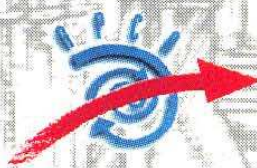


CONTROLO'2000

4th Portuguese Conference on Automatic Control

Conference Proceedings

Guimarães PORTUGAL
October 4 - 6, 2000



ISBN 972-98603-0-0

TEACHING ROBOT MODELLING AND CONTROL WITH *ROBLIB*

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

Institute of Engineering of Coimbra, Dept. of Electrical Engineering,
Quinta da Nora, Apartado 10057, 3031-601 Coimbra Codex, Portugal,
tel: 351 39 790200, fax: 351 39 790270, 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 2 8340500, fax: 351 2 8321159, 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 Controlo2000*

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

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 (S. Derby, 1986; R.B. White, 1989; R.E. Parkin and A.M.

Eydgahi, 1991) and software houses developed robot simulation and programming packages (WorkSpace, 1997; 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, (J.A. Tenreiro Machado and Alexandra M. Galhano, 1993, 1995). To cover this area, some researchers proposed the adoption of symbolic packages, (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 both basics and advanced 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 develops a numerical method for analyzing the manipulability of robotic systems. Based on the new algorithm section four studies the manipulation performance of one-arm and two-arm systems. 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.

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 1) 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.

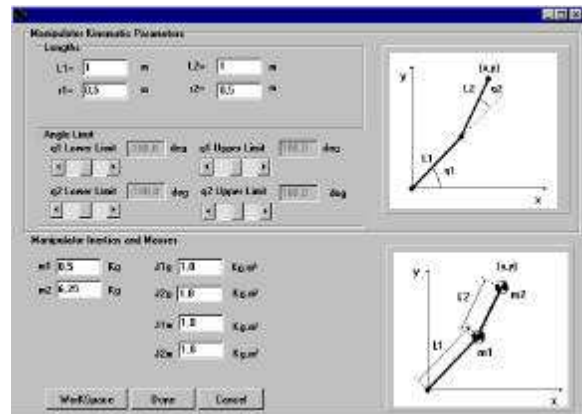


Fig. 1. Setup of a robot structure.

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.

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 2). 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.

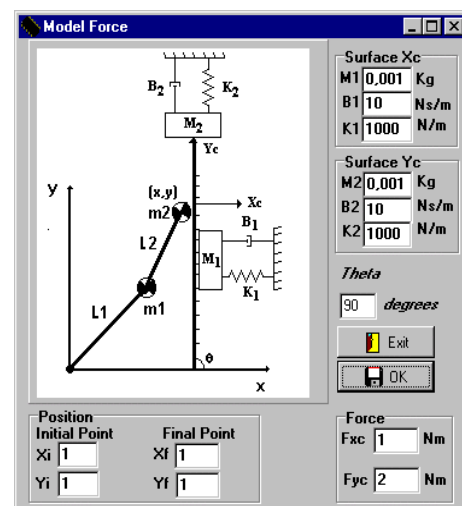


Fig. 2. Definition of a robot trajectory and the contact surface.

We have also the possibility of performing the robot motion animation as well as to save the trajectory in a integrated database (Figure 3).

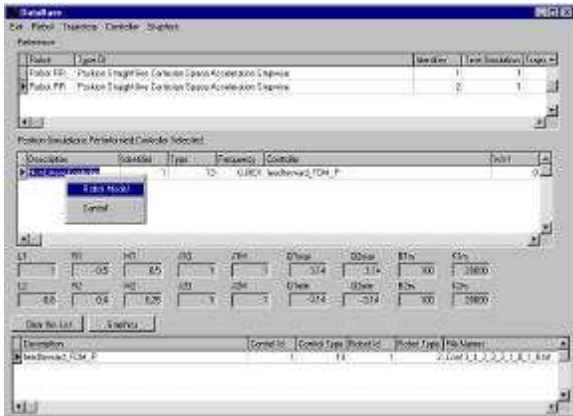


Fig. 3. The database integrated in the *RobLib*.

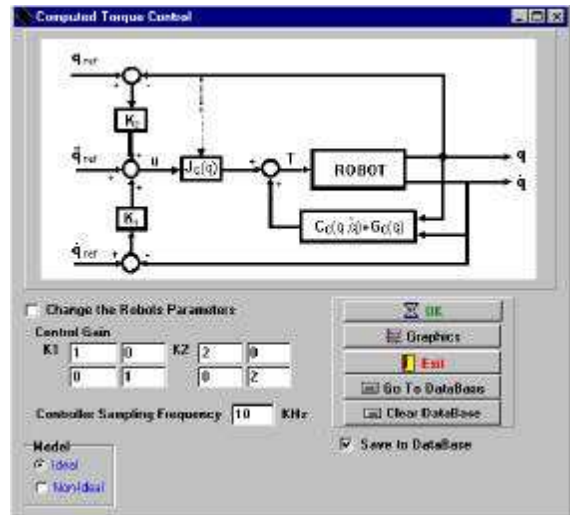
The database included in the *RobLib* simulator allows the user to search for previous simulations. In this way the user can compare experiments, namely for each type of controller and dynamical phenomena.

The *RobLib* allows the user to view all the trajectories variables. These charts depict the temporal evolution the joints variables both in the operational and joint spaces. 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.

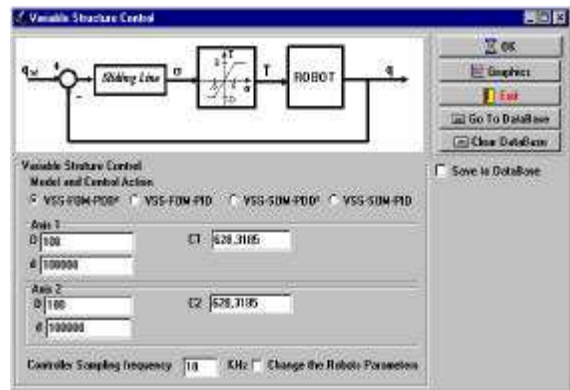
After defining the trajectory, the 'Inverse Dynamics' option is activated and the time evolution of the corresponding joint driving torques can be viewed through the 'Graphics' option in the 'Tools' menu. In the first case, the amplitude *versus* time evolution of the forces/torques can be taught as 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. With 'Graphics' the student can view 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.

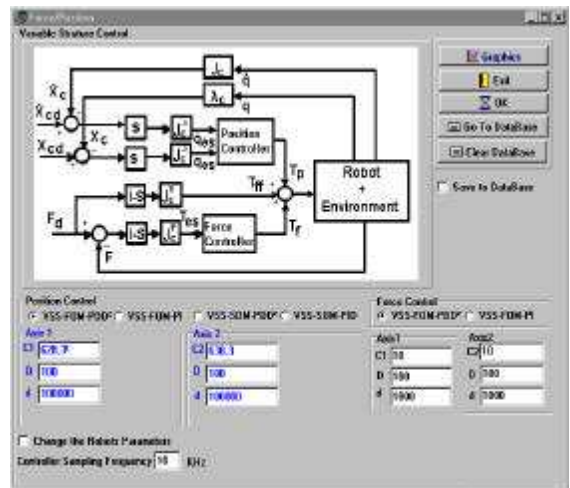
The Control (Figure 4) menu allows the choice of several controllers. In this case, the user can investigate the effects of a finite sampling frequency, distinct gain tuning and imperfect modelling. Moreover, the student can easily experiment and compare the performances of different algorithms in both spaces such as the PID controllers, the computed torque algorithm, the feedforward algorithm, the variable structure and, for position/force control, the hybrid controller.



a)



b)



c)

Fig. 4. Controller a) computed torque algorithm, b) variable structure, c) Hybrid controller

All simulated variables can be viewed namely, the positions, velocities, accelerations, errors and torques. For example, in a simple experiment, figure 5 shows the hand position of an ideal robot using the variable

structure controller with $\omega_n=100 \text{ rads}^{-1}$, $\zeta=1$ and a sampling frequency $f_c=10\text{kHz}$.

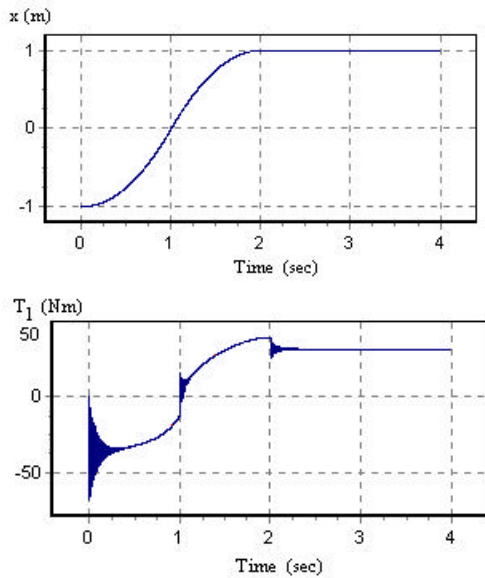


Fig. 5. The robot positions and torques for the variable structure algorithm with a $\omega_n=100 \text{ rads}^{-1}$, $\zeta=1$ and a sampling frequency $f_c=10\text{kHz}$.

In another experiment we analyse presents the position and force control (Figure 6). In this case we simulate a robot with flexibility at the joints. We consider a desired force $F=3 \text{ N}$ along the x_c axis, where the surface has $\theta=90^\circ$, while controlling the position along the y_c axis.

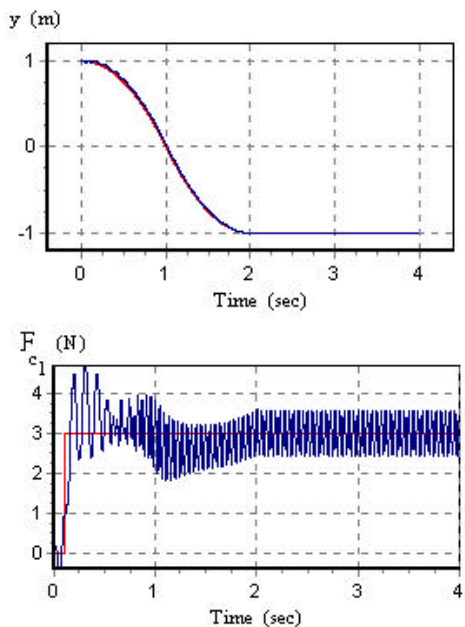


Fig. 6. Position and force control of a robot with flexibility joints using the hybrid controller with a $\omega_n=100 \text{ rads}^{-1}$, $\zeta=1$ and a sampling frequency $f_c=10\text{kHz}$.

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 7) allows the study of the main formulae and definitions adopted in the package.

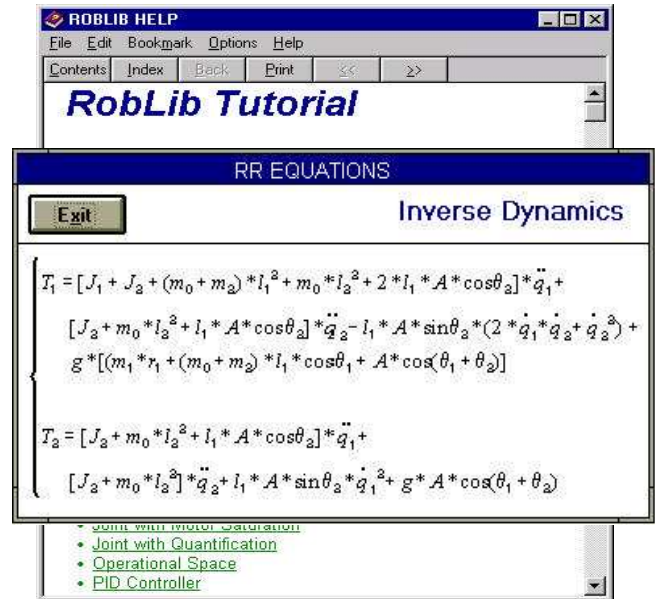


Fig. 7. Using the option Tutorial in the Help menu.

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.

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.

4. MANIPULABILITY OF ROBOTIC SYSTEMS

RobLib was designed for teaching the fundamental aspects of robotics. Nevertheless, it seems interesting to motivate the students to new research aspects. In this perspective, *RobLib* includes the entry 'Manipulability' in the tools menu.

For one arm, Yoshikawa proposed, the manipulability index μ given by:

$$\mu = |\det[\mathbf{J}(\mathbf{q})]| \quad (1)$$

where \mathbf{J} is Jacobian of the robot kinematics. For one robot the analytical development of μ is straightforward; nevertheless, for two or more robots

the definition of μ is more complex. To overcome this problem *RobLib* adopt a numerical approach inspired by the Monte Carlo method.

The method consists in generating a numerical sample of n points inside a sphere with radius ρ in the joint space, and to map them to the operational space, in order to obtain a set of points corresponding to different ellipsoids. The size and shape of the ellipsoids determine the “amplification” between the joint space and the operational space. The amplification is related to eigenvalues of the Jacobian robot kinematics and corresponds to the area of the ellipsoid.

The manipulability varies in the workspace, that is $\mu = \mu(x,y)$; therefore, we consider some sub-indices to simplify the study of the manipulability of several arms namely:

- The index μ_1 is defined as the maximum volume of μ in all the possible workspace W .

$$\mu_1 = \text{Max} [\mu(x,y), \forall x,y \in W] \quad (2)$$

- The index μ_2 is the average volume of μ considering only the workspace W where $\mu \neq 0$.

$$\mu_2 = \text{Av} [\mu(x,y), \forall x,y \in W: \mu(x,y) \neq 0] \quad (3)$$

- The index μ_3 represents the average volume of μ in all the possible workspace W .

$$\mu_3 = \text{Av} [\mu(x,y), \forall x,y \in W] \quad (4)$$

Figure 8 compares the analytical method (1) and the numerical algorithm for one RR robot.

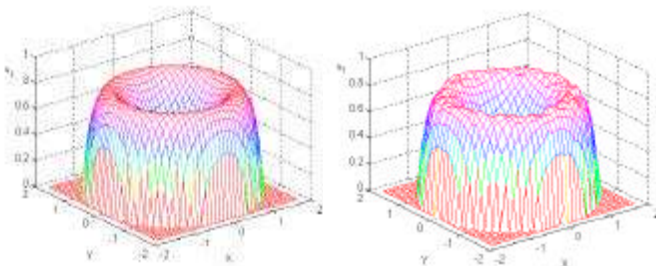


Fig. 8 - Manipulability μ of one RR robot with $l_1 = 1$ and $l_2 = 0.8$ obtained by the: a) analytical method, b) numerical algorithm for a sample of $n = 1000$ points.

Based on these concepts the following experiments adopt two robots with RR structure working in cooperation (Figure 9), in order to determinate the manipulability of the total system and the system configuration that leads to a superior performance.

Given the kinematic redundancy of the two-arm system, for each grasping point we consider that, the left and the right arms define, alternatively, the hand position.

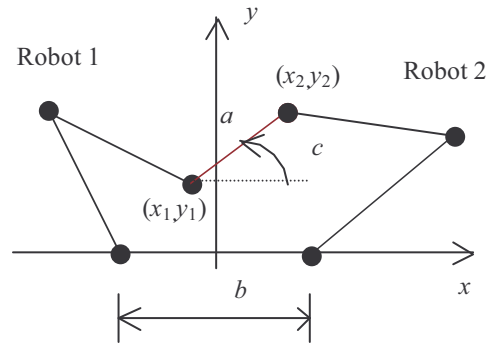


Fig. 9. Two RR robots working in cooperation for the manipulation of an object with length a , and orientation c .

Figures 10 - 12 show the indices μ_1 , μ_2 and μ_3 versus the parameters a , b , and c . This numerical experiment considers a grid of $m = 1000$ points and, for each of these points, a sample of $n = 1000$ points, inside a sphere with a radius of $\rho = 0.1$ rad in the joint space. We conclude that the maximum manipulability occurs for $l_1 = l_2$ and $c = 0$. Further analysis reveals that the best manipulability occurs when $a = b$ for $c = 0$.

4. 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, control and manipulability performance, 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.
- Ferreira, N. M and Tenreiro Machado (2000). RobLib: An Educational Program for Robotics, Syroco'00, 6th International IFAC Symposium on Vienna-Austria.
- Ferreira, N. M. and Tenreiro Machado (2000). Manipulability Analysis of Two Robotic Systems, Ines'00, IEEE International Conference on Intelligent Engineering Systems 2000.

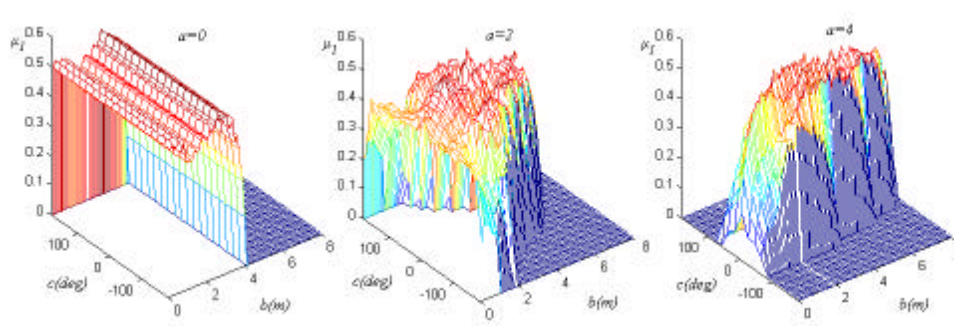


Fig. 10. The maximum manipulability μ_1 versus b and c for object lengths $a = \{0, 2, 4\}$, with $m = 1000$, $n = 1000$, $\rho = 0.1$ rad, Robot 1= Robot 2: $\{l_1 = l_2 = 1\}$ m}.

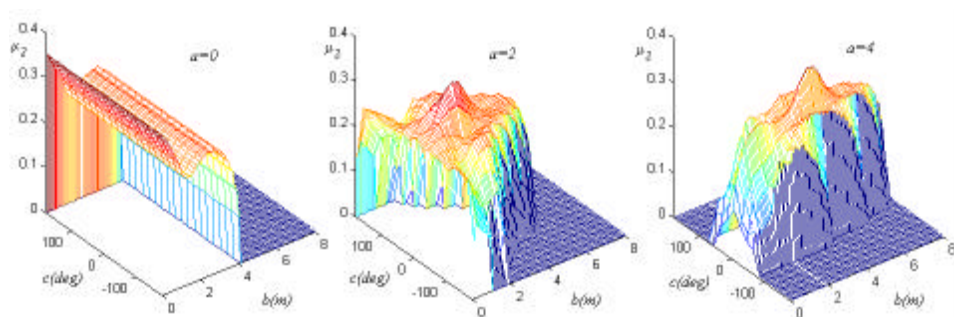


Fig. 11. Two arm average volume of the manipulability μ_2 versus b and c for object lengths $a = \{0, 2, 4\}$, with $m = 1000$, $n = 1000$, $\rho = 0.1$ rad, Robot 1 = Robot 2: $\{l_1 = l_2 = 1\}$ m}.

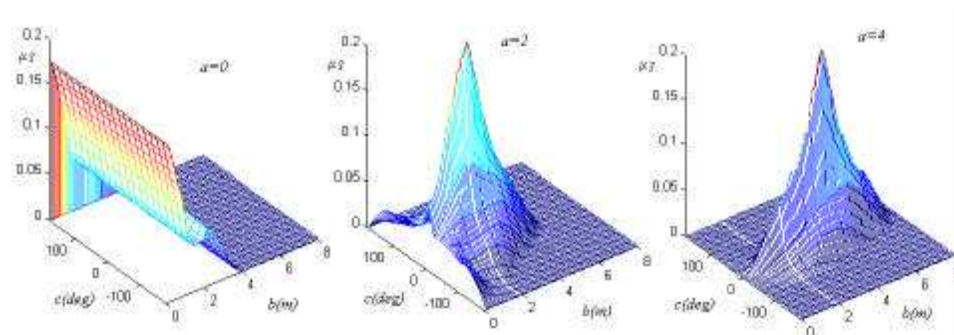


Fig. 12. – Two-arm average manipulability for the considered basis μ_3 versus b and c for object lengths $a = \{0, 2, 4\}$, with $m = 1000$, $n = 1000$, $\rho = 0.1$ rad, Robot 1 = Robot 2: $\{l_1 = l_2 = 1\}$ m}

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.

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 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 Robotic Arm. *IEEE Trans. Education*, vol. 32, no. 4, pp. 417-429.

Workspace (1997). Manual, Robot Simulations, Newcastle-Upon-Tyne England.