Int. J. Engng Ed. Vol. 9, No. 2, pp. 156–161, 1993
Printed in Great Britain.

0742–0269/91 $3.00+0.00

Computer-Aided Instruction in Robotics*

A. J. HEJASE
B. M. HASBINI
Electrical Engineering Department, American University of Beirut, Beirut, Lebanon

This paper presents an approach to increasing student interaction with robots via a replica of a robot manipulator. A user-friendly package is presented that so far has achieved the important goals of creating a complete dynamic animation of a robot manipulator for a computer user, and reducing the capital needed to equip a robotics laboratory with physical manipulators.

INTRODUCTION

DURING recent years, almost every engineering school has introduced robotics into its study programs in one form or another. It is a fact that industrial robots are replacing humans in many dirty, repetitive, tedious, difficult, dangerous and boring tasks in the industrial environment [1]. This field has generated more research in robotics, and consequently the need for engineers with preparation for this new area has increased. As usual, universities did not overlook this opportunity, but have introduced robotics courses since the early seventies. Today, many excellent textbooks cover the subject of robot manipulators, and the gamut of coverage extends from introductory concepts [2, 3, 4, 5, 6, 7] to more advanced and specialized theories and algorithms that deal with specific features of robots [8, 9, 10, 11].

Considering the set of robotics books whose main objective is to introduce the subject [2, 3, 4], it can be concluded that there are common important subjects that must be considered in any introductory treatment. Of these basic subjects the most important are direct kinematics (DR) and inverse kinematics (IR). Teaching experience has demonstrated that the domination of these two subjects together with their background and consequences, forms a firm basis in the process of learning about robotics. DK and IK are associated with the most important robotics concepts, such as frames, links, joints, transformations, axes, Jacobians, differential changes, and static point-to-point motion. Therefore most robotics instructors spend a reasonable amount of time teaching DK and IK during a robotics course. Moreover, most robotics curricula are associated with laboratory experiments where the student undergoes practical training, particularly in point-to-point positioning with DK and IK applications [12].

The fact that an industrial manipulator cannot be provided for every student has created difficulties in such laboratories; thus, the idea arose of using both real robot manipulators with their replicas via computer animation. The result is that a robotics laboratory will have a larger variety of equipment without much duplication of any unit. At the same time, when a student has practised on the real manipulator, he/she can complete their training and practical trials on an animated robot, available on any personal computer video monitor.

DESCRIPTION OF THE SIMULATED TASKS

The software basically deals with the problems of direct kinematics, inverse kinematics, and programming of the robot arm by using a teach-pendant. Direct or forward kinematics refers to the calculation of the position and orientation of the arm's tip or end-effector whenever a certain set of joint-values is given. The purpose is to identify how the gripper or tool is placed in space with respect to a reference coordinate frame. The result of such calculations is usually a 4 × 4 matrix known as an homogenous transformation [2, 3, 4].

Inverse kinematics, on the other hand, refers to the calculation of the joint-angles after examining the homogenous transformation that corresponds to the tip of the arm. This calculation is usually not unique, for an arm may reach a certain point with a certain orientation by taking different configurations and thus different joint-values. Similarly when the given homogeneous transformation corresponds to a reach-point that is outside the range of the robot, the calculation of the joint-values is not possible.

The last problem that the software emulates is that of positioning and orienting the tip of the robot arm by using a teach-pendant technique. Here the different joints are varied by direct control commands issued through the keys of the teaching board.

* Paper accepted 1 February 1993.

THE SOFTWARE

As Rhino robot arms [13] are the most commonly used training robots in educational institutes, it was decided to develop the simulation package around a Rhino architecture. Thus, the objective of the software is to achieve an exact scaled model of the Rhino arm that consists of a five-degrees-of-freedom manipulator mounted over an x–y sliding table. Figure 1 presents a diagram of the Rhino together with the assigned frames and joint-variables.

The software package is implemented under Turbo Pascal version 5.5 graphics utilities [14]. It can run on any IBM AT compatible system with an EGA or higher compatible graphics monitor (the existence of a co-processor will produce better results). The structure of the program consists of four modules called: **Rhino**, **Kinematics**, **Barunit**, and **Help**. The first three modules support the actions, commands, calculations, . . . and the initialization of Rhino controller; while the fourth module is available to provide the user with information that assists him/her in carrying out the simulation process.

THE PACKAGE LAYOUT

The program offers different functions in order to simplify the understanding and operation of the animated robot. Figures 2 to 7 show different options and windows that are offered. Each of these options may be selected from the main menu shown in Fig. 2 where full assistance is provided to smoothly guide the user through the different stages that the software offers. Figure 3 shows the arm with the joint variables and options available

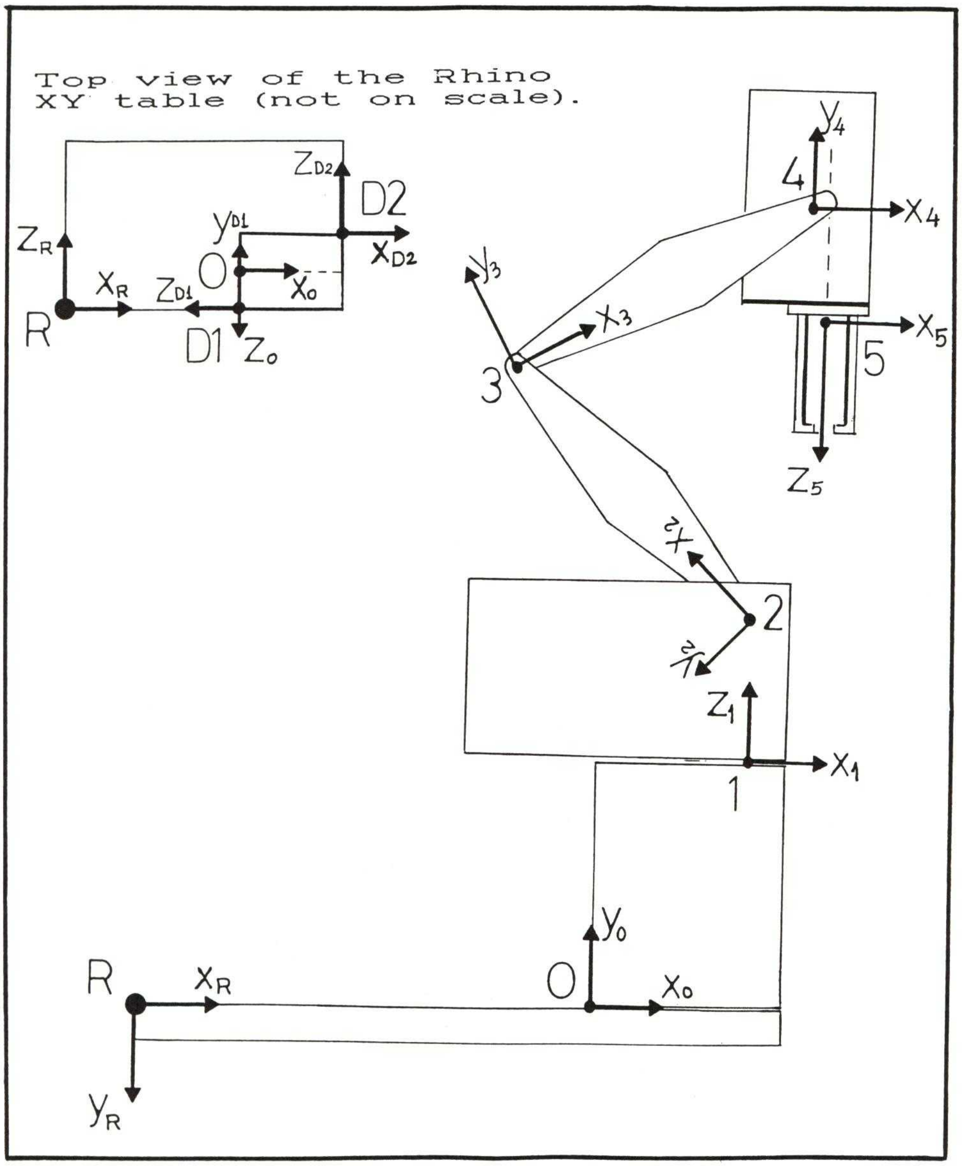

Fig. 1. Frames of the Rhino arm.

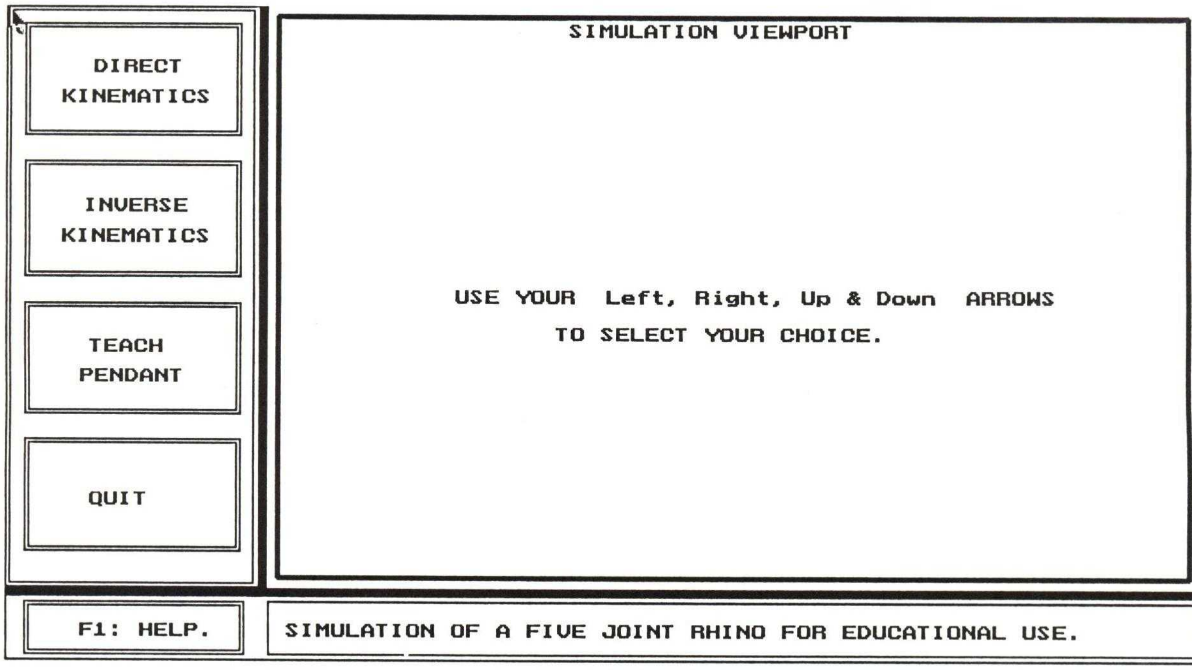

Fig. 2. The user's menu.

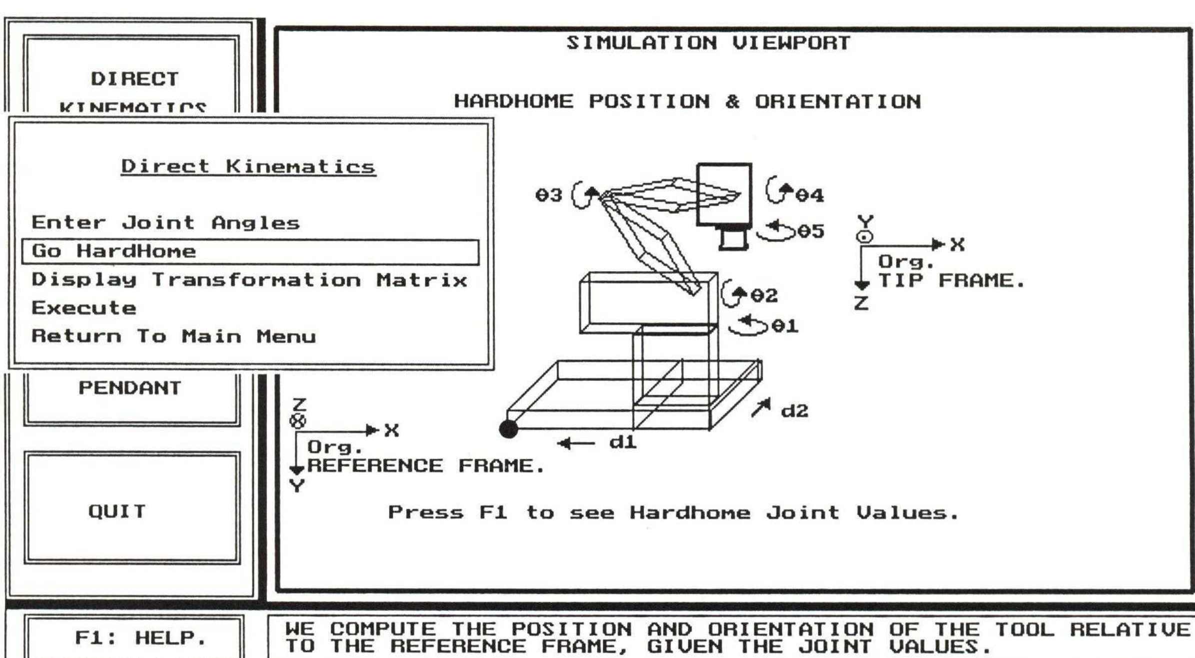

Fig. 3. Selecting the **Direct Kinematics** option.

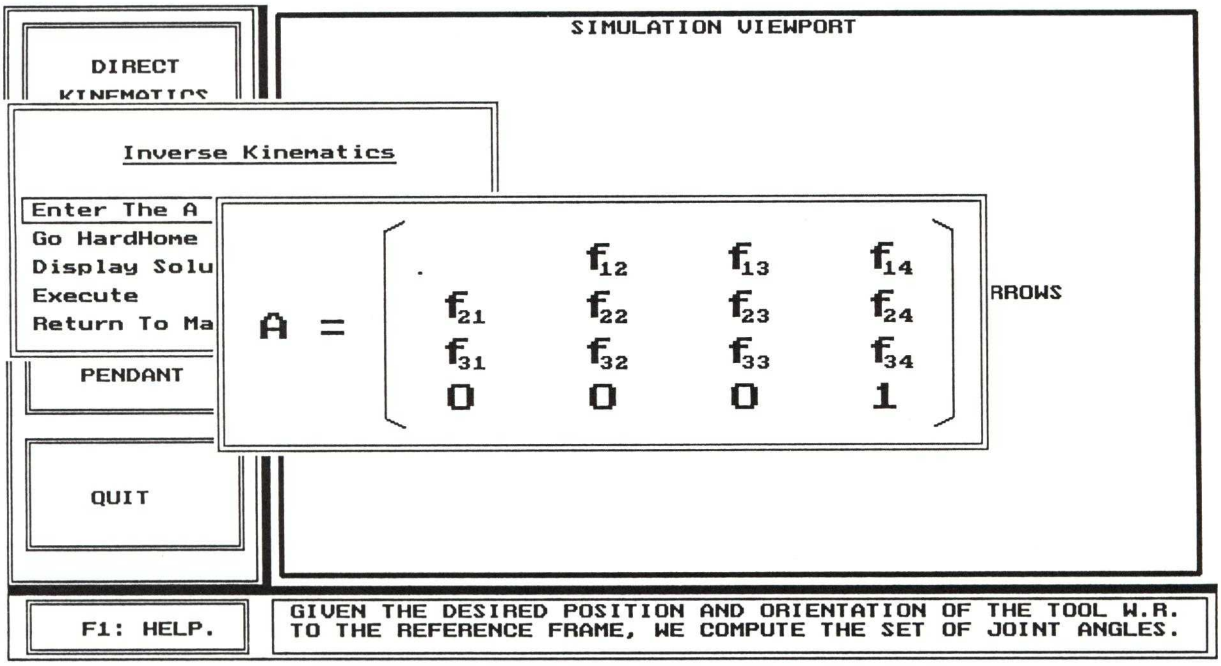

Fig. 4. Selecting the **Inverse Kinematics** option.

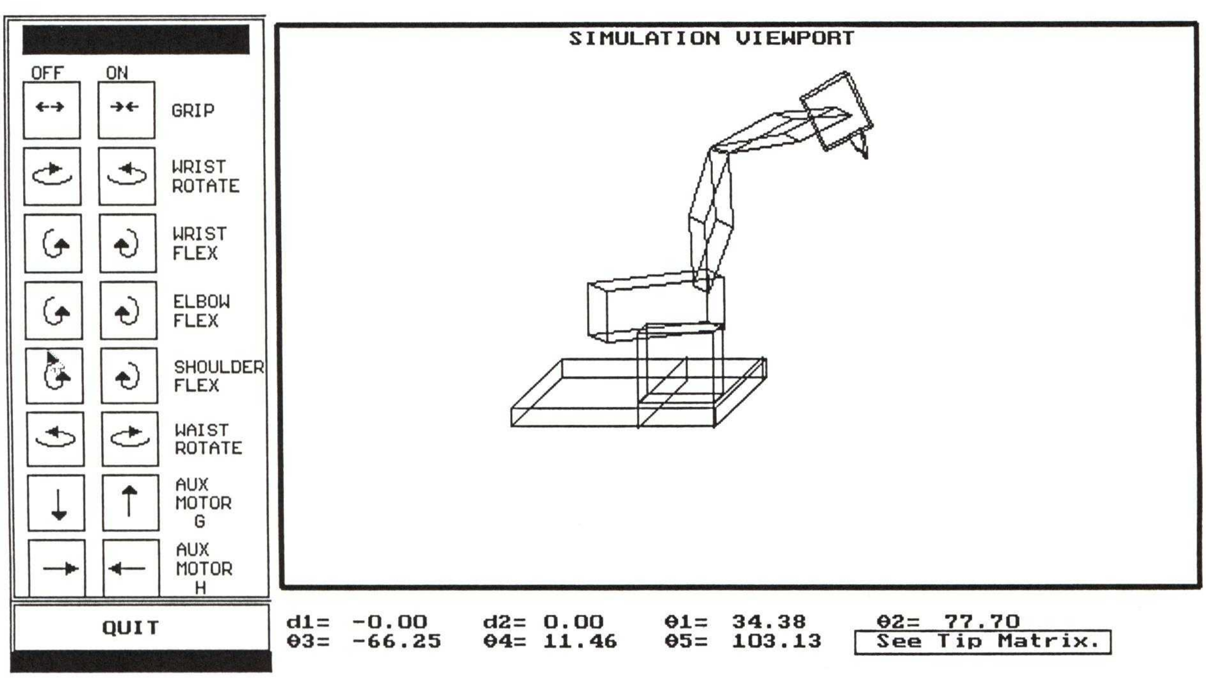

Fig. 5. The **teach-pendant** showing a configuration.

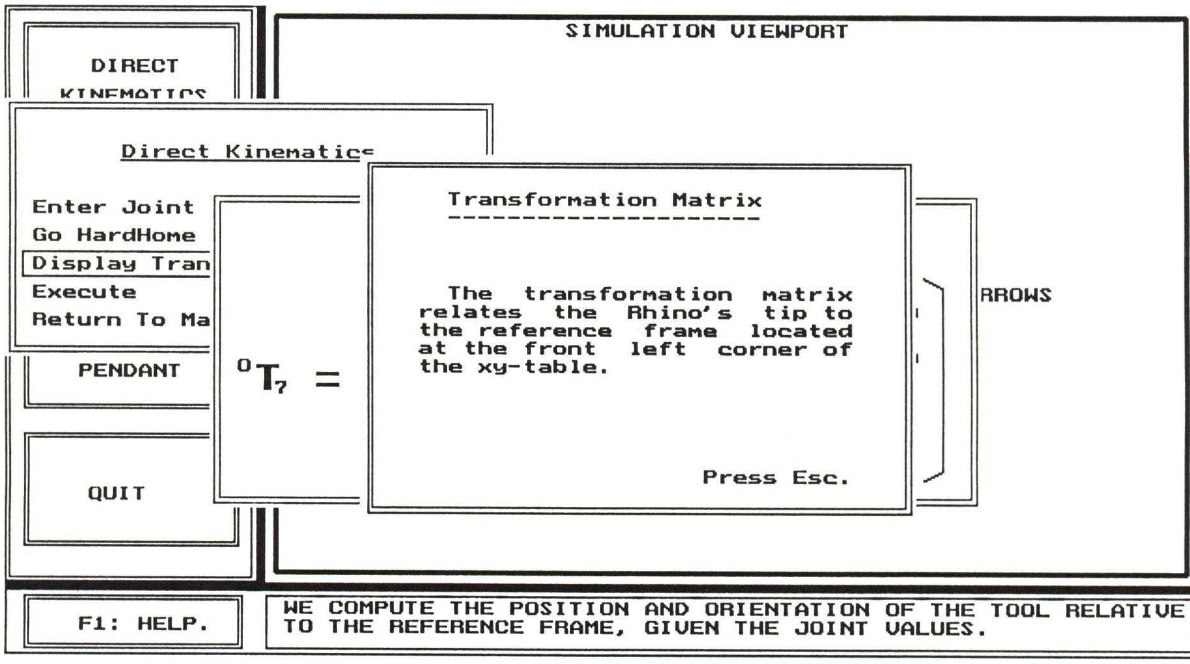

Fig. 6. Static help which relays explanations.

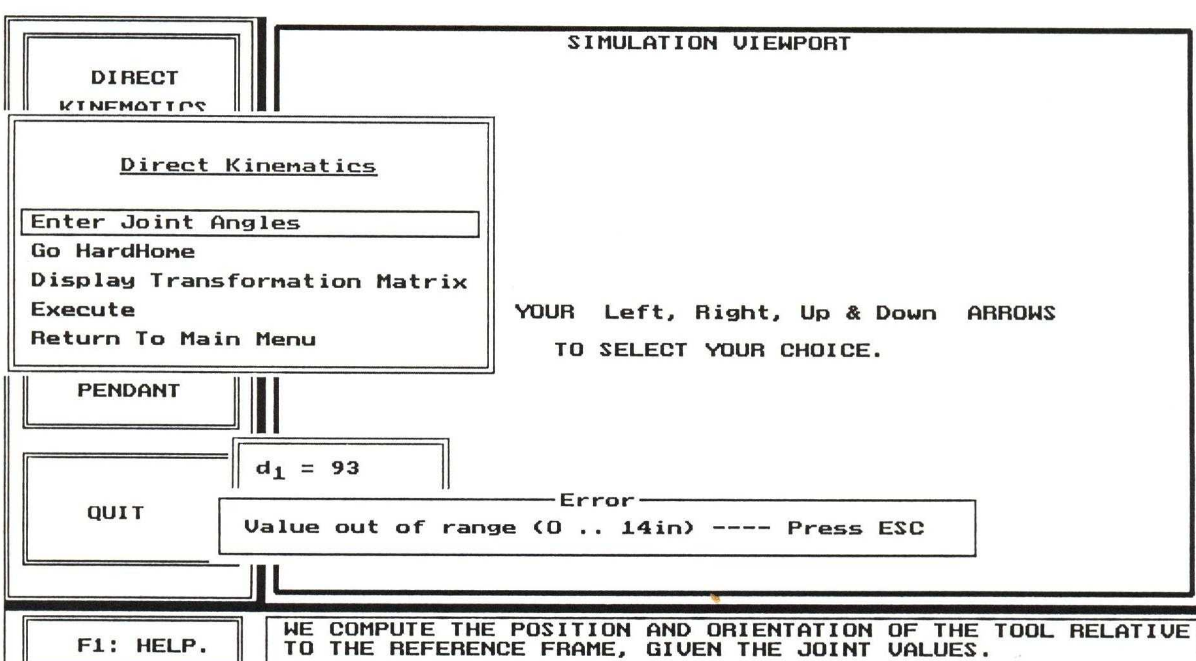

Fig. 7. Dynamic help which warns the user of an improper input.

when selecting **Direct Kinematics** in the main menu; here the homogenous transformation corresponding to any set of joint-values can be calculated and realized with the arm. The student can follow the arm's motion as it configures itself to comply with the desired transformation. A good feature is that at the bottom of the viewport, the user can always find the necessary instructions for performing the job successfully.

Figure 4 shows the **Inverse Kinematics** option; here the user may enter a homogenous transformation to verify and detect its corresponding joint-values which could be not unique. Again a great amount of help is provided to overcome any difficulties due to errors or impossible cases.

Figure 5 is an example of point-to-point motion by using the teach-pendant; at the bottom of the viewport the joint-values are repeated and updated successively as they undergo changes.

Figures 6 and 7 are examples of the static and dynamic help provided by the software. Static help includes the kind of information that can be found in printed manuals which is accessed by pressing the F1 key. Dynamic help involves automatic interaction with the user as shown in Fig. 7 where the user is warned that a joint-value out of range has been entered.

CONCLUSION

By giving the student of robotics the chance of interacting more frequently with animated robots, a more powerful understanding of robotics concepts is achieved. Such achievement is acquired without having to secure huge investments for teaching robot units.

Acknowledgements—The authors would like to thank B. Ghoussaini, R. Kadi, and A. Zobian for their efforts in developing the initial version of this software.

REFERENCES

1. R. C. Dorf, *International Encyclopedia of Robotics*, Wiley (1988).
2. R. P. Paul, *Robot Manipulators*, MIT Press (1981).
3. J. J. Craig, *Introduction to Robotics*, 2nd edn, Addison Wesley (1989).
4. M. Shahinpoor, *A Robot Engineering Textbook*, McGraw Hill (1989).
5. P. Coiffet and M. Chirouze, *An Introduction to Robot Technology*, McGraw Hill (1982).
6. A. J. Koivo, *Fundamentals for Control of Robotic Manipulators*, Wiley (1989).
7. M. Brady, et al., *Robot Motion*, MIT Press (1982).
8. M. W. Spong and M. Vidyasager, *Robot Dynamics and Control*, Wiley (1989).
9. A. C. Staugaard, *Robotics and AI*, Prentice-Hall (1987).
10. K. Warwick and A. Pugh, *Robot Control*, Peter Peregrinus (1988).
11. C. H. An, C. C. Atkeson and J. M. Hollerbach, *Model-Bawed Control of a Robot Manipulator*, MIT Press (1988).
12. A. J. Hejase and B. M. Hasbini, *An Educational Platform for Robotic Manipulators*, 34th Midwest Symposium on Circuits and Systems, Monterey, CA. (1991).
13. Rhino Mark III, Owner's Manual, Rhino Robots Inc. (1988).
14. Turbo Pascal, Manual, Borland International Inc. (1987).

Ale J. Hejase was born in Torreon, Mexico in 1956. He received a B.Eng. in electronics from Tecnologico de la Laguna, Mexico in 1980; his M. Eng. degree in control systems from Instituto Tecnologico y de Estudios Superiores de Monterrey, Mexico in 1982; and his Ph.D. in Electrical Engineering from Syracuse University, Syracuse, New York in 1987. He is currently an assistant Professor in the Electrical Engineering Department of the American University of Beirut. A. J. Hejase is now the Technical Manager of Future Publishers S.A.L. in Beirut-Lebanon (P.O. Box 11-236/3294).

Badr M. Hasbini was born in Beirut in 1969. He received his B.Eng. in computers and communications from the American University of Beirut, Lebanon in 1990. He is currently one of the owners and manager of the software and hardware company Analog Digital Systems S.A.R.L. located in Lebanon.