



ROBSIM - A SIMULATOR FOR ROBOTIC MOTION

A. S. Balamesh¹, T.D. Almatrafi², A.A.N. Aljawi¹ and M. Akyurt¹

1: College of Engineering, King Abdulaziz University, P.O. Box 80204, Jeddah 21589

2: Saudi Electricity Company, The Power Plant, Madinah Al-Munawwarah

E-mail: akyurt99@yahoo.com

ABSTRACT

RobSim, a software package developed at KAU for simulating robotic motion, is introduced. The MATLAB-based RobSim can be used to simulate the motions of Cartesian, cylindrical and spherical robots. The software further enables the simulation of motions of two different configurations of the KAU robot, which is a six-degree-of-freedom articulated robot.

Using a 3D viewer, the software enables the user to specify the dimensions and configurations of the robots. These include the arm left/right, elbow up/down and wrist up/down modes. In the manual mode, RobSim allows the viewing of the motion of the manipulator while varying any selected joint angle. When in the automatic mode, RobSim determines the end points as well as the type of trajectory, and then the simulator advances the manipulator through a selected path. The simulator likewise enables the changing of the view angle of the robot. Yet another feature of RobSim is that it can produce traces of the workspace of the arms in two planes. It is concluded that RobSim possesses the potential of an educational aid as well as a teaching pendant.

Keywords: arm, manipulator, motion, robot, simulator, trace, trajectory, viewer.

()

: RobSim

.MATLAB

RobSim

RobSim

1. INTRODUCTION

Simulation of motions of robots has been receiving much attention. To cite several recent efforts in this direction, Kim and associates [Kim, Naganathan and Dukkupati, 2000] worked on collision avoidance, [Koveces, 2000] worked on the dynamic modeling of manipulators, and [Filipovic and Vukobratovic, 2000] conducted simulation experiments while studying dynamic accuracy of robotic mechanisms. RobSim, which is a robot simulation program, was developed in its entirety at KAU [Dehlawi et. al., 1996; Al.Matrafi, 2000] to simulate the motions of robotic arms. More specifically RobSim can simulate the motions of five types of robots. These are the Cartesian, cylindrical and spherical robots as well as two different configurations of the articulated robot developed at KAU, i.e., the KAU PYR and the KAU RPR. The simulation language utilizes the MATLAB software.

2. THE INTERFACE SCREEN

The interface screen of RobSim is shown in Fig. 1. Detailed descriptions of some of the main features of RobSim are presented below by referring to Fig. 1.

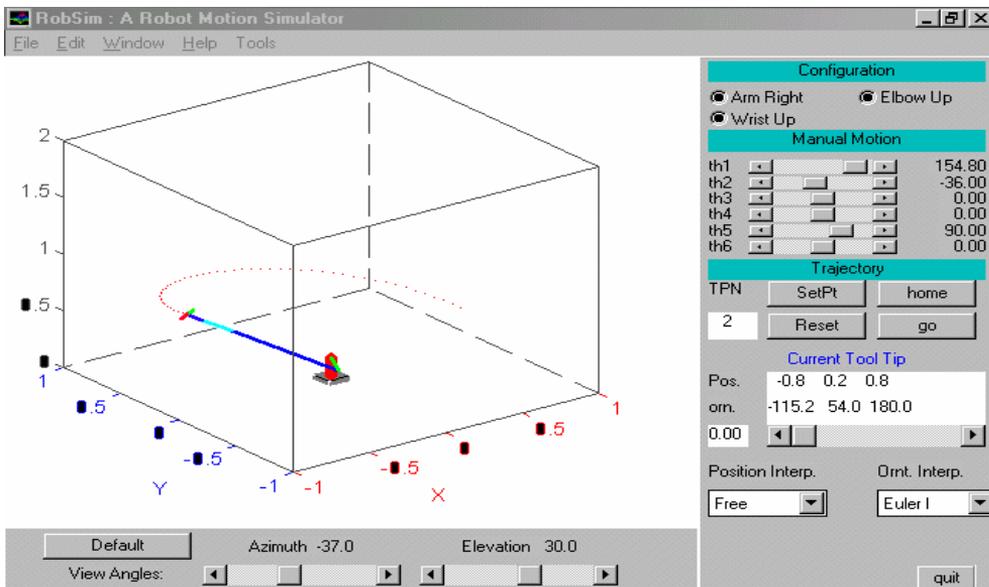


Fig. 1 The user interface screen of RobSim.

The interface screen consists essentially of the *Control Region*, which is the rectangular region on the right side of the screen, and the *Robot View Region*, which covers most of the rest of the screen. The *Control Region* consists of the following sections.

Configuration

This section contains radio buttons with binary selections for choosing the configuration of the arm (right or left), the elbow (up or down) and the wrist (up or down). These choices are used in inverse kinematics analysis [Dehlawi et. al., 1996; Al.Matrafi, 2000; Aljawi, 2002] to select the appropriate solution from among a number of possible solutions.

Manual Motion

This section contains six sliders, each slider for controlling the angle associated with one of the six joints. The angles associated with the joints are labeled as θ_1 through θ_6 . The current value (in degrees) for each joint angle is indicated to the right of the corresponding slider. As the user moves a joint, the value for the corresponding angle varies accordingly. The minimum and maximum joint values can be set by referring to the *parameters* screen shown in Fig. 2.

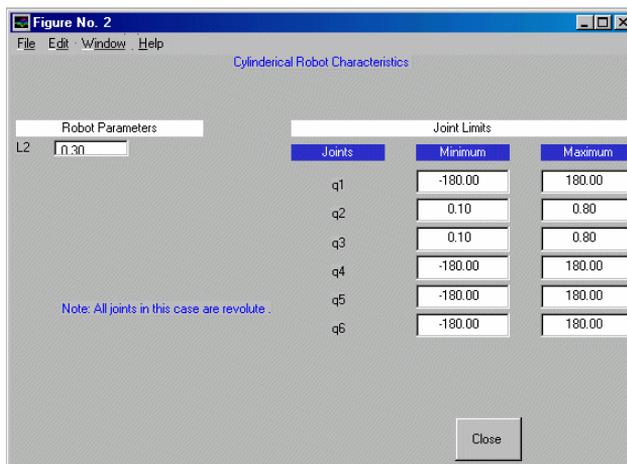


Fig. 2 Robot parameters.

Trajectory

Immediately below the section on sliders, there are trajectory-related buttons. Thus the *TPN* button indicates the total number of points utilized on the trajectory. The button labeled *home* moves the robot back to its home position. Activation of the *Reset* button deletes all stored points for trajectory planning, and the *Go* button causes the robot to move to the selected new position by following the trajectory path provided by the inverse kinematics solution.

The button labeled *SetPt* allows the setting of the current robot position as the *start* or the *end* position of the trajectory. As such the *SetPt* button is utilized to *teach* the robot. In the present case the robot is taught the start and end positions of the trajectory. The software may be readily modified to teach the robot more points along the trajectory. Then RobSim would assume the task of a *teaching pendant*.

The current position *Pos* and orientation *Orn* of the tool tip are displayed in this section. The orientation of the tool tip is given in terms of Euler-I angles relative to the base coordinate. The position of the tool tip is indicated by its coordinates. This portion of the display is read-only, and cannot be modified directly. However, the display changes continuously as the robot changes position.

The user is given the freedom to select the type of path interpolation to be used both for the position and the orientation of the tool. For tool position there are two available choices. One choice is *free*, where no path interpolation is used, and joints are allowed to move at constant angular steps until they reach their end positions. The second choice is *linear interpolation*, where the path is broken into a number of points along a straight line and free interpolation is used among those points.

The *Robot View Region* of the interface screen displays a 3D picture of the robot, with the robot enclosed inside a box, and the coordinate axes as edges of the box. The hidden edges of the box in Fig. 3 are shown as dashed lines to help the viewer imagine the 3D view at different viewing angles. The several links of the robot are shown indifferent colors. The height of the base is fixed as a fraction of H (we have used 0.25H). The two sides of the end-effector are shown in different colors to make it easy to observe the effect of rotating the tool around the approach vector.

Under manual control, when the robot is activated by the use of the sliders described above, the robot in the *robot view* region moves accordingly. When the *go* button in the trajectory section is pressed, the robot moves along the specified trajectory, and a trace of the trajectory is displayed.

Right under the window of the *robot view* region are the *view controls*, consisting of two sliders, one for the azimuth angle and the other for the elevation angle. The view angles vary continuously as the sliders are moved. The *Default* button returns the view angles to their default values (azimuth = -37° and elevation = 30°). The manipulation of view angles enables the viewing of the position of the robot as well as that of the trace of the trajectory in 3D.

At the top left of the user interface of Fig. 1 is located the command *tools*. When invoked, a submenu containing *Robot Parameters* and *Change Color* appears. The *Change Color* command allows the user to change the colors used in the *Robot View Region*. Opting for the

second command, *Robot Parameters*, causes a screen like that shown in Fig. 2 to be displayed, which screen displays the characteristics of a cylindrical robot. Each of the five types of robots is provided with its own screen of *Robot Parameters*. The user is given the opportunity to enter robot dimensions like base height, link lengths, and distances of offsets. By *joint limits* is meant the minimum and maximum values of displacement at a particular joint. It is possible to modify the joint work envelope by manipulating the joint limits. The version of Fig. 2 that is associated with articulated robots further allows the user to define the configuration of the last three joints (RPR/PYR) via a pop-up menu.

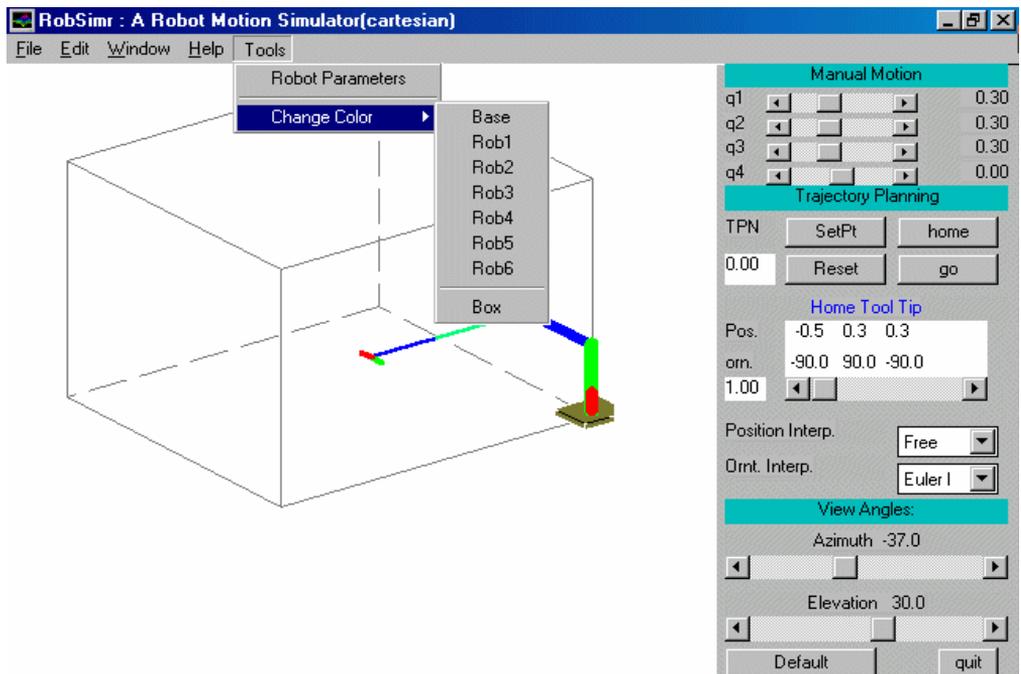


Fig. 3 The interface screen of RobSim.

3. ROBOT WORKSPACE

The work envelope of a robot is defined as *the locus of points in three-dimensional space that can be reached by the wrist*. The axes of the first three joints will be referred to as the major axes. It may be said that it is the major axes that are used to determine the position of the wrist. The axes of the remaining joints, the minor axes, are used to establish the orientation of the tool. As a consequence, the geometry of the work envelope is determined in general by the sequence of joints used for the first three axes.

In using the tool tip as a reference point, the effects of both major axes used to position the wrist and the minor axes used to orient the tool must be included. When viewed as a subset of \mathbf{R}^3 the shape or geometry of the work envelope varies from robot to robot. However, the work envelope can also be viewed within the framework of joint space. In joint space, the work envelope is typically characterized by bounds on linear combination of joint variables. Constraints of this nature generate a convex polyhedron in \mathbf{R}^n . We call this polyhedron the *joint-space work envelope*.

3.1 Cartesian Robot Workspace

The link-coordinate diagram for the Cartesian robot, which is a four-axis robot with tool roll motion, is shown in Fig. 4. To investigate the work envelope of this robot, one starts with the expression for the tool-configuration vector $W(q)$. It may be shown that [Al-Matrafi, 2000] the expression for the tool configuration vector of the Cartesian robot is:

$$W(q) = \begin{bmatrix} -L - q_3 \\ q_2 \\ q_1 \end{bmatrix}$$

The only joint variable of the minor axis that is used to orient the tool is the tool roll angle q_4 . However, it is evident from inspection of above equation that the tool roll angle has no effect on the tool-tip position \mathbf{p} . Consequently, the tool roll angle can be ignored in our investigation of the size and shape of the work envelope [Schilling, 1990].

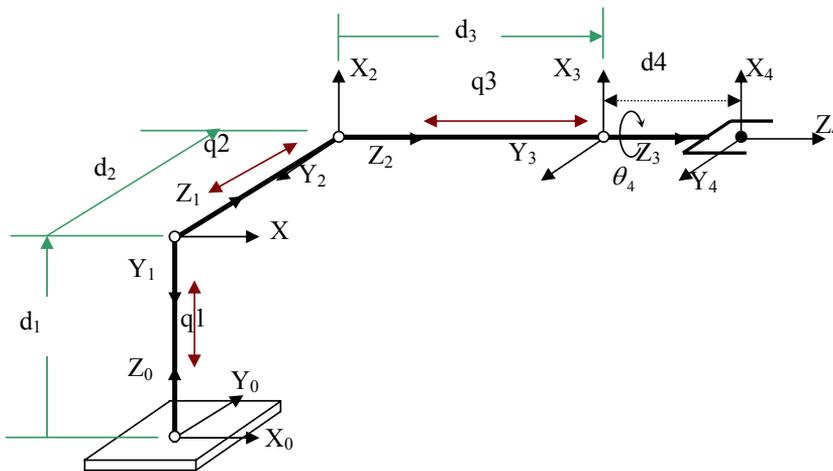


Fig. 4 Link-coordinate diagram for the Cartesian robot.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -\pi/2 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} q \leq \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ \pi/2 \end{bmatrix}$$

Note that there is effectively no constraint on the last angle (revolute) q_4 : it is allowed to range over one cycle $[-\pi/2, \pi/2]$. The constraints on the first three angle the distance limits of prismatic angles. The above equation, when applied in RobSim, enables RobSim to draw the workspace of the Cartesian robot (Fig. 5) instantaneously.

3.2 Workspace of the Cylindrical Robot

The workspace of this robot will be described depending on the first three joints. Consequently, the work envelope of the cylindrical robot is cylindrical in shape.

There is effectively no constraint on the base angle q ; it is allowed to range over on complete cycle $[-\pi, \pi]$. The second and third joints have the prismatic joint constraints, which depend on the maximum and minimum reachable of the joint sliding (Fig. 6).

$$\begin{bmatrix} -\pi \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} q \leq \begin{bmatrix} \pi \\ l_1 \\ l_2 \end{bmatrix}$$

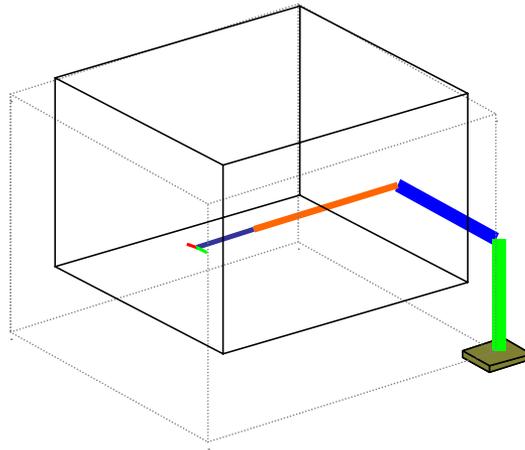


Fig. 5 3D-Cartesian robot workspace.

Figure 6, which was drawn by RobSim, illustrates the XY plane shape of the cylindrical robot workspace. The inner circle represents the minimum reachable and the outer one represents the maximum reachable. The distance between the two circles represents the horizontal stroke.

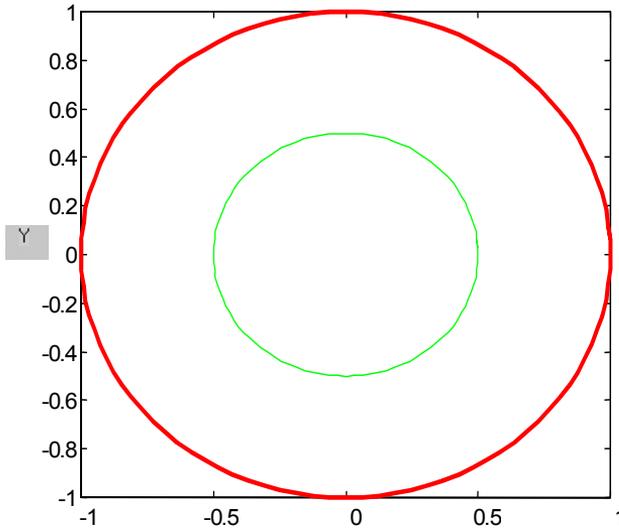


Fig. 6 The XY plane of a cylindrical robot work envelope.

3.3 Spherical Robot Workspace

The first two joints of the spherical robot are revolute. There are no constraints on joint one, it is allowed to complete cycle $[-\pi, \pi]$, Also joints four and five are allowed to make complete cycle $[-\pi, \pi]$. The range of joint five (pitch angle) is half cycle $[-\pi/2, \pi/2]$.

The following equation describes the work envelope of first three joints.

$$\begin{bmatrix} -\pi \\ -3\pi/2 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} q \leq \begin{bmatrix} \pi \\ \pi/2 \\ d_3 \end{bmatrix}$$

Figure 7 illustrates the XY plane of the spherical robot. The inner circle represents minimum reachable and the outer circle represents maximum reachable. The distance between the maximum and minimum circles are represents the horizontal stroke. In Fig. 8, illustrate the ZX plane of the spherical workspace shape. The inner semi circle represents the minimum reachable and the outer semi circle represents maximum reachable. Also, the distance between the maximum and minimum semi circles are represented the horizontal stroke.

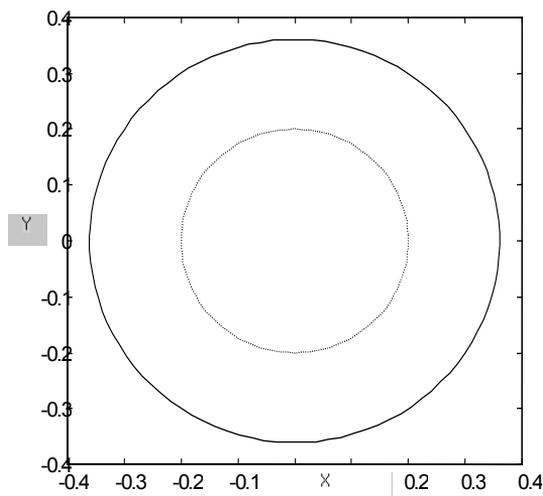


Fig. 7 The XY plane of spherical robot workspace.

3.4 Articulated Robot (KAU) Workspace

The first three joints of the articulated robot are revolute. There are no constraints on these joints, The cycle is $[-\pi/2, \pi/2]$. The following equation describes the work envelope of first three joints.

$$\begin{bmatrix} -\pi/2 \\ 0 \\ -\pi/2 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} q \leq \begin{bmatrix} \pi/2 \\ \pi/2 \\ \pi/2 \end{bmatrix}$$

Figure 9 shows a horizontal section of the workspace of the arm. This shape is generated by rotating joint one 360° . Joint two is required to rotate through a total angle 90° while joint three required to rotate a full revolution to generate the envelope shown in Fig. 10.

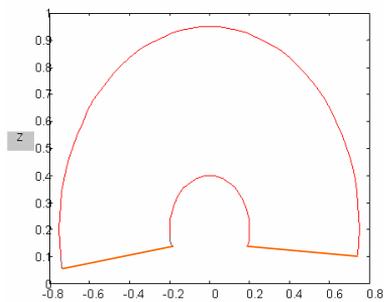


Fig. 8 The ZX plane of spherical robot

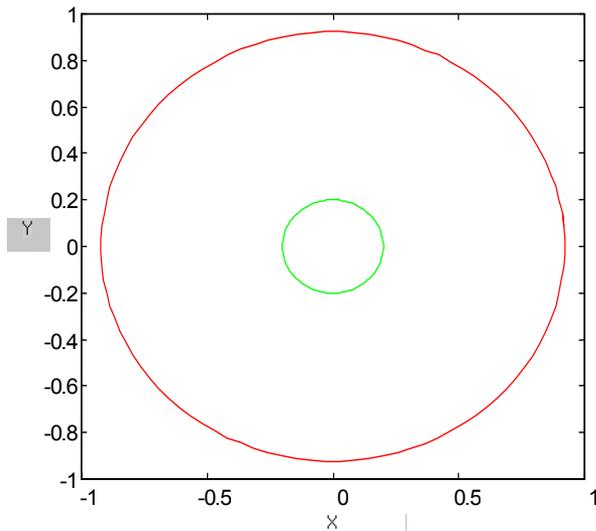


Fig. 9 The XY plane of the workspace of articulated robot..

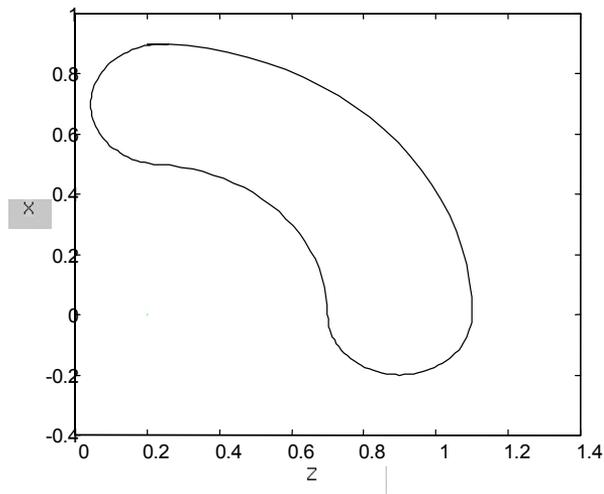


Fig. 10 The ZX plane of articulated robot workspace.

4. CONCLUSIONS

The direct kinematics and inverse kinematics solutions that were developed by the authors [Dehlawi, et al., 1996; Al-Matrafi, 2000; Aljawi et al., 2002] are used in RobSim for visualization purposes. In addition, workspace relationships are provided here for the accurate generation of the workspace of any of the five types of robots. These are the two

configurations of the KAU articulated robot as well as the Cartesian, cylindrical and spherical robots. RobSim allows the changing of robot arm parameters by the user, making the visual displays meaningful and immediately useful.

The robot workspace can be drawn instantaneously for any selected type of robot and for any set of selected parameters. Each link can be colored separately for easy identification. It may be concluded hence that the locally developed RobSim can be used as an educational tool. This is especially true if an industrial robot is physically available, such as the KAU robot which was also designed, manufactured and developed in house. The software further possesses the potential to be used as a teaching pendant.

It may be pointed out that RobSim is not the only software that is currently available for the simulation of robotic motion. Another program, also called Robsim [Anon., 2002], is available on the website of the University of Texas at Arlington. The latter software however, utilizes the Robotics Toolbox [Anon1, 2002] of MATLAB for its basic relationships and computations. This is in contrast with the RobSim software presented here, which package stands on its own feet.

REFERENCES

1. Aljawi, A.A.N., A.S. Balamesh, T. D. Almatrafi and M. Akyurt, 2002, "Symbolic modeling of robotic manipulators," *Proc. 6th Saudi Eng. Conf., KFUPM* (accepted).
2. Al-Matrafi, T.D., 2000, "Kinematic Simulation of a Six-Axis Robot Manipulator," Unpublished MS Thesis, King Abdulaziz Univ., Mechanical Eng. Dept., Jeddah 21589.
3. Anon, 2002, "Robsim", <http://robotics.uta.edu/me5337>.
4. Anon1, 2002, "Robotics Toolbox", <ftp.mathworks.com> in directory put/contrib/misc/robot.
5. Dehlawi, F.M., M. Akyurt, A.A. Aljawi, A.S. Balamesh and A.M. Al-Qasimi, 1996, "Design and Implementation Of An Industrial Robot", Progress Report 2, KACST Project 12-42, Riyadh.
6. Filipovic, M. and Vukobratovic, M., 2000, "Dynamic Accuracy of Robotic Mechanisms. II Simulation Experiments and Results Discussion", *Mechanisms and Machine Theory*, 35 (2) pp 239-270.
7. Kim, Y.G., Nagathan, N.G. and Dukkipati, R.V., 2000, "Real Time Collision Avoidance Algorithm," *International Journal of Vehicle Design*, 24 (2) pp 224-242.
8. Koren, Y., 1987, *Robotics for Engineers*, McGraw-Hill.
9. Kovacs, J., 2000, "Dynamic Modeling and Analysis of a Robot Manipulator Intercepting and Capturing of a Moving Object," *Shock and Vibration Digest*, 32 (1) pp 23 -24.
10. Schilling, R.J., 1990, *Fundamentals of Robotics, Analysis and Control*, Prentice-Hall.