Monograph:

Rana, A.S. and Zalzala, A.M.S. (1995) An Evolutionary Algorithm for the Collision Free Motion of Multi-Arm Robots. Research Report. ACSE Research Report 570 . Department of Automatic Control and Systems Engineering

RERVM COGNOSCERE CAVSAS

Department of
AUTOMATIC
CONTROL
and SYSTEMS
ENGINEERING
UNIVERSITY of SHEFFIELD

AN EVOLUTIONARY ALGORITHM FOR THE COLLISION FREE MOTION OF MULTI-ARM ROBOTS

A. S. RANA and A.M.S. ZALZALA

Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom

Research Report #570
19 April 1995

RRG
Tel : +44 (0)114 2825250
Fax : +44 (0)114 2731729
EMail : rrg@sheffield.ac.uk
Robotics Research Group

An Evolutionary Algorithm for the Collision Free Motion of Multi-arm Robots

A. S. Rana and A. M. S. Zalzala

Robotics Research Group
Dept. of Automatic Control and Systems Engineering
University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom.
email: rrg@sheffield.ac.uk

Abstract

This paper presents an evolutionary algorithm for the collision-free path planning of multi-arm robots. A global path planning technique is used where the paths are represented by a string of via-points which the robots have to pass through. The path planning algorithm uses evolutionary techniques to minimise path lengths, uneven distribution of via-points on the paths (decrease variation in velocity) and minimise the collision between the robots. Two particular features of this algorithm are that via-points are not limited to a fixed grid over the configuration space, and customised operators are used in the GA-based search to improve the performance. Simulation results are presented for collision-free paths planned for two planner arms and then for two 3-DOF PUMA®-like arms moving in three dimensional operational space.

Keywords: Multi-arm Robots, Genetic Algorithms, Collision-Free Motion Planning.

1. Introduction

The problem of path planning of multi-arm robots is different from that of single arm robots in that one arm may act as an obstacle to another arm. Most of the research in the field of robot path planning has been centred on avoiding collision with static obstacles [12,14]. Dynamically changing environments have been considered also [10], but simple problems have been addressed, e.g. path planning of mobile robots amongst moving obstacles. Path planning of articulated arms amongst moving obstacles becomes more complex due to the kinematic structure of the robotic arm.

Motion planning of multi-arm robots has received a lot of attention by researchers [3,15,17], but in most of the cases, assumptions have been made which make these planners impractical for application in actual practice, e.g. restricting the collision avoidance between end effectors only, considering elliptical paths for the end effectors etc. Koga and Latombe [13] report experiments in dual arm manipulator problems. They suggest algorithms based on searching over a grid in the workspace and randomised search techniques. While the former give better performance for simpler robots the latter are better for robots with higher DOF (degrees of freedom).

It has been shown that randomised methods when incorporated into conventional search techniques give better results for high DOF robots in a complex environment [2] for getting the robot out of local minima. Since GA's are guided random search techniques, they have been used to plan paths for robots by many researchers giving better results in certain aspects [5,7,9]. These algorithms perform the search for a path over a grid defined over the work-space of the robots. Chen and Zalzala [6] define a grid over the work-space of a mobile robot, where each grid point is given a certain potential depending upon the distance from obstacles. The search algorithm then tries to minimise the total potential field over the path which the robot takes. Local search has been used in addition to global search to improve the performance of the algorithm. These algorithms only give the path the robot would follow and do not provide information about time-history of robot on this path. Even though this time-history can be evaluated, it is not readily available for direct use by the algorithm. Such information is necessary for path planning of multiple agents, so that collision between them can be avoided. Hence these algorithms are not readily applicable to the path planning of multiple robots.

Shibita and Fukuda [18] have used GA's for motion planning of multiple mobile robots.

The planner proposed by them carries the search over a static visibility map. Paths are planned for the mobile robots, and the time which the robots have to wait for another robot to pass out of their way to avoid collision is incorporated into the fitness function of GA.

A planner has been proposed in this paper which uses a GA-based evolutionary algorithm for path planning of two 3-DOF PUMA®-like arms moving in three dimensional space. The problem is formulated in configuration space while ensuring that the information about the position of the robot on a path at a particular time instant is expressed implicitly, so that it can be used directly by the evolutionary algorithm for collision avoidance among the moving arms.

Path planning in configuration space has the advantage that the problem is reduced that of a point object, and hence complexities arising from considering the motion of an articulated chain of links are taken into account. It is, however, difficult in configuration space to determine the direction in which to move in order to avoid collision. Numerical methods have been suggested for the construction of potential fields in configuration space [1], and hill climbing techniques have been used to find a collision free path for the robot by following the gradient of these potential fields. Nonetheless, these methods deal with path planning in presence of static obstacles only, where the field has to be calculated only once. They are not applicable to moving obstacles, in which the field would change with the motion of the obstacles. Even though forming potential fields in presence of moving obstacles is difficult, it is quite easy to just determine whether the object whose path is being planned has collided with an obstacle or not. GA's do not need information about which way to move in the domain of a function in order to achieve its maxima or minima. This fact has been used in the algorithm reported in this paper by incorporating the collision of the moving robots in the fitness function of the GA.

This paper is organised as follows. In section 2, the problem of path planning of two planner robotic arms is formulated. In section 3, an evolutionary algorithm for these robots is presented, while section 4 extends the algorithm for the case of two 3-DOF PUMA®-like robotic arms moving in 3-D operational space. Conclusions are given in section five.

2. Problem Formulation for Two Planner Arms

Two planar 2-DOF robots are considered first. The operational space for these arms is shown in Figure 1. The configuration space for $\mathcal{A}_1$ (robot #1) is $C_1=T_1\times T_2\in\Re^2$ and that for $\mathcal{A}_2$ (robot #2) is $C_2=\alpha_1\times\alpha_2\in\Re^2$. The configurations spaces are considered to be decoupled, so that $\mathcal{A}_2$ appears as an obstacle $C\mathcal{A}_2\subset C_1$ to $\mathcal{A}_1$ and $\mathcal{A}_1$ appears as an obstacle $C\mathcal{A}_1\subset C_2$ to $\mathcal{A}_2$.

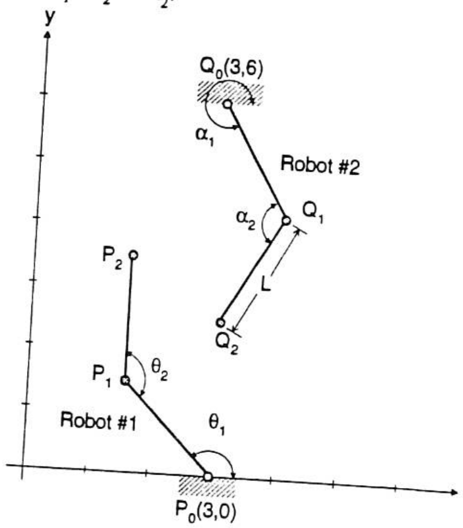


Figure 1. Operational space for two planner arms with two links

The paths for $\mathcal{A}_1$ and $\mathcal{A}_2$ are considered to be strings of via-points in C_1 and C_2, respectively, and the robots have to move through these via-points in order to reach their respective goal positions. Figure 2 shows a path in configuration space represented as via-points. Corresponding via-points on path #1 and path #2 represent the configurations in which $\mathcal{A}_1$ and $\mathcal{A}_2$ would be at the same instant of time.

To find out whether the arms are colliding with each other or not, the links are approximated with touching or overlapping circles (or with spheres in three dimensional work-space). The size, number and location of these circles for a particular size of link and degree of accuracy has been discussed in [4]. The distances between the centres of all of the circles on both arms are determined. If the distance between the centres of any two spheres is less than the sum of the two radii, then collision occurs.

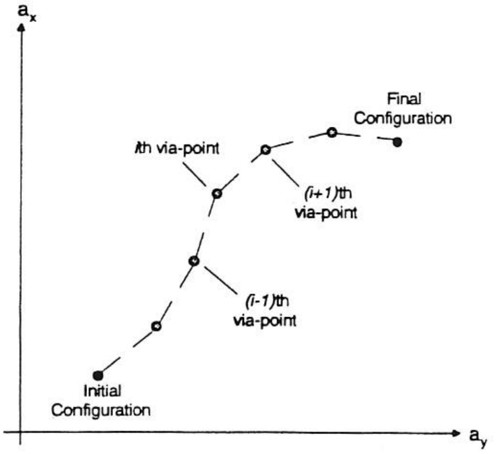


Figure 2. Via-points in the configuration space $C = a_x \times a_y \in \Re^2$ where a_s and a_y are joint space variables of a robot

Figure 3 shows the links of the robot approximated by touching circles. As the robot links are considered to be lines, the radii of the circles (or spheres) would determine the margin of safety in terms of collision between the links.

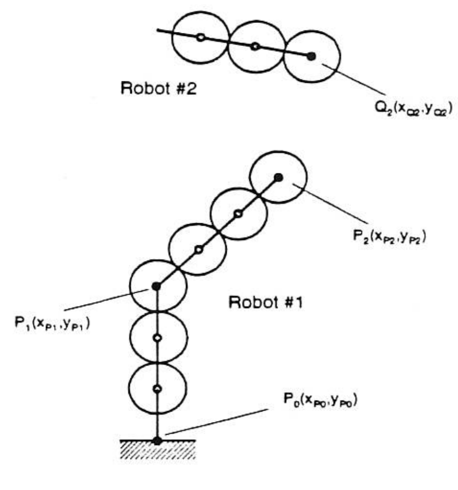


Figure 3. Robots approximated by touching circles

3. Evolutionary Algorithm for Motion Planning of Two Planner Arms

A GA-based search technique needs (i) the problem to be encoded as strings and (ii) formulation of a fitness function for these strings on the basis of which a guided random search would be performed.

3.1 Encoding of Paths as Strings

The paths are encoded directly as strings to be used as

$$\begin{array}{l} p_1 \vdots p_2 \vdots \cdots \vdots p_{i-1} \vdots p_i \vdots p_{i+1} \vdots \cdots \vdots p_{N-1} \vdots p_N \\ \quad \vdots q_1 \vdots q_2 \vdots \cdots \vdots q_{i-1} \vdots q_i \vdots q_{i+1} \vdots \cdots \vdots q_{N-1} \vdots q_N \end{array} \quad (1)$$

where $p_i \in C_1$ is the *i*th via-point on path #1 and $q_i \in C_2$ is the *i*th via-point on path #2.

3.2 Fitness Function

The fitness function puts a penalty on path lengths of both the robots in configuration space, uneven distribution of via-points on these paths so that the variation in velocity is minimised and finally the collision between the robots at corresponding time instants. Different components of the fitness functions are defined as follows:

a. Length of path #1 given by

$$C_1 = \sum_{i=1}^{N} \left(\sqrt{\left(x_{p_i} - x_{p_{i-1}}\right)^2 + \left(y_{p_i} - y_{p_{i-1}}\right)^2} \right) \quad (2)$$

where N is the total number of via-points in and $p_i = \left(x_{p_i}, y_{p_i}\right)$ is the *i*th via-point on path #1.

b. Length of path #2 given by

$$C_2 = \sum_{i=1}^{N} \left(\sqrt{\left(x_{q_i} - x_{q_{i-1}}\right)^2 + \left(y_{q_i} - y_{q_{i-1}}\right)^2} \right) \quad (3)$$

where N is the total number of via-points in and $q_i = (x_{qi}, y_{qi})$ is the *i*th via-point on path #2.

c. Measure for uneven distribution of via-points on path #1 given by

$$C_3 = \sum_{i=1}^{N} \left\| d_i - d \right\| \quad (4)$$

where d_i=distance between *i*th and *(i-1)*th via-point on path #1, and $d = \frac{1}{N}\sum_{i=1}^{N} d_i$

d. Measure for uneven distribution of via-points on path #2 given by

$$C_4 = \sum_{i=1}^{N} \left\| e_i - e \right\| \quad (5)$$

where e_i=distance between *i*th and *(i-1)*th via-point on path #2 and $e = \frac{1}{N}\sum_{i=1}^{N} e_i$

e. Collision penalty depending on collision between R_1 and R_2 in corresponding via-points on both paths, given by

$$C_5 = \sum_{i=1}^{N} C_i \qquad (6)$$

where

$$C_i = \begin{cases} 1 & \text{if } R_1 \text{ and } R_2 \text{ collide in } i\text{th configuration} \\ 0 & \text{otherwise} \end{cases}$$

Thus, the fitness function is thus given by

$$F = C_{\max} - k_1(C_1 + C_2) + k_2(C_3 + C_4) + k_3 C_5 \quad (7)$$

where C_{max}, k_1, k_2 and k_3 are constants. The value of k_3 is kept relatively higher, since collisions are to be avoided at all costs. The value of k_2 on the other hand is kept very low, since the via-points are redistributed evenly by one of the operators in the algorithm.

3.3 Generation of Initial Population

An initial population of strings is generated randomly for the evolutionary algorithm to start with. Each string is generated by starting from initial position of the robot in the configuration space, and iteratively generating the next via-point randomly in the general direction of the goal until a required number of via-points has been generated in the string. In this procedure, next via-point is generated by first generating a vector of fixed length in the direction of the goal position, and then adding or subtracting a randomly generated number to the angle of this vector in polar representation. When this vector is added to the current position from which the next via-point is being generated, it ensures that this position is random but at the same time is in the general direction of the goal position.

Via-points in paths in the population generated in this way are quite random and non uniformly distributed. They are then iteratively 'redistributed' in the following manner.

If $\mathcal{C} = a_x \times a_y \in \Re^2$ is the configuration space in which the path lies and if $a_x = \{a_{0x}, a_{1x}, \ldots, a_{Nx}\}$, $a_y = \{a_{0y}, a_{1y}, \ldots, a_{Ny}\}$ and the path is given by one-to-one correspondence between a_x and a_y, then both a_x and a_y can be expressed as a parametric functions of time. The time varies linearly with the index number of elements of a_x and a_y. Hence scaled time can be taken to be equal to the index number. Cubic splines are fitted to a_x and a_y as function of time as given in [8], and via-points are then distributed evenly over these splines at equal intervals of distance.

The path is then 'relaxed' by moving each via-point through a certain step in the direction of a point bisecting the line between the neighbouring via-points, i.e.

For i=1 to i=N-1 step 1

{

$$a_{ix} = a_{ix} + delta \times \left\{ \frac{a_{(i-1)x} + a_{(i+1)x}}{2} - a_{ix} \right\} \quad (8)$$

$$a_{iy} = a_{iy} + delta \times \left\{ \frac{a_{(i-1)y} + a_{(i+1)y}}{2} - a_{iy} \right\} \quad (9)$$

}

where *delta* is a positive real number less than 1. Effectively, relaxation makes each path act as a stretched elastic string [16].

3.4 The GA Operators

The following four operators are used in the evolutionary algorithm:

a. Reproduction: The strings are reproduced for next generation based on their fitness function. Weighted roulette wheel [11] is used to select the strings from an old population for a new population. A total of *keep_best* trajectories with best fitness are passed on to the new generation.

b. Cross-over: The strings in the new population are randomly grouped together into pairs, and double cross-over is performed on these pairs. One cross-over sight is chosen in the portion of the string representing path of $\mathcal{A}_1$ and the second cross-over sight is chosen in the portion representing path of $\mathcal{A}_2$ (see equation (1)). Cross-over is performed only if the distance between the chosen sites is less than a certain constant *cross_over_distance*.

c. Path Redistribution/ Relaxation: After cross-over, there may be large position jumps in the off-spring strings at the sites of cross-over. The via-points on these strings are then 'redistributed' evenly over the whole path and then 'relaxed' as described in section 3.3.

d. Mutation: New trajectories are generated as described in section 3.3 and randomly replaced with trajectories in the new population with a probability of *mutation_probability*. The number of new_trajectories generated in each generation is *new_trajectories*.

3.5 Simulation Results for Two Planar Arms

The algorithm was simulated with parameters given in Table 1, and paths were planned considering the initial and final configurations given in Table 2.

Table 1. Values of parameters used in GA-based search algorithm

k_1	0.5
k_2	0.005
k_3	100
C_{max}	500
population sizes	50
new_trajectories	5
keep_best	2
cross_over_distance	100°
cross_over_probability	0.9
mutation_probability	1.0

Table 2. Initial and Final Configurations for Robot #1 and Robot #2.

	Initial Value	Final Value
θ_1	45°	13°
θ_2	225°	135°
α_1	225°	315°
α_1	225°	315°

Figure 4 shows snapshots of the robots' motion at different intervals. The positions at which the robots seem to overlapping are infact occuring at different time instants. So collision is infact not occuring between them. Figure 5 shows the time history of average fitness and best fitness in the population. Figure 6 shows the paths in configuration space. Figure 7 shows the minimum distance between the two robots at different instants of time. The safety distance was considered to be 1 unit by taking the radii of circles equal to 0.5 unit. in this case. It can be seen that minimum distance does not go below this safety distance at any time.

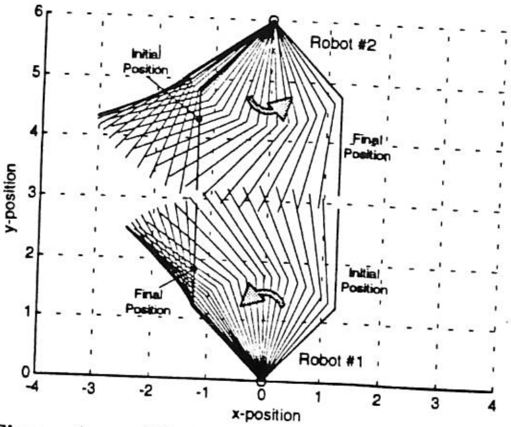


Figure 4. Motion of the planner arms in operational space

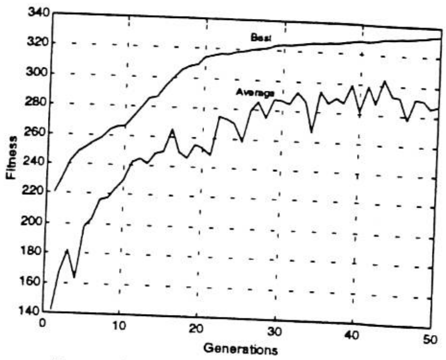


Figure 5. Time history of fitness function for planner arms

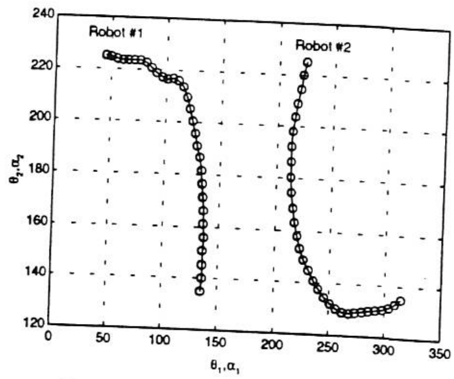


Figure 6. Paths of planner robots in configuration space.

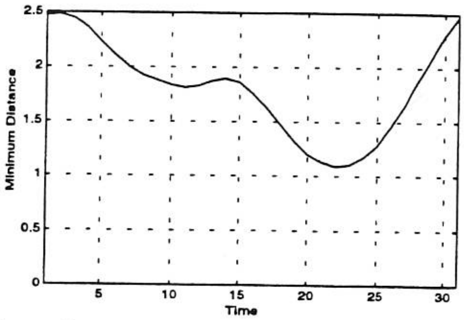
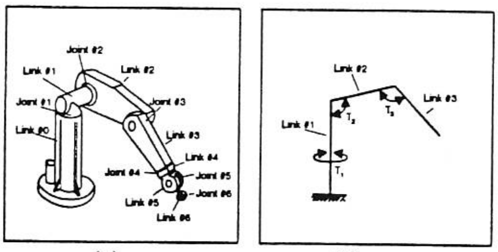


Figure 7. Minimum distance between the two robots.

4. Algorithm Formulation for Two 3-DOF PUMA®-Like Arms Moving in 3-D Operational Space

The path planning algorithm for planner robots discussed in section 3 is extended here for 3-D case considering two PUMA®-like robots. The PUMA®-560 series robot arm is a revolute (RRR) arm with six degrees of freedom. The first three degrees are used to take the wrist to a particular position, while the latter three degrees are used to orient the wrist in a desired configuration. Hence, it is the first three degrees of freedom which normally cause the collision. Therefore only the motion planning of first three links is considered for collision avoidance.

The PUMA®-560 arm is shown in Figure 8(a). Figure 8(b) shows approximation of this robot by line segments.



(a) (b)

Figure 8. (a) a PUMA®-560 series robot and (b) its approximation by line segments.

The problem is formulated in the same way as that for two planner arms, although appropriate changes are intorduced to incorporate the additional third dimension of the operational space.

4.1 Kinematics

The two robots are shown in Figure 9. Position of the base of first robot $\vec{p}_0$ is assumed to be at the origin. Positions of points $\vec{q}_0$, $\vec{p}_1$ and $\vec{q}_1$ remain fixed at (0,X,0), (0,0,L*) and (0,X,L*), where X is the distance between the bases of the two robots and L* is the length of the first link. Values of $\vec{p}_2$, $\vec{p}_3$, $\vec{q}_2$ and $\vec{q}_3$ are calculated as

$$
\begin{aligned}
\vec{p}_2 &= \underline{A_1(\theta_1)A_2(\theta_2)}\vec{p}_0 \\
\vec{p}_3 &= \underline{A_1(\theta_1)A_2(\theta_2)A_3(\theta_3)}\vec{p}_0 \\
\vec{q}_2 &= \underline{A_0}\,\underline{A_1(\alpha_1)A_2(\alpha_2)}\vec{p}_0 \\
\vec{q}_3 &= \underline{A_0}\,\underline{A_1(\alpha_1)A_2(\alpha_2)A_2(\alpha_3)}\vec{p}_0
\end{aligned}
\tag{10}
$$

where

$$
\underline{A_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & X \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ \underline{A_1}(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ \sin\theta & 0 & -\cos\theta & 0 \\ 0 & 1 & 0 & L^* \\ 0 & 0 & 0 & 1 \end{bmatrix};\ \underline{A_2}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & L\cos\theta \\ \sin\theta & \cos\theta & 0 & L\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

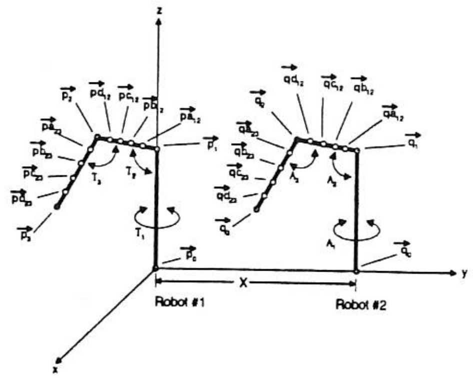


Figure 9. Two PUMA®-like arms sharing a portion of the operational workspace.

Once position of $\vec{p}_2$, $\vec{p}_3$, $\vec{q}_2$ and $\vec{q}_3$ has been determined, the positions of centres of all the touching spheres on the second and third links of both robots can be determined, since they are located at an equal distance from each other in a straight line on the link.

4.2 Simulation Results for 3-DOF Arms

Values of parameters given in Table 3 were used for the algorithm, and the initial and final configurations for the robots are given in Table 4.

Table 3. Values of parameters used in GA-based search algorithm for two 3-DOF arms moving in three dimensional space.

k_1	1.0
k_2	0.25
k_3	10.0
C_{max}	500
population size	100
new_trajectories	10
keep_best	1
cross_over_distance	100°
cross_over_probability	0.9
mutation_probability	1.0

Table 4. Initial and Final Configurations for Robot #1 and Robot #2.

	Initial Value	Final Value
θ_1	15°	150°
θ_2	-15°	-15°
θ_3	-30°	30°
α_1	-150°	-15°
α_2	-15°	-15°
α_3	30°	-30°

Figure 11 shows the motion of the two arms in operational space and Figure 12 (a) and (b) show the time history of fitness function and the paths in operational space, respectively. Figure 13 gives the minimum distance between the robots. In this case, the safety distance for the robots was 0.25 units, considering a collision spheres' radii to be equal to 0.125 units.

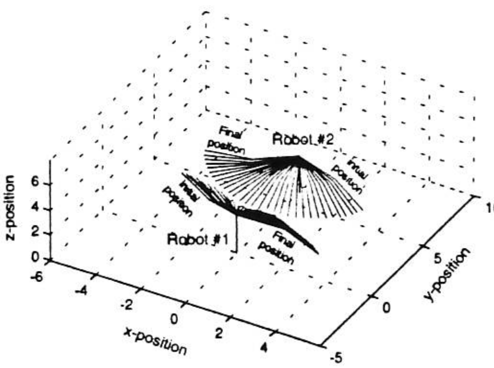
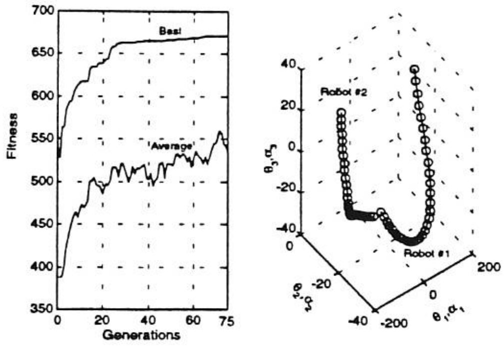


Figure 11. Motion of two 3-DOF arms in operational space



(a) (b)

Figure 12. (a) Time history of fitness function and (b)Paths in configuration space.

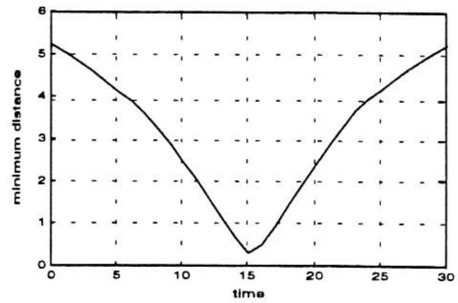


Figure 13. Minimum distance between the two robots.

5. Conclusions

This paper proposes a new strategy for path planning of multi-arm robotic systems based on an evolutionary algorithm. Distinct features in this algorithm include that the path's via-points are not fixed to lie on a particular grid and strings use floating point representation. In addition, customized operators have been intorduced (namely redistribution of via-points on the path and relaxation of the path). Hence, the incorporation of elastic relaxation of paths into GA operators give the planning algorithm an evolutionary structure, and improve its performance.

The algorithm has shown good results for motion planning of two PUMA®-like robotic arms. Even though at this stage, collision free motion of two arms is considered only, the algorithm can easily be extended to include static obstacles in the environment. Nonetheless, the algorithm is not limited to PUMA®-like configurations only, and can be applied to global path planning of any multi-arm robot system.

6. References

[1] Barraquand, J., Langlois, Latombe, J. C. "Numerical Potential Field Function Techniques for Robot Path Planning", Report No. STAN-CS-89-1285, Dept. of Computer Science, Stanford University, Oct. 1989.

[2] Barraquand, J. and Latombe, J. C. "A Monte Carlo Algorithm for Path Planning with Many Degrees of Freedom" Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1712-1717, 1990.

[3] Basta, R. A., Mehrotra, R., Varanasi, M. R. "Detecting and Avoiding Collision Between Robot Arms in a Common Workspace", Robot Control Theory and Applications, Copyright 1988 Institute of Electrical Engineers.

[4] Beaumont, R. G. and Crowder, R. M. "Two-Armed Robot Systems- A Review of Current Theory and the Development of Algorithms for Real-Time Collision Avoidance", IEE Colloquium on 'Controllers for Robotic Applications- Concepts and Implementations', Digest No. 127, London, UK, 1989.

[5] Zalzala, A.M.S. and Chan, K.K., "An evolutionary solution for the control of mechanical arms", In *Proc. 3rd Int. Conf. Automation, Robotics and Computer Vision (ICARCV)*, Vol.2, pp. 711-5, Singapore, 1994.

[6] Chen, M and Zalzala, A. M. S. "A Genetic Algorithm Approach to Multi-Criteria Position and Configuration Optimisation of Redundant Mobile Manipulator Systems", In *Proc. Int. Workshop on Advanced Robotics & Intelligent Machines*, University of Salford, 1995.

[7] Cleghorn, T. F., Baffes, P. T., Wang, L. "Robot Path Planning Using a Genetic Algorithm", Second Annual Workshop on Space Operation and Robotics, Vol. 3019, Ch. 44, pp.383-389, 1988.

[8] Craig, J. J. "Cubic Polynomials for a Path with Via-points", in 'Introduction to Robotics: Mechanics and Control (pp. 232-236)', Addison Wesley Publishing Company, Inc., 1989.

[9] Davidor, Y, "A Genetic Algorithm Applied to Robot Trajectory Generation", Handbook of Genetic Algorithms (Ed. L. Davis), pp. 144-165, Van Nostranes Reinhold, New York, 1991.

[10] Fujimura, K. "Motion Planning in Dynamic Environment", Computer Science Workbence (Ed. T. L. Kunii), Springer Verlag, Tokyo, 1991.

[11] Goldberg, D. E. "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley Publishing Company Inc., 1989.

[12] Hwang, Y. K. and Ahuja, N. "Gross Motion Planning- A Survey" ACM Computing Surveys, Vol. 24, No. 3, Sept. 1992.

[13] Koga, Y. and Latombe, J. C. "Experiments in Dual Arm Manipulation Planning", IEEE International Conference on Robotics and Automation, Nice, France, 1992.

[14] Latombe, J. C. "Robot Motion Planning", Kluwer Academic Publishers, Amsterdam, 1991.

[15] Lee, B. H. and Lee, C. S. G. "Collision Free Motion Planning of Two Robots" IEEE Transactions on Systems, Man and Cybernetics, Vol. SMV-17, 1987.

[16] Quinlan, S. and Khatib, O. "Elastic Bands: Connecting Path Planning and Control", Proceedings of IEEE International Conference on Robotics and Automation, pp. 802-807, 1993.

[17] Roach, J. W. and Boaz, M. N. "Co-ordinating the Motion of Robot Arms in a Common Workspace" IEEE Journal of Robotics and Automation, Vol. RA-3, No. 5, Oct. 1987.

[18] Shibata, T. and Fukuda, T. "Co-ordination in Evolutionary Multi-Agent Robotic Systems Using Fuzzy and Genetic Algorithms", Control Engineering Practice, Vol. 2, No. 1, p. 103-111, 1994.