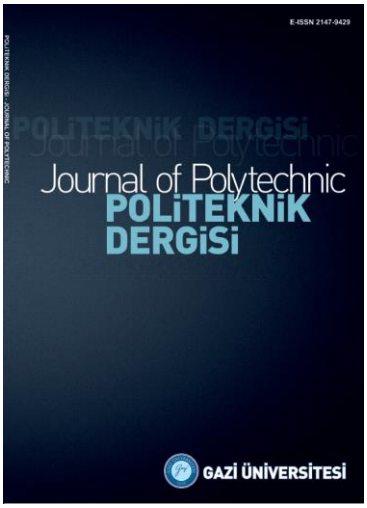

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

A comparison of multi-layer perceptron and inverse kinematic for RRR robotic arm

RRR robotik kol için çok katmanlı algılayıcı ve ters kinematik karşılaştırması

***Yazar(lar) (Author(s))**: Faruk Emre AYSAL[1], İbrahim ÇELİK[2], Enes CENGİZ[3] , Yüksel OĞUZ[4]*

ORCID[1]: 0000-0002-9514-1425

ORCID[2]: 0000-0002-8857-1910

ORCID[3]: 0000-0003-1127-2194

ORCID[4]: 0000-0002-5233-151X

To cite to this article: Aysal F,E.,Çelik İ., Cengiz E. ve Oğuz Y., "A comparison of multi-layer perceptron and ınverse kinematic for RRR robotic arm", ***Journal Of Polytechnic***, 27(1): 121-131, (2024).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Aysal F,E.,Çelik İ., Cengiz E. ve Oğuz Y., "A comparison of multi-layer perceptron and ınverse kinematic for RRR robotic arm", ***Politeknik Dergisi***, 27(1): 121-131, (2024).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1092642

A Comparison of Multi-Layer Perceptron and Inverse Kinematic for RRR Robotic Arm

Highlights

- *A Comparison of Multi-Layer Perceptron that is a neural network approach and Inverse Kinematic.*
- *The simulation of RRR type robotic arm with MLP model.*
- *Detailed explanation of inverse kinematic for robotic arms.*

Graphical Abstract

Kinematic calculations were made for the designed robot arm. Inverse kinematic analysis and MLP were used as two different approaches for the coordinate-joint calculations of the robot arm. The obtained results were compared with the three-dimensional simulation performed in Matlab environment. The general flow diagram of this study is given in the following figure.

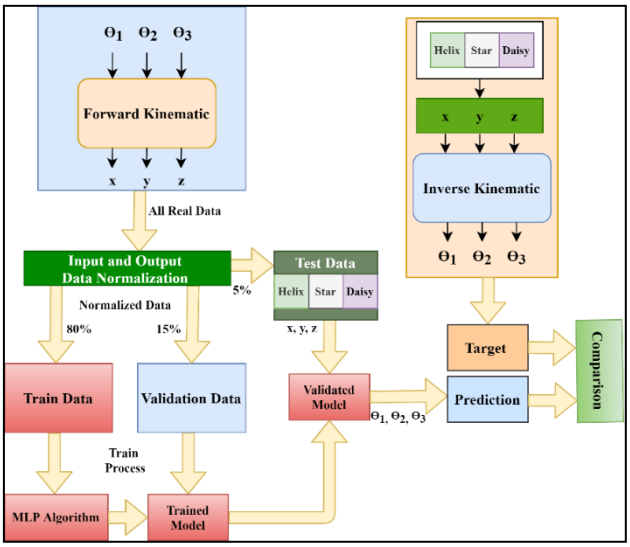

Figure. Flow chart of the study

Aim

The main purpose of this study is; to compare the inverse kinematics method in the kinematic analysis of robot arms and MLP, which is a common machine learning method, in a simulation environment in detail. Thus, it is aimed to bring a study that engineering students can access on an important subject such as inverse kinematic analysis to the literature.

Design & Methodology

The inverse kinematic analysis of a robot arm with three degrees of freedom RRR design has been made. As an alternative to inverse kinematics, the kinematic analysis of the robot arm was performed using the MLP machine learning algorithm.

Originality

Detailed explanation of geometric-analytic hybrid approach for inverse kinematic is performed. Also, an alternative algorithm for position control of robotic arms is proposed by using MLP with high accuracy.

Findings

Mean Relative Error (MRE) values for helix, star and daisy shapes were calculated as 0.0007, 0.0033 and 0.0011, respectively, in the tests performed.

Conclusion

Simulation results confirms that the proposed MLP model can operate this system at the desired stability.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission

A Comparison of Multi-Layer Perceptron and Inverse Kinematic for RRR Robotic Arm

Araştırma Makalesi / Research Article

Faruk Emre Aysal[1], İbrahim Çelik[1*], Enes Cengiz[2], Yüksel Oğuz[3]
[1]Afyon Kocatepe University, Mechatronics Engineering
[2]Sinop University, Electronic and Automation Department
[3]Afyon Kocatepe University, Electrical and Electronics Engineering

(Geliş/Received : 06.04.2022 ; Kabul/Accepted : 10.05.2022 ; Erken Görünüm/Early View : 27.06.2022)

ABSTRACT

In this study, the position control simulation of a 3 Degree of Freedom (3DOF) robot arm was compared with machine learning and inverse kinematic analysis separately. The considered robot arm is designed in RRR pattern. In the inverse kinematic analysis of the robot arm, the geometric approach and the analytical approach are used together. Multi-Layer Perceptron (MLP) was used as a machine learning method. Some of the coordinate data that the robot arm can reach in the working space are selected and the MLP model is trained with these data. When training was done with MLP machine learning method, the correlation coefficient (R^2) was obtained as 1. Coordinates of 3 different geometric models (helix, star and daisy) that can be included in the working space are used as test data of the MLP model. These tests are simulated in 3D in MATLAB environment. The simulation results were compared with the inverse kinematics analysis data. As a result, Mean Relative Error (MRE) values for helix, star and daisy shapes were calculated as 0.0007, 0.0033 and 0.0011, respectively, in the tests performed. Mean Squared Error (MSE) values were obtained as 0.0034, 0.0065 and 0.0040, respectively. This confirms that the proposed MLP model can operate this system at the desired stability.

Keywords: Inverse kinematics, multi-layer perceptron (MLP), robotic arm.

RRR Robotik Kol için Çok Katmanlı Algılayıcı ve Ters Kinematik Karşılaştırması

ÖZ

Bu çalışmada 3 DOF bir robot kolunun pozisyon kontrol simülasyonu makine öğrenmesi ve ters kinematik analiz ile ayrı ayrı yapılarak karşılaştırılmıştır. Ele alınan robot kol, RRR düzeninde tasarlanmıştır. Robot kolun ters kinematik analizinde geometrik yaklaşım ve analitik yaklaşım birlikte kullanılmıştır. Makine öğrenmesi yöntemi olarak Multi-Layer Perceptron(MLP) kullanılmıştır. Robot kolun çalışma uzayında ulaşabileceği koordinat verilerinin bir kısmı seçilerek, bu verilerle MLP modeli eğitilmiştir. MLP makine öğrenmesi yöntemiyle eğitim yapıldığında korelasyon katsayısı(R2) 1 olarak elde edilmiştir. Çalışma uzayı içerisinde yer alabilecek olan 3 farklı geometrik modelin (helix, star ve daisy) koordinatları MLP modelinin test verisi olarak kullanılmıştır. Bu testler MATLAB ortamında 3d olarak simule edilmiştir. Simülasyon sonuçları test kinematik analiz verileri ile karşılaştırılmıştır. Sonuç olarak gerçekleştirilen testlerde helix, star ve daisy şekilleri için Mean Relative Error (MRE) değerleri sırasıyla 0.0007, 0.0033 ve 0.0011 olarak hesaplanmıştır. Mean Squared Error (MSE) değerleri ise sırasıyla 0.0034, 0.0065 ve 0.0040 olarak elde edilmiştir. Bu da önerilen MLP modelinin bu sistemi istenilen kararlılıkta çalıştırabileceğini doğrulamaktadır.

Anahtar Kelimeler: Ters kinematik, çok katmanlı algılayıcı, robot kol.

1. INTRODUCTION

Robot arms are a type of robot used in industry for various industrial processes. Accordingly, the importance of robot arms in various industrial branches and medical applications is increasing day by day. Thus, by reducing the human factor in the studies carried out, more stable, faster and more sensitive processes can be performed [1]. In addition to the rapid development of semiconductor materials, developme nts in the production of micro and nano-sized machines also are increased the importance of studies in the field of robotics. Besides, the focus on robotics studies has increased cause of the increasing applications with high precision assembly [2,3].

A robot arm must be able to go directly to a specified coordinate location. This process can be done manually or automatically, and robot arms can be used statically or mobile [4]. One of the most important problems in the programming of robots is the inverse kinematics problem, which finds the values of the joint angles θ_1:θ_n, which allows the robot arm to reach the desired X, Y, Z coordinates with a certain orientation. This method converts the target coordinates into the angular motion of each robot joint. Various approaches such as Denavit-Hartenberg (D-H) [5], Screw Theory and Iterative Methods [6] are used to solve the inverse kinematics problem. In addition, instead of solving inverse

*Sorumlu Yazar (Corresponding Author)
e-posta : ibrahimcelik@aku.edu.tr

kinematics equations, statistical approaches called soft computation methods in the literature can be used [7,8]. Basically, some of these methods are used to find the best solution for calculating joint angles based on target coordinates. Many theoretical simulation studies have been carried out on inverse kinematics applications of robot arms with different degrees of freedom [9-13]. On the other hand, some researchers have developed a low-cost and small-scale prototype robot arm and applied inverse kinematics results [14-25]. However, in other studies, with the emergence of Internet of Things (IoT) technology, research that allows the robot to work through internet connection has begun to be discussed. Especially considering the development of IoT technology in the industrial sector, there is a great need for robot arm control algorithms that can act adaptively in accordance with this system [26-29].

The application of machine learning techniques to obtain the inverse kinematics solution of a given system has been widely discussed by researchers over the past two decades. Karlik et al tried to find the best Artificial Neural Network (ANN) configuration to solve the inverse kinematics problem of the 6-DOF robotic arm [30]. On the other hand, Chiddarwar and Babu compared the Radial Basis Function network (RBF) and Multi-layer Percepteron Network (MLP) in the inverse kinematics solution of the 6-DOF robot arm in 2010 [31]. In the study was performed by Köker, a neural network architecture combined with evolutionary techniques was used to solve the inverse kinematics problem of the 6-DOF Stanford robot manipulator [32]. Planar manipulators have been discussed in most of the studies in the literature [33]. In addition, Csiszar et al studied the inverse kinematics problem of spatial 3-DOF robot structure [34]. Toshani and Farrokhi presented an adaptive approach based on the online working Lyapunov Stability Theorem for the inverse kinematics problem of multi-joint manipulators as an alternative to offline machine learning techniques [35]. Ren and Ben-Tzv, unlike previous studies, developed a neural network to solve the inverse kinematics and dynamics of robots using real-world experimental data. The proposed neural network technique was applied to two different robot arms, 4 DOF and 8 DOF. As a result, it has been seen that the proposed method can solve inverse kinematics and dynamics problems with the desired accuracy [36].

In recent years, metaheuristic optimization algorithms, as well as machine learning algorithms, have been considered as an alternative for solving the inverse kinematics problem. Lopez and Franco in 2018 compared to solve the inverse kinematics problem of different robotic manipulators some metaheuristic algorithms such as Differential Evolution (DE), Artificial Bee Colony (ABC), Bat Algorithm (BA), Genetic Algorithm (GA) Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Cuckoo Search (CS), Differential Search (DS), and Particle Swarm Optimization (PSO) [7,37-44]. Shi et al proposed the Hybrid Mutation Fruit Fly Optimization Algorithm (HMFOA) to solve the inverse kinematics of a multi-degrees-of-freedom robot manipulator. An odour search based on multiple mutation strategies and a visual search based on dynamic real-time updates is adopted in HMFOA. The inverse kinematics problem of a 7-DOF manipulator with HMFOA has been solved and compared with various metaheuristic algorithms. The results revealed that HMFOA can be used to effectively solve the inverse kinematics problem [45]. Dereli and Köker were used a Quantum-based PSO for the inverse kinematics solution of a 7 DOF serial manipulator and the results were compared with other swarm techniques such as Firefly Algorithm (FA), PSO, and ABC. Firstly, the DH parameters of the robot manipulator were created and the transformation matrices were revealed. According to the results obtained; It has been shown that more efficient results are provided with quantum-based PSO than standard PSO, ABC, and FA [46].

Zhao et al 2020 presented a new approach to teaching robot kinematics to engineering students. The proposed teaching approach consists of the creation of the inverse kinematics algorithm, the implementation of the algorithm in the virtual experimental environment, and the implementation of the algorithm on the microcontroller, respectively. The sample application of the proposed approach has shown that the new teaching method has a significant effect on improving the robotic learning effect of students [47]. In their study, Al-Tahtawi et al performed the inverse kinematic analysis of a small-scale robotic arm designed as 3-DOF. Inverse kinematics calculations were performed using the geometric approach. Using the analysis results, it was observed that the real-time control of the robot arm was performed via Arduino and the error rate was obtained 3% [48]. When the literature studies are examined, it is seen that the kinematic analysis of the robot arm with many different degrees of freedom has been emphasized with various approaches and the studies in this field are currently continuing. However, it is known that the methods used in the majority of the studies are not explained in detail. Although this situation does not affect the academic value of the articles, there is an important deficiency in terms of education. Therefore, the main purpose of this study is; to compare the inverse kinematics method in the kinematic analysis of robot arms and MLP, which is a common machine learning method, in a simulation environment in detail. Thus, it is aimed to bring a study that engineering students can access on an important subject such as inverse kinematic analysis to the literature. Accordingly, an inverse kinematic analysis of a robot arm with three degrees of freedom RRR design has been made. As an alternative to inverse kinematics, the kinematic analysis of the robot arm was performed using the MLP machine learning algorithm. Coordinate information obtained from inverse kinematics and machine learning methods were compared in three dimensions in the Matlab environment. As a result, the correlation coefficient (R2) was obtained as 1 when training was done with the MLP

machine learning method. In addition, as a result of the test studies performed with MLP, the Mean Relative Error (MRE) value was 1.7 ‰ and the Mean Squared Error (MSE) was 4.6 ‰. This confirmed that the proposed machine learning model works with the desired stability.

2. METHODOLOGY

The solid model of the robot arm, designed in RRR type, was drawn in a CAD program. Kinematic calculations were made for the designed robot arm. Inverse kinematic analysis and MLP were used as two different approaches for the coordinate-joint calculations of the robot arm. The obtained results were compared with the three-dimensional simulation performed in Matlab environment. The general flow diagram of this study is given in figure 1. As seen from figure 1, firstly, forward kinematic equations are calculated and position coordinates are obtained as a function of joint angles. The position data were calculated in response to the angle data obtained by increasing the joint angles at 10-degree intervals. These data were normalized and divided into 80% training, 15% validation and 5% test data. The position and angle data of the data allocated to the training was used as the input and output, respectively. The artificial neural network was trained by this data. With the help of the obtained trained model, the joint angles were obtained for 3 different shapes. The same process is applied with mathematically calculated inverse kinematic equations. These two data sets were simulated by applying them to the robot arm and the results were compared.

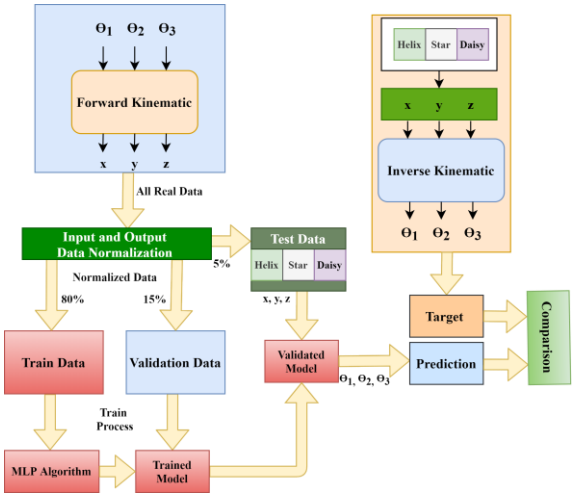

Figure 1. Flow Chart of the study

2.1 Kinematic Analysis

Robot kinematics are explained under two main headings as forward and backward kinematics. These topics are forward and inverse kinematic analysis. The forward kinematic equations are created easily with homogeneous transformation matrices. Several different approaches are used to solve the inverse kinematics problem. These are geometric, analytical, numerical and soft-computing approaches [49,50].

2.1.1 Forward Kinematic Analysis

Forward kinematic equations describe the position relationship between the base of the manipulator and the joints. The variables in these equations vary according to the joint state. If the joint used is radial, the equation variable is considered as angle, and if it is prismatic, the equation variable is considered as length. For the equations used in the transformations of the positions between the joints, D-H proposed a method that can be solved only with the x and z axes. Today, this method is known as the D-H method [51]. In this method, operations are performed with four parameters. These are the z-axis rotation angle (θ), the x-axis rotation angle (α), the z-axis translation (d), and the x-axis translation (a) parameters. These parameters are shown schematically in figure 2.

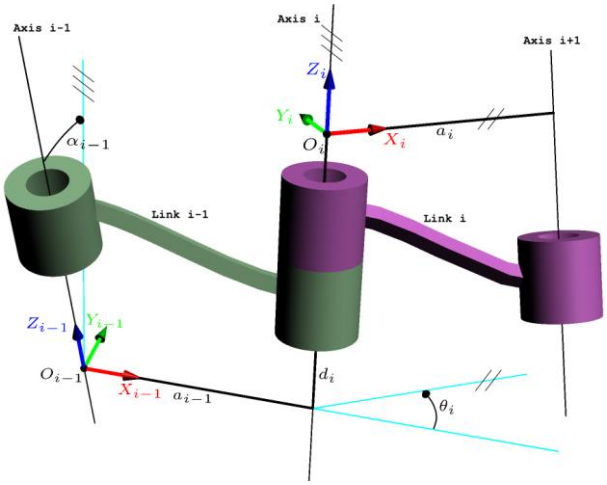

Figure 2. D-H parameters

The matrices obtained from the four parameters given above represent the position change between two neighbor joints. Eq. 1 provides the conversion between (n-1) and n axes. In this equation, "Rotational" and "Translational" expressions are showed as rot and trans, respectively.

$$ {}^{n-1}_{n}T = Rot_{x_{n-1}}(\alpha_{n-1}).Trans_{x_{n-1}}(a_{n-1}).Rot_{z_n}(\theta_n).Trans_{z_n}(d_n) \quad (1) $$

By analyzing Eq. 1, the matrix that defined the relationship between two joints and called the homogeneous transformation matrix, is calculated. The homogeneous transformation matrix is given in Eq. 2. The designed RRR type robot manipulator has 3 degrees of freedom. The parameters required for the D-H method, such as the main position, axes, and joint lengths of the robot arm, are given in figure 3.

$$ {}^{n-1}_{n}T = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & a_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & a_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

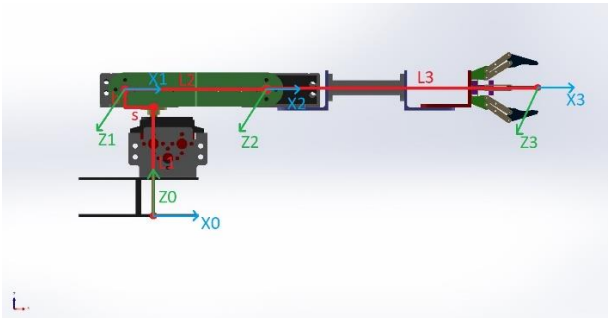

Figure 1. The physical parameters of the robot arm

The four D-H parameters and the angle limits of the joints are given in table 1. Here, s=21.15mm, b=13mm, L1=80.73mm, L2=104.4mm, and L3=201.39mm.

Table 1. D-H parameters

Link	θ_i	a_i	d_i	α_i	Limits (degree)
0-1	θ_1	s	b+L_1	$\pi/2$	
1-2	θ_2	L_2	0	0	
2-3	θ_3	L_3	0	0	

The D-H parameters in Table 1 are used in Eq. 2 to calculate the position correlation between the joints. Thus, the position relations between the neighbor joints were calculated as in Eq. 3.

$$ {}^{0}_{1}A = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & s\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & s\sin\theta_1 \\ 0 & 1 & 0 & b+L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}A = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

$$ {}^{2}_{3}A = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Here, cosine and sine are abbreviated as C and S, respectively. The total transformation between the robot's base axis and the gripper was calculated as in Eq. 4.

$$ {}^{0}_{3}T = {}^{0}_{1}A.{}^{1}_{2}A.{}^{2}_{3}A \quad (4) $$

If the inter-joint position matrices in Eq. 3 are substituted in Eq. 4, the total transformation matrix given in Eq. 5 is obtained.

$$ {}^{0}_{3}T = {}^{0}_{1}A.{}^{1}_{2}A.{}^{2}_{3}A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) $$

Here, the 3x3 size r denotes the rotation of the matrix from the base axis to the gripper. The 3x1 matrix p represents the translation between two axes. In Eq. 6, the values of r and p matrices are given. Thus, the forward kinematic analysis of the robot arm was calculated.

$$ \begin{aligned} r_{11} &= C\theta_1.C\theta_2.C\theta_3 - C\theta_1.S\theta_2.S\theta_3 \\ r_{21} &= C\theta_2.C\theta_3.S\theta_1 - S\theta_1.S\theta_2.S\theta_3 \\ r_{31} &= C\theta_2.S\theta_3 + C\theta_3.S\theta_2 \\ r_{12} &= -C\theta_1.C\theta_2.S\theta_3 - C\theta_1.C\theta_3.S\theta_2 \\ r_{22} &= -C\theta_2.S\theta_1.S\theta_3 - C\theta_3.S\theta_1.S\theta_2 \\ r_{32} &= C\theta_2.C\theta_3 - S\theta_2.S\theta_3 \\ r_{13} &= S\theta_1 \\ r_{23} &= -C\theta_1 \\ r_{33} &= 0 \\ p_x &= s.C\theta_1 + L_2.C\theta_1.C\theta_2 + L_3.C\theta_1.C\theta_2.C\theta_3 \\ &\quad - L_3.C\theta_1.S\theta_2.S\theta_3 \\ p_y &= s.S\theta_1 + L_2.S\theta_1.C\theta_2 + L_3.S\theta_1.C\theta_2.C\theta_3 \\ &\quad - L_3.S\theta_1.S\theta_2.S\theta_3 \\ p_z &= L_1 + b + L_2.S\theta_2 + L_3.C\theta_2.S\theta_3 + L_3.S\theta_2.C\theta_3 \end{aligned} \quad (6) $$

2.1.2 Inverse Kinematic Analysis

The desired result in a robot system is to place the robot gripper at a known point in the robot cartesian space. This result which is called inverse kinematic analysis is a mathematical process to calculate variable joint parameters. Well-known approaches for inverse kinematic analysis are geometric, analytical, and numerical approaches [49]. In this study, geometric and analytical approaches were used together.

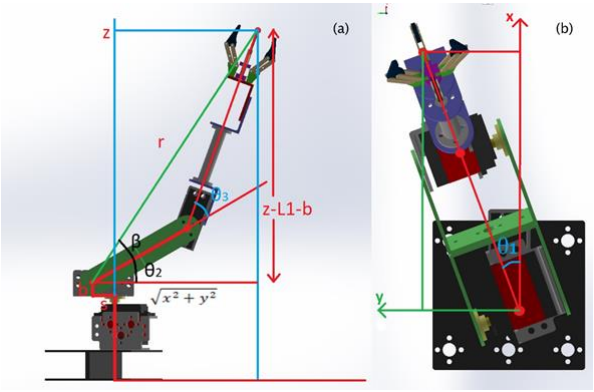

Figure 4. The parameters for the geometric approximation; a: side view, b: top view

While performing inverse kinematic analysis for the RRR type robot arm, the main position of the system was determined as in Figure 3. The final position is formed when the axes move by θ_1, θ_2 and θ_3 angles from the

home position. The side view and the top view of the final position are given in Figures 4-a and 4-b, respectively.

For the inverse kinematics problem, the input is the position of the gripper and the outputs are the angle values of the joints. In figure 4b, a triangle with one angle θ_1 is formed. This triangle has two legs, x and y. The hypotenuse is calculated as $\sqrt{x^2+y^2}$ according to the Pythagorean equation. Thus, the value of θ_1 is obtained by taking into account the following relations in Eq 7

$$\begin{aligned} &x\geq 0 \ \&\ y\geq 0 && tan(\theta_1) = \frac{y}{x} \\ &x<0 \ \&\ y>0 && tan(\pi-\theta_1) = -\frac{y}{x} \\ &x>0 \ \&\ y<0 && tan(2\pi-\theta_1) = -\frac{y}{x} \\ &x<0 \ \&\ y<0 && tan(\theta_1-\pi) = \frac{y}{x} \end{aligned} \quad (7)$$

Triangles related to angles θ_2 and θ_3 are seen, when is examined the side view in Figure 4-a. Equation 8 is obtained when the Pythagorean relation is written for right triangle had angle of $\beta+\theta_2$. Thus, the unknown final length value (r) is calculated. The cosine relation can be written for the angle θ_3 given in Figure 4-a, since the L_2, L_3, and r values are known. This relation is given in Eq. 9. If the cosine relation is rewritten for the β angle in the same triangle, the cos(β) value is obtained in Eq. 10. Thus, the $\beta+\theta_2$ value was calculated in Eq. 11.

$$r = \sqrt{\left(s+\sqrt{x^2+y^2}\right)^2 + (z-L_1-b)^2} \quad (8)$$

$$\cos(\theta_3) = \frac{r^2 - {L_3}^2 - {L_2}^2}{2\cdot L_2 \cdot L_3} \quad (9)$$

$$\cos(\beta) = \frac{r^2 + {L_3}^2 + {L_2}^2}{2\cdot L_2 \cdot r} \quad (10)$$

$$\tan(\theta_2+\beta) = \frac{z-L_1-b}{\sqrt{x^2+y^2}+s} \quad (11)$$

As a result of these processes, the joint angles, which are described as the solution of the inverse kinematics problem, are obtained in terms of the gripper's position.

2.2 Multi-Layer Perceptron (MLP)

In this study, the MATLAB interface was used to create an MLP model for the estimation of the joint angles of the robot arm. MLP is inspired by the biological nervous system in the human brain. These models consist of 3 layers: input, hidden and output layers. Layers contain a processing unit called a neuron. In these neurons, the addition process, and the activation function process are applied to the information from the previous layer. In Figure 5, the neural network model for a single neuron is given schematically. Here, the input values are x_1, x_2, …, x_n. The weight coefficients of model are defined as w_1, w_2, …, w_n. The V_k value is calculated by equation 12. Y_k is the value obtained as a result of subjecting the V_k value to the activation function. When modeling a neural network, this process is performed on all neurons separately [52].

$$V_k = \sum_{i=1}^{n} (w_i x_i) \quad (12)$$

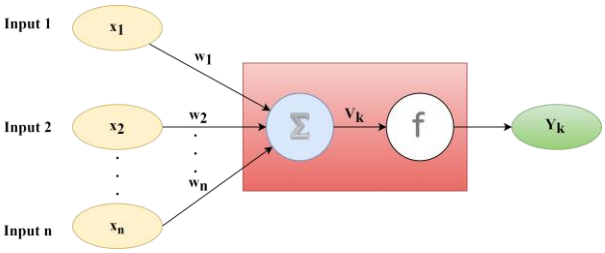

Figure 2. Single Neuron Architecture of MLP

Incoming data from the input layer is transferred to the fast layers to perform certain operations on the neurons. The number of hidden layers varies according to the problem, at least one. The output of each layer is the input of the next layer. Thus, the output layer is reached. An error value will occur if the result in the output layer is different from the target result. With back propagation, this error value is spread over the weights and the forward propagation process is performed again. The continuous f function in Figure 5 represents the differentiable activation function [52- 54]. Any mathematical functions are used as activation function in the model. Among them, the most preferred functions are sigmoid, tang, linear and threshold functions [55-57]. Figure 6 shows the MLP model created for estimating the joint angles of the robot arm.

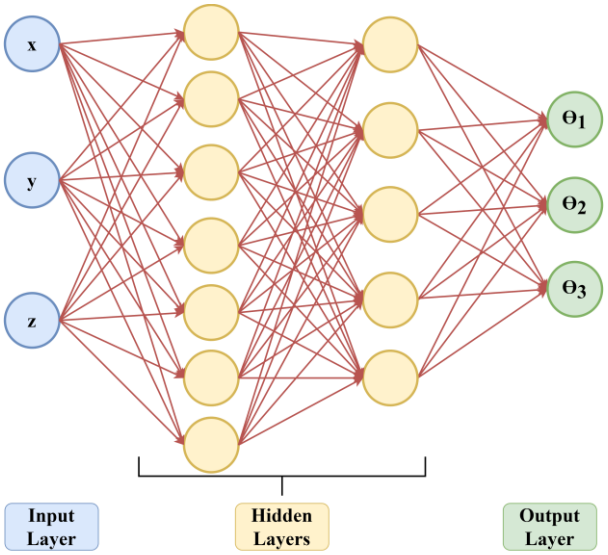

Figure 6. Problem-specific MLP model

In figure 6 above, the MLP model used for the estimation process is given. The weights are updated while the training process is carried out with the MLP model. With the test process, the performance of the MLP model is

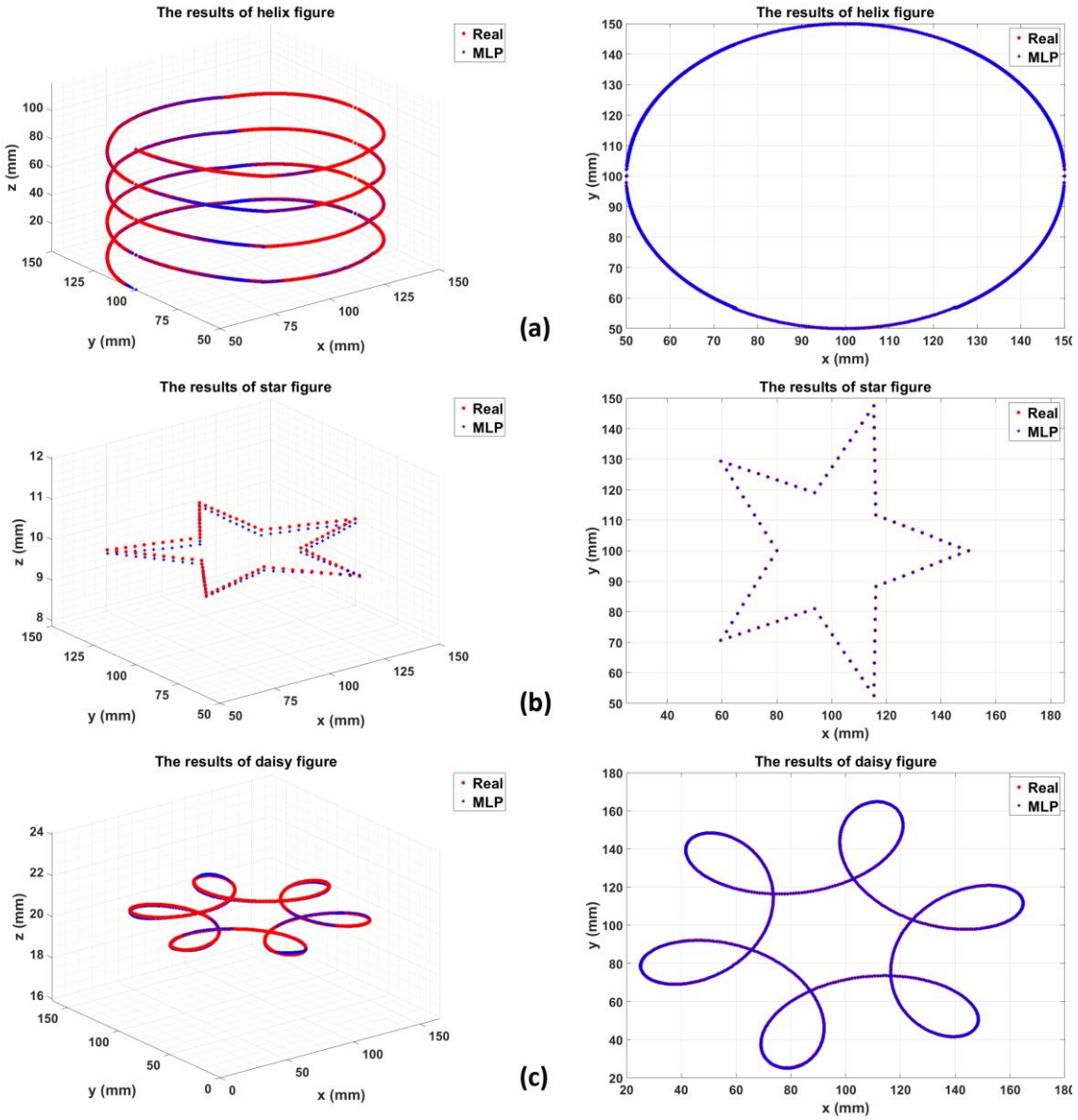

Figure 7. MLP Estimation Results of RRR robot arm's drawings, a:helix, b:star, c:daisy

tested. The training process is repeated by changing the hyperparameters until the appropriate performance result is reached. In the proposed MLP model there is an input layer consisting of 3 different input data (x, y and z positions). In general, the sigmoid function is used as the activation function in numerical estimation studies [58,59]. So, two hidden layers with 7 and 5 neurons using the sigmoid function (Eq.13) as the activation function are used in the model. Thus, the prediction data of joint angles (θ1, θ2 and θ3) are taken from the output layer in the developed model.

$$V(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

3. RESULTS AND DISCUSSIONS

The x, y, and z coordinates are determined for the helix, star and daisy shapes. The joint angles were calculated using both inverse kinematic equations and the MLP algorithm for these coordinates. Obtained results were compared in Matlab environment. The data of helix, star and daisy shapes as a result of this comparison are shown in two and three dimensions in Figures 7a, 7b and 7c, respectively.

It is seen that the estimation results made by MLP in all three shapes completely overlap with the inverse kinematics results in the x-y coordinates when Figure 7 is examined. However, it can be stated that there is a very small error between MLP estimation results and the actual results when the MLP estimation results are evaluated in three dimensions. When Tables 2, 3 and 4 are examined, the actual values of x, y and z coordinates, estimation values made with MLP and relative error values are presented for Helix, star and daisy shapes, respectively.

Table 2. Relative errors of Helix shape according to MLP estimation results

Helix			
Points	**Real x(mm)**	**MLP x(mm)**	**Relative Error**
1	57.3000	57.4473	0.0026
2	57.3500	57.4966	0.0026
3	57.4000	57.5458	0.0025
4	57.4500	57.5951	0.0025
5	57.5000	57.6444	0.0025
…	…	…	…
8821	50.1000	50.0766	0.0005
8822	50.0500	50.0294	0.0004
8823	50.0000	49.9829	0.0003
8824	50.0000	49.9904	0.0002
Points	**Real y(mm)**	**MLP y(mm)**	**Relative Error**
1	126.0137	126.0336	0.0002
2	126.0955	126.1148	0.0002
3	126.1771	126.1956	0.0001
4	126.2583	126.2762	0.0001
5	126.3391	126.3563	0.0001
…	…	…	…
8821	96.1299	96.1664	0.0004
8822	96.8393	96.8787	0.0004
8823	97.7645	97.8072	0.0004
8824	100.0000	100.0477	0.0005
Points	**Real z(mm)**	**MLP z(mm)**	**Relative Error**
1	1.6659	1.5961	0.0419
2	1.6772	1.6067	0.0421
3	1.6886	1.6173	0.0422
4	1.6999	1.6279	0.0424
5	1.7112	1.6385	0.0425
…	…	…	…
8821	99.9547	99.8721	0.0008
8822	99.9660	99.8836	0.0008
8823	99.9773	99.8950	0.0008
8824	99.9887	99.9063	0.0008

The robot arm must pass through 8824 different points in order to form the helix shape in the working space when Table 2 is examined. It is seen from both Table 2 and Figure 7a that the success rate is high when the actual and MLP estimation results of these points are compared. However, when the relative error values were examined, the highest error was obtained from the data on the "z" axis. On the other hand, millimeter (mm) is used as the unit of length in the coordinate system. Therefore, the error occurring at point 5 on the "z" axis, where the highest relative error value is observed, is calculated as 0.0727 mm. From the point of view of the helix shape, this reveals that the MLP prediction model developed for processes with a process precision of 73 micrometer (μm) and more is usable.

The data obtained for the Star shape that the robot arm will draw using the MLP prediction model is presented in Table 3. The robot arm must pass through 110 different points in order to create a star shape in the workspace. It is seen that the proposed machine learning approach has smaller relative error values for the Star shape compared to the Helix shape when Figure 7b and Table 3 are examined together. In addition, when each axis is considered separately, it has been determined that the highest error occurs at the starting points in the "z" axis, similar to the Helix shape. In the scaled examination, the highest error value can be obtained as 0.1269 mm (127 μm) at the first point of the "z" axis. In other words, the proposed machine learning approach for studies with a sensitivity value of less than 127 μm can be an alternative to inverse kinematic analysis. When the coordinate values of Helix and Star shapes are examined, apart from the first points of the "z" axis, it can be stated that higher precision values such as 30-50 μm can be obtained. Therefore, for work that requires higher precision, reducing the distance between the gripper's starting point and the zero position or the speed of the robot arm may increase the sensitivity.

Table 3. Relative errors of star shape according to MLP estimation results

Star			
Points	**Real x(mm)**	**MLP x(mm)**	**Relative Error**
1	116.1803	116.0862	0.0008
2	119.5623	119.4602	0.0009
3	122.9443	122.8354	0.0009
4	126.3262	126.2123	0.0009
5	129.7082	129.5912	0.0009
…	…	…	…
107	108.9615	108.9208	0.0004
108	111.1246	111.0828	0.0004
109	113.2877	113.2476	0.0004
110	115.4508	115.4156	0.0003
Points	**Real y(mm)**	**MLP y(mm)**	**Relative Error**
1	111.7557	111.6478	0.0010
2	110.5801	110.4726	0.0010
3	109.4046	109.2979	0.0010
4	108.2290	108.1234	0.0010
5	107.0534	106.9489	0.0010
…	…	…	…
107	61.00668	61.0071	0.00001
108	58.15351	58.1622	0.0002
109	55.30034	55.3166	0.0003
110	52.44717	52.4698	0.0004
Points	**Real z(mm)**	**MLP z(mm)**	**Relative Error**
1	10.0000	9.8731	0.01269
2	10.0000	9.8798	0.0121
3	10.0000	9.8855	0.0115
4	10.0000	9.8900	0.0110
5	10.0000	9.8931	0.0107
…	…	…	…
107	10.0000	9.9936	0.0006
108	10.0000	10.0058	0.0006
109	10.0000	10.0175	0.0018
110	10.0000	10.0288	0.0029

It is seen that the gripper needs to go to 1000 different coordinate points in order to create the daisy shape using the MLP prediction model of the robot arm when Table 4 is examined. Considering these points, the daisy shape drawn by the robot arm with the machine learning approach was obtained as seen in Figure 7c. Considering

Table 4 and Figure 7c together, it can be stated that the daisy shape has a higher success rate than the other two shapes in terms of relative error. The reason for this is that there are no linear sharp turns between points on the figure. However, when the relative error values are examined, it is seen that the highest error is obtained from the data in the "z" axis and the relative error is lower in the "z" axis compared to the other two figures. Similar to the previous two figures, when the "z" axis values with the highest relative error are considered on a scale, the highest error was calculated as 0.0326 mm (32.6 µm). Unlike the Helix and Star shapes, it is seen that the relative error values are close to each other at all points in the "z" axis, where the highest error is obtained.

Table 4. Relative errors of Daisy shape according to MLP estimation results

Daisy			
Points	**Real x(mm)**	**MLP x(mm)**	**Relative Error**
1	143.0000	142.9283	0.0005
2	143.7537	143.6833	0.0005
3	144.5047	144.4358	0.0005
4	145.2523	145.1848	0.0005
5	145.9958	145.9297	0.0004
…	…	…	…
997	140.7306	140.655	0.0005
998	141.4877	141.4134	0.0005
999	142.2444	142.1714	0.0005
1000	143.0000	142.9283	0.0005
Points	**Real y(mm)**	**MLP y(mm)**	**Relative Error**
1	119.0000	118.9819	0.0002
2	119.2900	119.2725	0.0002
3	119.5563	119.5394	0.0001
4	119.7989	119.7825	0.0001
5	120.0178	120.0018	0.0001
…	…	…	…
997	117.9876	117.9676	0.0002
998	118.3488	118.3294	0.0002
999	118.6862	118.6675	0.0002
1000	119.0000	118.9819	0.0002
Points	**Real z(mm)**	**MLP z(mm)**	**Relative Error**
1	20.0000	19.9731	0.0014
2	20.0000	19.9718	0.0014
3	20.0000	19.9705	0.0015
4	20.0000	19.9690	0.0015
5	20.0000	19.9674	0.0016
…	…	…	…
997	20.0000	19.9761	0.0012
998	20.0000	19.9752	0.0012
999	20.0000	19.9742	0.0013
1000	20.0000	19.9731	0.0014

In this part of the study, the simulation results for different shapes were compared with the error analysis techniques accepted in the literature. The error occurred here was determined by the MRE and MSE performance criteria, which are widely used for statistical analysis methods. Equations of the specified error analysis techniques are given in equations 14 and 15.

$$MRE = \frac{\sum_n \frac{|R_{real} - R_{MLP}|}{R_{real}}}{n} \quad (14)$$

$$MSE = \frac{\sum_n (R_{real} - R_{MLP})^2}{n} \quad (15)$$

MRE results for helix, star and daisy shapes are seen separately in x, y and z coordinates when Table 5 is examined. Accordingly, it was determined that the highest error values occurred in the z-axis in all three ways. MRE values for helix, star and daisy shapes in the working space were obtained as 0.0007, 0.0033 and 0.0011, respectively.

Table 5. MRE Results of MLP

	MRE x	**MRE y**	**MRE z**	**MRE xyz**
Helix	0.0005	0.0005	0.0010	0.0007
Star	0.0006	0.0006	0.0088	0.0033
Daisy	0.0009	0.0006	0.0019	0.0011
Mean	0.0007	0.0006	0.0039	0.0017

MSE values, another important performance criterion, are presented in Table 6. MSE values for helix, star and daisy shapes in the working space were obtained as 0.0034, 0.0065 and 0.0040, respectively. All of these values show that the proposed MLP prediction model provides reliable results.

Table 6. MSE Results of MLP

	MSE x	**MSE y**	**MSE z**	**MSE xyz**
Helix	0.0029	0.0027	0.0047	0.0034
Star	0.0050	0.0052	0.0094	0.0065
Daisy	0.0068	0.0031	0.0022	0.0040
Mean	0.0049	0.0037	0.0054	0.0046

It is seen that the shape in which the derivative value is a piecewise function or has sharp turns is difficult to draw. The function of Helix, daisy and star are given in eq. 16, 17, and 18, respectively. When these equations are examined, it is observed that difficulties of derivatives of equations may sort hard to easy as 18, 17 and 16.

$$\begin{aligned} x &= a * (\sin(4t) + 2) \\ y &= a * (\cos(4t) + 2) \\ z &= \frac{a}{\pi} * (\mathrm{t} - \frac{3\pi}{8}) \end{aligned} \quad (16)$$

$$\begin{aligned} x &= a * (0.5 * \cos(5t) + \sin(t) + 2) \\ y &= a * (0.5 * \sin(5t) + \cos(t) + 2) \\ z &= 20 \end{aligned} \quad (17)$$

$$\begin{aligned} x &= \frac{2}{3} * (y + 20) && 70 < \mathrm{y} < 100 \text{ and } 118 < \mathrm{y} < 148 \\ x &= 116 && 52 < \mathrm{y} < 88 \text{ and } 112 < \mathrm{y} < 148 \\ x &= 155 - \frac{3y}{4} && 52 < \mathrm{y} < 82 \text{ and } 100 < \mathrm{y} < 130 \\ x &= 450 - 3y && 100 < \mathrm{y} < 112 \text{ and } 118 < \mathrm{y} < 130 \\ x &= 3y - 150 && 70 < \mathrm{y} < 82 \text{ and } 88 < \mathrm{y} < 100 \\ & && z = 10 \end{aligned} \quad (18)$$

The results of error analysis also support this approximation. Therefore, it is seen that the shape with the sharpest turns is the star shape when the simulated shapes are examined. For this reason, the error results of the star shape are higher than the others. Considering the

derivative curves of other shapes, the higher the change in unit time, the more difficult it is to draw the shape. That is, the drawing difficulty levels of all simulated shapes are understood as star> daisy> helix.

4. CONCLUSION

In this study, the inverse kinematics method in the kinematic analysis of robot arms and its comparison with machine learning are presented. A three-degree-of-freedom manipulator with RRR design is considered as a robot arm. First of all, the joint angles required for the robot arm gripper tip to move to any desired position in the robot workspace are analytically modelled by inverse kinematic analysis. Inverse kinematics analysis was performed using a hybrid of geometric and algebraic approaches. Thus, the mathematical model of the joint angles (θ1, θ2, and θ3) that enables the robot to reach any position has been determined. Then, the linear kinematics calculation that gives the positions corresponding to the joint angles within the boundaries of the gripper joints in the working space of the robot was made analytically. The data obtained as a result of the forward kinematic analysis are used as input and output in the MLP algorithm, which is one of the widespread machine learning methods. In the trained MLP model, gripper positions (x, y, and z) are considered as input data, and joint angles are considered as output data. Thus, as an alternative to inverse kinematic analysis, a soft-computing estimation approach is proposed for the manipulator's joint angles. Coordinates of the helix, star, and daisy shapes were used as test data in order to determine the success rate of the developed MLP model. In order to compare the MLP model with the real results, the required joint angles for helix, star, and daisy shapes were calculated by using the mathematical models obtained as a result of inverse kinematic analysis. The performance of the machine learning model is revealed by comparing the prediction results obtained from the MLP model with the analytical calculation results. Accordingly, MRE and MSE values were considered as performance criteria. As a result, considering the operation of the robot arm in three-dimensional space, the MRE values for helix, star, and daisy shapes were calculated as 0.0007, 0.0033, and 0.0011, respectively. MSE values were obtained as 0.0034, 0.0065, and 0.0040, respectively.

As a result of this study, it has been revealed that machine learning approaches as a soft computing method can be used as an alternative to inverse kinematic analysis. In further studies, it is aimed to realize the positional control of robot arms with higher degrees of freedom and/or more complex articulated structures in the working space with the MLP method. In addition, it is foreseen that a similar approach will be made in future studies by using different machine learning methods.

ACKNOWLEDGEMENT

This study was supported by Afyon Kocatepe University Scientific Research Projects (AKU-BAPK) Unit with project number 21.KARİYER.03.

DECLARATION OF ETHICAL STANDARDS

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Faruk Emre AYSAL: Performed the MLP prediction, analyzed the results, and wrote the manuscript.

İbrahim ÇELİK: Performed the forward-inverse kinematic, analyzed the results, and wrote the manuscript.

Enes CENGİZ: Performed the MLP prediction, and analyzed the results of predictions.

Yüksel OĞUZ: Analyzed the results of comparisons.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Wu W and Rao SS. "Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis", ***Reliab. Eng Syst Safety***, 92 (1):54–64, (2007).

[2] Hasan AT, Hamouda AMS, Ismail N, and Al-Assadi HMAA. "An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F. serial robot manipulator", ***Sciencedirect Adv Eng Software***, 37(7):432–438, (2006)

[3] Caccavale F, Natale C, Siciliano B, and Villani L. "Integration for the next generation: embedding force control into industrial robots", ***IEEE Robot Autom Mag***;12(3):53–64, (2006).

[4] Gautam R., Gedam A., Zade A., and Mahawadiwar A., "Review on development of industrial robotic arm," ***International Research Journal of Engineering and Technology (IRJET),*** 4(3): 1752-1755 (2017).

[5] Becerra Y., Arbulu M., Soto S., and Martinez F., "A comparison among the Denavit-Hartenberg, the screw theory, and the iterative methods to solve inverse kinematics for assistant robot arm," ***In International Conference on Swarm Intelligence***, Springer, Cham, 447-457, (2019).

[6] Patil A., Kulkarni M., and Aswale A., "Analysis of the inverse kinematics for 5 DOF robot arm using DH parameters," ***In 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)***, 688-693 (2017)

[7] Lopez-Franco C., Hernandez-Barragan J., AlaniA. Y. s, and Arana-Daniel N., "A soft computing approach for inverse kinematics of robot manipulators," ***Engineering Applications of Artificial Intelligence***, 74: 104-120, (2018).

[8] Mahanta G. B., Deepak B. B. V. L., Dileep M., B. Biswal B., and Pattanayak S. K., "Prediction of inverse kinematics for a 6-DOF industrial robot arm using soft computing techniques," ***In Soft computing for problem solving***, Singapore, 519-530, (2019).

[9] Myint K. M., Htun Z. M. M., and Tun H. M., "Position control method for pick and place robot arm for object sorting system," ***International journal of scientific & technology research***, 5(6): 57-61, (2016).

[10] Han S. D., Feng S. W., and Yu J., "Toward Fast and Optimal Robotic Pick-and-Place on a Moving Conveyor," ***IEEE Robotics and Automation Letters***, 5(2): 446-453, (2019).

[11] Nu'man H. S., Sofyan Y., and, Al Tahtawi A. R, "Pengendalian Robot Lengan Pemilah Benda Berdasarkan Bentuk Menggunakan Teknologi Computer Vision", ***Seminar Nasional Teknologi dan Riset Terapan***, 2: 42-48 (2020).

[12] Zulfardi N. F., Saputra D. I., Ahkam A. D. A., "Aplikasi Deteksi Benda Menggunakan Metode Image Substraction Sebagai Masukan Koordinat Pada Robot Lengan 3 DOF," ***Seminar Nasional Teknologi dan Riset Terapan***, 1: 30-37 (2019).

[13] Ishak I., Fisher J., and Larochelle P., "Robot arm platform for rapid prototyping: Concept," ***Conference on Recent Advances in Robotics, Florida***, (2015).

[14] Tolani, D., Goswami, A., and Badler, N. I. "Real-time inverse kinematics techniques for anthropomorphic limbs". ***Graphical Models***, 62(5), 353-388, (2000).

[15] Kim H. S. and Song J. B., "Low-cost robot arm with 3-DOF counterbalance mechanism," ***IEEE International Conference on Robotics and Automation***, 4183-4188, (2013).

[16] Kato G., Onchi D. and Abarca M., "Low-cost flexible robot manipulator for pick and place tasks," ***10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)***, Jeju, 677-680, (2013).

[17] Andrews N., Jacob S., Thomas S. Sukumar M., S. and Cherian R. K., "Low-Cost Robotic Arm for differently abled using Voice Recognition," ***3rd International Conference on Trends in Electronics and Informatics (ICOEI)***, Tirunelveli, India, 735-739, (2019).

[18] Wu Y., Wang M. and Mayer N. M., "A new type of eye-on-hand robotic arm system based on a low-cost recognition system," ***International Conference on Advanced Robotics and Intelligent Systems (ARIS),*** Taipei, 110-114 (2017).

[19] Šuligoj F, Jerbić B., Švaco M., Šekoranja B., Mihalinec D and Vidaković J., "Medical applicability of a low-cost industrial robot arm guided with an optical tracking system," ***IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*** Hamburg, 3785-3790 (2015).

[20] Kroeger O., Wollschläger F. and Krüger J., "Low-Cost Embedded Vision for Industrial Robots: A Modular End-of-Arm Concept," ***25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)***, Vienna, Austria, 1301-1304, (2020)

[21] Luu T. H. and Tran T. H., "3D vision for mobile robot manipulator on detecting and tracking target," ***15th International Conference on Control, Automation and Systems (ICCAS)***, Busan, 1560-1565, (2015).

[22] Sarker P. P., Abedin F., and Shimim F. N., "R3Arm: Gesture controlled robotic arm for remote rescue operation," ***IEEE Region 10 Humanitarian Technology Conference (R10-HTC),*** Dhaka, 428-431, (2017).

[23] Mardiyanto R., Utomo M. F. R., Purwanto D. and Suryoatmojo H., "Development of hand gesture recognition sensor based on accelerometer and gyroscope for controlling arm of underwater remotely operated robot," ***International Seminar on Intelligent Technology and Its Applications (ISITIA)***, Surabaya, 329-333, (2017).

[24] Aggarwal L., Gaur V. and Verma P., "Design and implementation of a wireless gesture controlled robotic arm with vision," ***International Journal of Computer Applications***, 79(13): 39-43, (2013).

[25] Khajone S. A., Mohod S. W. and Harne V. M., "Implementation of a wireless gesture controlled robotic arm," ***International Journal of innovative research in computer and communication engineering***, 3(1), 375-379, (2015).

[26] Kadir W. M. H. W., Samin R. E., and Ibrahim B. S. K., "Internet controlled robotic arm," ***Procedia Engineering***, 41:1065-1071, (2012).

[27] Atmoko R. A., and Yang D., "Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol," ***IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics)***, Bandung, Indonesia, 12-16 (2018).

[28] Siagian P., and Shinoda K., "Web based monitoring and control of robotic arm using Raspberry Pi," ***International Conference on Science in Information Technology (ICSITech),*** Yogyakarta, 192-196 (2015).

[29] Zhang G., He Y., Dai B., Gu F., Han J. and Liu G., "Robust Control of an Aerial Manipulator Based on a Variable Inertia Parameters Model," ***IEEE Transactions on Industrial Electronics***, 67(11): 9515-9525, (2020).

[30] Karlik B., and Aydin S., "An improved approach to the solution of inverse kinematics problems for robot manipulators", ***Eng. Appl. Artif. Intell***. 13 (2) :159–164, (2000).

[31] Chiddarwar S.S., and Ramesh Babu N., "Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach", ***Eng. Appl. Artif. Intell***. 23 (7): 1083–1092, (2010).

[32] Köker R., "A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization", ***Information Sciences***. 222: 528–543, (2013).

[33] Duka A.-V., "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm", ***Procedia Technology***, 12: 20–27, (2014).

[34] Csiszar A., Eilers J., and Verl A., "On solving the inverse kinematics problem using neural networks", 24th ***International Conference on Mechatronics and Machine Vision in Practice, IEEE***, 1–6 (2017).

[35] Toshani H., and Farrokhi M., "Real-time inverse kinematics of redundant manipulators using neural networks and quadratic programming: A Lyapunov-based approach", Robotics and Autonomous Systems. 62 (6): 766–781, (2014).

[36] Ren, H., and Pinhas B-T. "Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks." ***Robotics and Autonomous Systems***, 124: 103386, (2020).

[37] Karaboga, D., and Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. ***Journal Of Global Optimization,*** 39: 459–471, (2007).

[38] González J.R., Pelta D.A., Cruz C., Terrazas G., and Krasnogor N. (Eds.), "Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)", ***Springer Berlin Heidelberg***, 65-74, (2010).

[39] Hansen N., Lozano J.A., Larrañaga P., Inza I., and Bengoetxea E. (Eds.), "Towards a new evolutionary computation: advances in the estimation of distribution algorithms", ***Springer Berlin Heidelberg***, 75-102, (2006).

[40] Yang X.S., and Deb S. "Nature Biologically Inspired Computing", ***World Congress on Nature & Biologically Inspired Computing IEEE***, 210-214, (2009).

[41] Storn, R., and Price, K. "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", ***Journal of Global Optimization,*** 11: 341–359 (1997).

[42] Civicioglu, P., "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm." ***Computers & Geosciences***, 46:229-247, (2012).

[43] Man, K. F., Tang, K. S., and Kwong, S., "Genetic algorithms: concepts and applications [in engineering design]". IEEE Transactions on Industrial Electronics, 43(5): 519-534 (1996).

[44] Kennedy J. and Eberhart R. "Particle Swarm Optimization", ***IEEE International Conference on Neural Networks***, Washington, DC, USA, 1942-1948, (1995).

[45] Shi, J., Mao, Y., Li, P., Liu, G., Liu, P., Yang, X., and Wang, D. (2020). "Hybrid mutation fruit fly optimization algorithm for solving the inverse kinematics of a redundant robot manipulator". ***Mathematical Problems in Engineering***, 2020.

[46] Dereli, S, and Köker R. "A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm." ***Artificial Intelligence Review*** 53(2): 949-964, (2020).

[47] Zhou, D., Xie, M., Xuan, P., and Jia, R. "A teaching method for the theory and application of robot kinematics based on MATLAB and V-REP". ***Computer Applications in Engineering Education***, 28(2): 239-253 (2020).

[48] Al Tahtawi, A. R., Agni, M., and Hendrawati, T. D. "Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics." ***Journal of Robotics and Control (JRC)*** 2(6): 469-475, (2021).

[49] Vasilyev, I. A., and A. M. Lyashin. "Analytical solution to inverse kinematic problem for 6-DOF robot-manipulator." ***Automation and Remote Control*** 71(10): 2195-2199, (2010).

[50] Kucuk S., and Bingul Z., "The Inverse kinematics solutions of industrial robot manipulators", ***IEEE Conferance on Mechatronics***, 274-279, (2004).

[51] Denavit J., and Hartenberg S., "A kinematic notation for lower-pair mechanisms based on matrices", ***Journal of Applied Mechanics*** 1: 215-221, (1955).

[52] Cengiz, E., Yılmaz, C., and Kahraman, H.T. "Classification of Human and Vehicles with The Deep Learning Based on Transfer Learning Method", ***Düzce Üniversitesi Bilim ve Teknoloji Dergisi***, 9(3): 215-225 (2021).

[53] Cengiz, E., Babagiray, M., Aysal, F. E., and Aksoy, F. "Kinematic viscosity estimation of fuel oil with comparison of machine learning methods" ***Fuel***, 316: 123422, (2022).

[54] Kelek, M. M., Cengiz, E., Oguz Y., and Yönetken, A. "RLBP Metodu ile Mamografi Görüntülerinin İncelenmesi ve Sınıflandırılması", ***Afyon Kocatepe Üniversitesi Uluslararası Mühendislik Teknolojileri ve Uygulamalı Bilimler Dergisi***, 4(2): 59-64, (2021).

[55] Güvenç, U., Dursun, M., and Çimen, H., "Artificial Neural Network Based Modeling Of Cutting Time In The Marble Cutting Process", ***International Journal of Technological Sciences***, 3(2): 9-16 (2011).

[56] Karadağ B., Arı A., and Karadağ M., "Derin öğrenme modellerinin sinirsel stil aktarımı performanslarının karşılaştırılması", ***Politeknik Dergisi***, 24(4): 1611-1622, (2021).

[57] Yumurtaci M, and Yabanova İ. "Yapay Sinir Ağları ile Dinamik Ağırlık Tahmin Uygulaması", ***Politeknik Dergisi***, 20(1): 37-41, (2017)

[58] Ngah, S., Bakar, R. A., Embong, A., and Razali, S., "Two-steps implementation of sigmoid function for artificial neural network in field programmable gate array", ***ARPN journal of engineering and applied sciences***, 11(7): 4882-4888 (2016).

[59] Wanto, A., Windarto, A. P., Hartama, D., and Parlina, I., "Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density", ***International Journal of Information System & Technology***, 1(1): 43-54, (2017)