Robotica (2005) volume 23, pp. 515–520. © 2005 Cambridge University Press
doi:10.1017/S0263574704001031 Printed in the United Kingdom

Learning control of robot manipulators by interactive simulation

Rafael Kelly*, Sebastian Dormido†, Carmen Monroy* and Elizabeth Díaz*

(Received in Final Form: September 6, 2004)

SUMMARY
Control systems of robot manipulators offer many challenges in education where the students must learn robot dynamics and control structures, and understand relations between the control parameters and the systems performance. Interactive simulation is aimed at improving the understanding of and intuition for the abstract parts of the control of robot courses. This paper presents an application of interactive simulation to teach control systems of robots. The application considers a nonlinear robot arm and two control modules: position control and motion control. Students can directly manipulate graphical representation of the systems such as a choice among seven control structures, controller gains, and desired trajectories, and obtain instant feedback on the effects. These features make the interactive learning tool stimulating and of high pedagogical value.

KEYWORDS: Robotics; Education; Interactive simulation; Robot control; Control systems.

1. INTRODUCTION

Robotics is a field of modern technology beyond the traditional engineering boundaries. Understanding the complexity of robots and their applications needs a knowledge of complementary engineering disciplines. This understanding can be made easy by using modern learning tools based on software for numerical computations and simulations.

Education in control of robots has become an important and integral part of engineering curricula in robotics related engineering such as electrical, mechanical and computer disciplines. On the other hand, industry continues to introduce more sophisticated control systems for robotic machines whose applications range from manufacturing factories to medical assistance.

In recent years, the advances in software and hardware has allowed to design tools with much better man–machine interaction, with intuitive graphical user interfaces and, more important perhaps, a high degree of interactivity. These tools are particularly useful in the study of dynamic systems in general, and specially in automatic control. Although many interactive tools have been developed for education in automatic control (see references 1–3) for recent surveys, the application of interactive tools for learning robot control is practically inexistent.

The objective of this paper is to present an interactive tool for education of robot control based on the CAD software SysQuake developed at the Federal Polytechnic of Lausanne.[4] This tool is focused on objects to explore different views of the systems, manipulate views directly using the mouse, and immediately see the consequences on the system behavior. More specifically, the presented interactive tool is composed of two modules: the position control module and the motion control module. The former deals with regulators to guide robot manipulators at desired joint configurations, and the latter concerns tracking controllers for following desired timed joint trajectories. Both modules consider a common robot arm model and a collection of seven control schemes. The underlying structure of the control schemes is based on a Proportional Derivative (PD) feedback part, plus a remaining feedforward or feedback term characteristic of each control strategy. We have found that the tool is a valuable complement to courses on control of robots and control of mechanisms because the high degree of interactivity makes the tool stimulating and quickly captures the interest of the user.

The remainder of the paper is organized as follows: Section 2 presents the model of robot manipulators. Section 3 describes the position control objective and summarizes four representative controllers. Section 4 is devoted to motion control and introduces three control algorithms. The functionality of the interactive tool modules is described in Sections 5 and 6. Finally, a brief summary is given in Section 7.

2. ROBOT MODEL

From a mechanical point of view, the robot manipulators are mechanisms formed by links jointed together through prismatic or revolute joints. The motion of each joint is carried out by an actuator which delivers a desired force or torque at the joint. The relationship between the time evolution of the joint displacements and the applied joint forces or torques, defines the robot dynamic model.

Under reasonable assumptions, the dynamic model of n joints robot manipulators has the general structure:[5]

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

* División de Física Aplicada, CICESE, Apdo. Postal 2615, Adm. 1, Ensenada, B.C., 22800 (MEXICO).
† Dept.de Informática y Automática, E. T. S. Ing. Informática, U.N.E.D Juan del Rosal 16, 28040 Madrid (SPAIN).
Corresponding author: E-mail: rkelly@cicese.mx

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press

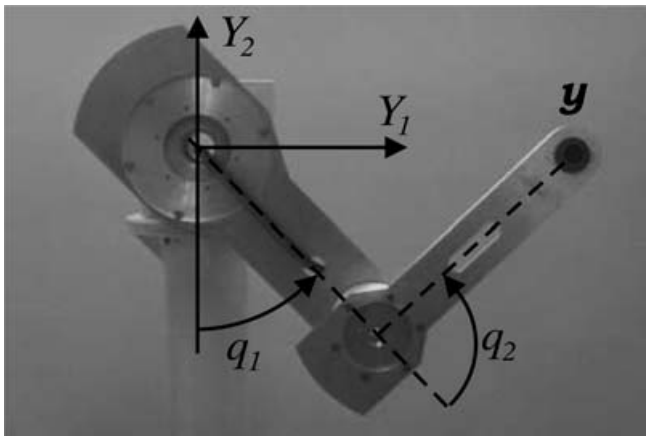


Fig. 1. Robot arm.

where $\boldsymbol{q}$ is the $n \times 1$ vector of joint displacements, $\dot{\boldsymbol{q}}$ is the $n \times 1$ vector of joint velocities, $\boldsymbol{\tau}$ is the $n \times 1$ vector of applied torque inputs, $M(\boldsymbol{q})$ is the $n \times n$ symmetric positive definite manipulator inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the $n \times n$ matrix of centrifugal and Coriolis torques, and $\boldsymbol{g}(\boldsymbol{q})$ is the $n \times 1$ vector of gravitational torques due to gravity.

In order to illustrate the basic concepts we have looked for a robot arm as simple as possible but keeping the main nonlinear features of general robot manipulators. We found that a robotic arm with two revolute joints ($n = 2$) moving in the vertical plane (see Figure 1) captures these ingredients. We borrow real numerical values of robot arm having this geometry from references [6] and [7]. The resulting form of the robot model elements – $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ has been chosen using Christoffel symbols –, is given by

$$M(\boldsymbol{q}) = \begin{bmatrix} 2.351 + 0.168\cos(q_2) & 0.102 + 0.084\cos(q_2) \\ 0.102 + 0.084\cos(q_2) & 0.102 \end{bmatrix},$$

$$C(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -0.084\sin(q_2)\dot{q}_2 & -0.084\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ 0.084\sin(q_2)\dot{q}_1 & 0 \end{bmatrix},$$

$$\boldsymbol{g}(\boldsymbol{q}) = 9.81 \begin{bmatrix} 3.921\sin(q_1) + 0.186\sin(q_1 + q_2) \\ 0.186\sin(q_1 + q_2) \end{bmatrix}.$$

It can be easily verified that the matrix $\frac{1}{2}\dot{M}(\boldsymbol{q}) - C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is skew symmetric. This fact is useful for stability analysis of robot control systems.

3. POSITION CONTROL

Position control of robot manipulators in joint space is perhaps the simplest control objective in the control of robots. In the automatic control terminology, it corresponds to set–point control, also referred as regulation. More specifically, given a constant desired joint position $\boldsymbol{q}_d \in \mathbb{R}^n$, the control objective revolves around the computation of the joint applied torques $\boldsymbol{\tau}$ such that the robot joint displacements $\boldsymbol{q}$ reach asymptotically the desired ones, i.e.

$$\lim_{t \to \infty} \boldsymbol{q}(t) = \boldsymbol{q}_d.$$

Equivalently, by defining the joint position error $\tilde{\boldsymbol{q}} \in \mathbb{R}^n$ as $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$, then the position control objective is to ensure that $\lim_{t \to \infty} \tilde{\boldsymbol{q}}(t) = \boldsymbol{0}$.

A number of control structure are able to solve the position control objective of robot manipulators. The basic and more intuitive control schemes are those having a linear Proportional and Derivative (PD) feedback part. This class of controllers has two control gains: the proportional gain K_p and the derivative gain, K_v; both are $n \times n$ symmetric positive definite matrices chosen by the user. According to the control structure and the gains selection, the closed–loop systems offer a wide spectrum of behaviors such as steady state position errors, fast transient, overshooting, etc. Since there is no a direct analytical relation to determine the quantitative effects of the gain changes, thus numerical methods must be applied by a trail and error tedious procedure to obtain a desired closed–loop system performance. This is the point where interactive simulation presents its useful potential to realize in a graphical and interactive mode such a relationship.

In the rest of this section we summarize four position controllers for robot manipulators based on PD structure.

3.1. PD control

The PD control has demonstrated its effectiveness in industrial applications when applied to second–order linear actuators, such as direct current motors.[5,8] This fact has motivated its application to solve the position control of robot manipulators. The control law structure of the PD control can be written as

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} - K_v\dot{\boldsymbol{q}} \tag{2}$$

where K_p and K_v are $n \times n$ symmetric positive definite matrices referred as the proportional and derivative gains. Surprisingly, this lineal controller when applying to robots whose nonlinear dynamics is governed by (1) but without gravitational torque term, i.e. $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$, leads to a globally asymptotically stable system. This is a by–product of the analysis reported in the key paper of Takegaki and Arimoto.[9]

3.2. PD control with gravity compensation

Although the PD control (2) is effective for controlling particular robots without gravitational torque vector, in the real world it has an important effect on the dynamics of most robots and it cannot be neglected. Notwithstanding, a clever and simple extension of the PD control is sufficient to address the position control of robots regardless of the gravitational torque vector. This results in a PD control with gravity compensation whose control law looks like

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} - K_v\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{3}$$

where K_p and K_v are $n \times n$ symmetric positive definite proportional and derivative gains matrices. This is a model–based control scheme because of the presence of the compensation term $\boldsymbol{g}(\boldsymbol{q})$ in the control law. This control technique was the first one reported in the robot control literature ensuring global asymptotic stability for general n degrees–of–freedom mechanical manipulators.[9]

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press

3.3. PD control with desired gravity compensation
The PD control with gravity compensation (3) can be simplified by substituting the gravity compensation part $\boldsymbol{g}(\boldsymbol{q})$ – which must be computed in real-time as the robot moves – by a suitable constant vector selected as the gravitational torque vector evaluated at the desired joint position $\boldsymbol{g}(\boldsymbol{q}_d)$. This leads to the PD control with desired gravity compensation whose control law reads

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} - K_v\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}_d) \tag{4}$$

where K_p and K_v are still $n \times n$ symmetric positive definite proportional and derivative gains matrices. However, the proportional gain matrix K_p must be carefully chosen. In order to have only one equilibrium and ensure global asymptotic stability, its smallest eigenvalue must be sufficiently large as described in the original proof provided by Takegaki and Arimoto.[9] A complete study of the control scheme and extensions for the case of parametric uncertainties and for elastic joint robots are presented in detail in reference [10].

3.4. PID control
Although the PD control with gravity compensation (3) or with desired gravity compensation (4) are able to solve the position control of robots, they need an exact knowledge of the gravitational torque vector $\boldsymbol{g}(\boldsymbol{q})$. The Proportional–Integral–Derivative (PID) control structure does not have such a limitation. The PID control law is given by

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} + K_i z - K_v\dot{\boldsymbol{q}} \tag{5}$$

where the new state variable z is

$$\dot{z} = \tilde{\boldsymbol{q}}. \tag{6}$$

The $n \times n$ matrices K_p and K_v are still the proportional and derivative gains, and K_i is the $n \times n$ symmetric positive definite matrix called the integral gain. Observe that structure of the PID control law (5)–(6) does not requires information about the robot dynamics. However, in order to ensure asymptotic stability the gain matrices must be carefully selected by taking into account the dynamics. A number of tuning procedures can be considered in references [11, 12, 13, 14].

4. MOTION CONTROL

The primary goal of motion control in joint space is to make the robot joint positions $\boldsymbol{q}$ track a given time–varying desired joint position $\boldsymbol{q}_d$. Rigorously, the motion control objective in joint space is achieved provided that

$$\lim_{t\to\infty} \tilde{\boldsymbol{q}}(t) = \boldsymbol{0}$$

where $\tilde{\boldsymbol{q}}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t)$ denotes the joint position error.

In the recent years, the PD–based control algorithms improved the control systems of robots. These control schemes applied to motion control require a knowledge of the robot dynamics, so they are called model–based control algorithms.

This section deals with three representative model–based controllers: PD+ control, PD control with feedforward compensation, and the non–adaptive Slotine–Li's control.

We use the following notation: $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$ stand for the desired joint position, velocity, and acceleration trajectories, respectively, which are chosen to be bounded functions. The joint position and velocity errors are denoted by $\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$, and $\dot{\tilde{\boldsymbol{q}}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$, respectively.

4.1. PD+ control
The PD+ control law can be written as

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} + K_v\dot{\tilde{\boldsymbol{q}}} + M(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}). \tag{7}$$

The first two terms correspond to the PD controller, the remaining depend on the robot dynamics. This control scheme was first introduced by Koditschek[15] and formally analyzed in references [16] and [17]. It has been proved that under exact robot dynamics, and symmetric positive definite matrices K_p and K_v, the PD+ control yields a globally asymptotically stable closed–loop system, thus asymptotic joint trajectory tracking.

4.2. PD control with feedforward compensation
This control consists of a linear PD controller plus a feedforward of the nominal dynamics computed along the desired trajectory $\boldsymbol{q}_d$. The control law is given by:[18]

$$\boldsymbol{\tau} = K_p\tilde{\boldsymbol{q}} + K_v\dot{\tilde{\boldsymbol{q}}} + M(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + C(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d). \tag{8}$$

The advantage of this controller is that once a desired trajectory ($\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{q}}_d$) for a given task has been specified, then the feedforward terms relying on the robot dynamics $M(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + C(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d)$ can be computed off–line thus reducing the computational burden.

The stability analysis of controller (8) can be found in references [19] and [20]. In summary, under exact robot dynamics, there exist symmetric positive definite matrices K_p and K_v, such that the closed–loop system is globally asymptotically stable, hence the position error $\tilde{\boldsymbol{q}}$ vanishes asymptotically.

4.3. Slotine–Li control (non–adaptive version)
First proposed as an adaptive control scheme by Slotine and Li,[21] this controller has become very popular, and is commonly named after its creators. The nonadaptive version of the control law can be written as

$$\begin{aligned}\boldsymbol{\tau} = {} & K_p\tilde{\boldsymbol{q}} + K_v\dot{\tilde{\boldsymbol{q}}} + M(\boldsymbol{q})[\ddot{\boldsymbol{q}}_d + \Lambda\dot{\tilde{\boldsymbol{q}}}] \\ & + C(\boldsymbol{q}, \dot{\boldsymbol{q}})[\dot{\boldsymbol{q}}_d + \Lambda\tilde{\boldsymbol{q}}] + \boldsymbol{g}(\boldsymbol{q})\end{aligned} \tag{9}$$

where Λ is defined as $\Lambda = K_v^{-1}K_p$. The Slotine–Li control scheme may be seen as an extension of the PD+ control (7) where Λ has been incorporated. The proof of global asymptotic stability can be extracted from references [22 and 23] under the requirement that the matrix gains be symmetric positive definite.

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press

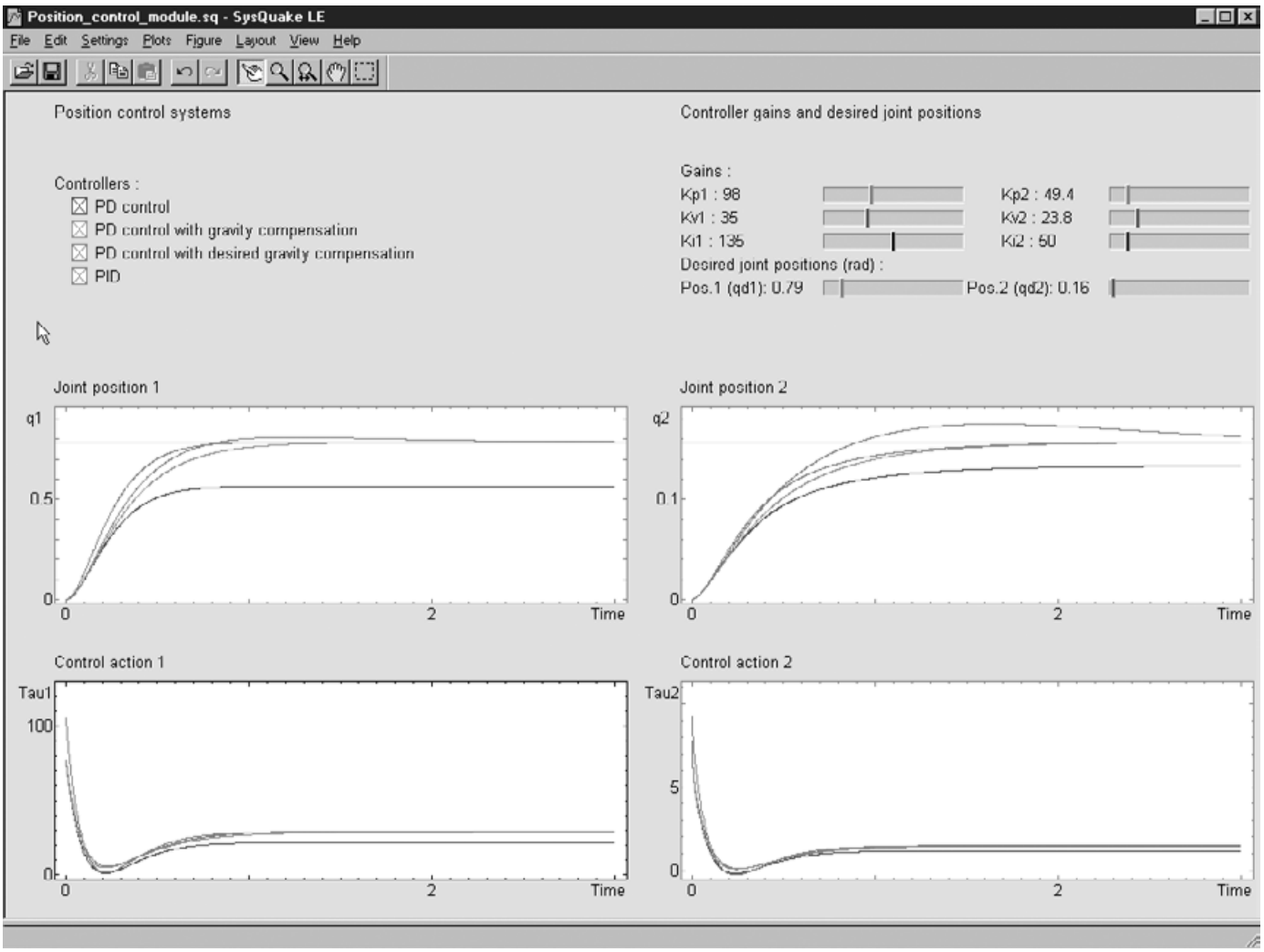


Fig. 2. Graphical user interface of the position control module.

5. POSITION CONTROL MODULE

The position control module for interactive simulation incorporates the four position control schemes described previously, namely: PD control, PD control with gravity compensation, PD control with desired gravity compensation, and PID control. The main graphical user interface is depicted in Figure 2. It is a dynamic picture in the sense that the interface is built up in an interactive manner where the user can change a parameter and immediately see its influence without typing a single command.

The interface shown in Figure 2 is divided in two main subscreens: The upper one contains on the left part boxes for selection of controllers and on the right part the parameters $Kp1$, $Kp2$, $Kv1$, and $Kv2$ corresponding to the elements of the diagonal matrix gains K_p and K_v, respectively. Also, the desired join positions $qd1$ and $qd2$ are included in this part of the interface. The lower subscreen of the interface shows the system responses, particularly the joint positions and the control actions.

The user can select – by clicking with the mouse – one or more of the controllers, change the proportional or derivative gains and set new desired joint positions by dragging the corresponding sliders with the mouse, and visualize immediately the system responses.

This new interactive computer learning experience in control of robots provides the user with a comparative tool to assess the advantages and limitations among different control schemes. The lessons to learn from this interactive simulation module are:

- The PD control is unable to achieve the position control objective because it leads to a steady state position error. Notwithstanding, this positioning error can be reduced by increasing the proportional gains.
- The PD control with gravity compensation leads to a vanishing position error. It presents a faster response as the proportional gain increases.
- The PD control with desired gravity compensation may produce a steady state position error whether the proportional gain is small enough. However, for sufficiently large proportional gains, the positioning error vanishes asymptotically.
- The PID control may achieve, in practice, the position control objective. The control gains must be carefully chosen. The integral gain should be maintained small at the price of a slow system response.
- For all controllers, oscillatory behavior may appear when the derivative gain is decreased.

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press

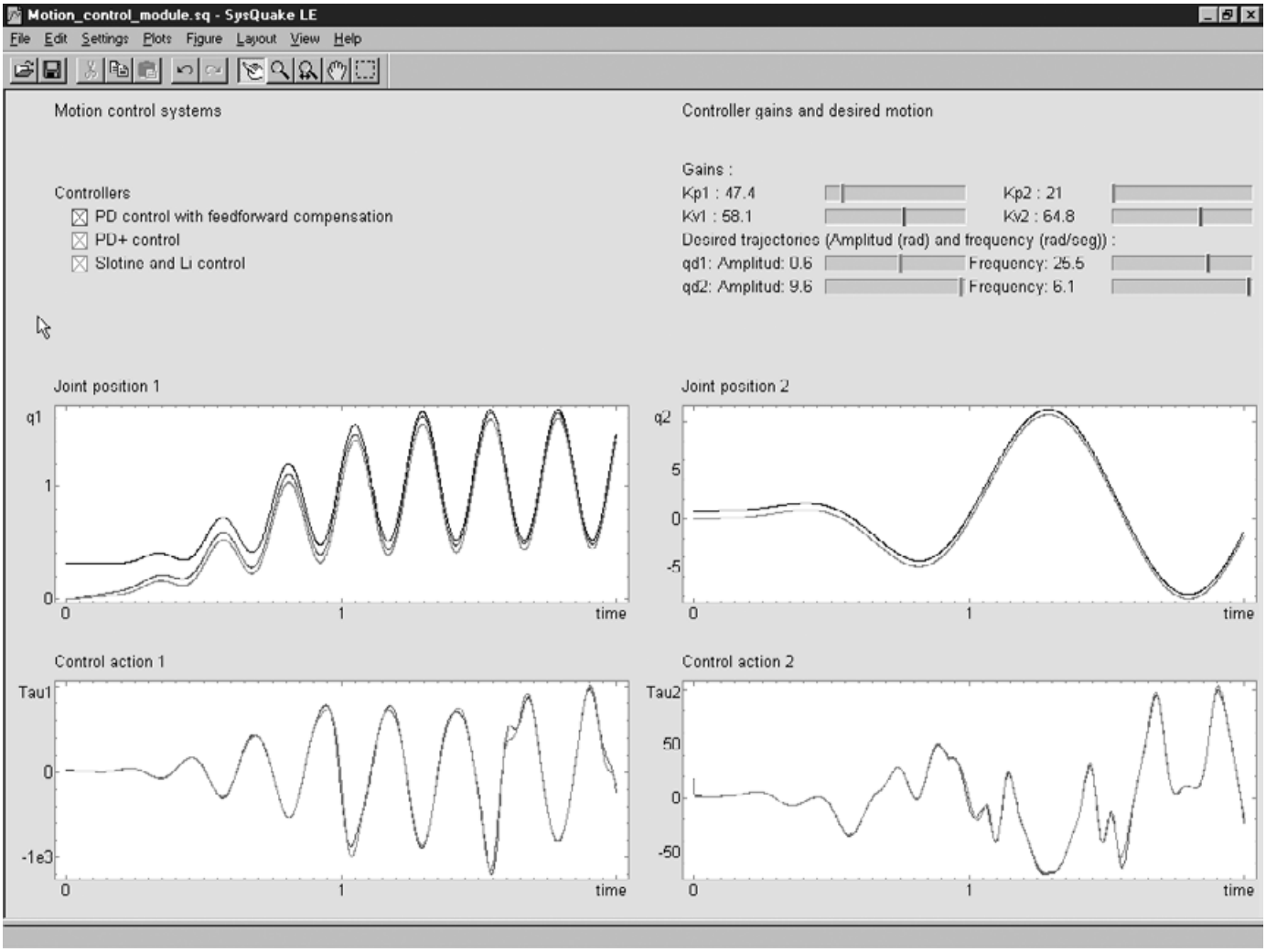


Fig. 3. Graphical user interface of the motion control module.

6. MOTION CONTROL MODULE

In contrast with the position control module where the desired joint positions are constant (step set–point), the motion control module addresses the situation of time–varying desired joint position trajectories $\boldsymbol{q}_d(t)$. The following form for such trajectory has been selected which has an asymptotic periodic behavior

$$\boldsymbol{q}_d(t) = \begin{bmatrix} \frac{\pi}{10} \\ \frac{\pi}{4} \end{bmatrix} + \begin{bmatrix} \frac{\pi}{4}\left[1 - e^{-2.0\,t^3}\right] + a_1\left[1 - e^{-2.0\,t^3}\right]\sin(\omega_1 t) \\ \frac{\pi}{3}\left[1 - e^{-1.8\,t^3}\right] + a_2\left[1 - e^{-1.8\,t^3}\right]\sin(\omega_2 t) \end{bmatrix} \quad \text{[rad]} \tag{10}$$

where a_1, a_2 are the desired amplitudes, and ω_1, ω_2 are the desired frequencies of the steady state desired joint positions. These parameters can be changed interactively by the user through the graphical interface.

The structure of the desired trajectory (10) was inspired by the desired ones used by other authors for experimental evaluation of control algorithms.[24,25] In our application, the second term of (10) was chosen in such a way as to exploit the arm in its fastest motion but without invading the actuators saturating zone, and the first one was chosen to add a step reference to demand an initial large torque.

The motion control module disposes of the graphical user interface shown in Figure 3. It is split into an upper sunscreen devoted to selection of controllers and manipulation of parameters, and a lower subscreen where the time responses of the control system are depicted.

In the upper subscreen a number of choices and parameters can be on–line selected by means of selection boxes and sliders. The selection boxes refers to the three model–based control techniques described in Section 4. It is emphasized that the one or more controller can be selected on–line and they are tuned simultaneously by the same proportional and derivative gains as well as they assume the same desired joint position trajectory. The parameter allowed to be changed by moving the sliders are the proportional gains $Kp1$ and $Kp2$, the derivative gains $Kv1$ and $Kv2$, and the amplitudes and frequencies of the desired position trajectory.

The lower subscreen contains four windows to show selected system responses. By default, two windows depict the desired position trajectories in black color and the corresponding robot joint position in distinctive colors. The

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press

remaining two windows trace the control actions utilizing the same colors than those of the joint position evolutions. All of these windows can be changed on–line to show a number of variables such as position errors, joint velocities.

Thanks to the interactive simulation, the motion control module permits the user to understand not only the effect of the manipulations on the controller gains, but its direction and amplitude becomes apparent. Therefore, the user learns quickly which parameter to use and how to push the design in the direction to better satisfaction of the specifications. In a natural manner, the user can extract the following guidelines from this learning process:

- The proportional gain plays an important role to achieve a rapid transient. As large it is, the position error reduces faster. However, this also increases the control actions, which in practice are limited.
- An other way to increase to response of the control system is by decreasing the derivative gains. Notwithstanding, very low gains yield undesirable overshooting and damped oscillations.
- The control actions increase when the amplitude and frequency of the desired trajectory increase. Thus, the proportional gain should be reduced to maintain acceptable control actions.

7. CONCLUSIONS

A tool for interactive simulation of control systems for robot manipulators has been described in this paper. The high degree of interactive of SysQuake has been found to be a key issue in the tool design. The tool is divided in two modules according to the control objective: position control module and motion control module. The former includes four control systems having a common underlying PD control structure; the latter incorporates three model–based control systems also with a PD control part.

The tool is based on objects for easy direct graphical manipulation. Control schemes, their parameters, and references can be changed on–line, thus during manipulations, objects are updated instantaneously and the relations between them are maintained all the time. The effect of changing the proportional or derivative gains of the controllers, compensating for gravity and allowings for larger reference amplitudes, is visualized immediately. This fact has a strong pedagogical potential to be used as a natural complements to traditional training and education on the control of robots to quickly gain an insight and motivation.

Acknowledgements

The authors would like to thank the partial support of CYTED, Mexican CONACyT, and Spanish CICYT under grant DPI2001-1012.

References

1. B. Wittenmark, H. Haglund and M. Johansson, "Dynamic pictures and interactive simulation," *IEEE Control Systems* **18**(3), 26–32 (June, 1998).
2. M. Johansson, M. Gäfvert and K. J. Åström, "Interactive tools for education in automatic control," *IEEE Control Systems* **18**(3), 33–40 (June, 1998).
3. S. Dormido, "Control learning: present and future," *Proc. of the 15th IFAC World Congress*, Barcelona, Spain (July, 2002) pp. 81–103.
4. Y. Piguet, U. Holmberg and R. Longchamp, "Instantaneous performance display for graphical control design methods," *Proc. 14th IFAC World Congress*, Beijing, P. R. China (July, 1999). **Vol L**, pp. 403–408.
5. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control* (John Wiley and Sons, New York, 1989).
6. F. Reyes and R. Kelly, "Experimental evaluation of identification schemes on a direct–drive robot," *Robotica* **15**, Part 4, 563–571 (1997).
7. F. Reyes and R. Kelly, "Experimental evaluation of model–based controllers on a direct–drive robot arm," *Mechatronics* **11**, 267–282 (2001).
8. R. Kelly, "A linear state feedback plus adaptive feedforward control for DC servomotors," *IEEE Trans. on Industrial Electronics* **34**(2), 153–157 (May, 1987).
9. M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *ASME Journal of Dynamic Systems, Measurement and Control* **103**, 119–125 (1981).
10. R. Kelly, "PD control with desired gravity compensation: A review," *Int. J. Robotics Research* **16**(5), 660–672 (October, 1997).
11. R. Kelly, "A tuning procedure for stable PID control of robot manipulators," *Robotica* **13**(2), 141–148 (March–April, 1995).
12. P. Rocco, "Stability of PID control of industrial robot arms," *IEEE Trans. on Robotics and Automation* **12**(4), 606–614 (1996).
13. J. Alvarez–Ramirez, I. Cervantes and R. Kelly, "PID regulation of robot manipulators: Stability and performance," *Systems and Control Letters* **41**, 73–83 (2000).
14. Y. Choi and W. K. Chung, "PID performance tuning methods for a robot manipulator based on ISS," *Asian Journal of Control* **5**(2), 206–216 (June, 2003).
15. D. Koditscheck, "Natural motion for robot arms," *Proc. of the 1984 IEEE Conference on Decision and Control*, Las Vegas, NV (December, 1984) pp. 733–735.
16. B. Paden and R. Panja, "Globally asymptotically stable PD+ controller for robot manipulators," *International Journal of Control* **7**(6), 1697–1712 (1988).
17. L. Whitcomb, A. Rizzi and D. Koditschek, "Comparative experiments with a new adaptive controller for robot arms," *IEEE Trans. on Robotics and Automation* **9**(1), 59–70 (1993).
18. C. H. An, C. G. Atkeson and J. M. Hollerbach, *Model–based Control of a Robot Manipulator* (The MIT Press, 1988).
19. R. Kelly and R. Salgado, "PD control with computed feedforward of robot manipulators: A design procedure," *IEEE Trans. on Robotics and Automation* **10**(4), 566–571 (1994).
20. V. Santibañez and R. Kelly, "PD control with feedforward compensation for robot manipulators: analysis and experimentation," *Robotica* **19**, Part 1, 11–19 (2001).
21. J. J. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robotics Research* **6**(3), 49–59 (1987).
22. M. W. Spong, R. Ortega and R. Kelly, "Comments on "Adaptive manipulator control: A case study", " *IEEE Trans. on Automatic Control* **35**(6), 761–762 (1990).
23. O. Egeland and J. M. Godhavn, "A note on Lyapunov stability for an adaptive robot controller" *IEEE Trans. on Automatic Control* **39**(8), 1671–1673 (1994).
24. D. M. Dawson, J. J. Carroll and M. Schneider, "Integrator backstepping control of a brush DC motor turning a robotic load," *IEEE Trans. on Control System Technology* **2**, 233–244 (1994).
25. M. S. De Queiroz, D. Dawson and T. Burg, "Reexamination of the DCAL controller for rigid link robots," *Robotica* **14**, Part 1, 41–49 (1996).

https://doi.org/10.1017/S0263574704001031 Published online by Cambridge University Press