Obstacle Avoidance for Redundant Manipulators Using a Genetic Algorithm

Marie-Hélène Lavoie *Roger Boudreau*

École de génie, Université de Moncton, Moncton, NB, Canada, E1A 3E9

1. Introduction

A task extensively performed in today's manufacturing process is the pick-and-place operation. Manipulators or robots used for such tasks are often working in environments clustered with obstacles that must be avoided. In order to perform such complex tasks, redundant manipulators, i.e. manipulators having a higher number of degrees-of-freedom (DOF) then that required for a configuration are used. The main problem associated with this higher number of DOF is the complexity of the inverse kinematics problem since it leads to an infinity of solutions. This whole problem can be viewed as an optimization problem with goals and constraints consisting of planning the end-effector (EE) trajectory, avoiding collisions between the robot and the obstacles, and finally, staying within the joints' physical limits.

Some researchers have tried to solve this problem using different methods such as Neural Networks [1], Simulated Annealing [2,3] and Genetic Algorithms (GA) [4]. In [4], Binary-Coded GAs are used in two different levels; one to find the angles and another to generate small changes in these angles. In this paper, a Real-Coded GA is used with one level only to solve the inverse kinematics of planar redundant manipulators for pick-and-place operations while avoiding obstacles in clustered environments. Obstacle avoidance is also verified as the robot moves between two points, contrary to [4].

2. Genetic Algorithm

Initially developed by Holland [5], the GA is based on the theory of evolution. In the GA, each individual in a population represents a possible solution to the problem. A new population of individuals is created from the reproduction, the transmission of genetic heritage and, sometimes, the mutation of individuals from a previous generation. Fitter individuals will be selected and reproduced more often. After several generations, this robust and powerful procedure will converge towards the solution.

Usually, Binary-Coded GAs are used as in [4], but Davies [6] demonstrated that Real-Coded GAs perform better, and are thus used in this work.

3. Application

The manipulators used here are serial planar ones with N actuated revolute joints. The obstacles in their working space are described by ellipses as in [3] since they are easily modelled mathematically. Starting from a specified initial position, the inverse kinematics must be solved such that the EE passes through a via point before finally attaining a goal point while avoiding any collisions with the obstacles. Two sub-paths have to be optimized. The first sub-path, SP1, is between the initial EE position and the via point, and the second one, SP2, between the via point and the final point.

Let P_A and P_E be, respectively, the actual and desired ending positions of the EE of a sub-path. A position error can thus be defined by

$$pos\_err = \left\| P_A - P_E \right\| \tag{1}$$

Then, let $\theta_{S\,i}$, $\theta_{A\,i}$ (i=1…N) respectively be the joint angles corresponding to the starting and actual positions. The total joint displacement can thus be defined by

$$jnt\_disp = \sqrt{\sum_{i=1}^{N} (\theta_{S\,i} - \theta_{A\,i})^2} \tag{2}$$

To verify if the path is collision-free, the manipulator displacement is discretized in M positions. Let θ_{Aij} be the actual angle of joint i at the discretized position j, then

$$\theta_{A\,i\,j} = \theta_{S\,i} + \tfrac{j}{M}(\theta_{A\,i} - \theta_{S\,i}) \tag{3}$$

For each position, any interference between the manipulator and an obstacle has to be found. The interference with an obstacle along the path of two successive positions of every joint must be determined as well.

4. Simulations

4.1. Seven dof manipulator

The manipulator, shown in Fig.1 has seven joints (N=7) with links 1 to 7 of length 10, 10, 10, 4, 4, 3 and 2 units. The joint angle domains, in radians, are [0…2.1] for the first joint and [-2.1,…3.9] for the others. In the initial position, the joint angles are 1.8, 1, -1, -1, 0, 0 and 0 radians for joints 1 to 7 respectively. The EE should pass through the via point (15, 5) and end up at position (18, 8.5) while avoiding any contact between the links and the box.

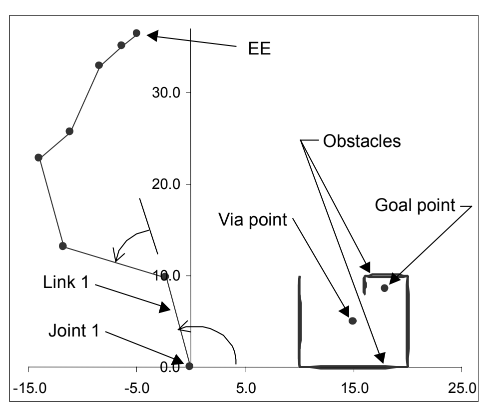

Fig.1: Initial position for 7 dof manipulator.

An objective function, OF, is formulated to evaluate the performance of the manipulator, based on the N variables $\theta_{A\,i}$ to be optimized. This function takes into account the positioning error, the total joint displacement and the collision avoidance criteria. Equation 4 represents this OF that is minimized in the GA. The parameters w_1 and w_2 are weighting factors set to 1.0 and 0.7 respectively. Moreover, another constraint is added to the OF which states that P_A of sub-path SP2 must lie within the box formed by the five ellipses. Each path is discretized in 10 positions (M=10) for the collision avoidance criteria.

$$OF = \begin{cases} \infty & \text{if collision} \\ \infty & \text{if } P_A \text{ outside box} \\ w_1\, pos\_err + w_2\, jnt\_disp & \text{otherwise} \end{cases} \quad (4)$$

The GA has a population of 800, a probability of crossover and of mutation of 0.3 and 0.2 respectively and elitism is applied. The algorithm is stopped whenever the OF is smaller than 6.0 for SP1, or 2.0 for SP2, or when a maximum number of 100 generations is reached.

4.2. Ten dof manipulator

The manipulator of the second simulation has ten joints (N=10). The lengths of links 1 to 6 are 6 units and the others are 2 units. The joint angle domains, in radians, are [-2.0,…2.0] for all the joints. The initial angles are 1, 1.5 and 1.5 radians for joints 1 to 3 respectively, -1 for joints 4 to 7 and 0 for the others. The via point is (10, -10), the final position is (13, -6.5), and the obstacles are the same as in the previous simulation, except that they are all translated by –5 units vertically and by –15 units horizontally. The GA parameters and the OF are the same as before.

5. Results

The two simulations are coded in C++. In the computer program used, the GA for SP1 is first called. Then, if the optimization is successful, the GA for SP2 is called. Finally, those two steps are in a loop that will end only if both optimizations are successful or when it has been executed up to ten times. Fig.2 shows a solution obtained with the 7 dof manipulator.

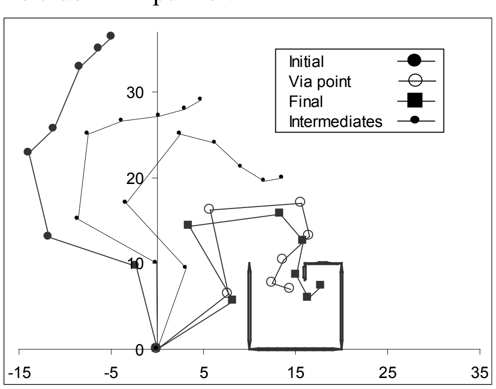

Fig. 2: Solution with 7 dof manipulator.

A thousand experiments were conducted for each simulation. Fig.3 displays the probabilities of obtaining a solution with a positional error less than *x* units. As we can see from the figure, the probabilities of obtaining a position error of less than 1 or 2 units are respectively 93.5 % and 99.8 % for the 7 dof manipulator. Those same probabilities are respectively 99.1 % and 100 % for the 10 dof manipulator. The average times for one run on a Pentium II 400 MHz computer are indicated in Fig. 3.

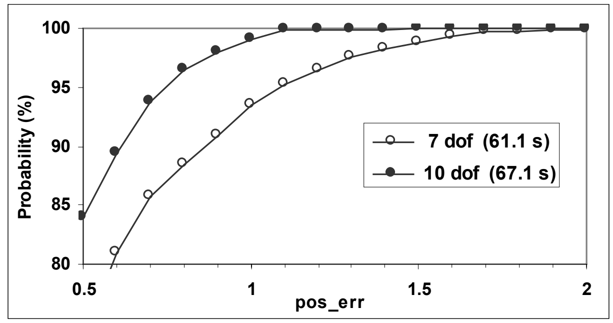

Fig.3: Probability of obtaining a position error less than x units

6. Conclusion

The probabilities indicated in Fig. 3 show the efficiency of the GA. The method used is simpler and has a better performance than that used in [4]. The performance of the GA in [4] was compared and found to be much better than the traditional Pseudoinverse method. No starting solution has to be given to the GA to assure a good convergence. Finally, if a better precision is required, the GA should be stopped at a smaller value of the OF. In this case, the maximum number of generations may have to be increased.

7. Acknowledgements

The financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

8. References

[1] Z. Mao and T.C. Hsia: “Obstacle avoidance inverse kinematics solution of redundant robots by neural network”, *Robotica*, 15(1), 3-10 (1997).

[2] F. Janabi-Sharifi and W.J. Wilson: “A Fast Approach for Robot Motion Planning” *J. of Intelligent and Robotic Systems*, 25, 187-212 (1999).

[3] H. Martinez-Alfaro and D.R. Flugrad: “Optimal collision-free path planning for robots using b-splines and simulated annealing ”, *Robotics: Kinematics, Dynamics and Controls*, 72, 323-331 (1994).

[4] A.C. Nearchou: “Solving the inverse kinematics problem of redundant robots operating in complex environments via modified genetic algorithm ”, *Mech. Mach. Theory*, 33(3), 273-292 (1998).

[5] J.H. Holland: Adaptation in natural and artificial systems, The University of Michigan Press, Ann Arbor, 1975.

[6] L. Davies: Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.