Nano World Journal

https://doi.org/10.17756/nwj.2023-s4-037

Research Article

Open Access

Prediction of Inverse Kinematics Solution of a Robotic Manipulator Using Python

Hrithik Varma Balerao[1*], Naresh Kumar Sarella[1], Venkateshwar Reddy Pathapalli[1] and Srinivasa Rao Pittam[2]

[1]Department of Mechanical Engineering, Vardhaman College of Engineering, Hyderabad, Telangana, India
[2]Department of Mechanical Engineering, Kommuri Pratap Reddy Institute of Technology, Hyderabad, Telangana, India

[*]Correspondence to:
Hrithik Varma Balerao
Department of Mechanical Engineering,
Vardhaman College of Engineering,
Hyderabad, Telangana, India.
E-mail: hrithikvarma08@gmail.com

Received: September 15, 2023
Accepted: November 23, 2023
Published: November 28, 2023

Citation: Balerao HV, Sarella NK, Pathapalli VR, Pittam SR. 2023. Prediction of Inverse Kinematics Solution of a Robotic Manipulator Using Python. *NanoWorld J* 9(S4): S214-S219.

Published by United Scientific Group

Abstract

In the current profession of global engineering have a feeling that robotics is somewhat a field with a main goal of a target. Important one of the making of machines that performs like human being. The kinematics of robots deals mostly with the motion of linkages which consist of velocity, displacement, and acceleration of robotic manipulator analytically. Deriving the kinematic model for a robotic mechanism of open chain for analyzing the performance of industrial manipulators. In this work, it will be examined closely a class of Popular 2-DOF and 3-DOF open chain mechanisms where its inverse kinematics admit a closed-form analytical Solution. By using python simple coding can be developed easily. A two-link planar connection manipulator and a three-link planar RR manipulator are considered to get an Inverse kinematic solution in a python environment. For this, a solution is considered to acquire the links of joint variables to reach the workspace position with corresponding values of input as the length of a link, position of the end effector.

Keywords

Robotic manipulator, Joint variables, Workspace, Inverse kinematics, Python

Introduction

Robotics a simpler interpretation that is defined as, an automated gadget that undertakes functions usually an attribute to People nor a machine in form as human beings. Karel Capek, a Czech the author, established the word "robot" in a year 1920. Robota is a word (CZECH word) that derived from the term Robot. By this robot we can perform several tasks and Variety of tasks [1]. Robotics is a topic that covers many disciplines which is useful from mechanical, electrical and electronic engineering, computer science, biology, and many other fields.

In engineering, robotic field, Kircanski and Vukobratovic [2, 3], Hussain and Noble [4], Tsai and Chiou [5], and Karlik and Aydin [6] are the researchers who used MACSYMA, REDUCE, SMP, and SEGM methods to reach forward and inverse solutions, these approaches need the use of standard mathematical principles. Khatib [7] worked in that field and provided a technique to avoid single locations obtained during path planning control inside the work volume. Lloyd and Hayward [8] designed a unique mechanism for a dual RCCL generator with a motion that is controlled of 'C' language in programming in an operating system environment. Mandava and Vundavilli [9], on the basis of inverse kinematics and the Zero Momentum Point approach, he offered a closed type of solution for an 18 degree of freedom (DOF) biped robot. Sreenivasulu [10] proposed the solutions for kinematics of inverse with a 2-DOF robotic manipulator utilising a procedure of geometric. Nugroho et al. [11] developed a design and executed a ki-

nematic strategy to construct a NAO robotic humanoid. Sadiq et al. [12], PSO technique was utilized to find the ideal way in a space of cartesian map for a 2-DOF robotic arm, also to get exact solutions. Chaitanya and Reddy [13] created representation for optimising the planning a path at 2-DOF robotic manipulator utilizing a genetic algorithm method. Kanayama et al. [14] suggested a stable control of tracking rule for non-holonomic vehicles to find appropriate targets applicable to mobile robots that are autonomous. Mohamed and Duffy [15] using the screw theory idea, in research he carried out a study on kinematics of instantaneous to the effector end position the area of completely parallel robot-like gadgets. Jones and Walker [16] proposed a modular method of approach for solving inverse kinematic problems of multi-section continuum robotics. Radavelli et al. [17], a comparison of the kinematics manipulators for robots between the DH convention and the dual quaternion method was given.

Chen et al. [18] refined a technique on theory of screw to predict the kinematics of inverse solutions of a robotic arm. Chirikjian [19] in addition, the kinematics of a system of robots were investigated by taking metamorphic stages into account. Robot kinematic and inverted solution methods for various robotic linkages are detailed in several textbooks by authors such as Craig [20], O'Malley [21], and Murray et al. [22]. Raheem et al. [23] shown on study, the way to improve the task in environment accompanied by the tip of effector of the robot in space using meta heuristic approaches. Hudgens [24] focused on research upon kinematic investigations of micro manipulators, particularly parallel linkages. Kumar et al. [25], in this discussed about the linkages of kinematics. Sun et al. [26], DH notations were used to suggest solutions on kinematics of inverse analytical. Tsai and Morgan [27] and Zhuang et al. [28], created a solution for universal manipulators of 6 and 5-DOF using continuation techniques. Veitschegger and Wu [29] investigated measuring methods in robot kinematics analysis to acquire positions of specific end effectors. Webster and Jones [30] created the kinematics structure for robots of a continuum of constant curvature. Yang et al. [31] they examine the use of automated strategies in the area of applied robotics in their book.

This requires accurate nanomanipulation supported by multiples adjacent operations. Nanomanipulation has received extensive attention due to large multidisciplinary applications ranging from biology to functional materials, electronic chipsets, and MEMS/NEMS. The industrial applications can be extended if the ultra-precision motion is maintained over relatively large workspace for automotive reflector optical reflection [32].

Other manipulators were based on micro- and nano-robotics focusing mainly on end-effectors e.g., micro-triggers and MEMS manipulation. All manipulators have outlined sub-micrometre resolution with workspace in the mesoscale. To improve the workspace range, manipulators were equipped with several positioning stages to reach 3-DOF within one cm^3 [33, 34]. According to literature, prior researchers concentrated on various areas of approaches involved in designing as well as developing solutions of kinematics, which is inverse, for various configurations of robotic arms. Programmable investigations on solutions for kinematics of inverse were found as limited in scope when compared to geometric techniques. It was further found that no one used software of python to obtain the joint angles as solution for robotic links with many DOF challenges particularly in the robot area.

Experimentation

Forward and inverse kinematics

Forward kinematics

The kinematics in forward has an interest in the connection between the effector's end location (x, y, or z) and orientation (φ) and each robot manipulator's specific joints (x, y, and z). The forward kinematics, to put it more precisely, is the process of figuring out the effector's end location also the orientation gives the robot's joint angles (θ_i, a_i, d_i, α_i). values. If the joint is revolute or rotating, the joint angles represent the angle b/w the 2 linkages; if joint is neither prismatic nor sliding, the joint angles are the link extension. The kinematics of inverse that will introduce in the chapter's following section and deals with figuring out the values of the variables related to joint motion that attain a desired position, is to be compared with the forward kinematics. This is shown in figure 1.

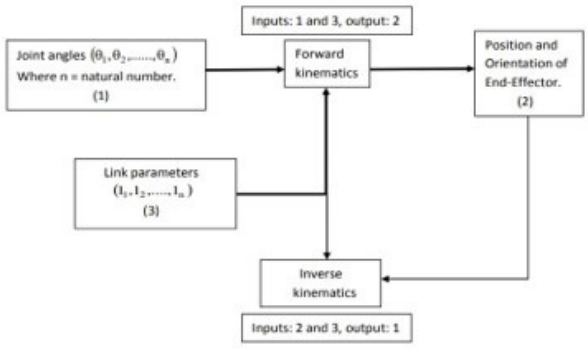

Figure 1: Inverse and forward kinematics scheme.

Inverse kinematics

Robotic kinematics of inverse problem has to do with the recognition of all viable as well as appropriate set of joint angles which comprehend a solution to determine the end effector locations and orientation. When compared to forward solutions, the inverse kinematics issue does not always specify a unique solved answer, i.e., the number of solutions to be acquired for 1 end effector to reach the required orientation and positions [32].

Case 1: 2 linkage planar manipulator

Contemplate a 2-link planar robotic arm with lengths of linkage as l_1, l_2 and joint angles as θ_1 and θ_2 to achieve the required positions as (P_x, P_y) represented and also [33] In this case, the inverse solution is found via a geometric method, as illustrated in figure 2 is as follows.

$$p_x = l_1 \cos\theta + l_2 \cos\left(\theta_1 + \theta_2\right) \quad (1)$$

$$p_y = l_1 \sin\theta + l_2 \sin\left(\theta_1 + \theta_2\right) \tag{2}$$

Now,

$$p_x^2 = l_1^2 \cos^2\theta_1 + l_2^2 \cos^2\left(\theta_1 + \theta_2\right) + 2l_1 l_2 \cos\theta_1 \cos\left(\theta_1 + \theta_2\right)$$

$$p^2 = l_1^2 \sin^2\theta_1 + l_2^2 \sin^2\left(\theta_1 + \theta_2\right) + 2l_1 l_2 \sin\theta_1 \sin\left(\theta_1 + \theta_2\right)$$

$$p_x^2 + p_y^2 = l_1^2\left(\cos^2\theta_1 + \sin\theta_1^2\right) + l_2^2\left(\cos^2\left(\theta_1 + \theta_2\right) + \sin^2\left(\theta_1 + \theta_2\right)\right) + 2l_1 l_2\left(\cos\theta_1 \cos\left(\theta_1 + \theta_2\right) + \sin\theta_1 \sin\left(\theta_1 + \theta_2\right)\right)$$

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2\left(\cos\theta_1\left(\cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2\right) + \sin\theta_1\left(\sin\theta_1 \cos\theta_2 + \cos\theta_1 \sin\theta_2\right)\right)$$

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2\left(\cos^2\theta_1 \cos\theta_2 - \cos\theta_1 \sin\theta_1 \sin\theta_2 + \sin^2\theta_1 \cos\theta_2 + \sin\theta_1 \cos\theta_1 \sin\theta_2\right)$$

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2\left(\cos\theta_2\left(\cos^2\theta_1 + \sin^2\theta_1\right)\right)$$

$$\cos\theta_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2}$$

Now,

$$\cos\theta_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{3}$$

$$\sin\theta_2 = \pm\sqrt{1 - \left(\frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)^2} \tag{4}$$

Finally, there are 2 solutions which are possible for θ_2 can be written as:

$$\theta_2 = \arctan 2\left(\pm\sin\theta_2, \cos\theta_2\right) \tag{5}$$

Then, multiply each side of equation 1 by $cos\theta_1$ and equation 2 by $sin\theta_2$ now add equations which gives solutions in a sequence to determine the answers of θ_1 in terms of parameter linkages and the angle which is known as θ_2.

$$\cos\theta_1 p_x = l_1 \cos^2\theta_1 + l_2 \cos^2\theta_1 \cos\theta_2 - l_2 \cos\theta_1 \sin\theta_1 \sin\theta_2$$

$$\sin\theta_1 p_y = l_1 \sin^2\theta_1 + l_2 \sin^2\theta_1 \cos\theta_2 + l_2 \sin\theta_1 \cos\theta_1 \sin\theta_2$$

$$\cos\theta_1 p_x + \sin\theta_1 p_y = l_1\left(\cos^2\theta_1 + \sin^2\theta_1\right) + l_2 \cos\theta_2\left(\cos^2\theta_1 + \sin^2\theta_1\right)$$

$$-\sin\theta_1 p_x = -l_1 \sin\theta_1 \cos\theta_1 - l_2 \sin\theta_1 \cos\theta_1 \cos\theta_2$$

$$+l_2 \sin^2\theta_1 \sin\theta_2 \cos\theta_1 p_y = l_1 \sin\theta_1 \cos\theta_1$$

$$+l_2 \cos\theta_1 \sin\theta_1 \cos\theta_2 + l_2 \cos^2\theta_1 \sin\theta_2 - \sin\theta_1 p_x$$

$$\cos\theta_1 p_x + \sin\theta_1 p_y = l_1 + l_2 \cos\theta_2$$

Now the equation which is simplified is obtained as follows:

$$\cos\theta_1 p_x + \sin\theta_1 p_y = l_1 + l_2 \cos\theta_2 \tag{6}$$

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = l_2 \sin\theta_2 \tag{7}$$

Now, we have to multiply every specific area of equation 6 by *px* and equation 7 by *py* also make a sum of equations of solutions in a sequence to get $cos\theta_1$:

$$\cos\theta_1\left(p_x^2 + p_y^2\right) = p_x\left(l_1 + l_2 \cos\theta_2\right) + p_y l_2 \sin\theta_2$$

So therefore,

$$\cos\theta_1 = \frac{p_x + \left(l_1 + l_2 \cos\theta_2\right) + p_y l_2 \sin\theta_2}{p_x^2 + p_y^2} \tag{8}$$

"$\sin\theta_1$" is obtained as:

$$\sin\theta_1 = \pm\sqrt{1 - \left(\frac{p_x\left(l_1 + l_2 \cos\theta_2\right) + p_y l_2 \sin\theta_2}{p_x^2 + p_y^2}\right)^2} \tag{9}$$

Therefore, the 2 solutions for θ_1 can be written as follows:

$$\theta_1 = \arctan 2\left(\pm\sqrt{1 - \left(\frac{p_x + \left(l_1 + l_2 \cos\theta_2\right) + p_y l_2 \sin\theta_2}{p_x^2 + p_y^2}\right)}, \frac{p_x + \left(l_1 + l_2 \cos\theta_2\right) + p_y l_2 \sin\theta_2}{p_x^2 + p_y^2}\right) \tag{10}$$

By using equation 5 and equation 10,

Joint angles are discovered by swapping the provided input variables. This approach involves lengthy determinations, which grow as the number of links rises. As a result, some programming routines must be written. Until now, the researchers have used ANN, MATLAB, C and C++, logic of fuzzy, and some approaches which are based upon. The work, a code of programming to kinematics of inverse type of an automated manipulator was built in an easy-to-understand manner for present generation humans. "Python" a highly systematic and simple coding language. Numerical analysis is one of python's most popular applications. The use of computer languages to solve mathematical analytic issues simplifies the work. Python requires fewer skills to code than other languages used for programming like C/C++/Java.

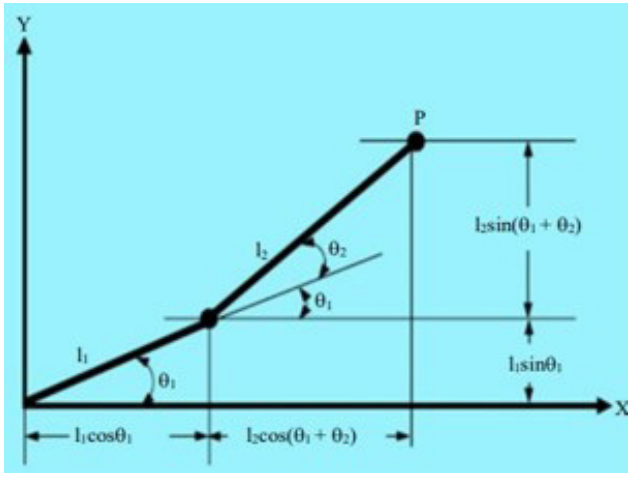

Figure 2: Geometrical model of a 2-link manipulator.

Case 2: 3 linkage RR planar manipulator

Contemplate a 3-linked RR planar robotic arm [34] that moves along the plane of x and y. A similar geometric method was used to derive the inverse solution for this mechanism.

However, the solutions of this type of method found as the no. of linkages increased, the series of action of answers became more complicated. As a result, the geometric method yields a complicated solution for serial planer manipulators. In this regard, python code provides simple methods to obtain a solution, which saves time and delivers correct performance while executing the software when compared to MATLAB. Thus, the programs output code of configurations such upside of elbow and downside of elbow.

Python

Python, which is well-known language in programming. [35] van Rossum has discovered it, so it came out in year 1991 and utilized as: Web developing (server-side), Software development, Mathematics, and System scripting. Python on a several type of sources, including Windows, Mac, Linux, and Raspberry Pi. This language has an easy-to-understand syntax which is similar to English. Syntax of this language enables programmers to develop program for less liners than in program of different languages.

This language is based on interpreter language that means code might be executed shortly when it is entered. This results that prototype can be skilfully done quickly. Python can be processed procedural, object-oriental, or functionally. Python stands apart from other languages used for programming due to its strong and huge standard library. Its standard library offers a large number of modules, operations, and web tools for service that you may use in your projects of applications without writing any more code.

Results and Discussion

The geometric method for inverse kinematics comprises a large number of mathematical expressions. Increasing the number of links adds greater difficulty to find solution by hand, but it is a less easy in MATLAB. By the study of this, a code that simplified was developed in the environment of python in a simple manner, of quick results generated which are output. So current discussed which is used 2 and 3 linked planer robotic arms, but the similar method will be used for complicated robot connections. About this scenario, links with lengths of 15 cm and 10 cm were chosen to reach a location (15 centimetres, 5 centimetres) that serves as the xy positions of the robotics arm end effector or tip. Equation 5 and equation 10 are used to calculate the variable joint of manipulator linkages and after results are compared with output results of python coding.

Case 1: Python code for 2 linked planar robotic arm

Figure 2 presents geometrical model of a 2-link manipulator. Figure 3 presents geometric model diagram of a 3 linked robotic arm. A code for kinematics of inverse of 2-linked RR planar robotic arm within an environment of python is provided below in figure 4. Table 1 presents inputs and outputs of 2D planar robotic manipulator.

Case 2: Python code for 3 linked planar robotic arm

Figure 5 presents 2 linked robotic arm of elbow upside orientation and position. Figure 6 presents 2 linked robotic arm of elbow downside orientation and position. A code for kinematics of inverse of 2-linked RR planar robotic arm within an environment of python is provided below in figure 7. Table 2 presents inputs and outputs of 3D planar robotic manipulator.

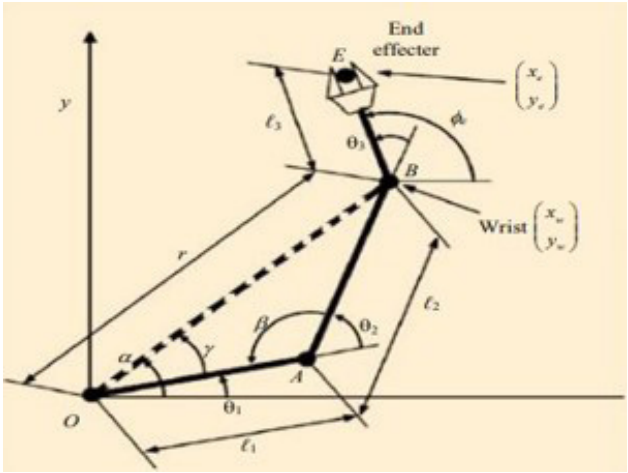

Figure 3: Geometric model diagram of a 3 linked robotic arm.

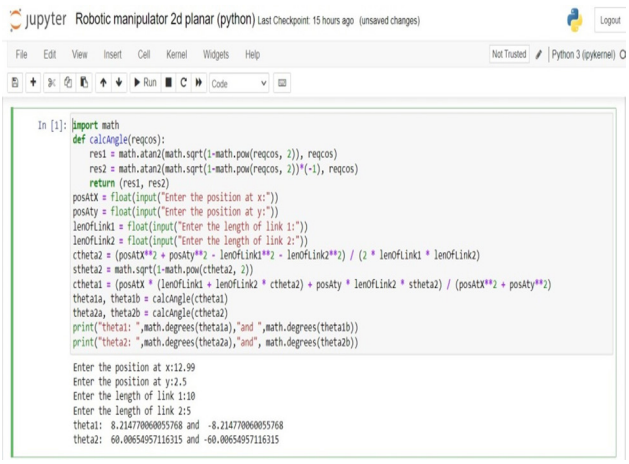

Figure 4: Python coding for 2D planar robotic arm.

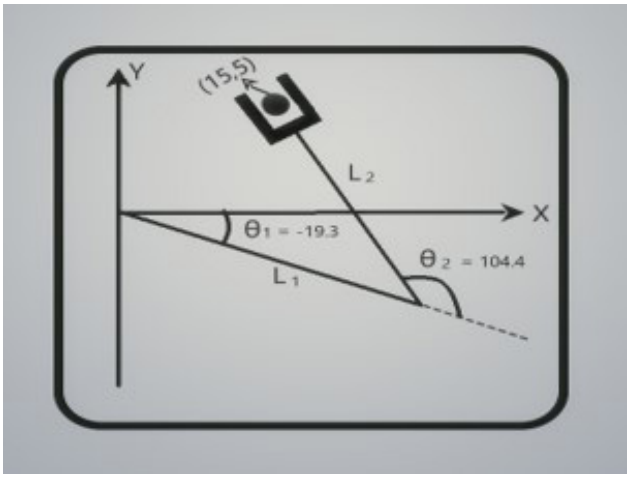

Figure 5: 2 linked robotic arm of elbow upside orientation and position.

Table 1: Inputs and outputs of 2D planar robotic manipulator

S. No.	p_x	p_y	l_1	l_2	θ_1 (Outputs)	θ_2 (Outputs)
1	16 cm	13 cm	10 cm	20 cm	33.258 and -33.258	100.806 and -100.806
2	15 cm	5 cm	15 cm	10 cm	19.326 and -19.326	104.477 and -104.477
3	12.99 cm	2.5 cm	10 cm	5 cm	8.214 and -8.214	60.006 and -60.006

Table 2: Inputs and outputs of 3D planar robotic manipulator.

S. No.	x_c	y_c	Phi	l_1	l_2	l_3	θ1 (Outputs)	θ_2 (Outputs)	θ_3 (Outputs)
1	16 cm	13 cm	180°	20 cm	10 cm	7 cm	10.275 and 48.676	60.330 and -60.330	109.394 and 191.653
2	15 cm	5 cm	110°	15 cm	10 cm	4 cm	-32.437 and 41.110	100.669 and -100.669	41.767 and 169.559
3	12.99 cm	2.5 cm	130°	10 cm	5 cm	2 cm	-8.297 and 16.055	37.128 and -37.128	101.169 and 151.072

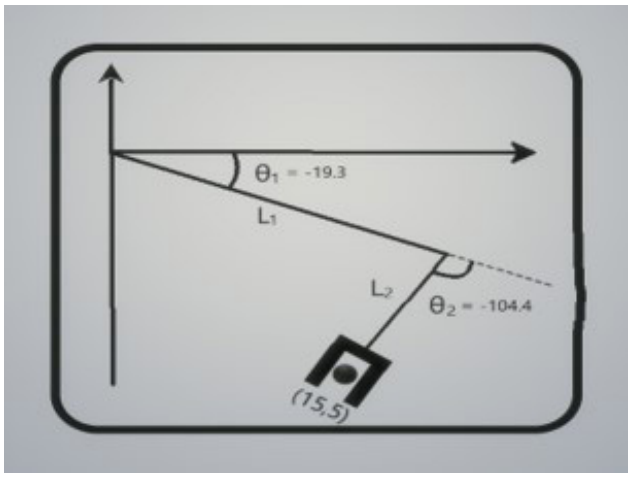

Figure 6: 2 linked robotic arm of elbow downside orientation and position.

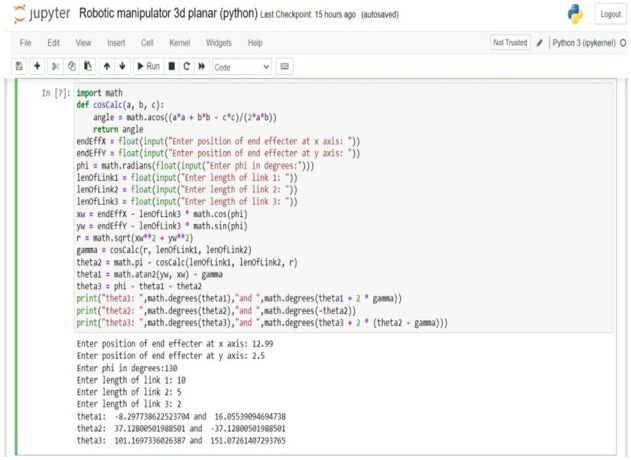

Figure 7: Python coding for 3D planar robotic manipulator.

Conclusion

A suggested technique is proved for 2D and 3D linked planer robotic arms; it must be investigated of couple more complex robot linkage configurations, so as larger than 2-DOF robotic arms and in links which are parallel. So, it may accomplish too many modelling scenarios where a solution to the numerical format in a mathematical form as the kinematics of inverse is implemented using coding of python for software. In this work python software has been utilized for a study of kinematics in inverse of a 2-DOF and 3-DOF planar robotic arms. Python software is adopted for solving the kinematics in inverse case of higher DOF of robotic manipulator. This technique is particularly efficient in anticipating the kinematics in inverse case for highest DOF robot manipulators because of its compactness and adaptable nature. As a result, this technology may be utilised in a variety of robots in a variety of fields to determine joint variables, positions, and the robotic space of working in order to keep away from obstacles. In the further generation we can develop for a higher degree of freedom robotic manipulators using python software like for 5-DOF and 6-DOF we can find the joint angles with python coding. When compared to other networks like fuzzy logic, C, and C^{++}. Python is highly well organized and simple. We can also develop for many applications of robotic manipulators.

Acknowledgements

I wish to acknowledge the help provided by the department of Mechanical Engineering and I would like to express my gratitude to the management, Vardhaman College of Engineering.

Conflict of Interest

None.

References

1. Deb SR, Deb S. 2010. Robotics Technology and Flexible Automation. McGraw-Hill Education.
2. Kircanski M, Vukobratovic M. 1985. Computer-aided generation of manipulator kinematic models in symbolic form. *Fifteenth ISIR* 1043-1049.
3. Kircanski M, Vukobratovic M. 1986. A new program package for generating symbolic kinematic models of arbitrary serial-link manipulators. *Sixteenth ISIR* 249-258.
4. Hussain MA, Noble B. 1984. Application of Macsyma to Kinematics and Mechanical Systems. In Pavelle R (ed) Applications of Computer Algebra. Springer, Boston, pp 262-280.
5. Tsai MJ, Chiou YH. 1989. Symbolic Equation Generation for Manipulators. In Waldron KJ (ed) Advanced Robotics: 1989. Springer, Berlin, pp 35-61.
6. Karlik B, Aydin S. 2000. An improved approach to the solution of inverse kinematics problems for robot manipulators. *Eng Appl Artif Intell* 13(2): 159-164. https://doi.org/10.1016/S0952-1976(99)00050-0

7. Khatib O. 1987. A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J Robot Automat* 3(1): 43-53. https://doi.org/10.1109/JRA.1987.1087068
8. Lloyd J, Hayward V. 1993. Trajectory generation for sensor-driven and time-varying tasks. *Int J Robot Res* 12(4): 380-393. https://doi.org/10.1177/027836499301200405
9. Mandava RK, Vundavilli PR. 2016. Forward and inverse kinematic based full body gait generation of biped robot. In International Conference on Electrical, Electronics, and Optimization Techniques, Chennai, Tamil Nadu, India.
10. Sreenivasulu R. 2012. Simulation of desired end point trajectory for a 2-dof planar manipulator. *Int J Adv Sci Tech Res* 5: 688-696.
11. Nugroho SA, Prihatmanto AS, Rohman AS. 2014. Design and implementation of kinematics model and trajectory planning for NAO humanoid robot in a tic-tac-toe board game. In IEEE 4th International Conference on System Engineering and Technology, Bandung, Indonesia.
12. Sadiq AT, Raheem FA, Abbas NAF. 2017. Optimal trajectory planning of 2-DOF robot arm using the integration of PSO based on D* algorithm and cubic polynomial equation. *First Int Conf Eng Res* 458-467.
13. Chaitanyaa G, Reddy S. 2016. Genetic algorithm based optimization of a two link planar robot manipulator. *Int J Lean Think* 7: 1-3.
14. Kanayama Y, Kimura Y, Miyazaki F, Noguchi T. 1990. A stable tracking control method for an autonomous mobile robot. In Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA.
15. Mohamed MG, Duffy J. 1985. A direct determination of the instantaneous kinematics of fully parallel robot manipulators. *J Mech Trans Automat* 107(2): 226-229. https://doi.org/10.1115/1.3258713
16. Jones BA, Walker ID. 2006. Kinematics for multisection continuum robots. *IEEE Trans Robot* 22(1): 43-55. https://doi.org/10.1109/TRO.2005.861458
17. Radavelli L, Simoni R, De Pieri E, Martins D. 2012. A comparative study of the kinematics of robots manipulators by Denavit-Hartenberg and dual quaternion. *Mecánica Comput* 31(15): 2833-2848.
18. Chen Q, Zhu S, Zhang X. 2015. Improved inverse kinematics algorithm using screw theory for a six-DOF robot manipulator. *Int J Adv Robot Syst* 12(10): 140. https://doi.org/10.5772/60834
19. Chirikjian GS. 1994. Kinematics of a metamorphic robotic system. In Proceedings of International Conference on Robotics and Automation, San Diego, CA, USA.
20. Craig J. 1986. Introduction to Robotics: Mechanics and Control. Pearson Publishing.
21. O'Malley M. 2011. Introduction to robotics, inverse manipulator kinematics. *Int J Soft Comput* 2: 1-8.
22. Murray RM, Li Z, Sastry SS. 2017. A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton.
23. Raheem FA, Sadiq AT, Abbas NAF. 2019. Robot arm free Cartesian space analysis for heuristic path planning enhancement. *Int J Mech Mechatron Eng* 19: 29-42.
24. Hudgens JC. 1988. A fully-parallel six degree-of-freedom micromanipulator: kinematic analysis and dynamic model. *ASME Design Tech Conf* 29-37.
25. Kumar SN, Shukur JA, Sriker K, Lavanya A. 2014. Design and structural analysis of skid landing gear. *Int J Curr Eng Technol* 635-642. http://dx.doi.org/10.14741/ijcet/spl.2.2014.121
26. Sun JD, Cao GZ, Li WB, Liang YX, Huang SD. 2017. Analytical inverse kinematic solution using the DH method for a 6-DOF robot. In 14th International Conference on Ubiquitous Robots and Ambient Intelligence, Jeju, South Korea.
27. Tsai LW, Morgan AP. 1985. Solving the kinematics of the most general six-and five-degree-of-freedom manipulators by continuation methods. *J Mech Trans Automat* 107(2): 189-200. https://doi.org/10.1115/1.3258708
28. Zhuang H, Roth ZS, Hamano F. 1990. A complete and parametrically continuous kinematic model for robot manipulators. In Proceedings of International Conference on Robotics and Automation, Cincinnati, OH, USA.
29. Veitschegger W, Wu CH. 1986. Robot accuracy analysis based on kinematics. *IEEE J Robot Automat* 2(3): 171-179. https://doi.org/10.1109/JRA.1986.1087054
30. Webster RJ, Jones BA. 2010. Design and kinematic modeling of constant curvature continuum robots: a review. *Int J Robot Res* 29(13): 1661-1683. https://doi.org/10.1177/0278364910368147
31. Yang C, Ma H, Fu M. 2016. Advanced Technologies in Modern Robotic Applications. Springer, Singapore.
32. Nanosciences, Nanotechnologies, Materials and New Production Technologies. [https://cordis.europa.eu/programme/id/FP7-NMP] [Accessed November 27, 2023]
33. Xu H, Wu X, Tian X, Li J, Chu J, et al. 2019. Dynamic structure-properties characterization and manipulation in advanced nanodevices. *Mater Today Nano* 7: 100042. https://doi.org/10.1016/j.mtnano.2019.100042
34. Yang YL, Lou JQ, Wu GH, Wei YD, Fu L. 2018. Design and position/force control of an S-shaped MFC microgripper. *Sens Actuators A Phys* 282: 63-78. https://doi.org/10.1016/j.sna.2018.09.021
35. van Rossum G. 2003. An Introduction to Python. Network Theory Limited.