v. 4, n. 19, 2024

Journal of
Engineering
Research

REVOLUTIONIZING 3D ROBOTICS SMART CONTROL WITH NEURAL NETWORKS

Márcio Mendonça
Universidade Tecnológica Federal do Paraná (UTFPR-CP)
Cornélio Procópio – Paraná
http://lattes.cnpq.br/5415046018018708

Emanuel Ignacio Garcia
Universidade Tecnológica Federal do Paraná (UTFPR-CP)
Cornélio Procópio – Paraná
http://lattes.cnpq.br/8501809850590859

Fabio Rodrigo Milanez
UniSENAI PR Campus Londrina
Londrina-PR
http://lattes.cnpq.br/3808981195212391

Francisco de Assis Scannavino Junior
Universidade Tecnológica Federal do Paraná - Academic Department of Electrical Engineering (DAELE)
Cornélio Procópio – Pr
http://lattes.cnpq.br/4513330681918118

Iago Maran Machado
Universidade Tecnológica Federal do Paraná
Master student - PPGEM-CP – Post-graduation Program in Mechanical Engineering - PP/CP
Cornélio Procópio - PR
http://lattes.cnpq.br/4733940365047328

Marcos Antônio de Matos Laia
Universidade Federal de São Joao Del Rei - Department of Computer Science – UFSJ
http://lattes.cnpq.br/7114274011978868

Vicente de Lima Gongora
Faculdade da Industria Senai
Londrina - Pr
http://lattes.cnpq.br/6784595388183195

Ricardo Breganon
Instituto Federal do Paraná, Campus Jacarezinho
Jacarezinho – Pr
http://lattes.cnpq.br/2441043775335349

Wagner Fontes Godoy
Universidade Tecnológica Federal do Paraná
Department of Electrical Engineering – Daele
Cornélio Procópio - PR
http://lattes.cnpq.br/7337482631688459

Gustavo Henrique Bazan
Instituto Federal do Paraná, Campus Jacarezinho
Jacarezinho - PR
http://lattes.cnpq.br/7076940949764767

Andre Luis Shiguemoto
Universidade Tecnológica Federal do Paraná
DAELE-CP – Academic Department of Electrical Engineering
Cornélio Procópio - PR
http://lattes.cnpq.br/9243656534211182

Andressa Haiduk
Dimension Engenharia
Ponta Grossa - PR
http://lattes.cnpq.br/2786786167224165

Abstract: Given these difficulties, the objective of this work is to present a possible solution to this problem through the application of an artificial neural network (ANN) in a robotic arm with 4 degrees of freedom in three dimensions. The results found demonstrate the feasibility of the proposed solution in comparison with traditional geometric techniques. The use of ANNs offers a robust method for solving inverse kinematics, providing a viable alternative to conventional approaches. This technique simplifies the complexity associated with high degrees of freedom and efficiently manages the multiple solution problem. The findings indicate that ANNs can effectively deal with the complexities of inverse kinematics, paving the way for future advances in robotic control systems. By taking advantage of the learning capabilities of ANNs, this approach shows promise for improving the control accuracy and efficiency of robotic manipulators, contributing to more sophisticated and adaptable robotic applications. And finally, the work ends with a conclusion and suggestion for future work.
Keywords: Robotic arm; Inverse Kinematics; Artificial neural networks.

INTRODUCTION

Technology has experienced exponential growth in recent years in areas such as Industry 4.0 and machine learning, specifically through deep learning, one of the branches of intelligent computing systems (AGGARWAL, 2018). In this context of accelerated technological advancement, robotics has significantly expanded its field of application, including space exploration, robotic surgeries, industrial applications, among others (SICILIANO; KHATIB, 2016).

Specifically, technologies such as IoT (Internet of Things), which are integrated with robotics, represent one of the pillars of Industry 4.0. A state-of-the-art example is the work of Zhu and Wu (2020), which proposes modern vocational education in China using the techniques discussed in this article.

Robotics has developed rapidly, despite facing challenges inherent to its progress. Classic problems include camera calibration and kinematic and dynamic modeling. In direct kinematics, using the Denavit and Hartenberg (DH) model (DENAVIT; HARTENBERG, 1955), it is possible to determine the final position of a terminal organ, such as a clamp of a robotic manipulator arm, knowing the angles of the joints and the dimensions of the arms.

MathWorks (2020) exemplifies a case of direct kinematics for a two-dimensional robotic arm with 2 degrees of freedom. Inverse kinematics, on the other hand, is not a trivial analytical task (NIKU, 2011). Even in simplified two-dimensional cases, at least two solutions are possible, which justifies the need for robust solutions to this problem. In short, inverse kinematics seeks to find the joint angles that position the arm in the desired final position.

Robotic manipulator arms find applications in several areas, such as agriculture, remote surgery, volcano inspection and the military. These examples demonstrate the versatility and transformative potential of robotics across multiple sectors.

Among other areas of knowledge. Agriculture has been modernizing in recent years towards robotics due to the increasing difficulty in finding skilled labor capable of carrying out tedious activities as in traditional agriculture.(ZHAO et al., 2016). In addition to their versatility in tasks such as harvesting, planting, cultivation and application of inputs, robots can reduce the risk of contamination and production costs without degrading the environment. Among the application examples found are: the extraction of wood from a forest in rough terrain, automated milk production and the automated harvesting of a plantation by a tractor combining vision sensors and GPS (SICILIANO; KHATIB, 2016).

Another important related topic is in the continuous development of society and science and technology, scientific management has received more and more attention. As institutions of higher education, universities need to accurately assess the workload, scientific achievements and academic level of each researcher. At the same time, universities are the main centers of scientific research and technological development. With the rapid advancement of computer science and networks, the management of scientific research in universities has become computerized and advanced towards network management.

Currently, most universities have established their own research management information systems. Although each has its characteristics, there are also some problems. Experiments have proven that the proposed algorithm has a good correspondence between the estimated value and the actual value of the evaluation object of teaching management in universities, and has a strong promotional value, thus promoting the better and faster development of teaching information management scientific research in universities (GUO; QU, 2022).

The inspirations for this research were projects such as the one developed by researchers at the University of Catania, Italy, who developed an automatic harvester capable of harvesting fruits based on their ripening color. Equipped with a mechanical arm and a caterpillar, the robot stipulated the distance using a dynamic incremental motion controller and a Kalman filter to determine the approximate diameter of the fruit to harvest. Its efficiency was approximately 5.85 seconds per fruit(MUSCATO et al., 2005).

This work seeks to solve inverse kinematics using an Artificial Neural Network (ANN) and a three-dimensional arm with 3 degrees of freedom(MATHWORKS, 2020). This problem is considered canonical for robotics, presenting its solution through the inverse Jacobian matrix. In addition to the difficulty of modeling, there is the classic problem of multiple solutions and the need to know the relationship between the arm and its joints, which form the Jacobian Matrix.(MATARIC, 2007).

The use of ANN among other intelligent computing systems is already known in the literature to solve classic problems, such as camera calibration using ANN(AHMED; HEMAYED; FARAG, 1999) and more recent works such as that of (DASH et al., 2011), (PEDRA et al., 2013), (MENDONÇA et al, 2022).

To solve the problem of inverse kinematics, we can cite as an example the work of (WELSFORD; PRETORIUS; DU PLESSIS, 2018).

This article is organized as follows: chapter 2 presents a brief background and contextualizes the problem of inverse kinematics using artificial neural networks. Section 3 presents a formalism of the DH model. Section 4 discusses the application of ANN in the inverse kinematics solution. Section 5 discusses results obtained. And finally, section 6 concludes, addresses future work and ends.

ARTIFICIAL NEURAL NETWORKS APPLIED IN THE INVERT KINEMATICS SOLUTION

The use of ANNs in solving inverse kinematics can be found in a considerable number of articles. As for its content, the present work is based on the most relevant articles on the application of ANN in tune with inverse kinematics for robotic actuators.

The work of Oyama et al. (2005) uses a Modular ANN with 4 layers in a robotic arm with 5 degrees of freedom. The training method used was back-propagation to select the best arrangement of angles in order to minimize position/orientation errors of the tip of the arm. To this end, the Gauss-Newton algorithm was modified for the least square's method.

In Kumar and Chand (2015) Two methods were used for the position of the joints in the kinematic analysis. The geometric model and the DH model served as training data for a feed-forward Neural Network (MLFF). The result was tested on hardware (SCORBOT-ER 4u).

In the article by Puheim and Madarász (2014), the authors seek to solve the problem of inverse kinematics in a 3-degree-of-freedom arm of a humanoid robot. The controller used was a feed-forward ANN. Its training is based on the normalization of inputs and outputs, thus increasing accuracy.

In current scientific circles, many articles address the use of ANNs together with various techniques for better results. From training to control, methods such as feed-forward, back-propagation and fuzzy logic can be employed. The goal is better training and better accuracy in results

In Morris and Mansor (1997) three-layer ANNs with training via backpropagation or backpropagation were used to analyze the inverse kinematics of manipulators with two and three degrees of freedom.

KINEMATIC MODELING OF THE ROBOTIC MANIPULATOR

Kinematics is a branch of mechanics that focuses on the motion of objects without considering the forces that cause this motion. In robotics, it involves the position and orientation of a robotic arm's end-effector (tool) given the angles of each joint in a system of equations. This forward kinematics process is crucial for controlling robotic manipulators. Conversely, the inverse of this process, known as inverse kinematics, involves determining the joint angles required to achieve a desired position and orientation for the end-effector. This process is significantly more complex due to the potential for multiple solutions or even no solution in certain configurations (Whitty, 2012).

For a simple case, such as a robotic arm with two degrees of freedom (2-DOF), the solution to the kinematic equations in two dimensions involves determining the angle between the first and second segments of the arm and the angle between the first segment and the ground. This can be visualized in a planar setup where the arm segments move within a two-dimensional plane.

However, as the number of degrees of freedom increases, the complexity of the inverse kinematics problem grows exponentially. For a robotic arm with 'n' angular joints representing 'n' degrees of freedom (n-DOF), the Denavit-Hartenberg (DH) parameters become essential. The DH parameters (αi, ai, di, and θi) provide a systematic way to express the kinematics of the manipulator for each joint, simplifying the complex relationships between the joints and the end-effector.

In essence, the DH parameters are used to create a transformation matrix that describes the position and orientation of each link relative to the previous one. This method transforms the complex spatial relationships into a series of simple transformations, which can be easily managed and computed. The four DH parameters are:

- () : The twist angle between the z-axes of consecutive frames a_i.
- () : The link length, which is the distance between the z-axes of consecutive frames a_i.
- () : The offset along the previous z-axis to the common normal d_i.
- () : The angle between the x-axes of consecutive frames θ_i.

For a 4-DOF robotic arm, the position of each joint and the end-effector can be calculated by sequentially applying the transformation matrices derived from the DH parameters. This systematic approach simplifies the process of determining the final position and orientation of the end-effector, even for complex multi-joint systems (Niku, 2011).

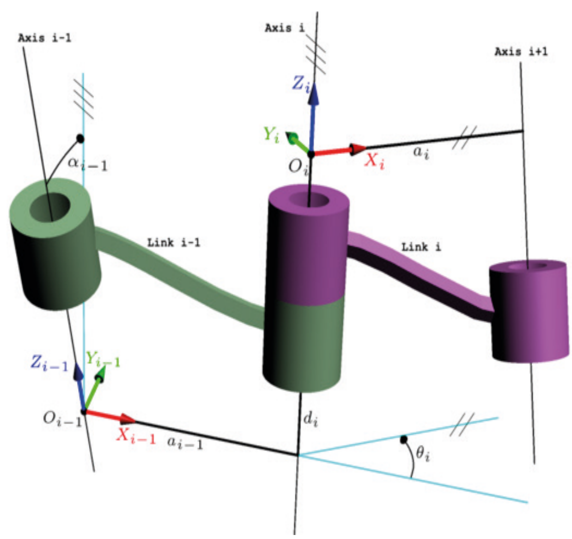

Figure 1 - Representation of DH parameters.

Source: Adapted from Corke (2011).

The DH parameters are described as follows:

- a_i is the distance from the measure about; $\hat{Z}_i \hat{Z}_{i+1} \hat{X}_i$

- a_i is the angle between and measure about; $\hat{Z}_i \hat{Z}_{i+1} \hat{X}_i$
- d_i is the distance from the measure about; $\hat{X}_i \hat{X}_{i+1} \hat{Z}_i$
- θ_i is the angle between the measure about. $\hat{X}_i \hat{X}_{i+1} \hat{Z}_i$

Originating equation 1, denoting the system of transformations (T) that the robotic arm must perform to reach a given position. For a robotic arm with 4 degrees of freedom, we then have equation (2).

$$T_i^{i-1}(\theta_i, d_i, a_i, \alpha_i) = T_z(\theta_i) T_z(d_i) T_x(a_i) T_x(\alpha_i) \quad (1)$$

$$T_4^0 = T_1^0 . T_2^1 . T_3^2 . T_4^3 \quad (2)$$

In matrix form we have equation 3:

$$T_4^0 = \begin{bmatrix} C\theta_1\, C(\gamma) & -C\theta_1\, S(\gamma) & S\theta_1 & g_1 C\theta_1 \\ S\theta_1\, C(\gamma) & -S\theta_1\, S(\gamma) & -C\theta_1 & g_1 S\theta_1 \\ S(\gamma) & C(\gamma) & 0 & g_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Considering the set of equations (4) below:

$$\begin{gathered} k_1 = (a_4\, C\theta_4 + a_3) C\theta_3 - a_4\, C\theta_4\, S\theta_3 + a_2 \\ k_2 = (a_4\, C\theta_4 + a_3) S\theta_3 + a_4\, S\theta_4\, C\theta_3 \\ g_1 = C\theta_1\, k_1 - S\theta_2\, k_2 \\ g_2 = S\theta_2 k_1 + C\theta_2\, k_2 + a_1 \\ \gamma = \theta_2 + \theta_3 + \theta_4 \end{gathered} \quad (4)$$

A specification of the temporal and spatial trajectories of an arm are prerequisites for the successful application of a manipulator for many tasks such as welding, painting, picking parts from conveyors and gluing. (BEN-ARI; MONDADA, 2018). The Jacobian matrix is the partial derivative obtained from the geometric equations that define the final position of a manipulator. For two degrees of freedom in 2D the Jacobian matrix is:

$$\begin{cases} x_f = l_1 . cos\theta_1 + l_2 . \cos(\theta_1 + \theta_2) \\ y_f = l_1 . sen\theta_1 + l_2 . \mathrm{sen}(\theta_1 + \theta_2) \end{cases} \quad (5)$$

Calculating the derivative of equation (5) :

$$\begin{cases} dx_f = -l_1 . d\theta_1 . sen\theta_1 - l_2 . (d\theta_1 + d\theta_2) . \mathrm{sen}(\theta_1 + \theta_2) \\ dy_f = \ \ l_1 . d\theta_1 . cos\theta_1 + l_2 . (d\theta_1 + d\theta_2) . \cos(\theta_1 + \theta_2) \end{cases} \quad (6)$$

In matrix form:

$$\begin{bmatrix} dx_f \\ dy_f \end{bmatrix} = \begin{bmatrix} -l_1 . sen\theta_1 - l_2 . \mathrm{sen}(\theta_1 + \theta_2) & -l_2 . \mathrm{sen}(\theta_1 + \theta_2) \\ l_1 . cos\theta_1 + l_2 . \cos(\theta_1 + \theta_2) & l_2 . \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} \quad (7)$$

In which:

- dx_f: the differential movement of the manipulator end;
- $l_1\ l_2$: the size of each arm;
- $\theta_1\ \theta_2$: the angle of each degree of freedom;
- $d\theta_1\ d\theta_2$: the differential movement of joints.

Figure 2 shows an example of generating the DH parameter table, from the Stanford robot (NIKU, 2011).

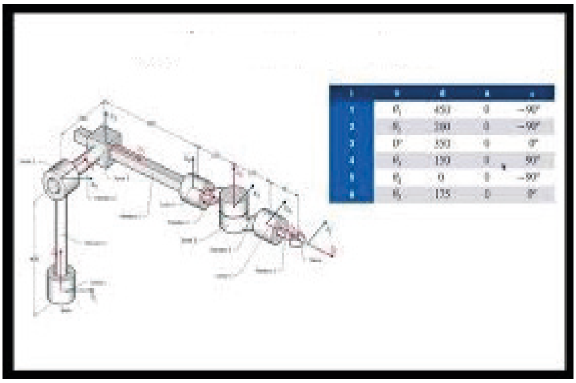

Figure 2.

Source: Adapted from Niku; 2011

Thus, using equation (7) it is possible to trace the trajectory of the robotic manipulator arm from its current point to the desired point(NIKU, 2011).

ARTIFICIAL NEURAL NETWORKS APPLICATION

Human beings excel in their ability to process complex information, whether precise or approximate. When there is a partial or total lack of knowledge about a problem, it becomes necessary to apply an imprecise strategy, which can be expressed linguistically, as seen in Fuzzy logic (ZADEH, 1976).

However, data about how a system is working can also provide valuable information. This type of manipulation can be carried out through identification and modeling techniques (MENDONÇA et al., 2016).

Among these techniques, Artificial Neural Networks (ANNs) stand out for their ability to perform parallel processing of large data sets. ANNs are models inspired by biology that simulate the functioning of the animal brain, using artificial neurons (HAYKIN, 2009). A possible investigation in the proposal of this work is the investigation between the performance of a classical neural network with Deep Learning, such as the work of Rayudu and Roseline, (2023) briefly presented an advantage of the classic version, some of the details of this research.

This study investigated the effectiveness of AI approaches such as Artificial Neural Networks (ANNs) and Deep Neural Networks (DNS), in the rain forecast. The methods were tested and compared in terms of efficiency. Group 1 used a Deep Neural Network (DNN), while Group 2 used an Artificial Neural Network (ANN) with a pretest power of 80% and alpha error rate of 0.05. 20 samples were analyzed, 10 from each group. The results of the MATLAB simulations showed that the accuracy of DNN was 92.59%, while that of ANN was 95.68%. Using the SPSS statistical program, it was verified that the accuracy ratio achieved is 0.034 ($p < 0.05$). The study concludes that the Artificial Neural Network (ANN) algorithm outperforms the Deep Neural Network method (DNN) in rainfall forecasting in an advanced meteorological forecasting context.

According to Coppin (2004), there is a remarkable similarity between biological neurons and artificial neurons in a neural network. The human brain contains billions of neurons interconnected by synapses, through which information is processed and exchanged. Specifically, each neuron receives input through its dendrites, processes the signal in its nucleus (soma), and transmits it through the axon as output.

In robotics, the Denavit-Hartenberg (DH) model is formed based on the structure of the robotic arm. The distance between the first and second joint constitutes the length L2 of the first arm section. The second section (L3) is between the second and third joints, and the third section (L4) is between the third and fourth joints. The length L1 is considered zero, as it represents the origin of the Cartesian plane at the base of the first joint. Each of the joints has degrees of freedom denoted by $\theta 0$, $\theta 1$, $\theta 2$ and $\theta 3$. Figure 2 illustrates a four degrees of freedom (GDL) manipulator in the xyz plane.

This detailed understanding of both neural networks and robotic kinematics is essential for advancement in the field of robotics, where ANNs can be employed to solve complex problems such as inverse kinematics, improving the accuracy and adaptability of robotic systems.

Figure 3: Generic example of a 3GDL handler.

Source: Authors, 2024.

The position data from the robotic arm generates the sample angles of the joints through kinematic, geometric and DH models. Then, the input to the ANN consists of the position values and angles of the joints. Each point corresponds to its respective joint angles. The departure angles must follow the same similarity (x_0; y_0; z_0) θ_0, θ_1, θ_2, θ_3, (KUMAR; CHAND, 2015).

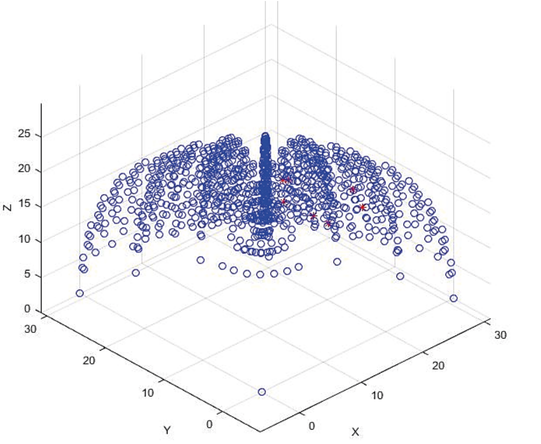

Figure 4: Representation of the action area and test points (front).
Source: Authors, 2024.

As previously mentioned, the actuator's area of action is made up of 4 GDL. Each presents an arc of action limiting the rotation of each joint. The base of the robotic arm rotates through 90º at the first joint, while the other joints are limited to 60º. The ANN uses hundreds of these points from the cloud generated by the DH model for training, as a relatively large mass of data, and after training, some points that were not used in training will be used as tests (in red) so that the ANN must be able to generate the set of angles that correspond to the desired points. Therefore, the final position of the arm must contain the smallest error compared to the position of the test points. To this end, it is necessary to create a set of points using direct kinematics, as shown in Figures 3, 4 and 5, action radius points in blue and test points in red. To verify the training accuracy, some points were selected for testing the network. A priori, such points must not be part of the cloud calculated by direct kinematics and, therefore, present the position of conventional points. As this is a simplified simulation, the radius of action is calculated for just one quadrant.

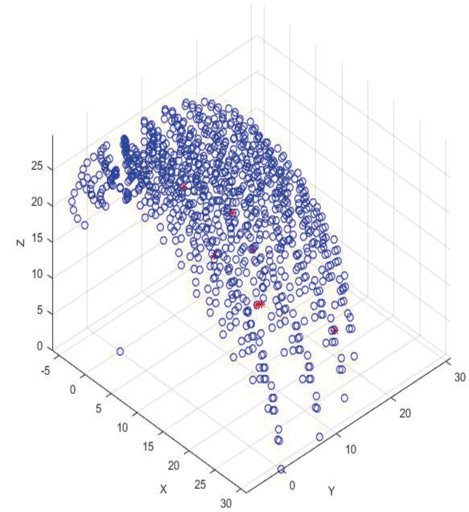

Figure 5: Representation of the action area and test points (side)
Source: Authors, 2024.

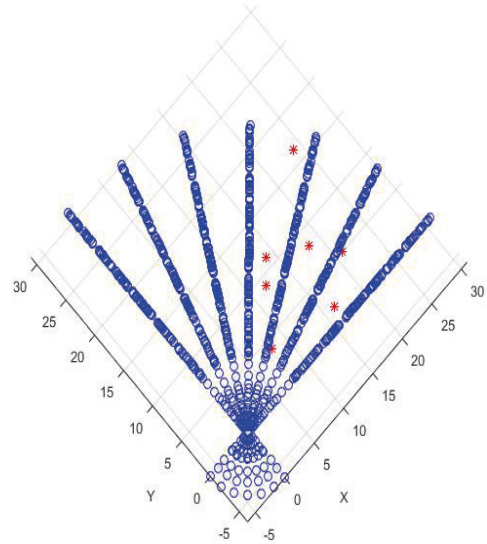

Figure 6: Representation of the action area and test points (top).
Source: Authors, 2024.

SIMULATION AND RESULTS

Limiting the number of training epochs to 3000, Tables 1 and 2 were obtained with the average relative error per coordinate and the total error. At the end of the simulation, the positions of the test points were compared with the points found using the angles obtained from the ANN output.

The configuration of the ANN architecture was with 40 neurons distributed in four layers. The transfer function used was sigmoid. This number of neurons was empirically defined incrementally(DA SILVA et al., 2017), (HAYKIN, 2009) the number of layers was due to the need to find 4 angles, a solution vector with 4 elements, each element has a different domain. The training algorithm was Levenberg Marquadt with tansig activation functions in the intermediate layers and ramp in the output layer. The network training error was on the order of 10-6.

Plausible solution found				
Point	**θ_0**	**θ_1**	**θ_2**	**θ_3**
1	5.847°	22.54°	15.157°	54.94°
2	13.439°	4.959°	13.167°	59.95°
3	23.68°	2.085°	3.724°	82.43°
4	34.82°	0.656°	3.922°	28.86°
5	36.81°	22.69°	15.31°	61.84°
6	26.69°	39.88°	15.32°	57.07°
7	38.24°	6.928°	13.94°	76.61°

Table 1: Angles corresponding to the use of RNA.

The results obtained may vary depending on adjustments in the topology of the neural network, layers, number of neurons, training algorithm, and targeted error. The points reached initially present angles calculated through inverse kinematics when trained by the ANN.

Point	Coordinate [cm]					
	Desired			**Obtained**		
	X	**Y**	**Z**	**X**	**Y**	**Z**
1	16.39	1,722	21.1	16.42	1.68	21.16
2	21.0	5.0	15.0	21.10	5.04	14.96
3	18.54	8.25	12.28	18.57	8.14	12.44
4	24.0	16.5	7.3	23.92	16.64	7.04
5	12.0	9.0	21.0	12.08	9.04	21.08
6	8.0	4.0	25.0	8.05	4.05	24.98
7	14.0	11.0	16.0	13.97	11.00	15.91

Table 2: Comparison between desired and obtained points.

These solutions take into consideration, that training and validation data and tests are not the same and were taken from the previously generated point cloud. This means that the test points were only reached in close proximity as they are within the action area of the manipulator's physical structure. Comparing the coordinates of the desired points with those found, we have the relative error of the points obtained by inverse kinematics at the end of the simulation, shown in Table 3. Analyzing the results, it can be inferred that the approximation obtained values close to those desired. The average distance between the test points and the desired points was 4 mm.

Error [cm]			
Between Desired and Obtained Points	**Coordinates**		
	X	**Y**	**Z**
0.42	-0.23	0.027	0.35

Table 3: Corresponding data on the use of RNA.

Through analysis of the results of the tables, it was suggested that, according to the errors found in the three axes, as mentioned above, the satisfactory use of ANN in finding a feasible solution to the inverse kinematics problem. However, the proposal requires data processing time, which requires pre-processing, which can increase complexity in a dynamic environment, due to the experiments being carried out in a static environment.

Therefore, in this situation, some dynamic adaptation algorithms must be used to obtain faster weights, due to the importance of the response time of a system in robotics (SICILIANO AND KHATIB, 2016).

Thus, when precision is necessary, ANN proves to be a promising tool, but used with caution to obtain the targeted errors. It must be noted that a robot also depends on its mechanical system, and there is no point in obtaining millimeters in the algorithm if the manipulator It is accurate to within a centimeter. Regarding the visual result, Figure 6 shows the position of the test points, represented by circles, and the points found as asterisks.

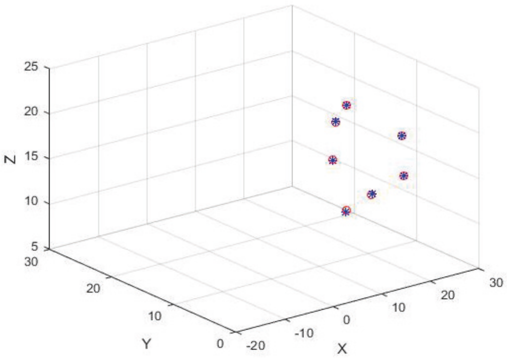

Figure 7: Representation of points in 3D space.

Source: Authors, 2024.

CONCLUSION

The use of ANNs was promising, as it suggested the ability to obtain the desired precision at the end of the simulations, although it did not provide evidence or proof that the solution is optimal, however plausible as mentioned above. Also, as mentioned above, the technique used, ANN, was efficient in solving the inverse kinematics problem for the case of 4 degrees of freedom.

A set of angles was generated within the actuator action limits and according to the training point cloud.

Future work aims to use other intelligent computational techniques, such as genetic algorithms, with which initial results have already been obtained in two dimensions, and finally the comparison with a deep neural network (Deep Learning), due to the massive amount of data.

REFERENCES

AGGARWAL, C. C. **Neural Networks and Deep Learning: A textbook**. 1. ed. Cham, Switzerland: Springer, 2018.

AHMED, M. T.; HEMAYED, E. E.; FARAG, A. A. **Neurocalibration: a neural network that can tell camera calibration parameters**. Proceedings of the Seventh IEEE International Conference on Computer Vision. **Anais**...Kerkyra, Greece: IEEE, 1999. Disponível em: <http://ieeexplore.ieee.org/document/791257/>

BEN-ARI, M.; MONDADA, F. **Elements of Robotics**. 1. ed. Cham, Switzerland: Springer International Publishing, 2018.

COPPIN, B. **Artificial Intelligence Illuminated**. 2. ed. Burlington, MA, USA: Jones & Bartlett Learning, 2004.

DA SILVA, I. N. et al. **Artificial neural networks: a practical curse**. 1. ed. Cham, Switzerland: Springer International Publishing, 2017.

DASH, K. K. et al. **A neural network based inverse kinematic problem**. 2011 IEEE Recent Advances in Intelligent Computational Systems. **Anais**...Trivandrum, Kerala, India: IEEE, 2011

DENAVIT, J.; HARTENBERG, R. S. **A kinematic notation for lower-pair mechanisms based on matrices**. ASME Journal of Applied Mechanics, v. 23, p. 215–221, 1955.

GUO, J.; QU, M. Research on optimization of scientific research management information system in colleges and universities based on artificial neural network. In: EUROPEAN CONFERENCE ON NATURAL LANGUAGE PROCESSING AND INFORMATION RETRIEVAL (ECNLPIR), 2022, Hangzhou, China. Anais [...]. Hangzhou: IEEE, 2022. p. 23-27. DOI: 10.1109/ECNLPIR57021.2022.00017

HAYKIN, S. S. Neural networks and learning machines. 3. ed. Upper Saddle River: Pearson Prentice Hall, 2009.

KUMAR, R. R.; CHAND, P. **Inverse kinematics solution for trajectory tracking using artificial neural networks for SCORBOT ER-4u**. 2015 6th International Conference on Automation, Robotics and Applications (ICARA). **Anais**...Queenstown, New Zealand: IEEE, 2015.

MATARIC, M. J. **The Robotics Primer**. [s.l.] The MIT press, 2007.

MATHWORKS. **Modeling Inverse Kinematics in a Robotic ArmHelp Center**, 2020. Disponível em: <https://www.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html>. Acesso em: 6 abr. 2020.

MENDONÇA, M. et al. **Hybrid Dynamic Fuzzy Cognitive Maps and Hierarchical Fuzzy Logic controllers for Autonomous Mobile Navigation**. 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016. **Anais**...Vancouver, BC, Canada: IEEE, 2016.

MENDONÇA, M.; PALÁCIOS, R. H. C.; BREGANON, R.; BOTONI DE SOUZA, L.; RODRIGUES CINTRA MOURA, L. **Analysis of the Inverse Kinematics and Trajectory Planning Applied in a Classic Collaborative Industrial Robotic Manipulator**. IEEE Latin America Transactions, v. 20, n. 3, p. 363-371, março 2022. doi: 10.1109/TLA.2022.9667133.

MORRIS, A.; MANSOR, A. **Finding the inverse kinematics of manipulator arm using artificial neural network with lookup table**. Robotica, v. 15, n. 6, p. 617–625, 1997.

MUSCATO, G. et al. **A prototype of an orange picking robot: Past history, the new robot and experimental results**. Industrial Robot, v. 32, n. 2, p. 128–138, 2005.

NIKU, S. B. **Introduction to Robotics: Analysis, Control, Applications**. 2. ed. Hoboken, New Jersey, USA: John Wiley & Sons, 2011. v. 53.

OYAMA, E. et al. **Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems**. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. **Anais**...Edmonton, AB, Canada: IEEE, 2005.

PEDRA, A. V. B. M. et al. **Camera Calibration Using Detection and Neural Networks**. IFAC Proceedings Volumes, v. 46, n. 7, p. 245–250, maio 2013.

PUHEIM, M.; MADARÁSZ, L. **Normalization of inputs and outputs of neural network based robotic arm controller in role of inverse kinematic model**. 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI). **Anais**...Herl'any, Slovakia: IEEE, 2014.

RAYUDU, D. V.; ROSELINE, J. F. **Accurate Weather Forecasting for Rainfall Prediction Using Artificial Neural Network Compared with Deep Learning Neural Network**. In: International Conference On Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), 2023, Chennai, India. Piscataway: IEEE, 2023. p. 1-6. DOI: 10.1109/ICECONF57129.2023.10084252.

SICILIANO, B.; KHATIB, O. (EDS.). **Springer Handbook of Robotics**. 2. ed. Heidelberg: Springer-Verlag Berlin Heidelberg, 2016.

WELSFORD, D. L.; PRETORIUS, C. J.; DU PLESSIS, M. C. **Neural Networks for Mobile Robot Inverse Kinematics**. 2018 Eighth International Conference on Information Science and Technology (ICIST). **Anais**...2018.

WHITTY, M. Robotics, **Vision and Control. Fundamental Algorithms in MATLAB**. Industrial Robot: An International Journal, v. 39, n. 6, p. ir.2012.04939faa.005, 12 out. 2012.

ZADEH, L. A. **A fuzzy-algorithmic approach to the definition of complex or imprecise concepts**. International Journal of Man-Machine Studies, v. 8, n. 3, p. 249–291, 1976.

ZHAO, Y. et al. **A review of key techniques of vision-based control for harvesting robot**. Computers and Electronics in Agriculture, v. 127, n. C, p. 311–323, 2016.

ZHU, G.; WU, Z. **The concept change and organization transformation of Vocational Education under the Internet thinking**. 2020 International Conference on Artificial Intelligence and Education (ICAIE). **Anais**...Tianjin, China: IEEE, 2020.