AN EDUCATIONAL FRAMEWORK FOR REMOTE CONTRO OF A PUMA ROBOT

Carlos Cesar Aparecido Eguti, eguti@ita.br
Carlos Eduardo Oliveira da Silva, csilva@ita.br
Emil Yoshigae Nakao, emilnakao@msn.com
Emília Villani, evillani@ita.br
Luis Gonzaga Trabasso, gonzaga@ita.br
Instituto Tecnológico de Aeronáutica – ITA
Divisão de Engenharia Mecânica-Aeronáutica
Pça. Mal. Eduardo Gomes, 50 - Vila das Acácias
CEP: 12228-900 – São José dos Campos/SP, Brazil

Abstract. *This paper presents the development of a low cost educational framework for the remote control of a PUMA robot. The framework is based on an open control architecture that allows a remote user to develop new control strategies for manipulating the robot and test them via Internet. This work is part of the a project which aims to provide students in remote locations with facilities of the shopfloor of the Center of Competence in Manufacturing (CCM) at ITA. This work is part of the FAPESP Program called TIDIA/KyaTera, which connects a number of research laboratories through an advanced high-speed optical network. The network is being used as a test bech for research in different areas including the remote control of robotic systems.*

Keywords*: PUMA robot, retrofitting, e-learning, Internet, e-manufacturing*

1. INTRODUCTION

This work merges the concept of e-manufacturing with another paradigm that is also based on the Internet: the e-learning. Basically, it explores the application of the e-learning concept to a manufacturing environment.

E-manufacturing can be defined as the use of Internet based tools for information interchanging with high flexibility and rapid response in manufacturing systems. While Internet and the e-business have accelerated product realization, manufacturing, delivery, and customer services, e-manufacturing provides the necessary support on the shop-floor, integrating manufacturing systems both vertically and horizontally. Vertical cooperation occurs along a supply chain (i.e., among suppliers, manufacturers, warehouses, distributed centers, and retailers), while horizontal cooperation is performed among peers (Hao et al, 2004).

On the other hand, e-learning is the unifying term used to describe the fields of online learning, web-based training, and technology-delivered instruction (Ageless Leaner, 2009). According to the site of Ageless Leaner (Ageless Leaner, 2009), e-learning features includes “the use of network technologies to create, foster, deliver, and facilitate learning, anytime and anywhere”; “the delivery of individualized, comprehensive, dynamic learning content in real time, aiding the development of communities of knowledge, linking learners and practitioners with experts”; and “a phenomenon delivering accountability, accessibility, and opportunity to allow people and organizations to keep up with the rapid changes that define the Internet world”.

In this context, the CCM (Manufacturing Competence Center) of the Aeronautics Institute of Technology (ITA) proposed the e-CCM Project, which aims at providing remote access to the its facilities. The e-CCM is part of the FAPESP Program TIDIA/KyaTera, which connects a number of research laboratories through an advanced high-speed optical network. The TIDIA/Kyatera network is been used as a testbed for research in different areas including the remote control of robotic systems.

Particularly, this paper presents the development of a low cost educational framework for the remote control of a PUMA robot. This framework is based on an open control architecture that allows a remote user to develop and test control strategies for manipulating the robot and perform different tasks using the Internet. The purpose of this platform is to support the development of teaching and research activities about robotic manipulators.

The key feature of this proposal is the bypass of the commercial robotic controller, which, in general is severely restricted in terms of modifications to software for real-time control, programming and integration. The original controller may be quite acceptable in most manufacturing environments where accurate and faithful repetition of a limited sequence of motions is all that is required (Goldenberg, Chan, 1988). However, in a research environment, the primary considerations are those of ease of modification, adaptability, and programmability.

This paper is organized as follows. Section 2 presents a review of the related work. It discusses some works that apply the e-learning paradigm to robotics and resumes many research works based on the PUMA robot. Section 3 presents the open platform developed at CCM. It details both hardware and software solutions. Section 4 discusses the remote access facilities provided by the platform. Finally, Section 5 drives some conclusion and discusses future work.

2. RELATED WORK

2.1. e-Learning in Robotics

There are a number of works that applies the e-learning concept to robotic systems. This section brings a brief review in order to contextualize the work presented in this paper.

In Fabri etl al (2004) the authors present an e-learning environment that aims to communicate the knowledge that can be learnt in a Mobile Robotics Laboratory and delivered through the web. This work focus on the development of experiments related to environment mapping, path planning, trajectory following, Navigation Guidance and Control (NUCC), obstacle detection and obstacle avoidance. Although some of these tasks also apply to industrial robotics, in mobile robotics image processing is a key issue.

La Russa e Faggiano (2004) is also concerned with the merging of e-learning and robotics but it focus on the human cognitive processes. This is an example of a dedicated solution. It presents the eLC Robot, which is essentially an adaptive learning environment that interacts with users via audio-visual communication and that acts accordingly by moving its head and upper limbs depending on the specific situation.

Mootien et al (2004) presents a web-based application developed for the introduction and study of robotics. The web site acts as a platform to program the Gryphon Robot found at the Faculty of Engineering, University of Mauritius. The undergraduated students can study robotics, write their programs and run the robot remotely. The language used to program the robot is WALLI3, which allows the specification of tasks for an automated or robotic workcell. Comparing to the work presented in this paper, the main difference is the level of control that the remote user has over the robot. In our case, the remote user can effectively define the control law of each electric motor, while in Mootien et al (2004) he only defines tasks at a high level of programming. The same consideration also applies to the work presented in Wirtz et al (2006), which focus on the remote programming of a multirobot system.

2.1. Research Activities based on PUMA Robot

According to (Dixon et al, 2001) the development of research on Robotics aims at subjects such as trajectory generation, actuator-level control, sensor development, flexible programming and applications. However, the researchers have to overcome a variety of issues, such as hardware interfacing, actuator-level control implementation, software development and user-interface development, before the issues that are of interest to the research can be addressed.

Many works in this area bypass this problem by developing a software or hardware simulator. An example is the work of Dixon et al (2001), which presents a software toolkit based on the MatLab/Simulink platform. Singaraju et al (2006) presents a remotely accessible hardware-in-the-loop (HIL) simulator, which is used for the development of the PUMA 560 dynamics and control. The basic structure of the HIL setup involves two motors coupled with one another through the shaft. One of the motors operates as the joint actuator, while the other motor is used for the generation of the total torque. This platform is also developed in MatLab/Simulink environment and has remote access using a client-server communication based on TCP/IP with sockets.

Although software and HIL simulations are important steps in the development or validation of a research proposal, it has its limitations. Simulation requires the modeling of the system under study. A model is always an abstraction of the real world, and, as such, incorporates simplifications that can or cannot impact on the results. When the purpose is teaching, the development of activities using a physical device (robot) has another important advantage that is the student motivation. These reasons motivated the retrofitting of a robot for the e-CCM project.

The robotic manipulator under retrofitting is the PUMA 560, a six-degree of freedom manipulator with a payload of 5kg, which is commonly used in welding and assembly applications.

The PUMA (Programmable Universal Machine for Assembly) robot was developed by Unimation in 1978. It has been widely used in countless researches over the last decades by academic institutions spread all over the world. The history of robotics in surgery begins with the PUMA 560 - a robot used in 1985 to perform neurosurgical biopsies with greater precision. It is probably the most common robot in universities.

The PUMA can be divided into two parts: the arm and the controller. The arm is illustrated in Figure 1. Each degree of freedom is driven by a DC servomotor, which is connected to an incremental encoder and a potentiometer.

While the robot arm itself is still robust and easily useable for research and teaching purposes, the original Unimation Mark II controller is outdated and has limited resources for integration, programming and control. According to Goldenberg and Chan (1988), the VAL II based controller is seriously lacking in terms of providing sufficiently flexible, -expandable, and powerful real-time control. Its major limitations are (Goldenberg, Chan, 1988):

- Lack of floating-point hardware to perform complex mathematical transformations in real-time.
- Limited capability for trajectory modification using sophisticated peripherals such as vision and contact sensors.
- VAL software is currently stored in erasable programmable read-only memory (EPROM's), which does not enable the user to examine or modify the software.

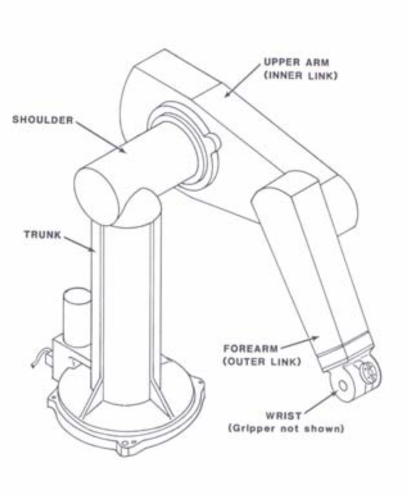


Figure 1. The PUMA robot (Bihr, Degrange, 1999).

In order to overcome these problems many academic works proposes the integration of the PUMA controller with an external computer. Goldenberg and Chan (1988) reviews some of the main works presented in the literature (Figure 2) and discusses the advantages and disadvantages of each alternative. These alternatives differ in terms of the extent to which the existing controller hardware is replaced. Basically, the conclusions of Goldenberg and Chan (1988)are that the alternatives (3)-(7) shown in Figure 2 fail to provide sufficient flexibility to overcome the limitations of VAL language. The alternative (1) is considered potentially the most powerful and flexible. However it requires the development of a complete new custom hardware and software. The alternative (2) maintains the low-level power hardware of the servo motors and replaces remaining boards of the controller. This alternative is chosen by Goldenberg and Chan (1988) and it achieves a great deal of flexibility without much complexity.

An example of work that follows the alternative (5) is presented in Wang et al (1991). A self-tuning adaptive control is used for control of the contact force between the end-effector of a PUMA560 robot and its environment. A PID controller is proposed, with its gains being continually updated using the most recent estimation of the dynamics of contact. The external force loop is closed around the robot positioning system.

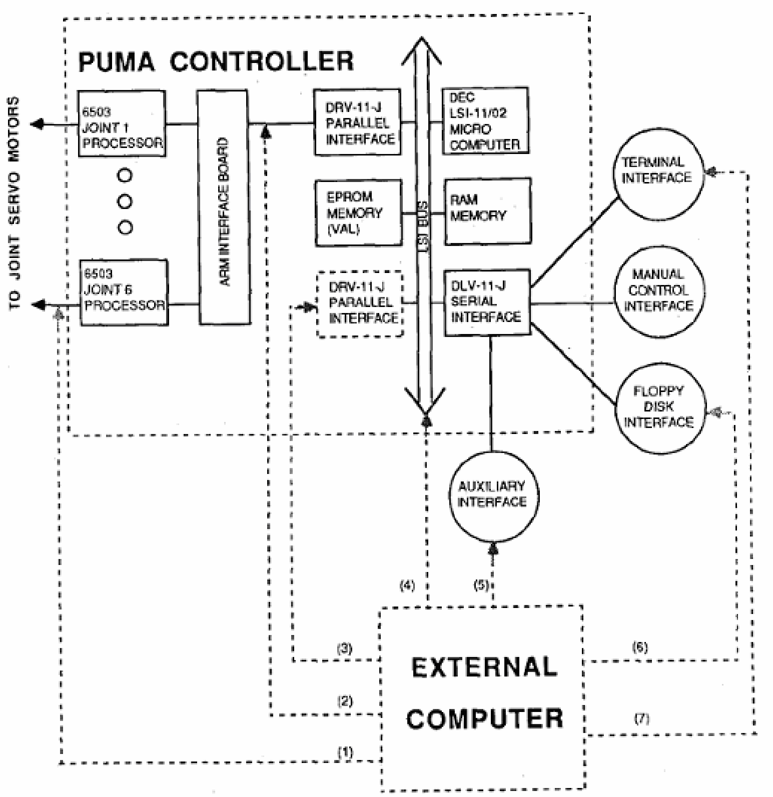


Figure 2. Integration and/or replacement of the PUMA controller (Goldenberg, Chan, 1988).

IEEE Control Systems Magazine of October 2004 has a special paper (Becerra et al, 2004) on the retrofitting of a PUMA Robot for educational purposes. The solution adopted in this case follows the alternative (2) of Figure 2. In order to control the PUMA arm using a PC, the authors replaced the original LSI/11 computer, EEPROM boards, CMOS boards, ad/ac interface, digital servo boards, and arm interface card. The original power amplifiers and current/torque controllers remain in the control architecture, as does the arm cable card. The software solution is developed using MATLAB-SIMULINK programming environment. It includes the trajectory generation routines and a PID computed torque control scheme.

3. The PUMA Retrofitting

3.1. The Hardware Architecture

The solution chosen for the PUMA retrofitting is the complete replacement of all control and power hardware. This solution corresponds to alternative (1) of Figure 2. The main motivations for the complete by-pass of the original PUMA controller are the availability of off-the-self power amplifier and control boards at low cost, the additional control and monitoring functionalities in new off-the-self amplifier boards and the difficult to integrate the original amplifier of the PUMA controller with new control boards, using the old Unimation wire wrapped backplane.

A number of hardware suppliers were consulted for the definition of the hardware architecture, such as National Instruments, Galil Motion Control, Trident Robotics and Servo-To-Go. The main criterion used in the choice was cost.

The chosen architecture is based on Galil Motion Control products and is illustrated in Figure 3. It is composed of two amplifier boards (AMP-19540), a control board (DMC-1800), a power supply of 40V/30A and an IBM standard PC.

Each amplifier board supports up to 4 servo motors (Galil, 2006a). The DMC-1800 (Galil, 2006b) is PCI bus motion controller for multi-axis applications and supports up to 8 servo motors. It is based on 32-bit Motorola 68331 microprocessor and has 4Mbytes of RAM and 4Mbytes of Flash memory. The communication interface with the host PC contains a primary and secondary communication channel. For synchronization with outside events, the DMC-1800 provide a number of digital inputs, digital outputs and analog inputs for interface to joysticks, sensors, and pressure transducers. Dedicated optoisolated inputs are provided for forward and reverse limits, abort, home, and other safety related applications.

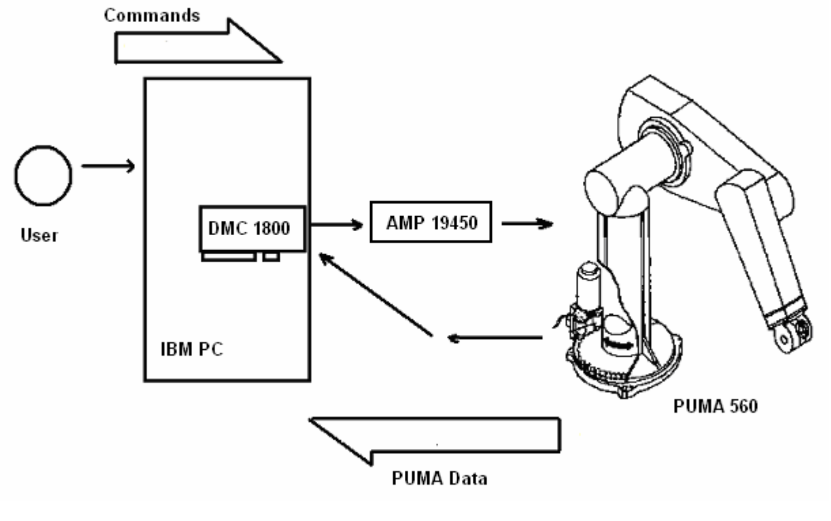


Figure 3. Hardware architecture of PUMA Retrofitting.

3.2. The Software Support Platform

The software solution for the PUMA retrofitting is developed in the LabView. The choice of LabView is due to its rapid and plug-and-play connection with different hardware equipment, the easy development of user friendly interface, the easy integration with other programs, such as MatLab routines, and the facilities provided for remote monitoring and control.

The software solution is divided into two parts. The first one is related to the basic structure that provides access to the PUMA axis and the routines of direct and reverse kinematics. It is called PUMA Support Platform and is detailed in this section. The second part is related to additional features and functionalities developed for educational purposes. It is presented in Section 3.3.

The controller implemented in the DMC-1800 board is a set of independent PID (Proportional Integral Derivative) controllers, each one associated to one axis of the PUMA.

The PUMA Support Platform is organized in the following blocks (Figure 4):

- DLL of the DMC-1800 controller.
- DLL for unit conversion.
- DLL for direct and inverse kinematics.
- User-friendly interface.

The DLL of the DMC-1800 controller is provided by Galil and is used to download commands and programs to the DMC-1800. For this purpose, Galil developed a dedicated programming language composed of basic commands that uses the number of encoder pulses to increment the position motor position or to define the motor speed and acceleration. An example of command is 'move the ith motor of n encoder pulses'. This programming language allows the access to all the functionalities provided by the DMC-1800 board. However, it is a general purpose programming language, not adequate for robotics applications.

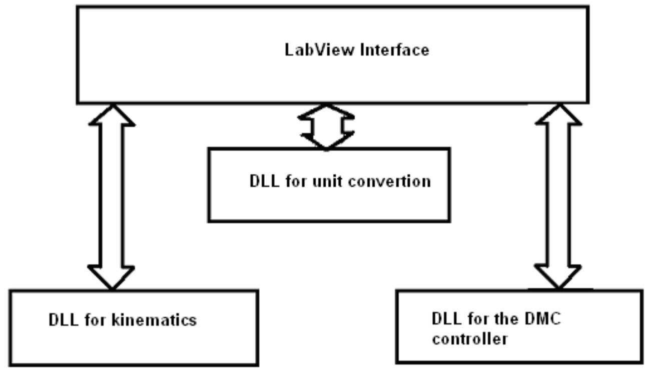


Figure 4. Main blocks of the PUMA Support Platform.

Due to the difficult of commanding a robot using encoder pulses, a set of routines is implemented in C++ for conversion of units and coordinate systems. These routines are organized in a set of different DLL, in order to assure easy software maintenance. Among these DLLs are one for the conversion of encoder pulses to degrees and one for the direct and reverse kinematics.

The direct kinematics determines the position and orientation of the end-effector in a Cartesian coordinate system for a particular set of joint angles. The reverse kinematics determine the angle of each joint of the PUMA robot in order to put the end-effector in a particular position and orientation. Among the solutions proposed in the literature for the problem of reverse kinematics, this work adopts the geometric solution presented in Lee et al. (1983). This choice is due to the possibility of selecting the desired solution for the end-effector positioning using a set of user defined variables. Table 1 presents the Denavit-Hartenberg parameters for the PUMA 560 robot used in the software.

Table 1. Denavit-Hartenberg parameter for PUMA 560.

Joint i	θ_i (degree)	α_i (degree)	a_i (mm)	d_i (mm)	Joint reach (degree)
1	90	-90	0	0	-160 a 160
2	0	0	431,8	149,09	-255 a 45
3	90	90	-20,32	0	-45 a 225
4	0	-90	0	433,07	-110 a 170
5	0	90	0	0	-100 a 100
6	0	0	0	56,25	-266 a 266

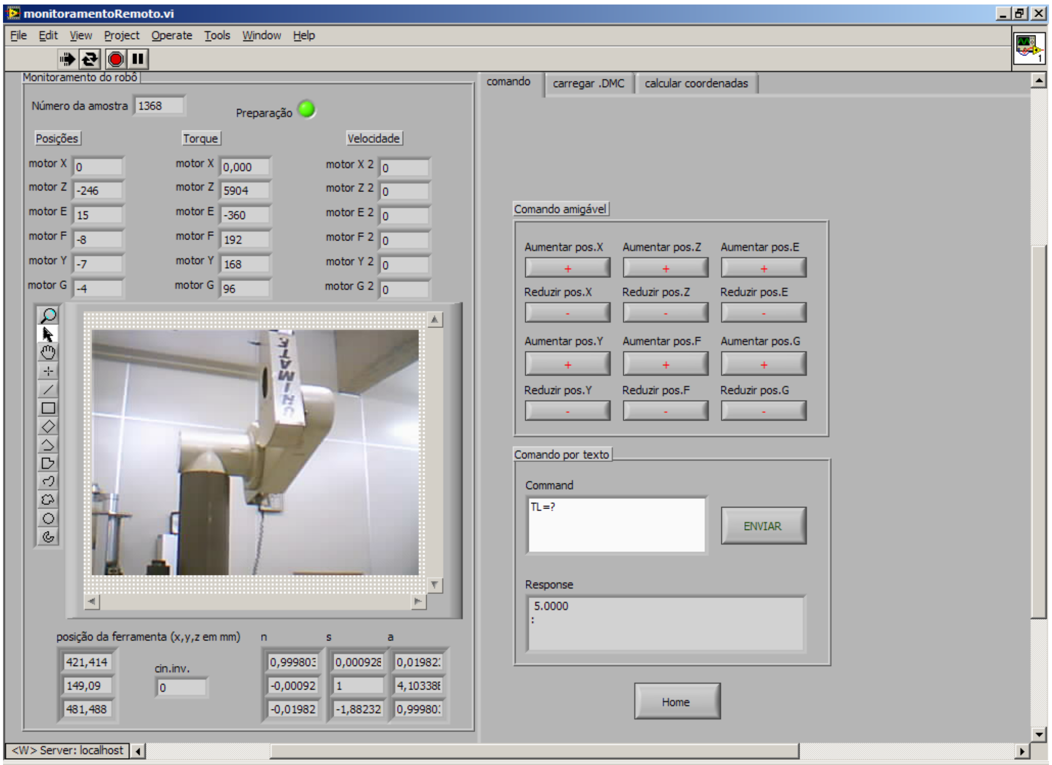
The user interface of the PUMA Support Platform provides the following functionalities:

- To send single commands to the DMC 1800 board using the Galil programming language and monitor the response of the DMC 1800 board;
- To load scripts of commands in the DMC 1800 board using the Galil programming language and start/halt its execution;
- To monitor the angles of the PUMA joints;
- To set the value of the angles of the PUMA joints;
- To monitor the position and orientation of the PUMA end-effector;

- To increment the position and orientation of the PUMA end-effector;
- To set the position and orientation of the PUMA end-effector.
- To provide the transmission of real-time video captured by a web-cam positioned near the PUMA robot.
- To make available the same interface in a webpage in order to allow a remote user to control and monitor the PUMA robot.

The user interface is divided into two sections (Figure 5). The first section, on the left of Figure 5, is dedicated to the PUMA monitoring, it contains the video and uses the direct kinematics routines to display the position and orientation of the end-effector. The second section, on the right of Figure 5, receives the user commands. It is divided in three abas, one for the increment of the position and orientation of the PUMA end-effector (illustrated in Figure 5), one for the command of the robot using the reverse kinematics routine or the joint angles (illustrated in Figure 6), and one for the execution of commands or the loading of scripts in the Galil programming language.

Figure 6 shows the same interface in a webpage, available to a remote user in via Internet.



Número da amostra 1368

Posições		Torque		Velocidade	
motor X	0	motor X	0,000	motor X 2	0
motor Z	-246	motor Z	5904	motor Z 2	0
motor E	15	motor E	-360	motor E 2	0
motor F	-8	motor F	192	motor F 2	0
motor Y	-7	motor Y	168	motor Y 2	0
motor G	-4	motor G	96	motor G 2	0

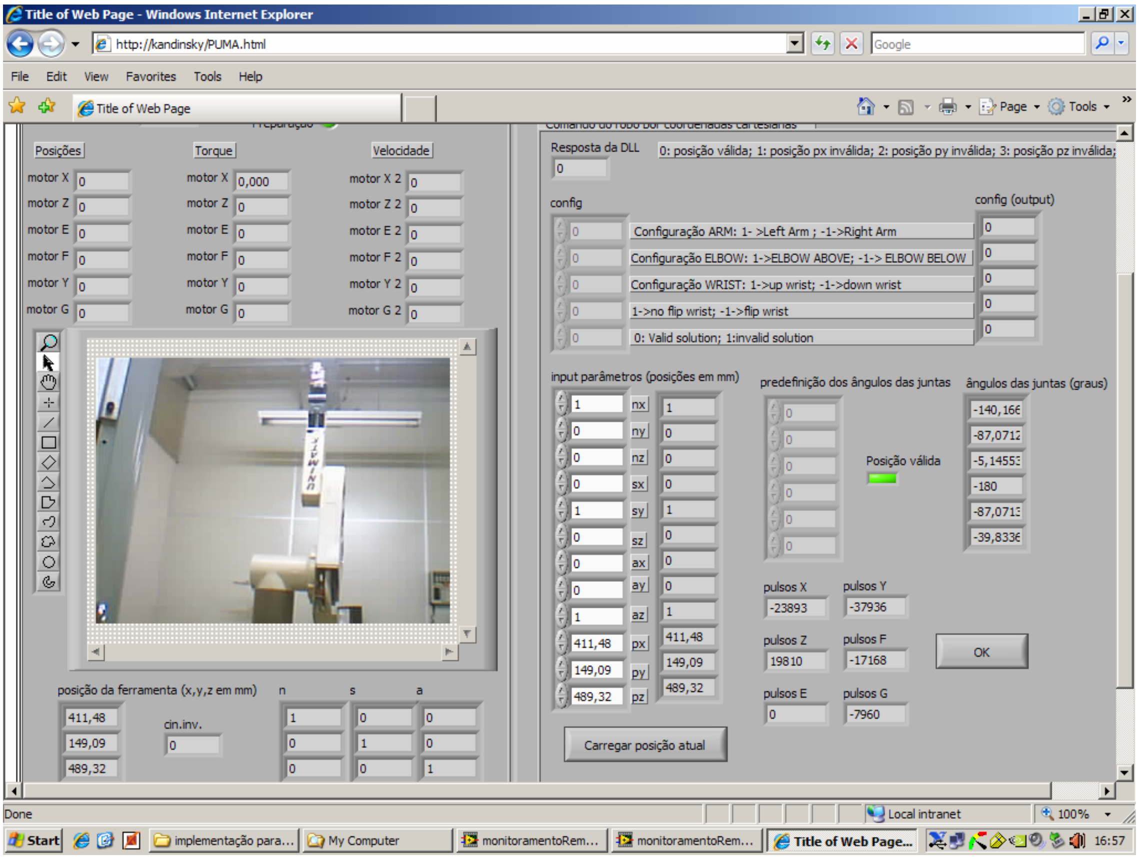
posição da ferramenta (x,y,z em mm)	n	s	a
421,414	0,999803	0,000928	0,01982
149,09	-0,00092	1	4,10338
481,488	-0,01982	-1,88232	0,99980

cin.inv. 0

Command
TL=?

Response
5.0000
:

Figure 5. User interface of the PUMA Support Platform.



Posições		Torque		Velocidade	
motor X	0	motor X	0,000	motor X 2	0
motor Z	0	motor Z	0	motor Z 2	0
motor E	0	motor E	0	motor E 2	0
motor F	0	motor F	0	motor F 2	0
motor Y	0	motor Y	0	motor Y 2	0
motor G	0	motor G	0	motor G 2	0

Resposta da DLL: 0

0: posição válida; 1: posição px inválida; 2: posição py inválida; 3: posição pz inválida;

config		config (output)
0	Configuração ARM: 1- >Left Arm ; -1->Right Arm	0
0	Configuração ELBOW: 1->ELBOW ABOVE; -1-> ELBOW BELOW	0
0	Configuração WRIST: 1->up wrist; -1->down wrist	0
0	1->no flip wrist; -1->flip wrist	0
0	0: Valid solution; 1:invalid solution	0

input parâmetros (posições em mm)		
1	nx	1
0	ny	0
0	nz	0
0	sx	0
1	sy	1
0	sz	0
0	ax	0
0	ay	0
1	az	1
411,48	px	411,48
149,09	py	149,09
489,32	pz	489,32

predefinição dos ângulos das juntas	ângulos das juntas (graus)
0	-140,166
0	-87,0712
0	-5,14553
0	-180
0	-87,0713
0	-39,8336

Posição válida 🟩

pulsos X	pulsos Y
-23893	-37936
pulsos Z	**pulsos F**
19810	-17168
pulsos E	**pulsos G**
0	-7960

posição da ferramenta (x,y,z em mm)	cin.inv.	n	s	a
411,48	0	1	0	0
149,09		0	1	0
489,32		0	0	1

Figure 6. PUMA Support Platform available in a webpage.

3.3. The Software Additional Modules

The additional modules supports a number of teaching activities related to robotics. They are developed in LabView and, with the exception of Module 3 that requires the commercial software RobCad, are all available to remote users. The additional modules are:

- Module 1 – PUMA 560 programming. This module specifies a programming language for the retrofitted PUMA robot that is similar to the original VALL II. The purpose is to train undergraduated students in robot programming. The module uses the PUMA Support Platform routines to convert the PUMA program into DMC 1800 basic programming language. It also and provides the necessary user interface to upload, start and monitor the execution of the programs.
- Module 2 – On line programming via joystick. This module provides the necessary interface to move the PUMA robot using a joystick with the purpose of developing teaching activities related to on line programming. The Module 1 interfaces with a Playstation 2 joystick. Although this joystick is not the most appropriate to robotic application (which is very common and easily found by remote users, connected in a USB port of the remote or local computer. The linear movement X, Y, Z are available on the commands of the left hand while the rotation movement E, G, F are available on the right hand (Figure 7).
- Module 3 – Simulation and off-line programming. This module integrates the PUMA programming module with the simulator RobCad. It supposes the user has access to RobCad. RobCad is a graphical simulator that provides the visualization and verification of the robot performance in a manufacturing environment. It helps on the determination of reachability issues, working area, time of process, collision problems, among others. Module 3 converts the output of the RobCad simulation (Figure 8), which contains the robot movements to the PUMA programming language of Module 1.
- Module 4 – Robot vision. This module considers that a camera is connected to the PUMA arm (Figure 9(a)). A number of pattern recognition routines are implemented in order to detect a set of standard parts (Figure 9(b)) and the corresponding positions. The parts are place in front of the robot. The purpose of this module is to provide remote or local students with the facilities necessary to elaborate a program for the robot that take each part and mount it in the correct position, just like a puzzle.

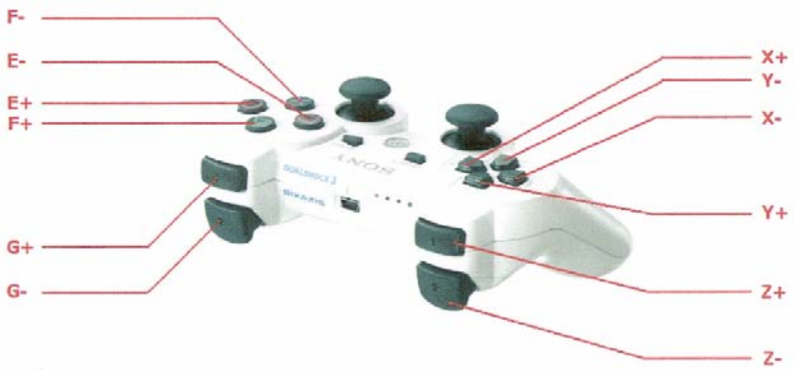
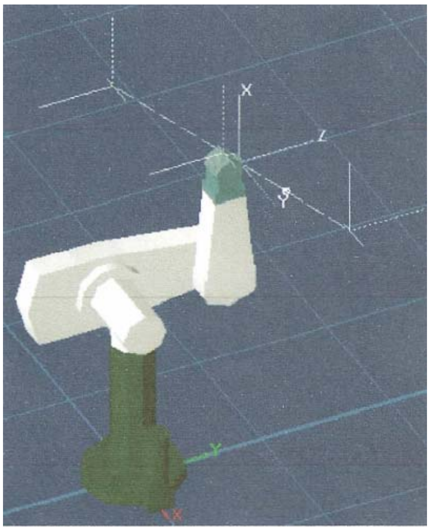


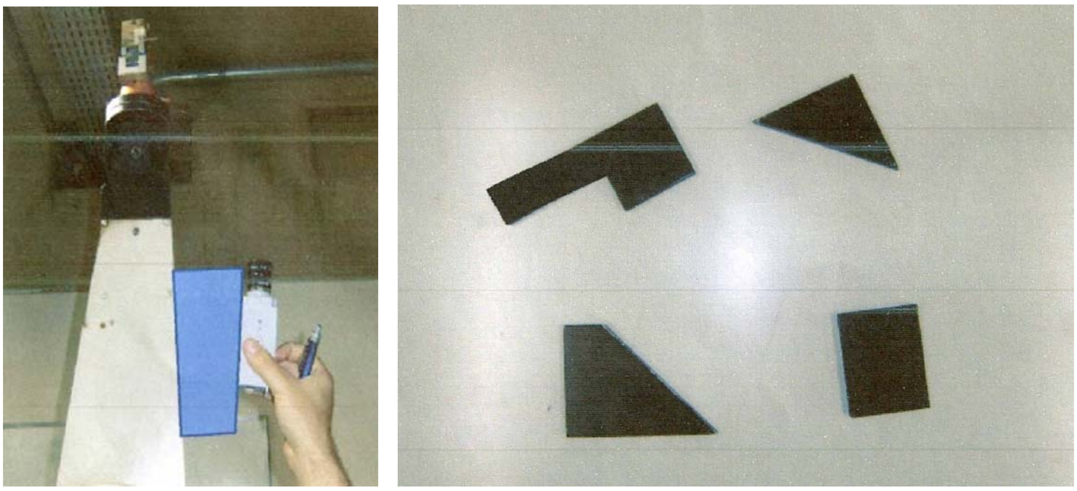
Figure 7. Mapping of the PUMA commands on a Playstation 2 joystick.



Example of ROBCAD output

```
prog_start pa8
robot_name pu560
tcpf 0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
path_start pa8
color 7
display 1
wp_start 1016
relative 0.000108,-150.000046,1590.000488,0.000000,0.000000,0.000000
reltcpf 489.999939,0.737686,-489.774567,-89.990168,89.913779,-89.903930
absolute -150.000046,-0.000108,1590.000488,0.000000,0.000000,-90.000003
attach puma
color 7
display 1
wp_start 1017
relative -919.773682,-151.384384,670.015930,80.106331,-89.912461,-80.020120
reltcpf -429.983826,-0.647189,430.000031,0.000000,0.015026,0.000000
absolute -151.384384,919.773682,670.015930,80.106344,-89.912461,-170.020130
attach puma
color 7
display 1
path_end pa8
prog_end
```

Figure 8. PUMA robot in the RobCad simulator and example of output program of the RobCad.



(a) (b)

Figure 9. a) PUMA with a camera, b) Standard parts used in Module 4, as they are viewed by the PUMA camera.

4. CONCLUSION

This paper presents the retrofitting of a PUMA 560 robot with the purpose of making it available for e-learning applications. The solution proposed in this paper is based in the complete bypass of the outdated Unimation Mark II controller. The hardware retrofitting is a low cost and open solution based on commercially available control and power boards.

The software solution is divided into two parts: a PUMA Support Platform and a set of additional modules. The PUMA Support Platform provides basic control and monitoring functions to remote users that are connected with a local computer via Internet. The additional modules support teaching activities related to robotics, such as on-line programming, simulation and off-line programming, and integration with vision systems. LabView is chosen as development environment for both parts. The facility to make the software interface available in an Internet webpage is one of the reasons for this choice. It can be easily integrated with control routines or trajectory planning strategies developed by remote users.

All the software modules and PUMA Support Platform were developed by undergraduate students, organized in independent groups. The experience confirmed that hardware platform can be easily used in teaching and laboratory activities.

3. ACKNOWLEDGEMENTS

This research is supported by governmental agencies CAPES, FAPESP, CNPq and FINEP.

4. REFERENCES

Hao, Q, et al, 2005, “Towards a cooperative distributed manufacturing management framework”, Computers in Industry, Vol.56, pp.71-84.

Ageless Leaner, 2009, “e-Learning”. 13 May 2009, <http://agelesslearner.com/intros/elearning.html>

Goldenberg, A.A., Chan, L., 1988, “An Approach to Real-Time Control of Robots in Task Space. Application to Control of PUMA 560 Without VAL-II”, IEEE Transactions on Industrial Electronics, Vol. 35, n.2, pp. 231-238.

Dixon, W.E., et al, 2001, “A Simulink-Based Robotic Toolkit for Simulation and Control of the PUMA 560 Robot Manipulator”, Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Mauí, Hawaii, USA, pp. 1-6.

Singaraju, T., et al, 2006, “Hardware-in-the-loop simulation of PUMA 560 via internet”, 32nd Annual Conference on IEEE Industrial Electronics, IECON 2006, pp. 5426-5432.

Bihr, W., Degrange, F., 1999, “Interface of a Force-Torque Sensor on PUMA 500 Robot”, Final Year Project 1999, Georgia Institute of Technologie.

Wang, Q. et al, 1991, “Adaptative Force Control of PUMA560 Industrial Robot Manipulator”, International Conference on Control '91, Vol.1, pp. 358-361.

Becerra, V. et al, 2004, “Hardware Retrofit and Computed Torque Control of a Puma 560 Robot”, IEEE Control Systems Magazine, pp. 78-82.

Galil, 2006a, “Multi-axis Brushless/Brush Servo Amplifier AMP-19520/40, Rev 1.0b”, Rocklin, California, 14 p.

Galil, 2006b, “DMC-1700/1800 User Manual, Rev 1.2m”, Rocklin, California, 253 p.

Fabri, D., et al, 2004, “Robot Control Designer Education on the Web”, Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, pp. 1364-1370.

La Russa, G., Faggiano, E., 2004, “Robo-eLC: Enhancing Learning Hypermedia with Robotics”, Proceedings of the IEEE International Conference on Advanced Learning Technologies, pp. 465- 469.

Mootien, S., et al, 2004, “A Web-Based Interface for the Gryphon Robot”, IEEE International Conference on Industrial Technology, IEEE ICIT '04, Vol. 2, pp. 842- 847

Wirz, R., et al, 2006, “Remote Programming of Multirobot Systems within the UPC-UJI Telelaboratories: System Architecture and Agent-Based Multirobot Control”, IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), pp.219-224.

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.