Annals of Emerging Technologies in Computing (AETiC)
Vol. 8, No. 1, 2024

Research Article

Trajectory Optimization for a 6 DOF Robotic Arm Based on Reachability Time

Mahmoud A. A. Mousa[1,2,*], Abdelrahman T. Elgohr[1,3] and Hatem A. Khater[3]

[1]Zagazig University, Zagazig, Egypt
mamosa@zu.edu.eg
[2]Heriot Watt University, Dubai, UAE
m.mousa@hw.ac.uk
[3]Horus University – Egypt, Damietta, Egypt
atarek@horus.edu.eg; hatem.a.khater@gmail.com
*Correspondence: mamosa@zu.edu.eg

Received: 9th December 2022; Accepted: 25th December 2023; Published: 1st January 2024

Abstract: The design of the robotic arm's trajectory is based on inverse kinematics problem solving, with additional refinements of certain criteria. One common design issue is the trajectory optimization of the robotic arm. Due to the difficulty of the work in the past, many of the suggested ways only resulted in a marginal improvement. This paper introduces two approaches to solve the problem of achieving robotic arm trajectory control while maintaining the minimum reachability time. These two approaches are based on rule-based optimization and a genetic algorithm. The way we addressed the problem here is based on the robot's forward and inverse kinematics and takes into account the minimization of operating time throughout the operation cycle. The proposed techniques were validated, and all recommended criteria were compared on the trajectory optimization of the KUKA KR 4 R600 six-degree-of-freedom robot. As a conclusion, the genetic based algorithm behaves better than the rule-based one in terms of achieving a minimal trip time. We found that solutions generated by the Genetic based algorithm are approximately 3 times faster than the other solutions generated by the rule-based algorithm to the same paths. We argue that as the rule-based algorithm produces its solutions after discovering all the problem's searching space which is time consuming, and it is not the case as per the genetic based algorithm.

Keywords: ***Forward Kinematics; Genetic Algorithm; Inverse Kinematics; Rule-Based Optimization; Trajectory Control; Trajectory Optimization; 6 DOF Robotic Arm***

1. Introduction

Robotics is a unique technical discipline concerned with the design, modelling, control, and application of robots. Today, robots assist people in their daily activities and take control of their daily routines. Robots are used in a wide range of applications, from toys to industrial robots and offices, to the very advanced technology necessary for space exploration [1].

Robotic arms are one of the most common types of robots in use today, from manufacturing to automotive to agriculture to laboratory purposes. One of the primary benefits of robotic arms is their adaptability to a wide range of applications, from simple to complex tasks. Material handling is one of these applications in which robotic arms can help to create a safe and efficient warehouse by ensuring goods and materials are properly stored, easy to find, or transported correctly. Pick-and-place robots are commonly used in modern manufacturing and logistics. Moreover, arm robots are being used in quality inspection at the end of production lines[1], [2].

Trajectory optimization is an important approach which determines the best path for the robot based on its kinematic and dynamic motion limitations [2]. Trajectory planning is a critical subject in robotics and automation in general. The capacity to produce trajectories with specified characteristics is critical for

Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater, "Trajectory Optimization for a 6 DOF Robotic Arm Based on Reachability Time", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 22-35, Vol. 8, No. 1, 1st January 2024, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2024.01.003, Available: http://aetic.theiaer.org/archive/v8/v8n1/p3.html.

achieving substantial outcomes in terms of quality and convenience of accomplishing the needed motion, particularly at the high operating speeds required in many applications. Optimal time trajectory arranging is the most commonly mentioned optimization approach in the literature, owing to its primary relationship with manufacturing time deduction and productivity gain. There are various forms for trajectory optimization problem in robotics. Furthermore, optimizing the trajectory by decreasing the energy usage is one of the most challenging problems which offers various economic benefits. This technique, on the other hand, conserves energy, which is important not only in terms of cost but also of quantitative constraint. Another minimization problem that could be considered in robotic arm's trajectory control is to create smooth, easy-to-follow trajectories and minimize mechanical stresses in actuators and the robot's frame. Jerk reduction also provides trajectory continuity, which leads to a decrease in joint positioning errors or more correctly tracking trajectory on the one hand, and a reduction in robot vibration limitations on the other, which leads to a reduction in wear and hence an increase in exploitation duration [3]. The difficulty of determining an ideal trajectory for a given route is then examined, along with several noteworthy solutions [4]. To solve the trajectory planning issue, reference inputs should be given to the robot's control system to assure that the required motion is carried out. The route defined by the path planner, as well as the robot's kinematic and dynamic limitations, are usually inputs to the algorithm used for trajectory planning. The trajectory of the joints, or the end-effector, is output by the trajectory-planning module as a series of position, velocity and acceleration numbers [5].

Artificial intelligence (AI) is an advanced science of computer science concerned with emblematic, non-algorithmic problem resolving methodologies, such as heuristic concepts to efficiently solve complex problems. One of its ways to overcome a certain problem is to build an expert system based on previous experience. A previous experience base stores both declarative knowledge (information about things, events, and situations) and procedural knowledge (data about courses of action). One of the most frequent previous experience rendition methodologies used in expert systems is the rule-based system approach. Procedural information is integrated with declarative knowledge in the form of 'if-then' rules in a rule-based system [6].

Genetic algorithm (GA) is a metaheuristic approach categorized as an evolutionary algorithm since it is based on the natural selection process (EA). Selection, crossover, and mutation are some of the bio-oriented operators utilized in genetic algorithms to deliver high-quality answers to search and optimization problems. There are a few downsides to the GA, however, and they need addressing in order to obtain better control over the population's exploitation and exploration and randomization involvement in the population during solution initialization. A perfect solution cannot be reached since the mutation is pressed onto the new chromosomes. A competitive selection process is used in each iteration of a GA to remove bad ideas. GA is one of the most stable and dependable optimization techniques since it does not need gradient information or an initial assumption [7].

The rest of this paper is organized as follows: Section 2 presents the kinematics analysis. Section 3 demonstrates the trajectory optimization using Rule-based techniques and Genetic Algorithm. Section 4 shows the proposed optimization techniques while Section 5 elaborates how these techniques are applied on a study case and demonstrates the tests and results for it. Section 6 concludes the paper and presents the future work.

2. Kinematics Analysis

A robotic arm is made of a series of links that are linked together in either a serial or a parallel fashion. The position and orientation analysis of a robotic arm is a critical step on its design and control. The Forward Kinematics (FK) analysis is used to analyse the model and calculate the position of the end-effector using the joint angles and other joint parameters. The reverse process of the Forward Kinematics is the Inverse Kinematics (IK) where the desired end-effector's position is known, and the challenge part is to find the joint angles to achieve the end-effector's position [8].

2.1. Forward Kinematics

Because the complexity of IK increases as the Degree of Freedom (DOF) increases, the Denavit & Hartenberg convention and transformation type solution are used to analyze it. Denavit-Hartenberg

parameters (also known as DH parameters) are four parameters related with a specific protocol for connecting reference frames to the links of a spatial kinematic chain or robot manipulator. Figure 1 [9] depicts a pair of adjacent links, link (i-1) and link i, as well as their associated joints, joints (i-1), i, and (i+1), and axes (i-2), (i-1) and i [9].

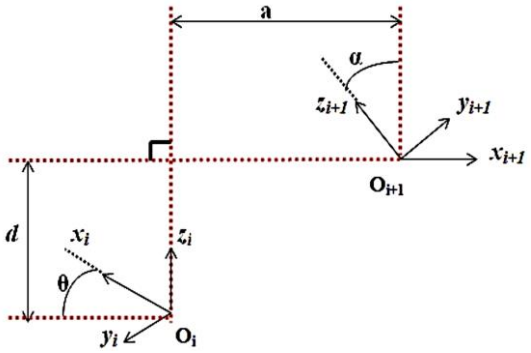


Figure 1. DH parameter assistance frames [9]

To calculate the movement between two joints crossing a link, multiply a set of basic homogenous matrices in a fixed sequence as shown in Eq. 1, and the output is a matrix representing the position and orientation of the final frame with the frame that comes before it as described by Eq. 2 [8].

$$A_i^{i-1} = R_{z,\theta} * T_{z,d} * T_{x,a} * R_{x,\alpha} \quad (1)$$

$$A_i^{i-1} = \begin{bmatrix} \text{Cos}\,[\theta i] & -\text{Sin}\,[\theta i] * \text{Cos}\,[\alpha i] & \text{Sin}(\theta i) * \text{Sin}\,[\alpha i] & ai * \text{Cos}\,[\theta i] \\ \text{Sin}\,[\theta i] & \text{Cos}\,[\theta i] * \text{Cos}\,[\alpha i] & -\text{Cos}\,[\theta i] * \text{Sin}\,[\alpha i] & ai * \text{Sin}[\theta i] \\ 0 & \text{Sin}\,[\alpha i] & \text{Cos}\,[\alpha i] & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2. Inverse Kinematics

The forward kinematic equations are utilized to deal with inverse kinematic analysis. The goal of IK is to determine the angles for the rotating joints or the displacement for the prismatic joints that will force the robotic arm's end-effector to reach the intended spot [10], [11].

Inverse kinematic solutions are classified into three types: full analytical solutions (closed form solutions), numerical solutions, and semi-analytical solutions. In the first kind, all joint variables are solved analytically using configuration data. In many situations where the manipulator supports or is to be supported by a sensory system, the results of kinematics calculations must be given quickly to apply appropriate control actions. The second form of solution obtains all of the joint variables using iterative computing processes. The semi-analytical solutions are obtained by applying analytical preprocessing steps that reduce the search space and numerical refinement steps to achieve high accuracy, especially for complex robots [10].

2.2.1. Spherical Wrist

The wrist refers to the joints in the kinematic chain between the arm and the end-effector. In the robotic arm, which is being studied, all wrist joints are revolute. To allow the end-effector to be oriented, a spherical wrist is commonly attached to the robotic arm end. As shown in Figure 2, the axes of the three joints intersect at the wrist centre point [Pc] in a spherical wrist. The Inverse Kinematic analysis allows for the decoupling of position and orientation [12].

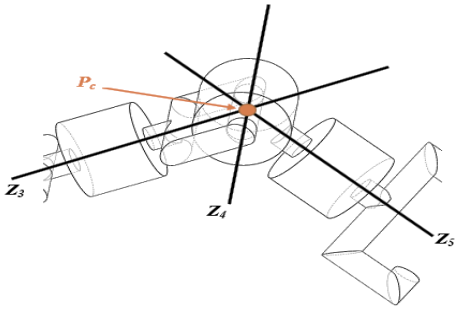


Figure 2. Spherical wrist joints intersecting at a single point [12]

Spherical wrists have the advantage of separating end-effector location from wrist orientation. The arm is in charge of locating the intersection point, while the wrist is in charge of determining end-effector orientation [13].

2.2.2. Kinematic Decoupling

Based on this kinematic decoupling, it is feasible to solve the arm's inverse kinematics independently from the spherical wrist's inverse kinematics [13]. For a robotic arm that has six joints (DOF = 6), the intersection of the last three joint axes is a single point which is the spherical wrist. The problem is then broken down into two sub-problems, which are inverse position kinematics and inverse orientation kinematics.

For inverse kinematics problem solving, the desired position and end-effector orientation are given in a homogeneous square matrix (H_6^0) with a dimension of (4×4) as shown in Eq. 3. Then by using decoupling method, which states that with the wrist center (Pc), the first three joints' parameters (x, y, and z) can be obtained. Defining the centre depending on the position (P6) can be defined as shown in Eq. 5 and shown in Figure 3 [14].

A translation of distance d6 along rotating axis of joint six (z5) from Pc yields the position of the end-effector centre, P6 as Figure 3. We can choose the third column of the desired rotation matrix (R_6^0) in Eq. 4 as the direction of z6 and z5 w.r.t. the base frame because z5 and z6 are on the same axis.

$$\mathrm{H}_6^0 = \begin{bmatrix} \mathrm{R}_6^0 & \mathrm{P}_6^0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

$$\mathrm{R}_6^0 = \begin{bmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} \end{bmatrix} \tag{2}$$

$$\mathrm{P}_6 = \mathrm{P}_\mathrm{c} + \mathrm{d}_6 \cdot \mathrm{R}_6^0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3}$$

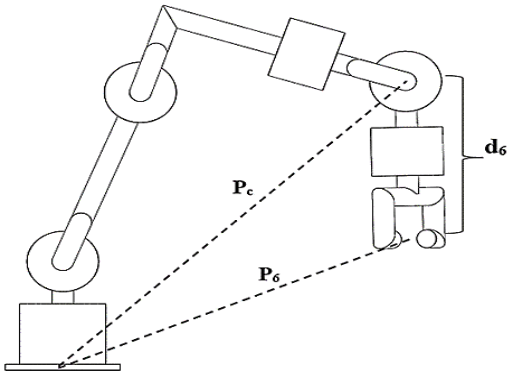


Figure 3. Six DOF arm configuration

Each of the first three angles of the robotic arm can be determined from the preceding equation. These angles represent the movement of the arm's body. This causes the arm to reach its intended destination, which is the centre of the spherical wrist (Pc) as shown in Figure 4 [14].

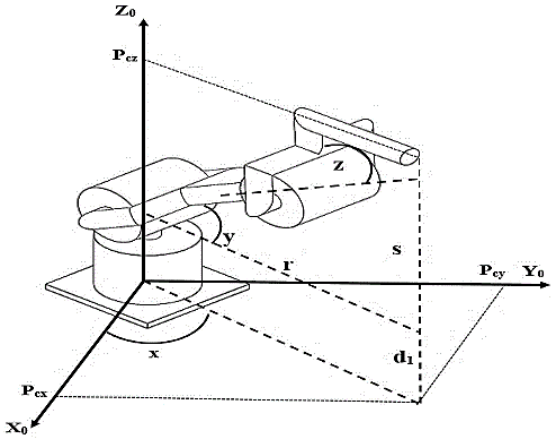


Figure 4. Arm body configuration describes the geometric of first three joints' angles [14]

When calculating the angle of the first joint, it can be observed that there are two potential results. One for the arm moving to the right direction and the other for the arm moving to the left direction according to the sign of the sine as in Eq. 6 but the amount of rotation of the joint motor must be taken into consideration [15].

$$x = \tan^{-1}\frac{\pm P_{cy}}{\pm P_{cx}} \tag{4}$$

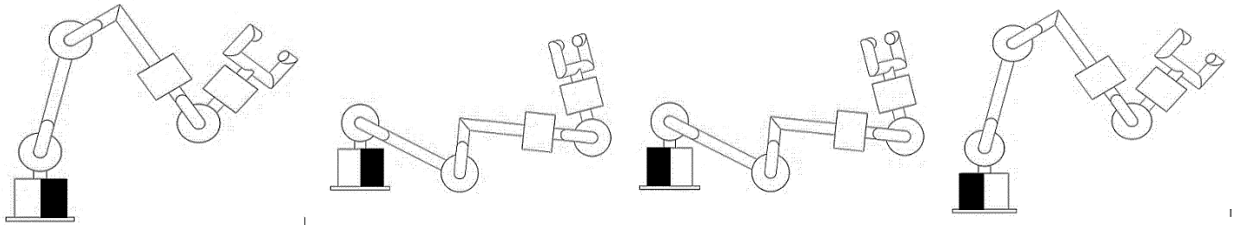
To compute the motors' rotation angles for both the second and third joints, we evaluate the body geometrics of the robotic arm as shown in Figure 4. Based on that, we conclude that the second angle of the arm can be calculated by Eq. 7 and the third angle by Eq. 8.

$$y = \tan^{-1}\left(\pm\frac{s}{r}\right) - \tan^{-1}\left(\frac{\left(\sqrt{a_3^2+{d_4}^2}\right)\operatorname{Sin}\beta}{a_2+\left(\sqrt{a_3^2+{d_4}^2}\right)\operatorname{Cos}\beta}\right) \tag{5}$$

$$z = \tan^{-1}\left(\frac{d_4}{a_3}\right) + \tan^{-1}\left(\frac{\pm\sqrt{1-D^2}}{D}\right) \tag{6}$$

All of s, r, γ, D, and β can be calculated by their deduction equations in Appendix A 1, 2, 3, 4, and 5.

In general, Figure 5 clarifies all the options for the robotic arm's body movement. There are four distinct options. Pose 1 demonstrates how to move the arm from its right position with the elbow up as in a and down as in b, while pose 2 shows how to move the arm to the left direction with the elbow down as in c and up as in d [12].



a. Right arm elbow up b. Right arm elbow down c. Lift arm elbow down d. Lift arm elbow up

Figure 5. Robotic arm's body poses with respect to first 3 joints angles

The orientation of the end-effector can be controlled by the angles of the spherical wrist joints. So, to obtain those angles, the wrist forward kinematics rotation matrix should be determined by Eq. 9. Then, the matrix rotation part will be compared with the homogenous matrix for the rotation part of a given desired target as in Eq. 3 multiplied by the first three joints rotation matrix inverse ($R_3^{0\,-1}$) Which can be deduced by multiplying the rotation matrices of the first three joints of the robotic arm [13].

$$H_6^3 = \begin{bmatrix} R_6^3 & P_6^3 \\ 0 & 1 \end{bmatrix} \tag{9}$$

$$R_6^3 = R_3^{0\,-1} * R_6^0 \tag{10}$$

$$R_6^3 = \begin{bmatrix} \operatorname{Cos}[a]\operatorname{Cos}[b]\operatorname{Cos}[c] - \operatorname{Sin}[a]\operatorname{Sin}[c] & -\operatorname{Cos}[c]\operatorname{Sin}[a] - \operatorname{Cos}[a]\operatorname{Cos}[b]\operatorname{Sin}[c] & \operatorname{Cos}[a]\operatorname{Sin}[b] \\ \operatorname{Cos}[b]\operatorname{Cos}[c]\operatorname{Sin}[a] + \operatorname{Cos}[a]\operatorname{Sin}[c] & \operatorname{Cos}[a]\operatorname{Cos}[c] - \operatorname{Cos}[b]\operatorname{Sin}[a]\operatorname{Sin}[c] & \operatorname{Sin}[a]\operatorname{Sin}[b] \\ -\operatorname{Cos}[c]\operatorname{Sin}[b] & \operatorname{Sin}[b]\operatorname{Sin}[c] & \operatorname{Cos}[b] \end{bmatrix} \tag{11}$$

One joint angle solution for fifth joint position (b) can be found by evaluating element (3,3) of matrix shown in Eq. 11; the associated fourth joint angle which is symbolled as (a) can be obtained by evaluating elements (1,3) and (2,3); and the relevant sixth joint angle which symbolled as (c) can be computed by evaluating elements (3,1) and (3,2). The fifth joint position (b) solution may be simply calculated using the preceding answer. The resulting equations are provided in Eq.12, Eq.13 and Eq. 14 [15], [16].

$$a = \tan^{-1}\left(\frac{\pm\operatorname{Sin}[a]\operatorname{Sin}[b]}{\pm\operatorname{Cos}[a]\operatorname{Sin}[b]}\right) = \tan^{-1}\left(\frac{r_{13}\operatorname{Sin}[x] - r_{23}\operatorname{Sin}[x]}{r_{13}\operatorname{Cos}[x]\operatorname{Cos}[y+z] + r_{23}\operatorname{Cos}[y+z]\operatorname{Sin}[x] + r_{33}\operatorname{Sin}[y+z]}\right) \tag{12}$$

$$b = \tan^{-1}\left(\frac{\pm\sqrt{1-F^2}}{F}\right) \tag{7}$$

Where F can be obtained with Appendix A 6.

$$c = \tan^{-1}\left(\frac{\pm\operatorname{Sin}[c]\operatorname{Sin}[b]}{\pm\operatorname{Cos}[c]\operatorname{Sin}[b]}\right) = \tan^{-1}\left(\frac{\operatorname{Cos}[x]\operatorname{Sin}[y+z]r_{12} + \operatorname{Sin}[x]\operatorname{Sin}[y+z]r_{22} - \operatorname{Cos}[y+z]r_{32}}{-(\operatorname{Cos}[x]\operatorname{Sin}[y+z]r_{11} - \operatorname{Sin}[x]\operatorname{Sin}[y+z]r_{21} + \operatorname{Cos}[y+z]r_{31})}\right) \tag{8}$$

3. Operating Time Equation

To find the operating time equation for a robotic arm, a set of equations must be deduced to calculate the operating time for each joint to rotate one unit with the maximum allowable velocity as shown in Eq. 15. Eq. 16 finds the operating time to move the end-effector from a point to another point by finding the sum of the angles of rotation of each joint to get from one point a to point b multiplied by the rotating unit operating time for that joint produced in Eq.16. Finally, the overall operating time equation to reach the designated points on a specific path ($O.T_{Path}$) can be calculated in Eq. 17 which is capable of collecting all of the operating time over the complete path, which dependent on the arm's number of joints (n) and the total number of points on the path (k) [17].

$$O.T_n = \frac{1\ rad}{\omega_n} \tag{9}$$

$$O.T_{P2P} = \sum_{i=1}^{n}(|(q_{bi} - q_{ai})|) * O.T_i \tag{10}$$

$$O.T_{Path} = \sum_{j=1}^{k}\sum_{i=1}^{n}(|(q_{bij} - q_{aij})|) * O.T_i \tag{11}$$

4. Trajectory Optimization

As shown in Figure 6, there are three main categories of optimization techniques: artificial intelligence techniques, metaheuristic methods, and mathematical methods. The If-then-rule-based technique, deep learning, and machine learning are all types of artificial intelligence. Many popular optimization methods fall within the category of metaheuristic algorithms. Linear Programming and Nonlinear Programming are examples of mathematical optimization approaches [18].

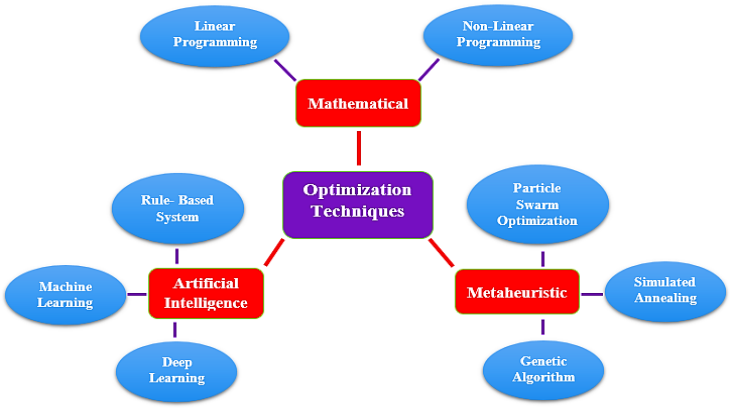


Figure 6. Optimization techniques [18]

It is clear from the Figure 6 that one of these strategies have to be employed to solve the optimization issue. The core of our paper is to use two different optimization techniques to solve the trajectory optimization problem and compare their performance with respect to the algorithm's running time. The two methods used in our study are "if then rule-based method" and the "genetic algorithm approach".

4.1. Rule-Based Technique

The Rule-Based Technique is an AI approach used to solve optimization problems. This is done by dealing with symbolic, non-algorithmic problem-solving approaches including heuristic principles to create knowledge-based expert systems by storing the declarative knowledge (facts about things, events, and situations) and the procedural knowledge (information about courses of action). The rule-based system method is one of the most common knowledge representation strategies used in expert systems. In a rule-based system, procedural information is combined with declarative knowledge in the form of 'if-then' rules [19].

4.2. Genetic Algorithm

Genetic Algorithm [GA] is an optimization approach based on natural evolution of organisms as shown in Figure 7. Therefore, biological background may be employed when GA is created to have a better understanding. The optimal features may be attributed to gene exchange between generations. In other

words, a gene is something that may improve outcomes when it is altered. Through genes exchange between different generations, the optimized properties can be credited to enhance the results when the gene is changed.

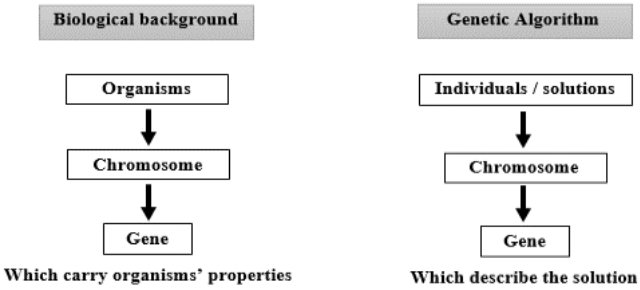


Figure 7. Genetic algorithm vs biological system

Starting with the establishment of the initial population, the genetic algorithm consists of numerous successive fixed phases as shown in Figure 8. To boost variety, this population may be produced from a Gaussian random distribution [18]. The population has numerous solutions that represent individual chromosomes. Each chromosome contains a collection of variables that represent the genes [20]. A subset of the current population is chosen to reproduce a new generation with each succeeding generation. Fitter solutions (as judged by a fitness function) are more likely to be chosen for implementation as individual solutions [21].

After a selection operator identifies the greatest candidates, they must develop the next generation. When male and female chromosomes unite, a new chromosome is formed. The roulette wheel chooses two solutions (parent solutions) and simulates it using the GA algorithm. Last evolutionary operator in which one or more genes are changed after children's treatments. High mutation rates turn GA into a random search problem. Mutation creates unpredictability to maintain population diversity. This operator helps prevent similar and local optimum solutions [22].

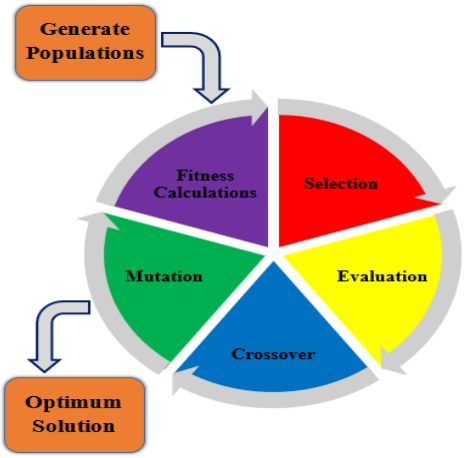


Figure 8. Genetic algorithm sequence [18]

To use the genetic algorithm to optimize the elements that affect the course of the robotic arm, the main function, we defined the objective function shown in Eq. 18 which depends on the fitness value, shown in Eq. 19, and the overall path time $O.T_{Path}$, described before by Eq. 17, which contains the optimization parameters for our problem. The selection of weights for multiple objective function optimization is an essential challenge in the optimization process. This challenge naturally arises in many reconstruction scenarios when one needs to rebuild a function from a limited class of signals based on noisy observed data [23].

$$\text{Objective Function} = \text{Fitness Value} + \text{Weight} * \text{Optimizing Parameter} \quad (12)$$

As the robotic arm optimization system enhances performance on operating time, the objective function as shown in Eq. 20 was used to adapt to the system considered in this study.

$$\text{Fitness Value (FV)} = \sum_{j=1}^{k} \sqrt{(P_x - GA_x)^2 + (P_y - GA_y)^2 + (P_z - GA_z)^2} \quad (13)$$

$$\text{Objevtive Function (OF. OT)} = \text{FV} + \text{Weight } (\delta) * O.T_{Path} \quad (20)$$

5. Case Study

For this paper, the KUKA Company's model KR 4 R600 robotic arm, whose dimensions are given in Figure 9, was used [24]. The Forward Kinematics was performed, and the DH parameters are inferred by establishing the XYZ axes for each joint of the arm.

The KR 4 R600 is a 6-axis (6 Degree of Freedom) industrial robot from KUKA Robotics with a payload of 3 kg, built primarily for laboratory businesses that utilize flexible robot-based automation. The robot has an open structure that lends itself well to variable applications, and it can communicate extensively with other systems [24].

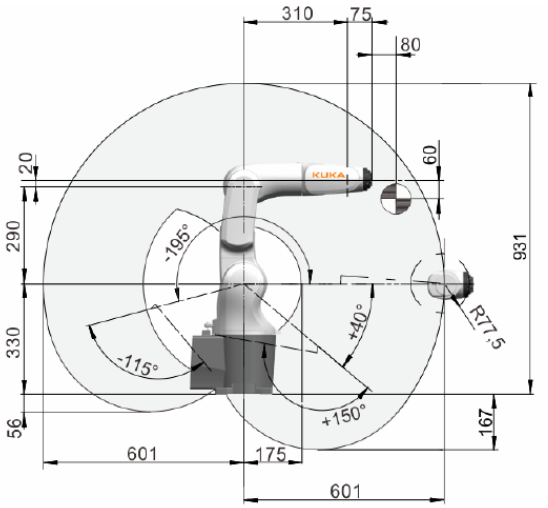


Figure 9. Robotic arm configuration [24]

Table 1. KUKA KR 4 R600 robotic arm specifications

Joints Motors specification	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Rotation range (rad)	2.96 to - 2.96	0.69 to - 3.4	2.62 to - 2	3.23 to - 3.23	2.09 to - 2.09	6.1 to - 6.1
Rotation Speed (rad/sec)	4.364	4.364	4.364	5.586	5.586	7.331

5.1. Robot Kinematics

The KUKA robotic arm has 6 degrees of freedom. The rules for positioning the joints frame can be implemented, with the Z-axis always passing through the axis of rotation of each joint. The X-axis is then positioned where it is needed, as long as it is vertical and intersects both the Z-axis of the current frame and the Z-axis of the previous frame. The third axis, the Y-axis, was then put for each frame, following the rule of counterclockwise rotation in the XYZ sequence as shown in Figure 10. The DH parameters for each link were calculated using four lines, the first two that describe the relationship between the next and current X-axis and the second two that explain the relationship between the previous and current Z-axis as shown in Table 2.

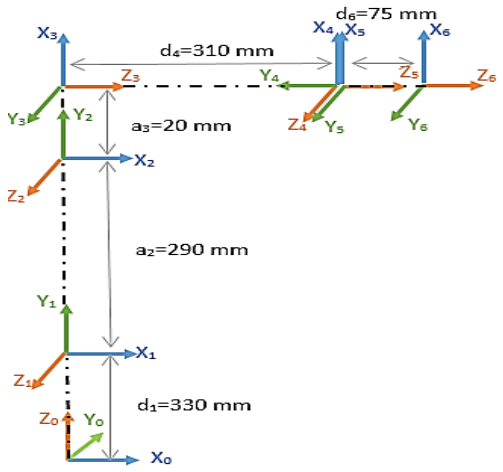


Figure 10. Forward kinematics frame

Table 2. DH parameters

Arm Links (Frame i to Frame i+1)	a (m)	α (rad)	d (m)	θ (rad)
Link 1 (F0 to F1)	0	1.57	0.330	x
Link 2 (F1 to F2)	0.290	0	0	y
Link 3 (F2 to F3)	0.020	1.57	0	z
Link 4 (F3 to F4)	0	-1.57	0.310	a
Link 5 (F4 to F5)	0	1.57	0	b
Link 6 (F5 to F6)	0	0	0.075	c

By inserting each of the DH parameters from Table 2 in the general matrix and substituting in the matrix specified in Eq.2, a homogeneous matrix was generated that depicts the movement of each joint relative to the joint before it.

After computing the homogeneous matrix for each link, the connection between the final frame, which represents the robotic arm's end-effector, and the zero frame, which represents the arm's base, can be described by multiplying all the individual matrices in the arm's sequence as in Eq.21. The robot's homogeneous matrix was derived as shown in Eq.22 that allows the end-effector of the arm to be regulated in terms of position and orientation [25].

$$H_6^0 = A1 * A2 * A3 * A4 * A5 * A6 \tag{21}$$

$$H_6^0 = \begin{bmatrix} r11 & r12 & r13 & p_{6x} \\ r21 & r22 & r23 & p_{6y} \\ r31 & r32 & r33 & p_{6z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

The forward kinematics results are used for each of the path points specified for the robotic arm's task. It is used to determine the robotic arm's end effector position, which can be seen in the matrix mentioned in Eq. 22 in the elements (1:3, 4), and is unique to each point. So, it can be said that after applying the forward kinematics there will be k matrices such as those mentioned in Eq. 22.

The inverse kinematic equations are then applied to the location of each point to obtain a set of equations for each point that describe the movement of each of the robotic arm's six joints.

As per our study case, a route was specified as a job for the arm to complete, which consists of five points. One of those points was initialized as a starting point, which is detected if all robotic arm joints are in their initial positions as shown in Table 3.

Table 3. Case 1: path points

Point	Coordinate (x, y, z)
P0	(0.310, 0, -0.055)
P1	(0.225, 0.11, 0.025)
P2	(0.09, 0.15, 0.25)
P3	(-0.085, 0.114, 0.12)
P4	(-0.18, 0.125, 0.117)

5.2. Optimization Approaches and Results

The following optimization algorithms proposed in sections 5.2.1 and 5.2.2 solves a reachability problem for the points in the defined path while maintain the minimum operating time. The MATLAB version 2020a application was used as a programming platform to implement the two algorithms. A computer with the following specifications: a laptop with an Intel Core i7 CPU and 16 gigabytes of RAM, was used to test the performance of the two algorithms described in the paper while applying the same testing environment.

5.2.1. Rule-Based Technique Results

Algorithm 1 shows the proposed rule-based technique written in MATLAB and used to solve the trajectory optimization problem for a KUKA 6 DOF robotic arm. It consists of a collection of If-statements and loops, which describe system constrains and strategies to explore all potential paths to finish the route. Then, the algorithm picks the best settings that ensures a minimum operating time for the overall trip.

Algorithm 1. IF-then rule-based algorithm based on minimum operating time

Input: DH parameters for six joints (ai, αi, di, and θi), and desired locations (Px, Py, Pz).

Output: Generate the robot's settings that grantee minimum operating time.

1: for each point j = 1:5

$Pcxj = Pxj - d6*r13; \quad Pcyj = Pyj - d6*r23; \quad Pczj = Pzj - d6*r33;$

$\theta 1j = tan\text{-}1(\frac{\pm Pcy}{\pm Pcx})$

$$\theta 2j = \boldsymbol{tan^{-1}}\left(\pm\frac{s}{r}\right) - \boldsymbol{tan^{-1}}\left(\frac{\left(\sqrt{a_3^2+d_4^{\,2}}\right)\boldsymbol{Sin\,\beta}}{a_2+\left(\sqrt{a_3^2+d_4^{\,2}}\right)\boldsymbol{Cos\,\beta}}\right)$$

$$\theta 3j = \boldsymbol{tan^{-1}}\left(\frac{d_4}{a_3}\right) + \boldsymbol{tan^{-1}}\left(\frac{\pm\sqrt{1-D^2}}{D}\right)$$

$$\theta 4j = \boldsymbol{tan^{-1}}\left(\frac{\pm Sin[\theta 4j]Sin[\theta 5j]}{\pm Cos[\theta 4j]Sin[\theta 5j]}\right)$$

$$\theta 5j = \boldsymbol{tan^{-1}}\left(\frac{\pm\sqrt{1-F^2}}{F}\right)$$

$$\theta 6j = \boldsymbol{tan^{-1}}\left(\frac{\pm Sin[\theta 6j]Sin[\theta 5j]}{\pm Cos[\theta 6j]Sin[\theta 5j]}\right)$$

$$solutions\ for\ j = \begin{bmatrix} \boldsymbol{\theta 11} & \dots & \boldsymbol{\theta i1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\theta 18} & \cdots & \boldsymbol{\theta i8} \end{bmatrix}$$

2: constraints for each joint for each point

for k = 1: length of solution j

if solution j (i, k)<= minimum range and solution j (i, k)>= maximum range

solution j (i) = null

end if

end for

3: trying all possible sequential solutions

for a = 1: length of solution 1

for b = 1: length of solution 2

for c = 1: length of solution 3

for d = 1: length of solution 4

for e = 1: length of solution 5

$$\boldsymbol{O.T_{Path}} = \sum_{j=1}^{k}\sum_{i=1}^{n}\left(\left|(\boldsymbol{q_{bij}} - \boldsymbol{q_{aij}})\right|\right) * \boldsymbol{O.T_i}$$

if $\boldsymbol{O.T_{Path}}(l) < \boldsymbol{O.T_{Path}}(l\text{-}1)$

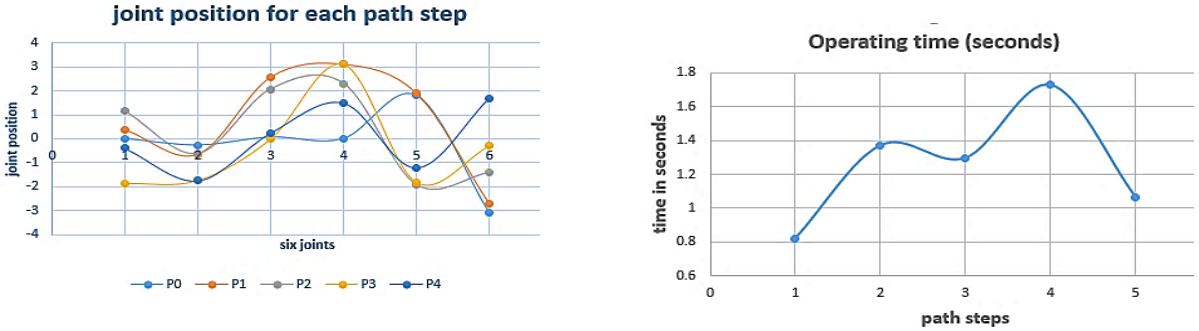
then minimum solution is $\boldsymbol{O.T_{Path}}(l)$

else the minimum solution is $\boldsymbol{O.T_{Path}}(l\text{-}1)$

end if

end for

Table 4 shows the results generated from Algorithm 1. Those results represent the values of the rotational angles that guarantee the improvement. Same results are represented graphically in Fig. 11(a) by drawing the change in the joint's settings to reach every point in the defined path. Using those results and with the help of Eq. 16 and Eq. 17, the minimal overall operating time for the specified path can be deduced which is shown in



a. Joints angles during the path b. Overall operating time

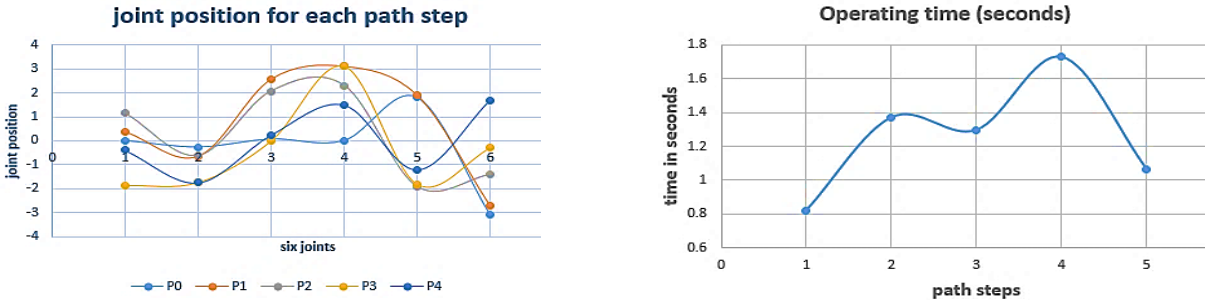
Figure 11. Rule-based characteristics during the path for minimum operating time

Table 5 and Figure 11(b).

Table 4. Rule-based results according to grantee the minimum operating time

Path steps	x (rad)	y (rad)	z (rad)	a (rad)	b (rad)	c (rad)
P0	0	-0.2700	0.0630	0	1.8000	-3.1000

P1	0.3800	-0.6400	2.5670	3.1000	1.9000	-2.7000
P2	1.1500	-0.6100	2.0660	2.3000	-1.9000	-1.4000
P3	-1.8568	-1.7300	-0.0090	3.1000	-1.8000	-0.3000
P4	-0.3883	-1.7400	0.2480	1.5000	-1.2000	1.7000



a. Joints angles during the path b. Overall operating time

Figure 11. Rule-based characteristics during the path for minimum operating time

Table 5. Operating time for each path step

Path	initial to P0	P0 to P1	P1 to P2	P2 to P3	P3 to P4	Total path
Operating time (seconds)	0.8201	1.3724	1.2981	1.7309	1.0632	**6.2847**

5.2.2. Genetic Algorithm Results

Each GA population member is in Eq.23, with 6 (DOF) x 5 (points) = 30 items (genes), corresponding to 6 rotation angles for reaching each working point. In Table 1, all genes' ranges are specified. Because of the task's specification, a single configuration will not appear. An individual is a string that contains of the optimized object's parameters. Mutation is the process of changing a parent person at random (mutation rate was 0.15). Crossover is a process in which the characteristics of two parents are mixed at random to generate a kid (crossover rate used was 0.6). The computation of the robot movement and the objective function evaluation are included in the fitness function evaluation.

$$Q = [q_{11}, q_{12}, q_{13} \cdots q_{1n}, q_{21}, q_{22} \cdots q_{k1}, q_{k2} \cdots q_{kn}] \quad (23)$$

When using the genetic algorithm as an algorithm to improve the task path of the robotic arm based on the operating time of the arm's six joints throughout the path, it may be degraded into numerous phases, beginning with a description of the input and what the needed output from the code is, and which can be discussed in depth via its steps using the pseudo code displayed in Algorithm 2. The core of an algorithm is to use the kinematics of a robotic arm to determine the best route it can travel while accounting for the fewest operating time.

Algorithm 2. Genetic algorithm based on minimum rotating angles

Input: Start and final destination, joints constraints (rotating range, and maximum velocity), and start and final joints position.

Output: Optimum solution with five knots that grantee minimum operating time.

1: Create MATLAB function contain objective function
function OT= OF_OT (theta)
$\boldsymbol{Objevtive\ Function\ (OF.OT) = FV + Weight\ (\delta) * O.T_{Path}}$

2: Inequalities and equalities
Joints constraints (rotating range (LB, UB), and maximum velocity)

3: Run MATLAB optimizer app
select weight ($\boldsymbol{\delta = 1}$)
Then run the optimizer app with start button

4: Calculate the total operating time as a function in optimizer output
for each knot (k = 1:5)
for each joint (i = 1:6)
$\boldsymbol{O.T_{Path} = \sum_{j=1}^{k}\sum_{i=1}^{n}(|(q_{bij} - q_{aij})|) * O.T_i}$

The results shown in

Table *6* were obtained as a result of running Algorithm 2 to trajectory control the KUKA robotic arm over the specified path. Those results represent the angles of rotation for each joint at each point to ensure the minimum operating time, which graphed in Figure 12(a). In addition, by utilizing the data in

Table *6*, we got the results in Table 7 and Figure 12(b), which indicate the total time for the complete journey.

Table 6. GA angles according to grantee the minimum operating time

Path steps	x (rad)	y (rad)	z (rad)	a (rad)	b (rad)	c (rad)
P0	0	-0.26	0.084	0	1.97	-3.1
P1	0.377	0.562	0.165	0.02	2.047	-3.1
P2	1.228	0.562	0.165	2.76	2.047	-3.1
P3	1.285	0.171	-0.044	2.76	1.352	-3.1
P4	2.552	0.171	0.025	2.76	1.352	-3.1

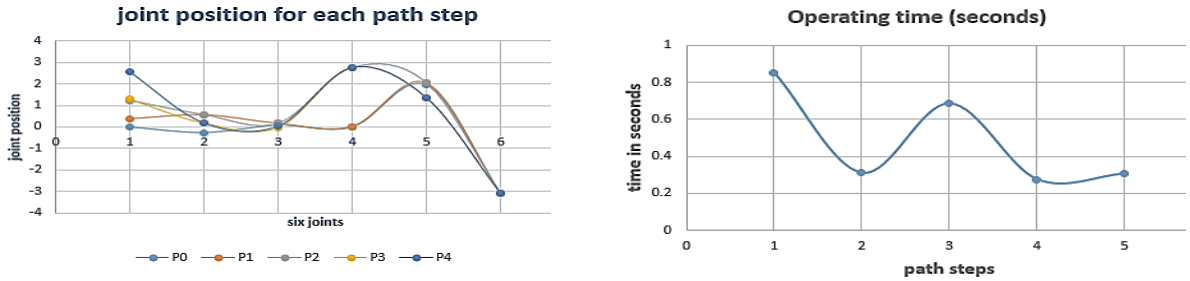
Table 7. Operating time for each path step

Path	initial to P0	P0 to P1	P1 to P2	P2 to P3	P3 to P4	Total path
Operating time (seconds)	0.8530	0.3105	0.6853	0.2749	0.3059	2.4296

Table 8. All optimization techniques results

NO.	Techniques	Average Path time (seconds)
1	Rule-based for mini. time	6.2847
2	GA for mini time	2.4296

To Summarize all the results obtained by applying the two proposed techniques (Rule-based for minimum operating time and GA for minimum operating time) used in the paper, the results were collected in Table 8, which presents the average running time after running the two algorithms over around 1760 different trails.



a. Joints angles during the path b. Overall operating time

Figure 12. GA characteristics during the path for minimum operating time

Based on the simulation results, we found that the genetic based approach is faster (in execution) than the rule-based technique while considering the same parameters. These times were calculated under the same operating and implementation conditions, on the same hardware and the same version of the program. We can argue this as the rule-based approach is generates its results after discovering all the possible solutions in the problem's searching space which takes more time than the genetic approach.

6. Conclusion and Future Work

Various design element, such as robot operating minimization, has been explored in the suggested robotic arm trajectory optimization. Because the inverse kinematics problem for a robot with several degrees of freedom is a complicated issue, two approaches have been proposed: the rule-based method and the genetic algorithm approach. The findings revealed that the genetic-based proposed algorithm can produce extremely excellent outcomes, as well as considerable reachability time and reduced wear and strain on the equipment owing to motors turning.

Through this idea, our future work will consider not on the reachability problem but also considering the power consumption to move the robotic arm end from a point to another.

Appendix A

1. $s = (P_{cz} - d_1)$
2. $r = \pm\sqrt{P_{cx}^2 + P_{cy}^2}$
3. $\gamma = \tan^{-1}\left(\frac{d_4}{a_3}\right)$
4. $D = \mathrm{Cos}(\beta) = \frac{s^2 + r^2 - a_2^2 - ({a_3}^2 + {d_4}^2)}{2a_2\left(\sqrt{{a_3}^2 + {d_4}^2}\right)}$
5. $\beta = \tan^{-1}\left(\frac{\pm\sqrt{1-D^2}}{D}\right)$

6. $F = (\mathrm{Cos}[x]\mathrm{Sin}[y+z])r_{13} + (\mathrm{Sin}[x]\mathrm{Sin}[y+z])r_{23} - (\mathrm{Cos}[y+z])r_{33} = \mathrm{Cos}\,[b]$

References

[1] Kazim Raza, Tauseef Aized Khan and Naseem Abbas, "Kinematic analysis and geometrical improvement of an industrial robotic arm", *Journal of King Saud University - Engineering Sciences*, Print ISSN: 1018-3639, Online ISSN: 2213-1558, pp. 218-223, Vol. 30, No. 3, July 2018, Published by Elsevier B.V., DOI: 10.1016/j.jksues.2018.03.005, Available: https://www.sciencedirect.com/science/article/pii/S1018363917304464.

[2] Steven M. Lavalle, *Planning Algorithms*, 1st ed. New York, USA: Cambridge University Press, 2006, Available: https://www.cambridge.org/core/books/planning-algorithms/FC9CC7E67E851E40E3E45D6FE328B768.

[3] Mariana Ratiu and Mariana Adriana Prichici, "Industrial robot trajectory optimization- a review", *in Proceedings of MATEC Web of Conferences, Annual Session of Scientific Papers IMT ORADEA 2017*, Sanmartin, Romania, 27-29 May 2017, E-ISSN: 2261-236X, DOI: 10.1051/matecconf/201712602005, pp. 1-6, Published by MATEC, Available: https://www.matec-conferences.org/articles/matecconf/pdf/2017/40/matecconf_imtoradea2017_02005.pdf.

[4] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti and Renato Vidoni, "Trajectory Planning in Robotics", *Mathematics in Computer Science*, Print ISSN: 1661-8270, Online ISSN: 1661-8289, pp. 269-279, Vol. 6, No. 1, 30th August 2012, Published by Springer, DOI: 10.1007/s11786-012-0123-8, Available: https://link.springer.com/article/10.1007/s11786-012-0123-8.

[5] Giuseppe Carbone and Fernando Gomez-Bravo, *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, 1st ed. New York: USA, Springer Cham, 2015, ISBN: 978-3-319-14704-8, Available: https://link.springer.com/book/10.1007/978-3-319-14705-5.

[6] Mauricio Restrepo, Claudio A. Cañizares, John W. Simpson-Porco, Peter Su and John Taruc, "Optimization- and Rule-based Energy Management Systems at the Canadian Renewable Energy Laboratory microgrid facility", *Applied Energy*, Print ISSN: 0306-2619, Online ISSN: 1872-9118, pp. 1-11, Vol. 290, 15th May 2021, DOI: 10.1016/j.apenergy.2021.116760, Available: https://www.sciencedirect.com/science/article/pii/S0306261921002671.

[7] Runqi Chai, Al Savvaris, Antonios Tsourdos and Senchun Chai, *Design of Trajectory Optimization Approach for Space Maneuver Vehicle Skip Entry Problems*, 1st ed. Singapore: Springer Singapore, 2020, ISBN: 978-981-13-9844-5, Available: https://link.springer.com/book/10.1007/978-981-13-9845-2.

[8] C. D. Crane, III and J. Duffy, *Kinematic Analysis of Robot Manipulators*, 1st ed. Cambridge, UK: Cambridge University Press, 1998, DOI: 10.1017/CBO9780511530159, Available: https://www.cambridge.org/core/books/kinematic-analysis-of-robot-manipulators/5D4339E1E7180EA7950D463F41640FAB.

[9] Hind Hadi Abdulridha and Tahseen Fadhel Abaas, "Differential Motion Analysis of Lab-Volt R5150 Robot System", *Global Journal of Engineering Science and Research Management*, ISSN: 2349-4506, pp. 152-160, Vol. 5, No. 10, October 2017, Published by GJESRM, DOI: 10.5281/zenodo.1034505, Available: https://www.gjesrm.com/Issues%20PDF/Archive-2017/October-2017/14.pdf.

[10] Serdar Kucuk and Zafer Bingul, "The inverse kinematics solutions of industrial robot manipulators", in *Proceedings of the IEEE International Conference on Mechatronics 2004 (IEEE ICM '04)*, Istanbul, Turkey, 5-5 June 2004, pp. 274-279, Print ISBN: 0-7803-8599-3, DOI: 10.1109/ICMECH.2004.1364451, Published by IEEE, Available: https://ieeexplore.ieee.org/document/1364451/.

[11] Sam Cubero, *Industrial Robotics: Theory, Modelling and Control*, 1st ed. Moscow, Russia: IntechOpen, 2006, Available: https://www.intechopen.com/books/6110.

[12] Said M. Megahed, "Inverse kinematics of spherical wrist robot arms: Analysis and simulation", *Journal of Intelligent and Robotic Systems*, Print ISSN: 0921-0296, Online ISSN: 1573-0409, pp. 211-227, Vol. 5, No. 1, June 1992, Published by Springer, DOI: 10.1007/BF00247418, Available: https://link.springer.com/article/10.1007/BF00247418.

[13] E. W. Kamen, J. K. Su, *Introduction to Optimal Estimation*, 1st ed. London, UK: Springer London, 1999, DOI: 10.1007/978-1-4471-0417-9, Available: https://link.springer.com/book/10.1007/978-1-4471-0417-9.

[14] Raffaele Di Gregorio, "Kinematics of a new spherical parallel manipulator with three equal legs: The 3-URC wrist", *Journal of Robotic Systems*, Print ISSN:1556-4959, Online ISSN: 1556-4967, pp. 213-219, Vol. 18, No. 5, 10th April 2001, Published by Wiley Periodicals, DOI: 10.1002/rob.1017, Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.1017.

[15] Seemal Asif and Philip Webb, "Kinematics Analysis of 6-DoF Articulated Robot with Spherical Wrist", *Mathematical Problems in Engineering*, Print ISSN: 1024-123X, Online ISSN: 1563-5147, pp. 1-11, Vol. 2021, 2nd February 2021, DOI: 10.1155/2021/6647035, Available: https://www.hindawi.com/journals/mpe/2021/6647035/.

[16] Zhijian Gou, Ying Sun and Haiying Yu, "Inverse kinematics equation of 6-DOF robot based on geometry projection and simulation", in *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering 2010 (IEEE CMCE 2010)*, Changchun, China, 24-26 August 2010, pp. 125-128, Print ISBN: 978-1-4244-7957-3, Electronic ISBN: 978-1-4244-7958-0, DOI: 10.1109/CMCE.2010.5610077, Published by IEEE, Available: https://ieeexplore.ieee.org/abstract/document/5609677/.

[17] Stanislav Števo, Ivan Sekaj and Martin Dekan, "Optimization of Robotic Arm Trajectory Using Genetic Algorithm", in *Proceedings of the 19th World Congress, The International Federation of Automatic Control 2014*, Cape Town, South Africa, 24-29 August 2014, pp. 1748-1753, ISSN: 1474-6670, DOI: 10.3182/20140824-6-ZA-1003.01073, Published by Elsevier IFAC Proceedings, Available: https://www.sciencedirect.com/science/article/pii/S1474667016418653.

[18] Abdelrahman O. Ali, Mohamed R. Elmarghany, Mohamed M. Abdelsalam, Mohamed Nabil Sabry and Ahmed M. Hamed, "Closed-loop home energy management system with renewable energy sources in a smart grid: A comprehensive review", *Journal of Energy Storage*, Print ISSN: 2352-152X, Online ISSN: 2352-1538, pp. 1-24, Vol. 50, June 2022, Published by Elsevier, DOI: 10.1016/j.est.2022.104609, Available: https://www.sciencedirect.com/science/article/pii/S2352152X22006259.

[19] H. Azaria and A. Dvir, "Algorithm optimization using a rule-based system. A case study: The Direct Kinematic Solution in robotics", *Journal of Intelligent & Robotic Systems*, Print ISSN: 0921-0296, Online ISSN: 1573-0409, pp. 309-324, Vol. 8, December 1993, Published by Springer, DOI: https://doi.org/10.1007/BF01257947, Available: https://link.springer.com/article/10.1007/BF01257947.

[20] Seyedali Mirjalili, *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 1st ed. Cham, Switzerland: Springer, 2019, Available: https://link.springer.com/book/10.1007/978-3-319-93025-1.

[21] Manoj Kumar, Mohammad Husain, Naveen Upreti and Deepti Gupta, "Genetic Algorithm: Review and Application", *International Journal of Information Technology and Knowledge Management*, Print ISSN: 1461-4111, Online ISSN: 1741-5179, pp. 451-454, Vol. 2, No. 2, 1st December 2010, DOI: 10.2139/ssrn.3529843, Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3529843.

[22] Sourabh Katoch, Sumit Singh Chauhan and Vijay Kumar, "A review on genetic algorithm: past, present, and future", *Multimedia Tools and Applications*, Print ISSN: 1380-7501, Online ISSN: 1573-7721, pp. 8091-8126, Vol. 80, No. 5, February 2021, Published by Springer, DOI: 10.1007/s11042-020-10139-6, Available: https://link.springer.com/article/10.1007/s11042-020-10139-6.

[23] M.A. Gennert and A.L. Yuille, "Determining the Optimal Weights in Multiple Objective Function Optimization", in *Proceedings of Second International Conference on Computer Vision 1998*, Tampa, FL, USA, 5-9 December 1998, pp. 87- 93, Print ISBN: 0-8186-0883-8, Published by IEEE, DOI: 10.1109/CCV.1988.589974, Available: https://ieeexplore.ieee.org/document/589974.

[24] Mahmoud A. A. Mousa, Abdelrahman Tarek Elgohr and Hatem Khater, "Path Planning for a 6 DoF Robotic Arm Based on Whale Optimization Algorithm and Genetic Algorithm", *Journal of Engineering Research*, Print ISSN: 2356-9441, Online ISSN: 2735-4873, pp. 160-168, Vol. 7, No. 5, November 2023, Published by Arab Journals Platform, DOI: 10.21608/erjeng.2023.237586.1256, Available: https://erjeng.journals.ekb.eg/article_325086.html.

[25] Jolly Atit Shah, S.S. Rattan and B.C. Nakra, "End-Effector Position Analysis Using Forward Kinematics For 5 Dof Pravak Robot Arm", *International Journal of Robotics and Automation (IJRA)*, pp. 112-116, Vol. 3, No. 3, September 2013, Print ISSN: 2089-4856, Online ISSN: 2722-2586, Published by Institute of Advanced Engineering and Science (IAES), DOI: 10.11591/ijra.v2i3.pp112-116, Available: https://ijra.iaescore.com/index.php/IJRA/article/view/921.