Development of Educational Program Code for Solving Kinematic Problems of RRR Type Robots

László Rónai

University of Miskolc, Miskolc-Egyetemváros 3515, Hungary,
laszlo.ronai@uni-miskolc.hu

Abstract. With the rapid development of technology, intuitive educational tools are becoming increasingly important and popular, not only in elementary and secondary schools, but also in universities. This paper deals with the development of a program code and a visualization window, which can help students to better understand robot kinematics. To solve the forward kinematic problem, the Denavit-Hartenberg convention is used. The inverse kinematic problem is described using a geometric approach. The article includes the presentation of the developed program code and its visualization interface for a 5 DOF robot, whose first three links are connected by revolute joints, forming an RRR kinematic chain.

Keywords: industrial robot, forward kinematics, inverse kinematics, Scilab

1 Introduction

Nowadays, it is increasingly important to use various technical tools in the education of students to help them understand better or clarify problems. This approach is also increasingly present in engineering, e.g., illustrating various machine elements with 3D design systems or visualizing a PLC-controlled process serves more to understand and clarify. Several articles address the field of robotics from an educational technology perspective.

Article [1] presents a robotics education system, which integrates virtual reality, remote control, and a physical laboratory environment to teach the kinematics of robots. A web-based simulation platform was presented including 6-DOF and SCARA robots that allow students to perform experiments safely compared to the real-world robotic tasks. This hybrid approach can improve learning efficiency and reduce resource requirements. A comprehensive overview of 17 robotics simulators used in education, highlighting their role in improving accessibility, reducing costs, and enabling safe experimentation is detailed in [2]. Furthermore, interactive, visual simulation environments significantly enhance understanding of complex robotic concepts.

Paper [3] provides a bibliometric and visualization-based analysis of research in educational robotics, identifying key authors, trends, and thematic developments in the field. It reports that educational robotics has a positive impact on

© The Author(s) 2026
S. Árvai-Homolya et al. (eds.), *Proceedings of the 2nd Innovation in Engineering Education Conference (IEE 2026)*, Atlantis Highlights in Social Sciences, Education and Humanities 58,
https://doi.org/10.2991/978-94-6239-823-8_15

learning success. An educational approach where students learn graph interpretation and construction in a kinematics context using a wireless robot and portable devices is presented in [4]. The study shows that this robot-supported collaborative learning environment significantly improves the ability to understand graphs. Another educational robotics approach for teaching advanced mathematical concepts such as Euclidean and Infinity metrics, as well as the convergence of Cauchy sequences, using a LEGO Mindstorms-based mobile robot is presented in [5]. The article shows that a physically implemented visualization can help students to better understand mathematical ideas by making them more intuitive. The robotic experiments are crucial to improve student engagement to use mathematical concepts.

Paper [6] presents a VR-based training and control framework where users can intuitively teach robot configurations that are later used for inverse kinematics control in medical robotic systems. The results show that task performance is improved using visual and interactive VR-based teaching. The study highlights that integrating visualization with inverse kinematics enables more intuitive understanding and more efficient manipulation of complex robotic systems.

In [7], authors present a reinforcement learning-based inverse kinematics solution algorithm (MAPPO-IK), they implicitly highlight the importance of simulation and data-driven visual environments in training and validating robotic control policies. A simulation program is developed in Python to make forward- and inverse kinematic calculations with visualization [8]. They highlighted that their program is useful for learners and educators to understand kinematics of robots. Teaching robot kinematics without visualization is not effective enough, interactive simulations significantly improve understanding. The future will promote greater involvement of digital systems for easier understanding.

One of the key tools of industrial automation is the robot. At the University of Miskolc, theoretical and practical education is given a prominent role in the mechatronics engineering education. The specialization of this training is called industrial robots. When describing the forward and inverse kinematics tasks of robots, a visualization interface can greatly assist in interpreting, e.g., the TCP point or the orientation unit vectors. The main aim of the article is to develop a program code, which demonstrates kinematics of industrial robots.

The article is organized as follows: Section 2 describes the forward- and inverse kinematic problems of robot composing of serial kinematic chain. Section 3 introduces the self-devised program for demonstrating the kinematics of such robots. The final Section contains the conclusions and the future plans.

2 Forward- and inverse kinematics of robots

Understanding forward and inverse kinematics problems is particularly important in the field of mechatronics education. When developing robotic arms, their geometric description must be ensured, therefore that the operating program can correctly generate the joint angles given the target position.

In the case of robotic arms constructed with single-degree-of-freedom joint constraints, forward kinematics problem means that the individual joint angles are known, and the goal is to calculate the position and orientation of the end-effector measured from the base. In contrast, during the inverse kinematics problem the individual joint angles are determined by knowing the target position.

2.1 Forward kinematics of RRR type robots

The Denavit-Hartenberg parametric description method is a widespread technique for the geometric description of robotic arms. It helps to solve the forward kinematic problem of industrial robotic arms in a uniform and easy-to-handle way. The result matrix gives the relationship between TCP point of the end-effector and the robot base, i.e., the position and orientation of the TCP point depending on the joint angles between the links in the case of rotational joints. It uses four parameters, interpreted between two coordinate systems, connected by the link of a robot. There are Denvait-Hartenberg classical and modified conventions [9]. Between the two techniques the difference lies in the placement of coordinate frames and the order in transformations applied. This leads to different formulations of the homogeneous transformation matrices. In this article, the classical convention is used.

Coordinate systems need to be attached to each joint, for which there is a rule:

- The axis of the i^{th} joint z_{i-1} is always the axis of rotation of the joint.
- The x_i axis is the common perpendicular between z_i and z_{i-1}, directed from z_{i-1} to z_i.
- The y_i axis is defined to form a right-handed coordinate system.

According to [10] the D-H matrix is the product of four elementary transformations, which gives the homogeneous transformation:

$$\mathbf{T}_i = \mathbf{Rot}_{z,\theta_i}\mathbf{Trans}_{z,d_i}\mathbf{Trans}_{x,a_i}\mathbf{Rot}_{x,\alpha_i}, \tag{1}$$

where a_i is the link length, d_i is called link offset. The remaining two parameters α_i, θ_i are link twist and joint angle, respectively. All of them belongs to the i^{th} robotic link. By performing the products of the elementary transformations the resulting matrix is the following:

$$\mathbf{T}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

By calculating each transformation matrix between two coordinate systems and then multiplying them, the relationship between the base and the end-effector can be obtained:

$$\mathbf{DH}_{0,n} = \prod_{i=1}^{n} \mathbf{T}_i, \tag{3}$$

where n indicates the degree of freedom of the given robot. The upper left 3x3 part of the matrix $\mathbf{DH}_{0,n}$ provides the rotational part, which is important to describe the orientation of the end-effector. Describes how the coordinate system of the end-effector is rotated relative to the base coordinate system. The first column gives the x axis of the end-effector expressed in the base coordinate system, the axes y and z are given by the 2^{nd} and 3^{rd} columns, respectively. The first three values of the last column of the 4x4 transformation matrix provide the end-effector position relative to the base.

During robotics education, students' understanding of the elements of the transformation matrix can be greatly facilitated by developing targeted program code and visualization.

2.2 Inverse kinematics problem

In order to solve inverse kinematics problem of a robot, there are several techniques: analytical method, numerical method and geometrical approach. The latter one is common if the robot consists of serial kinematic chain. Figure 1 represents a 3 DOF robot, and its parameters to formulate the joint angles with the geomteric approach. The coordinates of the TCP point are x_c, y_c and z_c, respectively.

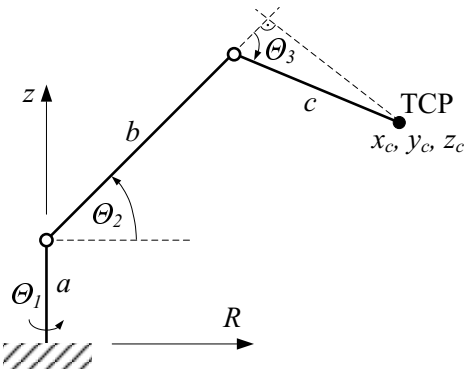


Fig. 1. Kinematic sketch of a 3 DoF robot

The first joint angle θ_1 can be calculated as

$$\theta_1 = \text{atan2}(y_c, x_c), \tag{4}$$

where atan2() is the four-quadrant inverse tangent. Thereafter the third joint angle is determined:

$$\theta_3 = \text{atan2}(\sin(\theta_3), \cos(\theta_3)), \tag{5}$$

where $\cos(\theta_3)$ and $\sin(\theta_3)$ can be defined as:

$$\cos(\theta_3) = \frac{L^2 - (b^2 + c^2)}{2bc}; \quad \sin(\theta_3) = \pm\sqrt{1 - \cos^2(\theta_3)}, \tag{6}$$

where $\cos\theta_3$ is determined by the use of law of cosines and supplementary angles, L is:

$$L = \sqrt{R^2 + (z_c - a)^2}, \tag{7}$$

where R is:

$$R = \sqrt{x_c^2 + y_c^2}. \tag{8}$$

θ_2 can be defined in two parts:

$$\theta_2 = \operatorname{atan}\left(\frac{z_c - a}{R}\right) + \operatorname{atan}\left(\frac{c\sin(\theta_3)}{b + c\cos(\theta_3)}\right). \tag{9}$$

With a larger degree of freedom, the task becomes redundant, therefore additional kinematic or task-specific conditions (e.g. fixing the orientation of the end-effector) must be specified to select a clear solution. For example, in the case of 5 DOF robot, the degrees of freedom remaining after determining the position give the orientation, therefore additional conditions (e.g. the perpendicular position of the gripper relative to the plane of the table) are required to clearly designate them.

3 Self-devised program and its visualization

A program code in Scilab software (SciNotes) has been written in order to visualise the links of a 5 DOF robot, the coordinate systems. The students can use this program to understand the building of the transformation matrices, the meaning of the orientation vectors and they can check the behaviour of the robotic arm if the parameters, e.g., length of the robot links, reference or target positions are modified. Furthermore, they can improve their programing skills.

Table 1. Defining the function for solving the inverse kinematics task

```
function [t1, t2, t3, t4, t5]=invkin(pos)
    R=sqrt(pos(1,1)^2+pos(2,1)^2);
    L=sqrt(R^2+(pos(3,1)+d-a)^2);
    t1=atan(pos(2,1),pos(1,1));
    ctet3=(L^2-b^2-c^2)/(2*b*c);
    stet3=sqrt(1-ctet3^2);
    t3=atan(stet3,ctet3);
    alfa=atan((pos(3,1)+d-a)/R);
    beta=atan((c*stet3)/(b+ctet3*c));
    t2=(alfa+beta);
    t4=(t2-t3)+%pi/2;
    t5=-t1;
endfunction
```

Figure 2 shows the process of the developed program code. The first task is to define the functions belonging to the formulae of the inverse kinematics, Denavit-Hartenberg classical convention and the visualisation. The 4 DH parameters for each link are also defined. In the next step, the joint angles of the reference position can be added by the user. Their correctness are checked in the next process. If the values are out of the defined range, the program is sending an error message. In case of correct values, the DH transformation matrices are calculated between two coordinate systems, followed by drawing the robotic arm in space. Thereafter the user can add the coordinates of the target position, which is checked. Currently, the parameters can be modified by the user in the program code. If the values are correct, the inverse kinematic task is solved.

In Table 1 the function for inverse kinematics is shown, joint angles θ_1 – θ_5 are denoted by t_1 – t_5. Most of the notations are consistent with those in subsection 2.2. The parameter d represents the length of the 4th link. Joint angles θ_4 and θ_5 are defined so that the robot's end-effector is perpendicular to the plane of the table and its orientation does not change compared to the reference position.

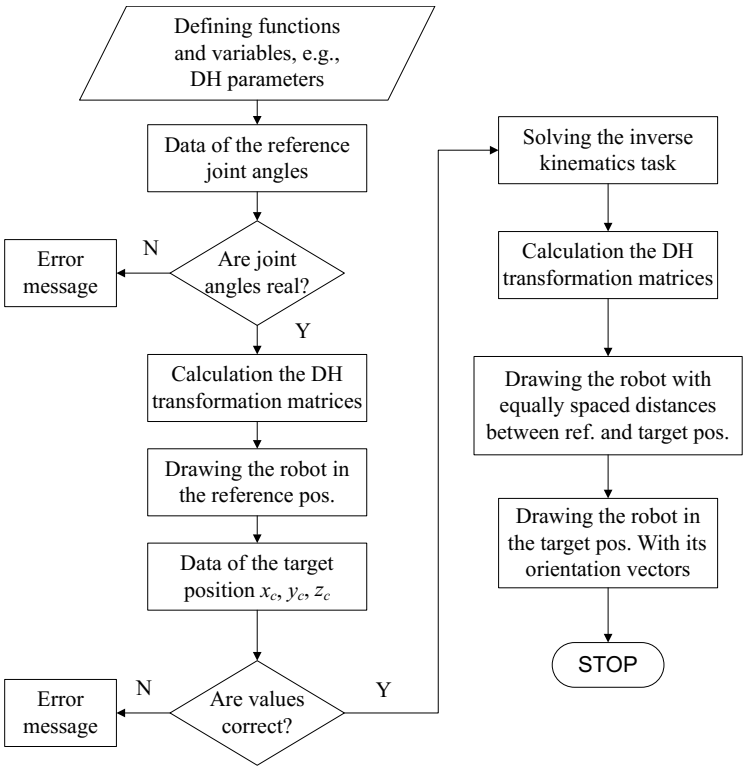


Fig. 2. Flowchart of the developed program code

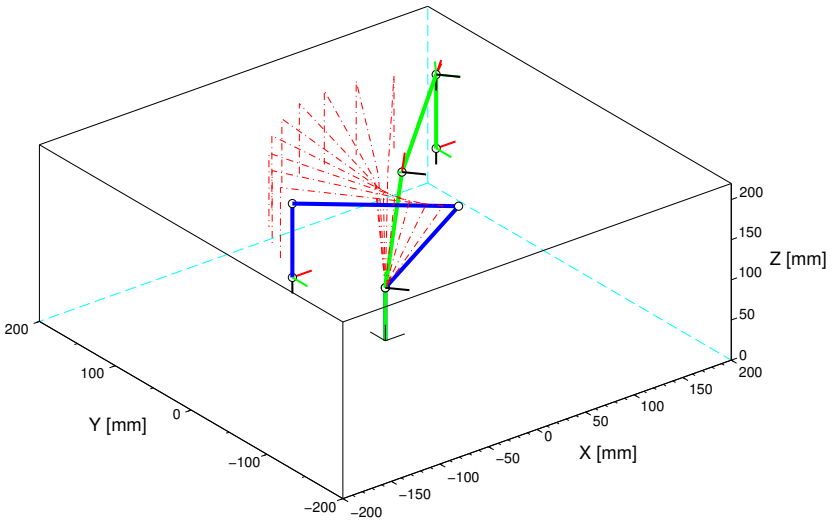


Fig. 3. Visualisation of the robot coordinate system, the orientations

After determining the necessary joint angles, the forward kinematics problem is solved. Finally, the robotic arm can be drawn in intermediate positions and finally in the target position, marking the orientation vectors, which can be seen in Figure 3. The robot centerline drawn in blue belongs to the reference position, while the robotic arm is denoted with green belongs to the target position. The number of intermediate drawn positions can be changed in the program, they are illustrated with red dotted lines. In the target position, all orientation vectors are depicted.

4 Conclusions

The article describes a program code developed in the Scilab environment for educational use, which facilitates students' kinematic studies related to robots. The literature review has shown that such educational visualization tools are relevant for understanding the curriculum. The developed program code is suitable for simulating and visualizing the motion of a 5 DOF robot arm whose first three links are connected by revolute joints (RRR type). Students can more easily understand forward and inverse kinematic calculations. Furthermore, they can also practice algorithmization.

In the future, the program code will be further developed to include a data input window enabling the modification of parameters, coordinates, and angles not

only within the source code. Further developments will also include the detection of singular configurations and the selection between elbow-up and elbow-down solutions. The degrees of freedom of the robotic arm built with a serial kinematic chain can be freely adjusted. Furthermore, as future work, the developed program code will be incorporated into education, and a student survey will be conducted to evaluate the extent to which it contributes to the understanding of robot kinematics.

References

1. Xu X., Guo P., Zhai J., and Zeng X.: Robotic kinematics teaching system with virtual reality, remote control and an on–site laboratory. *International Journal of Mechanical Engineering Education*, vol. 48, no. 3, pp. 197–220 (2018). doi:10.1177/0306419018807376
2. Tselegkaridis S., and Sapounidis T.: Simulators in Educational Robotics: A Review. *Education Sciences*, vol. 11, no. 1, article 11 (2021). doi:10.3390/educsci11010011
3. Dou A., Xu W., and Feng Y.: A Visualized Exploration of Educational Robotics Over Five Decades. *International Journal of Technology and Human Interaction*, vol. 20, no. 1 (2024). doi:10.4018/IJTHI.356662
4. Mitnik R.,Recabarren M.,Nussbaum M., Soto A.: Collaborative robotic instruction: A graph teaching experience. *Computers & Education*, vol. 53, no. 2, pp. 330–342 (2009). doi:10.1016/j.compedu.2009.02.010
5. Suárez-Gómez A.D., and Pérez-Holguín W.J.: Physical visualization of math concepts using LEGO Mindstorms. *Journal of Technology and Science Education*, vol. 10, no. 1, pp. 72–86 (2020). doi:10.3926/jotse.788
6. Karnam M., Zelechowski M., Cattin P.C. et al.: User-specified inverse kinematics taught in virtual reality reduce time and effort to hand-guide redundant surgical robots. *Communications Engineering*, vol. 4, article no. 20, (2025). doi:10.1038/s44172-025-00357-x
7. Zhao C., Wei Y., Xiao J., Sun Y., Zhang D., Guo Q., and Yang J.: Inverse kinematics solution and control method of 6-degree-of-freedom manipulator based on deep reinforcement learning. *Scientific Reports*, vol. 14, article 12467 (2024). doi:10.1038/s41598-024-62948-6
8. Hajjaj S.S.H., and Pan Y.: Simplified Forward and Inverse Kinematics with Python: A Visual and Interactive Approach. In: Chen W., et al.: *Selected Proceedings from the 2nd International Conference on Intelligent Manufacturing and Robotics, ICIMR 2024, 22–23 August, Suzhou, China*. Lecture Notes in Networks and Systems, vol. 1316. Springer, Singapore (2025). doi:10.1007/978-981-96-3949-6_66
9. Reddy C.A.: Difference between Denavit-Hartenberg (D-H) Classical and Modified Conventions for Forward Kinematics of Robots with Case Study *International Conference on Advanced Materials and manufacturing Technologies (AMMT)*, pp. 267–286 (2014).
10. Spong M.W., Hutchinson S., and Vidyasagar M.: Robot Modeling and Control. Wiley (2006).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

