Tikrit Journal of Engineering Sciences (2024); 31 (1): 1-11
DOI: http://doi.org/10.25130/tjes.31.1.1

IRAQI Academic Scientific Journals

ISSN: 1813-162X (Print); 2312-7589 (Online)

Tikrit Journal of Engineering Sciences

available online at: http://www.tj-es.com

TJES Tikrit Journal of Engineering Sciences

Dual Performance Optimization of 6-DOF Robotic Arm Trajectories in Biomedical Applications

Kian Raheem Qasim [a*], *Yousif Al Mashhadany* [b], *Esam Taha Yassen* [c]

a University of Information Technology and Communications, Baghdad, Iraq.
b Department of Electrical Engineering, College of Engineering, University of Anbar, Anbar, Iraq.
c Department of Computer Science, College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq.

Keywords:
Biomedical Application; Dual Performance; Inverse Kinematics; Optimal Trajectory; Optimization Problem; 6-DOF Robotic Arm.

Highlights:

- *To make the robotic system kinematically acceptable for biomedical applications, the suggested method modifies the kinematics of the robot arm by generating virtual points even when the robotic system is not redundant.*
- *The suggested approach chooses an appropriate objective function to reduce one or more robot arm kinematic properties while increasing one or more performance measures.*
- *The robot arm's end effector is fixed at strategic points, and the robot arm's self-motion causes the Dual Performance Precision algorithm to modify virtual joints and points.*
- *X-ray robot systems use this task, which is described as selecting the optimal path depending on the position and orientation of the input target.*

ARTICLE INFO

Article history:

Received	05 May	2023
Received in revised form	20 Aug.	2023
Accepted	14 Sep.	2023
Final Proofreading	19 Nov.	2023
Available online	03 Jan.	2024

Citation: Qasim KR, Al Mashhadany Y, Yassen ET. **Dual Performance Optimization of 6-DOF Robotic Arm Trajectories in Biomedical Applications**. *Tikrit Journal of Engineering Sciences* 2024; **31**(1): 1-11.
http://doi.org/10.25130/tjes.31.1.1

Corresponding author:

Kian Raheem Qasim

University of Information Technology and Communications, Baghdad, Iraq.

Abstract: For the first time, dual-performance perfection technologies were used to kinematically operate sophisticated robots. In this study, the trajectory development of a robot arm is optimized using a dual-performance perfection technique. The proposed approach alters the robot arm's Kinematics by creating virtual points even if the robotic system is not redundant to make it kinematically suitable for biomedical applications. In the suggested method, an appropriate objective function is chosen to raise one or maybe more performance measures while lowering one or more kinematic characteristics of a robot arm. The robot arm's end effector is set in place at the crucial locations, and the dual performance precision algorithm changes the joints and virtual points due to the robot arm's self-motion. As a result, the ideal values for the virtual points are established, and the robot arm's design is changed. Accordingly, this method's ability to visualize modifications made to the processor's design during the optimization problem is one of its benefits. The active robotic arm is used as a case study in this article. The task is defined as choosing the best path based on the input target's position and direction and is used in X-ray robot systems. The outcomes demonstrate the viability of the suggested approach and can serve as a useful prototype for an intelligent X-ray robot.

تحسين الأداء المزدوج لمسارات ذراع الروبوت بست درجات حرية في التطبيقات الطبية

كيان رحيم قاسم[1]، يوسف المشهداني[2]، عصام طه ياسين[2]

[1] جامعة تكنولوجيا المعلومات والاتصالات / بغداد – العراق.
[2] قسم الهندسة الكهربائية / كلية الهندسة / جامعة الانبار / الانبار – العراق.
[2] قسم علوم الحاسوب / كلية تكنولوجيا المعلومات وعلوم الحاسوب / جامعة الانبار / الانبار – العراق.

الخلاصة

لأول مرة، تم استخدام تقنيات الكمال ثنائية الأداء لتشغيل الروبوتات المتطورة بشكل حركي. في هذه الدراسة، تم تحسين تطور مسار ذراع الروبوت باستخدام تقنية الكمال ثنائية الأداء. يغير النهج المقترح الحركية لذراع الروبوت من خلال إنشاء نقاط افتراضية حتى لو لم يكن النظام الآلي زائداً عن الحاجة لجعله مناسبًا حركيًا للتطبيقات الطبية الحيوية. في الطريقة المقترحة، يتم اختيار وظيفة موضوعية مناسبة لرفع واحد أو ربما أكثر من مقاييس الأداء أثناء خفض واحد أو أكثر من الخصائص الحركية لذراع الروبوت. يتم وضع المستجيب النهائي لذراع الروبوت في مكانه في المواقع الحاسمة، وتغير خوارزمية دقة الأداء المزدوجة المفاصل والنقاط الافتراضية بسبب الحركة الذاتية لذراع الروبوت. نتيجة لذلك، يتم تحديد القيم المثالية للنقاط الافتراضية، وتغيير تصميم ذراع الروبوت. وفقًا لذلك، فإن قدرة هذه الطريقة على تصور التعديلات التي تم إجراؤها على تصميم المعالج أثناء مشكلة التحسين هي إحدى فوائدها. يتم استخدام الذراع الآلية النشطة كدراسة حالة في هذه المقالة. يتم تعريف المهمة على أنها اختيار أفضل مسار بناءً على موضع واتجاه هدف الإدخال وتستخدم في أنظمة روبوت الأشعة السينية. توضح النتائج جدوى النهج المقترح ويمكن أن تكون بمثابة نموذج أولي مفيد لروبوت ذكي للأشعة السينية.

الكلمات الدالة: التطبيقات الطبية الحيوية، الأداء المزدوج، معادلات الحركة العكسية، المسار الأمثل، مشكلة التحسين، الذراع الآلية DOF-6.

1.INTRODUCTION

Trajectory planning is cons indeed fundamental to controlling the robotic arm movements. The trajectory merits affect the completion of the task. At this time, trajectory planning impacts the robot's accuracy for trajectory track control as a prerequisite [1,2]. The trajectory planning affects motion efficiency, smoothness, and server consumption energy [3]. Planning is a problem-solving process in the modern artificial intelligence research context. The process starts with a specific problem and then constructing several operational tasks (operators) to achieve the objectives to solve that problem. The flow chart of trajectory planning in the missions' planners is shown in Fig. 1. For various on-orbit operations specifications, various trajectory planning operations must be conducted for the on-orbit robotics arms. The required trajectories must be planned based on the constraint of trajectory planning. The planning makes the robotic arms capable of reaching the selected positions to finish a particular objective [4,5]. Most scientists propose a hybrid robot and human joint activation as a future perspective. This work helps to take advantage of each of them. As an objective, robotics and automation play an important role in space explorations and ground-based experiments. The developments and evolutions of com-plex spacecraft systems and space exploration missions need the development of fast, intelligent automation systems and robotics prerequisites [6]. The robotic space tasks include some special space missions besides the conventional repair and assembly like planetary geological recon-naissance missions' orbital robots [7], and surgical robots for space mission [8].

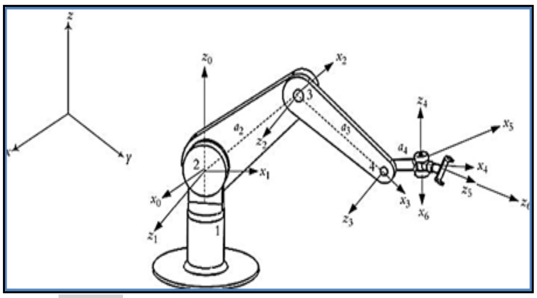


Fig.1 Trajectory Planning Diagram.

Robotic arms are frequently produced and sold globally as standard mechatronic items [9]. Numerous companies sell a vast array of firearms. Most of them are made to help with industrial manufacturing and assembly lines [10]. By mounting different tools on typical wrist joints (machining, spraying, and welding), those arms are employed in various applications where great repeatability and accuracy are necessary. Most studies on enhancing the functionality and design of control systems and mechanical systems for weapons that are now on the market are conducted by robotic firms and the laboratories that work with them [11]. They are significant weapons and vital testing grounds for research and study units into novel applications and intelligent control designs because of their durability and reliability. Thus, choosing a suitable controller and robot arm is an important research choice from the point of the researcher. Additionally, the degrees of freedom (DOF), payload, redundancy, the weight of weapons, and working space requirements are considered [12]. One must consider the (i) ability to add new sensors, (ii) control unit, (iii) possibility of combining existing and new software, and (iv) accessibility of a software emulator, after-sales service, etc., in this situation [13]. It is crucial to consider these factors early on before making a final purchasing decision to ensure future scalability. Six-degree-of-freedom (6-DOF) is highly sought-after in precision engineering, and robot trajectory planning is characterized as a collection of trajectory points depending on a set of projections and target positions [14]. Each robot joint's rotation angle is adjusted over time in a path-dependent manner to the end effector, then at each subsequent point until the target point is attained. Path planning in the joint space is simpler and more practical than in the Cartesian space [15]. The ends of multiple robotic arms are frequently given multiple fixed locations. Inverse Kinematics was used to construct such robot trajectory points, which were then converted from Cartesian space to a common coordination space [16]. The two types of motor shifts that are most frequently used are reverse and forward. The mapping regarding the robot's joint coordinates or its coupling to the position of the final effector is known as forward Kinematics. The location of the final effector in Cartesian space is difficult with the forward shift when the joint angles are known, whereas the inverse shift is just the reverse. The inverse kinematic transformation is frequently difficult to solve because of the considerable nonlinearity [17]. The decrease in energy use is viewed as one of the most significant goals in the industrial and manufacturing sectors for environmental and economic reasons [18]. Numerous energy-saving methods in this area were created over time, and it was shown that they could be applied to mechatronic and robotic systems [19]. Trajectory planning navigation can be classified according to the data received from the world. Three types of trajectory planning are available local, global, and personal navigation. Globally based navigations might perceive the environmental elements according to the referential axes and stir on the previously determined goals [20]. When it comes to Localized navigation techniques, we can say that it deals with anonymous and not fully known environments, whereas global navigations can address fully known environments. Localized navigation techniques are referred to as reactively-based techniques because they are smarter and can control and execute plans anonymously. Some of those intelligent techniques are Fuzzy Logic (FL), such as Evolutionary Algorithms (EGA), Artificial Neural Network (NN), Differential Evolutions (DE), Firefly Algorithm (FA), Cuckoo Search Algorithm (CSA) Swarm Intelligence (SI), Ant Colony Optimization (ACO), and others. Local search techniques illustrate higher elasticity. Moreover, such techniques follow an optimized path [21].

2.RELATED WORK

The conventional methods that use the polynomials interpolated trajectory planning technique might achieve effective object avoidance. In addition, such methods are considered robot arm trajectory selection technique that reasonably selects the movement. However, these techniques can fail to get the trajectory with the best performances. In the last few years, to get a feasible trajectory that fulfills the dynamical constraints and the Kinematics, studies have focused on applying smart techniques. Thus, such smart techniques have been applied to the space robotics arms for efficient trajectory planning. The Genetic Algorithm (GA) is one of the widely used evolutionary algorithms considered intelligent optimization algorithms. It has been used successfully in trajectory optimization solutions due to its efficiency in solving optimization problems [22]. The genetic Algorithm provides a generalized framework used to solve optimization issues for complex systems that does not depend on particular dimensions of the problem and have robustness. Muhammad et al. [23] instead of using the traditional Newtonian iteratively based methods the authors have introduced the GA in optimal control. Numeral algorithms for optimal controlling of movement planning were based on the GA optimization technique as was proposed. Simulation designs demonstrated the method effectiveness, and the motions planning problems of the non-complete constraints space robotics arms system were resolved. Seder and Petrovic [24] used the GA to handle the requirements of in bit trajectories planning of the space's robotics arm. They used GA fitness rate functions about the end trajectories length of Cartesian spaces robotic arms, the robotic arm angle in joint space, the max torque of the joints of motion, the entire motion latency of the robotics arms, and the collisions by the weight coefficients method. Last, the GA was used for planning an ideally based trajectory in jointed spaces without collisions, kinetic properties satisfy the marginal needs, and not long trajectories lengths and motions time. Zafar and Mohanta [25] the authors used optimization to enhance the trajectory based on energy conservation standards. Their suggested methods enable floating body operations in different platforms with manipulated joint failure. The enhanced-on board fuels remarkably affect the satellites' lifetime. Al-Mashhadany [26] the authors suggested new “feature-based” path optimization techniques for planetary explorations cars. In such a technique, they are discretizing the specified paths into segment parts and producing the energy-based data of these specified parts from the search environments maps [27]. As a result, GA was combined to search for the optimality in terms of path planner in conjunction with mesh environment maps utilizing cubic-shaped splines interpolations for searching the optimal and feasible solutions of the paths planner. Based on what has been mentioned earlier, it could be realized that even though the GA could be dealing with many different solutions simultaneously, minimizing risks of being trapped in optimal local areas. Nonetheless, the optimization algorithm is randomly based on searching techniques multiple calculations, and candidate solutions rendered are usually approximated solutions close the feasible solution. In addition, reliability might not be ensured. The various studies utilized to simulate and analyze the best trajectory for 6-DoF robotics using a VR model are listed in this section. Through interpolating several specified points of every joint of robot with a quintic polynomial function, mathematical expressions regarding every one of the joint variables of the robot with time are produced. To increase effectiveness of the search algorithm, the GA crossover operator and mutation operator of the Genetic Algorithm have been improved in the cosine form. The modified adaptive Evolutionary Algorithm (EA) is also employed to optimize time interval between each joint's interpolation points to achieve time-optimal trajectory planning. A MATLAB simulation is run, and results have shown that the approach recommended in this study cuts time required for robot tasks. Each joint curve for velocity, position, and acceleration is also suitably smooth, guaranteeing that it performs its function dependably and successfully [28]. Analyze the Kinematics of the robot arm, which is crucial for the motion of all robotic joints. Additionally, they are crucial for getting the signal to control or move the robot arm in the workplace. In this work, Lab VIEW will be used to implement the Kinematics regarding the ROB0036 DFROBOT arm. The kinematic equations of motion could be determined by determining the Denavit-Hartenberg representation's parameters, which resolves problem with the 6 revolute joints DFROBOT manipulator's automatic control. The kinematics solution of the Lab VIEW software was found to be nearest to the robot arms actual measurements that have been presented in [29]. Six linear motors were controlled using a control method depending on 6-DoF parallel robot kinematics to follow their respective intended trajectories under the guidance of a designed Fractional Order Active Disturbance Rejection Controller (FOADRC) [30]. The Denavit Hardenberg (DH) 6 DoF robot arm placement parametric technique is used in the forward kinematic model. Also, the forward kinematic model is investigated in MATLAB using Robotics Toolbox, whereas the inverse kinematic model is applied to an actual robot arm [31]. For each one of the robot joints, the quantic polynomial function is employed to satisfy several specified points, in which mathematical expressions are produced over time for each common robot variable. To accomplish the best path planning time, the optimized adaptive Genetic Algorithm enhances each joint's interval of interval points [32]. The robot's trajectory and real-time target are updated without further offline iterations. It extends to controllers of the linear quadratic regulator to circulate through globally or locally controllers to modify the robot's core path based on control optimization and gradient ratios [33]. Depending on the algorithms for Differential Evolution (DE) and Whale Optimization Algorithm (WOA), the Improved Whale Optimization Algorithm (IWOA) has been presented [34].

3.PROBLEM STATEMENT

Kinematic modeling issues normally can be split into a handful of sub-problems. First, direct Kinematics is a difficult problem due to determining the routing's Cartesian placement, which is given using standard coordinates. The second issue is the inversed movement, which is the Inverse Kinematics (IK) that represents a mathematical technique used to determine the joint configurations of a robotic or animated system that would result in a desired end-effector position or pose. It involves the calculation of covariates based on the orientation and location of the robot's end effector. The complexity of the second serial automatic weapon is greater than that of the first. Biomedical automation activities take advantage of the kinetics designs of robots because it permits them to be half-autonomous or entirely autonomous. Due to the operational environment and type of task, biomedical robots typically have several rigid connections mounted on platforms. As for a biomedical application, robot arms manipulating chemicals associated with 6-DoF are commonly utilized. Spot assembly spraying, welding, and fabrication are the most prevalent applications for biomedical robots. A considerable percentage of these apps must complete the selection and putting processes. Utilize the kinematics model of the robotic arms to complete this assignment. The kinematic issue compels us to define the Cartesian spaces for the co-teleports. Any robot manipulator agent can access two kinematic solutions: backward and forward Kinematics. Mobility forward can be determined after all joints are sensed. Suppose the selected locations and directions of the end stimuli are set. In that case, the total number of joints is calculated, and the joint spaces of the robotics arms are computed using the reversed kinematics motion in reversed motion.

4.METHODOLOGY

The various sorts of arm movement are the major topic of this study. Fig. 1 depicts the overall architectural layout and involves a processing unit and a robotic arm with six degrees of freedom (6-DoF). The Virtual Reality (VR) paradigm uses MATLAB as a development environment for software programs. Modeling in the Virtual Reality Modeling Language (VRML) utilizing an IK-based method to control a 6-DOF robotic arm. The whole processing unit, programs, and control modes files are recovered in MATLAB. The number of track points needed for the robot end effector in Cartesian space is converted by the inverse kinematic operation into the related joint variables, which are then interpolated to produce the robot joint variables as a function of a time-varying expression [35]. Eqs. (2)-(4) represent the placement and acceleration constraints.

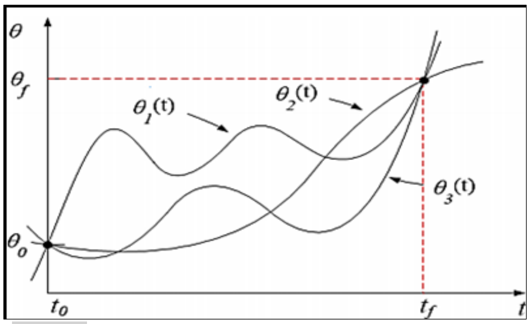


Fig. 2 Quantic Polynomial Function with the Same Beginning and Ending Points.

Find a smooth curve with o for the beginning and f for the ending. Additionally, Eq. (1) represents the quintic polynomial [33].

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_4 t^4 + a_5 t^5 \quad \textbf{(1)}$$

$$\begin{cases} \theta(0) = \theta_0 \\ \theta(tf) = \theta_f \end{cases} \quad \textbf{(2)}$$

$$\begin{cases} \dot{\theta}(0) = \dot{\theta}_0 \\ \dot{\theta}(tf) = \dot{\theta}(f) \end{cases} \quad \textbf{(3)}$$

$$\begin{cases} \ddot{\theta}(0) = \ddot{\theta} \\ \ddot{\theta}(tf) = \ddot{\theta}(f) \end{cases} \quad \textbf{(4)}$$

Eq. (5) calculates the robot's trajectory's velocity expression.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_4 t^4 + a_5 t^5 \quad \textbf{(5)}$$

Similarly, the acceleration function can be obtained using the second derivative of t in Eq. (6).

$$\dot{\theta}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad \textbf{(6)}$$

The second derivative of t in Eq. (6) could also be used to determine the acceleration function. Eqs. (2) - (4) above represent the quintic polynomial's coefficients.

$$\theta(0) = \theta_0 \quad \textbf{(7)}$$

$$a_1 = \dot{\theta_0} \quad \textbf{(8)}$$

$$a_2 = \frac{\ddot{\theta_0}}{2} \; n \quad \textbf{(9)}$$

$$a_2 = \frac{20f - 200\theta_0 - (12\dot{\theta}_0 + (8\dot{\theta}_f)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)r_f^2}{2t_f^3} \quad \textbf{(10)}$$

$$a_3 = \frac{30f - 30\theta_0 - (16\dot{\theta}_0 + (14\dot{\theta}_f)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)r_f^2}{2r_f^2} \quad \textbf{(11)}$$

$$a_4 = \frac{12f - 12\theta_0 - \left(6\dot{\theta}_0 + (6\dot{\theta}_f)\right)tf - (\ddot{\theta}_0 - \ddot{\theta}_f)\Big)_f^2}{2t_f^3} \quad \textbf{(12)}$$

Plugging the components as mentioned above into Eq. (3) allows one to construct the robot trajectory equation for the quantic polynomial.

5.MODELLING OF 6-DOF ROBOT

5.1.Inverse Kinematic Model

IK model has a wide range of applications in real robotic systems. The IK model calculates the joint angles required to achieve the desired orientation and position. IK is crucial not only in robotics, but in various fields, including 3D gaming [36]. IK does not have a single solution, in contrast to forwarding kinematics. The best systems guarantee minimal joint motion and collision-free operation. If the end position, i.e., Py, Px, and Pz, and the link parameters, are known, inverse Kinematics could be utilized to find all the joint angles. For a single end-effector location, inverse Kinematics will produce a significant number of joint angles [37]. Eqs. (13)–(24) are utilized for calculating the Inverse Kinematics.

$$T_{1-1} \times T = T_2 \times T_3 \times T_4 \times T_5 \times T_6 \quad \textbf{(13)}$$

$$\theta_1 = \text{atan2d}(p_y, p_x) \quad \textbf{(14)}$$

$$d = \sqrt{p_x^2 + p_y^2} \quad \textbf{(15)}$$

$$r_4 = d - a_4 \times \text{cosd}(\theta_{234}) \quad \textbf{(16)}$$

$$z4 = pz - a4 * sind(\theta 234) \quad \textbf{(17)}$$

$$\theta 3 = acosd\left(\frac{s^2 - {a^2}^2 - {a^2}^3}{2 * a2 * a3}\right) \quad \textbf{(18)}$$

$$\beta = \text{atan2d}(a_3 \times \text{sind}(\theta_3), a_2 + a_3 \times \text{cosd}(\theta_3) \quad \textbf{(19)}$$

$$\alpha = \text{atan2d}(z_4 - a_1, r_4) \quad \textbf{(20)}$$

$$\theta_2 = \alpha + \beta \quad \textbf{(21)}$$

$$\theta_4 = \theta_{234-}\theta_2 - \theta_3 \quad \textbf{(22)}$$

$$\theta_5 = \text{acosd}\left((\text{sind}(\theta_1) \times p_x) - \text{cosd}(\theta_1) \times p_y\right)/(\text{sind}(\theta_4) \times a_5)) \quad \textbf{(23)}$$

$$\theta_6 = \text{atand}(\text{cosd}(\theta_5) \times \text{atan}\left(\text{cosd}(\theta_5) \times \text{cosd}(\theta_4) \times \text{sind}(\theta_2 + \theta_3) + \text{sind}(\theta_5) \times \text{cosd}(\theta_2 + \theta_3) - \text{sind}(\theta_4) \times \frac{\text{cosd}(\theta_5)}{\text{cosd}(\theta_5) + \text{sind}(\theta_4) + \text{sind}(\theta_2 + \theta_3)}\right) \quad \textbf{(24)}$$

5.2.Forward Kinematics

The kinematic problem of a robot could be investigated using various methods. Denavit-Hartenberg (DH) PA screw displacements are applied in two widely utilized methods. Due to their systematic nature, both approaches are more suitable for imitating serial manipulators. Several researchers also employ geometric techniques for serial manipulators with rather basic geometry [38]. The DH method was used in this work to build the robot's kinematic model due to its suitability and adaptability for modelling any number of links and joints of a serial manipulator, independent of complexity [39, 40]. DH uses the quadruple (αi-1, ai-1, d, θ) notation to represent twist angle, link offset, link length, and joint angle. Each manipulator link was connected to an orthonormal coordinate system following DH standard [41], as shown in Fig. 3.

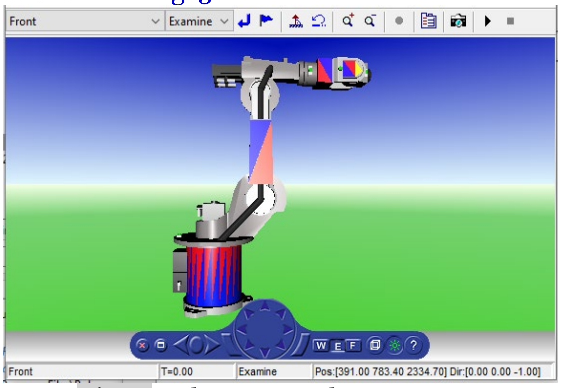


Fig. 3 Robot Forward Kinematics.

The necessary transformation matrices for each linkage regarding the robotic arm were created using the general shape of the transformation matrix for each link expression of the I joint in the preceding proximal i-1 joint acquired in [42]. Those transformation matrices separate when multiplied due to the complex transformation feature, producing an overall matrix reflecting the base's end terms [43]. The last column of the 3X3 matrix represents the location (x, y, z) of the end-effector about the base, whereas the first three columns and first three rows reflect the rotation [44].

$$^{0}_{6}T = \begin{bmatrix} C_1C_5S_{234}+S_1S_5 & -C_1S_{234}S_5+S_1C_5 & C_1C_{234} & C_1A \\ -S_1C_5C_{234}-C_1S_5 & S_1C_{234}C_5+C_1C_5 & S_1C_{234} & S_1A \\ C_{234}C_5 & -C_{234}S_5 & -S_{234} & B \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(25)}$$

Where

$$A = L_2S_{2+}L_3S_{23+}L_4C_{234} \quad \textbf{(26)}$$

$$B = L_1 + L_2C_{2+}\, L_3C_{23-}\, L_4S_{234} \quad \textbf{(27)}$$

$$\begin{bmatrix} Q1 \\ Q2 \\ Q3 \\ Q4 \\ Q6 \end{bmatrix} = \begin{bmatrix} px \\ py \\ pz \\ \alpha \\ \beta \\ \Upsilon \end{bmatrix} \quad \textbf{(28)}$$

Implementation of IK Model: A genuine manipulator equipped with a robotic arm was used to evaluate the IK model. An object has been given a defined orientation and location [45]. Using the user's known information, the generated procedure, as demonstrated in IK [46], first determines whether the object is inside the robot workspace. When the object is outside the work envelope, the Algorithm informs the user and stops. The IK model determines the necessary joint angles to point the end-effector in the proper orientation and position if it is not pointed in the needed orientation and position [47]. The joint angles are then allocated to the low-level encoder ticks after that. The program's Kernel-based instructions move the motors depending on the mapped instructions encoder ticks, then carry out the command. Fig. 4 shows the model implementation flowchart [48]. Before using the created IK model on robotic arm, it was ensured that the object (for instance, a vehicle key with a key) is positioned inside the workspace of the robotic arm. Two blocks are used to raise the platform on which the object is placed (in height z). The work at hand is moving an object from one area to another. Orientation and position of the destination and source have both been given as inputs. The robot moves depending on approximated joint angles derived from the user's object coordinates (IK model) [49]. The gripper shuts when the robot is in the desired position, enabling it to capture the object. The user has also taught the robot the object's coordinates by picking up the objects in a specific order. The operational area of the robot should also contain the target site. When the robot gets there, it drops the item and goes back to where it started [49]. Using inverse kinematics, convert the specified target position to the corresponding joint angles or positions for the robot arm. This step involves solving the inverse kinematics problem to determine the arm configuration that reaches the desired target. Below are the steps to follow to get the best results:

- **Step 1**: Define the target point (position) represents center point of working space as x,y and z.
- **Step 2**: Define the ratio of noise scale (0.001)
- **Step 3**: Generates number of random angles (number of angles set equal to possible number of points around target) as angles.
- **Step 4**: Define rotation matrix around Z for each angle as in Eq. (29).

$$Rot_z = angle * \begin{bmatrix} \cos(angle) & -\sin(angle) & 0 \\ \sin(angle) & \cos(angle) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(29)}$$

- **Step 5**: Final equation Eq. (30)

$$X = \text{noisescale} * (\text{rand}(1,3) - 0.5 * (\text{Rot}_x * \text{Rot}_x) + \text{center point} \quad \textbf{(30)}$$

- **Step 6**: Find nearest point to center point (xc, yc, and zc).

$$[\text{center}, \text{radius}] = \text{sum}((X(1) - xc)^2 + (X(2) - yc)^2 + X(3) - zc)^2 - r^2)^2 \quad \textbf{(31)}$$

The represents of generated points of dual performance method in 3D view within spherical space and around center point (target).

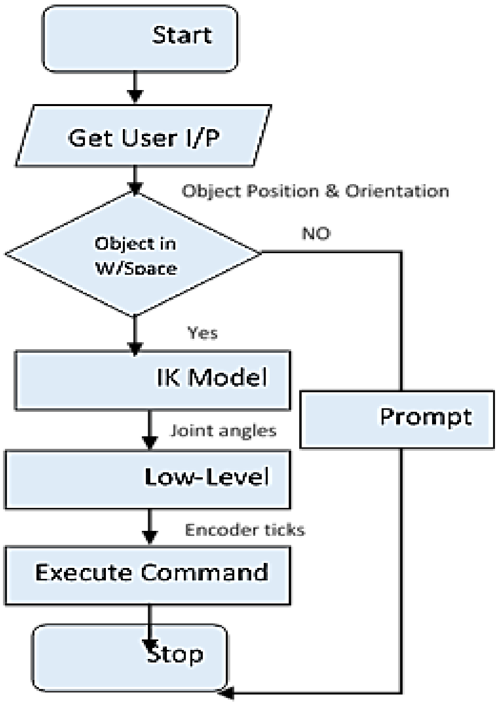


Fig. 4 Coordinate Assignment of DH Parameter.

6.SIMULATION RESULTS AND DISCUSSION

In this section, several trajectories are tested to achieve the target. To attain the target i.e. to let the arm reach to the exact point, it is necessary to determine the target position and the target orientation according to some criteria such as the arm's length, the movement angle that is subject to the set points of the arm which are the shoulder, elbow, and the wrist points. In addition, the grabber's length and flexibility are considered one of the most important constraints determining the arm's movement.

6.1.Trajectory Analysis Test

For adjusting and moving the robotic arms to the specified coordinate of angles and placed in between ones and the targeted objects, make the angles of the shift location could precisely allocate the directions of the entire axis of the robot's arms and obtain the best movement trajectory. In Table 1, one can observe the orientation error values Err_ert 3, Err_ert 2, and Err ert 1 of each coordinate, which are minimal in all the cases considered. Thus, it can be highlighted that the accuracy of the arm's movement toward the target is excellent, taking into account the error values of each of the three axes.

Table 1 Trajectory Analysis Test.

Err_ert 3	Err_ert 2	Err_ert 1	Error Z	Error Y	Error X
150.3937	-0.00602	-0.31617	0.055045	0.082833	0.002137
-0.00602	-0.44932	-0.31617	0.079884	0.053205	0.026659
-0.11289	-0.44932	-0.31617	0.099652	0.00304	0.001654
-0.11289	-0.44932	-0.31617	0.099358	0.003846	0.000532
-0.11289	-0.44932	-0.31617	0.006206	0.089897	0.001431
-0.11289	-0.44932	-0.31617	0.029801	0.089297	0.006015
-0.11289	-0.44932	-0.31617	0.095763	0.006295	0.006329
-0.11289	-0.44932	-0.31617	0.079201	0.047268	0.0263
-0.11289	-0.44932	-0.31617	0.096849	0.004541	0.000562

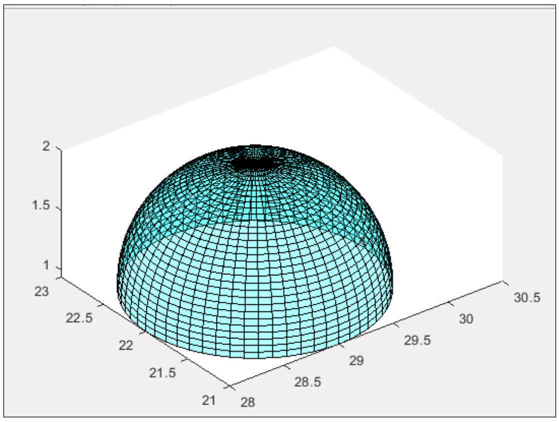
6.2.Target Orientation

The proposed robot arm must be distanced from the target by generating points at a specific distance in different orientations. The proposed method is generating points surrounding the target in a sphere shape. The suggested optimization algorithm would look for the shortest path between the robot arm and the target. Accuracy is proportional the distance between both points. When the distance between both points becomes less, the accuracy would become higher. The orientations are characterized in this context by XYZ position targets as shown in Table 2. The postures were chosen to be with in the robot workspaces. The test outcomes are illustrated in the Fig. 3 in all three orientations. The arm generated intermediary targets to trace the required trajectory. The robot arm reached target postures accurately. The test results approved that the local information of the robots is enough to produce a local model that could be utilized for actuating the terminal effectors towards the desired targets. In the Table 2, it can be noticed that the least distance is considered the best in terms of the accuracy. The least is obtained from the point vector (35, 20, 22.36069) which is the nearest to the target that equals (10, 20, 35). By using Euler equations (Eqs. 32 and 33) several points are generated near to the target point, the distance value is 10 units away from the target. The nearest point is selected as the best point.

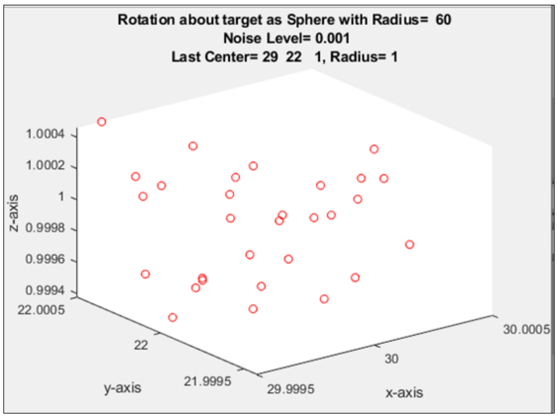
$$rx = \begin{bmatrix} cos\theta & \sin\theta & 1 \\ 1 & cos\theta & \sin\theta \\ 0 & 0 & 0 \end{bmatrix} \quad \textbf{(32)}$$

$$rz = \begin{bmatrix} \sin\theta & cos\theta & 0 \\ -\sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(33)}$$

It can be noticed from Fig. 5 (a) that the sphere dimensions are 10 for the radius while the noise level equals to 0.001, and the last centre (target) is 10, 20, 35. As for Fig. 5 (b) the number of the nearest points generated around the last centre (target) is 60. These points are all supposed to be distributed on the sphere shape shown in Fig. 5 (a).



(a)



(b)

Fig. 5 (a) The Domain where the Generated Euler Points are Distributed. (b) 60 Nearest Points Generated on the Sphere Shape Surrounding the Last Centre (Target).

6.3.Optimal Trajectory Finding

After finding the nearest point to the target, the forward and inverse kinematics are used to find the shortest trajectory distance representing the optimal distance. It can be noticed from Table 2 that we have the target points that are 9 points. It can be noticed that target number 8 is the best one in comparison to the other numbers of targets based on the obtained distance. The distance was 9.1376, which is the least among the others. We note that Fig.6 shows the best path obtained from the resulting paths, which is the eighth path, as in the above, and it is closest to the target point that was previously determined.

Table 2 Target Orientation Test.

Target number	POS target z	POS target y	POS target x	Distance
1	35	20	10	11.4571
2	35	20	20	9.4448
3	35	20	0	11.8618
4	35	30	10	12.0027
5	35	10	10	10.597
6	45	20	10	11.3893
7	25	20	10	9.7101
8	35	20	22.36068	9.1376
9	35	36.4005	10	12.8338

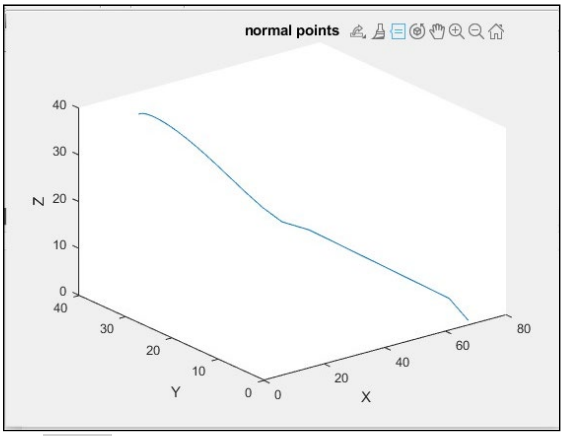


Fig.6 The Trajectory of the Best Target.

7.CONCLUSIONS

This research presents a novel method for the kinematics optimization of robot arms using dual performance and a virtual point resolution algorithm created to control complex trajectory robotics. The advantages of this approach over other analytical inverse kinematics methods include (1) the ability to visually monitor changes in the robot's configuration during optimization, (2) the ability to focus optimization on a subset of the robot manipulator's structural parameters, and (3) the fact that during the optimization process, design specifications are continuously modified until they attempt to reach near the optimum performance and no randomness exists. The designer can then feel the consequences of the desired performance index and the chosen structural variable for optimization. In this method, finding the best solutions for different design characteristics is possible by adding them as virtual joints in a complex trajectory robot that has been virtually built. Inverse kinematics solution approaches are used to modify these variables in the null space of the dual performance so that they do not affect the primary task or design constraint(s). However, the alteration directly impacts the chosen subtask, which is the goal of the optimization process. Thus, by adding these design specifications as virtual points of a virtually created complex trajectory robot, the design parameters are calculated following the chosen objective function that incorporates the chosen performance indices. The design parameters are modified according to the chosen objective function, which considers the manipulator performance indices.

ACKNOWLEDGEMENTS

The authors are grateful for support towards this research by the University of Anbar / College of Computer Science and Information Technology / Department of Computer Science.

NOMENCLATURE

a2	Represents an unknown or calculated acceleration.
f	Represents a constant or variable.
t_f	Represents a time variable
T_f	Represents a final time value
Greek Symbols	
θ_o	*Represents an initial angle or position*
$\dot{\theta}_o$	*Represents the initial angular velocity (first derivative of with respect to time).*
$\dot{\theta}_f$	*Represents the final angular velocity (first derivative of with respect to time).*
$\ddot{\theta}_o$	*Represents the initial angular acceleration (second derivative of with respect to time).*
$\ddot{\theta}_f$	*Represents the final angular acceleration (second derivative of with respect to time).*

REFERENCES

[1] Zhu S, Wang. **Time-Optimal and Jerk-Continuous Trajectory Planning Algorithm for Manipulators**. *Journal of Mechanical Engineering* 2010; **46**(3) :456–62.

[2] Al Mashhadany Y, Gaeid KS, Awsaj MK. **Intelligent Controller for 7-DOF Manipulator Based upon Virtual Reality Model**. *12th International Conference on Developments in eSystems Engineering (DeSE)* 2019; Kazan, Russia. IEEE: p. 687-692.

[3] Xu D, Liu M, Zhu L. **Single Frequency GNSS Integer Ambiguity Resolution with Adaptive Genetic Algorithm**. *International Conference on Information Science and Technology 2013*; Yangzhou, China. IEEE: p. 1049–1051.

[4] Al Mashhadany YI. **Virtual Reality Trajectory of Modified PUMA 560 by Hybrid Intelligent Controller**. *Bulletin of Electrical Engineering and Informatics* (2020); **9**(6): 2261-2269.

[5] Ying-Ying YU, Yan C, Tao-Ying LI. **Improved Genetic Algorithm for Solving TSP.** *Control and Decision* 2014; **29**(8):1483–8.

[6] Ren ZW, San Y. **Improved Adaptive Genetic Algorithm and its Application Research in Parameter Identification**. *Journal of System Simulation* 2006;**18**(1):41–43.

[7] Yingjie L, Shanwen Z, Xuwu L et al. (2005). **MATLAB Genetic Algorithm Toolbox and its Application**. Xi'an Electronic and Science University press, Xi'an (In Chinese).

[8] Ruggiero F, Lippiello V, Ollero A. **Aerial Manipulation: A Literature Review**. *IEEE Robotics and Automation Letters* 2018; **3**(3): 1957-1964.

[9] Sancak KV, Bayraktaroglu ZY. **Nonlinear Computed Torque Control of 6-Dof Parallel Manipulators**. *International Journal of Control, Automation and Systems* 2022; **20**(7): 2297-2311.

[10] Cheng C, Lv X, Zhang J, Zhang M. **Robot Arm Path Planning Based on Improved RRT Algorithm**. *3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT) 2021;* Changzhou, China: p. 243-247.

[11] Zhang Z, Chen Z. **Modeling and Control of Robotic Manipulators Based on Symbolic Regression**. *IEEE Transactions on Neural Networks and Learning Systems* 2023; **34**(5): 2440-2450.

[12] Jun-lin M, Ying L, Ying-kun Z, Cunming H. **Design of 3DOF Delta Parallel Capture Robot with High Speed and Light Weight**. *Journal of Physics: Conference Series* 2022; **2188**(1): 012008, (1-9).

[13] Shen N, Yuan H, Li J, Wang Z, Lu N, Lu Y. **Dynamic Modeling and Simulation of a Hybrid Robot**. *Journal of Mechanisms and Robotics* 2023; **15**(1): 011012, (1-15).

[14] Xia Y, Yang M, Ji Z, Jia L, Zhou Z, Wu D. **Design and Dimension Optimization of Cutter Disassembly Mechanism for Shield Tunneling Machine**. *Journal of Mechanical Science and Technology* 2021; **35**(7): 3005-3018.

[15] Al Mashhadany YI. **Scara robot: Modeled, simulated, and virtual-Reality Verified**. *In International Conference on Intelligent Robotics, Automation, and Manufacturing* (pp. 94-102). Berlin, Heidelberg: Springer Berlin Heidelberg. 2012.

[16] Yorozu T, Hirano M, Oka K, Tagawa Y. **Electron Spectroscopy Studies on Magneto-Optical Media and Plastic Substrate Interface**. *IEEE Translation Journal on Magnetics in Japan* 1987; **2**(8): 740-741.

[17] Al-Mashhadany YI. **Design and Analysis of 7-DOF Human-Link Manipulator Based on Hybrid Intelligent Controller**. *Информатика и автоматизация* 2022; **19**(4): 774-802.

[18] Hosseininejad S, Dadkhah C. **Mobile Robot Path Planning in Dynamic Environment Based on Cuckoo Optimization Algorithm.** *International Journal of Advanced Robotic Systems* 2019; **16**(2): 1729881419839575, (1-13).

[19] Kim YJ, Park SW, Yeom HG, Bang MS, Kim JS, Chung CK, Kim S. **A Study on a Robot Arm Driven by Three-Dimensional Trajectories Predicted from Non-Invasive Neural Signals**. *Biomedical Engineering Online* 2015; **14**(1): 1-19.

[20] Suchi M, Bader M, Vincze M. **Meta-Heuristic Search Strategies for**

Local Path-Planning to Find Collision Free Trajectories. *Proceedings of the Austrian Robotics Workshop (ARW-15) 2015*; Graz. Austria. Austrian Robotics Workshop (ARW): p. 1-8.

[21] El Haiek D, Aboulissane B, El Bakkali L, El Bahaoui J. **Optimal Trajectory Planning for Spherical Robot Using Evolutionary Algorithms**. *Procedia Manufacturing* 2019; **32**: 960-968.

[22] Patle BK, Pandey A, Parhi DRK, Jagadeesh AJDT. **A review: On Path Planning Strategies for Navigation of Mobile Robot.** *Defence Technology* 2019; **15**(4): 582-606.

[23] Muhammad A, Ali MA, Shanono IH. **Path Planning Methods for Mobile Robots: A Systematic and Bibliometric Review**. *Elektronika* 2020; **19**(3): 14-34.

[24] Seder M, Petrovic I. **Dynamic Window-Based Approach to Mobile Robot Motion Control in the Presence of Moving Obstacles**. *Proceeding of the IEEE International Conference on Robotics and Automation 2007*; Rome, Italy. IEEE: p. 1986-1991.

[25] Zafar MN, Mohanta JC. **Methodology for Path Planning and Optimization of Mobile Robots: A Review**. *Procedia Computer Science* 2018; **133**: 141-152.

[26] Al-Mashhadany YI. **A Posture of 6-DOF Manipulator by Locally Recurrent Neural Networks (LRNNs) Implement in Virtual Reality**. *IEEE Symposium on Industrial Electronics and Applications (ISIEA) 2010*; Penang, Malaysia. IEEE: p. 573-578.

[27] Qi RL, Zhou WJ, Wang TJ. **A Genetic Algorithm-Based Trajectory Planning Method for Spatial Robotic Arm Obstacle Avoidance**. *Robotics* 2014; **36**: 263–270.

[28] Shrivastava A, Dalla VK. **Failure Control and Energy Optimization of Multi-Axes Space Manipulator Through Genetic Algorithm Approach**. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 2021; **43**: 1-17.

[29] Fallah S, Yue B, Vahid-Araghi O, Khajepour A. **Energy Management of Planetary Rovers Using a Fast Feature-Based Path Planning and Hardware-in-the-Loop Experiments**. *IEEE Transactions on Vehicular Technology* 2013; **62**(6): 2389-2401.

[30] Chen Y, Chen S, Zhang T, Zhang S, Zheng N. **Autonomous Vehicle Testing and Validation Platform: Integrated Simulation System with Hardware in the Loop**. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 949-956). IEEE.

[31] Liu RJ, Wang F, Zhang Q, Nan L. **Research on Trajectory Planning of ROS-based Robot Arm**. *Navig. Position. Timing* 2016; **3**: 82–88.

[32] Kennedy J, Eberhart R. **Particle Swarm Optimization**. *Proceedings of ICNN'95-International Conference on Neural Networks 1995*; Perth, WA, Australia. IEEE: vol.4. p. 1942-1948.

[33] Jie DY, Lu HR, Wu HL, Ni FL. **Transporting Trajectory Optimization Method for Large Space Manipulator System**. *Acta Aeronautica et Astronautica Sinica* 2018; **39**: 111–119.

[34] Liu Y, Jia QX, Chen G. **Multi-Objective Particle Swarm Optimization Algorithm Based on Load-Maximizing Trajectory Optimization for Free-Floating Space Robots**. *Robotics*; 2014; **36**, 9.

[35] Xia HW, Zhai YB, Ma GC, Deng Y. **Spatial Robotic Arm Trajectory Planning Algorithm Based on Chaotic Particle Swarm Optimization Algorithm**. *Chinese Journal of Intelligent Science and Technology* 2014; **6**.

[36] Xu W, Liu Y, Liang B, Xu Y, Qiang W. **Autonomous Path Planning and Experiment Study of Free-Floating Space Robot for Target Capturing**. *Journal of Intelligent and Robotic Systems* 2008; **51**: 303-331.

[37] Liu Y, Du Z, Wu Z, Liu F, Li X. **Multiobjective Preimpact Trajectory Planning of Space Manipulator for Self-Assembling a Heavy Payload**. *International Journal of Advanced Robotic Systems* 2021; **18**(1): 1729881421990285, (1-26).

[38] Zhang J, Meng Q, Feng X, Shen H. **A 6-DOF Robot-Time Optimal Trajectory Planning Based on an Improved Genetic Algorithm**. *Robotics and Biomimetics* 2018; **5**(1): 1-7.

[39] Losey DP, O'Malley MK. **Learning the Correct Robot Trajectory in Real-Time from Physical Human Interactions**. *ACM Transactions on Human-Robot Interaction* 2019; **9**(1): 1-19.

[40] Wang T, Xin Z, Miao H, Zhang H, Chen Z, Du Y. **Optimal Trajectory Planning of Grinding Robot Based on Improved Whale Optimization Algorithm.** *Mathematical Problems in Engineering* 2020; **2020**:3424313, (1-8).

[41] Li Z, Yang C, Burdet E. **Guest Editorial an Overview of Biomedical Robotics and Bio-Mechatronics Systems and**

Applications. *IEEE Transactions on Systems, Man, And Cybernetics: Systems* 2016; **46**(7): 869-874.

[42] Gunantara N, Nurweda Putra I. **The Characteristics of Metaheuristic Method in Selection of Path Pairs on Multicriteria Ad Hoc Networks.** *Journal of Computer Networks and Communications* 2019; **2019**:7983583, (1-6).

[43] Reshamwala A, Vinchurkar DP. **Robot Path Planning Using an Ant Colony Optimization Approach: A Survey**. *International Journal of Advanced Research in Artificial Intelligence* 2013; **2**(3): 65-71.

[44] Korayem MH, Nazemizadeh M, Azimirad V. **Optimal Trajectory Planning of Wheeled Mobile Manipulators in Cluttered Environments Using Potential Functions**. *Scientia Iranica* 2011; **18**(5): 1138-1147.

[45] Llopis-Albert C, Rubio F, Valero F. **Optimization Approaches for Robot Trajectory Planning**. *Multidisciplinary Journal for Education, Social and Technological Sciences* 2018; **5**(1): 1-16.

[46] Benotsmane R, Dudás L, Kovács G. **Trajectory Optimization of Industrial Robot Arms Using a Newly Elaborated "Whip-Lashing" Method**. *Applied Sciences* 2020; **10**(23): 8666.

[47] Doan NCN, Lin W. **Optimal Robot Placement with Consideration of Redundancy Problem for Wrist-Partitioned 6R Articulated Robots**. *Robotics and Computer-Integrated Manufacturing* 2017; **48**: 233-242.

[48] Chicco G, Mazza A. **Metaheuristic Optimization of Power and Energy Systems: Underlying Principles and Main Issues of the 'Rush to Heuristics'**. *Energies* 2020; **13**(19): 5097, (1-38).

[49] Kim J, Kim SR, Kim SJ, Kim DH. **A Practical Approach for Minimum-Time Trajectory Planning for Industrial Robots**. *Industrial Robot: An International Journal* 2010; **37**(1): 51-61.