International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

The Study of Robot Hand Path Planning using GA with Multi Space Obstacle

Biniyam Anteneh Ayalew[1], Bingqiang Huang[2], Zhiyuan He[3]

School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Abstract: ***In this paper, a new method is proposed for solving the obstacles avoidance problem of 3-link redundant robot arm in 2D space by optimizing the point-to-point path planning using GA. The objective function in GA is set to minimize traveling time and path with their weights allocated according to their worth in actual situation. The constraint of GA is not to exceed the maximum torque limit while avoiding collisions with several obstacles in robot workspace. Forward and inverse kinematics are used to describe the path segments that connect initial, intermediate and final points.***

Keywords: path-planning, genetic algorithm

1. Introduction

Many kinds of recent robots show the possibilities to change the way we live. Robots are being developed for personal assistance, general-purpose manufacturing, health-care and military fields. Unlike traditional roboticsapplications in which a robot operates in a tightly controlledenvironment (e.g., on an assembly line where the environmentis static and known a priori), these applications require therobot to adapt to its environment at runtime, which meansthat it must perform path planning [1].

Path planning is a fundamental problem in robotics whereby one seeks to compute a dynamically-feasible trajectory to achieve a goal; for example, a robot may wish to move its arm to reach a desired position without colliding with itself or any other objects. Generally, the problem of robot path planning can be stated as follows: a) initial position of the robot, b) desired goal position of the robot, c) geometric description of the robot, d) geometric description of the workspace, e) to find a path that moves the robot gradually from start to goal while never touching any obstacle [2].

The above definition for from a) to d) can be set easily, so many interests are focused on resolving e). In the world, sometimes, there is need to find the shortest path under constraints such as obstacle avoidance, low cost, external force limitation and so on, which is belonged and extension to e). The optimization algorithms are caused to achieve this purpose and constraints. There are many optimization algorithms for path planning problem and their performance and runtime are different according to the given situation [3, 4].

This paper deals the 2D path planning problem of 3-link robot arm and suggests genetic algorithm (GA) as optimization algorithm which minimizes the path traveling time and path under the constraints of obstacles avoidance and torque limit. Due to the difficulty of achieving the purpose and constraints simultaneously, the weights or coefficients are introduced according to worth of purpose and constraints. Forward and inverse kinematics are used to describe the path segments that connect initial, intermediate and final points with fourth- and fifth- order polynomials.

2. Literature Review

Over the last fifteen years, many studies for path planning have been proposed [5]. For avoiding collision with obstacles, Motahari et al. [6] suggested a new obstacle avoidance method for discretely actuated hyper-redundant manipulators. In each step of the solution, the closest collision to the base isremoved and then the configuration of the next part of the manipulator is modified without consideringthe obstacles. The authors used this process repeatedly until no collision is found, while applying Suthakorn method to solve the inverse kinematics problem.Dong Han et al. [7] proposed Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection. They calculated nearest distances between the links of amanipulator and the convex hull of an arbitrarily-shaped dynamic obstacle obtained from Kinect-V2 camera inreal-time by Gilbert–Johnson–Keerthi algorithm, and defined the minimum one as the closest distance between the manipulator and the obstacle. When the closest distance is less than a safe value, whether the dynamic obstacle is located in the global path of the manipulator is determined by improved discrete collision detection, which can adjust detection step-size adaptively for accuracy and efficiency. If the obstacle will collide with the manipulator, set a local goal and re-plan a local path for the manipulator.

For smooth path planning generation, Papadopoulos et al. [8] suggested Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems. Polynomial is very fast, easy to use and computationally inexpensive so that it can be widely used to generate the smooth path.

Kucuk et al. [9] described forward and inverse kinematics for general robots, Erkan et al. [10] described for Denso robot, and Jiang et al. [11] did for lily picking mechanical arm.

Some studies of control for obstacle avoidance have been suggested. Nizar et al. [12] proposed the robust adaptive fuzzy controller of a manipulator robot under the assumption that we don't have the precise knowledge of the mathematical model what is generally the case.Sharma et al.

[13] considered the autonomous navigation problem of a nonholonomic mobile platform and an n-link nonholonomic manipulator fixed to the platform, and proposed Lyapunov-based nonlinear controllers to solve this problem. Jain et al. [14] proposed hierarchical partially observable Markov decision process (hierarchical POMDP) planner that develops cost-optimized motion plans for hybrid dynamics models and decompose the POMDP planning problem into smaller sub-parts that can be solved with significantly lower computational costs.

In recent years, multi-robot motion planningrelated workshave been suggested.Atias et al. [15] studied the effectiveness of metrics for Multi-Robot Motion-Planning (MRMP) when using RRT-style sampling-based planners, where these metrics play the crucial role of determining the nearest neighbors of configurations and in that they regulate the connectivity of the underlying roadmaps produced by the planners and other properties like the quality of solution paths.Duong Le et al. [16] presented an effective multi-robot motion planner with dynamics guided by multi-agent search. He also suggested cooperative, dynamics-based, and abstraction-guided multi-robot motion planning in [17]. Dobson et al. [18] studied scalable, sampling-based planner for coupled multi-robot problems that provides desirable path quality guarantees. He mentioned that the proposed dRRT∗is an informed, asymptotically-optimal extension of a prior method dRRT, which introduced the idea of building roadmaps for eachrobot and implicitly searching the tensor product of these structures in the composite space. Dutra et al. [19] suggested the program for determination of the movement trajectory of a locomotor robot through the Voronoi diagram.

Many algorithms for path planning have been developed for last decades.Moriarty et al. [20] focused on evolving obstacle avoidance behavior in a robot arm, and presented an alternative approach that evolves neural network controllers through genetic algorithms, where no input/output examples are necessary since neuro-evolution learns from a single performance measurement over the entire task of grasping an object.Duguleana et al. [21] proposed new approach for solving the problem of obstacle avoidance during manipulation tasks performed by redundant manipulators by using a double neural network that uses Q-learning reinforcement technique which has been applied in robotics for attaining obstacle free navigation or computing path planning problems. Q-learning is also used together with neural networks in order to plan and execute arm movements at each time instant. Pajaziti et al. [22] suggested robotic arm control with neural networks using genetic algorithm optimization approach for the given robotic arm with 5 DOF.

Zahra et al. [23] proposed ant colony algorithm to find the optimal path from an initial to a final position in the presence of five obstacles. Hassani et al. [24] suggested robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm which handles two different objectives which are the path safety and the path length.Cosicet al. [25] suggested an approach to intelligent robot motion planning and tracking in known and static environments, and divided this complex problem into several simpler problems: generation of a collision-free path from starting to destination point, which is solved using a particle swarm optimization (PSO) algorithm; interpolation of the obtained collision-free path, which is solved using a radial basis function neural network (RBFNN), and trajectory generationbased on the interpolated path; a trajectory tracking problem, which is solved using a proportional-integral (PI)controller. Due to uncertainties, obstacle avoidance is still not ensured, so the authors introduced anadditional fuzzy controller, which corrects the control action of thePI controller.

Masehian et al. [26] proposed chronological reviews for the major contributions to the motion planning (MP) field throughout a 35-year period, from classic approaches to heuristic algorithms. The authors mentioned that the proportion of heuristic/classic methods grew up rapidly, and ANN and GA approaches possess main part in heuristic method (ANN took 25.53% and GA took 37.23% in 2007).

This paper focuses on 3-link robot hand path planning from initial point to final point by using GA under the consideration of forward/inverse kinematics and the constraints of maximum torque limit.

The reminder of this paper is constructed as follow: section 3 mentions the problem formulation and objective, section 4 presents the methodology, and section 5 shows the results and discussion.

3. Robot Hand Kinematics, Dynamics and Path Planning Strategy

3.1 Problem statement

The robot manipulator with 3 DOF must execute the task from an initial to a final position without collision with any obstacle as soon as possible. The Fig. 1 shows the robot environment with 4 obstacles with circular shapes. Since each joint has its maximum torque limit, the problem is optimization to find the shortest path and traveling time under the constraints of maximum torque limit and obstacles avoidance.

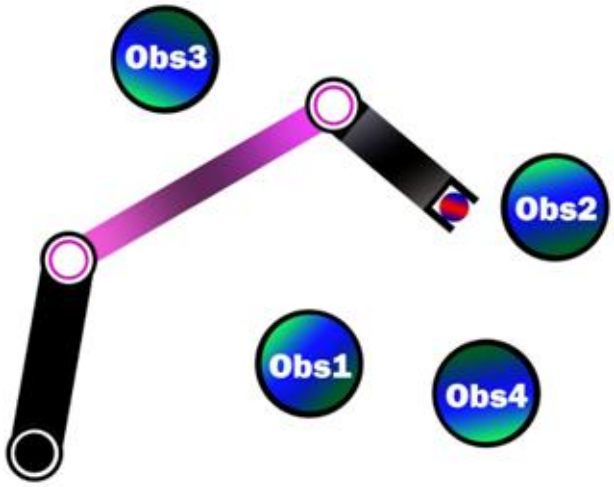

Figure 1: Diagram of the robot manipulator environment

3.2. Forward kinematics

A manipulator is composed of serial links which are affixed to each other revolute or prismatic joints from the base frame through the end-effector. Calculating the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics. In order to have forward kinematics for a robot mechanism in a systematic manner, one should use a suitable kinematics model. Denavit-Hartenberg method that uses four parameters is the most common method for describing the robot kinematics. These parameters α_{i-1}, d_i and θ_i are the link length, link twist, link offset and joint angle, respectively. A coordinate frame is attached to each joint to determine DH parameters. Zi axis of the coordinate frame is pointing along the rotary or sliding direction of the joints. The general transformation matrix T_i^{i-1} for a single link can be obtained as follows:

$$T_i^{i-1} = R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)Q_i(d_i) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where R_x and R_z present rotation, D_x and Q_i denote translation, and $c\theta_i$ and $s\theta_i$ are the short hands of $\cos\theta_i$ and $\sin\theta_i$, respectively. The forward kinematicsof the end-effector with respect to the base frame is determined by multiplyingall of the T_i^{i-1}matrices,

$$T_{end\_effector}^{base} = T_1^0 T_2^1 \cdots T_n^{n-1} \quad (2)$$

3.3 Inverse kinematics

The conversion of the position and orientation of a manipulator end-effector from Cartesian space to joint space is called as inverse kinematics problem. There are two solutions approaches namely, geometric and algebraic used for deriving the inverse kinematics solution, analytically. In this paper, geometric approach is used for inverse kinematics of 3-link robot arm as seen in Fig. 2, where we have to calculate θ_1, θ_2 and θ_3 from the position of the point $E(E_x, E_y)$ and the orientation angle φ of the last linkl_3.

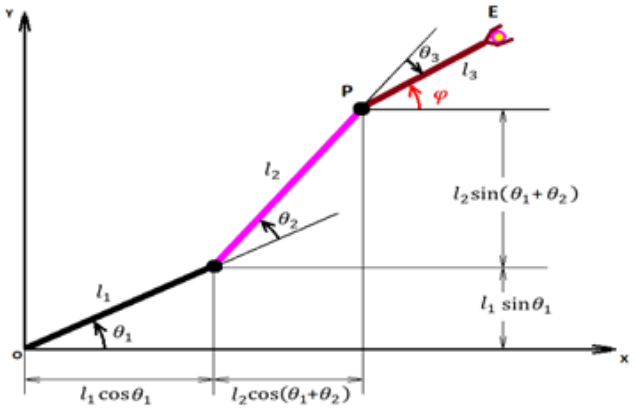

Figure 2: Solving the inverse kinematics based on trigonometry

The position of $P(P_x, P_y)$ is determined as follows:

$$P_x = E_x - l_3\cos\varphi \quad (3)$$
$$P_y = E_y - l_3\sin\varphi \quad (4)$$

On the other hand, the position of $P(P_x, P_y)$ is related to l_1, l_2, θ_1 and θ_2 as follows:

$$P_x = l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \quad (5)$$
$$P_y = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \quad (6)$$

Using the trigonometry equation$\cos^2\theta_i + \sin^2\theta_i = 1$ $(i = 1, 2)$, the following equation is obtained:

$$\theta_2 = \cos^{-1}\frac{P_x^2+P_y^2-l_1^2-l_2^2}{2l_1l_2} \quad (7)$$
$$\theta_1 = \cos^{-1}\frac{P_x(l_1+l_2\cos\theta_2)+P_y l_2\sin\theta_2}{P_x^2+P_y^2} \quad (8)$$
$$\theta_3 = \varphi - \theta_1 - \theta_2 \quad (9)$$

By substituting (3) and (4) into (7) and (8), the angles θ_1, θ_2 and θ_3 will be calculated from E_x, E_y andφ.

3.4 Dynamics of arm manipulator

The dynamic equations of the arm manipulator are represented by the following Lagrange method:

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (10)$$

where, $q, \dot{q}, \ddot{q} \in R^n$ denote the joint angle, the joint velocity and joint acceleration, $M(q) \in R^{n\times n}$ is the manipulator inertia matrix, $C(q, \dot{q}) \in R^{n\times n}$ is the centrifugal and Coriolis force matrix, $G(q) \in R^n$ is the gravitational force vector, and,$\tau(t)$ denotes the torque.

3.5 Path planning strategy with polynomials

The point-to-point trajectory can be constructed by several connected segments with continuous acceleration at the intermediate via point as shown in Fig. 3. 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 quantic polynomial can be used. Let us assume that there is mpintermediate via points between the initial and final points.

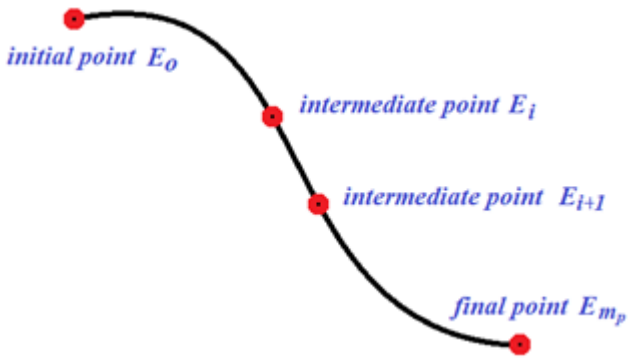

Figure 3: Intermediate points on the point-to-point trajectory

Between the initial points to mp intermediate via points, a quadrinomial is used to describe these segments as:

$$E_{i,i+1}(t) = a_{i0} + a_{i1}t_i + a_{i2}t_i^2 + a_{i3}t_i^3 + a_{i4}t_i^4, (i = 0, \dots, m_p - 1) \quad (11)$$

Where (ai0, ai1, ai2, ai3, ai4) are constants, and the constraint are given as:

$$E_i = a_{i0} \quad (12)$$
$$E_{i+1} = a_{i0} + a_{i1}T_i + a_{i2}T_i^2 + a_{i3}T_i^3 + a_{i4}T_i^4 \quad (13)$$
$$\dot{E}_i = a_{i1} \quad (14)$$
$$\dot{E}_{i+1} = a_{i1} + 2a_{i2}T_i + 3a_{i3}T_i^2 + 4a_{i4}T_i^3 \quad (15)$$
$$\ddot{E}_i = 2a_{i2} \quad (16)$$

Where $T_i$is the execution time from point E_i to point E_{i+1}. The five unknowns can be solved as:

$$a_{i0} = E_i \quad (17)$$
$$a_{i1} = \dot{E}_i \quad (18)$$
$$a_{i2} = \ddot{E}_i/2 \quad (19)$$
$$a_{i3} = (4E_{i+1} - \dot{E}_{i+1}T_i - 4E_i - 3\dot{E}_iT_i^2)/T_i^3 \quad (20)$$
$$a_{i4} = (\dot{E}_{i+1}T_i - 3E_{i+1} + 3E_i + 2\dot{E}_iT_i + \ddot{E}_iT_i^2/2)/T_i^4 \quad (21)$$

The acceleration of intermediate point E_{i+1} point can be calculated as:

$$\ddot{E}_{i+1} = 2a_{i2} + 6a_{i3}T_i + 12a_{i4}T_i^2 \quad (22)$$

The last segment between intermediate point E_{m_p} and the final point can be described by quantic polynomial as:

$$E_{i,i+1}(t) = b_{i0} + b_{i1}t_i + b_{i2}t_i^2 + b_{i3}t_i^3 + b_{i4}t_i^4 + b_{i5}t_i^5, (i = m_p) \quad (23)$$

Where the constants are given as:

$$E_i = b_{i0} \quad (24)$$
$$E_{i+1} = b_{i0} + b_{i1}T_i + b_{i2}T_i^2 + b_{i3}T_i^3 + b_{i4}T_i^4 + b_{i5}T_i^5 \quad (25)$$
$$\dot{E}_i = b_{i1} \quad (26)$$
$$\dot{E}_{i+1} = b_{i1} + 2b_{i2}T_i + 3b_{i3}T_i^2 + 4b_{i4}T_i^3 + 5b_{i5}T_i^4 \quad (27)$$
$$\ddot{E}_i = 2b_{i2} \quad (28)$$
$$\ddot{E}_{i+1} = 2b_{i2} + 6b_{i3}T_i + 12b_{i4}T_i^2 + 20b_{i5}T_i^3 \quad (29)$$

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

$$b_{i0} = E_i \quad (30)$$
$$b_{i1} = \dot{E}_i \quad (31)$$
$$b_{i2} = \ddot{E}_i/2 \quad (32)$$
$$b_{i3} = (20E_{i+1} - 20E_i - (8\dot{E}_{i+1} + 12\dot{E}_i)T_i - (3\ddot{E}_i - \ddot{E}_{i+1})T_i^2)/2T_i^3 \quad (33)$$
$$b_{i4} = (30E_i - 30E_{i+1} + (14\dot{E}_{i+1} + 16\dot{E}_i)T_i + (3\ddot{E}_i - 2\ddot{E}_{i+1})T_i^2)/2T_i^4 \quad (34)$$
$$b_{i5} = (12E_{i+1} - 12E_i - (6\dot{E}_{i+1} + 6\dot{E}_i)T_i - (\ddot{E}_i - \ddot{E}_{i+1})T_i^2)/2T_i^5 \quad (35)$$

From above equations, the total parameters to be determined are the joint angles of each intermediate via point (n×mp parameters), the joint angular velocities of each intermediate point (n×mp parameters), the execution time for each segment (mp+1 parameters), and the posture of the final configuration (n-m).

Therefore, for 3-link robot case, it used mp= 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 (22). 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.

4. GA appliance to Path Planning

4.1 GA path planning scheme

The GA planning scheme focuses an optimized trajectory having shortesttrajectory length, traveling distance of the manipulator and traveling time, without colliding with any obstacle in the workspace, while not exceeding a maximum pre-defined torque. The path 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:

$$[\theta_1, \theta_2, \theta_3, \theta_g, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, t_1, t_2] \quad (36)$$

Where θ_i and $\dot{\theta}_i$ are intermediate joint angle and velocity for ith joint respectively, $\theta_g$is the global angle ofthe final configuration of the end-effectors which is denoted as φ in Fig. 2, t_1 is execution time from initial point tointermediate point, and$t_2$is execution time from intermediate point to final point.

4.2 Genetic operators

The initial population of chromosomes can be generated at random in the pre-defined interval of eachchromosome, and the search is then carried out among this population. The main different operators adopted in the GA are reproduction, crossover and mutation.

The reproduction operator is used to check the eligibility of a string (path) to be in the mating pool. A solution with high fitness will get more number of copies in the mating pool; whereas a solution of low fitness may or may not have copies in the mating pool. Since the number of individuals in the population and the mating pool are same, week solutions will not be permitted to enter into the mating pool. The fitness level required to enter the mating pool is calculated based on the ration of string fitness to average fitness of the population.

The crossover operator is applied to two cross-sites which are randomly selected. The crossover operator adopted the single point technique with a given probability Pc, therefore, the crossover point will be allowed for only one gene randomly selected, in other words, the crossover operator will not disrupt the other genes.

The mutation operator refers to alteration of character values in individual string with a given probability Pm. In this paper, the mutation operator replaces only one gene value withanother one which is generated randomly with a specified range. The prime of objective of mutation is to keep diversity in population.

4.3 Fitness function in GA

Four indices are used to estimate the evolving trajectory of robotic manipulators in thefree workspace. All indices can

be translated into penalty functions to be minimized. Eachindex is computed individually, and all of them are integrated in thepenalty function evaluation with their weight factors which can be set differently according to the actual situation. The penalty function $F_{penalty}$ adoptedfor evaluating the candidate trajectories in the free workspace is defined as:

$$F_{penalty} = w_1 f_{torque} + w_2 f_\theta + w_3 f_{traj} + w_4 t_{total} \quad (37)$$

The optimization goal is to find a set ofdesign parameters in (36),which minimize $F_{penalty}$ according to thepriorities given by the weight factors w_i(i=1,.., 4),where each different set of weighting factors must resultsin a different solution.

The f_{torque} index represents the amount of excessivedriving with reference to the maximum torque $\tau_{i_{max}}$ which is pre-defined for the ith joint motor.

$$f_{torque} = \sum_{j=1}^{b} \sum_{i=1}^{a} f_i^j \quad (38)$$

$$f_i^j = \begin{cases} 0, & |\tau_i^j| < \tau_{i_{max}} \\ |\tau_i^j| - \tau_{i_{max}}, & \text{otherwise} \end{cases} \quad (39)$$

Where a is number of robot links, and b is number of intermediatepoints from the initial to final position in the path. The torque can be calculated from (10), where the mass of each link is considered as a point mass at the distal end of each link for simplicity.

The index $f_\theta$represents the total joint traveling distanceof the manipulator as follow:

$$f_\theta = \sum_{i=1}^{a} \sum_{j=2}^{b} |\theta_{ij} - \theta_{ij-1}| \quad (40)$$

The index f_{traj}represents the total Cartesian trajectory length as follow:

$$f_{traj} = \sum_{j=2}^{b} d(P_j, P_{j-1}) \quad (41)$$

Where P_j is the position of jth intermediate point in the trajectory, and $d(\cdot,\cdot)$ is a function which gives the distance between two points.

The index t_{total} represents the total traveling time for robot path planning as follow:

$$t_{total} = t_1 + t_2 \quad (42)$$

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

For the case of obstacle existence in the workspace, obstacle avoidance index function f_{ob} has to be combined with free workspace penalty function $F_{penalty}$ to form a final fitness functionF_{fit}, as shown in the following:

$$F_{fit} = f_{ob}/F_{penalty} \quad (43)$$

Where, collision avoidance index function f_{ob} can be set as follow:

$$f_{ob} = \begin{cases} 1, & \sum_{j=1}^{b} \sum_{i=1}^{a} (link_{ij} \cap obstacles) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (44)$$

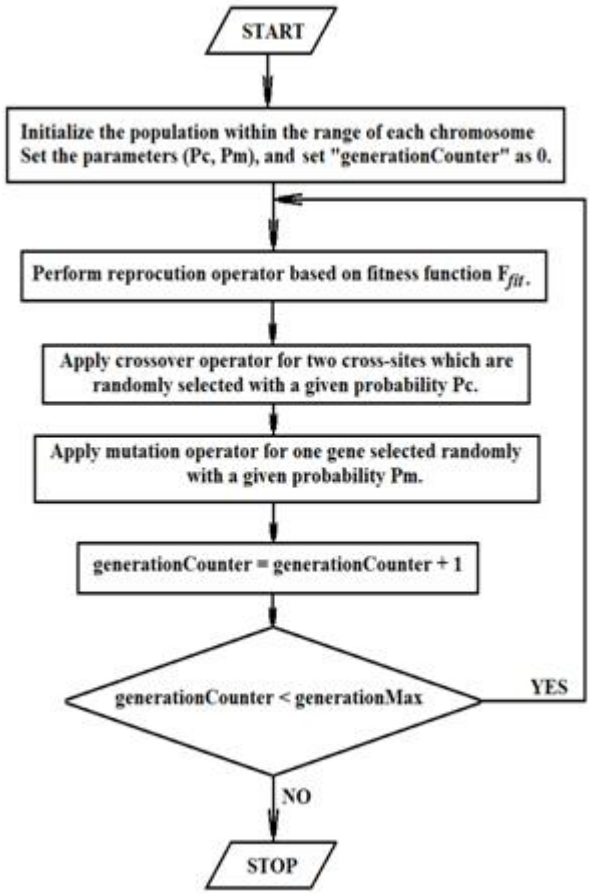

Figure 4: Flowchart of GA path planning

5. Result and discussion

Proposed GA path planning is implemented in MATLAB GUI.

The case of 3R robot hand arm withinitial point (x=0.686m, y=2.268m, θ_g=40°) and final point (x=-2 m, y=-0.5m) is considered while maximum torque for each joint is respectively given as $\tau_{1max} = 50Nm, \tau_{2max} = 30Nm, \tau_{3max} = 10Nm$. The parameters of robot hand arm are predefined as $l_1 = 1.2m$, $l_2 = 0.9m$, $l_3 = 0.7m$, $m_1 = 1.1kg$, $m_2 = 0.8kg$, $m_3 = 0.6kg$.The joint velocities and accelerationsfor the initial and final position are assumed as zeros, and all jointsare assumed to rotate 2π freely. The parameters of GA are given as Pc=0.85, Pm=0.07, *generationMax*=200 and *populationSize*=50. The weight factors are set as$[w_1, w_2, w_3, w_4] = [2, 1, 3, 4]$, and, the 4 obstacles have circular shape with radius 0.2m and their centers are placed at (-0.4, 1.5),(1.5, 0.7),(1.1, -0.9) and (-1.4, 0.5), respectively. The above settings are entered in MATLAB GUI as shown in Fig. 5.

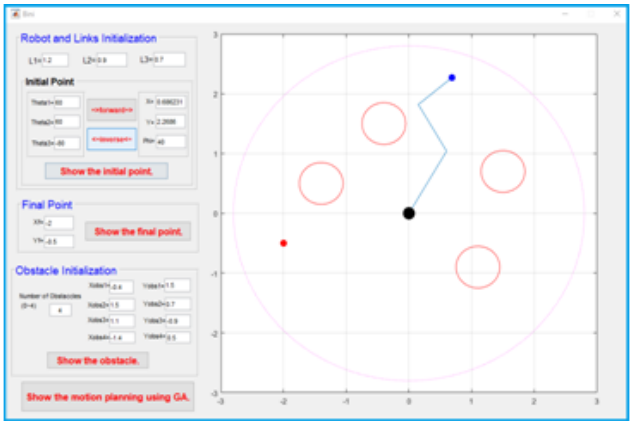

Figure 5: Simulation in MATLAB GUI

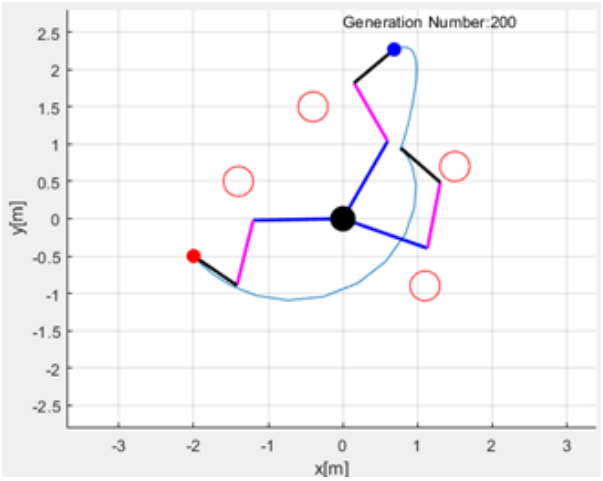

Figure 6: Optimal Cartesian path with obstacles avoidance

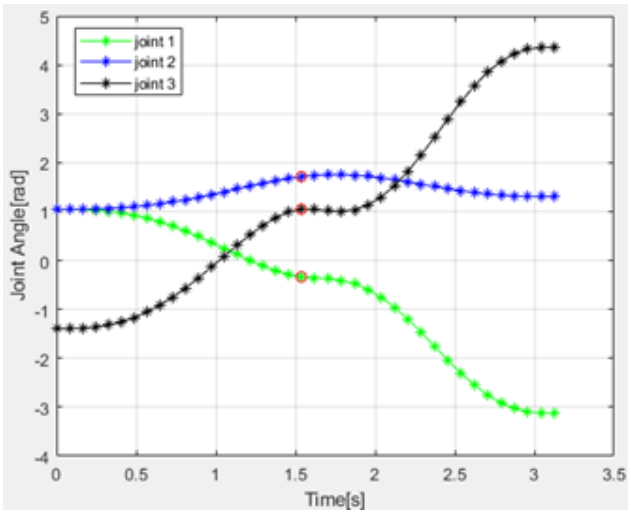

Figure 7: Joint angles versus traveling time

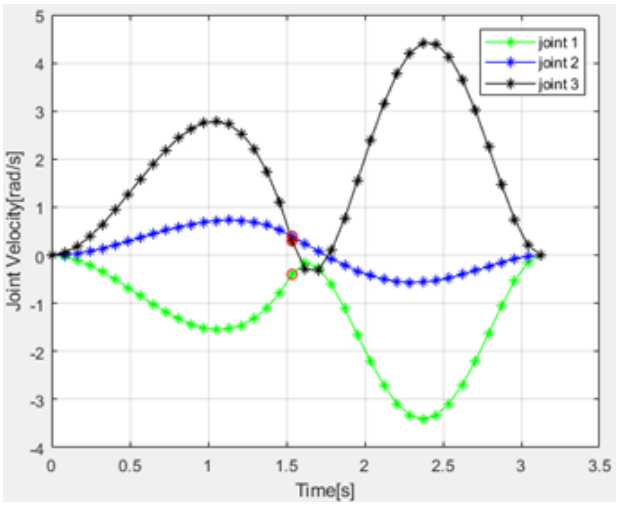

Figure 8: Joint velocities versus traveling time

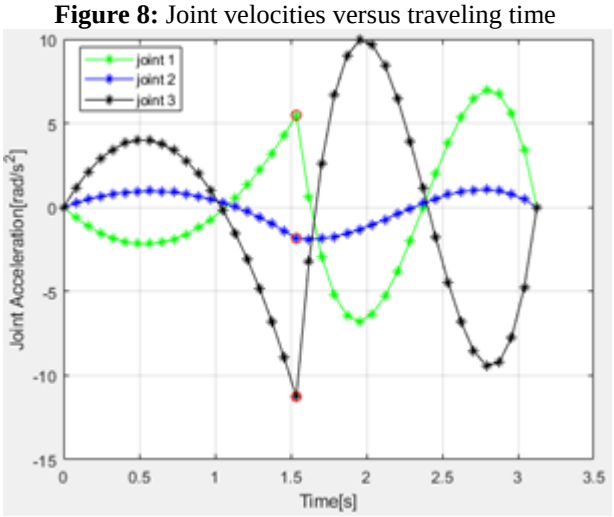

Figure 9: Joint accelerations versus traveling time

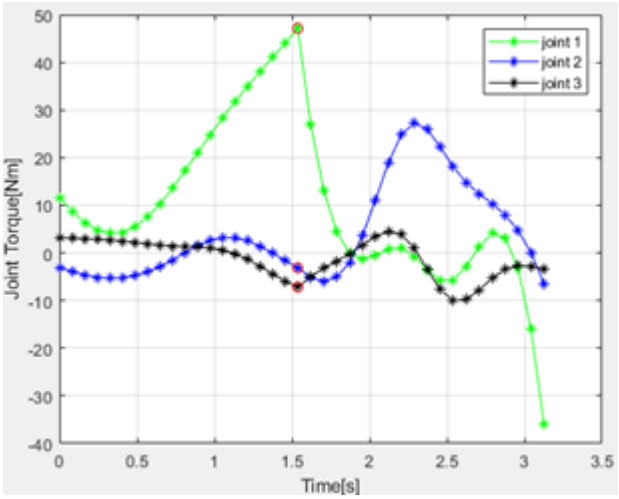

Figure 10: Joint torques versus traveling time

Fig. 6 to Fig. 13 show the simulation results.The optimized Cartesian path is shown in Fig. 6, where the path is not collided with any obstacle. The angle, angular velocity, angular acceleration and torque for each joint are shown in Fig. 7 to Fig. 10, where red spots denotes the joint angle, angular velocity, angular acceleration and torque, respectively, at the optimized intermediate time 1.54s. Especially, the joint torque curves in Fig. 10 are not exceeded 50Nm for green curve, 30Nm for blue one and 10Nm for black one, respectively, and these values are set as

maximum torque limit for each joint when starting simulation.Table 1 shows the values from Fig. 7 to Fig. 10, which are calculated from equation (7) to (10) and (36).

Table 1: Joint angles and torques

Index		Joint1	Joint2	Joint3
joint angle [rad]	t=0	1.047	1.047	-1.396
	t=intermediate time	-0.336	1.714	1.048
	t=final time	-3.125	1.306	4.364
	maximum absolute angle	3.125	1.754	4.364
joint torque [Nm]	t=0	11.691	-3.024	3.156
	t=intermediate time	47.143	-3.121	-7.196
	t=final time	-35.868	-6.442	-3.411
	maximum absolute torque	47.143	27.262	-9.998

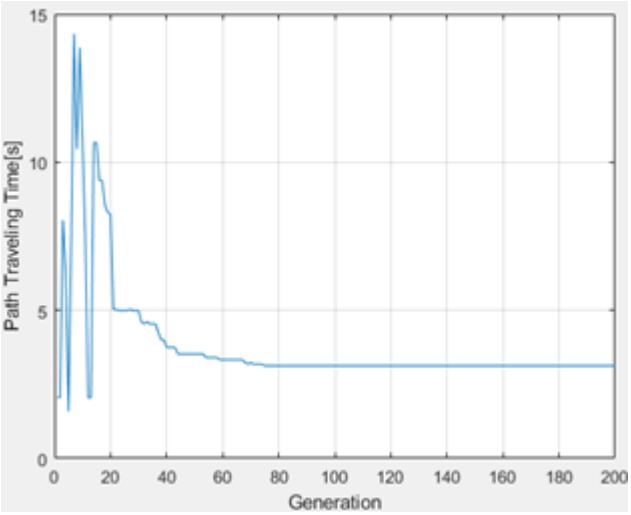

Figure 11: Path traveling time versus generation

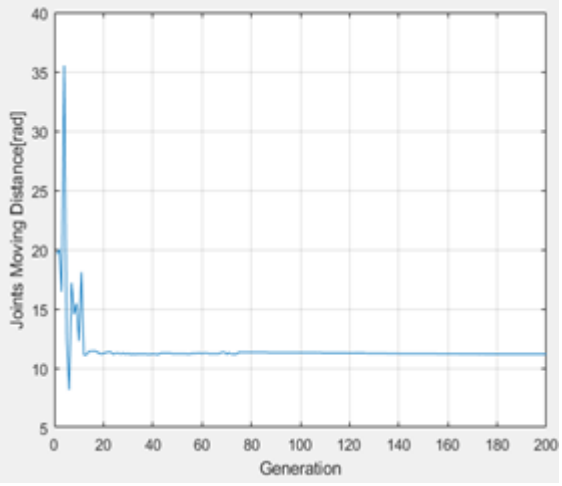

Figure 12: Total joints moving distance versus generation

Fig11 to Fig. 13 are for the convergence of path traveling time, total joints moving distance and path length versus generation. The convergence values at *generation Max*=200 show the optimal ones.

Table 2 shows the optimal path traveling time, total joints moving distance, path length and intermediate time after the final generation of GA, which are extracted Fig. 11 to Fig. 13 and equation (40) to (42).

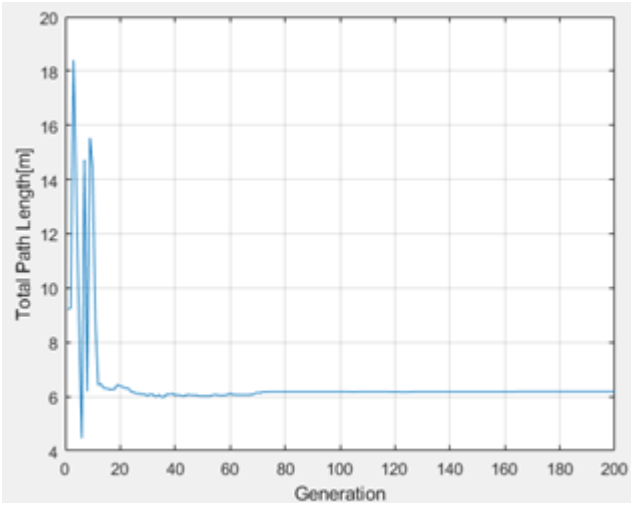

Figure 13: Path length versus generation

Table 2: Optimal values after final generation

Index	Optimal value
path traveling time, [s]	3.125
total joints moving distance, [rad]	11.177
path length, [m]	6.180
intermediate time, [s]	1.537

6. Conclusion

In this paper, robot hand path planning combined with GA was suggested. The forward/inverse kinematics, dynamics and polynomial path planning strategy of 3R robot arm were studied for redundancy of final configuration and GA approach. The chromosome was designed and GA operators were applied according to the case of 3R links. The fitness function of GA was designed from the needs that have to find the shortest Cartesian trajectory length, traveling time and joints moving distance while avoiding obstacles and not exceeding maximum torque. The proposed method was implemented by MATLAB. The simulation results show the validity of this method and satisfy the objectives and constraints.

References

[1] S. LaValle. “Planning Algorithms”. Cambridge University Press, 2006.
[2] Yang, Z.Q., Liu, L.B. and Yang, W.D. “Flexibleinspection path planning based on adaptive geneticalgorithm”, Proceedings of Control and Decision Conference,China, pp. 1558-1563, 2008.
[3] J. Bialkowski, S. Karaman, and E. Frazzoli. “Massively parallelizing the RRT and the RRT”,Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3513–3518, 2011.
[4] J. Pan and D. Manocha. GPU-based parallel collision detection for fast motion planning. International Journal of Robotics Research, 31(2), pp. 187–200, 2012.
[5] A. Kumar Kashyap and A. Pandey. “Different Nature-Inspired Techniques Applied for Motion Planning of Wheeled Robot: A Critical Review”, International Journal of Advanced Robotics And Automation, 2018.
[6] A. Motahari, H. Zohoor and M. HabibnejadKoranyem. “A new obstacle avoidance method for discretely

actuated hyper-redundant manipulators", Scientia IranicaB(2012) 19(4), pp.1081-1091, 2012.
[7] Dong Han, Hong Nie, Jinbao Chen and Meng Chen. "Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection", Robotics and Computer–Integrated Manufacturing 49, pp. 98-104, 2018.
[8] E. Papadopoulos, I. Papadimitriou and I. Poulakakis. "Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems", Robotics and Autonomous Systems 51(2005), pp. 229–247, 2005.
[9] S. Kucuk and Z. Bingul. "Robot Kinematics:Forward and Inverse Kinematics", Industrial Robotics: Theory, Modelling and Control, pp. 117-148, 2006
[10] M. Erkan, M. Taylan and L.Canan. "Forward and Inverse Kinematics Analysis of Denso Robot", Proceedings of the International Symposium of Mechanism and Machine Science, 2017
[11] A. Jiang, X. Yao, M. Cheng and J. Zhou. "Kinematics analysis and experiment of a lily picking mechanical arm", The 2nd 2018 Asian Conference on Artificial Intelligence Technology(ACAIT 2018), pp 1674-1681, 2018.
[12] Nizar CHATTI and Abderrazak CHATTI. "Robust adaptive fuzzy control and strategy of avoidance of obstacles for a manipulator arm to serve the people Disabilities", IFAC Proceedings Volumes (Vol. 43, Issue 8), pp. 402-409, 2010.
[13] B. Sharma, J. Vanualailai and S. Singh. "Lyapunov-based nonlinear controllers for obstacle avoidance with a planar n-link doubly nonholonomic manipulator", Robotics and Autonomous Systems 60 (2012), pp. 1484–1497, 2012.
[14] A. Jain and S. Niekum. "Efficient Hierarchical Robot Motion Planning Under Uncertainty and Hybrid Dynamics", 2nd Conference on Robot Learning(CoRL 2018), Zurich, Switzerland, 2018.
[15] A. Atias, K. Solovey, O. Salzman and D. Halperin "Effective Metrics for Multi-Robot Motion-Planning", Robotics: Science and System, 2017.
[16] Duong Le and ErionPlaku. "Multi-Robot Motion Planning with Dynamics Guided by Multi-Agent Search", Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 5314-5318, 2018.
[17] Duong Le and ErionPlaku. "Cooperative, Dynamics-Based, and Abstraction-Guided Multi-robot Motion Planning", Journal of Artificial Intelligence Research 63, pp. 361-390, 2018.
[18] A. Dobson, K. Solovey, R. Shome, D. Halperin and K. Bekris. "Scalable Asymptotically-Optimal Multi-Robot Motion Planning", 1st IEEE International Symposium on Multi-Robot and Multi-Agent Systems(MRS), 2017.
[19] M. Dutra and A. Carlos. "PROGRAM FOR DETERMINATION OF THE MOVEMENT TRAJECTORY OF A LOCOMOTOR ROBOT THROUGH THE VORONOI DIAGRAM", 17th International Congress of Mechanical Engineering, 2003.
[20] D. Moriarty and R. Miikkulainen. "Evolving Obstacle Avoidance Behavior in a Robot Arm", Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, 1996.
[21] M. Duguleana, F. Grigore, A. Teirelbar and G. Mogan. "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning", Robotics and Computer-Integrated Manufacturing 28, pp. 132–146, 2012.
[22] A. Pajaziti and H. Cana. "Robotic-Arm-Control-with-Neural-Networks-Using-Genetic-Algorithm-Optimization-Approach", International Journal of Mechanical and Mechatronics Engineering, Vol. 8, No. 8, 2014.
[23] F. Zahra, L. Bakkali, and Y. Lakhal. "Optimization of Arm Manipulator Trajectory Planning in the Presence of Obstacles by Ant Colony Algorithm", Procedia Engineering 181, pp. 560 – 567, 2017.
[24] I. Hassani, I. Maalej and C. Rekik. "Robot Path Planning with Avoiding Obstacles in Known Environment Using Free Segments and Turning Points Algorithm", Mathematical Problems in Engineering, Hindawi, 2018.
[25] A. Cosic, M. Susic and D. Katic. "Advanced Algorithms for Mobile Robot Motion Planning and Tracking in Structured Static Environments Using Particle Swarm Optimization", SERBIAN JOURNAL OF ELECTRICAL ENGINEERING, Vol. 9, No. 1, pp. 9-28, 2012.
[26] E. Masehian and D. Sedighizadeh, "Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review", International Journal of Mechanical and Mechatronics Engineering, Vol. 1, No. 5, 2007.