World Academy of Science, Engineering and Technology
International Journal of Electrical and Information Engineering
Vol:2, No:9, 2008

Heuristic Search Algorithms for Tuning PUMA 560 Fuzzy PID Controller

Sufian Ashraf Mazhari, Surendra Kumar Member IEEE

***Abstract*—**This paper compares the heuristic Global Search Techniques; Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing, Generalized Pattern Search, genetic algorithm hybridized with Nelder–Mead and Generalized pattern search technique for tuning of fuzzy PID controller for Puma 560. Since the actual control is in joint space ,inverse kinematics is used to generate various joint angles correspoding to desired cartesian space trajectory. Efficient dynamics and kinematics are modeled on Matlab which takes very less simulation time. Performances of all the tuning methods with and without disturbance are compared in terms of ITSE in joint space and ISE in cartesian space for spiral trajectory tracking. Genetic Algorithm hybridized with Generalized Pattern Search is showing best performance.

***Keywords*—**Controller tuning, Fuzzy Control, Genetic Algorithm, Heuristic search, Robot control.

I. Introduction

THERE are wide variety of linear and nonlinear control methods for Robotic manipulator [1-2].PID controller is most commonly used controller in industry[3].Most of the PID controllers are model based which require precise knowledge of dynamic model including the values of physical parameters involved. Control methods used in robot arm applications however face the major difficulty resulting from the indetermination of actual dynamic parameters of robots. Identification of actual dynamics of robot manipulators is a challenging task because of friction characteristic and some of assumptions made while driving the mathematical equations of manipulators. Also, Robotic arms constantly move among widely separated regions of their workspace, Therefore no linearization valid for all regions can be found, hence, methods of linear control and those of local linearization and moving linearization are not well suited for the control problem of robotic arms. Fuzzy controllers are among the recent nonlinear and nonmodel based control methods [4-6]. Fuzzy controllers are able to make decision on the basis of linguistic information. Majority of fuzzy controllers during past decade are fuzzy PID controllers [7-8]. In designing the fuzzy logic controller, the major task is to determine fuzzy rule base, membership function of input/output variables and input/output scaling factors. The membership functions and scaling factors affect the performance of fuzzy logic based controller. Therefore, tuning is needed for better performance of controller. Most of the conventional tuning methods are derivative driven and need continuous function. Genetic algorithm, Simulated Annealing and Particle Swarm Optimization are getting popular because of there ability to finding global minima in both continuous and non-continuous domain.

Genetic algorithms are stochastic search algorithm inspired by the principle of natural selection and natural genetics. Genetic Algorithm has considerably broadened the scope of optimization in engineering [9]. Simulated Annealing (SA) is motivated by an analogy to annealing in solids. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady frozen state. SA's major advantage over other methods is its ability to avoid becoming trapped at local minima. The algorithm employs a random search, which not only accepts changes that decrease objective function, but also some changes that increase it with some probability [10-11]. Particle Swarm Optimization (PSO) is a population based stochastic optimization technique. The underlying motivation for development of PSO algorithm is social behavior of bird flocking or fish schooling. PSO is also an optimizer based on population. The system is initialized with a set of randomly generated potential solutions, and then performs the search for the optimum one iteratively. PSO does not possess the crossover and mutation processes used in GAs, it finds the optimum solution by swarms following the best particle [12].Nelder and Mead nonlinear simplex algorithm is direct search method .This algorithm for nonlinear optimization is based on the concept of a simplex-a geometric object that is the convex hull of points not lying in the same hyper plane. The basic idea in the nonlinear simplex algorithm is that at each iteration a new point is generated in or near the current simplex. Usually, this new point replaces one of the current simplex vertices, yielding a new simplex. Generalized pattern search (GPS) algorithms are derivative free methods for the minimization of smooth functions, possibly with linear inequality constraints[13].Generalized pattern search algorithms for unconstrained or linearly constrained minimization generate a sequence of iterates with non-increasing objective function values. Iteration is divided into two phases: an optional search and a local poll. Various

Manuscript received June 2008
Sufian Ashraf Mazhari is working in GS E&C Gurgaon , India (Email: sufian.ashraf@gmail.com)
Surendra Kumar is Assistant Professor in the Department of Electrical Engineering, Indian Institute of Technology, Roorkee, Uttrakhand 247667, India (Email: surendra_iitr@yahoo.com)

hybrid search methods have been proposed for engineering optimization problem[14-16].In this paper, tuning performance of GA,PSO, SA,GPS, GA-NM (GA hybridized with NM local search technique) and GA-GPS (GA hybridized with GPS) is being compared for tuning of Fuzzy PD+I controller .

II. MANIPULATOR DYNAMICS AND KINEMATICS

The dynamics of n link robotic manipulator is represented as

$$A(\theta)\ddot{\theta} + B(\theta)[\dot{\theta}\dot{\theta}] + C(\theta)[\dot{\theta}^2] + g(\theta) = \tau$$

Where

$A(\theta)$ is the $n \times n$ kinetic energy matrix,

$B(\theta)$ is the $n \times n(n-1)/2$ matrix of coriolis torques,

$C(\theta)$ is the $n \times n$ matrix of centrifugal torques

$g(\theta)$ is $n \times 1$ vector of gravity torques

$\ddot{\theta}$ is $n \times 1$ vector of acceleration

τ is $n \times 1$ vector of joint torque.

The symbol $[\dot{\theta}\dot{\theta}]$ and $[\dot{\theta}^2]$ are vector of velocity and squared velocity product. The dynamic parameters of Puma 560 have been taken from [17]. Puma 560 joint actuators are DC servo motors with armature voltage as control input. The motor is connected to manipulator links through gear where the Robot dynamics appears as dynamic load. The dynamics of DC motor can be represented as

$$E_a = E_b + L\frac{dI}{dt} + RI \quad (1)$$

$$E_b = K_e N \Omega \quad (2)$$

$$I = \frac{E_a - K_e N \Omega}{Ls + R} \quad (3)$$

$$\tau = K_m I \quad (4)$$

Where E_a is the armature voltage, E_b the Back e.m.f, L and R are inductance and reactance of armature windings respectively, I is the armature current, N is gear ratio, K_e is the back e.m.f constant, K_m is motor constant and Ω is load angular velocity. Actuator data of puma 560 Robot is taken from [18].The transformation between the joint space and the Cartesian space of the robot is very important since robots are controlled in the joint space, whereas tasks are defined and object manipulated in the Cartesian space. While modeling the kinematics of manipulator, arm singularity and configuration must be checked. The Forward and Inverse kinematic equations of puma 560 are taken from [19]. Control system diagram of Puma 560 is shown in Fig.1 which consists of desired Cartesian space trajectory T, inverse kinematics block I, Fuzzy PD+I controller, servo motor M, dynamics D and forward kinematics Block F. Simulink based simulation model of puma 560 dynamics and complete simulation diagram is shown in Fig.2 and Fig.3 .

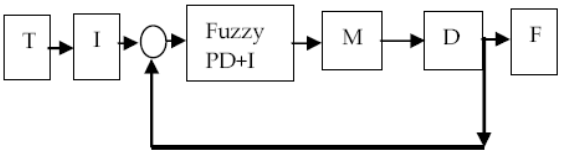

Fig. 1 Control system diagram of puma 560

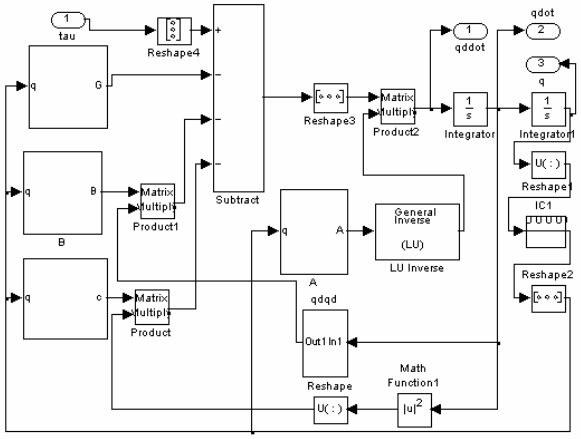

Fig. 2 Simulink diagram of PUMA560 dynamics

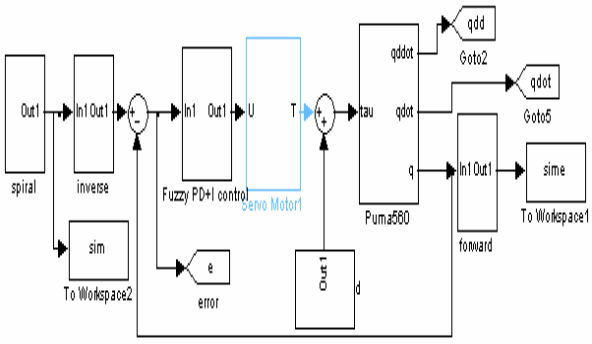

Fig. 3 Simulink simulation diagram of PUMA560

III. FUZZY PD+I CONTROLLER

Fuzzy controller can be designed using parameters of crisp PID controller. Fuzzy PID controller is implemented as Fuzzy PD+I controller .The input to fuzzy controllers are error and error change. Matlab simulation diagram of fuzzy PD+I Controller is shown in Fig.4.

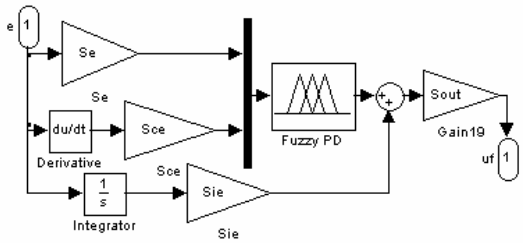

Fig. 4 Matlab simulation diagram of Fuzzy PD+I

Important step involved in designing fuzzy controller is rule base generation and input output gains setting. If $\theta_d(k)$ is desired joint angle and $\theta(k)$ is actual output angle at any sampling instant k , error *e(k)* , change in error $\dot{e}(k)$ and integral error *ie(k)* are given as

$$e(k) = \theta_d(k) - \theta(k) \tag{5}$$

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{T_s} \tag{6}$$

$$ie(k) = \sum_{k=1}^{n} \dot{e}(k) T_s \tag{7}$$

For classical PD controller the controller output is given as

$$u(k) = k_p \left(e(k) + T_d \dot{e}(k) \right) \tag{8}$$

Where k_p is gain of classical PD controller, T_d is derivative time constant and *u (k)* is control signal

When action signal *u(k)* is equal to zero

$$k_p \left(e(k) + T_d \dot{e}(k) \right) = 0 \tag{9}$$

$$\dot{e}(k) = -\frac{1}{T_d} e(k) \tag{10}$$

From (10) it is clear that ė(t) directly depends upon T_d. If state trajectory of the closed loop controlled system with PD controller for some constant PD value is plotted, it draws a sharp boundary between positive and negative control signals. This can be used to map rule base in discrete state space by taking the diagonal element of rule base as ZE. Rule base for fuzzy controller is given in Table I. Crisp PID controller parameters are used to initially set fuzzy input output gains. Input error scaling factor is S_e , error change scaling factor is S_{ce} and output scaling factor is S_{out} the fuzzy controller output u_f is given as

$$u_f = \left(S_e e(k) + S_{ce} \dot{e}(k) + S_{ie} ie(k) \right) S_{out} \tag{11}$$

$$u_f = S_e S_{out} \left(e(k) + \frac{S_{ce}}{S_e} \dot{e}(k) + \frac{S_{ie}}{S_e} ie(k) \right) \tag{12}$$

Comparing (12) with the crisp PID controller output, values of scaling factors come out to be

$$S_e S_{out} = k_p \qquad \frac{S_{ce}}{S_e} = T_d \qquad \frac{S_{ie}}{S_e} = \frac{1}{T_i}$$

If maximum probable error for any joint is e_{max} and input /output membership function universe is taken as [-1 1], the error scaling factor S_e can be set to $\frac{1}{e_{max}}$ .Error change and output scaling factor will be

$$S_{out} = k_p e_{max} \qquad S_{ce} = \frac{T_d}{e_{max}} \qquad S_{ie} = \frac{1}{e_{max} T_i}$$

Since better trajectory always starts from the current position of joint angle, initial tracking angle is zero. Taking a worst condition error of 10 radians. Initial value of error scaling factor S_{e0} is set to 0.1 and all other initial scaling factors $S_{ce0}, S_{ie0}, S_{out0}$ are calculated using values of classical PID parameters taken from [20] as given in Table II. All these values are initial setting and fine tuning is always needed.

TABLE I
RULE BASE FOR FUZZY CONTROL

$e \setminus \dot{e}$	NB	NM	NS	ZE	PS	PM	PB
PB	ZE	PS	PM	PB	PB	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PS	NM	NS	ZE	PS	PM	PB	PB
ZE	NB	NM	NS	ZE	PS	PM	PB
NS	NB	NB	NM	NS	ZE	PS	PM
NM	NB	NB	NB	NM	NS	ZE	PS
NB	NB	NB	NB	NB	NM	NS	ZE

TABLE II
INITIAL VALUE OF SCALING FACTORS

Joint	S_{e0}	S_{ce0}	S_{ie0}	S_{out0}
1	0.1	0.009	0.4	2109.375
2	0.1	0.0084	0.4267	2400
3	0.1	0.0084	0.4267	2400
4	0.1	0.0056	0.64	5400
5	0.1	0.0056	0.64	5400
6	0.1	0.0056	0.64	5400

IV. TUNING METHODS

In tuning of fuzzy controller in case of nonlinear system like robot dynamics, upper and lower bound of search space is always needed because there are chances of getting singular solution if search space is not bounded. Search space lower (*lb*) and upper (*ub*) bounds are taken as

$$lb = \frac{[S_{e0}, S_{ce0}, S_{ie0}, S_{out0}]}{3} \qquad ub = 3[S_{e0}, S_{ce0}, S_{ie0}, S_{out0}]$$

The performance index for tuning is taken as

$$f(s) = \sum_{j=1}^{6} \sum_{k=1}^{n} e^2(kj).kj^2$$

Where e*(kj)* is the system error at k^{th} sampling instant for j^{th} joint.

A. Genetic Algorithm

GA acts as a controller which modifies the set of parameters

$$\{S_{ej}\}_{j\in 1;p}, \{S_{cej}\}_{j\in 1;p}, \{S_{iej}\}_{j\in 1;p}, \{S_{outj}\}_{j\in 1;p}$$

of the j^{th} population which consist of p individual parameters of control system .This cycle is repeated until convergence criteria is met. In the evaluation step of GA, a simulation is performed for each u_f .

B. Simulated Annealing

This method simulates the annealing process in which a substance is heated above its melting temperature and then gradually cooled to produce the crystalline lattice, which minimizes its energy probability distribution. This crystalline lattice, composed of millions of atoms perfectly aligned, is a beautiful example of nature finding an optimal structure.

The algorithmic analog to this process begins with a random guess of the fitness function variable values. Heating means randomly modifying the variable values. Higher heat implies greater random fluctuations. The fitness function *f(s)* returns the output associated with a set of variables. If the output decreases, then the new variable set replaces the old variable set. If the output increases, then the output is accepted provided that

$$r \le e^{[f(S_{old})-f(S_{new})]/S}$$

Where r is a uniform random number .Otherwise, the new variable set is rejected. As the temperature of the system decreases, the probability of accepting a worse move is decreased. If the temperature is zero, then only better moves will be accepted. Thus, even if a variable set leads to a worse fitness value, it can be accepted with a certain probability. The new variable set is found by taking a random step from the old variable set

$$S_{new} = dS_{old} \tag{13}$$

The variable d is either uniformly or normally distributed about S_{old}. This control variable sets the step size so that, at the beginning of the process, the algorithm is forced to make large changes in variables values. At times the changes move the algorithm away from the optimum, which forces the algorithm to explore new regions of variable space. After a certain number of iterations, the new variable set no longer leads to lower fitness value. At this point the values of S and d decrease by a certain percent and the algorithm is repeated.

C. Particle Swarm Optimization

The algorithm is inspired by the social behavior of animals, such as bird flocking or fish schooling. PSO is similar to the continuous GA in that it begins with a random population matrix. Unlike the GA, PSO has no evolution operators such as crossover and mutation. The rows in the matrix are called particles (same as the GA chromosome). They contain the variable values and are not binary encoded. Each particle moves about the fitness function with a velocity.The particles update their velocities and positions based on the local and global best solutions:

$$v_{m,n}^{new} = v_{m,n}^{old} + \Gamma_1 \times r_1 \times (p_{m,n}^{localbest} - p_{m,n}^{old}) + ... \\ ....\Gamma_2 \times r_2 \times (p_{m,n}^{gobalbest} - p_{m,n}^{old}) \tag{14}$$

$$p_{m,n}^{new} = p_{m,n}^{old} + v_{m,n}^{new} \tag{15}$$

Where

$v_{m,n}$ Particle velocity, $p_{m,n}^{localbest}$ best local solution

$p_{m,n}$ Variables $p_{m,n}^{gobalbest}$ best global solution

r_1, r_2 random number, Γ_1, Γ_2 learning factor

The PSO algorithm updates the velocity vector for each particle then adds that velocity to the particle position . Velocity updates are influenced by both the best global solution associated with the lowest cost ever found by a particle and the best local solution associated with the lowest cost in the present population. If the best local solution has a cost less than the cost of the current global solution, then the best local solution replaces the best global solution. The particle velocity is reminiscent of local minimizers that use derivative information, because velocity is the derivative of position. The constant G1 is called the cognitive parameter. The constant G2 is called the social parameter. The advantages of PSO are that it is easy to implement and there are few parameters to adjust. The PSO is able to tackle tough cost functions with many local minima.

D. Generalized Pattern Search

Generalized pattern search algorithms for unconstrained or linearly constrained minimization generate a sequence of iterates with non-increasing objective function values. Iteration is divided into two phases: an optional search and a local poll. In the search step $f(s)$ is evaluated at a finite number of points on a mesh (a discrete subset of bounded search space *[lb, ub]*) to find one that yields a lower $f(s)$ value than the incumbent.

Mesh :

$$M_k = \left\{ S_k + \Delta_k D_z : z \in Z_+^{|D|} \right\} \tag{16}$$

D is a positive spanning set, Δ_k is mesh size parameter and S_k is a mesh local optimizer.

Poll set:

$$\{ S_k + \Delta_k d : d \in D_k \} \tag{17}$$

Search algorithm is given as

Step 0 :

Let S_0 be such that $f(S_0)$ is finite and M_0 be the intial mesh defined by $\Delta_0 > 0$, and D_0 .Set the iteration counter *k* to 0.

Step 1 :

Perform the Search and possibly the Poll steps (or only part of them) until an improved mesh point S_{k+1} with the lowest so far $f(s)$ value is found on the mesh M_k defined by (16). Evaluate $f(s)$ on the poll set defined in (17)

Step 2:

If the search or the poll step produced an improved mesh point, i.e., a feasible iterate $x_{k+1} \in M_k \cap \Omega$ for which $f(S_{k+1}) < f(S_k)$, then update $\Delta_{k+1} \ge \Delta_k$ by (18)

$$\Delta_{k+1} = \tau^{w_k} \Delta_k$$

for $0 < \tau^{w_k} < 1$ where $\tau > 1$ is a rational number that remains constant over all iterations, and $w_k \ge 0$ is an integer .If, $f(S_k) \le f(S_k + \Delta_k d)$ for all $d \in D_k$, Set $S_{k+1} = S_k$, update $\Delta_{k+1} < \Delta_k$ by (18) with $w_k \le -1$.

Increase $k \leftarrow k+1$ and go back to the **Step 1.**

E. Nelder-Mead method

This algorithm for nonlinear Nelder-Mead optimization method consists of the following steps :

Step.0 *Initialization*:

Generate an initial set of p + 1 extreme points in bounded search space say $S^i (i = 1, 2, ..., p+1)$ representing the vertices of the initial simplex. Calculate *f(s)* for each extreme

point. Set values for the algorithm coefficients α, β, γ and δ. Default settings are 1.0, 0.5, 2.0, and 0.5 respectively

Step.1 *Reflection*

Identify the vertices where the maximum, second highest, and minimum *f* values occur. Let $S_{\max}$, $S_{2\max}$ and $S_{\min}$ represent these points, respectively. Let S_{cent} represent the centroid of all S^i except for $S_{\max}$ Generate a new candidate vertex S_{refl} by reflecting $S_{\max}$ through S_{cent} according to $S_{refl} = (1+\alpha)S_{cent} - \alpha S_{\max}$. $(\alpha > 0)$

Step.2.a *Accept reflection*

If $f(S_{\min}) \le f(S_{refl}) < f(S_{2\max})$ then S_{refl} replaces $S_{\max}$ in the simplex and proceed to step 3; else go to step.2.b.

Step2.b *Expansion*

If $f(S_{refl}) < f(S_{\min})$ the reflection is expanded according to $S_{\exp} = \gamma S_{refl} + (1-\gamma)S_{cent}$

Where the expansion coefficient $\gamma > 1$, else go to step.2.c. If $f(S_{\exp}) < f(S_{refl})$, then $S_{\exp}$ replaces $S_{\max}$ in the simplex; otherwise, the expansion is rejected and S_{refl} replaces $S_{\max}$ .Go to step 1.

Step.2.c *Contraction*

If $f(S_{refl}) \ge f(S_{2\max})$,then the simplex contracts to reflect the poor S_{refl} .Consider the two cases: (1) $f(S_{2\max}) \le f(S_{refl}) < f(S_{\max})$ (outside contraction) (2) $f(S_{refl}) \ge f(S_{\max})$ (inside contraction).The contraction point is determined by $S_{cont} = \beta S_{\max/refl} + (1-\beta)S_{cent}$, $0 \le \beta \le 1$ where $S_{\max/refl} = S_{refl}$ in case(1) or $S_{\max/refl} = S_{refl}$ in case(2).In case(1) if $f(S_{cont}) \le f(S_{refl})$ the contraction is accepted .In case(ii),if $f(S_{cont}) < f(S_{\max})$ the contraction is accepted .If contraction is accepted replace $S_{\max}$ with S_{cont} and go to step3. If contraction is not accepted go to step.2.d.

Step.2.d *Shrink*

The contraction has failed and the entire simplex shrinks according to a factor of $0 < \delta < 1$ retaining only $S_{\min}$. This is done by replacing each vertex S^i (except $S_{\min}$) by $\delta S^i + (1-\delta)S_{\min}$. Go to step.3.

Step.3 *Termination*

Stop if convergence criterion is satisfied or if the maximum number of function evaluations has been reached; else, returns to step 1.

F. Hybrid GA Technique

Genetic algorithm is well suited for global search techniques. GA reaches its search space very fast. On the other hand Nelder-Mead and Generalized pattern search are very efficient in local search techniques. Initial starting point affects the performance of local search techniques. Combining the advantages of both GA and local search technique, the performance of search algorithm is improved .GA is used to set starting point of NM and GPS techniques. Algorithm consists of running GA for two time of population dimension. Start GPS or NM keeping starting point as solution coming out from GA.

V. Results

Initial values of Fuzzy PD+I scaling factors are choosen from crisp PID parameter value. Scaling factors are tuned with GA for 100 iteration with population size=40, crossover probability =0.8 and mutation probability=.01.Fitness function curve of GA tuning is shown in Fig.5 which clearly shows fast initial convergence characteristics of GA. It is clear from Fig.5 that GA is converging very fast initially but taking more iteration later to reach optimal point. Fuzzy PD+I is tuned with PSO for 100 iterations with population size=40, maximum particle velocity=6 and acceleration constant (local and global best)=2.Fitness value curve for tuning using PSO is shown in Fig.6.PSO is showing better performace compared to GA.Fig.7 shows fitness curve for Tuning using SA .It is clear from Fig.7 that SA fails to find global optimal solution. Fig.8 shows best function value for tuning using GPS.GPS performnce is found to be better compared to GA and SA.From Fig.5 and Fig.8 it is clear that GA is reaching near optimal value very fast while GPS shows just opposite result of GA .GPS initial convergence rate is slow but once it reaches optimal search space, it is decreasing very sharply. More precisely it is concluded that GA is showing capability of global search but GPS is more efficient in local search. Merits of both GA and GPS can be combined to give a better hybrid search techniques. NM method is another best method for local search. Controller is tuned using hybrid GA-GPS and GA-NM .GA is runned for iteration=30, output of GA is set to be starting point of GPS and NM. Fitness curve of both the hybrid techniques in Fig.9 and Fig.10 show that GA-GPS is converging fast compared to GA-NM. Both these are showing better performance compared to GA,SA and PSO. Comparing in terms of numbers of times fitness function is evaluated, GA and PSO takes 40x100=4000, while hybrid techniques take maximum of 40*30+100=1300 and searching global optima. Controller tuned using all these methods are tested for spiral trajectory tracking. The parametric equation of spiral trajectory is

$$x(t) = t\cos(t) + 400 \qquad y(t) = 0.0286(10 - 3t^2 / pi) + 200$$
$$z(t) = 3.33t\sin(t) + 250$$

All dimensions are in mm.To check the robustness of controller disturbance torque *d* is applied.

$$d = 1.5\sin(4.3575t) + \sin(9.825) + \sin(2.7075) + 1$$

Comparison in terms of performance index f(s) and cartesian space integral squared error *ISEX,ISEY* and *ISEZ* is shown in Table. III.

$$ISEX = \sum_{k=1}^{n}(dx)^2 \ , \quad ISEY = \sum_{k=1}^{n}(dy)^2 \ , \quad ISEZ = \sum_{k=1}^{n}(dz)^2$$
$$dx = x_d(k) - x(k) \ , \ dy = y_d(k) - y(k) \ , \ dz = z_d(k) - z(k)$$

Joint angles for spiral trajectory tracking without disturbance using untuned Fuzzy PD+I controller are shown in Fig.11.a and Fig.11.b.Fig.12 shows the spiral trajectory tracking using untuned Fuzzy PD+I controller without

disturbance and Fig.13 shows corresponding is tracking error in cartesian space. Since order of error is small compared to joint angle, the actual and desired joint angles are overlapping but the difference is clear in form of error in all the figures. Fig.14.a and Fig14.b show joint angles with disturbance using untuned controller .Fig. 15 and Fig.16 show tracked trajectory and corresponding Cartesian space error. A large error is coming out in case of untuned controller which is shown in Table.III .The controller is tuned with various methods but best is hybrid techniques. Therefore responses of only GA-GPS shown and others are being compared in form of Table.III .Fig.17.a and Fig.17.b shows joint angle using GA-GPS without disturbance. Fig.18 and Fig.19 shows tracked trajectory and tracking error. Fig.20.a and Fig.20.b show joint angles using GA-GPS with disturbance .Fig.21 and Fig.22 show tracked trajectory and tracking error. A large amount of error is being reduced with tuning using hybrid techniques.

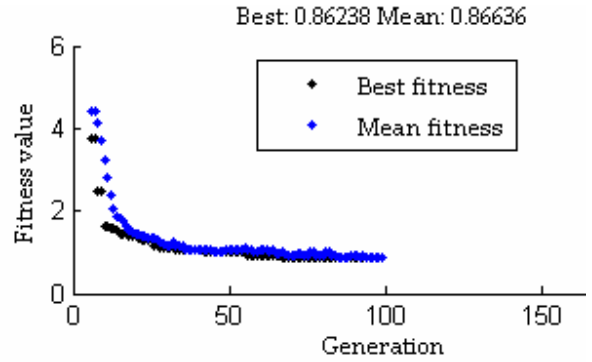

Fig. 5 Fitness function curve of GA

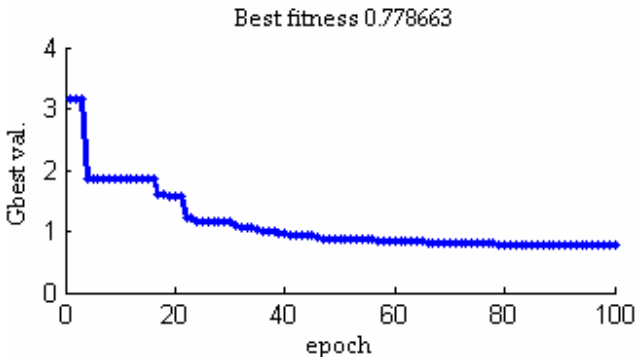

Fig. 6 Fitness function curve of PSO

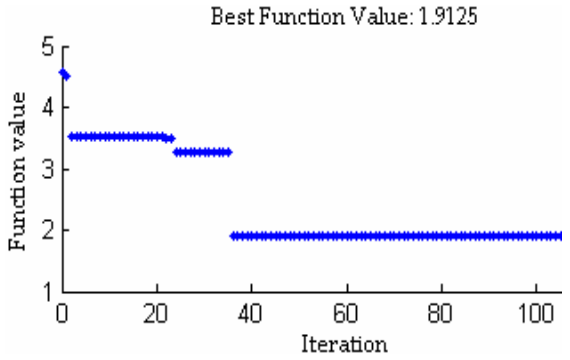

Fig. 7 Fitness function curve of SA

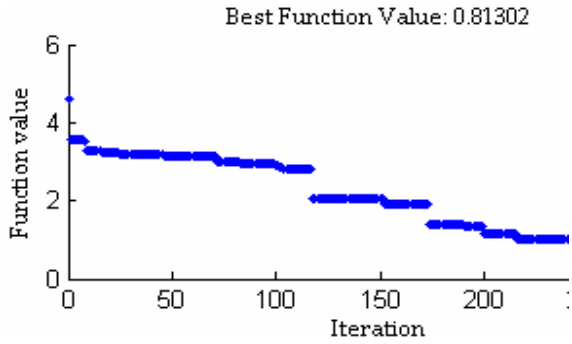

Fig. 8 Fitness function curve of GPS

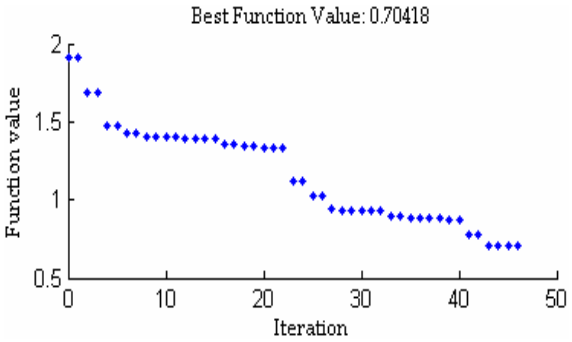

Fig. 9 Fitness function curve of GA-GPS

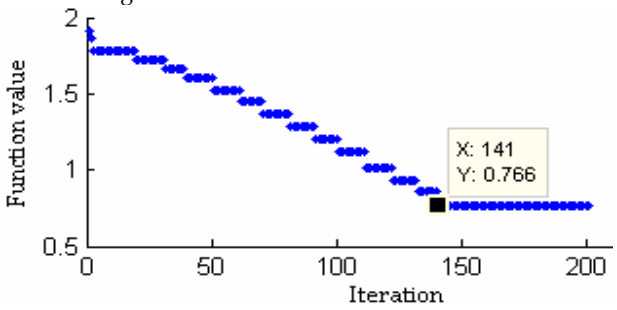

Fig. 10 Fitness function curve of GA-NM

TABLE III
ISE IN CARTESIAN SPACE FOR ALL METHODS

ISEX (mm) ISEY (mm) ISEZ (mm)	Without *d*	With *d*
Untuned	26.8915	599.0691
	0.2823	663.5670
	93.7527	554.9276
GA	15.3052	733.7772
	0.0638	179.2775
	55.5295	308.4535
SA	446.2807	802.5476
	2.7118	391.7494
	57.4151	668.8036
PSO	10.9776	437.1578
	0.0498	244.7739
	29.7441	230.5712
GPS	13.1754	595.4619
	0.1375	124.8692
	5.9448	40.3052
GA-NM	25.2411	458.5317
	0.0820	326.7774
	32.0323	311.3924
GA-GPS	19.2309	474.2470
	0.0383	133.0601
	27.7857	282.4736

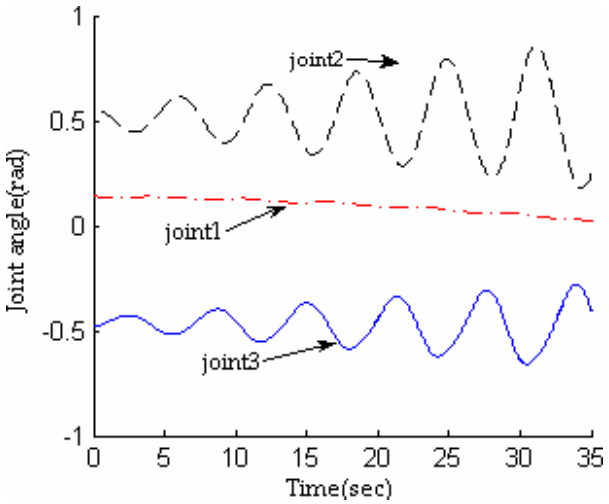

Fig. 11.a joint angles with untuned fuzzy PD+I without disturbance for spiral trajectory.

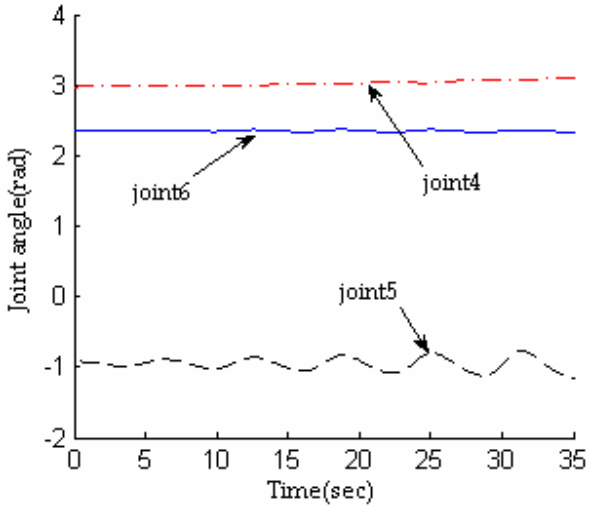

Fig. 11.b joint angles with untuned fuzzy PD+I without disturbance for spiral trajectory.

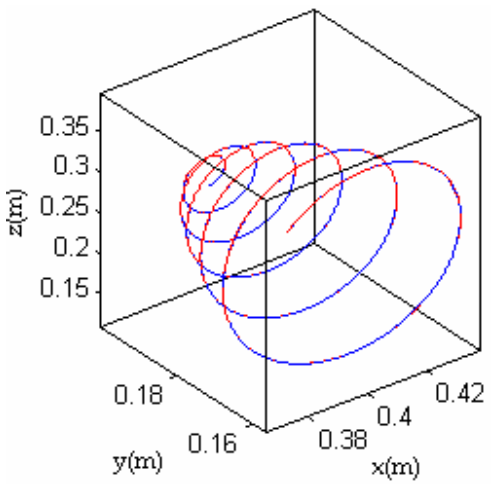

Fig. 12 Spiral trajectory with untuned fuzzy PD+I without disturbance

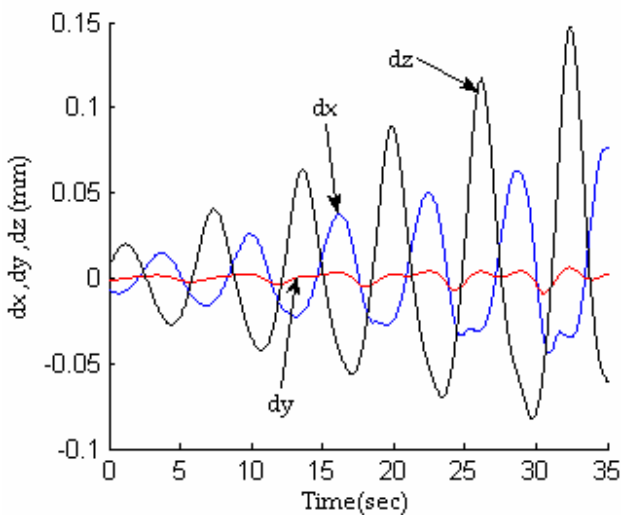

Fig. 13 cartesian space error with untuned fuzzy PD+I without disturbance for spiral trajectory

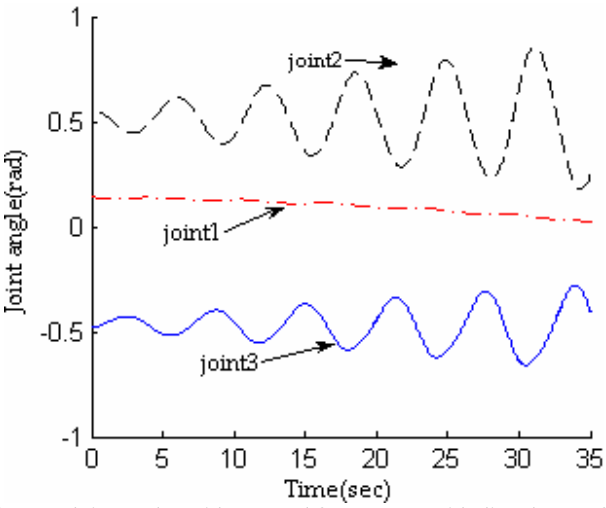

Fig. 14.a joint angles with untuned fuzzy PD+I with disturbance for spiral trajectory

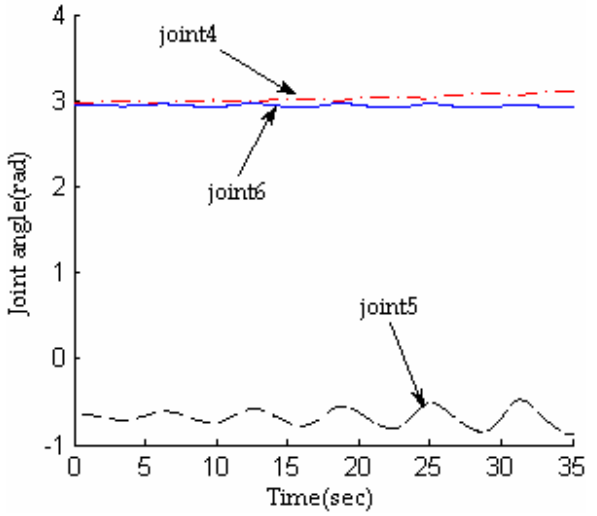

Fig. 14.b joint angles with untuned fuzzy PD+I with disturbance for spiral trajectory

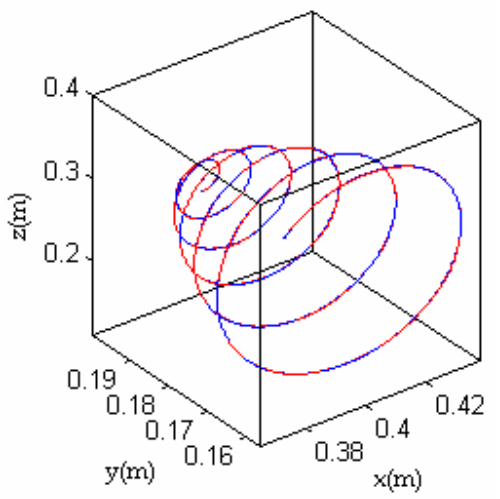

Fig. 15 Spiral trajectory with untuned fuzzy PD+I with disturbance

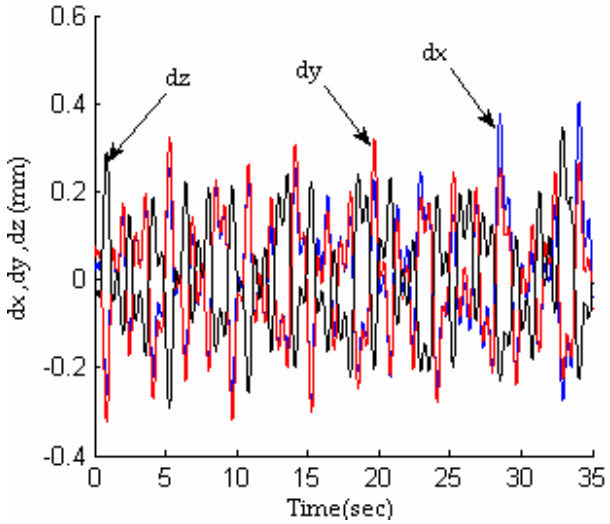

Fig. 16 cartesian space error with untuned fuzzy PD+I with disturbance for spiral trajectory

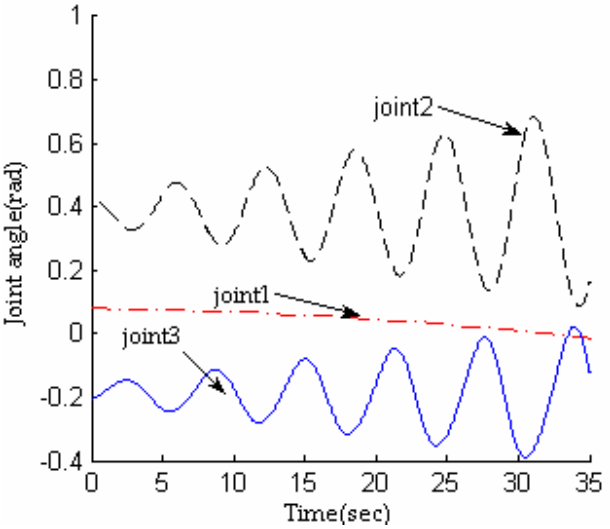

Fig. 17.a joint angles with GA-GPS tuned fuzzy PD+I without disturbance for spiral trajectory

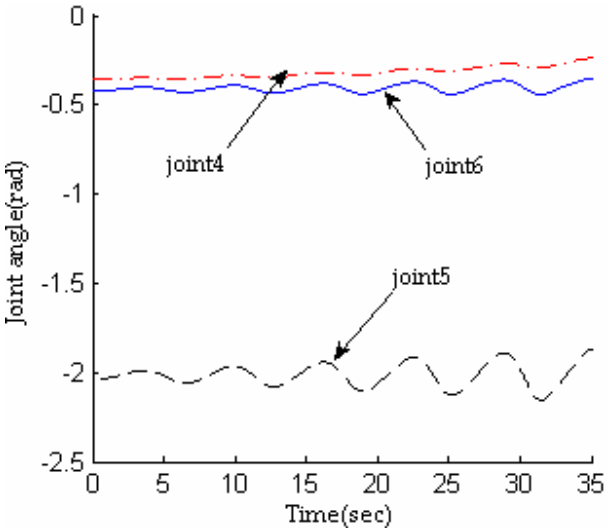

Fig. 17.b joint angles with GA-GPS tuned fuzzy PD+I without disturbance for spiral trajectory

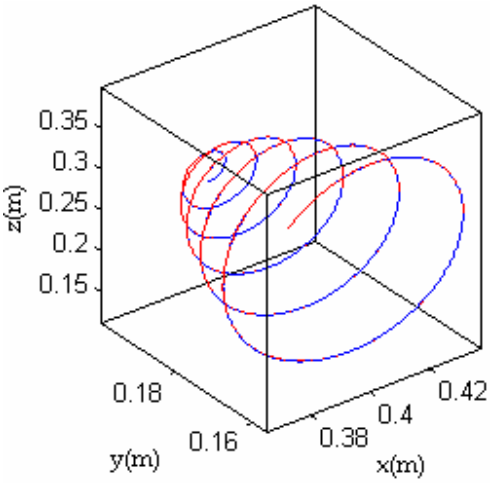

Fig. 18 Spiral trajectory with GA-GPS tuned fuzzy PD+I without disturbance

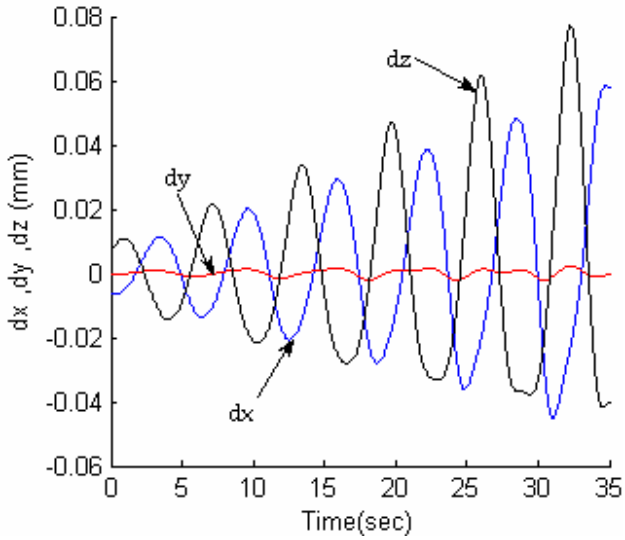

Fig. 19 cartesian space error with GA-GPS tuned fuzzy PD+I without disturbance for spiral trajectory

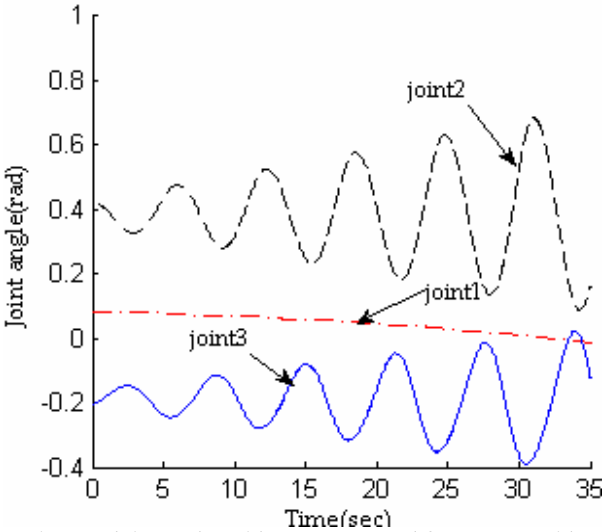

Fig .20.a joint angles with GA-GPS tuned fuzzy PD+I with disturbance for spiral trajectory

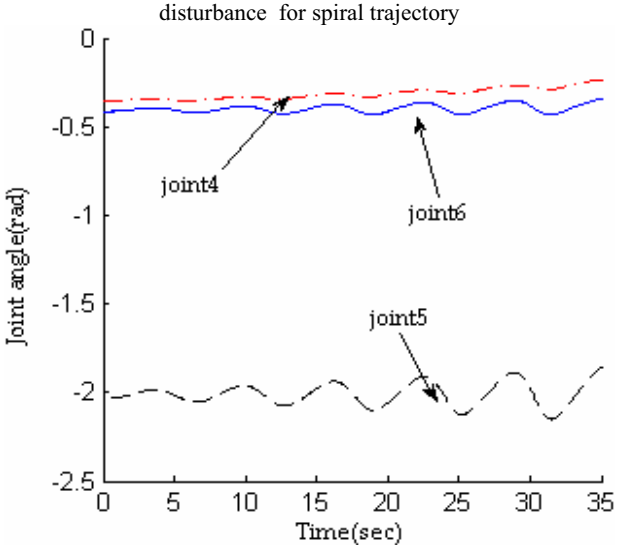

Fig. 20.b joint angles with GA-GPS tuned fuzzy PD+I with disturbance for spiral trajectory

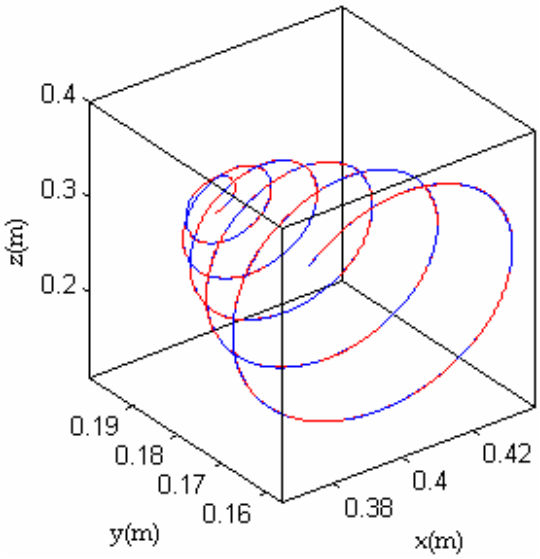

Fig. 21 Spiral trajectory with GA-GPS tuned fuzzy PD+I with disturbance

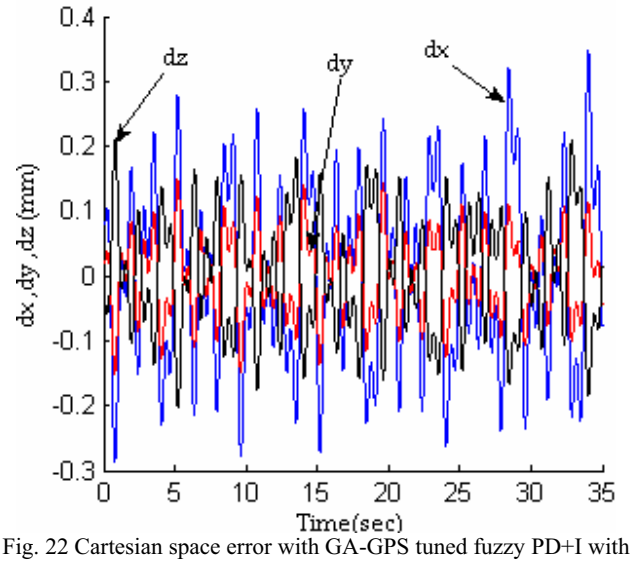

Fig. 22 Cartesian space error with GA-GPS tuned fuzzy PD+I with disturbance for spiral trajectory

VI. Conclusions

In this paper Fuzzy PD+I controller for Puma 560 is designed and tuned using different heuristic search algorithms. Since the actual control is in joint space, inverse kinematics is used in the beginning to generate various joint angles corresponding to desired cartesian space trajectory. The controlled cartesian space trajectory is later obtained using forward kinematics. The tuning of fuzzy PID parameters is carried out using GA, GPS, PSO, SA, hybridized GA-GPS and GA-NM and compared with those based on classical methods. Initial convergence rate of GA and PSO is high but takes more number of iterations later to reach optimal point. Initial convergence rate of GPS is low, but once it is reaching search space, it is quit fast. Tuning performance with and without disturbance is compared for all the above approaches in joint space as well as cartesian space. Hybridized GA-GPS is showing best performance compared to all. Extension of this work is planned for multi objective formulation.

References

[1] R.Kelly, V.santibanez and A.Loria , “Control of Robot Manipulators in joint space, Springer” Advanced Textbooks in Control and Signal Processing, series 2005.

[2] Naganna G.E and Surendra Kumar,” Conventional and intelligent controllers for robotic manipulator,” *IEEE International conference on industrial technology*, pp. 424-428, Dec 2006.

[3] John T. Wen, Steve H. Murphy (1990), “PID control for robot manipulators”; CIRSSE Document #54 Rensselaer Polytechnic Institute, May 1990.

[4] Yaochu Jin , "Decentralized adaptive fuzzy control of robot manipulators , *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: vol. 28, Issue: 1 ,Feb 1998.

[5] Hala Bezine , Nabil Derbel and Adel M. Alimi “ Fuzzy control of robot manipulators: some issues on design and rule base size reduction,” *Engineering Applications of Artificial Intelligence* , vol. 15, Issue 5, pp. 401-416,Sept 2002.

[6] Zong-Mu Yeh , “A systematic method for optimal design of Multivariable Fuzzy Logic Control System,” *IEEE Transaction on fuzzy systems*, vol. 7, no. 5, Oct 1999.

[7] G.MKhoury, M.Saad, H.Y.Kannan, C.Asmar , "Fuzzy PID control of a Five DOF Robot Arm," *Journal of intelligent and robotic systems*, vol. 40,pp. 299-320, ISSN: 0921-0296, , July 2004.
[8] Petr Pivonka , "Comparative analysis of fuzzy PI/PD/PID Controller Based on Classical PID Controller approach", *Proceedings of IEEE international conference on fuzzy systems*, vol. 1, pp.541-546, May 2002.
[9] Lance D. Chambers , *The Practical Handbook of Genetic Algorithms*, Chapman & Hall/CRC, ISBN: 1584882409
[10] Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, No. 459813, pp. 671-680 , May 1983.
[11] R.A.Rutenber, "Simulated Annealing Algorithms: An Overview," IEEE Circuits and Devices, vol. 5, Issue 1, pp. 19-26, ISSN: 8755-3996, Jan 1989.
[12] Maurice Clerc , Particle Swarm Optimization, ISBN 10: 1-905209-04-5.
[13] Robert Michael Lewis and Virginia Torczon , "Pattern Search Methods For Linearly Constrained Minimization," Institute for Computer Applications in Science and Engineering, Report no.98-3, 1998.
[14] D.H. Kim, A. Abraham ,"A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization and Robust Tuning of PID Controller with Disturbance Rejection", *Studies in Computational Intelligence* (SCI) 75, Springer Berlin / Heidelberg, ISBN: 978-3-540-73296-9, pp. 171–199, Aug 2007.
[15] Zahara E and Kao Y T , Hybrid Nelder–Mead simplex search and particle swarm optimization for Expert Systems with Applications, in press, doi:10.1016/j.eswa.2008.02.039.
[16] X.H. Shi , Y.C. Liang , H.P. Lee C. Lu and L.M. Wang ,An improved GA and a novel PSO-GA-based hybrid algorithm Information Processing Letters vol. 93, Issue 5, pp. 255-261, Mar 2005.
[17] B. Brian Armstrong, Oussama Khatib and Joel Burdick, "The explicit dynamic Model and Inertial Parameters of the Puma 560 Arms," Proceedings of IEEE International conference on Robotics and Automation, pp. 510-518, vol. 3, Apr 1986.
[18] T.J.Tarn, A.K.Bejczy, G.T.Marth and A.K.Ramadorai, "Performance Comparison of Four Manipulator Servo Schemes", *IEEE control system magazine*, vol. 13, Issue 1, pp. 22-29, ISSN: 0272-1708, ,Feb 1993.
[19] Shadia Elgazzar, "Efficient Kinematic Transformations for the PUMA 560 Robot," IEEE *Journal of Robotics and Automation*, vol. Ra-1, no. 3, pp. 142-151, ISSN: 0882-4967, Sept. 1985.
[20] Lee S.Wilfinger, "A comparison of Force control algorithm for Robots in contact with flexible environment", CIRSSE Report #135 Rensselaer Polytechnic Institute, Dec 1992.
[21] Matlab Fuzzy Logic Toolbox version 2.2.6

Sufian Ashraf Mazhari was born in India and received his Bachelor degree (Electrical Engineering) from Aligarh Muslim University, Aligarh India, in 2005 and M.Tech degree in Electrical Engineering from Indian Institute of Technology, Roorkee India.Currently is working with GS E & C ,Gurgaon ,India. His area of research includes Robotic Control, Optimization and Computer vision system.

Surendra Kumar (M'07) received B.E (Electrical), M.E(System Engineering and Operations Research) and Ph.D. Electrical in 1969,1971 and 1982 respectively in India. He joined the Department of Electrical Engineering, Indian Institute of Technology Roorkee, India as lecturer in 1972 .He has 36 year of teaching and research experience. Presently he is Assistant Professor .He has been on teaching assignment to University of Technology, Baghdad, IRAQ during 1987-1989.He is member IEEE, Fellow of Institution of Engineers India and member of ISTE India. His area of research interest is mainly Control, Optimization, AI application to Robotic Control and Fuzzy Reliability.