

Teaching and Learning of Robot Kinematics Using RoboAnalyzer Software

Jyoti Bahuguna
Chassis System PVBU, ERC
TATA Motors Ltd.
Pimpri, Pune-411018, India
91-7276468739
bahugunajyoti.iitd@gmail.com

Rajeevlochana G. Chittawadigi, Subir K. Saha
Department of Mechanical Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi-110016, India
91-1126581135
rajeevlochan.iitd@gmail.com,
saha@mech.iitd.ac.in

ABSTRACT

Robots are used at various places for different applications and hence the subjects related to robotics find their place in the courses of Mechanical and Electrical engineering disciplines. The concepts of robotics are typically difficult to understand from images and figures, thus several software to aid the learning of these concepts have been developed. RoboAnalyzer is one such software developed by the authors to perform kinematic and dynamic analyses of serial robots. It is an ongoing activity and in this paper, the modules of “Visualization of DH Parameters and Transformations”, “3D CAD Model Importer” and “Inverse Kinematics” are explained and illustrated. RoboAnalyzer software can be downloaded for free from <http://www.roboanalyzer.com> and can be used almost instantly.

Keywords

Robotics Learning Software; DH Parameters; 3D Robot Models; Inverse Kinematics

1. INTRODUCTION

With the rapidly growing world, demand for automation in industries is shooting high and robots are creating their need in those industries. Robots are being used in several industries such as automobile, electronics, medical healthcare, space etc. Hence learning of robotics is gaining importance in present era. Because of its relevance, robotics has found the place in the curriculum of various institutions all over the world.

Learning robotics in conventional way requires understanding of mathematical equations, transformations from one frame to another, etc. These generally scare a student and consequently student losses interest in robotics. Also it is difficult for a teacher to make student visualize robot movements while teaching the subject. Most of the industrial robots are described geometrically by their Denavit-Hartenberg (DH) parameters [1], which are also difficult to be perceived by the students. Students will find the

subject easier to learn if they are able to visualize in three dimensions. Hence, to make teaching and learning of robotics easy, the need of some aid in the form of a tool or software arises. Nowadays, a number of robotics visualization tools are available, both free as well as priced. A brief overview of few such tools is given in [2]. There are other advanced software developed by Jaramillo-Botero et al. [3] and Khan et al. [4] that cover Forward Dynamics (FDyn), Inverse Dynamics (IDyn) and trajectory planning. RoKiSim [5] has capability to display 3D CAD models of various industrial robots and perform joint-level and Cartesian-level jogging. It also has functionalities to ease the understanding of DH parameters. RobotAssist [6] is a software which helps in learning trajectory planning as well as Inverse Kinematics (IKin). On similar lines, one such robotic learning software named RoboAnalyzer was also developed by the authors [2, 7]. It is a 3D model based robotics learning software that models a serial robot based on its DH parameters. It has the capabilities to perform FKIn, IKin, IDyn and FDyn of a serial robot. As an output, it shows the animation of the analyses results and graph plots. In this paper, modules on “Visualization of DH Parameters and Transformation”, “3D CAD Model Importer” and “Inverse Kinematics” are presented. The first module lets user learn the meaning of DH parameters by visualizing the movement of Cartesian frame on a virtual robot. It also helps in understanding the physical meaning of transformation matrices between the Cartesian frames, referred as DH frames, attached to the robot links. The second module loads 3D CAD models of a few industrial robots such as PUMA 560, Stanford Arm and KUKA KR5, and perform their kinematic and dynamic analyses. “Inverse Kinematics” module can perform inverse kinematics of various robots with different architectures. Visualization of different solutions are provided to understand them in a better way. It also has the capability of providing the user an option of visualizing multiple inverse kinematics solutions. In addition, visualization of how a robot can go from one solution to another is a unique feature of IKin module, not found in the literature so far. This feature has better impact on students to understand that multiple solutions of inverse kinematics lead to the same configuration. The paper is arranged as follows: Section II gives an overview of kinematics formulations emphasizing more on IKin; Section III explains the software architecture of RoboAnalyzer followed by illustrations of new modules in Section IV.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.
Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

2. KINEMATIC FORMULATIONS

Kinematics of a serial robot plays a major role in understanding various concepts of robotics such as dynamics, control, trajectory planning etc. In this section, an overview of robot kinematics, especially IKin is presented, which is one of the recent modules of RoboAnalyzer [8].

2.1 Denavit- Hartenberg Parameters

A serial robot consists of several links connected by, usually, single degree of freedom (DOF) joints. They can be of revolute (R) or prismatic (P) type. In order to control the end-effector with respect to the base, it is necessary to find the relation between the coordinate frames attached to the end-effector and the base. For this, each robot link is attached with a DH frame as shown in Figure 1(a). The rules of attaching the frames are outlined in [9]. Four DH parameters are required to relate two consecutive DH frames as illustrated in Figure 1(b) and defined in Table I. Note that for a revolute joint, parameter θ_i is a variable whereas in case of a prismatic joint, b_i is a variable.

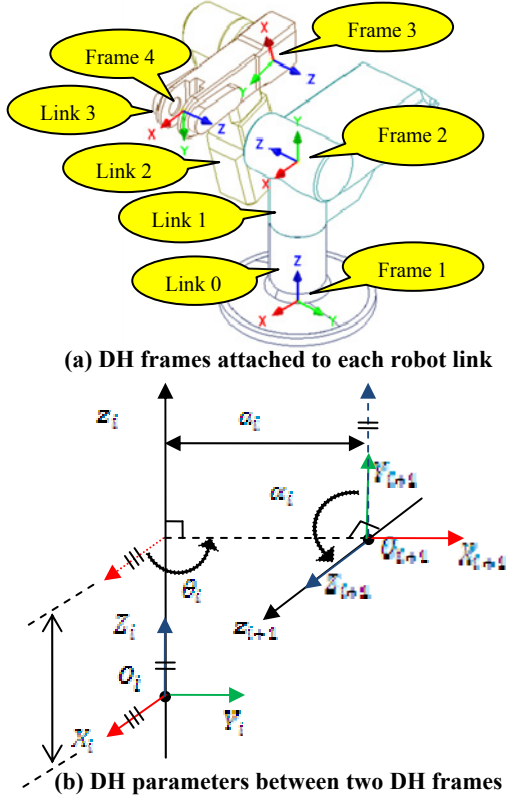


Figure 1. Denavit-Hartenberg (DH) Parameters

Table 1. Descriptions of DH parameters

Parameters	Description
Joint Offset (b_i)	Distance between X_i and X_{i+1} along Z_i
Joint Angle (θ_i)	Angle between X_i and X_{i+1} about Z_i
Link Length (a_i)	Distance between Z_i and Z_{i+1} along X_{i+1}
Twist Angle (α_i)	Angle between Z_i and Z_{i+1} about X_{i+1}

2.2 Forward Kinematics

In forward or direct kinematics, the joint positions, i.e., the angles of the revolute joints and the displacements of the prismatic joints, are prescribed. The task is to find the end-effector's pose, i.e., its position and orientation. This can be obtained from the closure equations as explained in the following steps:

- For a serial manipulator with n degree of freedom, attach a coordinate frame as per the rules outlined in [9] to each of the $n+1$ links of the robot with Frame 1 being attached to the fixed or base frame, and Frame $n+1$ to the end of the n^{th} link.
- Define the DH parameters.
- Write the homogeneous transformation matrices as $T_1, T_2, \dots, T_i, \dots, T_n$, where T_i , for $i=1, \dots, n$, represents the transformation of the Frame $i+1$ with respect to Frame i . Transformation matrix T_i is expressed as [9]
$$T_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$
- The homogeneous transformation matrix of the end-effector frame with respect to the Frame 1, i.e., T , is now obtained by the post multiplication of the above individual transformation matrices, T_i , for $i=1, \dots, n$, i.e.,

$$T = T_1 T_2 T_3 \dots T_n \quad (2)$$

which is known as the closure equation of the robot at hand.

If the expression for the homogeneous transformation matrix T_i of (1) is substituted in (2), two distinct relations in terms of the orientation of the bodies or links and the positions of the origins of the frames attached to the links are obtained. They are

$$Q = Q_1 Q_2 Q_3 \dots Q_n \quad (3)$$

$$p = a_1 + Q_1 a_2 + \dots + Q_1 \dots Q_{n-1} a_n \quad (4)$$

where Q_i is the orthogonal rotation matrix which represents the orientation of Frame $i+1$ with respect to the Frame i . and vector a_i is the vector from the origin of Frame i to the origin of Frame $i+1$. The orthogonal transformation matrix is nothing but the left 3x3 sub-matrix of T_i defined in (1), i.e.,

$$Q_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (5)$$

Moreover, Q is the orientation of the end-effector with respect to the fixed frame namely, Frame 1. Vector p is the position of the origin of the frame attached to the end-effector, i.e., Frame $n+1$, from the origin of Frame 1.

In forward position analysis, the right hand sides of (3) and (4) are given as inputs and the left hand sides are calculated as outputs.

2.3 Inverse Kinematics

Inverse kinematics problem can be stated as follows: finding a set of joint variables to achieve a desired configuration of the end-effector which is usually known from the task specification. While the position and orientation of end-effector is computed in

a unique manner once the joint variables are known (forward kinematics), the inverse kinematics problem is more complex. The equations to be solved are generally nonlinear. Thus it is not always possible to find a closed-form solution. Furthermore, multiple solutions may exist or no solution exist if the desired end-effector configuration lies outside the workspace. Even if multiple solutions exist, none may be admissible for the given architecture of the manipulator. Another problem lies in the visualization of these solutions.

One approach to solve the inverse kinematics problem is to find out a closed-form solution using algebra and geometry. In the geometric approach, joint angles are found by using geometrical heuristics to take advantage of the special structure of the robot. It is sometimes advantageous to use both the approaches together to solve the inverse kinematics problem. Another approach is to find a numerical solution by some successive approximation algorithm. Although the former approach is generally more desirable in applying the solutions to real time control of the robots, it is not always possible to obtain the closed-form solutions for the manipulators with arbitrary architectures. Rather, the class of manipulators for which the closed-form solutions are guaranteed is limited. Here, two such cases are considered.

2.3.1 Inverse Kinematics of 3R Planar Arm

Consider the arm shown in Figure 2. Here it is desired to find the joint angles, θ_1 , θ_2 and θ_3 , corresponding to a given end-effector's position and orientation. For planar motion, the position and orientation of the end-effector can be specified by the origin of Frame 4, i.e., (p_x, p_y) , and the orientation of the frame attached to the end-effector with respect to the X_1 -axis, i.e., angle ϕ , as indicated in Figure 2. Hence, they are specified as the inputs. The solutions are then obtained using an analytical approach given next. From the forward position kinematics analysis of the planar three-link arm, i.e., (3) and (4), one can write

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (6)$$

$$p_x = a_1 c_1 + a_2 c_{12} + a_3 c_{123} \quad (7)$$

$$p_y = a_1 s_1 + a_2 s_{12} + a_3 s_{123} \quad (8)$$

where $c_1 \equiv \cos \theta_1$, $s_1 \equiv \sin \theta_1$, $c_{12} \equiv \cos(\theta_1 + \theta_2)$, $s_{12} \equiv \sin(\theta_1 + \theta_2)$, and similarly c_{123} , s_{123} are defined. Note that these equations are nonlinear in joint angles, θ_1 , θ_2 and θ_3 . The inverse kinematics problem is simplified by subdividing the task, i.e., the end-effector is oriented first before it is positioned. Position of the wrist, W can be known from the position inputs of the end-effector i.e., point P and are given below:

$$w_x = p_x - a_3 c \phi = a_1 c_1 + a_2 c_{12} \quad (9)$$

$$w_y = p_y - a_3 s \phi = a_1 s_1 + a_2 s_{12} \quad (10)$$

Now these two equations can be used to solve the above stated nonlinear equations (6), (7) and (8) to get θ_1 , θ_2 and θ_3 .

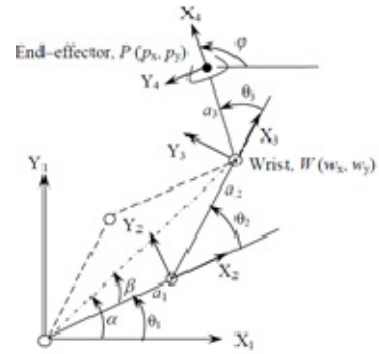


Figure 2.3R planar robot

2.3.2 Inverse kinematics of spatial 6R decoupled manipulator

Industrial robots usually have a special architecture that allows the decoupling of the positioning problem from the orientation problem. Inverse kinematics solution of such six-axis serial decoupled robot can be obtained by coupling the solutions of the 3R position problem and 3R orientation problem. Detailed derivation for the position as well as orientation problem has been discussed in [8, 10].

Let C denote the intersection of axes 4, 5, and 6, i.e., the centre of the spherical wrist, as shown in Figure 3. Let c denote the position vector of this point. Note that the position of C is independent of joint angles, θ_4 , θ_5 and θ_6 . Hence, only the first three joints are to be considered for the position analysis [10], i.e.,

$$c = p - Q Q_6^T a_6 \equiv p - Q b_6 \quad (11)$$

where b_6 is same as a_6 for revolute pairs and

$$Q \equiv Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 \quad (12)$$

As p and Q are input to the inverse kinematics problem and b_6 (joint offset) is known from the robotic architecture, one can find vector c . The positioning problem now becomes finding the first three joint angles necessary to position point C . Equation (11) is a vector equation containing three scalar equations in joint variables θ_1 , θ_2 and θ_3 only. Hence, first three joint variables can be known from the solution of the position problem.

Now the remaining joint angles can be obtained by solving the orientation problem. Since θ_1 , θ_2 and θ_3 are known, Q_1 , Q_2 and Q_3 can be evaluated. Hence, the orientation problem given by (12) can be rewritten in the following form

$$Q_4 Q_5 Q_6 = R \quad (13)$$

where R is defined as

$$R = Q_3^T Q_2^T Q_1^T Q \quad (14)$$

Note that e_i represents unit vector along the axis of Joint i . Moreover, the orientation of the end-effector is given, i.e., the components of vector e_6 are known. Let e_6 in Frame 4 be denoted as

$$[e_6]_4 = [\xi \quad \eta \quad \psi]^T \quad (15)$$

where ξ , η and ψ are the components of e_6 in Frame 4.

Similarly, the components of vector e_5 in Frame 4 are nothing but the entries of the third column of matrix Q_4 , i.e.,

$$[\mathbf{e}_5]_4 = [\mu_4 \sin \theta_4 \quad -\mu_4 \cos \theta_4 \quad \lambda_4]^T \quad (16)$$

in which $\mu_4 = \cos \alpha_4$ and $\lambda_4 = \sin \alpha_4$

Furthermore, vectors \mathbf{e}_5 and \mathbf{e}_6 make an angle α_5 and hence,

$$\mathbf{e}_6^T \mathbf{e}_5 = \lambda_5 \quad (17)$$

where $\lambda_5 = \cos \alpha_5$. Upon substitution of (15) and (16) into (17), one finds two values of joint variable θ_4 . Once θ_4 is known, one can derive \mathbf{Q}_4 . Furthermore expressions for θ_5 and θ_6 can be readily derived by solving first for \mathbf{Q}_5 as

$$\mathbf{Q}_5 = \mathbf{Q}_4^T \mathbf{R} \mathbf{Q}_6^T \quad (18)$$

Now, by virtue of the form of the \mathbf{Q}_i matrices, it is apparent that the third row of \mathbf{Q}_i does not contain θ_i . Hence, the third column of the matrix product of (18) is independent of θ_6 . Thus, two equations for θ_5 are obtained by equating the first two components of the third columns of that equation, hereby, obtaining a unique value of θ_5 for every value of θ_4 . Finally, with θ_4 and θ_5 known, it is a simple matter to calculate remaining joint angle, θ_6 . This is done upon solving for \mathbf{Q}_6 from the following equation

$$\mathbf{Q}_6 = \mathbf{Q}_5^T \mathbf{Q}_4^T \mathbf{R} \quad (19)$$

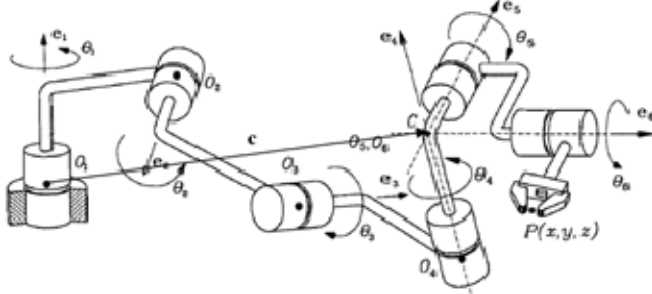


Figure 3. A General 6R manipulator with decoupled architecture [10]

3. ABOUT ROBOANALYZER

RoboAnalyzer software [2, 7] has been developed using the concepts of Object Oriented Modeling (OOM) in Visual C# programming language. For the 3D graphics rendering, OpenGL is used through Tao Framework [11]. For graph plots, an open source library named ZedGraph [12] is used. The software has been developed in modules so that addition and modification of modules do not affect the whole software. The module of Forward Kinematics (FKin) of serial robots with revolute joints was reported in [2]. This uses skeleton models. The analysis results were viewed in the form of animation and in-built graph plot module. Addition of the prismatic joints, Inverse Dynamics (IDyn) and Forward Dynamics (FDyn) analyses were reported in [7]. Here, further developed modules such as “Visualization of DH Parameters and Transformations”, “3D CAD Model Importer” and “Inverse Kinematics” (IKin) are presented.

The different modules of the software are as illustrated in Figure 4. For a given robot, the FKIn module takes joints trajectory (i.e., initial and final value of each joint variable and type of joint level trajectory) as input and determines the configuration of each robot

link over the simulation time, which is also known as Joint-level motion planning. The simulation results can be visualized by transforming each robot link for each time step in 3D Graphics Viewer. The results can also be viewed as graph plots.

IDyn module determines joint forces required to achieve the given joint trajectories, while mass and inertia properties of link and gravity acting on the robot are known. The motion can be viewed through FKIn module and graphs can be plotted for the results. Similarly, FDyn module determines the joint accelerations for gravity acting on the robot for a free-fall motion. The joint velocities and positional values are found using numerical integrators. The animation of the motion can be viewed through FKIn module and the joint motion can be plotted.

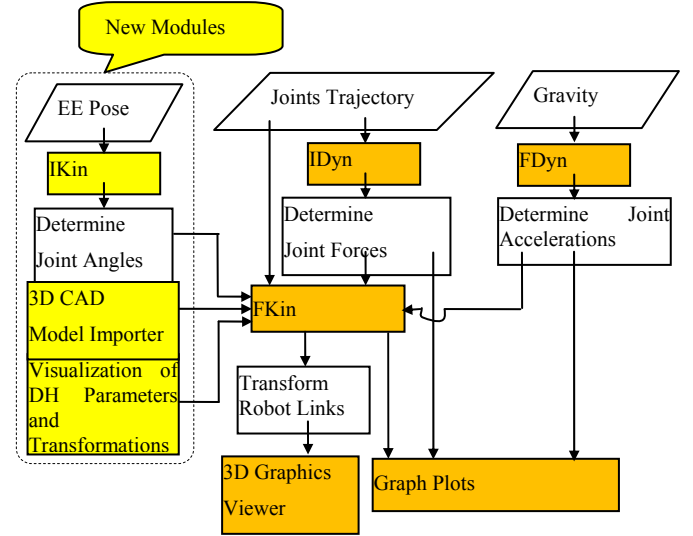


Figure 4. Modules of RoboAnalyzer

4. NEW MODULES IN ROBOANALYZER

In this section, illustrations of the newly added modules are presented.

4.1 Visualization of DH Parameters and Transformations

The DH parameters represent the transformation between two coordinate frames, referred as DH frames, attached to consecutive links of a robot. The usual text-book figures, which are in 2D, do not convey the concepts very well and hence teachers find it difficult to make students understand them. Not many software listed in Section I cover the DH parameters visualization to aid the learning process. The “Visualization of DH Parameters and Transformations” module inside RoboAnalyzer lets the user learn these concepts in an intuitive manner.

For a selected robot, user can vary the values of the DH parameters and see the effect of change in shape of robot links instantly. The four DH parameters can be thought of as translations and rotations of a coordinate frame about one axis at a time while moving from the DH frame attached to a link to the DH frame attached to the next link. A simple interface is designed to let user select a joint and then select any DH parameter for which, a coordinate frame is drawn at the start configuration. Another coordinate frame is translated or rotated and reaches the end configuration for the selected parameter in the form of an

animation. An example for Joint Offset and Twist Angle between Joints 1 and 2 is shown in Figure 5.

The user can select “All” option for a joint which moves the coordinate frame for all the four DH parameters. In addition, as the frame moves, the corresponding parameter in the DH table is also highlighted. One can also use “From Base to End-Effector” option to move the coordinate frame as animation. These features help in better understanding of the DH parameters.

The user can also learn the physical interpretation of the transformation between two selected DH frames.

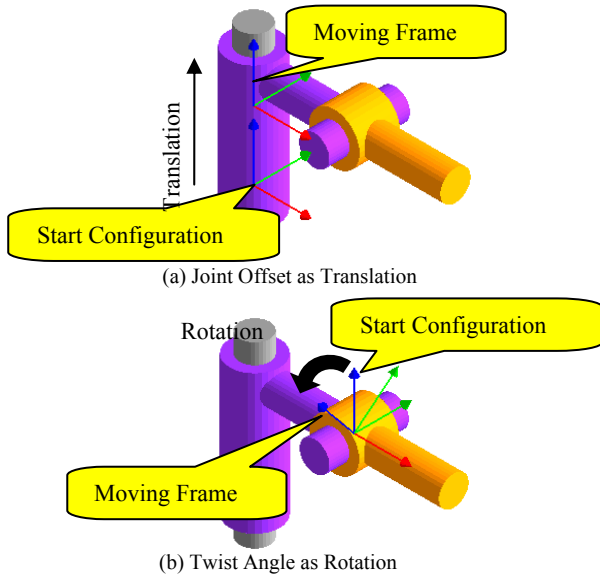


Figure 5. DH Visualization

Visually, the placement of the two frames on the virtual robots can be viewed as shown in Figure 6 (a). The value of the corresponding homogeneous matrix can be viewed as given in Figure 6 (b).

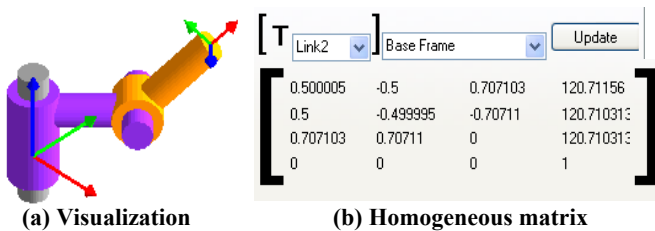


Figure 6. Visualization of transformation between the DH frames

4.2 CAD Model Importer

Robotics learning software such as RoKiSim [5] show 3D CAD model of a selected industrial robot. In RoboAnalyzer, similar module is added. From a CAD assembly model of an industrial robot inside Autodesk Inventor CAD software, its DH parameters were determined analytically using a methodology proposed in [13] and the DH frames were also attached to each robot link. Each link, modeled as a part file, was modified such that its origin frame coincides with the DH frame attached to it and exported as a Stereolithography (STL) file.

In “3D CAD Model Importer” module of RoboAnalyzer, STL CAD files of robots such as KUKA KR5 were imported. Each link was transformed and placed in the 3D environment using the DH parameters and the transformations arising between them, resulting in robot as shown in Figure 7. This makes the process of transforming the robot links easy and efficient.

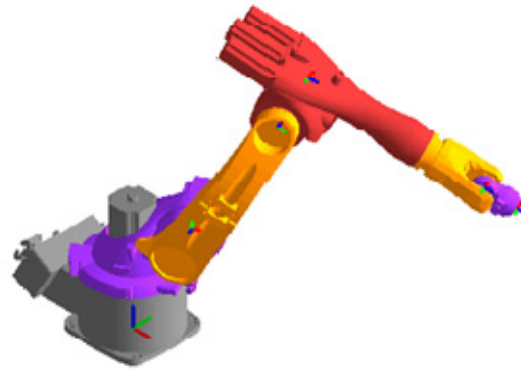


Figure 7. 3D CAD model of KUKA KR5 robot in RoboAnalyzer

4.3 Inverse Kinematics

For a given robot, the inverse kinematics module requires the pose, i.e., position and orientation, of the frame attached to the end-effector as input. As explained in Section II (C), multiple solutions may exist for the required end-effector pose. Similar to other concepts, visualization and understanding of existence of multiple solutions is not conveyed properly by figures in text-books. IKin module aids in its understanding.

IKin module opens in a separate window in which user can select a type of robot. Depending upon the DOF and architecture of the selected robot, specific inputs are to be supplied. For example, for a 3R planar robot as shown in Figure 8, one can obtain two possible solutions as values of joint angles θ_1 , θ_2 and θ_3 . When “Solution 1” is selected, the joint angles and the input parameters a_1 , a_2 and a_3 are sent to FKIn module, which visualizes the configuration of the three links, as shown in Figure 9 (a). Similarly, “Solution 2” corresponds to the display of robot links as shown in Figure 9 (b).

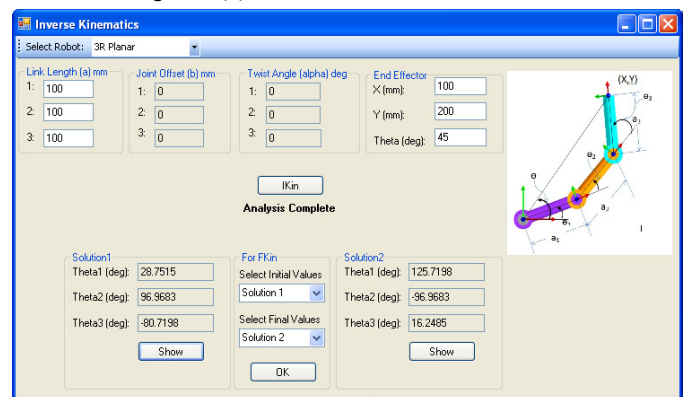


Figure 8. Inverse Kinematics (IKin) module of RoboAnalyzer

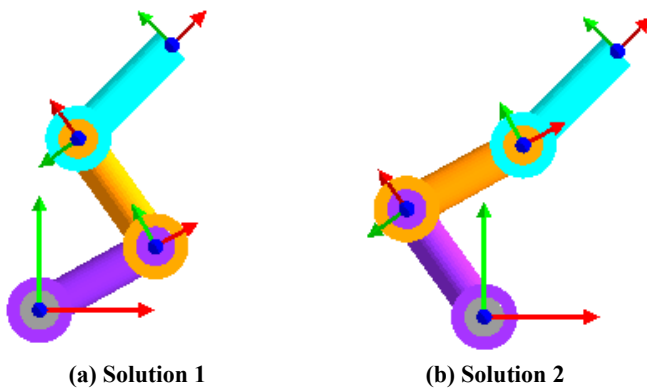


Figure 9. Inverse Kinematics solutions for 3R planar robot

A complex example of KUKA KR5 robot is now considered. It requires the transformation matrix corresponding to the end-effector frame with respect to the base frame as input. As it is a decoupled robot-arm with 6 DOF, a total of 8 solutions are possible [8, 9]. Selecting any solution, IKin module sends the joint angles of the corresponding solution to the FKIn module, which shows 3D model of KUKA KR5 robot in the configuration corresponding to the solution.

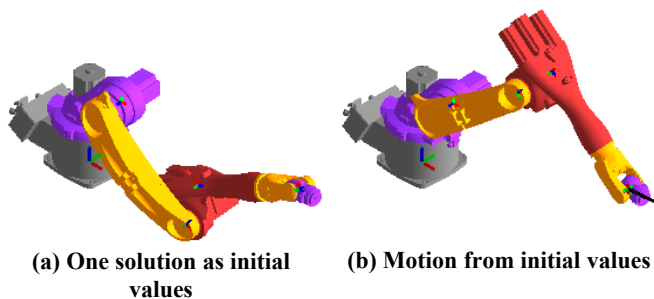


Figure 10. Motion between two IKin solutions of KUKA KR5 robot

In addition, the user can select any solution corresponding to “Initial Values”, given in Figure 10(a), of joint angles and any other solution for “Final Values”. IKin module sends these initial and final values of angles to the FKIn module, which then lets the user perform joint-level motion planning from initial values to final values as illustrated in Figure 10(b). As the robot end-effector configuration is same in initial and final values, the user gets a better understanding of the multiple solutions of IKin.

5. CONCLUSION

RoboAnalyzer is a software developed to ease the teaching and learning of the concepts related to subject of robotics, as these are typically difficult to understand from figures in a text-book. Earlier versions of RoboAnalyzer were reported to have forward kinematics, inverse and forward dynamics for serial robots with revolute and prismatic joints. In this paper, new modules are reported. A module on “Visualization of DH Parameters and Transformation” lets user learn the concepts of DH parameters by visualizing the movement of Cartesian frame on a virtual robot. It

also lets user learn the meaning of homogeneous transformation matrix between two DH frames in the form of display of the frames on a robot and their numerical values. The “3D CAD Model Importer” reads STL files of robot links and displays them in the 3D Cartesian space. The concepts of inverse kinematics, existence of multiple solutions and their visualization and animation from one solution to another are covered in “Inverse Kinematics” module. The readers are encouraged to download the RoboAnalyzer software from <http://www.roboanalyzer.com> and run “RoboAnalyzer6.exe” program to execute the examples given in this paper.

6. REFERENCES

- [1] Denavit, J and Hartenberg, R. S. 1955. A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *ASME Journal of Applied Mechanisms* 22, 2, 215–221.
- [2] Rajeevlochana, C. G. and S. K. Saha, 2011. RoboAnalyzer: 3D model based robotic learning software. In *Proceedings of the International Conference on Multi Body Dynamics* (Vijayawada, India, Feb 12-14, 2011), 3-13.
- [3] Jaramillo-Botero, A., Matta-Gomez, A., Correa-Caicedo, J. F. and Perea-Castro, W. 2006. ROBOMOSP, *IEEE Magazine on Robotics and Automation* 13, 4 (Dec. 2006), 62–73.
- [4] Khan, W., Zhuang, H. and Angeles, J. 2007. Robot Visualization System for Windows (RVS4W). *Centre for Intelligent Machines (CIM), User's Manual*, Dept. Mech. Eng., McGill University, Montreal, Canada.
- [5] RokiSim - <http://www.parallemic.org/RoKiSim.html>.
- [6] RobotAssist - <http://www.kinematics.com/products/robotassist.php>.
- [7] Rajeevlochana, C. G., Jain, A., Shah, S. V. and Saha, S. K. 2011. Recursive Robot Dynamics in RoboAnalyzer. In *Proceedings of the 15th National Conference on Machines and Mechanisms* (Chennai, India, Nov. 30-Dec. 2, 2011), 482-490.
- [8] Bahuguna, J. 2012, *Kinematics of Industrial Robots*, M. Tech Project Report, Dept. Mech. Eng, Indian Inst. of Tech. Delhi, India.
- [9] Saha, S. K. 2008. *Introduction to Robotics*, Tata McGraw Hill, New Delhi, India.
- [10] Angeles, J. 1997, *Fundamentals of Robotic Mechanical Systems. Theory, Method and Algorithms*, Springer.
- [11] Tao Framework, <http://www.sourceforge.net/projects/taoframework>.
- [12] ZedGraph Library, <http://www.sourceforge.net/projects/zedgraph>.
- [13] Rajeevlochana, C. G., Saha, S. K. and Kumar, S. 2012, Automatic Extraction of DH Parameters of Serial Manipulators using Line Geometry. In *Proceedings of the 2nd Joint Int. Conf. on Multibody System Dynamics*(Stuttgart, Germany, May 31-June 3, 2012).