Application of Swarm Intelligence to a Two-Fold Optimization Scheme for Trajectory Planning of a Robot Arm

Tathagata Chakraborti[1], Abhronil Sengupta[1], Amit Konar[1], and Ramadoss Janarthanan[2]

[1] Dept. of Electronics and Telecommunication Engg., Jadavpur University, Kolkata, India
tathagata.net@live.com, senguptaabhronil@gmail.com, konaramit@yahoo.co.in
[2] Department IT, Jaya Engineering College, Chennai, India
srmjana_73@yahoo.com

Abstract. Motion planning of a robotic arm has been an important area of research for the last decade with the growing application of robot arms in medical science and industries. In this paper the problem of motion planning has been dealt with in two stages, first by developing appropriate cost functions to determine a set of via points and then fitting an optimal energy trajectory. Lbest Particle Swarm Optimization has been used to solve the minimization problem and its relative performance with respect to two other popular evolutionary algorithms, Differential Evolution and Invasive Weed Optimization, has been studied. Experiments indicate swarm intelligence techniques to be far more efficient to solve the optimization problem.

1 Introduction

Motion planning of a robot arm requires the formulation of a specific set of via points through which the end effector must pass in order to avoid collision with obstacles in its environment while moving from an initial arm configuration to a final arm configuration. The next step requires the determination of smooth trajectory fitting through the obtained via points. However, the determination of optimal trajectory is the main area of concern.

Several research works have been conducted in this field during the past few years. Saab and VanPutte [1] used topographical maps to solve the problem. In [2] an algorithm has been proposed for obstacle avoidance using convexity of arm links and obstacles. Ziqiang Mao and T.C. Hsia et al. [3] employed neural networks to solve the inverse kinematics problem of redundant robot arms. Recently genetic algorithms (GA) have been used in this field. Traces of work by this method are found in [4-6]. Other researches in this field include Potential Field method [7].

In our paper we propose to solve the motion planning problem by developing an appropriate fitness function. In the first step a cost function has been developed to determine a set of via points subject to obstacle avoidance and other motion constraints. In the next step another cost function has been formulated to determine the trajectory joining the obtained via points by minimizing mechanical energy consumption.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part II, LNCS 7077, pp. 89–96, 2011.

Classical optimization techniques are not applicable here because of the roughness of the objective function surface. We therefore use derivative free optimization algorithms for this purpose. The first one used is Lbest PSO. The basic PSO algorithm is based on the sociological flocking behavior of birds. The Lbest PSO model is a variant of the basic PSO algorithm where each particle interacts directly with other particles of its local neighborhood [10]. The second optimization technique used for analysis is Differential Evolution (DE) which is guided by the Evolution of Chromosomes [9].The third algorithm used is Invasive Weed Optimization (IWO) proposed by Mehrabian and Lucas [8]. It mimics the colonizing behavior of weeds.

In this paper we present a comparative analysis of the efficiency of these evolutionary algorithms in finding the optimal solution. The promising results displayed by Lbest PSO algorithm may open up new avenues of application of swarm intelligence in control of robotic systems.

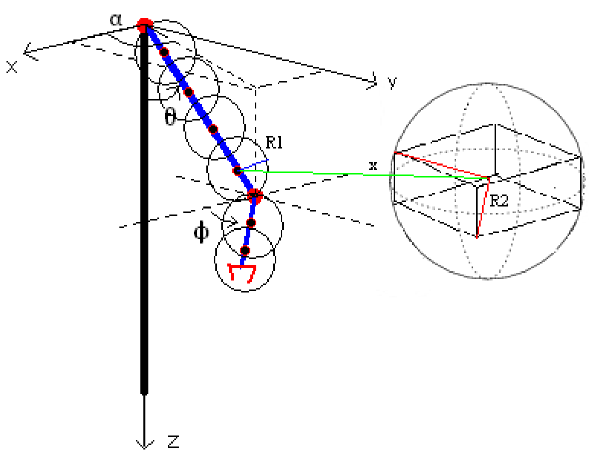

Fig. 1. Model of the Robot Arm

2 Description of the Robot Arm

In the following discussion we use a robot arm with two links and two movable sections each having one degree of freedom. Their movements are described by angles theta and phi which are defined with respect to the co-ordinate system as shown in Figure 1. The first section of length L1 moves in the vertical plane as described by theta (θ) measured from the positive z-axis. The vertical plane in which it moves is displaced from the x-axis by an angle α. The second section of length L2 moves in the horizontal plane as described by phi (ϕ) measured from the positive x-axis.

3 Lbest Particle Swarm Optimization

An initial population of particles (candidate solutions) is generated randomly over the D dimensional space. Each particle has the following characteristics associated with it: $\boldsymbol{x_i}(t)$ which represents the present location of particle and $\boldsymbol{v_i}(t)$which represents the present velocity of particle. Local neighborhoods each of size d are formed at each generation by grouping together particles with minimum Euclidean distance between them.

The individual best fitness and the corresponding location are updated in this step. The best position for each local neighborhood is also updated.

The equation for velocity update of the i^{th} particle is given by equation (1). ω is the inertial coefficient. Here r_1 and r_2 are two independent random numbers where r_1~U(0,1) and r_2~U(0,1). The values of r_1 and r_2 are multiplied by scaling factors c_1 and c_2 where $0<c_1<2$ and $0<c_2<2$. $\boldsymbol{p_i}(t)$represents personal best position of particle at time t and $\boldsymbol{l_i}(t)$ represents the best position that the local neighborhood has encountered so far.

$$\vec{v_i}(t+1) = \omega.\vec{v_i}(t) + c_1.r_1.\left(\vec{p_i}(t) - \vec{x_i}(t)\right) + c_2.r_2.\left(\vec{l_i}(t) - \vec{x_i}(t)\right). \quad (1)$$

In order to ensure that the particle remains bounded in the search space, the velocity of the particle is clamped to $[-v_{max}, v_{max}]$. The maximum velocity is usually chosen as $v_{max}= k*x_{max}$ where the search space is defined in the range $[-x_{max}, x_{max}]$.

Next the position of the particle is updated according to the equation:

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t+1). \quad (2)$$

This process is repeated until maximum number of iterations is reached.

4 Formulation of Cost Function for via Point Determination

4.1 Minimization of Redundant Joint Rotations

Here we purpose to minimize the energy consumed in redundant arm movements. Thus the problem reduces to one of ensuring that the final angle is attained as quickly as possible. Thus the cost function becomes

$$f = K1.\left|\theta_f - \theta\right| + K2.\left|\phi_f - \phi\right|. \quad (3)$$

where K1, K2 are constants of proportionality and (θ_f,ϕ_f) denotes the goal position. In the above equation $\theta = \theta_{prev} + \Delta\theta$ and $\phi = \phi_{prev} + \Delta\phi$ where $(\theta_{prev}, \phi_{prev})$ represents the previous via point and $(\Delta\theta, \Delta\phi)$ denotes the angular displacement between the two via points.

4.2 Obstacle Avoidance

We model the robot arm as a series of consecutive spheres of radius R1, where R1 is determined by the amount of safety margin required. Now the obstacle may be of different shapes and sizes and developing different penalty terms for each of them is a rather futile process. As a simple model we have approximated the obstacles in the robot environment by equivalent circumspheres.

If the distance 'x' between the centre of the sphere on the arm and the centre of the sphere representing the obstacle is less than R1 + R2 then there is a chance of collision and hence in such cases the cost function should incorporate a large penalty term. In cases where this distance is larger, the penalty term should be negligible. Thus the penalty component of the cost function finally takes the form (for the i^{th} sphere and j^{th} obstacle):

$$\sum_i \sum_j K3.C_{ij}.exp\,(-x_{ij}/(R1_i + R2_j))\,. \quad (4)$$

where K3 is a constant of proportionality and is in general much greater than K1 and K2 since this lends more weight to the penalty term as obstacles must be avoided at any cost. Here, $C_{ij} = 1$ if $x_{ij} < R1_i + R2_j$ and is equal to zero otherwise.

Thus in its final form, the cost function becomes:

$$f = K1.|\theta_f - \theta| + K2.|\phi_f - \phi| + \sum_i \sum_j K3.C_{ij}.exp\big(-x_{ij}/(R1_i + R2_j)\big)\,. \quad (5)$$

To ensure uniform distribution of via-points in the joint space, the optimizer has been employed to produce optimized values of $\Delta\theta$ and $\Delta\phi$ within a certain given range ($-\Delta\theta_{max}$, $\Delta\theta_{max}$) and ($-\Delta\phi_{max}$, $\Delta\phi_{max}$) which have been added to the θ_{prev} and ϕ_{prev} values, and the process is repeated till the goal is reached.

5 Formulation of Cost Function for Trajectory Planning

Here we develop an energy efficient method of fitting a smooth trajectory to the set of via-points found above. Let us consider that we have $n+1$ via-points (including initial and final points). We fit smooth cubic polynomials for theta and phi as functions of time in between each of these points as shown below:

$$\theta = a_0 + a_1 t + a_2 t^2 + a_3 t^3\,. \quad (6)$$

$$\phi = b_0 + b_1 t + b_2 t^2 + b_3 t^3\,. \quad (7)$$

where the coefficients are determined partly by a set of boundary conditions and partly by energy minimization criterion. The boundary conditions for the polynomial between the i^{th} and $(i+1)^{th}$ via-points are as follows:

$$\theta = \theta_i, \phi = \phi_i\,;\ \bar{\theta} = \bar{\theta}_i, \bar{\phi} = \bar{\phi}_i\ at\ t = 0. \quad (8)$$

$$\theta = \theta_{i+1}, \phi = \phi_{i+1}\ at\ t = T\,. \quad (9)$$

where i=1, 2,.., n and T is the time to move from the i^{th} to the $(i+1)^{th}$ via-point, and $\bar{\theta}, \bar{\phi}$ are the first time derivatives of theta and phi respectively.

Evidently the above conditions give three equations for four unknown coefficients, and the final equation is provided by the energy term. At any point the mechanical energy of the arm will be given by the summation of kinetic and potential energies of each of the arm sections. The total mechanical energy of the system is equal to

$$E = \frac{1}{2}m_2{L_1}^2\bar{\theta}^2 + \frac{1}{6}\left(m_2{L_1}^2\bar{\theta}^2 + m_2{L_2}^2\bar{\phi}^2\right) + \frac{1}{2}m_2L_1L_2\cos(\theta)\cos(\phi)\,\bar{\theta}\bar{\phi}$$
$$+gH(m_1+m_2) - gL_1\left(\frac{1}{2}m_1+m_2\right)\cos(\theta). \quad (10)$$

In this equation we put the expressions for θ, ϕ and $\bar{\theta}$,$\bar{\phi}$; and replace a_2, b_2 in terms of a_3,b_3 so that E is now a function of time, and a_3,b_3. The energy integrated over a single time interval will give a measure of the total energy consumed and this is a function of a_3 and b_3 (the integration is done using recursive adaptive Simpson quadrature technique). Thus we can find an optimum value of the remaining coefficient by minimizing F_K, where

$$F_K(a_3,b_3) = \int_0^T E.dt\,. \quad (11)$$

```
MOTION PLANNING ALGORITHM
INPUT: Initial and Final Arm Positions
OUTPUT: Energy Efficient Trajectory
Begin
Step1: Determination of Via Points
       Repeat
       Run Lbest PSO
       Input: cost function f, optimizer parameters;
       Output: optimized values of (Δθ,Δϕ);
       Update θprev and ϕprev in f;
       If goal is reached,
       Stop;
       Else Continue;
       Obtain n + 1 via points;
Step2: Determination of Optimal Energy Trajectory
       For i = 1 to n
       Run Lbest PSO
       Input: cost function Fk, optimizer parameters;
       Output: optimized values of a3,b3;
       End For
End
```

6 Experimental Results

6.1 Optimized Trajectory Generation

The robot arm starts from initial position defined by $(\theta, \phi)_{initial}$ and reaches the goal defined by $(\theta, \phi)_{final}$. We have considered cubical obstacles with edge length 1.5, so that R2 for the equivalent sphere becomes (√3/2)*1.5=1.29. The path taken by the robot arm is shown below in Figure 2. The optimal set of parameters required to generate the trajectory was determined by optimizing the cost functions with each of the 3 algorithms over a series of 10 runs and then averaging the result.

Table 1. Simulation Parameters

Simulation parameters	Estimated values	Simulation parameters	Estimated values
$(\theta, \phi)_{initial}$	(0,0) deg	m1	1
$(\theta, \phi)_{final}$	(90,90) deg	m2	1
K1	10	L1	5
K2	10	L2	5
K3	100	obstacle1	(3,8,3)
K4	10000	obstacle2	(3,9.5,3)
R1	0.5	obstacle3	(3,8,4.5)
R2	1.29	obstacle4	(3,9.5,4.5)
H	20	$(-\Delta\theta, \Delta\theta)$	(-20,20)deg
T	1	$(-\Delta\phi, \Delta\phi)$	(-20,20)deg

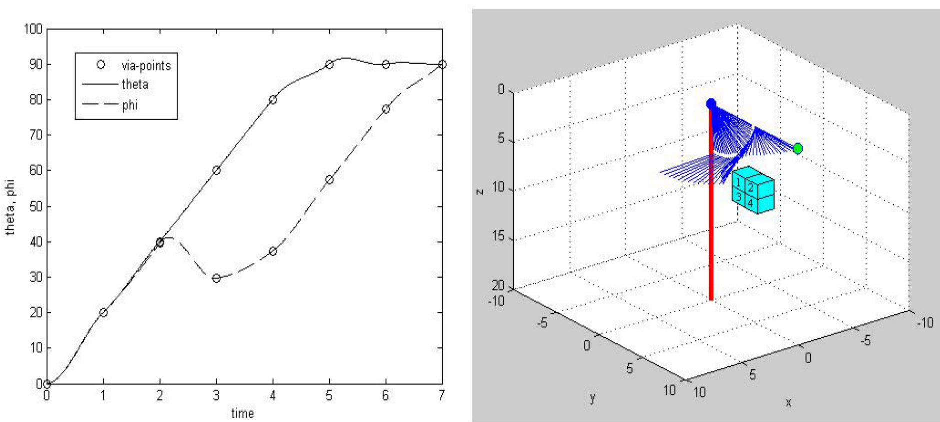

Fig. 2. Joint space trajectory plot

6.2 Optimizer Parameters

For the performance analysis of the three algorithms we use 10 particles to scourge the problem space. The parameters for the various optimization algorithms are described below.

Lbest PSO. Local neighborhoods each of size *d=5* are formed at each generation by grouping together particles with minimum Euclidean distance between them. The inertial coefficient has been made to decrease linearly over the iterations from a value of 0.9 to 0.4. The scaling factors for the social and cognitive components in the velocity update equation have been chosen to be equal to 2. The maximum velocity for each particle has been set equal to x_{max}.

IWO. The initial weed population has been chosen equal to 5. The maximum and minimum seed counts for each generation have been set equal to 5 and 0 respectively. The modulation index is set to 3. The initial and final values of the standard deviations have been taken to be equal to 10% and 0.004% of the search range respectively.

DE. The scaling factor used for the generation of donor chromosome is 0.8. After mutation, recombination takes place. The Crossover Constant used for the generation of the trial offspring vector was taken to be equal to 0.96.

6.3 Performance Evaluation

Each of the algorithms was run 10 times and the average number of fitness function evaluations required to attain the optimal set of parameters (within specified error limits) were evaluated. The following graphs illustrate the performance of the algorithms to track the optimal set of via points and trajectory respectively.

Table 2. Total Number of Fitness Function Evaluations Throughout Entire Journey

Description	Lbest PSO	DE	IWO
Via-Point Determination	**6099**	64142	36159
Trajectory Generation	**7850**	62228	17632

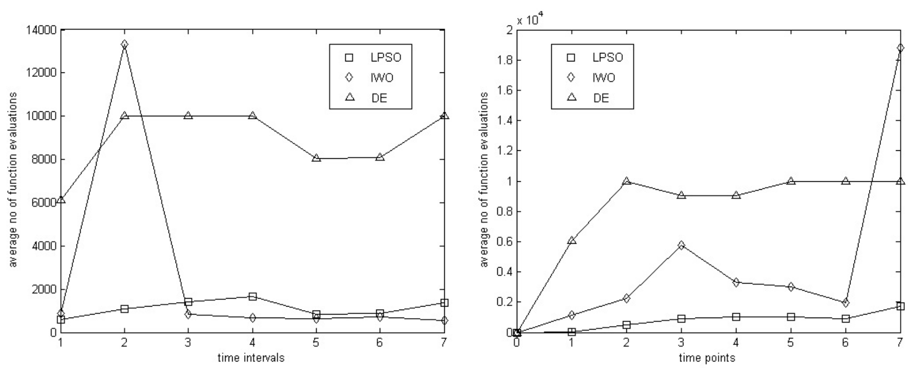

Fig. 3. Comparison of optimizer performances

7 Conclusions

The algorithm determines a set of potential via points for obstacle avoidance. The obtained trajectory ensures smooth motion of the end-effector and optimizes energy expenditure during motion from one via point to the next. The simulation results clearly indicate that swarm intelligence is a far better approach for optimization in this scenario and may be utilized as an effective optimization tool for evolutionary robotics.

References

1. Saab, Y., VanPutte, M.: Shortest Path Planning on Topographical Maps. IEEE Transactions on Systems, Man, and Cybernetics–Part A 29(1), 139–150 (1999)
2. Gilbert, E.G., Johnson, D.E.: Distance Function and Their Application to Robot Path Planning in the Presence of Obstacles. IEEE J. of Robotics and Automation RA-1(1) (1985)
3. Tian, L., Collins, C.: An Effective Robot Trajectory Planning Using a Genetic Algorithm. Elsevier, Mechatronics 14, 455–470 (2004)
4. Zalzala, A.M.S., Chan, K.K.: An Evolutionary Solution for the Control of Mechanical Arms. In: Proceedings of ICARCV 1994, Singapore (1994)
5. Pack, D., Toussaint, G., Haupt, R.: Robot Trajectory Planning Using a Genetic Algorithm. In: SPIE 1996, vol. 2824, pp. 171–182 (1996)
6. Mao, Z., Hsia, T.C.: Obstacle Avoidance Inverse Kinematics Solution of Redundant Robots by Neural Networks. Robotica 15, 3–10 (1997)
7. Khatib, O.: Real-time Obstacle Avoidance for Manipulators and Mobile Manipulators. Int. J. of Rob. Res. 5(1), 90–98 (1986)
8. Mehrabian, A.R., Lucas, C.: A Novel Numerical Optimization Algorithm Inspired from Weed Colonization. Ecological Informatics 1, 355–366 (2006)
9. Konar, A., Das, S.: Recent Advances in Evolutionary Search and Optimization Algorithms. In: NGMS 2006, BESU, Shibpur, Howrah, India, January 11-13 (1996)
10. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)