Programming Environment and a new positioning control for the PUMA 200 robot.

FERNANDO REYES CORTES, ALEJANDRO GUZMÁN RAMÍREZ, JAIME CID MONJARAZ
Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de Puebla.
Apartado Postal 542, Puebla 72001, México
Tel / Fax: +52 22 29 55 00 Ext. 7405 y 7857

ABSTRACT:- The development of the current work is based mainly on improving the hardware developed in previous works [6], by adding the necessary tools to support an open programming environment, and the elements of control to facilitate the experimenting of new algorithms of control. In order to accomplish this task it was necessary to perform an analysis of stability over a new controller for the PUMA 200 robot (Programmable Universal Manipulator for Assembly). The design of this controller is based on the energy molding technique.

1 Introduction

Every time that the word Robot is mentioned, most of the people think of machines with human or animal shapes, related mostly with Science Fiction movies. Even when that idea is not completely wrong, not only the robots on the movies exist; there are different fields of the human society where they are used, such as education, entertaining, manufacturing, the army, etc.

Within the articulated manipulators, also known as anthropomorphic ones, is situated the PUMA. This manipulator possesses six degrees of motion.

The PUMA system was designed to adapt itself to numerous applications. It is constituted by the following basic units: software, controller, peripherals, teach pendant and robot arm.

The software that controls the robot arm is placed on the memory of a computer placed on the controller; where the operative controls of the system are located.

There are many kinds of robots and almost one robot language for each trademark. Mainly two kinds predominate at this moment: the ones with their own semantics and syntax, and the ones compound of command libraries that take advantage of an existing programming language.

Figure 1. PUMA 200 System.

The most used of them are the last ones, as they save some of the work and it is not necessary to learn a specific programming language. Besides making easier the robot programming, to use libraries allows the utilization of several tools and algorithms of control that improve the efficiency and versatility of the robot arm.

The PUMA robot arm uses its own operative system. This system doesn't allow to program new algorithms of control, or the creation of applications that use visual messages for the user.

Thus, the advantages of having the required hardware to communicate the PUMA robot arm with a computer that uses a programming language like for example *C* or *Visual C++* are clear: in this way the manipulator keeps an open architecture in its programming environment.

The benefits of possessing an open system are not only the usage of powerful computing tools on the development of applications, but also the easy incorporation of these mechanisms in courses of

Robotics or Digital Control. Since these courses have as purpose the development of algorithms of control, the incorporation of these algorithms to the robot arm can be done almost directly.

2 Description

The development of the present work is based on establishing the required environment for the operation of the PUMA 200 robot, using a Personal Computer, and the required tools for an open programming environment; in addition to the elements of control to facilitate the experimentation of new algorithms of control.

A Personal Computer can transfer data from a Hard Drive Disk to the CPU, from the CPU to the memory, or from the memory to the Video Card. To have separated circuits for every pair of devices is considered as a waste on a PC system. The solution to this problem is frequently the usage of a *bus*. In a general classification, the Personal Computers use two kinds of interfaces to communicate with the outer world, serial and parallel ports. The difference between both is significant. For a digital interface of *n*, a parallel device uses *n* wires to transfer all the data in one cycle; on the other hand, the serial device uses one wire to transfer the same data in *n* cycles. Therefore, the parallel interface transfers data *n* times faster than the serial one.

The bus of a computer is formed by a number of tracks of conducting material. The cards – in communication with one another -, are connected to this tracks, where they make usage of some of the signals of the information flow. Only one of the cards is allowed to send or receive data from the bus at the same time.

The interface between the PUMA 200 robot and the PC uses three cards. They are connected to the ISA slots in the computer. The following diagram describes all the connections between the ISA Cards and the PUMA System components.

After conducting an investigation on programming environments, it was achieved the implementation of a totally open environment in programming. This environment integrates visual tools for the evaluation of the efficiency of the existing positioning controllers, and it allows the implementation of new controllers in a very simple mode thanks to the tools provided by MATLAB.

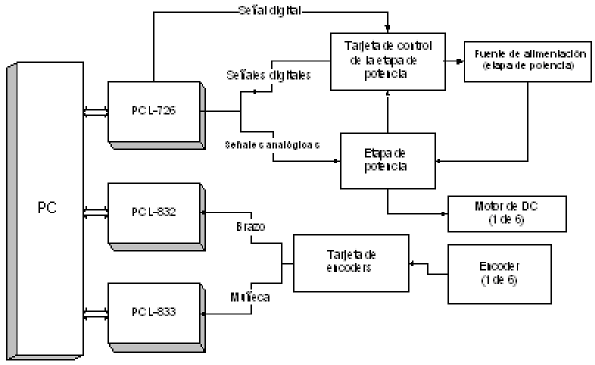

Figure 2. Connections PC - PUMA 200

3 Programming Environments

One of the big impediments in the programming of algorithms of control is the lack of effective communication between the user and the robot.

A general approach to solve this problem is the development of programming environments pointed to the experimental evaluation of the algorithms of control.

A significant number of researchers have focused their works in the development of programming environments, one of the disciplines in Robotics, due to the importance that it has over the easy operation in the implementation and experimental validation of the algorithms of control of the manipulator robot.

The programming environment also allows the user to take control of the experimental configuration, algorithm execution, display and register of data and it is also the way that the robot is protected against any contingency.

4 Dinamic Model for Manipulator Robots

The dynamics of a Manipulator Robot can be described as [1]:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F(\tau,\dot{q}) \qquad (1)$$

where q is a vector of $n\,x\,1$ and is associated with the position of the articulations, the vector $\dot{q}$ of $n\,x\,1$ describes the speed of the articulations, $M(q)$ is a symmetric matrix, defined as positive of $n\,x\,n$, which is associated with the inertia of the manipulator. $C(q,\dot{q})$ is the matrix of Coriolis and has the dimensions of $n \, x \, n$. The gravitational torque is represented by $g(q)$ and $F(\tau,\dot{q})$ represents the friction. Both are vectors of $n \, x \, 1$. τ is a vector of the applied torque.

An important property of the dynamic model is the following:

$$\dot{q}^T\left[\frac{1}{2}\dot{M}(q)-C(q,\dot{q})\right]\dot{q}=0 \quad \forall\, q,\dot{q}\in R^n \qquad (2)$$

Regulator

Let us consider that the regulator for the manipulator robot can be described through the following law of control:

$$\tau=\nabla U_a(Kp,\tilde{q})-F_v(Kv,\dot{q})+g(q)+F(\tau,\dot{q}) \qquad (3)$$

where $\tilde{q}=q_d-q$ denotes the error of position, q_d is the vector of the requested positions for the articulations, and where $U_a(Kp,\tilde{q})$ is the artificial potential energy and it must accomplish the following conditions:

$$\nabla U_a(Kp,\tilde{q})=\frac{\partial}{\partial\tilde{q}}U_a(Kp,\tilde{q}) \qquad (4)$$

$$U_a(Kp,\tilde{q})|_{\tilde{q}=0}=0$$

it is proponed the function of energy,

$$U_a(Kp,\tilde{q})=\frac{1}{2m}\tilde{q}^{m^T}Kp\tilde{q}^m+\frac{1}{2m}\begin{bmatrix}e^{-\alpha\tilde{q}_1^m}\\ \cdot\\ \cdot\\ \cdot\\ e^{-\alpha\tilde{q}_n^m}\end{bmatrix}^T Kp\begin{bmatrix}e^{-\alpha\tilde{q}_1^m}\\ \cdot\\ \cdot\\ \cdot\\ e^{-\alpha\tilde{q}_n^m}\end{bmatrix}-\sum_{i=1}^{n}Kp_i$$

which accomplishes the previous conditions.

$$\nabla U_a(Kp,\tilde{q})=Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1} \qquad (6)$$

$$\text{where } \tilde{q}^{2m-1}=\begin{bmatrix}\tilde{q}_1^{2m-1}\\ \cdot\\ \cdot\\ \cdot\\ \tilde{q}_n^{2m-1}\end{bmatrix} \qquad (7)$$

The tuning of the parameters is in function of the masses, center of mass, etc. of the manipulator robot.

$F_v(Kv,\dot{q})$ is a function that helps to set the system more stable.

For this regulator it is proponed an $F_v(Kv,\dot{q})$ function,

$$F_v(Kv,\dot{q})=Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1} \qquad (8)$$

finally, it is reached the following expression for the regulator,

$$\tau=Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}-Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}+g(q)+F(\tau,\dot{q})$$

Kp, Kv are diagonal matrixes defined as positives,

$$\dot{q}^{2m-1}=\begin{bmatrix}\dot{q}_1^{2m-1}\\ \cdot\\ \cdot\\ \cdot\\ \dot{q}_n^{2m-1}\end{bmatrix} \qquad (10)$$

the matrixes $\alpha e^{-\alpha\tilde{q}^{2m}}$ and $Be^{-B\dot{q}^{2m}}$ are diagonal matrixes of $n \, x \, n$, where $\alpha\in(0,1)$ and $B\in(0,1)$

The next step is to associate the regulator with the dynamics of the manipulator robot, putting both expressions of τ as equals.

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)+F(\tau,\dot{q})=Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}$$
$$-Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}+g(q)+F(\tau,\dot{q})$$

We obtained the representation of the states space, which describes the dynamics of the robot in a closed control system.

$$\frac{d}{dt}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}=\begin{bmatrix}-\dot{q}\\ M(q)^{-1}\left[Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}-Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}-C(q,\dot{q})\dot{q}\right]\end{bmatrix}$$

Stability analysis

Theorem of Liapunov [3]. Be $x=0$ a point of equilibrium of the system $\dot{x}=f(x)$ and $D\subset R^n$ a domain that contains $x=0$.

Be $V: D\to R$, a continuous differential function given that,

$$V(0)=0 \text{ y } V(x)>0 \text{ en } D-\{0\}$$
$$\dot{V}(x)\le 0 \text{ en } D$$

then $x=0$ is stable all the more if $\dot{V}(x)<0$ in $D-\{0\}$ then $x=0$ is stable asymptotically.

Equilibrium point

We searched for the value or values in which $\frac{d}{dt}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}=0$, and it is obvious that for this is necessary that $\dot{q}=0$ and that:

$$M(q)^{-1}\left[Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}-Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}-C(q,\dot{q})\dot{q}\right]=0$$

if $\dot{q}=0$, the previous equation reduces obtaining the following expression:

$$M(q)^{-1}\left[Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}\right]=0 \qquad (14)$$

because the matrix $M(q)$ is defined as positive and $\alpha\in(0,1)$:

$$Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}=0 \quad\Rightarrow\quad \tilde{q}=0$$

therefore the equilibrium point exists, it is only one and it is located in the origin $\begin{pmatrix}\dot{q}=0\\ \tilde{q}=0\end{pmatrix}$.

Now we propose the candidate function of Liapunov.

$$V(\dot{q},\tilde{q})=\frac{1}{2}\dot{q}^{T}M(q)\dot{q}+\frac{1}{2m}Kp\left[\tilde{q}^{2m}+e^{-\alpha\tilde{q}^{2m}}-I\right]$$

which in, $V(0,0)=0$

$$\dot{V}(\dot{q},\tilde{q})=\dot{q}^{T}M(q)\ddot{q}+\frac{1}{2}\dot{q}^{T}\dot{M}(q)\dot{q}+\tilde{q}^{2m-1}Kp\dot{\tilde{q}}-\alpha e^{-\alpha\tilde{q}^{2mT}}Kp\tilde{q}^{2m-1}\dot{\tilde{q}}$$

We include the dynamics of the robot in the closed control system,

$$\dot{V}(\dot{q},\tilde{q})=\dot{q}^{T}M(q)\left[M(q)^{-1}\left[Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}-Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}-C(q,\dot{q})\dot{q}\right]\right]$$
$$+\frac{1}{2}\dot{q}^{T}\dot{M}(q)\dot{q}+\tilde{q}^{2m-1}Kp\dot{\tilde{q}}-\alpha e^{-\alpha\tilde{q}^{2mT}}Kp\tilde{q}^{2m-1}\dot{\tilde{q}}$$

develop the previous equation and reduce the terms applying the characteristics of the dynamic model:

$$\dot{V}(\dot{q},\tilde{q})=\dot{q}^{T}Kp\left[I-\alpha e^{-\alpha\tilde{q}^{2m}}\right]\tilde{q}^{2m-1}-\dot{q}^{T}Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}-\tilde{q}^{2m-1}Kp\dot{q}$$
$$-\alpha e^{-\alpha\tilde{q}^{2mT}}Kp\tilde{q}^{2m-1}\dot{\tilde{q}}$$

Finally, we obtain:

$$\dot{V}(\dot{q},\tilde{q})=-\dot{q}^{T}Kv\left[I-Be^{-B\dot{q}^{2m}}\right]\dot{q}^{2m-1}<0$$

therefore the system is stable.

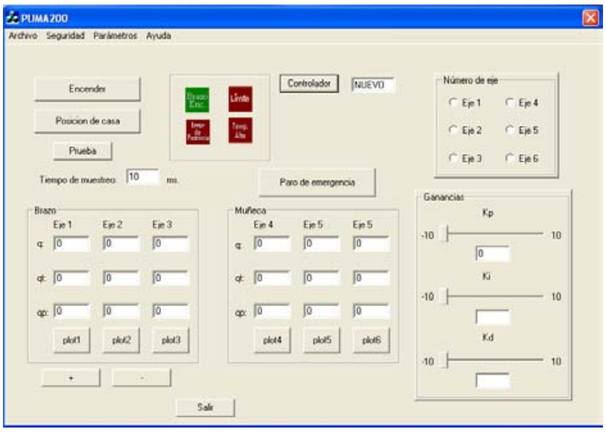

Figure 3. Programming Environment

5 CONCLUSIONS

An open architecture programming environment must contain the following as the main characteristics:

- Reliability.- it must include the necessary programming to avoid the robot damaging itself or damaging the user.
- Positioning Control.- the management of the variables ($q_x, \dot{q}_x, \tau_x, \tilde{q}_x, x, y, z$, etc.) used to control the positioning is essential in the design of programming environments.
- Speed limits and torque.- these values shall be variables; however, the environment should measure them based on the mechanical and electronic limitations of the system.
- Visualization and registration of the variables.- the environment must display the values of the variables selected by the user. In addition to allow this user to store those variables.

These were some of the main characteristics that will be added to our programming environment.

It is important to remark that it has been possible to demonstrate stability; not only for one controller but for a whole family of them, because the value of ***m*** can be a whole number. Moreover, the controller is built in the form $Kp(\tilde{q})\cdot\tilde{q}^{2m-1}$, this expresses that the proportional gain is in function of the error, which means that it is auto-adjustable.

Finally the experiment results are shown in figures 4 and 5. The desired joint positions as well as the initial positions and velocities were chose in this experiment test as:

$q_{d1} = 45[\text{deg}]$, $q_{d2} = 15[\text{deg}]$, $q_{d3} = 20[\text{deg}]$,

$q_{d5} = 30[\text{deg}]$, $q_{d6} = 30[\text{deg}]$

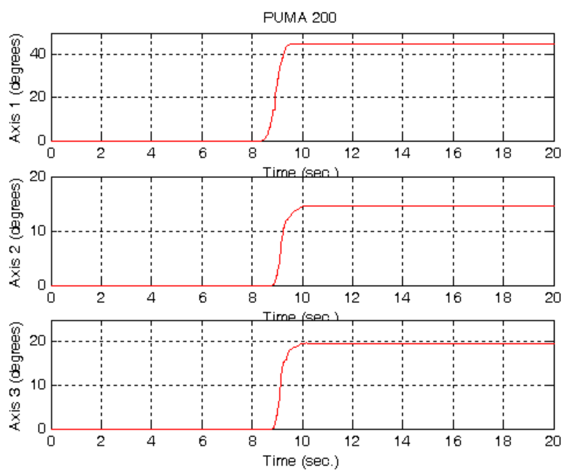

Figure 4. Experimental results

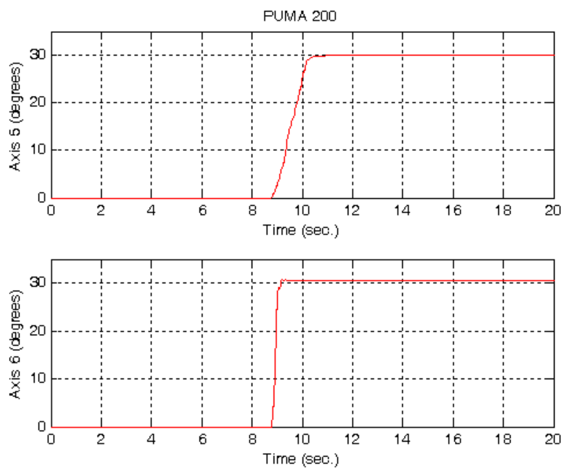

Figure 5. Experimental results

6 REFERENCES

[1] Spong, M and M.Vidyasagar, *Robot Dynamics and Control*, Wiley, New York, 1989.

[2] Koditscheck, D., ”Natural motion for robot arms”, Proc. IEEE Conf. On Decision and Control, Las Vegas, NV, 1984, pp 733-735.

[3] Hassan K. Khalil, *Nonlinear Systems Third Edition*, Prentice Hall, New Jersey, 2002

[4] Torres F., Pomares J., Gil P., Santiago T. & Aracil R., “Robots y Sistemas Sensoriales”. Prentice all.2002

[5] Reyes F. & Aguilar M. A. *Diseño de un Ambiente de Programación de Arquitectura Abierta para Robots Manipuladores*. FCE-BUAP, 2002

[6] J. Cid & Guzmán A. “Interfaz del Brazo Robot PUMA 200 con la PC”. FCE-BUAP,2002