Journal of Engineering Research

Volume 7
Issue 5 *This is a Special Issue from the Applied Innovative Research in Engineering Grand Challenges (AIRGEC) Conference, (AIRGEC 2023), Faculty of Engineering, Horus University, New Damietta, Egypt, 25-26 October 2023*

Article 53

2023

Path Planning for a 6 DoF Robotic Arm Based on Whale Optimization Algorithm and Genetic Algorithm

Mahmoud A. A. Mousa, Abdelrahman Tarek Elgohr, Hatem Khater

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

Path Planning for a 6 DoF Robotic Arm Based on Whale Optimization Algorithm and Genetic Algorithm

Mahmoud A. A. Mousa[1,2], Abdelrahman T. Elgohr[3,4], Hatem A. Khater[5]
[1] Computers and Systems Department, Faculty of Engineering, Zagazig University, Egypt. [2] School of Engineering and Computer Science, University of Hertfordshire Hosted by Global Academic Foundation (UH-GAF), Egypt – email: mamosa@zu.edu.eg; m.mousa@herts.ac.uk
[3] Mechatronics Program, Computers and Systems Department, Faculty of Engineering, Zagazig University, Egypt. [4] Department of Mechatronics Engineering, Horus University – Egypt, New Damietta, Damietta, Egypt – email: atarek@horus.edu.eg
[5] Department of Electrical Engineering, Horus University – Egypt, New Damietta, Damietta, Egypt – email: hatem.a.khater@gmail.com

***Abstract*- The trajectory planning for robotic arms is a significant area of research, given its role in facilitating seamless trajectory execution and enhancing movement efficiency and accuracy. This paper focuses on the development of path planning algorithms for a robotic arm with six degrees of freedom. Specifically, three alternative approaches are explored: polynomial (cubic and quantic), Whale Optimization Algorithm (WOA), and Genetic Algorithm (GA). The comparison of outcomes between different methods revealed that polynomial methods were found to be more straightforward to implement, albeit constrained by the intricacy of the pathway. Upon examining the functioning of the WOA, it has been shown that it is well suited for all types of pathways, regardless of their level of complexity. In addition, when GA is applied, it has been shown less smoothness than WOA but also less complexity. In brief, WOA is deemed superior in the path planning process since it is more thorough in determining the optimal path due to the conical spiral path technique it employs in offering optimized path planning. in comparison to GA, WOA is better in implementation speed and accuracy. However, GA is smoother in start and finish path.**

***Keywords*-Path planning; Cubic Polynomial; Quintic polynomial; WOA; GA; KUKA kr4 R600.**

I. Introduction

Path planning is defining an appropriate trajectory or series of configurations for a robotic arm to follow in order to reach its objective while avoiding collisions with impediments. Path planning is critical in robotics as it allows robotic arms to operate efficiently and precisely within their workspace. Because of the complexities of real-world jobs, robots must frequently traverse in a coordinated and obstacle-free manner. This is especially true for robotic arms with six degrees of freedom (6 DOF), which have more movement freedom and flexibility [1].

At its core, path planning for robotic arms involves determining a sequence of configurations that the arm must follow to traverse from a starting point to a desired goal while circumventing obstacles and adhering to kinematic constraints. This task is formidable due to the inherent complexities of real-world environments, which often entail irregular obstacles, limited visibility, and intricate workspace geometries [2].

Path planning for 6 DOF robotic arms has improved recently. Researchers have examined sampling, optimization, and machine learning methods. Rapidly exploring Random Trees (RRT) and its variations, as well as Probabilistic Roadmaps (PRM), can create feasible paths through high-dimensional environments. Optimization techniques solve the planning problem as an optimization, resulting in smooth, energy-efficient journeys. Path planning uses reinforcement learning and neural networks for adaptive and data-driven decision-making [3].

A. Related Work

Several articles have tackled path planning for a six degrees of freedom (DOF) robotic arm. Jiayan Zhang et al. (2018) proposed an improved genetic approach to optimize the time gap between interpolation points in robotic trajectory planning, resulting in smoother movement and shorter running time [4]. Lufeng Luo et al. (2018) presented and applied an energy-optimal and artificial potential field to develop a path planning method for a six-degree-of-freedom (DOF) serial harvesting robot in a dynamic uncertain environment, demonstrating that the proposed path planning algorithm can be applied to the harvesting robot [5]. Loubna Bouhalassa et al. (2020) described a path planning method based on soft computing techniques, which consisted of utilizing a neural network to model the end-effector workspace and then finding the ideal trajectory to reach a desired position [6]. Ali Abdi et al. (2021) proposed a hybrid path planning method with two separate parts: action-finding and angle-finding (passive approach), which significantly improves the slowness and complexity by using simplified agent-environment interaction in the active phase and simple computing of joint angles in the passive phase [7].

The rest of this paper is organized as follows: Section II presents the trajectory planning based on Cubic polynomial, Quantic polynomial. Section III describe optimized techniques include Whale Optimization Algorithm (WOA), and Genetic Algorithm (GA). Section IV simulate the planning results on specific case study. Section V discus results and limitations. Section VI concludes the paper and presents the future work.

Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

II. TRAJECTORY PLANNING

A. Cubic Polynomial

When considering the trajectory of a robotic arm between two known joint positions, it is common to employ polynomial equations, particularly those of higher degrees, to ensure smoothness and enable a broad range of motion in the arm's movement. The cubic path planning method is widely recognized for its effectiveness in controlling movement trajectories [8]–[10]. As implied by its name, this approach involves determining the path based on a third-degree polynomial equation, denoted as (1).

$$q_i(t) = a_{0,i} + a_{1,i}t + a_{2,i}t^2 + a_{3,i}t^3$$

$$\text{Where } i \in \mathbb{N} \wedge i \leq 6 \tag{1}$$

The symbol (i) indicates the joint represented by the planning equation, where each joint has its own path to move from the initial position to the final position which is being modeled as a third order polynomial equation with specific parameters.

Assume that the time period to move from the starting point to the desired point is represented by $[t_0 - t_1]$. Four initial conditions are being used at every joint to find its coefficients ash shown in (2). Knowing the start and end positions, the angle of the joint can be deduced at the beginning and end of time. Moreover, we assume zero velocity at the beginning and end of time interval. [11]–[13]. The velocity equation for each joint I is represented by (3). . By solving equations (1) and (3) subject to the boundary conditions in (2) we get the coefficients values for a joint I which are shown in (4).

$$\begin{cases} q_i(t_0) = q_{start} \\ q_i(t_1) = q_{end} \\ \dot{q}_i(t_0) = 0 \\ \dot{q}_i(t_1) = 0 \end{cases} \tag{2}$$

$$\dot{q}_i(t) = a_{1,i} + 2a_{2,i}t + 3a_{3,i}t^2 \tag{3}$$

$$\begin{cases} a_0 = q_{start} \\ a_1 = 0 \\ a_2 = \frac{3(q_{end} - q_{start})}{t_{final}{}^2} \\ a_3 = \frac{-2(q_{end} - q_{start})}{t_{final}{}^3} \end{cases} \tag{4}$$

Moreover, all these steps by representing the trajectories of all the joints of the robotic arm can be shown in the algorithm shown in Table I.

B. Quantic Polynomial

When considering the manipulation of robotic arms, it is crucial to assess the smoothness of the arm's trajectory to ensure compatibility with the joint's range of motion. Increased smoothness of the path results in enhanced efficiency of movement. In order to improve the trajectory of the robotic arm, it is possible to employ a polynomial function of a degree higher than cubic. This approach provides more flexibility and maneuverability than the cubic polynomial apprach, as demonstrated in (5) where a 5th degree polynomial is utilized.

$$q_i(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \tag{5}$$

Table I. Pseudo code for cubic polynomial path planning

Algorithm 1: Cubic Polynomial Path Planning
Inputs: Start pose : $\forall_{i\in\mathbb{N}, i\leq 6}\ q_i(t_0)$ End pose : $\forall_{i\in\mathbb{N}, i\leq 6}\ q_i(t_1)$ Joint angle constraints : $\forall_{i\in\mathbb{N}, i\leq 6}\ \theta_{i,min}, \theta_{i,max}$ Velocity constraints : $\forall_{i\in\mathbb{N}, i\leq 6}\ \dot{q}_i(t_0)$ and $\dot{q}_i(t_1)$ Number of points to be generated : M **Outputs**: 3rd polynomial trajectory equation for each joint, trajectory, velocity, and acceleration configuration Polynomial coefficients : $\forall_{i\in\mathbb{N}, i\leq 6}\ a_{0,i}, a_{1,i}, a_{2,i}, a_{3,i}$ **1. Generate cubic polynomial coefficients for each joint** $\Delta T = t_1 - t_0$ $\forall_{i\in\mathbb{N}, i\leq 6}$ $a_{0,i} = q_i(t_0)$ $a_{1,i} = 0$ $a_{2,i} = 3(q_i(t_1) - q_i(t_0))/\Delta T^2$ $a_{3,i} = -2(q_i(t_1) - q_i(t_0))/\Delta T^3$ **2. Generate waypoints along the trajectory** trajectory = () $\forall_{i\in\mathbb{N}, i\leq 6}$ Find $\{q_i(t) = a_{0,i} + a_{1,i}t + a_{2,i}t^2 + a_{3,i}t^3\}$ where $t \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \ldots, t_0 + \Delta T\right\}$ Append $\{q_i(t)\}$ into trajectory Find $\{\dot{q}_i(t) = a_{1,i} + 2a_{2,i}t + 3a_{3,i}t^2\}$ Find $\{\ddot{q}_i(t) = 2a_{2,i} + 6a_{3,i}t\}$ **2. Simulate result** $\forall_{i\in\mathbb{N}, i\leq 6}$ $\forall_{t\epsilon T}$ where $T \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \ldots, t_0 + \Delta T\right\}$ Plot $q_i(t)$ Plot $\dot{q}_i(t)$ Plot $\ddot{q}_i(t)$

Assume that the time period to move from the starting point to the desired point is represented by $[t_0 - t_1]$. Now, to solve equation (5), we consider boundary conditions not only from the joint's position and velocity but also acceleration. These boundary conditions are shown in (6) while (7) and (8) illustrate the velocity and acceleration equations, respectively.

$$\begin{cases} q_i(t_0) = q_{start} \\ q_i(t_1) = q_{end} \\ \dot{q}_i(t_0) = 0 \\ \dot{q}_i(t_1) = 0 \\ \ddot{q}_i(t_0) = 0 \\ \ddot{q}_i(t_1) = 0 \end{cases} \tag{6}$$

Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

$$\dot{q}_i(t) = a_1 + 2a_2t^1 + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \quad (7)$$

$$\ddot{q}_i(t) = 2a_2 + 6a_3t^1 + 12a_4t^2 + 20a_5t^3 \quad (8)$$

For each joint i, substituting the known initial and end conditions into (5 & 7 & 8) gives the coefficients as in (9):

$$\begin{cases} a_0 = q_{start} \\ a_1 = 0 \\ a_2 = 0 \\ a_3 = \dfrac{10(q_{end} - q_{start})}{t_{final}{}^3} \\ a_4 = \dfrac{-15(q_{end} - q_{start})}{t_{final}{}^4} \\ a_5 = \dfrac{6(q_{end} - q_{start})}{t_{final}{}^5} \end{cases} \quad (9)$$

In addition, all these phases are listed in the algorithm that is shown in Table II by calculating the trajectories of all of the joints that make up the robotic arm.

III. Optimized Trajectory Planning

Path planning optimization is important for robotic arm efficiency and other factors such as total trip time, accuracy, energy efficiency and motion smoothness. Robotic arms with optimized path planning finish tasks faster. In manufacturing, shorter cycle durations boost productivity. Accuracy: A well-planned path helps the robotic arm approach its target precisely. Surgery requires accuracy, thus this is crucial. Energy Efficiency: Optimized pathways reduce robotic arm energy use. Avoiding needless movements and optimizing joint trajectories reduces energy expenses. Path planning optimization avoids robot collisions by considering environmental impediments. This is essential to protect the robot and its surroundings. Smooth Motion: Smooth trajectories reduce sudden jerks and vibrations, extending the robotic arm's lifespan and improving its performance. Complex Tasks: Maintaining the end-effector's orientation or avoiding singularities in the robot's configuration space are complex limitations. Optimization helps solve these issues [14], [15].

Path planning optimization methods include search-based algorithms. These algorithms such as A*, RRT, and PRM search configuration space for collision-free paths . Optimization approaches formulate the problem as an optimization challenge to discover optimal paths. Gradient-based optimization, Whale Optimization Algorithm (WOA), genetic algorithms, and simulated annealing are examples of the optimization approaches that could work to achieve the path planning for robotic arms. Also, heuristic techniques are being used in real-time applications to achieve the same goal by sampling the configuration space randomly to generate a path between the start and end points. Furthermore, reinforcement learning and neural networks can be used to learn and optimize robot motion based on prior experiences and simulations [16]. In this paper, two optimization approaches were applied to solve the path planning problem on a robotic arm with 6 DoF. These algorithms are Whale Optimization Algorithm (WOA) shown in section A and Genetic Algorithm presented in section B.

Table II. Pseudo code for Quantic polynomial path planning

Algorithm 2: Quantic Polynomial Path Planning
Inputs: Start pose : $\forall_{i\in\mathbb{N}, i\le 6}\ q_i(t_0)$ End pose : $\forall_{i\in\mathbb{N}, i\le 6}\ q_i(t_1)$ Joint angle constraints : $\forall_{i\in\mathbb{N}, i\le 6}\ \theta_{i,min}, \theta_{i,max}$ Velocity constraints : $\forall_{i\in\mathbb{N}, i\le 6}\ \dot{q}_i(t_0)$ and $\dot{q}_i(t_1)$ Acceleration constraints : $\forall_{i\in\mathbb{N}, i\le 6}\ \ddot{q}_i(t_0)$ and $\ddot{q}_i(t_1)$ Number of points to be generated : M **Outputs**: 5th polynomial trajectory equation for each joint, trajectory, velocity, and acceleration configuration Polynomial coefficients : $\forall_{i\in\mathbb{N}, i\le 6}\ a_{0,i},\ a_{1,i}, a_{2,i}, a_{3,i}, a_{4,i}, a_{5,i}$ **1. Generate cubic polynomial coefficients for each joint** $\Delta T = t_1 - t_0$ $\forall_{i\in\mathbb{N}, i\le 6}$ $a_{0,i} = q_i(t_0)$ $a_{1,i} = 0$ $a_{2,i} = 0$ $a_{3,i} = 10(q_i(t_1) - q_i(t_0))/\Delta T^3$ $a_{4,i} = -15(q_i(t_1) - q_i(t_0))/\Delta T^4$ $a_{5,i} = 6(q_i(t_1) - q_i(t_0))/\Delta T^5$ **2. Generate waypoints along the trajectory** trajectory = () $\forall_{i\in\mathbb{N}, i\le 6}$ **Find** $\{q_i(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5\}$ where $t \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \ldots, t_0 + \Delta T\right\}$ **Append** $\{q_i(t)\}$ into trajectory **Find** $\{\dot{q}_i(t) = a_1 + 2a_2t^1 + 3a_3t^2 + 4a_4t^3 + 5a_5t^4\}$ **Find** $\{\ddot{q}_i(t) = 2a_2 + 6a_3t^1 + 12a_4t^2 + 20a_5t^3\}$ **3. Simulate results** $\forall_{i\in\mathbb{N}, i\le 6}$ $\forall_{t\in T}$ where $T \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \ldots, t_0 + \Delta T\right\}$ **Plot** $q_i(t), \dot{q}_i(t), \ddot{q}_i(t)$

A. *Whale Optimization Algorithm*

The Whale Optimization Algorithm (WOA) is inspired by humpback whale behavior. The algorithm simulates cetacean hunting to find the best optimization methods. WOA suitable in many optimization problems, especially in robotics systems. [17]. WOA consist of main stages as optimization strategy as shown in Table III, which describe the pseudo code of whole algorithm [18].

The Whale Optimization Algorithm (WOA) is a metaheuristic optimization algorithm inspired by the bubble-

Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

net feeding behavior of humpback whales. It uses three main operators: encircling, luring, and shrinking. The encircling operator mimics how whales encircle their prey before attacking. The luring operator imitates how whales use songs to attract prey. The shrinking operator gradually reduces the search space to focus on promising solutions. The WOA algorithm is effective for optimization problems like function optimization, machine learning, and engineering design. It has advantages like being easy to implement and not requiring gradient information. The WOA algorithm can be used for robot path planning by finding paths that minimize distance, avoid obstacles, minimize energy consumption, and minimize time to destination. The operators help achieve these goals. Though promising, the WOA algorithm's performance for path planning can be improved by tuning.

Table III. Pseudo code for WOA path planning

Algorithm 3: Whale Optimization Algorithm Path Planning

Inputs:
Start pose for each joint i : $\forall_{i\in\mathbb{N},i\le 6}\ q_i(t_0)$
End pose for each joint i : $\forall_{i\in\mathbb{N},i\le 6}\ q_i(t_1)$
Joint angle constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \theta_{i,min}, \theta_{i,max}$
Joint velocity constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \dot{q}_i(t_0)$ and $\dot{q}_i(t_1)$
Joint Acceleration constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \ddot{q}_i(t_0)$ and $\ddot{q}_i(t_1)$
Number of points to be generated : M
WOA parameters:
Number of whales : N_W
Number of iterations : N
Poly Coefficients lower boundary: $a_{(j,i),low}$ where $0 \le j \le 5$
Poly Coefficients higher boundary: $a_{(j,i),high}$ where $0 \le j \le 5$

Output: 5th polynomial trajectory equation for each joint, trajectory (desired and optimized) configuration.
Polynomial coefficients : $\forall_{i\in\mathbb{N},i\le 6}\ a_{(0,i)}, a_{(1,i)}, a_{(2,i)}, a_{(3,i)}, a_{(4,i)}, a_{(5,i)}$

1.
Initialize a matrix W of size $N_w \times 6$ where
$W[j,i] = a_{(j,i),low} + \text{rand} \times \left(a_{(j,i),high} - a_{(j,i),low}\right)\ ; i \le N_w, j \le 5$
2.
$\forall_{i\in\mathbb{N},i\le 6}\quad \forall_{t\in\mathbb{R}^+,t\in T}$
Find $q_{i,t} = poly\_fit\ [q_i(t_0)\ , q_i(t_1)]$
Where $T = \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$
And $\Delta T = t_1 - t_0$
3.
$\forall_{i\in\mathbb{N},i\le 6}\quad \forall_{1\le j\le |T|}$
Find $\widehat{q_{j,t}} = poly\_fit\ [a_{(0,i)}, a_{(1,i)}, a_{(2,i)}, a_{(3,i)}, a_{(4,i)}, a_{(5,i)}]$
Where $T = \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$
and $0 \le j \le 5$
4.
$\forall_{i\in\{1,2,\dots,6\}\wedge j\in\{0,1,\dots,5\}}$
Find fitness parameter $f(j,i) = \text{norm}(\widehat{q_{j,t}} - q_{i,t})$
5.
$\forall_{1\le i\le 6, 1\le k\le N, 1\le w\le N_W}$
Find best fitness $\boldsymbol{f_i^*}$ and best whale $\boldsymbol{W_i^*}$
after substituting with $W[j,i]$ in $f(j,i)$
6.
$\forall_{i\in\mathbb{N},i\le 6}$
$\forall_{t\in T}$ where $T \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$
Plot $q_i(t), \dot{q}_i(t), \ddot{q}_i(t)$

B. Genetic Algorithm

Genetic-based algorithm GA is an optimization algorithm that mimics biological evolution via crossover and mutation [19].

Crossover creates a new agent (child) by randomly selecting two agents from the population (parents) and creating the child agent's parameters from the parent agents' parameters. Crossover involves recombination to find the best solution by creating new agents and keeping the best-fitting ones in the population. To replace a poorly fitted gene in the population, the crossover should produce a better gene. If the new agent is inferior, the gene is maintained.

Mutation replaces an agent in the population by randomly selecting its parameters from the solution space without regard to fitness function values. A poorer mutated agent can replace the better one. Only using the crossover mechanism can lead to stagnation, hence the mutation mechanism is needed. It lets the GA discover the global optimum instead of the local optimum where all agents are. Crossover is employed 80% of the time, while mutation is used 1% [20].

The genetic algorithm (GA) can be used to solve path planning optimization problems for robotic arms. It works by generating an initial population of random paths that satisfy the constraints. Each path is evaluated based on fitness criteria like path length and energy consumption. The best paths are selected using selection methods and then crossed over and mutated to generate new paths. This process is repeated until a satisfactory path is found. The genetic algorithm is a robust algorithm and it can handle a variety of constraints.

GA uses in robotic arm path planning consist of main stages [13] as shown in Table IV, which describe the pseudo code of whole algorithm [21].

Table IV. Pseudo code for GA path planning

Algorithm 4: Genetic Algorithm Path Planning

Inputs:
Start pose for each joint i : $\forall_{i\in\mathbb{N},i\le 6}\ q_i(t_0)$
End pose for each joint i : $\forall_{i\in\mathbb{N},i\le 6}\ q_i(t_1)$
Joint angle constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \theta_{i,min}, \theta_{i,max}$
Joint velocity constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \dot{q}_i(t_0)$ and $\dot{q}_i(t_1)$
Joint Acceleration constraints : $\forall_{i\in\mathbb{N},i\le 6}\ \ddot{q}_i(t_0)$ and $\ddot{q}_i(t_1)$
Number of points to be generated : M
WOA parameters:
Number of generation : N_g
Number of iterations : N
Poly Coefficients lower boundary: $a_{(j,i),low}$ where $0 \le j \le 5$
Poly Coefficients higher boundary: $a_{(j,i),high}$ where $0 \le j \le 5$

Output: 5th polynomial trajectory equation for each joint, trajectory (desired and optimized) configuration.
Polynomial coefficients : $\forall_{i\in\mathbb{N},i\le 6}\ a_{(0,i)}, a_{(1,i)}, a_{(2,i)}, a_{(3,i)}, a_{(4,i)}, a_{(5,i)}$

1.
Initialize a matrix P of size $N_w \times 6$ where
$G[j,i] = a_{(j,i),low} + \text{rand} \times \left(a_{(j,i),high} - a_{(j,i),low}\right)\ ; i \le N_g, j \le 5$
2.
$\forall_{i\in\mathbb{N},i\le 6}\quad \forall_{t\in\mathbb{R}^+,t\in T}$
Find $q_{i,t} = poly\_fit\ [q_i(t_0)\ , q_i(t_1)]$
Where $T = \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$

Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

And $\Delta T = t_1 - t_0$

3.
$\forall_{i\in\mathbb{N}, i\le 6}$ $\forall_{1\le j\le |T|}$
Find $\widehat{q_{j,t}} = poly\_fit\ [a_{(0,i)}, a_{(1,i)}, a_{(2,i)}, a_{(3,i)}, a_{(4,i)}, a_{(5,i)}]$

Where $T = \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$
and $0 \le j \le 5$

4.
$\forall_{i\in\{1,2,\dots,6\}\wedge j\in\{0,1,\dots,5\}}$
Find fitness parameter $f(j,i) = \text{norm}(\widehat{q_{j,t}} - q_{i,t})$

5.
$\forall_{1\le i\le 6, 1\le k\le N, 1\le w\le N_w}$
Find best fitness f_i^* and best generation G_i^*
after substituting with $G[j,i]$ in $f(j,i)$

6.
$\forall_{i\in\mathbb{N}, i\le 6}$
$\forall_{t\in T}$ where $T \in \left\{t_0, t_0 + \frac{\Delta T}{M}, t_0 + 2\frac{\Delta T}{M}, \dots, t_0 + \Delta T\right\}$
Plot $q_i(t), \dot{q}_i(t), \ddot{q}_i(t)$

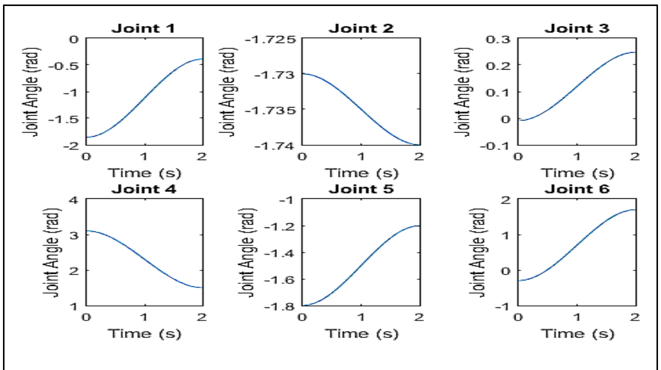
IV. SIMULATE ALGORITHMS AS CASE STUDY

All the previous algorithms were applied to the KUKA KR4R600 arm, Fig. 1. It can be defined by defining its DH coefficients that were deduced through kinematic analysis. Table IV describe the lengths of the arm's links.

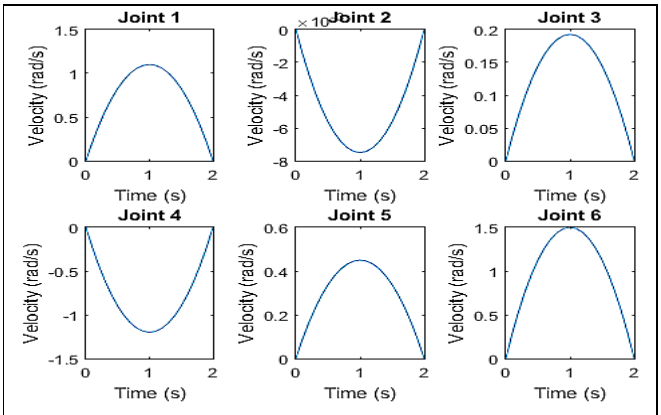
Figure 1. KUKA kr4 R600 configuration

Table IV. DH parameters of KUKA kr4 R600

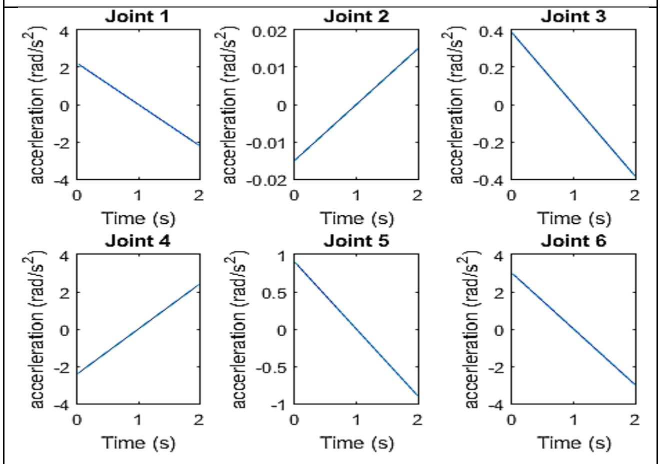
links	DH parameter			
	a (m)	α (rad)	d (m)	θ (rad)
1	0	1.57	0.330	q_1
2	0.29	0	0	q_1
3	0.02	1.57	0	q_1
4	0	-1.57	0.310	q_1
5	0	1.57	0	q_1
6	0	0	0.075	q_1



a) joint angles through path for each joint



b) Velocities through path for each joint



c) Accelerations through path for each joint

Figure 2. Cubic path planning configurations

When applying kinematics analysis to the robotic arm and determining the start point of the path by default and its end point as well. By ensuring that the movement limits of the six joints of the robot are appropriate for the ability to reach both points. So that, it is possible to deduce the angles of the joints necessary for the end effector to reach the end point from the start point to the end point (10 and 11).

$$start\ angles = [-1.856, -1.73, -0.009, 3.1, -1.8, -0.30] \quad (10)$$

$$end\ angles = [-0.388, -1.74, 0.248, 1.50, -1.30, 1.7] \quad (11)$$

The motion planning of the manipulator from the initial point to the desired point is achieved using the cubic polynomial interpolation and the quintic polynomial interpolation methods, respectively. The curve representing the relationship between angle, velocity, and acceleration is also derived. Both approaches have a zero velocity and acceleration -for quantic polynomial- at both the starting and target points.

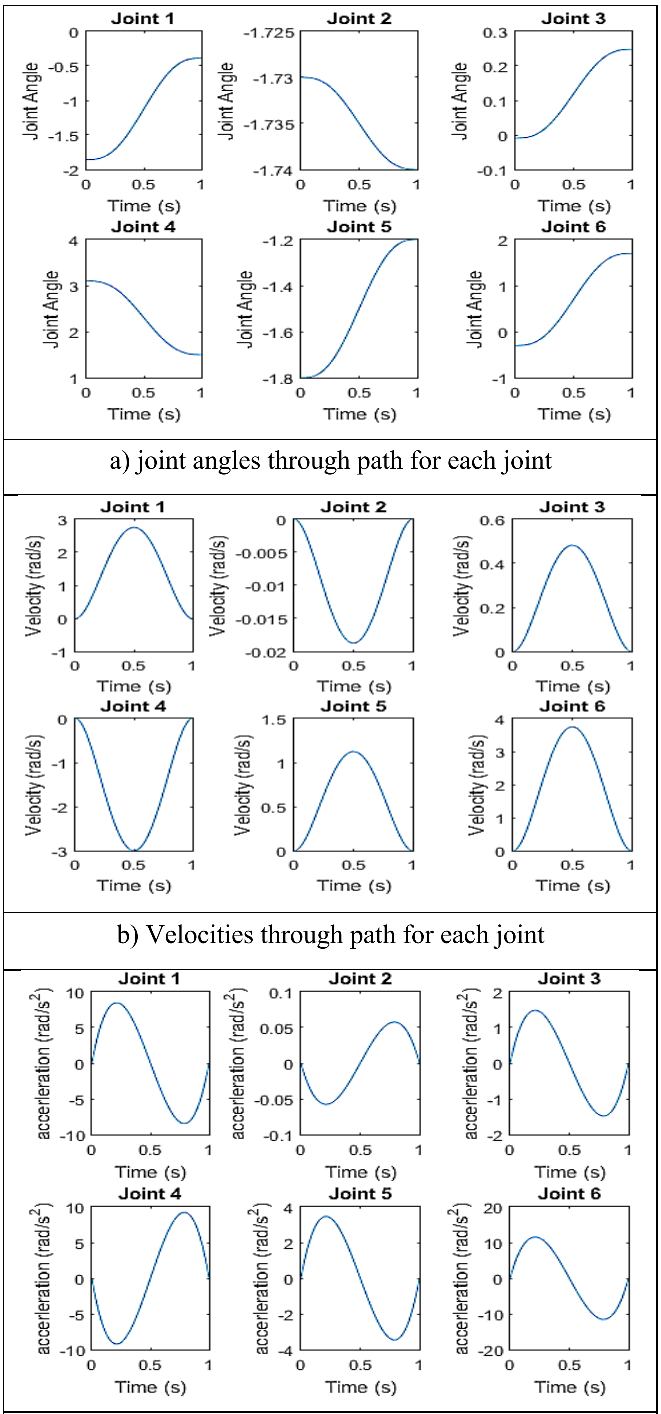
Fig 2 displays the curves of angular displacement, velocity, and acceleration for each joint, which were derived using cubic polynomials. The figure illustrates the continuity of the

Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

angular displacement function, angular velocity function, and angular acceleration function for each joint. Consequently, the robot exhibits a seamless movement during this particular motion.

Fig 3 illustrates that the joint curves generated by the quintic polynomial exhibit a higher degree of smoothness compared to those generated by cubic polynomials. Additionally, the joint acceleration curves exhibit curvature rather than linearity, resulting in a more favorable outcome.



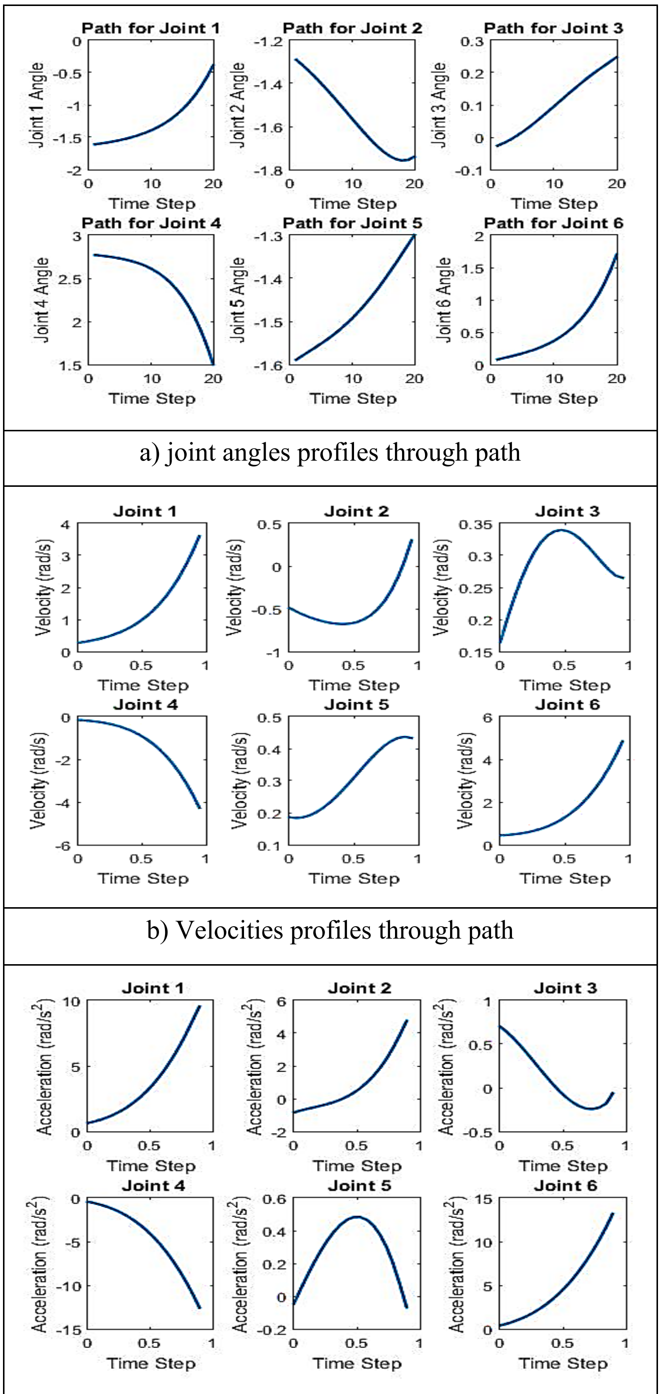
a) joint angles through path for each joint

b) Velocities through path for each joint

c) Accelerations through path for each joint

Figure 3. Quintic path planning configurations

When creating an integrated program for the WOA that includes all the necessary functions, (function of interpolate joint angles, function of generate joint angles, and function of evaluate fitness) as in algorithm 3, the path of each of the six joints of the robotic arm can be represented in (a), also velocity profiles in (b), and acceleration profiles in (c) all in Fig 4. Here, it can be noted the planned path is suitable with unstable environment not fixed as in polynomial planning.



a) joint angles profiles through path

b) Velocities profiles through path

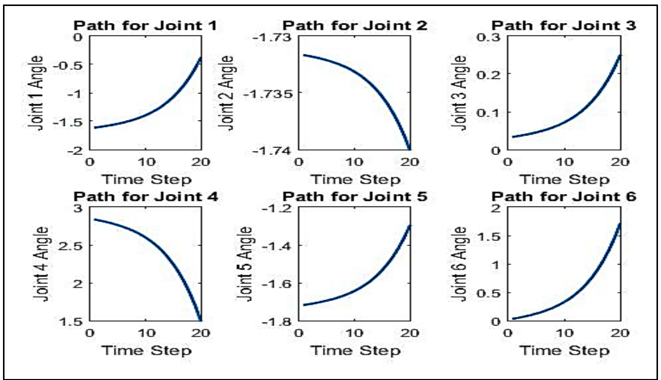
c) Accelerations profiles through path

Figure 4. WOA Joint angles, velocities, and accelerations configurations through path

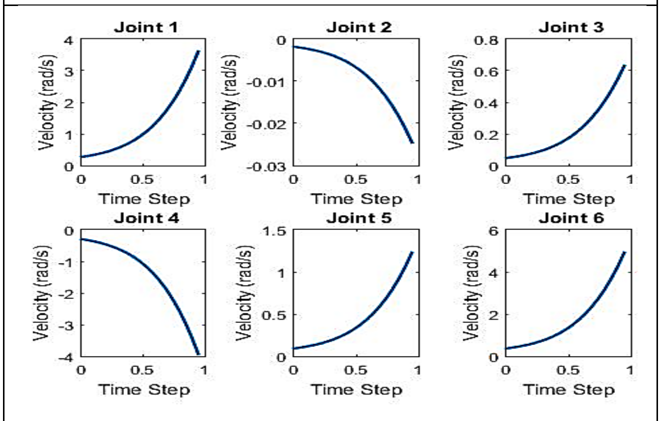
Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

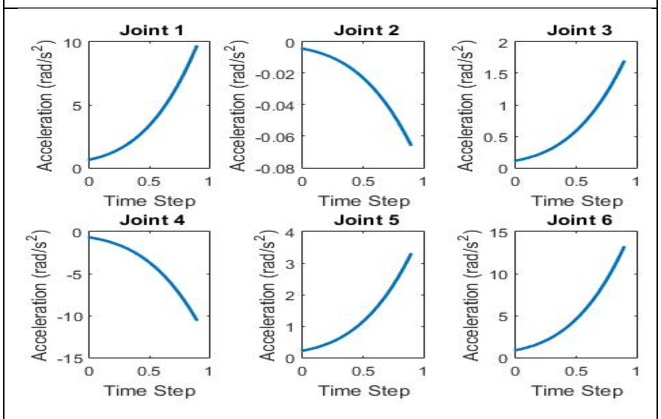
When designing a comprehensive integrated program for the GA, as in algorithm 4, it is essential to include all the requisite functions, such as the path calculation function and the fitness calculation function. In Fig 5, the path of each of the six joints of the robotic arm can be depicted in (a), while the velocity profiles can be represented in (b), and the acceleration profiles can be illustrated in (c). also, in those figures, the smoothness of planned path is less than in WOA. But on the other hand, it is more suitable and adaptive technique to unstable path than polynomial planning.



a) joint angles profiles through path



b) Velocities profiles through path



c) Accelerations profiles through path

Figure 5. GA Joint angles, velocities, and accelerations configurations through path

V. Results Discussion

For robotic arm path planning, this study compares nature-inspired optimization with the WOA, human-inspired optimization with GA, and mathematical curve fitting using polynomial approaches. Based on its route analysis technique, the WOA may be better at robotic arm path planning than other methods.

When the arm is restricted to suboptimal pathways, polynomial path planning may result in constrained optima. WOA's exploration and exploitation capabilities enable it to avoid local optima, resulting in better global solutions.

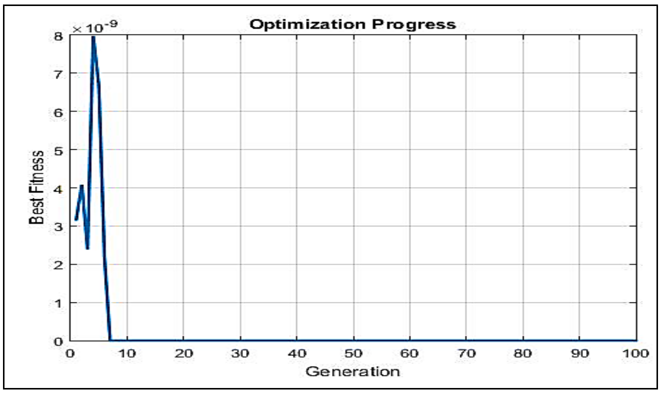
In many instances, adaptability and flexibility are essential. Polynomials are mathematical functions that represent trajectories. The WOA and GA, on the other hand, may tailor the search methodology to the problem at hand, increasing its adaptability in path planning settings.

In multi-objective optimization, WOA and GA use to optimize time efficiency, energy consumption, and collision avoidance. In practice, polynomials may struggle to balance various goals.

Polynomial approaches can be sensitive to initial conditions, producing different results for slightly varied initial trajectories. WOA's stochasticity and exploratory processes reduce its sensitivity to initial conditions. To alleviate collision avoidance constraints, the WOA and GA algorithms intelligently reject infeasible pathways. Restriction implementation in polynomial methods may be more challenging and require more explanation.

Polynomial algorithms, on the other hand, are simple and efficient when applied to well-defined trajectories under controlled conditions. The needs and complexities of robotic arm path planning should determine whether adaptive or polynomial algorithms are used. Combining both approaches may be a useful strategy for maximizing their benefits.

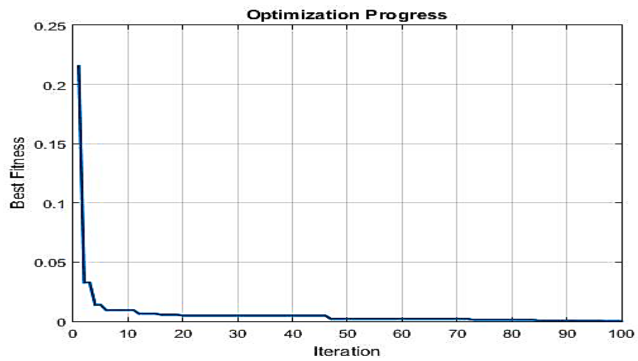
It can now be observed that there is a close similarity between the GA and the WOA, as it was discovered that both are good at dealing with path planning, particularly unpaved paths. The performance of both algorithms in the optimization process can be compared. Looking at Fig 6 (a and b), it can be seen that whale algorithm is more stable in the process of searching for the best results and faster in implementing the algorithm, but they are close when looking at the lower value of the fitting. GA, on the other hand, got off to a better start than WOA.



Doi: 10.21608/erjeng.2023.237586.1256

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
https://erjeng.journals.ekb.eg/
ISSN: 2356-9441 e ISSN: 2735-4873

(a) optimization progress of GA



(a) optimization progress of WOA

Figure 6. GA and WOA progress

The choice between cubic polynomial, quintic polynomial, WOA, or GA depends on the specific context and requirements of the path-planning problem. For simple paths in controlled environments, polynomial methods offer efficiency and simplicity. However, as the complexity of the environment, constraints, and objectives increase, WOA emerges as a robust contender. WOA's nature-inspired optimization enables it to tackle real-world challenges, making it a strong candidate for scenarios involving dynamic obstacles, changing environments, and multi-objective optimization.

VI. Conclusion

The field of trajectory planning for robotic arms is of great importance in research, as it plays a crucial role in enabling smooth execution of trajectories and improving the efficiency and precision of movements. The primary objective of this study was to investigate and analyze the progression of path planning algorithms specifically designed for a robotic arm possessing six degrees of freedom. This study examined three distinct methodologies, namely polynomial (cubic and quantic), WOA, and GA, as potential alternative techniques. The analysis of results across various methodologies indicated that the implementation of polynomial approaches was comparatively more straightforward, however it was limited by the complexity of the pathway. After conducting an analysis of the operational mechanisms of the WOA, it has been demonstrated that this approach is very compatible with many types of routes, irrespective of their degree of intricacy. Furthermore, empirical evidence suggests that the use of GA exhibits a lower level of smoothness compared to WOA, while simultaneously demonstrating reduced complexity. Nevertheless, the presence of compatibility between the two methods also necessitates a proportional increase in algorithmic complexity to effectively address the diverse constraints imposed by the path in both the WOA and the GA.

Acknowledgment

We thank Horus University (Egypt) for its explicit support throughout this research and for providing access to KUKA robotic arm which was used in our case study.

Software: MATLAB software was used as a platform for modeling the robotic arm and implementing the proposed algorithms.

Author Contributions: All authors contributed to the research, development, and testing of the proposed techniques. Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater wrote the manuscript. Abdelrahman T. Elgohr and Hatem A. Khater leaded the mathematical modeling for the robotic arm. Mahmoud A. A. Mousa was responsible for developing the proposed optimization techniques and its implementation was done by Abdelrahman T. Elgohr. The testing was performed by Abdelrahman T. Elgohr and analyzed by Mahmoud A. A. Mousa and Hatem A. Khater.

Funding: Not Applicable

Declarations: Ethics approval and consent to participate: All applicable institutional and national guidelines were followed.

Competing interests: The authors declare that they have no conflict of interest.

References

[1] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, “Path Planning and Trajectory Planning Algorithms: A General Overview,” *Mechanisms and Machine Science*, vol. 29, pp. 3–27, Mar. 2015, doi: 10.1007/978-3-319-14705-5_1.

[2] M. Ali and M. Mailah, “Path Planning and Control of Mobile Robot in Road Environments Using Sensor Fusion and Active Force Control,” *IEEE Trans Veh Technol*, vol. PP, p. 1, Jan. 2019, doi: 10.1109/TVT.2019.2893878.

[3] J. Yi, Q. Yuan, R. Sun, and H. Bai, “Path planning of a manipulator based on an improved P_RRT* algorithm,” *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2227–2245, 2022, doi: 10.1007/s40747-021-00628-y.

[4] J. Zhang, Q. Meng, X. Feng, and H. Shen, “A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm,” *Robotics Biomim*, vol. 5, no. 1, p. 3, Dec. 2018, doi: 10.1186/s40638-018-0085-7.

[5] L. Luo *et al.*, “Collision-Free Path-Planning for Six-DOF Serial Harvesting Robot Based on Energy Optimal and Artificial Potential Field,” *Complexity*,

Journal of Engineering Research (ERJ)
Vol. 7 – No. 5, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

vol. 2018, pp. 1–12, Nov. 2018, doi: 10.1155/2018/3563846.

[6] L. Bouhalassa, L. Benchikh, Z. Ahmed-Foitih, and K. Bouzgou, “Path Planning of the Manipulator Arm FANUC Based on Soft Computing Techniques,” *International Review of Automatic Control (IREACO)*, vol. 13, no. 4, p. 171, Jul. 2020, doi: 10.15866/ireaco.v13i4.18506.

[7] A. Abdi, D. Adhikari, and J. H. Park, “A Novel Hybrid Path Planning Method Based on Q-Learning and Neural Network for Robot Arm,” *Applied Sciences*, vol. 11, no. 15, p. 6770, Jul. 2021, doi: 10.3390/app11156770.

[8] J. Sudharsan and L. Karunamoorthy, “Path planning and co-simulation control of 8 DOF anthropomorphic robotic arm,” *International Journal of Simulation Modelling*, vol. 15, no. 2, pp. 302–312, 2016, doi: 10.2507/IJSIMM15(2)9.339.

[9] A. Abdullah Farid, G. Selim, and H. Khater, *Applying Artificial Intelligence Techniques to Improve Clinical Diagnosis of Alzheimer's Disease*. 2020.

[10] M. Alhamdany, “PROPOSED APPROACH FOR AUTOMATIC UNDERWATER OBJECT CLASSIFICATION,” *ICIC Express Letters*, vol. 12, pp. 1205–1212, Sep. 2018.

[11] S. Fang, X. Ma, Y. Zhao, Q. Zhang, and Y. Li, “Trajectory Planning for Seven-DOF Robotic Arm Based on Quintic Polynormial,” in *Proceedings - 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2019*, Institute of Electrical and Electronics Engineers Inc., Aug. 2019, pp. 198–201. doi: 10.1109/IHMSC.2019.10142.

[12] W. Abdelmoez, H. Khater, and N. El-shoafy, “Comparing maintainability evolution of object-oriented and aspect-oriented software product lines,” in *2012 8th International Conference on Informatics and Systems (INFOS)*, 2012, p. SE-53-SE-60.

[13] H. Khater, S. Mesbah, and A. Anwar, “Enhanced Navigation System for AUV Using Mobile Application,” 2015. [Online]. Available: www.ijeijournal.com

[14] F. G. Flores and A. Kecskeméthy, “Time-Optimal Path Planning Along Specified Trajectories,” in *Multibody System Dynamics, Robotics and Control*, H. Gattringer and J. Gerstmayr, Eds., Vienna: Springer Vienna, 2013, pp. 1–16.

[15] A. Abdullah Farid, “Applying Artificial Intelligence Techniques for Prediction of Neurodegenerative Disorders: A Comparative Case-Study on Clinical Tests and Neuroimaging Tests with Alzheimer's Disease,” in *Proceedings of The 2nd International Conference on Advanced Research in Applied Science and Engineering*, GLOBALKS, Mar. 2020. doi: 10.33422/2nd.rase.2020.03.101.

[16] R. R. Santos, D. A. Rade, and I. M. da Fonseca, “A machine learning strategy for optimal path planning of space robotic manipulator in on-orbit servicing,” *Acta Astronaut*, vol. 191, pp. 41–54, 2022, doi: https://doi.org/10.1016/j.actaastro.2021.10.031.

[17] A. Chhillar and A. Choudhary, “Mobile Robot Path Planning Based Upon Updated Whale Optimization Algorithm,” in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2020, pp. 684–691. doi: 10.1109/Confluence47617.2020.9058323.

[18] Y. Dai, J. Yu, C. Zhang, B. Zhan, and X. Zheng, “A novel whale optimization algorithm of path planning strategy for mobile robots,” *Applied Intelligence*, vol. 53, no. 9, pp. 10843–10857, 2023, doi: 10.1007/s10489-022-04030-0.

[19] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[20] S. M. Lim, A. B. M. Sultan, M. N. Sulaiman, A. Mustapha, and K. Y. Leong, “Crossover and mutation operators of genetic algorithms,” *Int J Mach Learn Comput*, vol. 7, no. 1, pp. 9–12, Feb. 2017, doi: 10.18178/ijmlc.2017.7.1.611.

[21] S. Baressi Šegota, N. Anđelić, I. Lorencin, M. Saga, and Z. Car, “Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms,” *Int J Adv Robot Syst*, vol. 17, no. 2, p. 1729881420908076, Mar. 2020, doi: 10.1177/1729881420908076.

Doi: 10.21608/erjeng.2023.237586.1256

