Received: 21 September 2016 | Accepted: 17 March 2017
DOI: 10.1002/cae.21828

RESEARCH ARTICLE

WILEY

Robot kinematics made easy using RoboAnalyzer software

Ratan S. Othayoth[1] | **Rajeevlochana G. Chittawadigi**[2] | **Ravi P. Joshi**[3] | **Subir K. Saha**[4]

[1] Department of Mechanical Engineering, Johns Hopkins University, Baltimore, Maryland

[2] Department of Mechanical Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amrita University, Bengaluru, India

[3] Department of Human Intelligence Systems, Kyushu Institute of Technology, Kitakyushu-shi, Fukuoka, Japan

[4] Department of Mechanical Engineering, Indian Institute of Technology Delhi, New Delhi, India

Correspondence
Rajeevlochana G. Chittawadigi, Department of Mechanical Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amrita University, Kasavanahalli, Carmelaram Post, Bangalore, 560035 Karnataka, India.
Email: rg_chittawadigi@blr.amrita.edu

Funding information
QIP (Quality Improvement Programme) CD (Curriculum Development) Cell at IIT Delhi

Abstract

RoboAnalyzer is a software based on 3D model of robots. It was developed primarily for teaching and learning of robot mechanics, although it is robust enough for the use by researchers as well. The motive behind the development of RoboAnalyzer was mainly to help teachers and students get started with teaching/learning of robotics using template-based skeleton models or CAD models of serial robots. This minimizes the time otherwise spent on modeling, programming, and simulating the robots from scratch. In this article, we focus on the visualization of the Denavit Hartenberg (DH) parameters used to define a robot's architecture, and the modeling of the robot's input-output motion characteristics, that is, robot kinematics, using them. The advantages of using RoboAnalyzer to overcome several challenges of learning robotics in a classroom environment are also discussed.

KEYWORDS
education, robot kinematics, robotics, simulation, visualization

1 | INTRODUCTION

Among various types of robots, serial-chain robots are used extensively in diverse applications in industry, space, healthcare, etc. Hence, it has been observed over the years that majority of the introductory level courses on robotics emphasize substantially on the mechanics of serial robots. Their kinematics and dynamics are generally not very intuitive to teach or learn, as they involve topics from linear algebra, coordinate transformations, and fundamentals of mechanics. Perceiving the geometry, architecture, and motion of the robot using only a textbook as a sole reference medium can be difficult. These factors underscore the need to have a robot learning/teaching software which can complement any textbook on robotics. A good learning tool can make the learning/teaching process more productive. With reduced or almost no effort to create, visualize, and simulate the model of a robot in the CAD environment, one can spend more time in learning its mechanics. It would also allow the course instructor to demonstrate the concepts and the robot motion in a classroom setup more conveniently.

Numerous software have been reported in the literature for the simulation of different kinds of robots. A thorough review of the software packages available for dynamic simulation of robots is reported in Ref. [14], where majority of these software need prerequisites such as a clear understanding of robot kinematics and dynamics, and a good understanding of an associated programming language. Although these can be excellent tools for students with firm grasp of robot mechanics to perform further in-depth analyses or research, they are not as effective in aiding a novice learner or student to comprehend robot mechanics. On the contrary, it was observed that the number of software that devote attention to teaching/learning of robotics is less, which are also in great need. A good review of such robotics learning/teaching software is provided in Refs. [2,31].

One of the software for learning/teaching of robotics basics is Robotics Toolbox [4] based on MATLAB functions. The latest version of it includes Machine Vision [5] as well. Robotica [21] is a software package for robot analysis in Mathematica environment. ROBOT-DRAW [25] is another software designed with the aim of aiding the visualization of robot geometry. It is an internet-based visualization tool based on Virtual Reality Markup Language (VRML). It presents several customizable robot models which help to study the effect of the DH (Denavit-Hartenberg) parameters on the robot architecture. It also has provisions for forward and inverse kinematics. RIO (Robotics Illustrative Software) [17] is an attempt in creating a web-based framework for learning robotics. It has a user interface that can be displayed using a web browser where virtual models of several robots are available. An educational Virtual Laboratory [3] was also introduced for teaching robotics. It allows the simulation of a virtual robot using a teach pendant. An advanced version of it was later released as RobUALab, which was integrated into a robotics course. The same research group has recently released a Java based robotics learning framework known as EjSRL [15]. It is an interactive software that allows modeling and simulation of generic serial-link robot manipulators. It also comes with implementation of a computer-vision algorithm and advanced functions for robot analysis. Simulation and remote-triggered control of an actual robot was discussed in Ref. [22] as a part of Virtual Labs, which allows a user to control a robot through internet. However, only one user can control the robot at a time. Some web-based approaches for aiding robotics education were reported in Refs. [7,18]. Robotect [20] is a software aimed at designing and analysis of serial-robot manipulators. V-REP [10] is another one for robotics learning and simulation. It allows a user to simulate robot manipulators and mobile robots in various environments by introducing virtual sensors and actuators. RoKiSim [27] is yet another software for robot simulation that allows Cartesian and joint-level jogging of six-axes serial robots. Recently, it was upgraded as RoboDK [26] and is being developed under the new name. On a similar note, Webots [19] is a development environment which focuses on modeling, programming, and simulation of mobile robots. It aims to reduce the time spent on developing mobile robot applications and has functionalities to interface with mobile robot hardware. ARTE [11] is another MATLAB toolbox that allows simulation and visualization of robot manipulators, both serial and parallel. It allows visualization of robot models, jogging of robot models using teach pendant, and plotting of simulation results. Recently, Build-A-Robot [9] was reported which uses MATLAB-Simulink to generate VRML models of robot links for effective visualization of DH parameters. Though it is a good tool to learn DH parameters, it has a dependency on MATLAB, which is not accessible to many students. An offline robot simluation toolbox called ROBOLAB is reported in Ref. [16] that focuses on educational users and helps to understand robot mechanics using 3D simulations.

While the above software emphasize on robotics education, other robot simulators cater to industrial and advanced research applications. For example, Architecture Design and Robot Simulation (ADRS) [12] is a tool that helps a user to design serial manipulators—a GUI that allows one to interactively build a serial robot. This was developed as a part of ADEFID [1] industrial package which is a set of graphics based modules for research in mechanical systems and industrial applications. Recently, a new module called SnAM (Serial n-Axis Manipulators) [13] has been added to the ADEFID package that allows forward and inverse kinematics of serial robots. Of late, Robot Operating System (ROS) [23] has been introduced into industrial (ROS Industrial) and research applications as a common standard. For ROS based applications, simulation, and visualization are often based on the Gazebo [28] simulator. While the software mentioned in this paragraph are powerful for industrial and research applications, they often have steep learning curves for the beginners in a classroom environment.

As evident from the above literature survey, several good functionalities are required in a teaching software. Some of them were implemented in the above mentioned software based on the importance felt by their developers. However, many are left out which may appear in their future versions. One of them is the concept of Denavit and Hartenberg (DH) [8] parameters used to define the architecture of a robot. It is ever confusing due to the existence of several versions in the literature that are evident from many standard text books on robotics, for example, [32,6]. Hence, this feature was introduced in robot teaching/learning software, RoboAnalyzer (RA), developed by the authors [2,24,29,30] but never emphasized or highlighted in the earlier publications. The RA encapsulates most of the other important features which are specifically required by the students to learn the subject of robotics in a more enjoyable way. A typical screenshot of the RA application is shown in Figure 1, whereas, the flowchart of programming-level implementations was reported in Refs. [24,29]. In order to avoid repetition and save space,

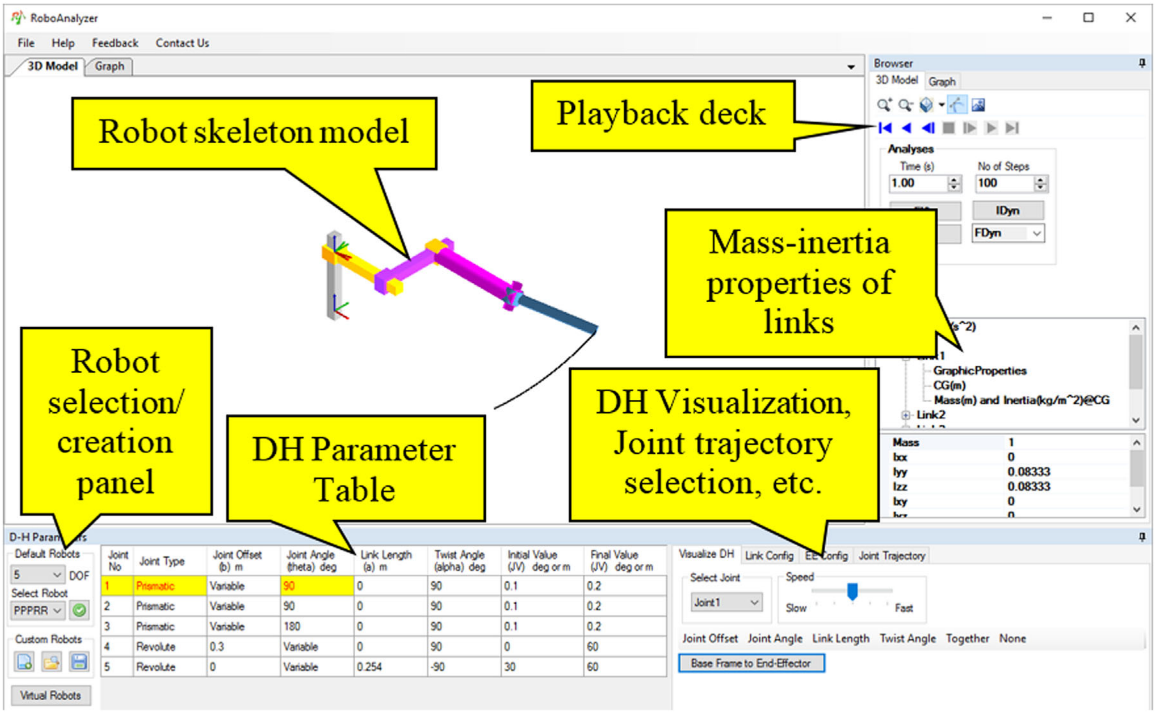


FIGURE 1 Graphical User Interface (GUI) of RoboAnalyzer (RA)

they are not given here. Interested readers may kindly refer Refs. [24,29].

The design philosophy behind RoboAnalyzer was as follows: While learning robot mechanics, its physics must not be obscured by the underlying mathematics, and must be clearly understood by a student as if he or she is moving a real robot. It has been developed with an objective of teaching and learning robotics using few regular shapes of the links. That way a student can get started with the kinematic and dynamic analyses of the robots without spending too much time in learning CAD modelling and/or programming a real robot if it exists. Though the authors agree that modeling and programming a robot model using a CAD or simulation software or programming a real robot could be a constructive experience, students may loose valuable time that could be otherwise spent on learning robotics. On the other hand, derivations of kinematic and dynamic equations by the students could be a real learning and rewarding experience, but some may not even attempt it due to the associated complexities. Hence, the need of RoboAnalyzer is justified.

In this paper, RoboAnalyzer (RA) software and its application in teaching only robot kinematics are presented even though the software can perform dynamics as well as some trajectory generation. This is intentional as it can be elaborated on how to fully exploit the RA software with respect to at least one important topic in robotics. The emphasis is given on the visualization of the DH parameters used to define a robot's architecture, and the problems or challenges that are faced while teaching or learning robot kinematics and how RA can be effective. To bring out the strengths of this software with regard to other similar existing software, a comparison is done in Table 1, where "•" and "X" imply feature "available" and "not available," respectively.

2 | TEACHING ROBOT ARCHITECTURE

In a typical course on robotics, its architecture or geometry is described using an approach which is commonly referred to as the Denavit and Hartenberg (DH) parameters [8]. Further, robot kinematics requires matrix algebra, coordinate transformations and multivariate equations, which will be taken up in section 3. They are not very intuitive if only a textbook is used. However, a robotics teaching/learning software with a visualization environment can help understand their physical manifestations, and thus helping one to understand the concepts better.

2.1 | Visualization of DH parameters

The description of robot architecture using the DH parameters can be fully appreciated if they are related to their underlying coordinate transformations. Given the limitations in a classroom environment, it may be difficult to understand the essence of these 3-dimensional transformations using the

TABLE 1 Comparison of features of different robotics teaching/learning software

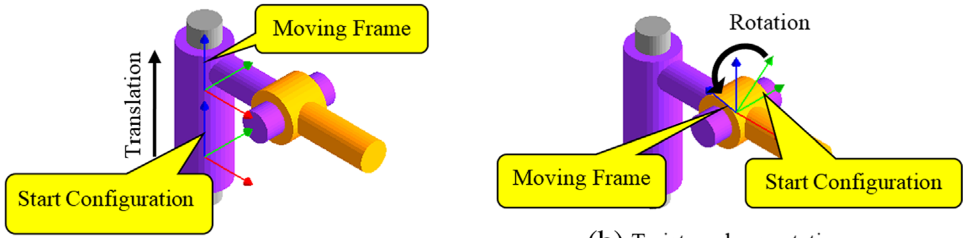
	Software						
Features	**Robotics toolbox [4,5]**	**V-REP [10]**	**RoboDK [26]**	**ARTE [11]**	**EJS + EjSRL [3,15]**	**Build a robot [9]**	**RoboAnalyzer**
Schematic skeleton models	•	•	X	•	•	•	•
Actual robot CAD models	•	•	•	•	•	X	•
Customizable robots	•	•	X	•	•	•	•
Degrees-of- freedom	Any, using commands	Any, using commands	Up to 6	Up to 6	Any, using commands	Up to 7	Any, using GUI
Multiple robots within same environment	•	•	•	•	•	X	X
Visualization of coordinate frames	•	•	•	•	•	•	•
Animation of homogeneous transformations	Using commands	Using commands	Not available	Using code	X	X	Using GUI
Animation of robot motion	•	•	•	•	•	•	•
End-effector trace	•	•	•	•	•	X	•
Plots	•	•	X	•	•	X	•
Trajectory models	•	•	•	•	•	X	•
Multiple solutions for inverse kinematics	•	•	•	•	X	X	•
Coding Prerequisites required	MATLAB	MATLAB	For advanced operation	MATLAB	JAVA	No	No
Display of transformation data	By command	By command	On-screen	By command	By command	On-screen	On-screen
Interfacing with MATLAB	MATLAB based	•	X	MATLAB based	•	MATLAB based	•

textbook figures alone. A 3D animation environment can help demonstrate the coordinate transformations associated with the four DH parameters, that is, joint offset (b), joint angle (θ), link length (a), and twist angle (α) of two neighboring links coupled by a one-degree-of-freedom (DOF) joint, and how they correspond to the physical architecture of the robot. This is a fundamental concept a beginner must understand or his/her instructor must provide. RoboAnalyzer provides a readymade solution towards that objective. It essentially helps a user to correlate the four DH parameters to the four underlying elementary transformations, which are either a translation or a rotation along or about an axis, respectively. More details of the DH parameters used in RoboAnalyzer are available in Refs. [24,32].

Figure 2 illustrates how two elementary transformations associated to joint offset and twist angle can be visualized in RoboAnalyzer. Figure 2A,B show them, respectively, for a Revolute-Revolute (RR) manipulator. A coordinate frame is drawn in the initial configuration which is then animated by translation and rotation depending on the DH parameter, as indicated in Figure 2A,B, respectively. Simultaneously, the DH parameter is highlighted in the DH parameter table of the user interface shown in Figure 1.

2.2 | Effect of DH parameters on robot architecture

Once the mathematical description of DH parameters has been associated with their underlying coordinate transformations, the next step is to understand its effect on the physical configuration of the robot, that is, how a change in DH parameters can affect the architecture of the robot and



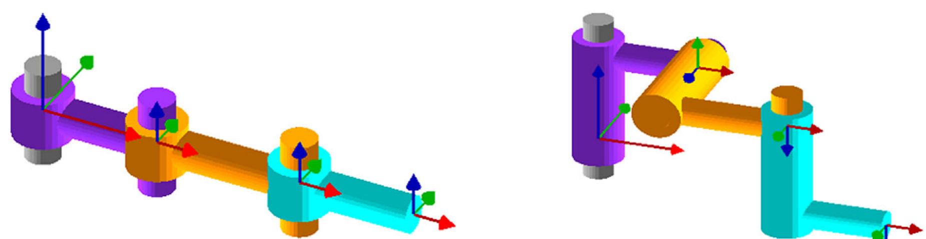
(a) Joint offset as translation (b) Twist angle as rotation

FIGURE 2 Visualization of DH parameters in RoboAnalyzer

vice versa. Explaining and understanding the same in a classroom environment necessitates the use of multiple diagrams. A better approach toward overcoming this hurdle would be to have a 3D visualization environment like RoboAnalyzer using its skeleton models. An actual robot model cannot be used for this purpose, as its DH parameters were decided by its manufacturer.

The proposed software offers an alternate representation of an existing industrial robot, say, KUKA KR5 Arc, using the skeleton models whose kinematic and dynamic analyses results are same, provided the DH and other parameters are entered correctly. It is just the visualization of the link shapes which differ. Otherwise kinematically and dynamically, they are same. One can then vary the link parameters by changing the DH parameters to see the effects on kinematic and dynamic performances. This provides learners a way to explore variations in the given architecture of an industrial robot. It will also help in conceptualizing a new design. Basically, a user can experiment with a multitude of robots, circumventing the hassles of coding, or creating different models in a 3D environment. This capability is illustrated in Figure 3A in which an RRR robot is shown whose motion is restricted to a plane. The corresponding DH parameters are also shown. By varying the DH parameters b and under α, as shown in Figure 3B, a different configuration of the RRR robot can be obtained whose motion is spatial. Similar customization is possible for robots with different DOFs.

In-line with the above discussion, RoboAnalyzer can offer the user with the ability to create robots of different architecture and DOFs. Different industrial/research applications require different robot architectures. For example, SCARA (Selective Compliance Assembly Robot Arm) is preferred for pick and place operations, while a 7-DOF arm allows to reach more number of configurations in the workspace. To facilitate the need to illustrate the concept and to provide better flexibility to robotics researchers, RoboAnalyzer allows the creation of such customized robot models of any DOF and architecture. To a novice user, this would aid his or her imagination of the type of various robot architectures, while an advanced user can use the software to arrive at an optimal robot architecture required for an application. The available feature is shown in Figure 4. The "Add New Robot" window of Figure 4A allows the user to create serial robots of any DOF by specifying the type of joints and corresponding DH parameters. The generated 3D model of the robot is shown in Figure 4B.



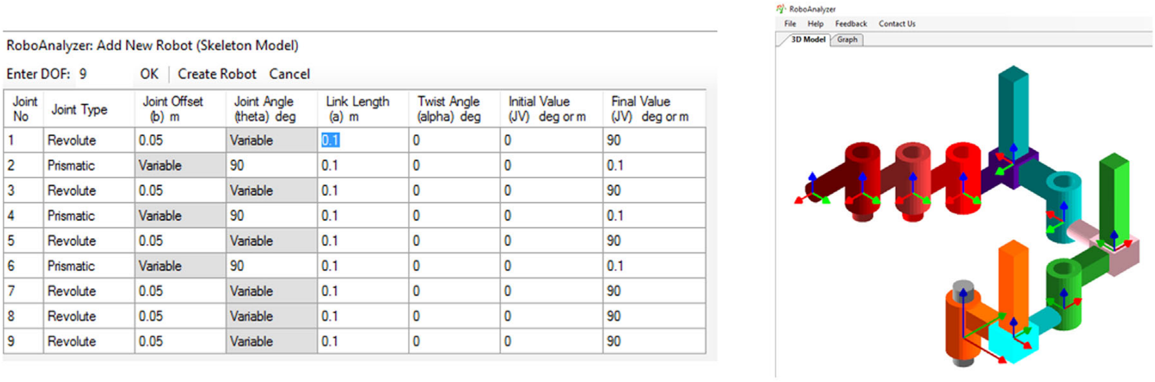
i	$b_i\,(m)$	$\theta_i\,(°)$	$a_i\,(m)$	$\alpha_i\,(°)$
1	0	0	0.1	0
2	0	0	0.125	0
3	0	0	0.1	0

(a) Planar RRR robot

i	$b_i\,(m)$	$\theta_i\,(°)$	$a_i\,(m)$	$\alpha_i\,(°)$
1	0.1	0	0.1	90
2	0.1	0	0.125	90
3	0.1	0	0.1	90

(b) Spatial RRR robot

FIGURE 3 Customizable serial robots in RoboAnalyzer



RoboAnalyzer: Add New Robot (Skeleton Model)

Enter DOF: 9 OK | Create Robot Cancel

Joint No	Joint Type	Joint Offset (b) m	Joint Angle (theta) deg	Link Length (a) m	Twist Angle (alpha) deg	Initial Value (JV) deg or m	Final Value (JV) deg or m
1	Revolute	0.05	Variable	0.1	0	0	90
2	Prismatic	Variable	90	0.1	0	0	0.1
3	Revolute	0.05	Variable	0.1	0	0	90
4	Prismatic	Variable	90	0.1	0	0	0.1
5	Revolute	0.05	Variable	0.1	0	0	90
6	Prismatic	Variable	90	0.1	0	0	0.1
7	Revolute	0.05	Variable	0.1	0	0	90
8	Revolute	0.05	Variable	0.1	0	0	90
9	Revolute	0.05	Variable	0.1	0	0	90

(a) 'Add New Robot' module in RoboAnalyzer

(b) Nine-DOF RPRPRPRRR robot in RoboAnalyzer

FIGURE 4 Creating robots of different architecture

2.3 | Homogenous transformation matrices

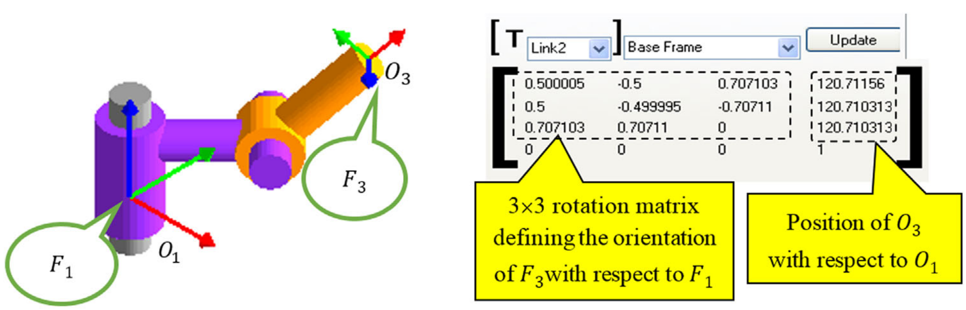
The numeric values of the 4 × 4 Homogeneous Transformation Matrices (HTMs) are required for kinematic and dynamic analyses, as they describe the position and orientation of the robot links. But it is tedious and time-consuming to calculate the HTMs for a pair of DH frames fixed to any two links at every time instant of the robot's movement. The target audience of beginner-level students would find it difficult to do the same for learning and doing assignments. In RoboAnalyzer, the HTMs for different links are available to the user in the GUI. This would allow them to validate their results, especially in a classroom and during practical sessions. The HTM for a particular configuration of an RR robot is shown in Figure 5, where F_1 and F_3 are the frames attached to the base link (Link 0) and the last link (Link 2), respectively.

3 | TEACHING ROBOT KINEMATICS

Robot kinematics involves two fundamental tasks, namely, forward and inverse kinematics. The former deals with the calculation of the pose of a robot's end-effector (EE) from the joint variables, while the latter involves the determination of all possible joint variables corresponding to a given EE pose. Similar to the problems discussed in section 2, there are some inherent difficulties faced by the students and teachers while understanding and teaching the robot kinematics, respectively. They are discussed below.

3.1 | Forward kinematics

In forward kinematics, pose, that is, the position and orientation of the EE of a robot is determined by multiplying the HTMs sequentially. While the numerical calculation is straightforward, the symbolic forward kinematic equations in terms of different joint variables can be lengthy. As a result, understanding the robot motion when all the joint variables change can be difficult. Also, the robot motion can be done in the joint space as well as in EE's Cartesian space. It is important to clarify these points in an introductory course on robot kinematics. Using the "FKin" module of RoboAnalyzer one can perform the forward kinematics analysis and visualize the robot's motion between initial and final joint



(a) Visualization (b) Homogeneous transformation matrix (HTM)

FIGURE 5 Visualization of an HTM between two DH frames configurations without immediately bothering about the overall analytical expressions. The motion of the robot and the change in its EE pose or configuration can be visualized. The animation of forward kinematic analysis of KUKA KR5 Arc robot is shown in Figure 6A. The corresponding plots for the EE position are shown in Figure 6B. It is suggested here that once a student understood the physical behavior of a robot through the animation in RoboAnalyzer, he or she should perform the matrix operations himself or herself for further clarity on the subject.

3.2 | Trajectories for joint motion

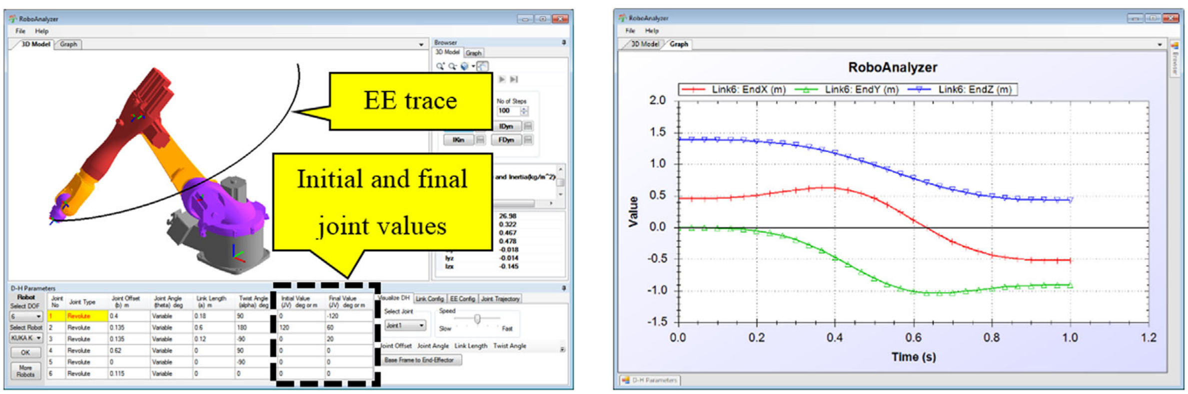
While moving a joint from an initial to a final position within certain time, different joint trajectories can be employed. Each joint trajectory is characterized by its position, velocity, and acceleration. Commonly studied trajectories to provide smooth motion to a robot avoiding jerky motion or vibration are cosine, cubic, quintic, cycloidal, etc. Based on a selected joint trajectory, the robot's joint will move. In general, calculation of a trajectory for each joint is time consuming. RoboAnalyzer provides the user with commonly used trajectories which can be selected from its GUI. After this, the simulation can be performed. This would help a user to quickly compare and demonstrate the effects of different joint trajectories on kinematics of a robot. Advanced users can provide their joint trajectory in a specified file format and use it for performing forward kinematics and animation of the robot's motion. The plots for position, velocity, and acceleration for a revolute joint when it is moved in a cycloidal trajectory from 0° to 30° are shown in Figure 7.

3.3 | Inverse kinematics

Unlike forward kinematics problem which has a unique solution, inverse kinematics problem of a typical industrial robot is not straight forward, mainly, owing to the existence of multiple solutions of the highly non-linear trigonometric functions. While the forward kinematics has a generic procedure for all robot architectures, there is no generic inverse kinematics solution possible that can accommodate all robot architectures. One has to resort to a numerical algorithm for the solution of corresponding kinematic constraint equations. To obtain solutions to the inverse kinematics problem, one is required to solve multiple multivariate transcendental equations. Sometimes no solution may exist for a given input pose. Such aspects make the topic of inverse kinematics relatively difficult in an introductory course on robotics.

The inverse kinematics module of RoboAnalyzer [2] was designed to tackle the above issues. The closed-form solutions of the inverse kinematics problem of several commonly discussed robots in a textbook were implemented. The users can supply the position and orientation of the EE in the form of the Homogeneous Transformation Matrix (HTM) containing 3×3 rotation matrix and the 3-dimensional end-effector position, and then obtain all possible solutions, if they exist. The solver will supply multiple solutions, after which each solution can be visualized in the 3D model viewer. This would help the user in appreciating various solutions for a single EE pose. The focus here is not the computational efficiency of the inverse kinematics solution, but the ease with which the solutions can be obtained and visualized. Using these, students can verify their results. The inverse kinematics solutions obtained for a given EE configuration of the planar RRR robot of Figure 3A is shown in Figure 8A.

It also has the functionality to perform the joint motion between any two of the solutions. This helps a user to confirm that the same pose of the EE can be achieved using different joint configurations. Visualization of multiple solutions is a unique feature of the inverse kinematics module of RoboAnalyzer. The



(a) Animation of the robot

(b) Plots of EE's position

FIGURE 6 Forward kinematics of KUKA KR5 Arc in RoboAnalyzer

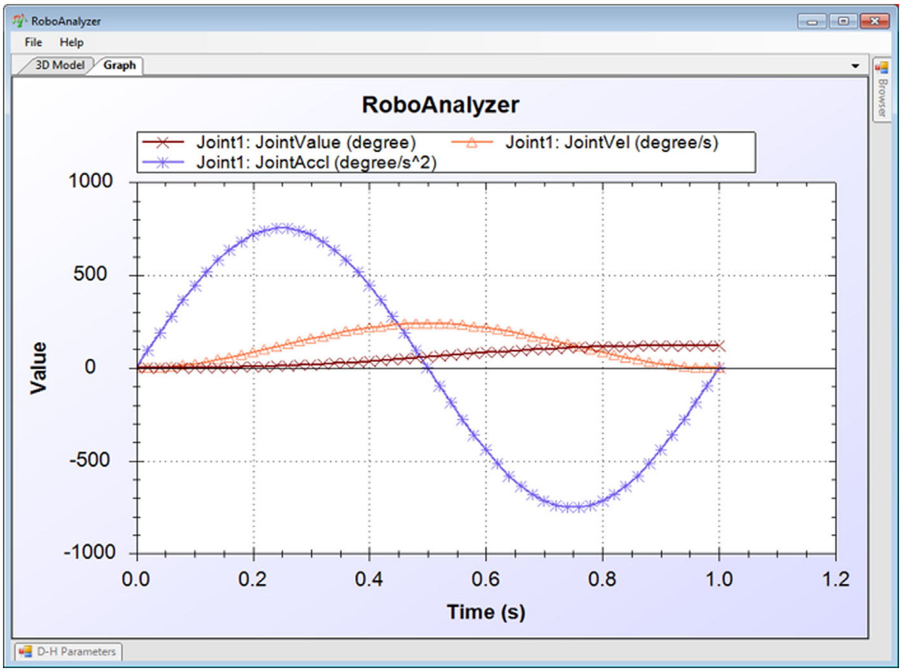


FIGURE 7 Cycloidal trajectory for the position of a joint and its corresponding velocity and acceleration

two possible solutions for a planar RRR robot as obtained in RoboAnalyzer are shown in Figures 8B,C.

3.4 | Motion in joint and cartesian spaces

A robot can be controlled in joint or in cartesian space. The difference between them should be thoroughly understood by the students to program an industrial robot. The Virtual Robot Module (VRM) of RoboAnalyzer [29] was designed to mimic a teach-pendant of a commercially available industrial robot. It has over 17 CAD models of commonly used industrial robots. The teach-pendant-like interface can be used to jog the robot at joint-level or at cartesian-level, and can be used to emphasize the differences between the two types of motion inputs. The joint control panel of the VRM shown in Figure 9A allows the user to move the robot in the joint space by varying each joint angle. The robot can also be moved in the Cartesian space along the X, Y, or Z axes. The orientation can be changed by modifying the Euler angles (A, B, and C). The Cartesian control panel of the VRM is shown in Figure 9B. The user can move the robot relative to its EE frame or with respect to the world frame. These features help students to understand different concepts used in robot motion and programming, even in the absence of an actual robot. The VRM also provides the user to teach a trajectory to a robot either in joint space or cartesian space, record it and then

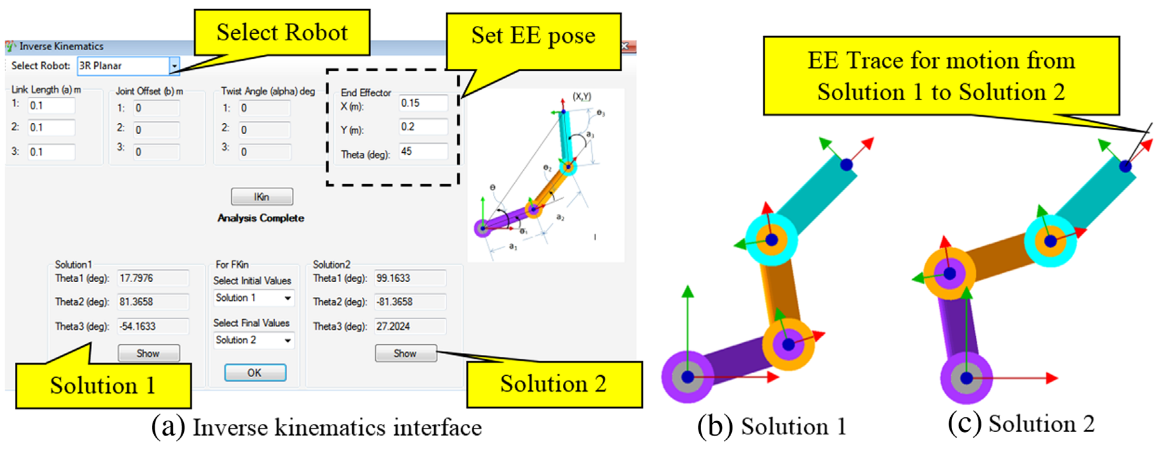
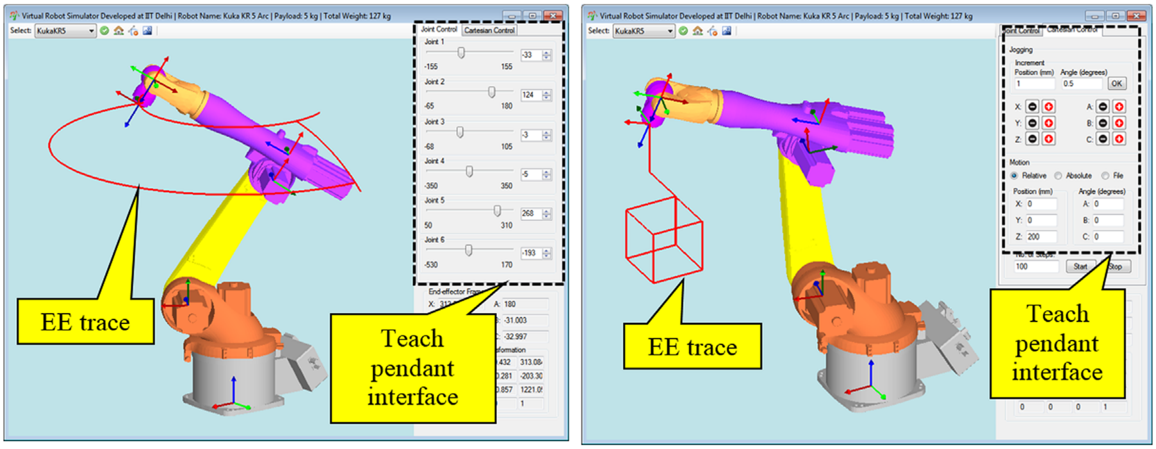


FIGURE 8 Inverse kinematics of the RRR robot of Figure 3A



(a) Jogging the robot in joint space (b) Motion in Cartesian space

FIGURE 9 Joint and cartesian motions of the robot in Virtual Robots Module (VRM)

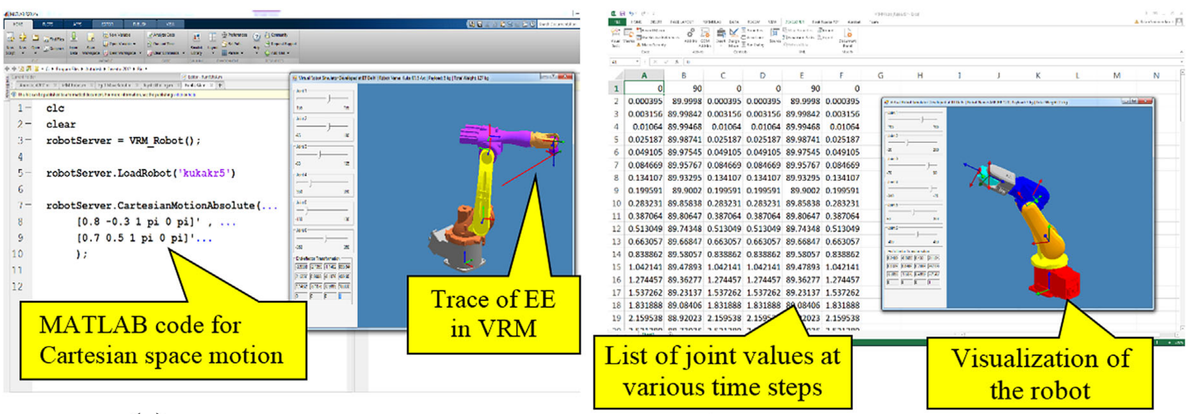
playback later. The authors believe that this would help in introducing offline robot programming to a beginner, in a virtual environment. The VRM was also developed to act as a Microsoft COM (Component Object Model) server which can be used from applications such as MATLAB and Microsoft Excel [30]. In addition, it has been interfaced with Robotics Toolbox [4] to act as a good visualization tool for the analyses performed in the MATLAB-based functions. The screenshots of VRM COM server are shown in Figure 10.

4 | ADVANTAGES OF ROBOANALYZER

The most obvious advantages of RoboAnalyzer are as follows:

1) It allows quick demonstration and simulation of a serial robot.
2) It can be easily employed to demonstrate the concepts of DH parameters, coordinate transformations, HTMs, etc.
3) Since programming and the knowledge of CAD are not prerequisites for using RoboAnalyzer, it helps students to get started almost instantly. When students do not have access to an actual robot, near realistic motion simulations and animation can be performed using RoboAnalyzer and the results can be verified.

The features of RoboAnalyzer discussed in sections 2 and 3 help in learning and teaching of robots, along with any standard textbook in Robotics. It has been used by one of the authors at his institute for creating a structured practical coursework for the course "Mechanics of Robots," where students were asked to perform virtual experiments using RoboAnalyzer and validate their results with data from an actual robot. Various workshops and training programs were also conducted to familiarize students and faculty with the use of RoboAnalyzer for robotics education in different



(a) Interface with MATLAB (b) Interface with Microsoft Excel

FIGURE 10 Virtual Robots Module (VRM) as a COM server

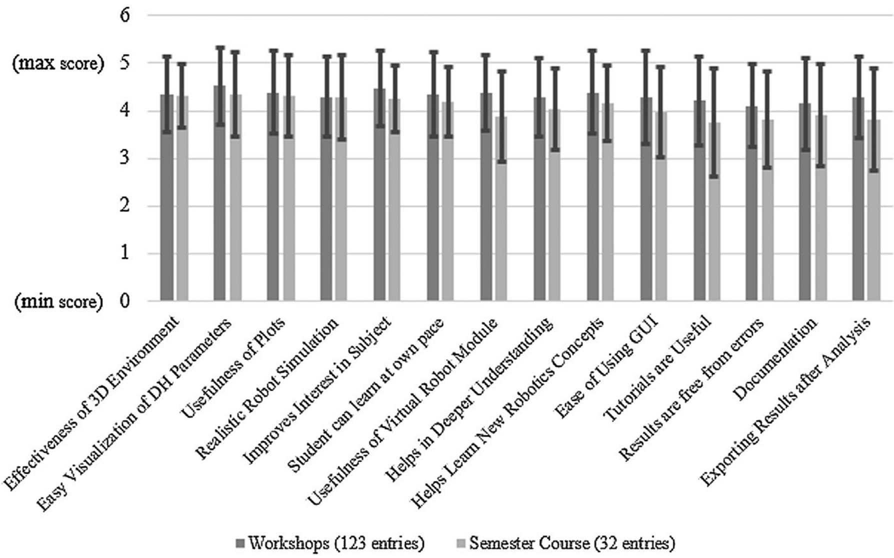


FIGURE 11 User feedback results of RoboAnalyzer

parts of India. An introductory robotics course with integrated virtual experiments using RoboAnalyzer was proposed in Ref. [31].

Based on the feedback received from students and teachers who used RoboAnalyzer in academia, the salient features of RoboAnalyzer have been rated on a scale of 5. The results are shown in Figure 11 with the mean and standard deviation. The user feedback also suggests that the feature of visualizing DH parameters is the second most accepted one among users. It is to be noted here that this a unique feature of RoboAnalyzer and has not yet been reported in any other robotics teaching software. This was also not discussed in detail by the authors in their earlier publications.

Although RoboAnalyzer is good for teaching and learning the concepts of robotics, compared to other software like Robotics Toolbox [4] and VREP [10], it currently does not provide the flexibility and programming environment to perform advanced analyses. This is because the original objective of the RoboAnalyzer was to assist a student to learn robot mechanics in a fun and independent way outside the classroom at his or her pace. It also aids a teacher to make the subject interesting by showing robot animations and the effects of kinematic parameters on a robot architecture almost instantly. It is currently available only on Windows platform, while VREP [10], Robotics Toolbox [4], and RoKiSim [27] are available for Windows, Linux, and MAC operating systems. To overcome some of the shortcomings, continuous efforts are being made to make RoboAnalyzer a better software to teach and learn robotics. Considering the above discussed advantages and shortcomings, the authors believe that RoboAnalyzer is a good tool for beginners.

5 | CONCLUSIONS

Teaching and learning robotics involve advanced level concepts from mathematics to mechanics to trajectory planning and control. Explaining them in a classroom environment may be challenging. A teaching software like RoboAnalyzer presents an effective way to overcome most of the above challenges. It has different modules for visualization, kinematics, dynamics, and plotting, which can help a student to correlate the physics of a robot to the mathematics involved. The strongest feature of the software and the main contribution of this paper is the visualization of the DH parameters and their effective use in relating the input-output motion characteristics of a given robot. RoboAnalyzer software can be downloaded free from http://www.roboanalyzer.com, where its video demos are also available.

ACKNOWLEDGMENTS

The authors would like to thank QIP (Quality Improvement Programme) and CD (Curriculum Development) Cell at IIT Delhi for several partial financial supports toward the development of RoboAnalyzer. The software is a combined effort of many students and researchers whose names are listed on RoboAnalyzer's website. Comments received from several people which helped to improve the software are also acknowledged in the software's website.

REFERENCES

1. ADEFID – available online at: http://adefid.com. Accessed June 2016.

2. J. Bahuguna, R. G. Chittawadigi, S. K. Saha, *Teaching and Learning of Robot Kinematics Using RoboAnalyzer Software,* 1st Int. Conf. Advances in Robotics, 2013.
3. F. A. Candelas, S.T. Puente, F. Torres, F. G. Ortiz, P. Gil, J. Pomares, *A virtual laboratory for teaching robotics*, Int J. Eng. Educ., **19** (2003), 363–370.
4. P. Corke, *A robotics toolbox for MATLAB*, IEEE Robot. Autom. Mag., **3** (1996), 24–32.
5. P. Corke, *MATLAB toolboxes: robotics and vision for students and teachers*, IEEE Robot. Autom. Mag., **14** (2007), 16–17.
6. J. J. Craig, *Introduction to Robotics: mechanics and Control*, 3rd ed., Pearson India, New Delhi, 2005.
7. E. E. Danahy, A. Goswamy, C. B. Rogers, *Future of robotics education: the design and creation of interactive notebooks for teaching robotics concepts,* IEEE Int. Conf. Technologies for Practical Robot Applicat., 2008, 131–136.
8. J. Denavit, R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, *Trans. ASME J. Appl. Mechan.*, 1955, 215–221.
9. M. Flanders, R. Kavanaugh, *Build-A-Robot: using virtual reality to visualize the Denavit– Hartenberg parameters*, Comput. Appl. Eng. Educ., **23** (2015), 846–853.
10. M. Freese, S. Singh, F. Ozaki, N. Matsuhira, Virtual robot experimentation platform V-REP: a versatile 3D robot simulator, *Int. Conf. Simulation, Modeling, and Programming for Autonomous Robots.* 2010, 51–62.
11. A. Gil, O. Reinoso, J. M. Marin, L. Paya, J. Ruiz, *Development and deployment of a new robotics toolbox for education*, Comput. Appl. Eng. Educ., **23** (2015), 443–454. https://doi.org/10.1002/cae.21615
12. M. A. Gonzalez-Palacios, *Advanced engineering platform for industrial development*, J. Appl. Res. Tech., **10** (2012), 309–326.
13. M. A. Gonzalez-Palacios, E. A. Gonzalez-Barbosa, L. A. Aguilera-Cortes, *SnAM: a simulation software on serial manipulators*, Eng. Comput., **29** (2013), 87–94.
14. S. Ivaldi, V. Padois, F. Nori, Tools for dynamics simulation of robots: a survey based on user feedback, *arXiv preprint*, arXiv: 1402.7050, 2014.
15. C. A. Jara, F. A. Candelas, J. Pomares, F. Torres, *Java software platform for the development for advanced robotic virtual laboratories*, Comput. Appl. Eng. Educ., **21** (2013), 14–30.
16. S. Kucuk, Z. Bingul, *An off-line robot simulation toolbox*, Comput. Appl. Eng. Educ., **18** (2010), 41–52.
17. A. Lobov, J. L.M. Lastra, R. Tuokko, A collaborative framework for learning robot mechanics: RIO—robotics illustrative software, ASEE/IEEE Frontiers in Educ. Conf., 2003, 12–16.
18. G. T McKee, *The development of Internet-based laboratory environments for teaching robotics and artificial intelligence*, IEEE Int Conf Robot Autom. **3** (2002), 2695–2700.
19. O. Michel, *Webots: professional mobile robot simulation*, Int J. Adv. Robot. Syst., **1** (2004), 39–42.
20. H. D. Nayar, *Robotect: serial-link manipulator design software for modeling, visualization and performance analysis,* 7th Int. Conf. Control, Autom., Robot. and Vision, 2002, 1360–1364.
21. Nethery, M.W. Spong, *Robotica: a Mathematica package for robot analysis*, IEEE Robot. Autom. Mag., **1** (1994), 13–20.
22. Navaraja, N. Jain, D. Sengupta, C. S. Kumar, *Web based simulation and remote triggered laboratory for robots*, 28th International Conference on CAD/CAM, Robotics and Factories of the Future, 2016, 665–677.
23. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, Vol. 3, No. 3.2, 2009.
24. C. G. Rajeevlochana, S. K. Saha, RoboAnalyzer: 3D model based robotic learning software, *Int. Conf. Multi Body Dynamics*, 2011, 3–13.
25. M. F. Robinette, R. Manseur, *ROBOT-DRAW, an internet-based visualization tool for robotics education*, IEEE Trans. Educ., **44** (2001), 29–34.
26. RoboDK – available at: http://www.robodk.com. Accessed in June 2016.
27. RoKiSim – available at: http://www.parallemic.org/RoKiSim.html. Accessed in June 2016.
28. ROS – Gazebo Plugin, available at: http://wiki.ros.org/gazebo. Accessed on June 2016.
29. R. Sadanand, R. G. Chittawadigi, S. K. Saha, *Virtual robot simulation in RoboAnalyzer,* 1st Int. and 16th Nat. Conf. Mach. and Mechanisms, 2013.
30. R. Sadanand, R. G. Chittawadigi, R. P. Joshi, S. K. Saha, Virtual robots module: an effective visualization tool for robotics toolbox, *2nd Int. Conf. Advances in Robotics*, 2015.
31. R. Sadanand, R. P. Joshi, R. G. Chittawadigi, S. K. Saha, *Virtual experiments for integrated teaching and learning of robot mechanics using RoboAnalyzer,* 28th International Conference on CADCAM, Robotics and Factories of the Future (CARS-FOF2016), Kolaghat, India, 2016.
32. S. K. Saha, *Introduction to Robotics*, 2nd ed., McGraw-Hill Higher Education, New Delhi, 2014.

R. S. OTHAYOTH received his BTech degree in Mechanical Engineering from National Institute of Technology Calicut, India in 2014. Thereafter, he worked as a research assistant on robot simulation, with the Programme for Autonomous Robotics (PAR) Lab at Indian Institute of Technology Delhi. He is currently a graduate student at the Johns Hopkins University. His research interests are robotics, control systems, bio-inspired robots, and system modeling.

R. G. CHITTAWADIGI received his BTech degree in Mechanical Engineering from Motilal Nehru National Institute of Technology, Allahabad, India in 2006 and MS (Research) from Indian Institute of Technology Delhi, New Delhi, India. He has industry and research experience from Hero Honda Motors Limited, Gurgaon, AR-CAD.com, IIT Delhi, and 3dPLM Software Solutions Limited. He is currently a faculty in the Mechanical Engineering Department at Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Bangalore, India. His research interests are robotics, mechanisms, railway dynamics, and CAD.

R. P. Joshi received his BTech degree from IIITDM Jabalpur, India in 2012. Thereafter, he worked in Altair Engineering, Bangalore, India. Later, he moved to Indian Institute of Technology Delhi, New Delhi, India, where he worked as senior research fellow in the Department of Mechanical Engineering until 2015. Currently, he is a master's degree student in the Kyushu Institute of Technology, Fukuoka, Japan. His research interests are assistive robotics, human-robot interaction, and machine learning.

S. K. Saha, a 1983 Mechanical Engineering graduate from the RE College (Now NIT), Durgapur, India, completed his MTech from IIT Kharagpur, India in 1985, and PhD from McGill University, Canada in 1991. Upon completion of his PhD, he joined Toshiba Corporation's R&D Center in Japan. Since 1996, he has been with Indian Institute of Technology Delhi, New Delhi, India, where he is currently a professor in the Mechanical Engineering Department. He was awarded the Humboldt Fellowship in 1999 by the AvH Foundation, Germany. His research interests are robotics, dynamics, mechanisms and also the usage of these in rural areas.

How to cite this article: Othayoth RS, Chittawadigi RG, Joshi RP, Saha SK. Robot kinematics made easy using RoboAnalyzer software. *Comput Appl Eng Educ*. 2017;25:669–680. https://doi.org/10.1002/cae.21828