OPTIMIZATION OF ROBOT ARM PATH PLANNING WITHOUT COLLISION IN ITS WORKSPACE USING GENETIC ALGORITHM

A. Chennakesava Reddy[#], P. Ram Reddy[$] and B. Kotiveerachari[*]

[#]Associate Professor in Mechanical Engineering, JNTU College of Engineering
Kukatpally, Hyderabad –500 072, India
dr_acreddy@yahoo.com

[$]Registrar, Jawaharlal Nehru Technological University, Kukatpally, Hyderabad – 500 072
[*]Professor of Mechanical Engineering, National Institute of Technology, Warangal

ABSTRACT: In the last decade genetic algorithms have been applied in a plethora of fields such as in control, parameter and system identification, robotics, planning and scheduling, image processing, pattern recognition and speech recognition. This paper addresses the generation of a robotic manipulator structure and the planning of trajectories, namely in finding a continuous motion that takes the hand from a given starting configuration, without collision with any obstacle, up to a desired end position in the workspace.

Key words: robot arm, path planning, collisions, workspace, genetic algorithm

1.0 INTRODUCTION

Various methods for trajectory planning, collision avoidance and manipulator structure definition have been proposed. A possible approach consists in adopting the differential inverse kinematics, using the Jacobian matrix, for generating the manipulator trajectories [1, 2, 3]. However, the algorithm must take into account the problem of kinematic singularities that may be hard to tackle. To avoid this problem, other algorithms for the trajectory generation are based on the direct kinematics [3 -8].

This paper proposes a method to obtain a robot arm and its path. This method is based on a genetic algorithm (*GA*) adopting the direct kinematics. The optimal manipulator is the one that minimizes both the path trajectory length and the ripple in the time evolution, without any collision with the obstacles in the workspace.

2.0 ROBOT ARM AND ALGORITHM FORMULATION

In this study robotic manipulators are required to move from an initial point up to a given final point. In the experiments, the planar manipulators are used with rotational and prismatic joints. The link length arms are in the range [0, 1] *m*, and the robot rotational joints are free to rotate 360º. Therefore, the manipulator workspace is a circle with a 4 *m* maximum radius, which may have obstacles such as rectangles and circles. To test a possible collision between the manipulator and the obstacles, the arm structure is discretized into several points and then these points are checked in order to verify if they are inside any obstacle.

In what concern the structure generator, it is adopted a *GA* to search for a global optimal robot which presents the best performance. The mechanical structure consists of a set of strings that represent the type of joint and the link lengths. On the other hand, the trajectory generator uses a GA scheme to search for an optimal robot path. The trajectory consists in a set of strings that represent the joint positions between the initial and final robot configurations. In conclusion, in this work are adopted four *GAs*. One *GA* is used to calculate the robot's structure. For each arm two *GAs* are used to calculate the initial and final configurations of the trajectory. Finally, another *GA* determines the intermediate configurations between the two points calculated previously.

3.0 MODELING

The robotic structure is encoded as:

$$[(J_1, l_1),..., (J_i, l_i),..., (J_k, l_k)] \quad (1)$$

where J_i represents the type of the ith joint (R for rotational and P for prismatic joints) and l_i is the ith link length, in the range [0, 1] m. In order to limit the computational time the number of dof is limited to $k \leq 4$. All values used in this work are encoded through real values except the type of the robotic link.

The initial and the final configuration are encoded as:

$$[q_1,...,q_k] \quad (2)$$

The path is encoded, directly, as strings in the joint space to be used by the *GA* as:

$$[(q_{11},...,q_{k1}),..., (q_{1j},...,q_{kj}),..., (q_{1n},...,q_{kn})] \quad (3)$$

The ith joint variable for a robot intermediate jth position is q_{ij}, the chromosome is constituted by n genes (configurations) and each gene if formed by k values. The values of q_{ij} are initialized in the range [360º, +360º] for R joints and [0, 1] m for the case of P-joints. It should be noted that the initial and final configurations have not been encoded into the string because this configuration remains unchanged throughout the trajectory search. Without losing generality, for simplicity, it is adopted a normalized time of $\Delta t = 1$ sec between two consecutive configurations, because it is always possible to perform a
time rescaling.

4.0 OPERATORS IN THE GENETIC ALGORITHM

The initial populations of strings are generated at random. The search is then carried out among these populations. The three different operators used in the genetic planning are reproduction, crossover and mutation, as described in the sequel. In what concern the reproduction operator, the successive generations of new strings are reproduced on the basis of their fitness function. In this case, it is used a tournament selection [8] to select the strings from the old population, up to the new population. For the crossover operator, the strings in the new population are grouped together into pairs at random. Single crossover is then performed among pairs. The crossover point is only allowed between genes (*i.e.* the crossover operator may not disrupt genes). The mutation operator consists on several actions namely, commuting the type of the joint, modifying the link length and changing the joint variable. Therefore, the mutation operator replaces one gene value with a given probability that follows the equations:

$$q_{ij}(t + 1) = q_{ij}(t) + \mathrm{k_m}\, \varphi_i \quad (4)$$

$$l_i(t + 1) = l_i(t) + \mathrm{k_m}\, \psi_I \quad (5)$$

$$\{\varphi_i, \psi_i\} \sim \mathrm{U}[-1; 1] \quad (6)$$

at generation t, while φi, ψi are uniform random numbers and km a parameter.

Finally, at the end of each GA structure iteration two operators take into action, randomly, over the (J_i, li) genes. One duplicates a given gene while the other removes another gene, with probabilities p_r and p_d, respectively.

5.0 EVOLUTION CRITERIA

Several criteria have been selected to qualify the evolving robotic manipulators. All constraints and criteria are translated into penalty functions to be minimized. Each criterion is computed individually and then, is used in the fitness function evaluation [10].

The fitness function f adopted to evaluate the candidate robots is defined as:

$$f = \beta_1 f_T + \beta_2 f_I + \beta_3 f_F \quad (7)$$

where β_i (i=1,2,3) are weighting factors. The f_I and f_F functions give a measurement of the distance between the initial or final desired point and the point actually reached by the robot configuration. The fitness function f_T, adopted to evaluate the candidate trajectories, is defined as:

$$f_T = \begin{cases} \alpha_1 \dot{q} + \alpha_2 \ddot{q} + \alpha_3 \dot{p} + \alpha_4 \ddot{p} & \text{nap} = 0 \\ +\infty & \text{nap} \neq 0 \end{cases} \quad (8)$$

where $\dot{q}, \ddot{q}, \dot{p}, \ddot{p}$ and *nap* are the criteria defined in the sequel. The optimization goal consists in finding a set of design parameters that minimize *f* according to the priorities given by the values of α_i (i = 1,…,4).

The joint velocities $\dot{q}$ are used to minimize the manipulator traveling distance yielding the criteria:

$$\dot{q} = \sum_{j=1}^{n} \sum_{i=1}^{k} \dot{q}_{ij}^2 \quad (9)$$

This equation is used to optimize the traveling distance because if the curve length is minimized, then the ripple in the space trajectory is indirectly reduced. For a function $y = g(x)$ the distance curve length is $\int[1 + (dg/dt)^2]\, dx$ and, consequently, to minimize the distance curve length it is adopted the simplified expression $\int(dg/dt)^2\, dx$. The fitness function maintains the quadratic terms so that the robot configurations are uniformly distributed between the initial and final configurations.

The joint accelerations $\ddot{q}$ are used to minimize the ripple in the time evolution of the robot trajectory through the criteria:

$$\ddot{q} = \sum_{j=1}^{n} \sum_{i=1}^{k} \ddot{q}_{ij}^2 \quad (10)$$

The Cartesian velocities $\dot{p}$ are introduced in the fitness function *f* to minimize the total trajectory length, from the initial point up to the final point. This criteria is defined as:

$$\dot{p} = \sum_{w=2}^{n} d(p_w, p_{w-1})^2 \quad (11)$$

where 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 Cartesian acceleration $\ddot{p}$ in the fitness functions is responsible for reducing the ripple in time evolution of the arm velocities. This criteria is formulated as:

$$\ddot{p} = \sum \left[d(p_w, p_{w-1}) - d(p_{w-1}, p_{w-2})\right]^2 \quad (12)$$

The points that are not admissible give a conflict measure between the robot and the obstacles. In this perspective, each manipulator link is discretized and the *nap* value is a criterion consisting on the sum of the manipulator points that are inside the obstacles.

6.0 SIMULATION RESULTS

In this section are presented the results of several simulations. The experiments consist on moving a robotic arm from the starting point A up to the final point B (Table 1), for two types of situations:

- the algorithm optimizes the robot structure for a sequence of *r* trajectories (series optimization), tacking each trajectory at a time;
- the algorithm optimizes the robot structure for the *r* trajectories (parallel optimization), considering all trajectories simultaneously.

The algorithm adopts crossover and mutation probabilities of $p_c = 0.8$ and $p_m = 0.05$ respectively, $p_r = p_d = 0.01$, $k_m = 1.8$, a 30-string population for the robots, a 50-string population for the initial and final configurations and a 100-string population for the intermediate configurations. For the experiment are used strings length of $n = 10$ and the selection operator is based on tournament selection with elitism. The workspace contains an obstacle, a circle with center at the point (0, 2) and radius 0.5.

6.1 OPTIMIZATION OF THE TRAJECTORY 1

For one trajectory only, there is no distinction between the series and parallel optimization methods. Therefore, this section presents the results of trajectory 1 optimization yielding a manipulator with structure {[R: 1.0000] [P: 0.8824] [P: 0.5945] [P: 0.8366]}, where [J_i: l_i] identifies the type of the *i*th joint and the link length. Figure 1 to 2 shows some results of the robotic manipulator obtained.

6.2 SERIES TRAJECTORY OPTIMIZATION

This section presents the results when optimizing sequentially the group of trajectories, one trajectory at a time. The robot structures obtained for the five trajectories are:

{[R: 1.0000] [P: 0.7393] [P: 0.5641] [R: 0.6718]}
{[R: 0.5632] [P: 0.4643] [P: 0.5460] [P: 0.7479]}
{[R: 0.9504] [P: 0.7374] [P: 0.6805] [P: 0.8839]}
{[R: 0.9803] [P: 1.0000] [P: 1.0000] [P: 0.8312]}
{[R: 1.0000] [P: 0.8347] [P: 0.9961] [P: 0.8087]}

The results are shown in Figures 4 to 5.

The abrupt transitions (Figure 5) of the best individual function are due to the change of the optimization trajectory. The step *c* is negative because the length of the new trajectory ($A_4 \rightarrow B_4$) is smaller.

6.3 PARALLEL TRAJECTORY OPTIMIZATION

This section presents the resultant robot when optimizing the five trajectories simultaneously. The final robot mechanical structure is {[R: 1.0000] [P: 0.7053] [P: 1.0000] [P: 0.6010]}. Figures 6 to 8 show the results for this case.

This section shown the results when is optimizing sequentially the group of the trajectories for a workspace with two obstacles. The robot structures obtained for the trajectories are:
{[P: 0.3875] [R: 1.0000] [P: 1.0000] [P: 0.7689]}
{[R: 0.4888] [P: 1.0000] [P: 0.9250]}
{[R: 0.6641] [R: 0.8629] [P: 1.0000] [P: 0.7881]}
{[P: 0.3359] [R: 0.9353] [P: 0.9802] [P: 1.0000]}
{[P: 0.2894] [R: 1.0000] [P: 1.0000] [P: 0.9780]}

The results for the second robot are shown in Fig. 9-10.

6.4 RESULT ANALYSIS

The results are satisfactory because the solutions avoid the obstacles, and the time evolution of the variables presents a small ripple. Moreover, analyzing the final number of axis, we conclude that the larger the number of dof the better the robot ability to maneuver and to reach the desired points.

The different experiments simulated both a workspace without obstacles and a workspace with several types and positions of obstacles. For one obstacle the results reveal that for the first axis we have 88.3%-rotational and 11.7%- prismatic joints, respectively. Therefore, it seems that a robot with a first rotational joint has a superior performance (in the sense of being more adaptable) to execute different tasks. Nevertheless, in the present form, the study does not consider energy requirement [11]. In this line of thought, future work will take into account the robot dynamics and we expect to have clear conclusions about the total number of dof. For more obstacles in the workspace the convergence sees more difficult and further experiments are still required.

7.0 SUMMARY AND CONCLUSIONS

A *GA* robot constructor and its trajectory planner, based on the kinematics, were presented. The algorithm is able to reach a determined goal with a reduced ripple both in the space trajectory and the time evolution. Moreover, any obstacles in the workspace do not represent a difficulty for the algorithm to reach the solution. Since the *GA* uses the direct kinematics the singularities do not constitute a problem. Furthermore, the algorithm determines the robot structure more adaptable to a given number and type of tasks, maintaining good manipulating performances.

REFERENCES

1. Mingwu, C., and Ali, M. S. Z., A Genetic Approach to Motion Planning of Redundant Mobile Manipulator Systems Considering Safety and Configuration, Journal of Robotic System, 14, n. 7, 529-544, 1997

2. Yaval, D., Genetic Algorithms and Robotics, a Heuristic Strategy for Optimization, World Scientific, 1991.

3. Chennakesava Reddy, A., Ram Reddy, P. and Kotiveerachari, B., Optimization of path dependent trajectory for an arm - like serial chain robot using genetic algorithm, National Conference on Recent Trends in CAD/CAM, Bidar, July 2004.

4. Doyle, A.B., and Jones, D.I., Robot Path Planning with Genetic Algorithms, 2nd Portuguese Conference on Automatic Control, pp. 312-218, Porto, Portugal, 1996.

5. Rajasekhara Reddy, B. and Chennakesava Reddy, A., Obstacle avoidance of planar redundant manipulators for pick-and-place operations using real-coded genetic algorithm, National Conference on Advances in Manufacturing Technology, Palghat, February 2003.

6. Rana, A., and Zalzala, A., An Evolutionary Planner for Near Time-Optimal Collision-Free Motion of Multi Arm Robotic Manipulators, International Conference on Controls, vol.1, pp. 29-35, 1996.

7. Chennakesava Reddy, A., Ram Reddy, P., Hazard detection for multi – agent reactive robotic system using on - line software analysis of safety constraints, National Conference on Computer integrated Design and Manufacturing, Coimbatore, November 2003.

8. Naoyuki, K., Takemasa Arakawa, Toshio Fukuda, Trajectory Generation for Redundant Manipulator Using Virus Evolutionary Genetic Algorithm, IEEE International Conference on Robotics and Automation, pp. 205-210, Albuquerque, New Mexico, 1997

9. Q. Wang, Q., and Zalzala, A.M.S., Genetic Control of Near Time-optimal Motion for an Industrial Robot Arm, IEEE International Conference on Robotics and Automation, pp. 2592-2597, Minneapolis, Minnesota, 1996

10. David, E.G., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison – Wesley, 1989

11. Solteiro Pires, E.J., and Tenreiro Machado, J.A., A *GA* Perspective of the Energy Requirements for Manipulators Maneuvering in a Workspace with Obstacles, CEC 2000 - Congress on Evolutionary Computation, pp. 1110-1116, 16-19 July 2000, San Diego, California, USA.

Table-1: Trajectory simulations

Trajectory	Initial point A	Final Point B
1	(2,2)	(-1,2)
2	(-1,2)	(1,1)
3	(1,3)	(1,0)
4	(3,-1)	(1.5,-1)
5	(-1,-3)	(-1,-1)

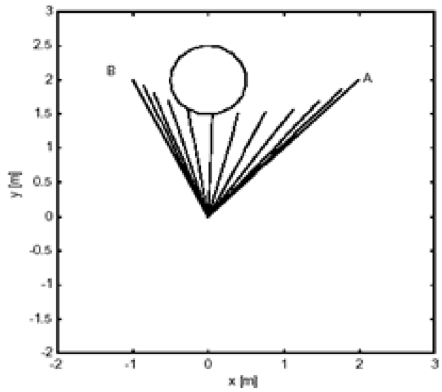

Fig.1 Successive robot configuration for trajectory-1

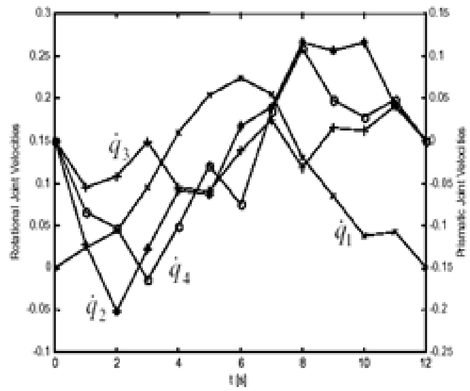

Fig.2 Variation of joint velocities

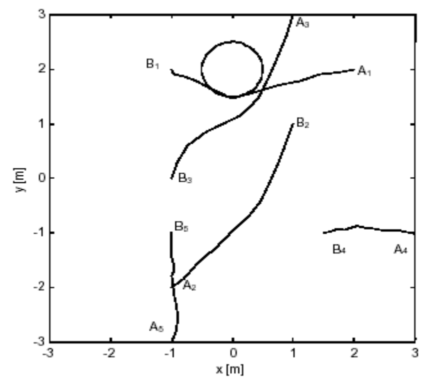

Fig.3 Robot arm trajectories

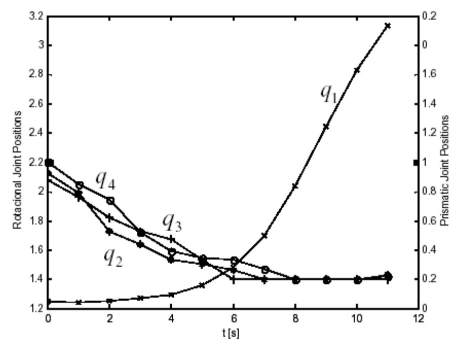

Fig.4 Variation of joint positions of trajectory-3

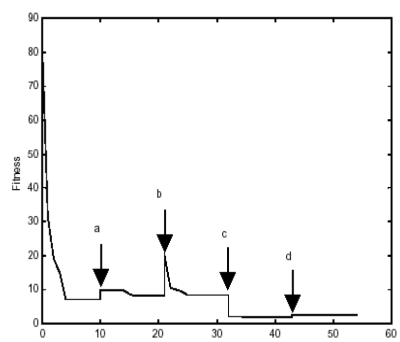

Fig.5 The best individual evolution versus the generation

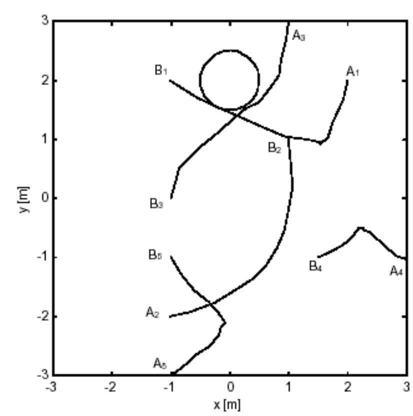

Fig.6 Terminal arm position for the trajectories

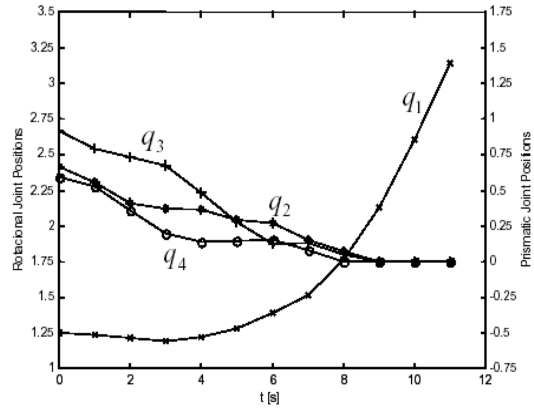

Fig.7 Joint position of trajectory - 3

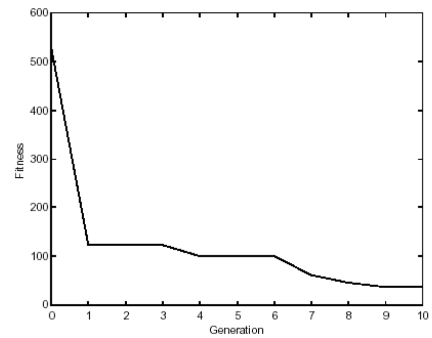

Fig.8 The best individual evolution versus the generation

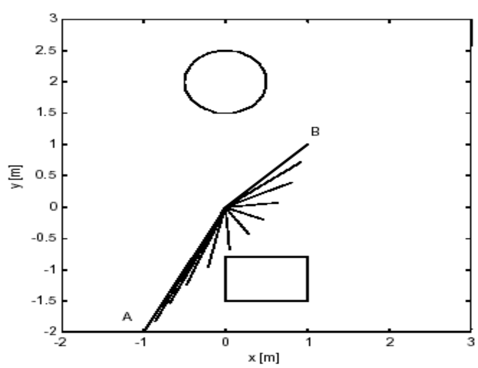

Fig.9 Successive robot configurations

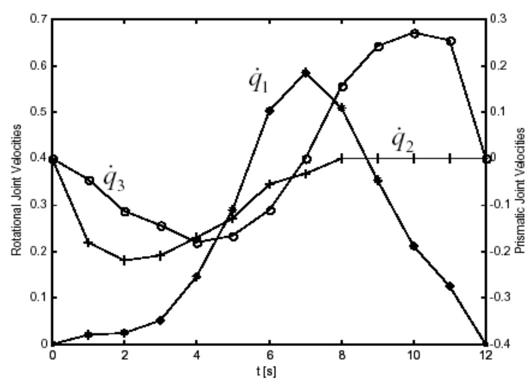

Fig.10 joint velocities