Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

Application of Pareto-Based Multiobjectives Genetic Algorithm in Minimum Time Motion Planning

Ang Mei Choo
B.Sc (Maths.), M.Sc (Eng. in Control Systems)
Dept. of Industrial Computing, Faculty of Information, Science and Technology
National University of Malaysia, 43600 Bangi, Malaysia.

Dr. A.M.S. Zalzala
BEng, PhD, MIEEE, AMIEE
Dept. of Automatic Control and Systems Engineering
University of Sheffield, S1 3JD, UK

Abstract

This paper looks into the application of Pareto-based Genetic Algorithm (GA) in obtaining the minimum time motion planning for an industrial robot. A common practice in multiobjective optimisation GAs for minimum time motion planning is to apply classical aggregation approach to the objective formulation while in this study, Pareto-ranking method is used. A suitable objective vector is organised to include the total travel time and the two joint constraints: velocity and acceleration limits. The objective is to obtain a optimal motion with minimum travel time and within the kinematics limitations. The optimisation process involves first producing a fixed number of joint displacements using the genetic operators, and then scaling the travel time such that it is not violating the kinematics constraints. The feasibility of this method is shown by simulation results with an RTX SCARA robot. Cubic spline functions are used in the construction of the joint trajectory.

1. Introduction

Minimum time motion planning for a industrial manipulator is an important subject in the area of robotics. However, most multiobjective optimisation GAs used in minimum time motion planning apply classical aggregation approach to the objective formulation (see for e.g., [8, 9, 11]). This paper is intended to develop a Pareto-based multiobjective genetic algorithm, which has recently being proposed [4, 5], for the minimum time motion planning of an RTX robot (with six joints). The travel distance and kinematics characteristics of each joint are varied. Thus, to obtain a feasible minimum motion time, the combination of the kinematics characteristics for their joints such as velocities, accelerations and jerks have to be optimised. The objective vector in this project includes the total travel time, the criticality to joint velocity and acceleration limits. The GA motion planning is using the cubic spline theory [7] to construct the joint trajectories. The method allows manipulator motion

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

time, velocity, acceleration and jerk to be scaled such that the kinematics constraints are met. Note that the path planning is carried out in joint space and only kinematics constraints (joint velocities, accelerations and jerks) are considered in this study. This paper is organised as follows: Section 2 explains the application of cubic spline joint trajectories and the time scaling method. Section 3 incorporates the parameters obtained in Section 2, i.e. optimal motion time, the criticality to velocity and acceleration limits, into the objective function formulation. Section 4 gives the Pareto-based GA motion planning. Section 5 shows the results when applying this method. Section 6 gives the discussions and conclusions.

2. Cubic Spline Joint Trajectories and Time Scaling Method

Each joint trajectory is fitted to a number of joint displacements at a sequence of time instants by using piecewise cubic polynomials. Let $t_1 < t_2 < t_3 < t_4 < ... < t_{n-2} < t_{n-1} < t_n$ be an ordered time sequence and the position (or knot) of the jth joint at time $t = t_i$ is $\theta_{ji}(t_i)$. Thus, the vector of knots for the jth joint along the path is given as $[\theta_{j1}(t_1), \theta_{j2}(t_1), ..., \theta_{jn}(t_n)]$. The interval time is defined as $h_i = t_{i+1} - t_i$ (i = 1,2,...,n-1) whereas the velocity and acceleration of joint j at knot i are denoted as v_{ji} and w_{ji} respectively. The cubic spline joint trajectories allow the joint velocities, accelerations and jerks to be evaluated at every instant of motion time [7]. Every initial motion time will be tested such that it is optimal and the motion will not violate the kinematics constraints. The intervals must be scaled up or down depending on the scaling factor in (2.4). The knot velocities, accelerations and jerks should be compared with their own limits to obtain the time-optimal path satisfying the joint constraints. Let the absolute values of the jth joint velocity, acceleration and jerk limits denoted as VC_j, WC_j, and JC_j respectively. The scaling factor λ can be obtained as follows:

$$\lambda_1 = \max_j \left[\max_{i=1,...,n} \left\{ \left|v_{ji}\right|, \left|v_{ji}\left(\bar{t}_i\right)\right| \right\} / VC_j \right] \tag{2.1}$$

where $\bar{t}_i$ satisfies $w_{ji}\left(\bar{t}_i\right)$ =0 and is in the interval [t_i , t_{i+1}],

$$\lambda_2 = \max_j \left[\max_i \left| w_{ji} \right| / WC_j \right] \tag{2.2}$$

$$\lambda_3 = \max_j \left[\max_i \left| \frac{w_{j,i+1} - w_{ji}}{h_i} \right| / JC_j \right] \tag{2.3}$$

and $\lambda = \max\left(\lambda_1, \sqrt[2]{\lambda_2}, \sqrt[3]{\lambda_3}\right)$. (2.4)

If the time interval h_i is replaced by λh_i for i =1,2,..., n-1, then the velocity, acceleration and jerk will be replaced by factors of $1/\lambda$, $1/\lambda^2$, $1/\lambda^3$, respectively. These changes assure the satisfaction of constraints on velocities, accelerations and jerks [7].

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

3. Objective Formulation

The objective vector following from the definition of Pareto optimality [13] would look like this:

$$\text{Minimise: } \left(\sum_{i=1}^{n-1} h_i\ ,\ 1\text{-}\lambda_1,\ 1\text{-}\lambda_2\right) \tag{3.1}$$

subjects to constraints: $\lambda_3 \leq 1$,

where h_i is the time interval i, 1-λ_1 is the criticality to the velocity constraints, 1-λ_2 is the criticality to acceleration constraints. The parameters: λ_1, λ_2 and λ_3 are computed by equations (2.1), (2.2) and (2.3) respectively. Criticality is a measurement of a trajectory on how close it is to the joints' velocity and acceleration limits.

4. Pareto-based GA Motion Planning

The schematic diagram for GA minimum time motion planning is illustrated as follows:

Initialisation of parameters

Generate Initial Population

Evaluate the Initial Population

do {
    Selection
    Recombination
    Update-generation
    Evaluate:
        Pareto-Ranking
        Fitness Assignment and Sharing
    Record the best-time trajectory
}
while (generation < maximum generation)
Report.

4.1 Generate Initial Population

From the knots vector described in Section 2, the path is encoded directly as string of floating point to be used by the GA as

$$[\theta_{11},\ \theta_{12},\ldots,\ \theta_{1n};\ \theta_{21},\ \theta_{22},\ldots,\theta_{2n};\ \ldots;\ \theta_{m1},\ \theta_{m2},\ldots,\theta_{mn}]$$

The knot values of each joint are generated randomly using a transition scheme. The joint is restricted to move in only 5 directions from any knot. To create the initial population, two trajectories, one begins from start position and the other one begins

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

from end position will be generated. The trajectories start from the end position will have higher tendency to move downward whereas the other one will have higher possibility to move upward. A valid trajectory will be created once they meet at a particular point. The travel time between intermediate knots is initially set to their lower bounds for each joint, according to the formula as follows:

$$(h_1, h_2, h_3, \dots, h_{n-1}) = \left(\max_j \frac{|\theta_{j2} - \theta_{j1}|}{VC_j}, \max_j \frac{|\theta_{j3} - \theta_{j2}|}{VC_j}, \dots, \max_j \frac{|\theta_{j,n} - \theta_{j,n-1}|}{VC_j} \right) \quad (4.1)$$

and then the time scaling method will convert the time into a feasible one so that the trajectories will not exceed the limits on all the six joints

4.2 Evaluate The Population

All trajectories are ranked based on its total travel time, the criticality to the joints' velocity and acceleration limits (see Section 3) based on Pareto ranking. According to the procedure proposed by [4], an individual in a population can be ranked by counting the number of individuals that dominate it. When all individuals are ranked, the fitness values will be assigned to them according to their rank. In this paper, the fitness assignment is done by interpolating a linear function [12] from the best individual (rank = 0) to the worst individual (rank<maxpop). Following that, same rank individuals will receive the same fitness values by averaging the total values assigned to them. Fitness sharing uses a sharing parameter *h* to control the extend of sharing, which is a measurement of the maximum distance between individuals that could form niches. It is computed as follows [4]:

$$h = [8C_n^{-1}(n+4)(2\sqrt{n})^n / N]^{1/(n+4)} \quad (4.2)$$

where *n* is the number of decision variables, and *N* is the population size. It is also known as smoothing parameter for the Epanechnikov kernel [10]. Each trajectory's share count is set to zero initially, and then it is incremented by a certain amount for every trajectory in the population, including the trajectory itself. Raw fitness value is then recalculated by dividing the values with the total share count. The share count can be obtained by using the Epanechnikov kernel [4]. However, the calculation of Epanechnikov kernel function will require the inverse of the sample covariance matrix and also the determinant of the covariance matrix [10]. To avoid the problem of matrix singularity, an approximation is implemented and the share count function will look like the following [6]:

$$sh(d^*) = \begin{cases} 1 - (d^* / h)^2 & if\ d^* / h < 1 \\ 0 & otherwise \end{cases} \quad (4.3)$$

where d^* is computed such that each decision variable x_i is weighted by its variance. The variance is calculated as follows:

$$\sigma_i^2 = \frac{N\left(\sum_{j=1}^{N} x_{ij}^2\right) - \left(\sum_{i=1}^{N} x_{ij}\right)^2}{N(N-1)} \quad (4.4)$$

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

where N is the number of observations or in other word the population size and d^* is computed as:

$$\left(d^*\right)^2 = \sum_{i=1}^{n} \frac{\left(x_i - y_i\right)^2}{\sigma_i^2}. \tag{4.5}$$

4.3 Selection Scheme And Mating Restriction

Selection scheme is a process to determine the number of trials a particular individual is chosen for reproduction. The selection technique adopted in this project is based on stochastic universal sampling (SUS) introduced by [1]. This method uses a single spin and N equally spaced pointers, where N is the number of population size. Arbitrary recombining pairs of trajectories may produce new offspring that do not represent any niche [3]. It is therefore desirable to reduce crossover between trajectories of different niches. A simple mating restriction scheme is implemented by setting the mating parameter to be equalled to the sharing parameter [3]. The mating parameter is actually the measurement of maximum distance between individuals (trajectories) that allow them to be paired for recombination. Thus, if an individual (trajectory) is within a distance of mating parameter, then a mating companion is found and mating can be performed, otherwise another individual (trajectory) is tried.

4.4 Genetic Recombination

The selected trajectories will be paired up for crossover or recombination subject to their mating distance (i.e. mating restriction) and cross-over probability. A robot trajectory consists of joint angles that may produce large position jump in the offspring strings after conventional crossover. To tackle this problem, a customised genetic operator named as path redistribution and relaxation operator is used [9]. The technique involves fitting cubic splines onto the offspring's knots with each time interval set to one. The path length is then computed as the Euclidean distance between the start and end knots along the splines. Each joint knots are then 'redistributed' evenly over these splines at equal intervals. The paths are then relaxed by moving each knots with a small step towards the point that will bisect the line between its neighbouring knots. Single point cross-over is applied in this GA motion planning. A special purpose mutation operator [9] named as injection is implemented where a fixed number of new trajectories are injected into the population and randomly replace selected trajectory. The number of new trajectories is kept low but it has the effect of preventing premature convergence and creating new search space. After the recombination, the parent trajectory motion time is passed on to the child trajectory as the initial time. The child path together with the motion time will be tested by time scaling method to obtain an optimal motion which will not violate the kinematics constraints. The resulted optimal time and the criticality values to joint velocity and acceleration limits will be used to access the performance of the child trajectory.

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

5. Results

The multiobjective optimisation GA incorporating all the techniques described in the sections above is implemented. In this simulation, the robot is assumed at rest initially, and comes to a full stop at the end of the time interval. In other words, at the initial time $t = t_1$ and the terminal time $t = t_n$, the joint velocity, and joint acceleration are given as: $v_{j1} = 0 = v_{jn}$, and $w_{j1} = 0 = w_{jn}$. The initial and final configurations of the path planning are shown in Table 1. The limits of the velocities, accelerations and jerks are given in Table 2 [2]. The path redistribution-relaxation operator is experimented with different population size, and 0.9 crossover probability [9]. The injection rate is 2.5% of the population size. The results for 100 generations with different population size are shown in Table 3. The motion profiles for 200 population size are given in Figures 5.1-5.4.

Table 1: Initial and final configurations for the path planning of RTX robot.

	Column (m)	**Shoulder (rad)**	**Elbow (rad)**	**Yaw (rad)**	**Pitch (rad)**	**Roll (rad)**
Initial Configuration	0.4	$-\pi/6$	$-\pi/3$	$-\pi/2$	0	$-\pi/4$
Final Configuration	0.8	$\pi/6$	$\pi/3$	$\pi/2$	$-\pi/6$	$\pi/4$

Table 2: Velocity, acceleration and jerk constraints for RTX robot.

	Column	**Shoulder**	**Elbow**	**Yaw**	**Pitch**	**Roll**
Velocity	0.1116	0.6154	1.2092	1.9715	1.3780	1.2412
Acceleration	1.7755	6.2018	14.081	31.055	28.063	26.180
Jerk	297.59	894.67	3718.9	3377.6	3933.1	4172.7

Note: The zed velocity, acceleration, and jerk are in m/s, m/s^2, and m/s^3, respectively. The other joint angle velocities, accelerations and jerks are in rad/s, rad/s^2, and rad/s^3, respectively.

Table 3: Results From GA Minimum Time Motion Planning

Population Size	**Minimum Time(sec)**
100	3.9530
200*	3.9335
300	3.9049

* this is the population size chosen to generate the profiles below.

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

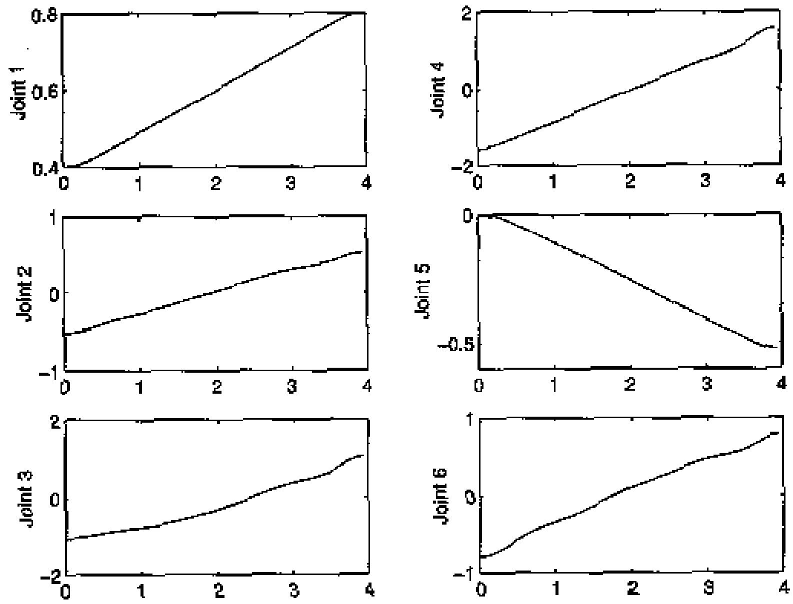

Figure 5.1: Position Profiles

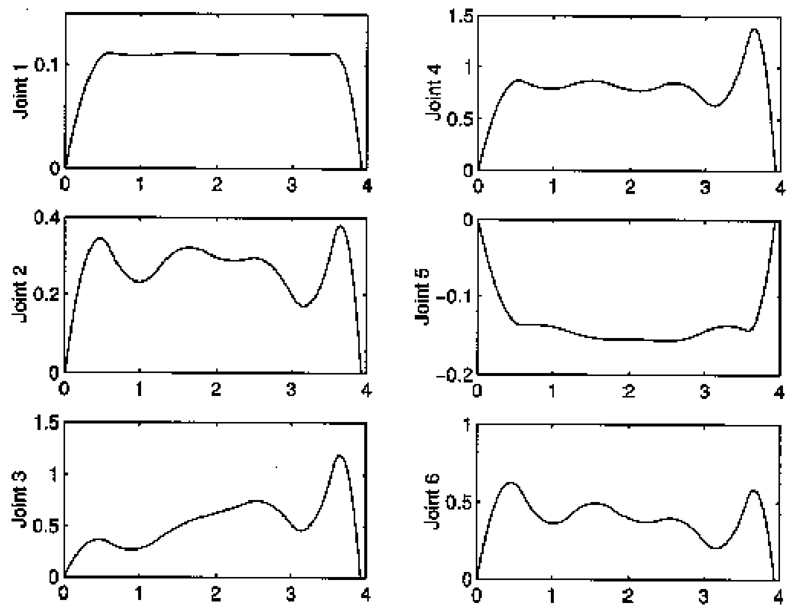

Figure 5.2: Velocity Profiles

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

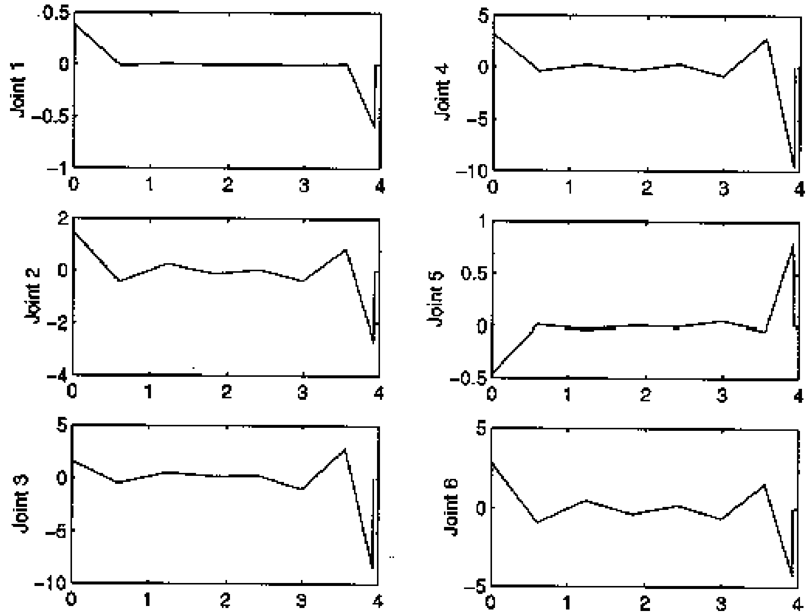

Figure 5.3: Acceleration Profiles

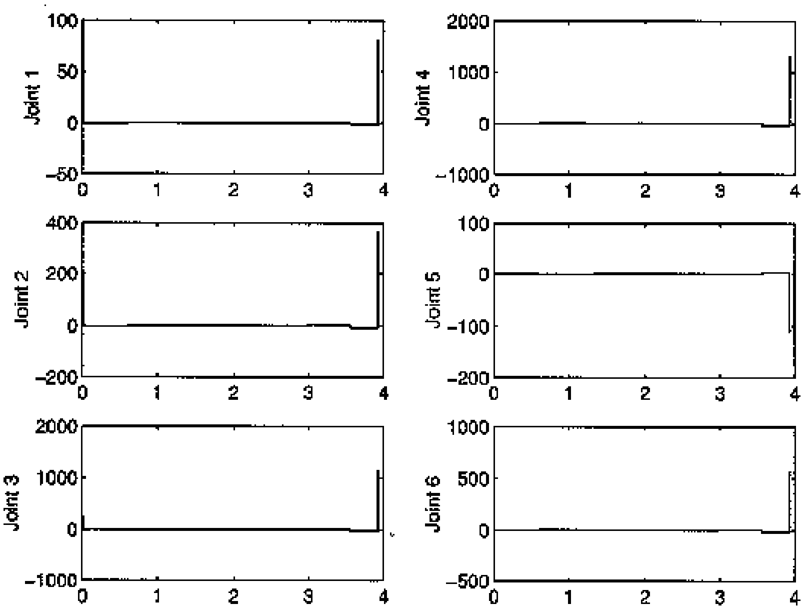

Figure 5.4: Jerk Profiles

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

6. Discussions And Conclusions

Pareto-based multiobjective GA involves ranking process of non-dominated solution to generate the joints trajectories using cubic spline function can produce a optimum path which results in minimum time. This result of minimum motion time improves as the population size increases. However the processing time in obtaining results also increases. Therefore an appropriate population size has to be selected to obtain a reasonably minimum motion time with acceptable length of processing time.

References

1. Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21 Grefenstette, J. J. ed., Lawrence Erlbaum Associates, Publishers.

2. Cao, B., and Dodds, G. I. (1994). Time-Optimal and smooth joint path generation for robot manipulators. *Int. Conf. CONTROL'94,* (Coventry, UK), pp. 1122-1127.

3. Deb, K and Goldberg, D. E. (1989). An Investigation of Niche and Species Formation in Genetic Function Optimization. *Genetic algorithms: Proceedings of the Third International Conference on Genetic Algorithms,* pp. 42-50. Schaffer, J. D. ed., Morgan Kaufmann.

4. Fonseca, C. M. and Fleming, P. J. (1995a). Multiobjective genetic algorithms made easy: selection, sharing, and mating restriction. *First International Conference on 'Genetic Algorithms in Engineering Systems: Innovations and application', GALESIA '95,* (Sheffield, UK, 1995), pp. 45-52.

5. Fonseca, C. M. and Fleming, P. J. (1995b). An Overview of evolutionary algorithms in multiobjective optimisation. *Evolutionary Computation.*, **3**, No. 1, pp. 1-16.

6. Fonseca, C. M. (1996). Personal communication.

7. Lin, C.-S., Chang, P.-R. and Luh, J. Y. S. (1983). Formulation and optimisation of cubic polynomial trajectories for industrial robots. *IEEE Trans. Automatic Control*, Vol. AC-28, No. 12, pp. 1066-1074.

8. Rana, A. S. and Zalzala, A. M. S. *An Evolutionary Algorithm for the Collision Free Motion of Multi-Arm Robots.* Research Report 570, Depart. of ACSE, University of Sheffield, UK, April, 1995.

Published in Proceedings of the 4th International Conference Computer Integrated Manufacturing, 21-24 October, 1997, Singapore (ICCIM 97) (A. Sen, A.I. Sivakumar, R. Gay, Eds.), Vol.2, pp. 1338-1347, ISBN 981-3083-68-9, Springer.

9. Rana, A. S. and Zalzala, A. M. S. (1996). *Minimum Time Motion Planning of the RTX Robot Using an Evolutionary Algorithm*. Research Report 620, Depart. of ACSE, University of Sheffield, UK, April 1996.

10. Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

11. Wang, Q. and Zalzala, A. M. S.(1995). *Genetic Optimisation and Experimentation for the PUMA 560 Manipulator.* Research Report 567, Depart. of ACSE, University of Sheffield, UK, April 1995.

12. Whitley, D. (1989). The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. *Genetic algorithms: Proceedings of the Third International Conference on Genetic Algorithms*, pp. 116-121. Schaffer, J. D. ed., Morgan Kaufmann.

13. Yacov, Y. H., Tarvainen, K, Shima, T. and Thadathil, J. (1990). *Hierarchical Multiobjective Analysis of Large-Scale Systems.* New York: Hemisphere Publishing Corporation.