Coordinated motion planning of manipulators by evolution strategies

M.A. Ridao[a], J. Riquelme[b], E.F. Camacho[a], M.Toro[b]

[a]*Departamento Ingeniería de Sistemas y Automática, Universidad de Sevilla, Spain*

[b]*Departamento Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Spain*

Abstract

A method for obtaining coordinated motion plans of manipulator robots is presented. This planning can be easily implemented by programs written in any industrial robot programming language, such as VAL II. The generated programs minimize the total motion time of the robots along their paths, with some constraints directed at avoiding collision between the robots. A method based on Evolution Strategies is used for the optimization.

1 Introduction

Finding collision-free motion planning algorithms has been the focus of research in many works [1]-[4]. The problem is to plan a collision-free motion (obstacles and other robots), from an initial configuration to a goal configuration. The algorithm presented in this paper is based on a Decoupled Planning approach, that is, the path of each robot is planned independently of the others, and afterwards it is necessary to consider the interactions among the paths. These algorithms reduce the computational complexity considerably but this gain results in a loss of completeness. Different methods have been presented to solve the interactions among the paths [2]-[4].

In this paper, a method is presented to minimize the total motion time of the robots along their paths, avoiding collision regardless of the model used. The method uses the Coordination Diagram concept and obtains trajectories that can easily be implemented on any industrial robot programming language. Selection of the model is not a critical aspect, because a collision will never be produced, even with a non accurate model. However, as the algorithm tries to minimize the total motion time, precise models are required to obtain optimum results.

Evolutionary Computation (EC) is a global optimization method based on biological observations [5]: the means of natural evolution (natural selection and natural genetics) and the survival of the fittest. In the problem presented in this paper, the use of Evolution Strategies (ES) [6], one of the pathways characterizing EC, with a codification appropiate to the problem has been used, permitting the implementation of the goal function, search strategies and constraints to the solutions in a natural way.

2 Problem statement

The use of a Decoupled Planning approach needs a fixed obstacle collision-free path for each of the robots to be previously obtained. These paths are obtained by considering only the fixed obstacles and not taking into account the other robots. Let P^1 and P^2 denote the two collision-free paths generated for each of the manipulators. Let λ^1_{max} and λ^2_{max} denote the length of both paths. The P^1P^2-*space* is defined as the following $\mathbb{R}^2$ region:

$$P^1P^2\text{-}space=\{(\lambda^1,\lambda^2)\ /\ 0\leq\lambda^1\leq\lambda^1_{max} \text{ and } 0\leq\lambda^2\leq\lambda^2_{max}\}$$

Any continuous path from (0,0) to $(\lambda^1_{max},\lambda^2_{max})$ determines a coordinated execution of the two paths, and is called a *Coordination Path* (*CP*).

Let the *Collision Region* (CR) be defined as the set of points in the P^1P^2-space where a collision between the two manipulators is produced. If a Coordination Path does not cross through the Collision Region, it is called a Collision-Free CP.

In order to reduce the search space in the P^1P^2-space, a discretization of each path has to be made, so the path is divided into several equal intervals. Let's number the intervals of both paths from 0 to max_1 and 0 to max_2 respectively. Let δ^i_k be the k-th interval along the path of the manipulator i. Now, a path can be defined as a set of intervals, that is $\Omega_i=\{\delta^i_k\ /\ 0\leq k\leq max_i\}$

A cell is defined as a pair (δ^1_m,δ^2_n) where $\delta^1_m\in\Omega_1$ and $\delta^2_n\in\Omega_2$. For simplicity, a cell will be represented as (m,n) and $\Omega_i=\{k\ /\ 0\leq k\leq max_1\}$. So, the P^1P^2-space is transformed into an array of cells, the *Coordination Diagram* (CD).

A cell $(m,n)=\delta^1_m\ x\ \delta^2_n$ is considered collision FREE if $\forall\ \lambda^1\in\delta^1_m,\ \lambda^2\in\delta^2_n\Rightarrow(\lambda^1,\lambda^2)\notin$ CR. Otherwise, it is considered an OBSTACLE cell.

In order to implement a trajectory in the CD, the motion of both robots must be syncronized. Robots are syncronized by using Syncronization Points in this paper. A *Syncronization Point* (SP) is a point in the CD, which any Coordination Path will necessarily pass through. A SP defines a spot on the path of each robot. When the robot arrives at that place on its path, it will stop until the other robot arrives at its respective point. Once both robots are at these points, motion can continue.

A collision-free coordinated motion of two robots can be found by searching for a *Syncronization Point Sequence* that minimizes the total coordinated motion time. Let's consider a rectangle formed by free cells in the CD and let's consider the motion of the robots from the lower left corner cell to the upper right corner cell. Any trajectory defined for each robot between these two points in the CD will be a collision-free CP. This class of rectangles is going to be called *Free Rectangles*. Let's consider a set of Free Rectangles, connected as can be seen in Figure 1. This set of rectangles is a *Free Rectangle Sequence*, and every intersection points between two rectangles will be a SP.

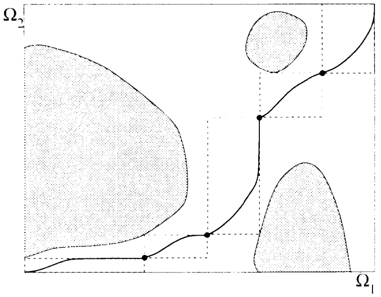


Figure 1. A Free Rectangles Sequence.

Given a Free Rectangle Sequence, any CP constrained to pass over every SP of the sequence will be a collision-free CP.

3 The proposed algorithm

ES are basically search algorithms that start with some population of structures, called individuals, that are initial solutions to the problem, and repeatedly perform the following cycle of operations until a termination condition is satisfied: Evaluation, Selection, Reproduction and Replacement. Next paragraphs show the main aspects that define the application of ES.

Chromosomic representation of the individual

Each individual will represent an increasing SP Sequence. The solution adopted here is to consider fixed length chromosomes, that is, all the sequences have the same number of SPs, but two or more points having the same coordinates are allowed. These types of SPs are called *Multiple Points*. The length of the chromosome defines the maximum number of SPs that any sequence can have.

Let N be the length of the chromosome. A SP Sequence will be considered to be determined by $N+2$ points, where $(x_0,y_0) = (0,0)$ and $(x_{N+1},y_{N+1}) = (max_1,max_2)$ and with:

$$\{ \ P_i=(x_i,y_i) \in \Omega_1 \ x \ \Omega_2 \ / \ \forall \ i, \ x_i \leq x_{i+1} \ and \ y_i \leq y_{i+1} \ \} \quad 0 \leq i \leq N$$

where x_i, y_i are points corresponding to the paths of the first and second robot, respectively.

Transactions on Information and Communications Technologies vol 8, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

An individual is *valid* if it forms an increasing Sequence of Free Rectangles, and so, a Colission-Free CP, otherwise it is a *non-valid* individual. Finally, when a genetic operator is applied, it is possible for a non-increasing SP Sequence to be obtained. It is not strictly an individual, but in this paper, it will be called a *non-acceptable individual.*

Individuals in the Initial Population

The initial population is selected randomly when using Evolution Strategies, however, it seems evident that this individuals must be acceptables ones. The proposed form of creating an initial population consist of generating two sets of N random values from [0,1], arrange them in ascending order and project them on each axis. This way of creating initial populations does not secure the validity of the individuals.

Fitness measure

For valid individuals, the fitness function measures the total execution time needed by the robots to complete their paths, when the SPs are placed in the positions defined by the individual specifications.

The fitness function for non-valid individuals must measure how far it is from a valid individual. Obviously, the fitness value for this kind of individual must be higher than any valid individual value. The function considered is $f(N)=K+nco$, where K is a high value in respect to the value associated to the valid individuals, and nco is the number of Obstacle Cells inside the rectangle sequence.

Proposed Crossover operators

Rectangular crossover The crossover of two individuals $S^1=\{(x_i^1,y_i^1) \ / \ 1\leq i\leq N\}$ and $S^2=\{(x_i^2,y_i^2) \ / \ 1\leq i\leq N\}$ produces a child $H=\{(hx_i,hy_i)\}$ so that hx_i is an integer random value in the interval $[x_i^1,x_i^2]$ and hy_i is an integer random value in the interval $[y_i^1,y_i^2]$. As can be seen in Figure 2, the SPs of the child must be placed over the grey rectangles.

Segment crossover The crossover of two individuals S^1 and S^2 as defined above, produces a child $H=\{(hx_i,hy_i)\}$ so that hx_i is a random value in the segment defined between (x_i^1,y_i^1) and (x_i^2,y_i^2) for each i.

Extended Segment crossover Each point (hx_i,hy_i) is a random value in an extended segment. It is also considered to be the prolongation of the segment connecting the points (x_i^1,y_i^1) and (x_i^2,y_i^2) for a fixed percentage of the total length (See Figure 2).

Simple crossover An integer value k is chosen randomly verifying $1\leq k\leq N$, and two new individuals are formed: H^1 with the first k points of S^1 and the remaining ones of S^2. Also, H^2 is formed with the first k points of S^2 and the rest of S^1.

Transactions on Information and Communications Technologies vol 8, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

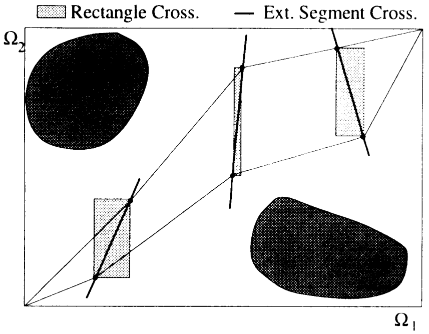


Figure 2: Rectangular and Extended Segment Crossover.

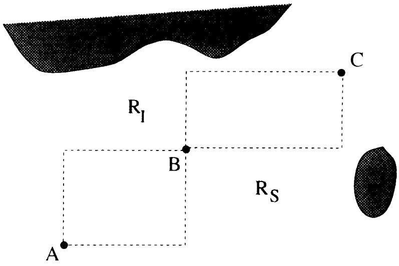


Figure 3: Redundant Control Points.

Proposed mutation operators

Proportional mutation Given an individual $S=\{(x_i,y_i)\ /\ 1\leq i\leq N\}$, an integer value k is radomly chosen so that $1\leq k\leq N$. Then, the coordinates of the point (x_k,y_k) will be changed by new values, in such a way that they will be randomly distributed over the rectangle defined by the lower left vertex (x_{k-1},y_{k-1}) and the upper right vertex (x_{k+1},y_{k+1}). Every individual generated by this mutation is a valid one. This mutation has the variation consisting of modify only one of the coordinate of P_k chosen randomly (Single Proportional Mutation).

Fixed limits mutation Given an individual $S=\{(x_i,y_i)\ /\ 1\leq i\leq N\}$, two integer values are randomly chosen, the first one k so that $1\leq k\leq N$, and the second m with $-MUTMAX \leq m \leq MUTMAX$, where $MUTMAX$ is the maximum permitted mutation. Then, this mutation consists of substituting point (x_k,y_k) by (x_k+m,y_k+m). This mutation has three variations:

- Double mutation: this is the result of selecting two different values of m for each coordinate.
- Single mutation: this second possibility is the result of applying the mutation to only one of the coordinates of P_k
- Non uniform mutation: the third variant is obtained when $MUTMAX$, varies as a function of the number of the executed generations.

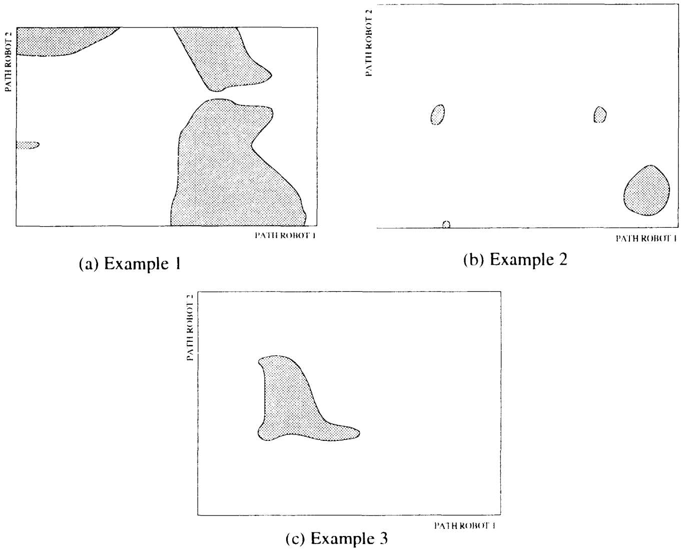
Reduction operator

Given three consecutive SPs A, B and C, as shown in Figure 3, B is a *Redundant Syncronization Point* when A and C are considered, if the R_S and R_I rectangles are free.

When a Redundant Syncronization Point appears in a Sequence, it means that the middle point B can be eliminated, obtaining a new valid sequence with a better fitness function. The use of this specific operator tends to reduce the number of different SP in individuals.

Constraints on the individuals

When these operators are applied, the resulting individuals may not verify the necessary conditions for being an acceptable individual. The action selected in order to resolve these conflicts consist of merging points, that is, modifying the two coordinates of a point in order to avoid the conflict, merging the conflictive point with the following one, in order to form a multiple point.



(a) Example 1

(b) Example 2

(c) Example 3

Figure 4. Coordination Diagrams corresponding to the examples.

4 Application examples

The proposed algorithm has been implemented and applied to several examples in order to study its efficiency, and the comparative behaviour of the different genetic operators described in the previous sections.

The first two examples correspond to the coordinated motion for a two SCARA-type articulated manipulator in two different cases. Figure 4-(a) and 4-(b) represent the respective CD. The example shown in Figure 4-(c) corresponds to the coordinated motions of two PUMA-type manipulators.

The results obtained for these three examples modifying the parameters are described in the following paragraphs. In all the examples, the interval for path discretization is 4 degrees. Examples 1 and 2 have been executed with a population of 100 individuals considering 200 generations, while the population selected for the third example was 25 individuals, also with 200 generations. The selection of the population will follow an elitist model with selection probability proportional to fitness, 20% being the ratio of parents that will be duplicated in the next generation, the rest being offspring obtained by crossover and mutation of these. Typical CPU times on an I.B.M. RISC-6000 320H are

1m30s. for 200 generations and a 100 individual populations and 25s. for populations with 25 individuals.

Tables 1-3 show the average, the standard desviation and the minimum value of the fitness function (an asterisk indicates the global minimum) for different tests. The number of simulations for each test has been 50.

Better results are obtained with greater mutation probability, 0.5 or even 0.8. Quicker convergence results are achieved in example 3 with 0.5.

There are no important differences when the size of the chromosome is substantially increased. In this result, the reduction specific operator plays an important role. As can be seen in Table 1, better results are obtained using the reduction specific operator.

The results with different mutation types are shown in Table 2. As a first conclusion, it can be said that fixed limits mutations obtain better results than proportional types (in the Table pr1 and pr2). Using fixed limits mutations, the best results have been obtained with the single mutation (ra1 in Table 2) and with the non-uniform mutation (ranu). Also, when both coordinates are mutated, better results are obtained by selecting different mutation values for each coordinate (ra2d), than by selecting the same value for both coordinates (ra2i).

Table 1 Time results (in seconds) when the reduction specific operator is used

Ex.	Red	Avg.	σ	Min.
1	yes	5.4604	0.0359	5.4060
	no	5.5406	0.0647	5.4607
2	yes	4.7458	0.0150	4.7421*
	no	4.9110	0.0743	4.7424
3	yes	3.4651	0.0000	3.4651*
	no	3.7495	0.1324	3.5405

Table 3 shows the results with different crossover types. Best results are obtained by using the Simple Crossover.

5 Conclusion

This paper describes a method based on Evolution Strategies to generate programs written in any robot language able to execute the coordinated motion of two robots without collision with the enviroment fixed obstacles or between the robots. The algorithm tries to find a SP Sequence that minimizes the execution motion time. The tests have demonstrated the capability of the proposed evolutionary algorithm to find satisfactory solutions and its efficiency The algorithm has been implemented on two robots, but it can easily be extended to an environment with a greater number of robots.

References

1. Latombe J.C., *Robot Motion Planning*, Klewer Academic Publishers, 1991.

Transactions on Information and Communications Technologies vol 8, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

Table 2 Time result in seconds with different crossover types

Ex.	Type	Avg.	σ	Min.
1	sim.	5.4604	0.0359	5.4060
	rec.	5.4961	0.0246	5.4681
	seg.	5.5068	0.0415	5.4060
	eseg	5.4893	0.0258	5.4735
2	sim	4.7458	0.0150	4.7421*
	rec.	4.7637	0.0314	4.7423
	seg.	4.7722	0.0320	4.7422
	eseg	4.7470	0.0155	4.7423
3	sim.	3.4651	0.0000	3.4651*
	rec.	3.5152	0.0805	3.4651*
	seg.	3.4677	0.0110	3.4651*
	eseg	3.4730	0.0231	3.4651*

Table 3 Time results (in seconds) with different mutation types

Ex.	Type	Avg.	σ	Min.
1	ranu	5.4551	0.0309	5.3930*
	ra1	5.4519	0.0312	5.4060
	ra2d	5.4660	0.0207	5.4060
	ra2i	5.4780	0.0123	5.4735
	pr1	5.4741	0.0220	5.4060
	pr2	5.4782	0.0123	5.4735
2	ranu	4.7422	0.0001	4.7421*
	ra1	4.7422	0.0001	4.7421*
	ra2d	4.7458	0.0151	4.7421*
	ra2i	4.7531	0.0246	4.7421*
	pr1	4.7618	0.0308	4.7422
	pr2	4.7531	0.0246	4.7421*
3	ranu	3.4651	0.0000	3.4651*
	ra1	3.4651	0.0000	3.4651*
	ra2d	3.4651	0.0000	3.4651*
	ra2i	3.5642	0.1060	3.4651*
	pr1	3.5538	0.0787	3.4651*
	pr2	3.5815	0.0974	3.4651*

2. Lee B.H. and Lee C.S.G., Collision-Free Motion Planning of Two Robots, *IEEE Transations on System, Man and Cybernetics*, Jan-Feb 1987, Vol.17, no 1, pp 21-32.
3. O´Donnell P.A. and Lozano-Pérez T. Deadlock-Free and Collision-Free Coordination of Two Robots Manipulators, in ICRA 89, pp 484-489, *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, , 1989.
4. Ridao M.A. and Camacho E.F. A Heuristic Algorithm for Coordinated Motion Planning of Two Manipulators, in EURISCON'94. pp 1089-1096. Málaga (Spain). August 1994.
5. Goldberg D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley 1989.
6. I. Rechemberg, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution.* Frommann-Holzboog Verlag. Stuttgart. 1973.