

Robustness Improvement of a Nonlinear H_∞ Controller for Robot Manipulators via Saturation Functions

Manuel G. Ortega, Manuel Vargas, Carlos Vivas, and Francisco R. Rubio
*Departamento Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros
University of Seville
Camino de los Descubrimientos
s/n—41092 Sevilla, Spain*

Received 20 April 2004; accepted 7 May 2005

In this paper, previous works on nonlinear H_∞ control for robot manipulators are extended. In particular, integral terms are considered to cope with persistent disturbances, such as constant load at the end-effector. The extended controller may be understood as a computed-torque control with an external PID, whose gain matrices vary with the position and velocity of the robot joints. In addition, in order to increase the controller robustness, an extension of the algorithms with saturation functions has been carried out. This extension deals with the resulting nonlinear equation of the closed-loop error. A modified expression for the required increment in the control signal is provided, and the local closed-loop stability of this approach is discussed. Finally, simulation results for a two-link robot and experimental results for an industrial robot are presented. The results obtained with this technique have been compared with those attained with the original controllers to show the improvements achieved by means of the proposed method.

© 2005 Wiley Periodicals, Inc.

1. INTRODUCTION

During the past few years, the control systems community has been paying increasing attention to the nonlinear H_∞ control theory,^{1–3} whose solution for the continuous nonlinear case was given by van der Schaft in his prominent article.⁴ The general approach leads to two Hamilton-Jacobi-Bellman-Isaacs partial derivative equations (HJBI PDEs), which replace the Riccati equations present in the linear H_∞ control for-

mulation. The main problem with the nonlinear case is that there is not a general method to solve these HJBI PDEs. Therefore, solutions have to be found for each particular case. Although some algorithms exist,^{5,6} they cannot be applied in cases where time appears explicitly in the equations.

Nonlinear H_∞ controllers have been applied to different kinds of systems, such as power converters,⁷ rigid spacecraft,⁸ chemical processes,⁹ and engine compressors.⁶ In this paper, applications related to ro-

botics are of special interest, where this theory has been used successfully in refs. 10–14, among others.

In ref. 11, a vector of disturbance signals acting upon the input channels is used to represent the combined effect of modelling errors and external disturbances. This analytical solution to the nonlinear H_∞ control for robot manipulator motion provides the control system with the ability to reject these disturbances (maintaining small tracking error) without excessive control effort. However, these results (successfully applied in ref. 14) present the following main restrictions:

1. **Null-average disturbances are assumed. This hypothesis is not very realistic if, for example, the robot has to carry a weight at its final element. This implies a persistent disturbance.**
2. **A perfect robot model is supposed, and it is assumed that uncertainties may be interpreted as null-average disturbances.**

The contributions of this paper are intended to solve these two handicaps:

1. **On the one hand, the nonlinear differential equation of the error has been modified, including a new term which allows us to penalize the integral of the robot position error. An analytical solution (with a similar structure to the original one) is also given. The conditions for formulating the controller in the form of a nonlinear PID are established, where the control signal can be penalized, as well as the error signals, their integral and their derivative.**
2. **On the other hand, despite the robustness of the preceding controller, it should be noted that it is designed exclusively for disturbance rejection, and a perfect model is assumed. To solve this limitation, a solution is proposed in this paper based on an extension to nonlinear systems of the classical algorithms with saturation functions (see, for example, refs. 15 and 16).**

The remainder of the paper is organized as follows: An approach upon the concepts of L_2 gain and H_∞ optimization in the context of nonlinear systems is introduced in Section 2. In Section 3 a suboptimal nonlinear controller is derived to maximize the robot manipulator's ability to reject external disturbances acting on the input channel, assuming a perfect system model. In Section 4, the nonlinear H_∞ controller is formulated as a computed-torque control with an

external nonlinear PID controller. To increase the robustness of the controller, the algorithms with saturation functions are extended to nonlinear systems in Section 5. A two-link robot and the *RM-10* industrial manipulator are used in Sections 6 and 7, respectively, as examples to show the performance of these controllers. Finally, the main conclusions drawn are given in Section 8.

2. NONLINEAR H_∞ CONTROL APPROACH

The dynamic equation of an n th order smooth nonlinear system which is affected by an unknown disturbance can be expressed as follows:

$$\dot{x} = f(x, t) + g(x, t)u + k(x, t)\omega, \quad (1)$$

where $u \in \mathcal{R}^p$ is the vector of control inputs, $\omega \in \mathcal{R}^q$ is the vector of external disturbances and $x \in \mathcal{R}^n$ is the vector of states. Performance can be defined using the cost variable $z \in \mathcal{R}^{(m+p)}$ given by the expression

$$z = W \begin{bmatrix} h(x) \\ u \end{bmatrix}, \quad (2)$$

where $h(x) \in \mathcal{R}^m$ represents the error vector to be controlled and $W \in \mathcal{R}^{(m+p) \times (m+p)}$ is a weighting matrix. If states x are assumed to be available for measurement (which is usual in robotics), then the optimal H_∞ problem can be posed as follows:⁴

Find the smallest value $\gamma^* \geq 0$ such that for any $\gamma \geq \gamma^*$ there exists a state feedback $u = u(x, t)$, such that the L_2 gain from ω to z is less than or equal to γ , that is,

$$\int_0^T \|z\|_2^2 dt \leq \gamma^2 \int_0^T \|\omega\|_2^2 dt. \quad (3)$$

The integral expression on the left-hand side of inequality (3) can be written as

$$\|z\|_2^2 = z^T z = \begin{bmatrix} h^T(x) & u^T \end{bmatrix} W^T W \begin{bmatrix} h(x) \\ u \end{bmatrix}$$

and the symmetric, positive definite matrix $W^T W$ can be partitioned as follows:

$$W^T W = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}.$$

Matrices Q and R are symmetric, positive definite and the fact that $W^T W > 0$ guarantees that $Q - SR^{-1}S^T > 0$.

The following structures, considered in this paper for matrices Q and S , constitute an extension of the original formulation:¹¹

$$Q = \begin{bmatrix} Q_1 & Q_{12} & Q_{13} \\ Q_{12} & Q_2 & Q_{23} \\ Q_{13} & Q_{23} & Q_3 \end{bmatrix}, \quad S = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}.$$

Under these assumptions, an optimal control signal $u^*(x, t)$ may be computed for system (1) if there exists a smooth solution $V(x, t)$, with $V(x_0, t) \equiv 0$ for $t \geq 0$, to the following HJBI equation:

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\partial^T V}{\partial x} f(x, t) + \frac{1}{2} \frac{\partial^T V}{\partial x} \left[\frac{1}{\gamma^2} k(x, t) k^T(x, t) \right. \\ \left. - g(x, t) R^{-1} g^T(x, t) \right] \frac{\partial V}{\partial x} - \frac{\partial^T V}{\partial x} g(x, t) R^{-1} S^T h(x) \\ + \frac{1}{2} h^T(x) (Q - SR^{-1}S^T) h(x) = 0 \end{aligned} \quad (4)$$

for each $\gamma > \sqrt{\sigma_{\max}(R)} \geq 0$, where σ_{\max} stands for the maximum singular value. In such a case, the optimal state feedback control law is derived as follows:¹¹

$$u^* = -R^{-1} \left(S^T h(x) + g^T(x, t) \frac{\partial V(x, t)}{\partial x} \right). \quad (5)$$

3. NONLINEAR H_∞ OPTIMIZATION IN MANIPULATOR MOTION CONTROL

The following Euler-Lagrange equations of motion are used to describe the behavior of an n degree-of-freedom (DOF) robot manipulator (see, for example, refs. 17 and 18):

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau + \tau_d \quad (6)$$

with

$$N(q, \dot{q}) = C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q),$$

where $q \in \mathfrak{R}^n$ is the vector of joint variables (joint positions) and \dot{q} is its time derivative (joint speeds). It is assumed that these two vectors are available for measurements. Vector τ (generalized torques applied on the joint axes) is the input signal of the system and τ_d represents the total effect of system modelling errors and external disturbances. The inertia matrix $M(q)$ is symmetric positive definite, $C(q, \dot{q})\dot{q}$ is the vector of centripetal and Coriolis terms, $F(\dot{q})$ represents the friction terms, and $G(q)$ denotes the gravity terms.

As is known,¹⁹ matrix $C(q, \dot{q})$ of the centripetal term is not unique. For the sake of convenience, in this paper this matrix will be computed through the following expression:

$$C(q, \dot{q}) = \frac{1}{2} \dot{M}(q, \dot{q}) + \mathcal{N}(q, \dot{q}), \quad (7)$$

where the terms $\dot{M}(q, \dot{q})$ and $\mathcal{N}(q, \dot{q})$ are given by

$$\dot{M}_{ij} = \frac{d}{dt} M_{ij} = \frac{\partial M_{ij}}{\partial q} \dot{q} = \sum_{k=1}^n \frac{\partial M_{ij}}{\partial q_k} \dot{q}_k, \quad (8)$$

$$\mathcal{N}_{ij} = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right) \dot{q}_k. \quad (9)$$

Using q_r , \dot{q}_r and \ddot{q}_r to denote the desired joint position, speed and acceleration, respectively, the tracking error vector, x , and its derivative, \dot{x} , can be defined as follows:

$$x(t) = \begin{bmatrix} \dot{e}(t) \\ e(t) \\ \int e(t) dt \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \ddot{e}(t) \\ \dot{e}(t) \\ e(t) \end{bmatrix}, \quad (10)$$

where

$$\ddot{e} = \ddot{q} - \ddot{q}_r,$$

$$\dot{e} = \dot{q} - \dot{q}_r,$$

$$e = q - q_r,$$

$$\int_0^t \omega dt = \int_0^t (q - q_r) dt.$$

Note that an integral term has been included in the vector error. This term will allow us to achieve a null steady-state error when persistent disturbances are acting on the system.

For system (6), a control law with the following structure is considered:

$$\tau = M(q)\ddot{q} + N(q, \dot{q}) - \frac{1}{\rho}(M(q)T\dot{x} + C(q, \dot{q})Tx) + \frac{1}{\rho}u. \quad (11)$$

This proposed control law can be split up into three different parts: the first one consists of the first two terms of that equation, which are designed in order to compensate for the robot dynamics [see Eq. (6)]. The second part consists of terms including error vector x and its derivative, \dot{x} . Assuming $\tau_d \equiv 0$, these two terms of the control law enable perfect tracking, which means that they represent the *essential* control effort needed to perform the task. Finally, the third part includes a vector u , which represents the *additional* control effort needed for disturbance rejection.

It can also be pointed out that, despite the preceding control law might seem a not well posed system, it will be shown afterwards that the computed torque does not rely on joint accelerations, but on their references.

Matrix T in Eq. (11) can be partitioned as follows:

$$T = [T_1 \ T_2 \ T_3] \quad (12)$$

with $T_1 = \rho I$, where ρ is a positive scalar and I is the n th-order identity matrix.

Substituting the expression of the control law from (11) into the Euler-Lagrange equation of the robot [Eq. (6)] and defining $\omega = \rho\tau_d$, yields

$$M(q)T\dot{x}(t) + C(q, \dot{q})Tx(t) = u(t) + \omega(t). \quad (13)$$

This expression represents the *dynamic equation of the system error*. It is a *nonlinear* 3nth order equation since its coefficients [matrices $M(q)$ and $C(q, \dot{q})$] vary with time. Taking into account this nonlinear equation, the nonlinear H_∞ control problem can be posed as follows:

“Find a control law $u(t)$ such that the ratio between the energy of the cost variable $z = W[h^T(x)u^T]^T$ and the en-

ergy of the disturbance signals ω is less than a given attenuation level γ .”

To apply the theoretical results presented in Section 2, it is necessary to rewrite the nonlinear dynamic equation of the error [Eq. (13)] into the standard form of the nonlinear H_∞ problem [see Eq. (1)]. This can be done by defining the following expressions:

$$f(x, t) = T_o^{-1} \begin{bmatrix} -M^{-1}C & O & O \\ \frac{1}{\rho}I & I - \frac{1}{\rho}T_2 & -I - \frac{1}{\rho}(T_3 - T_2) \\ 0 & I & -I \end{bmatrix} T_o x, \quad (14)$$

$$g(x, t) = k(x, t) = T_o^{-1} \begin{bmatrix} M^{-1} \\ O \\ O \end{bmatrix}, \quad (15)$$

where I is the identity matrix, O the zero matrix, both of n th order and

$$T_o = \begin{bmatrix} T_1 & T_2 & T_3 \\ O & I & I \\ O & O & I \end{bmatrix}. \quad (16)$$

As stated in Section 2, the solution of the HJBI equation depends on the choice of the cost variable, z , and particularly on the selection of function $h(x)$ [see Eq. (2)]. In this paper, this function is taken to be equal to the error vector, that is, $h(x) = x$. Once this function has been selected, computing the control law, u , will require finding the solution, $V(x, t)$, to the HJBI equation posed in the previous section [see Eq. (4)]. The following theorem will help to do this.

Theorem: Let $V(x, t)$ be the scalar function

$$V(x, t) = \frac{1}{2} x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X - Y \\ O & X - Y & Z + Y \end{bmatrix} T_o x, \quad (17)$$

where X , Y and $Z \in \mathfrak{R}^{n \times n}$ are constant, symmetric, and positive definite matrices such that $Z - XY^{-1}X + 2X > 0$, M is the inertia matrix of the robot, and T_o is as defined in (16). Let $T = [T_1 \ T_2 \ T_3]$ be the matrix appearing in dynamic equation (13). If these matrices verify the following equation:

$$\begin{bmatrix} O & Y & X \\ Y & 2X & Z+2X \\ X & Z+2X & O \end{bmatrix} + Q + \frac{1}{\gamma^2} T^T T - (S^T + T)^T R^{-1} (S^T + T) = 0 \quad (18)$$

then, function $V(x, t)$ constitutes a solution to the HJBI (4), for a sufficiently high value of γ .

The proof of this theorem can be found in the Appendix at the end of the paper.

The algorithm for obtaining matrix T is the following:

1. Compute T_1 and T_3 by solving the following Riccati algebraic equations:

$$\begin{aligned} T_1^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) T_1 - S_1 R^{-1} T_1 - T_1^T R^{-1} S_1^T \\ - S_1 R^{-1} S_1^T + Q_1 = O, \end{aligned}$$

$$\begin{aligned} T_3^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) T_3 - S_3 R^{-1} T_3 - T_3^T R^{-1} S_3^T \\ - S_3 R^{-1} S_3^T + Q_3 = O. \end{aligned}$$

2. Compute matrix X by means of the following expression:

$$\begin{aligned} X = - \left[T_1^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) T_3 - S_1 R^{-1} T_3 - T_1^T R^{-1} S_3^T \right. \\ \left. - S_1 R^{-1} S_3^T + Q_{13} \right]. \end{aligned}$$

3. Compute T_2 by solving the Riccati algebraic equation:

$$\begin{aligned} T_2^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) T_2 - S_2 R^{-1} T_2 - T_2^T R^{-1} S_2^T \\ - S_2 R^{-1} S_2^T + Q_2 + 2X = O. \end{aligned}$$

After this, $V(x, t)$ can be computed. Then, substituting $V(x, t)$ in Eq. (5), control law u^* corresponding to the H_∞ optimal index γ is given by

$$u^* = -R^{-1} (S^T + T)x.$$

4. THE CONTROL LAW FORMULATED AS A COMPUTED-TORQUE NONLINEAR PID

Several methods for synthesizing robot controllers in the form of PIDs have been reported in the control literature (see, for example, ref. 20). In this section it is shown how the obtained control law may be interpreted as a computed-torque control with an external nonlinear PID controller.

If we replace the expressions for T , \dot{x} and u^* in (11), after some manipulation, the optimal control law can be written as

$$\begin{aligned} \tau^* = M\ddot{q}_r + N - \frac{1}{\rho} \left(M \begin{bmatrix} T_2 & T_3 \end{bmatrix} \begin{bmatrix} \dot{e} \\ e \end{bmatrix} + CTx \right) \\ - \frac{1}{\rho} R^{-1} (S^T + T)x. \end{aligned}$$

Keeping in mind the definition of x , the control law can be rewritten as follows:

$$\begin{aligned} \tau^* = M\ddot{q}_r + N - \frac{1}{\rho} M \left(\begin{bmatrix} T_2 & T_3 & 0 \end{bmatrix} + M^{-1} CT \right) \\ + M^{-1} R^{-1} (S^T + T)x \end{aligned}$$

or in a more compact form:

$$\tau^* = M(q)\ddot{q}_r + N(q, \dot{q}) - M(q) \left(K_D \dot{e} + K_P e + K_I \int e dt \right), \quad (19)$$

where

$$K_D = \frac{1}{\rho} (T_2 + M^{-1} CT_1 + M^{-1} R^{-1} (S_1^T + T_1)),$$

$$K_P = \frac{1}{\rho} (T_3 + M^{-1} CT_2 + M^{-1} R^{-1} (S_2^T + T_2)),$$

$$K_I = \frac{1}{\rho} (M^{-1} CT_3 + M^{-1} R^{-1} (S_3^T + T_3)).$$

It is easy to see that Eq. (19) represents a *computed-torque control law* with an *external PID controller*. This external PID is a nonlinear one since its gain matrices are time-varying, like matrices $M(q)$ and $C(q, \dot{q})$. Besides this, from Eq. (19) it can be concluded that the

control law does not depend on the joint accelerations, preventing the system to be ill-posed as it could seem from Eq. (11).

A particular case can be obtained when the components of the weighting compound $W^T W$ verify:

$$Q_1 = w_1^2 I, \quad Q_2 = w_2^2 I, \quad Q_3 = w_3^2 I, \quad R = w_u^2 I,$$

$$Q_{12} = Q_{13} = Q_{23} = O, \quad S_1 = S_2 = S_3 = O. \quad (20)$$

In this case, the following analytical expressions for the gain matrices have been obtained:

$$K_D = \frac{\sqrt{w_2^2 + 2w_1 w_3}}{w_1} I + M^{-1} \left(C + \frac{1}{w_u^2} I \right),$$

$$K_P = \frac{w_3}{w_1} I + \frac{\sqrt{w_2^2 + 2w_1 w_3}}{w_1} M^{-1} \left(C + \frac{1}{w_u^2} I \right),$$

$$K_I = \frac{w_3}{w_1} M^{-1} \left(C + \frac{1}{w_u^2} I \right).$$

These expressions have an important property: *they do not depend on the parameter γ* . Thus, we have algebraic expressions for computing the general optimal solution for this particular case.

Finally, it should be pointed out that the controller proposed in ref. 11 is a particular case of the previous control law. In fact, the original controller is obtained if parameter w_3 is set to a null value. Therefore, the controller presented in ref. 11 may be interpreted as a computed-torque control scheme, with an *external nonlinear PD controller*.

5. ROBUST CONTROL DESIGN

Despite the robustness provided by the preceding controller, in the previous sections a complete knowledge of the robot model is presumed. It is assumed that uncertainties can be interpreted as disturbances included in the term τ_d .

Since this hypothesis is not very realistic, in this section we propose an additional control term to be added to the previous controller, in order to increase the system robustness when there are modeling errors. Specifically, an extension of the algorithms with saturation functions for the preceding nonlinear controller is proposed. A interesting survey about differ-

ent ways to implement this method can be found in ref. 15, all of them based on linear error equations.

Next, a brief summary of a classical algorithm with saturation functions is given. A more detailed description can be found in ref. 16.

5.1. The Classical Algorithm with Saturation Functions

Let $M(q)$ and $N(q, \dot{q})$ be the dynamic matrices of the Euler-Lagrange equations (6) and $\hat{M}(q)$ and $\hat{N}(q, \dot{q})$, their respective estimations. The following hypotheses are assumed:

1. $\sup_{t \geq 0} \|\ddot{q}_r\| < Q_{max} < \infty$.
2. $\|E(q)\| \leq \alpha \leq 1$ for some value of α , and for all $q \in \mathfrak{R}^n$, where $E(q) \equiv M(q)^{-1} \hat{M}(q) - I$.
3. $\|\Delta N(q, \dot{q})\| \leq \phi(x, t)$ for some time-bounded function $\phi(x, t)$, where $\Delta N(q, \dot{q}) = N(q, \dot{q}) - \hat{N}(q, \dot{q})$ and, in this case, $x(t) = [\dot{e}(t) e(t)]^T$.

Once the values of the bounds have been calculated, this method proposes a computed-torque structure where the robot dynamics is linearized with the estimated matrices (see Figure 1). If there were no uncertainty and the external controller were a linear PD control law (with *constant* gain matrices K_P and K_D , respectively), the closed-loop error dynamics would satisfy the following *linear* differential equation:

$$\dot{x}(t) = \bar{A}x(t) = (A - BK)x(t), \quad (21)$$

where the *constant* matrices A , B and K are as follows:

$$A = \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix}, \quad B = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad K = [K_D \ K_P].$$

Since estimations are not perfect, the following algorithm is proposed to achieve an appropriate control signal $v(t)$ (see Figure 1):

1. **Design a control law $v(t)$ as follows:**

$$v(t) = \ddot{q}_r(t) - Kx(t) + \Delta v(t).$$

The next dynamic equation of the closed-loop error is obtained with this linearization:

$$\dot{x} = \bar{A}x + B(\Delta v + \eta),$$

where K should be designed such that $\bar{A} = A - BK$ is Hurwitz, and function η has the following expression:

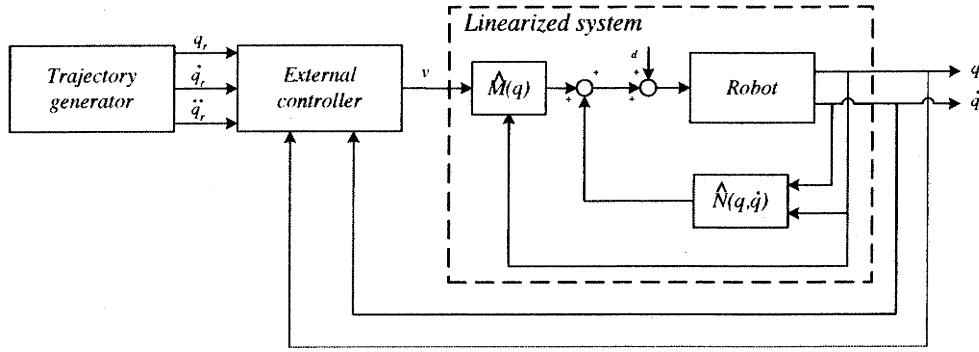


Figure 1. Computed-torque control scheme.

$$\eta = E\Delta v + E(\ddot{q}_r - Kx) + M^{-1}\Delta N.$$

2. Compute the function $\xi(x, t)$ satisfying

$$\|\eta\| < \xi(x, t), \quad \|\Delta v\| < \xi(x, t) \quad (22)$$

as follows:

$$\xi(x, t) = \frac{1}{1 - \alpha} [\alpha Q_{max} + \alpha \|K\| \|x\| + \bar{M} \phi(x, t)], \quad (23)$$

where $\|M(q)^{-1}\| \leq \bar{M} \forall q \in \mathcal{R}^n$.

3. Since \bar{A} is Hurwitz, select a symmetric, positive definite matrix Φ and find the unique symmetric, positive definite solution P of the following Lyapunov's equation:

$$\bar{A}^T P + P \bar{A} + \Phi = 0. \quad (24)$$

4. Finally, compute the term $\Delta v(t)$ by means of the following expression:

$$\Delta v(t) = \begin{cases} -\xi(x, t) \frac{B^T P x}{\|B^T P x\|} & \text{if } \|B^T P x\| \geq \epsilon, \\ -\frac{\xi(x, t)}{\epsilon} B^T P x & \text{if } \|B^T P x\| < \epsilon. \end{cases} \quad (25)$$

5.2. Extension of the Algorithms with Saturation Functions to Nonlinear Systems

The main idea is to include an algorithm similar to the one described above in the control law stated in

Section 4. However, the classical method just described cannot be used since the external controller consists of *nonlinear gain matrices*, which implies a *nonlinear closed-loop error equation*.

This difficulty can be solved taking into account that:

- The dynamics of the closed-loop error attained with the controller in Section 4 is stable for the nominal case (null uncertainty).
- Lyapunov's equation (24) in the linear case is obtained from the assumption of a quadratic function $\Psi(x) = x^T P x$. The constant, symmetric, positive definite matrix P is the one used for the calculation of $\Delta v(t)$ in Eq. (25) through the term $B^T P x$.

Therefore, an analogy between linear and nonlinear optimal control (see ref. 2, for instance) can be carried out by comparing the control laws obtained for the linear case ($-B^T P x$) and for the nonlinear one $[-\sigma^T \partial \Psi(x, t) / \partial x]$.

Bearing in mind this analogy, the classical algorithm may be modified as follows:

Let $M(q)$ and $N(q, \dot{q})$ be the dynamic matrices of the Euler-Lagrange equations (6) and $\hat{M}(q)$ and $\hat{N}(q, \dot{q})$ their respective estimates. Assuming the same hypotheses as for the classical method, a control signal $v(t)$ can be attained by means of the following algorithm:

1. Design an external control law v as follows:

$$v(t) = \ddot{q}_r(t) - Kx(t) + \Delta v(t), \quad (26)$$

where x is the error vector defined in Section 4, $K = [K_D K_P K_I]$ is the matrix of the external nonlinear PID controller in Eq. (19) (substituting matrices M and N for their estimates) and $\Delta v(t)$ has the same meaning as in the classical method.

Following the methodology described in Section 5.1 for the linear PD external controller, an expression for the dynamics of the closed-loop error can be obtained as follows:

$$\dot{x} = \bar{f}(x, t) + B(\Delta v + \eta), \quad (27)$$

where $\bar{f}(x, t) \equiv \bar{A}(q, \dot{q})x(t)$. In this case, the dynamic matrix of the error is expressed as follows:

$$\bar{A}(q, \dot{q}) = A - BK(q, \dot{q}), \quad (28)$$

where, in this case, matrices A and B are as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}, \quad B = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}.$$

Notice the time dependency of the vector $\bar{f}(x, t)$ through the time variation of the joint positions and velocities.

2. Compute a scalar function $\xi(e, t)$ by means of Eq. (23), where the inequalities in (22) are assumed to be satisfied. Notice that only one boundary of the uncertainties is computed, which is independent of the method applied.
3. Find a scalar function $\Psi(x, t) \geq 0$ that satisfies the following inequality:

$$\frac{\partial \Psi(x, t)}{\partial t} + \frac{\partial^T \Psi(x, t)}{\partial x} \bar{f}(x, t) < 0, \quad \forall x \neq 0. \quad (29)$$

It should be remembered that matrix K has been designed such that the dynamic vector of the error, $\bar{f}(x, t)$, is stable in the nominal case (null uncertainties).

4. Finally, to complete the control law of Eq. (26), design the term $\Delta v(t)$ according to the following expression:

$$\Delta v(t) = \begin{cases} -\xi(x, t) \frac{B^T \frac{\partial \Psi(x, t)}{\partial x}}{\left\| \frac{\partial \Psi(x, t)}{\partial x} \right\|} & \text{if } \left\| B^T \frac{\partial \Psi(x, t)}{\partial x} \right\| \geq \epsilon, \\ \frac{-\xi(x, t) \frac{\partial \Psi(x, t)}{\partial x}}{\epsilon} & \text{if } \left\| B^T \frac{\partial \Psi(x, t)}{\partial x} \right\| < \epsilon. \end{cases} \quad (30)$$

Notice that $\Delta v(t)$ is again linearized for a value of $\|B^T[\partial \Psi(x, t)/\partial x]\|$ less than ϵ .

The proof of the closed-loop stability when (30) is applied may be carried out by means of Lyapunov's second method. Thus, let $\Psi(x, t) \geq 0$ be a scalar function that satisfies (29). The total time derivative of $\Psi(x, t)$ is

$$\frac{d\Psi(x, t)}{dt} = \frac{\partial \Psi(x, t)}{\partial t} + \frac{\partial^T \Psi(x, t)}{\partial x} \dot{x}(t).$$

Taking into account Eq. (27), it follows that

$$\begin{aligned} \frac{d\Psi(x, t)}{dt} &= \frac{\partial \Psi(x, t)}{\partial t} + \frac{\partial^T \Psi(x, t)}{\partial x} [f(x, t) + B(\Delta v + \eta)] \\ &< \frac{\partial^T \Psi(x, t)}{\partial x} B(\Delta v + \eta), \end{aligned}$$

where relation (29) has been used. Hence, the time derivative of $\Psi(x, t)$ is negative if the following inequality is satisfied:

$$\frac{\partial^T \Psi(x, t)}{\partial x} B (\Delta v + \eta) \leq 0.$$

Therefore, the term Δv must be chosen such that

$$\frac{\partial^T \Psi(x, t)}{\partial x} B \Delta v \leq - \frac{\partial^T \Psi(x, t)}{\partial x} B \eta.$$

Assuming the worst case, that is, vectors $B^T[\partial\Psi(x, t)/\partial x]$ and η with the same direction. Equation (22) enables us to conclude that the direction of Δv must be opposite that of $B^T[\partial\Psi(x, t)/\partial x]$ (in order to counteract the effect of η) and with its module equal to $\xi(x, t)$, that is

$$\Delta v = - \xi(x, t) \left\| \frac{B^T \frac{\partial \Psi(x, t)}{\partial x}}{\left\| \frac{B^T \frac{\partial \Psi(x, t)}{\partial x} \right\|} \right\|.$$

Due to the complexity of the resulting HJ equation, no analytical expression is available yet for scalar function $\Psi(x, t)$ in Eq. (29). In this application, we have used the expression $x^T P(t)x$ as an approximation of $\Psi(x, t)$. Matrix $P(t)$ is obtained as the solution to the following Lyapunov's equation:

$$\bar{A}_0^T P + P \bar{A}_0 + \Phi = 0,$$

where Φ is again a symmetric, positive definite matrix and \bar{A}_0 is the Jacobian of vector $\tilde{f}(x, t)$ evaluated at (t_0, x_0) , that is

$$\bar{A}_0 = \left. \frac{\partial \tilde{f}(x, t)}{\partial x} \right|_{t=t_0, x=x_0}.$$

It should be noted that \bar{A}_0 is Hurwitz [see Eq. (28)] and therefore $P(t)$ is well defined. Aizerman's conjecture (see ref. 21) is satisfied when the time derivative is assumed to be small enough with respect to the spatial one. Hence, under this assumption, the stability of this design can be guaranteed.

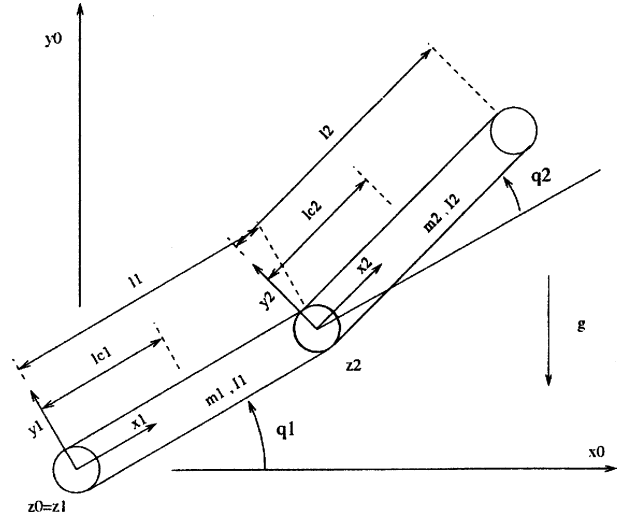


Figure 2. Scheme of the 2-DOF robot.

6. A CASE STUDY: TWO-LINK ROBOT MANIPULATOR

In this section, the preceding control laws are tested on an academic example, in which the model used for simulation is detailed. Thus, simulation experiments on a two-link planar robot were carried out to assess the performance of the three control schemes: the original one proposed in ref. 11, the *external nonlinear H_∞ /PID* proposed in Section 4, and the *robust external nonlinear H_∞ /PID* proposed in Section 5.2. These three controllers will be referred in the following as NL PD, NL PID and RNL PID, respectively.

Figure 2 shows the scheme of the 2-DOF planar robot considered. It is made of two links and two revolute joints. Each link is characterized by the following parameters: mass (m_i), length (l_i), mass center position (l_{ci}), and inertia (I_i), where $i=1, 2$.

In this case, the matrices and vectors involved in the Euler-Lagrange equations (6) are the following:

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{12}(q) & M_{22}(q) \end{bmatrix},$$

where

$$M_{11}(q) = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2,$$

$$M_{12}(q) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2,$$

Table I. Nominal robot parameters.

Parameter	Nominal value
Lengths	$l_1=l_2=1$ m
Position of mass centers	$l_{c1}=l_{c2}=0.5$ m
Masses	$m_1=m_2=3$ Kg
Inertia moments	$I_1=I_2=0.2536$ Kg m ²

$$M_{22}(q) = m_2 l_{c2}^2 + I_2.$$

Matrices $\dot{M}(q, \dot{q})$ and $\mathcal{N}(q, \dot{q})$, computed using expressions (8) and (9), are as follows:

$$\dot{M}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin q_2 \begin{bmatrix} 2\dot{q}_2 & \dot{q}_0 \\ \dot{q}_0 & 0 \end{bmatrix},$$

$$\mathcal{N}(q, \dot{q}) = m_2 l_1 l_{c2} \sin q_2 \left(\dot{q}_1 + \frac{1}{2} \dot{q}_2 \right) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

and, therefore, the next expression is obtained for matrix $C(q, \dot{q})$:

$$C(q, \dot{q}) = \frac{1}{2} \dot{M} + \mathcal{N} = m_2 l_1 l_{c2} \sin q_2 \begin{bmatrix} -\dot{q}_2 & -\dot{q}_2 - \dot{q}_1 \\ \dot{q}_1 & 0 \end{bmatrix}.$$

Finally, the gravity term is as follows:

$$G(q) = g \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) \cos q_1 + m_2 l_{c2} \cos(q_1 + q_2) \\ m_2 l_{c2} \cos(q_1 + q_2) \end{bmatrix},$$

with g being the gravity constant.

Table I gives the nominal values for the robot parameters, which are employed for the controller synthesis.

In order to check the performance of the controllers, some uncertainty has been included in the robot parameters. Each model parameter used for simulation contains a pseudo-random deviation with respect to its nominal value, with a maximum deviation of $\pm 20\%$. In addition, in order to introduce some disturbances into the system, persistent torques of values 7 Nm and 2 Nm were applied at joints 1 and 2, respectively, at simulation times 1.25 seconds and 0.75 seconds.

In the simulations presented in this section, the position references (provided by a trajectory generator) are fifth degree polynomials between the initial

Table II. Weights for the controller of the 2 DOF robot.

Signal	Weighting submatrix	NL PD	NL PID and RNL PID
Speed error \dot{e}	Q_1	$(\frac{1}{2})^2 I$	$(\frac{1}{2})^2 I$
Position error e	Q_2	I	I
Error integral $\int e dt$	Q_3		$2^2 I$
Control effort u	R	$0.01^2 I$	$0.01^2 I$

position $[q_1 \ q_2] = [0 \ 0]$ rad to the final position equal to $[\pi/4 \ \pi/4]$ rad, with initial and final speeds and accelerations equal to zero. The transition time is 2 seconds.

For controller design, a diagonal $W^T W$ weighting matrix was considered. Table II shows the values for the diagonal weighting submatrices [see Eq. (20)] for the 2 DOF planar robot.

Figure 3 presents the simulation results obtained with the NL PD controller. It should be noted that the position errors do not tend to zero when a persistent disturbance is applied.

This limitation can be overcome with the NL PID controller. Figure 4 shows the simulation results attained with this control law. In can be seen that the position errors tend to null values; furthermore, the maximum position error values are smaller than those of the preceding results, with an improvement of approximately 65%.

Finally, simulation results obtained with the RNL

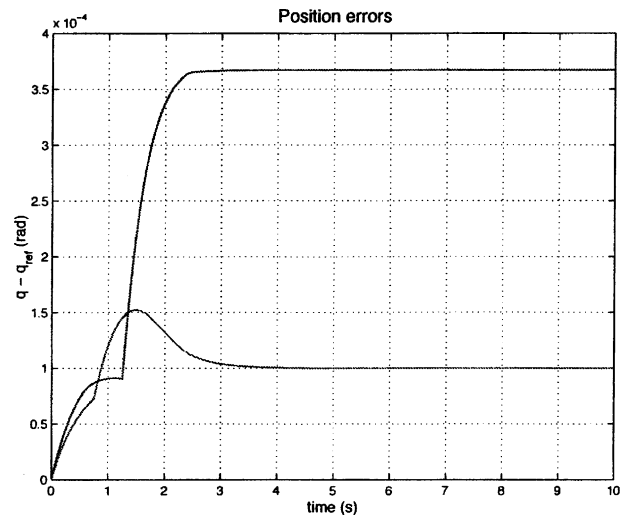


Figure 3. Results with the NL PD controller.

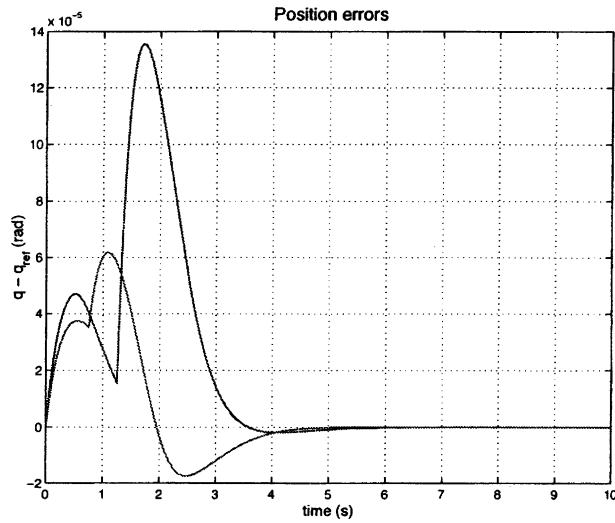


Figure 4. Results with the NL PID controller.

PID controller are shown in Figure 5. The following values of the parameters described in Section 5 have been numerically computed: $Q_{max}=6.505$, $\alpha=0.7053$, $\bar{M}=3.5981$. Function $\phi(x,t)$ has been chosen such that²²

$$\begin{aligned} \|\Delta N(q, \dot{q})\| &= \|\Delta C\dot{q} + \Delta G\| \leq \|\Delta C\dot{q}\| + \|\Delta G\| \leq \gamma_v \|\dot{e}\|^2 + \gamma_g \\ &\equiv \phi(x,t). \end{aligned}$$

The values of $\gamma_v=26.683$ and $\gamma_g=1.7266$ also have

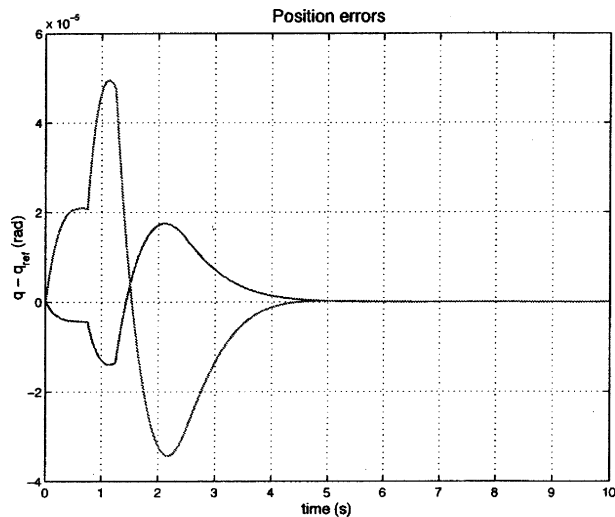


Figure 5. Results with the RNL PID controller.

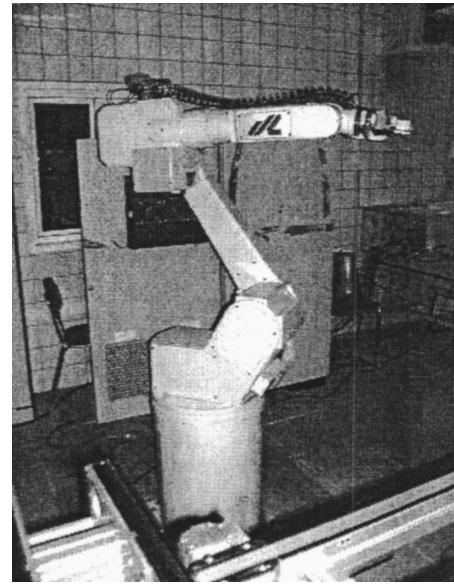


Figure 6. The RM-10 robot manipulator.

been computed numerically. Finally, parameter ϵ has been chosen as 10^{-6} .

With regard to the results attained by the last controller, it is clear that, besides having a null steady-state position error, their maximum peaks are again smaller than with the preceding controller, with an additional reduction of approximately 60%.

7. EXPERIMENTAL RESULTS: APPLICATION TO THE RM-10 INDUSTRIAL ROBOT

The RM-10 robot, shown in Figure 6, is a six-degree-of-freedom revolute joint manipulator arm.²³ All the six joints are driven by dc-brushless low-inertia electric motors which provide a uniform torque for every joint angle, and enable high control torque peaks. Torque is delivered to the joint axes through gear reductions; thus the RM-10 is an indirect-drive manipulator. An electric brake is also provided for each joint in order to lock the manipulator arm in any given position.

Coupled to each motor axis there is a two-pole resolver device which provides an accurate measurement of the corresponding joint position. These measures allow, as usual, the closed-loop control of the system.

The original control hardware used by the RM-10 robot system was replaced by a DS1103 real-time control board from dSPACE (see refs. 24–26).

Table III. Estimated masses of the links.

Link	Mass (Kg)
1	38.65
2	51.80
3	84.10
4	33.89
5	7.36
6	5.00

This control board, holding a 333 MHz PowerPC as a main processor and a DSP as an input/output processor, was plugged into the expansion bus of a standard personal computer.

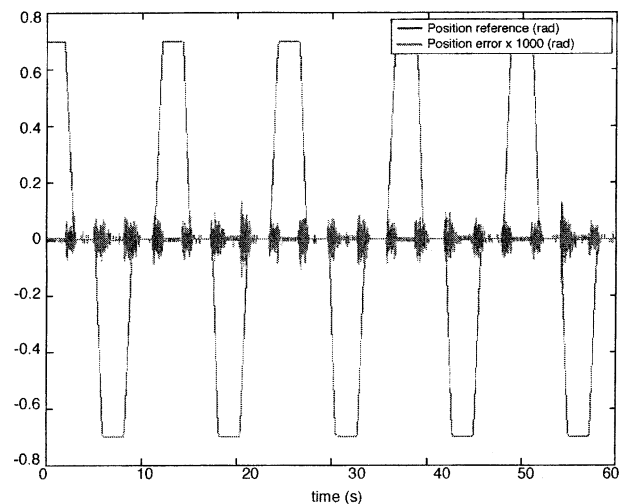
Before designing any controller it is necessary to obtain a dynamic model of the robot manipulator. According to the Euler-Lagrange formulation, the dynamic model of a general n -link rigid-body robot is a second order nonlinear equation, as shown in Eq. (6). This yields a complex, hard-to-handle nonlinear motion equation.²⁷ Since the considered robot is an indirect-drive manipulator, and since the effects of its gear reductions are quite relevant, there is a clear dominance of the diagonal terms in the inertia matrix. Thus, some dynamic coupling is only detected at the lowest joints. The parameters of the robot model were estimated by geometric measurements and dynamic experiments on the robot arm. In Table III, the estimated masses of the different links of the robot are shown. These values may help to give an idea of the robot's bulkiness.

The values of the diagonal weighting submatrices used for the *RM-10* control synthesis are the following: $Q_1=0.5^2I$, $Q_2=I$, $Q_3=3^2I$, $R=0.4^2I$. The remainder of the parameters needed for the RNL PID controller were computed numerically, as in the previous section.

References were supplied by a trajectory generator in the form of smoothed trapezoidal waves. In order to make a comparison, Figure 7 shows the experimental results for the first joint obtained with a computed torque structure with a *linear* PID as external controller.

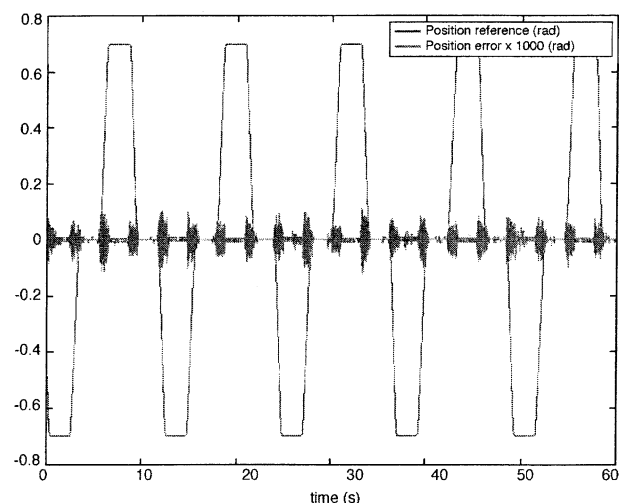
The results obtained with the NL PD were not satisfactory. Due to the lack of integral effect, the robot was not able to follow the reference after several periods of the trajectory. Therefore, these results have not been included.

Figures 8 and 9 depict the experimental results (also for the first joint) obtained with the NL PID and

**Figure 7.** Results with a *linear* PID controller.

RNL PID controllers, respectively. Despite they look like similar to the results of Figure 7, it has been numerically confirmed that the NL PID controller provides a reduction of the maximum error of a 15% approximately. Likewise, in the case of the RNL PID controller, an additional reduction of a 8% is attained respect to the NL PID controller.

It should be emphasized that improvements were only appreciated at the lowest joints. This is due to gear reductions, which produces that some dynamic coupling only appears at these joints. Actually, additional simulation experiments were carried out with

**Figure 8.** Results with the NL PID controller.

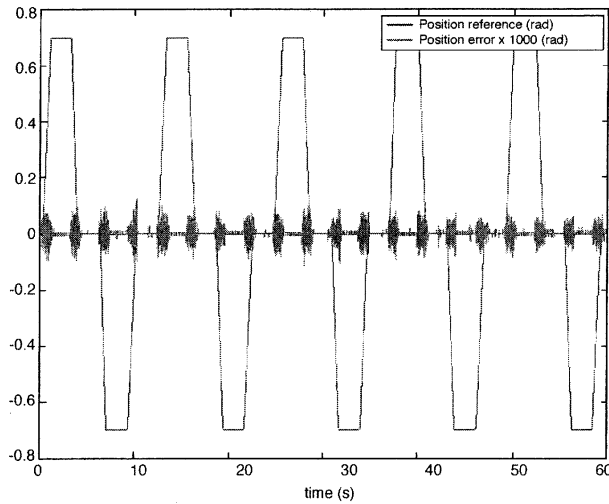


Figure 9. Results with the RNL PID controller.

a tested model of the robot. These simulations showed that if gear reductions are included, results similar to the ones performed with the real robot are attained. However, if gear reductions are removed, similar improvements to those obtained in the case study presented in Section 6 are achieved for all joints.

8. CONCLUSIONS

In this paper, a nonlinear H_∞ controller for robot manipulators is proposed which deals with persistent disturbances due to the inclusion of an integral term. This controller can be interpreted as a computed-torque structure with an external nonlinear PID controller. A particular case has been formulated in which the nonlinear PID gains do not depend on the value of the attenuation level γ .

In addition, some robustness improvement was provided to the previous controller by means of an extension of the traditional algorithms with saturation functions. Based on the classical method, where a linear equation is used, this extension is intended to handle a nonlinear error equation. This way, the resulting Lyapunov's equation was replaced by a Hamilton-Jacobi-Bellman-Isaacs inequality, which was solved by linearization at each operating point. A modified expression for the control signal increment was supplied, showing that closed-loop stability is satisfied.

Simulation results on a two-link planar robot and

experimental results on the *RM-10* industrial robot have been presented. Several tests were carried out taking into account the differences between the model used for controller synthesis and the one implemented on the simulator. Finally, improvements obtained using the proposed algorithm have been shown by comparison with the results attained with the original controllers.

9. APPENDIX: PROOF OF RESULTS IN SECTION 3

First of all, in this appendix, we will show that scalar function $V(x,t)$ defined in (17) is positive definite. Next, the fact that this function constitutes a solution for the Hamilton-Jacobi equation (4) will be proven.

9.1. Positive-Definiteness of $V(x,t)$

Function $V(x,t)$ is positive definite if and only if

$$\begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} > 0.$$

Since the inertia matrix M is symmetric positive definite, and under the assumption of X , Y and $Z \in \mathfrak{R}^{n \times n}$ are constant, symmetric, and positive definite matrices, the preceding inequality is verified if and only if

$$Z + Y - (X - Y)Y^{-1}(X - Y) > 0.$$

Expanding this expression yields

$$Z - XY^{-1}X + 2X > 0,$$

which is a theorem assumption.

9.2. $V(x,t)$ Is a Solution of the Hamilton-Jacobi Equation

Next, let us show that $V(x,t)$ constitutes a solution for the Hamilton-Jacobi equation (4). The gradient of $V(x,t)$ is given by

$$\frac{\partial^T V(x,t)}{\partial x} = x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} T_o + \frac{1}{2} [0_{1 \times n} \ \Omega \ 0_{1 \times n}],$$

where $\Omega \in \mathfrak{R}^{1 \times n}$ is equal to

$$\Omega = \left(x^T T_o^T \begin{bmatrix} \frac{\partial^T M}{\partial q_1} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x, \dots, x^T T_o^T \begin{bmatrix} \frac{\partial^T M}{\partial q_n} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x \right).$$

It is easy to verify that

$$[0_{1 \times n} \ \Omega \ 0_{1 \times n}] [g(x,t)u + k(x,t)\omega] = 0$$

and, consequently,

$$\begin{aligned} \frac{\partial^T V(x,t)}{\partial x} f &= \left(x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} T_o + \frac{1}{2} [0_{1 \times n} \ \Omega \ 0_{1 \times n}] \right) f \\ &= x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} T_o f + \frac{1}{2} [0_{1 \times n} \ \Omega \ 0_{1 \times n}] f + \frac{1}{2} [0_{1 \times n} \ \Omega \ 0_{1 \times n}] (gu + k\omega) \\ &= x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} T_o f + \frac{1}{2} [0_{1 \times n} \ \Omega \ 0_{1 \times n}] \dot{x}. \end{aligned}$$

Using Eqs. (14) and (7), the first term of the last expression can be written as follows:

$$\begin{aligned} x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} T_o f &= x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X-Y \\ O & X-Y & Z+Y \end{bmatrix} \times \begin{bmatrix} -M^{-1} \left(\frac{1}{2} \dot{M} + \mathcal{N} \right) & O & O \\ \frac{1}{\rho} I & I - \left(\frac{1}{\rho} T_2 \right) & -I - \frac{1}{\rho} (T_3 - T_2) \\ O & I & -I \end{bmatrix} T_o x \\ &= x^T T_o^T \begin{bmatrix} - \left(\frac{1}{2} \dot{M} + \mathcal{N} \right) & O & O \\ \frac{1}{\rho} Y & X - \frac{1}{\rho} Y T_2 & -X - \frac{1}{\rho} Y (T_3 - T_2) \\ \frac{1}{\rho} (X - Y) & X + Z - \frac{1}{\rho} (X - Y) T_2 & - (X + Z) - \frac{1}{\rho} (X - Y) (T_3 - T_2) \end{bmatrix} T_o x, \end{aligned}$$

while the second term can be posed as

$$\frac{1}{2}[0_{1 \times n} \quad \Omega \quad 0_{1 \times n}]\dot{x} = \frac{1}{2}[0_{1 \times n} \quad \Omega \quad 0_{1 \times n}] \begin{bmatrix} \ddot{e} \\ \dot{e} \\ e \end{bmatrix} = \frac{1}{2}\Omega\dot{e} = \frac{1}{2}x^T T_o^T \begin{bmatrix} \sum_{k=1}^n \frac{\partial^T M}{\partial q_k} (\dot{q}_k - \dot{q}_{rk}) & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x.$$

The derivative of $V(x, t)$ with respect to time is

$$\frac{\partial V}{\partial t} = \frac{1}{2}x^T T_o^T \begin{bmatrix} \sum_{k=1}^n \frac{\partial^T M}{\partial q_k} \dot{q}_{rk} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x.$$

Adding the previous expressions, the following result is obtained:

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\partial^T V(x, t)}{\partial x} f &= \frac{\partial V}{\partial t} + x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X - Y \\ O & X - Y & Z + Y \end{bmatrix} T_o f + \frac{1}{2}[0_{1 \times n} \quad \Omega \quad 0_{1 \times n}]\dot{x} \\ &= \frac{1}{2}x^T T_o^T \begin{bmatrix} \sum_{k=1}^n \frac{\partial^T M}{\partial q_k} \dot{q}_{rk} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x + x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X - Y \\ O & X - Y & Z + Y \end{bmatrix} T_o f \\ &\quad + \frac{1}{2}x^T T_o^T \begin{bmatrix} \sum_{k=1}^n \frac{\partial^T M}{\partial q_k} (\dot{q}_{rk} - \dot{q}_{rk}) & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x \\ &= x^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X - Y \\ O & X - Y & Z + Y \end{bmatrix} T_o f + \frac{1}{2}x^T T_o^T \begin{bmatrix} \sum_{k=1}^n \frac{\partial^T M}{\partial q_k} \dot{q}_k & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x \\ &= x^T T_o^T \begin{bmatrix} -\left(\frac{1}{2}\dot{M} + \mathcal{N}\right) & O & O \\ \frac{1}{\rho}Y & X - \frac{1}{\rho}YT_2 & -X - \frac{1}{\rho}Y(T_3 - T_2) \\ \frac{1}{\rho}(X - Y) & X + Z - \frac{1}{\rho}(X - Y)T_2 & -(X + Z) - \frac{1}{\rho}(X - Y)(T_3 - T_2) \end{bmatrix} T_o x \\ &\quad + \frac{1}{2}x^T T_o^T \begin{bmatrix} \dot{M} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x \end{aligned}$$

$$= x^T T_o^T \begin{bmatrix} -\mathcal{N} & O & O \\ \frac{1}{\rho}Y & X - \frac{1}{\rho}YT_2 & -X - \frac{1}{\rho}Y(T_3 - T_2) \\ \frac{1}{\rho}(X - Y) & X + Z - \frac{1}{\rho}(X - Y)T_2 & -(X + Z) - \frac{1}{\rho}(X - Y)(T_3 - T_2) \end{bmatrix} T_o x.$$

Taking into account that matrix \mathcal{N} is skew-symmetric and due to the particular structure of T_o , we obtain

$$x^T T_o^T \begin{bmatrix} -\mathcal{N} & O & O \\ O & O & O \\ O & O & O \end{bmatrix} T_o x = 0.$$

The substitution of this last equation in the preceding expression yields

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\partial V^T(x,t)}{\partial x} f &= x^T T_o^T \begin{bmatrix} O & O & O \\ \frac{1}{\rho}Y & X - \frac{1}{\rho}YT_2 & -X - \frac{1}{\rho}Y(T_3 - T_2) \\ \frac{1}{\rho}(X - Y) & X + Z - \frac{1}{\rho}(X - Y)T_2 & -(X + Z) - \frac{1}{\rho}(X - Y)(T_3 - T_2) \end{bmatrix} T_o x \\ &= x^T \begin{bmatrix} O & O & O \\ Y & X & O \\ X & 2X + Z & O \end{bmatrix} x \\ &= \frac{1}{2} x^T \left(\begin{bmatrix} O & O & O \\ Y & X & O \\ X & 2X + Z & O \end{bmatrix} + \begin{bmatrix} O & Y & X \\ O & X & 2X + Z \\ O & O & O \end{bmatrix} \right) x = \frac{1}{2} x^T \begin{bmatrix} O & Y & X \\ Y & 2X & Z + 2X \\ X & Z + 2X & O \end{bmatrix} x. \end{aligned}$$

The computation of the expression $g^T(x,t)[\partial V(x,t)/\partial x]$ can be made as follows:

$$g^T(x,t) \frac{\partial V(x,t)}{\partial x} = [M \ O \ O] (T_o^{-1})^T T_o^T \begin{bmatrix} M & O & O \\ O & Y & X - Y \\ O & X - Y & Z + Y \end{bmatrix} T_o x = [I \ O \ O] T_o x = Tx.$$

In addition, since $g(x,t) = k(x,t)$, the following equality is obtained:

$$\frac{1}{2} \frac{\partial^T V}{\partial x} \left(\frac{1}{\gamma^2} k(x,t) k^T(x,t) - g(x,t) R^{-1} g^T(x,t) \right) \frac{\partial V}{\partial x} = \frac{1}{2} x^T T^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) Tx.$$

Finally, substituting the computed expressions for the terms of the Hamilton-Jacobi equation given in Eq. (4) and with the value of $h(x) = x$:

$$\frac{1}{2}x^T \begin{bmatrix} O & Y & X \\ Y & 2X & Z + 2X \\ X & Z + 2X & O \end{bmatrix} x + \frac{1}{2}x^T T^T \left(\frac{1}{\gamma^2} I - R^{-1} \right) T x - x^T T^T R^{-1} S^T x + \frac{1}{2}x^T (Q - SR^{-1}S^T)x = 0$$

and by simplifying this expression, Eq. (18) is obtained.

ACKNOWLEDGMENTS

The authors are grateful to CICYT for funding this work under grants DPI2004-06419 and DPI2003-00429. Likewise, the authors appreciate the work by Francesco Dreoni in the realization of the real experiments.

REFERENCES

1. A. Isidori, Nonlinear control systems, third edition, Springer-Verlag, London, 1996.
2. W. Helton and M. James, Extending H_∞ control to nonlinear systems. Control of nonlinear systems to archive performance objectives, SIAM, Philadelphia, 1999.
3. A. van der Schaft, L_2 -gain and passivity techniques in nonlinear control, Springer-Verlag, New York, 2000.
4. A. van der Schaft, L_2 -gain analysis of nonlinear systems and nonlinear state feedback control, IEEE Trans. Autom. Control 37:(6) (1992), 770–784.
5. D.L. Lukes, Optimal regulation of nonlinear dynamical systems, SIAM J. Control 7:(1) (1969), 75–100.
6. M. Hardt, J.W. Helton, and K. Kreutz-Delgado, Numerical solution of nonlinear H_2 and H_∞ control problems with applications to jet engine compressors, IEEE Trans. Control Syst. Technol. 8:(1) (2000), 98–111.
7. A. Kugi and K. Schlacher, Nonlinear H_∞ -controller for a dc-to-dc power converter, IEEE Trans. Control Syst. Technol. 7:(2) (1999), 230–237.
8. W. Kang, Nonlinear H_∞ control and its applications to rigid spacecraft, IEEE Trans. Autom. Control AC-40, (1995), 1281–1285.
9. S. Li and W. Zhang, Nonlinear H_∞ control of neutralization processes, 14th World Congress of IFAC, China, vol. N (1999), 133–138.
10. B.S. Chen, T.S. Lee, and J.H. Feng, A nonlinear H_∞ control design in robotic systems under parameter perturbation and external disturbance, Int. J. Control 59, (1994), 439–462.
11. W. Feng and I. Postlethwaite, Robust nonlinear H_∞ /adaptive control of robot manipulator motion, Proc. Inst. Mech. Eng. 208 (1994), 221–230.
12. A. Astolfi and L. Lanari, Disturbance attenuation and setpoint of rigid robots via H_∞ control, Proc. 33rd Conf. on Decision and Control, Orlando, FL, pp. 2578–2583, 1994.
13. W.C.A. Maas and M. Dalsmo, Tracking of a rigid body in $SE(3)$ using singular H_∞ control, Report, University of Twente, 1995.
14. I. Postlethwaite and A. Bartoszewicz, Application of nonlinear H_∞ control to the tetrabot robot manipulator, Proc. Inst. Mech. Eng. 212:(1) (1998), 459–465.
15. H.G. Sage, M.F. De Mathelin, and E. Ostertag, Robust control of robot manipulators: a survey, Int. J. Control 72:(16) (1999), 1498–1522.
16. M.W. Spong and M. Vidyasagar, Robot dynamics and control, John Wiley & Sons, New York, 1989.
17. L. Sciavicco and B. Siciliano, Modeling and control of robot manipulators, McGraw-Hill, New York, 1996.
18. H.K. Khalil and E. Dombre, Modeling, identification and control of robots, Hermes Penton, London, 2002.
19. J.J. Craig, Introduction to robotics. Mechanics and control, 2nd ed., Addison-Wesley, New York, 1989.
20. J. Alvarez-Ramirez, I. Cervantes, and R. Kelly, PID regulation of robot manipulators: Stability and performance, Syst. Control Lett. 41:(2) (2000), 73–83.
21. M. Vidyasagar, Nonlinear systems analysis, second edition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
22. F.L. Lewis, C.T. Abdallad, and D.M. Dawson, Control of robot manipulators, MacMillan, New York, 1993.
23. System Robot, RM-10 Manuale d'Uso, Robótica Industriale, 1991.
24. Implementation Guide, Real Time Interface (RTI and RT-MP), dSPACE, 1999.
25. MOOG GmbH, DS1103 PPc Controller User Manual, 1998.
26. MOOG GmbH, T158-11 Controller User Manual, 1990.
27. C. Perez, C. Vivas, and F.R. Rubio, PID gain scheduling controller for a robot manipulator, PID'00 IFAC Workshop on Digital Control: Past, present and future of PID control, in Terrassa, Spain, pp. 585–589, 2000.