

Adaptive Control of Robotic Manipulators with Improvement of the Transient Behavior Through an Intelligent Supervision of the Free-Design Parameters and the Sampling Period

M. De La Sen, A. Almansa and J. C. Soto

Instituto de Investigación y Desarrollo de Procesos, Leioa (Bizkaia). Aptdo. 644 de Bilbao, Spain

Abstract: An adaptive control scheme for mechanical manipulators proposed. The control loop consists of a network for learning the robot's inverse dynamics and on-line generating the control signal. Some simulation results were provided to evaluate the design. A supervisor was used to improve the performances of the system during the adaptation transients. The supervisor exerts two supervisory actions. The first one consists of updating the free-design adaptive controller parameters so that the value of a quadratic loss function maintained sufficiently small. The second supervisory action consists of an on-line adjustment of the sampling period within an interval centered at its nominal value.

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

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. In recent years, adaptive control results for robotic systems have included rigorous stability analysis (for instance^[1-4]). On the other hand, over the last few years the possible use of learning networks within a control systems environment has been considered (^[5-6] and references therein). Thus, it is important the achievement of good transient performances when synthesizing adaptive control laws. 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 be improved during the transient. 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. The proposed

supervisory scheme consists of two major actions, namely:

- * An on-line updating procedure of one of the free-design parameters of the estimation algorithm. An optimization horizon including a set of samples including past measurements and eventually, tracking error predictions is considered.
- * The sampling period is generated from an updating sampling law as dependent of the tracking error rate.

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

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

which is an nx1 vector of joint torques; Θ , $\dot{\Theta}$ and $\ddot{\Theta}$ are the nx1 vectors of joint positions, speed and accelerations, respectively; $M(\Theta)$ is the nxn mass matrix of the manipulator; $V(\Theta, \dot{\Theta})$ is an nx1 vector of centrifugal and Coriolis terms; $G(\Theta)$ is an nx1 vector of gravitational terms and $F(\Theta, \dot{\Theta})$ is an nx1 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

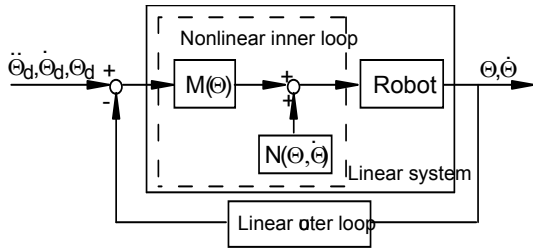


Fig. 1: Model-based feedback linearization control scheme

so-called computed-torque control using feedback linearization. The method basically consists in introducing a nonlinear model-based feedback to compensate for all the nonlinearities present in the robot (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 $\hat{G}(\Theta) + \hat{F}(\Theta, \dot{\Theta})$ be:

$\hat{\tau} = \hat{M}(\ddot{\Theta})\tau' + \hat{V}(\Theta, \dot{\Theta}) +$
 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 Θ' has been calculated as:

$$\tau' = \ddot{\Theta}_d + K_v \dot{E} + K_p E \quad (2)$$

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

$$\begin{aligned} \tilde{\tau}_k &= \tau_k - \hat{\tau}_k \\ &= \hat{M}(\Theta)\ddot{E} + \tilde{M}(\Theta)\ddot{\Theta}_d + \tilde{M}(\Theta)\ddot{E} - \tilde{M}(\Theta)\ddot{\Theta}_d \end{aligned} \quad (3)$$

(2) using (3) and then substituting in the second-order differential equation obtained from (4) the closed-loop dynamics equation is found to be:

$$\ddot{E} + K_v \dot{E} + K_p E = \hat{M}^{-1}(\Theta) [\tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta})] \quad (4)$$

where the modeling parametrical errors are ;
 $\tilde{M}(\Theta) = M(\Theta) - \hat{M}(\Theta)$
 $\tilde{V}(\Theta, \dot{\Theta}) = V(\Theta, \dot{\Theta}) - \hat{V}(\Theta, \dot{\Theta})$

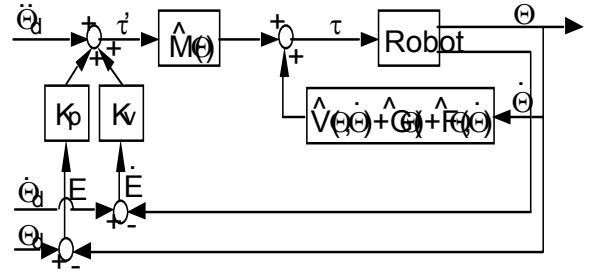


Fig. 2: Computed-torque control scheme

$$\begin{aligned} \tilde{G}(\Theta) &= G(\Theta) - \hat{G}(\Theta) \quad (5) \\ \tilde{F}(\Theta, \dot{\Theta}) &= F(\Theta, \dot{\Theta}) - \hat{F}(\Theta, \dot{\Theta}) \end{aligned}$$

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 cannot, 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.

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:

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

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} fulfill:

$$\hat{M}(\Theta)\ddot{\Theta} + \hat{V}(\Theta, \dot{\Theta}) + \hat{G}(\Theta) + \hat{F}(\Theta, \dot{\Theta})$$

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

and thus:

$$\tilde{M}(\Theta)\ddot{\Theta} + \tilde{V}(\Theta, \dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\Theta, \dot{\Theta})\tilde{M}(\Theta)$$

$$= W(\Theta, \dot{\Theta}, \ddot{\Theta})\tilde{P} \quad (9)$$

where the parameter estimation error \tilde{P} has been defined as $\tilde{P} = P - \hat{P}$. Figure 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 eqn. (7)), while its outputs are estimates of the input torques:

$$\hat{\tau}(t) = \hat{\tau}_k = \sum_{i=1}^n \sum_{j=1}^r w_{ij}(\Theta, \dot{\Theta}, \ddot{\Theta})\hat{p}_{j,k} \quad (10)$$

for $t \in [t_k, t_{k+1})$ 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:

$$\hat{P} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_r]^T, \hat{\tau} = [\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_n]^T$$

Eqn. 10 can be expressed in a familiar matrix form:

$$\hat{\tau}_k(t) = \hat{\tau}_k = W(\Theta, \dot{\Theta}, \ddot{\Theta})\hat{P}_k \quad (11)$$

for $t \in [t_k, t_{k+1})$ 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 as follows:

$$\hat{P}_{k+1} = \hat{P}_k + \frac{F_k W_k^T E_{\tau k}}{c_k + \|W_k F_k W_k^T\|} \quad (12)$$

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

where $E_{\tau k}$ 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 $F_0 = F_0^T > 0$. The parameter sequence $\lambda_k \in (0, 1)$ is the forgetting factor used to update the adaptation gain

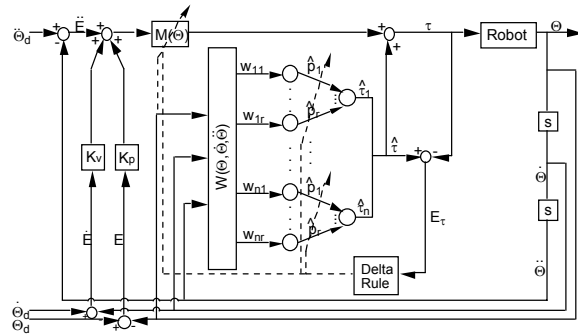


Fig. 3: Proposed neural control scheme

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) 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 then used in the simulated example 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.

Supervisor design

A. Heuristic motivation: Note, by inspection, that the learning rule associated with the adaptation algorithm (13)-(14) 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. The adaptation rate is very low when the c_k -sequence takes very large values compared to the square of the regressor norm. A second action of the supervisor 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. 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 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 at the preceding sample was to increase (decrease) the value of such a sequence. [In other words, if the tracking performance is deteriorating then make an action to correct the supervision philosophy of the last action exerted on the value of the free parameter f 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:

$$[T_{\min}, T_{\max}] = [T_0 - \Delta T, T_0 + \Delta T]$$

centred around a prefixed nominal sampling period T_0 . Thus, the sampling period decreases as the tracking error absolute value grows faster and vice versa,. In Rule 3, $\Delta T / T_0$ has to be small since the discrete parameterization of the plant is time- varying under time- varying sampling. Thus, small variations of the sampling period lead to small deviations of the plant parameterization from time- invariance and the estimation algorithm 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 $T_{\min}(k) \rightarrow T_0, T_{\max}(k) \rightarrow T_0$ as $k \rightarrow \infty$.

On the other hand, Rule 3 can also be modified by involving higher- order time - derivatives of the tracking error. The above three actions are completed with the design philosophies:

1. The sizes of the modifications in the successive values of the c_k - sequence of the parameter adaptive algorithm are 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 art isolated sampling instants .

B. Supervisory action on a free - design parameter of the adaptation algorithm: Define the loss function

$$J_k^e = \sum_{i=k-N_1}^{k+N_2} \sigma^{k-i} E_i^T Q_i E_i$$

for each current k - th sample as supervisory criterion where $E(\cdot) = [E_1(\cdot), E_2(\cdot), E_3(\cdot)]^T$ 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 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 top 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 is large compared to the older ones and subsequent future ones, respectively.
- 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 over c_k is described in the following algorithm:

C. Supervision 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 J according to the above supervisory design criteria (a) to (c). Initialize $k \rightarrow 0$.

Step 1: For each current k -th sampling instant, Make

$$\bar{c}_k = \rho_k \left\| W_k F_k W_k^T \right\| + \bar{c}$$

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

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

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

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

Step 2: Apply the parameter -estimation algorithm of Eqn. (13) and Generate the torque Eqn. (2).

Step 3: $k \rightarrow k+1$ and then Go to Step 1.

End

If the loss function value decreases the supervisory policy has to be kept. The saturation g_k for the modification of Θ_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.

D. Error prediction: The measurements of the loss function in the prediction horizon are calculated by

extrapolations of preceding predictions or real measurements by using a Taylor series expansion approximations with 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. 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. Numerical experimentation involving different sizes of the correction and prediction horizons will help the designer in the choice of their more convenient values.

E. 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 as follows:

$$T_k = \begin{cases} T_{min} & \text{if } \bar{T}_k \leq T_{min} \\ \bar{T}_k = \frac{CT_{k-1}}{|\epsilon_k - \epsilon_{k-1}|} & \text{if } T \in [T_{min}, T_{max}] \\ T_{max} & \text{if } \bar{T}_k \geq T_{max} \end{cases} \quad (14)$$

where a discrete approximation $\bar{T}_k = \frac{CT_{k-1}}{|\dot{\epsilon}(t_k)|}$ is used

by monitoring the tracking error rate and $C > 0$ is an arbitrary real constant. 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. 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 supervisory technique can be also applied to the

forgetting factor by making it time-varying with $\lambda_k \in (0, 1]$ 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

$$\frac{\lambda_k \|W_k F_k W_k^T\|}{T_r (1 - \lambda_k - \delta)}$$

with

$T_r \geq \text{Trace}(F_0) \geq \text{Trace}(F_k) > 0$. 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.

F. Closed-loop stability: The following result, whose proof is omitted, 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 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 $(0, 1]$) 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.

Simulation example: We consider the control of the simple planar mechanical manipulator with three revolute joints. 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 $I = \text{Block Diagonal} [I_{xx}, I_{yy}, I_{zz}]$. 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 . The simplified planar manipulator with three degrees of freedom can reach arbitrary positions and orientations in the plane. The elements of the dynamic equation (1) for this robot are found to be:

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

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

$$G(\Theta) = \begin{bmatrix} (m_1 + m_2 + m_3)g l_1 c_1 + (m_2 + m_3)g l_2 c_{12} \\ (m_2 + m_3)g l_2 c_{12} \\ 0 \end{bmatrix}$$

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:

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

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): $m_1=4.6$; $m_2=2.3$; $m_3=1.0$; $I_{zz}=0.1$; $l_1=l_2=0.5$; $v_1=v_2=v_3=0.5$; $k_1=k_2=k_3=0.5$. Different levels and forms of supervision follow:

A. Supervision of the free parameter c_k : Neural control is applied to a mechanical manipulator. In this

case the robot starts at position $(q_1, q_2, q_3) = (0^\circ, 30^\circ, 20^\circ)$ and the control objective is to reach the position $(q_1, q_2, q_3) = (10^\circ, -50^\circ, -20^\circ)$ in 0.5 seconds, following smooth cubic trajectories, defined by:

$$q_1(t) = 10 + 0.12 t^2 - 0.016 t^3, t \leq 0.5; q_1(0) = 10, t \geq 0.5$$

$$q_2(t) = -50 - 0.12 t^2 + 0.016 t^3, t \leq 0.5; q_2(0) = -50, t \geq 0.5$$

$$q_3(t) = -20 + 0.72 t^2 - 0.096 t^3, t \leq 0.5; q_3(0) = -20, t \geq 0.5$$

It is 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. A two-layered neural network with twelve inputs and three outputs will be used to learn the robot's inverse dynamics and to on-line generate the control signal. In particular, the values $\bar{c} = 5$, $c_{\min} = 5$, $c_{\max} = 5 \times 10^7$ and $c_0 = 5 \times 10^6$ have been chosen in the learning rule (13). In the absence of supervisor, $c = c_0 = 5 \times 10^6$, for all $k \geq 0$. The Θ weighting matrix of the loss function J^e 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.b); $\bar{c} = 5$, $\rho_0 = 2$ and $\Delta \rho = 0.1$. The correction and prediction horizons are chosen with $N_1 = 5$ and $N_2 = 2$. The 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 $K_p = K_v = \text{Diag} (100, 100, 100)$.

B. Supervision of the sampling period: 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 nominal sampling period is $T_0 = 0.6 \times 10^{-3}$ secs.

with variation domain $[T_{\min}, T_{\max}] = [0.5, 0.7]$ with $C=1$. If the adaptive sampling rate operates within a neighborhood of a suitable nominal sampling period then the bandwidth and closed-loop stability requirements are satisfied.

C. Combined supervisory of the free algorithm parameter and the sampling period: 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 $\text{Diag} [10^4, 10^4, 10^4, 10^4]$. Results are displayed on Fig. 4.

D. Summary of the performance efficiency: Exhaustively worked examples led to the following conclusions. The free-design parameter supervision and the use of a tracking error time-variation dependent adaptive sampling are successful to improve the transient performances. 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.

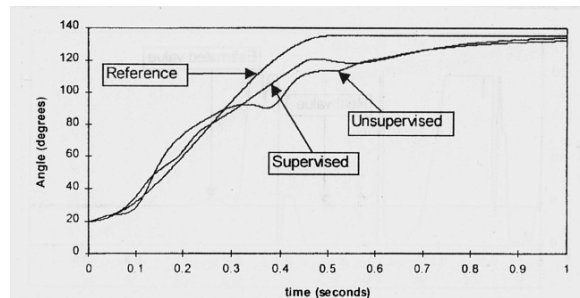


Fig. 20. Position of the third joint with and without supervision of c_k and T_k .

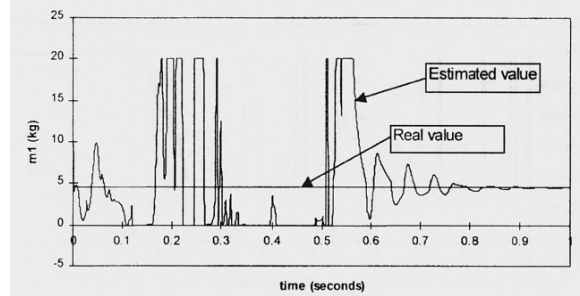


Fig. 4: Combined supervision performance

ACKNOWLEDGMENT

The authors are grateful to MCYT for its partial support of this work through DPI 2006-0714.

REFERENCES

1. Craig, J.J., 1986. Introduction to Robotics, Mechanics and Control. Reading, Mass.: Addison-Wesley.
2. Craig, J.J., 1988. Adaptive Control of Mechanical Manipulators. Reading, Mass.: Addison-Wesley.
3. Slotine, J.J.E. and LI, W., 1989. Composite adaptive control of robot manipulators. Automatica, 25: 509-519.
4. Li, W. and J.J.F. Slotine, 1989. An indirect adaptive controller. Systems and Control Letters, 12: 259-266.
5. Sadegh, N. and R. Horowitz, 1990. Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. Intl. J. Robotics Res., 9: 74-92.
6. De la Sen, M., 2005. Adaptive control of time-varying systems with time-varying delays. Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis, 12: 45-66.