Multi-Criteria Optimization Manipulator Trajectory Planning

E. J. Solteiro Pires and P. B. de Moura Oliveira
Universidade de Trás-os-Montes e Alto Douro
epires,oliveira@utad.pt
Portugal

J. A. Tenreiro Machado
Instituto Superior de Engenharia do Porto
`jtm@isep.ipp.pt`
Portugal

1. Introduction

In the last twenty years genetic algorithms (GAs) were applied in a plethora of fields such as: control, system identification, robotics, planning and scheduling, image processing, and pattern and speech recognition (Bäck et al., 1997). In robotics the problems of trajectory planning, collision avoidance and manipulator structure design considering a single criteria has been solved using several techniques (Alander, 2003).

Most engineering applications require the optimization of several criteria simultaneously. Often the problems are complex, include discrete and continuous variables and there is no prior knowledge about the search space. These kind of problems are very more complex, since they consider multiple design criteria simultaneously within the optimization procedure. This is known as a multi-criteria (or multi-objective) optimization, that has been addressed successfully through GAs (Deb, 2001). The overall aim of multi-criteria evolutionary algorithms is to achieve a set of non-dominated optimal solutions known as Pareto front. At the end of the optimization procedure, instead of a single optimal (or near optimal) solution, the decision maker can select a solution from the Pareto front. Some of the key issues in multi-criteria GAs are: i) the number of objectives, ii) to obtain a Pareto front as wide as possible and iii) to achieve a Pareto front uniformly spread.

Indeed, multi-objective techniques using GAs have been increasing in relevance as a research area. In 1989, Goldberg suggested the use of a GA to solve multi-objective problems and since then other researchers have been developing new methods, such as the multi-objective genetic algorithm (MOGA) (Fonseca & Fleming, 1995), the non-dominated sorted genetic algorithm (NSGA) (Deb, 2001), and the niched Pareto genetic algorithm (NPGA) (Horn et al., 1994), among several other variants (Coello, 1998).

In this work the trajectory planning problem considers: i) robots with 2 and 3 degrees of freedom (*dof*), ii) the inclusion of obstacles in the workspace and iii) up to five criteria that are used to qualify the evolving trajectory, namely the: joint traveling distance, joint velocity, end effector / Cartesian distance, end effector / Cartesian velocity and energy involved. These criteria are used to minimize the joint

and end effector traveled distance, trajectory ripple and energy required by the manipulator to reach at destination point.
Bearing this ideas in mind, the paper addresses the planning of robot trajectories, meaning the development of an algorithm to find a continuous motion that takes the manipulator from a given starting configuration up to a desired end position without colliding with any obstacle in the workspace.
The chapter is organized as follows. Section 2 describes the trajectory planning and several approaches proposed in the literature. Section 3 formulates the problem, namely the representation adopted to solve the trajectory planning and the objectives considered in the optimization. Section 4 studies the algorithm convergence. Section 5 studies a 2*R* manipulator (*i.e.*, a robot with two rotational joints/links) when the optimization trajectory considers two and five objectives. Sections 6 and 7 show the results for the 3*R* redundant manipulator with five goals and for other complementary experiments are described, respectively. Finally, section 8 draws the main conclusions.

2. Trajectory planning

Trajectory planning for robotic manipulators is the process of creating trajectories free of collisions allowing the manipulators to perform the required task. The robotic planners substitute the human operator to specify explicitly the trajectory. Consequently, the operator is free to focus his attention on the task instead of worrying about the robot movement. Therefore, the operator only needs to specify the start and the end path points, leaving the planner to generate the appropriate trajectory that the manipulator must perform.
The trajectories are made by successive displacements of the robotic end effector and a trajectory can be seen as a sequence of points in which the end effector must pass. As a result of the end effector movement over the discrete points, it is obtained a continuous trajectory (Fig. 1). Optimizing the robot trajectory involves the identification of optimal points and the corresponding intermediate positions. Nevertheless, the trajectory optimization is difficult due to the non-linearity dynamics and the dimension of the trajectory search space.
The trajectory planning can be implemented using either the direct or the inverse kinematics. When it is adopted the direct kinematic, the problem is directly solved in the joint space. On the other hand, when the problem is solved through the inverse kinematics, it is determined the trajectory of the manipulator end effector in the operational space. Then, the values of the joints are evaluated using the inverse kinematics. The resulting values of the joints variables are then used by the planner to form the final trajectory. However, when the inverse kinematics is considered, due to the existence of singularities, some problems may arise (Duarte, 2002), such as:

- The mobility of the manipulator is reduced and it way be not possible to impose certain body movements to the end effector;
- The inverse kinematics problem can have multiple solutions;
- In the neighborhood of the kinematic singularities low speeds in operational space may require high speeds in joint space.

In order to solve the planning problem it is necessary to model the trajectory. Therefore, the representation of a manipulator trajectory can be seen as a string representing all the joint positions between the initial and final robot configurations. The trajectory is determined in order to satisfy some specifications with the best performance. Best performance may have a different meaning such as minimum energy consumption, trajectory duration, singularities avoidance, *etc*. Therefore, the trajectory optimization for robotic manipulators is a problem involving the use of multiple criteria. The resolution of such problems can benefit from the multi-objective optimization, particularly when the objectives

are conflictous. This means that an improvement of one criterion leads, necessarily, to the degradation of another one. In these cases, the optimal compromise between them is possible considering the concepts of non-dominance theory proposed by Pareto and successfully integrated in evolutionary multi-objective (EMO). The major disadvantage of using EMO is the increasing of the computational time with the number of objectives under analysis, which is progressively reduced with advent of faster processors. This makes the techniques proposed in this chapter to have viability in the context of industrial applications.

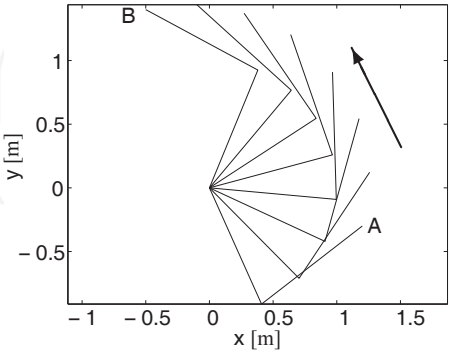


Fig. 1. Trajectory of a manipulator with two rotational degrees of freedom (2*R* robot)

2.1 Trajectory planning: A review

Trajectory planning is a fundamental research issue in robotics and, GAs that have been successfully adapted to solve many problems, also tracked this problem. This section describes GA-methods, proposed by several authors, to solve the planning of robot trajectories.

Davidor (1991) was the first researcher that applied GAs to robot path planning. He uses a string with a non fixed size to represent paths for a 3*R* robotic manipulator. The performance was evaluated considering the error between the desired path and the path yielded by the GA. To find the manipulator configurations he used the inverse kinematics. Other application similar was proposed by Nearchour and Aspragathos (1995) for a 7*R* robot. In this work a point-to-point movement was considered in a workspace with obstacles. The problems consisted in finding the configuration that the manipulator should adopt in order to put the end effector in the correct position. Another problem, using the inverse kinematics, was addressed by Lavoie and Boudreau (2001) for 7 and 10 *dof* manipulators. The end effector had to travel through several pre-defined points and the objective was to minimize the joint displacement without robot colliding with obstacles. Doyle and Jones (1996) proposed a GA to find the successive robot configurations with the algorithm searching the joint space in order to find the best path. Lee *et al.* (1999) proposed two applications based on evolutionary algorithms (EAs) for a 2*R* robot: One application using a GA with the penality method, and the second adopting a evolutionary strategy with a heuristic methods and using a time scheduling to lead with the joint motors torque constraints. A comparison with linear programming was included in the work.

Other algorithms inspired in nature have also been applied to robot trajectory planning. Kubota *et al.* (1997; 1996) proposed a hierarchical EA to finds the trajectory of a redundant manipulator without colliding with the workspace obstacles. The algorithm was based on the co-evolution of virus and host populations. One population calculates some manipulator collision-free positions and the other generates a collision-free trajectory by combining these intermediate positions. The objective consisted

in minimizing an aggregate function including the distance traveled and the energy required to perform the task. Luo and Wei (2004) addressed the problem for a 3*R* manipulator based on a biological immune system. The problem was reduced to one objective, aggregating all the criteria together, and to one parameter (joint) making all the other joints dependent of that variable.

GAs can be used in trajectory generation to determine the intermediate points of the polynomial that describes the manipulator movement. In this kind of application the velocities and accelerations are considered null in the extreme points. Wei-Min and Yu-Geng (1996) developed a method for the 2*R* and 3*R* manipulators incorporating kinematic, dynamical and control constraints. The method uses polynomials of 4^{th} and 5^{th} degree and optimizing the time duration of the trajectory. Wang and Zalzala (1996) also developed an algorithm for an industrial robot to provide a short travel time duration. The algorithm divided the joint space through a grid, using a 6^{th} degree polynomial and a binary GA with heuristic search techniques. Another application was proposed by Tian and Collins (2004), adopting a binary GA and a cubic polynomial for the trajectory generation of a 2*R* manipulator in a workspace containing point obstacles.

GAs are also applied in cooperative manipulator systems that is having more than one manipulator operating in the same workspace. Rana and Zazala (1996) proposed a technique evolving two manipulators. This algorithm minimized the trajectory duration while avoiding collisions between the manipulators. The planning was carried out in the joint space and the trajectory was represented as a string of via-points connected through cubic splines.

Another method based on a hybrid EA was described by Ridao *et al.* (2001). The GA had to determine the sequence of synchronization points where the length of the path is minimal. For this purpose, the algorithm used predetermined manipulators paths (given by other algorithms like local search). Ali *et al.* (2002) proposed a method, in a three-dimensional space, for two 2*R* manipulators which consisted in minimizing the trajectory distance without colliding with obstacles. Another application, where the objective is the minimization of the energy required by the manipulator to perform the trajectory, was proposed by Garg and Kumar (2001; 2002) using both GAs and simulated annealing.

In the trajectory planning there are some approaches that optimize independently the objectives. Ortmann (2001) presented a multi-criteria scheme for a manipulator where the workspace has one obstacle. The algorithm tries to find multiple joint positions of the trajectory.

Chen and Zalzala (1997) proposed a GA to generate the position and the configuration of a mobile manipulator. In this report the inverse kinematics optimizes the least torque norm, the manipulability, the torque distribution and the obstacle avoidance.

In table 1 are depicted the main differences of the mentioned works. The column 'Kinematics' indicates if the work solves the planning trough the direct (D) or the inverse (I) kinematics. The manipulator type and the number of obstacles are described in the next columns. The column 'Objective' specifies the criteria used in the optimization. The space dimension and the information if the algorithm uses cooperative techniques are described in next columns. Column eight indicates if exists more than one manipulator, and its number, in the workspace. 'Dynamics' and 'Polynomial interpolation' indicates if the algorithm uses the dynamic equations and polynomial functions, or splines, respectively. Finally, the last two columns indicate if the algorithm adopts immune systems, hybrid GAs, and if it incorporate a mobile base.

3. Problem description

This work considers robotic manipulators that are required to move between two configurations taking into consideration several objectives. The objectives considered in this work are: the joint distance (O_q), the Cartesian / gripper distance (O_p), the joint ripple ($O_{\dot{q}}$), the Cartesian / gripper ripple ($O_{\dot{p}}$) and the energy required by the manipulator to make the movement (O_{E_a}). Thus, it is intended to determine

Authors	Kinematics	Manipulator	Obstacles	Objective	Workspace	Co-Evolutive	Cooperative	Dynamics	Polinomial interpolation	Immune systems	Hybrid GA	Mobile manipulator
Davidor	I	3*R*	-	Cartesian distance	2*D*	-	-	-	-	-	-	-
Nearchour and Aspragathos	I	7*R*	Y	Cartesian distance	2*D*	-	-	-	-	-	-	-
Lavoie and Boudreau	I	7;10*R*	Y	Joint distance	2*D*	-	-	-	-	-	-	-
Doyle and Jones		3*R*	Y	Joint distance	2*D*	-	-	-	-	-	-	-
Lee *et al.*		2*R*	-	Trajecty duration	2*D*	-	-	Y	-	-	-	-
Kubota *et al.*	D	7 gdl	Y	Distance and Power	3*D*	Y	-	Y	-	-	-	-
Luo and Wei	I	3*R*	-	Cartesian distance	2*D*	-	-	-	-	Y	-	-
Wei-Min and Yu-Geng	I	2;3*R*	-	Joint distance	2*D*	-	-	Y	Y	-	-	-
Wang and Zalzala	D	6*R*	-	Joint distance	3*D*	-	-	-	Y	-	-	-
Tian and Collins	I	2*R*	Y	Joint distance	2*D*	-	-	-	Y	-	-	-
Rana and Zazala	I	2;3*R*	-	Trajecty duration	2;3*D*	-	2	Y	Y	-	-	-
Ridao *et al.*		2;3*R*	Y	Trajecty duration	2;3*D*	-	2	-	-	-	Y	-
Ali *et al.*	D	2*R*	Y	Cartesian distance	3*D*	Y	2	Y	Y	-	Y	-
Garg and Kumar	D	2*R*	-	Energy	2*D*	-	2	Y	Y	-	Y	-
Ortmann	D	3*R*	-	Arm ripple	3*D*	-	-	-	Y	-	Y	-
Chen and Zalzala	I	-	Y	Torque and manipulability	3*D*	-	-	-	-	-	-	Y

Table 1. Genetic applications in manipulator trajectory planning

a set of non-dominated solutions from the Pareto optimal front. The final solution is then chosen by the decision maker taking into account the commitment of the objectives that he finds more appropriate. Two (see figure 2) and three *dof* planar manipulators are used in the experiments. However, this algorithm can be extended to hyper-redundant robots. The rotational joint of each link, is free to rotate 2π rad. To test a possible manipulator / obstacle collision, the end effector structure is analyzed to verify if it is inside of any obstacle.

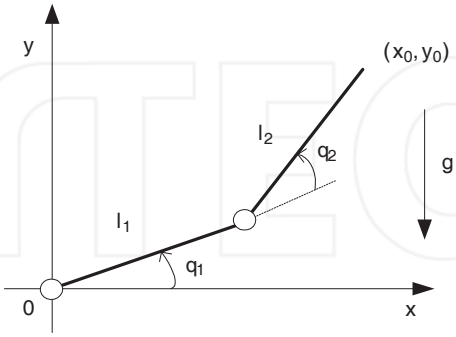


Fig. 2. Two joint (link) manipulator (2*R* robot) (g – gravity constant)

The path for a *iR* manipulator ($i = 2, 3$), at generation T, is directly encoded as a string in the joint space to be used by the GA. This string is represented by expression (1), where i represents the number of *dof* and Δt the sampling time between two consecutive configurations. Therefore, one string is codified as:

$$[\{ q_1^{(\Delta t,T)}, q_2^{(\Delta t,T)} \}, \{ q_1^{(2\Delta t,T)}, q_2^{(2\Delta t,T)} \}, \ldots, \{ q_1^{((n-2)\Delta t,T)}, q_2^{((n-2)\Delta t,T)} \}] \tag{1}$$

where the joints values $q_l^{(j\Delta t,0)}$ ($j = 1, \ldots, n-2$; $l = 1, \ldots, i$) are randomly initialized in the range $]-\pi, +\pi]$ rad. The robot movement is described by $n = 8$ configurations. However, the initial and final configurations are not encoded into the string because they remain unchanged throughout the trajectory search. Without losing generality, for simplicity, it is adopted a normalized time of $\Delta t = 0.1$ s, but it is always possible to perform a time re-scaling.

Five indices $\{O_q, O_{\dot{q}}, O_p, O_{\dot{p}}, O_{E_a}\}$ presented in (2) quantify the qualify of the evolving trajectory robotic manipulators. The indices represents the joint distance, O_q, the joint ripple, $O_{\dot{q}}$, the Cartesian / gripper distance and ripple (O_p and $O_{\dot{p}}$), and the energy, O_{E_a}.

$$O_q = \sum_{j=1}^{n} \sum_{l=1}^{i} \left(\dot{q}_l^{(j\Delta t,T)} \right)^2 \tag{2a}$$

$$O_{\dot{q}} = \sum_{j=1}^{n} \sum_{l=1}^{i} \left(\ddot{q}_l^{(j\Delta t,T)} \right)^2 \tag{2b}$$

$$O_p = \sum_{j=2}^{n} d\left(p_j,\ p_{j-1} \right)^2 \tag{2c}$$

$$O_{\dot{p}} = \sum_{j=3}^{n} \left\{ d\left(p_j,\ p_{j-1} \right) - d\left(p_{j-1},\ p_{j-2} \right) \right\}^2 \tag{2d}$$

$$O_{E_a} = (n-1)T\,P_a = \sum_{j=1}^{n}\sum_{l=1}^{i} |\tau_l . \Delta q_l^{(j\Delta t,T)}| \tag{2e}$$

The function O_{E_a} evaluates the average mechanical energy during the trajectory whereas is assumed that power regeneration is not available by motors doing negative work, that is, by taking the absolute value of the power (Silva & Tenreiro Machado, 1999). The index i specifies the total number of manipulator joints. These criteria are minimized by the planner to find the non-dominated front. Before evaluating any solution, in order to remove virtual jumps, all the values such that $|q_i^{((j+1)\Delta t,T)} - q_i^{(j\Delta t,T)}| > \pi$ are readjusted, by adding or removing a multiple value of 2π in the strings.

The joint distance O_q represented in (2a) is used to minimize the manipulator joints traveling distance. In fact, for a function $y = g(x)$ the curve length is defined by equation (3) and, consequently, to minimize the curve length distance the simplified expression (4) is adopted.

$$\int [1 + (dg/dx)^2]dx \tag{3}$$

$$\int (dg/dx)^2 dx = \int \dot{g}^2 dx \tag{4}$$

Throughout this chapter the results are only presented in the objective space, since the size of the space of attributes is of the order of $i(n-2)$ which makes the analysis difficult.

To solve the problem a multi-objective algorithm is used with selection operator based on the Pareto ranking proposed by Goldberg (1989) and a sharing scheme with $\sigma_{sharing} = 0.01$ and $\alpha = 2$ parameters (5), where $d_{k,j}$ is the distance between solution k and j. So, the fitness value is initially assigned according to the sorted rank of the solution in the population, and then this values is affected by the number of solutions in their neighborhood (niche count). To evaluate the niche count metric, all the neighborhood solutions of the population are considered regardless of their rank. After the crossover and the mutation operations, the elements that are selected to the next generation are the best of all between parents and their offspring.

$$f'^{(k)} = \frac{f^{(i)}}{n_{c_k}} \tag{5a}$$

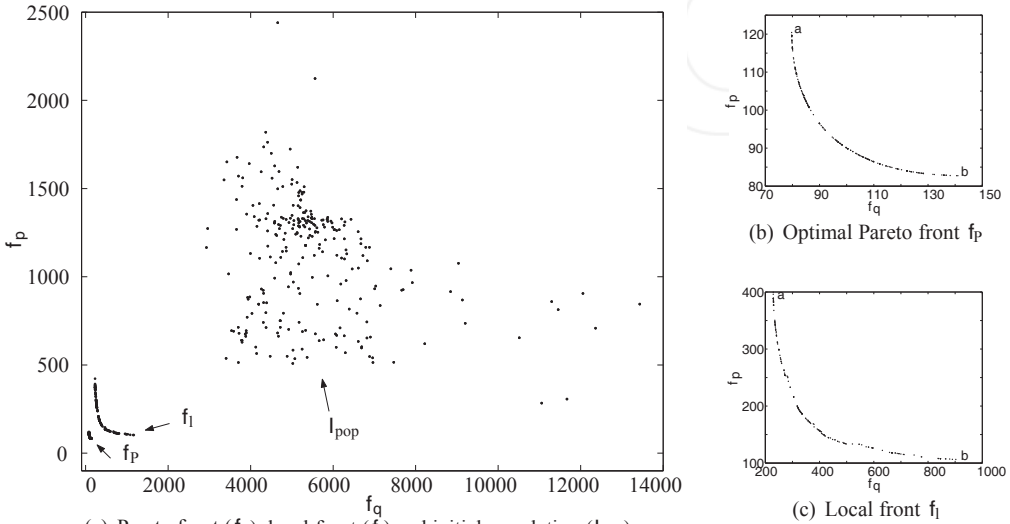
$$n_{c_k} = \sum_{j=1} d_\sigma(d_{k,j}) \tag{5b}$$

$$d_\sigma(d_{k,j}) = \begin{cases} 1 - \left(\frac{d_{k,j}}{\sigma}\right)^\alpha, & d_{k,j} < \sigma \\ 0, & d_{k,j} \geq \sigma \end{cases} \tag{5c}$$

4. Algorithm convergence

This section analysis the convergence of the algorithm to the locals fronts when the trajectory is optimized. Several experiments consisting in moving a $2R$ manipulator from the point $A \{x_0, y_0\} = \{1.2, -0.3\}$ up to the point $B \{x_1, y_1\} = \{-0.5, 1.4\}$ are performed. The optimization considers the angular and the Cartesian end effector distance criteria, while the initial and final configurations are determined through the inverse kinematics. The simulations adopt the parameters (defined in advance): $pop_{lim} = 300$ population solutions, during $T_t = 1500$ generations, and probabilities of crossover and mutation of $p_c = 0.6$ and $p_m = 0.05$, respectively. Each link has a length of $l = 1$ m and mass of $m = 1$ kg.

The algorithm converges to the fronts illustrated in Fig. 3. One of the fronts is obtained when the planar manipulator moves around its base in a counterclockwise direction (Figure 4(a)). The corresponding optimal Pareto front is shown in Fig. 3(a) as f_P and is expanded separately illustrated in Fig. 3(b). The other local front is obtained when the manipulator circumvents the base in the clockwise direction (Fig. 4(b)). The front is denoted by f_l in Fig. 3(a) and is represented separately in Fig. 3(c).
Fig. 3(a) depicts the initial population I_{pop}, giving rise to the Pareto front f_P. In figure it is also illustrated the local front obtained from another experiment. In 80.1%of the tests carried out (in a set of 21 simulations) the algorithm converges to the optimal Pareto front f_P. In the rest of the cases the algorithm converges to the local front f_l.



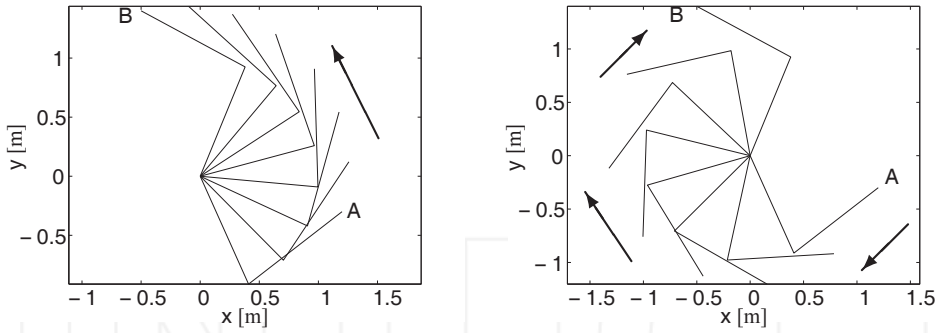
(a) Pareto front (f_P), local front (f_l) and initial population (I_{pop})

(b) Optimal Pareto front f_P

(c) Local front f_l

Fig. 3. The local fronts and the initial population

One of the problems associated with the use of EAs, in the context of optimization problems, is called premature convergence, which consists in the incapacity of the population to converge the global optimal (Bäck, 1996). This phenomenon arises when the objective function has an local optimal with a large base location or the local optimal is in a small spectrum. Therefore, the relationship between the convergence to the global optimum and the geometry function is very important. If the population of the EA is 'trapped' in a large local geometry, then it is difficult for the variation operators to produce an offspring with better performance than its parents. In the second case, if the global optimum is located in a geometry relatively small and the population did not found it until the moment, then variation operators have a low probability of producing descendants in these regions is very low. This happens in 19.9%of the cases, when the algorithm converges to local front getting trapped in it. Indeed, the optimal front solutions are so different from Pareto front, that operators have an extremely low probability to transform a solution of the local front in a solution of Pareto front.
In Fig. 5 is represented the configuration of the robotic manipulator and the joints displacement for extreme solutions of the Pareto optimal front, that are represented by solutions '*a*' and '*b*' in Fig. 3(b).



(a) A trajectory of Pareto optimal front f_P (b) A trajectory of local (non-optimal) front f_l

Fig. 4. Trajectories of the local fronts for the 2*R* manipulator

5. Experiments with the 2*R* manipulator

This section presents other experiments for the 2*R* manipulator, when are optimized five objectives, two by two at a time (2*D*), or the five criteria simultaneously (5*D*).

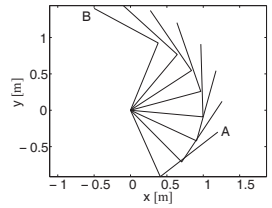
In Fig. 6 the Pareto optimal fronts are illustrated when two criteria are optimized a time, namely the ten cases: $O(q, \dot{q})$, $O(q, E_a)$, $O(q, p)$, $O(q, \dot{p})$, $O(\dot{q}, E_a)$, $O(\dot{q}, p)$, $O(\dot{q}, \dot{p})$, $O(E_a, p)$, $O(E_a, \dot{p})$ and $O(p, \dot{p})$. In the figures it can be seen a good distribution of the solutions along the Pareto fronts for all experiments.

In Fig. 7 are illustrated the results when the algorithm optimizes the joint distance and the joint velocity $O(q, \dot{q})$. The solution $s_{\dot{q},min}$ corresponds to the case where the $\dot{q}$ objective value is minimal. As expected the solution does not present any ripple (see Fig. 7(b)). On the other hand, in Fig. 8 is illustrated the results of the solution $s_{E_a\ min}$ for the optimization $O(q, E_a)$. It can be observed that the gripper describes a trajectory spending the lowest possible energy. Finally, Fig. 9 depicts the results when is minimized the joint distance and the Cartesian ripple. When is chosen an intermediate solution of a previous Pareto front, the behavior is a combination of the extreme solutions of that Pareto front.

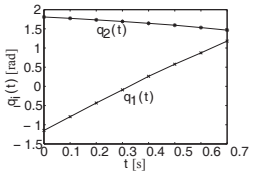
In a second phase, the 2*R* manipulator motion is optimized when the five criteria (5*D*) are considered simultaneously. Figures 10 and 11 show the results for a number of generations $T_t = 50,000$ and a population size of $pop_{lim} = 1000$.

Fig. 10 shows the tradeoff of normalized values for the final population and the best solutions $s_i = \{s_1, s_2, s_3, s_4, s_5\}$ for the objectives $O_i = \{O_q, O_{\dot{q}}, O_{E_a}, O_p, O_{\dot{p}}\}$, respectively. Table 2 depicts the range of values obtained in the simulation. It can be observed that the algorithm 5*D* does not leads to results so good as those obtained with the algorithm 2*D*, due to the significant increase of the search space complexity. However, the solutions show a good distribution (Fig. 10(a)) and the results are close to those of the 2*D* fronts. It must be noted that, when it is considered only the 2*D* optimization it may happen that the 2*D* optimal Pareto front contains dominated solutions from the 5*D* optimization. These solutions are not taken into account when considering the 5*D* optimization.

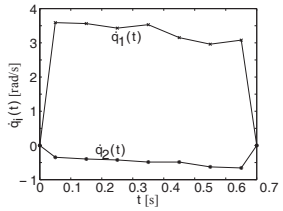
From Fig. 10(b) it can be concluded that O_q and $O_{\dot{q}}$, or O_p and $O_{\dot{p}}$, are conflicting objectives with a small tradeoff between them. Moreover, the objective O_{E_a} presents the greatest compromise between the other objectives. Fig. 11 presents the best solution s_i, $i = \{1, ..., 5\}$, in the viewpoint of the objective i. For the trajectory under study, the results indicate that the closer the gripper moves near its base lower is the energy consumption (figures 11(e), 11(g) and 11(h)).



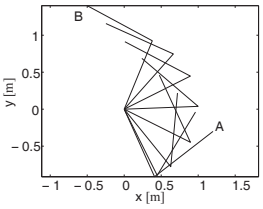
(a) Successive configurations, solution a of Fig. 3(b))



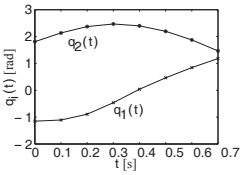
(b) Joint position trajectory vs. time, solution a of Fig. 3(b)



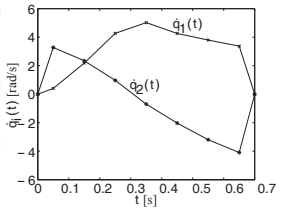
(c) Joint time evolution vs. time, solution a of Fig. 3(b)



(d) Successive configurations, solution b of Fig. 3(b)



(e) Joint position trajectory vs. time, solution a of Fig. 3(b)



(f) Joint time evolution vs. time, solution b of Fig. 3(b)

Fig. 5. Trajectories of optimal the Pareto front for the $O(q, p)$ optimization and the 2*R* manipulator

	O_q (rad^2/s^2)	$O_{\dot{q}}$ (rad^4/s^4)	O_{E_a} (J)	O_p (m^2/s^2)	$O_{\dot{p}}$ (m^4/s^4)
Minimum	79.8	18.2	1056.7	83.5	15.0
Maximum	182.3	101.7	4602.7	121.8	56.4

Table 2. Range of the objectives with 5*D* optimization and the 2*R* manipulator

Fig. 12 illustrates the projections of the 5*D* front into the distinct 2*D* planes. In each plane, is also plotted the 2*D* non-dominated front obtained when the corresponding optimization is executed. It can be observed that the 5*D* optimization does not produce results as good as the 2*D* optimization. This is due to the increasing of the search space, reducing the ratio between the population size and the search space dimension. The distance between the fronts 2*D*, 3*D*, 4*D* and 5*D* increases with the number of objectives.

6. The 3*R* manipulator trajectory with 5*D* optimization

In this section the trajectory of a 3*R* manipulator is optimized considering simultaneously the five objectives listed in (2) and one circular obstacle with center at $c = (1.0, 0.4)$ and radius $\rho = 0.4$. Each link of the manipulator has a length of $l = 0.67$m and a mass of $m = 0.66$Kg. The initial and final configurations, $A \{q_1, q_2, q_3\} = \{-1.374, 1.129, 1.129\}$ and $B \{q_1, q_2, q_3\} = \{-1.050, 0.909, 0.909\}$ are comparable to the one adopted for the 2*R* manipulator.

Fig. 13 illustrates the normalized fitness values of the population and the solutions that have the best performance for each objective in the optimization. This simulation yields also a good distribution of the solutions. The objectives are also quarrelsome but the relative tradeoffs between them have changed. Table 3 depicts the range of objective values achieved by the manipulator in a single execution.

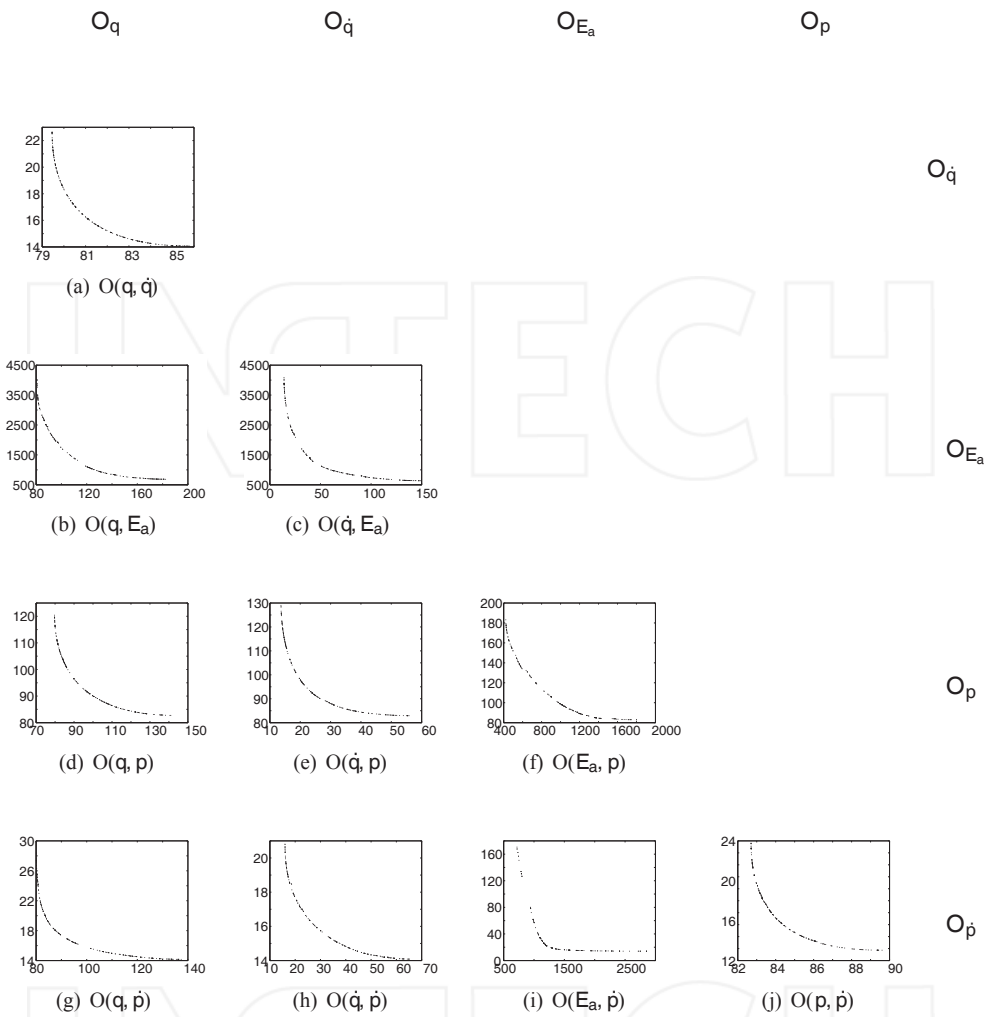
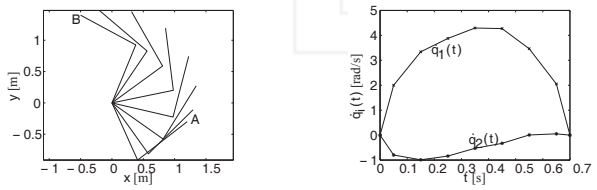


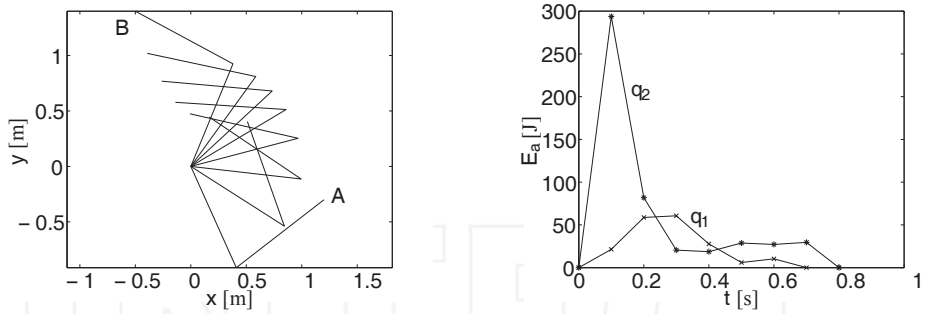
Fig. 6. The 2*D* Pareto optimal front for the ten cases: $O(q, \dot{q})$, $O(q, E_a)$, $O(q, p)$, $O(q, \dot{p})$, $O(\dot{q}, E_a)$, $O(\dot{q}, p)$, $O(\dot{q}, \dot{p})$, $O(E_a, p)$, $O(E_a, \dot{p})$ and $O(p, \dot{p})$ and the 2*R* manipulator



(a) Successive 2*R* robot configurations

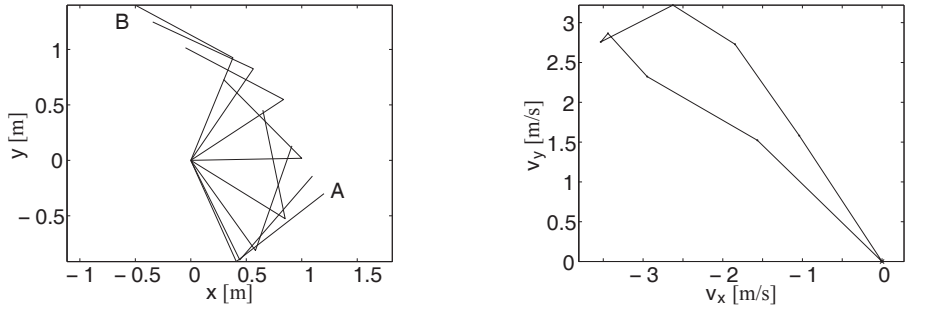
(b) joint time evolution vs. time

Fig. 7. The solution $s_{\dot{q}\,min}$ of the $O(q, \dot{q})$ optimization for the 2*R* manipulator



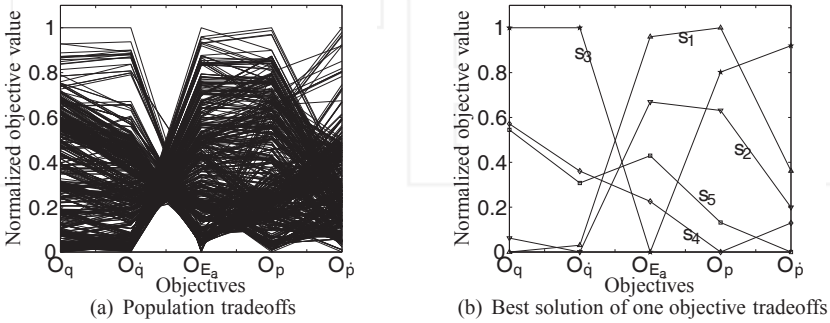
(a) Successive 2*R* robot configurations (b) Energy required E_a

Fig. 8. The solution $s_{E_a\,min}$ of the $O(q, E_a)$ optimization for the 2*R* manipulator



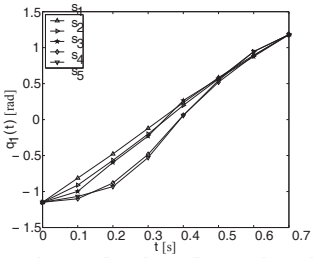
(a) Successive 2*R* robot configurations (b) Cartesian time evolution v_x *vs.* v_y

Fig. 9. The solution $s_{\dot{p}\,min}$ of the $O(q, \dot{p})$ optimization for the 2*R* manipulator

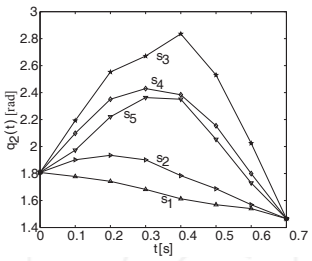


(a) Population tradeoffs (b) Best solution of one objective tradeoffs

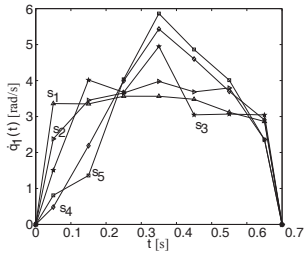
Fig. 10. Value path method representation of the O_q, $O_{\dot{q}}$, O_{E_a}, O_p and $O_{\dot{p}}$ objectives for the 2*R* manipulator



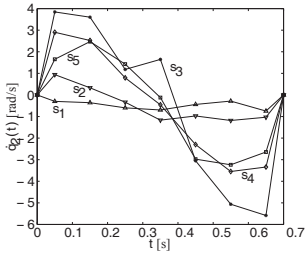
(a) Joint 1 position vs. time



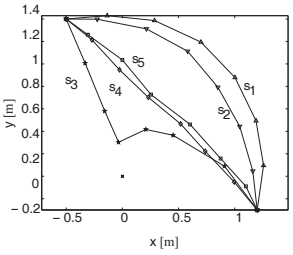
(b) Joint 2 position vs. time



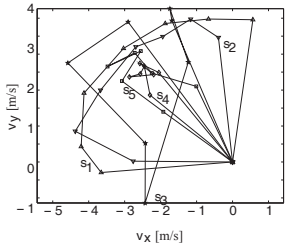
(c) Joint 1 time evolution vs. time



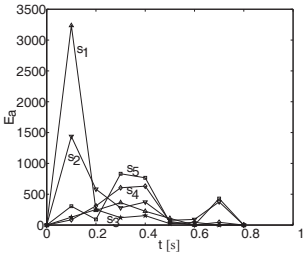
(d) Joint 2 time evolution vs. time



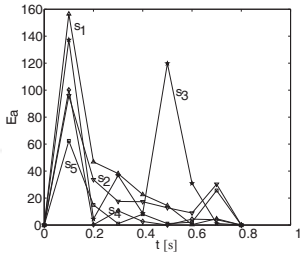
(e) Cartesian movement



(f) Cartesian time evolution



(g) Joint 1 energy required vs. time



(h) Joint 2 energy required vs. time

Fig. 11. Behavior of the best solutions obtained for the 2*R* robot with 5*D* optimization

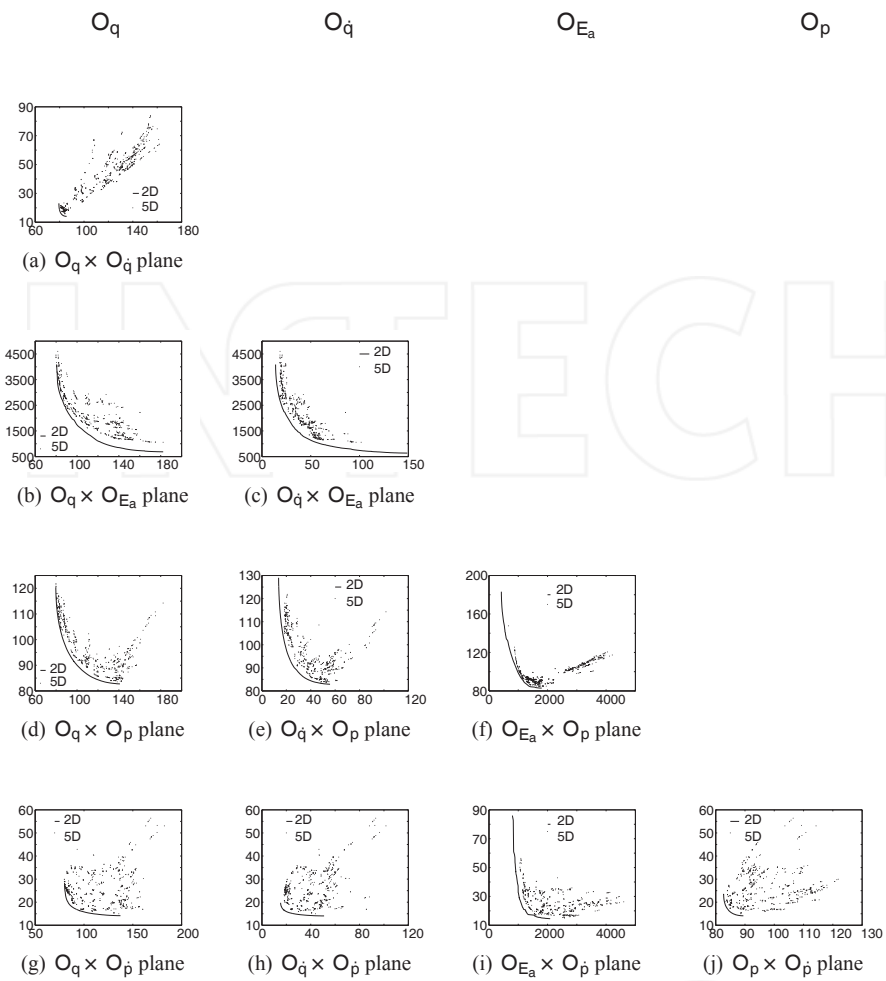


Fig. 12. Projections of the 5*D* Pareto optimal front for the 2*R* manipulator

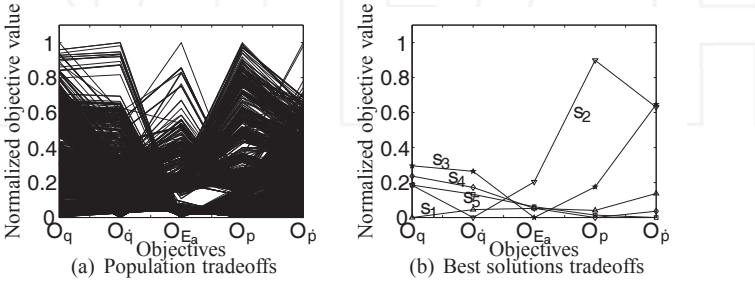


Fig. 13. Value path method representation of the O_q, $O_{\dot{q}}$, O_{E_a}, O_p and $O_{\dot{p}}$ objectives for the 3*R* manipulator

	O_q (rad^2/s^2)	$O_{\dot{q}}$ (rad^4/s^4)	O_{E_a} (J)	O_p (m^2/s^2)	$O_{\dot{p}}$ (m^4/s^4)
Minimum	155.1	52.8	446.9	74.4	12.9
Maximum	638.0	731.9	20531.3	333.7	108.3

Table 3. Range of the objectives with 5*D* optimization and the 3*R* manipulator

Fig. 14 shows the projections of 5*D* front into the different 2*D* planes. It is apparent that the solutions have a good distribution across the search space. From the figures it can be observed two optimal fronts. This 'discontinuity', of the optimal solutions, is due to all optimal solutions are not obtained when the manipulator moves in same direction.

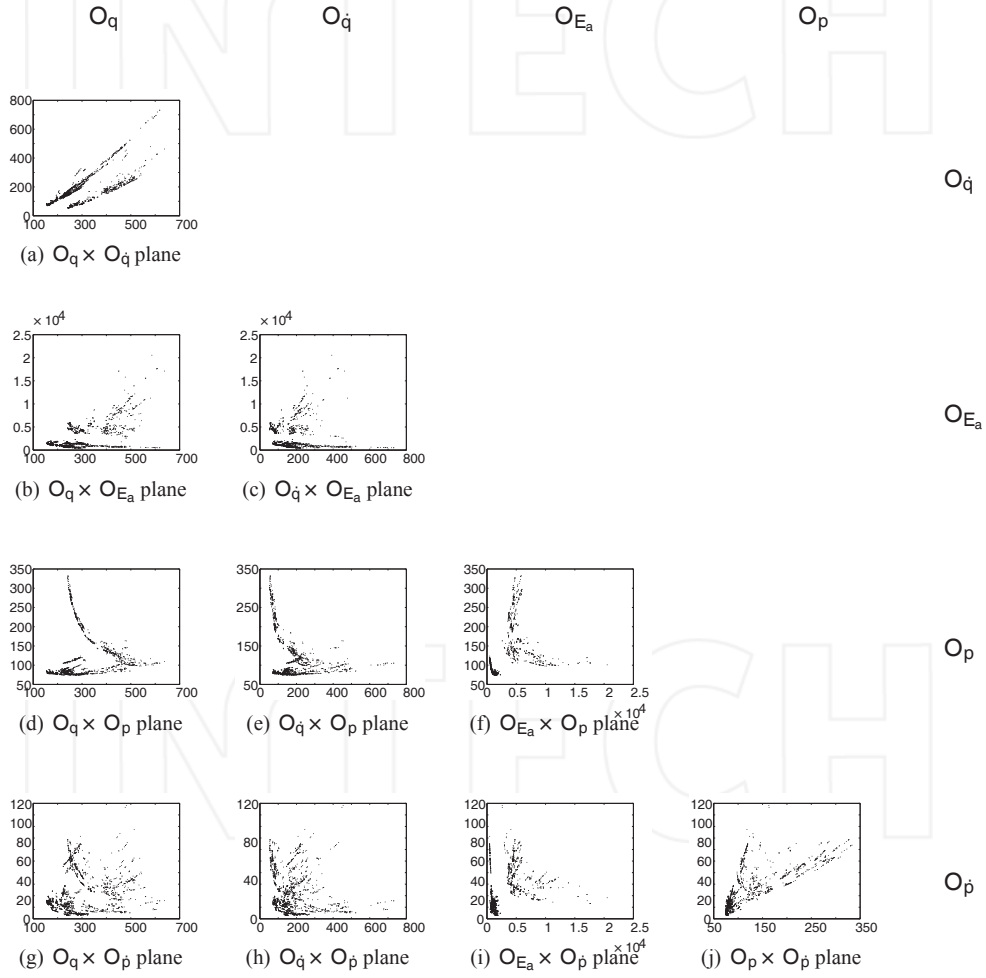
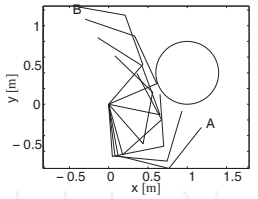
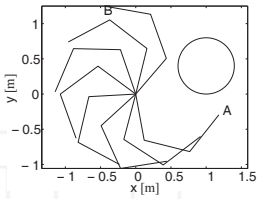


Fig. 14. The 5*D* Pareto optimal front projection for the 3*R* manipulator

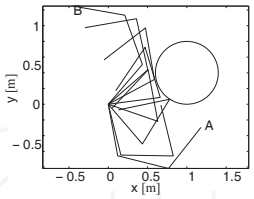
Fig. 15 presents the best solutions for each objective. Most of them are obtained when the manipulator moves in the counterclockwise direction. However, the solution with lower $O_{\dot{q}}$ results from manipulator clockwise movement.



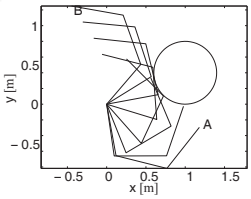
(a) Successive configurations for the achieved best solution in the viewpoint of the joint distance (O_q) objective



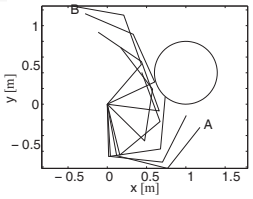
(b) Successive configurations for the achieved best solution in the viewpoint of the joint ripple ($O_{\dot{q}}$) objective



(c) Successive configurations for the achieved best solution in the viewpoint of the energy (O_{E_a}) objective



(d) Successive configurations for the achieved best solution in the viewpoint of the Cartesian distance (O_p) objective



(e) Successive configurations for the achieved best solution in the viewpoint of the Cartesian ripple ($O_{\dot{p}}$) objective

Fig. 15. Results of the best solutions for 3*R* manipulator with one obstacle in the workspace, considering the five objectives optimization criteria

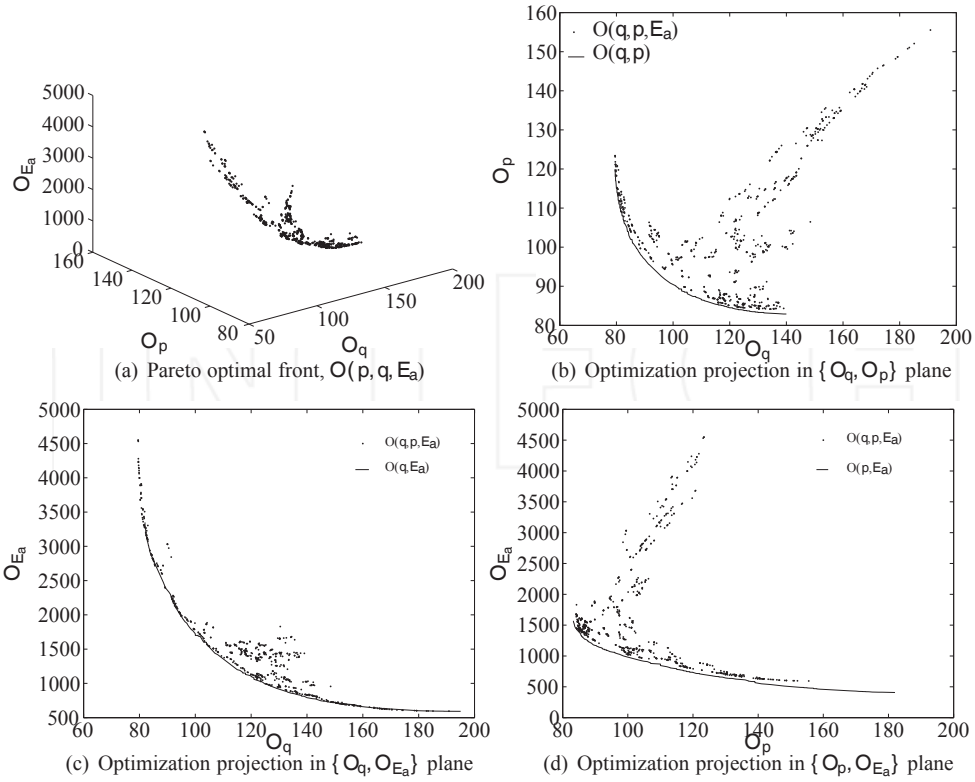
7. Additional experiments

In this section two tests are presented. The first experiment involves a 2*R* manipulator and three objectives. This simulation intends to emphasize the form of the optimal Pareto front for the 3*D* case. The second experiment involves a 3*R* manipulator and one obstacle, in order to study the effect of the obstacle in the non-dominated front.

During the experiments a 600 population elements is used over a total of 30000 generations. The other parameters remain the same as those earlier sections.

Fig. 16(a) depicts the results for the 2*R* manipulator when considering three objectives: the joint distance, the Cartesian distance and the energy required by the end effector to perform the task. The obtained front is continuous and has an 'Y' shape.

Figures 16(b)–16(d) show the Pareto optimal front projected into the different planes: $\{q, p\}$, $\{q, E_a\}$ and $\{p, E_a\}$. The chats are also overlapped the solutions obtained through the optimizations considering the two objectives under analysis. Comparing the results, it can be concluded that the algorithm with three objectives converges for the optimal front similarly to the Pareto front obtained with the two corresponding objectives.



(a) Pareto optimal front, $O(p, q, E_a)$ (b) Optimization projection in $\{O_q, O_p\}$ plane

(c) Optimization projection in $\{O_q, O_{E_a}\}$ plane (d) Optimization projection in $\{O_p, O_{E_a}\}$ plane

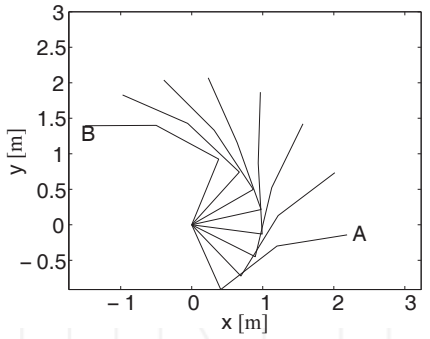
Fig. 16. The Pareto optimal front and its projections for the 2*R* manipulator and three objectives

In Fig. 17, a 3*R* manipulator is considered to move between two configurations $A = \{q_1, q_2, q_3\} = \{-1.15, 1.81, -0.50\}$ and $B = \{q_1, q_2, q_3\} = \{1.18, 1.47, 0.50\}$. The manipulator has link length $l_i = 1$ m length and mass $m_i = 1$ kg of mass, for $i = \{1, 2, 3\}$. The results concern environment with and without obstacles. The obstacle is a circle with center at $c = (2.0, 2.0)$ and radius $\rho = 1$. In Fig. 17(c) are represented the optimal Pareto fronts f_1 and f_2 for the workspace with and without obstacles, respectively. Thus, when the obstacle is introduced the optimal Pareto front is reduced. Consequently, the environment with obstacles does not lead to values with a cost as low as in the case of no obstacles, as can be seen by Figures 17(a), 17(b) and 17(c). At the other end of the front, the resulting solutions '*b*' and '*d*' are almost identical, therefore, it can be concluded that the optimization of the gripper Cartesian distance is not affected by inclusion of the obstacle in the workspace.

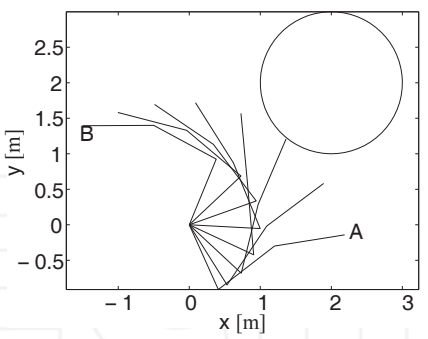
8. Conclusions

This chapter addresses the planning of robotic trajectories in a multiobjective optimization perspective, for finding a set of solutions belonging to the Pareto optimal front. The results demonstrate clearly that the algorithm finds the Pareto front or at least one very close.

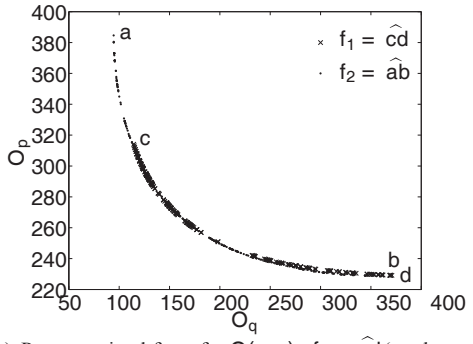
Additionally, it is concluded that the algorithm leads to solutions with a good distribution along the front. The presence of obstacles in the environment can affect the optimal Pareto front but is not an



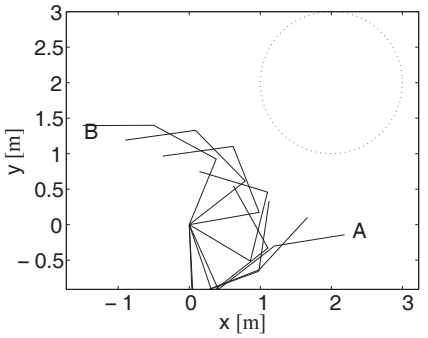
(a) Successive configurations, solution *a*



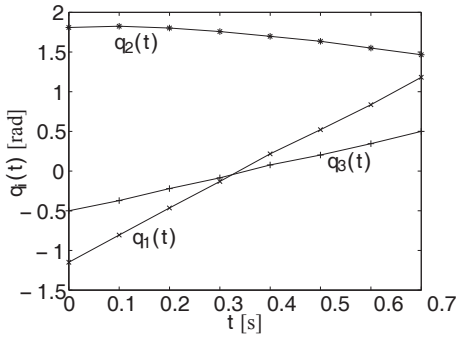
(b) Successive configurations, solution *c*



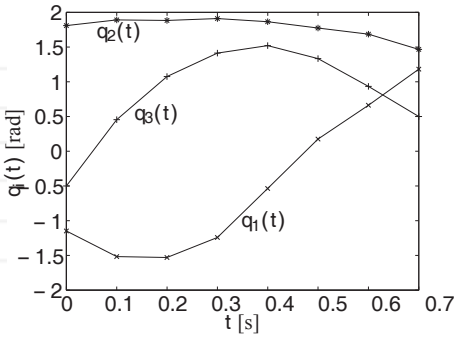
(c) Pareto optimal front for $O(q, p)$: $f_1 = \widehat{cd}$(workspace with one obstacle), $f_2 = \widehat{ab}$(workspace with one obstacle)



(d) Successive configurations, solution *b*



(e) Joint position trajectory, solution *a*



(f) Joint position trajectory, solution *b*

Fig. 17. Pareto optimal front, successive configurations and joint trajectory for the 3*R* manipulator

obstacle to solve the problem. When it is increased the number of objectives, the burden and complexity of the algorithm increases and the quality of final solutions decreases. This disadvantage can be reduced by increasing population and the number of generations. However, the final population considering only the 2*D* optimization may include dominated solutions in the 5*D* optimization point of view. The proposed method helps the decision maker in choosing the best solution since the method gives a set of representative solutions belonging to the non-dominated front.

9. Acknowledgment

The authors would like to acknowledge the GECAD unit.

10. References

Alander, J. (2003). An indexed bibliography of genetic algorithms in robotics, *Technical report*, Department of Electrical Engineering and Production Economics, University of Vaasa.

Ali, A. D. M. S., Babu, N. R. & Varghese, K. (2002). Offline path planning of cooperative manipulators using co-evolutionary genetic algorithm, *Proceedings of International Symposium on Automation and Robotics in Construction, 19th (ISARC)*, National Institute of Standards and Technology, Gaithersburg, Maryland, pp. 415–424.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolutionary Strategies, Evolutionay Programming, Genetic Algorithms*, Oxford University Press, Oxford, New York.

Bäck, T., Hammel, U. & Schwefel, H.-P. (1997). Evolutionary computation: Comments on the history and current state, *IEEE Trans. on Evolutionary Computation* **1**(1): 3–17.

Chen, M. & Zalzala, A. M. S. (1997). A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration, *Journal Robotic Systems* **14**(7): 529–544.

Coello, C. A. C. (1998). A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems* **1**(3): 269–308.

Dae Lee, Y., Hee Lee, B. & Gyoo Kim, H. (1999). An evolutionary approach for time optimal trajectory planning of a robotic manipulator, *Information Sciences* **113**: 245–260.

Davidor, Y. (1991). *Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization*, number 1 in *Series in Robotics and Automated Systems*, World Scientific Publishing Co. Pte Ltd.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, LTD.

Doyle, A. B. & Jones, D. (1996). Robot path planning with genetic algorithms, *2nd Portuguese Conf. on Automatic Control*, Porto, Potugal, pp. 312–318.

Duarte, F. B. M. (2002). *Anï£¡lise de Robots Redundantes*, Phd, Faculdade de Engenharia da Universidade do Porto.

Fonseca, C. M. & Fleming, P. J. (1995). An overview of evolutionary algorithms in multi-objective optimization, *Evolutionary Computation Journal* **3**(1): 1–16.

Garg, D. P. & Kumar, M. (2001). Optimal path planning and torque minimization via genetic algorithm applied to cooperating robotic manipulators, *IMECE – Congress of American Society of Mechanical Engineers*, Ney York.

Garg, D. P. & Kumar, M. (2002). Optimization techinques applied to multiple manipulators for path planning and torque minimization, *Engineering Applications of Artificial Intelligence* (15): 241–252.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison – Wesley.

Horn, J., Nafploitis, N. & Goldberg, D. (1994). A niched pareto genetic algorithm for multi-objective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 82–87.

Kubota, N., Arakawa, T. & Fukuda, T. (1997). Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm, *IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 205–210.

Kubota, N., Fukuda, T. & Shimojima, K. (1996). Trajectory planning of cellular manipulator system using virus-evolutionary genetic algorithm, *Robotics and Autonomous systems* **19**: 85–94.

Lavoie, M.-H. & Boudreau, R. (2001). Obstacle avoidance for redundant manipulators using a genetic algorithm, *Proc. of the 2001 CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, Montréal.

Luo, X. & Wei, W. (2004). A new immune genetic algorithm and its application in redundant manipulator path planning, *Journal of Robotic Systems* **21**(3): 141–151.

Nearchou, A. C. & Aspragathos, N. A. (1995). Application of genetic algorithms to pointo-to-point motion of redundant manipulators, *Mecha. Mach. Theory, Pergamon* **31**(3): 261–270.

Ortmann, M. (2001). Multi-criterion optimization of robot trajectories with evolutionary strategies, *FACTA UNIVERSITATIS, Electronics and Energetics* **14**(1): 19–32.

Rana, A. & Zalzala, A. (1996). An evolutionary planner for near time-optimal collision-free motion of multi-arm robotic manipulators, *UKACC International Conference on Control*, Vol. 1, pp. 29–35.

Ridao, M. A., Camacho, E. F., Riquelme, J. & Toro, M. (2001). An evolutionary and local search algorithm for motion planning of two manipulators, *Journal of Robotic Systems* **18**(8): 463–476.

Silva, F. & Tenreiro Machado, J. (1999). Energy analysis during biped walking, *Proc. IEEE Int. Conf. Robotics and Automation*, Detroit, Michigan, pp. 59–64.

Tian, L. & Collins, C. (2004). An effective robot trajectory planning method using a genetic algorithm, mechatronics, *Mechatronics, In Press* **14**: 455–470.

Wang, Q. & Zalzala, A. M. S. (1996). Genetic control of near time-optimal motion for an industrial robot arm, *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 2592–2597.

Wei-Min, Y. & Yu-Geng, X. (1996). Optimum motion planning in joint space for robots using genetic algorithms, *Robotics and Autonomous Systems* **18**(4): 373–393.

Robot Manipulators New Achievements

Edited by Aleksandar Lazinica and Hiroyuki Kawai

ISBN 978-953-307-090-2
Hard cover, 718 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

Robot manipulators are developing more in the direction of industrial robots than of human workers. Recently, the applications of robot manipulators are spreading their focus, for example Da Vinci as a medical robot, ASIMO as a humanoid robot and so on. There are many research topics within the field of robot manipulators, e.g. motion planning, cooperation with a human, and fusion with external sensors like vision, haptic and force, etc. Moreover, these include both technical problems in the industry and theoretical problems in the academic fields. This book is a collection of papers presenting the latest research issues from around the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

E. J. Solteiro Pires, P. B. de Moura Oliveira and J. A. Tenreiro Machado (2010). Multi-Criteria Optimization Manipulator Trajectory Planning, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), ISBN: 978-953-307-090-2, InTech, Available from: http://www.intechopen.com/books/robot-manipulators-new-achievements/multi-criteria-optimization-manipulator-trajectory-planning

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821