Link Folding Algorithm (LFA) for Inverse Kinematics of an Industrial Robot

Tiruvidula Eswara Rao[1], Nalluri Mohan Rao[2]

Department of Mechanical Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India[1]

Department of Mechanical Engineering, University College of Engineering, Jawaharlal Nehru Technological (JNT) University Kakinada, Kakinada, India[2]

Corresponding author: Tiruvidula Eswara Rao, Email: terao1@rediffmail.com

The present work aims to depict computationally efficient, model free, singularity free, generalized, non-iterative, single pass, and exact Link Folding (LF) Algorithm to tackle inverse kinematics (IK) problem. This paper presents a 'single scan' numerical algorithm conceived from geometric method to offer suitable IK solution for the robots coordinated and controlled by low resource controllers like Arduino Uno. This approach will reduce computational complexities of the methods like cyclic coordinate descending (CCD) algorithm and target triangle (TT) algorithm. An IK solution is guaranteed when the position to be reached is in robots' workspace and for kinematic structures with unconstrained joints. The method does not require forward kinematics model which is being used in majority of the inverse kinematic solution methods. Test results are provided and analyzed to illustrate the performance improvements of the presented algorithm over the other geometric methods. For algorithm development MATLAB was used, verified in custom built simulator developed by authors, and validated with prototype built and controlled using Arduino.

Keywords: Inverse Kinematics (IK), Geometric method, Link Folding Algorithm (LFA), cyclic coordinates descending (CCD) Algorithm, Target triangle (TT) Algorithm.

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 359–369. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-93-91842-08-6-36

1 Introduction

Robotization and late improvements in innovation made ready for expanded utilization of robots with accentuation for accuracy tasks embraced in the industry 4.0 scenario. Robotic arm's performance is critical in implementation of industry 4.0. In any case, controllers experience difficulties as for their performance that emerge from their kinematic structure; arrive at limits inside their work area, singularity conditions and so forth. For robust manipulator control, it is essential that an end-effector must fallow a singularity free geometric path. A satisfactory degree of practical execution and its control is in this manner imperative to the adequacy and effective execution of such controllers.

A manipulator comprises of a sequential chain of rigid members associated with one another by revolute or prismatic joints. A revolute joint is one in which one of the links rotates about motion axis and in prismatic joint one of the link slides along motion axis on another link. A manipulator is designed to perform a task in 3-D space. To perform any task in three-dimensional space it is required to control a robot arm to move along a specified three-dimensional path. By and large, movement of robot end effector is specified in the Cartesian space, while the mechanical arm movement is constrained by joint movements. In robotics the determination of joint values for a desired Cartesian position on the end effector path is known as inverse kinematics (IK). An answer for the IK problem is crucial in robotics. One of the chief problems in robot kinematics is, finding the solution to IK problem.Answer for the IK issue requires the presence and uniqueness of an answer, in addition to its viability and proficiency. Solution to IK problem is more difficult to solve for sequential manipulators. The troubles are amplified by the prerequisite of continuous movement on the path of the robot tasks. Kinematically Redundant Manipulators (KRM) are those with a greater number of degrees of freedom than that needed to position and situate the end effector in specified workspace. KRMs are utilized to stay away from the inside singularity arrangements and snags and to upgrade dynamic execution. Existing methods are not suitable because of computational effort required to solve IK problem of KRMs which if required to be controlled by low resource controllers like Arduino. Therefore, real-time solution procedures to the IK problem of KRMs are of importance in robotics.In this way, in situ response to IK issue of redundant manipulator is of significance in Robotics.

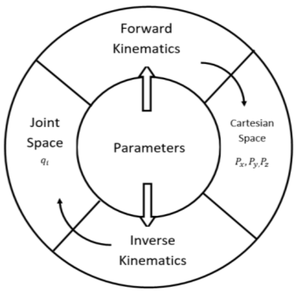


Fig. 1. Relation between FK and IK

A set of joint values determine current position and orientation of end effector called as posture. Determining the end effector posture using current joint values is forward kinematics (FK). Calculating joint values needed to achieve desired end effector posture is inverse kinematics (IK). Relation between FK and IK is illustrated in Fig.1. IK is a challenging task for serial manipulator as it involves singularity problems, non-reachable positions, computational complexities, mathematical formulation difficulties, nonexistence of solutions, multiple solutions, implementation issues and collision detection etc. We needed an inverse kinematic solution which minimizes or eliminates all or some of problems depending upon the application. Splitting complex inverse kinematic problem into simpler problems geometrically seems to be an interesting solution.

Solutions of IK of a robot manipulator are analytic or algebraic, iterative, and geometric. The analytic IK solution requires solution of non-linear equations which is time-consuming. Algebraic and geometric methods are faster and easily identify all possible solutions. However, algebraic methods do not provide closed form solutions. Majority utilize forward kinematics to get the IK solution. Forward kinematics uses Denavit Hardenberg method which is cumbersome and tedious to obtain FK model.

The organization of this paper is as follows. It starts with literature survey in section 2. Comparison of geometric methods in section 3. The proposed link folding algorithm (LFA)with theoretical analysis is in section -4. Section 5 deals with numerical implementation, development of algorithm analysis and results illustrating the effectiveness of the approach. Section 6 is devoted to concluding remarks.

2 Literature Survey

Most of the current IK strategies can be categorized into Analytical or Arithmetic, Computational or iterative and geometric. Somasundar & Yedukondalu [1] presented inverse kinematics using Jacobian inverse method. Jaladi & Rao [2] have used GA (iterative) approach for tackling IK problem of SSRMS robotic arm and obtained results are close to exact. But this method may not be suitable for real time control as computational effort is more and hence time consumption is more. Kumar & Rao [3] have used ANFIS approach for handling IK issue of SSRMS robotic arm. ANFIS approach may be suitable for real time control as computational effort is less, but the results obtained using ANFIS approach are approximate, and it requires significantly more time to train the network and regression at some of the positions may not be satisfactory. Tao, & Yang [4] have presented FABRIK algorithm for avoiding collision. Song & Huc [5] reported Target Triangle algorithm that evaluates IK solution fast and eliminates improper and large-scale angle rotations. Cajar and Mukundan [6] introduced a method for solving IK problem utilizing the law of cosines which incurs low computational cost. Petrenko et al. [7] have provided an inverse kinematic solution for any kinematic chain with rotational joints using Denavit-Hartenberg notation and analytic geometry. Tokarz & Kieltyka [8] have discussed various methods of IK and described Mechanical and Electronic constructional aspects of a robotic arm. Tokarz & Manger [9] presented geometric approach-based IK suitable to be used by low resource controllers. However, most IK strategies experience ill effects of high computational expense and the peculiarity or singularity conditions. F. A. Candelas et al [10] described implementation and control aspects of robotic arms using Arduino.

3 Comparison of Geometric Methods

The following table 1 gives comparison of various geometric methods for IK: cyclic coordinate descent (CCD); Forward and Backward Reaching Inverse Kinematics (FABRIK); Target triangle algorithm (TTA); and Link folding Algorithm (LFA).

Table 1. Comparison of various geometric methods for IK

Aspect\Algorithm	CCD	FABRIC	TTA	LFA
Implementation of low resource control unit	not suitable	not suitable	suitable	suitable
Joint rotations	Improper	All links rotate	All links rotate	links rotate on the position
Iterative method	multi pass	multi pass	single pass	single pass
Computational effort	More	More	less	Very less
Implementation of low resource control unit	not suitable	not suitable	suitable	suitable

4 Development of Link Folding algorithm (LFA)

Uniform, continuous, and singularity free motion is required for development of path planning models. To obtain a solution for the IK problem Link Folding algorithm (LFA) is introduced. LFA uses heuristic approach to find joint values beginning from the end of the kinematic chain. An answer for IK issue is ensured when utilized with unconstrained joints and the necessary position is in range. This technique is like CCD, as each joint endeavors to track down its best direction disregarding the direction of any previous joints. Joint values affected meanwhile in transit that will move the end-effector towards the goal. Current work presents a Link Folding Algorithm (LFA), aimed to offer simplified solution to Inverse kinematic problem avoiding multiple solutions. The created algorithm presents a non-traditional and computationally productive answer for the IK issue utilizing LFA. Utilization of the proposed approach supports creation of a uniform, constant, and singularity free path development. Proposed LFA enjoys many benefits in execution, for example, exact control and less calculating time, as well as beating some numerical issues in path generation. Proposed method is suitable for real time applications as it is computationally fast.

The above characteristics make the proposed algorithm both fast and useful for robots involving more-joints. In the proposed technique the calculation emphasizes through each joint from end-effector to base. At each joint we tackle similar arrangement of conditions, consequently by the last joint an answer will be found. Accordingly, we require just a single pass to arrive at the objective.

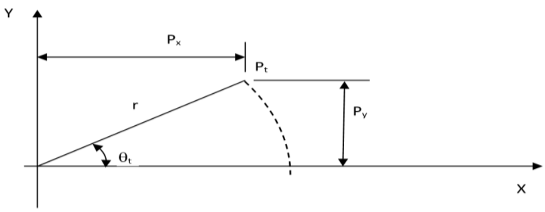


Fig. 2. (a) Target position P_t in XY plane

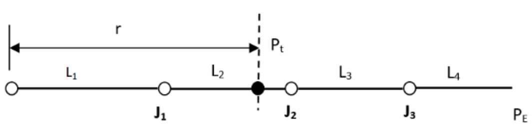


Fig. 2. (b) Links in Fully extended condition considered as home postion

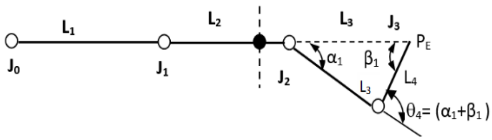


Fig. 2. (c) Links L_3 and L_4 are folded to align P_E with J_3

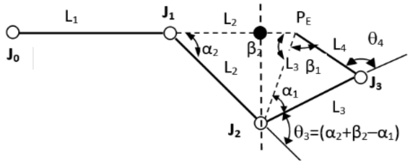


Fig. 2. (d) Links L_2 and edge J_2P_E are folded to align P_E with J_2

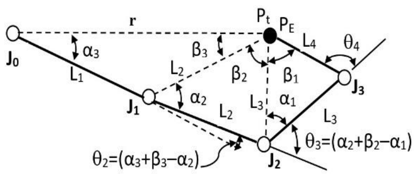


Fig. 2. (e) Links L1 and edge J1PE are folded to align PE with Pt

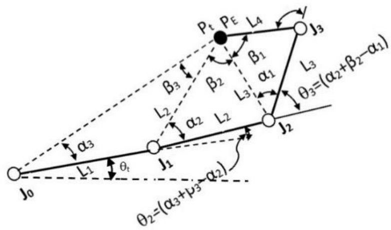


Fig. 2. (f) Solution diagram is rotated through θ_t

Fig. 2. Geometric implementation of Link folding algorithm

The developed algorithm is mostly planned to tackle IK issue of a planar robot. But the algorithm can easily be extended to three dimensional robots. Accepting it at home situation of the robot as displayed in fig. 2(b) the calculation attempts to decrease the distance between target position and the end-effectors position by folding the links each pair in turn beginning from the end until the target position is reached. Once the required target position is reached final angular displacements of each link are obtained by rotating the solution diagram by angle θ_t as depicted in fig.2.(f)

5 Proposed Geometric Approach for the Solution

(i) Assume the home position of the chain is at θ_i=0° for i is 1 to 4 as shown in fig.(a).

(ii) Rotate the target vector on to the home position diagram. Target position will be at distance'r' from the origin as displayed in fig.2(a).
(iii) Fold the last two links i.e., L_3 and L_4 rotating about joint J_3 to move the end effector P_E along horizontal axis coinciding position of J_3 before folding. θ_4 is the angular rotation of the link L_4. as displayed in fig.2(b).
(iv) Fold the link L_2 and edge J_2 P_E of the triangle J_2 J_3P_E rotating about J_2 to move the end effector P_E along the horizontal axis coinciding with position of J_2 before folding. θ_3 is the angular rotation of the link L_3. as displayed in fig.2(d).
(v) Fold the link L_1 and edge J_1P_E of the quadrilateral $J_1J_2J_3P_E$ rotating about J_1 to move the end effector P_E along the horizontal axis coinciding with target point P_t. θ_2 is the angular rotation of the link L_2. as displayed in fig.2(e).
(vi) Rotate the solution diagram through an angle of θ_t determined in fig.2. (a) and as depicted in fig.2. (f)

5.1 Numerical implementation

Referring fig.2. (c),

$$L_4^2 = 2L_3^2 - 2L_3^2 \cos \alpha_1$$

Defining, $\kappa = \frac{L_4}{L_3}$

$$\alpha_1 = \cos^{-1}\left(1 - \frac{1}{2}\kappa^2\right)$$

$$\mathrm{L}_3 \sin \alpha_1 = \mathrm{L}_4 \sin \beta_1$$

$$\beta_1 = \sin^{-1}\left(\frac{\sin \propto_1}{\kappa}\right)$$

Referring fig.2. (d) Defining, $\kappa = \frac{L_3}{L_2}$

$$\alpha_2 = \cos^{-1}\left(1 - \frac{1}{2}\kappa^2\right)$$

$$\beta_2 = \sin^{-1}\left(\frac{\sin \propto_2}{\kappa}\right)$$

Referring fig.2. (e),

$$L_2^2 = L_1^2 + r^2 - 2L_1 r_1 \cos \alpha_3$$

$$\alpha_3 = \cos^{-1}\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1 r_1}\right)$$

$$\beta_3 = \sin^{-1}\left(\frac{\sin \propto_3}{\kappa}\right)$$

Let L_1=100, L_2=100, L_3=70 and L_4=60 in mm. and for reaching target point $P_t(100,10)$

Referring fig.3, $\kappa = \frac{L_4}{L_3} = 0.8571, \propto_1 = 50.75^0, \beta_1 = 64.62^0, \theta_4 = 115.3769^0$

$\kappa = \frac{L_3}{L_2} = 0.7000, \propto_2 = 40.97^0, \beta_2 = 69.51^0, \theta_3 = 59.7334^0$

$\kappa = \frac{L_2}{L_1} = 0.666666, \propto_3 = 38.9420, \beta_3 = 70.52890, \theta_2 = 78.6951^0$

$\theta_1 = -54.1243^0$

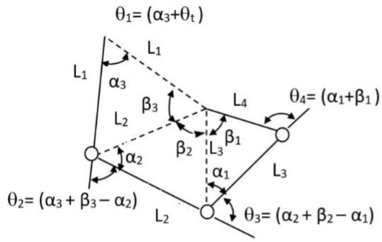


Fig. 3. Schematic of numerical implementation of the algorithm

5.2 Algorithm

Read x_t,y_t target point
Read n, number of links
Read L_i, i is 1 to n, Length of each link
Set SL=0, Sum of lengths

```
        x1=0
        Repeat  i, 2 to n+1
                xi=xi-1+Li-1
                SL=SL+Li-1
                thi=0
        end
```

r= $\sqrt{x_t^2 + y_t^2}$
theta_t=arctan2(y_t,x_t) ,rotation angle for the solution at origin

```
thp=0
if r is less than or equal to SL
for i=1 to n-1
if(r<xn+2-i&& r<SL)
        j=n-i
        k=n+1-i
           if(r>xn+1-i)
             a=r-xn-i
            th1=cos-1((Lj2+a2-Lk2)/(2Lja))
            th2=sin 1(Ljsin (th1)/Lk);
           else
             R=Lk/Lj
             th1=cos-1(1-0.5R2);
             th2=sin-1(sin(th1)/R);
if(r<x2&&i=n-1)
               a=r-xn-i
               alpha=cos-1((Lj2+a2-Lk2)/(2Lja))
              beta=sin-1(Lj*sin(th1)/Lk);
             else
             end
           end
thk=alpha+beta-thp
thp=alpha
thj=-alpha
    else
    end
  end
```

```
 th1=th1+theta_t
 else
   message ('Point is out of range')
end
```

5.3 Results

Development of the algorithm and numerical simulation was carried out in MATLAB and graphical simulation was done in customized simulator(fig.4) developed by the authors using three.js which is webGL wrapper java script library.Three.js is a multi-browser supported JavaScript library and Application Programming Interface used to make and show 3D computer graphics in an internet browser. Three.js utilizes WebGL. The source code is facilitated in an archive on GitHub. Algorithm control was tested using Autodesk's TinkerCAD circuits simulator(fig.5).

Table 2. Various configurations achieved by algorithm for two link manipulator

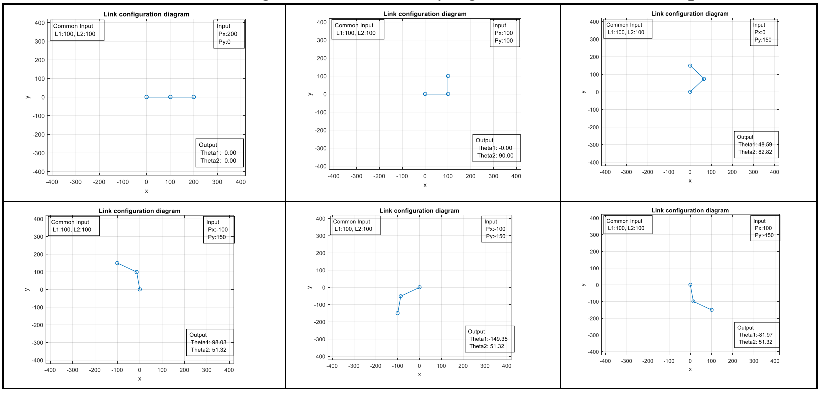


Table 3. Various configurations achieved by algorithm for three link manipulator

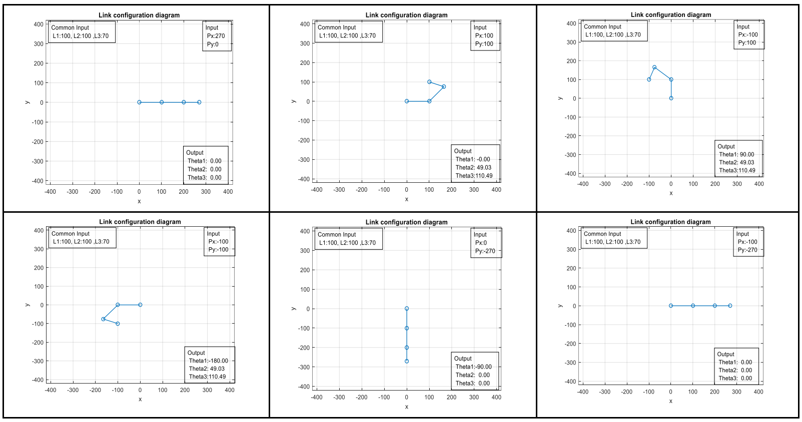


Table 4. Various configurations achieved by algorithm for four link manipulator

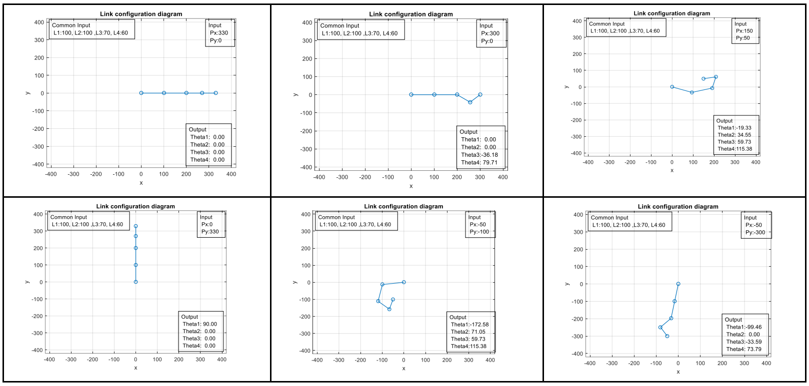
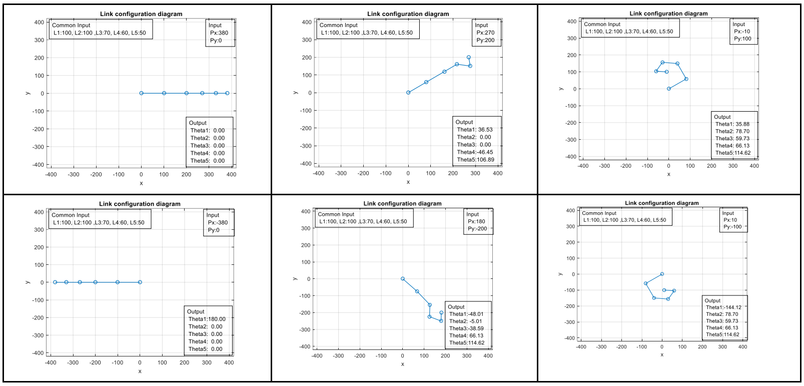


Table 5. Various configurations achieved by algorithm for five link manipulator



From the results displayed in tables 2,3,4,5 we can see the various typical positions reached by the end effector in its workspace. For reaching far positions only the links in the end are rotated which looks unnatural. In arms with more than 3links the far links move around the target position which is practically difficult to implement.

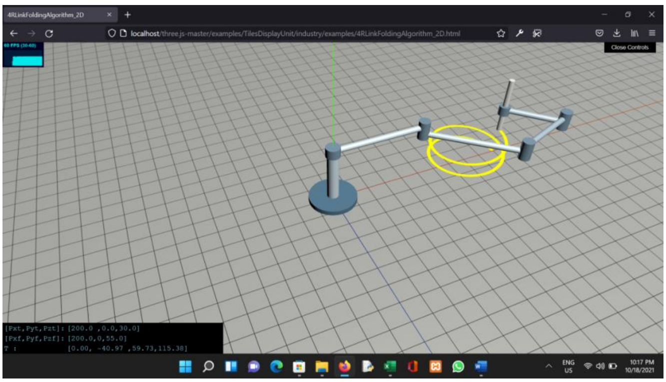


Fig. 4. Simulator developed using webGL

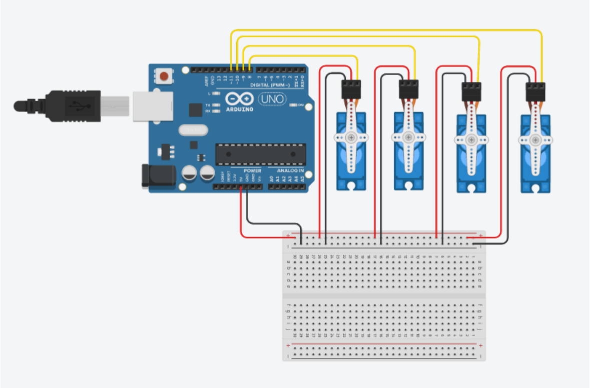


Fig. 5. Arduino circuit connection diagram

6 Concluding Remarks

The main limitations of the CCD, TTA and FABRIC algorithms have been outlined. This paper has discussed the IK solution for an n-link joint chain using link folding algorithm. In this study, a non-iterative, computationally efficient, geometric link folding algorithm has been presented for solving IK of serial manipulator. The proposed method best suits for real-time applications, as it needs few computational steps to process each link and it processes each link at most once to obtain a solution. A comprehensive analysis has been made to show the benefits of the presented algorithm over existing other geometric methods. Test results show that the introduced algorithm is computationally proficient, general, powerful, and versatile than customary or different related strategies and can be tuned for individual execution and conduct references. The algorithmic design is modular and generic and thus might also be applied to various other robot kinematic configurations that can be solved by LFA. In future this work can be extended to derive the solution for IK of a redundant manipulators.

References

[1] Somasundar, A.V.S.S. and Yedukondalu, G. (2018). Robotic path planning and simulation by jacobian inverse for industrial applications. *Procedia Computer Science,* 133:338-347.

[2] Jaladi, S. and Rao, T. E. (2019). An Inverse Kinematics Analysis of Space Station Remote Manipulator System (SSRMS) Using Genetic Algorithms. *International Journal of Mechanical and Production Engineering Research and Development,* 9(1):217–226.

[3] Kumar, R. L. and Rao, T. E. (2018). An inverse kinematics of a 7 DOF space station remote manipulator system (SSRMS) using ANFIS. . *International Journal of Mechanical and Production Engineering Research and Development*, 8(7):1364–1375.

[4] Tao, S. and Yang, Y. (2017). Collision-free motion planning of a virtual arm based on the FABRIK algorithm. *Robotica*, 35(6):1431–1450.

[5] Song, W. and Hu, G. (2011). A fast inverse kinematics algorithm for joint animation. *Procedia Engineering,* 24:350–354.

[6] Cajar, R. M. and Mukundan, R. (2007). Triangualation - A New Algorithm for Inverse Kinematics. In *Proceedings of Image and Vision Computing New Zealand,* 181-186.

[7] Petrenko, V. I. et al. (2018). The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only. *IOP Conference Series: Materials Science and Engineering,* 450(4):1-6.

[8] Tokarz, K. and Kieltyka, S. (2010). Geometric approach to inverse kinematics for arm manipulator. In *International Conference on Intelligent Systems,* 1(II):682-687.

[9] Tokarz, K. and Manger, C. (2011). Geometric and rough set approach to inverse kinematics for arm manipulator. *International Journal of Mathematical Models and Methods in Applied Sciences,*5(1):1-8.
[10] Candelas, F. A. et al. (2015). Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. *IFAC-Papers On Line*, 48(29):105–110.