Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://journal.utem.edu.my/index.php/jet/index

INVERSE KINEMATICS SOLUTION FOR A 6-DEGREE-OF-FREEDOM ROBOTIC MANIPULATOR BASED ON FABRIK

J. Yem[1], T. F. Chay[*1]

[1]Faculty of Engineering and Technology, Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia.

*chaytf@tarc.edu.my

Article history:
Received Date: 2021-10-26
Accepted Date: 2021-12-06

Keywords: FABRIK, Industrial robots, Inverse kinematics

Abstract— This paper aims to present a novel algorithm to solve the inverse kinematics (IK) problem of a 6 Degree-of-Freedom (DOF) industrial robotic manipulator based on FABRIK (Forward And Backward Reaching Inverse Kinematics). The algorithm calculates and locates the joint position for every joint of the robotic manipulator in Cartesian space to bring the end-effector to the desired target position. Based on the obtained joint positions, the rotational angle for each joint is then calculated to complete the IK solutions. The algorithm is tested for its feasibility and accuracy in MATLAB Robotic Toolbox. The results show that with an average of 13 iterations of computing, the joint positions are obtained with an average error of 1.256×10^{-3} mm in Cartesian space. The algorithm presented in this paper yields high accuracy with fewer iterations of computing thus offering a new way to solve the IK problem of industrial robotic manipulators.

I. Introduction

Inverse kinematics is used to calculate the required and optimal movement of the joints of a linked system to reach the required destination. It is used in industrial robotic manipulators to determine how the robotic manipulator should move to bring the end-effector to the desired target position. In 2011, there was a new method named Forward And Backward Reaching Inverse Kinematics (FABRIK) invented by Dr. Andreas Aristidou to solve the inverse kinematics problem in computer animation. This iterative solution requires less computation time and at the same time able to provide a smooth trajectory compared to other numerical methods [1].

While FABRIK is widely used in computer animation, the application of this method to solve inverse kinematics problems of robotic manipulators has started to attract the attention of researchers in the field of robotic research in recent years. In searching the studies related to FABRIK, it was found that FABRIK was applied to develop solutions to solve inverse kinematics problems of rigid link 2-dimensional (2D) planar manipulators with 3 DOF, 4 DOF, 6 DOF, 8 DOF, 10 DOF [2-4]; and 3-dimensional (3D) non-planar manipulator with 2 DOF [5]. Besides the rigid link manipulators as mentioned, FABRIK was also applied to solve inverse kinematics problems of multi-section flexible link continuum robots (elephant trunk-like robots) which are commonly used in the medical field [6].

From the review, it was found that inverse kinematics solutions using FABRIK for the commonly used 6 DOF non-planar industrial robotic manipulator in 3D Cartesian space have not been carried out yet. Therefore, this paper aims to formulate and test the algorithm to solve the inverse kinematics problems of a 6 DOF non-planar robotic manipulator based on FABRIK.

II. Research Methodology

A. Introduction to FABRIK

FABRIK iteratively solves inverse kinematics problems by calculating and updating the joint positions forward and backward regarding previous joint positions until the result is significantly close to the target [1]. It is started by calculating the distance between each joint and checking whether the target is reachable. The target is only reachable if the distance between the root and target is smaller than the sum of the distances of all of the inter-joints. For the case, if the target is unreachable, the algorithm will be terminated if the error tolerance is saturated.

As illustrated in Figure 1, the iteration starts by moving the end-effector inwards to the target position t. The new position of end-effector is noted as p_4', where $p_4' = t$. Next, the new position of p_3' is determined. The length of the link from p_3 to p_4 defined the position of p_3' on the vector across p_4' and p_3. A similar concept is applied to find $p_2'$and p_1', and the first stage of iteration ends when all the positions of joints were updated.

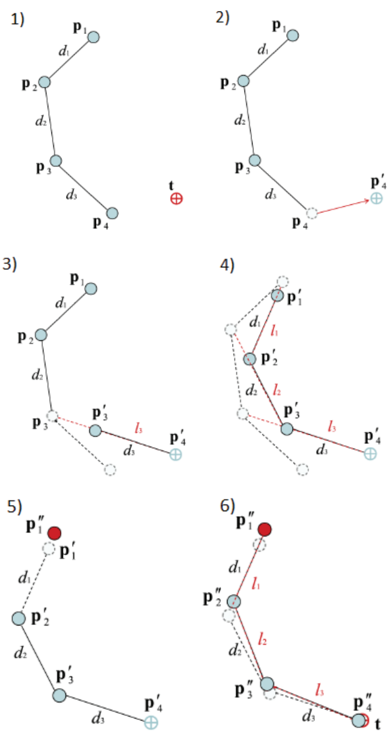

Figure1: FABRIK algorithm

However, the position of p_1' is now different from its initial position. Hence, the second stage of iteration is needed to complete the algorithm. The second stage of iteration begins with the root joint p_1. The new position regarding p_1', known as p_1'' is determined using the same concept used in the first stage. The procedure repeats for all the joints including the end-effector. As the position of the end-effector will not reach the desired target position with one complete iteration, this two-stage iteration procedure is repeated until the end-effector reached a position identical or significantly closer to the target [1].

The advantage of using the FABRIK algorithm is that it requires less computational cost due to its simplicity. By repositioning or reorienting the target in the allowable bound, FABRIK can be applied on most of the robotic joints without much modification [2][7]. However, the FABRIK method does not consider joint limitation and orientation of robotic manipulators [5]. It works better only if all the joints of the links are located on the same plane as those 2D planar manipulators [3]. Therefore, an extended version of FABRIK is necessary to be formulated to solve inverse kinematics for 6 DOF non-planar manipulators in 3-dimensional space.

B. Selected Robot and Its D-H Representation

In this paper, the original FABRIK algorithm is modified to solve the inverse kinematics problems of a non-planar serial robotic manipulator. To conduct this study, a commonly used 6 DOF industrial robot manipulator, Mitsubishi MELFA F-Series RV-35F as shown in Figure 2 is selected and its Denavit-Hartenberg (DH) parameters such as link length, link twist, and offset are summarized in Table 1.

To test the inverse kinematics solutions using the extended FABRIK algorithm, the robot modeling of RV-35F was done in MATLAB Robotic Toolbox with the DH parameters listed in Table 1. Figure 3 shows the link and joint representation of the robot modeling in MATLAB Robotic Toolbox.

Figure 2: Mitsubishi MELFA F-Series RV-35F

Table 1: DH parameter of RV-35F

Link	Link length	Link twist	Link offset	Joint angle
1	0	pi/2	355	0
2	900	0	0	0
3	150	pi/2	0	0
4	0	-pi/2	990	0
5	0	pi/2	0	0
6	0	0	375	0

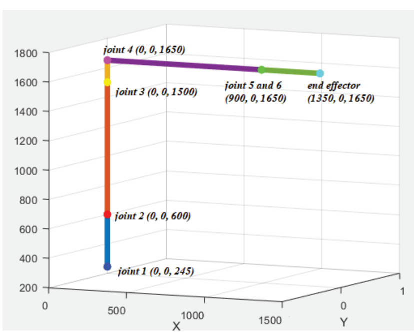

Figure 3: Robot Modelling of RV-35F in MATLAB Robotic Toolbox

C. Extended FABRIK Algorithm

The original FABRIK was created to find the joint positions for every joint of a linked system in Cartesian space. To use FABRIK to solve inverse kinematics of a non-planar industrial robotic manipulator in 3D, the original FABRIK has to be extended to find the joint angles for every joint of the robot.

The extended FABRIK algorithm presented in this paper to solve the inverse kinematics problems of a 6 DOF robotic manipulator is split into 2 major steps. The first step is to find the position of every joint in 3D Cartesian space and the second step is to calculate the required angle for each joint to move the end-effector to the target position based on the joint positions found.

With P_6 representing the position of the end-effector of the robotic manipulator, the algorithm starts with setting the new end-effector position, P_6' to the target position, t , where $P_6' = t(x, y, z)$. Next, to find the new position of Joint

5, P_5', a vector is formed from P_6' to the initial joint position of Joint 5, P_5 which λ will be the ratio between Link 5 with the vector. As a result, the position of P_5' can be determined with Equations (1) and (2):

$$\lambda_5 = \frac{link\ 5}{|P_5 - P_6'|} \quad (1)$$

$$P_5' = P_6' + \lambda_5(P_5 - P_6') \quad (2)$$

Position P_5' is now set as the new target and it has to be checked if it is a reachable target. If P_5' is an unreachable target, the error tolerance will be set as 0.1 to terminate the iteration in MATLAB, otherwise 0.

Before FABRIK is implemented, the position of Joint 3 is relocated via projection so that it is on the same plane as Joint 5. The projection can be carried out with Equations (3) and (4):

$$tan\theta_1 = \frac{P_5'(y)}{P_5'(x)} \quad (3)$$

$$(P_3(x'))^2 + (P_3(y'))^2 = (P_3(x))^2 + (P_3(y))^2 \quad (4)$$

After projecting Joint 3 to the plane of Joint 5, FABRIK is implemented to find its new position.

As shown in Figure 4, the positions of Joint 2 and Joint 1 remain unchanged even if both Joint 2 and Joint 1 rotate, the forward iteration hence ends at Joint 2. Next, backward iteration starts at Joint 2 towards Joint 3 and then Joint 5 as these three joints are now on the same plane.

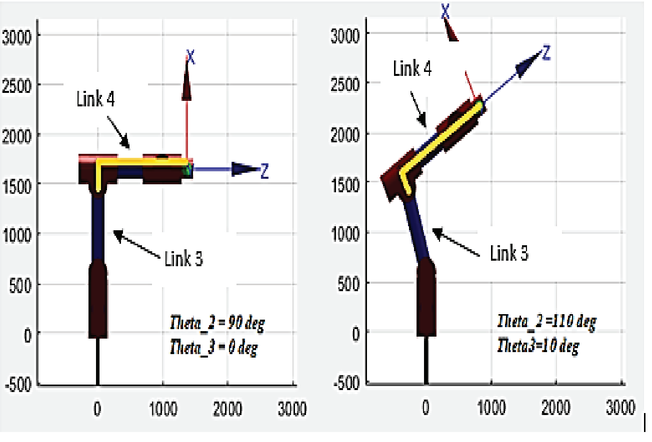

Figure 4: Position of Joint 2 remains unchanged when Joint 1 rotates

The FABRIK algorithm will be terminated when the error between the computed position P_5'' and P_5' reaches the error tolerance after several iterations. However, to prevent the algorithm falls into an endless loop, the algorithm will be terminated as well if the error is saturated. At this stage, position for Joint 5, Joint 3, Joint 2 and Joint 1 would have been determined by FABRIK.

When applying FABRIK algorithm on this robot, position of Joint 4 is not included in the calculation, as Link 3 and Link 4 which connected by Joint 4 are always perpendicular. This is illustrated in Figure 5. This speeds up the computation and unconstraint FABRIK is able to be applied.

However, the finding of Joint 4 position is still necessary when determining the joint angle for each joint. Position of Joint 4 can be calculated by the interception of Link 3 (i.e. from Joint 3 to Joint 4) and Link 4 (i.e. from Joint 4 to Joint 5). Several cases to define the position of Joint 4 are illustrated in Figure 6.

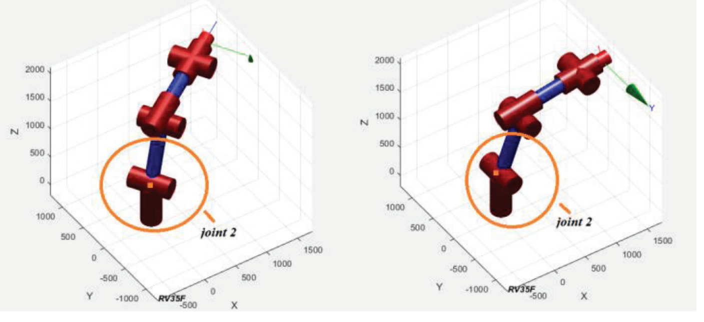

Figure 5: Link 3 and Link 4 are perpendiculars

After finding the joint position for each joint, the next step is to calculate the angle for each joint so that the end-effector can reach the desired target position. The angle of Joint 1, θ_1 can be determined by the x and y coordinate of Joint 5. Angles for Joint 2, 3, and 5 ($\theta_2, \theta_3,$ *and* θ_5) can be found by the vector dot product between the links. Equations (5) to (8) are hence formulated and programmed in

MATLAB for calculating the mentioned angles:

$$tan\theta_1 = \frac{P_5''(y)}{P_5''(x)} \quad (5)$$

$$cos\theta_2 = \frac{\overrightarrow{P_1P_2''}.\overrightarrow{P_2''P_3''}}{|\overrightarrow{P_1P_2''}||\overrightarrow{P_2''P_3''}|} \quad (6)$$

$$cos\theta_3 = \frac{\overrightarrow{P_2''P_3''}.\overrightarrow{P_3''P_4''}}{|\overrightarrow{P_2''P_3''}||\overrightarrow{P_3''P_4''}|} \quad (7)$$

$$cos\theta_5 = \frac{\overrightarrow{P_4''P_5''}.\overrightarrow{P_5''P_6''}}{|\overrightarrow{P_4''P_5''}||\overrightarrow{P_5''P_6''}|} \quad (8)$$

On the other hand, the angle for Joint 4 can be obtained by comparing the translation matrix in the homogeneous transformation matrix of the robot and its end-effector position, as shown in Equations (9) and (10):

$$T_6^0 = \begin{bmatrix} R_6^0 & O_6^0 \\ 0 & 1 \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} P_6''(x) \\ P_6''(y) \\ P_6''(z) \end{bmatrix} = \begin{bmatrix} O_6^0(x) \\ O_6^0(y) \\ O_6^0(z) \end{bmatrix} \quad (10)$$

The angle of Joint 6 is not determined in this paper as the end-effector's orientation is not given. If the orientation of the end-effector is given, the angle of Joint 6 can be found by comparing the rotation matrix with the orientation matrix from Joint 6 to Joint 1.

It is to take note that FABRIK does not consider the limits of joint angles especially for the rigid link robotic manipulator that is studied in this paper. Hence, angle correction is needed if the joint angles for Joint 2 and Joint 3 exceed the joint limit. A similar concept introduced to find the position of Joint 4 will be applied to determine the new joint position for Joint 3 and Joint 4 as shown in Figure 7.

In summary, the flowchart to perform the extended FABRIK algorithm to solve the inverse kinematics problems of the 6 DOF articulated robotic manipulator is shown in Figure 8. The algorithm is programmed into MATLAB to test its feasibility and accuracy.

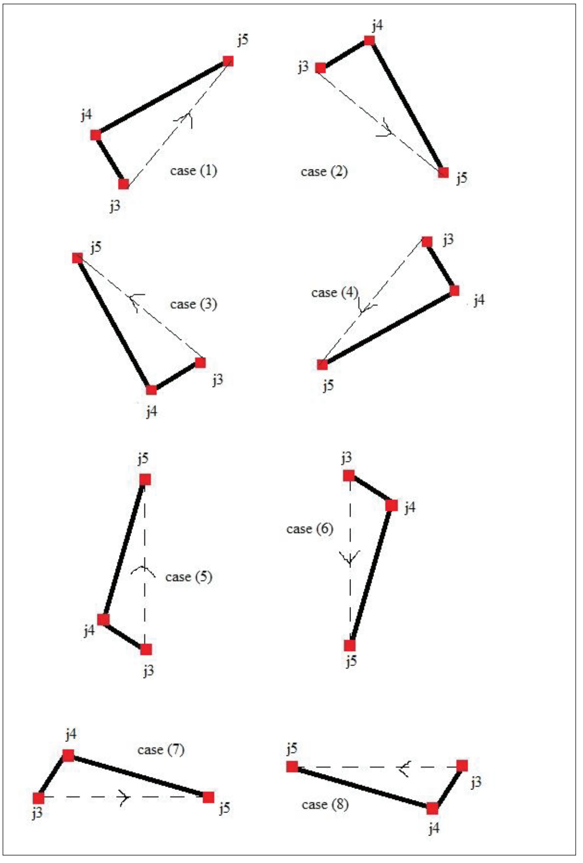

Figure 6: Position of Joint 4 in different possible cases

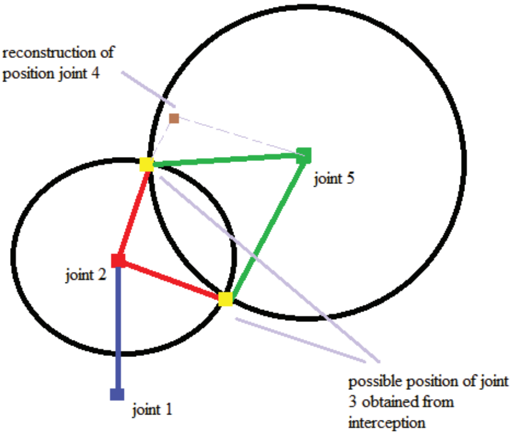

Figure 7: Angle correction to relocate the joint position

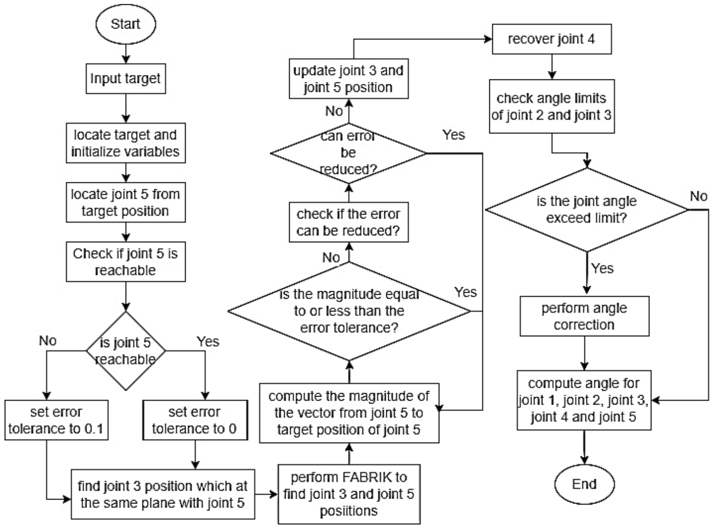

Figure 8: The flowchart of the extended FABRIK algorithm

III. Result and Discussion

Table 2 shows the result of a series of point-to-point positioning performed by the 6 DOF robot model in MATLAB Robotic Toolbox using the extended FABRIK algorithm. The results show that the extended FABRIK can solve the inverse kinematics of the robot by calculating the end-effector position with an average of 13 iterations and an average positioning error of 2.754×10^{-4} mm.

The joint angles of Joint 1 to Joint 5 (θ_1, θ_2, θ_3, θ_4, θ_5) calculated by the inverse kinematics solutions using the extended FABRIK were verified with the end-effector position obtained from forward kinematics, *robot.teach* function in the MATLAB Robotic Toolbox. The end-effector's position is obtained by feeding the rotation angles (calculated by the extended FABRIK) for each joint into *robot.teach* function. The verification data is tabulated in Table 3. The error between the end-effector position obtained from *robot.teach* function and the desired target position (of the end-effector) is then calculated and tabulated in Table 4. The average error found is

1.256×10^{-3} mm which is considered accurate for industrial robotic manipulators.

The results show that the end-effector can reach the desired target positions in a series of point-to-point positioning [1600, 900, 150], [150, 250, 2500], [20, -300, 2500], and [600, -1800, 150] in a back and forth iteration manner and the extended FABRIK algorithm is proven to be able to calculate the rotation angle of each joint. The average positioning error of 1.256×10^{-3} mm is within a very good range. This shows that the extended FABRIK algorithm can solve inverse kinematics for the non-planar 6 DOF manipulator in 3D Cartesian space.

As mentioned, FABRIK does not consider the limits of joint angles especially for rigid link robotic manipulators, the extended FABRIK algorithm presented in this paper has successfully calculated the joint angle for every joint of the robot and can perform an angle correction algorithm if any joint exceeds their limit as demonstrated in Joint 2 and Joint 3.

Table 2: Number of iterations needed for end-effector to reach target position using extended FABRIK algorithm

Target Position	Number of Iteration	End Effector Position	Error between Target Position and End Effector Position, mm
[1600, 900, 150]	8	[1600.0001, 900.00017, 149.99972]	0.0003424
[150, 250, 2500]	11	[149.9999, 249.99997, 2500.0002]	0.0002449
[20, -300, 2500]	17	[20.000148, -299.99972, 2499.9999]	0.0002976
[600, -1800, 150]	16	[600.00005, -1800.0002, 149.99975]	0.000324
[20, -300, 2500]	11	[19.999946, -299.99987, 2500.0002]	0.0002446
[150, 250, 2500]	18	[149.99998, 249.99977, 2499.9999]	0.0002311
[1600, 900, 150]	13	[1600.0001, 900.00006, 149.99987]	0.0001746
[1365, 0, 1650]	9	[1365, -0.000169, 1650.0003]	0.0003443
Average number of iteration = 13			Average position error = 0.0002754 mm

Table 3: Verification of the calculated rotation angle for each joint via comparing the end-effector position using forward kinematics

Computed End Effector Position	θ_1	θ_2	θ_3	θ_4	θ_5	End Effector Position from function
[1600.0001, 900.00017, 149.99972]	25.9270	26.6072	21.4964	-131.9268	23.1877	[1600.000, 899.999, 150.001]
[149.9999, 249.99997, 2500.0002]	42.4674	104.2905	22.9737	-12.8556	85.1738	[150.000, 250.000, 2500.000]
[20.000148, -299.99972, 2499.9999]	0.96425	107.7217	36.6183	53.3847	86.0836	[19.843, -299.862, 2500.11]
[600.00005, -1800.0002, 149.99975]	-70.7849	24.9727	23.5667	153.2462	9.4703	[598.371, -1801.220, 150.701]
[19.999946, -299.99987, 2500.0002]	-78.5496	103.0759	23.5423	6.1277	86.4821	[20.000, -299.999, 2500.000]
[149.99998, 249.99977, 2499.9999]	-36.8693	108.2074	37.1486	-50.6936	88.0772	[150.001, 249.999, 2500.000]
[1600.0001, 900.00006, 149.99987]	28.6719	28.5279	16.9009	-153.1842	7.4629	[1600.000, 899.998, 149.998]
[1365, -0.000169, 1650.0003]	8.3203	64.1995	12.7977	33.4209	106.9946	[1365.000, 0.000, 1650.000]

Table 4: Error between end-effector position and target position

Target Position	End Effector Position from function	Error, mm
[1600, 900, 150]	[1600.000, 899.999, 150.001]	0.00141
[150, 250, 2500]	[150.000, 250.000, 2500.000]	0.00000
[20, -300, 2500]	[19.998, -300.000, 2500.000]	0.00200
[600, -1800, 150]	[599.999, -1800.000, 149.999]	0.0014
[20, -300, 2500]	[20.000, -299.999, 2500.000]	0.00100
[150, 250, 2500]	[150.001, 249.999, 2500.000]	0.00141
[1600, 900, 150]	[1600.000, 899.998, 149.998]	0.00283
[1365, 0, 1650]	[1365.000, 0.000, 1650.000]	0.00000
		Average error = 0.001256 mm

IV. Conclusions

FABRIK has attracted the attention of robotics researchers in recent years but most of the projects implement FABRIK to solve inverse kinematics of 2D planar manipulators or 3D planar manipulators with low DOF. Therefore, the study of inverse kinematics solutions of a non-planar 3D 6 DOF robotic manipulator is carried out in this paper and a novel algorithm based on FABRIK has been formulated and tested to solve the robotics inverse kinematics problems. The results show that this algorithm can solve inverse kinematics problem for a 6 DOF robotic manipulator (Mitsubishi RV-35F) with an average position error of 1.256×10^{-3} mm by 13 iterations of computation. Due to the working principle of the calculation which is started from the end-effector to the robot base, this extended FABRIK algorithm can solve the inverse kinematics problem for other industrial robotic manipulators with 3 to 7 DOF.

V. Acknowledgment

The authors would like to express their gratitude to the editors and reviewers of the Journal of Engineering and Technology for their constructive comments and time in reviewing the manuscript.

VI. References

[1] M Aristidou, A. and Lasenby, J., "FABRIK; A fast, iterative solver for the inverse kinematics problem," Graphical Models, vol. 73, no. 5, 2011, pp. 243-260.

[2] M. Abdallah, Baziyed, M.s., Fareh, R. and Rabie, T., "Tracking control for robotic manipulator based on FABRIK algorithm," Advances in Science and Engineering Technology International Conferences, 2018, pp. 1 – 5.

[3] G. Plouffe, P. Payeur and A. M. Cretu, "Real-time posture control for a robotic manipulator using natural human-computer interaction," Multidisciplinary Digital Publishing Institute Proceedings, 2018, vol. 4, p. 31.

[4] X. Ren, W. Dinghui and H. Haibo, "A heuristic iterative method for solving the inverse solution of kinematics of manipulators," Chinese Control and Decision Conference (CCDC), IEEE, 2019, pp. 3521-3526.

[5] R. A. Tenneti and A. Sarkar, "Implementation of modified FABRIK for robot manipulators," Proceedings of the Advances in Robotics, 2019, pp. 1-6.

[6] D. Y. Kolpashchikov, N. V. Laptev, V. V. Danilov, I. P. Skimevskiy, R. A. Manakov and O. M. Gerget, "FABRIK-based inverse kinematics for multi-section continuum robots," 18th International Conference on Mechatronics - Mechatronika (EE), IEEE, 2018, pp. 1-8.

[7] A. Aristidou, Lasenby, J., Chrysanthou, Y. and Shamir, A., "Inverse Kinematics techniques in computer graphics: A survey," Computer Graphics Forum, vol. 37, 2018, pp. 35-58.