

ROBLIB: AN EDUCATIONAL PROGRAM FOR ROBOTICS

N. M. Fonseca Ferreira, J.A. Tenreiro Machado

Institute of Engineering of Coimbra, Dept. of Electrical Engineering,
Quinta da Nora, 3031-601 Coimbra Codex, Portugal,
tel: 351 239790200, fax: 351 239790220, e-mail: nunomig@isec.pt
Institute of Engineering of Porto, Dept. of Electrical Engineering,
Rua Dr António Bernardino de Almeida, 4200-072 Porto Codex, Portugal,
tel: 351 228340500, fax: 351 228321159, e-mail: jtm@dee.isep.ipp.pt

Abstract: The paper presents a program for robotics education that runs on standard PCs under the Microsoft Windows environment. The *RobLib* package is designed for undergraduate students and emphasis the fundamental aspects of robot modelling and control. The software is self-explanatory and uses menus, dialog boxes with figures and context-dependent on-line help. In this perspective, students are motivated to investigate on the workspace, kinematics, dynamics, trajectory planning, position and force control of manipulators. Based on this first experience, further studies on robotics, using more sophisticated packages and concepts, are, then, more attractive from the students point of view. *Copyright ©2000 IFAC*

Keywords: Robots, manipulators, mechatronics, software, education.

1. INTRODUCTION

In the last two decades robotics became a common subject in courses of electrical, computer, control and mechanical engineering. Progress in scientific research and developments on industrial applications lead to the appearance of educational programs on robotics, covering a wide range of aspects such as kinematics and dynamics, control, programming, sensors, artificial intelligence, simulation and mechanical design. Nevertheless, courses on robotics require laboratories having sophisticated equipment, which pose problems of funding and maintenance.

Computer-based education is a well-established alternative that overcomes some of the referred problems. In this line of thought, university teams, see (S. Derby, 1986; R.B. White and T. Raz, 1989; R.E. Parkin and A.M. Eydgahi, 1991) and software houses developed robot simulation and programming packages such as (WorkSpace, 1989; SYMORO, 1990) running on inexpensive computers. Nevertheless, often, computer programs emphasise capabilities such as the 3D graphical simulation and the programming language and give a smaller importance to mathematical aspects of modelling and control, see (J.A. Tenreiro Machado and

Alexandra M. Galhano, 1993, 1995). To cover this area, some researchers proposed the adoption of symbolic packages, see (J. Lloyd, 1988; M.C. Leu and Y.S. Wang, 1990; N. Vira, 1992; J.F. Nethery and M.W. Spong, 1994) to illustrate robot kinematics and dynamics. However, undergraduate students with no prior experience may feel difficulties in getting into the robotics experiments before overcoming the symbolic packages procedures and commands. This state of affairs, motivated the development of a computer program highlighting the fundamentals of robot mechanics and control. Given the popularity of Microsoft Windows it was decided to implement the software in this environment using the Borland Delphi language. The project lead to the *RobLib* program which was adopted as an educational tool in a first course on robotics. This paper introduces the package and discusses the related educational aspects.

The paper is organised as follows. Section two, presents a general overview of the software and describes its main features and commands. Section three guides us through a typical session for exploring *RobLib*. Based on the results, section four discusses future software enhancements. Finally, section five outlines the main conclusions.

2. OVERVIEW OF THE PROGRAM

The *RobLib* educational package was designed to take full advantage of the Windows environment. All the commands and the required parameters are entered through pull-down menus and dialog boxes. The software is intended to be self-explanatory to the extent possible to encourage students exploring the program. For the same purpose, help menus are available throughout the different windows. Several dialog boxes include figures to clarify context-dependent definitions. Experiments are restricted to planar manipulators that is, with two degrees of freedom (dof). This strategy intends to decrement the 'weight' of the graphical simulation and to give importance to the mathematical aspects of kinematics, dynamics and control, while maintaining the program at an introductory level. Moreover, this option allows the use of inexpensive PCs because of the low computational requirements. On the other hand, numerical data files and graphical data files have formats compatible with other Windows applications to allow data interchange and external data calculation.

As usual in Windows programs the top of the screen contains the main menu bar. The top bar includes the entries: File, Planning, Model, Tools, Control and Help. Figure 1 shows the selections available on each menu. These commands and the related actions are further explained in the next session.

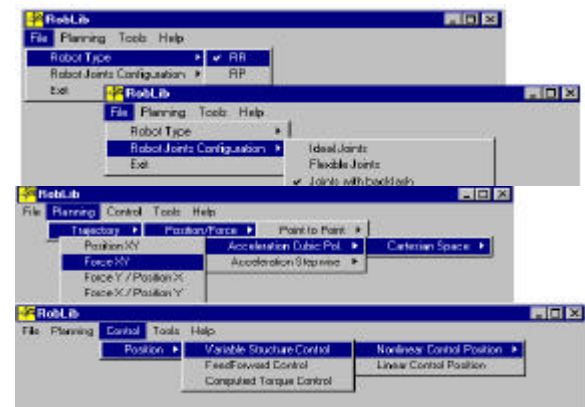


Fig. 1. Selections available in the main menu.

3. A TYPICAL CLASSROOM SESSION

In order to demonstrate the capabilities of the package the paper shall now follow a typical classroom session. The tour begins with the definition of the type of robot within two possibilities: the RP and RR planar structures (R-Rotational, P-Prismatic, axis) and the dynamical phenomena at the robot joints such as flexibility, backlash, motor saturation, non-linear friction and signal quantification. After selecting the robot type and dynamical phenomena in robot joints, *RobLib* presents a dialog box (Figure 2) for entering the robot numerical parameters, namely the joint limits, the link lengths and masses and the actuator inertias. Later on, these numerical parameters can be viewed/changed and saved.

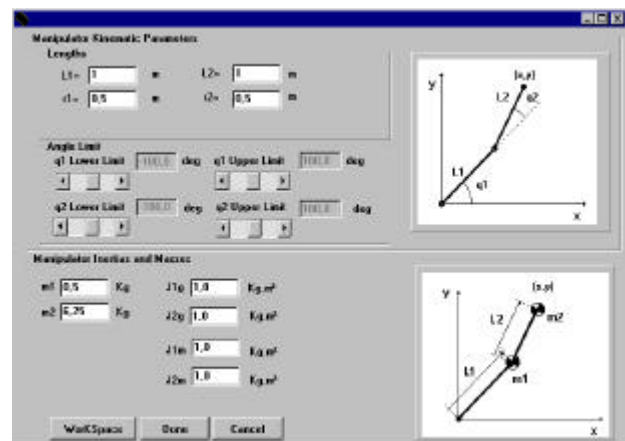


Fig. 2. Setup of a robot structure.

The second phase of the session consists on the definition of the robot trajectory. The option Trajectory allows to define the straight line trajectory planning based on the definition of the initial and final points. This planning can be done in both operational and joint spaces and requires the definition of parameters such as the time for the simulation and the trajectory, the initial points and the sampling frequency (Figure 3).

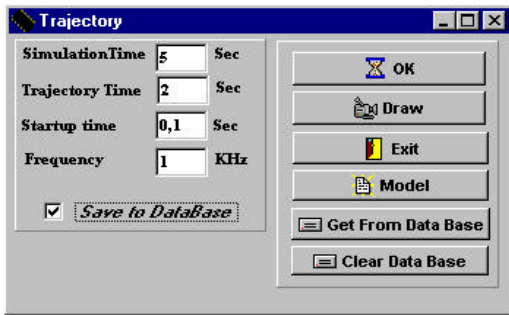


Fig. 3. Definition of a robot trajectory.

The trajectory planning involving the variables position and force presents some differences. In that perspective, we consider the separation of the two possibilities in the option Planning. When planning the force we can configure the environment in both axis through the button 'Model' (Figure 4). The environment is modelled through a linear system with a spring, a mass and a viscous friction. The surface model in the axis x_c and y_c directions is accomplished through the constants M_i , B_i and K_i ($i=1,2$), while the variable θ allows the definition of the trajectory slope.

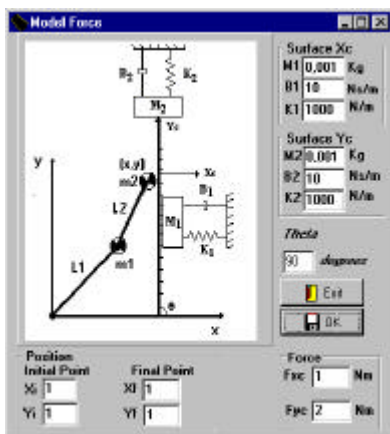


Fig. 4. Definition of a robot trajectory and the contact surface.

We have also the possibility of performing the robot motion animation (Figure 5) as well as to save the trajectory in an integrated database (Figure 6).

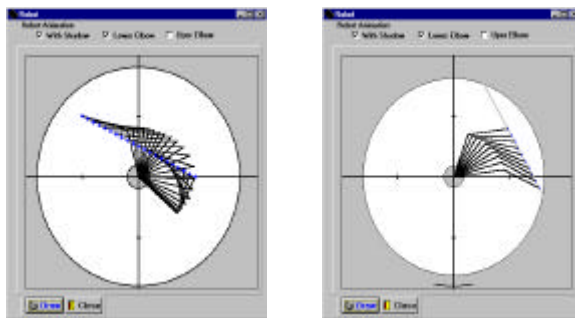


Fig. 5. Robot animations in the operational and joint spaces.

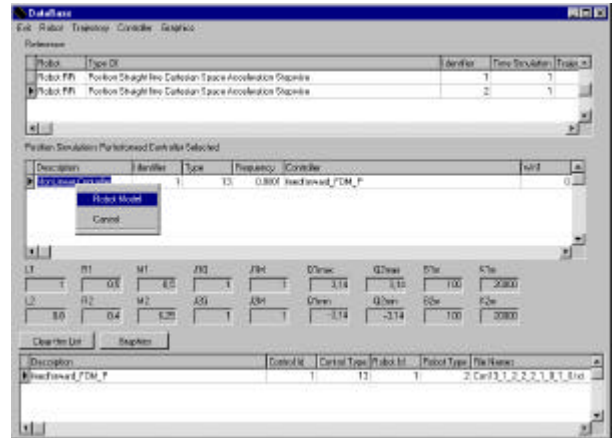


Fig. 6. The database integrated in the RobLib.

The database included in the *RobLib* simulator allows the user to search for previous simulations. In this way the user can compare experiments, namely for each type of controller and dynamical phenomena.

The *RobLib* allows the user to view all the trajectories variables. These charts depict the temporal evolution in the joints variables in both in the operational and joint spaces (Figure 7). The 'Print' button performs a hardcopy of the active window and the 'Save' button enables saving files in BMP format. Moreover, the 'Zoom in' and 'Zoom out' buttons allow to maximise/minimise the graphic size while there is also the possibility of zooming one specific area of the graphic.

The simulation data is stored in files, available for later treatment and analysis through other programmes. The student can analyse the non-linear effect of the kinematics in the interpolation method. During the calculation, a graphical form (Figure 7) depicts the successive interpolation points, both in the cartesian and joint spaces.

After defining the trajectory, the 'Inverse Dynamics' option is activated and the time evolution of the corresponding joint driving torques can be viewed through the 'Graphics' option in the 'Tools' menu. In the first case, the amplitude *versus* time evolution of the forces/torques can be taught as a piecewise linear function approximation whose points are inserted/deleted through the mouse. In the second case, the numerical data can be loaded from an external software application.

The third phase of the session consists on selecting the 'Graphics' on the 'Tool' menu. In this case the student can view (Figure 7) the evolution (both in the time and the phase plane domains) of the variables. During the rest of the session the kinematic, dynamic and control variables can be viewed, printed and compared through the Graphics option.

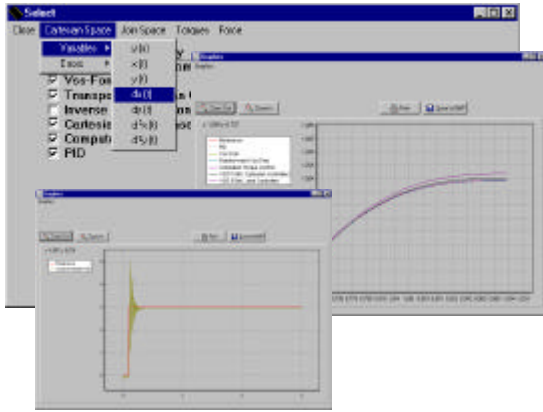


Fig. 7. Using the Graphics selection to view the time evolution of positions, velocities and accelerations in the operational and joint spaces.

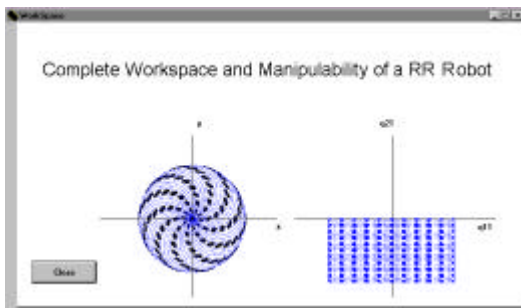


Fig. 8. The Tools integrated in the RobLib for analysing the Workspace and the Manipulability.

The Tools menu includes other entries such as 'Workspace' and 'Manipulability' for the analysis of the robot properties (Figure 8). This tool also shows the manipulability in the workspace of one or two robots working in cooperation (Figure 9). The manipulability varies in the workspace, that is $m = m(x, y)$; therefore, we consider some indices to simplify the study of the manipulability of several arms (Figure 10).

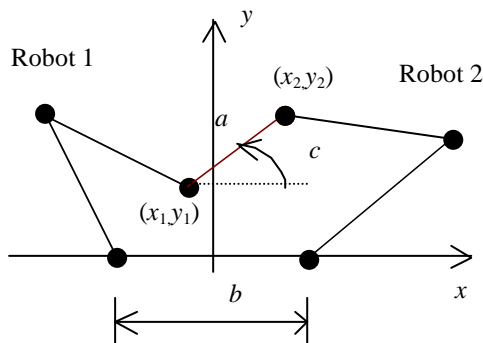


Fig. 9. Two RR cooperating robots for the manipulation of an object with length a and orientation c .

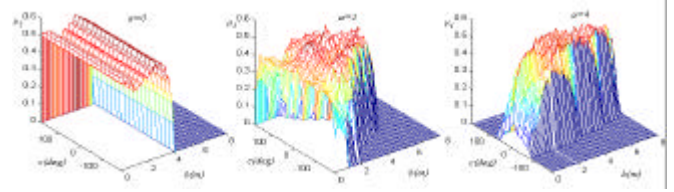
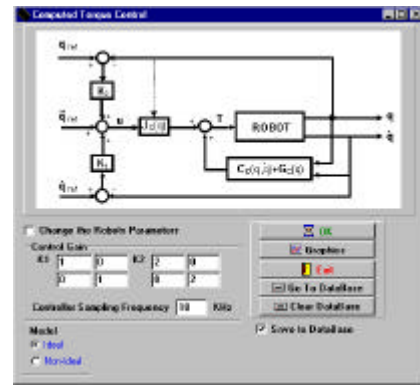
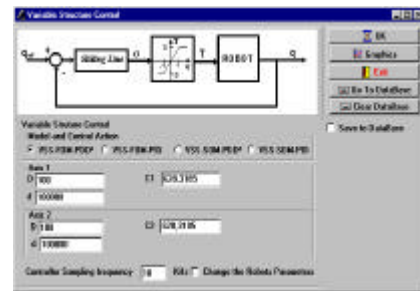


Fig. 10 – The maximum manipulability m versus b and c for object lengths $a = \{0, 2, 4\}$, with $m = 1000$, $n = 1000$, $r = 0.1$ rad, Robot 1 = Robot 2: $\{l_1 = l_2 = 1 \text{ m}\}$.

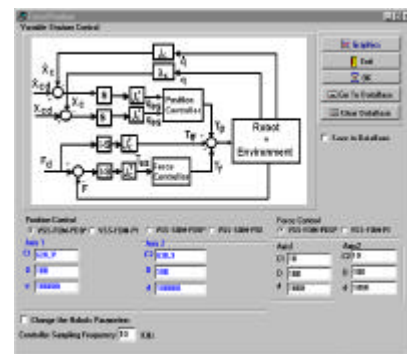
The Control (Figure 11) menu allows the choice of several controllers. In this case, the user can investigate the effects of a finite sampling frequency, distinct gain tuning and imperfect modelling. Moreover, the student can easily experiment and compare the performances of different algorithms.



a)



b)



c)

Fig. 11. Controllers a) Computed torque algorithm, b) Variable structure algorithm, c) Hybrid controller.

The figure consists of four subplots arranged vertically, all sharing a common x-axis representing Time in seconds (sec) from 0 to 4.

- Top Plot:** Position x (m) vs Time (sec). The curve starts at $x = -1$ m at $t = 0$ s, rises smoothly, crosses $x = 0$ m at $t = 1$ s, and reaches $x = 1$ m at $t = 2$ s, where it remains constant.
- Second Plot:** Position y (m) vs Time (sec). This is a step function that is 0 until $t = 0$ s, where it jumps to 1 m and remains constant.
- Third Plot:** Torque T_1 (Nm) vs Time (sec). The plot shows a sharp negative spike at $t = 0$ s, a sharp positive spike at $t = 1$ s, and then settles to a steady-state value of 40 Nm for $t > 2$ s.
- Bottom Plot:** Torque T_2 (Nm) vs Time (sec). The plot shows a sharp negative spike at $t = 0$ s, a sharp positive spike at $t = 1$ s, and then settles to a steady-state value of 25 Nm for $t > 2$ s.

In another experiment we analyse presents the position and force control. In this case we simulate a robot with

The figure consists of three vertically stacked plots sharing a common x-axis representing Time in seconds, ranging from 0 to 4.

- Top Plot:** The y-axis is labeled $x \text{ (m)}$ and ranges from 0.99328 to 1.04448. It shows a blue signal that starts at approximately 1.0035, exhibits high-frequency oscillations between 0.99328 and 1.04448 for the first second, and then settles into a lower-frequency oscillation around 1.03. A red horizontal line is drawn at $y = 1.0035$.
- Middle Plot:** The y-axis is labeled $y \text{ (m)}$ and ranges from -1 to 1. It shows a blue curve that starts at 1, decreases smoothly, crosses 0 at approximately 0.8 seconds, and reaches -1 at approximately 2 seconds, where it remains constant. A red horizontal line is drawn at $y = 0$.
- Bottom Plot:** The y-axis is labeled $F \text{ (N)}$ and ranges from 0 to 5. It shows a blue signal that starts at 0, jumps to approximately 3, and then exhibits high-frequency oscillations between approximately 2 and 4. A red horizontal line is drawn at $y = 3$.

At any time during the worksession, users can get on-line help in each form. Nevertheless, *RobLib* has a separate Help menu where additional guidelines can be found. A Tutorial option (Figure 14) allows the study of the main formulae and definitions adopted in the package.

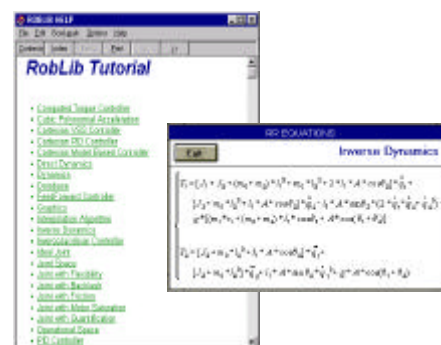


Fig. 14. Using the option Tutorial in the Help menu.

4. FUTURE SOFTWARE ENHANCEMENTS

RobLib is planned to include several additional educational features. In the robot definition, selection on structures such as counterweights and parallelogram linkages will allow the study of compensation schemes leading to a new dynamics (J.A. Tenreiro Machado, 1997). The present version of *RobLib* includes an entry for the option force control algorithms and, consequently, it is being implemented the 'impedance architecture' for the force/position controller.

RobLib has not yet available the toolbox that includes an entry for the analysis of different methods of calculation of the inverse dynamics (J.A. Tenreiro Machado, 1995). This perspective, will allow the student to investigate numerical recursive methods such as the Lagrange-Euler, the Newton-Euler and the adoption of symbolic formulae. The analysis of the computational load posed by the different algorithms will motivate students for the aspects of software optimisation for real-time applications.

An important educational aspect consists on the students' introduction to *RobLib*. In this line of thought, a Internet AVI video has been prepared for the teacher use and a manual of exercises experiments with the *RobLib* is presently being developed.

5. CONCLUSIONS

A program for robotics education has been presented. *RobLib* is a package that runs on standard PCs under the Microsoft Windows environment. The program is designed to be used at an introductory level for undergraduate students and emphasises the aspects of robot mathematical modelling and control. The software is self-explanatory and uses menus, dialog boxes with figures and context-dependent on-line help. In this perspective, students are motivated to investigate on the kinematics, dynamics, trajectory planning and control in order to gain insight in robotics, before getting into expensive laboratory experiments with real robots.

Those interested in obtaining the *RobLib* package should send their request to the authors.

REFERENCES

- Derby, S. (1986). In Position: Simulating Robotic Workcells on a Micro, *Computers in Mechanical Engineering*, pp. 34-37.
- Eydgahi, A.M. and J.J. Sheehan (1991). A Computer Animation of Robotic Manipulators and Workcells, *IEEE Control Systems*, vol. 11, n. 4, pp. 56-59.
- Leu, M.C., Y.S. Wang (1990). Studying Robot Kinematics and Dynamics with the Aid of MATHEMATICA, *Int. J. of Mechanical Engineering Education*, vol. 19, n. 3, pp. 213-228.
- Lloyd, J., V. Hayward (1988). Kinematics of Common Industrial Robots, *Robotics and Autonomous Systems*, vol. 4, no. 2, pp. 169-191.
- Megahed, S.M., (1993). Principles of Robot Modelling and Simulation, *John Wiley and Sons*.
- Nethery, J.F. and M.W. Spong, (1994). Robotica: A Mathematica Package for Robot Analysis. *IEEE Robotics & Automation Magazine*, vol. 1, no. 1, pp. 13-20.
- Parkin, R.E. (1991). An Interactive Robotic Simulation Package. *Simulation*, vol. 56, no. 5, pp. 337-345.
- Raz, T.(1989). Graphics Robot Simulator for Teaching Introductory Robotics. *IEEE Trans. on Education*, vol. 32, no. 2, pp. 153-159.
- SYMORO (1990). Software Package of SYmbolic MOdelling of Robots. *Rapid Data Ltd*.
- Tenreiro, J.A. Machado, J.L. Martins de Carvalho and Alexandra M. Galhano, (1993). Analysis of Robot Dynamics and Compensation Using Classical and Computed Torque Techniques. *IEEE Trans. on Education*, vol. 36, no. 4, pp. 372-379.
- Tenreiro, J.A. Machado and Alexandra M. Galhano, (1995). Benchmarking Computer Systems for Robot Control. *IEEE Trans. on Education*, vol. 38, no. 3, pp. 205-210.
- Tenreiro, J.A. Machado and Alexandra M. Galhano, (1997). WinRob: A Educational Program for Robotics. *Int. Journal of Electrical Engineering Education*, vol. 34, no. 1, pp. 37-47.
- Vira, N., E. Tunstel, (1992). Use of Symbolic Computation in Robotics Education. *IEEE Trans. on Education*, vol. 35, no. 1, pp. 18-30.
- White, R.B., R.K. Read, M.C. Koch and R.J. Schilling, (1989). A Graphics Simulator for a RoboticArm. *IEEE Trans. on Education*, vol. 32, no. 4, pp. 417-429.
- WorkSpace (1989). Manual, Robot Simulations, Newcastle-Upon-Tyne, England.

