



Genetic Algorithm Based Optimal Trajectories Planning for Robot Manipulators on Assigned Paths

Obe O. O.¹, Ajayi E. A.², Odewale O. O.³

¹Computer Science Department, Federal University of Technology, Akure, Nigeria, ooobe@futa.edu.ng

²Faculty of Information Science and Technology, Multimedia University, Malaysia, ebeseun@gmail.com

³Computer Science Department, Ibadan City Polytechnic, Ibadan, Nigeria, omolaraodewale@gmail.com

ABSTRACT

This research work focuses on finding an optimal trajectory for a robot manipulator taking into consideration the dynamics constraints of the manipulator. A 3-DOF planar robot was modeled using MATLAB/SIMULINK toolbox. The simulation of the robot manipulator was carried out using the Genetic Algorithm to find the optimal trajectory both in the workspace with and without obstacles. In both cases, the Genetic Algorithm (GA) generated optimal trajectories. The results of both environments were also compared. The increase in the simulation result in an obstacle existence environment made it possible for optimal trajectory devoid of collision with any obstacle in the working area. Thereafter, result comparison was done with a similar work and the GA method produced a more desired result in terms of execution time.

Key words: Cartesian space, Constraints, Genetic algorithm, Manipulators, Obstacle, Optimization, Trajectory planning.

1. INTRODUCTION

Robots of various types have found applications in human endeavors. Robots are either playing assistive roles or have entirely substituted human experts. The area of applications ranges from simple industrial robots to complex fully autonomous space robots for Mars exploration. The widespread is as a result of the excellent versatility and flexibility of robots, enabling them to perform multi tasks. Speed, precision and repeatability with intelligence are robots desirable features. Manipulator robots used in the industries are essentially a mechanical structures comprising of fixed rigid bodies (arms) interlocked with one another [1], [3]. Wrists and end-effectors of the manipulators are the parts that provide mobility: a wrist gives dexterity, while an end-effector performs the task for which the robot is designed. Task execution is attained based of implementation of one or more movements imposed on the end-effector(s). The

movement may be free or bound depending on the interaction of the end-effector with its operational environment.

Optimal motion scheduling is very important to the operation of robot manipulators. This entails the generation of a path from start to goal that fit the desired objectives. These include path traveling distance/time interval minimization, least energy consumption or obstacle avoidance, and satiating the robot kinematics and dynamics

Industrial robots must have a high degree of flexibility to carry out various technical operations and operate with manual workers. A manipulator robot has many constraints compared to humans in terms of mobility in order to realize even basic movements in a workspace. Different tasks have to be solved to move between two space points. It is necessary to find the best trajectory, avoid obstacles and collisions, consider other limitations, and achieve the high efficiency and productivity of work. The path or route planning is the planning of the whole way from one point to another, including ending in defined path points. The route entails many continuous motion routes that need the route scheduling. If a path cannot be previously scheduled because of insufficient previous data, the motion task is called pathfinding [4] [5]

Control tasks in robots can be categorized in various ways. In different situations, different path scheduling techniques may be used. In path scheduling, there exists two types of constraints that must be addressed. One, it is possible that obstacles could restrict the movement of a robot and so obstacle constraints have to be considered. Secondly, path selection may have some kind of constraint. These constraints are identified as path constraints. Robots trajectories are usually filled with some obstacles. Robots must intelligently navigate its way through the constraints.

Path planning strategies can be used to solve robotic path planning tasks. The position-controlled motion is the movement between assigned and referred path points along interpolated trajectories. The path assigned is the path that has regular defined points that must be passed through

unconditionally. Robots are subject to all the mechanical constraints. Robot manipulators with many joints (prismatic or revolute) are obviously constrained by its physical limit of motion. Optimal solutions must be found for the limits of joint and actuator positions, jerk, velocity and acceleration. The physical nature also means that there are proportions that must be considered, hence collision and kinematics are of outmost consideration. The energy dispensed to accelerate, hold and brake by manipulator must be optimized by reducing unnecessary energy expenditures.

2. METHODOLOGY

In this work, a robotic manipulator system was developed using a planar articulated robot with three degrees of freedom and a program based on graphical interface in MATLAB/SIMULINK for manipulator control and simulation. The graphical interface enables users to pass instructions to the robotic handler through simple pre-programmed options. The Initial position and the destination or targeted position which is the goal of the manipulator can be specified. The control interface was developed using the MATLAB GUIDE (Graphical User Interface Development Environment) toolbox which provides GUI creation tools for custom applications. It comprises boxes and tools to present the graphical User Interface. The software built on the principle of the Genetic Algorithm (GA) as shown in figure (1), and it is able to simulate a robot manipulator of three degrees of freedom.

The path planning problem was addressed in this paper along the following lines: for a specified robot manipulator route and set of its motion constraints, a trajectory is to be computed that permits traversal of the route in the least possible time. Limitation in the joint speeds, limitation in the joint accelerations, and limitation in the torques constraints are assigned. Using Genetic Algorithm, the initial population of strings is generated at random and the search is then carried out among this population. The strings generated are synonymous with different possible solutions. The progression of the population elements is non-generational.

Genetic Algorithms are optimization technique that is stochastic in search for optimal solution. It is similar to natural evolution typified with normal selection and nature-based genetics [6] [9] [11] [12] [13]. The GA uses its three operators (reproduction, crossover and mutation) to evolve and generate the most fitting solutions among the generated string structures.

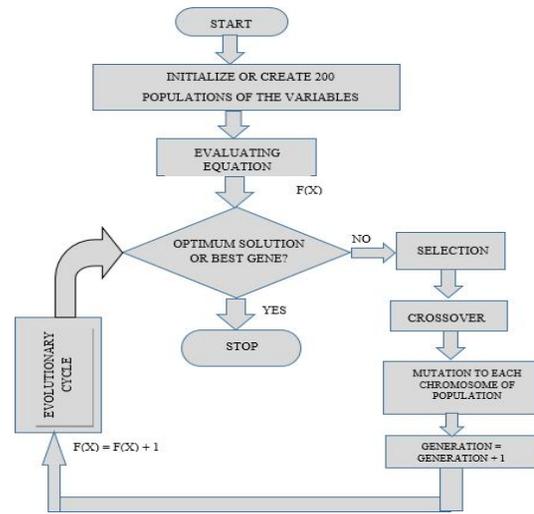


Figure 1: System Flow Chart

2.1 The genetic algorithm motion planning model

The GA scheduling model delivers an optimized route for the manipulator with minimum space, optimized time, and equally not surpassing an extreme pre-specified torque, without collision with any obstacle in the working area. The motion scheduling implements direct kinematics to circumvent singularity complications. Using real codification, the route parameters are encoded as strings (population) for GA implementation.

Equation (1) expresses the nine parameters to be optimized.

$$[q_1, q_2, q_3, q_g, \dot{q}_1, \dot{q}_2, \dot{q}_3, t_1, t_2] \quad (1)$$

where q_i and \dot{q}_i are intermediate joint angle and velocity of the i^{th} joint respectively. q_g represents the global angle of the final configuration of the end-effectors which equals the sum of joint angles of the manipulator [7][8], t_1 is execution time from the starting point to the transitional via point, and t_2 is execution time from transitional to final point.

Transition Conditions

The evolving trajectory of the robotic manipulator's free working area are qualified with four keys. All the keys are decoded into penalty functions to be minimized. Each key is calculated discretely and is incorporated into the fitness function estimation. Equation (2) represents the fitness function f_f chosen for evaluating the candidate routes:

$$-f_f = \beta_1 f_{ot} + \beta_2 f_q + \beta_3 f_c + \beta_4 t_T \quad (2)$$

Finding a set of design parameters that minimize f_f according to the priorities given by the weighting factors β_i ($i = 1, \dots, 4$), where each different set of weighting factors must result in a different solution is set as the optimization goal.

In equation (3), the f_{ot} index stands for 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 route under consideration using the equation (4) which is called the cost function [10][2]

$$f_{or} = \sum_{j=1}^a b \sum_{i=1}^b f_i^j \quad (3)$$

$$f_i^j = \begin{cases} 0 & |\tau_i^j| < \tau_{i,max} \\ |\tau_i^j| - \tau_{i,max} & \text{otherwise} \end{cases} \quad (4)$$

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

Iterative Newton-Euler dynamics algorithm [9] is used to obtain the dynamics of equations of the 3-DOF manipulator. In equation (5), the index f_q denotes the sum total of joint

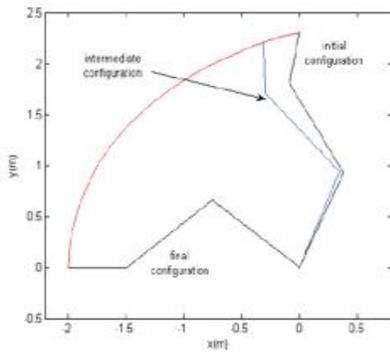


Figure 2: Cartesian path for the end effectors

traveling distance of the robot.

$$f_q = \sum_{i=1}^a \sum_{j=2}^b |q_{iy} - q_{y-1}| \quad (5)$$

In equation (6), the index f_c represents the total Cartesian trajectory length of the manipulator.

$$f_c = \sum_{j=2}^b d(p_j, p_{j-1}) \quad (6)$$

The parameter p_j is the robot j^{th} intermediate arm Cartesian position and the function $d(.,.)$ gives the distance between the two arguments. In equation (7), the index t_T denotes the sum total time spent for robot motion.

$$t_T = t_1 + t_2 \quad (7)$$

where t_1 and t_2 are the execution times from start to transitional configuration, and from transitional to destination configuration respectively.

Equation (8) represents the objective function of collision avoidance f_{ob} .

$$f_{ob} = \begin{cases} 1 & \sum_{j=1}^a b \sum_{i=1}^b a(\text{link}_{ij} \cap \text{obstacle}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

3. RESULT AND DISCUSSION

In this section, the result of a manipulator robot case study is presented. The GA adopts a crossover probability, P_c of 0.8 per individual (chromosome), a mutation probability P_m of value 0.005 per locus, a population of 200 elements (individual) for intermediate joints angle, joint velocity and traveling time of the arm, a 5-tournament Roulette wheel selection scheme with elitism, and maximum generation m_g of value 80.

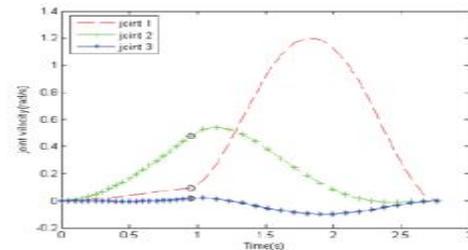


Figure 4: Joint velocity versus time

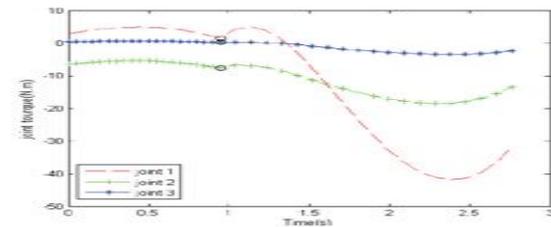


Figure 5: Joints torque versus time

3.1 Case 1: Obstacle free workspace

A 3-DOF robot moving arm from starting coordinate point ($x = 0m, y = 2.3m, q_b = 80^\circ$) to final coordinate point ($x = -2m, y = 0m$.) case study is considered. The robot links have length of ($l_1 = 1m, l_2 = 1m$ and $l_3 = 0.5m$) and mass of ($m_1 = 1kg, m_2 = 1kg$ and $m_3 = 0.5kg$) the maximum allowed torques for joint 1, 2 and 3 of $\tau_{1,max} = 45 Nm, \tau_{2,max} = 20 Nm$ and $\tau_{3,max} = 5 Nm$, respectively. The joints acceleration and velocity of the initial and final settings are taken as zeros. Additionally, all robot joints are free rotate 2π .

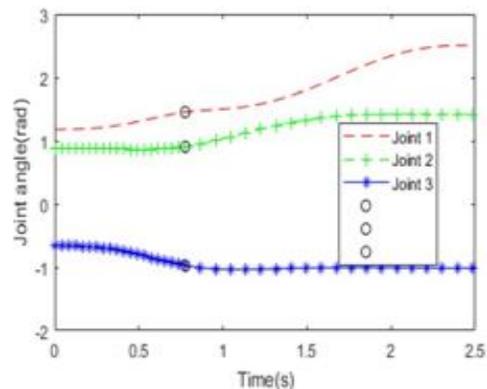


Figure 3: Joint angle versus time

3.2 Case 2: Obstacle existence

A 3-DOF robot arm moving from initial coordinate point ($x = 0m, y = 2.3m, q_b = 80^\circ$) to final coordinate points ($x = -2m, y = 0m,$) and ($x = -0.5, y = 1.8$). The manipulator links possess length of ($l_1 = 1m, l_2 = 1m$ and $l_3 = 0.5m$) and mass of ($m_1 = 1kg, m_2 = 1kg$ and $m_3 = 0.5 kg$) the maximum assigned torques for joint 1, 2 and 3 of $\tau_{1max} = 45 Nm, \tau_{2max} = 20 Nm$ and $\tau_{3max} = 5 Nm$, respectively. The joints acceleration and velocity of the initial and final settings are assigned zeros. Also, all robot joints are free to rotate 2π .

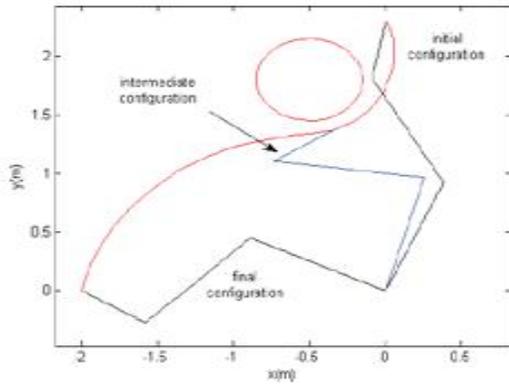


Figure 6: Cartesian path for the end effectors

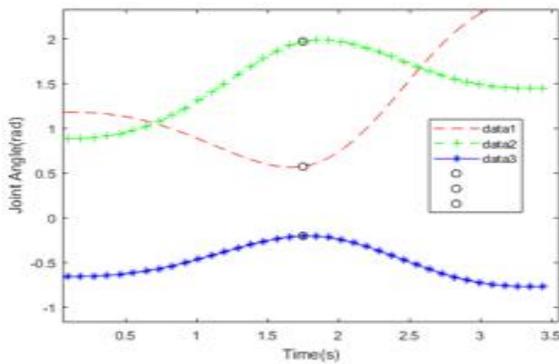


Figure 7: Joint angle versus time

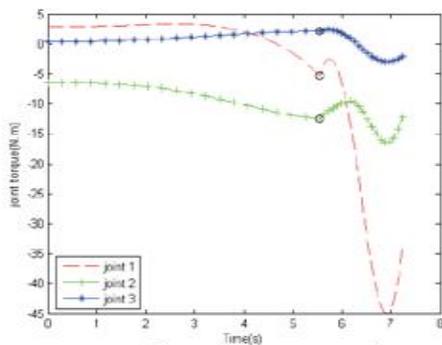


Figure 8: Joint velocity versus time

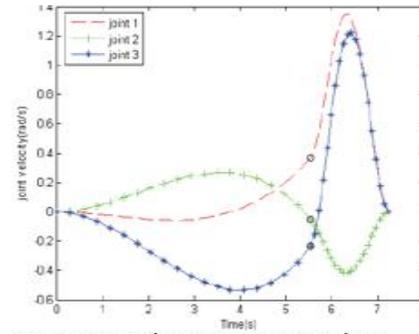


Figure 9: Joint torque versus time

The shorter Cartesian path is depicted in Figure 2. On the other hand, if a straight line is drawn from the starting to the end point it will be the shortest path. This is not possible in real life and will be far from the best one according to the Genetic Algorithm Optimization result. Figures 3 and 4 shows the joint angle and joint velocity with time respectively at the first case of obstacle free workspace. Figure 6 shows the ability of the Genetic Algorithm to decide the parameters that generate the optimal trajectory with an obstacle in the working area with regards to the definite objective functions. In figures 5 and 9, the torque that is calculated along the joint space trajectory in the case of free working area is less than the results from the case study of obstacle existence working area.

Table 1: Optimization Result Comparison

Result value	Without Obstacle	With Obstacles
Total travelling time (secs)	2.76	7.23
Total joint travelling Distance (rad)	1.91	5.78
Total Cartesian trajectory length (m)	3.28	3.42

Still, in the two cases, the joint torque does not exceed the maximum pre-set torque. The final trajectory in figures 2 and 6 has been chosen, hence the Genetic Algorithm is able to solve the optimal route problem.

3.3 Comparison of results

The results of the free working area and the obstacle incidence working area are compared in this sub section. Table 1 shows the value of the total traveling time, total joint traveling distance, and Cartesian trajectory length by equations (7), (5) and (6) respectively, for both free and obstacle incidence working area.

Figure 10 presents the graphical view of the result in Table 1. The aggregate traveling time, aggregate joint traveling distance, and aggregate Cartesian trajectory length values of the obstacle incidence exceeded the results computed from the free working area. The variance empowers the robot to maneuver during its navigation from the starting point all through to the set target point avoiding collision with obstacles.

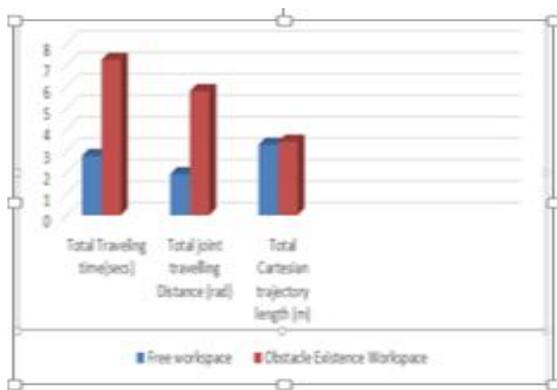


Figure 10: Results of comparison between free and obstacle existence workspace.

Furthermore, we compare the results above with that obtained in [2], where Decision Algorithm was developed to solve time optimality where limitations for the torque was considered. The algorithm was simulated with a model of ABB IRB140 robot manipulator with 6 degrees of freedom. The result computed by the algorithm is as shown in figure 11. The red line is the speed limit. In figure 12, the law computed in terms of joint torque is presented. The time spent by the algorithm to compute the whole trajectory, in this case, is 0.4166 s.

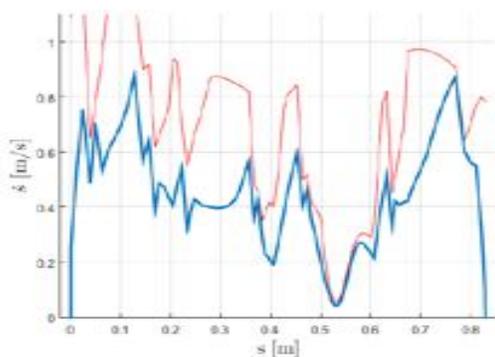


Figure 11: Space's vs Velocity's [2]

Comparing figure 5 in free workspace, figure 9 in obstacle incidence with Figure 12, it shows that using the Genetic Algorithm and Decision Algorithm in both cases did not exceed the torque limit and both algorithms seem to generate a time-optimal trajectory within the minimum time. Conversely, with the genetic algorithm used here, the aggregate trajectory traveling time, aggregate joint traveling distance, and aggregate Cartesian trajectory length can be computed and this could make the Genetic Algorithm ideal.

4. CONCLUSION

The modeling and simulation of optimal trajectories using an articulated planar robot with three degrees of freedom were carried out considering dynamic constraints. The trajectory planning method based on Genetic Algorithm with specific objective functions was presented. The objective function of the Genetic algorithm is to minimize traveling time and space and at the same time not exceeding the maximum pre-specified torque, without collision with any obstacle present in the manipulator working area. Using the case study of a planar robot manipulator revealed that the

method is effective in both obstacle-free and obstacle-incidence workspace for avoiding obstacle collision with minimum traveling time and distance.

REFERENCES

[1] G. Ahmad and S. Noorani. **Optimal Trajectory Planning for Design of a Crawling Gait in a Robot Using Genetic Algorithm.** *International Journal of Advanced Robotic Systems*, Vol. 8, No. 1, pp. 29-36, 2011.

[2] C. Andrea, M. Z. Andrea and R. Paolo. **Online planning of optimal trajectories on assigned paths with dynamic constraints for robot manipulators.** *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2, No. 1, pp. 24-32, 2016.

[3] A. A. Atef. **Optimal Trajectory Planning of Manipulators.** *Journal of Engineering Science and Technology*. Vol. 2, No. 1 pp. 32-54, 2007

[4] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni. **Path Planning and Trajectory Planning Algorithms: A General overview.** (2nd ed.). Italy: Springer Basel International publishing, 2015

[5] A. Harish, A. Chandrashekar, and B. Satish. **Trajectory planning of a Robot Manipulator using Genetic Algorithm.** *India Journal of Mechanical Engineering Department* Vol. 7, Issue 11. pp 284 – 293, 2018.

[6] R. L. Haupt, and S. E. Haupt. **Practical Genetic Algorithms.** (2nd ed.). USA: John Wiley & Sons Interscience publication, 2004.

[7] K. G. Mukul, K. S. Arun, and B. Kamal. **Trajectory Tracking Control of Robot Manipulators.** *International Journal of Computer Applications* Vol. 64, Number 10. pp. 28-32, 2013. <https://doi.org/10.5120/10672-5458>

[8] V. Rajan. **Minimum time trajectory planning.** *IEEE Proceedings of the International Conference on Robotics and Automation*. Vol. 2, Issue 3. pp. 759-764, 2008.

[9] P.E.J. Solteiro, M.J.A Tenreiro, and O.P.B. Moura. **Fractional Order Dynamic in a Genetic Algorithm.** *The 11th International Conference on Advanced Robotics*, Colombia, Portugal, pp. 264 -269, 2003.

[10] S.G. Yue, D. Henrich, X.L. Xu, and S.K. Toss. **Point-to-Point Trajectory Planning of Flexible Redundant Robot Manipulators using Genetic Algorithms.** *Journal of Robotics*. Vol. 20, pp. 269-280, 2002.

[11] D. Zhang and W. Zhang. **Simulation of Robot Manipulator Trajectory Optimization Design.** *International Journal of Research in Engineering and Science*. Vol. 5, Issue 2. pp. 47-54, 2

[12] N. Alsharman, F. Alzyoud, A. Saaidah, and O. Almomani (2020). **Machine Learning and Answer Set Program Rules towards Traffic Light Management.** *International Journal of Advanced Trends in Computer Science and Engineering* 9(3). <https://doi.org/10.30534/ijatcse/2020/08932020>

[13] A. D. M., Esteban, M. L. B., & Orines, H. M. (2020). **Vehicular Speed Limiting Control System in Critical Zones Using Global Positioning System.** *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5). <https://doi.org/10.30534/ijeter/2020/72852020>