Using Artificial Neural Network for Solving Inverse Kinematic Task of 2R Planar Robot-Manipulator

Kamila Tanyrbergenova[1], Tolkyn Mirgalikyzy[1], Balgaisha Mukanova[2], and Mikhail Posypkin[3]

[1] *L.N. Gumilyov Eurasian National University, Astana, 010000, Kazakhstan*
[2] *Astana IT University, Astana, 010000, Kazakhstan*
[3] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119991, Russia*

Abstract

The research considered the motion control of a 2R planar robot consisting of two links connected in series through two revolute joints, suitable for the task of simulating the human arm and other similar mechanisms. The aim of the study was to apply Machine Learning (artificial neural networks, ANN) methods to control the position of the 2R planar robot's end-effector within the workspace along a given trajectory. The possibility of using ANNs to achieve a single solution of the Inverse Kinematics Task is shown.

Keywords

2R robot-manipulator, Neural Networks, ANN, robotics

1. Introduction

Nowadays, robots have deeply penetrated into our everyday life and are used in a variety of practical tasks. Robotics is an actively developing field, the achievements of which are used in manufacturing, medicine and many other spheres. For example, the work [1] gives a voluminous overview of robot models used in the medical sphere. Today, robotic manipulators of the "multi-link" type are popular and have not lost their relevance for research purposes for many years. Palleschi and his co-authors [2] "presented a time-optimal trajectory planning algorithm for moving a flexible joint robot along smooth parametrized paths" for a 2R planar robot. In work [3] 2R planar robot is also considered, and the authors "designed a genetic algorithm to perform time-optimal control of robotic manipulators along specified paths, subject to torque constraints". Work [4] also deals with canonical SCARA 2DoF. Based on all of the above, robotic manipulators of the "multi-link" type are popular and in demand due to their simple design and convenience in performing calculations and experiments.

The Inverse Kinematics Task (IKT) in robotics is important. The solution of the IKT is often the basis for solving other problems facing the mechanical system, for example, for calculating the workspace of a robot. It is known that the IKT is often has more than one solution. The solution of robot's IKT is considered in many works, the problem does not lose its relevance [5]-[8]. It is solved in different ways for different mechanical systems. The most popular are classical mathematical (including geometric) methods, as well as widely used methods of machine learning. In work [5] solution of GLC for multilink with spherical joints (7 links) with application of mathematical apparatus is considered. In the work of Mukanova B.G. solves IKT and Direct Kinematics Task (DKT) for RPR robot also with the use of classical mathematical methods and transformations [6].

Proceedings of the 7th International Conference on Digital Technologies in Education, Science and Industry (DTESI 2022), October 20–21, 2022, Almaty, Kazakhstan
EMAIL: kamila.tanyrbergen@gmail.com (Kamila Tanyrbergenova); m_t85@mail.ru (Tolkyn Mirgalikyzy); mbsha01@gmail.com (Balgaisha Mukanova), mposypkin@frccsc.ru (Mikhail Posypkin)
ORCID: 0000-0002-0703-4057 (Kamila Tanyrbergenova); 0000-0002-0248-9220 (Tolkyn Mirgalikyzy); 0000-0002-0823-6451 (Balgaisha Mukanova), 0000-0002-4143-4353 (Mikhail Posypkin)

W.H. Zayer with co-authors proposed the use of FuzzyNeuralPetriNet to solve the IKT and the DKT of a 2DoF robot manipulator [7]. Using ANN to solve the IKT for multi-link is also applied in [8] for PUMA 560 robot, and the training result equals to 90 %. Solving IKT is an important step in solving robotics problems.

Machine learning methods are extensively used in robotics not only for solving DKT (direct kinematics task) and IKT. Machine learning makes it possible to solve problems where it is difficult to derive dependencies by classical methods. The review by Laith Alzubaidi and his co-authors[9] states that "recently, machine learning (ML) has become very widespread in research and has been incorporated invariety of applications,including text mining, spam detection, video recommendation, image classification, and multimedia concept retrieval" and that "the effectiveness of an algorithm is highly dependent on the integrity of the input-data representation". For example, holonomy systems in solving DKT, IKT have the problem of geometric inaccuracy of the obtained result. Article [10] discusses a solution to this problem using the Artificial Neural Network (ANN) and "is trained to predict the position error in the workspace for the given input coordinates of the planar geometry to be sewed in the fabrics".

Robotics is an actively developing field, but it is likely that the highest peak of development in this area has not yet been reached. First, there is no single solution (framework) that fits all robot models. You have to use highly specialized tools to solve narrow problems. Second, each robot is a mechanism with unique features that require an individual approach. This relates to the description of the kinematics, the dynamics of a robot, how links are connected and their order, whether closed or open kinematic chains are used, whether the system is holonomous or whether odometry must be used, whether a solid body forms the basis of the mechanical system, and other nuances. Therefore, there are still many discoveries to be made in robotics, and often today they refer to specific robot models aimed at solving a narrow range of problems. In [11] described trends in the industrial robotics sector and graphically shows the growth of number of publications in the field of industrial robotics. The paper [12] shows trends in the evolution of service robots and lists some unresolved issues in robotics, such as "the lack of generalization and formalism in classifications and taxonomy, the current perceived utilitarian value, battery and autonomy modelling and estimation, ethics, and even design problems related to gender biases based on the occupation of the robot".

It is a well-known fact that in robotics there are two kinematics tasks: direct and inverse (IKT and DKT). The solution of the DKT gives us an unambiguous answer, while the IKT does not always give a single or unambiguous solution. We must rely on the goal pursued in solving the IKT, which may be to find multiple solutions and further process the results, or to derive a single solution that fits the criteria. IKT is an important step in solving many kinematics problems in robotics, e.g., solving IKT is an important step in computing the Workspace of a robot. It should be noted that the application of Artificial Intelligence (AI) increasingly overlaps with robotics problems and provides many fresh ways to solve different problems. The exact location of not only the end-effector, but also other links in the path can be critical, depending on the tasks the robot is designed to perform. For example, when teaching a robot to repeat movements to train flexor-extensor muscles for those in need of rehabilitation, it is not enough for a robot to follow the end-effector movement along a given trajectory; we also need the most precise movement of all links due to the limited degree of freedom of the human hand. There are many tasks where this criterion is important according to the condition or formulation of the problem. Also, the IKT solution is important in calculating the robot's workspace. Knowing the robot's workspace helps to ensure that the robot's mechanisms and parts work less traumatically for the robot. This study proposes to solve the inverse kinematics problem for 2R planar robot-arm using Artificial Neural Networks (ANN) apparatus and further use the results to build the workspace of the robot, to move the robot along the trajectory taking into account the more exact location of all links. The features and advantages of the proposed solution are given in further sections of the paper.

2. Description of the robot-manipulator

In this work we operate with 2R planar robot-arm to test hypotheses and do research. Due to the simple design and ease of prototyping the robot, the demand for this model in different industries, its accessibility and canonicality, the significance of the results for both, training purposes and practical purposes, is achieved.

This work considers the IKT for a 2R planar robot manipulator using Machine Learning methods to obtain a single (unambiguous) solution on output.

As mentioned above, the study deals with a 2R planar robot because it has a simple construction consisting of two links connected in series through two revolute joints, and is suitable for the task of simulating the human hand and other similar mechanisms. The robot mechanism under consideration is a holonomous system and can be described geometrically. The geometrical scheme of 2R planar robot with notations is shown in Figure 1.

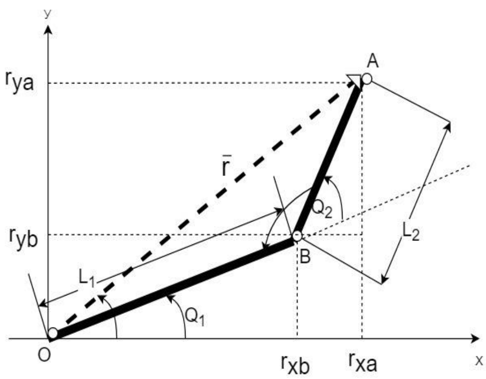

Figure 1: 2R Planar Robot Geometric Schema

Let introduce the following notations:

O – is the center of the first link (base) or the junction point of the base and the first link,

OB – is the first link,

BA – is the second link,

point*B*- is point of connection of the first and second links,

point*A* – is the end-effector coordinate,

Q_1 – is the angle between the *x*-axis and the link *OB*,

Q_2 – is the angle between the *OB* link and the *BA* link,

L_1-length of the first link,

L_2 – length of the second link,

r_{xb}, r_{yb} –projections of the length of link L_1 on coordinate axes,

r_{xa}, r_{ya}– projections of the radius-vector determining the position of the end-effector on coordinate axes,

$\bar{r}$ - the radius-vector determining the position of the end-effector in space (the vector connecting the 0 and the target point).

The degree of freedom of the robot equals two (DOF=2).

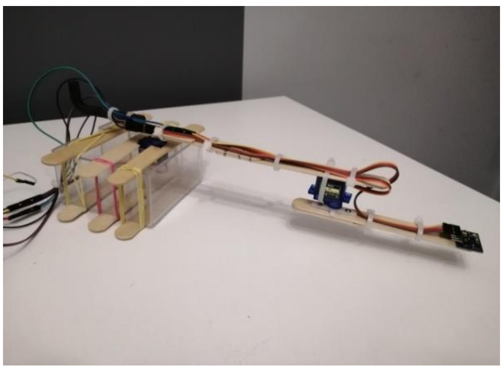

Figure 2: 2R Planar Robot Prototype

According to the described geometrical scheme, the prototype 2R Planar Robot - manipulator was assembled using the ArduinoUno platform based on the ATMEGA328 microcontroller and using two servo motors at the points O and B.

Thus, two servo motors (model SG 90) play actuators roles in the physical implementation. The implemented physical model of the 2R Planar Robot prototype is shown in Figure 2.

The principal model of the robot, implementedin the CoppeliaSim environment, is shown in Figure 3.

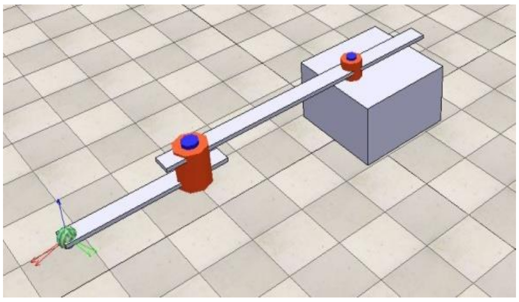

Figure 3: 2R Planar Robot CoppeliaSim model

Note that due to the fact that 2R Planar Robot is a holonomous system, only the dimensions of the first and second links mattered for calculations. The links length of the mechanical system prototype was L1 =0.16 m and L2 =0.1 m. The parameters of the robot adopted in the study are described in Table 1. Robot design was chosen based on characteristics of used servomotor models specifications and the weight of the elements that were used in the construction. The characteristics of the SG 90 servomotor can be taken from the manufacturer's official website. According to them the servomotor can hold up to 1 kg of weight on the end of the arm with the length 0,01 m, the servomotor force is 1.2-1.6 kg·cm at voltage 4.8 V. The maximum speed of the SG 90 servomotor is 1.046 radians in 0.1 s (the specification show it as 60 degrees/ 0.1 s).

Table 1
Robot parameters

Parameter	Length (m)	Width (m)	Height(m)	Weight(kg)
First Link (L_1)	0,16	0,017	0,002	0,016
SecondLink (L_2)	0,10	0,017	0,002	0,023
Base	0,095	0,095	0,05	Not considered

Let's present the specifications of servomotors in table (see Table 2). As you can see from the table, SG 90 servomotor is restricted and can rotate from 0 to 180 degrees (90 degrees to one side and 90 degrees to the other side). In the calculations we denote the range of motion from -90 to 90 degrees for a convenient representation on the Cartesian coordinate system. In fact, the maximum boundary values are not attainable for the working area. We take this fact into account when constructing the robot workspace.

Table2
Servoparameters

Parameter	Max angular velocity	Min position (degrees)	Max position (degrees)	Mass (kg)
Servo 1	1,046 rad/0.1sec	-90	90	0,009
Servo 2	1,046 rad/0.1sec	-90	90	0,009

By projecting the radius-vector$\bar{r}$onto the coordinate axes, express the values of its projections onto the coordinate axes using the law of cosines. Thus, we obtain the kinematic equations of motion describing the mathematical solution to the DKT (1).

$$\begin{cases} r_{xa} = L_1 * \cos(Q_1) + L_2 * \cos(Q_1 + Q_2) \\ r_{ya} = L_1 * \sin(Q_1) + L_2 * \sin(Q_1 + Q_2) \end{cases} \quad (1)$$

Performing calculations by formula (1) in the loop, taking into account the constraints on the servomotors, we obtain the set of workspace points of the robot-manipulator. The boundaries of the workspace are unreachable for the end-effector, so we remove them from the set of points of the workspace (highlighted in black). The calculation step equal to 1 degree, angles from -90 to 90 degrees. Then subtract from the set the points being the boundary points. Based on this, with regard to the given lengths of the links, the workspace has the surface shown in Figure 4.

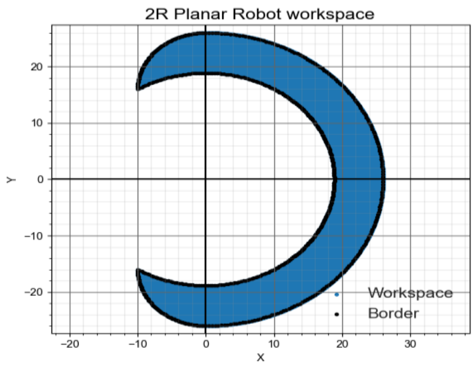

Figure 4: 2R Planar Robot Workspace with given constraints

3. Problem Formulation

The following definitions and terms should be introduced for the convenience of the following presentation:

- *A Transitional link*–is any movable link that participates in the movement and affects the coordinates of the end-effector, on which the end-effectoris not located and the coordinates of which do not coincide with the coordinates of the end-effector. In our case, we highlight the link L_1as the only intermediate link.
- The coordinates of point B are *coordinates of the Transitional link.*
- The set of allowable *coordinate points of the Transitional link* is called *a workspace of the Transitional link.* Obviously, in this case *the workspace of the Transitional link is an arc of a circle with radius L_1.*
- *A Target link* – is a link those coordinates are associated with the coordinates of the end-effector.
- When controlling the motion of the robot, we will be based on the *"exactly-the-same"* principle. This principle implies that all intermediate links of the robot and the target link of the robot will repeat the necessary movements as accurately as possible, and not only the trajectory set for the end-effector.
- Thus, we implement the *ESM method* with with application to ANNs with different topologies and models, inputs and outputs, to obtain a single output. This makes it possible to train the robot to move by following the correct location of the target and intermediate links on the selected surface according to the "exactly-the-same" principle.

The formulation of the problem is below:

Given: The workspace of the robot is defined and the trajectory is given by *a sequence of discrete points* along which the end-effector should move, with a given time step Δt=(t1-t0)/H -H number of steps.

It is required to set the angles of rotation on the actuators in real time, so that at the next moment of time +Δt the working tool moves to a given point on the trajectory. Perform a calculation for a given time interval, with regular step Δt, such that the robot follows the given trajectory from the initial to the final position.

The restrictions on the workspace and the technical limitations on the torques on the actuators must be satisfied.

4. Concept of the proposed method

We propose the following method for solving IKT for a 2R multi-link robot:

- The IKTis solved using Artificial Neural Networks (ANN). The trajectory, time step of links rotations is given.
- At the input of the ANN is the coordinates of the intermediate link and the coordinates of the end-effector.
- The proposed ANN model gives a single solution on output.

Thus, ANN gives a single and unambiguous output result. The input data, which contain the coordinates of the intermediate link, gives opportunity to avoid the ambiguity of the solution. The specific feature of the proposed solution is that the input of the ANN is consist of, in addition to the coordinates of the end-effector, also the coordinates of the intermediate link (point B).

When data are sequentially sent to the input of the ANN, it should be noted next statements:

- taking into account that each subsequent input vector is the coordinates of points of an arbitrary trajectory, when moving with a step Δt;
- taking into account that the coordinates of points are taken with a fixed time step, by the results of the output vector it is possible to reproduce the laws of change of linear speeds, linear and angular accelerations on the route, if necessary.

5. ANN Model

The advantage of *the proposed ANN topology*, that implements the *ESM method*, is that, with taking into account of the intermediate link position, the output is a single solution of the IKT for 2R planar robot-arm. This allows the trained robot not just to repeat the required trajectory by the end-effector, but also to repeat completely the movements of all intermediate links.

So, using these solutions, it is possible to set the trajectory and positions of its intermediate links to the robot and teach it to repeat exactly all movements according to the "exactly the same" principle. The total time, the resulting law of velocity change, the energy expended on the motion, and other dynamic indicators are not optimal. These indicators are called intuitive outputs based on the results of the conversion of the ANN output data.

The modules developed to implement the goal of the research were written in Python. Different models of a fully coupled ANN with different topologies were implemented using the frameworks Tensor Flow and Keras. The graphs were constructed with the help of TensorBoard, Matplotib, and Graphviz libraries.

Implemented research steps to make work on the simulator and the robot model prototype and to find a suitable ANN topology are below:

1. Designing the circuit of the future robot prototype, description of the robot kinematics, model implementation in CoppeliaSimEDU environment.
2. Developing of a module to build the robot's workspace and its boundaries.
3. Developing of a module for generating a pseudo-random trajectory within the workspace.
4. Assembling the construction of the robot prototype on the ArduinoUno board using Atmega328 microcontroller.

5. Developing of a module for generating training and test samples based on the movements of the robot model in the CoppeliaSim environment.
6. Designing the ANN topologies, verification, training, testing.
7. Visualization of results, evaluation of ANN performance, and selection of the best models.
8. Developing of module for testing, verification of training results and recreation of robot movements on robot model in CoppeliaSim environment.
9. Testing of the model implemented on ArduinoUno board.

The training samples was generated in 1-degree increments, because a larger increment had insufficient effect on the learning process of the ANN, while a smaller increment did not produce better learning results.

Pre-testing of the training results was performed using the simulation on the robot model implemented in the CoppeliaSim environment. Then, after passing the pre-test, the learning results were tested physically on the prototype of the serial 2R Planar robot assembled on the ArduinoUno board with the Atmega328 microcontroller. The computational load of working of the Neural Network was performed on the PC, and training results was transferred to the microcontroller through the interface Firmata.

5.1.1. The ANN model implements the "exactly the same" principle. Topology description and training result

Preparation of the training and test samples. The set of consecutively arranged trajectory points from the working area, calculated in the range of the angular actuator [-90º, 90º] with step 1º and the corresponding set of intermediate points is a training sample for the ANN. The training sample with this step is chosen due to the good result obtaining while working. A larger step did not give a better result in the test. A smaller step did not give a better result. The size of the training sample is 32040. Four parameters are input:

- r_{xa}, r_{ya} – determine the coordinates of point A on the coordinate plane;
- r_{xb}, r_{yb} – determine the coordinates of point B on the coordinate plane.

Calculated by the following formula (Formula 2):

$$\begin{cases} r_{xb} = L_1 * \cos(Q_1) \\ r_{yb} = L_1 * \sin(Q_1) \end{cases} \quad (2)$$

The injection of the coordinates of *the Transitional link* into the ANN allows us to avoid the problem of choosing solution between many solutions of the IKT (in our case is two solutions). At the output we have two parameters: Q1 and Q2-angles of rotation of the actuator that determine the position of the links in the angular coordinate system.

The ANN, which gave the best result in realization of "exactly the same" principle consists of an input layer of dimension 4, hidden layers of dimension 9 and 12 neurons, an output layer of two neurons (see Table2). Visual diagram (topology) of the full-connected ANN is shown in Figure 5.

Table2

Model: "sequential"

Layer (type)	Output Shape	Param #
Input Layer	4	4
Dense	9	45
Dense_1	12	120
Dense_2	2	26
Total params:	195	
Trainableparams:	195	

The following activation functions built into Keras are used for the layers:

- The activation function - hyperbolic tangent calculates according to the following formula (3) and gives the result at the neuron output in the range [-1,1].

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (3)$$

- The sigmoid activation function, gives a result on the neuron output in the range [0,1] (Formula 4):

$$f(x) = \frac{1}{1+e^{-x}} \quad (4)$$

These activation functions were used in the final ANN model because they gave the best learning results and are suitable for solving similar problems, are the most commonly used and well-proven. The custom activation functions were not necessary and their development was not the purpose of the study.

To measure the accuracy we used *Accuracy* metric, which is also an embedded metric. This metric was chosen for the task due to its simplicity and clarity. The built-in losses was chosen as *Loss function MAPE* (mean absolute percentage error) calculated using formula (5) below. The Inverse Error Propagation Algorithm was used for training. The *Adamax* function was used as an optimizer and the *learning rate* parameter for the ANN was equal to 0.01. More details on metrics, loss functions, and optimizers can be found in the Keras documentation.

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{(\text{y\_true} - \text{y\_pred})}{\text{y\_true}}\right| , \qquad (5)$$

Where y_{true} is the actual value and y_{pred} is the forecast value.

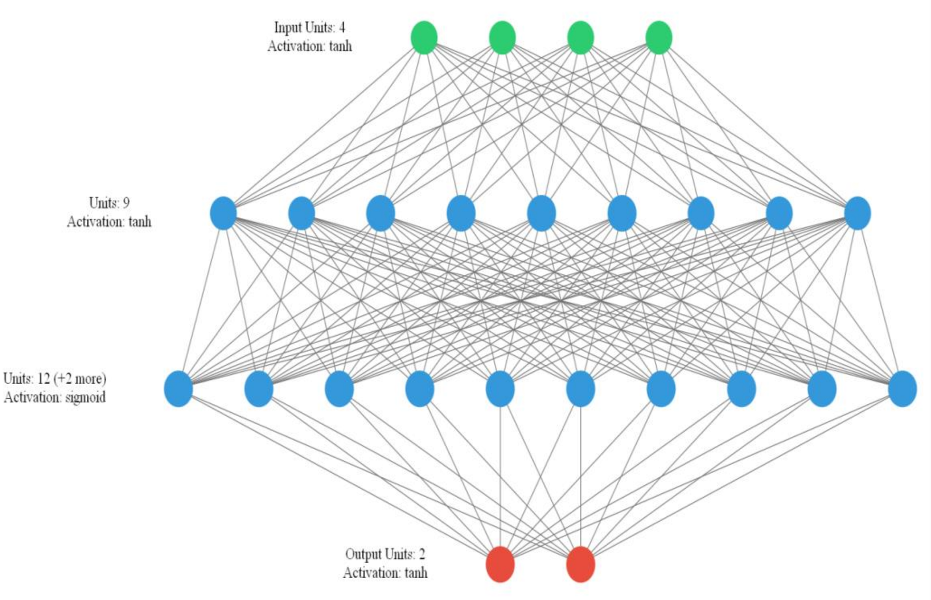

Figure 5: Neural network graph indicating the activation function on the layer

6. Results and Discussion

The results of training and testing of the ANN are given in this section. The evaluation of the trained ANN during testing with test samples (model.predict()) was performed with the standard function model.evaluate().

6.1. Model Training and Testing Results

The designed ANN model was trained with a final Accuracy equals 98.84% (see Figure 6a). A total of 500 epochs were allocated for training to achieve the specified Accuracy. The final value of the loss function was 5.16% (see Figure 6b). Optimizer Adamax was used with a learning rate equal to 0.01.

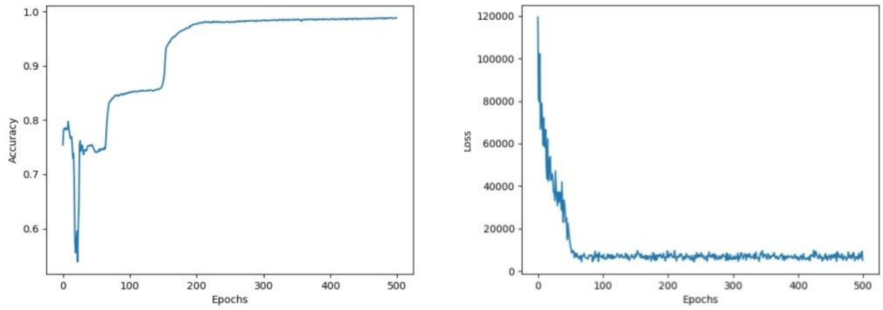

Figure 6: a) Accuracy b) Loss

Next, the Random Trajectory Generation module generated a test sample with a step Δt=10 ms and an angle of no more than 50 degrees, taking into account the maximum speed (Figure 7). It represents the points of the trajectory that the robot prototype should repeat.

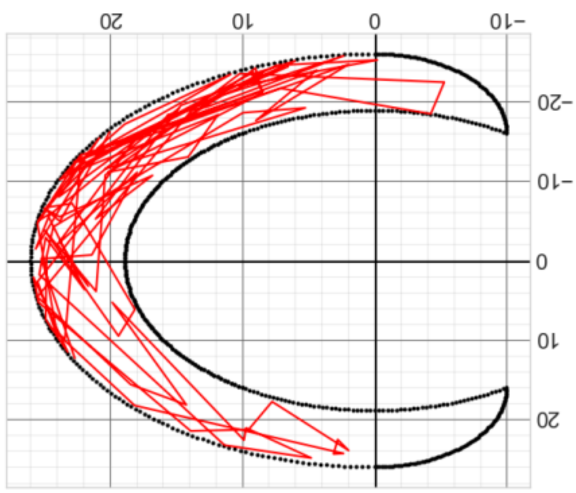

Figure 7: Random generated trajectory

The ANN performed well in testing and showed Accuracy equals 95%. The graphical result of the output parameters Q_1 and Q_2 is shown in Figure 8. The blue line indicates the test data, the orange line is the result of the ANN operation. As shown in the graphs, the lines are almost repeats each other.

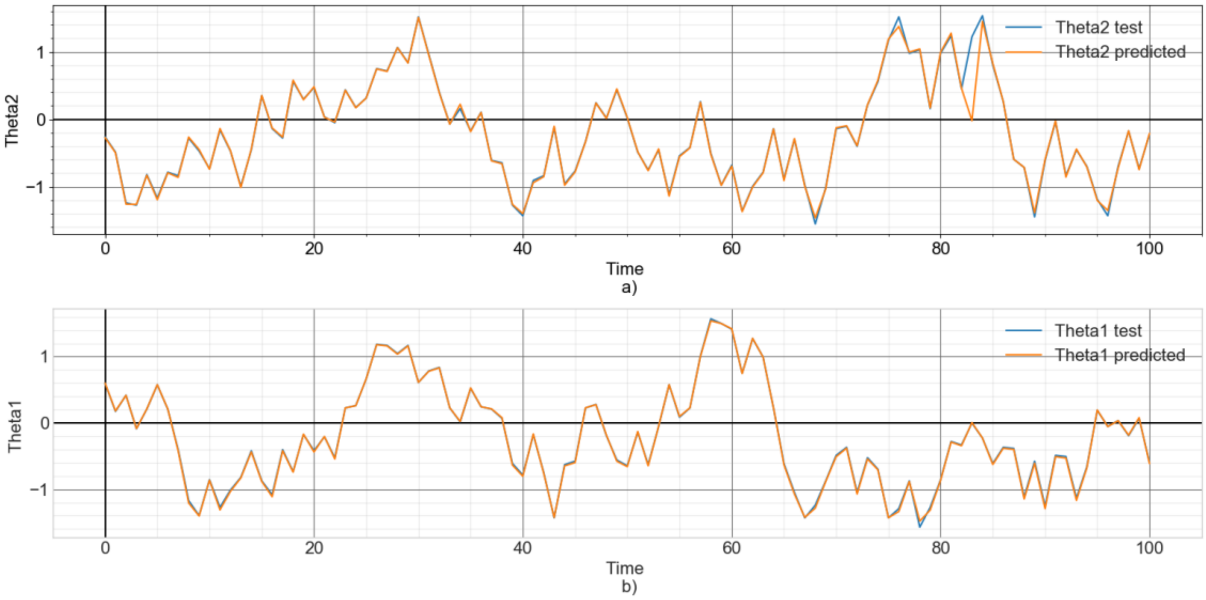

Figure 8: a) Q_2 is Theta2 angle b) Q_1 is Theta1 angle

Another property of the proposed ANN is that it always produces a result that is within the workspace of the robot even when input data has values outside of the workspace. This allows to save the mechanisms of the robot. And it allows to put the constraints associated with the robot design into the process of the ANN learning and training.

7. Conclusion

The main result of the study is the proposed model of the ANN for solving the IKT, which implements the proposed ESM method. The ANN model solves the problem of finding angles Q_1 and Q_2, having information about the coordinates of the target link and the coordinates of the transitional link, it implements the "exactly the same" principle. The accuracy of the proposed final ANN was 98% in training and not lower than 95% in test samples. The proposed ANN model is called ESMNeuralNetwork 1.0v.

The results of this study can be applied to robots with a similar configuration, if:

- It is important to obtain a single solution of IKT;
- The working area of an end-effector and transitional links is known.
- There is a need to specify or calculate the total time taken to move along a trajectory.
- There is a need to obtain the law of angular velocity change on a trajectory and other characteristics of movement.
- The coordinates of target trajectory coordinates and transitional links coordinates.

The results of the work can be adapted for flat artist robots, flexor-extensor muscle trainer robots, and others with a similar configurations and problem formulation.

Results of work can be developed for other robot models, as well as to improve the mechanical characteristics of motion, continue research in the direction of calculating the optimal trajectory, the laws of change of the control moments on the actuators, improve the comfort of movement and minimize energy costs.

8. Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

9. References

[1] J. Holland, L. Kingston, C. McCarthy, E. Armstrong, P. O'Dwyer, F. Merz, and M. McConnell, Service Robots in the Healthcare Sector, Robotics 10.1 (2021) 47. doi:10.3390/robotics10010047.
[2] A. Palleschi, R. Mengacci, F. Angelini, D. Caporale, L. Pallottino, A. De Luca, M. Garabini, Manolo, Time-Optimal Trajectory Planning for Flexible Joint Robots, IEEE Robotics and Automation Letters (2020) 1–1. doi:10.1109/LRA.2020.2965861.
[3] E. Ferrentino, A. Della Cioppa, A. Marcelli, P. Chiacchio, An Evolutionary Approach to Time-Optimal Control of Robotic Manipulators, Journal of Intelligent & Robotic Systems (2019). doi:10.1007/s10846-019-01116-9
[4] K. Khalid, A. A. Zaidi and Y. Ayaz, Optimal Placement and Kinematic Design of 2-DoF Robotic Arm, in: Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies, IBCAST, 2021, pp. 552-559, doi: 10.1109/IBCAST51254.2021.9393255.
[5] S. Ondočko, J. Svetlík, M. Šašala, Z. Bobovský, T. Stejskal, J. Dobránsky, L. Hrivniak, Inverse Kinematics Data Adaptation to Non-Standard Modular Robotic Arm Consisting of Unique Rotational Modules, Applied Sciences 11.3 (2021) 1203. doi:10.3390/app11031203.
[6] B. Mukanova, Control of Actuators Torques for Optimal Movement along a Given Trajectory for the DexTAR Robot, Journal of Applied and Computational Mechanics 7.1 (2021) 165-176. doi: 10.22055/JACM.2020.34650.2449.
[7] W. H. Zayer, Z. A. Maeedi, and A. J. F. Ali, Solving forward and inverse kinematics problem for a robot arm (2DOF) using Fuzzy Neural Petri Net (FNPN), Journal of Physics: Conference Series 1773 (2021) 012009. doi:10.1088/1742-6596/1773/1/012009.
[8] L. Aggarwal, K. Aggarwal, R. J. Urbanic, Use of Artificial Neural Networks for the Development of an Inverse Kinematic Solution and Visual Identification of Singularity Zone(s), Procedia CIRP 17 (2014) 812–817. doi:10.1016/j.procir.2014.01.107.
[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, Journal of Big Data 8.1 (2021). doi:10.1186/s40537-021-00444-8.
[10] M. Praveen Kumar, S. D. Ashok, Artificial neural network based geometric error correction model for enhancing positioning accuracy of a robotic sewing manipulator, Procedia Computer Science 133 (2018) 1048–1055. doi:10.1016/j.procs.2018.07.069.
[11] A. Dzedzickis, J. Subačiūtė-žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, and V. Bučinskas, Advanced applications of industrial robotics: New trends and possibilities, Applied Sciences (Switzerland) 12.1 (2022). doi:10.3390/app12010135.
[12] J. A. Gonzalez-Aguirre, R. Osorio-Oliveros, K. L. Rodríguez-Hernández, J. Lizárraga-Iturralde, R. Morales Menendez, R. A. Ramírez-Mendoza, M. A. Ramírez-Moreno, J. D. J. Lozoya-Santos, Service Robots: Trends and Technology, Appl. Sci. 11 (2021) 10702. URL: https://doi.org/10.3390/app112210702.