Archives of Control Sciences
Volume 21(LVII), 2011
No. 3, *pages 269–285*

Evolutionary computation approaches to tip position controller design for a two-link flexible manipulator

BIDYADHAR SUBUDHI, SUBHAKANTA RANASINGH and AJAHA KUMAR SWAIN

Controlling multi-link flexible robots is very difficult compared rigid ones due to inter-link coupling, nonlinear dynamics, distributed link flexure and under-actuation. Hence, while designing controllers for such systems the controllers should be equipped with optimal gain parameters. Evolutionary Computing (EC) approaches such as Genetic Algorithm (GA), Bacteria Foraging Optimization (BFO) are popular in achieving global parameter optimizations. In this paper we exploit these EC techniques in achieving optimal PD controller for controlling the tip position of a two-link flexible robot. Performance analysis of the EC tuned PD controllers applied to a two-link flexible robot system has been discussed with number of simulation results.

Key words: flexible manipulator, genetic algorithm, bacteria foraging, fitness function

1. Introduction

Flexible robot manipulators, unlike industrial robots, are utilized for specific purposes like in a space shuttle arm. These flexible robots exhibit superior performance with respect to rigid robot in terms of an increased payload carrying capacity, lesser energy consumption, cheaper construction, faster movements, and longer reach. However, link flexibility causes significant control problems owing to distributed nature of flexure giving rise to complex dynamics. The dynamics of flexible manipulator being a distributed parameter system, it is highly non-linear in nature. Control algorithms will be required to compensate for both the vibrations and static deflections that result from the flexibility. This provides a challenge to design control techniques that gives precise control of desired parameters of the system in desired time, ability to cope up with sudden changes in the bounded system parameters, and robust performance [1].

The ultimate goal of such robotic designs is to accurate tip position control in spite of the flexibility in a reasonable amount of time. Conventional control system design is generally a trial and error process which is often not capable of controlling a process, which varies significantly during operation. So, a more promising controller design is to be done in order to overcome these challenges.

The Authors are with Departmen of Electrical Engineering, National Institute of Technology, Rourkela-769008, India. Email: bidyadhar@nitrkl.ac.in

Received 16.07.2011. Revised 10.09.2011.

The proportional-derivative (PD) controllers have been widely used in industry for many years due to their simplicity of operation, robustness of performance. Unfortunately, it has been a problem to achieve optimal PD gains because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities. For a wide range of practical applications, the tuning approaches like trial and error method and Ziegler-Nichols method works quite well. However, the tuning process is too laborious and time consuming. Due to these reasons, it is highly desirable to find new approaches to the tuning of PD controllers. Many artificial intelligence (AI) techniques have been employed to improve the controller performances, such as neural network [2], fuzzy system [3], and neural-fuzzy logic [4] have been widely applied to proper tuning of PD controller parameters.

The advent of evolutionary computation (EC) has revolutionized the approach to optimal PD controller design. Some of the practical advantages of using evolutionary algorithms as compared with classic methods of optimization or artificial intelligence are reviewed in [5]. Specific advantages include the flexibility of the procedures, as well as the ability to self-adapt the search for optimum solutions on the fly. In this paper a new design method for PD controller based on Genetic Algorithm (GA) and Bacteria Foraging Optimization (BFO) scheme is presented. The controller is implement-ed on the flexible manipulator system in a loop and a certain fitness function is minimized over the loop thus ensuring better tip position control of the flexible manipulator. A common fitness function is chosen for both the processes and the simulation results are compared.

The paper is organized as follows: Section 2 gives the dynamics of a two-link flexible manipulator system. The PD controller design based on GA optimization technique is described in Section 3 and controller design based on BFO in Section 4. Section 5 reports simulation results for the system, using both GA and BFO based PD controller. Conclusions are drawn in the final section.

2. Dynamics of two-link flexible manipulator

Consider a planar 2-link flexible arm with rotary joints between the two flexible links whose first link is clamped at its base on the rotor of a motor and second link is loaded with a point mass at its tip as shown in Fig. 1. As the flexible manipulator is subjected to motion only in the horizontal plane, gravitational effects are not considered while modeling. The (X_0', Y_0') coordinate frame is considered as base frame and other coordinate frames are transformed into the base frame with help of different transformation matrices as presented in [6]. The rigid body moving frame associated to link i is (X_i, Y_i), and the flexible body moving frame associated to link i is (X_i', Y_i'). The rigid motion is described by the joint angles θ_i, while $y_i(x_i)$ denotes the transversal deflection of link i at abscissa x_i, $0 \leqslant x_i \leqslant l_i$, l_i being the link length.

The dynamic equations of motion of a planar 2-link flexible arm can be derived following the standard Lagrangian approach , i.e by computing the kinetic energy T and the potential energy U of the system and then forming the Lagrangian $L = T - U$. The

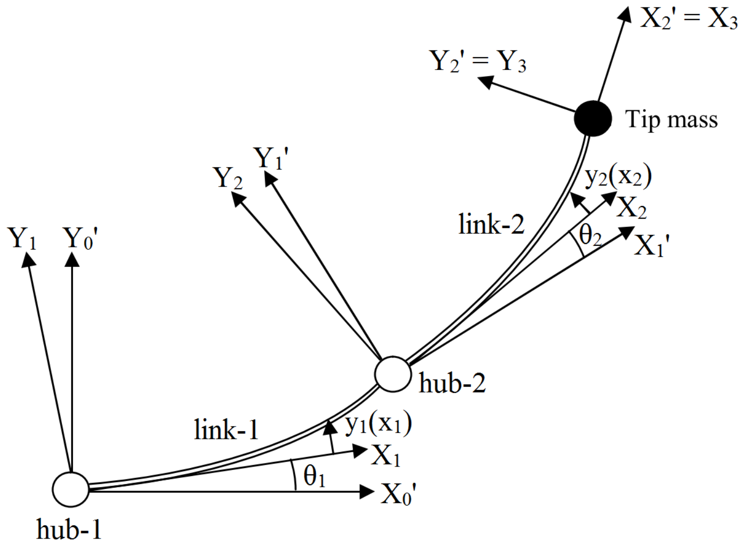


Figure 1. A two-link flexible manipulator.

total kinetic energy is given by the sum of the following contributions [6]:

$$T = \sum_{i=1}^{2} T_{h_i} + \sum_{i=1}^{2} T_{l_i} + T_p. \tag{1}$$

The kinetic energy of the rigid body located at hub i of mass M_{h_i} and moment of inertia J_{h_i} is

$$T_{h_i} = \frac{1}{2} M_{h_i} \dot{r}_i^T \dot{r}_i + \frac{1}{2} J_{h_i} \dot{\alpha}_i^2 \tag{2}$$

where, $\dot{r}_i$ is the absolute linear velocity of the end point of link i in frame (X_0', Y_0') and

$$\dot{\alpha}_i = \sum_{j=1}^{i} \dot{\theta}_j + \sum_{k=1}^{i-1} \dot{y}_{ke}'.$$

with $y_{ke}' = (\partial y_k / \partial x_k)|_{x_k = l_k}$ is the angular velocity of frame (X_i, Y_i). The kinetic energy pertaining to link i of linear density ρ_i is

$$T_{l_i} = \frac{1}{2} \int_0^{l_i} \rho_i(x_i) \dot{p}_i^T(x_i) \dot{p}_i(x_i) dx_i \tag{3}$$

where $\dot{p}_i(x_i)$ represents the absolute linear velocity of an arm point. The kinetic energy associated to a payload of mass M_p and moment of inertia J_p located at the end of the second link is

$$T_p = \frac{1}{2} M_p \dot{r}_3^T \dot{r}_3 + \frac{1}{2} J_p (\dot{\alpha}_2 + \dot{y}'_{2e})^2. \tag{4}$$

where, $\dot{r}_3$ is the absolute linear velocity of tip mass and $y'_{2e} = (\partial y_2/\partial x_2)|_{x_2=l_2}$ and the upper dot denotes time derivative.

In absence of gravity (horizontal plane motion), the potential energy is given by [6]

$$U = \sum_{i=1}^{2} U_i = \sum_{i=1}^{2} \frac{1}{2} \int_0^{l_i} (EI)_i(x_i) \left[\frac{d^2 y_i(x_i)}{dx_i^2} \right]^2 dx_i \tag{5}$$

where U_i is the elastic energy stored in link i, being $(EI)_i$ its flexural rigidity.

Links are modeled as Euler-Bernoulli beams of uniform density ρ_i, and constant flexural rigidity $(EI)_i$, with deformation $y_i(x_i,t)$ satisfying the partial differential equation,

$$(EI)_i \frac{\partial^4 y_i(x_i,t)}{\partial x_i^4} + \rho_i \frac{\partial^2 y_i(x_i,t)}{\partial t^2} = 0, \quad i = 1,2. \tag{6}$$

With proper boundary conditions [6], a finite-dimensional model (of order m_i) of link flexibility can be obtained by the assumed modes technique. Exploiting separability in time and space of solutions to (6), the link deflection can be expressed as

$$y_i(x_i,t) = \sum_{j=1}^{m_i} \phi_{ij}(x_i) \delta_{ij}(t) \tag{7}$$

where $\phi_{ij}(x_i)$ denotes the assumed mode shapes of link i for specific beam boundary conditions and $\delta_{ij}(t)$ are the time-varying variables associated with the assumed spatial mode shapes.

The dynamic model of the two-link flexible manipulator is obtained by satisfying the Lagrange-Euler equations

$$\frac{d}{d\tau} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i, \quad i = 1,2,\ldots \tag{8}$$

where $\{f_i\}$ are the generalized forces performing work on $\{q_i\}$.

As a result of this procedure, the equations of motion for a planar two-link flexible arm can be written in the familiar closed form

$$M(q)\ddot{q} + h(q,\dot{q}) + Kq = Qu \tag{9}$$

where $q = (\theta_1, \theta_2, \delta_{11}, \delta_{12}, \ldots, \delta_{1m}, \delta_{21}, \delta_{22}, \ldots, \delta_{2m})^T$ and u is the 2-vector of joint (actuator) torques. M is the positive- definite symmetric inertia matrix, h is the vector of coriolis and centrifugal forces, K is the stiffness matrix, and Q is the input weighting matrix due to the clamped link assumption.

A case of finite dimensional dynamic model with two assumed mode shapes for each link has been described in [6]. But due to the model complexity and simulation run-time a more simple dynamic model with one mode shape for each link has been considered for optimal algorithm simulation.

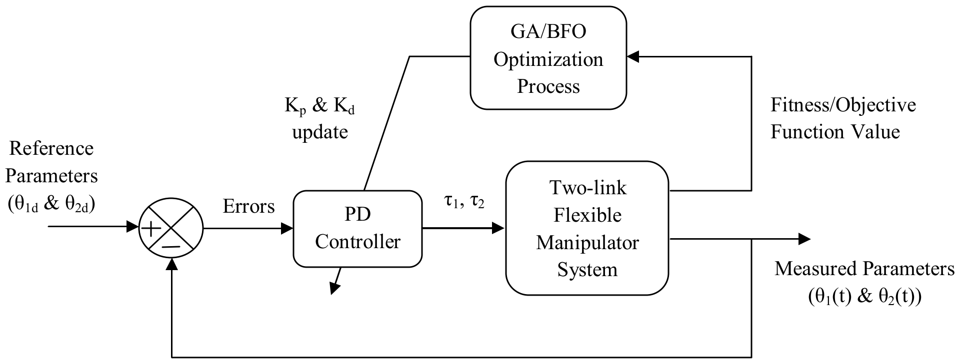


Figure 2. Evolutionary computation based controller design.

3. GA based PD controller design

Genetic algorithm (GA) is a search algorithm based on the observation that the reproduction and the principle of survival of the fittest, enables biological species to adapt to their environment and compete effectively for its resources. Control analysis for a two-link flexible manipulator includes tip position control and end effector path following control. In this paper tip position control analysis is considered for controller design. The objective of tip position control is to drive the tip of the flexible manipulator to a desired position as fast as possible with minimum vibration in the links. Let $P(x,y)$ is the desired position discussed above. Using inverse kinematics, the angles by which each link to is to driven to reach $P(x,y)$ is calculated as $(\theta_{1d}$ & $\theta_{2d})$. A simple PD controller is proposed for this propose and the controlled torque developed using the controller is given by,

$$\begin{aligned} \tau_1 &= K_{p1}[\theta_{1d} - \theta_1(t)] - K_{d1}\dot{\theta}_1(t) \\ \tau_2 &= K_{p2}[\theta_{2d} - \theta_2(t)] - K_{d2}\dot{\theta}_2(t) \end{aligned} \tag{10}$$

where τ_1 ans τ_2 are the input torques to link-1 and link-2 respectively, K_{p1} and K_{p2} are the proportional gains and K_{d1} and K_{d2} are the derivative gains for both links. $\theta_1(t)$ and $\theta_2(t)$ are the total angles made by the tip of link-1 and link-2 respectively which also includes the deflection factors.

The proportional gain in the PD controller helps in reducing rise time. It also decreases the steady state error of the system. Derivative gain is necessary for increasing stability of the system and reducing the overshoot, and improving the transient response. Positive values of K_p and K_d are chosen to ensure the stability of the closed loop system. Also different values of K_p and K_d are chosen for both links for better performance.

In our case PD gains used in the controller (10) is optimally selected using GA under certain suitably specified performance fitness functions to achieve the desired tip motion performance of the flexible link system. The advantages of GA optimization over the conventional optimization method are that it is global, data independent and robust [7]. Further GA can be directly applied to solve an optimization problem with a certain fitness function without reformulating the problem into a suitable form [8]. The GA optimization technique consists of three basic processes; randomly initialization of the parameters to be optimized from a suitable search space, evaluation of performance fitness function and application of GA operators. First a suitable search space is chosen based on the convergence of tip position error and limitation of the model parameters. Then the first generation population is randomly generated within the range of the search space. The parameters set (population), which are called chromosomes, are of decimal form. For second process a proper fitness function should be chosen such that minimizing it will lead us to optimal parameter. In the case of flexible manipulator control, where the main objective being the tip position error minimization, the Integral Time-multiplied Absolute value of Errors (ITAE) is chosen as a suitable fitness function because of its advantages over Integral of Squared Errors (ISE) in minimizing oscillations and overshoots [8]. The fitness function is given by:

$$ff = \text{ITAE} = \int_0^T t|P(x,y) - p(x,y)|dt \tag{11}$$

where, $P(x,y)$ is the desired tip position and $p(x,y)$ is the actual tip position. Generally the evaluation the time interval should be $[0, \infty]$. However, only the finite time interval $[0,T]$ will be considered for practicality and realizability. The smaller the values of the two fitness functions for the two links, the better tip performances we can achieve.

The above fitness function is evaluated for each chromosome passing through the system model and then for that generation convergence condition is checked. If the optimization convergence limit is reached, then the optimization process is terminated as shown in Fig. 3. Otherwise the next generation is obtained through the genetic operators. GA operators play an important role in GA optimization process and are described in detail in the following.

3.1. Selection

This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce. So out of N chromosomes best $N/2$ chromosomes are selected based on the performance fitness function values determined using (11) (the chromosomes which give minimum fitness function value are chosen).

3.2. Cross-over

Randomly two parents are selected from the best $N/2$ chromosomes to exchange genetic information with each other and generate two new individuals. The cross-over operator is repeated until $N/2$ new individuals are generated so that population size re-

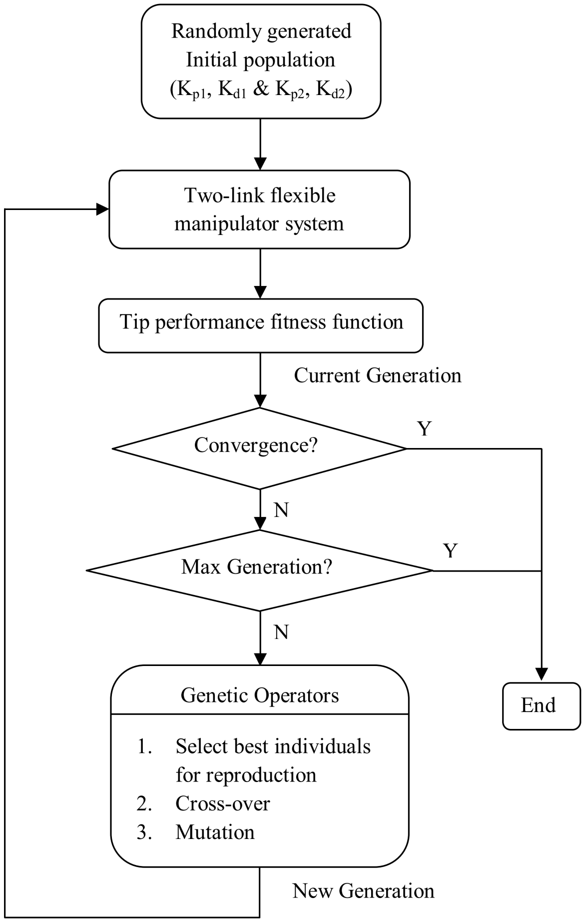


Figure 3. GA optimization procedure.

mains constant for next generation. Mathematically the cross-over operator is described as follows:

If the parents are (K_{p1}, K_{d1}) and (K_{p2}, K_{d2}), then [8]

$$\begin{aligned} K_{p1new} &= rK_{p1} + (1-r)K_{p2} \\ K_{d1new} &= rK_{d1} + (1-r)K_{d2} \\ K_{p2new} &= rK_{p2} + (1-r)K_{p1} \\ K_{d2new} &= rK_{d2} + (1-r)K_{d1} \end{aligned} \tag{12}$$

where $r \in (0,1)$ is a random number. New pairs of PD controller gains are formed using above equations (12) where cross-over operation is carried out taking r part from one parent and $(1-r)$ part from the other.

3.3. Mutation

Mutation occurs with a certain probability knows as mutation rate. Mutation operator introduces new genetic information replacing that of the selected individual. Mathematically the mutation operator is described as,

$$K_{new} = |K_{old} + 2(r_1 - 0.5)K_{max}| \tag{13}$$

where $r_1 \in (0,1)$ is a random number and the above mutation process is done for PD controller gain pairs (K_p, K_d).

The GA is repeated for certain generation and finally the optimal values of (K_p, K_d) pairs are obtained for the flexible manipulator system for which the fitness function value is the least in the last generation.

4. BFO based PD controller design

Bacteria Foraging Optimization (BFO) algorithm is a widely accepted global optimization technique inspired by the social foraging behavior of *Escherichia coli* [9]. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFO algorithm is mimicking chemotactic movement of virtual bacteria in the problem search space.

The design of PD controller (10) for both the links of two link flexible robot requires two pairs of PD gains (K_{p1}, K_{d1}) and (K_{p2}, K_{d2}) which are optimized using BFO technique. So the search space desired in the BFO algorithm is taken as four for the above four gains. In the bacterial foraging process, three motile behaviors are mimicked. They are described as follows.

4.1. Chemotaxis

In this process each bacteria cell moves through swimming and tumbling via flagella. It can swim for a period of time in the same direction or it may tumble and alternate between these two modes of operation for the entire lifetime. Suppose $B(i, j, k, l)$ represents i-th bacterium at j-th chemotactic, k-th reproductive and l-th elimination-dispersal step and $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may

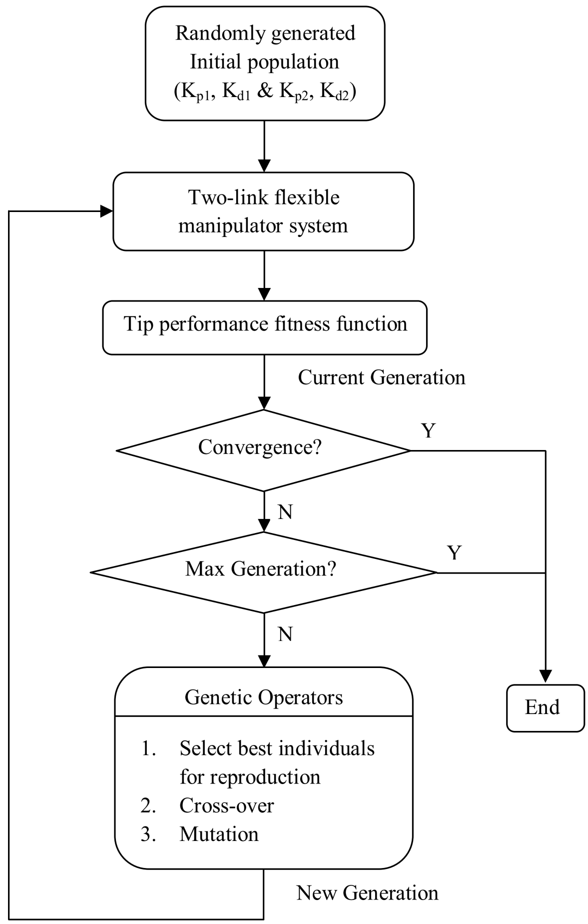


Figure 4. BFO algorithm.

be represented by [10]

$$B(i, j+1, k, l) = B(i, j, k, l) + \frac{C(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{14}$$

where Δ indicates a vector in the random direction whose elements lie in $[-1, 1]$. The bacterium $B_1(i, j, k, l)$, $B_2(i, j, k, l)$, $B_3(i, j, k, l)$ and $B_4(i, j, k, l)$ represent the K_{p1}, K_{d1}, K_{p2} and K_{d2} gains of the PD controller respectively. The objective function, $J(i, j, k, l)$, evaluated using the above four bacterium in each Chemotaxis process loop is defined as

$$J(i, j, k, l) = \int_0^T t\left|P(x, y) - p\left(x(i, j, k, l), y(i, j, k, l)\right)\right| dt \tag{15}$$

where $P(x,y)$ is the desired tip position and $p(x(i,j,k,l),y(i,j,k,l))$ is the actual tip position for each bacterium quadruple (PD controller gains) in the Chemotaxis process.

4.2. Reproduction

The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function (15)) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant. Mathematically this process can be represented as

$$B(i+Sr,k+1,l) = B(i,k+1,l) \tag{16}$$

where $B(i,k,l)$ represents the i-th bacterium at k-th reproductive and l-th elimination-dispersal step and Sr is the number of bacteria reproductions (splits) per generation.

4.3. Elimination and dispersal

Due to various reasons sudden changes in the local environment where a bacterium population lives may occur. In *elimination and dispersal* process all the bacteria in a region can be killed or a group can be dispersed into a new location. They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near lower objective function (15) region. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

The above processes are repeated in loops for certain number of generation. Each generation has the Chemotaxis process loop of Nc with bacteria swim length taken as Ns and the Reproduction process loop of Nre and the *elimination and dispersal* loop of Ned. And finally the PD gains in form of bacteria B are found for which the objective function (15) value is the least.

5. Results and discussion

In this section the simulation results of the dynamic model of the two-link flexible manipulator described in section II using PD controller is presented. The dynamic model is simulated for both the optimization algorithm and the optimized PD gains are obtained. Using the PD gains found from both the algorithm the link tip position is analyzed.

The two-link flexible manipulator with system parameters given in Tab. 1 is considered for simulation. A single mode of vibration for each link is considered for simplicity of model and less run time. For both optimization processes same fitness function (the function to be minimized to get optimum result) has been taken which is the integral

Table 5. Flexible manipulator parameters

ρ_1	0.2 kg/m	ρ_2	0.2 kg/m
l_1	0.5 m	l_2	0.5 m
m_1	0.1 kg	m_2	0.1 kg
J_{o_1}	0.0083 kg m^2	J_{o_2}	0.0083 kg m^2
J_{h_1}	0.1 kg m^2	J_{h_2}	0.1 kg m^2
M_{h_2}	1 kg	M_p	0.1 kg
$(EI)_1 = (EI)_2$	1 N m^2	J_p	0.0005 kg m^2

time-multiplied absolute value of errors (ITAE):

$$ff = \text{ITAE} = \int_0^T t|P(x,y) - p(x,y)|dt$$

where $P(x,y)$ is the desired tip position and $p(x,y)$ is the actual tip position. And the evaluation time interval is set to be $0 \leqslant t \leqslant 5$ s (i.e. $T = 5s$). $P(x,y)$ is mathematically represented as:

$$P(x,y) = \begin{bmatrix} xd \\ yd \end{bmatrix}$$

where

$$\begin{aligned} xd &= l_1 \cos(\theta_{1d}) + l_2 \cos(\theta_{1d} + \theta_{2d}) \\ yd &= l_1 \sin(\theta_{1d}) + l_2 \sin(\theta_{1d} + \theta_{2d}). \end{aligned} \tag{17}$$

In the simulation θ_{d1} and θ_{d2} (desired link angles) are both taken as 20 degree or 0.3491 rad and (xd, yd) denotes the desired tip position for the two-link flexible manipulator. And $p(x,y)$ vector that denotes the x and y coordinates of the tip of the manipulator is defined by

$$p(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}.$$

Using different transformation matrices used in deriving the dynamic model of the two-link flexible manipulator system defined in [6], the x and y coordinates of the tip of the manipulator are calculated as follows:

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) - y_{1e} \sin(\theta_1) - y_{2e} \sin(\theta_1 + \theta_2)$$

$$
\begin{aligned}
&- \quad y_{2e}y'_{1e}\cos(\theta_1+\theta_2) - l_2 y'_{1e}\sin(\theta_1+\theta_2) \\
y \quad &= \quad l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) + y_{1e}\cos(\theta_1) + y_{2e}\cos(\theta_1+\theta_2) \\
&- \quad y_{2e}y'_{1e}\sin(\theta_1+\theta_2) + l_2 y'_{1e}\cos(\theta_1+\theta_2).
\end{aligned} \tag{18}
$$

So, the absolute value of error is defined as

$$|P(x,y) - p(x,y)| = \sqrt{(xd-x)^2 + (yd-y)^2}. \tag{19}$$

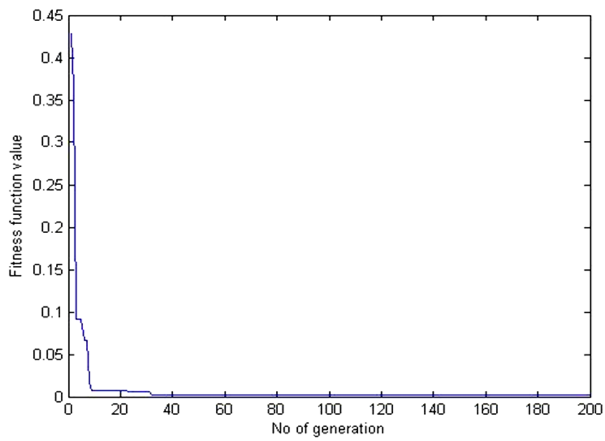


Figure 5. Fitness function values over 200 generations.

Fig. 5. shows the minimization of fitness function using GA optimization process over 200 generations.

The updating of PD controller gains through genetic operators (crossover (12), mutation (13)) are shown in Fig. 6(a), and Fig. 6(b).

In BFO process the PD gain pairs are chosen from different regions of search space and only those pairs are considered which are found in local minimas of objective function. And finally the PD gain pairs, which yield lowest value of objective function, are considered as optimal values. The optimal PD gain values thus obtained, compared to those obtained in GA process, is given in the Tab. 2.

The angle rotated by link-1 tip (hub-2) including the link vibration defined by:

$$ta_1 = \theta_1 + y_{1e}/l_1 \tag{20}$$

is shown in Fig. 7(a) with PD controller whose gains are optimized through both GA and BFO process. Similarly the link-2 tip (payload mass) revolution w.r.t. the inertial coordinate frame at hub-2 defined by:

$$ta_2 = \theta_2 + y_{2e}/l_2 \tag{21}$$

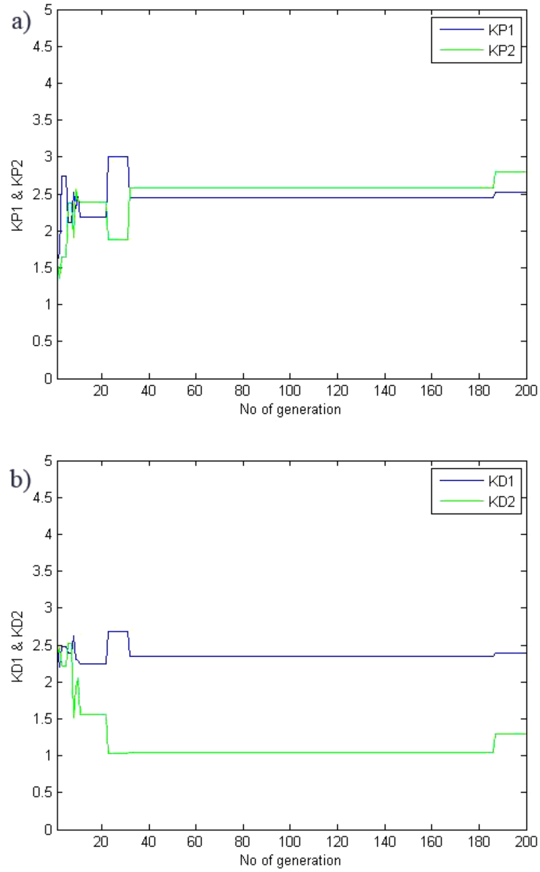


Figure 6. a) K_p update and b) K_d update for both the links.

Table 6. GA and BFO optimimzed PD gains

Optimization process	K_{p1}	K_{d1}	K_{p2}	K_{d2}
GA	2.5148	2.3961	2.7959	1.2933
BFO	2.4718	2.4967	2.2789	1.5524

is plotted against the desired link angle in Fig. 7(b).

The observed overall tip response datas of two-link flexible manipulator using both GA and BFO based PD controller are provided in the Tab. 3. From the table it can

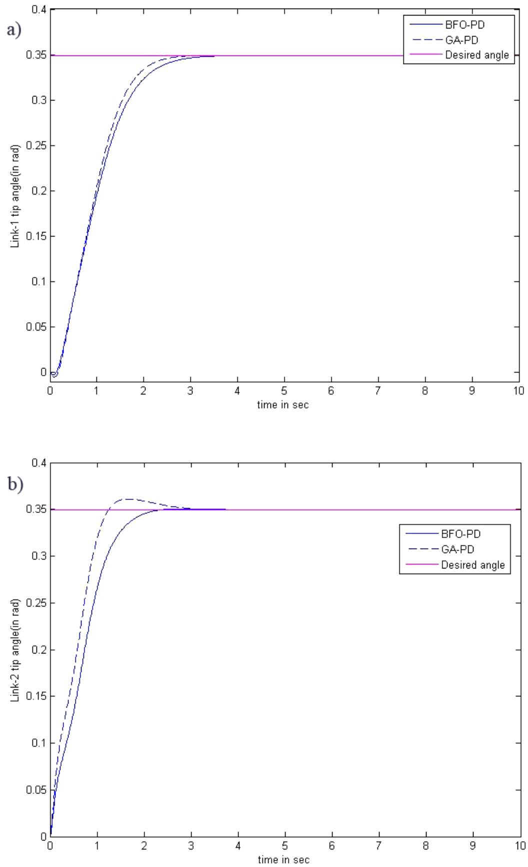


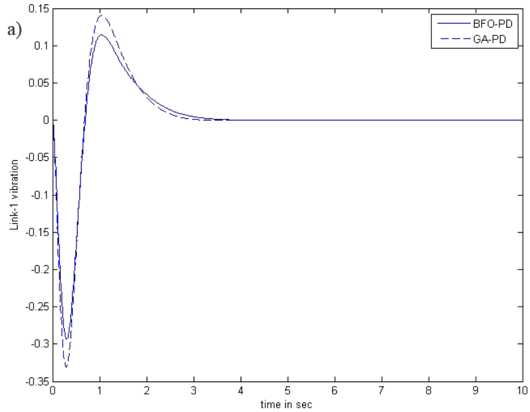
Figure 7. a) link-1 tip revolution, b) link-2 tip revolution using both GA and BFO based PD controller.

Table 7. GA and BFO based PD controller properties

Optimization process	Settling time	Rise time	Peak overshoot
GA	2.8 secs	1 secs	0.016
BFO	2.2 secs	1.3 secs	0.0009

be inferred that in case of GA-PD controller rise time is less but settling time is more compared to the case of BFO-PD controller.

The link vibrations shown in Fig. 8(a) and (b) indicate less vibration in case of PD controller gains obtained in BFO process compared to those obtained in GA process.



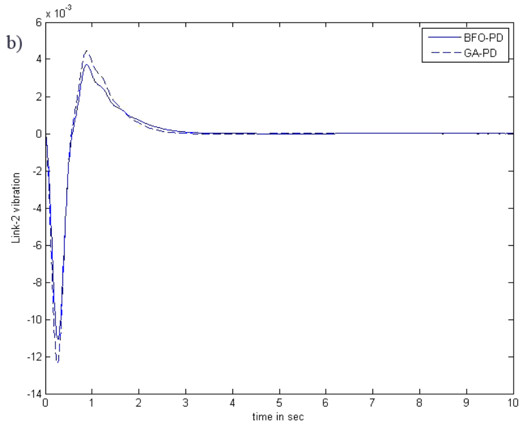


Figure 8. a) Link-1 vibrations, b) Link-2 vibrations using optimal PD gains obtained in both GA and BFO process.

Therefore we concluded that optimization of PD controller gains is best found in BFO technique. The cause of the above result can be deduced as this way; the BFO process involves in finding most of local minimas of fitness (objective) function and global best among them, whereas GA optimization process sticks to one local minima until another local minima is found by mutation operator. So, sometimes GA process could not find the global best of fitness function under consideration as it get stuck in a local minima. Comparing the responses of link tip positions shown in Fig. 7, it also observed that BFO optimized PD gains yield smooth tip performances.

6. Conclusions

In this paper two important types of evolutionary computation technique have been employed successfully to achieve optimal PD controller gains for tip position control of two-link flexible manipulator. Both the optimization algorithms ensure the stability of the closed-loop system simply by a choice of a set of positive valued feedback gains. Both the process show good tip motion performance suppressing the unwanted vibration in the links. But BFO process is found to be more efficient than GA in tuning the feedback gains to optimize certain desired fitness functions and also in smooth tracking. The paper has also included results of simulation experiments demonstrating the effectiveness of the BFO approach compared to the GA approach.

References

[1] S. OZCELIK and E. MIRANDA: Output feed-back direct adaptive control for a two-link flexible robot subject to parameter changes. *Adaptive Control*, Edited by Kwanho You, In-tech Publisher, Croatia, 2009.

[2] W-Y. HAN, J-W. HAN and C-G. LEE: Development of a self-tuning PID controller based on neural network for nonlinear systems. *Proc.of the 7th Mediterranean Conf. on Control and Automation (MED99)*, Haifa, Israel, (1999), 28-30. 1999.

[3] B. SUBUDHI, A.S. MORRIS: Fuzzy and neuro-fuzzy approaches to control a flexible single-link manipulator. *Proc. Instn. Mech. Engrs*, **217** Part I: J. Systems and Control Engineering, 2003.

[4] Y. YONGQUAN, H. YING, W. MINGHUI, Z. BI and Z. GUOKUN: Fuzzy Neural PID controller and tuning its weight factors using genetic algorithm based on Werent location crossover. *IEEE Int. Conf. on Systems, Man and Cybernetics*, (2004).

[5] D.B. FOGEL: The advantages of evolutionary computation. *Natural Selection, Inc.*

[6] A. DE LUCA and B. SICILIANO: Closed-form dynamic model of planar multi-link lightweight robots. *IEEE Trans. on Systems, Man, and Cybernetics*, **21**(4), (1991), 826-839.

[7] D.E. GOLDBERG: Genetic algorithm in search, optimization and machine learning. Reading. Reading MA Addison-Wesley, 1989.

[8] S.S. GE, T.H. LEE and G. ZHU: Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system. *IEEE Trans. on Industrial Electronics*, **43**(5), (1996), 567-574.

[9] K.M. PASSINO: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE. Control Syst. Mag.*, (2002), 52-67.

[10] S. DAS, A. BISWAS, S. DASGUPTA and A. ABRAHAM: Bacterial foraging optimization algorithm: Theoretical foundations, analysis, and applications. *In: Foundations of Computational Intelligence*, **3** ser. Studies in Computational Intelligence, A. Abraham, A.E. Hassanien, P. Siarry and A. Engelbrecht, (Eds). Springer Berlin Heidelberg, **203** (2009), 23-55.