JJMIE

Volume 2, Number 3,Sep. 2008
ISSN 1995-6665
Pages 131 - 136

Jordan Journal of Mechanical and Industrial Engineering

Motion Planning for a Robot Arm by Using Genetic Algorithm

Bahaa Ibraheem Kazem *, Ali Ibrahim Mahdi, Ali Talib Oudah

Mechatronics Engineering Dept., Al-Khawarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

Abstract

This paper proposes genetic algorithm (GA) to optimize the point-to-point trajectory planning for a 3-link (redundant) robot arm. The objective function for the proposed GA is to minimizing traveling time and space, while not exceeding a maximum pre-defined torque, without collision with any obstacle in the robot workspace. Quadrinomial and quintic polynomials are used to describe the segments that connect initial, intermediate, and final point at joint-space. Direct kinematics has been used for avoiding the singular configurations of the robot arm.

Keywords: robot motion planning; genetic algorithm; obstacle avoidance

1. Introduction

In the last decade, genetic algorithms (GAs) have been applied in large number of fields such as in control, parameter, and system identification, robotics, planning and scheduling, image processing, pattern recognition, speech recognition. This paper addresses the area of robotics, namely the point-to-point trajectory planning for mechanical manipulators. At start, some of conventional methods have been used for trajectory planning.

For generating smooth trajectory planning for specified path, Z. Zoller and P. Zentay [1] focused on the problem of the trajectory planning and dealt with constant kinetic energy motion planning. The authors used Euclidean space to provide the equation of dynamic of robot motion with constant kinetic energy. This method produced trajectory characteristics smoother and better than which did obtained from time optimal method. Nevertheless, it can be implemented only for pre-specified path.

Zhe Tang et al. [2] proposed a third–order spline interpolation based trajectory-planning method to plan a smooth biped swing leg trajectory by reducing the instant velocity change, which occurs at the time of collision of the biped swing leg with the ground. The authors demonstrate that the impact effects can be avoided at the time of the swing foot's heel touching with the ground.

About on line trajectory planning, Chwa et al. [3] proposed "Missile Guidance Algorithm" to generate on–line trajectory planning of robot arms of the interception of a fast maneuvering object. The authors employed the guidance law throughout the tracking phase, and dynamic constraints such as torque and velocity constraints and satisfied the matching condition of the position and velocity at the time of the interception altogether. This was carried out by introducing body axis (as well as joint and inertia axis) as trajectory planning coordinates and separating the trajectory-planning problem into direction planning and speed planning of robot arm.

Various methods for trajectory planning schemes based on GAs have been proposed. P. Garg and M. Kumar [4] use GA techniques for robot arm to identify the optimal trajectory based on minimum joint torque requirements. The authors use polynomial of 4^{th} degree in time for trajectory representation to joint space variables.

Pires and Machado [5] propose a path planning method based on a GA while adopting the direct kinematics and the inverse dynamics. The optimal trajectory is the one that minimize the path length, the ripple in the time evolution and the energy requirements, without any collision with the obstacle in the workspace.

Pires et al. [6] optimized robot structure while optimizing the required manipulating trajectories using GA. The objective is to minimize the space/time ripple in the trajectory without colliding with any obstacles in the workspace, while optimizing the mechanical structure.

S. G. Yue et al. [7] focused on the problem of point-to-point trajectory planning of flexible redundant robot manipulator (FRM) in joint space. The proposed trajectory to minimize vibration of FRMs is based on GA. The authors use quadrinomial and quintic polynomials to describe the segment, which connects the initial, intermediate, and final points in joint space.

Pires et al [8] use genetic algorithm to optimize a planar robot manipulator trajectory. The GA objective is to minimize the trajectory space/time ripple without

* Corresponding author. e-mail: drbahaa@gmail.com

exceeding the maximum pre-defined torque. The authors use direct kinematics to avoid the singularities.

In this line of through, this paper, propose a point-to-point trajectory planning method based on GA while adopting the direct kinematics and the inverse dynamics. The optimum trajectory is the one that minimize both traveling time and traveling space, while not exceeding the maximum pre-defined torque, without collision with any obstacle in the workspace.

Bearing these ideas in mind, this paper is organized as following. Section 2 presents the robot motion planning strategy. Section 3 introduces GA motion planning scheme. Section 4 presents operators in genetic algorithm. Section 5 presents evolution criteria. Based this formulation, section 6 presents the case studies and simulation, the results are also discussed in this section. Finally, section 7 outlines the main conclusions.

2. Motion Planning Strategy

The supposed point-to-point trajectory is connected by several segments with continuous acceleration at the intermediate via point as shown in figure 1. The intermediate points can be given as particular points that should be passed through.

For a robot, the number of degrees of freedom of a manipulator is n and the number of end-effectors degree of freedom is m. If one wishes to be able to specify the position, velocity, and acceleration at the beginning and the end of a path segment, a quadrinomial and a quintic polynomial can be used. Let us assume that there is m_p intermediate via points between the initial and final points.

Figure 1: Intermediate points on the point-to-point trajectory

Between the initial points to m_p intermediate via points, a quadrinomial is used to describe these segments as [7]:

$$\theta_{i,i+1}(t) = a_{i0} + a_{i1}t_i + a_{i2}t_i^2 + a_{i3}t_i^3 + a_{i4}t_i^4, \ (i=0,\dots,mp-1) \quad (1)$$

Where ($a_{i0},\dots,a_{i4}$) are constants, and the constraint are given as:

$$\theta_i = a_{i0} \quad (2)$$

$$\theta_{i+1} = a_{i0} + a_{i1}T_i + a_{i2}T_i^3 + a_{i4}T_i^4 \quad (3)$$

$$\dot{\theta}_i = a_{i1} \quad (4)$$

$$\dot{\theta}_{i+1} = a_{i1} + 2a_{i2}T_i + 3a_{i3}T_i^2 + 4a_{i4}T_i^3 \quad (5)$$

$$\ddot{\theta}_i = 2a_{i2} \quad (6)$$

Where Ti is the execution time from point i to point i+1. The five unknowns can be solved as:

$$a_{i0} = \theta_i \quad (7)$$

$$a_{i1} = \dot{\theta}_i \quad (8)$$

$$a_{i2} = \ddot{\theta}_i / 2 \quad (9)$$

$$a_{i3} = (4\theta_{i+1} - \dot{\theta}_{i+1}T_i - 4\theta_i - 3\ddot{\theta}_i T_i^2) / T_i^3 \quad (10)$$

$$a_{i4} = (\dot{\theta}_{i+1}T_i - 3\theta_{i+1} + 3\theta_i + 2\dot{\theta}_i T_i + \ddot{\theta}_i T_i^2 / 2) / T_i^4 \quad (11)$$

The intermediate point *(i+1)*'s acceleration can be obtained as:

$$\ddot{\theta}_{i+1} = 2a_{12} + 6a_{i3}T_i + 12a_{i4}T_i^2 \quad (12)$$

The segment between the number m_p of intermediate points and the final point can be described by quintic polynomial as:

$$\theta_{i,i+1}(t) = b_{i0} + b_{i1}t_i + b_{i2}t_i^2 + b_{i3}t_i^3 + b_{i4}t_i^4 + b_{i5}t_i^5, \ (i=mp) \quad (13)$$

Where the constants are given as:

$$\theta_i = b_{i0} \quad (14)$$

$$\theta_{i+1} = b_{i0} + b_{i1}Ti + b_{i2}T_i^2 + b_{i3}T_i^3 + b_{i4}T_i^4 + b_{i5}T_i^5 \quad (15)$$

$$\dot{\theta}_i = b_{i1} \quad (16)$$

$$\dot{\theta}_{i+1} = b_{i1} + 2b_{i2}Ti + 3b_{i3}T_i^2 + 4b_{i4}T_i^3 + 5b_{i5}T_i^4 \quad (17)$$

$$\ddot{\theta}_i = 2b_{i2} \quad (18)$$

$$\ddot{\theta}_{i+1} = 2b_{i2} + 6b_{i3}Ti + 12b_{i4}T_i^2 + 20b_{i5}T_i^3 \quad (19)$$

In addition, these constraints specify a linear set of six equations with six unknowns whose solution is:

$$b_{i0} = \theta_i \quad (20)$$

$$b_{i1} = \dot{\theta}_i \quad (21)$$

$$b_{i2} = \ddot{\theta}_i / 2 \quad (22)$$

$$b_{i3} = (20\theta_{i+1} - 20\theta_i - (8\dot{\theta}_{i+1} + 12\dot{\theta}_i)T_i - (3\ddot{\theta}_i - \ddot{\theta}_{i+1})T_i^2) / 2T_i^3 \quad (23)$$

$$b_{i5} = (12\theta_{i+1} - 12\theta_i - (6\dot{\theta}_{i+1} + 6\dot{\theta}_i)T_i - (\ddot{\theta}_i - \ddot{\theta}_{i+1})T_i^2) / 2T_i^5 \quad (24)$$

As formulated above, the total parameters to be determined are the joint angles of each intermediate via point ($n \times m_p$ parameters), the joint angular velocities of each intermediate point ($n \times m_p$ parameters), the execution time for each segment (m_p+1 parameters), and the posture of the final configuration (n-m).

Therefore, for 3-link robot case, it used m_p= 1, n =3 and one degree of freedom of redundancy for the final point, there are nine parameters to be determined.

It should be point out that joint angular acceleration at each intermediate point could be obtained via equation (12). If all the intermediate points are connected by quintic polynomial, there will be eight parameters to be determined. This would be more time-consuming, which is why we choose both quadrinomial and quintic polynomial to generate the segments.

3. The GA Motion Planning Scheme

The GA planning scheme renders an optimized trajectory having minimum space, minimum time, while not exceeding a maximum pre-defined torque, without colliding with any obstacle in the workspace. The motion planning adopts direct kinematics to avoid singularity problems. The trajectory parameters are encoded directly, using real codification, as strings (chromosomes) to be used by GA.

For 3R, redundant robot there are nine parameters should be optimized as shown in the following chromosome:

$$[q_1, q_2, q_3, q_g, \dot{q}_1, \dot{q}_2, \dot{q}_3, t_1, t_2] \tag{25}$$

Where q_i and $\dot{q}_i$ are intermediate joint angle and velocity for *i*th joint respectively, q_g is the global angle of the final configuration of the end-effectors which equals the sum of joint angles of the manipulator [16], t_1 is execution time from initial to intermediate via point, and t_2 is execution time from intermediate to final point.

4. Operators in genetic algorithm

The initial population of strings is generated at random and the search is then carried out among this population. The evolution of the population elements is non-generational, meaning that the new replace the worst elements. The main different operators adopted in the GA are reproduction, crossover, and mutation.

In what concerns the reproduction operator, the successive generations of new strings are generated based on their fitness values. In this case, a 5-tournament is used to select the strings for reproduction.

With a given probability P_c, the crossover operator adopted the single point technique and, therefore, the crossover point is only allowed between genes or, in other words, the crossover operator cannot disrupt genes.

The mutation operator replaces one gene value x_t with another one generated randomly with a specified range by a given probability P_m. figure 2 shows the flow chart of the above steps of GA.

5. Evolution Criteria

Four indices are used to qualify the evolving trajectory robotic manipulators free workspace. All indices are translated into penalty functions to be minimized. Each index is computed individually and is integrated in the fitness function evaluation. The fitness function f_f adopted for evaluating the candidate trajectories is defined as:

$$f_f = \beta_1 f_{ot} + \beta_2 f_q + \beta_3 f_c + \beta_4 t_T \tag{26}$$

The optimization goal consists in finding a set of design parameters that minimize f_f according to the priorities given by the weighting factors β_i (i = 1,.., 4), where each different set of weighting factors must results in a different solution.

The f_{ot} index represents the amount of excessive driving, in relation to the maximum torque $\tau_{i\,max}$, that is demanded for the *i*th joint motor for the trajectory under consideration using the equation (28) which is called the cost function [9].

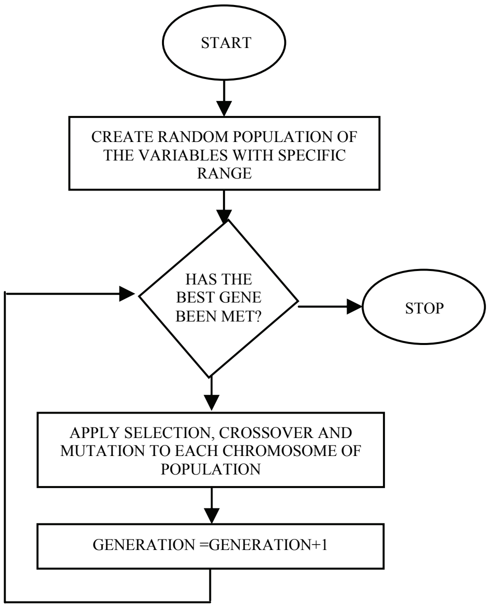

Figure 2: flow chart of GA

$$f_{ot} = \sum_{j=1}^{b} \sum_{i=1}^{a} f_i^j \tag{27}$$

$$f_i^j = \begin{cases} 0 & \left|\tau_i^j\right| < \tau_{i\max} \\ \left|\tau_i^j\right| - \tau_{i\max} & \text{otherwise} \end{cases} \tag{28}$$

Where a is number of robot links, and b is number of joint positions from the initial to final configuration.

The dynamic equations of the 3R manipulator can be easily obtained from the iterative Newton-Euler dynamics algorithm [8]. For simplicity, all mass exist as a point mass at the distal end of each link as shown in figure 3.

The index f_q represents the total joint traveling distance of the manipulator as criteria:

$$f_q = \sum_{i=1}^{a} \sum_{j=2}^{b} |q_{ij} - q_{ij-1}| \tag{29}$$

The index f_c represents total Cartesian trajectory length, as criteria:

$$f_c = \sum_{j=2}^{b} d(p_j, p_{j-1}) \tag{30}$$

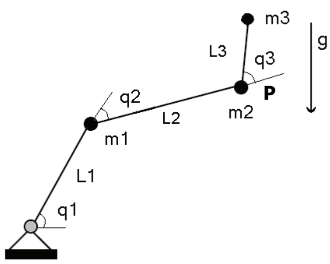

Figure 3: The 3R robot

Where p_j is the robot jth intermediate arm Cartesian position and d(.,.) is a function that gives the distance between the two arguments. The index t_T represents the total consumed time for robot motion, as criteria:

$$t_T = t_1 + t_2 \tag{31}$$

Where t_1 and t_2 are the execution time form start to intermediate configuration, and from intermediate to target configuration, respectively.

For obstacle existence workspace, obstacle avoidance objective function f_{ob} has been combined with free space fitness function to form over all fitness function f, as shown below:

$$f = f_f / f_{ob} \tag{32}$$

By f_{ob}, the robot manipulator has the ability to avoid the obstacle collision during its movement from point to point in side the workspace. f_{ob} can be depicted as [11]: all links of configurations, which formed, by the joint positions between the initial and final robot configurations do not intersect obstacle region. The fitness value is f_{ob}=1.

Therefore,, the objective function of collision avoidance f_{ob} can be written as equation (33).

$$f_{ob} = \begin{cases} 1 & \sum_{j=1}^{b}\sum_{i=1}^{a}(\text{link}_{ij} \cap \text{obstacle}) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

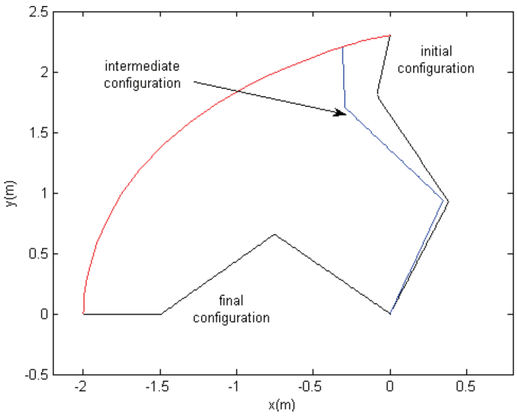

Figure 4: Cartesian path for the end effectors of the robot in free workspace

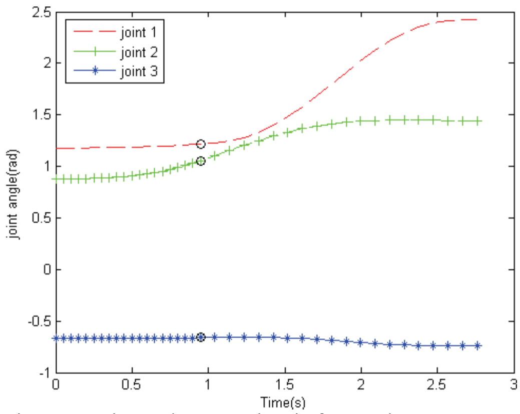

Figure 5: Joint angle versus time in free workspace

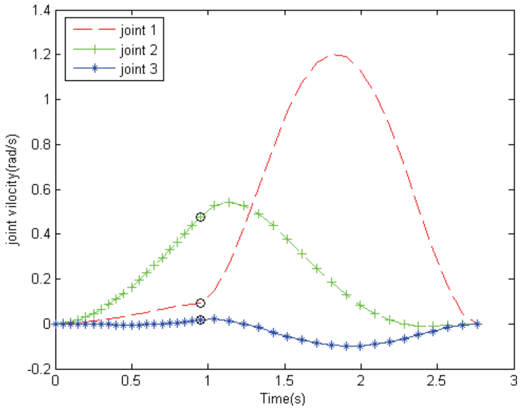

Figure 6: Joints velocity versus time in free workspace

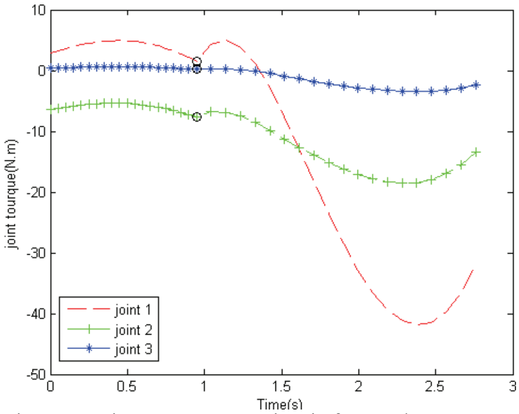

Figure 7: Joints torque versus time in free workspace

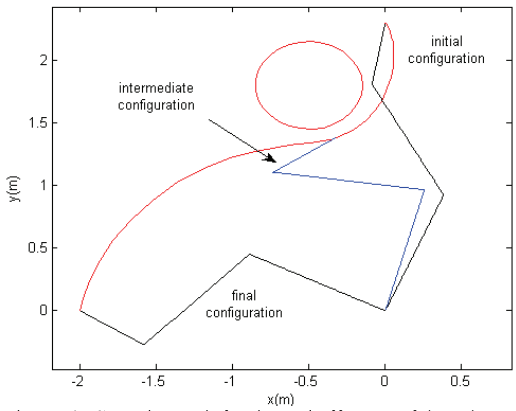

Figure 8: Cartesian path for the end effectors of the robot with obstacle existence

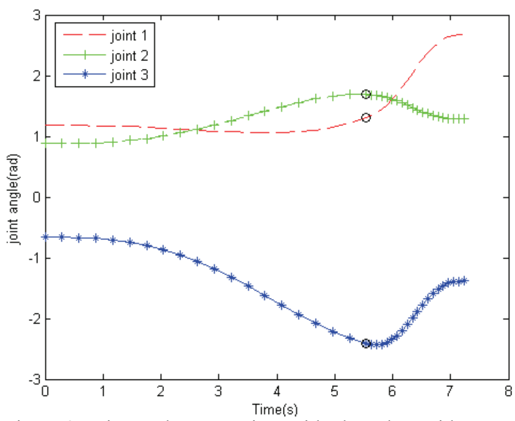

Figure 9: Joint angle versus time with obstacle avoidance

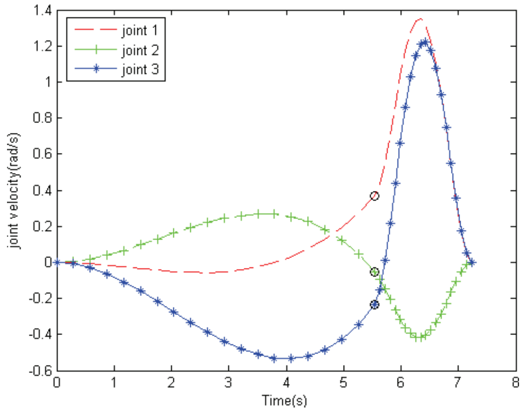

Figure 10: Joint velocity versus time with obstacle avoidance

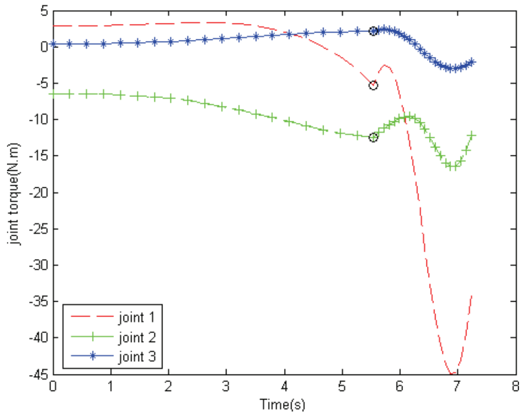

Figure 11: Joint torque versus time with obstacle avoidance

6. Simulation Results

This section presents the results of robot case study. This case consists on moving 3R robot arm form starting point (x=0 m, y=2.3 m, $q_g=80^o$) to final point (x=-2 m, y=0 m). The robot links have length of (l_1=1m, l_2=1m and l_3=0.5 m) and mass of(m_1=1kg, m_2=1kg and m_3=0.5 kg) the maximum allowed torques for joint 1, 2 and 3 of τ_{1max} = 45 Nm, τ_{2max}= 20 Nm and τ_{3max} =5 Nm, respectively. The joints velocity and acceleration of the initial and final configuration are assumed zeros. More over all robot joints are free rotate 2π .in the following case studies, the obstacle has circular shape with radius 0.35 m.

GA adopts a crossover probability P_c=0.8 per chromosome, a mutation probability P_m=0.05 per locus, a population of 200 elements for intermediate joints angle, joint velocity and traveling time of the arm, a string size ss=9 robot respectively, a 5-tournament selection scheme with elitism, and maximum generation mg=80. The weight factors set of fitness faction is $[\beta_1, \beta_2, \beta_3, \beta_4] = [1,2,2,1]$. At the initial generation of population, GA generates the chromosome elements with specified range, as following:

$$-\pi \le q_i \le +\pi \quad rad \quad (i=1,2,3) \tag{34}$$

$$-\pi \le q_g \le +\pi \quad rad \tag{35}$$

$$-\pi/4 \le \dot{q}_i \le +\pi/4 \ rad/sec \quad (i=1,2,3) \tag{36}$$

$$0.1 \le t_i \le 8 \ sec \quad (i=1,2) \tag{37}$$

From figure 4 to figure 7, show the optimization results free workspace. When an obstacle found in the workspace with coordinates(x=-0.5, y=1.8), the optimized results are shown in figures from 8 to 11.

Figure 4 shows the shorter Cartesian path. However, the straight line from the initial to final point is the shortest one, but is far from the best one according to the GA optimization result. Whereas figure 8 shows the ability of GA to decide, the parameters that generate the shorter Cartesian path with obstacle presence in the workspace with regard the other specified objective functions.

At each generation, GA chooses an adequate q_g. By final point coordinates and q_g, the joint angles of the final configuration can be evaluated by inverse kinematics of planar 3-link articulated robot [16]. As shown, the final tool orientation in figure 4 and 8 has been chosen according to the specified objective functions, therefore GA able to solve the kinematics redundancy in the absence and presence the obstacle in the workspace.

The black spots in figures 5 and 9 and figures 6 and 10 represent the optimized intermediate joint angle and joint velocity, respectively. The black spot in figures 7 and 9) represent the joint torque at the optimized joint angle and velocity. As shown in figures 7 and 9 the torque that calculated along the joint space trajectory in the case of free workspace is less than which results from case of obstacle existence workspace. However, in both cases, the joint torque does not exceed the maximum pre-defined torque. Since direct kinematics has proposed, tool-configuration matrix not used [12]. Therefore, singularity has been not concerned.

Table 1 shows the value of total traveling time, total joint traveling distance and Cartesian trajectory length by equations (31), (29) and (30) respectively, for both free and obstacle existence workspace.

Table 1: Optimization results

Result value	Free workspace	Obstacle existence workspace
Total traveling time (sec)	2.76	7.23
Total joint traveling distance (rad)	1.91	5.78
Total Cartesian trajectory length (m)	3.28	3.42

As noted from figure 12, the values of the total traveling time, total joint traveling distance and total Cartesian trajectory length of the obstacle existence are more than which are resulted from free work space. The amount of the increment makes the robot to be able to maneuver during its motion for avoiding the collision with the obstacle.

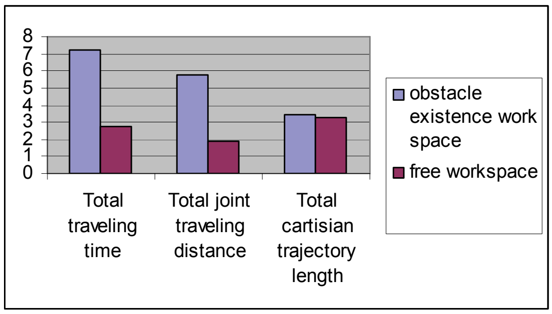

Figure 12: Results comparison between free and obstacle existence workspace

7. Conclusions

In the previous sections, the problem of the point-to-point trajectory planning was studied in detail. Trajectory planning method based on GA with specific objective functions was presented. Kinematics redundancy for the final configuration was considered as planning variable in the presented method. Case study of 3R planar robot showed that the method is effective, especially for avoiding the obstacle collision with the other objective functions. Since the proposed motion planning is based on the joint space, the total traveling time depends only on the joint traveling distance. The joint torque of the robot did not exceed its maximum pre-defined torque in both free and obstacle existence workspace case. Since GA uses the direct kinematics, the singularities do not constitute a problem. GA showed that it is able to achieve multi objective optimization efficiently. Finally, kinematics redundancy can be solved within GA according to the specified objective functions.

References

[1] Zoltan Zoller, & Peter Zentan " Constant Kinetic Energy Robot Trajectory Planning ", PERIDICA POLYTECHNICA SER. MECH. ENG., Vol. 43, No.2, 213 – 228, 1999.

[2] Zhe Tang, Changjiu Zhou, Zenqi Sun," Trajectory Planning For Smooth Transition Of A Biped Robot," IEEE International Conference on Robotic & Automation, Vol. 2, 2455-2460, Sept. 14 -19, 2003.

[3] Dong Kyoung Chwa, Jungo Kang, Kihong Im, Jin Young Choi, " Robot Arm Trajectory Planning Using Missile Guidance Algorithm,", SICE Annual Conference, vol. 2, pp. 2056-2061, August 4-6, 2003.

[4] Devendra P. Grag, & Manish Kumar, "Optimization Techniques Applied To Multiple Manipulators for Path Planning and Torque Minimization", Engineering Applications of Artificial Intelligence, Volume 15, Number 3, June 2002 , pp. 241-252(12).

[5] E.J. Solteiro Pires & J.A. Tenreiro Machado, " A Ga Perspective Of The Energy Requirement For Manipulators Maneuvering In A Workspace With Obstacles", Cec 2000-Congress On Evolutionary Computation , pp. 1110- 1116, 16-19 July 2000, Santiago, California, USA.

[6] E.J. Solteiro Pires, Tenreiro Machado JA & De Moura Oliveira CR, "An Evolutionary Approach to Robot Structure and Trajectory Optimization", 10th IEEE-International Conference on Advanced Robotics, ICAR'01, pp. 333-338, 22-25 August, 2001, Budapest, Bulgaria.

[7] S.G. Yue, D. Henrich, X. L. Xu, & S.K. Toss, "Point-to-Point Trajectory Planning of Flexible Redundant Robot Manipulators Using Genetic Algorithms", Journal of Robotica, vol 20, pp 269-280, 2002.

[8] E.J. Solteriro Pires, J.A. Tenreiro Machudo & P.B. Moura Oliveira, "Fractional Order Dynamic In A Genetic Algorithm", The 11th International Conference On Advanced Robotics, Colombia, Portugal, pp. 264 -269 June 30- July 3, 2003.

[9] D. E. Goldberg, "Genetic Algorithm in Search, Optimization, and Machine Learning," Addison-Wiley, 1989.

[10] Craig, J.J “Introduction to Robotics: Mechanics and Control”, Addison- Wesley, 1989.

[11] Dv Xin, Cher Hua –Hua, Gv Wei – Kang, "Neural Network And Genetic Algorithm Based Global Path Planning In A Static Environment", Journal Of Zhejiang University Science, Vol. 6A No. 6, 2005, 549- 554.

[12] Schilling J., "Fundamentals of Robotics Analysis and Control," Prentice Hall, 1990.

[13] Hartmut Pohlheim, "Geatbx Tutorial Genetic and Evolutionary Algorithm Toolbox for Use with Matlab," Version 3.5a, July 2004. Available at www.Geatbx.Com/

[14] Fu, K.S., Gonzalez, R.C., and Lee, C.S.G “Robotics: Control, Sensing, Vision and Intelligence”, McGraw Hill, 1987.

[15] Asada and Slotine, "Robot Analysis and Control," John Wiley &Sons, Inc., 1976.

[16] Lung-Wen T. "Robot Analysis, the Mechanism of Serial and Parallel Manipulators" John Wiley &Sons, Inc., 1999.