Trajectory Optimization for Redundant Robots Using Genetic Algorithms

E. J. Solteiro Pires
Univ. Trás-os-Montes e Alto Douro – Dep. Eng.
5001 Vila Real, Portugal
epires@utad.pt

J. A. Tenreiro Machado
Inst. Superior de Engenharia do Porto – Dep.Eng.Elec.
4200-072 Porto, Portugal
jtm@dee.isep.ipp.pt

Abstract

This paper proposes a genetic algorithm to generate trajectories for robotic planar manipulators, based on the kinematics and the dynamics.

1 REPRESENTATION AND OPERATORS

The path is encoded as a fix string length in joint space as

$$[(q_{11},...,q_{k1}),..., (q_{1j},...,q_{kj}),..., (q_{1n},...,q_{kn})] \quad (1)$$

The ith joint variable for a robot intermediate jth position is q_{ij}, the chromosome is constituted by n genes (configurations) and each gene is formed by k values, where k is the number of robot links. The value of q_{ij} is represented as a floating-point number.

The tournament selection with elitism is adopted to select the strings along the evolution. Single crossover is used and the crossover point is only allowed between genes (*i.e*. the operator may not disrupt genes). For the mutation operator one gene value is replaced with a given probability and follows the equation:

$$q_{ij}(t+1) = q_{ij}(t) + \mathrm{k_m}\, x \qquad x \sim \mathrm{U}[-1; 1] \quad (2)$$

2 EVOLUTION CRITERIA

The fitness function f, adopted is defined as:

$$f = \begin{cases} -\alpha_1 \dot{q} - \alpha_2 \ddot{q} - \alpha_3 \dot{p} - \alpha_4 \ddot{p} - \alpha_5 \varepsilon - \alpha_6 P_a & nap = 0 \\ +\infty & nap \neq 0 \end{cases} \quad (3)$$

$$\dot{p} = \sum_{w=2}^{n} d\left(p_w, p_{w-1}\right)^2 \qquad \dot{q} = \sum_{j=1}^{n} \sum_{i=1}^{k} \dot{q}_{ij}^{\,2}$$

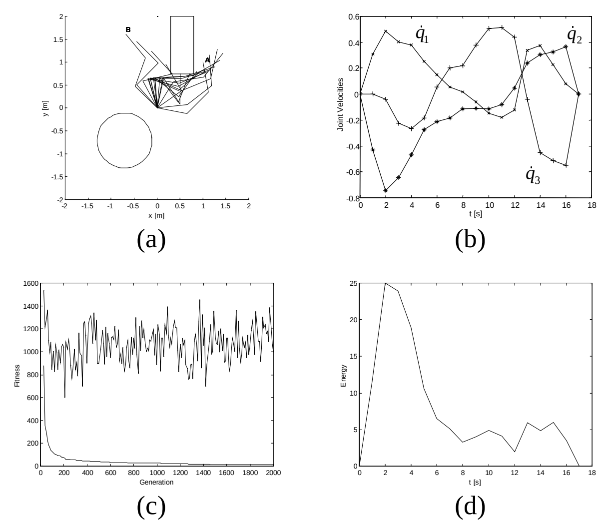
$$\ddot{p} = \sum_{w=3}^{n} \left| d\left(p_w, p_{w-1}\right) - d\left(p_{w-1}, p_{w-2}\right)\right|^2 \qquad \ddot{q} = \sum_{j=1}^{n} \sum_{i=1}^{k} \left|\ddot{q}_{ij}\right|^2$$

$$P_a = \frac{1}{T} P = \frac{1}{T} \sum_{j=1}^{n} \sum_{i=1}^{k} \left| \tau_j \Delta \theta_{ji} \right|$$

The functions $\dot{q}$, $\ddot{q}$, $\dot{p}$, $\ddot{p}$, ε and P_a are responsible for minimizing the manipulator traveling distance, the ripple in time evolution of positions and velocities, the total trajectory length, the distance between the simulation and desired points and the manipulator energy consumption, respectively. The variable p_w is the robot w intermediate arm cartesian position and $d(\cdot,\cdot)$ is a function that gives the distance between the two arguments. The points not admissible (*nap*) give a conflict measure between the robot and the obstacles.

3 SIMULATION RESULTS

The experiments consist on moving a robotic arm from the starting point A≡(1,1) up to the final point B ≡ (–0.6697, 1.6168). The results for a 3*R* robot are given in figure 1 for a crossover and mutation probabilities of $p_c = 0.8$ and $p_m = 0.1$ respectively, $\mathrm{k_m} = 1.8$, 100-string population, string lengths of $n = 16$ and α =(1/3, 1/3, 2, 2, 20, 0.01).



(a) (b) (c) (d)

Figure 1: Results for the 3*R* robot with 2 obstacles. (a) Successive configurations, (b) Joint velocities *versus* time, (b) The best individual evolution and the fitness mean evolution *versus* generation, (d) Power *versus* time.

4 CONCLUSIONS

An off-line *GA* trajectory planner for robots, based on the kinematics and the dynamics was presented. The algorithm is able to reach a determined goal with a reduced ripple both in the space trajectory and the time evolution.