International Journal of Computer Applications (0975 – 8887)
Volume 178 – No. 39, August 2019

Inverse Kinematics Solution of a Robot Arm based on Adaptive Neuro Fuzzy Interface System

Gjorgji Vladimirov
Faculty of computer science, University of Goce Delchev – Stip
blvd. Krste Misirkov 10/A, 2000 Stip, R. North Macedonia

Saso Koceski
Faculty of computer science, University of Goce Delchev – Stip
blvd. Krste Misirkov 10/A, 2000 Stip, R. North Macedonia

ABSTRACT
Nowadays, robot arms are used as standalone or intrinsic part of many robot systems in various fields and applications. The design and structure of robot arms varies depending on multiple constraints such as the tasks they have to perform, working environment in which they have to operate, the dimensions of the objects they have to peak, etc. Every robot arm, regardless of its design and structure, is aimed to perform some movement that has to be carefully analyzed and planned. During this process, usually two types of motion are analyzed. The first one aims at finding the position of the end effector when the angles between the robot arm links are known. This problem is usually denoted as direct kinematics. The second one, known as inverse kinematics, aims at solving the opposite problem i.e. to determine the angles between links when the position of the end effector is known. This paper presents an inverse kinematics solution of two degrees of freedom planar robot arm based on Adaptive Neuro Fuzzy Interface System (ANFIS). The proposed model is experimentally evaluated and the obtained results are discussed.

General Terms
Kinematics, Inverse kinematics, ANFIS, fuzzy logic.

Keywords
Robot arm, ANFIS, Robot arm with two joint, Robot arm with two links, inverse kinematics, inverse kinematic on 2 joint robot.

1. INTRODUCTION
Since the very beginning of robotics robot arms are developed and used as stand-alone robots or parts of more complex robot systems. First robot arms were used in industrial applications. They were used for performing pick and place tasks and their design was inspired from the flexibility and performance of the human arm. Nowadays, the usage of robot arms has been extended to multiple applications in various sectors. This is a result not only on the advancements in material design and their low prices but, also due to the development of novel control paradigms as well as development of Artificial Intelligence (AI). The growing popularity of the Artificial Intelligence (AI) and its application in various fields [1], starting from tourism [2], through medicine [3-5], biology [6], education [7], robotics [8-11], and also in economy [12], is mainly due to the apparatus i.e. the models and techniques used to mimic the human reasoning, learn and improve during time.

When it comes to research and develop something in the field of robotics, forward and inverse kinematics are the most common problems that may occur. The forward kinematics problem reference to the relationship between the robot arms joints and the position and orientation of the robot arm itself. With other words said, the forward kinematics problem is determination of the position and orientation of the robot arm, when values for the individual joints are given.

However, opposite of forward kinematics problem is the inverse kinematics problem, which may be described as a problem of computing the angles between links in the case of known position and orientation of the end effector.

While solving forward kinematics is almost straightforward, the solution of the inverse kinematics is very difficult and challenging task mainly due to the non-linearity of the problem as well as multiple solutions existence. Therefore, computation wise, solution of the inverse kinematics is usually time consuming and does not always guarantee the convergence [13, 14].

Different methodologies for solving the inverse kinematic problem of robot arms are reported in the literature. They can be divided into three main categories: algebraic [15], geometric [16] and iterative [17]. The main problem of algebraic methods is that the closed form solutions cannot be guaranteed. Geometric methods guarantee a closed for solution for the first three joints of the manipulator while the iterative methods will not work near singularities and may converge to a single solution depending on the starting point.

In contrary, artificial intelligence models and methods, are offering huge flexibility mainly due to their improvement ability and the nonlinear functions approximations capability.

Neural networks and genetic algorithms have been applied in combination to solve the inverse kinematics problem of a six-joint Stanford robotic manipulator in [18]. This paper evidenced the drawback of application of genetic algorithms for solving the inverse kinematics problems. The proposed hybrid approach uses best features of neural networks and evolutionary techniques to obtain more precise solutions.

In [19] authors proposed a solution to the inverse kinematics problem of a robot arm using the continuous genetic algorithm. The algorithm operators (initialization phase, crossover, mutation) were designed in a way to produce smooth joints paths while they maintain an excellent accuracy along the Cartesian path. The authors found that the smoothness of the joints path mainly depends on the initialization phase. Moreover, it was proved that the continuous genetic algorithm outperforms the standard genetic algorithm in terms of both the number of generations for convergence and the average execution time.

Solution to inverse kinematics of robot arm in both cluttered and obstacle free environment using bidirectional particle swarm optimization method is presented in [20]. The main idea behind the bidirectional search is to speed up the convergence in respect to traditional unidirectional search methodology. However, in order to make it applicable the authors proposed a novel manipulator decoupling technique. A multi-objective optimization problem is formulated with minimum total joint movement as another objective function and collision avoidance as constraint. Proposed algorithm was evaluated experimentally on a four degree of freedom robot arm and the reported results are promising.

In [21] a firefly algorithm has been used to solve the inverse kinematics problem of a robot arm. This approach starts with the known forward kinematic model, and later on the firefly is used to generate iteratively a set of joint motions. Afterwards, the forward kinematic solution is used to compute the relative Cartesian positions of a specific end-segment. Actually the firefly algorithm is used to minimize the fitness function defined as a distance between the obtained forward positions and the desired. Proposed algorithm has been evaluated over a 2 links and 3 links articulated planar system. The obtained results confirmed a convergence for one hundred tests performed.

An accurate solution for inverse kinematics using artificial neural network has been proposed in [22]. Authors proposed novel design of the artificial neural network controller in Cartesian coordinates. The proposed design showed improvement in performance of end effector in some aspects.

There are many real-life applications that are based on a planar robot arms composed of two links. This paper presents a solution of a robot arm with two links and two joints that operate in a single plane. Considering it as a simple design it is often used to test novel technologies that will improve the actuation of the arm. One such technology is the Pneumatic artificial muscles (PAMs). Pneumatic artificial muscles (PAMs) belong to the group of nonconventional actuators with remarkable force/weight ratio that can be used for the construction of soft mechanisms safe in contact with humans. This kind of actuators are usually exhibiting huge non-linearity. For this kind of problems Adaptive Neural Fuzzy Inference System (ANFIS) has been introduced as a possible solution for inverse kinematics [23].

Design and implementation of two DOF (Degrees Of Freedom) robot arm capable of teaching children how to write and enhance their writing and drawing ability is presented in [24]. End effector's trajectory and motion were planned using using multi-segment parametric Cartesian equations based on the Adaptive Neuro-Fuzzy Inference System (ANFIS). Two ANFIS controllers were designed for modeling the inverse kinematics to PWM directly. The inputs of each ANFIS are the desired Cartesian coordination represents the desired letter points while the outputs are the two Pulse Width Modulation (PWM) servomotor commands needed to actuate each robot link to the desired position.

This paper presents an inverse kinematics solution of two degrees of freedom planar robot arm based on Adaptive Neuro Fuzzy Interface System (ANFIS). The case of a two links and two joint planar robot arm is analyzed. It is supposed that the robot arm has links with known length, and one of the joints creates a fix contact with the ground as depicted in Figure 1. In this case the inverse kinematics solution, given the position of the end effector, should calculate two angles as output. The first angle is the one between the first ling and the ground, and the second one is the angle between the first and the second link (Figure 1).

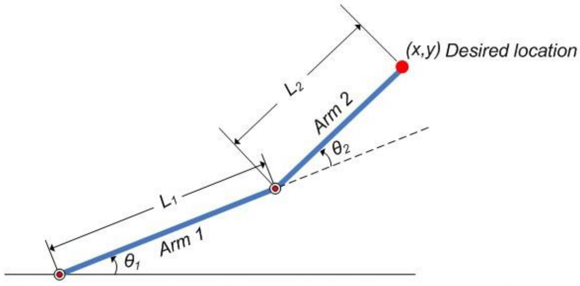

Fig 1: Robot arm design. All parameters of inverse kinematics

2. ANALYTICAL SOLUTION OF THE INVERSE KINEMATICS PROBLEM

Considering a two degrees of freedom planar robot arm with given constraints for the joint angles the analytical solution can be derived using the geometrical dependencies and equations.

Moreover, it will be assumed that the first joint is static i.e. it is fixed to the ground. It will also be assumed that the lengths of the links are known and they will be denoted with l_1 and l_2 (Figure 2). In this case given the coordinates of the end effector, denoted as $M_1(x_1, y_1)$, the angles between the first link and the ground (marked as θ_1) as well as the angle between the line of the first link and the second one (marked as θ_2) should be computed.

For this purpose, let's analyze the triangle ΔOM_1M_2. The lengths of all three sides of this triangle are known or can be calculated and the coordinates of two out of three vertices are known. Unknown remain the coordinates of the third vertex denoted as $M_2(x_2, y_2)$. In order to calculate them, several mathematical equations will be used in the following.

$$\begin{cases} \boldsymbol{l_1}^2 = \boldsymbol{x_2}^2 + \boldsymbol{y_2}^2 \quad \rightarrow \quad \boldsymbol{x_2}^2 = \boldsymbol{l_1}^2 - \boldsymbol{y_2}^2 \\ \boldsymbol{l_2}^2 = (\boldsymbol{x_1}^2 - \boldsymbol{x_2}^2)^2 + (\boldsymbol{y_1}^2 - \boldsymbol{y_2}^2)^2 \end{cases} \qquad \textbf{(1)}$$

This system of equations can be further transformed and simplified, resulting in:

$$l_2{}^2 = x_1{}^2 - 2x_1{}^2x_2{}^2 + x_2{}^2 + y_1{}^2 - 2y_1{}^2y_2{}^2 + y_2{}^2 \qquad (2)$$

$$l_2{}^2 = x_1{}^2 + x_2{}^2 + y_1{}^2 + y_2{}^2 - 2(x_1x2 + y_1y_2) \qquad (3)$$

$$l_2^2 = x_1{}^2 + l_1{}^2 - y_2{}^2 + y_1{}^2 + y_2{}^2 - 2(x_1x2 + y_1y_2) \qquad (4)$$

$$l_2^2 = l_1{}^2 + x_1{}^2 + y_1{}^2 - 2(x_1x2 + y_1y_2) \qquad (5)$$

$$2(x_1x_2 + y_1y_2) = l_1 - l_2 + x_1{}^2 + y_1{}^2 \qquad (6)$$

$$C = x_1x_2 + y_1y_2 \qquad (7)$$

$$D = l_1 - l_2 + x_1{}^2 + y_1{}^2 \qquad (8)$$

$$2C = D \;\rightarrow\; C = \frac{D}{2} \quad \rightarrow \quad C = \frac{l_1 - l_2 + x_1{}^2 + y_1{}^2}{2} \qquad (9)$$

$$C = x_1\sqrt{l_1{}^2 - y_2{}^2} + y_1y_2 \qquad (10)$$

$$C - y_1y_2 = x_1\sqrt{l_1{}^2 - y_2{}^2} \qquad /^2 \qquad (11)$$

$$(C - y_1y_2)^2 = x_1{}^2(l_1{}^2 - y_2{}^2) \qquad (12)$$

$$C^2 - 2Cy_1y_2 + y_1{}^2y_2{}^2 - x_1{}^2l_1{}^2 + x_1{}^2y_2{}^2 = 0 \qquad (13)$$

$$\boldsymbol{ay^2 + by + c = 0} \;\rightarrow\; \boldsymbol{y_{1/2}} = \frac{\boldsymbol{-b \pm \sqrt{b^2 - 4ac}}}{\boldsymbol{2a}} \quad \rightarrow \qquad (14)$$

$$y_{2_1} = \frac{2Cy_1 + \sqrt{4C^2{y_1}^2 - 4({x_1}^2 + {y_1}^2)(-{y_1}^2{y_1}^2 + C^2)}}{2({x_1}^2 + {y_1}^2)} \quad \textbf{(15)}$$

$$y_{2_2} = \frac{2Cy_1 - \sqrt{4C^2{y_1}^2 - 4({x_1}^2 + {y_1}^2)(-{y_1}^2{y_1}^2 + C^2)}}{2({x_1}^2 + {y_1}^2)} \quad \textbf{(16)}$$

With equations (15) and (16) the y coordinate of the point M_2 can be calculated, because the variables in the last equation are known except the y coordinate itself. Once y coordinate is found, the x coordinate can be easy deduced from $\boldsymbol{{x_2}^2 = {l_1}^2 - {y_2}^2}$. At the end as an outcome four different alternatives for the x coordinate will be produced and calculated with (17) and (18) equations. This is because both equations are quadratic equations.

$$x_{2_1} = \sqrt{{l_1}^2 - {y_{2_1}}^2} \qquad x_{2_3} = -\sqrt{{l_1}^2 - {y_{2_1}}^2} \quad (17)$$

$$x_{2_2} = \sqrt{{l_1}^2 - {y_{2_2}}^2} \qquad x_{2_4} = -\sqrt{{l_1}^2 - {y_{2_2}}^2} \quad (18)$$

Therefore, we will have 8 possible candidate points for M_2:

$$\boldsymbol{A(x_{2_1}, y_{2_1}), B(x_{2_2}, y_{2_1}), C(x_{2_3}, y_{2_1}), D(x_{2_4}, y_{2_1}),}$$
$$\boldsymbol{E(x_{2_1}, y_{2_2}), F(x_{2_2}, y_{2_2}), G(x_{2_3}, y_{2_2}), H(x_{2_4}, y_{2_2})}$$

Finding these points does not mean that the robot arm's second joint can be placed in each of them. The robotic arm has physical limits, as it is mention at the very begging and it can move with angle of θ_{1min}- θ_{1max} and θ_{2min}- θ_{2max}. For that reason, additional validation and selection needs to be done. Only those angles falling into intervals between θ_{1min}- θ_{1max}, θ_{2min}- θ_{2max} will be taken into consideration and will be used to determine a single possible solution.

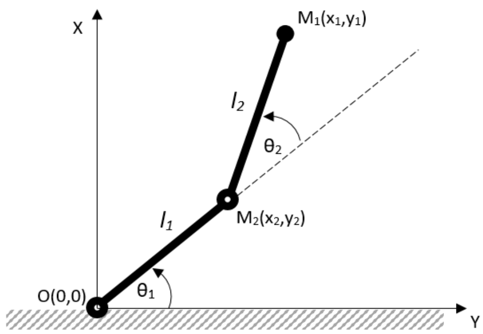

Fig 2: Spatial position of the robot arm links

3. ANFIS CONTROLLER FOR SOLVING THE INVERSE KINEMATICS PROBLEM

Adaptive Neuro Fuzzy Interface System, or shortly ANFIS, generally represents combination of fuzzy logic and neural network. Fuzzy systems have the ability to represent comprehensive linguistic knowledge (given for example by a human expert and preform reasoning by means of rules). However, fuzzy systems do not provide a mechanism to automatically acquire or tune those rules. On the other hand, neural networks are adaptive systems that can be trained and tuned from a set of samples. Therefore, this combination of the positive characteristics of the neural networks with the fuzzy logic could be a viable option for solution of non-linear problems, such as inverse kinematics of a robot manipulators.

In our specific case of a planar two degrees of freedom robot arm the ANFIS controller is composed of two ANFIS networks aimed at computing the joint angles θ_1 and θ_2 taking as input the coordinates of the end effector. The proposed controller architecture is presented in Figure 3.

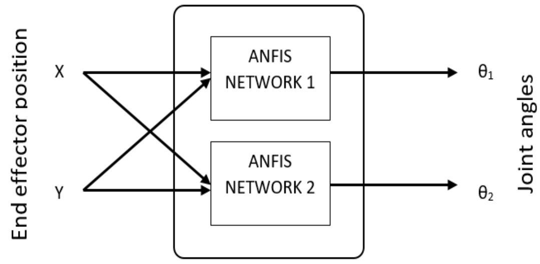

Fig 3: ANFIS Controller architecture for 2DOF planar manipulator inverse kinematics problem calculation

The first phase of development of an ANFIS controller foresees the network training. For this reason, training data are required so that later on ANFIS output can be computed.

In order to generate the training dataset, the forward kinematics equations are used:

$$\boldsymbol{x_2 = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2)} \quad \textbf{(19)}$$

$$\boldsymbol{y_2 = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2)} \quad \textbf{(20)}$$

Data that has been generated here represents all available end points of the planar robot arm in the operating area.

The process starts by generating all values for θ_1 and θ_2 within the operating range using a given step that is usually correlated with the precision of the encoders or stepper motors used to actuate the arm.

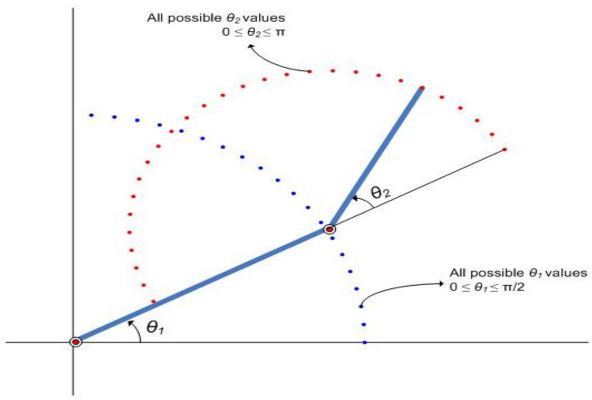

Fig 4: Incremental generation of the training dataset

The following pseudo code, combined with Matlab syntax is explaining the data generation process:

$$l1 = 10; \; -first\ link\ length \quad (21)$$

$$l2 = 7; \; -second\ link\ length \quad (22)$$

$$step = 0.1; \; -arm\ step \quad (23)$$

$$\theta_1 Min = 0; \; -\min angle\ of\ first\ link \quad (24)$$

$$\theta_2 Min = 0; \; -\min angle\ of\ second\ link \quad (25)$$

$$\theta_1 Max = \frac{\pi}{2}; \ -\max angle\ of\ fisrt\ link \qquad (26)$$

$$\theta_2 Max = \pi; \ -\max angle\ of\ second\ link \qquad (27)$$

$$\boldsymbol{\theta_1 = \theta_1 Min : step : \theta_1 Max} \qquad \mathbf{(28)}$$

$$\boldsymbol{\theta_2 = \theta_2 Min : step : \theta_2 Max} \qquad \mathbf{(29)}$$

At this point there are two arrays, one with the all possible values of *Θ1* and the other with all possible values of *Θ2*. Next follows the process of combining these two arrays into a matrix. After the angle matrix is given *X* and *Y* coordinates are calculated based on *Θ1, Θ2* from the matrix using the following formulas:

$-generate\ grid\ of\ angles$

$$[\theta_{11}, \theta_{22}] = meshgrid(\theta_1, \theta_2); \qquad (30)$$

$-\ calculation\ of\ x\ coordinates$

$$\boldsymbol{X = l1 \cdot \cos\theta_{11} + l2 \cdot \cos(\theta_{11} + \theta_{22})}\ ; \qquad \mathbf{(31)}$$

$-calculation\ of\ y\ coordinates$

$$\boldsymbol{Y = l1 \cdot \sin\theta_{11} + l2 \cdot \sin(\theta_{11} + \theta_{22})}\ ; \qquad \mathbf{(32)}$$

Once X and Y coordinate arrays are calculated, joining the two arrays into one, that array is going to represent array of all possible end points (all possible desired location) of the robot arm. Code syntax:

$$struct\ Point\ \{float\ x; \quad float\ y;\}$$

$$\boldsymbol{EndPoint = Array[Point \ldots.] - array\ of\ points}$$

That is illustrated on the Figure 5 and Figure 6. These points are not coordinates of the second joint.

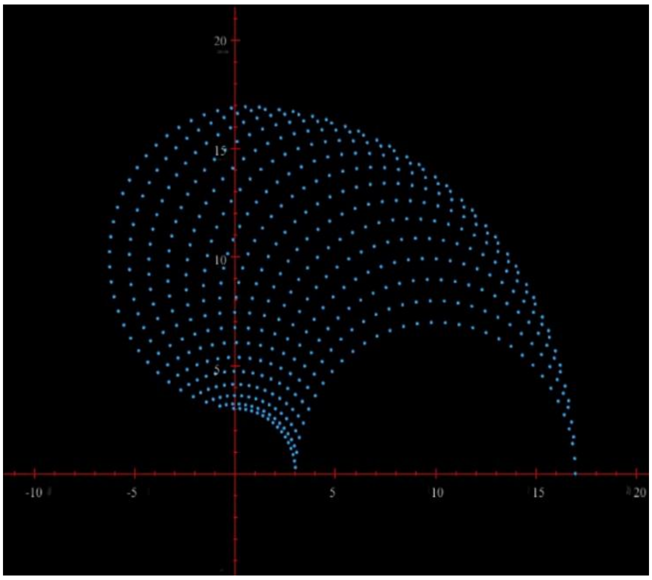

Fig 5: All possible end points (desired points) of the robot arm

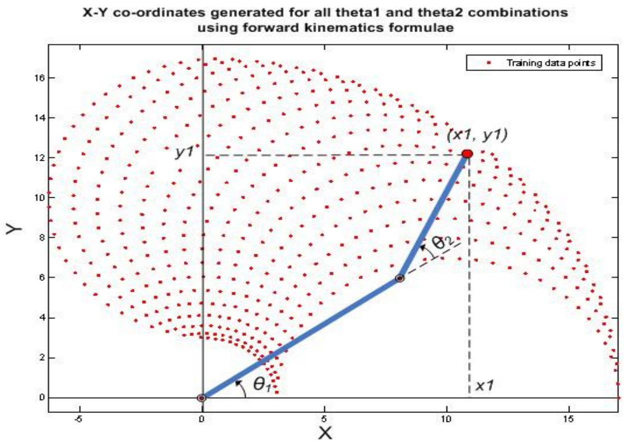

Fig 6: Example of robot arm position to point at some desired point

Once the primary training dataset is generated, duplicates in the input data are determined and the corresponding datasets (both input and target) are removed.

By doing so, both ANFIS networks during the process of training will learn to perform a unique mapping from Cartesian space to joint space.

Both networks have five membership functions for each of the input variables *x* and *y*, that results in a database of rules composed of 25 rules in total for each of the networks. All the membership functions are in the form of a generalized bell function.

$$\boldsymbol{\mu(x, a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}} \qquad \mathbf{(33)}$$

The shape of this function is determined by the coefficients *a*, *b* and *c*. These coefficients represent the premise parameters that are modified i.e. tuned during the process of learning.

Output variables are linear combination of the input parameters and could be described with the following general equation:

$$\boldsymbol{f(x, y, p, q, r) = p \cdot x + q \cdot y + r} \qquad \mathbf{(34)}$$

Both networks in the controller were trained with training datasets ranging from 500 to 1200 input-output data samples and up to 150 training epochs.

Later on the controller was validated against the analytical solution. During the evaluation 50 trials were performed for various positions of the endpoint in the working range. The difference between the joint angles computed using analytical solution and the ones computed using ANFIS controller was in average in the range of 1e-3. This kind of error is acceptable for most applications. However, the precision of the controller can be further improved by increasing the training datasets and epochs as well as with further modification of the architecture.

4. CONCLUSION

Nowadays, robot arms are used in various applications as standalone devices as well as a part of more complex robot systems. Robot arms are usually associated with tasks such as gripping, picking up and placing down objects of various sizes. Although solving the direct kinematics problem is almost a straight forward, solving the inverse kinematics can be very demanding and may rise various issues during the process of control. The complexity of the problem increases with the complexity of the robot arm architecture.

In this paper a solution of the inverse kinematics problem for a two-degree planar robot arm is presented. The proposed solution is based on Adaptive Neuro Fuzzy Interface System (ANFIS). The paper proposes an architecture of the ANFIS controller. Dataset for training the controller is generated and the procedure is explained.

The controller has been validated against an analytical solution. The difference between the joint angles computed using analytical solution and the ones computed using ANFIS controller was in average in the range of 1e-3. This kind of error is acceptable for most applications. However, the precision of the controller can be further improved by increasing the training datasets and epochs as well as with further modification of the architecture.

5. REFERENCES

[1] Loshkovska, Suzana, and Saso Koceski, eds. ICT innovations 2015: Emerging technologies for better living. Vol. 399. Springer, 2015.

[2] Koceski, Saso, and Biljana Petrevska. "Empirical evidence of contribution to e-tourism by application of personalized tourism recommendation system." Annals of the Alexandru Ioan Cuza University-Economics 59, no. 1 (2012): 363-374.

[3] Trajkovik, Vladimir, Elena Vlahu-Gjorgievska, Saso Koceski, and Igor Kulev. "General assisted living system architecture model." In International Conference on Mobile Networks and Management, pp. 329-343. Springer, Cham, 2014.

[4] Stojanov, Done, and Saso Koceski. "Topological MRI prostate segmentation method." In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pp. 219-225. IEEE, 2014

[5] Koceski, Saso, and Natasa Koceska. "Evaluation of an assistive telepresence robot for elderly healthcare." Journal of medical systems 40, no. 5 (2016): 121.

[6] Stojanov, Done, Aleksandra Mileva, and Sašo Koceski. "A new, space-efficient local pairwise alignment methodology." Advanced Studies in Biology 4, no. 2 (2012): 85-93.

[7] Koceski, Saso, and Natasa Koceska. "Challenges of videoconferencing distance education-a student perspective." International Journal of Information, Business and Management 5, no. 2 (2013): 274.

[8] Koceski, Saso, Natasa Koceska, and Ivica Kocev. "Design and evaluation of cell phone pointing interface for robot control." International Journal of Advanced Robotic Systems 9, no. 4 (2012): 135.

[9] Koceski, Saso, Stojanche Panov, Natasa Koceska, Pierluigi Beomonte Zobel, and Francesco Durante. "A novel quad harmony search algorithm for grid-based path finding." International Journal of Advanced Robotic Systems 11, no. 9 (2014): 144.

[10] Koceska, Natasa, Saso Koceski, Francesco Durante, Pierluigi Beomonte Zobel, and Terenziano Raparelli. "Control architecture of a 10 DOF lower limbs exoskeleton for gait rehabilitation." International Journal of Advanced Robotic Systems 10, no. 1 (2013): 68.

[11] Serafimov, Kire, Dimitrija Angelkov, Natasa Koceska, and Saso Koceski. "Using mobile-phone accelerometer for gestural control of soccer robots." In Embedded Computing (MECO), 2012 Mediterranean Conference on, Bar, Montenegro, pp. 140-143. 2012.

[12] Koceska, Natasa, and Saso Koceski. "Financial-Economic Time Series Modeling and Prediction Techniques–Review." Journal of Applied Economics and Business 2, no. 4 (2014): 28-33.

[13] Kucuk, Serdar, and Zafer Bingul. "Robot kinematics: Forward and inverse kinematics." In Industrial Robotics: Theory, Modelling and Control. IntechOpen, 2006.

[14] Pérez-Rodríguez, Rodrigo, Alexis Marcano-Cedeño, Úrsula Costa, Javier Solana, César Cáceres, Eloy Opisso, Josep M. Tormos, Josep Medina, and Enrique J. Gómez. "Inverse kinematics of a 6 DoF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical Neurorehabilitation." Expert Systems with Applications 39, no. 10 (2012): 9612-9622.

[15] J. J. Craig, Introduction to Robotics: Mechanisms and Controls, Addison-Wesley, Reading, MA, 1989.

[16] G. C. S. Lee, Robot Arm Kinematics, Dynamics and Control, Computer, Vol. 15, Issue. 12, pp. 62-79, 1982.

[17] J. U. Korein, N. I. Balder, Techniques for generating the goal-directed motion of articulated structures, IEEE Computer Graphics and Applications, Vol. 2, Issue. 9, pp. 71-81, 1982.

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

[19] Momani, Shaher, Zaer S. Abo-Hammour, and Othman MK Alsmadi. "Solution of inverse kinematics problem using genetic algorithms." Applied Mathematics & Information Sciences 10, no. 1 (2016): 225.

[20] Ram, R. V., P. M. Pathak, and S. J. Junco. "Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling." Mechanism and Machine Theory 131 (2019): 385-405.

[21] Dereli, Serkan, and Raşit Köker. "Calculation of the inverse kinematics solution of the 7-DOF redundant robot manipulator by the firefly algorithm and statistical analysis of the results in terms of speed and accuracy." Inverse Problems in Science and Engineering (2019): 1-13.

[22] Almusawi, Ahmed RJ, L. Canan Dülger, and Sadettin Kapucu. "A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242)." Computational intelligence and neuroscience 2016 (2016).

[23] Hošovský, A., J. Piteľ, K. Židek, M. Tóthová, J. Sárosi, and L. Cveticanin. "Dynamic characterization and simulation of two-link soft robot arm with pneumatic muscles." Mechanism and Machine Theory 103 (2016): 98-116.

[24] Raheem, Firas A., Hind Z. Khaleel, and Mostafa K. Kashan. "Robot Arm Design for Children Writing Ability Enhancement using Cartesian Equations based on ANFIS." In 2018 Third Scientific Conference of Electrical Engineering (SCEE), pp. 150-155. IEEE, 2019.