Robot Axis Dynamics Control Using A Virtual Robotics Environment

E. R. Cassemiro J. M. Rosario
UNICAMP - University of Campinas
Faculty of Mechanical Engineering
13083-970 Campinas - SP, Brazil
rosario@fem.unicamp.br

D. Dumur
SUPELEC
Department of Automatic Control
91192 Gif-sur-Yvette, France
Didier.dumur@supelec.fr

Abstract

Robots are complex electromechanical systems where several electric drives are employed to control the movement of articulated structures. In industrial environments they must perform tasks with rapidity and accuracy in order to produce goods and services with minimal production time. These procedures require the use of flexible robots which can act in a large workspace, thus subject to important parameters variations, with efficient control algorithms. The progress of technology and the close relationships among several sciences, such as micro-electronic, software engineering and communications, open space for a great development in the robotics area and automation process. Generalized Predictive Control (GPC) has shown to be an effective strategy in many fields of applications, with good time-domain and frequency properties (small overshoot, improved tracking accuracy and disturbance rejection ability, good stability and robustness margins), able to cope with important parameters variations. This paper presents an application of GPC to a robot trajectory control with a comparison between classical PID and GPC controllers within an original Virtual Environment.

1 Introduction

The number of robots working in industry significantly increases due to their operation capacities in terms of flexibility, rapidity and accuracy [5]. In most industrial applications, the robot tasks are programmed by learning without need of a geometrical model. In this way, its trajectory is defined through a set of angles associated to the angular movement of each degree of freedom of the robot. After interpolation, these angles will act as reference signals for positioning controllers located at each joint that compare the signals deriving from the position sensors of the joints [5].

However, as quick flexible manipulators are essential to achieve this performance leading to a minor production time and small energy consumption [6, 7], more resourceful control algorithms must be implemented, which can cope with important parameters variations, e.g. inertia variations.

Usually, the integration of industrial robots and mechatronic devices into a Flexible Manufacturing Cells (FMC) involves modeling methodologies using automation formalism. The environment modeling by Computer Aided Design (CAD) system can been associated with robot's control performance including: equipment and correlated mechanisms, mathematical modeling of the robot (direct and inverse kinematics) and its connected devices, as also, coordination and integration of the robot's movements with other devices.

Figure 1 shows a FMC particular application based on the coordination and integration of two industrial robots and a mechatronic device (table) with 3 DOF (PRR robot) developed for accurate welding work purposes. This device can assist in tasks for which traditional manipulators have difficulties to reach some parts of the piece. For that, the table is synchronized with the manipulators allowing them to carry out complex tasks without adjustments.

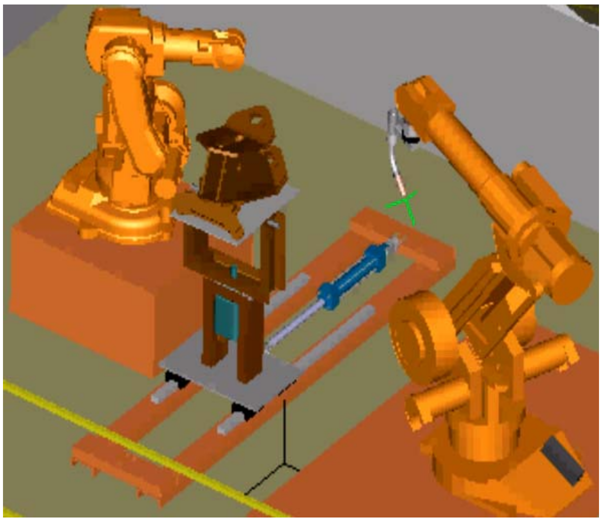

Figure 1: Flexible Manufacturing Cell.

This paper only focuses on the study of the 3 DOF table, without taking into account the synchronism with the robots, as well as the modeling and simulation of this table, particularly emphasizing the development

0-7803-9402-X/05/$20.00 © 2005 IEEE

and implementation of robotic joint position controllers. The paper is organized as follows. Section 2 provides a description of the table, including kinematics, dynamic and actuator modeling. Section 3 presents the advanced predictive axis control structure implemented under the RST formalism. Section 4 is dedicated to the results obtained within a virtual robotics environment. Finally Section 5 proposes some conclusions and future trends.

2 Virtual robot environment description

This section presents the modeling and simulation of the three degree-of-freedom (3 DOF) robot, leading to the design of a virtual environment using electric and mechanical libraries blocks in combination with SIMULINK™ blocks (Fig. 2).

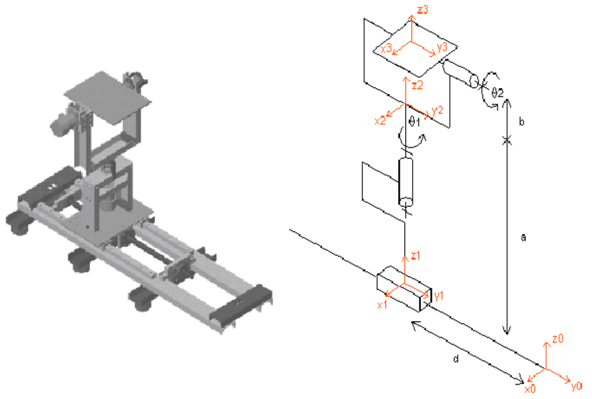

Figure 2: 3 DOF robot and related movements.

The design of such a simulator requires definition and modeling of the three main joints connected to the other parts of the manipulator through gear boxes. The main elements of these robotics joints are brushless DC motor drives, axes inertia, gears and control blocks. The control system itself, which essentially consists of cascaded control loops (for each axis), is built with SIMULINK™ blocks. The inner speed and torque control loops are part of the drive model; only the position loop is explicitly modeled, as different control strategies will be further compared. In fact, the position control of the manipulator can be implemented through the control feedback of each isolated joint [4], requiring the model of each joint. At the end, all joints must be coordinated as shown on Figure 3, so that the dynamic model of the structure has to be defined.

Other elements of the manipulator (including loads) are represented by three nonlinear models, one for each motor drive. The simulator also includes a path generation module, providing the joints with the axis trajectories as reference signals to the control parts. Finally, a graphical interface is available, showing results of joints movements obtained through typical trajectories.

2.1 Control structure including kinematics

For many operations, the operator defines the tasks, or reference trajectories of the controller in relation to a coordinate system, that is fixed to the end-effector of the robot (in the Cartesian space). But the desired movements (expressed in angular coordinates) and the control laws are in different coordinate systems, requiring the implementation of fast algorithms for the inversion of the kinematics model and generation of the reference trajectory in angular coordinates (Fig. 3).

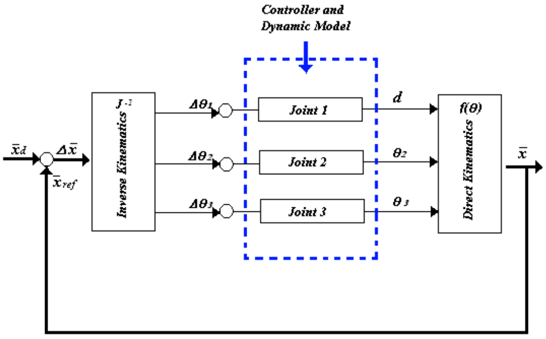

Figure 3: Control structure with kinematics.

2.2 Kinematics model

The geometrical model of a 3 DOF robot provides the position $(p_x \;\; p_y \;\; p_z)$ and orientation $(\psi, \;\theta, \;\phi)$ of the table with respect to a coordinate system fixed at the centre of the table, as a function of its generalized coordinates (translational and rotational joints), that is:

$$\mathbf{x} = f(\boldsymbol{\theta}) \qquad (1)$$

where $\boldsymbol{\theta} = \left(d \;\; \theta_2 \;\; \theta_3\right)$ as joints position
$\mathbf{x} = \left(p_x \;\; p_y \;\; p_z \;\; \psi \;\; \theta \;\; \phi\right)$ as position table vector

The function f enables to calculate the movement of the end-effector resulting from the movement of the joints. This function is nonlinear and has nontrivial analytical solution.

This relation may be expressed mathematically by a matrix connecting the system of fixed coordinates in the base of the robot with a system of coordinates associated to the robot. This matrix, so called homogeneous transition matrix, is obtained through the product of the homogeneous transformations matrix[8], $\mathbf{A}_{i,i-1}$, linking the system of coordinates of element i with the system of the previous element $i-1$, that is:

$$\left\lfloor \mathbf{n}^{\mathrm{T}} \;\; \mathbf{s}^{\mathrm{T}} \;\; \mathbf{a}^{\mathrm{T}} \;\; \mathbf{p}^{\mathrm{T}} \right\rfloor = \mathbf{T}_n = \mathbf{A}_{0,1}\,\mathbf{A}_{1,2}\,\mathbf{A}_{2,3} \qquad (2)$$

with $\mathbf{p} = \left\lfloor p_x \;\; p_y \;\; p_z \right\rfloor$ position vector
$\mathbf{n} = \left\lfloor n_x \;\; n_y \;\; n_z \right\rfloor$, $\mathbf{s} = \left\lfloor s_x \;\; s_y \;\; s_z \right\rfloor$, $\mathbf{a} = \left\lfloor a_x \;\; a_y \;\; a_z \right\rfloor$
orthonormal vector that describes the orientation

The kinematics description of this robot is performed through the Denavit-Hartenberg procedure, after isolating the four parameters $\boldsymbol{\theta}_i$ $\mathbf{r}_i$, $\mathbf{d}_i$ and $\boldsymbol{\alpha}_i$ (Table 1 and Figure 2). These coefficients enable to represent the different positions of the parts of this mechanical device.

Axis	α_i	d_i	$\boldsymbol{\theta}_i$	$\mathbf{r}_i$
1	0	d	0	0
2	0	0	θ_2	a
3	θ_3	0	0	b

Table 1: Denavit-Hartenberg parameters.

The position vector $\left(p_x \ p_y \ p_z\right)$ and the orientation angles $(\psi \ \theta \ \phi)$ solution coming from an object with dimensions $\left(L_x \ L_y \ L_z\right)$ placed in the centre of the table are:

$$\begin{aligned} p_x &= L_x\, c_2\, c_3 - L_y\, s_1 + L_z\, c_2\, s_3 \\ p_y &= L_x\, s_2\, c_3 + L_y\, c_2 + L_z\, s_2\, s_3 + d \\ p_z &= -L_x\, s_2 + L_z\, c_2 + (a+b) \end{aligned} \qquad (3)$$

$$(\psi \ \ \theta \ \ \phi) = \mathrm{rot}(x,\phi)\,\mathrm{rot}(y,\theta)\,\mathrm{rot}(z,\psi)$$
$$= \left[\begin{array}{c|c|c} c\phi\, c\theta & -c\phi s\theta s\psi - s\phi\, c\psi & c\phi\, s\theta\, c\psi + s\phi\, s\psi \\ s\phi\, c\theta & -s\phi s\theta\, s\psi + c\phi\, c\psi & s\phi\, s\theta\, c\psi - c\phi\, s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{array}\right] \qquad (4)$$

with

$$\theta = ATAN\,2\left[\frac{-n_Z}{c\phi\, n_x + s\phi\, n_y}\right], \ \phi = ATAN2\left[\frac{n_y}{n_x}\right]$$

$$\psi = ATAN\,2\left[\frac{s\varphi\, a_x - c\varphi\, a_y}{-s\varphi\, s_x + c\varphi\, s_y}\right]$$

$$\begin{aligned} & n_x = c_2 \qquad n_y = s_2 \qquad n_z = 0 \\ & s_x = -s_2\, c_3 \quad s_y = c_2\, c_3 \quad s_z = s_3 \\ & a_x = s_2\, s_3 \quad a_y = -c_2\, s_3 \quad a_z = c_3 \\ & c_i = \cos(\theta_i) \quad s_i = \sin(\theta_i) \\ & c\phi = \cos(\phi) \quad s\phi = \sin(\phi) \ \text{(idem for } \theta \text{ and } \psi\text{)} \end{aligned}$$

2.3 Inverse Kinematics model

The elaboration of references in angular coordinates referring to the tasks defined in the Cartesian space is expressed mathematically by the inversion of the geometrical model, that is:

$$\boldsymbol{\theta} = f^{-1}(\mathbf{x}) \qquad (5)$$

This transformation elaborates the references given to the axis control parts.

2.4 Dynamic model

As previously mentioned, the control of each joint is considered in an independent way, with any coupling effect. To take these effects into account, and to solve the trajectory problem, the dynamic control involves the determination of the inputs, so that the drive of each joint moves its links to the position values with required speed. The dynamic model of a robotic joint can be derived through the Euler-Lagrange formulation that expresses the generalized torque [2]. The manipulator dynamic behavior is described by a group of differential equations called dynamic equations of motion. For a three degree of freedom rigid manipulator, the equations are:

$$\tau_i(t) = J_i(\theta(t))\,\ddot{\theta}_i(t) + C_i\left(\theta(t),\dot{\theta}(t)\right) + Q_i(\theta(t)) \qquad (6)$$
$$i = 1,\Lambda\ ,3$$

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, as given in the block diagram of Figure 3, the input references, obtained in angular coordinates from the trajectory interpolator, are then 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 (Eq. 6).

2.5 Actuator model

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

$$\begin{aligned} u(t) &= L\frac{\mathrm{d}\,i(t)}{\mathrm{d}\,t} + R\,i(t) + K_E\,\frac{\mathrm{d}\,\theta_m(t)}{\mathrm{d}\,t} \\ T_m(t) &= J_{eq}\,\frac{\mathrm{d}^2\,\theta_m(t)}{\mathrm{d}\,t^2} + B_m\,\frac{\mathrm{d}\,\theta_m(t)}{\mathrm{d}\,t} \\ T_m(t) &= K_T\ i(t) \end{aligned} \qquad (7)$$

where $T_m(t)$ is the motor torque, $\theta_m(t)$ the angular position of the motor, $i(t)$ the motor current, L, 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 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 …

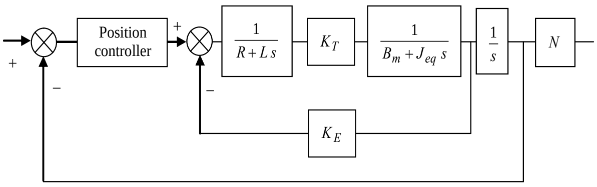

Figure 4: Block diagram of the joint axes.

3 Generalized Predictive Control (GPC)

One advantage of the virtual environment is the possibility to implement and test advanced axis control strategies, in particular Predictive Control, well known structure providing improved tracking performances. This philosophy aiming at creating an anticipative effect using the explicit knowledge of the trajectory in the future, can be summarized as follows [1, 3,]:

- Definition of a numerical model of the system, to predict the future system behavior,
- Minimization of a quadratic cost function over a finite future horizon, using future predicted errors,
- Elaboration of a sequence of future control values, only the first value is applied both on the system and on the model,
- Repetition of the whole procedure at the next sampling period according to the receding horizon strategy.

3.1 CARIMA model

The CARIMA (Controlled AutoRegressive Integrated Moving Average Model) form is used as numerical model of the system, in order to cancel steady state error in case of step input or disturbance by inducing an integral term in the controller:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{\xi(t)}{\Delta(q^{-1})} \qquad (8)$$

where u is the control signal applied to the system, y the output of the system, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, A, B polynomials in the backward shift operator q^{-1}, of respective order n_a and n_b, ξ an uncorrelated zero-mean white noise.

3.2 Prediction equation

Assuming the best estimate of disturbance in the future equals the mean value of the signal, i.e. zero, a j-step ahead predictor is then defined in a polynomial form, as follows:

$$\hat{y}(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1})\Delta u(t-1)}_{\text{free response}} + \underbrace{G_j(q^{-1})\Delta u(t+j-1)}_{\text{forced response}} \qquad (9)$$

where all required unknown polynomials are derived solving Diophantine equations.

3.3 Cost function

The GPC strategy minimizes a weighted sum of square predicted future errors and square control signal increments:

$$J = \sum_{j=N_1}^{N_2} \left(\hat{y}(t+j) - w(t+j)\right)^2 + \lambda \sum_{j=1}^{N_u} \Delta u(t+j-1)^2 \qquad (10)$$

assuming $\Delta u(t+j) = 0$ for $j \geq N_u$. Four tuning parameters are required: N_1, the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control weighting factor.

3.4 RST form of the controller

The minimization of the previous cost function results in the predictive controller derived in the RST form according to Figure 5 and implemented through a difference equation:

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q)w(t) \qquad (11)$$

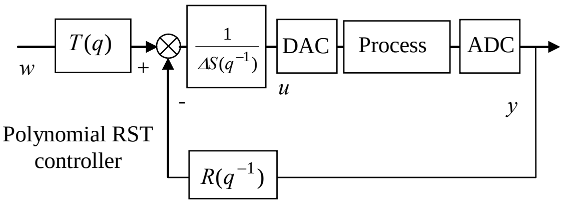

Figure 5: GPC in a RST form.

The main feature of this RST controller is the non causal form of the T polynomial, creating the anticipative effect of this control law. The degrees of the 3 polynomials are as follows:

$$\begin{aligned} &\text{degree}\left(R(q^{-1})\right) = \text{degree}\left(R(q^{-1})\right) \\ &\text{degree}\left(S(q^{-1})\right) = \text{degree}\left(B(q^{-1})\right) \\ &\text{degree}\left(T(q)\right) = N_2 \end{aligned} \qquad (12)$$

4 Tests in the virtual environment

Previous sections have described the whole virtual environment related to the 3 DOF manipulator, including motor drives, gear boxes, kinematics and dynamics models, and predictive axis controllers, designed with electric drives and SIMULINK™ libraries (Figure 5). This section will now present results obtained within this environment, comparing for this 3 DOF manipulator performances obtained with classical PID controllers and GPC advanced control laws. Simulations describe below consider 3D trajectories issued from the path generation module.

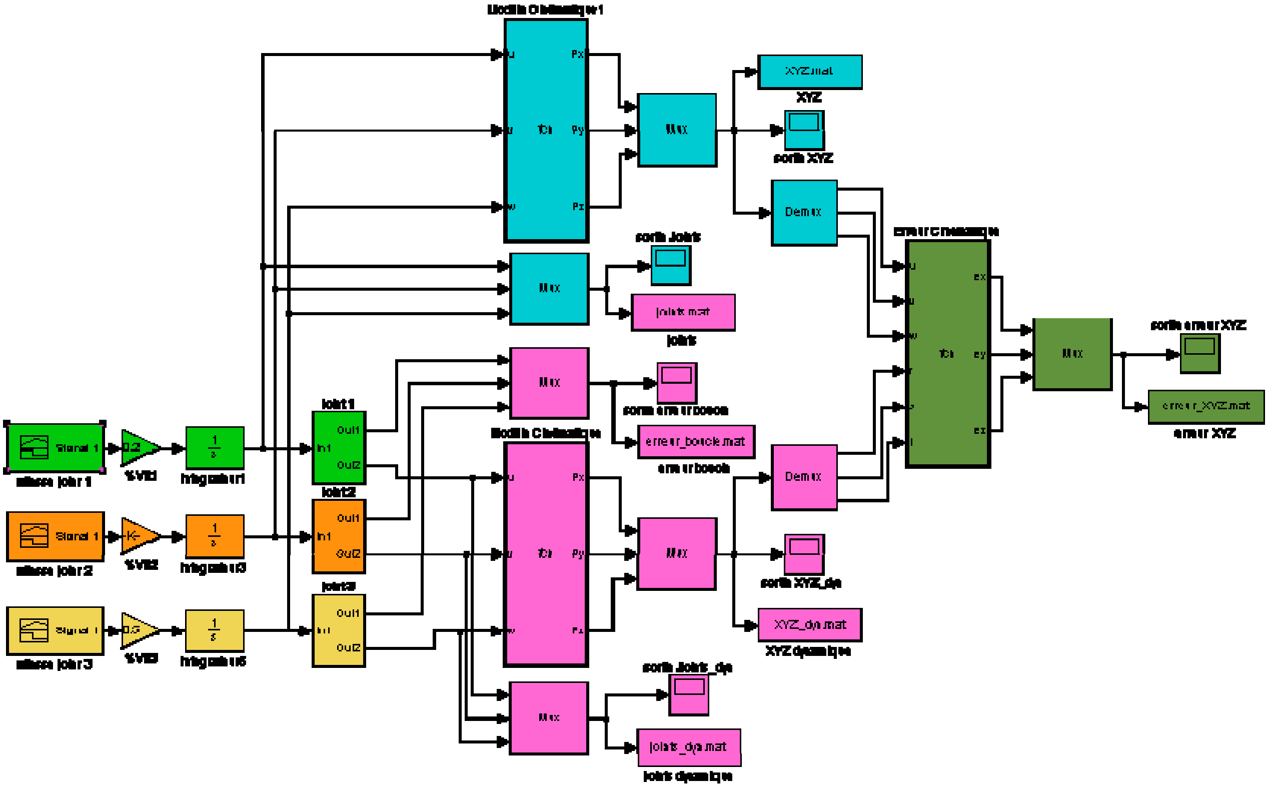

Figure 5: Architecture of the virtual environment in SIMULINK™.

4.1 Actuator parameters

The system considered here, used for supervision and control [6, 7], developed at UNICAMP, Brazil, includes three DC motors, a 1:100 gear box, a ball screw transmission (only for axis 1) and incremental encoders, with parameters given in Table 2.

Motor Siemens 1FK7060	
Inertia (kgm^2)	$0.71\ 10^{-3}$
Weight (kg)	8
Mechanical time constant (ms)	1.94
Voltage constant (V/rad/s)	0.807
Torque constant (Nm/A)	1.33
Inductance (mH)	14.7
Resistance (Ω)	1.44

Table 2: Motor Parameters.

4.2 GPC tuning parameters

The axis controllers are designed independently following the mechanism developed in Section 3, resulting in three RSTs, considering the same axis motor but with different inertia on the motor side due to different geometrical features for each axis. The tuning parameters given in Table 3 have been chosen to provide good stability and robustness margins [1].

Axis	N_1	N_2	N_u	λ
1	1	8	1	92
2	1	8	1	107.3
3	1	8	1	126

Table 3: GPC tuning parameters for each joint.

4.3 Simulation scenario

The scenario considers the rotational trajectory around the z-axis followed by a rod of 100 mm length (L), located at the center of the table of the 3 axis robot of Figure 2, with a 30^o inclination angle. The desired path is thus a revolution cone, as shown in Figures 6 and 7. To do that, only the second axis is moving, but disturbances are added on each axis.

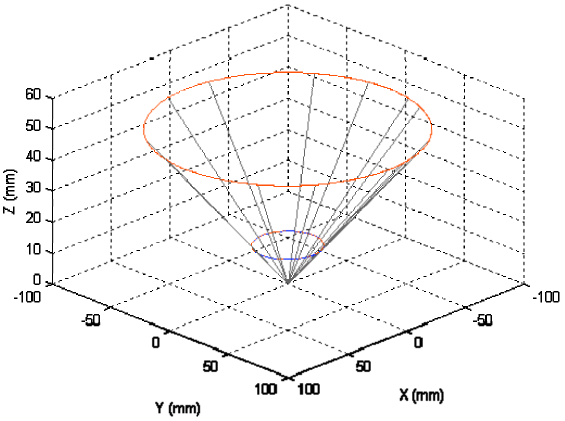

Figure 6: Rod spatial trajectory

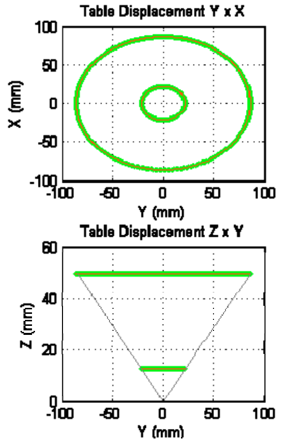

Figure 7: Rod trajectory in YX, XZ, YZ planes

4.4 Simulation results

A first simulation presents results obtained with RST/GPC axis controllers, considering the previous scenario, with velocity outputs given in Figure 8 and position errors in Figure 9. As mentioned before, only the second joint is operating, but disturbances can be seen on each joint.

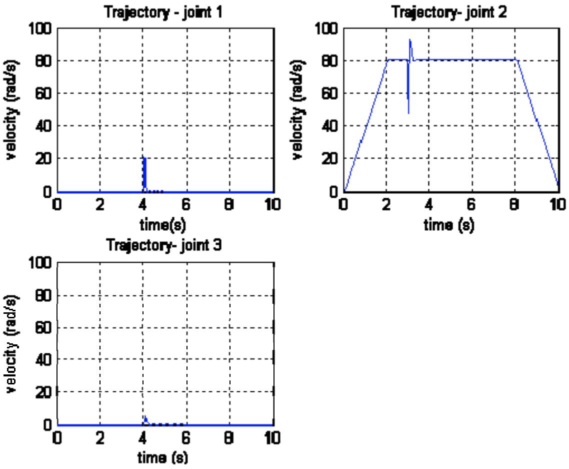

Figure 8: Velocity output for each joint GPC case with disturbances.

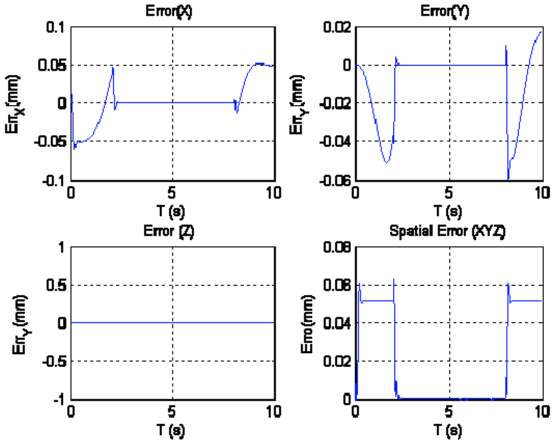

Figure 9: Axis and spatial position errors GPC case with disturbances.

These results are now compared with those obtained using classical industrial PID axis controllers, with the same scenario and same disturbances. Similarly, Figure 10 shows the resulting velocity outputs and Figure 11 the position errors. The PID was tuned as best as possible, showing a slower disturbance rejection dynamics compared to GPC (Figure 8 vs. Figure 10) a more unstable behavior (Figure 9 vs. Figure 11). Tracking performances offered by GPC laws are clearly emphasized on Figure 9, with very small tracking errors.

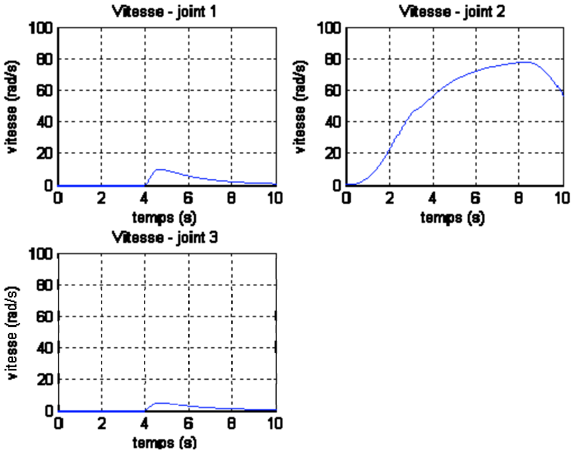

Figure 10: Velocity output for each joint PID case with disturbances.

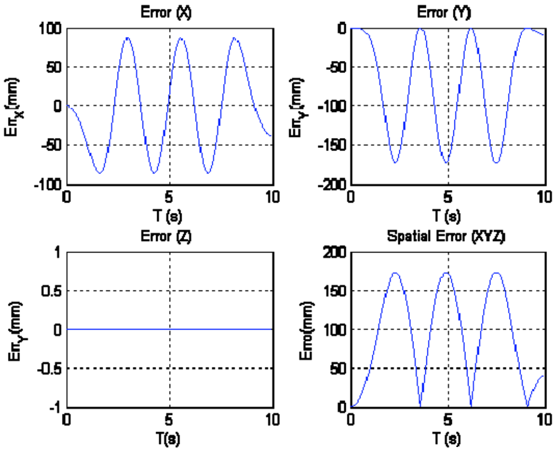

Figure 11: Axis and spatial position errors PID case with disturbances.

Globally, the results analysis shows that the anticipative effect of the GPC law can provide better performances, even if the controllers were designed neglecting the coupling effect between each axis. In this direction, GPC is less sensitive to inertia variations (appearing as every axis acts on the other ones) than PID. This significant simulation shows the robustness of GPC, so that the inertia variation can be considered as a disturbance performing on the system.

5 Conclusions and further works

This paper has developed an advanced control architecture for a 3 DOF manipulator, composed of a table moved by one translational and two rotational movements. For that purpose, a complete modular virtual environment has been designed using SIMULINK™ framework. This simulator includes the kinematics and dynamics models as well as the axis controlled loops built around the actuator model, i.e. DC motor with gear box and ball screw for the translational joint. A trajectory generation module and a graphical interface have been developed to help the user in testing realistic manipulator configurations. To emphasize the modularity of this virtual environment, electrical drive and controller libraries have been integrated as well; additional user-defined plug-in modules can be added very easily.

In this direction, the axis controllers have been structured under the RST formalism, which corresponds to the generic framework for numerical control. From this form, a GPC control law has been implemented on each axis, simply designed without taking into account coupling effects, providing improved performances in terms of rapidity, cancellation of overshoot, accuracy, disturbance rejection and robustness towards inertia variations and non linearities. This last point is one of the main challenges of robot control, mainly when large workspace is involved, because inertia can present important variations.

For comparison purposes, simulations have been given obtained with a classical PID axis control strategy. In this case, the errors remain important, and influence of disturbances and coupling effects not sufficiently minimized.

Furthers works will look at robustification of the GPC strategy against measurement noise and parameters uncertainties, as well as real validation of the developed control algorithms accomplished through experimental implementation.

6 Acknowledgement

The authors acknowledge support of the “Conselho Nacional de Desenvolvimento Científico e Tecnológico”- CNPq, and “Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES”- Brazil, through a collaborative work between the Control Department of SUPELEC, France, and the Laboratory of Automation and Robotics of UNICAMP, Brazil.

References

[1] P. Boucher, D. Dumur, “Predictive Motion Control”, *Journal of Systems Engineering, Special Issue on Motion Control Systems*, Vol. 5, pp.148-162, *Springer-Verlag,* London, 1995.

[2] M.W. Spong, M. Vidyasagar, “Robot Dynamics and Control”, *John Wiley & Sons*, New York, 1989.

[3] D. W. Clarke, C. Mohtadi, P.S. Tuffs, “Generalized Predictive Control”, Part I "The Basic Algorithm", Part II "Extensions and Interpretation, *Automatica,* 23(2):137-160, 1987.

[4] J.J. Craig, “Introduction to Robotics: Mechanics and Control”, Second edition, *Addison-Wesley Publishing Company*, 1989.

[5] S. David, J.M. Rosário, “Modeling, Simulation and Control of Flexible Robots”, *CONTROLO’98*, pp. 532-539, Coimbra, Portugal, 1998.

[6] K.B. Pimenta, J.P. Souza, J.M. Rosário, D. Dumur, “Control of Robotic Joints with Generalized Predictive Control (GPC)”, *RADD’2001*, Vienne, Austria, 2001.

[7] K.B. Pimenta, J.M. Rosário, “The Using Techniques of Generalized Predictive Control (GPC) Applied on Control of Robotic Joints”, *Asiar*, Bangkok, Thailand, 2001.

[8] J.M. Rosário, C. Oliveira, C.E.A. Sa,.“Proposal Methodology for the Modeling and Control of Manipulators”, *International Journal of the Brazilian Society of Mechanical Engineering,* Vol. XXIV – N 3, July 2002.