TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol.68, Issue Special I, May, 2025

STUDY ON THE POSITIONING ACCURACY FOR A NEURAL NETWORK CONTROLLED ROBOTIC ARM END-EFFECTOR

Cristian Marius BACICAN, Cristian Emil MOLDOVAN, Mihai MICEA

Abstract: *This paper explores the potential of using a neural network to position a robotic arm's end effector. Traditional methods rely on direct and inverse kinematics, but this study draws inspiration from human hand-eye coordination—how people learn their hand's workspace and reach for objects. In robotics, a similar approach could involve using a depth camera to detect and grasp objects directly. The neural network learns the relationship between joint angles and end-effector positions within the camera's field of view, with applications in manipulation, object handling, and collaborative robotics in dynamic environments. Simulations on a 3-DOF robotic arm tested 10 architectures, revealing feasibility for direct kinematics but challenges in inverse kinematics requiring further research.*
Keywords: *Robotics, Neural Networks, Position, Kinematics, Industry 4.0.*

1. INTRODUCTION

Modern times are witnessing the rapid evolution of technology and the emergence of Industry 4.0 based on the inclusion of cyber physical systems, interconnected and artificial intelligence (AI) driven systems. Furthermore, there is increased emergence of the term Industry 5.0 focused on a human-centric, resilient and sustainable approach towards manufacturing [1, 2].

In this context, robotic systems play a vital role to achieve these ambitious goals. With this in mind, we propose a study on the possibilities of integrating neural networks in robotic-based systems. Our work studies the possibilities of using a neural network to position the end effector of a robotic arm with 3 Degrees of Freedom (DoF). Of interest is to discover the limitations of this approach with respect to positioning accuracy and neural network configuration. Our approach relies on the correspondence between joint angles and resulting tool center point (TCP) position, approach that is similar to the hand-eye coordination in human manipulation. Humans, starting in their infancy, explore their environment also through touch and learn the correlation between the reached arm position and muscle commands. This whole process takes place at a subconscious level, and is achieved through the human's neural network [3].

For a robotic system, in order to train the network, we use synthetic data produced using the kinematic model of the arm, also presented in this paper.

This method can potentially be used in a scenario when robotic manipulation is needed in an unstructured environment with video feedback. The robot arm and camera calibration process can be done through exploration, using machine learning techniques recording the position in the image. TCP positions could be sampled in the camera field of view, generating correspondence points to joint angles, simplifying the calibration process.

Similar work was done in [4] where different network architectures were studied to assess the behavior to control a 2 DoF robotic arm and in [5] where a 5 DoFs manipulator is controlled, our paper proposes a robotic arm with 3 DoFs and we test the positioning accuracy of the manipulator. In [6] the authors propose a neural network to control a parallel robot, in comparison, our research focuses on a serial

robotic arm. In [7] Duka proposes a 3DoF planar manipulator, in comparison, we propose a manipulator capable of moving in 3D space. More recent work on the use of neural networks to control a robotic arm is presented in [8] and [9] papers published in 2024.

This approach, neural networks for robotic arm control, is suited if the robotic arm operates in a highly dynamic, uncertain, or unstructured environment, when the task is too complex for traditional methods (e.g., learning from vision), if the system dynamics are difficult to model mathematically or when using human demonstrations or reinforcement learning to train the robot.

During the next chapters we will present our study, starting with the kinematic model of the robot, training data generation, a presentation on proposed neural networks architectures and their benchmarks and finally a conclusion on our study.

2. 3DOF KINEMATIC MODEL FOR THE ROBOTIC ARM AND DATA GENERATION

In order to generate the training data and test the results, we created a kinematic model for a 3DOF manipulator. The kinematic schema is presented in Figure 1.

The equations used to determine the TCP position are shown in (1)

$$\begin{cases} x = L_1 \cos(\theta_1)\cos(\theta_2) + L_2 \cos(\theta_1)\cos(\theta_2 + \theta_3) \\ y = L_1 \sin(\theta_1)\cos(\theta_2) + L_2 \sin(\theta_1)\cos(\theta_2 + \theta_3) \\ z = L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) \end{cases} \quad (1)$$

Using the equations in (1) we developed a Direct Kinematics (DK) graphical user interface using the Python environment, shown in Figure. 2

For the data generation is considered the direct kinematic model, where joint angles are given and the position of the TCP can be calculated.

The three joint angles were considered in all possible positions considering 1, 5 and 10 degree steps. Rotation angle Theta_1 is considered to be within a range from [0,2π] [radians], while Theta_2 rotation angle is in range [0, π]. This limit is considered to simulate a mechanism for collision avoidance with the surface on which the robot arm is installed. The angle between links L1=100 [mm] and L2=200 [mm] is Theta_3, is assuring the 3rd DOF, is limited to [0, π/2].

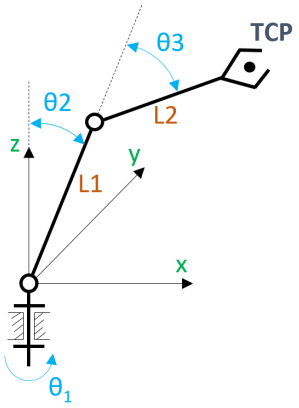
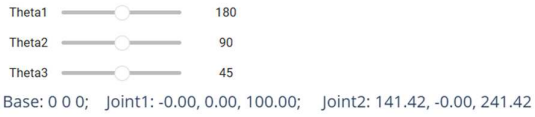


Fig. 1. Kinematic schema for the proposed 3DOF arm.



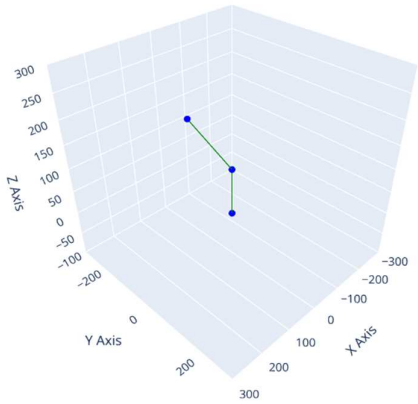


Fig. 2. Graphical user interface for the robotic arm

If all values of all three angles are put into equations, with a change step of 1, 5 and 10 degrees, the total number of end-effector positions will result - see Table 1.

The number of data points increases exponentially with the resolution of the movement. It can be observed that decreasing two times the step of robot movement, from 10 to 5 deg., the number of points is 8 times bigger while decreasing the resolution 10 times, to 1 deg., the number of data points is three orders of magnitude higher.

Table 1

End-effector positions (Data points) and the split considered for the ML model training and test.

Angles change step [deg]	Number of data points	Number of points of training	Number of points for test
10	5832	5000	832
5	46656	40000	6656
1	5832000	-	-

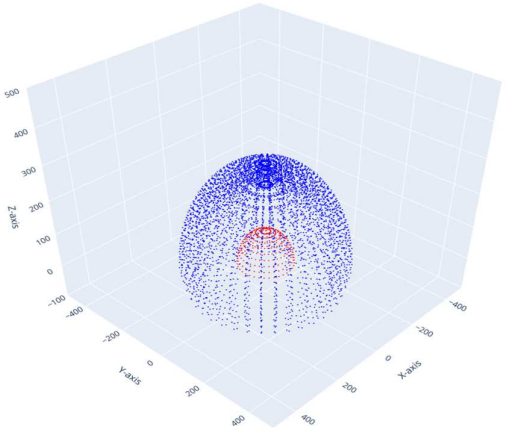


Fig. 3. Plot of all end effector positions.

The study was made with changes of input angles of 10 deg. generating a dataset with 5832 entries. Values were split 5000 for training (~86% of total) and 832 for test (~14% of total).

Even though the machine learning (ML) model is trained with angle values that are divisible with 10 this does not necessarily mean that good predictions cannot be obtained for any angle in between. This implies that the model offers predictions of end-effector position in 3D space for any input angles values even if they are not divisible by 10.

All 5832 data points corresponding to 10 deg. steps were rendered for a brief visual verification in Fig 3, with blue are the end-effector positions and with red the joint between L1 and L2 links of the arm. In other words, the blue points show all possible locations of the end-effector while the red ones represent the corresponding location of the joint of L1 and L2 links, given that all 3 angles swipe in their specific range with a resolution step of 10 deg. The origin of the robot arm, the bottom end of L1 link (the end that is not connected with L2) is in the origin of the coordinate system - point [0, 0, 0].

3. THE NETWORK EVALUATION

For this problem, a fully connected network was considered, shown in Figure 4.

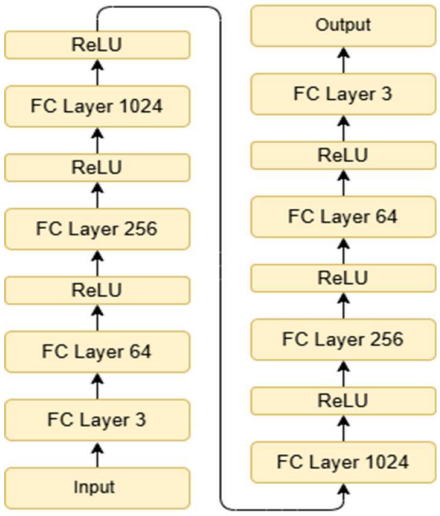


Fig. 4. Neural Network architecture with best performance.

Beforehand, several configurations we tested and results are compared in Table 2, winner being configuration 6. All variants share the same input and output layer configuration.

Table 2

Experiments with different configurations of the network. With bold the best configuration according to RMSE applied on the test set.

No.	No. of hidden layers	Nodes	Test set RMSE [mm]
1	8	64,128,512,1024,1024,512, 128,64	9,56
2	7	64,128,512,1024,512,128,64	4,81
3	7	64,128,256,512,256,128,64	5,10
4	7	64,128,128,256,256,128,64	5,72
5	6	64,128,128,256,128,64	6,73
6	6	64,256,1024,1024,256,64	4,48
7	5	64,512,1024,512,64	6,80
8	5	64,128,256,128,64	9,88
9	5	64,256,2048,256,64	5,11
10	4	256,128,64,32	12,52

The input layer has three neurons corresponding to the three variable angles of the arm. Output layer also has three neurons, each

representing a coordinate of the end-effector in 3D space. In order to find a model that best fits the training data, the network architecture was modified in the sense of number or hidden layers, number of parameters, or both. In all cases after each hidden layer activation function Rectified Linear Unit (ReLU) is used. Implementation was done using the PyTorch library.

3.1 Network Training and evaluation

For training the 10 deg. change in angle was considered resulting in 5832 data points. At training time, 5000 entries are used and 832 are kept for testing. Even though the dataset is generated with angles having ordered values, with increment of 10, during training, the angles - positions pairs are fed to the network in a randomized order, and this order changes at each training epoch. This is normal for the machine learning training process and contributes to a better training performance thus better generalization over new data during inference time.

However, in the robot arm case, each end-effector position is related to the previous one and influences the following since the values of the input angles are consecutive. Unfortunately, this information is lost by randomization thus, the network has no chance to learn the correlation. A more complex architecture could benefit from the relation of consecutive points and expectation is that prediction performance would improve.

At every training step a batch of 32 data entries are used (batch size = 32). Training loss function used is Mean Squared Error (MSE) and the optimizer is Adaptive Moment Estimation (ADAM) configured with a learning rate of 0.001. Each configuration of the network is trained for 100 epochs.

For all models the inference performance is measured using 832 inputs and predictions are checked against the analytic generated values using Root Mean Squared Error (RMSE) method. RMSE was chosen since it has the same unit as the predicted variable making the result more intuitive for understanding model performance.

The results of the experiment are presented in Table 2 and the architecture of the configuration with best RMSE is shown in detail in Figure 3.

3.2 Analysis of the results

Table 3 contains five examples of predictions using the model from Figure 1. Best and worst predictions and their errors are highlighted with bold.

Table 3

Examples of predictions for five different angle configurations.

Point	1	2	3	4	5
Theta 1	150	190	40	280	290
Theta 2	70	60	90	170	90
Theta 3	70	0	30	30	70
x pred.	104,72	-149,21	-74,39	-51,86	**-72,5**
y pred.	-56,57	-25,72	-60,14	274,64	171,84
z pred.	213,99	**261,58**	270,82	-53,59	161,22
x GT	103,06	-147,72	-76,60	-49,74	**-64,28**
y GT	-59,50	-26,05	-64,28	282,07	176,60
z GT	222,53	**259,81**	273,21	-51,04	168,40
error x [%]	1,61	1,01	2,89	4,26	**12,79**
error y [%]	4,92	1,27	6,44	2,63	2,70
error z [%]	3,84	**0,68**	0,87	5,00	4,26

Figure 5 is the graphical representation of the five input angle configurations from Table 3.

Corresponding end-effector position given by analytical method is represented with blue circles while the network predicted end-effector position is represented with green squares.

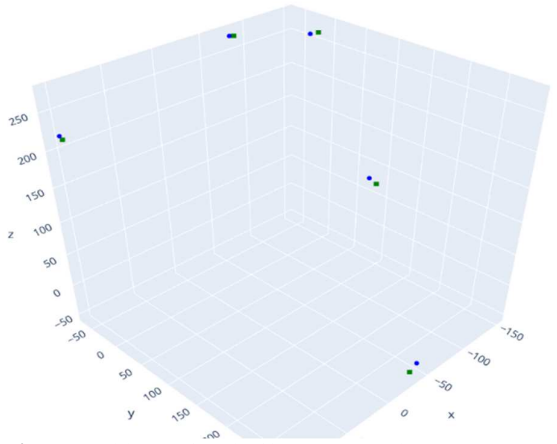


Fig. 5. Positions of end-effector (blue circle) and model prediction (green square) corresponding to values from Table 3.

However, this is just for the five values randomly extracted from the total of 832 test values. From the experiments, we can conclude that the mean error for all 3 predicted values (x, y, z) for all 832 cases is 4.48mm, in the context of an arm with a maximum extension of 300mm (L1 link configured to 100 mm and L2 link to 200 mm).

3.3 Inverse kinematics (IK)

Another experiment is to use the same network architecture and number of hyper parameters for inverse kinematics. It is to note that the network is retrained with input-output pairs on TCP [x, y, z] position and joint angles $[\theta_1, \theta_2, \theta_3]$ values. The results show that over 832 positions the mean error of all three angles is 57.29 [deg]. This is a very high value, which demonstrates that the simple approach for IK has significant less accuracy making is not usable.

Since we use the neural network as a function approximator, the mapping between joint and TCP position is a non-injective surjective function which causes problems in the situation of inverse kinematics. To be more explicit, for the direct kinematics, one set of joint angles produces one TCP position, while for the inverse kinematics, one position can be reproduced by multiple sets of joint angles, this situation is shown in Figure 6.

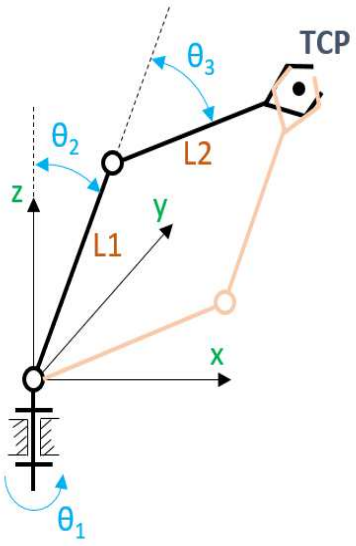


Fig. 6. TCP Positions reached in two configurations

4. CONCLUSION

The analytic and Machine Learning approach for direct kinematic method show rather similar results thus proving that the correlation between input angles and end-effector position can be learned effectively even by a relatively simple deep learning model with fully connected layers.

As per the results shown in paragraph 3.2 and 3.3 the predictions for direct kinematics are far more accurate than inverse kinematics thus a more adequate approach for inverse kinematics is needed rather than just a fully connected deep learning model.

Future research could address this problem by taking into consideration motion, not just points. When in motion, the robotic arm follows a sequence of arm configurations that change minimally when moving from one point to the next in order to reproduce the whole trajectory. Considering this, a restricted model could be developed.

The neural network control for robotic arm methods could have potential impact in situations where visual feedback is used to allow automatic handling of objects and the environment is highly unstructured. Potential industries that could benefit from this technology are smart manufacturing and assembly, for example in the automotive industry, the production of wiring harness systems; warehouse automation and logistics, pick, sort and pack operations; agriculture and smart farming; food processing and packaging; construction, such as brick or tiles laying; recycling and waste management.

This method extends the state of the art and provides an additional option to solve tasks that can be automatized using robotics

5. REFERENCES

[1] Xu, X., Lu, Y., Vogel-Heuser, B., Wang, L., *Industry 4.0 and Industry 5.0—Inception, conception and perception*, Journal of Manufacturing Systems, ISSN 0278-6125, 2021.

[2] Mocan, B., Fulea, M., Olaru, M., Buchmüller, M., *From Intuitive*

Programming of Robotic Systems to Business Sustainability of Manufacturing SMEs. Amfiteatru Economic, 18(41), pp. 215-231. 2016.

[3] Hoffmann, M., Marques, H., Arieta, A., Sumioka, H., Lungarella, M., Pfeifer, R., *Body Schema in Robotics: A Review*, IEEE Transactions on Autonomous Mental Development, vol. 2, no. 4, pp. 304-324, Dec. 2010.

[4] Călin, A, Pop,a C., Moldovan, C, Davidescu, A., *Model-Free Reaching of a 2-DOF Robotic Arm Using Neural Networks*, Mechanisms and Machine Science, Springer, ISBN978-3-031-25655-4, 2023.

[5] Aravinddhakshan, S., Apte, S., Akash, S.M, *Neural Network Based Inverse Kinematic Solution of a 5 DOF Manipulator for Industrial Application*, Journal of Physics: Conference Series, 2021.

[6] Gholami, A., Homayouni, T., Ehsani, R., Sun, J.Q., *Inverse Kinematic Control of a Delta Robot Using Neural Networks in Real-Time*, Robotics, 2021.

[7] Duka, AV, *Neural network based inverse kinematics solution for trajectory tracking of a robotic arm*, Procedia Technology 12, 2014.

[8] Dorman, J,D,, Gupta, A., Gill H,S., *FNN-Based Inverse Kinematics for Efficient Trajectory Planning in Industrial Robots*, IACIS, ISBN:979-8-3503-6066-0, 2024.

[9] Ojer, M, Etxezarreta, A, Kortaberria, G., et al., *High accuracy hybrid kinematic modeling for serial robotic manipulators*, Robotica. 2024.

Studiu cu privire la precizia de poziționare a unui braț robotic controlat cu rețele neuronale

În această lucrare de cercetare se explorează posibilitatea utilizării unei rețele neuronale artificiale pentru a poziționa efectorul final al unui braț robotic. Metoda clasică utilizează modelul cinematic direct și invers pentru a realiza această sarcină, dar inspirația pentru acest studiu provine din analiza modului în care omul rezolvă această problemă, anume coordonarea ochi-mână. Situația echivalentă în robotică ar fi utilizarea unei camere video pentru detectarea și apucarea obiectelor. Abordarea bazată pe rețele neuronale artificiale ar permite învățarea funcției de corespondență între valorile unghiurilor articulațiilor robotului și poziția efectorului final în imagine, permițând realizarea de aplicații de manipulare. Experimentele cu rețele neuronale artificiale din această lucrare au fost realizate pe un model matematic al unui braț robotic cu trei grade de libertate. Acesta a fost folosit pentru generarea setului de date de antrenament, după care s-au evaluat 10 arhitecturi de rețele neuronale din punct de vedere al performanței.

Cuvinte cheie: *robotică, rețele neuronale, poziție, cinematică, industrie 4.0.*

Cristian Marius BACICAN, PhD candidate, Eng., University Politehnica Timisoara, Department of Computer and Information Technology**,** cristian.bacican@student.upt.ro

Cristian Emil MOLDOVAN, PhD, Associated Professor, University Politehnica Timisoara, Department Mechatronics, cristian.moldovan@upt.ro, 0744575840

Mihai MICEA, PhD, Professor, University Politehnica Timisoara, Department of Computer and Information Technology, mihai.micea@upt.ro, 0725890914