Asian Journal of Scientific Research

ISSN 1992-1454

science alert
http://www.scialert.net

ANSInet
an open access publisher
http://ansinet.com

Asian Journal of Scientific Research 6 (3): 488-497, 2013
ISSN 1992-1454 / DOI: 10.3923/ajsr.2013.488.497

Polynomial Joint Angle Based Motion Planning for Robotics Arm

Affiani Machmudah and Setyamartana Parman
Department of Mechanical Engineering, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Corresponding Author: Affiani Machmudah, Department of Mechanical Engineering, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

ABSTRACT

This study addresses a point-to-point of an arm robot motion planning in complex geometrical obstacle considering all kinematics and dynamics constraints. A continuous function of a seventh degree polynomial is utilized as a joint angle path. The path planning optimization objective is to minimize a joint angle traveling distance under avoiding collision constraint. After the best path has been discovered, the trajectories will be optimized with an objective is to minimize the total traveling time and the torque under the maximum velocity, the maximum acceleration, the maximum jerk and the maximum torque constraints. Three Degree of Freedom (3-DOF) planar robot will be utilized to simulate the proposed method. The computational strategy utilizing a Genetic Algorithm (GA) will be presented. There is no information regarding the region of the feasible seventh degree polynomial joint angle path so that the GA should search it first. At the first computation where the population is generated randomly, all individuals commonly collide with obstacles. It needs a requirement to escape from zero fitness. After the feasible individual has been discovered, the GA should evolve this individual to find the best one with the highest fitness value. Results show that the feasible joint angle path which is very smooth in the motion has succeeded to be found. The trajectories are also discovered successfully without exceeding the constraint values.

Key words: Arm robot motion planning, seventh degree polynomial joint angle path, genetic algorithm, jerk, torque

INTRODUCTION

In line with higher demand to achieve production efficiency, the motion planning is very important for industrial robot. The recent issue is finding the best strategy to move the robot from the initial configuration to the final configuration safely and optimally considering all constraints.

Some strategies to gain the optimality has been proposed. Boriga and Grabos (2009) presented a planning mode of trajectory motion for serial-link manipulators with the polynomials of degrees 9, 7 and time courses of displacements, velocities, accelerations and jerk for the rectilinear path of end-effector motion for a three Degrees of Freedom (DOF) manipulator were presented. An inverse kinematics problem was solved and on this basis the runs of displacements, velocities, accelerations and angular jerks of each kinematic chain link were established.

Gasparetto and Zanotto (2007) presented the trajectory planning of robot manipulators with the objective function containing a term proportional to the integral of the squared jerk along the trajectory is considered. Fifth-order B-splines are then used to compose the overall trajectory.

Pires *et al.* (2007) presented the point-to-point manipulator trajectories using a multi-objective genetic algorithm for two DOF and three DOF manipulators when one circular obstacle was present. They discover the collision-free joint angle trajectories using the discrete analysis with nine configuration points.

Saravanan *et al.* (2009) utilized a cubic B-spline for point-to-point robot motion planning under the velocity, the acceleration and the jerk constraints. It has twelve variables for each links that should be found by the trajectory planner.

This study considers improving the previous works in numbers of ways. It utilizes complex geometrical obstacles. It considers the kinematics constraints as well as the dynamics constraint. The constraints are the maximum velocity, the maximum acceleration, the maximum jerk and the maximum torque. It also utilizes continuous function as joint angle path to achieve the continuity of the link configuration. Generating the continuous joint angle path in the complex obstacle geometry as well as finding the feasible trajectories which satisfy the kinematics and dynamics constraints is very complex optimization problem. The GA will be utilized to solve this challenging problem.

PERFORMANCE OF ROBOTICS ARM IN THE PRESENCE OF OBSTACLES

It is very important for the arm robot to have the connectivity for all configurations during the manipulation. Some tools to analyze the robotics arm performance are the workspace, the aspect and the connectivity.

Without the existence of the obstacles, the workspace of 3-DOF planar arm robot will be equal to $l_1+l_2+l_3$, with l is the length of the link. The workspace of the robot can be very complicated when the obstacle is present (Haddad *et al.*, 2002).

Refer to Fig. 1a, the workspace analysis give an information that points A and B lay in the free workspace so that there are feasible trajectories between these points; however, the path in Fig. 1a. is not the part of these feasible path since points C and D as well as points E and F do not connect each other. Aspect analysis needs to be done in point C and point D. The aspect concerns with the posture of the link robot between trajectories that should be reachable. Figure 1b shows one of the feasible path from point A to point B where all trajectories are connect each other.

The configurations can be considered consist of n discrete trajectories. Among these discrete trajectories, it should be guaranteed that there are n-connectivity. The detail of the connectivity tool can be found by Haddad *et al.* (2002). Figure 1b has n-connectivity so that the motion is reachable. In the proximity of the obstacle, the analysis of n-connectivity is also very complicated (Haddad *et al.*, 2002). By keeping the joint angle traveling distance into the minimal one incorporating with the avoiding collision requirement, the GA will search the path that lays in the free configuration and has n-connectivity while the optimality has also should be achieved.

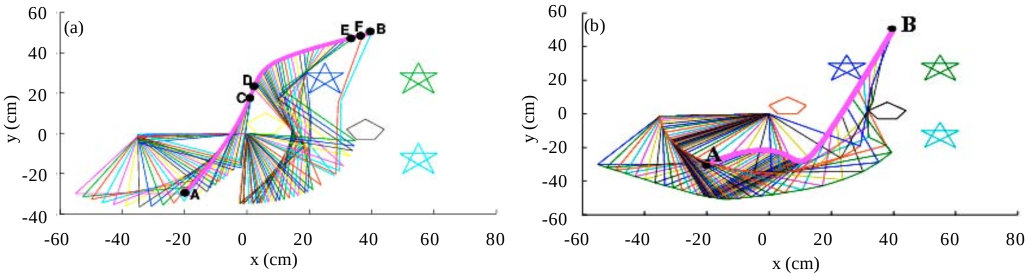

Fig. 1(a-b): Robot motion (a) Unfeasible and (b) Feasible

KINODYNAMIC CONSTRAINTS

This study considers the arm robot motion planning under kinodynamic constraints that contain the kinematics constraints as well as the dynamic constraint. It considers that the joint angles can be operated from $0 \le \theta \le 2\pi$.

Kinematics constraints: The path planning does not give any information about the motion as function of time. The trajectory planning involves discovering the total travelling time in such a way that it will give the optimal motion. This paper will govern linear time-scale, thus the path planning is done from parameter 0 to 1. The kinematics constraints consist of the maximum velocity, the maximum acceleration and the maximum jerk.

The arm robot has the limitation in the maximum velocity as follows:

$$\dot{\theta}_{(i)} = \frac{d\theta_{(i)}}{dr}\frac{1}{T} \le \omega_{max(i)} \tag{1}$$

where, $\dot{\theta}_{(i)}$, $\omega_{max(i)}$ are the joint angle velocity of ith link and its maximum limit value.

The maximum acceleration and the maximum jerk are described as follow, respectively:

$$\ddot{\theta}_{(i)} = \frac{d^2\theta_{(i)}}{dr^2}\frac{1}{T^2} \le \alpha_{max(i)} \tag{2}$$

$$\dddot{\theta}_{(i)} = \frac{d^3\theta_{(i)}}{dr^3}\frac{1}{T^3} \le J_{max(i)} \tag{3}$$

where $\ddot{\theta}_{(i)}$, $\alpha_{max(i)}$, $\dddot{\theta}_{(i)}$ and $J_{max(i)}$ are the joint angle acceleration, the maximum joint angle acceleration, the jerk and the maximum jerk, respectively.

Dynamic constraint: Dynamics model of link robot can be derived using the Lagrange-Euler method (Angeles, 2002).

The dynamic constraint can be expressed in the following:

$$\tau_i \le \tau_{imax} \tag{4}$$

where, τ_i and τ_{imax} the torque and the maximum torque of ith link, respectively.

AVOIDING COLLISION CONSTRAINT

Since the obstacles are present, it is also involve avoiding collision constraint. The collision detection is very important activity to achieve the collision-free path. To satisfy the avoiding collision constraint, the collision detection rules can be constructed as guidance for the motion planner to discover the feasible path.

The rules can be expressed in the following:

$$\text{if } \exists \text{ link position (i)} \cap \text{obstacle areas, then the collision happen otherwise is free from collision} \tag{5}$$

It needs to check whether any link positions are inside the obstacles areas. In this execution, after the joint angles are generated by the GA, the forward kinematics equation is utilized directly instead of the inverse kinematics.

SEVENTH DEGREE POLYNOMIAL JOINT ANGLE PATH

In the first step of arm robot motion planning, the best path should be discovered utilizing 7th degree polynomial function as follows:

$$\theta_k = a_{7k}r^7+a_{6k}r^6+a_{5k}r^5+a_{4k}r^4+a_{3k}r^3+a_{2k}r^2+a_{1k}r+a_{0i} \quad (6)$$

where, r is linear time scaled, t/T, with t is time and T is total transfer time. a_{nk} is the nth polynomial coefficient, with k is the number of the link.

This study assumes that an initial speed, a final speed, an initial acceleration and a final acceleration are zero. The starting point and the final point are known.

By taking all known variable and all constraints analysis into Eq. 6, the genes in the path planning are in the following:

a_{61}	a_{71}	a_{62}	a_{72}	a_{63}	a_{73}

Each link will have two unknown variables where the GA should code it to find the best individual.

FEASIBLE JOINT ANGLE PATH AND TRAJECTORIES

The continuous function of the joint angle trajectories starts from the initial joint angle to the final joint angle. The trajectories continuously move from the initial to final joint angles. Figure 2b illustrates the continuous joint angle trajectories of path configuration of Fig. 2a. The problem becomes how to find the intersection area of the sixth degree polynomial joint angle trajectories with the free configuration.

From Eq. 1-4, it can be seen that after the optimal path has been discovered, the trajectories generation is done by searching the total transfer time, T, so that the kinodynamic constraints are satisfied.

GENETIC ALGORITHM

The GA is a search technique to find exact solutions of optimization and search problems based on genetic theory and Darwin evolution theory. Figure 3 gives an illustration of the GA.

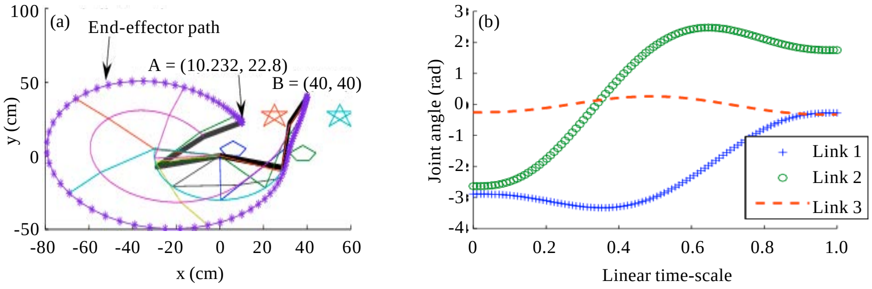

Fig. 2(a-b): Path configuration (a) Arm robot path and (b) Its continuous joint angle path

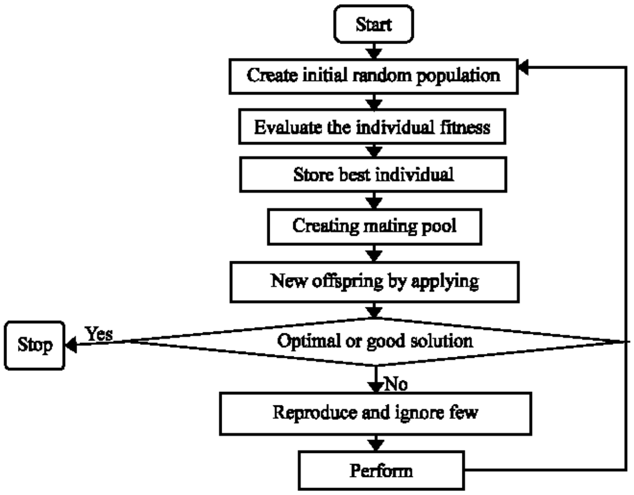

Fig. 3: Genetic algorithm scheme

Fitness function: For path planning, the objective optimization is to minimize the joint angle traveling distance. There is one strict constraint in this case, the avoiding collision. To guarantee the collision-free, the collision detection should be executed in the computation. Due to there is no information of the location of the joint angle feasible range, the death penalty is chosen in the GA. The death penalty will turn the GA into purely randomized searching method. It will kill every individual that collides with the obstacles by giving the zero fitness. The GA needs to escape from zero fitness condition to discover the range of feasible seventh degree polynomial joint angle path.

After the optimal joint angle is discovered, the GA needs to search the total traveling time, T, so that the trajectories lays in the allowable zone considering all constraints.

The death penalty rules can be expressed as follows:

- If any constraint does not satisfy by the joint angle path, then it is failure
- Otherwise, the path is success (7)

$$\theta_k = a_{7k}r^6+a_{6k}r^5 ... +a_{1k}r+a_{0k} \tag{7}$$

Considering the rules, the membership functions have been composed to solve the first step of the motion planning as follows:

$$F_i = \begin{cases} 0\exists\left(x_{\text{link}(i)}, y_{\text{link}(i)}\right) \cap \text{obstacle area} \\ \dfrac{1}{f_1+f_2+f_3} \text{ Otherwise} \end{cases} \tag{8}$$

$$f_i = \int_0^1 \sqrt{1+\left(\frac{d\theta(r)}{dr}\right)^2}\, dr \tag{9}$$

where, f_i is the joint angle traveling distance of ith link.

For the joint angle as function of time-scale, r, the joint angle traveling distance is the integration formula in Eq. 9. This study utilizes the Simpson's rule to compute the integration.

The second step of the motion planning is to find the best total travelling time, T, that is optimal and satisfies all constraints. In this trajectories planning, the gene is the total traveling time, T.

Such as in the first step, the membership function is composed for the trajectory planning as follows:

$$F_2 = 0\exists\left\{\begin{array}{c}\left\{\dot{\theta}_{(i)} = \frac{d\theta_{(i)}}{dr}\ \frac{1}{T} \geq \omega_{max}\ \text{or}\ \ddot{\theta}_{(i)} = \frac{d^2\theta_{(i)}}{dr^2}\ \frac{1}{T^2} \geq \alpha_{max}\ \text{or}\ \dddot{\theta}_{(i)} = \frac{d^3\theta_{(i)}}{dr^3}\ \frac{1}{T^3} \geq J_{max}\ \text{or}\ \tau_i \geq \tau_{imax}\right\} \\ F_2 = \text{Otherwise}\end{array}\right. \tag{10}$$

where, F_2 is the fitness function of the trajectories generation as follows:

$$F_2 = \frac{1}{\left\{aT + (1-a)\sum\sqrt{\tau_1^{\ 2} + \tau_2^{\ 2} + \tau_3^{\ 2}}\right\}} \tag{11}$$

where, a and τ are the weighting factor and the torque, respectively. Equation 11 will have the dimension of time. In this case, the joint angle path, θ, has been known from the first step result.

RESULTS AND DISCUSSION

A simulation in MATLAB has been done, by coding in m file. The simulations use 20 individuals in the population, 100 generations and 1000 generations for the path planning and the trajectories planning, respectively, 0.3 mutation rate, 0.5 selection rate. Searching area for path planning is $-100 \leq a_{6k} \leq 100$ while for trajectory planning is $0 \leq T \leq 100$. The genes are represented by real code value. Since in this case, it needs to turn the GA into purely randomized searching method, the high mutation rate, namely hyper-mutation, is used (Tsang and Lajbcyger, 2002; Srinivas and Patnaik, 1994).

There are three simulation cases carried out in this study. Figure 5-7 show the path planning results as well as the trajectories results for case I, case II and case III, respectively. For all cases, the point A and the point B are the initial point and the final point, respectively. Link constrains are presented in Table 1.

The first computation is done by generating the individual in the population randomly. All individuals commonly collide with the obstacles. This paper continues the searching process and investigates whether the GA can escape from this collision condition and find the feasible individual or not.

Table 1: Link constraints

Constraints	1st link	2nd link	3rd link
$\dot{\theta}$ (rad sec^{-1})	3	3	3
$\ddot{\theta}$ (rad sec^{-2})	8	8	8
$\dddot{\theta}$ (rad sec^{-3})	10	15	20
Mass (kg)	3	3	2
length (cm)	30	30	20
Torque (N m)	30	25	25

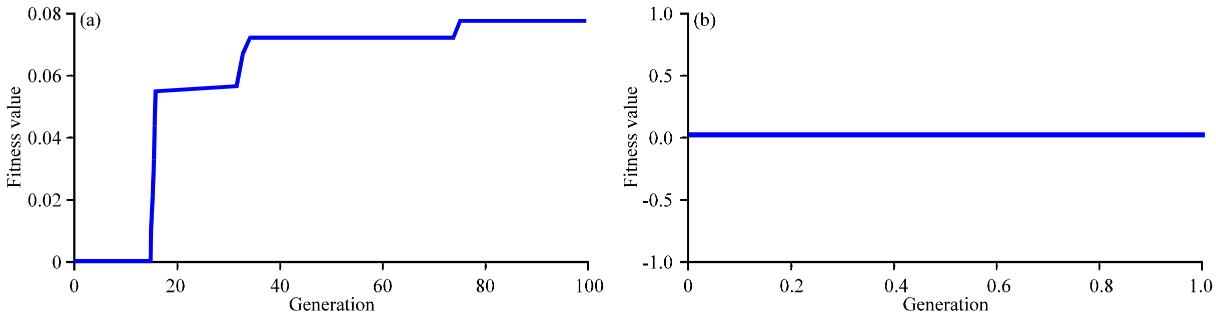

Fig. 4(a-b): Escaping process (a) Success and (b) Failure

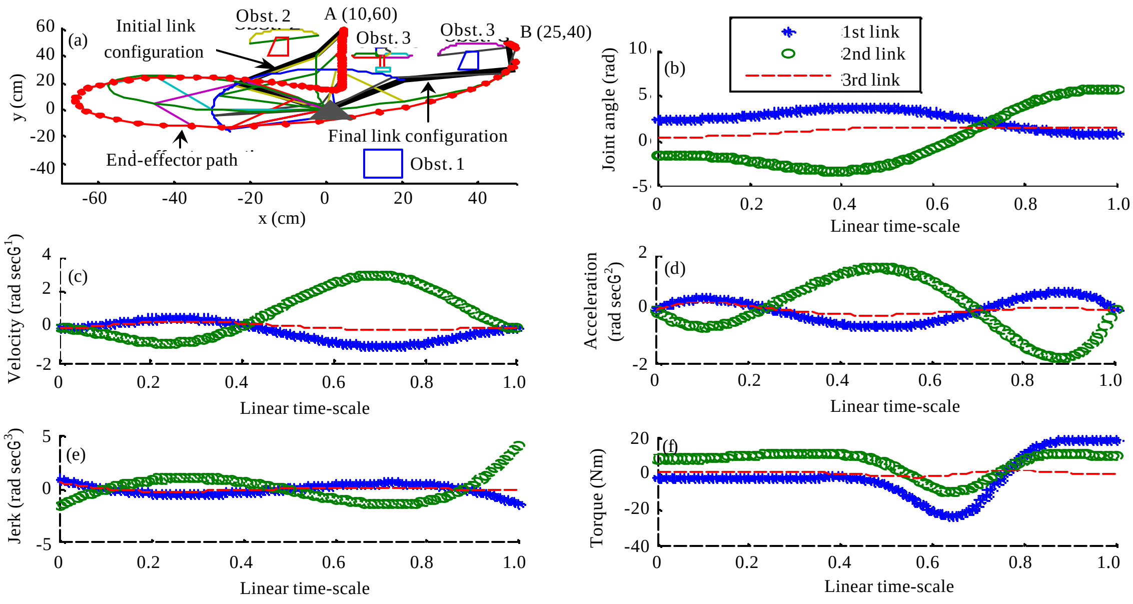

Fig. 5(a-f): Case I robot (a) Motion, (b) Joint angle, (c) Velocity, (d) Acceleration, (e) Jerk and (f) Torque

Figure 4a illustrates the success of the escaping process while Fig. 4b shows the failure of the escaping process during 100 generations. This paper just utilizes 100 generation since it want to investigate the GA performance in very short computational time.

At the beginning of the computation, the GA searches the feasible individual utilizing the death penalty. This is due to no information of the feasible individual yet. Thus, at the first iteration, where the individuals are generated randomly, all individuals in the population collide with the obstacles. The GA continues the searching process until the feasible zone is met.

For case 1, four running have been done to investigate the escaping and evolving process till 100 generations. The investigation is done in both path planning and trajectory planning. The results of this activity are presented in Table 2.

During these four running times, the path planning can be failure in escaping process. For the path planning, it utilizes just 100 generations since it needs to investigate the GA performance in short computational time. The computational time of the path planning is longer than that of the

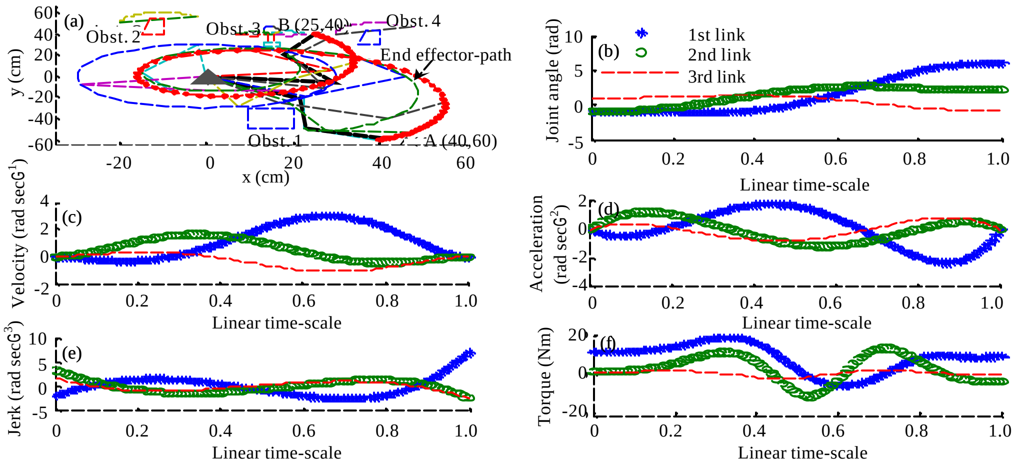

Fig. 6(a-f): Case II robot (a) Motion, (b) Joint angle, (c) Velocity, (d) Acceleration, (e) Jerk and (f) Torque

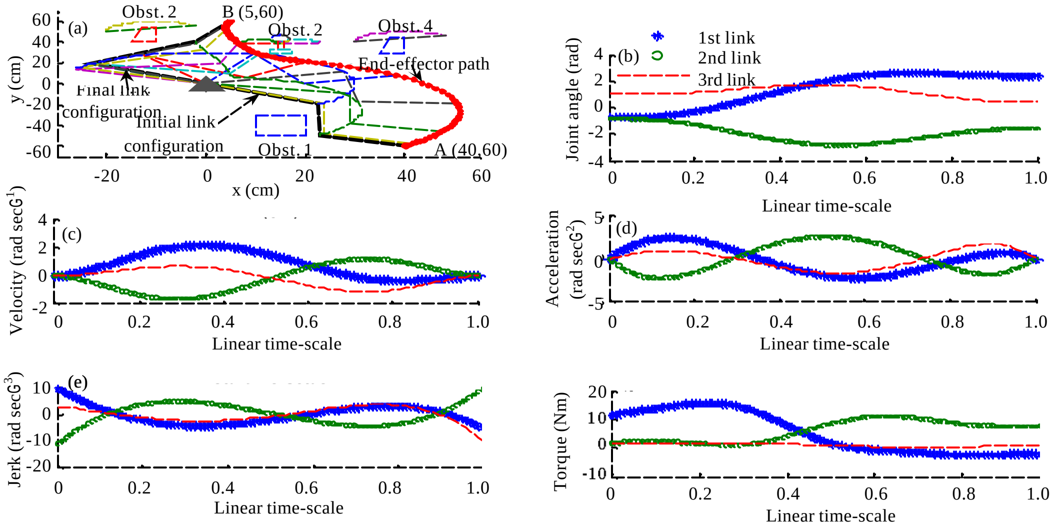

Fig. 7(a-f): Case III robot (a) Motion, (b) Joint angle, (c) Velocity, (d) Acceleration, (e) Jerk and (f) Torque

trajectory planning due to it involves the collision detection procedure. For trajectory planning, run 3 is utilized since it is the best result. The feasible transfer time is always discovered successfully since at the beginning. This is due to the path planning is more difficult optimization problem compare to the trajectory planning. It involves the requirement of avoiding collision and as n-connectivity that are very hard to be computed in the proximity of the obstacle.

From Table 2, it can be seen that the time of successful escaping process met in the path planning is unpredictable. The most important thing is the GA succeeded in finding this feasible joint angle path so that the evolution to get the optimal particles can be done easily now. After the non-zero fitness has been met, the evolution is begun to find the best individual in the population. These are very challenging result since it is generally known that the analysis of the robotic arm

Table 2: Escaping and evolving of case I

Case 1	Run 1 (gen, fitness)	Run 2 (gen, fitness)	Run 3 (gen, fitness)	Run 4 (gen, fitness)
Path planning gen	Fail	Escaping (60,0.036924) a_{71} = 67.813 a_{61} = -57.814 a_{72} = -61.702 a_{62} = 48.997 a_{73} = -48.887 a_{63} = 7.2503 Evolving (100, 0.03926) a_{71} = 67.584 a_{61} = -57.814 a_{72} = -61.124 a_{62} = 76.594 a_{73} = -14.423 a_{63} = -68.237	Escaping (21, 0.04113) a_{71} = 57.01 a_{61} = -74.871 a_{72} = -40.25 a_{62} = 77.882 a_{73} = -3.0836 a_{63} = -30.919 Evolving (100,0.08701) a_{71} = 22.236 a_{61} = 32.412 a_{72} = -27.59 a_{62} = 40.202 a_{73} = -32.821 a_{63} = 41.946	Fail
Trajectory planning fitness	Escaping (1,0.10361) T = 9.5482 Evolving (1000,0.1036) T = 4.682	Escaping (1, 0.13) T = 7.6128 Evolving (1000, 0.21135) T = 4.6825	Escaping (1, 0.10185) T = 9.7126 Evolving (1000,0.2115) T = 4.6789	Escaping (1,0.20462) T = 4.8371 Evolving (1000,0.2106) T = 4.6992

Gen: Generation, Fitness: Fitness value

performance is very complicated in the proximity of the obstacle (Haddad *et al.*, 2002). The calculation of free workspace, free aspect and n-connectivity is extremely difficult to be solved by the analytical technique. The existing methods to calculate them have many limitations. This study considers utilizing the natural computation technique by using its searching ability. The most difficult thing is finding the feasible joint angle path starting from randomly generated population where at this stage all individuals are unfeasible due to the constraint. It needs the requirement to escape from zero-fitness condition. After the non-zero fitness has been met, the computation is continued to evolve this non-zero fitness individual. Although it is very hard computation problem, the GA has succeeded to solve it easily.

CONCLUSIONS

The arm robot path planning utilizing 7th degree polynomial function in the complex geometrical environments has been presented. The trajectory planning has also succeeded to be optimized where all constraints are satisfied. The GA has succeeded in solving this complex optimization problem where it can escape from zero fitness and then evolve the non-zero fitness particle into the best one ATM.

REFERENCES

Angeles, J., 2002. Fundamentals of Robotic: Mechanical Systems: Theory, Methods and Algorithms. 3rd Edn., Springer, New York, USA.

Boriga, M. and A. Grabos, 2009. Planning of manipulator motion trajectory with higher-degree polynomials use. Mech. Mach. Theor., 44: 1400-1419.

Gasparetto, A. and V. Zanotto, 2007. A new method for smooth trajectory planning of robot manipulators. Mech. Mach. Theor., 42: 455-471.

Haddad, M., T. Chettibi, W. Khalil and H. Lehtihet, 2002. Trajectory Generation. In: Robot Manipulators: Modelling, Performance, Analysis and Control, Dombre, E. and W. Khalil (Eds.). 3rd Edn. ISTE Ltd., USA.

Pires, E.J.S., P.B.M. Oliveira and J.A.T. Machado, 2007. Manipulator trajectory planning using a MOEA. Applied Soft Comput., 7: 659-677.

Saravanan, R., S. Ramabalan and C. Balmurugen, 2009. Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles. Eng. Appli. Artif. Intell., 22: 329-342.

Srinivas, M. and L.M. Patnaik, 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybernet., 24: 656-667.

Tsang, R. and P. Lajbcyger, 2002. Optimizing Technique Trading Strategies with Split Search Genetic Algorithms. In: Evolutionary Computation in Economics and Finance, 1st Edn., Chen, S.H. (Ed.). Springer, New York, USA., ISBN-13: 9783790814767, Pages: 460.