Int. J. Engng Ed. Vol. 11, No. 6, pp. 441–449, 1995
Printed in Great Britain.

0949-149X/91 $3.00+0.00

A Multimedia Approach to Teaching Robot Kinematics*

S. C. FOK
F. L. TAN
P. NG

School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 2263

A multimedia approach to teaching robot kinematics is presented in this paper. Graphical three-dimensional animation of robot motions with sound and text is a useful teaching tool for undergraduates in robotics. A multimedia package developed in-house, MMROBOT, to assist beginning students in robotics to understand elementary robot kinematics is described. The paper recommends the implementation of the multimedia package at a university robotics laboratory.

INTRODUCTION

MULTIMEDIA has become a household word in recent years with the rapid evolution of software and hardware to generate graphics, animation, sound and text on a personal computer (PC). Educational software has been developed using multimedia technology on PCs to facilitate learning at home. Multimedia on PC makes the learning of a difficult topic more interesting and easier. Multimedia is also slowly making inroads into the teaching of engineering to undergraduates in universities. Courses such as thermodynamics, robotics, materials science, fluid mechanics and others can be taught using multimedia on PCs.

Courses in robotics can be found in almost any engineering school. Robotics is usually taught in conjunction with practical applications in the laboratory, using small 'teaching robots' [1]. Large robotic systems, which are most representative of the actual production equipment in the industry, are expensive to acquire for the university laboratory. As such, computer graphical simulation [2] of large robotic systems can be most easily accomplished using a PC. Multimedia software can be developed on the PC to teach engineering undergraduates about the kinematics of large robotic systems, without having to spend large sums of money on large robots in the laboratory.

The literature on simulation of robotic systems is quite extensive. Soroka [3] has discussed some of the possibilities available for the graphical simulation of robots. The three major display modes obtainable are monocular, stereo and two-camera (i.e. plan and elevation views); while drawing techniques available include splines and wire-frame.

The earlier simulation packages were typically developed for workstations, due to the limited capabilities of PCs. Mulvenna and Fritz [4] have discussed the development of programs on an IBM 4341 workstation for graphical simulation as part of a university course. Their programs featured three-dimensional wire-frame displays. Simulation programs developed at Cornell at the beginning of the 1980s included the work of Leu and Park [5], and Mahan and Walter [6]. The programs by Leu and Park [5] consisted of two parts. The first was a solid geometric modeller, PADL-2, which could be used to simulate the geometry of various types of robots. The second was a wire-frame modeller, PICSYS, which served to simulate robot motion. The equipment used included VAX 11/780 and VAX 11/750 computers. Mahan and Walter [6] developed the program IGRIP on similar platforms. Their program, for the emulation of both robot kinematics and dynamics, was initially intended for a robotics course but later enhanced for industry.

One of the first simulation package developed on an IBM PC was the SIM-7535 package by Meyer and Jayaraman [7] in 1983. This package simulates the IBM 7535 robot. Input to the simulator is a file containing an AML/Entry language program. Output includes a splines display of the arm motion seen in plan view. However, three-dimensional representations of motion are not covered mainly due to the limitations of both hardware and software at that time. A recent paper by Hejase and Hasbini [8] presented a user-friendly package developed on a PC to animate the kinematics of a robot manipulator. The motion of the robot arm is displayed using wire-frame, which does not give a

---

* Paper accepted 15 July 1994.

realistic physical representation of the robot arm. With the dramatic development of computer technology, the above limitations have been overcome. Software on solid modelling and 3-D animation are now readily available for the PC. Such capabilities have the potential to be incorporated into robotic simulation programs to make the emulation more realistic. Furthermore, these capabilities are accessible at a relatively low cost and it is now highly cost-effective to use PC animation programs to provide 'hands-on' experience for large robotics classes.

One of the basic topics of robotic education is the kinematics of robots [9]. In this paper the application of multimedia on a PC for the teaching of robots' kinematics is presented. The paper first gives an overview of the problems faced by many students studying robot kinematics. Then the multimedia educational software package developed in-house, MMROBOT, is described. The paper concludes with recommendations for the implementation of the multimedia package in a university robotics laboratory.

KINEMATICS OF ROBOTS

The arm of a robot consists of links connected by joints. Each link is actuated by an actuator and its motion can be angular rotation or rectilinear translation. Definite relationships can be established between the joint actuators' displacement and the final position and orientation of the end effector. These mathematical relations depend on the robot's configuration.

The kinematics analysis of a robot involves the following steps.

1. Define a world coordinate system and systematically assign the Denavit–Hartenberg coordinate frames to the links [8].
2. Determine the structural parameters for each link. These include the length a_i and the offset angle α_i. Figure 1 shows the structural kinematics parameters for a general link i.
3. Determine the joint variables, i.e. either angle θ_i for revolute joint or translation d_i for prismatic link.
4. Compute the link transformation matrix relating the coordinates of two successive frames $i-1$ and i with

$$A_{i-1}^{i}=\begin{bmatrix}\cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos y_i\\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos_i\sin\alpha_i & a_i\sin\theta_i\\ 0 & \sin\alpha_i & \cos\alpha_i & d_i\\ 0 & 0 & 0 & 1\end{bmatrix}\tag{1}$$

5. Determine the expression for the hand matrix. For an n degree of freedom manipulator:

$$T_0^n=A_0^n=A_0^1A_1^2\ldots A_{n-1}^n=\begin{bmatrix}n_x & s_x & a_x & P_x\\ n_y & s_y & a_y & P_y\\ n_z & s_z & a_z & P_z\\ 0 & 0 & 0 & 1\end{bmatrix}\tag{2}$$

6. Given the values of the joint variables, determine the final position and orientation of the end effector using equation (2).

The inverse kinematics problem is to determine the appropriate values of the joint variables that correspond to a given location and orientation of the end effector. This involves comparing the various elements in the matrix equation (2) and deducing the governing equations for the various joint variables in terms of $n_x, s_x, a_x, P_x, n_y, s_y, a_y, P_y, n_z, s_z, a_z, P_z$.

The major problems faced by students in the

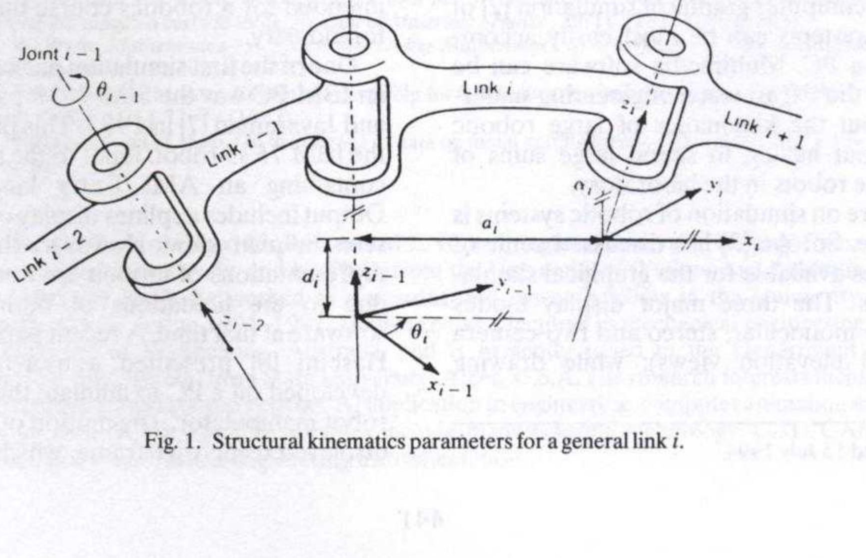


Fig. 1. Structural kinematics parameters for a general link *i*.

study of robot kinematics are the assignments of the coordinate frames for different robot configurations and the determination of T_0^n. Another difficulty faced by many students is the determination of the inverse kinematics equations for the various joint variables. To assist the students in overcoming these difficulties, realistic visualization aids using computer graphics are highly desirable. Furthermore, computer graphics can also be used to show the applications of robot kinematics in a trajectory control to students who have mastered the theory.

MULTIMEDIA PACKAGE—MMROBOT

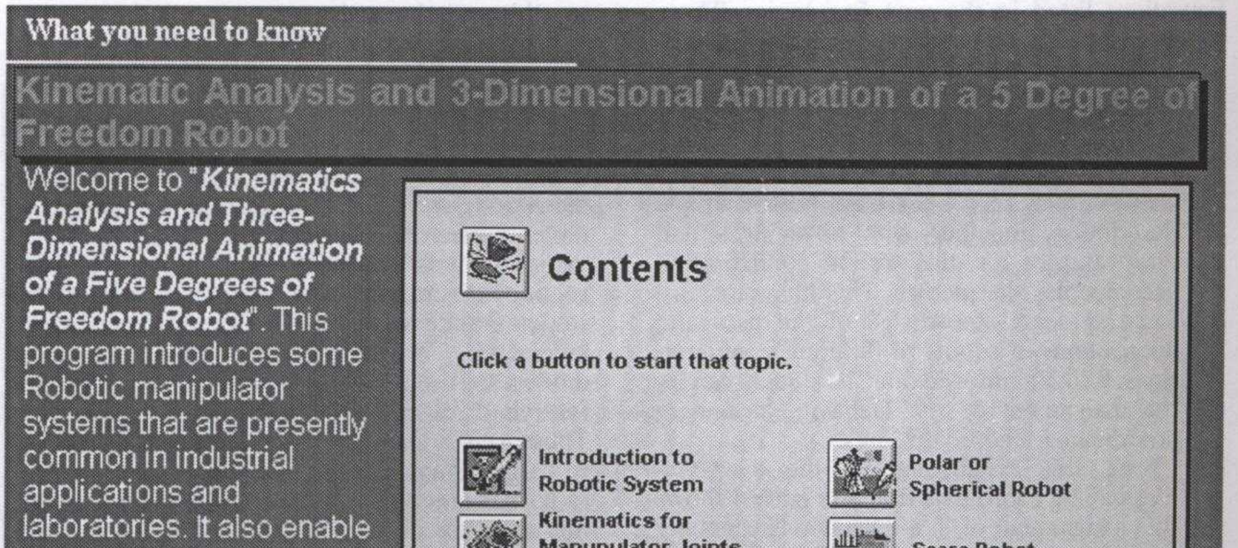
The MMROBOT multimedia package was developed to assist beginning students in robotics to understand elementary robot kinematics. It can cater for any five degree-of-freedom robots of the following configurations: rectangular, cylindrical, spherical, SCARA and revolute manipulators. It is written for an IBM compatible PC (4 MB RAM recommended) operating under MS.DOS 3.1 (or higher). It uses Microsoft Windows 3.1, Autodesk 3D studio to create Autodesk animation and Compel for multimedia presentation. An MCI-compatible sound card to play wave or MIDI audio is required.

MMROBOT is designed to be user-friendly and navigation through the package can be easily accomplished with mouse and keys. A menu page allows the user to tour selected topics (see Fig. 2). The menu consists of nine topics which can be separated into two main modules. These are the static display module and the animation module (see Table 1).

The static display module is designed in the form of a tutorial program to help reinforce student knowledge. Figures drawn in the static display module mainly consist of three-dimensional wire-frame representations of objects.

The topic entitled ‘Content’ merely explains the methods of navigation through the program. It also contains an overall summary of the other topics in the package. The first lesson gives an introduction on robotic systems which explains the functions of a robot.

The next lesson is on kinematics of manipulator joints, links and grippers. This gives a fundamental explanation of coordinate frames and the transformation between frames. It also contains documentation on the procedural assignment of links' coordinate frames. Information on the calculation



What you need to know

Kinematic Analysis and 3-Dimensional Animation of a 5 Degree of Freedom Robot

Welcome to "*Kinematics Analysis and Three-Dimensional Animation of a Five Degrees of Freedom Robot*". This program introduces some Robotic manipulator systems that are presently common in industrial applications and laboratories. It also enable you to grasp the important concepts and features of Robots.

To know more about the program, please click the bottom right button.

Contents

Click a button to start that topic.

Fig. 2. Menu page showing the contents.

Table 1. Menu topics separated into modules.

MODULES	MENU TOPICS
Static display	Contents
	Introduction to Robotic System
	Kinematics for Manipulator Joints, Links and Grippers
	Rectangular Robot
	Cylindrical Robot
	Spherical Robot
	SCARA Robot
	Revolute Robot
Animation	Kinematics and Controlled Path Programs

of the inverse kinematics solution can also be found in this section.

After navigating through the first two lessons, the student can select any of the specific robot configurations listed in the next five topics. These topics include the rectangular, cylindrical, spherical, SCARA and revolute robots. In each of these topics, the students can access the following information.

(a) View the specific robot configuration. Figure 3 shows a view of a SCARA robot. A simple description, together with advantages and disadvantages of this robot configuration accompanies the picture. This helps to reinforce students' knowledge of the uses and applications of robots of different configurations. With this information, the student can try the assignment of the Denavit–Hartenberg coordinate frames to the links.

(b) Check the assignment of the Denavit–Hartenberg coordinate frames. Figure 4 shows the assignment of the Denavit–Hartenberg coordinate frames for the SCARA robot. This information is meant mainly as a check of the students' results on coordinate frames assignment. Once the frames assignments are checked, the students can proceed to derive the various linkages transformations and then to combine them to get the overall hand matrix.

(c) Check the derived expressions for the hand matrix (equation (2)). Figure 5 shows the hand matrix for the revolute robot with which the students can check their results. Following this, the students can try to obtain the inverse kinematics relationship for the robot.

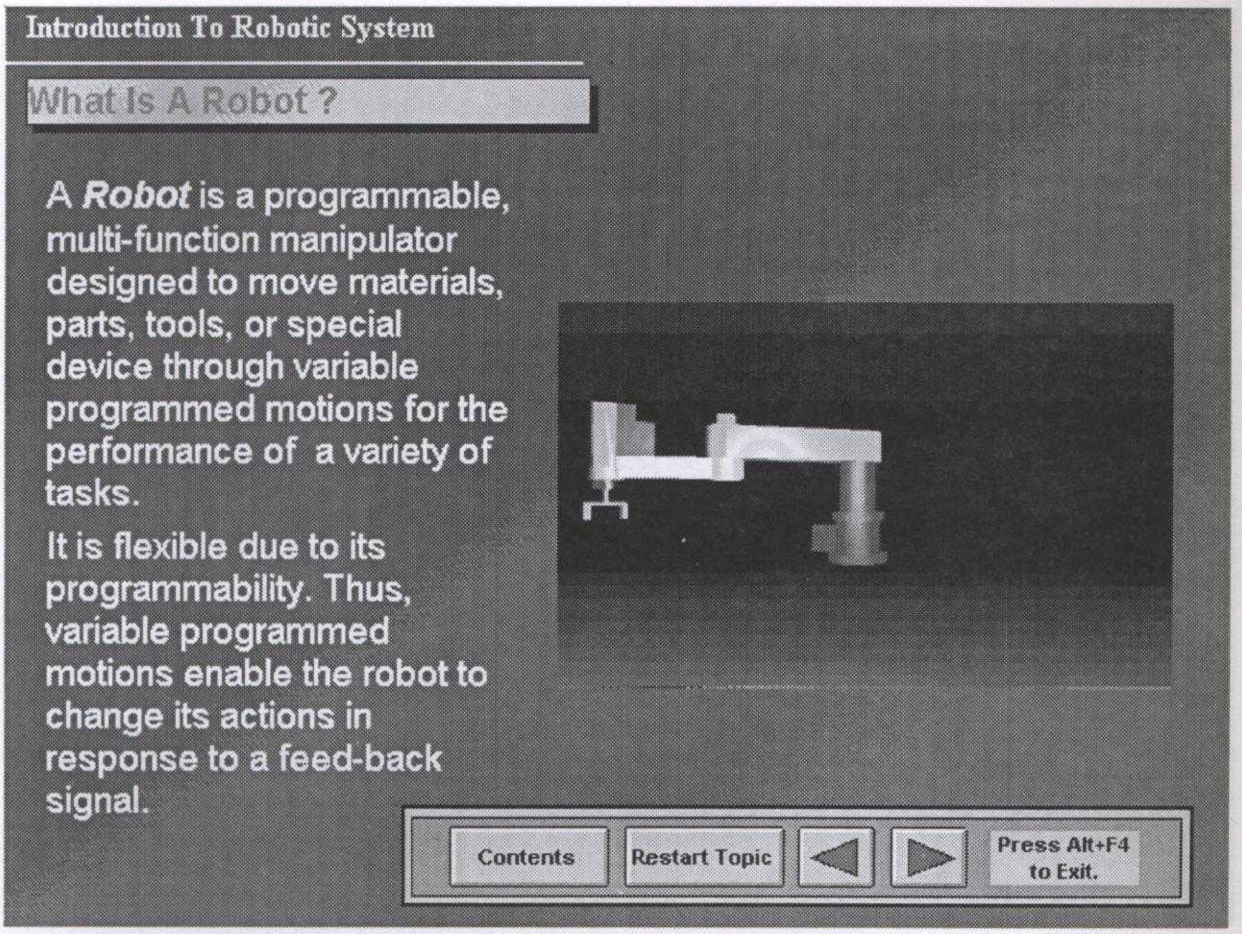
(d) Check the inverse kinematics equations. Figure 6 shows the inverse kinematics solutions for the revolute robot. If the students get this correct, they can proceed further.

(e) Find the required joint variables when given the end effector position and orientation. In this part, the students are supposed to use the inverse kinematics equations to try to solve the joint variables for a few end effector positions and orientations.

In the animation module, the program has the ability to simulate the controlled path when given the starting and final locations and orientations of the end effector. This feature integrates the kinematics analysis with the animation of the various robot configurations. It is designed to enhance the students' understanding of the application of robot kinematics. To provide realistic animation, solid models are used. Figures 7 and 8 show the images for cylindrical and spherical robots, respectively. These were created using 3D Studio and rendered to Targa format at 24-bit resolution.

To make the animation module more interesting, the users are allowed to set the links' structural parameter and select the trajectory path that the manipulator should follow. These trajectories include linear and planar interpolations. Figure 9 shows options for setting the parameters for a SCARA robot.

The integration of the static and animation modules is accomplished using Compel (a multimedia package). To make the learning process more interesting, sound and music are used. Furthermore, human voice can be embedded to read the text of the static modules. The whole package, MMROBOT, is stand-alone and can be ported to other PCs in the laboratory. The student can even port to his or her home PC for self-learning.



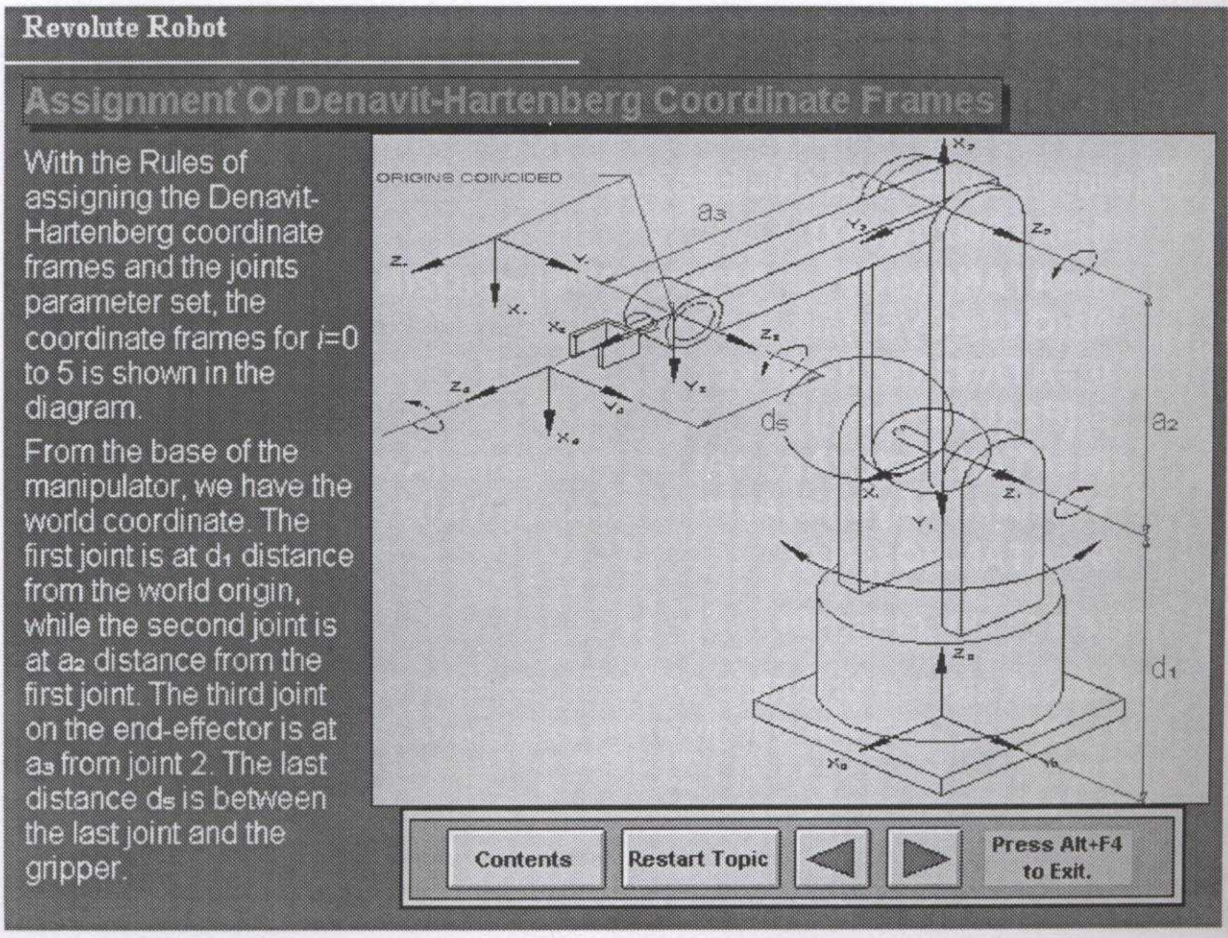
Introduction To Robotic System

What Is A Robot ?

A ***Robot*** is a programmable, multi-function manipulator designed to move materials, parts, tools, or special device through variable programmed motions for the performance of a variety of tasks.

It is flexible due to its programmability. Thus, variable programmed motions enable the robot to change its actions in response to a feed-back signal.

Fig. 3. Configuration of a SCARA robot.



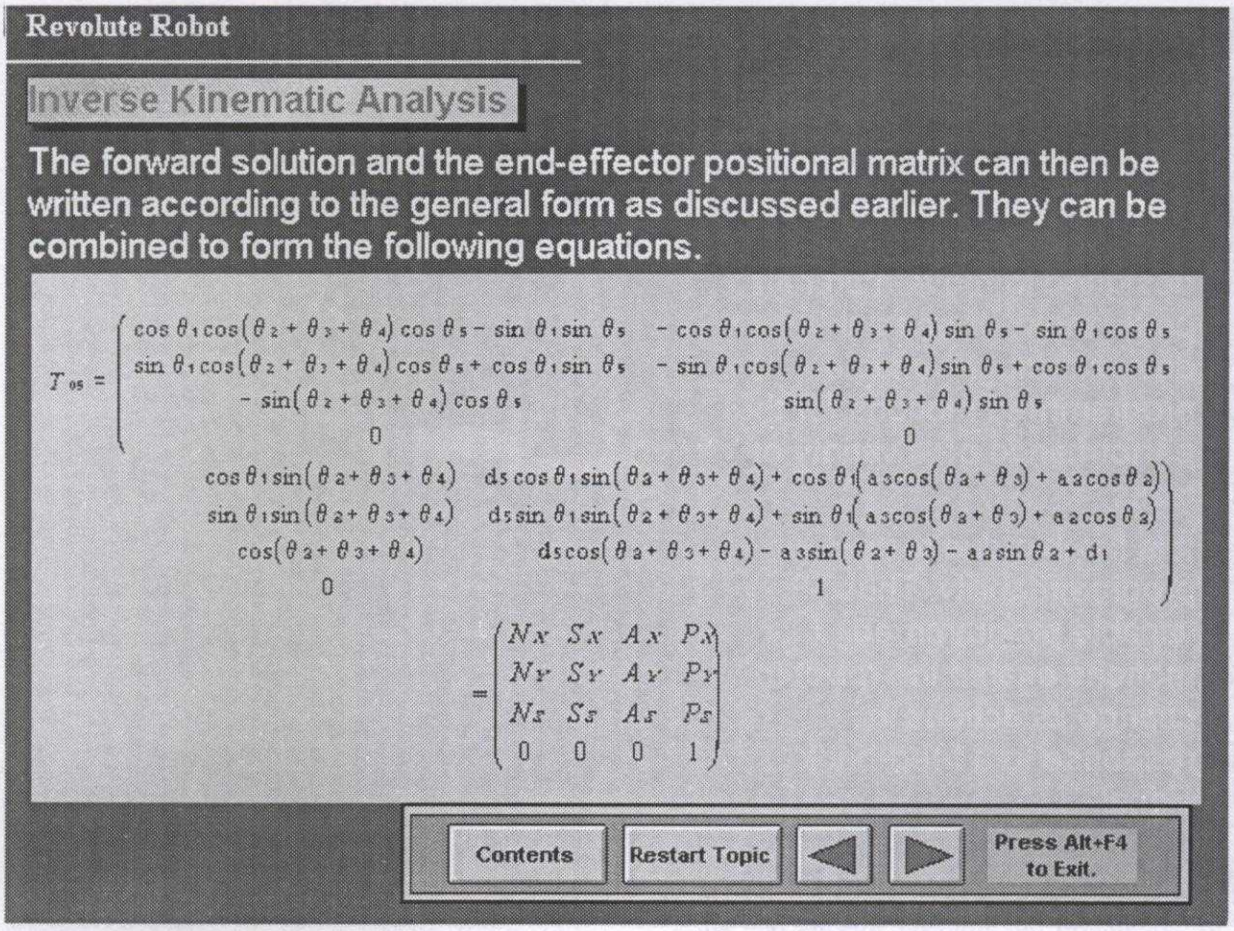
Revolute Robot

Assignment Of Denavit-Hartenberg Coordinate Frames

With the Rules of assigning the Denavit-Hartenberg coordinate frames and the joints parameter set, the coordinate frames for i=0 to 5 is shown in the diagram.

From the base of the manipulator, we have the world coordinate. The first joint is at d_1 distance from the world origin, while the second joint is at a_2 distance from the first joint. The third joint on the end-effector is at a_3 from joint 2. The last distance d_5 is between the last joint and the gripper.

Fig. 4. Assignment of D–H coordinate frames to a SCARA robot.



Revolute Robot

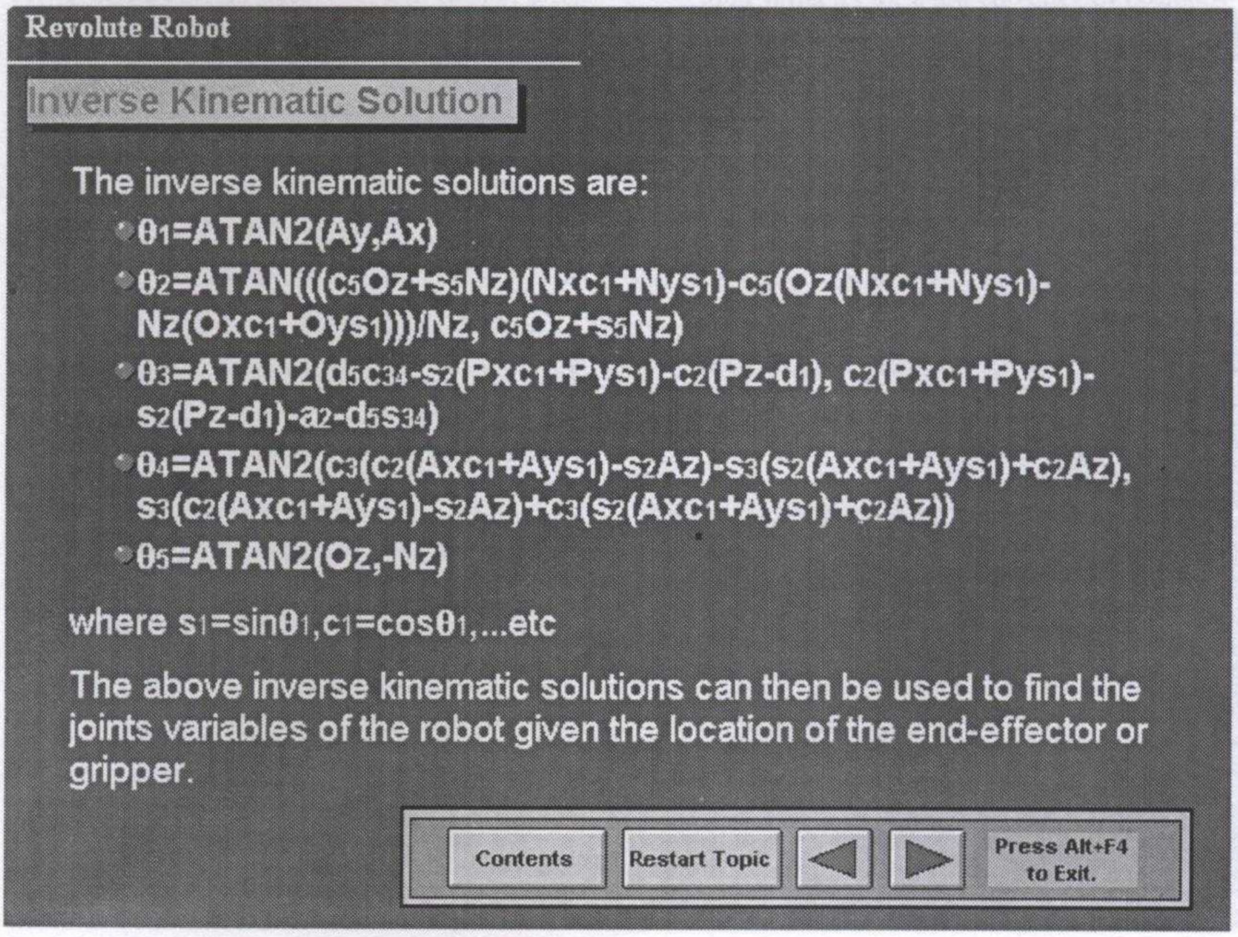
Inverse Kinematic Analysis

The forward solution and the end-effector positional matrix can then be written according to the general form as discussed earlier. They can be combined to form the following equations.

$$T_{05} = \begin{pmatrix} \cos\theta_1\cos(\theta_2+\theta_3+\theta_4)\cos\theta_5 - \sin\theta_1\sin\theta_5 & -\cos\theta_1\cos(\theta_2+\theta_3+\theta_4)\sin\theta_5 - \sin\theta_1\cos\theta_5 & \cos\theta_1\sin(\theta_2+\theta_3+\theta_4) & d_5\cos\theta_1\sin(\theta_2+\theta_3+\theta_4) + \cos\theta_1(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2) \\ \sin\theta_1\cos(\theta_2+\theta_3+\theta_4)\cos\theta_5 + \cos\theta_1\sin\theta_5 & -\sin\theta_1\cos(\theta_2+\theta_3+\theta_4)\sin\theta_5 + \cos\theta_1\cos\theta_5 & \sin\theta_1\sin(\theta_2+\theta_3+\theta_4) & d_5\sin\theta_1\sin(\theta_2+\theta_3+\theta_4) + \sin\theta_1(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2) \\ -\sin(\theta_2+\theta_3+\theta_4)\cos\theta_5 & \sin(\theta_2+\theta_3+\theta_4)\sin\theta_5 & \cos(\theta_2+\theta_3+\theta_4) & d_5\cos(\theta_2+\theta_3+\theta_4) - a_3\sin(\theta_2+\theta_3) - a_2\sin\theta_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} Nx & Sx & Ax & Px \\ Ny & Sy & Ay & Py \\ Nz & Sz & Az & Pz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 5. Hand matrix for the revolute robot.



Revolute Robot

Inverse Kinematic Solution

The inverse kinematic solutions are:

- θ_1=ATAN2(Ay,Ax)
- θ_2=ATAN(((c_5Oz+s_5Nz)(Nxc_1+Nys_1)-c_5(Oz(Nxc_1+Nys_1)-Nz(Oxc_1+Oys_1))))/Nz, c_5Oz+s_5Nz)
- θ_3=ATAN2(d_5c_{34}-s_2(Pxc_1+Pys_1)-c_2(Pz-d_1), c_2(Pxc_1+Pys_1)-s_2(Pz-d_1)-a_2-d_5s_{34})
- θ_4=ATAN2(c_3(c_2(Axc_1+Ays_1)-s_2Az)-s_3(s_2(Axc_1+Ays_1)+c_2Az), s_3(c_2(Axc_1+Ays_1)-s_2Az)+c_3(s_2(Axc_1+Ays_1)+c_2Az))
- θ_5=ATAN2(Oz,-Nz)

where $s_1=\sin\theta_1$,$c_1=\cos\theta_1$,...etc

The above inverse kinematic solutions can then be used to find the joints variables of the robot given the location of the end-effector or gripper.

Fig. 6. Inverse kinematics for a revolute robot.

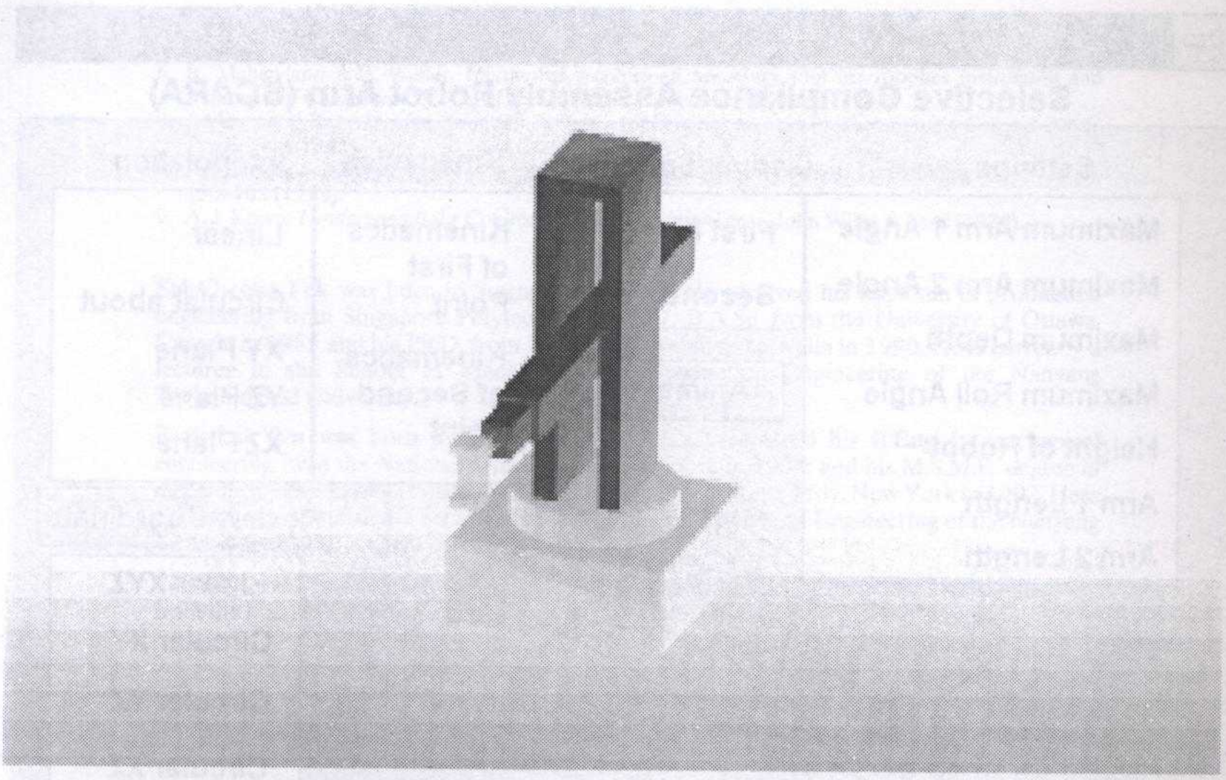


Fig. 7. Image for the cylindrical robot.

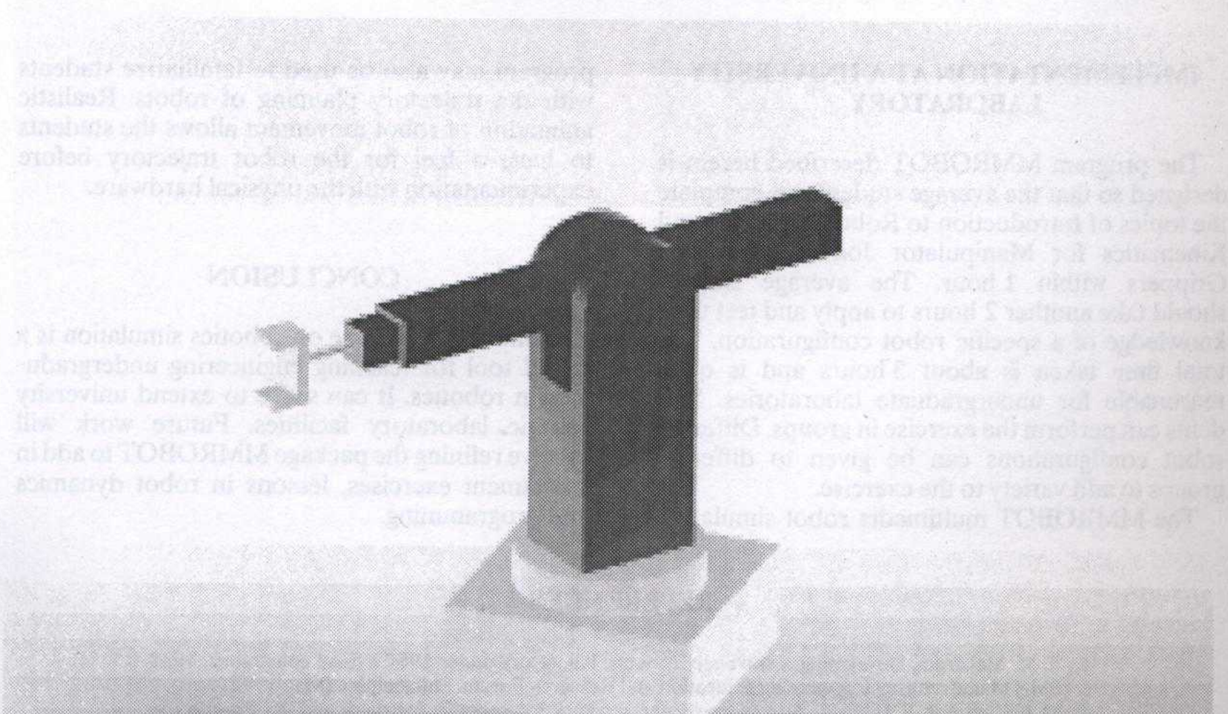


Fig. 8. Image for spherical robot.

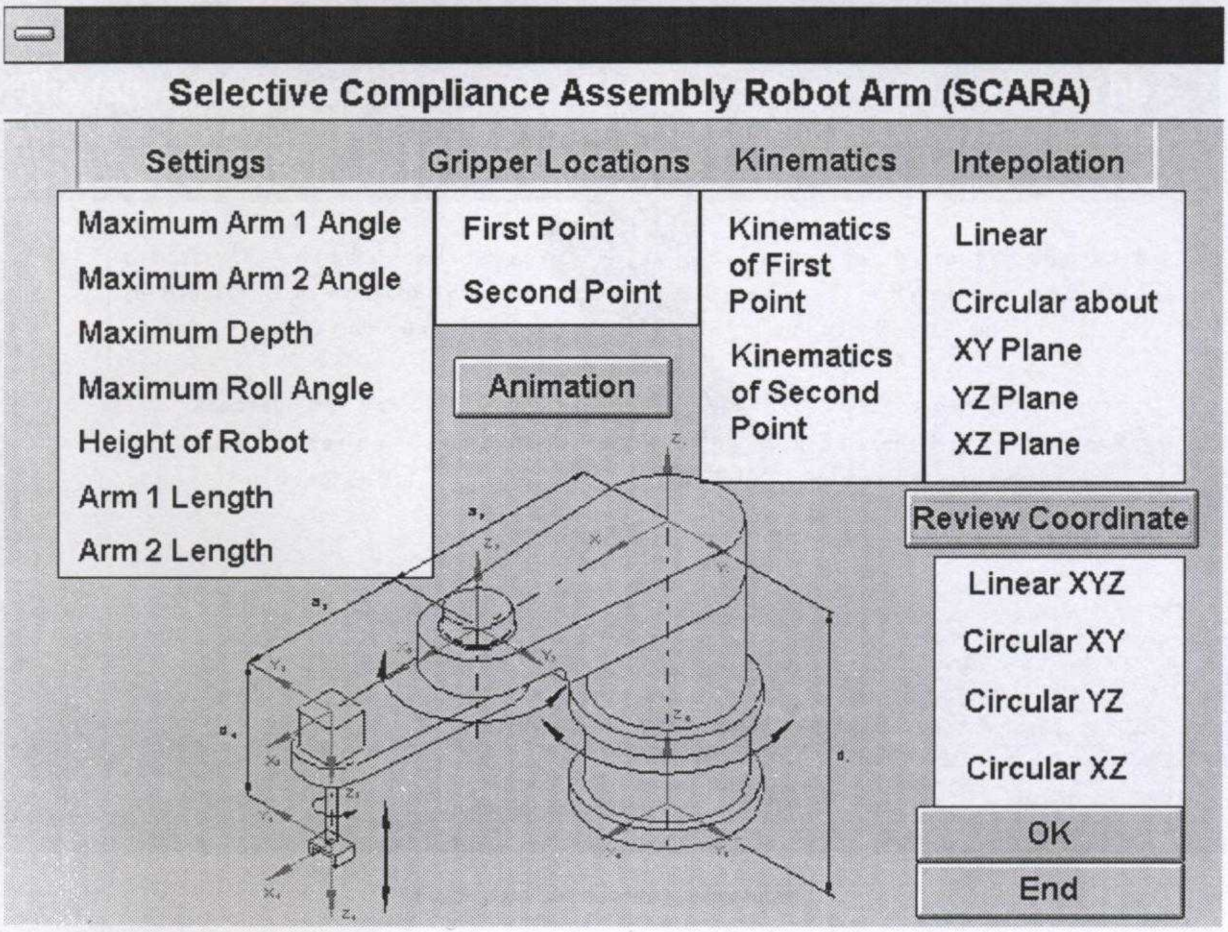


Fig. 9. Settings for the SCARA robot.

IMPLEMENTATION AT A UNIVERSITY LABORATORY

The program MMROBOT described herein is designed so that the average student can complete the topics of Introduction to Robotic Systems and Kinematics for Manipulator Joints, Links and Grippers within 1 hour. The average student should take another 2 hours to apply and test their knowledge of a specific robot configuration. The total time taken is about 3 hours and is quite reasonable for undergraduate laboratories. Students can perform the exercise in groups. Different robot configurations can be given to different groups to add variety to the exercise.

The MMROBOT multimedia robot simulation program may also be used to familiarize students with the trajectory planning of robots. Realistic animation of robot movement allows the students to have a feel for the robot trajectory before experimentation with the physical hardware.

CONCLUSION

Multimedia package on robotics simulation is a useful tool for teaching engineering undergraduates in robotics. It can serve to extend university robotic laboratory facilities. Future work will involve refining the package MMROBOT to add in assessment exercises, lessons in robot dynamics and programming.

REFERENCES

1. E. M. Malstrom, Developing a university robotics laboratory under 1980's fiscal constraints, 1982 SME Manufacturing Engineering Education and Research Forum, Philadelphia (May 1982).
2. M. C. Leu and S. H. Park, Use of computer graphics for robotics instruction, *1983 Frontiers in Education Conference Proceedings*, IEEE (1983).
3. B. I. Soroka, Debugging robot programs with simulator, Computer and Automated Systems Association of SME (1980).
4. C. A. Mulvenna and D. W. Fritz, An engineering computer graphics/computer aided design course, *3rd Canadian CAD/CAM and Robotics Conference*, Toronto (1984).

5. M. C. Leu and S. H. Park, Application of PADL-2 solid modeller to robot simulation, *3rd Canadian CAD/CAM and Robotics Conference*, Toronto (1984).
6. R. Mahan and S. E. Walter, Interactive graphics—a necessary tool for robotics instruction and training, *3rd Canadian CAD/CAM and Robotics Conference*, Toronto (1984).
7. J. Meyer and R. Jayaraman, Simulating robotic applications on a personal computer, *Comput. Mech. Engng* (July 1983).
8. A. J. Hejase and B. M. Hasbini, Computer-aided instruction in robotics, *Int. J. Engng Education*, **9**, 156–161 (1993).
9. A. J. Koivo, *Fundamentals for Control of Robotic Manipulators*, John Wiley & Sons (1989).

Sai-Cheong Fok was born in Singapore in 1957. He received his Diploma in production engineering from Singapore Polytechnic in 1977, B.A.Sc. from the University of Ottawa, Canada in 1985 and his Ph.D. from Monash University, Australia in 1990. He is currently a lecturer in the School of Mechanical and Production Engineering of the Nanyang Technological University.

Fock-Lai Tan was born in Singapore in 1959. He received his B.Eng. in mechanical engineering from the National University of Singapore in 1984; and his M.S.M.E. degree in mechanical engineering from Rensselaer Polytechnic Institute, Troy, New York in 1992. He is currently a lecturer in the School of Mechanical and Production Engineering of the Nanyang Technological University.

Patsy Ng was born in Singapore in 1969. She received her B.Eng. in mechanical engineering from the Nanyang Technological University in August 1994.