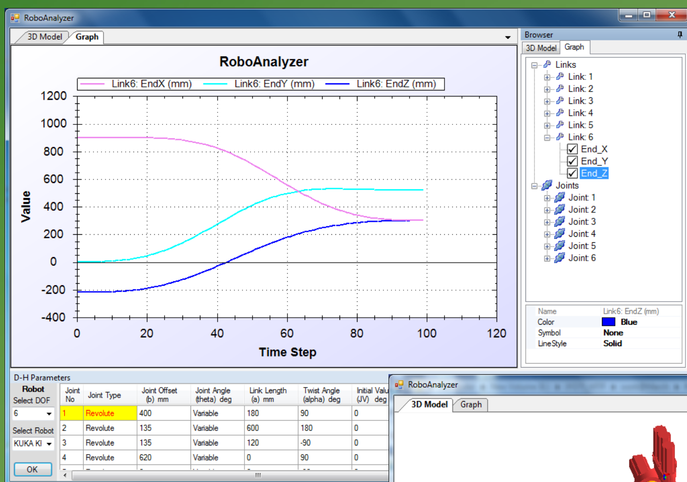
RoboAnalyzer

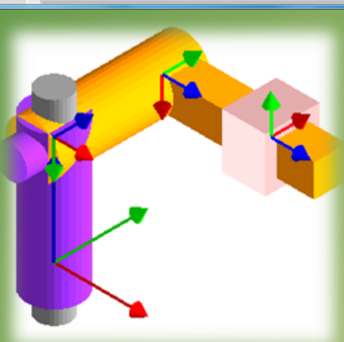
3D Model Based Robotics Learning Software

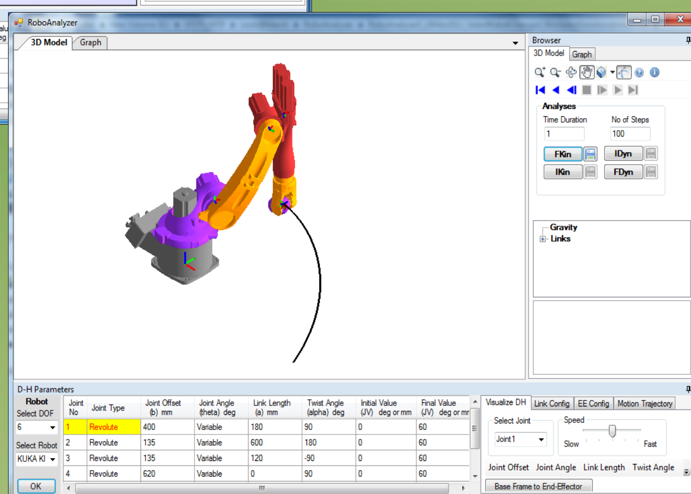
User Manual



Features:

Forward Kinematics
Inverse Kinematics
Forward Dynamics*
Inverse Dynamics*
Animation and Plots
Virtual Robot Module
Integration with MATLAB





Freely Available for Academic Use !!!

September 2014

Developed by Prof S. K. Saha & Team
Mechatronics Lab, Mechanical Engineering Department, IIT Delhi, New Delhi, India

Courtesy: CD Cell, QIP, IIT Delhi

http://www.roboanalyzer.com

* Uses **ReDySim** Dynamic Formulation (Appendix A)

PREFACE

Robotics is a field related to the design, development, control and application of robots in industry, education, research, entertainment, medical applications etc. Since the mathematics involved in the study of robotics, e.g., kinematics and dynamics is initially difficult to understand by students and same is the case by a teacher to convey the essence of mathematics of robotics to the students. Also it has been difficult for students to learn because of limited ability to perceive and visualize the concepts appropriately at the time of teaching. Without seeing a real robot it is very difficult to comprehend its motion in three-dimensional Cartesian space. Hence, there is a need for a robotics learning software.

The work of combining the robot analyses algorithms in the form of software started in 1996 in the name of RIDIM (Recursive Inverse Dynamics for Industrial Manipulator). But it had only analysis part with plot facilities. The development of RoboAnalyzer started in 2009. RoboAnalyzer, a 3D model based software, can be used to teach robotics subjects to undergraduate and postgraduate courses in engineering colleges in India and elsewhere. It can be used to learn DH parameters, kinematics and dynamics of serial robots and allows 3D animation and graph plots as output.In essence, learn/teach the physics of robotics with the joy of RoboAnalyzer animations before attempting to learn the mathematics of robots.

RoboAnalyzer is developed in the Mechatronics Lab, Department of Mechanical Engineering at IIT Delhi, India under the guidance of Prof. S. K. Saha. The following students are given due credits in its development.

- S. Goel and S. Ramakrishnan (1996-97) : Algorithm development for Recursive Inverse Dynamics for Industrial Manipulators (RIDIM)
- A. Patle (2000-01) : Windows-interface for RIDIM
- Rajat Jain (2009-10) : Added Graph-plots to RIDIM
- Suril V Shah (2007-11) : Recursive Dynamics Simulator (ReDySim) Algorithm [Appendix A]
- Rajeevlochana C.G. (2009 - 2013) : User Interface, 3D Modeling of robot, Forward Kinematics, Animation, Graph-plot, DH Visualize, Virtual Robot Module, Integration with MATLAB
- Amit Jain (2010-11) : C# implementation of DeNOC-based Inverse and Forward Dynamics (ReDySim)
- Jyoti Bahuguna (2011-12) : Inverse Kinematics Module and Motion Planning
- Ratan Sadanand O.M. (2012-14): Conversion of CAD models for Virtual Robot Modules, Version 7 Enhancements
- Ravi Joshi (2014): Version 7 Enhancements

CONTENTS

1. GETTING STARTED

This section helps you get started with the installation of RoboAnalyzer, a 3D Model Based Robotics Learning System. It has been developed using OpenTK and Visual C#.

1.1. MINIMUM SYSTEM REQUIREMENT

- Processor: Atleast 1.5 GHz
- RAM: Atleast 512 MB
- Operating System: Windows XP, Windows Vista, Windows 7
- Dependencies: Microsoft .Net 2.0 framework

1.2. INSTALLATION

RoboAnalyzer can be installed on a computer by downloading it from our website. The latest version of the software (Version 7) is available for free at http://www.roboanalyzer.com. The following are the steps to install RoboAnalyzer:

Step 1: Visithttp://www.roboanalyzer.com
Step 2: Click on **Downloads** tab
Step 3: Click on **RoboAnalyzer V7**(or latest version) to download a .zip file
Step 4: A popup window will appear. Select the folder where the file has to be saved and click on **Save**
Step 5: After downloading is complete, unzip RoboAnalyzer7.zip to any folder on your computer. Open the folder RoboAnalyzer7
Step 6: Double-click on RoboAnalyzer.exe to start RoboAnalyzer.

2. INTRODUCTION TO ROBOANALYZER

RoboAnalyzer is a 3D Model Based Robotics Learning Software. It has been developed to help the faculty to teach and students to learn the concepts of Robotics. It also acts as a supporting material for the contents on various robotics topics in text book entitled "Introduction to Robotics", S. K. Saha, 2014 [1].

2.1. MODEL A ROBOT

Double click on RoboAnalyzer.exe starts RoboAnalyzer. By default, it shows a robot model (2-R). Users can select a robot from the options given at the left bottom corner of the application as shown in Figure 1.
After selecting a robot model, users can modify the Denavit-Hartenberg (DH) parameters shown in the tabular form and the robot model updates automatically. A few industrial robots are also listed, when selected shows a 3D CAD model of the robot (Figure 2).

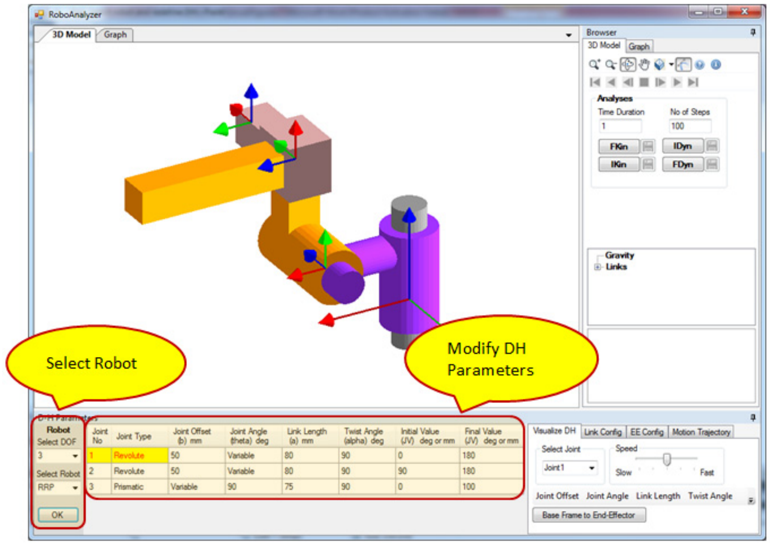


Figure 1: Select Robot Model and Redefine DH Parameters

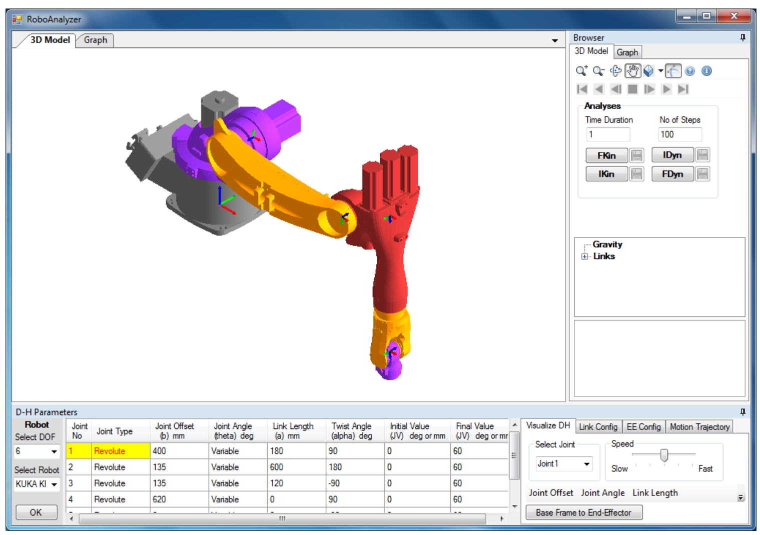


Figure2: 3D CAD Model of Industrial Robot

2.2. FEATURES OF ROBOANALYZER

RoboAnalyzer can be used to perform kinematic and dynamic analyses of serial chain robots/manipulators. The following are the main features of RoboAnalyzer:

- DH Parameter Visualization (Section 3)
- Forward Kinematics (Section 4)
- Inverse Kinematics (Section 5)
- Inverse Dynamics (Section 6)
- Forward Dynamics (Section 7)
- Motion Planning (Section 8)

2.3. OVERVIEW OF USER INTERFACE

RoboAnalyzer's easy to use Graphical User Interface (GUI) consists of the following as shown in Figures 3, 4 and 5.

1. Robot Selection and DH Parameters section
2. Visualize DH section
3. 3D Model Browser
4. 3D Model View
5. Graph Plot Tree View
6. Graph Plot Window
7. Inverse Kinematics Window

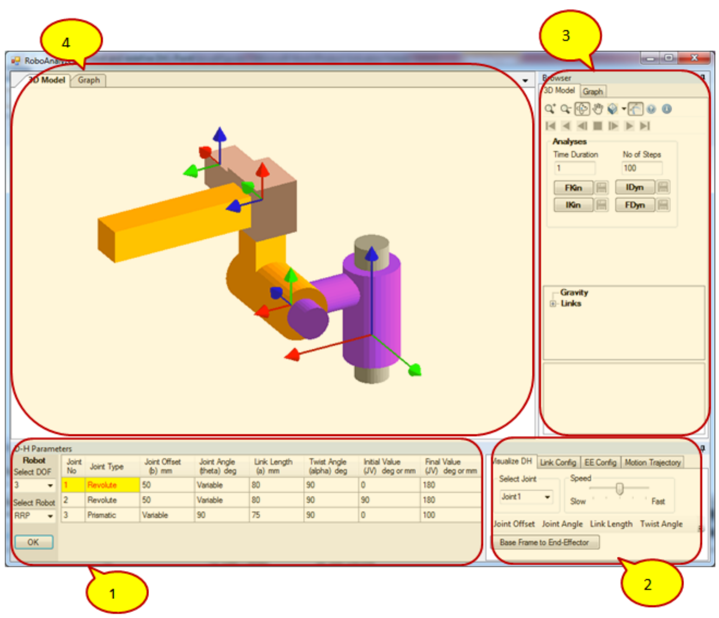


Figure 3: User Interface of 3D Model View

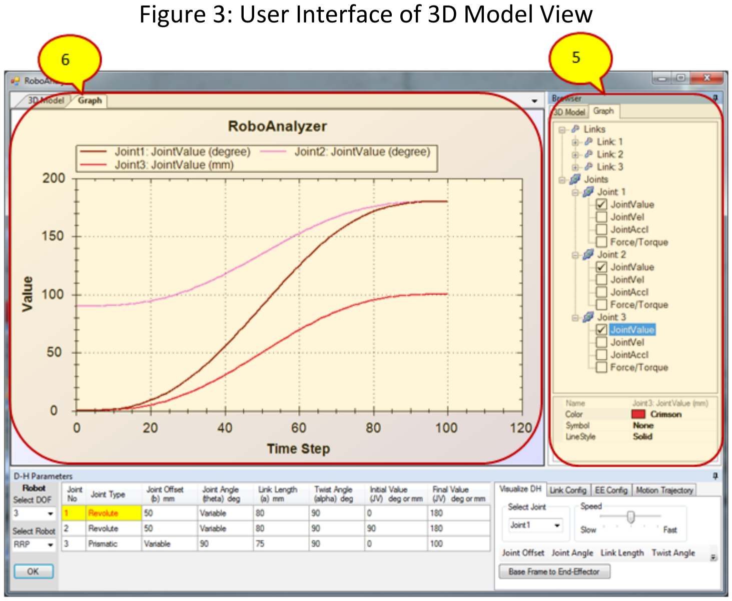


Figure 4: User Interface of Graph Plot View

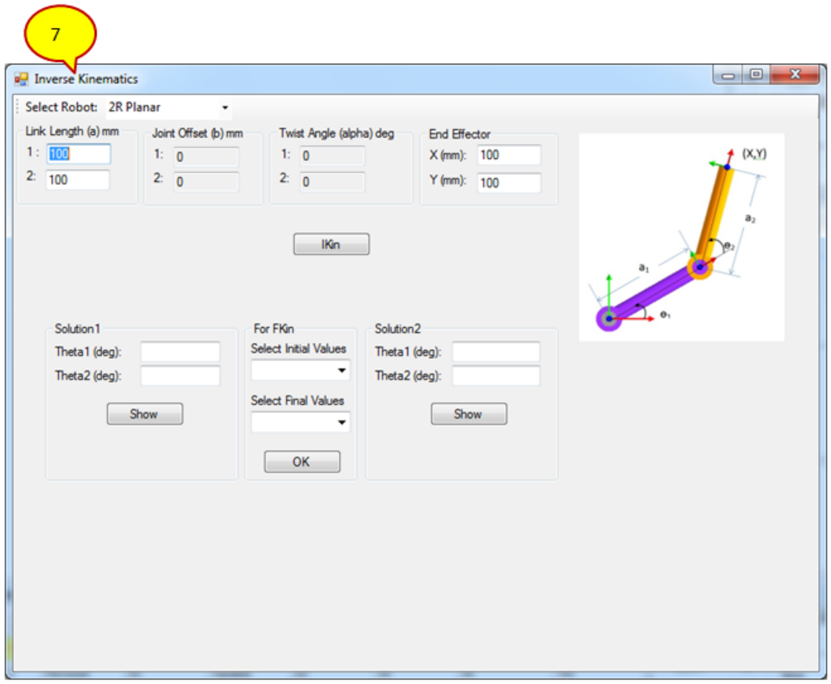


Figure 5: User Interface of Inverse Kinematics Window

2.4. 3D MODEL VIEW OPTIONS

RoboAnalyzer lets the user to zoom, rotate and pan the 3D model to have better visualization. These can be used as explained below and shown in Figure 6.

- **Zoom:** Place the mouse cursor anywhere on 3D Model View and use the mouse-wheel to zoom in and zoom out. It can also be done by clicking on **Zoom In** and **Zoom Out** buttons.
- **Rotate:** Press the right mouse button and drag the mouse cursor anywhere on the 3D Model View to rotate the model in the browser form.
- **Pan:** Press the left mouse button and drag the mouse cursor anywhere on the 3D Model View to pan/translate the model in the browser form.
- **Standard Views:** Select any standard view from the dropdown and the model view updates.

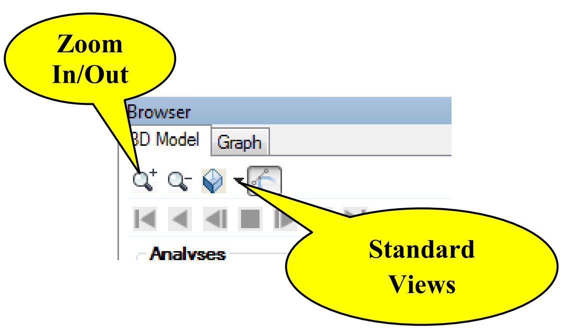


Figure 6: 3D Model View Options

3. DENAVIT-HARTENBERG PARAMETERS VISUALIZATION

The architecture of industrial robots is usually represented by Denavit-Hartenberg (DH) parameters. It forms the basis for performing kinematic and dynamic analyses of robots. A set of four DH parameters is used to represent the position and orientation of a robot-link with respect to its previous link. More details on DH parameters can be found in Chapter 4 of [1].

3.1. VISUALIZE DH

After selecting a robot and redefining DH parameters as explained in Section 2.1, users can visualize each DH parameter by selecting a joint and then selecting a DH parameter type as shown in Figure 7. Once it is done, the corresponding DH parameter is highlighted in the DH parameter input table and a transformation frame moves in the 3D robot model. It shows the two co-ordinate frames corresponding to the selected DH parameter.

Users can click on **Together** button and a co-ordinate frame moves covering all the four DH parameters corresponding to the selected joint.

Users can click on **Base Frame to End-Effector** button to see a co-ordinate frame moving from base frame to end-effector frame covering all the DH parameters of the robot model.

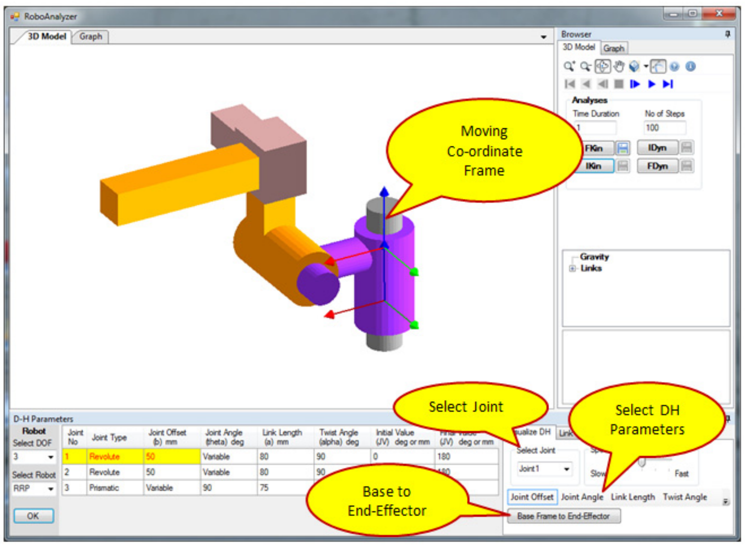


Figure 7: Visualize DH Parameters

3.2. LINK CONFIGURATION

The configuration/ transformation of a co-ordinate frame (DH frame) attached on each robot-link can be determined with respect to a frame attached to its previous link or base frame by following the steps below and as shown in Figure 8.

- Select a joint. If **Joint1** is selected, it corresponds to co-ordinate frame attached on Link1.
- Select **Previous Link Frame** or **Base Frame** as the reference frame with respect to which the transformation needs to be determined.
- Click on **Update** button and 4X4 transformation matrix is populated. A pair of co-ordinate frames is shown in 3D robot model to help user in visualizing the transformation.

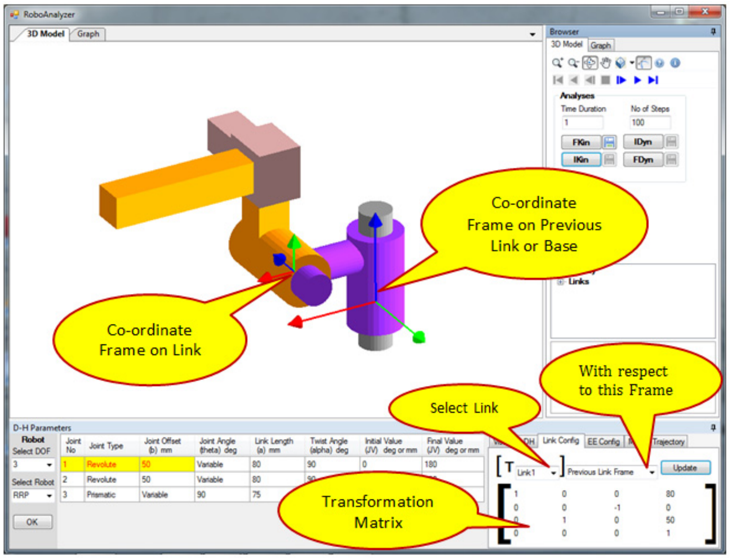


Figure 8: Link Configuration

3.3. END-EFFECTOR CONFIGURATION

The end-effector configuration/transformation can be determined with respect to the base frame directly by using the **Update** button as shown in Figure 9. The 4X4 transformation matrix is populated and a pair of co-ordinate frames is shown in 3D robot model to help user in visualizing the transformation.

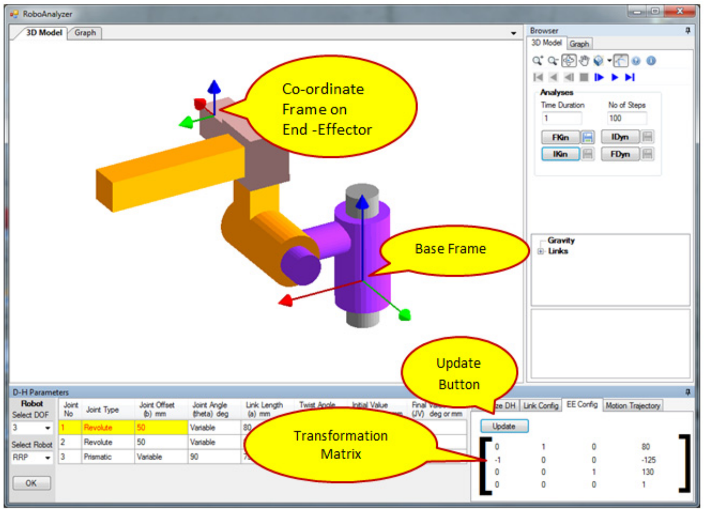


Figure 9: End-Effector Configuration

4. FORWARD KINEMATICS

In the 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 configuration/transformation consisting of its position and orientation. More details can be found in Chapter 6 of [1]. After selecting a robot and redefining DH parameters as explained in Section 2.1, forward kinematics (FKin) is performed which updates the 3D model.

4.1. ANIMATION OF FKIN

To perform animation of the robot motion between two sets of initial and final values of joint variables, the following are the steps as shown in Figures 10 and 11. The trajectory of joint values, joint velocities and joint accelerations follow Cycloidal trajectory mentioned in Chapter 8 of [1]. The trajectory can be changed as explained in Section 8.

1. Set the **initial** and **final** values of joint variables
2. Set **Time Duration** and **Number of Steps**
3. Click on **FKin** button
4. Click on **Play** button to see the animation
5. The end-effector trace can be viewed

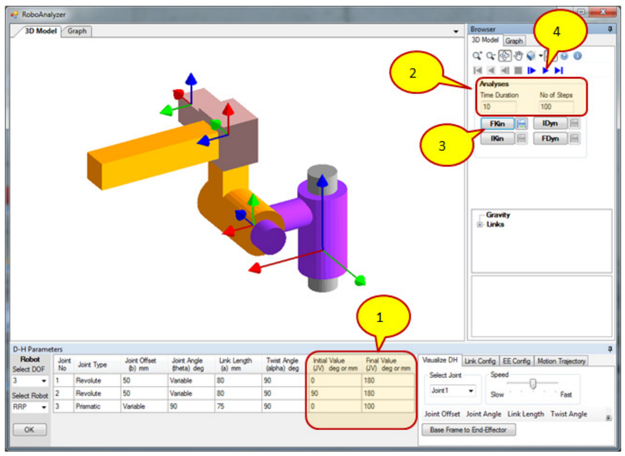


Figure 10: Initial Position of all Joints

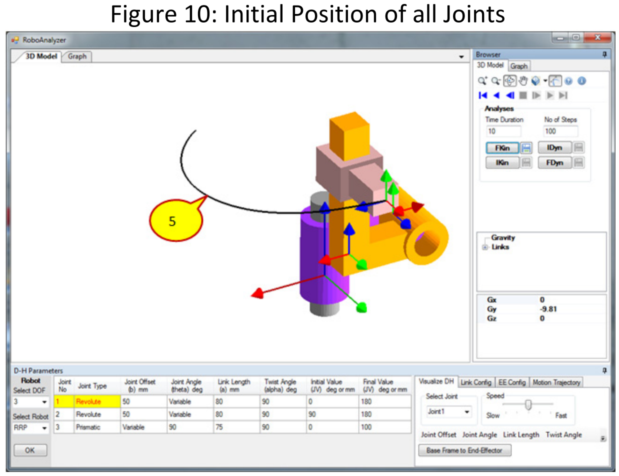


Figure 11: Final Position of all Joints and Trace of End-Effector

4.2. GRAPH PLOTS OF FKIN

To view the graph plots of a forward kinematics (animation) analysis, the following are the steps as shown in Figures 12, 13 and 14.

1. Click on **Graph** tab

2. Click on **+** next to the link of which the plots are to be viewed
3. Click on **box** to plot graph of a particular node to see X, Y and Z plots
4. Click on **+** next to the joint of which the plots are to be viewed
5. Click on **box** to plot graph of a particular node to see joint value (joint angle for revolute joint and joint offset for prismatic joints), joint velocity and joint acceleration

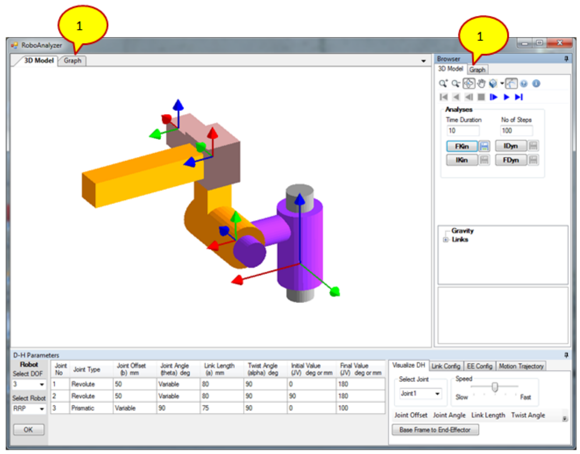


Figure 12: Graph Plots of FKin Data

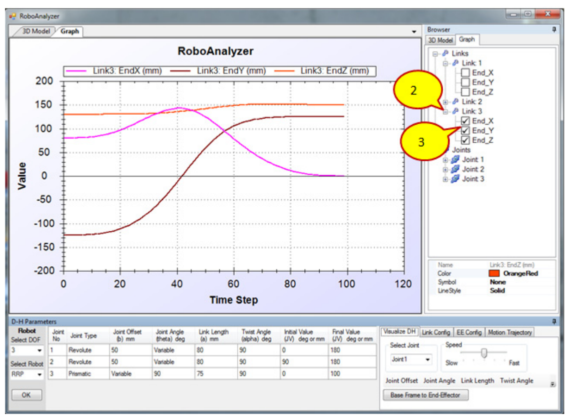


Figure 13: Graph Plots of Position of Coordinate Frame attached to Link3

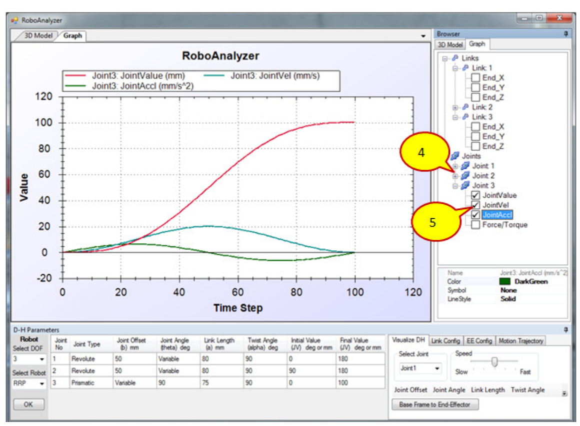


Figure 14: Graph Plots of Input Trajectory to Joint3 (Cycloidal Trajectory)

More details and information on the Graph Plot options can be found in Section 9.

5. INVERSE KINEMATICS

Inverse Kinematics (IKin) consists of determination of the joint variables corresponding to a given end-effector's orientation and position. The solution to this problem is of fundamental importance in order to transform the motion specifications assigned to the end-effector in the operational space into the corresponding joint space motions. There may be multiple or no results possible for a given end-effector position and orientation. More details can be found in Chapter 6 of [1].

5.1. SOLUTIONS OF IKIN

To select a robot and view the solutions of its Inverse Kinematics, the following are the steps as shown in Figures 15 and 16. In future, an IKin solution can be selected and 3D model will be updated accordingly.

1. Click on **IKin** button. It shows a separate window (Figure 16)
2. Select a Robot
3. Enter Input parameters
4. Click on **IKin** button
5. View the possible solutions
6. Click on **Show** button. It shows the selected solution in 3DModel window. To see this go back to main window by minimizing IKin window
7. Select any of the obtained solution as initial and final solution
8. Click on **OK.** This step replaces the initial and final joint values in DH Parameter table
(Main window)by values selected in step 7
9. Click on **FKin**button to view animation i.e. how robot moves from one solution to another solution selected in step 7

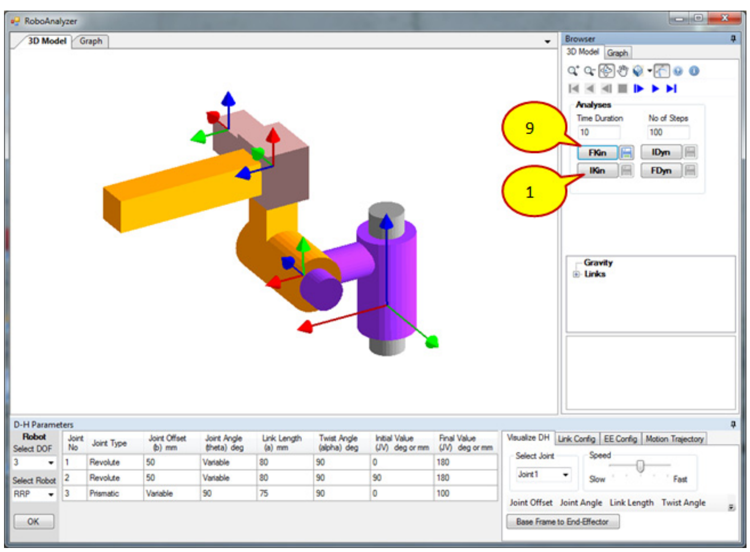


Figure 15: Inverse Kinematics Button

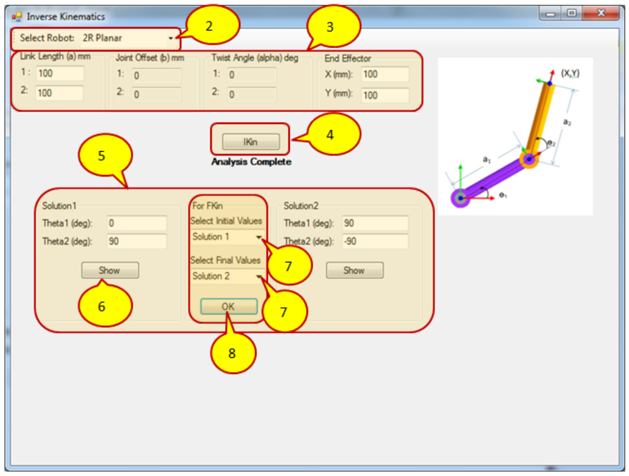


Figure 16: Inverse Kinematics of 3R Articulated Robot

6. INVERSE DYNAMICS

Inverse Dynamics (IDyn) is a dynamics problem, where the robot geometric, inertial parameters, and the joint motions i.e its positions, velocities and acceleration are given and the corresponding joint torques or forces are calculated. In RoboAnalyzer, the dynamics solver is based on ReDySim algorithm, which uses Decoupled Natural Orthogonal Complement (DeNOC) Matrices based recursive formulation. More details on ReDySim can be found at [2]. More details on Inverse Dynamics and DeNOC can be found in Chapters 8 and 9 respectively of [1].

6.1. SOLUTION OF IDYN

Select a robot and redefine DH parameters as explained in Section 2.1, to solve for IDyn of the robot between two sets of initial and final values of joint variables, the following are the steps as shown in Figures 17, 18, and 19. The trajectory of input joint values, joint velocities and joint accelerations follow Cycloidal trajectory mentioned in Chapter 8 of [1]. The trajectory can be changed as explained in Section 8

1. Set the **initial** and **final** values of joint variables
2. Set **Time Duration** and **Number of Steps**
3. Set **Gravity** (all values should be in SI units, i.e. m/s^2)
4. Select a robot-link to enter its Center of Gravity (**CG**) location. It corresponds to a vector from the CG of the robot-link to the origin of the co-ordinate frame attached to that link, measured in the reference of the co-ordinate frame attached to that link.
5. Select **MassProperties** of a robot-link. Set **Mass** of each robot-link (values should be in SI units, i.e. kg) and set **Inertia** tensor of each robot-link with respect to the co-ordinate frame attached at the CG of the robot-link and the co-ordinate frame is parallel to the one attached to the robot-link (values should be in SI units, i.e. kgm^2). These values are to be entered manually and not calculated automatically from the shape of the robot-links.
6. Click on **FKin** button (required to populate the input joint trajectory)
7. Click on **Play** button to see the animation (only for visualization purpose, not necessary for IDyn)
8. Click on **IDyn** button to performInverse Dynamics
9. lick on **Graph** tab to view the graph

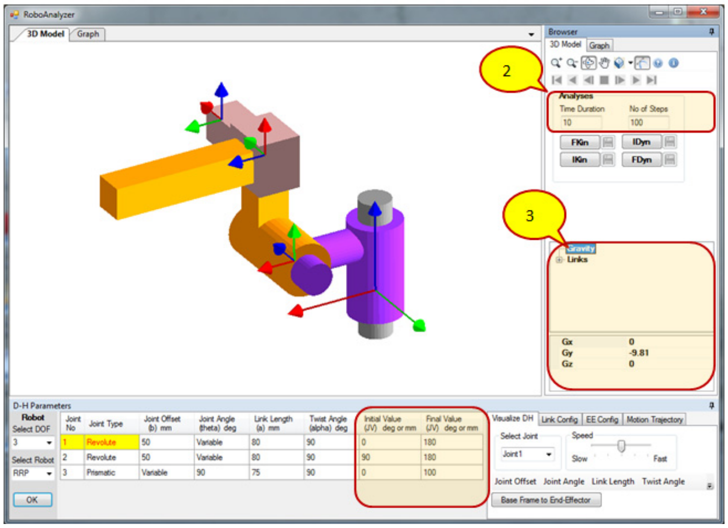


Figure 17: Inverse Dynamics Settings

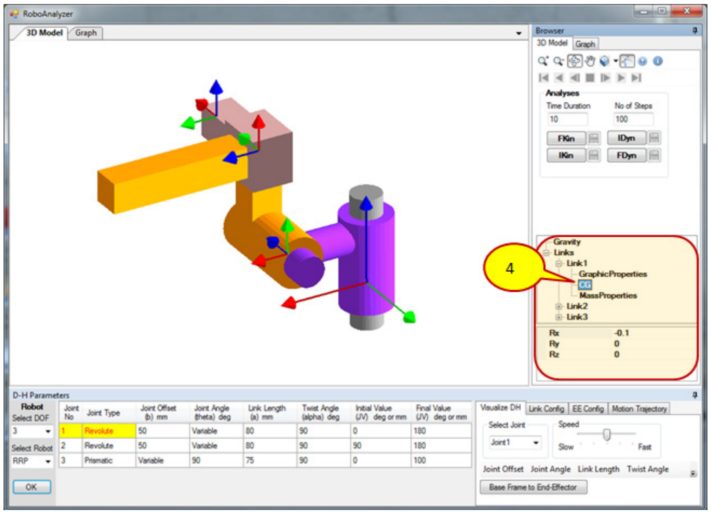


Figure 18: Set Center of Gravity for Inverse Dynamics

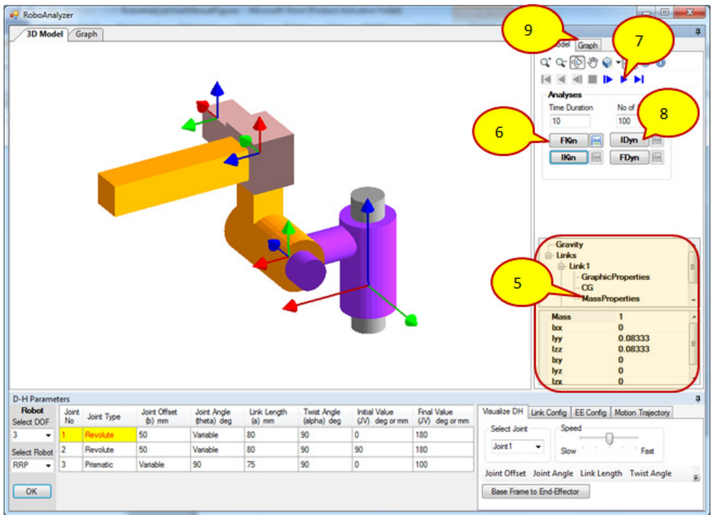


Figure 19: Set Mass and Inertia Properties and Perform Inverse Dynamics

6.2. GRAPH PLOTS OF IDYN

To view the graph plots of joint torques and forces, the following are the steps as shown in Figure 20.

1. Click on **+** next to the joint of which the plots are to be viewed
2. Click on **box** to plot graph of joint torque/force

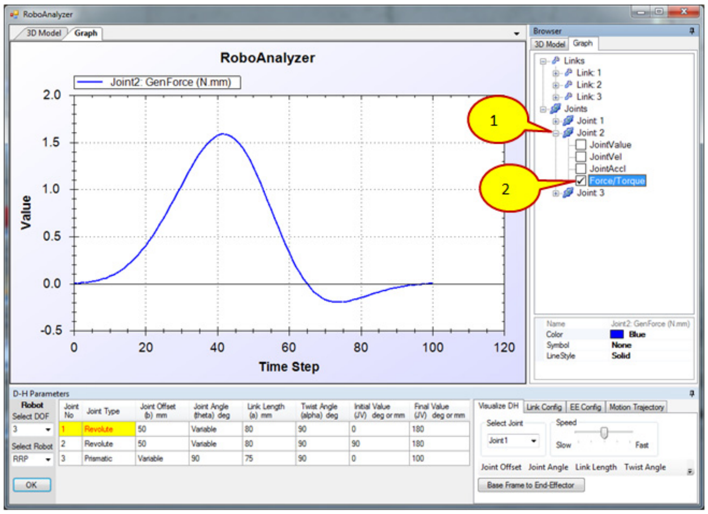


Figure 20: Graph Plot of Joint Torque/Force

More details and information on the Graph Plot options can be found in Section 9.

7. FORWARD DYNAMICS

Forward Dynamics (FDyn) is a dynamics problem, where the robot geometric, inertial parameters, and the joint torques and forces are given and the joint accelerations are calculated. The dynamics solver uses ReDySim[2] as in IDyn. More details on Forward Dynamics can be found in Chapters 8 and 9 of [1].

7.1. SOLUTION OF FDYN

Select a robot and redefine DH parameters as explained in Section 2.1, to solve for FDyn of the robot for a given initial values of joint variables, please refer to Section 6.1 to perform steps 1 to 5 mentioned below. Then perform steps 6, 7 and 8 as shown in Figure 21.

1. Set the **initial** value of joint variables
2. Set **Time Duration** and **Number of Steps**
3. Set **Gravity** (all values should be in SI units, i.e. m/s^2)
4. Select a robot-link to enter its Center of Gravity (**CG**) location. It corresponds to a vector from the CG of the robot-link to the origin of the co-ordinate frame attached to that link, measured in the reference of the co-ordinate frame attached to that link.
5. Select **MassProperties** of a robot-link. Set **Mass** of each robot-link (values should be in SI units, i.e. kg) and set **Inertia** tensor of each robot-link with respect to the co-ordinate frame attached at the CG of the robot-link and the co-ordinate frame is parallel to the one attached to the robot-link (values should be in SI units,

i.e. kgm^2). These values are to be entered manually and not calculated automatically from the shape of the robot-links.

6. Click on **FDyn** button to perform Forward Dynamics. The robot is simulated for free-fall due to the action of gravity. In future, joint torques/forces can be set as input.
7. Click on **Play** button to see the animation
8. Click on **Graph** tab to view the graph

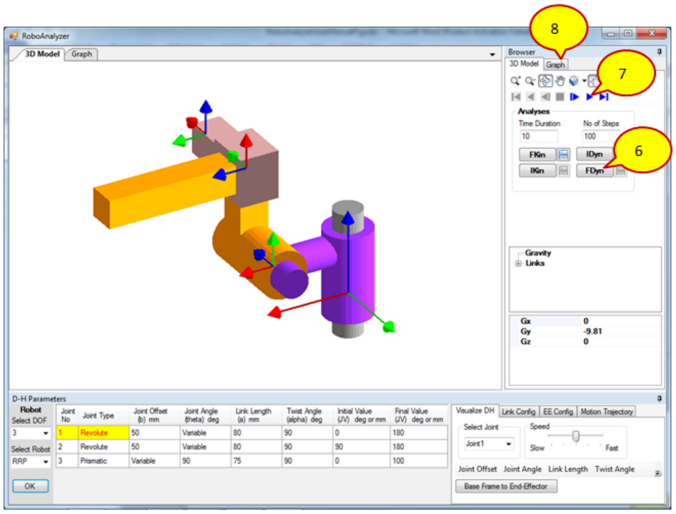


Figure 21: Forward Dynamics

7.2. GRAPH PLOTS OF FDYN

To view the graph plots of joint accelerations and position of links, follow the steps mentioned in Section 4.2.

8. MOTION PLANNING

The goal of motion planning of a robot is to generate a function according to which a robot will move. This function generation depends on the robot tasks. A robot user typically specifies a number of parametersto describe a point-to-point or continuous-path task. Trajectory planning algorithm then generates thereference inputs for the control system of the manipulator, so as to be able to execute the motion. The geometric path, the kinematic and dynamic constraints are the inputs of the trajectory planning algorithm, whereas the trajectory of the joints (or of the end effector), expressed as a time sequence ofposition, velocity and acceleration values, is the output.Trajectory planning can be done either in the joint space, i.e., in terms of joint positions, velocities and accelerations, or Cartesian space (also called operational space) i.e., in terms of the end-effector positions, orientations, and their time derivatives.More details on Motion Planning can be found in Chapter 11 of [1].

8.1. SOLUTION OF MOTION PLANNING

Select a robot and redefine DH parameters as explained in Section 2.1.For a given initial values of joint variables,Motion planning of the selected robot can be performed by selecting particular motion trajectory as shown in Figure 21 followed by steps 1 to 5 mentioned in section 4.1.

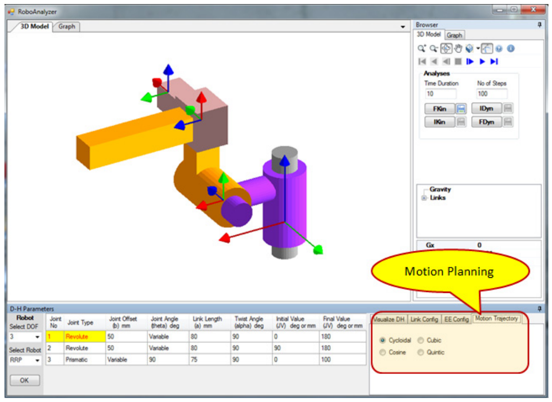


Figure 21: Motion Planning

9. GRAPH PLOT OPTIONS

The analyses results for FKin, IDyn and FDyn can be viewed as graph plots as explained in Sections 4.2, 6.2 and 7.2 respectively. Several options for graph plot functionalites are explained below and as shown in Figure 22 .

1. Select a graph plot node
2. Set the plot color, symbol and line style
3. Right click on graph to show a menu. Here you can use various options to zoom, print etc
4. Export Data as CSV: Export plot data that can be opened in a spreadsheet such as MS Excel
5. Use Mouse wheel to zoom in and out
6. Press Mouse wheel and drag the mouse to pan around the graph

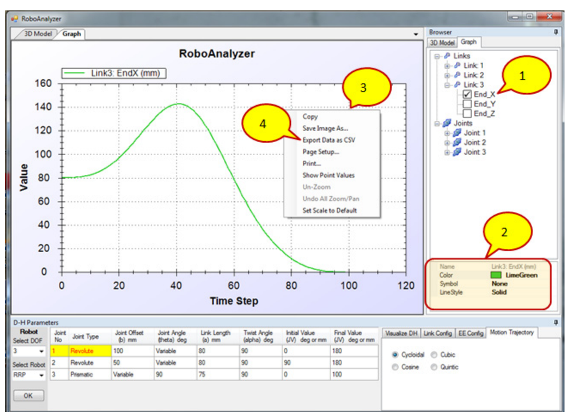


Figure 22: Graph Plot Options

10. SAVING AND OPENING SKELETON MODELS

The skeleton models can be saved after modifying the DH parameters and mass-inertia properties as per the convenience.

1. To save a skeleton model:
   a. Click on ''**File**'' menu
   b. Select the **"Save"** option.
   c. Save the robot in the required directory with a suitable filename. The file extension must be '.xml'.
   d. The Save window can be opened using the shortcut, 'CTRL+S'.

2. To open a previously saved skeleton model:
   a. Click on ''**File''** menu
   b. Select the **"Open"** option.
   c. Navigate to the required directory and open the previously saved model.
   d. The open window can be opened using the shortcut, 'CTRL+O'.

11. VIRTUAL ROBOT MODULE

The Virtual Robot Module inside RoboAnalyzer lets the user select an industrial robot model (CAD Model) and change the joint angles using a slider or using buttons. It can be used as a learning tool to teach joint-level jogging of robots. As of this version, CAD models of 17 industrial robots can be loaded. The following steps are to be followed (as illustrated in Figures 23-24).

1. Click on "More Robots" button.
2. A new window/form is shown. By default CAD model of a robot isdisplayed.
3. Select a robot from the drop-down and click on "Load".
4. Use slider on the left to change each joint angle. Note that all the joint angles have minimum and maximum values as per their specifications (joint limit).
5. Buttons can also be used to change the value of joint angle.
6. The end-effector transformation is updated with every change in joint angle(s).

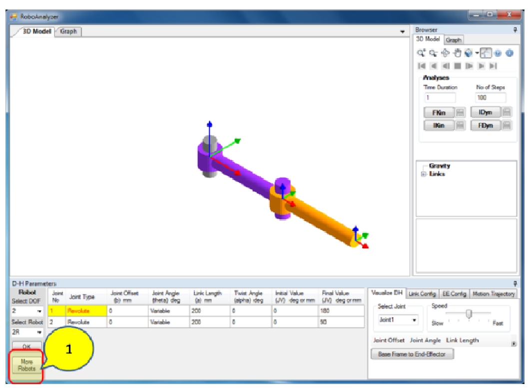


Figure 23: Click on "More Robots" Button in RoboAnalyzer Window

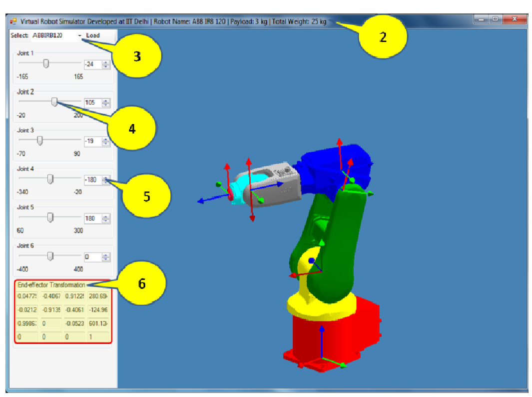


Figure 24: Virtual Robot Module

12. MATLAB INTEGRATION

Virtual Robot Application (explained in Section 10) is made available as a COM interface which can be called and controlled from MATLAB.Other software which can interface with COM objects (such as MS Excel, programs developed in C++ etc.) can also used to control Virtual Robot. At present, only Virtual KUKA KR5 robot can be controlled from MATLAB. The following steps are to be followed to install “VirtualRobotCOM” on a computer:

1. Download "VirtualRobotsCOM.zip" and unzip it in C:\ drive
2. If MATLAB installation is 32 bit version, right click on C:\VirtualRobotsCOM\register32.bat and "**Run as Administrator**" else if MATLAB installation is 64 bit version, right click on C:\VirtualRobotsCOM\register64.bat and "**Run as Administrator**"
3. A command window opens, executes some code and closes on its own (Installation is complete!).
4. Open C:\VirtualRobotsCOM\virtualRobotInMatlab.m in MATLAB and run it. Virtual Robot Application as shown in Figure 25 is displayed.
5. MATLAB file has brief comments on the workflow. Joint angles of the robot (KUKA KR5) have to be updated as per the user’s requirement and algorithm (such as Jacobian control, Cartesian motion planning etc.)
6. Denavit-Hartenberg (DH) parameters of Kuka KR5 robot is given in " C:\VirtualRobotsCOM\ DHParameters_KukaKR5.pdf"

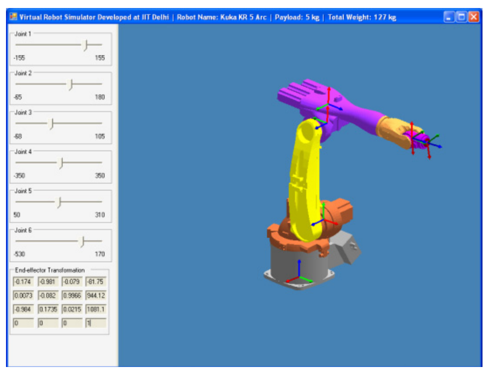


Figure 25: Virtual Robot Application Being Controlled from MATLAB

13. REFERENCES

[1] S. K. Saha, “Introduction to Robotics,” 2^{nd} Edition, Tata McGraw Hill, New Delhi, 2014
[2] ReDySim, website accessed on December 28, 2011, http://www.roboanalyzer.com/redysim.html

APPENDIX A: RECURSIVE DYNAMICS SIMULATOR (REDYSIM)

A.1. GENERAL DESCRIPTION

Recursive Dynamics Simulator (ReDySim) is a MATLAB-based recursive solver [A1] for dynamic analysis of robotic and multibody systems. ReDySim has capability to incorporate any control algorithm and trajectory planner with utmost ease. This ability provides flexibility to user/researcher in incorporating any customized algorithms. ReDySim showed considerable improvement [A1] over the commercial software and existing algorithms in terms of both computational time and numerical accuracy. ReDySim has three modules as shown in Table 1.

Table 1: Details of modules in ReDySim

	Modules of ReDySim	Demos within the module
1	Basic Module: Fixed-base Systems (Open and Closed-loop Systems)	3-link robot, gripper, KUKA robot, four-bar mechanism, biped, long chain, 3-RRR parallel robot and robotic leg
2	Specialized Module: Floating-base Systems (a) Module for Space Robots (b) Module for Legged Robots	3-and 7-link space robots, dual-arm space robot, and biped, quadruped and hexapod legged robots
3	Symbolic Module (a)Module for Fixed-base systems* (b)Module for Floating-base Systems	Fixed-base PR-robot, 2-link robot and KUKA robot, and floating-base 2-link space robot

*Presently, only fixed-base symbolic module can model prismatic joint. It will soon be added to other modules.

A.2. HOW TO INSTALL AND USE

a) Require MATLAB 2009a or higher version in order to use ReDySim.
b) Go to the webpage http://www.redysim.co.nr/download.
c) Download the required module and unzip the downloaded folder.
d) Follow the instruction manual provided in the module to start solving a problem.
e) For analysis of any system, the inputs are entered in the files shown in Table 2 (See instruction manual).

Table 2: Details of the files in which inputs are required to be entered for analysis

	Module	Sub module	*input.m*	*initial.m*	*trajectory.m*	*torque.m*	*jacobian.m*	*inv_kine.m*
1	Fixed-based	Inverse	Yes	-	Yes	-	-	Yes++
		Forward	Yes	Yes	Yes+	Yes	Yes++	-
2	Floating-base	Inverse	Yes	Yes	Yes	-	-	-
		Forward	Yes	Yes	Yes+	Yes	-	-
3	Symbolic	Fixed/Floating	Yes	-	-	-	-	-

+ Desired trajectory for controlled simulation; ++ This file is required only in the case of closed-loop systems

A.3. RUN DEMOS

a) Open the downloaded module and select Inverse /forward dynamics subfolder.
b) Within inverse/forward dynamics folder, open the subfolder with name of the system to be analyzed, i.e., open folder named biped for analyses of biped. This folder has the required input files shown in Table 2.
c) Copy all the files from this folder and paste them in inverse dynamics/forward dynamics folder.
d) Run function file *run_me.m*. This will simulate the system and plot the results of inverse dynamics (joint torques) or forward dynamics (joint motions), total energy, etc.
e) Run *animate.m* for animating the system.

A.4. SNAPSHOTS OF SIMULATION RESUTLS OF A FLAOTING-BASE BIPED USING REDYSIM

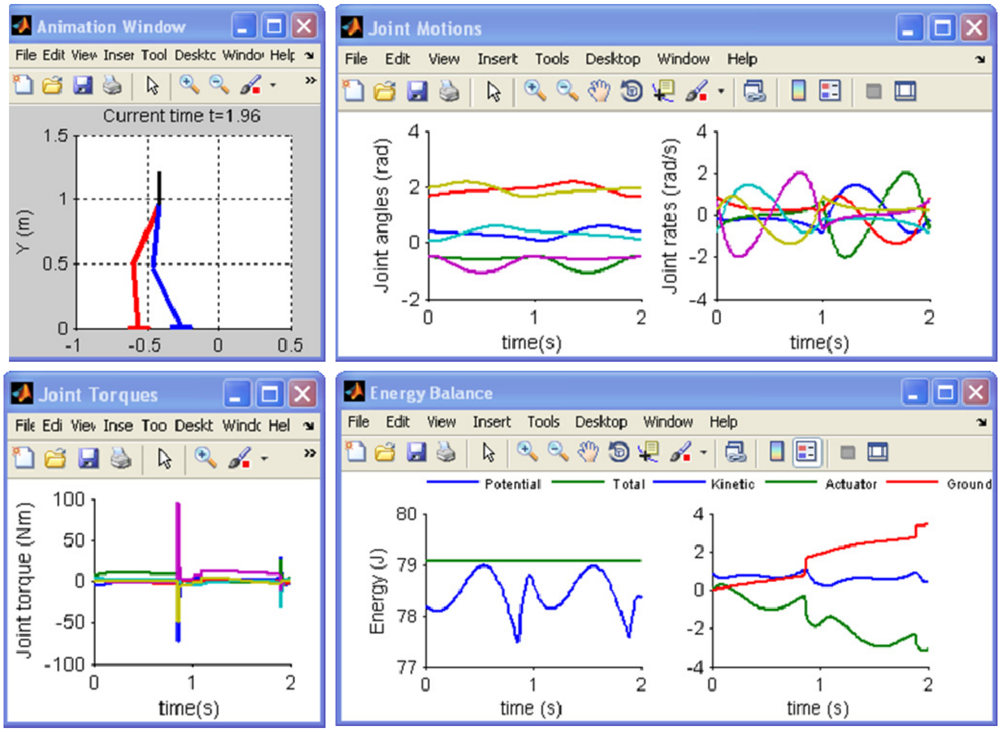
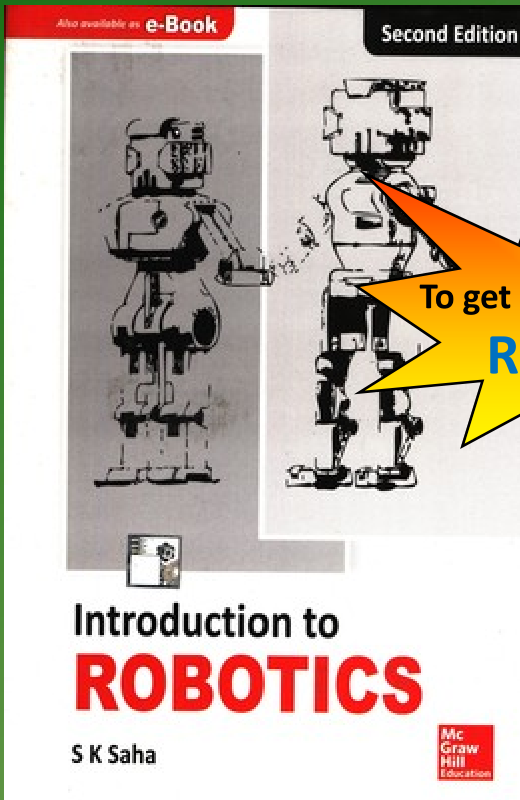


Figure A.1: Snapshots of simulation results of a biped using floating-base module of ReDySim

REFERENCE

A1. Shah S. V., Saha S. K., and Dutt J. K., "Dynamics of Tree-type Robotic Systems," Intelligent Systems, Control and Automation: Science and Engineering Book series, Springer, Netherlands (ISBN 978-94-007-5005-0).



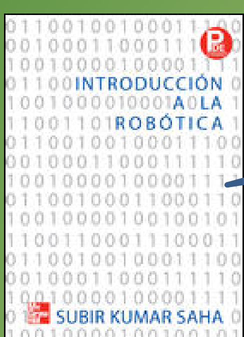
To get more insight on Robotics

"Comprehensive book that presents a detailed exposition of the concepts using a simple and student friendly approach“

“Excellent coverage of Robotic applications, Homogeneous transformation and Robotic programming etc.”

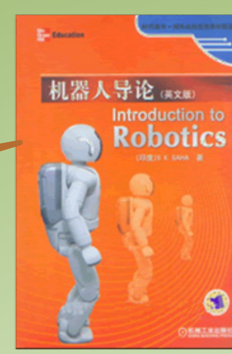
“comprehensive coverage on **Drive Systems, Robot Control,** and **Robot Applications**”

http://www.mhhe.com/saha/robotics



Published in Mexico (Spanish)

Published in P.R. China



RoboAnalyzer

3D Model Based Robotics Learning Software

Developed by Prof S. K. Saha & Team
(Mr. Rajeevlochana C. G., Mr. Amit Jain, Mr. Suril V. Shah, Ms. Jyoti Bahuguna, Mr. Ratan Sadanand and Mr. Ravi Joshi)
Mechatronics Lab, Mechanical Engineering Department, IIT Delhi, New Delhi, India
Contact: saha@mech.iitd.ac.in, roboanalyzer@gmail.com Website: http://www.roboanalyzer.com