

Adaptive Stable Control of Manipulators with Improved Adaptation Transients by Using On-line Supervision of the Free-Parameters of the Adaptation Algorithm and Sampling Rate

Manuel De la SEN, Ana ALMANSA

*Instituto de Investigación y Desarrollo de Procesos, Dpto. de Ingeniería de Sistemas y Automática
Facultad de Ciencias, Universidad del País Vasco
Leioa (Bizkaia), Apto. 644 de Bilbao, Spain
e-mail: msen@we.lc.ehu.es*

Received: February 2002

Abstract. An adaptive control scheme for mechanical manipulators is proposed. The control loop essentially consists of a network for learning the robot's inverse dynamics and on-line generating the control signal. Some simulation results are provided to evaluate the design. A supervisor is used to improve the performances of the system during the adaptation transients. The supervisor exerts two supervisory actions. The first one consists basically of updating the free-design adaptive controller parameters so that the value of a quadratic loss function is maintained sufficiently small. Such a function involves past tracking errors and their predictions both on appropriate time horizons of low performances during the adaptation transients. The supervisor exerts two supervisory actions. The second supervisory action consists basically of a on-line adjustment of the sampling period within an interval centered in a nominal value of the sampling period. The sampling period is selected so that the transient of the tracking error is improved according to the simple intuitive rule of using a sampling rate faster as the tracking error changes faster.

Key words: adaptive control, adaptation transients, supervised adaptive control.

1. Introduction

The problem of designing adaptive control laws for rigid robot manipulators has interested researchers for many years. The development of effective adaptive controllers represents an important step towards high-precision robotic applications. Since robotic manipulators are inherently nonlinear systems with time-varying inertia and gravitational loads, the adaptive control approach has been proposed as a feasible technique to achieve consistent performance in the presence of configuration and payload variations.

In recent years, adaptive control results for robotic systems have included rigorous stability analysis (Craig, 1988; Slotine and Li, 1989; Li and Slotine, 1989; Sadegh and Horowitz, 1990; Feng and Ren, 1995) and the existence of globally convergent adaptive control laws has been established. On the other hand, over the last few years the possible

use of learning networks within a control systems environment has been considered by several authors (Psaltis *et al.*, 1988; Narendra and Parthasarathy, 1990; Antsaklis, 1990; Hunt *et al.*, 1992; Sadegh, 1993; Gupta and Rao, 1994). The typical encountered approach is the use of the network as a modeling technique to approximate either the plant's direct or inverse dynamics, taking advantage of the network's capability of nonlinear mapping. Concerning the application of learning controllers to control mechanical manipulators, several works have appeared in the literature, for example Miyamoto *et al.* (1988), Ozaki *et al.* (1991), Sundararajan *et al.* (1993), Teshnehlab and Watanabe (1994), Pham and Oh (1994). Many of these works use the neural network's learning capability to off-line estimate the inverse dynamics of the manipulator so that the control law can be generated.

On the other hand, it is usually a key design point obtaining a good performance when any control design is performed. In this context, it is important the achievement of good transient performances when synthesizing adaptive control laws since it is known that those designs are subject to poor performances if special tools are not involved in the design. Particular useful tools for that purpose are the on-line updating of the free parameters of the adaptation algorithm and the on-line generation of the sampling period so that the tracking error is improved during the transient (see, for instance, De la Sen, 1984a; 1986). The off-line selection of the sampling period has been proved to be also useful for the improvement of the performance in regulator-type problems (Schlueter and Levis, 1973; Barry, 1975; Glasson, 1980).

In this paper, an adaptive control scheme for mechanical manipulators is presented that takes advantage of the relationships between adaptive and neural controllers.

The control loop basically consists of a simple neural network which learns the robot's inverse dynamics, so that the control signal can be on-line generated. The synthesized controller involves the use of a supervisor to improve the transient performances since such a strategy was proved to be useful in classical problems of adaptive control to improve the adaptation transients. See De la Sen (1984a; 1985). One takes the design advantage arising from the freedom in selecting online the so-called free-design parameters of the algorithm. Such parameters of the estimation algorithm are those which can be freely selected in an admissibility domain being compatible with convergence and stability. *The proposed supervisor consists of two actions, namely:*

1. *An on-line updating procedure of one of the free-design parameters of the estimation algorithm so that the adaptation rate is governed according to an extra adaptation loop in such a way that the tracking error during the adaptation transient exhibits improved performances. Such a strategy may be interpreted as a high-level supervisor which implements a simple empirical suboptimization procedure of the prefixed admissible values in the free-design parameter of the adaptation algorithm so that a weighted generalized quadratic tracking error is minimized. An optimization horizon including a set of samples including past measurements and, eventually, tracking error predictions is considered.*
2. *The sampling period is generated from an updating sampling law within an interval centered around its nominal value being suitable for operation in each particular*

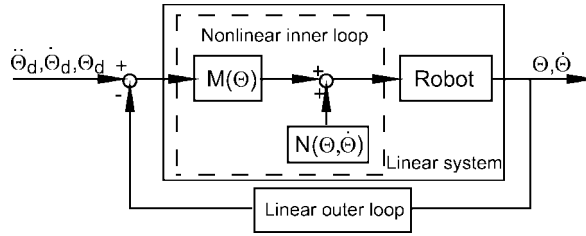


Fig. 1. Model-based feedback linearization control scheme.

application. The adaptive sampling law guarantees that sampling occurs at a faster rate as the tracking error varies rapidly with time. See, for instance, Hsia (1972; 1974) and De la Sen (1986) for a general description of the adaptive sampling problem as well as description of a general analytic method to derive wide sets of adaptive sampling laws so that the transient performances are improved compared to the usual case when constant sampling rates are used. Simulation examples show that the obtained results, without involving the use of supervisor, are comparable to those achieved using adaptive computed-torque control techniques. The use of the supervisor improves those performances during the transients. Note that the use of a supervisory scheme is linked to the potential use of a multimodel control scheme since the some parameters, like the sampling period or the adaptation gain, which are very crucial in the dynamics can be online updated by the supervisory action.

The paper is organized as follows. Section 2 formulates the problem of controlling a mechanical manipulator with unknown parameters. Sections 3 solves the controller synthesis by using an adaptive approach to the control problem. Section 4 discusses the proposed supervisor for improvement of the adaptation transients and the closed-loop stability is also discussed. Section 5 presents simulation examples to evaluate the designs and comparisons between the situations of absence and implementation of the proposed supervisor. Finally, some concluding remarks are given in Section 6.

2. Problem Formulation

The vector equation of motion of an n -link robot manipulator can be written as:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}), \quad (1)$$

where τ is an $n \times 1$ vector of joint torques; $\Theta, \dot{\Theta}$ and $\ddot{\Theta}$ are the $n \times 1$ vectors of joint positions, speed and accelerations, respectively; $M(\Theta)$ is the $n \times n$ mass matrix of the manipulator; $V(\Theta, \dot{\Theta})$ is an $n \times 1$ vector of centrifugal and Coriolis terms; $G(\Theta)$ is an $n \times 1$ vector of gravitational terms and $F(\Theta, \dot{\Theta})$ is an $n \times 1$ vector of friction terms.

The equations of motion (1) form a set of coupled nonlinear ordinary differential equations which are quite complex, even for simple manipulators. One of the most widely

used techniques to design a trajectory following control system for such a device is the so-called computed-torque control using feedback linearization (Craig, 1986). The method basically consists in introducing a nonlinear model-based feedback to compensate for all the nonlinearities present in the robot (see Fig. 1). If the model of the system is accurately known, this nonlinear inner loop decouples and linearizes the robot's dynamics in such a way that a linear outer loop can be used to efficiently control the resulting linear system to track a desired trajectory $\Theta_d, \dot{\Theta}_d, \ddot{\Theta}_d$. A frequently used computed-torque control scheme is shown in Fig. 1 where $N(\cdot, \cdot)$ is a nonlinear block which includes nonlinear effects on the plant. From the block diagram, the nonlinear, model-based control law is found to be:

$$\hat{\tau} = \hat{M}(\ddot{\Theta})\tau' + \hat{V}(\Theta, \dot{\Theta}) + \hat{G}(\Theta) + \hat{F}(\Theta, \dot{\Theta}), \quad (2)$$

where $\hat{M}(\ddot{\Theta})$, $\hat{V}(\Theta, \dot{\Theta})$, $\hat{G}(\Theta)$, $\hat{F}(\Theta, \dot{\Theta})$ are estimates of $M(\Theta)$, $V(\Theta, \dot{\Theta})$, $G(\Theta)$, $F(\Theta, \dot{\Theta})$, respectively, where $\hat{\tau}$ is identical to τ in (1) with the parameters being replaced by their estimates and τ' has been calculated as:

$$\tau' = \ddot{\Theta}_d + K_v \dot{E} + K_p E \quad (3)$$

with K_v and K_p $n \times n$ constant diagonal matrices and the servo error E defined as $E = \Theta_d - \Theta$.

From (1) and (2) and the servo error, the error torque becomes:

$$\begin{aligned} \tilde{\tau}_k &= \tau_k - \hat{\tau}_k = \tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta}) - \hat{M}(\Theta)(K_v \dot{E} + K_p E) \\ &= \tilde{M}(\Theta)\ddot{\Theta}_d + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta}) \\ &\quad + \hat{M}(\Theta) \left(\ddot{E} - \hat{M}^{-1}(\Theta) \left(\tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta}) \right) \right) \\ &= \hat{M}(\Theta)\ddot{E} + \tilde{M}(\ddot{\Theta})\ddot{\Theta}_d + \tilde{M}(\Theta)\ddot{E} - \tilde{M}(\Theta)\ddot{\Theta}_d = M\ddot{E}. \end{aligned} \quad (4)$$

Using (1–4), by calculating the torque from (2) using (3) and then substituting in the second-order differential equation obtained from (4) the closed-loop dynamics equation is found to be:

$$\ddot{E} + K_v \dot{E} + K_p E = \hat{M}^{-1}(\Theta) \left[\tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta}) \right], \quad (5)$$

where the modeling parametrical errors are

$$\begin{aligned} \tilde{M}(\Theta) &= M(\Theta) - \hat{M}(\Theta); & \tilde{V}(\Theta, \dot{\Theta}) &= V(\Theta, \dot{\Theta}) - \hat{V}(\Theta, \dot{\Theta}); \\ \tilde{G}(\Theta) &= G(\Theta) - \hat{G}(\Theta); & \tilde{F}(\Theta, \dot{\Theta}) &= F(\Theta, \dot{\Theta}) - \hat{F}(\Theta, \dot{\Theta}). \end{aligned}$$

If all the robot's parameters are perfectly known, the closed loop equation (5) takes the following linear and decoupled form since the terms in the right-hand side brackets of (5) become zero:

$$\ddot{E} + K_v \dot{E} + K_p E = 0, \quad (6)$$

so that it becomes clear that a simple suitable selection of K_p and K_v can easily regulate the evolution of the servo error. However, although some parameters of a robot are easily measurable, some other effects, such as friction, mass distribution or payload variations can not, in general, be accurately modeled, and thus the assumption of obtaining negligible modeling errors is quite unrealistic in practice. In these conditions, it looks apparent that some sort of adaptive parameter estimation mechanism should be included in the control loop, so that equation (5) became approximately linear and uncoupled and the servo errors could be asymptotically eliminated.

3. Adaptive Control Scheme

The equations of motion (1), although quite complex and nonlinear in general, can be expressed in a linear in the parameters form, since all the potentially unknown parameters (link masses, lengths, friction coefficients, etc.) appear as coefficients of known functions of the generalized coordinates. In an adaptive control system design context, one usually takes the advantage of the above property of linearity in the parameters by rewriting (1) as:

$$M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}) = W(\Theta, \dot{\Theta}, \ddot{\Theta})P, \quad (7)$$

where P is an $r \times 1$ vector containing the robot's unknown parameters and $W(\Theta, \dot{\Theta}, \ddot{\Theta})$ is an $n \times r$ matrix of known nonlinear functions, often referred to as regression matrix. In the same way, the $r \times 1$ estimated parameters vector \hat{P} fulfil:

$$\hat{M}(\Theta)\ddot{\Theta} + \hat{V}(\Theta, \dot{\Theta}) + \hat{G}(\Theta) + \hat{F}(\Theta, \dot{\Theta}) = W(\Theta, \dot{\Theta}, \ddot{\Theta})\hat{P}, \quad (8)$$

and thus:

$$\tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta}) = W(\Theta, \dot{\Theta}, \ddot{\Theta})\tilde{P}, \quad (9)$$

where the parameter estimation error \tilde{P} has been defined as $\tilde{P} = P - \hat{P}$. Figs. 2 and 3 show the adaptive control scheme. The design is a neural extension of the computed-torque control strategy. A two-layered learning network with $n \times r$ inputs and n outputs is used to learn the manipulator's inverse dynamics, so that the control law can be on-line generated. The network's inputs are known nonlinear functions of the system response (more concretely, the elements w_{ij} of the regression matrix $W(\Theta, \dot{\Theta}, \ddot{\Theta})$ are defined in (7)), while its outputs are estimates of the input torques to the robot:

$$\hat{\tau}(t) = \hat{\tau}_k = \sum_{i=1}^n \sum_{j=1}^r w_{ij}(\Theta, \dot{\Theta}, \ddot{\Theta})\hat{p}_{jk} \quad \text{for } t \in [t_k, t_{k+1}), \quad (10)$$

which is a piecewise constant signal from the zero-order sampling and hold device. Defining the connection weights vector and the estimated torques vector as

$$\hat{P} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_r]^T; \quad \hat{\tau} = [\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_n]^T, \quad (11)$$

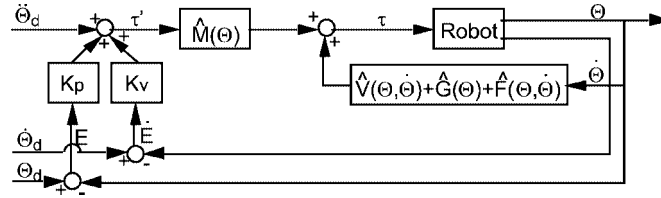


Fig. 2. Computed-torque control scheme.

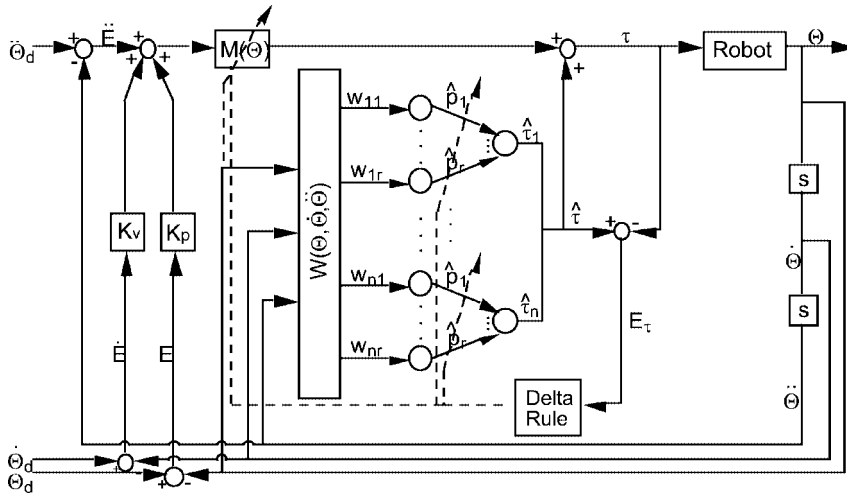


Fig. 3. Proposed neural control scheme.

(10) can also be expressed in a familiar matrix form:

$$\hat{\tau}_k(t) = \hat{\tau}_k = W(\Theta, \dot{\Theta}, \ddot{\Theta})\hat{P}_k \quad \text{for } t \in [t_k, t_{k+1}), \tag{12}$$

which is formally the same as (8) where \hat{P}_k is a parameter vector which is estimated in a discrete-time way, i.e., it is only updated at sampling instants by the adaptation algorithm. The inverse dynamics is learned by measuring the input and output signals in the robot and then adjusting the connection weights vector at each discrete instant kT_i (with T_i being the i -th sampling period) by using an extension of the well-known Widrow–Hoff delta rule (Widrow and Lehr, 1990):

$$\hat{P}_{k+1} = \hat{P}_k + \frac{F_k W_k^T E_{\tau k}}{c_k + \|W_k F_k W_k^T\|}, \tag{13a}$$

$$F_{k+1} = \frac{1}{\lambda_k} \left(F_k - \frac{F_k W_k^T W_k F_k}{c_k + \|W_k F_k W_k^T\|} \right), \tag{13b}$$

where E_{τ_k} is the prediction error vector, defined as $E_{\tau_k} = -\tilde{\tau}_k = \hat{\tau}_k - \tau_k$, $W_k^T(\cdot)$ is the regression matrix used for updating the parameters, and F_k is an adaptation gain matrix which satisfies $F_k = F_k^T$ and is positive definite for all samples $k \geq 0$. Both symmetry and positive definiteness are achieved automatically for all sample by initializing the algorithm with $F_0 = F_0^T > 0$. The parameter sequence $\lambda_k \in (0, 1)$ is the forgetting factor used to update the adaptation gain matrix and $c_k \in (0, \infty)$ is a scalar for all $k \geq 0$. Both free parameters of the algorithm have to satisfy the given stability constraints in order to achieve closed-loop stability. The matrix sequence obtained from (13.b) is positive definite (at the limit it can become semidefinite) and time-decreasing. The norms taken in (13) are the Euclidean norms. The above approach is used in the simulated example given in Section 5.3 to evaluate the supervision efficiency.

If the manipulator's inverse dynamics is correctly learned by the neural network, both nonlinear dynamics cancel each other according to the block diagram shown in Fig. 3. Thus, the closed-loop system turns linear and the closed-loop tracking properties are adjusted with a suitable selection of the proportional and derivative gain matrices K_p , K_v . This is the same effect obtained using the conventional adaptive control approach described in the previous section.

Remark 1. The given neural control scheme uses an extremely simple linear neural network to learn the manipulator's nonlinear inverse dynamics. This can be accomplished by using the nonlinear prefilter W at the network's input, so that the composite device (nonlinear filter plus linear network) is able to map a nonlinear function. This partitioned design greatly simplifies the analysis of the overall control scheme, since it formally reduces to a conventional adaptive control system. However, it seems clear that, in general, nonlinear neural networks should be employed to efficiently learn a nonlinear dynamical behavior.

4. Design of a Supervisor for Improvement of the Transient Performances

4.1. Heuristic Motivation

Note, by inspection, that the learning rule (or adaptation algorithm) Eqs. (13) has an adaptation rate highly dependent on the size of the c_k -updating parameter which is a free-design parameter provided that it is positive and bounded. Assume that the gain λ_k is constant. The adaptation rate is very low when the c_k -sequence takes very large values compared to the square of the regressor norm. The adaptation rate increases as such a sequence takes smaller values compared to the square of the regression matrix norm. *Thus, a good solution for improvement of the transients will be the on-line updating of such a sequence to govern the rate of the parametrical adaptation.* A similar reasoning applies when using γ_k as free-design on-line updated parameter within the interval $(0, 2)$. A second action of the supervisor as it was commented in the introductory section is concerned with the on-line choice of the sampling period within an interval centered around a nominal sampling period. The boundary of the variation domain of the sampling

period is established according to ‘a priori’ knowledge about guaranteeing closed-loop stability and a prefixed bandwidth. Other considerations as, for instance, the upper limit of the sampling rate or the achievable performance of the application at hand (Hsia, 1972; 1974; De la Sen, 1984b) have to be also taken into account. The overall supervisor is designed for:

- an on-line calculation of a free parameter of the adaptation algorithm;
- a calculation of a time-varying sampling period dependent on the time variation of the tracking error.

It is based upon three main heuristic rules, namely:

Rule 1. *If the tracking error is increasing with respect to preceding samples then decrease (increase) the last value of the sequence of the free-design parameter provided that the previous action (i.e., the kind of supervisory action at the preceding sample) was to increase (decrease) the value of such a sequence.*

In other words, if the tracking performance is deteriorating then correct the supervision philosophy of the last action exerted on the value of the free parameter of the algorithm.

Rule 2. *If the tracking error is decreasing with respect to preceding samples then decrease (increase) the last value of the sequence of the free design parameter provided that the previous action was to decrease (increase) it. [In other words, if the tracking performance is being improved then do not modify the last action exerted on the value of the free parameter].*

Rule 3. *Compute a time-varying updated sampling period as being inversely proportional to the absolute value of the tracking error time-derivative within a predefined interval $[T_{\min}, T_{\max}] = [T_0 - \Delta T, T_0 + \Delta T]$ centered around a prefixed nominal sampling period T_0 .*

Thus, the sampling period decreases as the tracking error absolute value grows faster and vice versa, (Hsia, 1972; 1974; De la Sen, 1986). In Rule 3, $\frac{\Delta T}{T_0}$ has to be small since the discrete parametrization of the plant is time-varying under time-varying sampling (De la Sen, 1984b; Fuster and Guillen, 1987). Thus, small variations of the sampling period lead to small deviations of the plant parametrization from time-invariance and the estimation algorithm (13) is still valid, in practical situations. This is the philosophy used in this paper to design the admissibility domain of the sampling period. Another useful variation would be, for instance, to use a time-varying whose length decreases asymptotically converging to the nominal sampling period; i.e., $[T_{\min}(k), T_{\max}(k)]$ is designed so that $T_{\min}(k) \rightarrow T_0$ and $T_{\max}(k) \rightarrow T_0$ as $k \rightarrow \infty$. On the other hand, Rule 3 can also be modified by involving higher-order time-derivatives of the tracking error (Hsia, 1974; De la Sen, 1986).

The above three actions are completed with the following design philosophies:

- (1) The sizes of the modifications in the successive values of the free-design parameter sequence (i.e., the c_k -sequence) of the parameter adaptive algorithm have to be

related to the ‘amplitudes’ of the improvement or deterioration of the transient performances within the stability constraints for c_k in (13) of its values being positive and bounded.

- (2) It is better to analyze the transient tracking errors over a finite horizon of preceding samples and, eventually, also on a finite horizon of its future predictions over each current sample in order to include both a correcting and a predictive-correction effects of registered tracking errors to calculate the current value of the sequence of free-design parameters. The use of a unique sample in the supervisory loop would lead, in general, to unsuitable actions when measurement failures arise or when abrupt changes in the control input appear at isolated sampling instants.

4.2. Supervisory Action on a Free-Design Parameter of the Adaptation Algorithm

Define the loss function $J_k^\varepsilon = \sum_{i=k-N_1}^{k+N_2} \sigma^{|k-i|} \mathbf{E}_i^T Q_i \mathbf{E}_i$, for each current k -th sample as supervisory criterion where $\mathbf{E}(\cdot) = [E_{1(\cdot)}, E_{2(\cdot)}, E_{3(\cdot)}]^T$ is the tracking error vector, N_1 and N_2 are, respectively, the sizes of the correction and prediction horizons $[k - N_1, k]$ and $[k, k + N_2]$ associated with the current k -th sample, $Q_{(\cdot)}$ is a (at least) positive semidefinite weighting matrix and $0 < \sigma \leq 1$ is the forgetting factor of the loss function. Note that \mathbf{E}_j for $j > k$ are predicted tracking errors in the loss function for each k -th sample. In this paper, the free design parameter in (13) is c_k which has to belong to an admissibility interval compatible with the stability constraint, i.e., it has to be positive and bounded. The horizon sizes, weighting matrix and forgetting factor of the loss function are chosen by the designer according to the next design criteria:

- (a) *How relatively important each robot articulation is compared to the remaining ones.* This idea is relevant to the choice of the $Q_{(\cdot)}$ -matrix. In Fig. 4, the third articulation could be considered more important, if suited, since it has to follow a reference related to the final trajectory for each specific application. If the matrix is chosen as diagonal with positive identical diagonal entries then all the articulations are considered equally relevant and then all the tracking error components are introduced with identical weights in the supervisory loss function.
- (b) *The relative weight in the loss function given to the more recent measured errors and their next immediate predictions compared to the older ones and subsequent future ones, respectively.* This idea is relevant to the choice of σ .
- (c) *The relative weight in the loss function given to the past tracking errors (correction horizon) compared to the predicted errors (prediction horizon).*

The supervisory action for c_k is described in the following algorithm.

Supervisor of c_k

Step 0. Define $[c_{\min}, c_{\max}]$ with $c_{\min} > 0, c_{\max} > c_{\min} > 0$ as the admissibility domain for the free parameter c_k of the adaptation algorithm (13). **Define** also the loss function \mathbf{J} according to the above supervisory design criteria (a) to (c). Initialize $k \leftarrow 0$.

Step 1. For each current k -th sampling instant, **make** $\bar{c}_k = \rho_k \|W_k F_k W_k^T\| + \bar{c}$

$$c_k = \begin{cases} c_{\min} & \text{if } \bar{c}_k \leq c_{\min}, \\ \bar{c}_k & \text{if } \bar{c}_k \in (c_{\min}, c_{\max}), \\ c_{\max} & \text{if } \bar{c}_k \geq c_{\max}, \end{cases}$$

$$\rho_k = \rho_{k-1} + \text{Min} \left(g_k, \text{Int. part} \left[\frac{J_k - J_{k-1}}{J_{k-1}} \right] \right) \cdot \text{Sign} [\rho_{k-2} - \rho_{k-1}] \cdot \Delta\rho,$$

$$g_k = \frac{K - \bar{c}}{\|W_k F_k W_k^T\| \Delta\rho},$$

with $\text{sign}(0) = 0$.

Step 2. **Apply** the parameter-estimation algorithm of (13) and **generate** the torque (2).

Step 3. $k \leftarrow k + 1$ and **go to** Step 1.

Remark 2. Note that Step 1 follows the heuristic rules given in Section 4.1 since ρ_k and then c_k increase when J_k increases (decreases) with respect to its preceding value provided that c_{k-1} decreased (increased) with respect to its preceding value. If ρ_k decreases (increases) when J_k decreases (increases) then c_k is decreased. The choice of according to a proportional factor of the regression matrix square norm is proposed to follow such empirical design guidelines. Thus, if the loss function value increases then the action on the free-design parameter should be changed of sign with respect to the previous one. If the loss function value decreases the supervisory policy has to be kept. The saturation g_k for the modification of ρ_k in Step 1 guarantees that c_k is upper-bounded by a predefined positive design constant K . The small positive constant \bar{c} is used to avoid division by zero in the parameter estimation (13) when the measurement regressor is zero. The supervisory learning rule also ensures, apart from the above mentioned saturation, that the eventual corrections on the choice of the parameters increase as the efficiency deteriorates. Such an efficiency is given by the loss function and it decreases with the sizes of those positive or negative increments in ρ_k being given by the relative variation of the loss function between two consecutive sampling instants.

Remark 3. The use of the above supervisory action maintains the stability of the adaptive closed-loop scheme since the admissible variation domains of the $g_{(\cdot)}$ and $c_{(\cdot)}$ sequences of free-design parameters are maintained.

4.3. Error Prediction

The measurements of the loss function in the prediction horizon are calculated by simple extrapolations of preceding predictions or real measurements by using a Taylor series expansion approximating the derivatives by finite differences using sampled values according to:

$$f_{k+1} \cong \sum_{i=0}^{\infty} \frac{T^i f_k^{(i)}}{i!} = f_k + (f_k - f_{k-1}) + \frac{1}{2}(f_k - f_{k-1} + f_{k+2}),$$

with T being the sampling period for any signal $f(t)$ and the i -th derivative $f_k^{(i)}$ being defined recursively from $f_k^{(1)} = \frac{f_k - f_{k-1}}{T}$ for $i \geq 1$. Note that even in the case when the predictions are very rough, this is not very relevant for the supervisory algorithm efficiency because of the saturation effect included in Step 1 which guarantees that the obtained value for the current sample of the free-design updated parameter is upper-bounded by a prefixed bound K . In this context, *it is suitable to have acceptable predictions of the signs of the next tracking errors for each current sample rather than good estimations of their real values*. A simple estimation procedure as the proposed one can be sufficient as shown through simulations in Section 5.3. It is also important to find an efficient balance between the sizes of the correction and prediction horizons and, for such a purpose, it becomes apparent by simple empirical considerations that *the size of the prediction horizon should be not larger than that of the correction horizon* which operates with real previously registered measurements. Numerical experimentation involving different sizes of the correction and prediction horizons will help the designer in the choice of their more convenient values.

4.4. *Supervisory Action on the Sampling Rate (i.e., on-line updating rule for the time-varying sampling period T_k)*

The sequence of sampling instants $\{t_k, k \geq 0\}$ is generated as $t_{k+1} = t_k + T_k$ with $t_0 = 0$ and T_k being the sampling period after the i -th sampling instant. The following adaptive sampling law function is used for on-line adjustment of the sampling period T_k :

$$T_k = \begin{cases} T_{\min} & \text{if } \bar{T}_k \leq T_{\min}, \\ \bar{T}_k & \text{if } T \in [T_{\min}, T_{\max}], \\ T_{\max} & \text{if } \bar{T}_k \geq T_{\max}, \end{cases} \tag{14}$$

where $\bar{T}_k = \frac{CT_{k-1}}{|\bar{\varepsilon}(t_k)|}$ and $C > 0$ is an arbitrary real constant. The above adaptive sampling law is a particular case of that general one proposed in Hsia (1974) and then used De la Sen (1986). Such a general law was obtained analytically from a penalty function which has two additive terms. The first term penalized the deviation of the tracking error from its last sampled value within a sampling interval and the second term penalized the sampling action itself. The admissibility interval $[T_{\min}, T_{\max}]$ of the sampling period is selected according to considerations of stability, bandwidth and the requirements on performance of each particular application (De la Sen, 1986). The above sampling law is tested in the simulations to evaluate the performance improvement of the sampling rate updating for the transient adaptation. The above sampling law as well as other five updating sampling laws listed below are then comparatively tested in the simulations to evaluate the various improvements caused by the sampling rate adaptation over the basic free-parameter adaptation. The tracking error derivatives are approximated by finite differences at sampling instants in order to get an easier implementation. The sampling laws take the following specific structures:

Law 1:
$$\bar{T}_k = \frac{T_{\max} T_{k-1}^2}{C|\varepsilon_k - \varepsilon_{k-1}|^2 + T_{k-1}^2},$$

$$\begin{aligned} \text{Law 2: } \bar{T}_k &= \frac{CT_{k-1}}{|\varepsilon_k - \varepsilon_{k-1}|}, \\ \text{Law 3: } \bar{T}_k &= \frac{CT_{k-1}^{2/3}}{|\varepsilon_k - \varepsilon_{k-1}|^{2/3}}, \\ \text{Law 4: } \bar{T}_k &= T_{\max} - \frac{C|\varepsilon_k - \varepsilon_{k-1}|}{T_{k-1}}, \\ \text{Law 5: } \bar{T}_k &= \frac{T_{\max}T_{k-1}}{C|\varepsilon_k - \varepsilon_{k-1}| + T_{k-1}}, \\ \text{Law 6: } \bar{T}_k &= \sqrt{\frac{CT_{k-1}}{|\varepsilon_k - \varepsilon_{k-1}|}}, \end{aligned}$$

where the tracking error used for updating is taken for the third articulation.

Remark 4. The supervisory technique can be also applied to the forgetting factor by making it time-varying so that $\lambda_k \in (0, 1]$ to ensure closed-loop stability of the adaptive scheme. A useful technique is to modify the Supervision Algorithm of Section 4.2 to on-line estimate the forgetting factor which has to belong to the admissibility domain $[\lambda_{\min}, \lambda_{\max}] = [\delta, 1 - \delta]$ for some constant $\delta \in (0, 1)$ with the change $\bar{c} \rightarrow \bar{\lambda} > 0$ and Step 2 is modified with the replacement $\bar{c}_k \rightarrow \bar{\lambda}_k = \rho_k \lambda_{k-1} + \bar{\lambda}$ with ρ_k being computed as above and

$$\lambda_k = \begin{cases} \lambda_{\min} & \text{if } \bar{\lambda}_k \leq \lambda_{\min}, \\ \bar{\lambda}_k & \text{if } \bar{\lambda}_k \in (\lambda_{\min}, \lambda_{\max}), \\ \lambda_{\max} & \text{if } \bar{\lambda}_k \geq \lambda_{\max}. \end{cases}$$

Subsequently, the free parameter of the adaptive algorithm c_k is chosen according to the rule $c_k = \frac{\lambda_k \|W_k F_k W_k^T\|}{T_r(1-\lambda_k-\delta)}$ where $T_r \geq \text{Trace}(F_0) \geq \text{Trace}(F_k) > 0$. Such a rule ensures that the trace of the adaptation matrix remains upper-bounded by a prefixed finite bound T_r for all time in spite of the fact that the adaptation gain matrix is not necessarily time-decreasing.

4.5. Closed-Loop Stability

The following result proves that both the basic (supervision-free) system and the supervised ones are stable.

Theorem 1 (Stability results). *The following two items hold:*

(i) *In the absence of supervision, the estimated parameters are bounded if their initial conditions are bounded and the initial adaptation covariance matrix is positive definite. Also, the closed-loop system is globally Lyapunov's stable so that the output, input, estimation error and tracking error are all bounded provided that the reference trajectory is bounded.*

(ii) *If only the algorithm free-parameter c_k (or, alternatively, the forgetting factor) is supervised by the given rule while respecting its positivity and boundedness (while belonging to the range $(0, 1]$) for all sample then (i) holds. If the sampling period is supervised (with the free-parameter being supervised or not) during a finite time interval within its admissibility domain then the (i) still holds.*

Sketch of Proof. (i) Direct calculations with (2), (3), (8), (9) and (13) yield for all sampling instant:

$$E_{\tau k} = M_{1k} \hat{P}_k + \hat{M}_k \ddot{\Theta}_k = M_k \ddot{\Theta}_k + E'_{\tau k} - M_{1k} \tilde{P}_k, \quad (15a)$$

$$E'_{\tau k} = W_k \tilde{P}_k = M_{1k} - M_k \ddot{\Theta}_k = M_k (K_v \ddot{\Theta}_k - K_p \ddot{\Theta}_l), \quad (15b)$$

where $\tilde{P}_k = P - \hat{P}_k$ is the parametrical error for the auxiliary parameter vector P . Thus,

$$E_{\tau k} = (W_k - M_{1k}) \tilde{P}_k + M_k \ddot{\Theta}_k.$$

On the other hand, one gets from (13) and the above error expression:

$$\lambda_k F_{k+1} F_k^{-1} = \left(I - \frac{F_k W_k^T W_k}{c_k + \|W_k F_k W_k^T\|} \right); \quad \lambda_k F_k^{-1} \tilde{P}_k = F_{k+1}^{-1} \tilde{P}_{k+1}.$$

If the Lyapunov's-like sequence $V_k = \tilde{P}_k^T F_k^{-1} \tilde{P}_k$ is defined then it follows that $V_{k+1} \leq \lambda_k V_k \leq V_k \leq V_0$ since

$$V_{k+1} - \lambda_k V_k = - \frac{\lambda_k}{c_k + \|W_k F_k W_k^T\|} \tilde{P}_k^T W_k^T W_k \tilde{P}_k \leq 0$$

for $c_k \in (0, \infty)$ and $\lambda_k \in (0, 1]$, all integer $k \geq 0$ with

$$\tilde{P}_{k+1} - \tilde{P}_k = \left(I - \frac{F_k W_k^T W_k}{c_k + \|W_k F_k W_k^T\|} \right) \tilde{P}_k - \tilde{P}_k = - \frac{F_k W_k^T W_k \tilde{P}_k}{c_k + \|W_k F_k W_k^T\|}.$$

Since the sequence $\{V_k\}_0^\infty$ is nonnegative and bounded for V_0 bounded and (non-strictly) monotonically decreasing then it has a finite nonnegative limit so that

$$\infty > V_0 \geq V_k \geq \lambda_{\max}(F_k^{-1}) \|\tilde{P}_k\|_E^2.$$

This implies that the parameter error \tilde{P}_k and its associate estimate are bounded for all sample since the above maximum eigenvalue of the covariance inverse is always strictly positive. As a result, all the estimates of the direct parameters used in the calculations in (2) and (5) are bounded. If the regressor is bounded then $E_{\tau k}$ and the auxiliary one $E'_{\tau k}$ are also bounded from the initial identities of this proof and then the estimated and error torques $\hat{\tau}_k$ and $\tilde{\tau}_k$ are bounded and $W_k \tilde{P}_k$ converges asymptotically to zero. It follows that the output and the tracking error are bounded, see (15), if the reference is bounded. Finally, if the regressor fulfils a standard type of asymptotic persistent excitation condition then the parametrical error converges asymptotically to zero. This proves (i). The proof of (ii) follows in the same way since the free parameters of the basic estimation scheme always belong to their admissibility domains compatible with stability if the supervisor scheme for any of the free-parameters is in operation. Finally, assume that the sampling period is on-line updated within its admissibility domain during a finite time

interval and then it is fixed to a constant value within such an interval. Thus, the overall system becomes time-invariant after a finite time which may be set as initial time for analysis and the above results still hold.

Remark 5. Note that the stability is also ensured if the time-varying sampling period tends exponentially to any constant value within its admissibility domain (a particular situation is when such a limit is its nominal value). This property may be proved by extending directly Theorem 1 (ii) by adding to the identification and parametrical error bounded and exponentially decaying terms. The key point ensuring that the closed-loop stability holds under supervision is that the free-parameter of the parameter-adaptive algorithm and the sampling period are kept within their admissible domains. Those domains are compatible with convergence of the updating algorithm and stability.

5. Simulation Examples

In this section we will consider the control of the simple planar mechanical manipulator with three revolute joints shown in Fig. 4. For simplicity, it will be assumed that the masses m_1 and m_2 of elements 1 and 2 are concentrated at the distal end of each link, while mass m_3 is distributed according to a diagonal inertia tensor

$$I = \text{Block Diagonal} [I_{xx}, I_{yy}, I_{zz}]. \quad (16)$$

Moreover, we assume that the center of mass of link 3 is located at the proximal end of the link, that is, it coincides with the center of mass of m_2 .

It is worth to note that although the mechanical manipulator being considered here is very simple, it is complex enough to illustrate nearly all the principles of general manipulators. In the same way that a typical robot with six degrees of freedom can reach arbitrary positions and orientations in the space, the simplified planar manipulator with three degrees of freedom can reach arbitrary positions and orientations in the plane. The dynamics in three dimensions of the robot is compacted in a matrix form notation and

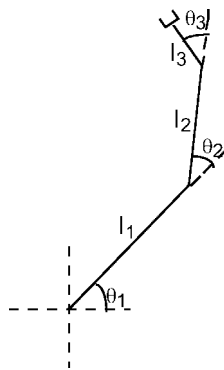


Fig. 4. The simple planar manipulator with three revolute joints.

it is calculated from the mechanical equations involving mass, inertia moments, frictions and applied external forces and torques in a standard way (see Craig, 1986; 1988). The elements of the dynamic equation (1) for this robot are found to be:

$$M(\Theta) = \begin{bmatrix} I_{zz} + m_1 l_1^2 + (m_2 + m_3)[l_1^2 + l_2^2 + 2l_1 l_2 c_2] \\ I_{zz} + (m_2 + m_3)[l_2 + l_1 l_2 c_2] \\ I_{zz} \\ I_{zz} + (m_2 + m_3)[l_2 + l_1 l_2 c_2] & I_{zz} \\ I_{zz} + (m_2 + m_3)l_2 & I_{zz} \\ I_{zz} & I_{zz} \end{bmatrix}, \quad (17a)$$

$$V(\Theta, \dot{\Theta}) = \begin{bmatrix} -(m_2 + m_3) l_1 l_2 s_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \\ (m_2 + m_3) l_1 l_2 s_2 \dot{\theta}_1^2 \\ 0 \end{bmatrix}, \quad (17b)$$

$$G(\Theta) = \begin{bmatrix} (m_1 + m_2 + m_3)gl_1 c_1 + (m_2 + m_3)gl_2 c_{12} \\ (m_2 + m_3)gl_2 c_{12} \\ 0 \end{bmatrix}, \quad (17c)$$

where c_i , s_i represent $\cos \theta_i$ and $\sin \theta_i$, respectively, and c_{12} represents $\cos \theta_1 + \theta_2$. Concerning the friction terms, a combination of viscous and Coulomb friction is assumed:

$$F(\Theta, \dot{\Theta}) = \begin{bmatrix} v_1 \dot{\theta}_1 + k_1 \text{sgn}(\dot{\theta}_1) \\ v_2 \dot{\theta}_2 + k_2 \text{sgn}(\dot{\theta}_2) \\ v_3 \dot{\theta}_3 + k_3 \text{sgn}(\dot{\theta}_3) \end{bmatrix}, \quad (17d)$$

where v_i and k_i are the viscous and Coulomb friction coefficients, respectively. In all the subsequent examples the following values for the robot's parameters will be assumed (SI units):

$$\begin{aligned} m_1 = 4.6; \quad m_2 = 2.3; \quad m_3 = 1.0; \quad I_{zz} = 0.1; \\ l_1 = l_2 = 0.5; \quad v_1 = v_2 = v_3 = 0.5; \quad k_1 = k_2 = k_3 = 0.5. \end{aligned} \quad (18)$$

5.1. Supervision of the Free Parameter c_k

In this second example will apply the neural control structure proposed in Section 3 to control the same mechanical manipulator of the previous example. In this case the robot starts at position $(\theta_1, \theta_2, \theta_3) = (0^\circ, 30^\circ, 20^\circ)$ and the control objective is to reach the position $(\theta_1, \theta_2, \theta_3) = (10^\circ, -50^\circ, -20^\circ)$ in 0.5 seconds, following again smooth cubic trajectories, given now by:

$$\theta_1(t) = 10 + 0.12t^2 - 0.016t^3, \quad t \leq 0.5; \quad \theta_1(t) = 10, \quad t \geq 0.5, \quad (19a)$$

$$\theta_2(t) = -50 - 0.12t^2 + 0.016t^3, \quad t \leq 0.5; \quad \theta_2(t) = -50, \quad t \geq 0.5, \quad (19b)$$

$$\theta_3(t) = -20 + 0.72t^2 - 0.096t^3, \quad t \leq 0.5; \quad \theta_3(t) = -20, \quad t \geq 0.5. \quad (19c)$$

In order to compare both control methods in similar conditions, it will again be assumed that the link masses m_1, m_2, m_3 , as well as the mass distribution I_{zz} and the viscous friction coefficient v_1 of the manipulator are unknown. Thus, a two-layered neural network with 12 inputs and 3 outputs will be used to learn the robot's inverse dynamics and to on-line generate the control signal. In particular, the values $\lambda_k = 1$ in (13b), $\bar{c} = 5$, $c_{\min} = 5$, $c_{\max} = 5 \cdot 10^7$ and $c_0 = 5 \cdot 10^6$, have been chosen in the learning rule (13). In the absence of supervisor, $c_k = c_0 = 5 \cdot 10^6$, for all $k \geq 0$. The loss function for the supervisor of the free algorithm parameter is $N_1 = 3$ (Correction Horizon Size) and $N_2 = 1$ (Prediction Horizon Size). The network's connection weights have been initialized without using a priori information as $\hat{p}_{10} = 10$, $\hat{p}_{20} = 5$, $\hat{p}_{30} = 1$, $\hat{p}_{40} = 1$. The values for the proportional and derivative gain matrices $K_p = K_v = \text{Diag}(100, 100, 100)$ have been employed in the outer control loop.

Fig. 5 displays the robot's second articulation time response for unsupervised free parameter which is kept constant to c_0 and supervised situations. The supervised design yields a much better transient response which is practically close to the reference signal. Figs. 6–7 show the torques for the first and second articulations, respectively and Fig. 8 shows the time evolution of the supervised algorithm free parameter c_k as well as that of $\|W_k F_k W_k^T\|$ within the time interval [50, 65]. Finally, Fig. 9 shows the evolution of the estimate \hat{m}_1 from the initial condition $\hat{m}_1(0) = 9.2$. It can be observed that the neural network is able to learn the robot's inverse dynamics quite well. As a result, the position errors are quickly eliminated and the control objective is achieved. From the given examples, it can be concluded that this neural approach to the control of mechanical manipulators lead to results comparable to those obtained using conventional adaptive control designs.

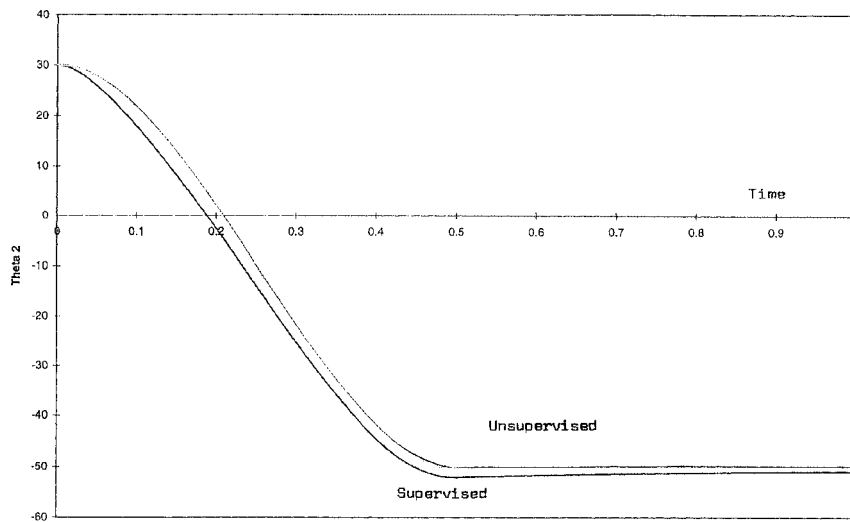


Fig. 5. Angle of the Second Articulation θ_2 versus time (seconds) with and without supervisory action in the algorithm free parameter c_k . Correction and Prediction Horizon Sizes $N_1 = 5$ and $N_2 = 2$.

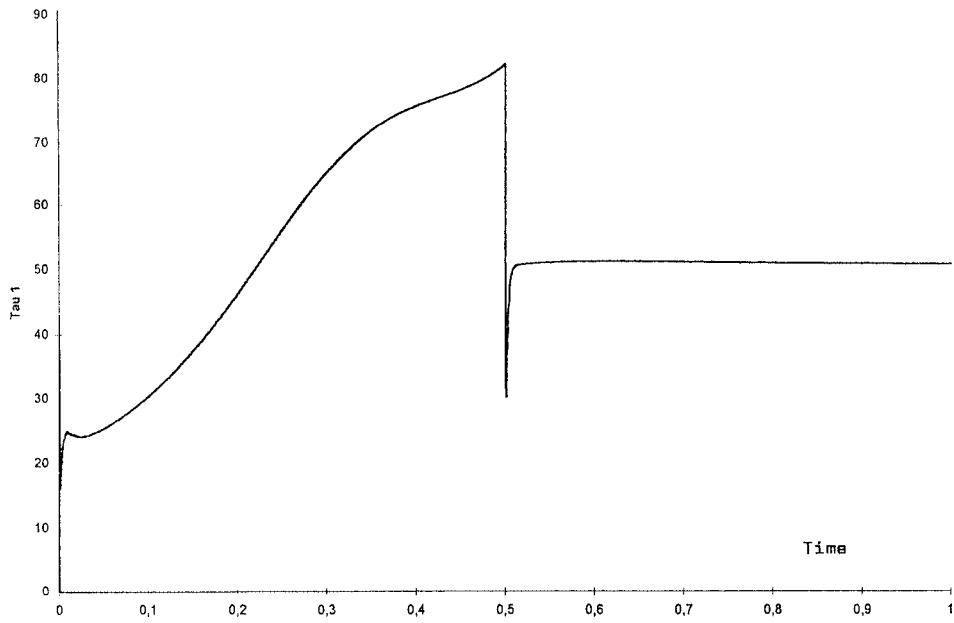


Fig. 6. Torque of the First Articulation τ_1 versus time (seconds) with supervisory action in the algorithm free parameter c_k under the Conditions of Fig. 5.

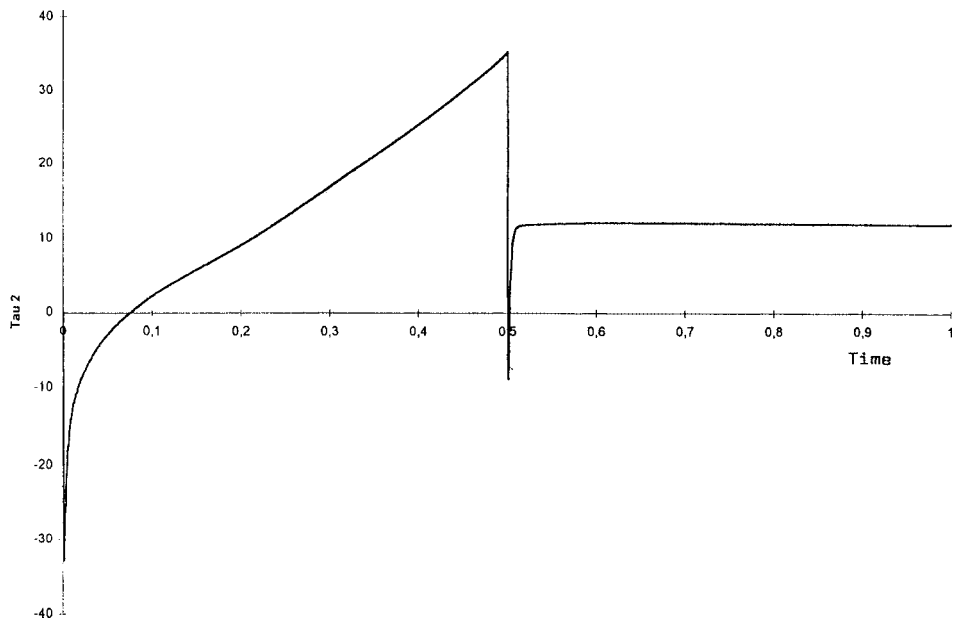
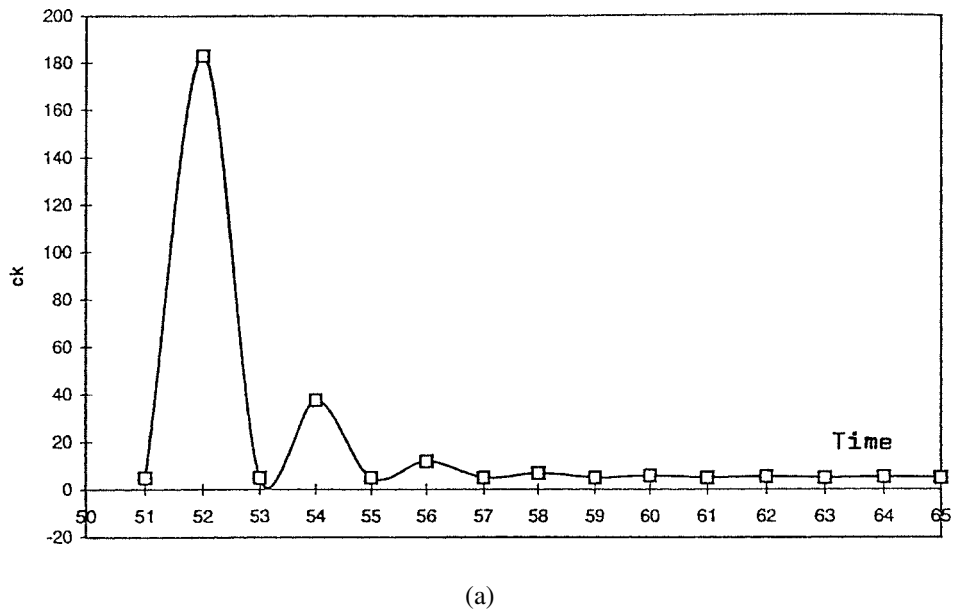
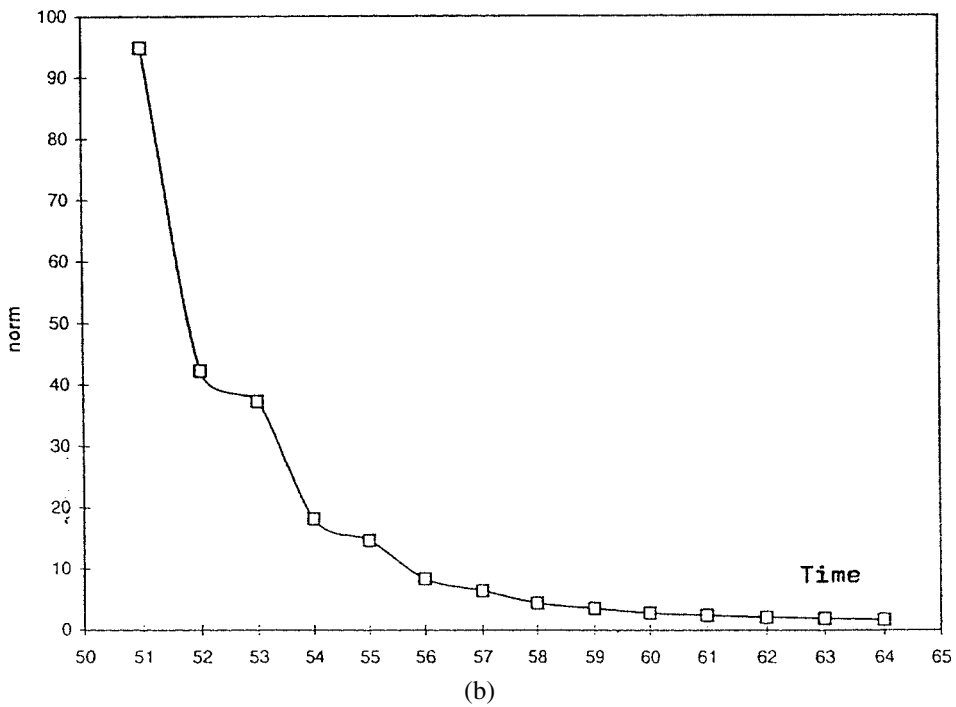


Fig. 7. Torque of the Second Articulation τ_2 versus time (seconds) with supervisory action in the algorithm free parameter c_k under the Conditions of Fig. 5.



(a)



(b)

Fig. 8. (a) Evolution versus time (seconds) of the free parameter c_k from time 50 secs to time 65 secs with supervision under the Conditions of Fig. 5. (b) Evolution versus time (seconds) of $\|W_k F_k W_k^T\|$ from time 50 secs to time 65 secs with supervision under the Conditions of Fig. 5.

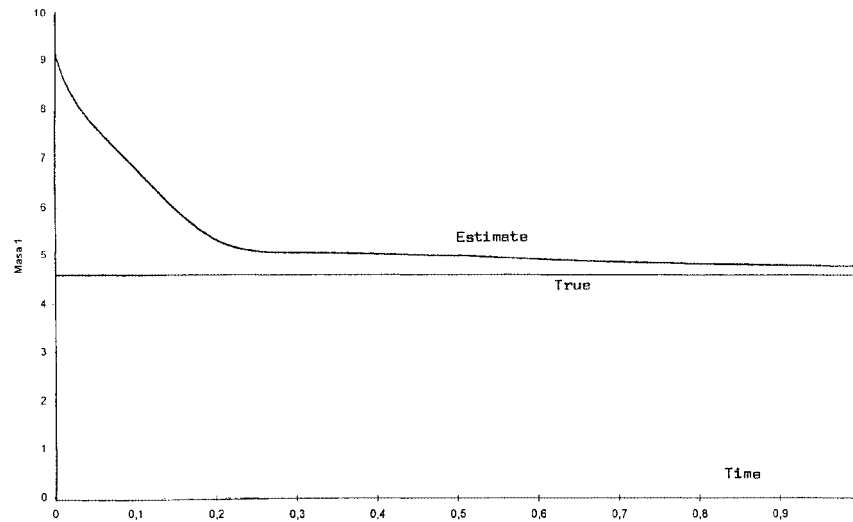


Fig. 9. \hat{m}_1 versus time (seconds) and m_1 .

5.2. Combined Supervisory of the Free Algorithm Parameter and the Sampling Period

Now, the efficiency of the use/non use of the supervisory control of Section 4 is discussed. The closed-loop stability without or with supervisory loop is kept since all the free-parameters are always kept within their admissible domains guaranteeing stability. The parameters of the robot, K_p and K_v are now chosen as in Section 5. 1 with the same units. The initial and final positions for the robot links are chosen as in the example of Section 5.2 with the same smooth cubic trajectories defined in (19). The (unknown) parameters are defined as in the above example and estimated with initial conditions [6, 4, 0.2, 0.8]. The adaptation gain matrix is initialized to $\text{Diag}[10^4, 10^4, 10^4, 10^4]$. The parameters of the supervisor are re-updated online as follows.

Supervision of the free parameter c_k of the algorithm:

The weighting matrix of the loss function J^ε is $Q_{(\cdot)} = \text{Diag}[0.2, 0.2, 0.2]$ for the samples of the prediction horizon and $Q_{(\cdot)} = \text{Diag}[0.9, 0.9, 0.9]$ for those of the correction horizon; $\sigma = 0.5$; $K = 10$ (Step 2 of the Supervisory Algorithm of Section 4.2); $\bar{c} = 5$, $\rho_0 = 2$ and $\Delta\rho = 0.1$. The correction and prediction horizons are chosen with $N_1 = 5$ and $N_2 = 2$. The associated horizon sizes lead to the better registered transient performances by the use of the supervisory loop.

Supervision of the sampling period:

The nominal sampling period is $T_0 = 0.6 \times 10^{-3}$ seconds. The time-varying sampling period is chosen within the interval $[T_{\min}, T_{\max}] = [0.5, 0.7]$ by using the constant $C = 1$ in the sampling law 2 of (14) by approximating the derivatives by finite differences. Note that the use of alternative values for this constant is in most practical examples irrelevant during the transient since the sampling rate results to be bang-bang under the sampling law (14), i.e., the sampling period takes values at the boundary of its admissibility domain provided that such a variation interval is small. The error versus number of samples of the third articulation and the updated estimates of $m_2 + m_3$ and I_{zz} are shown, respectively, in Figs. 10–12. The comparison with the corresponding unsupervised experiment is made by fixing in the unsupervised case a constant free-design parameter c_k in (13) to the initial value under supervised control $c_0 = 5 \cdot 10^6$. Fig. 10 shows the solutions of

the trajectories for the third articulation under constant nominal sampling rate and under adaptive sampling both under correction and prediction horizons $N_1 = 3$ and $N_2 = 0$ for the supervision of c_k . It can be seen that the transient is improved under adaptive sampling. Fig. 11 display the trajectories for the third articulation under correction horizons $N_1 = 0, 1, 6$ in the absence of prediction horizon ($N_1 = N_2 = 0$ corresponds to the absence of supervision on c_k). All those trajectories are obtained under adaptive sampling with the sampling period constrained to the given admissible interval. It is seen that the transient performance of the tracking error becomes improved as the prediction horizon increases. Fig. 12 shows the trajectories for the same constraints of the sampling period and horizon sizes $N_1 = 5$ (correction); $N_2 = 0, 2$ and 3 (prediction). It is seen that the transient tracking error of the third articulation is smaller, and even the output reference constant set-point error is smaller, under supervised adaptive control with use of prediction horizon compared to the use of only correction horizon in the supervisory algorithm.

5.3. Discussion of the Simulations

Worked examples led to the following conclusions on the influence of the supervisor on the improvement of the transient performance compared to the performance being achieved without supervisory actions on the free parameter of the algorithm and sampling period.

1. The moderate increase of the sizes of the optimization and correction horizons N_1 and N_2 improves the achieved performance. Good performances have been obtained for sizes of lengths less than six samples in the studied examples. The loss function decreases as the sizes of those horizons increase from zero to six. Extra increments of the horizon sizes can deteriorate the registered performances. The prediction error should not exceed significantly the size of the correction one in order to obtain good performances. An important key feature is that the stability is always maintained since the algorithm parameter is always obtained within its stability admissibility do-

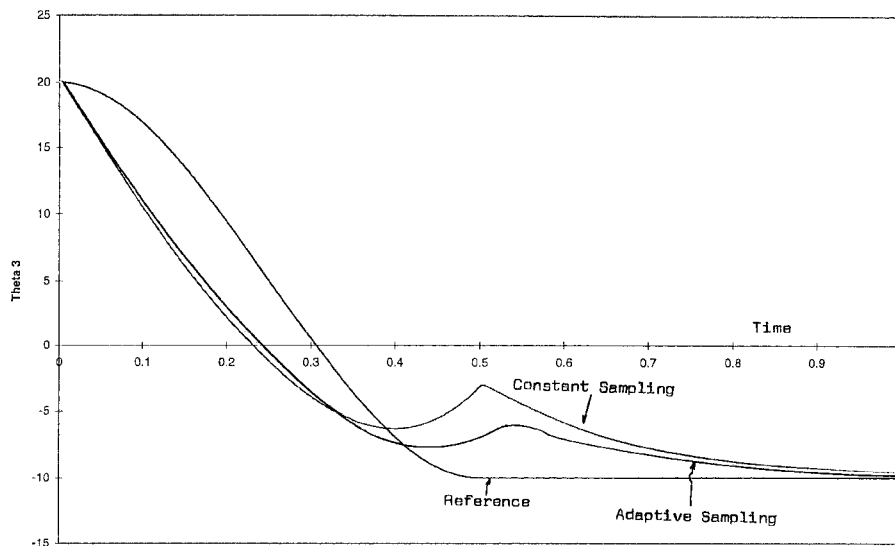
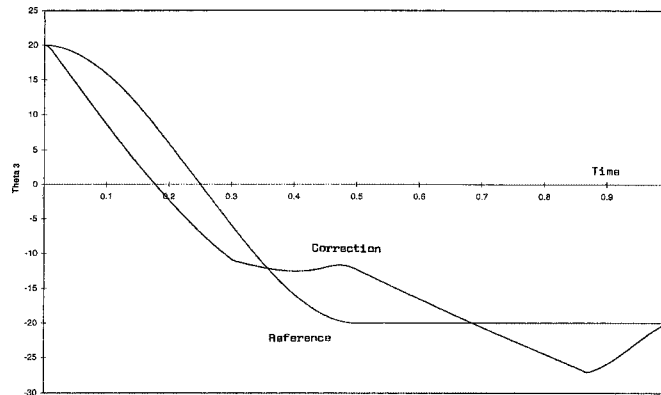
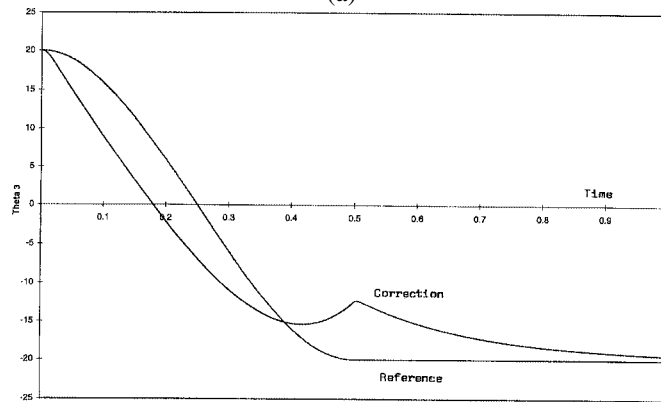


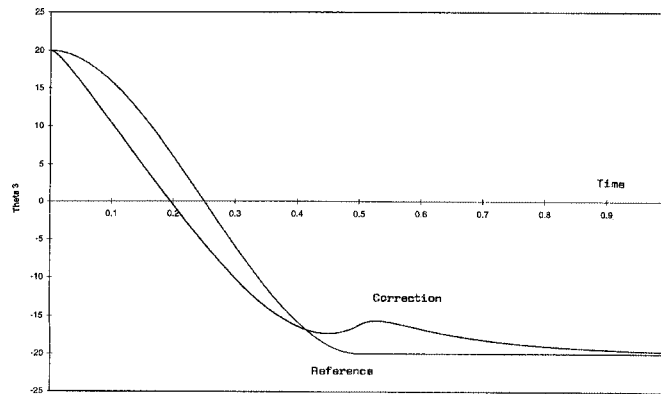
Fig. 10. Angle of the Third Articulation θ_3 versus time (seconds) with and without supervisory action in the sampling period with nominal $T_0 = 0.6 \times 10^{-3}$ seconds and $T_i \in [0.5, 0.7]$. Correction and Prediction Horizon Sizes $N_1 = 3$ and $N_2 = 0$.



(a)



(b)



(c)

Fig. 11. Angle of the Third Articulation θ_3 versus time (seconds) with and without supervisory action in the sampling period with nominal $T_0 = 0.6 \times 10^{-3}$ and $T_i \in [0.5, 0.7]$. (a) Correction and Prediction Horizon Sizes $N_1 = N_2 = 0$ (Unsupervised action in the free algorithm parameter). (b) Ibid. $N_1 = 1$ and $N_2 = 0$. (c) Ibid. $N_1 = 6$ and $N_2 = 0$.

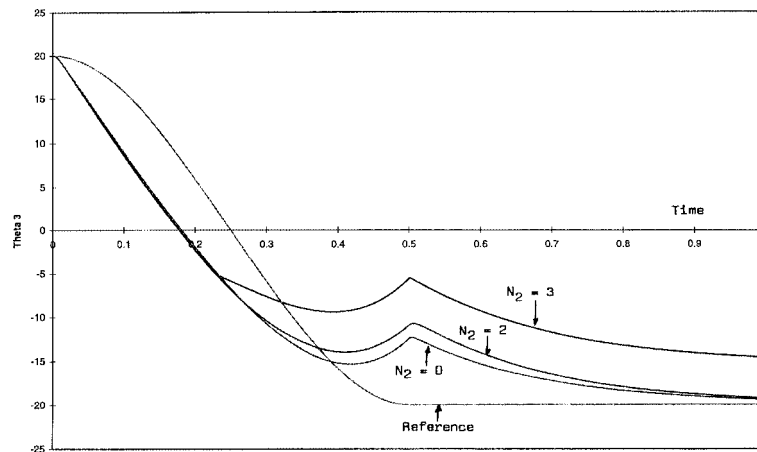


Fig. 12. Angle of the Third Articulation θ_3 versus time (seconds). Supervisory Action in the Sampling Period as in Fig. 11 with Correction and Prediction Horizon Sizes $N_1 = 5$ and $N_2 = 0, 2$ and 3 .

main (i.e., a positive finite value) and, at the same time, the supervisory action is exerted prior to the computation of the control torque.

2. The modification of the supervisory technique proposed in Remark 4 is also successful in practice to improve the transient performances as it has been verified in other alternative worked examples. In such a case, the parameter which is primarily supervised is the forgetting factor while the c_k -parameter is then on-line adjusted to maintain a bounded trace of the adaptation gain and, thus, to ensure the closed-loop stability.

3. The use of the adaptive sampling law also improves the transient performances since the signals are sampled faster as the tracking error becomes smaller. It is important to select properly the bounds for the sampling period according to the stability, bandwidth and applications requirements from 'a priori' knowledge on the system. An important feature is not allowing large sampling rate variations (i.e., to choose an admissibility domain for the sampling period of small measure around its nominal value) so as to obtain a sampling law with small sampling period variations. Acceptable values of the maximum and minimum values of the admissibility interval of the sampling period are until $\pm 20\%$ of its nominal value. *The technical reason is that the controlled discretized system becomes time-varying under adaptive sampling and it has to be 'slowly' time-varying for obtaining improved closed-loop performances since the controller parameters are adaptively re-updated.*

6. Conclusions

An approach to adaptive neural control for robot manipulators has been presented. The proposed neural design has been developed using conventional adaptive control schemes for mechanical manipulators as a starting point while using analogies between neural and adaptive controllers. In particular, the presented control scheme is a neural extension of the classical computed-torque control philosophy, with a two-layered neural network which learns the robot's inverse dynamics and on-line computes the control law. A controller supervisor is also proposed for improvement of the tracking error during the adaptation transients. The supervisor consists of two parts, namely: (1) An algorithm that selects on-line one of the free parameters of the adaptation algorithm so that the scheme performance is improved. The mechanism used is the minimization of a loss function of quadratic type of the tracking error; and (2) A sampling law which calculates on-line each next

sampling period which runs faster as the tracking error increases and vice-versa. The adaptive sampling rate operates within a neighborhood of a suitable nominal sampling period so that the bandwidth and close-loop stability requirements are also satisfied. An adaptive control scheme has been applied to the control of a simple planar manipulator with three revolute joints. The simulation results have shown that the proposed controller leads to better transient performances compared to that obtained in the unsupervised situation.

Acknowledgment

The authors are grateful to MCYT for its partial support of this work through DPI 2000-0244. They are also grateful to the reviewers by their interesting suggestions which greatly helped them to improve the original version of this manuscript.

References

- Antsaklis, P.J. (1990). Neural networks in control systems. *IEEE Control Systems Magazine*, **10**, 3-5.
- Barry, P.E. (1975). Optimal control of multirate systems. *Report RN-361*, Research Department, Grumman Aerospace Corporation.
- Craig, J.J. (1986). *Introduction to Robotics, Mechanics and Control*. Reading, Mass.: Addison-Wesley.
- Craig, J.J. (1988). *Adaptive Control of Mechanical Manipulators*. Reading, Mass.: Addison-Wesley.
- De la Sen, M. (1984a). On-line optimization of the free parameters in discrete adaptive control systems. In *Proc IEE-D*, Vol. 131, No. 4, pp. 146-157.
- De la Sen, M. (1984b). A new modeling for aperiodic sampling systems. *Int. J. of Systems Science*, **15**(3), 315-328.
- De la Sen, M. (1985). Modified adaptive controllers for improving the transients in adaptive control. *Computer Aided Design CAD*, **17**(1), 9-14.
- De la Sen, M. (1986). Multirate hybrid adaptive control. In *IEEE Trans. Automat. Control*, Vol. 17, No. 1, pp. 9-14.
- Hsia, T.C. (1972). Comparisons of adaptive sampling control laws. In *IEEE Trans. Automat. Contr.*, Vol. AC-17, pp. 830-831.
- Hsia, T.C. (1974). Analytic design of adaptive sampling control laws. In *IEEE Trans. Automat. Contr.*, Vol. AC-19, pp. 39-42.
- Feng, G., R.L. Ren (1995). Robust adaptive control for robot manipulators. *International Journal of Systems Science*, **26**, 1017-1028.
- Fuster, A., J.M. Guillen (1987). New modeling technique for aperiodic sampling of nonlinear systems. *Int. J. of Control*, **45**(3), 951-968.
- Glasson, D.P. (1980). Research in multirate estimation and control. *Report TR 1356 /1*. The Analytic Science Corporation, Reading, MA.
- Gupta, M.M., D.H. Rao (1994). *Neuro-Control Systems*. Piscataway, N.J.: IEEE Press.
- Hunt, K.J., D. Sbarbaro, R. Zbikowski, P.J. Gawthrop (1992). Neural networks for control systems - a survey. *Automatica*, **28**, 1083-1112.
- Li, W., J.J.E. Slotine (1989). An indirect adaptive controller. *Systems and Control Letters*, **12**, 259-266.
- Miyamoto, H., M. Kawato, T. Setoyama, R. Suzuki (1988). Feedback error learning neural network for trajectory control of a robotic manipulator. *Neural Networks*, **1**, 251-265.
- Narendra, K.S., K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. In *IEEE Transactions on Neural Networks*, Vol. 1, pp. 4-27.
- Ozaki, T., T. Suzuki, T. Furuhashi, S. Okuma, Y. Uchikawa (1991). Trajectory control of robotic manipulators using neural networks. In *IEEE Transactions on Industrial Electronics*, Vol. 38, pp. 641-657.
- Psaltis, D., A. Sideris, A. Yamamura. (1988). A multilayered neural network controller. In *IEEE Control Systems Magazine*, Vol. 8, pp. 17-21.
- Pham, D.T., S.J. Oh (1994). Adaptive control of a robot using neural networks. *Robotica*, **12**, 553-561.

- Sadegh, N. (1993) A perceptron network for functional identification and control of nonlinear systems. In *IEEE Transactions on Neural Networks*, Vol. 4, pp. 982–988.
- Sadegh, N., R. Horowitz (1990). Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *International Journal of Robotics Research*, **9**, 74–92.
- Slotine, J.J.E., W. Li (1989). Composite adaptive control of robot manipulators. *Automatica*, **25**, 509–519.
- Schlueter, R.A., A.H. Lewis (1973). The optimal linear regulator with state dependent sampling. In *IEEE Trans. Automat. Control*, Vol. AC-18, pp. 512–515.
- Sundararajan, N., L. Chin, Y.K. San (1993). Selection of network and learning parameters for an adaptive neural robotic control scheme. *Mechatronics*, **3**, 747–766.
- Teshnehlab, M., K. Watanabe (1994). Self-tuning of computed torque gains by using neural networks with flexible structures. In *IEE Proceedings – Control Theory and Applications*, Vol. 141, pp. 235–242.
- Widrow, B., M.A. Lehr (1990). 30 years of adaptive neural networks: perceptron, madaline and back propagation. In *Proceedings of the IEEE*, Vol. 78, pp. 1415–1442.

M. De la Sen was born in Arrigorriaga, Bizkaia, Spain in 1953. He received his M.Sc. degree with honors in applied physics from the University of the Basque Country, his Ph.D. degree also in applied physics from the same University in 1979, and the degree of "Docteur d'Etat-ès-Sciences Physiques" (specialité Automatique et Traitement du Signal) with "mention très honorable" in 1987. He has had several teaching positions in the University of the Basque Country at Bilbao, Spain, where he is currently Professor of Systems and Control Engineering. He has also had the positions of Visiting Professor in the University of Grenoble, France and the University of Newcastle, New South Wales, Australia. He has authored or co-authored a number of papers in the fields of adaptive systems, discrete systems, ordinary and functional (time-delay) differential equations, and mathematical systems theory. He had supervised fifteen Doctoral Thesis and a number of Master Research works in those fields.

A. Almansa received her M.Sc. in applied physics from the University of the Basque Country and her Ph.D. degree also in applied physics from the same University in 2000. She currently works in the private industry.

Adaptyvus ir stabilus manipiliatorių valdymas esant pagerintiems adaptacijos pereinamiesiems procesams, kai operatyviai tikrinami adaptacijos algoritmo ir diskretizacijos dažnio parametrai

Manuel De la SEN, Ana ALMANSA

Pasiūlyta mechaninių manipuliatorių adaptyvaus valdymo schema. Valdymo grandyje analizuojama roboto atvirkštinė dinamika ir operatyviai generuojamas valdymo signalas. Sistemos adaptacijos pereinamųjų procesų pagerinimui naudojamas supervizorius, kuris atlieka dvi funkcijas. Pirmą, supervizorius koreguoja adaptacijos parametrus taip, kad kvadratinė nuostolių funkcija būtų pakankamai maža. Antra supervizoriaus funkcija yra operatyviai keisti diskretizacijos periodą. Diskretizacijos periodas parenkamas taip, kad pereinamojo proceso sekimo paklaida (tracking error) pagerinama naudojant paprastą taisyklę: kuo sparčiau kinta proceso sekimo paklaida, tuo diskretizacijos dažnis turi būti didesnis.