GENERATION OF ROBOTIC MANIPULATORS TRAJECTORIES USING GENETIC ALGORITHMS

Luiz Eduardo Nicolini do Patrocínio Nunes, luiz@unitau.br
Victor Orlando Gamarra Rosado, victor@feg.unesp.br
Universidade de Taubaté, Rua Daniel Danelli, s/n, Taubaté – SP, CEP 12060-440
Universidade Estadual Paulista, Av. Dr Alberto P. da Cunha, Guaratinguetá – SP, CEP 12516-410

Francisco José Grandinetti, grandi@unitau.br
Universidade de Taubaté, Rua Daniel Danelli, s/n, Taubaté – SP, Cep 12060-44

Abstract. *The trajectories planning for robotic manipulators consists in finding continuous movements that take the arm of a given initial configuration until the desired position in the work space. The objective of this work was to develop an application for robotic manipulator trajectories planning of three degrees of freedom, in a free obstacles environment, using two approaches: genetic algorithms and cubic trajectories. As the initial configuration was known, the inverse kinematics problem for the final configuration was invetigated using genetic algorithms. Thus a Cubic trajectory was employed to calculate the angles for all the intermediate points of the trajectory between the initial and the final manipulator configuration. The used methodology has produced soft trajectories with low computing costs.*

Keywords*: Cubic trajectory, genetic algorithms, inverse Kinematics, robotic manipulator*

1. INTRODUCTION

The inverse kinematics determines the joint angles that result in the manipulator desired position in relation to the coordinate of reference system. The inverse kinematics solution is difficult whereas the mapping between the Cartesian space and the joint space is non-linear and involves equations that can have multiple solutions (Craig, 1989). Several methods for the inverse kinematics solution, based on Genetic Algorithms (GAs) have been proposed in the literature. In Kalra et. al. (2003) the inverse kinematics problem using genetic algorithms was explored for isolated points. In that work, the algorithm provided the two possible solutions to the inverse kinematics problem, the better solution is the one which presented better fitness value.

The inverse kinematics problem for isolated points also was studied by Parker et. al. (1989). In this case, the objective was to place the manipulator in the correct position and to minimize the joint displacements.

Eydgahi and Ganesan (1998) have presented a genetic algorithms application for the fuzzy sets generation and adjustment. This application is used to for the manipulator inverse kinematics solution. The presented method converges quickly to the final solution in comparison with methods based only on fuzzy systems.

The robotic manipulators trajectories planning consists in finding continuous moviments that take the arm of the given initial configuration until a desired position in the work space (Pires and Machado, 1999).

Toogood et. al. (1995) used a genetic algorithm to get a free collisions trajectory for a manipulator in a space containing fixed and known obstacles. Besides avoiding collisions, the trajectory could be optimized for smaller distance, less time or minimum torques.

Tian and Collins (2003, 2004) have proposed a genetic algorithm using real codification for the search of an optimal trajectory of a redundant manipulator. The evaluation function was based on multiple criteria such as total displacement of all the joints and uniform speed in the Cartesian spaces and joints. For validation of this approach, simulations were carried out in a workspace with and without obstacles.

1.1. Objective

This work presents an application for the planning of robotic manipulator trajectories of three degrees of freedom, in a free obstacles environment, using two approaches: Genetic algorithms and Cubic Trajectories. A Cubic trajectory was used to calculate the angles for all the intermediate points of the trajectory between the initial and final configurations. Being the initial configuration known, the inverse kinematics problem for the optimal final configuration was investigated using genetic algorithms with real codification. In the implemented algorithm, singularities do not constitute problem, therefore the algorithm uses the direct kinematics of the manipulator. In the optimizatiom process, the algorithm chose, as solution, the individual better adapted, i.e., that one which generated minor angular displacement of the manipulator joints, with smaller position error.

2. PROBLEM FORMULATION

In this work was considered a robotic manipulator with three degrees of freedom (Figure. 1). The joint angles θ_1, θ_2 and θ_3 can vary between -π and π. The links l_1, l_2 and l_3 have 15cm, 10cm and 5cm of length respectively. The grip must follow the simulated trajectories, starting in the Cartesian coordinate (12.0004, 14.9995) and finishing in different points.

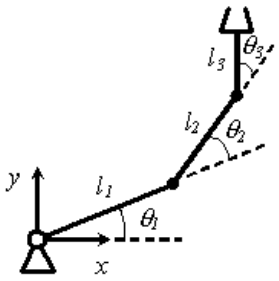


Figure 1. Robotic manipulator

The initial configuration corresponds to the joints variable vector $\{0.0307\ 1.8756\ 1.5691\}^T$ in radians, as illustrated in the Figure 2. In this figure, the area between the circumferences of radius R_1 and R_2 corresponds to manipulator workspace, being $R_1 = l_1 + l_2 + l_3$ and $R_2 = (l_2^2 + (l_1 - l_3)^2)^{0.5}$. The final configuration is the coordinates (x, y) of the grip, that is gotten through the direct kinematics equation, with current angles generated by the GA.

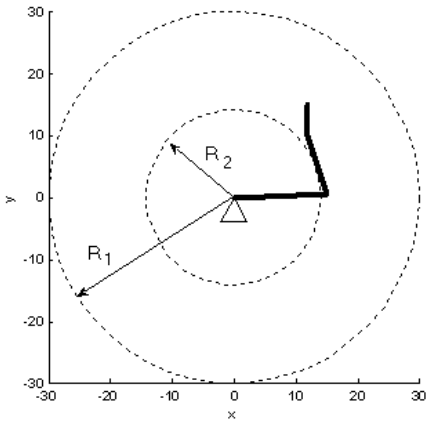


Figure 2. Initial configuration of the manipulator

The grip location for this robotic manipulator is given through the direct kinematics equation:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) + l_3 cos(\theta_1 + \theta_2 + \theta_2) \\ l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) + l_3 sin(\theta_1 + \theta_2 + \theta_2) \end{Bmatrix} \quad (1)$$

The manipulator trajectory consists in a group of strings that represent the joint positions between the initial and final robot configurations. A GA was adopted to look for the optimal angles.

3. GENETIC ALGORITHMS

The Genetic Algorithms (GAs) are techniques for search and optimization, inspired in a Darwin evolution principle. The natural principles, on which the GA's were inspired, are simple. The selection principle privileges the most apt individuals and, therefore, with more probability of reproduction. The individuals with more descendants have more chance of transmitting their genetic codes to the next generations (Michalewicz, 1994). Such genetic codes constitute the identity of each individual and are represented in the chromosomes. These principles are emulated in the construction of computing algorithms that search the best solution for one determined problem; and then through the evolution of codified populations with artificial chromosomes. The components of a GA include: iniciation, selection, crossing and mutation as illustrated the Figure 3 (Kalra et. al., 2003):

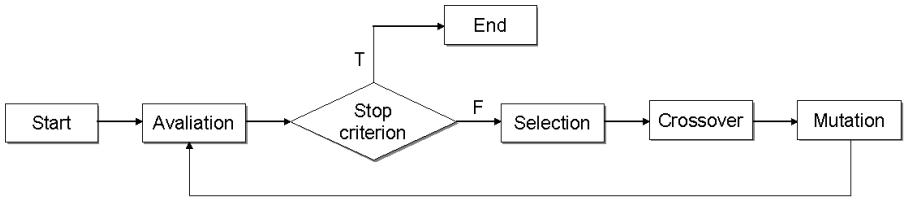


Figure 3. Component of a GA

3.1. Individual representation

The success of a genetic algorithm for a specific optimization problem depends on the representation of an individual in the population (Kalra et. al., 2003). Each possible solution in the search space is represented by a sequence of symbols s generated from an alphabet (binary or real). Each s sequence corresponds to a chromosome and each element of s is equivalent to a gene. For a robotic manipulator, the individuals in a population can be represented, with real codification, through the joint angles: $\{\theta_1\ \theta_2\ \theta_3\}$. The real codification was chosen to avoid succeeding conversions of the binary code or gray for real values, saving, thus, computing time.

3.2. Inicialization

In the initialization process, a population of chromosomes is generated randomly. The size of the population affects the efficiency and the performance of the GA (Goldberg, 1989). A population of small dimension can take the GA to converge quickly to a maximum local, while a very big population, damages the computing performance of the algorithm. The initial population for a robot with three degrees of freedom is generated randomly respecting the inferior (L) and superior (U) limits of each joint variable:

$$\theta_i^L \ \leq \ \theta_i \ \leq \ \theta_i^U \qquad i = 1,2,3 \tag{2}$$

3.3. Evaluation

To each structure (solution) is associated a numerical value (fitness) that represents the quality of this structure and indicates its aptitude degree. The value of fitness is gotten through the objective function. The objective function of this study aims at the minimization of the manipulator position error and the smallest joint angular displacement. The positioning error is calculated through the Euclidean Distance between the current and final manipulator coordinates:

$$E_p = \sqrt{(x_i - x_f)^2 + (y_i - y_f)^2} \tag{3}$$

being (x_f, y_f) the desired position and (x_i, y_i) the current position, gotten through the Direct Kinematics calculation (Eq. 1) with the use of the current angles generated by the GA. The angular error is also gotten through the Euclidean Distance between the initial and final configurations of the joint angles:

$$E_a = \sum_{i=1}^{3} \left\| \{\theta_{i,f}\} - \{\theta_{i,in}\} \right\| \tag{4}$$

Where $\{\theta_{i,in}\}$ is the initial configuration angles, $\{\theta_{i,f}\}$ is the current angles generated by the GA and $\|.\|$ denotes the Euclidean distance. In this work, the positioning error and the angular displacement are studied as multi-objective function (Nunes et. al., 2006). Using the weighting factors method, that satisfies the restriction $\omega_1 + \omega_2 = 1$, the optimization problem is defined as the inverse of the error:

$$fitness = \frac{1}{\omega_1 E_p + \omega_2 E_a} \tag{5}$$

3.4. Seletion

The selection process in GAs chooses individuals for the reproduction. The selection is based on the individuals aptitude: better individuals have more probable of being chosen for the reproduction. The selection method chosen for this work was Stochastic Universal Sampling (SUS), in which the individuals are located in adjacent segments. This

segments length is the same as the value given for the evaluation function to each individual. In this method it is used *n* pointers equally spaced between them (*n* = number of parents to be selected) and the roulette spins only once. The chosen parents are the individuals marked for the *n* pointers. The distance between the pointers will be 1/*n* and the position of the first pointer is given for a number generated randomly between 0 and 1/*n*. The method SUS is considered fast enough for the serial processing and more efficient than the Roulette selection methods, Stochastic Rest and Ranking (Baker, 1987).

3.5. Crossing and mutation

The individuals selected for the following population are recombined through the Crossover operator. This operator is considered the main characteristic of the GAs. The pairs of individuals are chosen randomly and new individuals are created from the interchange of the genetic material. The descendants will be different, however, with genetic characteristics of both. This method (single-point crossover) is the most applied one and was used in this work. The chromosomes created from the crossover operator are, later, submitted to the mutation operation. Based on the probability pm of mutation, the content of a chromosome position is modified.

4. CUBIC TRAJECTORY

This section investigates, in a constant time, the trajectory problem of the manipulator from an initial point until a final point in the Cartesian space. In the search of a function for each joint between the initial position, θ_0, and the final position, θ_f, the set of angles can be calculated, on the basis of the position and final orientation of the manipulator, through the use of its kinematic equations. It is necessary four restrictions in the function $\theta(t)$ that belong to the joints. Here, $\theta(t)$ represents the angular position in the instant of time *t*.

$$\begin{aligned} \theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \end{aligned} \tag{6}$$

Two additional restrictions come from a function that is continuous in the point of the joint speed, i.e. the speed will be zero in the initial and final points of the joint.

$$\begin{aligned} \dot{\theta}(0) &= 0 \\ \dot{\theta}(t_f) &= 0 \end{aligned} \tag{7}$$

These four conditions can be provided for a polynomial of third order. Whereas a cubic polynomial has four coefficients, then, a cubic trajectory can be written as:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{8}$$

The speed and the acceleration of the joint along the trajectory are given respectively by:

$$\begin{aligned} \dot{\theta}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{\theta}(t) &= 2a_2 + 6a_3 t \end{aligned} \tag{9}$$

If the equations (8) and (9) are combined, four equations with four unknown values are gotten:

$$\begin{aligned} \theta_0 &= a_0 \\ \theta_f &= a_0 - a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ 0 &= a_0 \\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned} \tag{10}$$

Finally, with solution of the equations above, the coefficients are found as follow:

$$
\begin{aligned}
a_0 &= \theta_0 \\
a_1 &= 0 \\
a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) \\
a_3 &= \frac{-2}{t_f^3}(\theta_f - \theta_0)
\end{aligned}
\tag{11}
$$

Using the equations above, the equation (12) is derived and used to compute the function of reference position and speed:

$$\theta_i(t) = \theta_{i0} + \frac{3}{t_f^2}(\theta_{if} - \theta_{i0})t^2 - \frac{2}{t_f^3}(\theta_{if} - \theta_{i0})t^3 \quad i = 1,..,m \tag{12}$$

5. METHODS

The hardware used in this work was a microcomputer equipped with a processor Intel® Pentium IV, with 512 MB of RAM memory and processing speed of 2.8 GHz. The inverse kinematics problem for the final configuration was investigated using genetic algorithms, through computing library GAUL (Genetic Algorithm Utility Library). Library GAUL has free distribution and was developed by Adcock (2000). This library was compiled using the Bloodshed Dev-C++, that is an integrated environment of development for the programming language C/C++, it is also free distribution.

The following parameters of control were used for the Genetic Algorithm: population size = 80 individuals, crossing probability = 0.9 and probability of mutation = 0.01. It was adopted respectively the values 0.12 and 0.88 for the the weighting factors ω_1 and ω_2 of the objective function. The genetic algorithm develops while the position error is bigger that 0.1 cm.

6. RESULTS

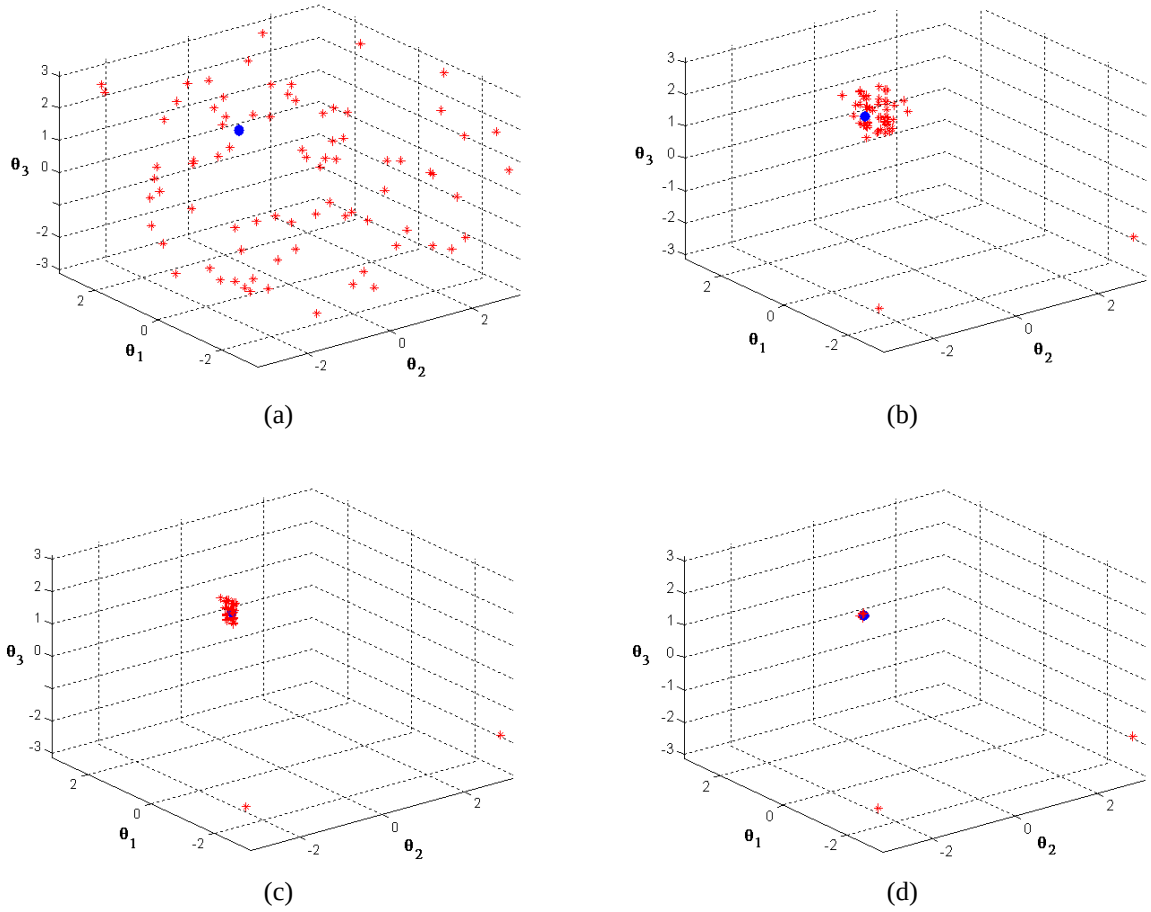
The described GA was, then, used to get an optimal configuration of the trajectory final point. To facilitate the identification of the simulated trajectories, these will be mentioned by numbers, according to Table 1. All the simulations were carried out with the same initial configuration and different final points.

Table 1. Simulations Identification

Simulation nº.	Final Point
1	(20, 10)
2	(15, -20)
3	(-5, 25)

6.1. Inverse kinematics of the final point

In relation to the simulation 1, the GA calculated the optimal angular configuration for the final point (20, 10). Figure 4 shows in (a) the distribuition of the individuals of the initial population occupying all the search space of the joint angles, in (b), the population evolution around the optimal point after 18 cycles, after 100 cycles in (c) and in (d) the proximity of the individuals of the population around the optimal point in the last cycle of evolution. It is observed that in the Figure 4, only two individuals were very distant of the optimal solution.

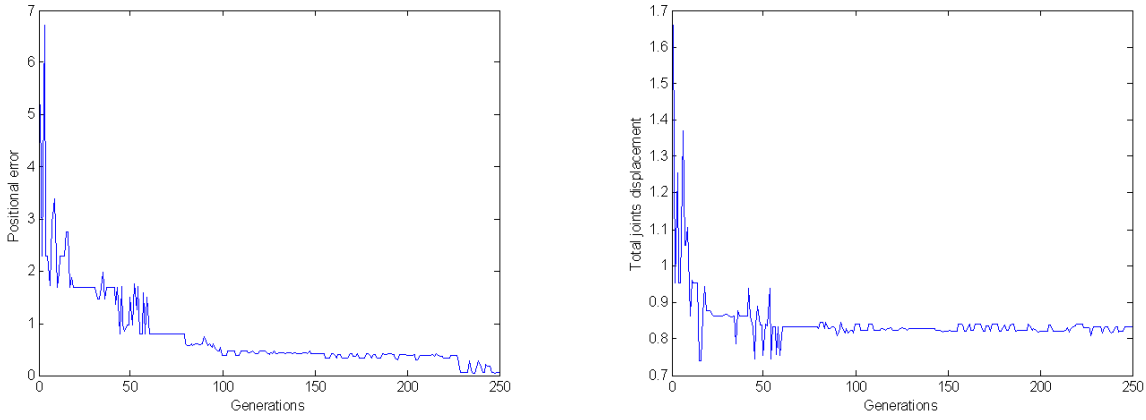


(a) (b)

(c) (d)

Figure 4. Evolution - (a) initial population, (b) 18 cycles, (c) 100 cycles, (d) final population

The positioning error of the manipulator grip and the total joint displacement during the evolution cycle of the Genetic Algorithm are illustrated in the Figures 5 (a) and (b) respectively. The positioning error is stabilized after 100 cycles of evolution. In the first generations, the total displacement of the joint angles presents big leaps, due to the random iniciation of the population.



(a) (b)

Figure 5: (a) error of position and (b) angular displacement.

The angles generated by the Genetic Algorithm for the final point of each simulated trajectory were substituted in the equation of the direct kinematics, given for the Eq. (1), in order to get the positions x and y of the manipulator in that point. The results gotten are presented in the Table 2, that shows the bigger relative error of the position, in relation to the final point was 0.0467% in x and 0.0275% in y, relative errors referring to simulation 2.

Table 2. Relative errors of Position

Final Point (*x*, *y*)	**Actual Positin (*x*, *y*)**	**Relative Error in *x* (%)**	**Relative Error in *y* (%)**
(20, 10)	(20.0000, 9.9999)	0	0.001
(15, -20)	(14.9930, -20.0055)	0.0467	0.0275
(-5, 25)	(-4.9993, 24.9991)	0.014	0.0036

6.2. Generation of Trajectories

Making use of the initial and final configurations of the manipulator, a cubic trajectory was used to find all the intermediate angles between those two configurations. The simulated trajectories were fixed in 120 points, in a time of 5 seconds.

The simulations results are shown in Figure 6 that respectively corresponds to the simulations from 1 to 3. These figures show the successive configurations of the manipulator from the initial point until the final point of each trajectory. As it can be observed in these figures, the joint angles movement of the manipulator is without deviations.

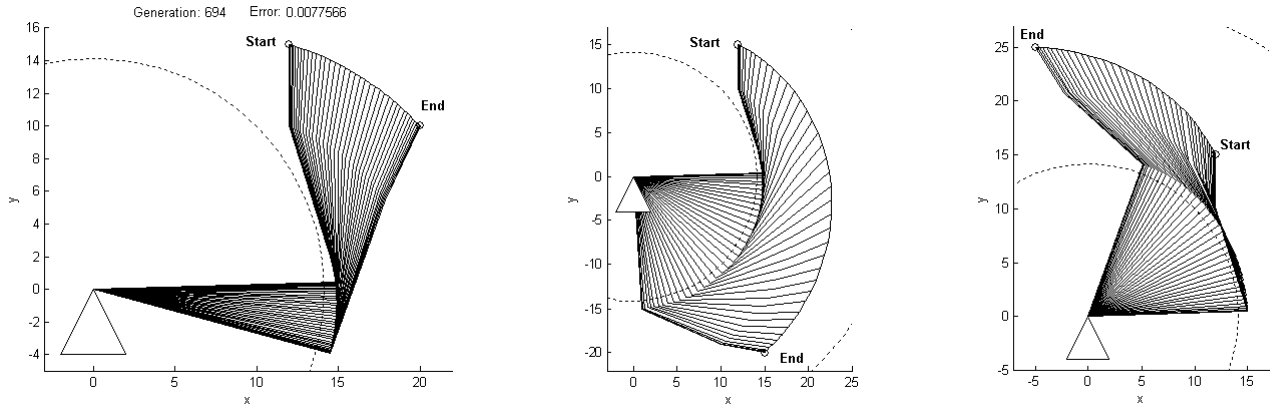


Figure 9. Successive configurations of the trajectories

The proposed algorithm has presented fast computing performance with low cost (executed in less than one second).

7. CONCLUSION

Genetic algorithms had been implemented to find the optimized angles to the manipulator reaches the end point in the Cartesian space. The evaluation function (fitness) had multi-objective character and was defined on two criteria: minimization of positional errors in the Cartesian space and minimum joints displacement. With the known initial configuration and the final configuration, gotten by the Genetic Algorithm, a cubical trajectory was implemented to find the intermediate angles between these configurations. The results had shown that the implemented method is efficient, computationally fast and viable in real applications.

REFERENCES

Adcock, S., 2002. “Genetic Algorithm Utility Library”. 20 Feb. 2005, <http://gaul.sourceforge.net/>.

Ata, A. A., Myo, R. T., 2005. “Optimal Point-to-Point Trajectory Tracking of Redundant Manipulators Using Generalized Pattern Search”, International Journal of Advanced Robotic Systems, Vol. 2, N. 3, pp. 239-244.

Baker, J., 1987, “Reducing bias and inefficiency in the selection algorithm”, In Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, pp. 14–21.

Buckley, K. A., Hopkins, S. H., Turton, B. C. H, 1997, “Solution of Inverse Kinematics Problems of a Highly Kinematically Redundant Manipulator Using Genetic Algorithms”, 2nd Int. Conf. on Genetic Algorithms on Engineering Systems: Innovations and Applications, 2-4 Sep., pp. 264-269.

Craig, J, J., 1989, “Introduction to Robotics: mechanics and control”, Addison-Wesley Publishing, 2nd ed., Mass, USA.

Davis, L., 1996, “Handbook of Genetic Algorithms”, International Thomson Computer Press, Boston, MA, USA, pp. 144-165.

Eydgahi, A. M., Ganesan, S., 1998, “Genetic Based Fuzzy Models for Inverse Kinematics Solution of Robotic Manipulators”, IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, pp. 2196-2201.

Goldberg, D. E., 1989, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Publishing, Mass, USA.

Kalra, P., Mahapatra, P. B., Aggarwal, D. K, 2003, “On the Solution of Multimodal Robot Inverse Kinematics Functions using Real-coded Genetic Algorithms”, IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp. 1979-1984.

Marques, A., Sequeira, J., Ramalho, M., Gonçalves, R., 1996, “Planeamento de Trajectórias de um Manipulador Robótico usando Algoritmos Genéticos”, Proceedings of 1st workshop on Genetic Algorithms and Artificial Life - AGVA96, pp. 26-30.

Michalewicz, Z., 1994, “Genetic Algorithms + Data Structures = Evolution Programs”, Springer-Verlag, 3 ed.

Nunes, L. E. N. P., Rosado, V. O. G., Grandinetti, F. J., 2006, “Geração de Trajetória de um Manipulador Robótico Planar de Três Graus de Liberdade Através de Algoritmos Genéticos”, XVI CBA – Congresso Brasileiro de Automática, Out. 3-6, Salvador – BA.

Parker, J. K., Khoogar, A. R., Goldberg, D. E., 1989, “Inverse Kinematics of Redundant Robots using Genetic Algorithms”, Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, pp. 271-276.

Pires, E. J. S.; Machado, J. A. T., 1999, “A Trajectory Planner for Manipulators using Genetic Algorithms”, Proceedings of IEEE International Symposium on Assembly and Task Planning, pp. 163-168.

Tian, L., Collins, C., 2003, “Motion Planning for Redundant Manipulators Using a Floating Point Genetic Algorithm”, Journal of Intelligent and Robotic Systems, n. 38, pp. 297-312.

Tian, L., Collins, C., 2004, “An Effective Robot Trajectory Planning Method Using a Genetic Algorithm”, Elsevier Mechatronics, n. 14, pp. 455-470.

Toogood, R., Hao, H., Wong, C., 1995, “Robot Path Planning Using Genetic Algorithms”, Proceedings of IEEE Systems, Man and Cybernetics Conference, Vol. 1, pp. 489-494.