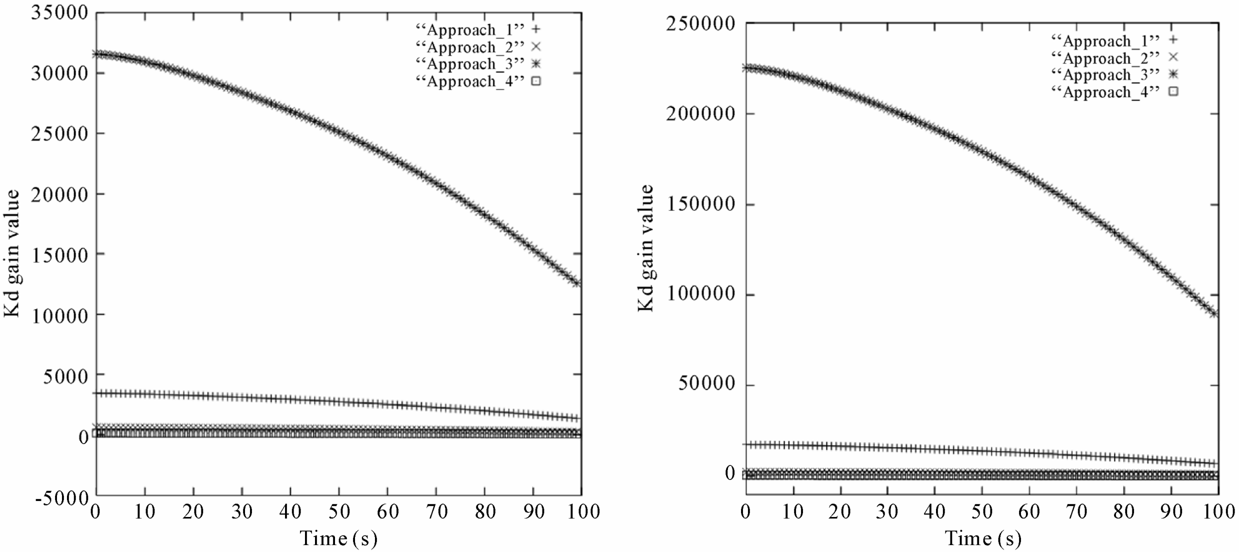
Table 1. Optimized values of the variables obtained by the GA in approach 2 for straight path tracking.

Variable	Range	Optimized value	Variable	Range	Optimized value
t'	0.001, 0.03	0.001	*W*23	−1.0, 1.0	−1.00
ρ	2600.0, 2800.0	2600.0	*W*24	−1.0, 1.0	−0.99
*V*11	−1.0, 1.0	−1.00	*W*31	−1.0, 1.0	−0.99
*V*12	−1.0, 1.0	−1.00	*W*32	−1.0, 1.0	−1.00
*V*13	−1.0, 1.0	−1.00	*W*33	−1.0, 1.0	−0.99
*V*14	−1.0, 1.0	−0.99	*W*34	−1.0, 1.0	0.89
*V*21	−1.0, 1.0	−1.00	*W*41	−1.0, 1.0	−0.99
V22	−1.0, 1.0	−1.00	*W*42	−1.0, 1.0	−0.99
*V*23	−1.0, 1.0	−1.00	*W*43	−1.0, 1.0	−0.99
*V*24	−1.0, 1.0	−1.00	*W*44	−1.0, 1.0	−0.99
*W*11	−1.0, 1.0	−0.99	a_1*_hid*	0.1, 1.0	0.099
*W*12	−1.0, 1.0	−0.99	a_2*_out*	2.0, 3.0	1.999999
*W*13	−1.0, 1.0	−0.99	a_3*_out*	1.0, 2.0	0.99999
*W*14	−1.0, 1.0	−1.00	a_4*_out*	1.0, 2.0	1.000
*W*21	−1.0, 1.0	−0.99	a_5*_out*	1.0, 2.0	0.999999
*W*22	−1.0, 1.0	−0.99	*b*	0.000001 to 0.0001	0.00000023

ing the straight path. **Figures 7** and **8** show the variations of K_d, K_p values for joint 1 and joint 2, respectively, while tracing the given straight trajectory. Comparisons of these approaches are shown on the entire trajectory in **Figure 9(a)**, whereas **Figure 9(b)** displays the same comparison on a finer scale within the smaller ranges of X and Y. **Figures 10** and **11** display the variations of percent deviation in predictions of X and Y values in a cycle, respectively, for the four approaches. The deviations in tracking the trajectory are found to be less in approaches 2 and 4 compared to that in approaches 1 and 3. Approach 4 is found to perform better than other approaches in terms of accuracy in path tracking and power consumption. The better performance of approach 4

Table 2. Optimized values of the variables obtained by the PSO in approach 4 for straight path tracking.

Variable	Range	Optimized value	Variable	Range	Optimized value
t'	0.001, 0.03	0.001	*W*23	−1.0, 1.0	1.00
ρ	2600.0, 2800.0	2600.0	*W*24	−1.0, 1.0	−0.7125
*V*11	−1.0, 1.0	−0.38296	*W*31	−1.0, 1.0	−0.9599
*V*12	−1.0, 1.0	0.1	*W*32	−1.0, 1.0	0.911465
*V*13	−1.0, 1.0	0.1	*W*33	−1.0, 1.0	0.62298
*V*14	−1.0, 1.0	1.00	*W*34	−1.0, 1.0	−0.67631
*V*21	−1.0, 1.0	1.00	*W*41	−1.0, 1.0	−0.049
*V*22	−1.0, 1.0	0.622	*W*42	−1.0, 1.0	0.955
*V*23	−1.0, 1.0	0.761	*W*43	−1.0, 1.0	−0.84
*V*24	−1.0, 1.0	0.761	*W*44	−1.0, 1.0	0.8364
*W*11	−1.0, 1.0	−0.84	a_1_hid	0.1, 1.0	0.2437
*W*12	−1.0, 1.0	−0.44	a_2_out	2.0, 3.0	2.1013
*W*13	−1.0, 1.0	−0.85	a_3_out	1.0, 2.0	2.00
*W*14	−1.0, 1.0	−0.35	a_4_out	1.0, 2.0	2.00
*W*21	−1.0,1.0	−0.71	a_5_out	1.0, 2.0	1.744
*W*22	−1.0, 1.0	−0.6924	*b*	0.000001 to 0.0001	0.000023



(a)

(b)

Figure 8. Variations of PD controller's gain values: (a) K_d; (b) K_p for Joint 2.

could be due to the use of a PSO algorithm in place of a GA. The latter is a potential tool for global search only but its local search capability is poor, whereas the former carries out both the global and local searches simultaneously. Moreover, PSO is a greedier algorithm compared to the GA.

4.2. Circular Path Tracking

The end-effector of the manipulator will have to trace a circular path in Cartesian coordinate system. The corresponding joint angle values are determined using Equation (2). A second-order polynomial is found to be suit-

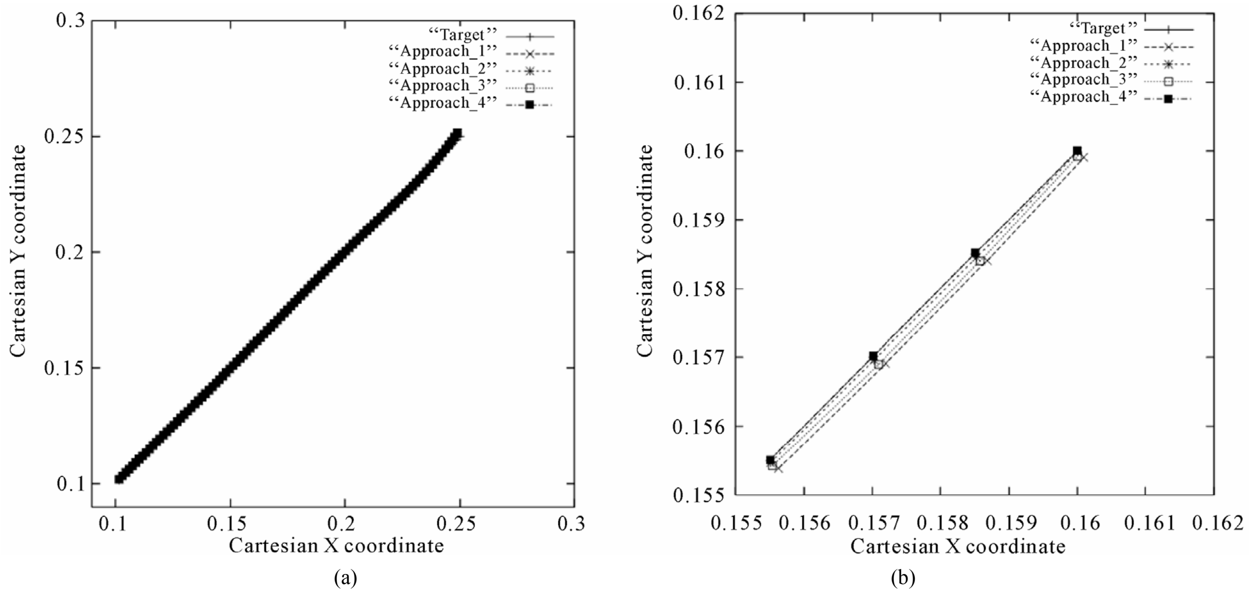


Figure 9. Comparisons of the paths tracked by the robot using approaches 1, 2, 3 and 4: (a) entire trajectory; (b) a part of the trajectory shown on the finer scales.

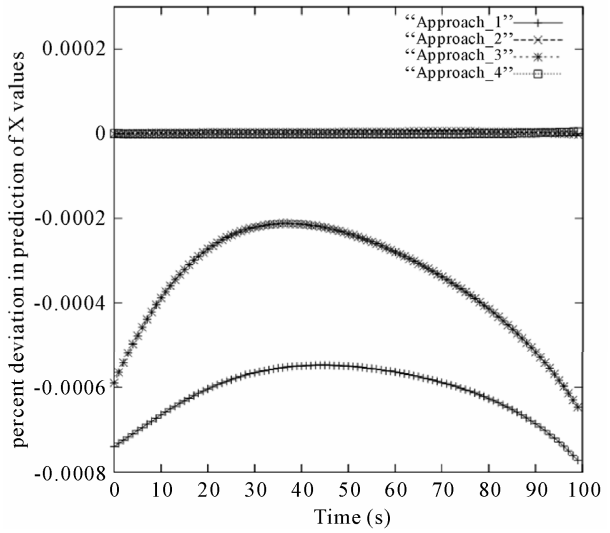


Figure 10. Variations of percent deviation in prediction of *X* values in a cycle for four approaches.

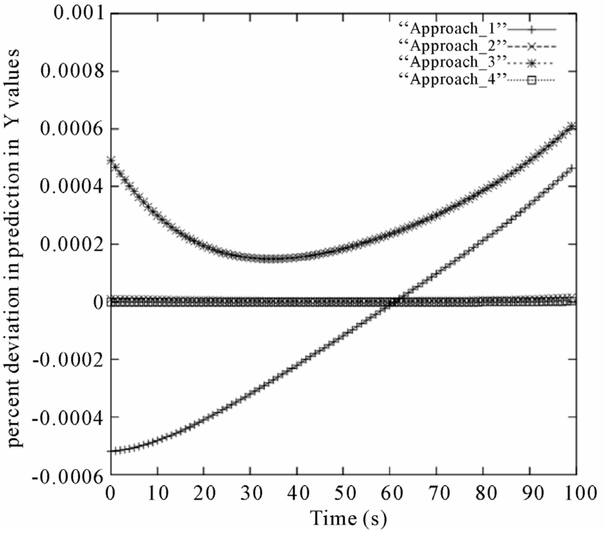


Figure 11. Variations of percent deviation in prediction of *Y* values in a cycle for four approaches.

able to represent the trajectory of $\theta_1(t)$ as $\theta_1(t) = a_{10}+a_{11}t + a_{12}t^2$, where $a_{10} = -0.39713758259$, $a_{11} = 0.01657138221$, $a_{12} = -0.0000095$. Similarly, the trajectory function of $\theta_2(t)$ has been expressed using another second-order polynomial as $\theta_2(t) = a_{20}+a_{21}t + a_{22}t^2$, where $a_{20} = 0.92207514343$, $a_{21} = 0.00000008916$, $a_{22} = -0.00000000085$.

4.2.1. Results of Approach 1

The following GA-parameters (obtained through a careful parametric study as explained earlier) are found to yield the best results: $P_c = 0.96$, $P_m = 0.0055$, Population size $P = 180$, Maximum number of generations $G_{\max} = 240$. The optimized values of the parameters: t', ρ, ω_1, s_1, ω_2 and s_2 are seen to be equal to 0.001m, 2600.00 kg/m^3, 7.017, −0.02, 10.00, and −0.0116, respectively.

4.2.2. Results of Approach 2

The best results have been obtained with the following GA-parameters: $P_c = 0.96$, $P_m = 0.0055$, Population size $P = 150$, Maximum number of generations $G_{\max} = 240$. The optimized values of the parameters: t', ρ,

ω_1, s_1, ω_2 and s_2 are seen to be equal to 0.001 m, 2600.00 kg/m^3, 7.017, −0.02, 10.00, and −0.0116, respectively. The optimized values of the variables obtained by the GA are shown in **Table 3**.

4.2.3. Results of Approach 3

The following PSO-parameters (obtained through a careful parametric study as explained earlier) are found to yield the best results: k = 3; d = 6; number of runs = 100; number of executions = 1000. The optimized values of the parameters: t', ρ, ω_1, s_1, ω_2 and s_2 are seen to be equal to 0.001 m, 2600.00 kg/m^3, 20.00, −0032556, 10.00 and 0.004085, respectively.

4.2.4. Results of Approach 4

The following PSO-parameters (obtained through a systematic study as explained earlier) are found to yield the best results: k = 3; d = 36; number of runs = 100; number of executions = 7000. The optimized values of the variables obtained by the PSO algorithm are shown in **Table 4**.

4.2.5. Comparisons

Results of the above four approaches have been compared here. The variations of K_d and K_p values for the joints: 1 and 2 are displayed in **Figures 12** and **13**, respectively. The manipulator is found to consume 0.864591, 0.771185, 0.755222, 0.752206 kW of energy using approaches 1, 2, 3 and 4, respectively. **Figure 14(a)** shows the circular path tracked by the manipulator using the above four approaches. In order to clearly distinguish the path traced by the robot, a segment of this path has been displayed in **Figure 14(b)** The values of percent deviation in prediction of X- and Y-values have been determined as obtained by the said four approaches and these are shown in **Figures 15** and **16**, respectively. Approaches 1, 2, 3 and 4 have yielded the values of average absolute percent deviation in prediction of the path as 0.000001103, 0.00000032, 0.00000038, and 0.00000023, respectively. Once again, approach 4 has proved its supremacy over other approaches in terms of accuracy in path tracking and energy consumption. It has happened so, due to the reasons explained above.

4.3. Comparisons with Others' Studies

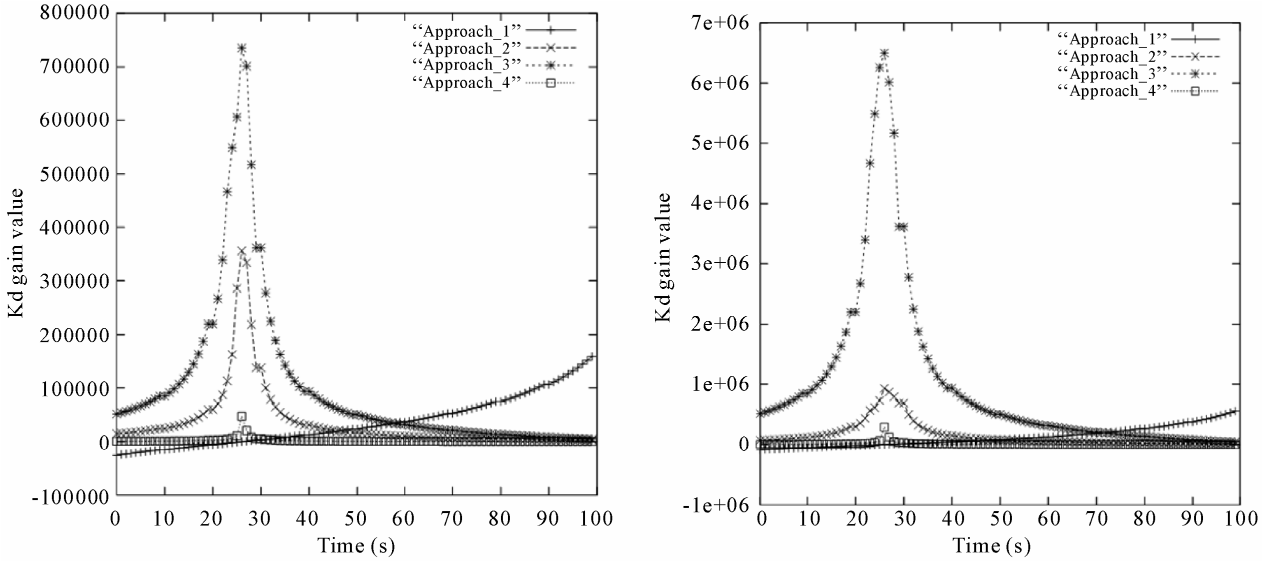
Path tracking problems of a 2-dof manipulator had been solved by various researchers. In this connection, the studies of Qu [1], Homsup and Anderson [2], Kelly and Salgado [3] and Ouyang and Zhang [4] are worth mentioning. Soft computing-based approaches had also been developed for the said purpose [7,9,10]. This study also deals with trajectory tracking problems of a manipulator. In the present paper, an integrated scheme has been de-

Table 3. Optimized values of the variables obtained by the GA in approach 2 for circular path tracking.

Variable	Range	Optimized value	Variable	Range	Optimized value
t'	0.001, 0.03	0.001	$W23$	−1.0, 1.0	−1.00
ρ	2600.0, 2800.0	2600.00	$W24$	−1.0, 1.0	−1.00
$V11$	−1.0, 1.0	−0.906785	$W31$	−1.0, 1.0	−1.00
$V12$	−1.0, 1.0	−0.861631	$W32$	−1.0, 1.0	−1.00
$V13$	−1.0, 1.0	−1.00	$W33$	−1.0, 1.0	−1.00
$V14$	−1.0, 1.0	−1.00	$W34$	−1.0, 1.0	−1.00
$V21$	−1.0, 1.0	−1.00	$W41$	−1.0, 1.0	−1.00
$V22$	−1.0, 1.0	−1.00	$W42$	−1.0, 1.0	−1.00
$V23$	−1.0, 1.0	−1.00	$W43$	−1.0,1.0	−1.00
$V24$	−1.0, 1.0	−1.00	$W44$	−1.0, 1.0	−1.0
$W11$	−1.0,1.0	−1.00	a_1_hid	0.1, 1.0	0.1
$W12$	−1.0, 1.0	−1.00	a_2_out	2.0, 3.0	2.00
$W13$	−1.0,1.0	−1.00	a_3_out	1.0, 2.0	0.999999
$W14$	−1.0, 1.0	−1.00	a_4_out	1.0, 2.0	1.00
$W21$	−1.0, 1.0	−1.00	a_5_out	1.0, 2.0	0.999999
$W22$	−1.0, 1.0	−1.00	b	0.000001 to 0.0001	0.000003

Table 4. Optimized values of the variables obtained by the PSO in approach 4 for circular path tracking.

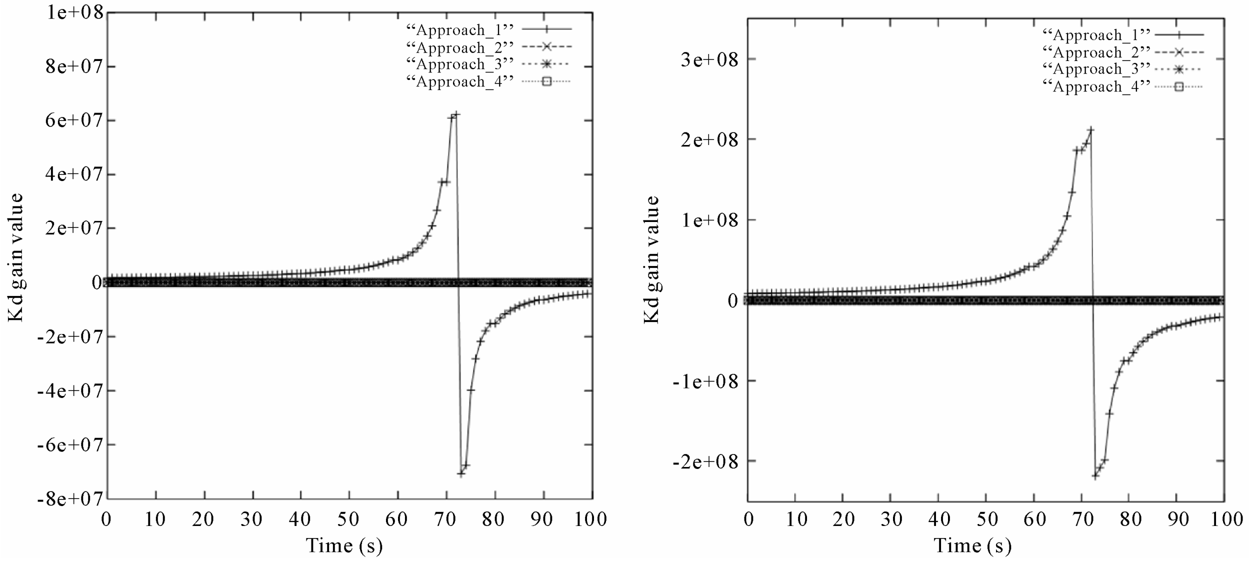
Variable	Range	Optimized value	Variable	Range	Optimized value
t'	0.001, 0.03	0.001	*W*23	−1.0, 1.0	−0.99
ρ	2600.0, 2800.0	2600.00	*W*24	−1.0, 1.0	−0.99
*V*11	−1.0, 1.0	−0.999	*W*31	−1.0, 1.0	−1.00
*V*12	−1.0, 1.0	−0.98	*W*32	−1.0, 1.0	−1.00
*V*13	−1.0, 1.0	−0.99	*W*33	−1.0, 1.0	−1.00
*V*14	−1.0, 1.0	−0.99	*W*34	−1.0, 1.0	−0.99
*V*21	−1.0, 1.0	−0.99	*W*41	−1.0, 1.0	−0.99
*V*22	−1.0, 1.0	−0.99	*W*42	−1.0, 1.0	−0.99
*V*23	−1.0, 1.0	−1.00	*W*43	−1.0,1.0	−1.00
*V*24	−1.0, 1.0	−0.99	*W*44	−1.0, 1.0	−1.00
*W*11	−1.0,1.0	−0.99	a_1*_hid*	0.1, 1.0	0.10
*W*12	−1.0, 1.0	−0.99	a_2*_out*	2.0, 3.0	2.00
*W*13	−1.0, 1.0	−1.0	a_3*_out*	1.0, 2.0	1.00
*W*14	−1.0, 1.0	−.0.99	a_4*_out*	1.0, 2.0	1.00
*W*21	−1.0, 1.0	−0.99	a_5*_out*	1.0, 2.0	1.00
*W*22	−1.0, 1.0	−0.99	*b*	0.000001 to 0.0001	0.000 0001691



(a) (b)

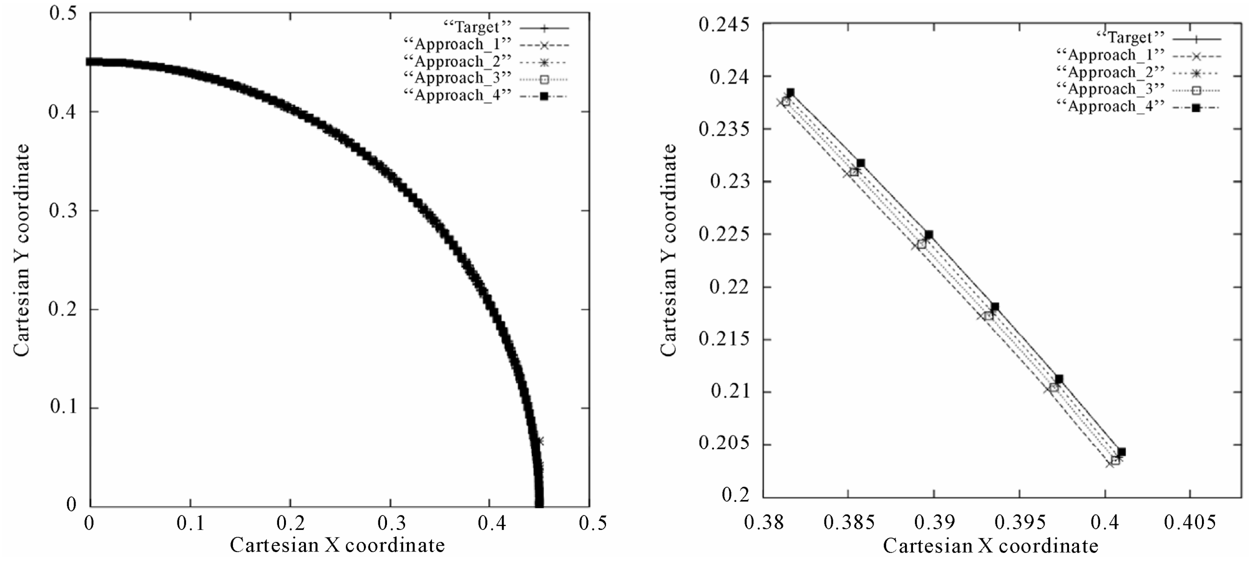
Figure 12. Variations of PD controller's gain values: (a) K_d; (b) K_p for Joint 1.

veloped to obtain optimal mechanical structure and PD controllers simultaneously for a 2-dof manipulator, so that it can track the trajectories accurately after consuming the minimum power. To the best of the authors' knowledge, this study is unique for a rigid link manipulator, although an attempt (not exactly the same) was made by Park and Asada [11] for a non-rigid link manipulator. It had been reported in Abe *et al.* [18], Braik *et al.* [17], Chen *et al.* [15] that computational cost of the PSO is less compared to that of a GA. The performance of PSO algorithm has been compared with that of a GA, in the present study.



(a) (b)

Figure 13. Variations of PD controller's gain values: (a) K_d; (b) K_p for Joint 2.



(a) (b)

Figure 14. Comparisons of the circular paths tracked by the robot in approaches 1, 2, 3 and 4: (a) entire trajectory; (b) a segment of the trajectory shown in finer scales.

5. Conclusions

An integrated scheme for obtaining optimal mechanical structures and adaptive PD controller for a 2-dof manipulator has been developed and its performance has been tested through computer simulations on two trajectory tracking problems. The robot studied in this paper is a simple one. However, the main strength of this study lies with the design and development of the above integrated scheme. The robot should be able to track the trajectory accurately, after consuming the minimum power and ensuring no mechanical failure of the same. Four approaches have been developed. In approaches 1 and 3, natural frequency ω and stability locus point s have been kept constant throughout the cycle, whereas these values are selected adaptively in the cycle in approaches 2 and 4. Approach 4 has outperformed other three approaches in terms of both power consumption as well as accuracy in trajectory tracking due to the reasons explained earlier. Moreover, approach 4 has yielded a more stable system compared to other approaches. The better performance of the PSO algorithm than that of the GA

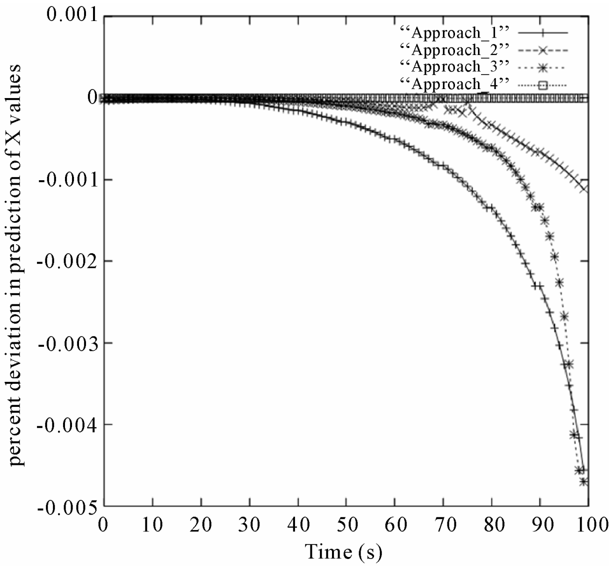


Figure 15. Variations of percent deviation in prediction of *X* values in a cycle for four approaches.

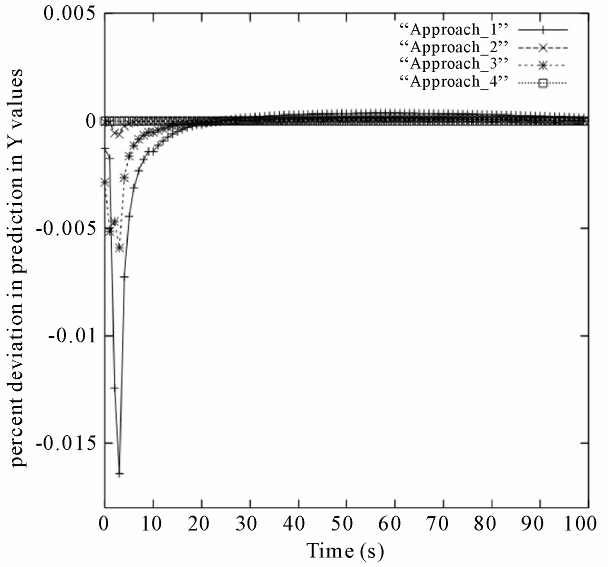


Figure 16. Variations of percent deviation in prediction of *Y* values in a cycle for four approaches.

could be due to its inherent ability to carry out the global and local searches simultaneously. On the other hand, the GA is a potential tool for global search, although it may not be so much powerful in local search.

6. Scope for Future Work

In the present study, the performances of developed approaches have been tested through computer simulations. However, it will be more interesting to test their performances in real-experiments. An improved version of PSO algorithm [37,38] may also be used in future to solve the said problem. The authors are working on these issues.

7. Acknowledgements

Rega Rajendra thanks financial support of the All India Council of Technical Education (AICTE), New Delhi, India, under the Quality Improvement Programme (QIP) Scheme, to carry out this study.

8. References

[1] Z. Qu, "Global Stability of Trajectory Tracking of Robot under PD Control," *Dynamics and Control*, Vol. 4, No. 1, 1994, pp. 59-71. doi:10.1007/BF02115739

[2] W. Homsup and J. N. Anderson, "PD Control Performance of Robotic Mechanisms," *Proceedings of American Control Conference*, Minneapolis, 10-12 June 1987, pp. 472-475.

[3] R. Kelly and R. Salgado, "PD Control with Computed Feed-Forward of Robot Manipulators: A Design Procedure," *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, 1994, pp. 566-571. doi:10.1109/70.313108

[4] P. R. Ouyang and W. J. Zhang, "A Novel Evolutionary PD Control and Application for Trajectory Tracking," *Proceedings of 7th International Conference on Control*, *Automation*, *Robotics and Vision*, Singapore, 2-5 December 2002, pp. 1337-1342.

[5] T. Ravichandran, D. Wang and G. Heppler, "Simultaneous Plant-Control Design Optimization of a Two-Link Planar Manipulator," *Mechatronics*, Vol. 16, No. 3-4, 2006, pp. 233-242. doi:10.1016/j.mechatronics.2005.09.008

[6] D. K. Pratihar, "*Soft Computing*," Narosa Publications, New Delhi, 2008

[7] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma and Y. Uchikawa, "Trajectory Control of Robotic Manipulators Using Neural Networks," *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 3, 1991, pp. 195-202. doi:10.1109/41.87587

[8] J. J. Craig, "Adaptive Control of Mechanical Manipulators," Addison-Wesley, Reading, 1988.

[9] M. B. Ghalia and A. T. Alouani, "A Robust Trajectory Tracking Control of Industrial Robot Manipulators Using Fuzzy Logic," *Proceedings of 27th South Eastern Symposium on System Theory* (*SSST*'95), Mississippi, 12-14 March 1995, pp. 268-271.

[10] A. Rueda and W. Pedrycz, "A Hierarchical Fuzzy-Neural-PD Controller for Robot Manipulators," *Proceedings of IEEE Conference on Computational Intelligence*, Orlando, 26-29 June 1994, pp. 673-677.

[11] J. H. Park and H. Asada, "Concurrent Design Optimization of Mechanical Structure and Control for High Speed Robots," *Journal of Dynamic Systems*, *Measurements*, *and Control*, Vol. 116, No. 3, 1994, pp. 344-356. doi:10.1115/1.2899229

[12] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Perth, 27 November-1 December 1995, pp. 1942-1948. doi:10.1109/ICNN.1995.488968

[13] Y. Shi and R. C. Eberhart, "A Modified Particle Swarm Optimizer," *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway, 1998, pp. 69-73.

[14] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability and Convergence in a Multi-Dimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 58-78. doi:10.1109/4235.985692

[15] Q. Chen, G. Guo and C. Li, "An Improved PSO Algorithm to Optimize BP Neural Network," *Proceedings of 5th International Conference on Natural Computation*, Tianjian, 14-16 August 2009, pp. 357-360.

[16] M. Han and L. Jiang, "Endpoint Prediction Model of Basic Oxygen Furnace Steelmaking Based on PSO-ICA and RBF Neural Network," *Proceedings of International Conference on Intelligent Control and Information Processing*, Dalian, 13-15 August 2010, pp. 388-393. doi:10.1109/ICICIP.2010.5565236

[17] M. Braik, A. Sheta and A. Arieqat, "A Comparison between GAs and PSO in Training ANN to Model the TE Chemical Process Reactor," *Proceedings of the AISB Symposium on Swarm Intelligence Algorithms and Applications*, Aberdeen, 1-4 April 2008, pp. 25-31.

[18] A. Abe and K. Komuro, "Trajectory Planning for Saving Energy of a Flexible Manipulator Using Soft Computing Methods," *Proceedings of International Conference on Control*, *Automation and Systems*, Gyeonggi-Do, 27-30 October 2010, pp. 1462-1467.

[19] K. S. Fu, R. C. Gonzalez and C. S. G. Lee, "Robotics: Control, Sensing, Vision, and Intelligence," McGraw-Hill Inc., Boston, 1987.

[20] J. Nishii, K. Ogawa and R. Suzuki, "The Optimal Gait Pattern in Hexapods Based on Energetic Efficiency," *Proceedings 3rd International Symposium on Artificial Life and Robotics*, Oita, 19-21 January 1998, pp. 106-109.

[21] F. P. Beer, R. E. Johnston Jr. and J. T. Dewolf, "Strength of Materials," Tata-McGraw-Hill Publishing Company Limited, New Delhi, 2004.

[22] J. R. Davis, "ASM Specialty Handbook Al and Al Alloys," ASM International Materials Part, Ohio, 1993.

[23] E. G. Dieter, "ASM Handbook, Material Selection and Design," Vol. 20, ASM International Materials Park, Ohio, 1997.

[24] W. F. Gale and T. C. Totemeier, "Smithhells Metal Reference," 8th Edition, Butterworth-Heinemann, Waltham, 2004.

[25] E. T. George and S. Mackenzie, "Handbook of Aluminum. Vol. 1. Physical Metallurgy and Processes," Marcel Dekker Inc., New York, 2003.

[26] S. L. Semiatin, "ASM Handbook Forging and Forming," Vol. 14, ASM International Materials Park, Ohio, 1998.

[27] J. E. Temple, "Handbook of Structural Design in Al Alloys," Temple Press, Brooks, 1953.

[28] C. Wu, 2010. www.efunda.com

[29] J. J. Craig, "Introduction to Robotics: Mechanics and Control," Pearson Education, Upper Saddle River, 2006.

[30] M. K. Spong, S. Hutchinson and M. Vidyasagar, "Robot Modeling and Control," John Wiley& Sons, Inc., New York, 2006.

[31] M. Akar and I. Temiz, "Motion Controller Design for the Speed Control of DC Servomotor," *International Journal of Applied Mathematics and Informatics*, Vol. 4, No. 1, 2007, pp. 131-137.

[32] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems*, Vol. 9, No. 2, 1995, pp. 115-148.

[33] J. H. Holland, "Adaptation in Natural and Artificial Systems," The University of Michigan Press, Ann Arbor, 1975.

[34] D. E. Goldberg, "Genetic Algorithm in Search Optimization, and Machine Learning," Addison-Wesley, Reading, 1989.

[35] R. C. Eberhert, P. Simpson and R. Dobbins, "Computational Intelligence PC Tools: Dalian," Chapter 6, Academic Press, San Diego, 1996, pp. 212-226.

[36] Y. Shi and R. C. Eberhart, "Empirical Study of Particle Swarm Optimization," *Proceedings of IEEE International Congress on Evolutionary Computation*, Washington DC, 6-9 July 1999, pp. 101-106.

[37] J. J. Liang and P. N. Suganthan, "Defining a Standard for Particle Swarm Optimization," *Proceedings of Swarm Intelligence Symposium*, Pasadena, 8-10 June 2005, pp. 124-129. doi:10.1109/SIS.2005.1501611

[38] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, 2006, pp. 281-295. doi:10.1109/TEVC.2005.857610

List of Symbols and Abbreviated Terms

$a_{10},\cdots,a_{14}$	Coefficients of fourth-order polynomial
$a_{20},\cdots,a_{32}$	Coefficients of cubic polynomial
$a_{1\_}hid$	Coefficient of transfer function of hidden neurons
$a_{2\_}out$	Coefficient of transfer function for first output neuron
$a_{3\_}out$	Coefficient of transfer function for second output neuron
$a_{4\_}out$	Coefficient of transfer function for third output neuron
$a_{5\_}out$	Coefficient of transfer function for fourth output neuron
b	Bias value
b_1,b_2	Acceleration coefficients
B	Effective damping value
c_1,c_2	Gravity terms of torque
d	Dimension of search space
D	Inertia term of torque
$D(s)$	Disturbance on the system
E'	Energy, kW
E	Elastic Modulus, N/m^2
Err_{total}	Total error
Err_{avg}	Average error
f	Fitness
g	Acceleration due to gravity, m/s^2
$Gbest$	Swarm global best solution
$G_{\max}$	Maximum number of generations
h	Dynamic Coriolis term
I	Moment of inertia, kg.m^2
k	Number of particles interacting with each particle
$jterr$	Joint error
J	Inertia of links, motor and gear, kg·m^2
K_d	Derivative gain
K_p	Proportional gain
L_1,L_2	Length of the links, m
m	Mass of the link, kg
M_1	Bending moment of first link
M	Moment, N-m
P_c	Probability of crossover
P_i	i^{th} Particle best fitness
P_g	Swarm's global best fitness
P_m	Probability of mutation
P	Population size
P'	Penalty term
$Pbest$	Particle best solution
R	Outer radius of hollow circular link
r	Inner radius of hollow circular link
s	Stability locus point
t	Time, s
t'	Thickness, m
v_{id}	Particle's updated velocity in *d*th dimension
x_{id}	Particle's updated position in *d*th dimension
w	Uniformly distributed load, N/m
$U(s)$	Control input
V_i	Velocity Vector
$[V]$	Connecting weights between input and hidden layers
$[W]$	Connecting weights between hidden and output layers
W	Inertia weight
W_1	Concentrated load acting on first link, N
W_2	Concentrated load acting on second link, N
W_{m2}	Weight of second motor
X_i	Position Vector
X_i,Y_i	Coordinates of initial position
X_f,Y_f ,	Coordinates of final position
ζ	Damping ratio
$\Theta(s)$	System response in Laplace Transform
θ	Joint angle, rad
$\theta(s)$	Angle in Laplace domain, rad
ρ	Density, kg/m^3
σ_y	Yeild Stress, N/m^2
τ	Torque, N-m
ω	Natural frequency
$\Omega(s)$	Closed-loop characteristic polynomial
GA	Genetic Algorithm
PD	Proportional Derivative
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
SBX	Simulated binary crossover
Rand( )	Random number generator in the range of (0,1)