

*This paper investigated the dependence of the stability of the knee joint, when exposed to external rotational load on the lower leg, on the position of the graft of the tendon of the popliteal muscle. Estimation finite-element models of the right knee joint of an adult were constructed, which included the articular ends of the bones that form this joint, as well as its main ligaments. The models reflected a surgery to restore posterolateral angle structures and differed only in the position of the popliteal tendon graft. That position was set by the point of attachment of the graft to the posterior surface of the tibia. At the same time, the fixation point changed both vertically and horizontally, in the frontal plane. In addition, a control model was built in which the hamstring tendon was completely absent. As a result of the calculations, patterns of the distribution of the fields of movement of the points of the finite-element model were obtained. As criteria for assessing the effectiveness of the selected position of the graft, movements of the lower leg model in the horizontal plane were proposed. Analysis of the results of the calculations showed that the greatest movements in all directions were obtained in the control model, in which the hamstring of the popliteal muscle was absent. The magnitudes of the considered movements derived from the control model exceeded the same values in the model with minimal movements by 17, 37, 17, 32, and 16%. From the point of view of the stability of the tibia under rotational load, the most effective was the fixation of the graft on the posterior surface of the tibia as laterally as possible and closer to its articular surface. This is indicated by the magnitude of the movements, which, in this case, turned out to be the smallest in all directions*

**Keywords:** knee joint, hamstring tendon, external rotational load, graft, finite-element model, stiffness characteristics

# DEVELOPING APPLICATION TECHNIQUES OF KINEMATICS AND SIMULATION MODEL FOR INMOOV ROBOT

**Chingis Kenshimov**

PhD, Associate Professor, Leading Researcher\*

**Talgat Sundetov**

Corresponding author

Doctoral Student, Researcher\*

E-mail: [sundetovtalgat@gmail.com](mailto:sundetovtalgat@gmail.com)

**Magzhan Sarzhan**

Doctoral Student, Researcher\*

**Murat Kunelbayev**

Research Associate\*

**Madina Kutubayeva**

Doctoral Student Computer Science and Software

L. N. Gumilyov Eurasian National University

Satpayev str., 2, Nur-Sultan, Republic of Kazakhstan, 010008

**Arman Amandykuly**

Student, Researcher\*

\*Laboratory of Artificial Intelligence and Robotics

Institute of Information and Computational Technologies

Shevchenko str., 28, Almaty, Republic of Kazakhstan, 050010

Received date 27.06.2022

Accepted date 22.08.2022

Published date 30.08.2022

**How to Cite:** Kenshimov, C., Sundetov, T., Sarzhan, M., Kunelbayev, M., Kutubayeva, M., Amandykuly, A. (2022). Developing application techniques of kinematics and simulation model for InMoov robot. *Eastern-European Journal of Enterprise Technologies*, 4 (7 (118)), 79–88. doi: <https://doi.org/10.15587/1729-4061.2022.261039>

## 1. Introduction

The prerequisite for the realization of this work is a steady growth of interest in mobile robotics systems on the part of both scientific and educational institutions. The implementation of such a system is a complex and time-consuming process, which cannot be performed at the modern level without using a set of specialized software and hardware tools to help the developer at various stages of design, programming and debugging.

The solution to this problem is to develop a verbal-interactive robot-technical complex, with support for modern mobile control technologies. This will allow to immediately begin to get acquainted with the hardware of electronic devices for various purposes, get skills in programming microcontrollers and program debugging. Availability of verbal-interactive training platform at a reasonable price will allow engineers and scientists from different fields, who need to automate their approaches, to test their ideas and algorithms on prototypes of robotics, to quickly prototype and program such models.

The main difference of this work from similar (ongoing) works in other countries is its focus on the development of a humanoid robotic system with an open architecture based on modern mobile technologies.

Humanoid robots are commonly referred to as those robots that at least partially resemble humans. Most such humanoid robots have a torso, two legs, two arms, and a certain shape of head. Some of them may have a face that can change expression to varying degrees. Although the idea of creating this type of robot has been around for a long time, it is only in the last decade that relatively humanoid robots have actually made quite a lot of progress.

The term android is often used as simply a synonym for humanoid robots, but it can also be used to describe a robot more specifically. Some people believe that, strictly speaking, android is a robot that looks like a man, while gynoid would be the technically correct term for robots with a female appearance.

A truly humanoid robot should have somewhat limited autonomous capabilities, not just a human-like appearance. For example, a simple calculator placed in a humanoid shell

will not become a humanoid robot. Humanoid robots are capable, to some extent, of adapting to their environment and are usually built on some form of self-learning system, so that they can develop and improve their abilities to solve problems.

The ability to move is one of the most challenging tasks in creating good humanoid robots, since the human body is actually quite a complex system in terms of its motor capabilities. For example, it is incredibly difficult to create a robot that can jump, because moving a heavy robot requires a significant amount of energy, and adjusting and fine-tuning the motors to ensure that it maintains its balance during a collision is extremely difficult. For example, the humanoid robot Dexter can jump, but it was designed almost exclusively for this purpose, and yet its ability to jump is still very limited compared to humans.

Humanoid robotic engineering recently has become much studied and an important topic for academic community due to its potential usage for domestic and medical aims. In the work [1] there were developed several complicated humanlike robots, for instance, ASIMO robot, created by Honda Motor Company; QRIO robot [2], fabricated by Sony; HUBO robot [3], offered by KAIST.

But even with its limited capabilities, humanoid robots already have a number of applications, and in the future they will be able to solve many other important tasks. Humanoid robots can be used to perform dangerous work that requires human involvement, especially to operate equipment that is already designed to be operated by humans. They can also be used to serve the elderly and to care for and entertain young children. In fact, one area with increasing involvement of humanoid robots is the education of preschool children, who can actively interact with them without experiencing many of the problems that arise between robots and adults. Meanwhile, humanoid robots continue to improve, they can already replace humans in many cases, especially for work in space, underwater or when exploring dangerous areas on the ground.

The InMoov project is an open and publicly available project; it is possible to make exactly such a robot yourself, using the list of standard components and drawings of robot parts posted on a special website for storing 3D projects Thingiverse. Particularly relevant are the studies devoted to dedicate for optimization of kinematical solutions via neural networks, which are relevant for several main fields of computational problems nowadays, since the flexibility and speed of the neural networks. The solutions for forward and inverse kinematics problems are crucially important for the further development of robotics.

---

## 2. Literature review and problem statement

---

Humanoid robotic engineering recently has become much studied and an important topic for academic community due to its potential usage for domestic and medical aims. In the work [1] there were developed several complicated humanlike robots, for instance, ASIMO robot, created by Honda Motor Company; QRIO robot [2], fabricated by Sony; HUBO robot [3], offered by KAIST.

In [4, 5] the authors developed and investigated simple, analytical and illustrative kinematic models for kinematic control of robot gait. Aldebaran Robotics and Robotis, developed and created small humanoid robots NAO [4] and

DARwIn-OP [5]. Robots of this type have found applications in the fields of learning and entertainment, offering an accessible platform for students, researchers and non-professionals.

In [6], a model of robot leg movement was developed, which gave a number of advantages compared to movement on wheels. Also in this work, the developers proposed legs that can overcome barriers and achieve a smooth gait on uneven surfaces, effectively changing the length of the legs in accordance with the geometry of the surface. In [7], the authors developed forward and inverse kinematics, enabling humanoids to perform complex dynamic movements. In paper [8], the authors developed the first conceptual design that defines the positions and orientation of the finite state effector when the active configurations of the robot's joints are specified. And in [8], a second conceptual design was developed that defines joint variables to determine the actual positions and orientation of the finite state effector.

The first incredibly significant step to controlling a robot is to understand the mathematical model of the system. The INMOOV robot has been designed to operate safely in interaction with human, and, as a result, has become one of the most used research platforms in robotics labs.

Planning a robot's motion requires understanding of the relationship between the actuators it is possible to control and the robot's resulting position in the environment. Forward Kinematics of a robotic arm is the process used to find the position of the end-effector of the robot using the knowledge of the angle of each joint. If it is necessary to find the angle of each joint for a particular end-effector position, it is necessary to invert this relationship. This process is known as Inverse Kinematics. Through this study the process of inverse kinematics for INMOOV robot is optimized by developing software application in Python.

In [9], the complexity of the inverse kinematic problem for open kinematics was investigated, in particular, for the reason that a set of solutions led to a nonlinear mapping of parallel and Cartesian spaces. In [10], the authors investigated analytical kinematic solutions for real-time applications, since the computation time of numerical solutions can vary significantly. In [11], the researchers proposed a closed form solution to complex algebraic and geometric problems that uses a reduced number of unknowns to express the position and orientation of the finite state effector.

In [12], the authors investigated the relationship between the constituent parameters of the Denavit Hartenberg (DH) robot leg by applying a geometric approach based on the properties of a triangle. In [13], the authors used the forward and backward decoupling method to solve the kinematics of the humanoid leg using the Hernandez-Santos inverse transformation method. In [14], the authors investigated the sagittal and frontal planes to obtain closed form solutions for the inverse and forward kinematics of the humanoid robot. In [15–17], the authors, using three variables, manipulated both sides of the kinematic matrix equation to express the motion of one leg.

The forward kinematics of a serial manipulator is a very well-established concept in robotics research, the most commonly used technique being Denavit-Hartenberg parameters [1]. Inverse Kinematics is more complicated, as there is no universal mechanism to derive the inverse kinematics equations of a manipulator. In fact, closed-form solutions do not even exist for redundant (7-DOF or more) manipulators. In the article [18], a method of numerical inverse kinematics

of robotic manipulators was developed. The main idea is to develop an algorithm that is complex and efficient. The study used the method of zero position analysis and  $3 \times 3$  rotation matrices (instead of  $4 \times 4$  displacement matrices) directly provides updated geometric information required by the algorithm, and therefore significantly increases the computational efficiency of the numerical method. The algorithm can also be used as a temporary replacement for other numerical methods when the hand is close to a special position. The article [19] provides quantitative comparisons using several humanoid platforms between an improved implementation of the inverse Jacobian algorithm KDL, a set of sequential quadratic programming algorithms (SQP) IK, which use various quadratic error metrics, and a combined algorithm that simultaneously runs the most efficient SQP algorithm and an improved inverse Jacobian implementation. The article [20] describes a new heuristic method called forward and inverse kinematics (FABRIK) and compares it with some of the most popular existing methods in terms of reliability, computational cost and conversion criteria. FABRIK avoids using rotation angles or matrices and instead finds the position of each connection by locating a point on the line. The article [21] proposes an adaptive learning strategy using an artificial neural network ANN to control the movement of a robot manipulator 6 D.O.F. and overcoming the inverse kinematics problem, which is mainly related to features and uncertainties in hand configurations. In this approach, the network was trained to study the desired set of connection angle positions from a given set of end effector positions, experimental results showed an excellent mapping of the robot's workspace to confirm the ability of the developed network to make predictions and a good generalization for any data set, new training using a different data set was performed using the same network, experimental the results showed a good generalization for new data sets. The article [22] presents a new and very effective algorithm for calculating the inverse kinematics of a common sequential kinematic chain 6R. The basic idea is to use classical multidimensional geometry to structure the problem and use geometric information before starting the elimination process. For geometric preprocessing, a training model of Euclidean displacements, sometimes called a kinematic image, was used, which identifies the displacement with a point on a six-dimensional quadric in a seven-dimensional projective space  $P^7$ . Algebraically, a system of seven linear equations and one resultant was solved to obtain a one-dimensional polynomial of degree 16. At this step of the algorithm, two of the six connection angles were obtained, and the remaining four angles are obtained rectilinearly by solving the inverse kinematics of two 2R circuits. In the article [23], the configuration of levers is developed, which are the main problems in the kinematic control of the robot that arise as a result of the use of the robot model. In this study, a solution based on the use of an artificial neural network (ANN) is proposed. The main idea of this approach is to use ANN to study the characteristics of a robotic system instead of specifying an explicit model of a robotic system. The network was designed to have one hidden layer, where the input data were Cartesian positions along the  $X$ ,  $Y$ , and  $Z$  coordinates, orientation according to the RPY representation, and linear velocity of the end effector, while the output data were angular position and velocities for each connection, in the obstacle-free workspace. autonomous smooth geometric trajectories in the connection space of the manipulator. The resulting network was tested on a new set of

data that were recorded in singular configurations to show the generality and effectiveness of the proposed approach, and then the test results were tested experimentally. The article [24] shows the study of the influence of some more important parameters on the final errors of the trajectory of the final effector of the proposed neural network model solving the inverse kinematics problem. The work studied the number of neurons in each layer, the sensitive function for the first and second layers, the coefficient of increase in the trajectory error, the variable step of the time delay and the position of this block, various cases of target data and the case when hidden target data were corrected. The obtained results were verified using the corresponding direct kinematics virtual LabVIEW toolkit. One optimal sigmoid bipolar hyperbolic tangent neural network with time delay and recurrent connections (SBHTNN (TDRL)) of the type is obtained that can be used to solve the inverse kinematics problem with a maximum of 4 % trajectory errors.

The study of previous work shows that the most of the researchers [25, 26] have adopted methods like ANN, ANFIS etc. for simple problem. The features of MLPNN are found quite matching and hence suitable for the present problem having complexity and involving multiple parameters. Therefore, the main aim of this work is focused on minimizing the mean square error of the neural network-based solution of inverse kinematics problem. The result of each network is evaluated by using direct kinematics equations to obtain information about their error. In other words, the angles obtained for each joint are used to compute the Cartesian coordinate for end effector. The training data of neural network have been selected by iterating through angle combinations with sufficiently precise increment.

---

### 3. The aim and objectives of the study

---

The aim of this study is to Identification of patterns forward and inverse kinematics of both hands of the robot by creating imitational model and analysis of the functionality of the manipulator (hand) with 4-dof and workspace for the both hands. This will allow the created robot to achieve the goals, also manipulate with variety of sign languages for the educational systems.

To achieve this aim, the following objectives are accomplished:

- building model of forward kinematics for both arms of the robot by using method of the Denavit-Hartenberg parameters and finding the corresponding workspace for these arms;
- development of the iterative algorithm for the chosen configurations of the arms of the robot for achieving the destination in the space;
- creation of the learning algorithm for the neural network and experiments in order to reduce the root mean square error of summative difference of angles.

---

### 4. Materials and methods of research

---

The object of the study is the INMOOV robot, including kinematics for both robot arms. Research hypothesis: the developed kinematic model based on the machine learning method permits to reach an effective solution for the orientation of both robot arms compared to the iterative

method. A direct kinematic model for the left and right arms of the robot is researched. Let's use Python-based software libraries for calculations, using the matplotlib library for the robot arms workspace. Let's apply an iterative method algorithm to find points in space for development. Using this algorithm, let's develop a simulation model for both robot hand. This paper proposes a structured artificial neural network (ANN) model to find a solution to the inverse kinematics of the INMOOV 4-dof robot. The ANN model used is a multilayer perceptron neural network (MLPNN), which applies gradient descent type learning rules. An attempt was made to find the best ANN configuration to solve this problem. The improvement of the solution is based on minimizing the root-mean-square error by changing each angle by the joints of the robot arm.

**5. Results of the direct kinematics study and InMoov robot simulation model**

**5. 1. Forward pose kinematics**

InMoov robot is equipped with SDK MyRobotLab experimental package, provided by its developers. Apart from that, it has several interfaces of simulating model's telecontrol, as well, real InMoov model. In the work herein there were offered the methods and algorithms on the tasks of direct and inverse kinematics for robot's arms. There is discussed the methodology, tools and resources which have been used for robot's arms orientation.

The task of direct kinematics is to solve the problem of orientation and position of finite element with account of joint angles. It is solved with the help of robot geometry and coordinates system, which are denoted in Denavit-Hartenberg (DH) parameters [1]. DH parameters represent the set of four variables, which define spatial relationship between two permissible coordinate frames. Those variables are  $d_i$  (moving along old  $z$ ),  $\Theta_i$  (rotation around old  $z$ ),  $r_i$  (moving along new  $x$ ) and  $\alpha_i$  (rotation around new  $x$ ). The Table 1 shows the DH parameters for the INMOOV robot model. Those parameters are united, using DH structure to get transformation matrix, which connects two coordinates system by matrices multiplying. General homogeneous transformation from one chain to the next with account of DH parameters is given in matrix form in terms of:

$$T_i^{i-1} = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & \alpha_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \alpha_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1)$$

where  $T_i^{i-1}$  – transformation from coordinates system  $i-1$  into coordinates system  $i$ .

Product of multiplication of all transformational matrices should be defined.

$$T_n^0 = \prod_{i=0}^{n-1} T_{i+1}^i, \quad (2)$$

After which the vector of position should also be extracted from this product.  $p(\theta)$  represents the vector of position, components of which are the coordinates

$$T_n^0 = \begin{pmatrix} \dots & p(\theta) \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

$$p(\theta) = \begin{pmatrix} x(\theta) \\ y(\theta) \\ z(\theta) \end{pmatrix}. \quad (4)$$

Six transformation matrices  $T_i^{i-1}$  for the arm of InMoov robot can be found, based on coordinates systems, shown in Fig. 1.  $T_i^{i-1}$  matrices are enumerated below.

$$[T_1^0] = \begin{bmatrix} 1 & 0 & 0 & r_0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$$[T_2^1] = \begin{bmatrix} c_1 & 0 & -s_1 & c_1 * r_1 \\ s_1 & 0 & c_1 & s_1 * r_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$[T_3^2] = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$[T_4^3] = \begin{bmatrix} c_3 & 0 & -s_3 & 0 \\ s_3 & 0 & c_3 & 0 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$[T_5^4] = \begin{bmatrix} c_4 & 0 & s_4 & c_4 * r_4 \\ s_4 & 0 & -c_4 & s_4 * r_4 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

$$[T_6^5] = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

**Table 1**  
DH parameters of a left arm with six degrees of freedom

$i$	$\alpha_{i-1}$ (rad)	$r_{i-1}$ (m)	$d_i$ (m)	$\Theta_i$ (rad)
1	$\pi/2$	0.2	0	0
2	$-\pi/2$	0.05	0	$\Theta_2 - \pi/2$
3	$\pi/2$	0	-0.18	$\Theta_3 + \pi/2$
4	$-\pi/2$	0	0.286	$\Theta_4 - \pi/2$
5	$\pi/2$	0.0127	0.0135	$\Theta_5$
6	0	0	0.28	$\Theta_6$

Equation of direct shoulder positioning kinematics gives positioning 4-dof (3 positions and 3 orientations) of finite element depending on six joint angles of arm manipulator Fig. 2. Computation of DH parameters is the first step towards obtaining the equations system to compute finite point coordinates.

dinates Table 2. Direct kinematics is computed using transformation matrices. For the shoulder under consideration the matrix of transformation from the base frame to the upper one, using several transition transformation matrices, is set as following:

$$[T_7^0] = [T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^4 * T_6^5], \tag{11}$$

$$p_x = r_0 + c_1 * r_1 - d_2 * s_1 - d_5 * (s_4 * (s_1 * s_3 - c_1 * c_2 * c_3) - c_1 * c_2 * s_2) - d_4 * (c_3 * s_1 + c_1 * c_2 * s_3) - c_4 * r_4 (s_1 * s_3 - c_1 * c_2 * c_3) + c_1 * d_3 * s_2 - c_1 * r_4 * s_2 * s_4, \tag{12}$$

$$p_y = c_2 * r_4 * s_4 - d_5 * (c_2 * c_4 - c_3 * s_2 * s_4) - c_2 * d_3 - d_4 * s_2 * s_3 + c_3 * c_4 * r_4 * s_2 * c_2 * r_4 * s_4 - d_5 * (c_2 * c_4 - c_3 * s_2 * s_4) - c_2 * d_3 - d_4 * s_2 * s_3 + c_3 * c_4 * s_2, \tag{13}$$

$$p_z = c_1 * d_2 + r_1 * s_1 + d_5 * (s_4 * (c_1 * s_3 + c_2 * c_3 * s_1) + c_4 * s_1 * s_2) + d_4 * (c_1 * c_3 - c_2 * s_1 * s_3) + c_4 * r_4 * (c_1 * s_3 + c_2 * c_3 * s_1) + d_3 * s_1 * s_2 - r_4 * s_1 * s_2 * s_4. \tag{14}$$

Presented above equations might be used directly for obtaining finite positions of effectors by entering input data for joint angles.

Let's denote  $c_i = \cos(\Theta_i)$  and  $s_i = \sin(\Theta_i)$  in equations.

Table 2

DH parameters of the right arm with six degrees of freedom

$i$	$\alpha_{i-1}$ (rad)	$r_{i-1}$ (m)	$d_i$ (m)	$\Theta_i$ (rad)
1	$\pi/2$	-0.2	0	0
2	$-\pi/2$	0.05	0	$-\Theta_2 - \pi/2$
3	$\pi/2$	0	-0.18	$-\Theta_3 + \pi/2$
4	$-\pi/2$	0	0.286	$\Theta_4$
5	$\pi/2$	0.0127	0.0135	$-\Theta_5$
6	0	0	0.28	$\Theta_6$

Parameter DH is a mechanical mathematical model of the mathematical model and coordinate system that uses four parameters to express two pairs of position angle relations between the connectors' connections [2].

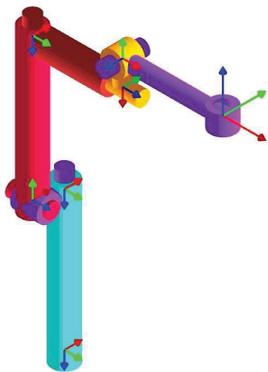


Fig. 2. Kinematic scheme of the right shoulder with six degrees of freedom

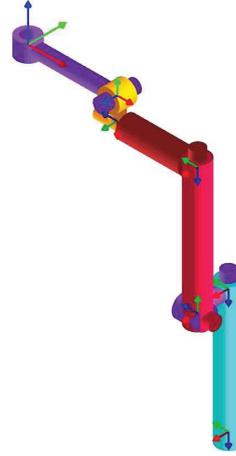


Fig. 1. Kinematic scheme of a left arm with six degrees of freedom with coordinate systems

Research of working space. Fig. 3 shows accessible working 3-D space for left/right manipulators of InMoov robot with length units in centimeters. Visualization was obtained using Matplotlib library in Python programming language in all joint angles' limits (except for wrist joint rotation angle,  $\Theta_6$  which by no means makes influence at positioning XYZ of the finite point). There were obtained the given space mappings, which are shown below in Fig. 3–7.

Upon completing creation of arms movement area, coordinates and points orientation at desired path in Cartesian space might be applied as input data for neural network, as well, there will be generated general parameters, which place the finite effector to the required path.

Fig. 6 shows the resulting joint spatial trajectories corresponding to a circular trajectory defined in Cartesian space.

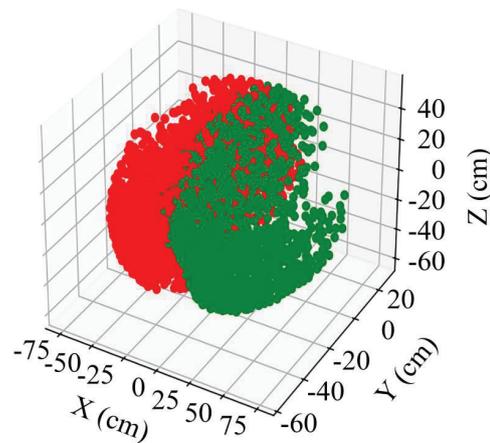


Fig. 3. Chart of working space for left/right arm of InMoov

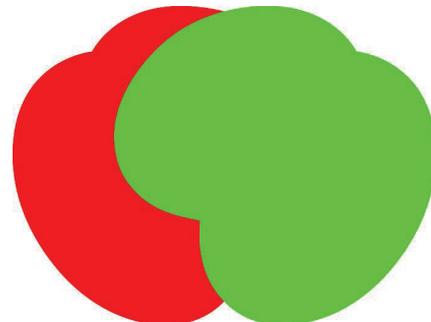


Fig. 4. Mapping of working space of left/right arm at XZ plane

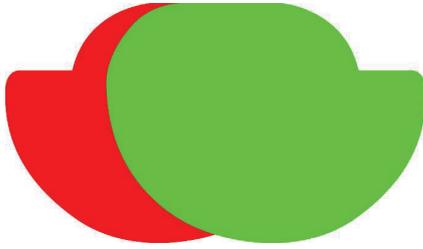


Fig. 5. Mapping of working space for left/right arm on XZ plane

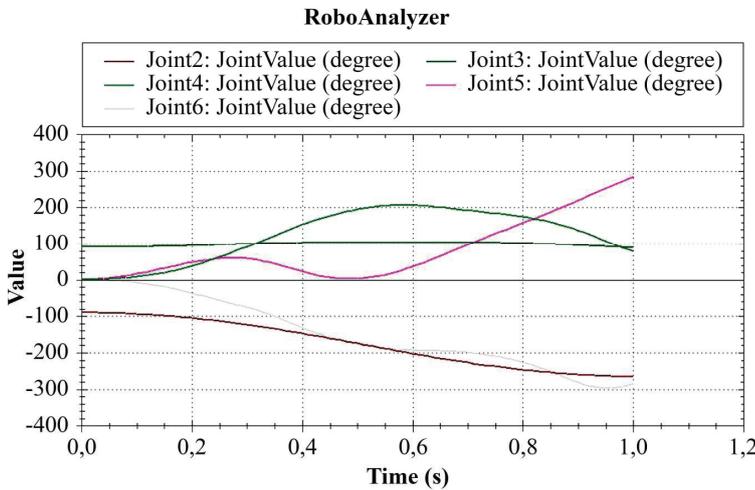


Fig. 6. Combined paths

This section shows all possible working areas of both robot arms. The projection of both hands on different coordinate systems is shown. The figure shows the interval of the maximum minimum degree of degrees of freedom of both robot arms.

**5. 2. Iterative method of inverse pose kinematics solution**

Iterative method through Jacobian matrix is the most basic method for iterative solution of inverse kinematics task. Computing pseudoinverse non-quadratic Jacobian matrices allows find the speed of changing input angles of arm’s parts to reach the point in space at every time point [3].

Initially, there is defined (current) position of the finite effector ( $x \leftarrow FPK(\Theta_{seed})$ ). Afterwards, angles current state vector is assigned values of angles initial positions ( $\Theta \leftarrow \Theta_{seed}$ ). Then, there is started cyclic fulfillment of computations and operations of angle values change, which might be executed, until the distance between the aim point and current position of finite effector is bigger, than satisfying magnitude (*repeat... until*  $\|x_{des} - x\| > \epsilon$ ).

Number of those operations represent:

1. Computation of v ector difference between location of a finite effector and aim point  $\Delta x \leftarrow x_{des} - x(\cdot)$ .
2. Computation of the vector, transmitting movement direction of a finite effector, which is production of the unit vector, vector position difference and scalar value, regulat-ing speed of angles change ( $x' \leftarrow \Delta x / \|\Delta x\| \times step$ ).
3. Assigning to angles vector the new magnitudes, which are vector production of pseudo inverse matrix for Jaco-bian matrix and direction, computed earlier by the vec-tor ( $\Theta \leftarrow \Theta + J^+(\Theta)x'$ ).
4. Updating the values of finite effector current position in the space ( $x \leftarrow FPK(\Theta)$ ).

In the given case Jacobian matrix is computed, applying numerical method, i. e., computed by means of calculating

vector differences at small changes of each angle for some small quantity, which are divided into that small magnitude. Algorithm 1 shows pseudocode for that computation.

Algorithm 1: Iterative Method.

- 1) PROCEDURE Iterative ( $x_{des}, \theta_{seed}, step$ );
  - 2)  $x \leftarrow FPK(\theta_{seed})$ ;
  - 3)  $\theta \leftarrow (\theta_{seed})$ ;
- repeat

```

 $\Delta x \leftarrow x_{des} - x,$ 
 $x' \leftarrow \frac{\Delta x}{\|\Delta x\|} \circ \times step,$ 
 $\theta \leftarrow \theta + J^+(\theta)x',$ 
 $x \leftarrow FPK(\theta),$ 
until  $\|x_{des} - x\| < \epsilon.$ 
    
```

Jacobian matrix’s formal definition, the matrix’s vector components’expressions and corresponding Moore-Penrose pseudoinverse forms are shown below.  $J^+(\theta)$  – the pseudoinverse of Jacobi-an matrix

$$J(\theta) = \begin{pmatrix} \frac{\partial v}{\partial \theta_0} & \frac{\partial v}{\partial \theta_1} & \dots & \frac{\partial v}{\partial \theta_n} \end{pmatrix}, \tag{15}$$

$$\frac{\partial v}{\partial \theta_i} = \begin{pmatrix} \frac{\partial x}{\partial \theta_i} \\ \frac{\partial y}{\partial \theta_i} \\ \frac{\partial z}{\partial \theta_i} \end{pmatrix}, \tag{16}$$

$$J^+(\theta) = \left( J(\theta)^T J(\theta) \right)^{-1} J(\theta)^T. \tag{17}$$

Approximate evaluation of the pseudoinverse of Jacobian matrix can be done just by finding approximate values of the partial derivatives, which can be done by finding the ratio of the change of the position vector and the small increment of the considered joint angle

$$\frac{\partial v}{\partial \theta_i} \approx \frac{v(\alpha) - v(\theta)}{\Delta \theta_i}, \tag{18}$$

$$\alpha = \theta + \begin{pmatrix} \dots \\ \Delta \theta_i \\ \dots \end{pmatrix} = \begin{pmatrix} \dots \\ \theta_i + \Delta \theta_i \\ \dots \end{pmatrix}. \tag{19}$$

Method of cyclic descend along coordinates iteratively tries to make the finite effector, first, converge at sphere with a radius, equal to distance between manipulator basis and finite effector, and then do the same to convergent in the aim position. Fig. 7 demonstrates the initial and final states of the end-effector with applying the Iterative technique.

The proposed method is widely used for simulation of movement of object containing joints and perfectly serves for the task if it doesn’t require significant amount of time. However, this method does have some caveats, such as the inability to converge to a certain orientation and inherent computational limitations, while each iteration tries to make the final effector fall on the line connecting the actuated connection and the target position.

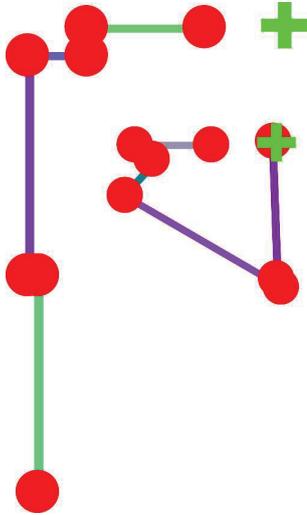


Fig. 7. Results of iterations of the inverse kinematics solver in action

### 5. 3. Usage of multilayer neural network for inverse pose kinematics solving

Solution of robot’s inverse kinematics makes direct influence at robot control accuracy. Inverse kinematics solutions conventional techniques, such as numerical, algebraic and geometrical, have insufficient speed and accuracy, and solution process is complicated. Thanks to ability of neural network to mapping, usage of neural networks for solving the tasks of robot’s inverse kinematics attracted wide attention.

It is well known, that neural networks own the better possibilities, than other methods to solve different complicated tasks. Inverse kinematics is transformation of system’s coordinates ( $X, Y$  and  $Z$ ) into the system of coupling coordinates ( $\Theta_1, \Theta_2, \Theta_3, \Theta_4$ ). That transformation might be fulfilled for input/output work, which uses unknown transfer function. MLP (Multilayer neural net) neuron of neural network is a simple working element and has local memory. Neuron accepts multidimensional input signal, and then transfers it to other neurons according to their weights. It gives scalar result at neuron’s output. MLP transfer function, acting at local memory, uses learning rule to create interrelation between input and output. To introduce activation the time function is indispensable.

The authors offered the solution, using multilayer perceptron with an algorithm of inverse distribution for learning. Afterwards, the network is learnt, using the data for a number of finite effectors positions, expressed in Cartesian coordinates and corresponding joint angles. Data consists of various configurations, accessible for robot’s arm.

The network uses learning regime, in which input data is introduced into the network together with the desired result, and weights get customized in the way, in order the network gives the desired result. The weights after learning contain valuable information, while prior to learning they are random and have no importance.

Clean input of buried neurons (for  $k$  inputs) is defined as

$$n_h = \sum_{j=1}^k w_j \times o_{jn} . \tag{20}$$

Output signal  $O_{mj}$  of buried neuron, as function of its network input is described with equation (13). Sigmoid function is:

$$output = O_{mj} = \frac{1}{1 + e^{-n_k}} . \tag{21}$$

When output data of buried layer neurons was computed, clean input signal for every input layer is calculated in a similar way, as in equation (14).

$$\delta = f'(n)(d - o_m) , \tag{22}$$

$$\delta = o_m(1 - o_m) , \tag{23}$$

where  $d$  – aim or desired value, and  $o_m$  – actual value, obtained from output neuron after direct computation. Error computation was implemented based on the principle ‘neuron after neuron’ along total set (epoch) of patterns. That error value  $\delta$  was used to fulfill corresponding corrections of coupling weight between output layer and buried layer.

$$\delta_h = f'(n_h) \sum_{l=0}^{k_l} w_{lh} \delta_l = o_h(1 - o_h) \sum_{l=0}^{k_l} w_{lh} \delta_l , \tag{24}$$

where  $\delta_h$  – error value of buried layer,  $\delta_l$  – error value of output layer,  $o_h$  – output signal of sigmoid function, and  $w_{lh}$  – coupling weights between output and buried layers. Within the weights of couplings between output and buried layers, weights changes are computed according to equation (16).

$$w(old) = w(new) + \eta \delta_o + \alpha [w(old)] . \tag{25}$$

Learning stage aim consists in minimizing the root-mean-square error in all learning models. Network convergence rate depends on learning curve and kinetic momentum coefficient  $\alpha$ .

In the work herein, double-layer neural network with three inputs  $P_x, P_y$  and  $P_z$ , and four outputs ( $\Theta_1, \Theta_2, \Theta_3, \Theta_4$ ), was learnt, using algorithm of inverse distribution, described earlier, along the path of finite effector in XYZ space.

The proposed work was executed on Python Keras platform. Herein, learning data sets were generated, using equations (12)–(17). First, there was generated the set of 40000 data in compliance with the formula for input parameters of  $P_x, P_y$  and  $P_z$  coordinates in millimeters. Those data sets served as the basis for learning, assessment and testing of MLP model. Out of data sets, 20 % were used as learning data, and 80 % – for testing at MLP, as it is shown in Fig. 8.

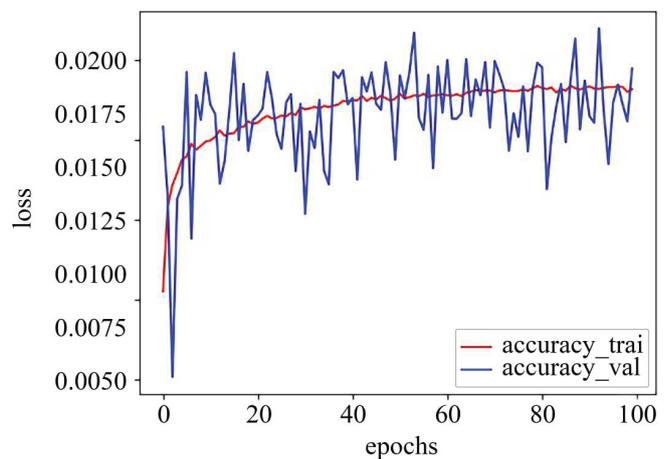


Fig. 8. Results of searching points in space ( $t=100\%$ )

Inverse distribution algorithm was used for learning the network and obtaining the desired weights matrix. In the work herein there was applied the learning method, based on learning periods in epochs. MLPNN model formulation is generalized and might be used for solving direct and inverse kinematic task of any configuration manipulator. However, in the present work there was considered definite configuration only to illustrate method applicability and solution quality, comparing with other alternative techniques.

Mean-root-square curves, shown in Fig. 8, 9 demonstrate the procedure of vector construction for a new path, which gives an idea of proposed model success. As it is shown, in the result of the experiment, the used solution method gives possibility to select the result, which has the least error in the system. Therefore, the solution can be obtained with less measure of inaccuracy, as it is shown in Fig. 9, for the best efficiency of obtained data verification with required data.

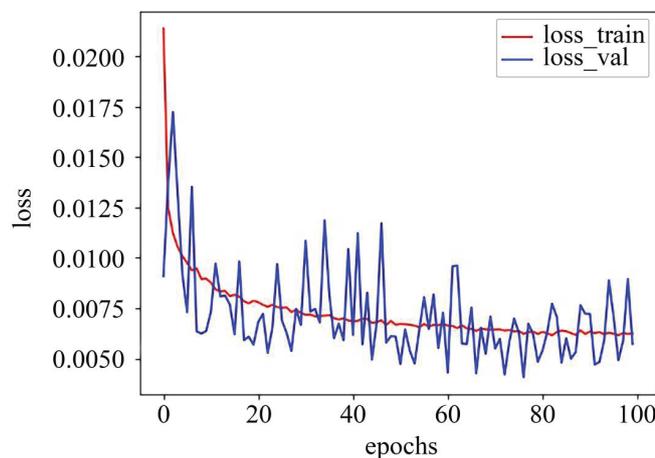


Fig. 9. Errors of finding points in space ( $\neq 100\%$ )

Tests for generalization were carried out with random aim positions, showing, that the studied MLP generalized well along the total space. Regression coefficient analysis, in compliance with Fig. 9, shows conformity of 95.6% for all coupling variables, which is acceptable for getting the inverse kinematics of InMoov robot.

Also, Fig. 10 shows the red dots show errors in finding the coordinates of the manipulator, the green dots are the exact location of the coordinates of the manipulator.

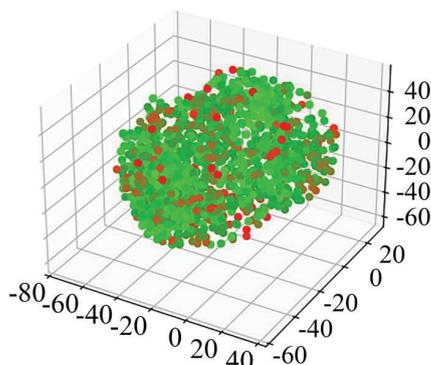


Fig. 10. Error points in space in various views

Main attention in the article was paid to applying analytical approach and neural network to solve inverse kinematics InMoov 4-dof robot. Mathematical models are based

on assumption about advanced structure of the model, which might be non-optimal. Consequently, many mathematical models are not able to model complex behavior for solving the problem of inverse kinematics. Unlike it, ANN is based on pairs of input/output data to define the structure and parameters of the model. Moreover, they always might be updated to obtain better outcomes by submitting new, learning examples, while entering new data. In the given task the error value (root-mean-square error) practically equals to zero, which is much acceptable in distinction to accuracy indicators and values of any typical manipulator's recurrence errors. From the given research it is seen, that MLP gives minimal root-mean-square error for solution and unites variables as performance factor. The model of predicting joint angles, based on artificial neural network might be a useful tool for engineers – technologists for precise validation of manipulator movement.

### 6. Discussion of forward kinematics study and InMoov robot simulation model

Firstly, the shown transformational matrices (2)–(7) are received by replacing some components of the general transformational matrix (1) which are represented as trigonometrical expressions with constant values regarding to the DH characteristics (Table 1, 2). After which, by sequential multiplication (8) of these matrices gave us the final equations (9)–(11) describing the coordinates of the end-effector in the space. Secondly, we got the working space (Fig. 3–5) by marking all points in the space, which can be reached by the end-effector, so we could visually estimate the movement of the manipulator. Thirdly, the standard iterative technique is applied to the manipulator and we have gained positive results (Algorithm 1, Fig. 9). Despite the achieved goals, this technique's disadvantage is its time complexity, which requires computation of the Jacobs matrix and further movement's unit vector. Although it was negligible in the experiment, optimization of the time complexity was considered to avoid time delay to proceed other computations in the computer. Therefore, the better solution for that problem is multilayer neural network, which can maintain properties of the complex trigonometric functions in the computations. The further training (12)–(17) of the neural network has been done in order to gain all the correct properties of the network for adequate functioning. After which, sufficient accuracy (Fig. 8, 9) are gained. Also distribution of the loss per points (Fig. 10) is visualized in the space. Speaking of obstacles in these studies, there could be problems with overtraining the neural network for particular part of the space as an input.

The article herein presents detailed kinematic analysis of two arms of 4-DOF InMoov humanoid robot. There were presented Denavit-Hartenberg parameters for every sequencing circuit, definite parameters of length and joint angles limits. There were developed common 4-dof solutions for FPK kinematics with analytical results. For 4-dof inverse pose kinematics general solution there was proposed numerical approach. There were used such modules as, Matplotlib, Keras, TensorFlow and SciPy, which were implemented in Python programming language, for all developments in kinematics area. The authors elaborated own version of kinematics and dynamics library for InMoov, the library contains implementation of direct kinematics and

iteration algorithms for inverse kinematics with six degrees of freedom.

The accuracy of the neural network cannot be higher because to the fact that the method of evaluation is based on the summative difference between two vectors of the expected output and prediction of the neural network. This is caused by the properties of the Keras module and it is not possible to implement Keras for evaluation of the distance between input position and the destination because the Keras tensors are not available for parsing. Therefore, at some rare cases the error could reach about 1.5 sm. It is possible to reduce the error by enlarging the learning database for the neural network, however the time for learning also increases exponentially.

The simulation model does not consider dynamical properties of the system. Therefore, there could be some errors caused by the inertia of the joints. The neural network itself is implemented in Python programming language, which is easy for implementation, but has drawbacks related to the speed of programming interpretation.

One of the main advantages of the neural networks is its flexibility, hence it can be used for adapting the real conditions as additional mathematical features of the system. In the further research, it is planned to develop algorithm for avoidance of collision between two arms.

---

## 7. Conclusion

---

1. A detailed kinematic analysis of the two arms of IN-MOOV's 4-dof humanoid robot is presented. Denavit-Hartenberg parameters for each sequential chain, specific length parameters, and limits of the connection angle were given. General 4-dof solutions for FPK kinematics with analytical results were developed. The workspaces for the corresponding arms were calculated for the further MLPNN algorithm.

2. The iterative algorithm were developed for the IN-MOOV robot using Jacobian matrix's pseudo-inverse. The simulative model was built for testing the iterative algorithm in Python programming language. The visualization was conducted via PyGame library.

3. The MLPNN algorithm was created via Keras library in Python programming language with less time complexity, which is constant  $O(1)$ , while the iterative method has linear time complexity  $O(n)$ . For the comparison samples of the experiments were obtained, where the iterative algorithm shows about 180 ms in average, while the MLPNN shows approximately 24 ms. The result is caused by the fact that the iterative algorithm conducts matrix multiplications throughout the distance from initial position to destination, while the MLPNN only needs constant amount of calculations. The regression coefficient according to the analysis, which shows a 95.6 % fit for all joint angles, is acceptable for obtaining the inverse kinematics of the INMOOV robot.

---

## Acknowledgments

---

This work is supported by grant from the Ministry of Education and Science of the Republic of Kazakhstan within the framework of the Work No. AP08053034 «Development of new methods for modeling and recognition of Kazakh sign language», Institute Information and Computational Technologies CS MES RK.

---

## Conflict of interest

---

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

---

## References

- Denavit, J., Hartenberg, R. S. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *Journal of Applied Mechanics*, 22 (2), 215–221. doi: <https://doi.org/10.1115/1.4011045>
- Yang, C., Ma, H., Fu, M. (2016). Robot Kinematics and Dynamics Modeling. *Advanced Technologies in Modern Robotic Applications*, 27–48. doi: [https://doi.org/10.1007/978-981-10-0830-6\\_2](https://doi.org/10.1007/978-981-10-0830-6_2)
- Smith, A., Yang, C., Li, C., Ma, H., Zhao, L. (2016). Development of a dynamics model for the Baxter robot. 2016 IEEE International Conference on Mechatronics and Automation. doi: <https://doi.org/10.1109/icma.2016.7558740>
- Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P. et. al. (2009). Mechatronic design of NAO humanoid. 2009 IEEE International Conference on Robotics and Automation. doi: <https://doi.org/10.1109/robot.2009.5152516>
- Williams, R. L. (2012). DARwIn-OP Humanoid Robot Kinematics. Volume 4: 36th Mechanisms and Robotics Conference, Parts A and B. doi: <https://doi.org/10.1115/detc2012-70265>
- Todd, D. J. (1985). *Walking Machines: An Introduction to Legged Robots*. Springer, 190. doi: <https://doi.org/10.1007/978-1-4684-6858-8>
- Kofinas, N., Orfanoudakis, E., Lagoudakis, M. G. (2013). Complete analytical inverse kinematics for NAO. 2013 13th International Conference on Autonomous Robot Systems. doi: <https://doi.org/10.1109/robotica.2013.6623524>
- Spong, W., Hutchinson, S., Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons.
- Aydin, Y., Kucuk, S. (2006). Quaternion Based Inverse Kinematics for Industrial Robot Manipulators with Euler Wrist. 2006 IEEE International Conference on Mechatronics. doi: <https://doi.org/10.1109/icmech.2006.252591>
- Ho, T., Kang, C.-G., Lee, S. (2012). Efficient closed-form solution of inverse kinematics for a specific six-DOF arm. *International Journal of Control, Automation and Systems*, 10 (3), 567–573. doi: <https://doi.org/10.1007/s12555-012-0313-9>
- Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. (2009). *Robotics. Modelling, Planning and Control*. Springer, 632. doi: <https://doi.org/10.1007/978-1-84628-642-1>
- Pieper, D. (1968). *The kinematics of manipulators under computer control*. Stanford University. Available at: <https://apps.dtic.mil/sti/pdfs/AD0680036.pdf>

13. Graf, C., Hartl, A., Rofer, T., Laue, T. (2009). A Robust Closed-Loop Gait for the Standard Platform League Humanoid. Proceedings of the 4th Workshop on Humanoid Soccer Robots (Humanoids '09). Paris. Available at: <http://www.informatik.uni-bremen.de/kogrob/papers/Humanoids-Graf-et-al-09.pdf>
14. Hernández-Santos, C., Rodríguez-Leal, E., Soto, R., Gordillo, J. L. (2012). Kinematics and Dynamics of a New 16 DOF Humanoid Biped Robot with Active Toe Joint. *International Journal of Advanced Robotic Systems*, 9 (5), 190. doi: <https://doi.org/10.5772/52452>
15. Kofinas, N., Orfanoudakis, E., Lagoudakis, M. G. (2014). Complete Analytical Forward and Inverse Kinematics for the NAO Humanoid Robot. *Journal of Intelligent & Robotic Systems*, 77 (2), 251–264. doi: <https://doi.org/10.1007/s10846-013-0015-4>
16. Kalimoldayev, M., Akhmetzhanov, M., Kunelbayev, M., Sundetov, T. (2019). Information systems of integrated machine learning modules on the example of a verbal robot. *NEWS of National Academy of Sciences of the Republic of Kazakhstan*, 6 (438), 215–222. doi: <https://doi.org/10.32014/2019.2518-170x.173>
17. Kenshimov, C., Sundetov, T., Kunelbayev, M., Amirgaliyeva, Z., Yedilkhan, D., Auelbekov, O. (2021). Development of a Verbal Robot Hand Gesture Recognition System. *Wseas Transactions on Systems and Control*, 16, 573–583. doi: <https://doi.org/10.37394/23203.2021.16.53>
18. Kazerounian, K. (1987). On the Numerical Inverse Kinematics of Robotic Manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 109 (1), 8–13. doi: <https://doi.org/10.1115/1.3258791>
19. Beeson, P., Ames, B. (2015). TRAC-IK: An open-source library for improved solving of generic inverse kinematics. 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids). doi: <https://doi.org/10.1109/humanoids.2015.7363472>
20. Aristidou, A., Lasenby, J. (2011). FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 73 (5), 243–260. doi: <https://doi.org/10.1016/j.gmod.2011.05.003>
21. Hasan, A. T., Hamouda, A. M. S., Ismail, N., Al-Assadi, H. M. A. A. (2006). An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *Advances in Engineering Software*, 37 (7), 432–438. doi: <https://doi.org/10.1016/j.advengsoft.2005.09.010>
22. Husty, M. L., Pfulner, M., Schröcker, H.-P. (2007). A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mechanism and Machine Theory*, 42 (1), 66–81. doi: <https://doi.org/10.1016/j.mechmachtheory.2006.02.001>
23. Hasan, A. T., Ismail, N., Hamouda, A. M. S., Aris, I., Marhaban, M. H., Al-Assadi, H. M. A. A. (2010). Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations. *Advances in Engineering Software*, 41 (2), 359–367. doi: <https://doi.org/10.1016/j.advengsoft.2009.06.006>
24. Olaru, A., Olaru, S., Paune, D., Aurel, O. (2012). Assisted Research and Optimization of the Proper Neural Network Solving the Inverse Kinematics Problem. *Advanced Materials Research*, 463-464, 827–832. doi: <https://doi.org/10.4028/www.scientific.net/amr.463-464.827>
25. Mohammed Jasim, W. (2011). Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network. *International Journal on Soft Computing*, 2 (4), 59–66. doi: <https://doi.org/10.5121/ijsc.2011.2406>
26. Mayorga, R. V., Sanongboon, P. (2005). Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An Artificial Neural Network approach. *Robotics and Autonomous Systems*, 53 (3-4), 164–176. doi: <https://doi.org/10.1016/j.robot.2005.09.011>