RAPID PROTOTYPING WITH EMPHASIS ON ARCHITECTURE CONTROL IMPLEMENTED IN MECHATRONIC DEVICES

PARACÊNCIO, Luís Gustavo de Mello
UNIP - University Paulista
lumello@terra.com.br

ROSÁRIO, João Mauricio
UNICAMP - University of Campinas'
rosario@fem.unicamp.br

ABSTRACT: *In recent years, w*ith technological advances in mechatronics engineering it is necessary to monitor and improve the studies of these new trends. This research aims to present a methodology to integrate robotic manufacturing cells with emphasis on modeling and controlling of a robotic device. The model consists of three degrees of freedom, driven hydraulically allowing the placement of a table where a piece can be worked by two industrial robots constituting a system of collaborative manufacturing. Studies are presented concerning kinematic and dynamic models and the calculation of control system´s parameters using MatLab-Simulink™. An interface was developed in LabVIEW™ language for acquisition and processing of the information from the sensors of the joints and the implementation of the system of supervision and control. Since the publication of an HTML page, a cell collaborative application may be available on the collaborative WEB allowing the creation of a virtual laboratory directed to scientific and technological research and the possibility to connect with other laboratories for teaching and research. For instance, this will allow carrying out the implementation of distance learning experience and performing complex tasks in real time.

Keywords: Collaborative systems, robotics workcell, e-learning.

1 Introduction

The number of robots working in industry increases significantly due to its ability to operate in terms of flexibility, speed and accuracy (David and Rosário, 1998). In most industrial applications tasks robots are programmed by learning without the need for a geometric model. Thus, its trajectory is defined by a set of angles associated with the angular motion of each degree of freedom robot. After interpolation, these angles will act as reference signals for positioning controllers located at each joint that compare the signals from the sensor position of the joints (Rosário, Oliveira and Sá, 2002).

However, flexible manipulators faster are essential to achieving this performance leading to reduced production time and small energy consumption (Oliveira, 2008), control algorithms more resources should be implemented, which can deal with important parameters variations, for example, variations of inertia (Rojas, 2004).

Typically, the integration of industrial robots and mechatronic device in Flexible Manufacturing Cell (FMC) involves modeling methodologies using the formalism of automation. The environment modeling Computer Aided Design (CAD) may have been associated with the performance of robot control, including equipment and related mechanisms, mathematical modeling of the robot (forward and inverse kinematics) and its connected devices, but also the coordination and integration of robot movements with other devices (Aihara, 2001).

1.1 Steps for Rapid Prototyping

This paper only Focus on the study of the Flexible Manufacturing Cells (FMC) with two more robots mechatronic device that emphasis on rapid prototyping of mechatronic control systems:

a) Functional specification of the robot cell using SFC as a function of defining the problem under study (the welding operation of complex device) by specifying automated system (Sequential Modeling of Collaborative Systems using SFC);
b) Functional specification / technology mechatronic system to work cooperatively with other elements of the cell (robots);

c) Modeling kinematics, dynamics and implementation of predictive control strategy;
d) Implementation of the monitoring system with monitoring and control through WEB based in LabVIEW™;
e) System Initialization and Calibration;
f) Achievement Test Validation.

This work only focuses on the study of 3 DOF mechatronic devices without regard to the timing with the robots, as well as modeling and simulation of this framework with particular emphasis on the development and implementation of robotic controllers common position. The paper is organized as follows. Section 2 provides a description of the framework, including kinematics, dynamics and modeling of the actuator. Section 3 presents the structure of PID control. Section 4 is devoted to the results obtained within a virtual environment for robotics. Finally Section 5 offers some conclusions and future trends.

2 Mechatronics Device

The Figure 2.1 shows a particular application FMC based on coordination and integration of two industrial robots and mechatronic device (table) with 3DOF (robot RRP) developed for purposes of work needs to solder. This device can help with tasks for which traditional manipulators have difficulty reaching some parts of the piece and the table is synchronized with the handlers by allowing them to perform complex tasks.

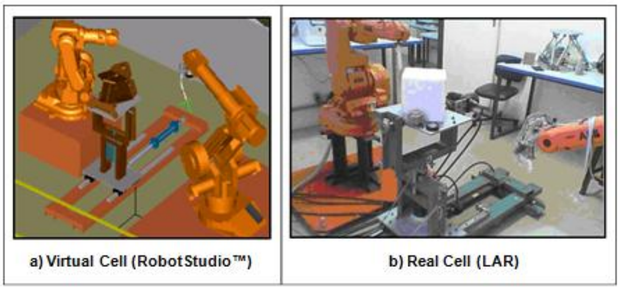


Figure 2.1: Flexible manufacturing cell

This section presents the modeling and simulation of the three degrees of freedom (3 DOF) robot, which leads to the concept of a virtual environment using electrical and mechanical libraries blocks in combination with SIMULINK™ blocks (Figure 2.2).

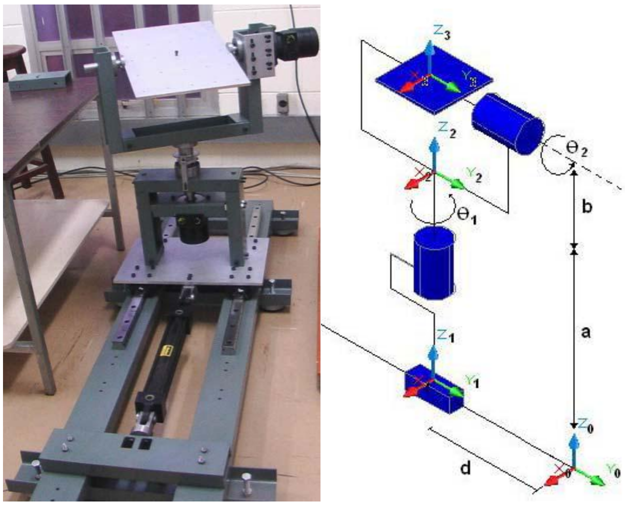


Figure 2.2: 3DOF robot and related motions

The simulator design requires definition and modeling of the three major joints connected to other parts of the manipulator through the gearbox. The main elements are the robotic joints, brushless DC motor drives, and the inertia of the shafts, gears and control blocks. The control system which consists primarily of control loops in cascade (in each axis) is built with Simulink blocks.

Position control of the manipulator can be implemented via feedback control of each isolate requiring that the model of each joint (Rosário and Cassemiro, 2003). In the end, all joints must be coordinated as shown in Figure 2.3 so that the dynamic model of the structure must be set.

The simulator includes a module for generating path providing joints with the trajectories of axes as reference signals for the control parts. Finally, a graphical interface is available by presenting results of joints movements obtained by typical trajectories.

2.1 Control structure including kinematics

For many operations, the operator defines the tasks or trajectories of controller´s reference for a coordinate system that is fixed to the end effector of the robot (in cartesian space) but the desired movements (expressed in angular coordinates) and the laws of control are different coordinate systems requiring the implementation of fast algorithms for the inversion of the kinematic model and generating the reference trajectory in angular coordinates (Figure 2.3).

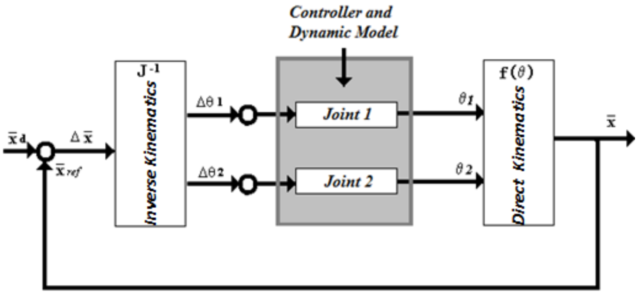


Figure 2.3: Control structure with kinematics

2.2 Dynamic Model

The Mentioned previously, the control of each joint is considered in an independent way with any coupling effect. These effects to take into account and to solve the problem the dynamic trajectory control involves the determination of the inputs so that the drive of each joint motion its links to the position values with required speed. The dynamic model of the robotic joint can be derived through the Euler-Lagrange formulation expressions that the generalized torque (Henriques, 2005). The manipulator dynamic behavior is described by a group of differential equations called dynamic equations of motion. For a 3 DOF rigid manipulator, the equations are:

$$\tau_i(t) = J_i(\theta(t))\,\ddot{\theta}_i(t) + C_i\big(\theta(t),\dot{\theta}(t)\big) + Q_i(\theta(t)) \qquad i = 1,\ldots,3 \tag{1}$$

where $\tau_i(t)$ is the generalized torque vector, $\theta_i(t)$ the generalized frame vector (joints), $J_i(t)$ the inertial matrix, $C_i(\theta,\dot{\theta})$ the non-linear forces (for example centrifugal) matrix, $Q_i(\theta)$ the gravity force matrix.

Combining all this (Figure 2.3) the input references obtained in angular coordinates from the trajectory interpolator are compared with the angular position sensor information of each joint (incremental encoder). The controller makes the corrections taking into account the robot's dynamic model developed above. These corrections are transmitted to the manipulator through the actuator described in the next subsection including a gearbox. These gearboxes are characterized by their ratio, inertia and stiffness and damping of input and output shafts. The gearboxes output shafts are connected to the other parts of the robot structure, which results in the effective torque reflected to each joint. For each three joints, the other links effects are globally considered as a single load inducing to the joint a torque composed of three terms (Equation 1).

2.3 Actuator Model

Each robotic joint commonly includes a DC motor, a gear and an encoder.. The three classical equations are the following considering DC motor:

$$u(t) = L\frac{\mathrm{d}i(t)}{\mathrm{d}t} + R\,i(t) + K_E\frac{\mathrm{d}\theta_m(t)}{\mathrm{d}t}$$

$$T_m(t) = J_{eq}\frac{\mathrm{d}^2\theta_m(t)}{\mathrm{d}t^2} + B_m\frac{\mathrm{d}\theta_m(t)}{\mathrm{d}t} \qquad (2)$$

$$T_m(t) = K_T\, i(t)$$

where $T_m(t)$ is the motor torque, $\theta_m(t)$ the angular position of the motor, $i(t)$ the motor current, *L* and *R* respectively the inductance, resistance of the motor, J_{eq} the inertia of axis load calculated on the motor side, resulting in the block diagram of Figure 2.4.

A specific library has been elaborated which includes complete axis models with controllers, motor drive, gear boxes and mechanical parts. This library enables easy change of controllers' structures or motor types.

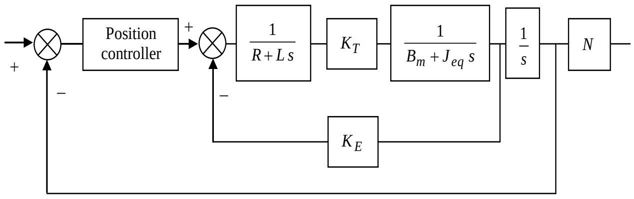


Figure 2.4: Block diagram of the joint axes

2.4 Kinematic Modeling and Testing Program

Given the setting angles of joints of the robot calculating the position and orientation of the end of the handler is called direct kinematics and it is always possible to obtain the solution of direct kinematic problem. In contrast, the inverse kinematic problem solution is somewhat more complex depending on the characteristics of the robot. Furthermore, multiple solutions and singularities of the problem may occur. Figure 2.5 illustrate the relationship between the two cinematic (Sanchez, Rosário, Uribe and Paracêncio, 2008).

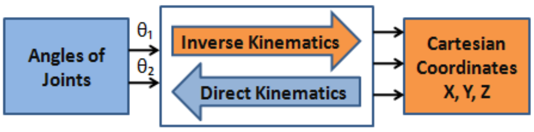


Figure 2.5: Direct kinematic and inverse kinematic of robot manipulators

2.5 Direct Kinematics

Obtaining direct kinematic model can be accomplished using Vectors Local (*VL*) which is considered the position of the center seat to the inertial frame and the end point *P* of the tool relative to the center of the table, θ_1 and θ_2 are rotation angles of two rotational joints (rotation of the base and rotation of the table). In Figure 2.6 are shown a schematic model of the table and an indication of the movements of the joints. The length of the rod revolution will be designated L_h.

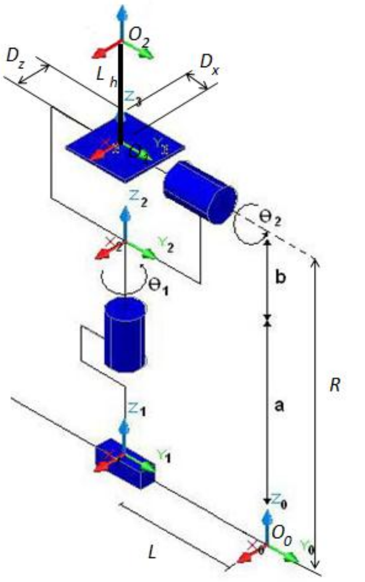


Figure 2.6: Table with 3 DOF

i) Rotation Matrices

$T_{\theta_1} = \begin{pmatrix} \cos\theta_1 & -sen\theta_1 & 0 \\ sen\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, due to rotation around the z-axis (base).

$T_{\theta_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_2 & sen\theta_2 \\ 0 & sen\theta_2 & \cos\theta_2 \end{pmatrix}$, due to rotation around the x-axis (rotation table).

ii) Calculating Vectors Places

$O_0O_1 = \begin{pmatrix} 0 \\ 0 \\ R \end{pmatrix}$ $\qquad O_1O_2 = \begin{pmatrix} 0 \\ 0 \\ L_h \end{pmatrix}$

$O_1 = 0_O + T_{\theta 1} \times O_{11}$

$O_2 = 0_1 + T_{\theta 1} \times T_{\theta 2} \times O_{12}$

Logo, $O_2 = O_0 + O_1 = \begin{pmatrix} -sen\theta_1 * sen\theta_2 * L_h \\ \cos\theta_1 * sen\theta_2 * L_h \\ \cos\theta_1 * L_h \end{pmatrix}$

O_0 = positioning the table in relation to the inertial frame.

O_1 = positioning the rod at the center of the table (*x*, *y*).

Conversion factor *X* = Encoder degrees / pulse.

Position in degrees = Number of pulses * conversion factor Encoder.

$\theta_1 = \theta_{1init} + \theta_{1lido}$

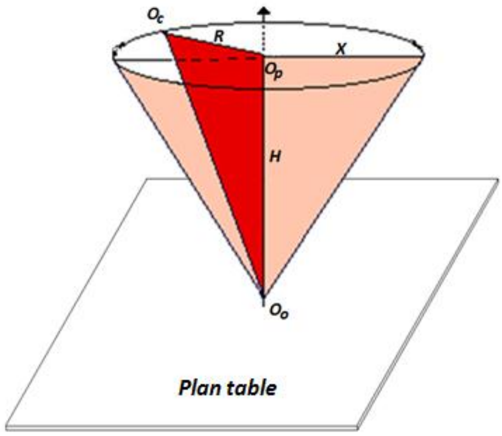
$\theta_2 = \theta_{2init} + \theta_{2lido}$

From the definition of fixed-angle positioning or displacement of a profile table (θ_2) is performed to drive a shaft fixed on the table (L_h) from the rotation of the base table (θ_1) describing a cone of revolution (θ_2 different from *0*) or a cylinder of revolution (θ_2 equals *0*, if the stem be positioned at a distance *d* from the center of the table).

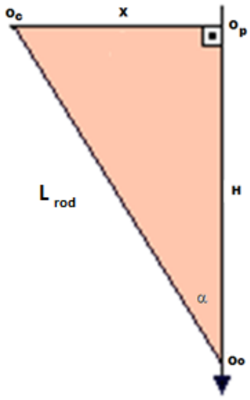
2.6 Inverse Kinematic

Considering that the rod rotation on the table (Figure 2.7) it held a cone in space. To obtain the inverse kinematic model we consider as input parameters the radius of the cone of rod revolution (cylinder in case of change of the center) of revolution (*x*), the current position of the table (θ_{1init}), value and direction of travel rod in degrees ($\theta_{1deslocl}$) and speed of table´s base. For the description of a cone space, the program will automatically calculate the desired angle to the table ($a = \theta_2$), allowing rotation of the base table from the initial position to the desired final position (θ_{1init} and θ_{1final}), as:

ISSN 1679-8902



a) Desktop: rod in the center of the table



b) Ligth cone

Figure 2.7: Cone rotation

i - Parameters

X: Radius of the circle of revolution.

L_h: Stem length sets the base table.

$\theta_{1desloc}$: Desired offset from the base of the table in degrees.

θ_{1init} : initial position.

ii - Equations Model

- **Case 1: Cone Rod Space Center Rotation Table:**

$$\theta_2 = ATAN2\left(\frac{x}{\sqrt{L_h^2 - x^2}}\right)$$

$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$

- ***Case 2: Cone Rod Displaced Space Center Rotation Table:***

X: Radius of the circle of revolution (distance from the shank to the center of rotation of the table).

X_o: Distance from the stem to the center of revolution of the table.

L_h: Stem length sets the base table.

$\theta_{1desloc}$: Desired offset from the base of the table in degrees.

$$\theta_2 = ATAN2\left(\frac{x}{\sqrt{L_h^2 - x^2}}\right)$$

$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$

- ***Case 3: Cylinder Rod Revolution Displaced Center of Rotation of the Table***

X: Radius of the circle of revolution (distance from the shank to the center of rotation of the table).

L_h: Stem length sets the base table.

$\theta_{1desloc}$: Desired offset from the base of the table in degrees.

$\theta_2 = 0$

$\theta_{1final} = \theta_{1init} + \theta_{1desloc}$

ISSN 1679-8902

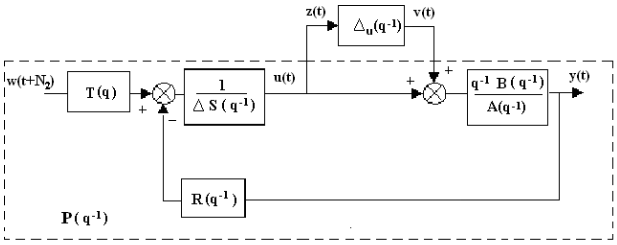


Figure 2.8: Direct multiplicative system

The stability limit is given by small gain theorem where the robustification to uncertainties is maximized by H_∞ norm minimization.

3 VIRTUAL SIMULATOR - MatLab/ Simulink

The control system which consists of cascade control loop for each axis was built in Simulink™ blocks. The set of control loops for position, velocity and torque can be part of the model of the drive system and control of a robotic joint. Position control of the handler can be implemented via feedback to each joint alone requiring the model of each joint. Finally, all joints must be coordinated for the dynamic model of the structure must be set (Figure 2.4).

In the problem study was implemented only position control loop coupled to the complete model of a robotic joint using open architecture so as to be easily implemented different control strategies for subsequent simulation, analysis and comparison of performance (Paracêncio, Rosário, Hermini and Sanchez, 2008).

Other elements of mechatronic systems (including possible external load) are represented by non-linear models, one for each engine. The simulator also includes a module for generating trajectories giving together with the trajectories of the axes as reference signals to the controller. And a graphical interface is available showing the results of the motions obtained from simple trajectory (Figure 3.1).

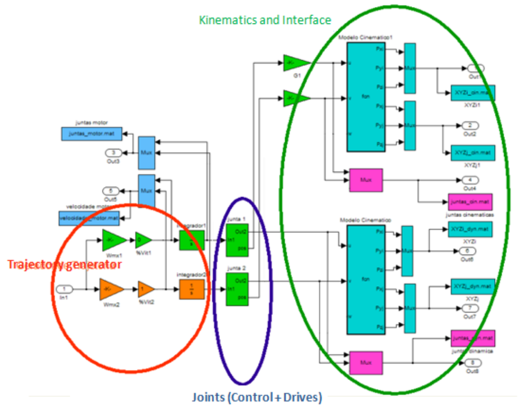


Figure 3.1: Simulator of a table of 2 DOF using Simulink™ environment

3.1 Module Generation of Trajectories

The trajectory generation module was implemented in MatLab™ with the objective of generating reference signals for each joint (velocity profile) based on kinematic characteristics of them. Allows an integrator to obtain the reference position of each joint. Figure 3.2 shows a profile path to be followed by each joint, with the acceleration, constant speed and braking at an interval of 10 seconds. The times of acceleration and braking were chosen based on the dynamic characteristics of the system (mechanical time constant). The trajectory generation module can be implemented by inverse kinematics from the kinematic model of the structure to be controlled using the inverse *Jacobian*, as presented earlier.

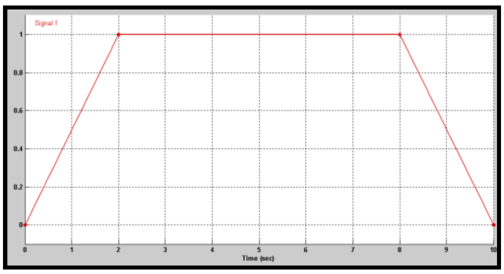


Figure 3.2: Reference signal for each board position

3.2 Drive Module

The drive module (Figure 3.3) presents the components for the electrical and mechanical drive system model and also the position controller.

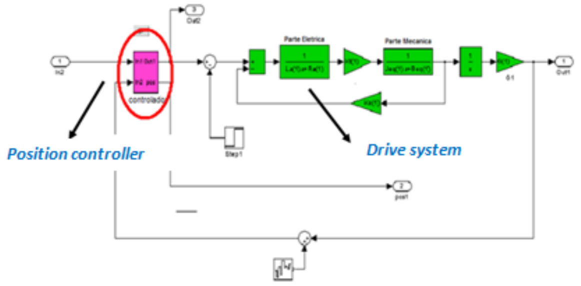
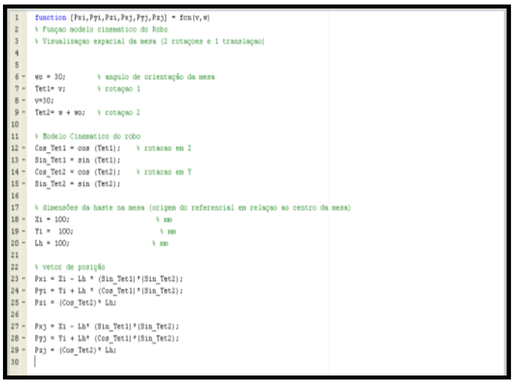


Figure 3.3: Loop position control of joint

3.3 Kinematic Module

The kinematic model of the mechatronic system was implemented using S-function features within the Matlab™ environment integrated with Simulink™ blocks (Figure 3.4).



```
function [Pxi,Pyi,Pzi,Pxj,Pyj,Pzj] = fcn(v,w)
% Funçao modelo cinematico do Robo
% Visualizaçao espacial da mesa (2 rotaçoes e 1 translaçao)


wo = 30;        % angulo de orientação da mesa
Tet1= v;        % rotaçao 1
v=30;
Tet2= w + wo;   % rotaçao 2

% Modelo Cinematico do robo
Cos_Tet1 = cos (Tet1);    % rotacao em Z
Sin_Tet1 = sin (Tet1);
Cos_Tet2 = cos (Tet2);    % rotacao em Y
Sin_Tet2 = sin (Tet2);

% dimensões da haste na mesa (origem do referencial em relaçao ao centro da mesa)
Xi = 100;                    % mm
Yi =  100;                    % mm
Lh = 100;                   % mm

% vetor de posição
Pxi = Xi - Lh * (Sin_Tet1)*(Sin_Tet2);
Pyi = Yi + Lh * (Cos_Tet1)*(Sin_Tet2);
Pzi = (Cos_Tet2)* Lh;

Pxj = Xi - Lh* (Sin_Tet1)*(Sin_Tet2);
Pyj = Yi + Lh* (Cos_Tet1)*(Sin_Tet2);
Pzj = (Cos_Tet2)* Lh;
```

Figure 3.4: Function *S-function* – Implemented in *Matlab*™ blocks *Simulink*™

3.4 Graphic interface

The simulator implemented in Simulink™ allows visualization of temporal outputs and inputs of the system. To better understand and analyze the spatial behavior of the system becomes essential to implement a graphic simulator of spatial motions that is described below.

After the simulations in the time domain through the simulator implemented in Simulink™ are obtained corresponding temporal data files all the variables (angular and cartesian, speed, power, control signal) that after treatment appropriate it is possible to verify important results available in the menu of Figure 3.5.

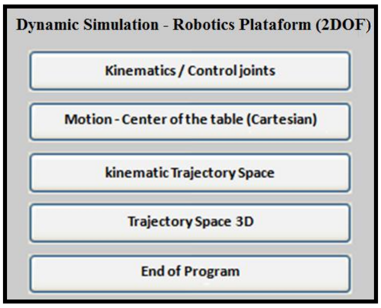


Figure 3.5: *Menu* - Robotic simulation with 2 DOF

A rod was inserted *100mm* in length (between points *A* and *B* of Figure 3.6) centered at the bottom of the table and tilted at an angle *30* degrees to enable the visualization of motions in relation to the cartesian coordinates of the joints . In order to better visualize the behavior of the system, the simulations presented show the rod rotating around the *Z* axis, in other words, moving only by the actuator 2. As the rod is in the center of the table position (*0*, *0*, *0*), the projected figure is a cone. Moving the rod to another position (P_x, P_y, *0*) and to move the junta appears to be a cylinder.

ISSN 1679-8902

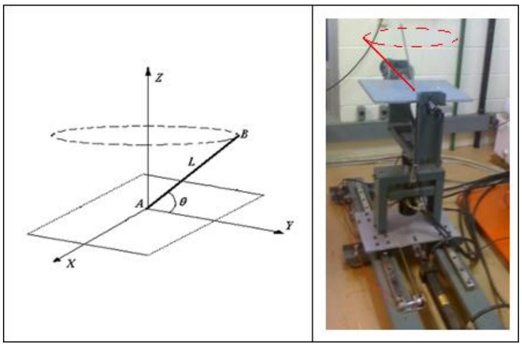


Figure 3.6: Rod placed on the table for visualization on motions during the simulations

In Figure 3.7, shows the positioning system of rotational joints where:

- the joint 1: puts the angles to drive and displays the graph of velocity and position;

- the joint 2:

a) Module 1: place the value of the angle of inclination of the table and is automatically calculated the radius of the rod.

b) Module 2: place the value of the radius rod and is automatically calculated by the slope of the table.

- has yet to enter the field full speed and time.

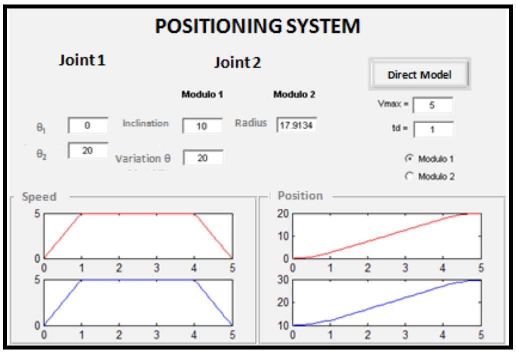


Figure 3.7: Positioning system

Therefore, we introduce some results for these simulation conditions:

i – Engine Speed:

Can you check the motion of every joint separately or together the three together and the speed of each joint is the chosen input parameter in the simulator (Figure 3.8).

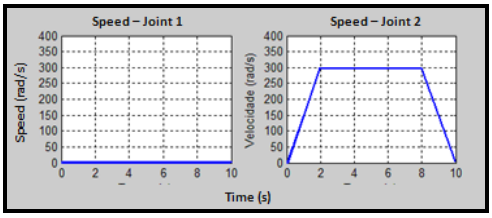


Figure 3.8: Velocity profile of the Joints - only driven actuator 2

ii – Engine Displacement:

Figures 3.9 and 3.10 has been motion considering the kinematic and dynamic displacement of the joints.

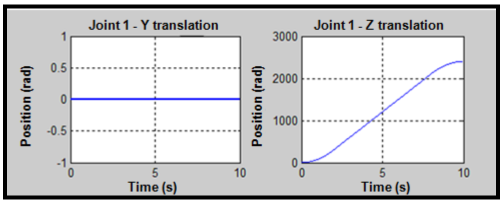


Figure 3.9: Kinematic motion - only driven joint 2

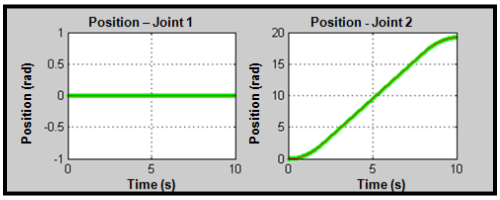


Figure 3.10: Dynamic motion - only driven joint 2

iii - Control Signal

The control signal can be viewed as the Figure 3.11.

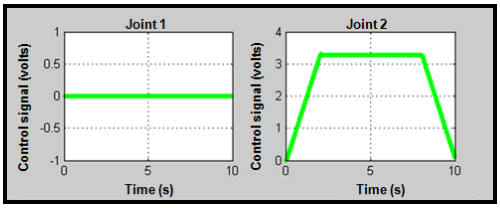


Figure 3.11: Signal control

iv - Motion – Central of the Table

The motion of the table´s center with respect to the probe placed on it. The idea is to show through the spatial visualization of the position presented by the rod at each instant of time (range 10 seconds) (Figure 3.12).

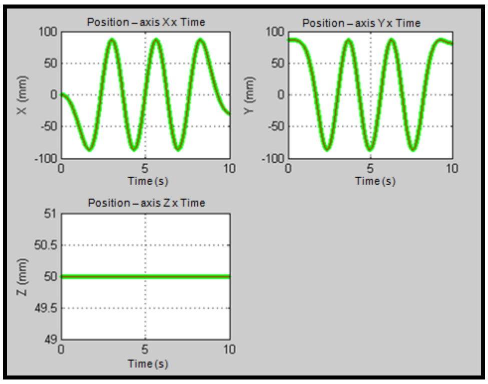


Figure 3.12: Motion of central table - - only driven actuator 2

v - Kinematic Trajectory Space

The Figure 3.13 represents the spatial motion of the table from the rod connected to the axes x, y and z.

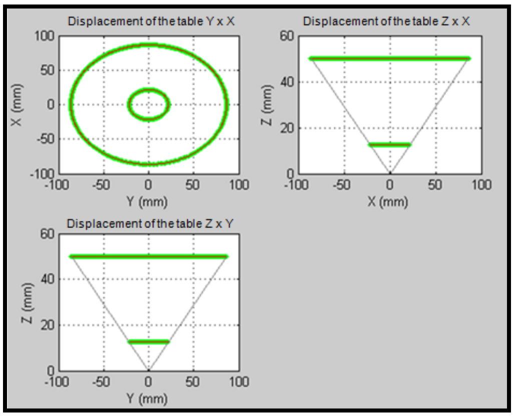


Figure 3.13: Vista axes X, Y and Z of the bureau in motion - Motion space rod

vi - Trajectory Space 3D

The Figure 3.14 shows the spatial motion of the rod, allowing to check possible errors and disturbances of displacement in cartesian space.

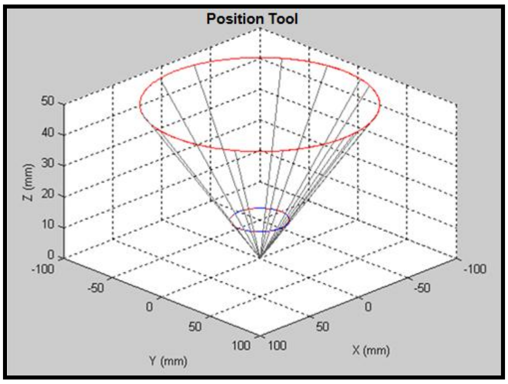


Figure 3.14: Stem position Space - Operation of only two actuator

4 Description of the Cell Robotics Study

In the laboratory of Integrated Automation and Robotics - FEM-UNICAMP unable to observe this condition in cooperative virtual programming the system illustrated in Figure 4.1. So it was implemented an integrated machining cell that consisting of two robots ABB: IRB1400 with load capacity of

ISSN 1679-8902

5 Kgf; IRB140 with load capacity of *5 Kgf* and a 3 DOF mechatronic device (Figure 4.1).

For implementation of off-line programming were considered the 3 DOF mechatronic devices as external axes robots ABB IRB 140 and IRB 1400. This is possible in the real robot allowing adjustments and integration of external axes and devices to be controlled by the control unit of the robot.

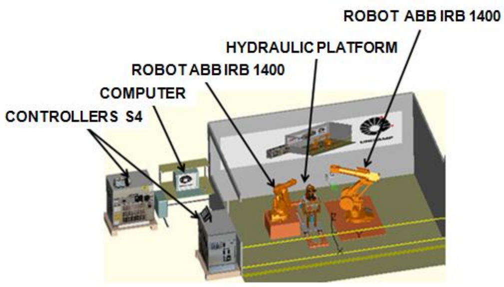


Figure 4.1: Automated virtual cell cooperative to be implemented in the FEM-UNICAMP

For validation of the robotic platform used for virtual operation was implemented a system of supervision and control developed in LabVIEW™ environment representing a task of implementation of complex generic work in the five faces of a cube using the 3 DOF hydraulic platform described in a previous working cooperatively with two industrial robots from ABB (IRB 140 and IRB 1400) and the study of activation and control of 3 DOF platform with drive and control interface were implemented in LabVIEW™ environment from the movement of a rod ready on the table.

In Figure 4.2a contains a piece on the table to be worked out cooperatively by two industrial robots where the piece will be positioned by mechatronic device to then be worked by industrial robots (case 1). Already in Figure 4.2b presents a rod placed under the table kinematic model validation, interface and drive control of the mechatronic device described in the previous section.

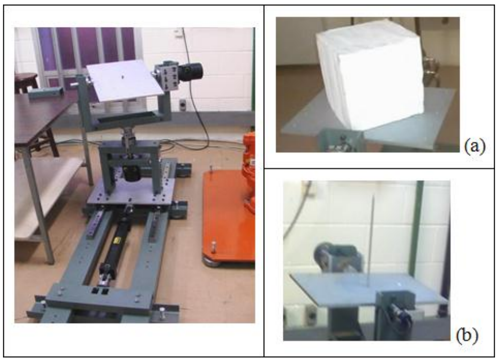


Figure 4.2: Platform positioning – LAR

4.1 Case 1: Collaborative System in Automation

4.1.2 Validation Environment

A complex infrastructure for experimental work in robotics is not always possible to be implemented because of high costs involved. The search for appropriate solutions is constant with the use of mathematical models capable of representing part of a real system without losing its generality to experimentally validate new control structures.

The group implemented a structured environment for cooperative work involving the integration between two robots ABB and 3 DOF for positioning platform, featuring a modular, hierarchical and open, can easily be used for integration of various mechatronic devices. The Figure 4.3 shows the structure of the task execution using a generic job indexed table and two industrial robots from ABB (IRB 140 and IRB 1400). And Figure 4.4 presents some validation tests.

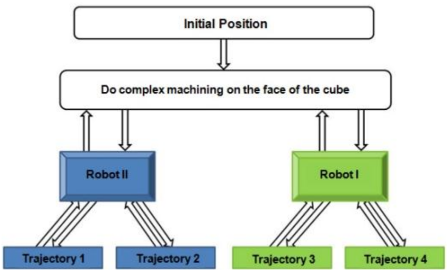


Figure 4.3: Structuring tasks

ISSN 1679-8902

The interface can be finding at the following address http:// 143.106.9.151. The Figure 4.4 presents images of operative cooperative tasks performed.

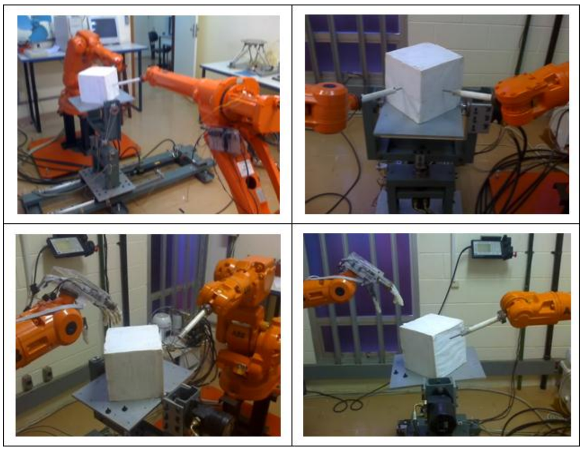


Figure 4.4: Integrated work cell – LAR

4.2 Collaborative Environments for Teaching and Research based on the WEB

Internet is revolutionizing science, industry and society through evolution of Information Technologies and Telecommunications. Thus, use of educational environments with computational resources allows greater access to new knowledge more quickly, leading to social impacts and new areas of research and development (Henriques, 2005).

The use of learning platforms or environments should provide information for the learner to evolve at their own rhythm and flexibility (Traylot, Heer and Fiez, 2003). These environments should promote the integration of knowledge, innovation and experience to solve small problems to motivate and improve the visualization of the continuity of learning. Aiming to use these aspects of virtual laboratories and Internet become allies in the learning process (Figure 4.5 illustrates the relationships shown) (López, Romeo and Guerrero, 2009).

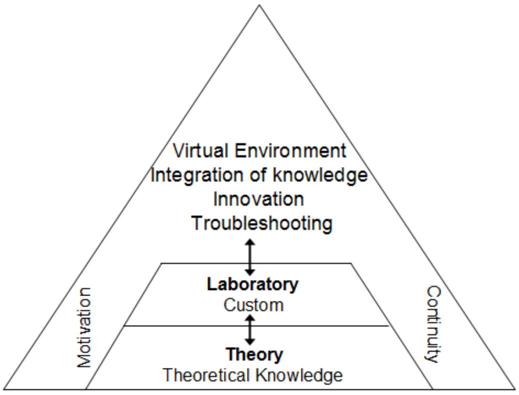


Figure 4.5: Environment for teaching and research virtual aspects

4.2.1 Screens Environment Implemented in LabVIEW™

The Figures 4.6, 4.7 and 4.8 show the screens implemented in LabVIEW™ of cooperative tasks performed.

Figure 4.6: HTML page proposal - Control panel implemented in LabVIEW™

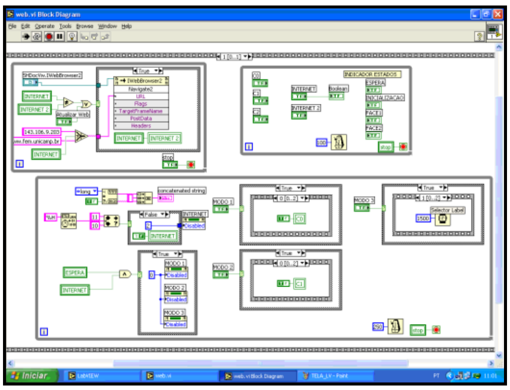


Figure 4.7: Screen block diagram implemented in *LabVIEW*™

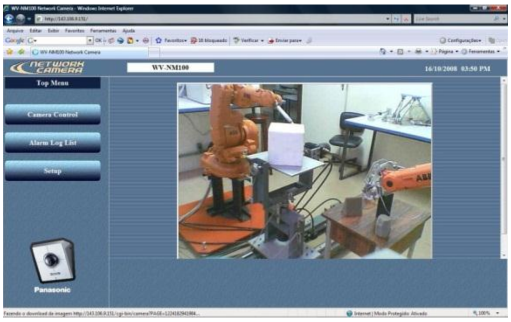


Figure 4.8: Screen display of CCD camera with fixed IP WEB CAM

4.3 Case 2: Implementation of Supervision and Control System Platform 3 DOF

Implementation of a simulator using building blocks, providing a modular, hierarchical and open, can easily be used for the simulations of various mechatronic devices.

Figure 4.9 show schematically the simulator implemented. This simulator consists of the following modules:

- Generation of reference trajectory signal joints;
- direct and inverse kinematic model;
- Drive and control;
- Graphical interface to allow viewing of the results obtained by the movement trajectories of references.

From subsection 4.3.5 will be presented at startup screens, positioning, orientation, calibration and control.

In the Laboratory of Integrated Automation and Robotics Faculty of Mechanical Engineering, UNICAMP, has implemented a platform for positioning with three degrees of freedom (robot PRR) to work cooperatively with two industrial robots (IRB 140 and IRB 1400 ABB™) with the objective to enable the cooperative work of robot manipulators in conventional machining operations and welding of complex mechanical devices that need more degrees of freedom to carry out complex paths (Figure 4.9).

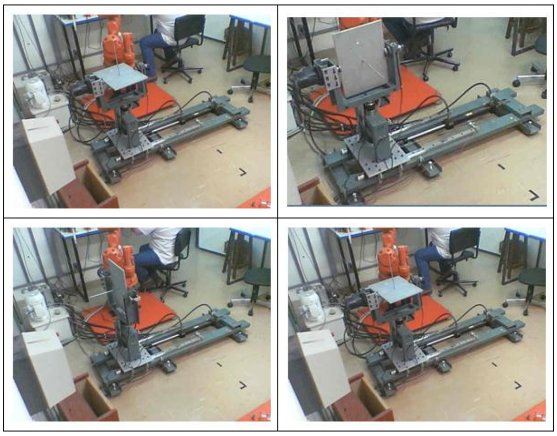


Figure 4.9: Integrated manufacturing cell – LAR

The following subsections will attempt to present a description of the software implemented in LabVIEW™ to drive and control of the three hydraulic actuators to drive a platform capable of positioning and orientation of a base of a table with two degrees of freedom.

4.3.1 Description of Operative Party

This platform consists of hydraulic actuators with positioning sensors. The first degree of freedom has a positioning cylinder for linear movement of the base in a particular direction (translational motion), inductive sensors final course and central positioning of the table. Two degrees of freedom are responsible for spatial orientation (robot RR) by angular motion of rotation hydraulic motors (rotation) consists of rotation sensors (incremental encoders). At the base of this table provided for the placement of a calibration rod with dimensions (length) will be available as variable parameter of the startup screen.

The software LabVIEW™ can be used for control and supervision. Control of robotic joints is accomplished through an interface D/A that drives the motors of the joints. A program of supervision and control in the computer is responsible for management and control of information from sensors and actuators of the system.

A display interface has been implemented in Windows environment and in this environment were developed supervisory screens that collect information from sensors of treatment programs, mathematical (kinematic modeling of the table) information system (speed controller parameters, number of points of the trajectory, etc..) initialization and automated calibration.

The interface of data acquisition used was the PCI 9112 from ADLINK Technology. This interface receives information from the encoders and the pot to determine the current position of the constituents together at the table. This information is compared with the reference values and after this interface is calculated using control algorithm send information to the controller output (position, velocity) drives to each degree of freedom of the table (Figure 4.10).

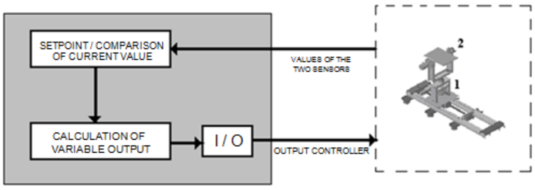


Figure 4.10: Outline of the controlled system

All references values are calculated in real time through a database, the same inverse kinematics with high nonlinearities. Other types of sensors can be included on the bench or simply through VI (Virtual Interface) software LabVIEW™.

4.3.2 System Supervision and Control

The system of supervision and control was implemented in LabVIEW™ language, often used industrially, in the design of devices related areas of measurement and control can use it to create custom applications that run on platforms NI (National Instruments) with I/O Reconfigurable based FPGAs. Together, LabVIEW™ FPGA and NI hardware for I/O Reconfigurable allow the creation of a flexible platform for developing sophisticated systems that were previously only possible with hardware designed so dedicated.

4.3.3 Description of Operating System

For implementation of the System of Supervision and Control of Hydraulic Platform was used to communication the interface PCI-9112 16-CH 12-Bit 110 kS/s Multi-Function DAQ Card/ Low-Profile National Instrumentation™ DAQ Card (Figure 4.13), consisting a module with eight analog inputs (A/D) to reading two encoders incremental rotational movement of the joints, a module of eight analog outputs (D/A) for powering the rotary cylinders and forward and reverse cylinder linear motion, a module of eight inputs and eight digital outputs to the sensors reading cylinder linear position sensors and final stroke, and pressing of keys (for example, keys, start and end of operation) and drive.

Figure 4.11: Communication Interface PCI-9112 16-CH 12-Bit 110 kS/s Multi-Function DAQ Card/ Low-Profile DAQ Card of NI Instrumentation

4.3.4 Control Interface

The user interface was developed using the LabVIEW™ software compatible with this interface that is enable to monitore and control of information two models: learning and reading the data file, thus, learning and recording of trajectories from kinematic modeling platform. And tool trajectories for industrial robots used terminal, storing information, read the position sensors (incremental encoders) via interface A/D and driving the hydraulic linear actuator for positioning the platform (forward and backward) and actuators Hydraulic rotating through the digital output interface after treatment of PID controller for each joint and digital input interface are responsible for acquiring information from external sensors (for example, keys, start and end of operation).

Thus, the entire management of this information is performed by a computer program of supervision and control resident on a PC where it was implemented an interface for viewing from the development of monitoring screens of information from sensors (end of course, security, keys logics, etc…), treatment programs and mathematical (kinematic modeling of the device), information about the status of the system (percentage of speed, the PID parameters, number of points of the trajectory, etc...) startup and automated calibration.

4.4 Implemented Screens

4.4.1 Startup Screen

The startup screen of the program allows the application to computational work in manual mode (controlled through dedicated buttons (calibration and control) on the control panel) or so computer through menus triggered off mouse (Figure 4.12). Four procedures are available to you: Calibration, Learning (Manual), Generate File and Control.

Upon initial implementation, the system is automatically driven to the calibration mode platform, which performs the procedure of reading and storing information from the position sensors for each of the actuators (encoders).

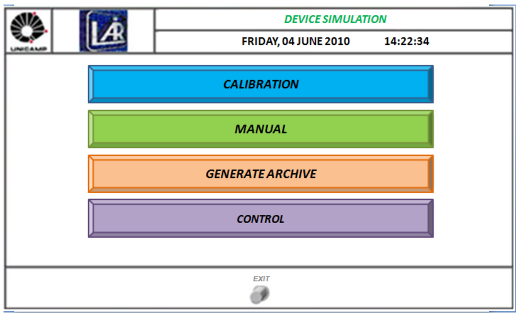
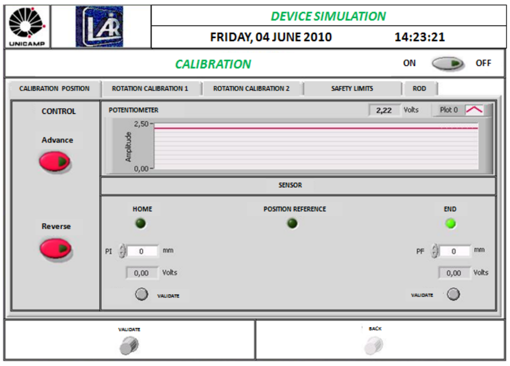


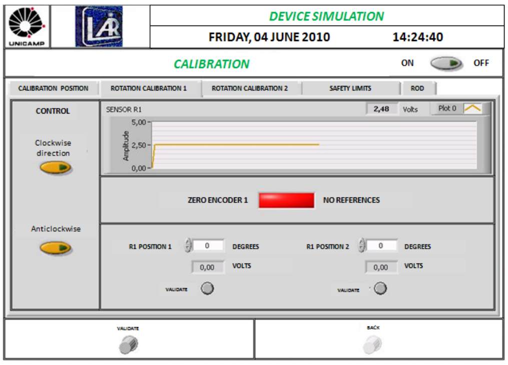
Figure 4.12: Screen primary program

4.4.2. Calibration Mode

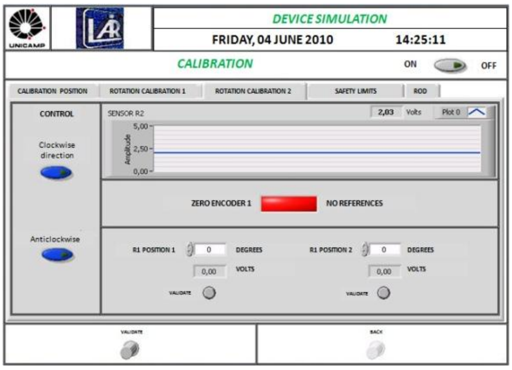
This mode allows the calibration and should always be used at startup and in case of checking user-positioning. The main screen consists of two buttons located on the left side of the screen calibration and make a display starting position responsible for information relating to position sensors for each joint (pot and encoders) indicating the position linear (translational motion) and angle (referring to the two rotational motions of the table). The Figure 4.13 shows this screen.



a) Calibration position linear actuator



b) Calibration position actuator 2



c) Calibration position actuator 3

Figure 4.13: Screenshots calibration

Other indicator lights show the user the zero position of the encoders (rotary motion) sensors and final course in the case of linear motion of the base platform. The parameters of sensors calibration and size of rod calibration ready at one point referenced (*x, y*) of the table (in this case, by default, $x = 0$, $y = 0$ and $L_{haste} = 0{,}20m$) shall be entered by the user and can no longer be modified after the calibration phase. Two buttons located on the bottom of the screen allow you to record these positions and get out of returning to the main menu screen shown above.

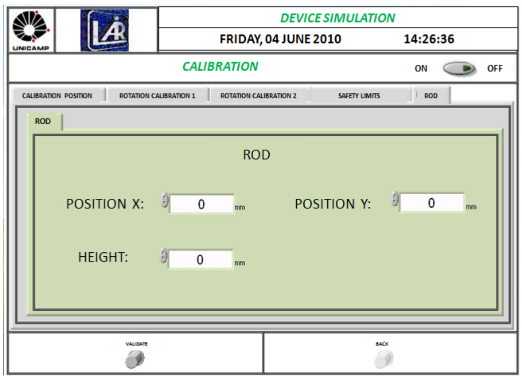


Figure 4.14: Screen calibration rod

Each step of the calibration procedure is performed by the user with interactive screens to allow the motion of actuators, sensors monitoring the positioning and initialization routine (zero). Figure 4.15 shows typical messages menu calibration.

Figure 4.15: Messages typical calibration menu

By completing the entire calibration process for each degree of freedom and validate them, automatically generates a screen as in Figure 4.16, with:

- limits placement of each joint;

- the value of the current position of each joint;

- power button / shutdown;

- buttons for activating the linear actuator back and forth for position;

- buttons to drive the rotary actuators clockwise and counterclockwise to the orientation.

ISSN 1679-8902

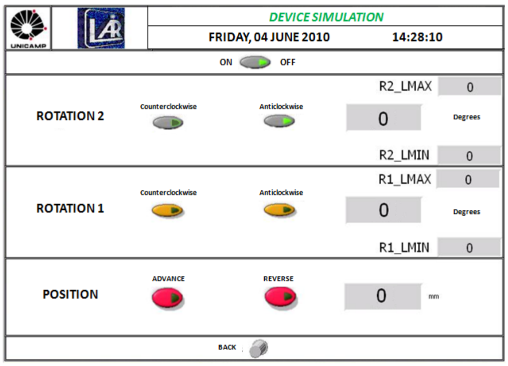


Figure 4.16: Validation of the generated screen after calibration

4.5 Generation File

4.5 Learning Mode

After startup of the platform (position of linear actuator to move the table until the center position sensor and calibration of positioning sensors for angular orientation of the table operations menu described above) the user can enter the learning mode. On this screen the user can perform motions in joint mode (rotations of the two degrees of freedom of the table) or in Cartesian over the end of the rod calibration (in this mode the program uses the kinematic model of the table considering the size and positioning of stem from the center of the rotary table). This mode allows you to make the generation of files rotational motions of the table from the motion of learning.

Angular mode (Figure 4.17a), the user performs the motions of each joint angle interpolation and storing for subsequent generation of motions in automatic mode. The position of the terminal stem in Cartesian coordinates is displayed on the screen. Cartesian mode (or position) shown in Figure 4.17b the user moves the table performing Cartesian motions in relation to the termination of the tool used (in this case the rod calibration) is shown on the screen the respective angular positions corresponding to the motion.

Tilt Mode:

Joint 1: rotation (degrees)

Joint 2: angle (degrees)

- Variation of displacement (degrees)

- Shows the Radius value for this slope

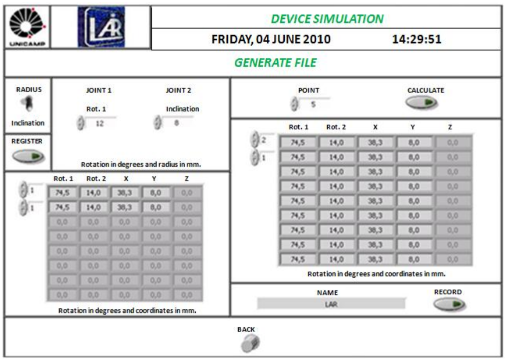
Mode Radius:

Joint 1: rotation (degrees)

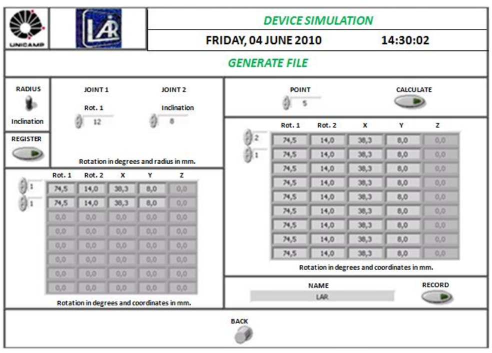
Joint 2: radius (mm)

- Variation of displacement (degrees)

- Display the value of slope (degree)



a) Mode of motion joints



b) Cartesian mode - positioning rod

Figure 4.17: Screen operation-generated files trajectories

4.5.1 Motion from standard file EXCEL™

From the file generated in EXCEL™ representing the motion of joints was implemented using LabVIEW™ software module to allow:

i) From the kinematic model presented in section 2 and 3 to convert these motions into reference signals (expressed in volts) to be compared with the position sensors (potentiometer of precision) to be sent to a PID position controller that provides signals command to the hydraulic actuators corresponding to the two degrees of freedom of rotation of the device.

ii) Implementation of Supervision and Control Software and Hardware Drive and Control in LabVIEW™ environment that will allow the automatic generation of motions together.

4.5.2 Mode Control

The mode control allows you to read a file motions generated menu file generation made before or after reading a typical file waveform generated in Excel™ presented earlier, that by a factor of scale will reduce the number of points in the reading file. The Figure 4.18 shows examples of typical screens of this module.

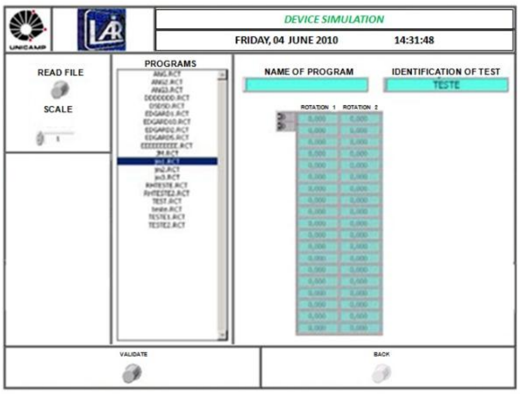


Figure 4.18: Typical screen control module

4.5.3 Screen Control (automatic mode)

The screen control in automatic mode (Figure 4.19) consists of basic modules for control, status of the program, graphic display of position sensors (encoders and inductive sensors for positioning the platform) and adjust the parameters of the position controller (PID).

At the same time the display consists of a button enabling the program to generate motions for the actuation of hydraulic valves and interrupt (with signage on display).

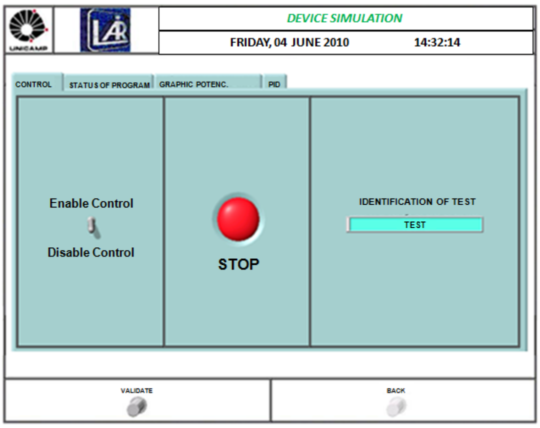


Figure 4.19: Screen control mode (auto)

4.5.4 Status Display

The status screen shows the different step of trajectory is shown the step number and value of current reference to be sent to the controller for comparison with the encoder and the percentage of evolution of the trajectory (Figure 4.20).

ISSN 1679-8902

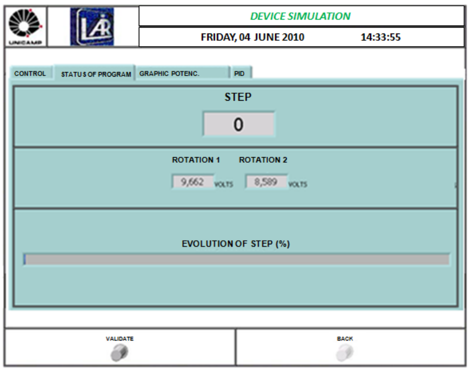


Figure 4.20: Typical screens of status evolution of trajectory

4.5.5 Screen Monitoring

The monitoring screen allows the monitoring time in the values relating to the joint position sensors (encoders) as shown in Figure 4.21.

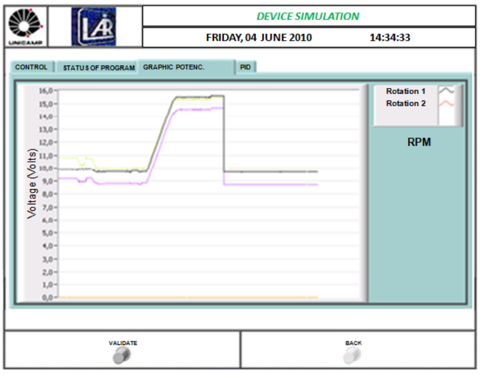


Figure 4.21: Screen sensors monitoring of joints (voltage x time)

4.5.6 Screen Adjustment Parameters of PID Position Controller

This screen allows adjustment of parameters of the PID (Proportional, Integral and Derivative) and choose the percentage of maximum error (Figure 4.22).

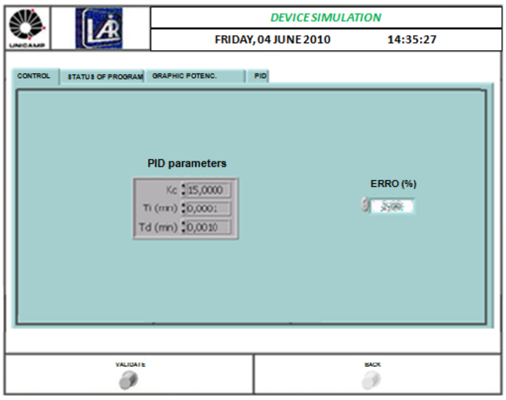
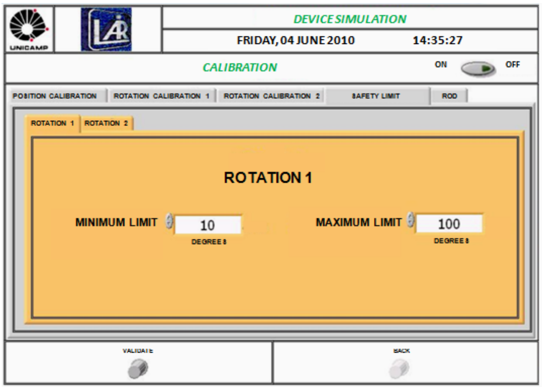


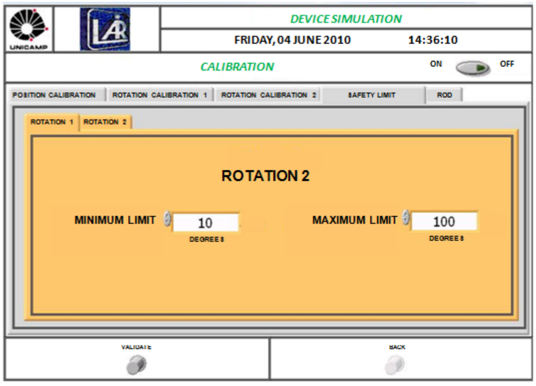
Figure 4.22: Screen adjustment parameters of PID controller

4.5.7 Safe Mode

In the implementation of the final program in LabVIEW™ were covered aspects related to safe use of the device. The aspects considered were setting limits on software and limits of travel of each rotary joint (information provided during the calibration phase) and final year of the base (information provided by inductive sensors arranged in the base). This protection consists basically in stopping the program (shut down the system drive) and alarm message on the screen to the user as shown in Figure 4.23.



a) Limits of safety rotary actuator 1



b) Limits of safety rotary actuator 2

Figure 4.23: Screen alarm operation mode (security)

4.5.8 Generation File Database

After loading the files a database consisting of information of operation of the file is automatically generated (Figure 4.24).

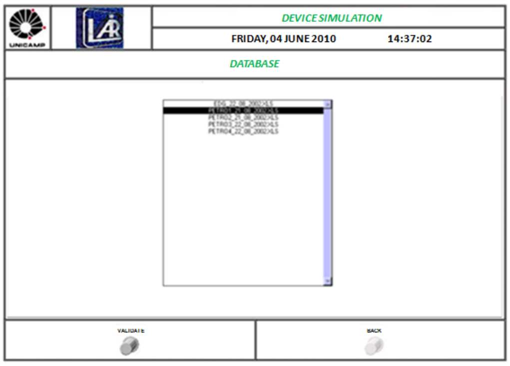


Figure 4.24: Database Files (Drive)

5 CONCLUSION

As a solution to integrate and implement a program executes the instructions programmed by a simple logic of communication. This solution does not meet generally a few types of cases, especially in activities that require quick responses in relation to motions of the robotic arms as well as the synchrony of motion.

To validate this research were implemented two studies:

a) Platform WEB collaborative automation: integration of two industrial robots with 3 DOF robotic devices to perform collaborative tasks through automated WEB;

b) Implementation of System of Supervision and Control of a robotic device with 3 DOF with emphasis on kinematic modeling and motion control.

REFERENCES

Aihara, C. K., Cosso, S. G., Saramago, M. A. P. e Rosário, J. M.: "Desenvolvimento de Aplicativos para Monitoramento de Variáveis de Controle de Processos Industriais", In: Aplicon, EEUSP São Carlos, Julho 2001.

David, S., Rosário, J. M.: "Modeling, Simulation and Control of Flexible Robots", CONTROLO'98, pp. 532-539, Coimbra, Portugal, 1998.

Henriques, R., V., B.: "Desenvolvimento de Metodologia para Planejamento de Trajetórias de Soldagem Robotizada", Escola de Engenharia da UFMG, Belo Horizonte, Tese de Doutorado, 2005.

López, G., Romeo, A. and Guerrero, J.: "Simulation Tools for Active Learning in Robot Control and Programming", 20th EAEEIE Annual Conference: Innovation in Education for Electrical and Information Engineering, Valencia. Spain, 2009.

Morcelli, J. C. M.: "Simulador Seqüencial de sistemas de Filas", Disponível em: http://www.inf.pucpcaldas.br/~morselli/, Acessado em Agosto de 2007.

Paracêncio, L. G. de Mello., Doctor Thesis, Department of Mechanical Design, FEM, UNICAMP, 2009.

Paracêncio, L. G. M., Rosário, J. M., Hermini, H. A., Sanchez, O. F. A., “Collaborative Mechatronical Platform for Integrated Robotics Environment”, International Workshop ROBOCONTROL´08, 2008, Bauru - SP.

Oliveira, E.: “Prototipagem Rápida em Sistemas Mecatrônicos Baseada em Instrumentação Virtual”, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2008, 271p., Tese (Mestrado).

Rosário, J. M., Cassemiro, E. R.: “Modelling and Control of Anthropomorphic Gripper in Industrial Robot”, Proc. IEEE 12th International Conference on Advanced Robotics, Coimbra, Portugal, 2003.

Rojas, J. H. C.: “Metodologia de Modelagem, Simulação e Programação *off-line* de Robôs e Mecanismos Mecatrônicos Integrados e Direcionados a Células de Manufaturas Flexíveis”, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2004, 168p. Tese (Doutorado).

Rosário, J. M., Oliveira C. and Sá C.E.A.: “Proposal Methodology for the Modeling and Control of Manipulators”, International Journal of the Brazilian Society of Mechanical Engineering, Vol. XXIV - nº. 3, Julho 2002.

Sanchez, O. F. A., Rosário, J. M., Uribe, A. J., Paracêncio, L. G. M.,: “Estratégia de Controle para Mão Mecânica Baseado em Informação de Forma: Experiências com Mão Mecatrônica MUC-1”, International Workshop ROBOCONTROL´08, 2008, Bauru - SP.

Traylot, R. L., Heer, D. and Fiez T.S.: “Using an Integrated Platform for Learning™ to Reinvent Engineering Education”, In: IEEE Transactions on Education, Vol. 46, nº. 4, p. 409-419, November 2003.