AN EVOLUTIONARY APPROACH TO THE TRAJECTORY PLANNING OF REDUNDANT ROBOTS

MARIA DA GRAÇA MARCOS
Instituto Politécnico do Porto, Instituto Superior de Engenharia, Dep. Matemática, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal

J. A. TENREIRO MACHADO
Instituto Politécnico do Porto, Instituto Superior de Engenharia, Dep. Engenharia Electrotécnica, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal

T-P AZEVEDO-PERDICOÚLIS
Universidade de Trás-os-Montes e Alto Douro, Dep. Matemática, 5001 Vila Real, Portugal

Several kinematic techniques for the trajectory optimization of redundant manipulators control the gripper using the pseudoinverse of the Jacobian. Nevertheless, these algorithms lead to a kind of chaotic motion with unpredictable arm configurations. This paper presents a new technique for solving the inverse kinematics problem for redundant manipulators that combines the closed-loop pseudoinverse method with genetic algorithms.

1. Introduction

Kinematic redundancy occurs when a manipulator possesses more degrees of freedom than the required to execute a given task. In this case the inverse kinematics admits an infinite number of solutions, and a criterion to select one of them is required. Most of the research on redundancy deals with the use of these extra degrees of freedom and is referred to in the literature as the resolution of redundancy.

Several techniques for solving the kinematics of redundant manipulators, that have been suggested, control the end-effector, indirectly through the rates at which the joints are driven, using the pseudoinverse of the Jacobian [1]. The pseudoinverse of the Jacobian matrix guarantees an optimal reconstruction of the desired end-effector velocity – in the least-squares sense – with the minimum-norm joint velocity. However, even though the joint velocities are instantaneously minimized, there is no guarantee that the kinematic singularities

are avoided [2]. Moreover, this method has the generally undesirable property that repetitive end-effector motions do not necessarily yield repetitive joint motions. Klein and Huang [3] were the first to observe this phenomenon for the case of the pseudoinverse control of a planar three-link manipulator.

Baillieul [4] proposed a modified Jacobian matrix called the extended jacobian matrix. The extended jacobian is a square matrix that contains the additional information necessary to optimize a certain function. The inverse kinematic solutions are obtained through the inverse of the extended jacobian. The algorithms based on the computation of the extended jacobian matrix have a major advantage over the pseudoinverse techniques because they are locally cyclic [5]. The disadvantage of this approach is that, while mechanical singularities may be avoided, typical algorithmic singularities arise from the way the constraint restricts the motion of the mechanism.

Another class of methods resolves the motion through a direct mapping from the workspace to the joint space. The main advantage of the inverse kinematics method is that the solution yields directly in terms of the joint variables, while, when the pseudoinverse or the extended jacobian method are adopted, the joint velocities must be integrated in order to obtain the joint positions. Chang [6] developed a closed-form solution for the inverse kinematics of manipulators with redundancy using the Lagrangian multiplier method. Another approach is to find a numerical solution by a successive approximation algorithm. For example, Goldenberg *et al*. [7] introduced a generalized solution to the inverse kinematics of robots, that uses the modified Newton-Raphson technique, for solving the system of nonlinear kinematic equations. The solution can be obtained subject to specified constraints based performance criteria. Alternatively, the kinematic model of the manipulator can be divided into subsystems such that an iterative procedure determines some joint variables, while the rest of the variables are obtained through a closed-form solution [8-9].

One optimization method that is gaining popularity for solving complex problems in robotics is the Genetic Algorithm (GA). GAs are population-based stochastic and global search methods. Their performance is superior to that revealed by classical techniques [10] and has been used successfully in robot path planning.

Kubota *et al*. [11] studied a hierarchical trajectory planning method for a redundant manipulator using a virus-evolutionary GA. This method runs, simultaneously, two processes. One process calculates some manipulator collision-free positions and the other generates a collision-free trajectory by

combining these intermediate positions. Nishimura *et al*. [12] proposed a motion planning method using an artificial potential field and a GA for a hyper-redundant manipulator whose workspace includes several obstacles. Peng and Wei [13] presented a trajectory planning method of redundant manipulators by combining a stochastic search algorithm (simulated annealing algorithm) and a GA. In this algorithm the selection, crossover and mutation operators are adjusted by using an adaptive mechanism based on the fitness value.

Having these ideas in mind, the paper is organized as follows. Section 2 introduces the fundamentals of the kinematics and dynamics of redundant manipulators. Based on these concepts, section 3 presents the proposed closed-loop inverse kinematics algorithm with genetic algorithms (CLGA). The simulation results are presented in section 4 and, finally, in section 5 are drawn the main conclusions.

2. Kinematics and Dynamics of Redundant Manipulators

We consider a manipulator with n degrees of freedom, whose joint variables are denoted by $\mathbf{q} = \left[q_1, q_2, \ldots, q_n \right]^T$, and a class of tasks described by m variables, $\mathbf{x} = \left[x_1, x_2, \ldots, x_m \right]^T$, $m < n$. The relation between $\mathbf{q}$ and $\mathbf{x}$ is given by:

$$\mathbf{x} = f\left(\mathbf{q}\right) \tag{1}$$

where f is a function representing the direct kinematics.

The inverse kinematics equation is given as the vector equation:

$$\mathbf{q} = f^{-1}\left(\mathbf{x}\right) \tag{2}$$

Mapping from the world coordinates into the joint coordinates is not one to one, and there may exist an infinite number of joint solutions which result in a given end-effector configuration. Moreover, because of the complexity of (1), the inverse mapping (2) is hard to express in closed form and $\mathbf{q}$ does not necessarily exists.

Differential kinematics of robot manipulators was introduced by Whitney [14] that proposed the use of differential relationships to solve for the joint motion from the Cartesian trajectory of the end-effector. Whitney named this method *resolved motion rate control*. Differentiating (1) with respect to time yields:

$$\dot{\mathbf{x}} = \mathbf{J}\left(\mathbf{q}\right)\dot{\mathbf{q}} \tag{3}$$

where $\dot{\mathbf{x}} \in \Re^m$, $\dot{\mathbf{q}} \in \Re^n$ and $\mathbf{J}(\mathbf{q}) = \partial f(\mathbf{q}) / \partial \mathbf{q} \in \Re^{m \times n}$. Hence, it is possible to calculate a path $\mathbf{q}(t)$ in terms of a prescribed trajectory $\mathbf{x}(t)$ in the operational space.

Equation (3) can be inverted to provide a solution in terms of the joint velocities:

$$\dot{\mathbf{q}} = \mathbf{J}^{\#}(\mathbf{q})\dot{\mathbf{x}} \tag{4}$$

where $\mathbf{J}^{\#}$ is the Moore-Penrose generalized inverse [1] of the Jacobian $\mathbf{J}$.

The dynamic equation of motion for a general *n*-link manipulator can be described by:

$$\mathbf{T} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{5}$$

where $\mathbf{T}$ is the $n \times 1$ joint torque vector, $\mathbf{M}(\mathbf{q})$ is the $n \times n$ inertia matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ is the $n \times 1$ Coriolis/centripetal vector, and $\mathbf{g}(\mathbf{q})$ is the $n \times 1$ gravity vector.

3. Robot Trajectory Control

The Jacobian of a 3-link planar manipulator (*i.e.*, $n = 3$, $m = 2$) has a simple recursive nature according with the expression:

$$\mathbf{J} = \begin{bmatrix} -l_1 S_1 - \ldots - l_3 S_{123} & \cdots & -l_3 S_{123} \\ l_1 C_1 + \ldots + l_3 C_{123} & \cdots & l_3 C_{123} \end{bmatrix} \tag{6}$$

where l_i is the length of link $i = 1, 2, 3$, $q_{i \ldots k} = q_i + \ldots + q_k$, $S_{i \ldots k} = Sin(q_{i \ldots k})$ and $C_{i \ldots k} = Cos(q_{i \ldots k})$.

In the experiments the arms have identical link lengths, $l_1 = l_2 = l_3$.

In the closed-loop pseudoinverse (CLP) method the joint positions can be computed through the time integration of the velocities according with the block diagram depicted in Figure 1.

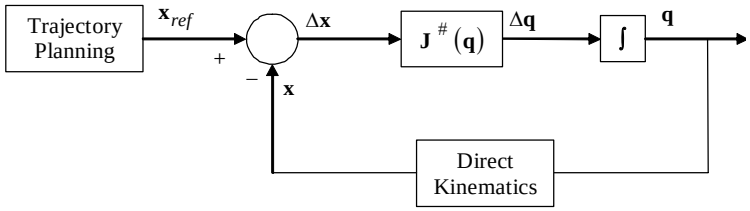

Figure 1. Block diagram of the closed-loop inverse kinematics algorithm with the pseudoinverse.

In a previous study, addressing the CLP method [15], we concluded that this method leads to unpredictable arm configurations and reveals properties similar to those that occur in chaotic systems.

Genetic algorithms (GAs) are a method for solving both constrained and unconstrained optimization problems, based on the mechanics of natural genetics and selection, that was first introduced by Holland [16]. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the fitness or the cost function. The GA modifies repeatedly the population of individual possible solutions. At each step, the genetic algorithm selects individuals at random, from the current population, to be parents, and uses them to produce the offspring for the next generation. Over successive generations, the population evolves towards an optimal solution. The GAs can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, not differentiable, stochastic, or highly nonlinear.

Bearing these facts in mind, in this paper we propose a new method that combines the CLP with a GA, that we call closed-loop inverse kinematics algorithm with genetic algorithms (CLGA).

3.1. *The CLGA Formulation*

The CLGA adopts the closed-loop structure without requiring the calculation of the pseudoinverse. The CLGA uses an extended Jacobian matrix $\mathbf{J}^*$, with dimension $n \times n$, and an extended vector $\Delta\mathbf{x}^*$, with dimension $n \times 1$, as a way to limit the joint configurations for a given end-effector position.

The definition of $\mathbf{J}^*$ and $\Delta\mathbf{x}^*$ take the form:

$$\mathbf{J}^* = \begin{bmatrix} -l_1 S_1 - \ldots - l_3 S_{123} & \cdots & -l_3 S_{123} \\ l_1 C_1 + \ldots + l_3 C_{123} & \cdots & l_3 C_{123} \\ \hline j_{31} & j_{32} & j_{33} \end{bmatrix} \qquad \Delta\mathbf{x}^* = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \hline \Delta x_3 \end{bmatrix} \tag{7}$$

where the matrix elements j_{3i}, $i = 1, 2, 3$, and Δx_3 are values generated by the GA, satisfying the additional imposed constraints.

The flowchart of the CLGA is depicted in Figure 2.

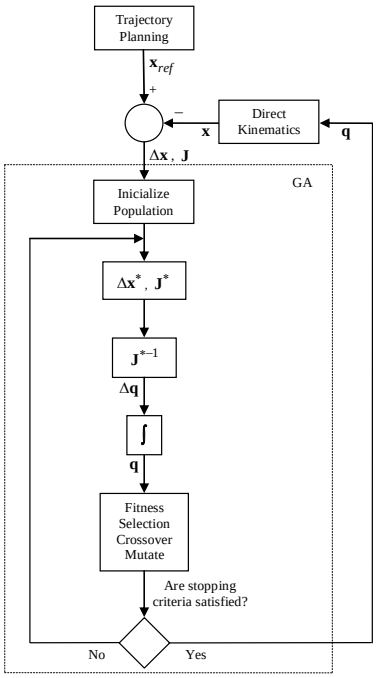

Figure 2. Flowchart of the CLGA algorithm.

3.2. *Optimization Criteria*

The fitness function is designed according to the goal we want to achieve. Four criteria have been selected *a priori*. All constraints and criteria are translated into penalty functions to be minimized and are defined in the sequel.

1. In order to minimize the largest joint displacement between two adjacent configurations the fitness function to be minimized is:

$$f_1 = \max\left\{ \left[q_j(k+1) - q_j(k) \right]^2 \right\}, \quad j = 1,2,3 \tag{8}$$

where k and $k+1$ are two consecutive sampling instants.

2. The total level of joint velocities must be minimized at each configuration leading to the fitness function:

$$f_2 = \sum_{j=1}^{3} \dot{q}_i{}^2 \tag{9}$$

3. In order to minimize the total joint torque in each joint configuration the fitness function is:

$$f_3 = \sum_{i=1}^{3} T_i^2 \tag{10}$$

4. To minimize the total joint power consumption the fitness function is:

$$f_4 = \sum_{j=1}^{3} P_i^2 \tag{11}$$

where the power P_i at each joint i is defined as $P_i = T_i \dot{q}_i$ and T_i is the generalized force/torque for joint i $(i = 1,2,3)$.

4. Simulation Results

This section presents the results of several simulations. The experiments consist in the analysis of the kinematic performance of a planar manipulator with 3 rotational joints (3*R*-robot) that is required to repeat a circular motion in the operational space with frequency $\omega_0 = 7.0 \text{ rad sec}^{-1}$, center at $r = \left(x_1^2 + x_2^2\right)^{1/2}$, radius $\rho = 0.5$ and a step time increment of $\Delta t = 10^{-3}$ sec. The goal here is to position the end-effector of the 3*R*-robot at a target location while satisfying a given optimization criterion. Moreover, the simulations are divided into two groups: workspace without obstacles and workspace with obstacles.

The CLGA algorithm adopts crossover and mutation probabilities of $p_c = 0.5$ and $p_m = 0.2$, respectively, a $n_P = 100$ string population, and the results are obtained for $n_G = 100$ consecutive generations.

If the robot´s end-effector current position is $P_c = \left(x_c, y_c\right)$ and the desired final position is $P_f = \left(x_f, y_f\right)$, then the positional error, P_{error}, is defined as:

$$P_{error} = \sqrt{\left(x_c - x_f\right)^2 + \left(y_c - y_f\right)^2} \tag{12}$$

and the average of the positional error, $\overline{P}_{error}$, is defined as:

$$\overline{P}_{error} = \frac{\sum P_{error}}{\frac{2\pi}{\omega_0 \Delta t} n_C} \tag{13}$$

where n_C represents the number of cycles to be performed by the end-effector.

4.1. *The CLGA Performance in a Workspace without Obstacles*

The 3*R*-robot is firstly tested for the criteria f_1. Figure 3 show some successive robot configurations and positions of the end-effector when using the CLGA for $r = \{0.7,\ 2.0\}$. The initial joint space configurations are assumed as $q_0 = (0^\circ, 14.9^\circ, 154.3^\circ)$ and $q_0 = (0^\circ, 24.5^\circ, 65.3^\circ)$ for $r = \{0.7,\ 2.0\}$, respectively.

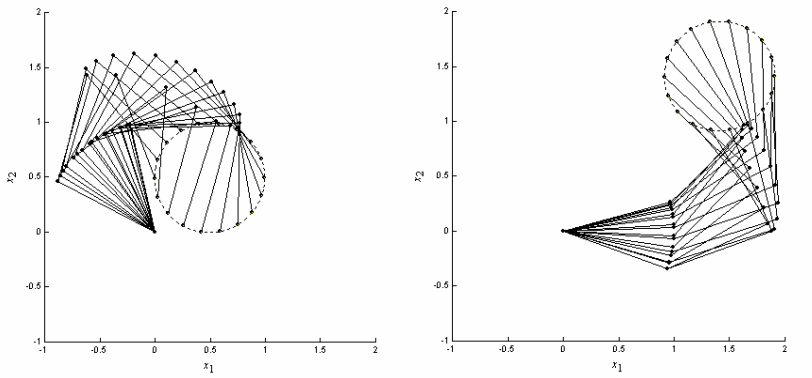

Figure 3. Successive robot configurations in a workspace without obstacles for $r = \{0.7,\ 2.0\}$, respectively, and for the first cycle.

The results are satisfactory because the robot approaches the desired position without trajectory errors.

Next we test the CLGA, when considering all criteria, for $n_C = 100$ cycles and $r = \{0.7,\ 2.0\}$. The average of the positional error, $\overline{P}_{error}$, is presented in Figure 4.

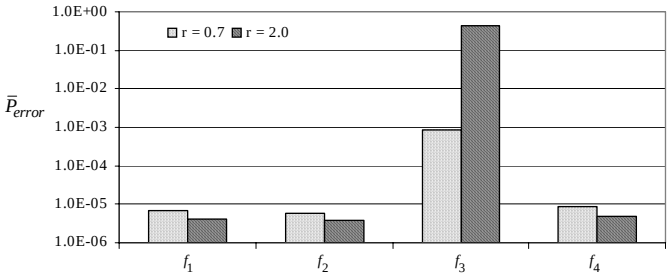

Figure 4. $\overline{P}_{error}$ of the 3*R*-robot, under the action of the CLGA for $n_P = 100$, during $n_C = 100$ cycles for $r = \{0.7,\ 2.0\}$.

We observe that:

- a high precision is achieved when using the CLGA with $\{f_1, f_2, f_4\}$;
- the maximum value of $\overline{P}_{error}$ occurs when the total joint torque in each joint configuration is minimized.

The Fourier transform of the robot joint velocities are depicted in Figure 5 revealing that:

- we achieve repetitive joints positions when using f_1 as we can see in figure 6. For $r = 2.0$ we verify the occurrence of a transient phase for $t \leq 40$ sec, from which the positions of the joints start to be repetitive;
- the signal energy is concentrated in the fundamental and multiple higher harmonics when we minimize the largest joint displacement f_1;
- we get a signal energy distribution along all frequencies when minimizing the total joint torque f_3.

We verify that the CLGA has a better performance than the CLP, because, for example, when minimizing the largest joint displacement f_1, we get not only a good positioning but also a repetitive trajectory.

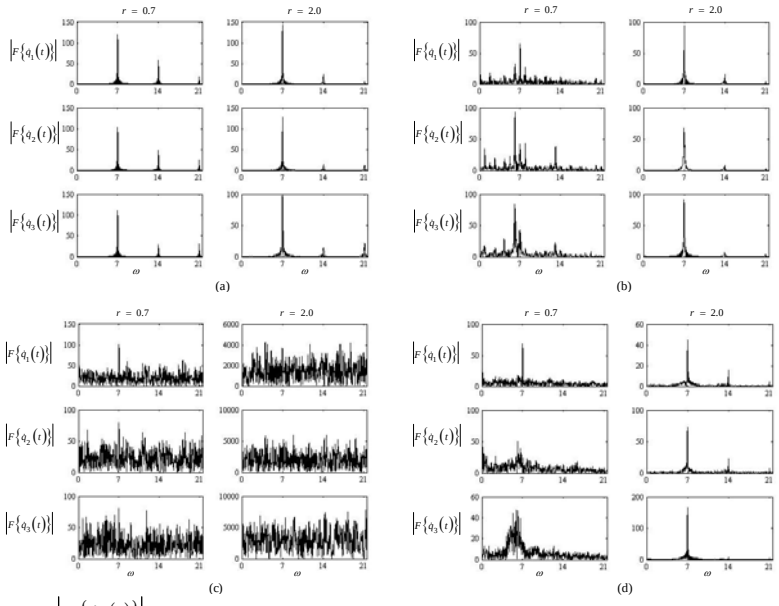

Figure 5. $\left|F\left\{\dot{q}_i(t)\right\}\right|$ *vs* ω of the 3*R*-robot, under the action of the CLGA, during $n_C = 100$ cycles for $r = \{0.7,\ 2.0\}$ and the fitnesses (a) f_1 (b) f_2 (c) f_3 (d) f_4.

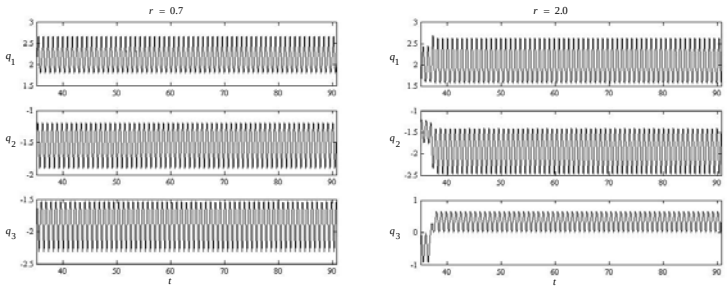

Figure 6. The 3*R*-robot joint positions *vs* time, under the action of the CLGA, for the fitness f_1 and $r = \{0.7,\ 2.0\}$.

4.2. *The CLGA Performance in a Workspace with Obstacles*

This section presents the results of several simulations, for the criteria f_1, when considering two obstacles in the workspace. When, for a joint configuration, some part of the manipulator is inside an obstacle, the CLGA rejects this configuration and generates a new population element.

For $r = 0.7$, the obstacles consist on one circle with center at $(0.3,\ 1.3)$ and radius 0.2, and one rectangle, with the upper left corner and the lower right corner with coordinates $(1.3,\ 0.9)$ and $(1.8,\ 0.5)$, respectively. For $r = 2.0$, the obstacles consist on one circle with center at $(1.6,\ 0.6)$ and radius 0.2, and one rectangle, with the upper left corner and the lower right corner with coordinates $(0.1,\ 1.5)$ and $(0.6,\ 1.1)$, respectively.

Firstly, the 3*R*-robot is tested for a motion with two cycles. The initial joint configurations are identical to the previous ones without obstacles, namely $q_0 = (0^\circ, 14.9^\circ, 154.3^\circ)$ and $q_0 = (0^\circ, 24.5^\circ, 65.3^\circ)$ for $r = \{0.7,\ 2.0\}$, respectively. Figures 7 and 8 show successive robot configurations, for $r = \{0.7,\ 2.0\}$, during the first and second cycles, respectively.

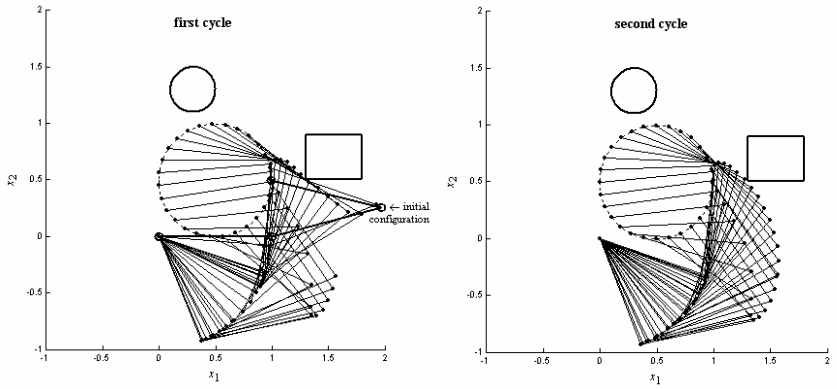

Figure 7. Successive robot configurations in a workspace with obstacles for $r = 0.7$, for the first and second cycles, respectively.

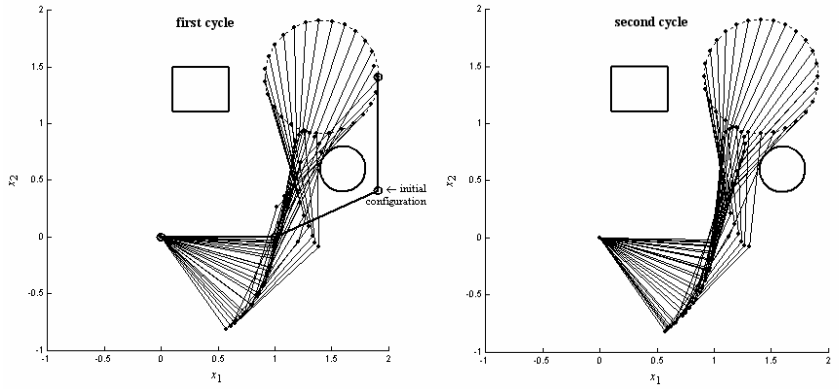

Figure 8. Successive robot configurations in a workspace with obstacles for $r = 2.0$, for the first and second cycles, respectively.

We observe that:

- for $r = 0.7$ the robot approaches the desired position while avoiding the obstacles, for the two cycles;
- for $r = 2.0$ the robot can not reach some points in the circle for the first cycle but, for the second circle, there is no problem to reach the desired points.

Secondly, we repeat the experiment for $r = 2.0$, with the initial joint configuration $q_0 = \left(-16.5^\circ, 90^\circ, -25.8^\circ\right)$. The results for the first cycle are shown in figure 9, revealing that the performance of the CLGA depends on the initial configuration of the manipulator.

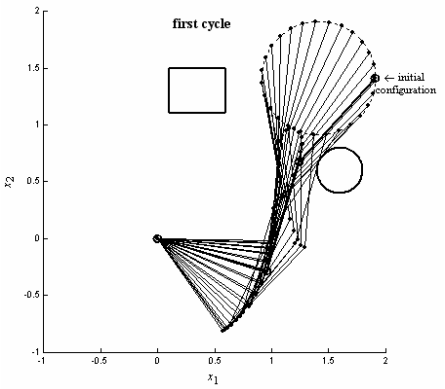

Figure 9. Successive robot configurations in a workspace with obstacles for $r = 2.0$. The first and second cycles are identical.

In a second set of experiments we tested the 3*R*-robot for $n_C = 100$ cycles and $r = \{0.7,\ 2.0\}$. The average of the positional error is $\overline{P}_{error} = \{6.55E-06,\ 9.44E-06\}$ for $r = \{0.7,\ 2.0\}$, respectively. The Fourier spectra of the joint velocities is depicted in figure 10.

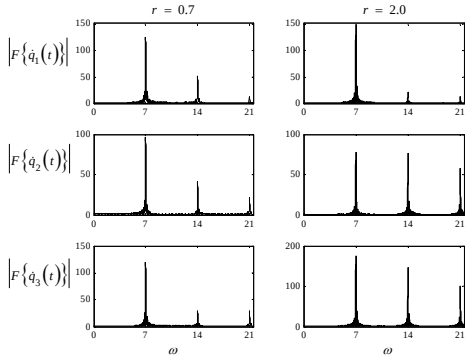

Figure 10. $\left|F\{\dot{q}_i(t)\}\right|$ *vs* ω of the 3*R*-robot, under the action of the CLGA, during $n_C = 100$ cycles for $r = \{0.7,\ 2.0\}$ in a workspace with obstacles.

The results reveal that the average of the positional error, $\overline{P}_{error}$, and the Fourier transform of the robot joint velocities, are consistent with those of the previous section.

5. Conclusions

A CLGA scheme that combines the CLP with a GA scheme was presented. Several experiments were developed to study the performance of the CLGA when the manipulator is required to repeat a circular motion in the operational space while satisfying some different optimization criteria.

The results show that, in general, the CLGA gives superior results in what concerns the repeatability and positioning than the CLP method. The better result occurs when the CLGA minimizes the largest joint displacement between two adjacent configurations since not only we get a good positioning, but also the joint motion is repetitive and the chaotic phenomena observed in the CLP disappear. Moreover, it is shown that the presence of obstacles does not present an additional complexity for the algorithm to reach the solution, when the CLGA minimizes the largest joint displacement between two adjacent configurations, as long as the selected initial joint configurations are adequate for the required task.

References

1. K. L. Doty, C. Melchiorri and C. Bonivento, A Theory of Generalized Inverses Applied to Robotics. *The Int. Journal of Robotics Research*, **12(1**): 1–19 (1993).
2. J. Baillieul, J. Hollerbach and R. Brockett. Programming and control of kinematically redundant manipulators. *Proc. of the23rd IEEE Conf. on Decision and Control*, 768–774 (1984).
3. C. A. Klein and C. H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulator. *IEEE Trans. Systems, Man, Cybern*, **SMC(13)**:245–250 (1983).
4. J. Baillieul. Kinematic programming alternatives for redundant manipulators. *Proc. of the 1985 IEEE Int. Conf. on Robotics and Automation*, 722–728 (1985).
5. D. R. Baker and C. W. Wampler II. On the Inverse Kinematics of Redundant Manipulators. *The Int. Journal of Robotics Research*, **7(2-11)**:3–21 (1988).
6. P. Chang. A closed-form solution for inverse kinematics of robot manipulators with redundancy. *IEEE Journal of Robotics and Automation*, **RA-3(5)**:393–403 (1987).
7. A. A. Goldenberg, B. Benhabib and R. G. Fentor. A complete generalized solution to the inverse kinematics of robots. *IEEE Journal of Robotics and Automation*, **RA-1(1)**:14–20 (1985).

8. V. J. Lumelsky. Iterative coordinate transformation procedure for one class of robots. *IEEE Trans. Systems, Man, Cybern*, **SMC(14)**:500–505 (1984).
9. R. Featherstone. Position and velocity transformations between robot end effector coordinates and joint angles. *The Int. Journal of Robotics Research*, **2(2)**:35–45 (1983).
10. D. E. Goldenberg. Genetic algorithms in search optimization, and machine learning. Reading, MA: Addison-Wesley. (1989).
11. N. Kubota, T. Arakawa, T. Fukuda and K. Shimojima. Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm. *Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation*, 205–210 (1997).
12. T. Nishimura, K. Sugawara, I. Yoshihara and K. Abe. A motion planning method for a hyper multi-joint manipulator using genetic algorithm. *Proc. of the 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 645–650 (1999).
13. Y. Peng and W. Wei. A New Trajectory Planning Method of Redundant Manipulator Based on Adaptive Simulated Annealing Genetic Algorithm (ASAGA). *Proc. of the 2006 IEEE Int. Conf. on Computational Intelligence and Security*, 262–265 (2006).
14. D. E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, **MMS-10(2)**:47–53 (1969).
15. M. G. Marcos, F. B. M. Duarte and J. A. T. Machado. Complex Dynamics in the Trajectory Control of Redundant Manipulators. *Transactions of Nonlinear Science and Complexity*, 134–143 (2006).
16. J. H. Holland, (1975). Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor. 2nd ed. Mit Press (1992).