We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Intelligent Control

Maouche Amin Riad
Houari Boumedien University of Algiers
Algeria

1. Introduction

This chapter presents a novel family of intelligent controllers. These controllers are based on semiphysical (or gray-box) modeling. This technique is intended to combine the best of two worlds: knowledge-based modeling and black-box modeling.

A knowledge-based (white-box) model is a mathematical description of the phenomena that occur in a process, based on the equations of physics and chemistry (or biology, sociology, etc.); typically, the equations involved in the model may be transport equations, equations of thermodynamics, mass conservation equations, etc. They contain parameters that have a physical meaning (e.g., activation energies, diffusion coefficients, etc.), and they may also contain a small number of parameters that are determined through regression from measurements.

Conversely, a black-box model is a parameterized description of the process based on statistical learning theory. All parameters of the model are estimated from measurements performed on the process; it does not take into account any prior knowledge on the process (or a very limited one). Very often, the devices and algorithms that can learn from data are characterized as intelligent. The human mental faculties of learning, generalizing, memorizing and predicting should be the foundation of any intelligent artificial device or smart system (Er & Zhou, 2009). Even if we are still far away from achieving anything similar to human intelligence, many products incorporating Neural Networks (NNs), Support Vector Machines (SVMs) and Fuzzy Logic Models (FLMs) already exhibit these properties. Among the smart controller's intelligence is its ability to cope with a large amount of noisy data coming simultaneously from different sensors and its capacity to plan under large uncertainties (Kecman, 2001).

A semiphysical (or gray-box) model may be regarded as a tradeoff between a knowledge-based model and a black-box model. It may embody the entire engineer's knowledge on the process (or a part thereof), and, in addition, it relies on parameterized functions, whose parameters are determined from measurements. This combination makes it possible to take into account all the phenomena that are not modeled with the required accuracy through prior knowledge (Dreyfus, 2005).

A controller based on gray-box modeling technique is very valuable whenever a knowledge-based model exists, but is not fully satisfactory and cannot be improved by further analysis, or can only be improved at a very large computational cost (Maouche & Attari, 2008a; 2008b). Physical systems are inherently nonlinear and are generally governed by complex equations with partial derivatives. A dynamic model of such a system, to be used in control design, is by nature an approximate model. Thus, the modeling error

introduced by this approximation influences the performance of the control. Choosing an adaptive control based on neural network, allows dealing with modeling errors and makes it possible to compensate, until a certain level, physical phenomena such as friction, whose representation is difficult to achieve (Maouche & Attari, 2007).

We will consider as an application to this type of control, a robot manipulator with flexible arms. Flexible manipulators are a good example of complex nonlinear systems difficult to model and to control.

In this Chapter we describe a hybrid approach, based on semiphysical modelling, to the problem of controlling flexible link manipulators for both structured and unstructured uncertainties conditions (Maouche & Attari, 2008a; 2008b). First, a neural network controller based on the robot's dynamic equation of motion is elaborated. It aims to produce a fast and stable control of the joint position and velocity, and to damp the vibration of each arm. Then, an adaptive neural controller is added to compensate the unknown nonlinearities and unmodeled dynamics, thus enhancing the accuracy of the control. The robustness of the adaptive neural controller is tested under disturbances and compared to a classical nonlinear controller. Simulation results show the effectiveness of the proposed control strategy.

2. Lightweight flexible manipulators

The demand for increased productivity in industry has led to the use of lighter robots with faster response and lower energy consumption. Flexible manipulator systems have relatively smaller actuators, higher payload to weight ratio and, generally, less overall cost. The drawbacks are a reduction in the stiffness of the robot structure which results in an increase in robot deflection and poor performance due to the effect of mechanical vibration in the links.

The modeling and control of non-rigid link manipulator motion has attracted researchers attentions for almost three decades. A non-rigid link in a manipulator bears a resemblance to a flexible (cantilever) beam that is often used as a starting point in modeling the dynamics of a non-rigid link (Book, 1990). Well-known approaches such as Euler-Lagrange's equation and Hamilton's principle are commonly used in modeling the motion of rigid-link manipulators and to derive the general equation of motion for flexible link manipulators. The infinite-dimensional manipulator system is commonly approximated by a finite-dimensional model for controller design. The finite element method is used in the derivation of the dynamical model leading to a computationally attractive form for the displacement bending.

The motion control of a flexible manipulator consists of tracking the desired trajectory of the rigid variables which are the angular position and velocity. But due to the elasticity of the arms, it has also to damp the elastic variables which are, in our case, deflection and elastic rotation of section of the tip. The main difficulty in controlling such a system is that unlike a rigid manipulator, a flexible manipulator is a system with more outputs to be controlled (rigid and elastic variables) then inputs (applied torques), that involves the presence of dynamic coupling equations between rigid and elastic variables.

Moreover, the dynamic effect of the payload is much larger in the lightweight flexible manipulator than in the conventional one.

However, most of the control techniques for non-rigid manipulators are inspired by classical controls. A multi-step control strategy is used in (Book et al., 1975; Hillsley & Yurkovitch,

1991; Ushiyama & Konno, 1991; Lin & Lee, 1992; Khorrami et al., 1995; Azad et al., 2003; Mohamed et al., 2005) that consists of superimposing to the control of the rigid body, the techniques of shaping or correction of the elastic effects. Other algorithms use the technique of decoupling (De Shutter et al., 1988; Chedmail & Khalil, 1989), others are based on the method of the singular perturbation approach (Siciliano & Book, 1988; Spong, 1995; Park et al., 2002), use noncollocated feedback (Ryu et al., 2004) or use model-based predictive control for vibration suppression (Hassan et al., 2007).

Neural network-based controllers were also used as they reduce the complexity and allow a faster computation of the command (Kuo & Lee, 2001; Cheng & Patel, 2003; Tian & Collins, 2004; Tang et al., 2006).

With recent developments in sensor/actuator technologies, researchers have concentrated on control methods for suppressing vibration of flexible structures using smart materials such as Shape Memory Alloys (SMA) (Elahinia & Ashrafiuon, 2001), Magnetorheological (MR) materials (Giurgiutiu et al., 2001), Electrorheological (ER) materials (Leng & Asundi, 1999), Piezoelectric transducers (PZT) (Shin & Choi, 2001; Sun et al., 2004; Shan et al., 2005), and others.

The use of knowledge-based modeling, whereby mathematical equations are derived in order to describe a process, based on a physical analysis, is important to elaborate effective controllers. However, this may lead to a complex controller design if the model of the system to be controlled is more complex and time consuming.

Therefore, we propose a controller based on artificial neural networks that approximate the dynamic model of the robot. The use of artificial neural networks, replacing nonlinear modeling, may simplify the structure of the controller and, reduce its computation time and enhance its reactivity without a loss in the accuracy of the tracking control (Maouche & Attari, 2008a; 2008b). This is important when real time control is needed.

The main advantage of neural networks control techniques among others is that they use nonlinear regression algorithms that can model high-dimensional systems with extreme flexibility due to their learning ability.

Using dynamic equations of the system to train the neural network presents many advantages. Data (inputs/outputs set) are easily and rapidly obtained via simulation, as they are not tainted with noise, and they can be generated in sufficient number that gives a good approximation of the model. Moreover, it is possible to generate data that have better representation of the model of the system.

To reduce the modeling error between the actual system and its representation, we propose to add an adaptive neural controller. Here, the neural network is trained online, to compensate for errors due to structured and unstructured uncertainties, increasing the accuracy of the overall control.

The control law presented here has several distinguished advantages. It is easy to compute since it is based on artificial neural network. This robust controller design method maximizes the control performance and assures a good accuracy when regulating the tip position of the flexible manipulator in the presence of a time-varying payload and parameter uncertainties.

3. Dynamic modeling

The system considered here consists of two links connected with a revolute joint moving in a horizontal plane as shown in Figure 1. The first and the second link are composed of a

flexible beam cantilevered onto a rigid rotating joint. It is assumed that the links can be bent freely in the horizontal plane but are stiff in the vertical bending and torsion. Thus, the Euler-Bernoulli beam theory is sufficient to describe the flexural motion of the links. Lagrange's equation and model expansion method can be utilized to develop the dynamic modeling of the robot.

As shown in Figure 1, $\{O_0\,\vec{x}_0\,\vec{y}_0\}$ represents the stationary frame, $\{O_1\,\vec{x}_1\,\vec{y}_1\}$ and $\{O_2\,\vec{x}_2\,\vec{y}_2\}$ are the moving coordinate frames with origin at the hubs of links 1 and 2, respectively. $\vec{y}_1$ and $\vec{y}_2$ are omitted to simplify the figure. θ_1 and θ_2 are the revolving angles at the hub of the two links with respect to their frames. f_1, α_1, f_2 and α_2 are the elastic displacements, they describe the deflection and the section rotation of the tip for the first and the second arm, respectively.

The motion of each arm of the manipulator is described by one rigid and two elastic variables:

$$\mathbf{q} = [\,\mathbf{q_r}\ \mathbf{q_e}\,]^{\mathrm{T}} \tag{1}$$

where $\mathbf{q_r} = [\,\theta_1\,\theta_2\,]^{\mathrm{T}}$ and $\mathbf{q_e} = [\,f_1\ \alpha_1\ f_2\ \alpha_2\,]^{\mathrm{T}}$.

The torques applied to the manipulator joints are given by:

$$\mathbf{\Gamma} = [\,\Gamma_1\ \Gamma_2\,]^{\mathrm{T}} \tag{2}$$

Let consider an arbitrary point M_i on the link i ($i = 1, 2$). The kinetic energy of the link i is given by:

$$T_i = \frac{1}{2}\rho_i \int_0^{L_i} \int_0^{S_i} V(M_i)^2 \, ds\, dx \tag{3}$$
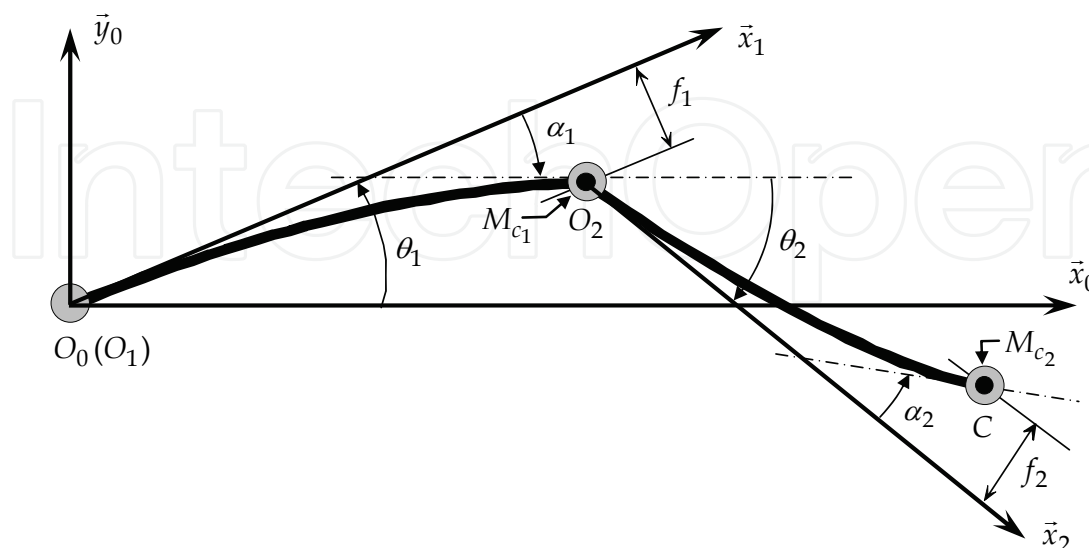

Fig. 1. Two-link manipulator with flexible arms

where $V(M_i)$ is the velocity of M_i on the flexible link i. L_i, S_i and ρ_i are the length, the section and the mass density of link i ($i = 1, 2$), respectively.

Now, the total kinetic energy T can be written as (Ower & Van de Vegt, 1987):

$$T = T_1 + T_2 + \frac{1}{2}J_{A_1}\dot{\theta}_1^{\,2} + \frac{1}{2}J_{B_1}(\dot{\theta}_1 + \dot{\alpha}_1)^2 + \frac{1}{2}J_{A_2}(\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2)^2$$

$$+ \frac{1}{2}J_{B_2}(\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2 + \dot{\alpha}_2)^2 + \frac{1}{2}M_{C_1}V(O_2)^2 + \frac{1}{2}M_{C_2}V(C)^2 \qquad (4)$$

where J_{A_i} and J_{B_i} are, respectively, the mass moment of inertia at the origin and at the end of the link i ($i = 1, 2$). Note that the first and the second terms on the right-hand side in (4) are kinetic energy of the flexible links 1 and 2, respectively. The third term is due to moment of inertia of the portion of the mass of the first actuator relative to link 1. The fourth and the fifth terms are due to moment of inertia of the portion of the mass of the second actuator in relation to link 1 and portion of the mass of the second actuator in relation to link 2, respectively. The sixth term is due to moment of inertia of mass at C (payload). The seventh and the eighth terms are kinetic energy of mass at O_2 and C respectively.

The potential energy U can be written as:

$$U = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{K}\mathbf{q} \qquad (5)$$

with $\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_e} \end{bmatrix}$, $\mathbf{K_e} = \begin{bmatrix} \mathbf{K_{E_1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{E_2}} \end{bmatrix}$ and $\mathbf{K_{E_i}} = \begin{bmatrix} 12E_iI_i\,/\,L_i^{\,3} & -6E_iI_i\,/\,L_i^{\,2} \\ -6E_iI_i\,/\,L_i^{\,2} & 4E_iI_i\,/\,L_i \end{bmatrix}$ ($i = 1, 2$).

The term on the right-hand side in (5) describes the potential energy due to elastic deformation of the links. Note that the term relative to the gravity is not present here as the manipulator moves on a horizontal plane. $\mathbf{K}$ is the stiffness matrix. The first two rows and columns of $\mathbf{K}$ are zeros as U does not depend on $\mathbf{q_r}$. E_i is the Young modulus and I_i the quadratic moment of section of the considered link.

The dynamic motion equation of the flexible manipulator can be derived in terms of Lagrange-Euler formulation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \ell}{\partial \dot{q}_r(i)}\right] - \left[\frac{\partial \ell}{\partial q_r(i)}\right] = \Gamma_i \quad (i = 1, 2) \qquad (6a)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \ell}{\partial \dot{q}_e(j)}\right] - \left[\frac{\partial \ell}{\partial q_e(j)}\right] = 0 \quad (j = 1, 4) \qquad (6b)$$

where ℓ is the Lagrangian function and $\ell = T - U$.
Substituting (4) and (5) into (6a) and (6b) yields to:

$$\mathbf{L_r}\mathbf{\Gamma} = \mathbf{A(q)}\ddot{\mathbf{q}} + \mathbf{B(q)}\left[\dot{\mathbf{q}}\dot{\mathbf{q}}\right] + \mathbf{C(q)}\left[\dot{\mathbf{q}}^2\right] + \mathbf{K}\mathbf{q} \qquad (7)$$

where $A(q)$ is the (n x n) inertia matrix, $B(q)$ is the ($n \times (n^2 - n)/2$) matrix of Coriolis terms and $[\dot{q}\dot{q}]$ is an ($(n^2 - n)/2 \times 1$) vector of joint velocity products given by: $\left[\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \dot{q}_1\dot{q}_4, ..., \dot{q}_{n-1}\dot{q}_n\right]^T$, $C(q)$ is the (n x n) matrix of centrifugal terms and $\left[\dot{q}^2\right]$ is an (n x 1) vector given by: $\left[\dot{q}_1{}^2, \dot{q}_2{}^2, ..., \dot{q}_n{}^2\right]^T$, K is the (n x n) stiffness matrix and $L_r\Gamma$ is the n torque vector $[\Gamma_1...\Gamma_r, 0...0]^T$ applied to the joints. n is the total number of variables: $n_r + n_e$ (rigid and elastic, respectively) of the system, in our case, $n = 6$, $n_r = 2$, $n_e = 4$.

If we suppose known the length of the two links, we define (Pham et al., 1991):

$$X = [J_{A_1} + J_{B_1}, J_{B_1}, M_{C_1} + M_{C_2}, \rho_1 I_1, M_1, E_1 I_1, J_{A_2} + J_{B_2}, J_{B_2}, M_{C_2}, \rho_2 I_2, M_2, E_2 I_2]^T \quad (8)$$

where X is the vector of the robot dynamic parameters.

4. Nonlinear control

This control is a generalization of the classically known 'computed torque' used to control rigid manipulator (Slotine & Li, 1987). It consists of a proportional and derived (PD) part completed by a reduced model which contains only the rigid part of the whole nonlinear dynamic model of the flexible manipulator (Pham, 1992). Let:

$$h(q, \dot{q}) = B(q)\left[\dot{q}\dot{q}\right] + C(q)\left[\dot{q}^2\right] \quad (9)$$

Then, the model can be reduced to:

$$L_r\Gamma = A(q)\ddot{q} + h(q, \dot{q})\dot{q} + Kq \quad (10)$$

or even:

$$\begin{bmatrix} \Gamma \\ 0 \end{bmatrix} = \begin{bmatrix} A_r & A_{re} \\ A_{er} & A_e \end{bmatrix}\begin{bmatrix} \ddot{q}_r \\ \ddot{q}_e \end{bmatrix} + \begin{bmatrix} h_r & h_{re} \\ h_{er} & h_e \end{bmatrix}\begin{bmatrix} \dot{q}_r \\ \dot{q}_e \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_e \end{bmatrix}\begin{bmatrix} q_r \\ q_e \end{bmatrix} \quad (11)$$

we deduce from (11) that:

$$\Gamma = A_r \ddot{q}_r + h_r \dot{q}_r + A_{re} \ddot{q}_e + h_{re} \dot{q}_e \quad (11a)$$

$$0 = A_{er} \ddot{q}_r + h_{er} \dot{q}_r + A_e \ddot{q}_e + h_e \dot{q}_e + K_e q_e \quad (11b)$$

we can then use the following control law:

$$\Gamma_{NL} = A_r(q_r, q_e)\ddot{q}_r^d + h_r(q_r, q_e, \dot{q}_r, \dot{q}_e)\dot{q}_r^d + K_{pr}\tilde{q}_r + K_{vr}\dot{\tilde{q}}_r \quad (12)$$

where, q_r^d, $\dot{q}_r^d$ and $\ddot{q}_r^d$ define the desired angular trajectory. $\tilde{q}_r = q_r^d - q_r$, $\dot{\tilde{q}}_r = \dot{q}_r^d - \dot{q}_r$ are angular position and velocity errors. K_{vr} and K_{pr} are positive definite matrices of gain. If we consider the ideal case where no errors are made while evaluating the dynamic parameters X, a Lyapunov stability analysis of this control law is presented on Appendix A.

5. Adaptive neural control

The control system structure proposed in this paper is a combination of two controllers. The construction of the first controller is based on the approximation of the nonlinear functions in (12) by neural network to reduce the computation burden. The second controller is based on adaptive neural network. Here the network is trained online, to compensate for errors due to structured and unstructured uncertainty, increasing the precision of the overall controller.

5.1 Reducing the computation burden using Neural Network

The nonlinear law presented in (12) has some major advantages as it uses information extracted from the dynamic motion equation of the system to control the manipulator. Physical characteristics like the passivity of the system can then be used to elaborate a stable controller (Kurfess, 2005).

The drawback is that, using dynamic motion equation of the system in the construction of the controller can lead to a complex controller. Computing such a controller can be time consuming. This is mainly the case with flexible manipulators as they are governed by complex equations which lead generally to a huge model. Using such a model can be incompatible with real time control.

To avoid this problem we propose to approximate the part of the model which is used in the controller with neural networks. The main feature that makes neural network ideal technology for controller systems is that they are nonlinear regression algorithms that can model high-dimensional systems and have the extreme flexibility due to their learning ability. In addition their computation is very fast.

The functions $\mathbf{A_r(q_r,q_e)}$ and $\mathbf{h_r(q_r,q_e,\dot{q}_r,\dot{q}_e)}$ are approximated with the artificial neural networks $\mathbf{A_rNN}$ and $\mathbf{h_rNN}$. We will then use their outputs in addition to the PD part of (12) to elaborate the first controller:

$$\mathbf{\Gamma_{NN}} = \mathbf{A_rNN}\,\ddot{\mathbf{q}}_r^d + \mathbf{h_rNN}\,\dot{\mathbf{q}}_r^d + \mathbf{K_{pr}}\,\tilde{\mathbf{q}}_r + \mathbf{K_{vr}}\,\dot{\tilde{\mathbf{q}}}_r \qquad (13)$$

In the neural network design scheme of $\mathbf{A_rNN}$ and $\mathbf{h_rNN}$, there are three-layered networks consisting of input, hidden and output layers. We use sigmoid functions in the hidden layer and linear functions in the output layer.

The back-propagation algorithm is adopted to perform supervised learning (Gupta et al., 2003). The two distinct phases to the operation of back-propagation learning include the forward phase and the backward phase.

In the forward phase the input signal propagate through the network layer by layer, producing the response $\mathbf{Y}$ at the output of the network:

$$\mathbf{Y} = f_o(\,f_h(\mathbf{Xi}\cdot\mathbf{Wij})\cdot\mathbf{Wjk}) \qquad (14)$$

where $\mathbf{Xi}$ is the input signal, $\mathbf{Y}$ is the actual output of the considered neural network. In this control scheme, the input signals of the input layer for $\mathbf{A_rNN}$ are the rigid and elastic position of the two links: $[\theta_1, \theta_2, f_1, \alpha_1, f_2, \alpha_2]^T$. For $\mathbf{h_rNN}$ the inputs are rigid and elastic position and velocity of the two links: $[\theta_1, \theta_2, f_1, \alpha_1, f_2, \alpha_2, \dot{\theta}_1, \dot{\theta}_2, \dot{f}_1, \dot{\alpha}_1, \dot{f}_2, \dot{\alpha}_2]^T$. $\mathbf{Xi}\cdot\mathbf{Wij}$ is the weighted sum of the outputs of the previous layer, Wij_{ij} and Wjk_{jk} denote the

weights between units i and j in the input layer to the hidden layer and between units j and k in the hidden layer to the output layer, respectively.

In this paper, the function f_o is a linear function and f_h is a tangent sigmoid function expressed by:

$$f_h(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{15}$$

The actual responses of $\mathbf{A_r NN}$ and $\mathbf{h_r NN}$ so produced are then compared with the desired responses of $\mathbf{A_r}$ and $\mathbf{h_r}$ respectively. Error signals generated are then propagated in a backward direction through the network.

In the backward phase, the delta rule learning makes the output error between the output value and the desired output value change weights and reduce error.

The training is made off line so that it does not disturb the real time control. The free parameters of the network are adjusted so as to minimize the following error function:

$$\mathbf{E_{NN}} = \frac{1}{2}(\mathbf{Y^d} - \mathbf{Y})^2 \tag{16}$$

where $\mathbf{Y^d}$ and $\mathbf{Y}$ are the desired and actual output of the considered neural network ($\mathbf{A_r NN}$ or $\mathbf{h_r NN}$).

The connect weight Wjk_{jk} is changed from the error function by an amount:

$$\Delta Wjk_{jk} = \gamma \cdot \delta k_k \cdot H_j \tag{17}$$

where γ is the learning rate and H_j is the jth hidden node.

The connect weight Wij_{ij} is changed from the error function by an amount:

$$\Delta Wij_{ij} = \gamma \cdot \delta j_j \cdot Xi_i \tag{18}$$

Delta rule learning for the units in the output layer is given by:

$$\delta k_k = Y_k^d - Y_k \tag{19}$$

Delta rule learning for the units in the hidden layer is given by:

$$\delta j_j = (1 - H_j^2) \cdot \sum_k (\delta k_k \cdot Wjk_{jk}) \tag{20}$$

Neural networks corresponding to $\mathbf{A_r NN}$ and $\mathbf{h_r NN}$ have been trained over different trajectories (training set). The stop criterion is a fundamental aspect of training. We consider, that the simple ideas of capping the number of iterations or of letting the system train until a predetermined error value are not recommended. The reason is that we want the neural network to perform well in the test set data; i.e., we would like the system to perform well in trajectories it never saw before (good generalization) (Bishop, 1995).

The error in the training set tends to decrease with iteration when the neural network has enough degrees of freedom to represent the input/output map. However, the system may

be remembering the training patterns (overfitting or overtraining) instead of finding the underlying mapping rule. To overcome this problem we have used the 'Cross Validation' method.

To avoid overtraining, the performance in a validation set (data set from trajectories that the system never saw before) must be checked regularly during training. Here, we performed once every 50 passes over the training set. The training should be stopped when the performance in the validation set starts to decrease, despite the fact that the performance in the training set continues to increase.

5.2 Construction of the adaptive neural controller

Let us consider now the case where the estimated parameters $\hat{\mathbf{X}}$ used in the dynamic equations to model the system are different from the actual parameters $\mathbf{X}$ of the manipulator. This will introduce an error in the estimation of the torque.

In addition to the structured uncertainties, there are also unstructured uncertainties due to unmodeled phenomena like frictions, perturbations etc. A more general equation of motion of the horizontal plane flexible robot is given by:

$$\mathbf{L_r \Gamma} = \mathbf{A(q)\ddot{q}} + \mathbf{B(q)\dot{q}\dot{q}} + \mathbf{C(q)\dot{q}}^2 + \mathbf{Kq} + \mathbf{F(q,\dot{q})} \tag{21}$$

where $\mathbf{F(q,\dot{q})}$ is the unstructured uncertainties of the dynamics, including frictions (viscous friction and dynamic friction) and other disturbances.

We will then add a second controller to the system based on adaptive neural network in order to compensate the errors induced by the structured and unstructured uncertainties.

The basic concept of the adaptive neural network used in the second controller is to produce an output that forms a part of the overall control torque that is used to drive manipulator joints to track the desired trajectory.

The errors between the joint's desired and actual position/velocity values are then used to train online the neural controller.

In the adaptive neural network design scheme there are also three layers. Sigmoid and linear functions are used in the hidden and the output layer respectively.

The input signals of the input layer are angular position and velocity: $[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T$ at the hub 1 and 2, and the output signals $\mathbf{Y}$ of the output layer is the torque $\mathbf{\Gamma_{AN}} = [\Gamma_{AN_1}, \Gamma_{AN_2}]^T$.

Training is made online and the parameters of the network are adjusted to minimize the following error function:

$$\mathbf{E_{AN}} = \frac{1}{2}(\mathbf{Y^d} - \mathbf{Y})^2 = \frac{1}{2}(\mathbf{K_{pn}\tilde{q}_r} + \mathbf{K_{vn}\dot{\tilde{q}}_r})^2 \tag{22}$$

where $\mathbf{Y^d}$ and $\mathbf{Y}$ are the desired and actual output of the neural network, $\mathbf{K_{pn}}$, $\mathbf{K_{vn}}$ are positive definite matrices of gain.

As the training of the adaptive neural controller is made online, we must minimize its computational time. The learning rate is designed relating the network learning, local minimum, and weight changes which can be overly large or too small in the neural network learning. A momentum factor is then used to help the network learning (Krose & Smagt, 1996).

The formulation of the weight change is then given by:

$$\Delta \mathbf{W}(t+1) = \gamma \cdot \frac{\partial \mathbf{E}}{\partial \mathbf{W}} + \eta \cdot \Delta \mathbf{W}(t) \tag{23}$$

where $\mathbf{W}$ designates $\mathbf{Wij}$ or $\mathbf{Wjk}$, t indexes the presentation number and η is a constant which determines the effect of the previous weight change.

When no momentum term is used, it takes a long time before the minimum has been reached with a low learning rate, whereas for the high learning rates the minimum is never reached because of the oscillations. When adding the momentum term, the minimum will be reached faster. This will drive the adaptive neural controller to produce a faster response. A better control can then be achieved.

The overall robotic manipulator control system proposed is shown in Figure. 2. It can be written:

$$\Gamma = \Gamma_{NN} + \Gamma_{AN} \tag{24}$$

where Γ is the overall controller output (torque); Γ_{NN} is the first controller output based on the neural model of the robot, as defined in (13); Γ_{AN} is the second controller output based on the adaptive neural network.
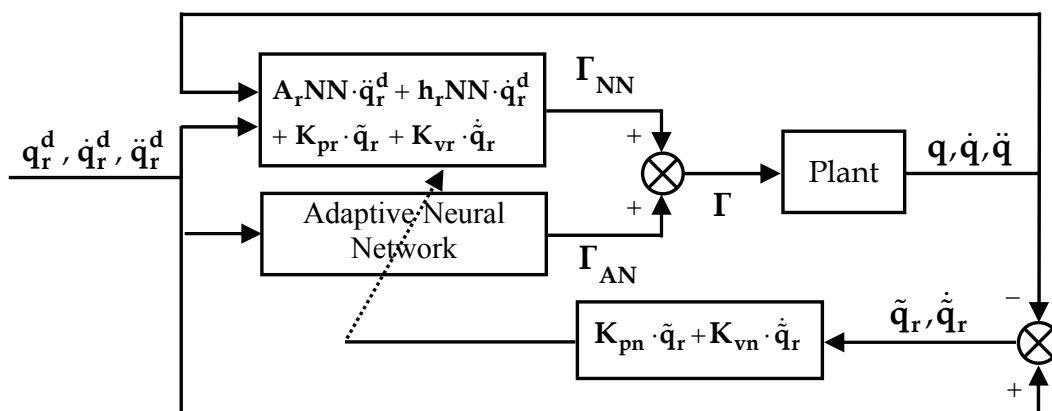

Fig. 2. The overall control system

6. Simulation results

Performance of the control strategy proposed is tested using a dynamic trajectory having 'Bang-Bang' acceleration, with a zero initial and final velocity:

$$\ddot{\theta}_1^d(t) = \ddot{\theta}_2^d(t) = \begin{cases} \dfrac{16\pi}{T^2} & \text{for } t \in \left[0, \dfrac{T}{4} \right] \\[2mm] -\dfrac{16\pi}{T^2} & \text{for } t \in \left[\dfrac{T}{4}, \dfrac{3T}{4} \right] \\[2mm] \dfrac{16\pi}{T^2} & \text{for } t \in \left[\dfrac{3T}{4}, T \right] \end{cases} \tag{25}$$

with $\theta_1^d(0) = \theta_2^d(0) = 0$ and $\dot{\theta}_1^d(0) = \dot{\theta}_2^d(0) = 0$.

To avoid the destabilization of the control induced by fast dynamics, we choose $T = 30$ *sec*. The maximum angular velocity is reached for $t = T/4$ and for $t = 3T/4$ and its absolute value is $4\pi/T$ *rad/sec* or 24 *deg/sec*.

The gain matrices are adjusted as follows:

- in the nonlinear control law (12), $\mathbf{K_{pr}} = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}$ and $\mathbf{K_{vr}} = \begin{bmatrix} 4 & 0 \\ 0 & 0.8 \end{bmatrix}$

- in the error function (23), $\mathbf{K_{pn}} = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.4 \end{bmatrix}$ and $\mathbf{K_{vn}} = \begin{bmatrix} 4.8 & 0 \\ 0 & 1.5 \end{bmatrix}$.

Let us suppose now that the actual values of the parameters of the robot are such as specified in Table 1. To test the robustness of the proposed control strategy, we consider the extreme case where the estimation error on the dynamic parameters $\mathbf{X}$ is:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{100} \qquad (26)$$

then, we use these values ($\hat{\mathbf{X}}$) in the training of $\mathbf{A_rNN}$ and $\mathbf{h_rNN}$. This will drive the first controller to produce an incorrect torque. We will see how the second controller deals with this error and how it will correct it.

Our goal here is to simulate an important error due to a bad estimation of the dynamic parameters (or ignorance of some of them). We can suppose that if the hybrid controller can handle this important error, it can a fortiori handle a small one.

For simplicity on the simulation tests, dynamic parameters are equally bad estimated in (26). Even if it is not always the case on practice, this will not affect the adaptive neural controller, which is in charge of compensating these errors, because the adaptive neural controller considers the global error (the resultant of the sum of all errors).

In order to better appreciate the effectiveness of the overall adaptive neural controller we compare its results with the nonlinear controller given in (12).

Figures 3 to 12 illustrate the results obtained with the adaptive neural controller applied to the two-link flexible manipulator. They describe the evolution of: angular position, error on position, deflection, angular velocity and error on the angular velocity, for the joints 1 and 2, respectively.

Results of the nonlinear control are reported in dashed line for comparison. The desired trajectory (target) is reported on Figures 3, 6, 8 and 11 in dotted line.

Table 2 and Table 3 presents the maximum error and the Root Mean Square error (RMS) of the angular position and velocity obtained with the two types of control strategy used.

The desired trajectory imposes a fast change of acceleration on moment $t = T/4 = 7.5$ *sec* and $t = 3T/4 = 22.5$ *sec*. This radical change from a positive to a negative acceleration for the first moment and from a negative to positive acceleration for the second one stresses the control. We can see its impact on the control of the angular velocity in Figure 6 and Figure 11.

However, the trajectory following obtained with the adaptive neural control is good and the error induced is acceptable. Whereas, the nonlinear control strongly deviates from the target.

We can see from Table 2 and Table 3 that the error on velocity, obtained with the conventional nonlinear control, reaches 0.19 *rad/sec* (10.9 *deg/sec*) for the first joint and 0.13 *rad/sec* (7.7 *deg/sec*) for the second one. With the adaptive neural control, results are

significantly better with an velocity error lower than 0.01 *rad/sec* (lower than 0.5 *deg/sec*) for the two joints.

For the position control (see Figures 3 and 8), we notice that the angular trajectory obtained with the adaptive neural controller matches perfectly the target, with an error of no more than 0.003 *rad*, (0.2 *deg*) for the first and the second links, whereas it exceeds 0.34 *rad* (20 *deg*) with the nonlinear controller for the two links (see Table 2 and Table 3).

The hybrid controller proposed deals well with the flexibility of the link as the deflection is lessened (see Figure 5 and Figure 10). However, results obtained with the nonlinear control alone are slightly better.

The deflection of the first flexible link, shown in Figure 5, is within ± 0.055 *m* with the hybrid control where as it is lower than 0.036 *m* with the nonlinear control. For the second flexible link and as shown in Figure 10, the deflection reaches 0.017 *m* with the hybrid control where as it is lower than 0.011 *m* with the nonlinear control

We notice also from Figure 5 and Figure 10, the appearance of vibrations with the hybrid control. However, their amplitude is lessened.

Therefore, we can make the following conclusion. On the one hand, the use of the nonlinear model based controller (Γ_{NN}) alone reduces the precision of the control in the presence of structured and unstructured uncertainties. But, on the other hand, the use of the adaptive part of the neural controller (Γ_{AN}) alone increases the deflection of the links and no damping of vibrations is achieved which can lead to an unstable system.

Combining these two control technique schemes gives a good compromise between stability and precision. Simulation results show the effectiveness of the control strategy proposed.

Physical parameters	Link 1	Link 2
Length (m)	$L_1 = 1.00$	$L_2 = 0.50$
Moment of inertia at the Origin of the link (kg m²)	$J_{A_1} = 1.80 \ 10^{-3}$	$J_{A_2} = 1.85 \ 10^{-4}$
Moment of inertia at the end of the link (kg m²)	$J_{B_1} = 4.70 \ 10^{-2}$	$J_{B_2} = 0.62$
Mass of the link (kg)	$M_1 = 1.26$	$M_2 = 0.35$
Mass at the tip (kg)	$M_{C_1} = 4.0$	$M_{C_2} = 1.0$
Mass density (kg/m³)	$\rho_1 = 7860$	$\rho_2 = 7860$
Young's modules	$E_1 = 1.98 \ 10^{11}$	$E_2 = 1.98 \ 10^{11}$
Quadratic moment of section (m⁴)	$I_1 = 3.41 \ 10^{-11}$	$I_2 = 6.07 \ 10^{-12}$

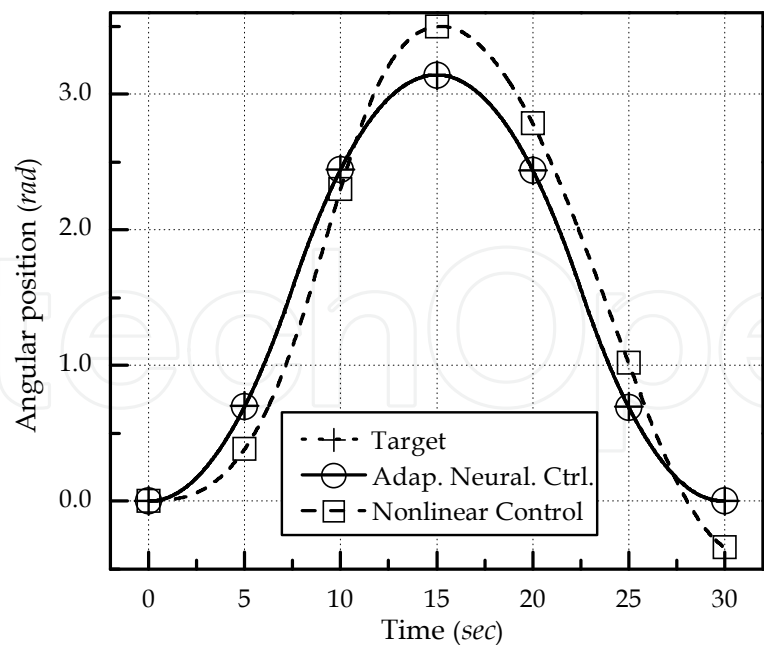Table 1. Manipulator characteristics

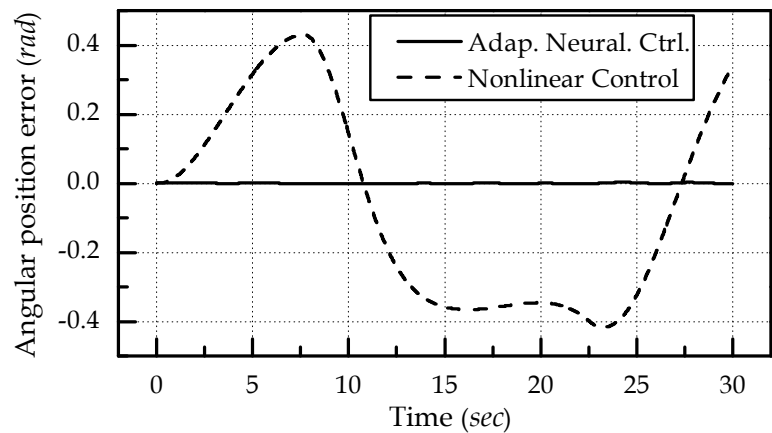Fig. 3. Evolution of the angular position θ_1 (*rad*)

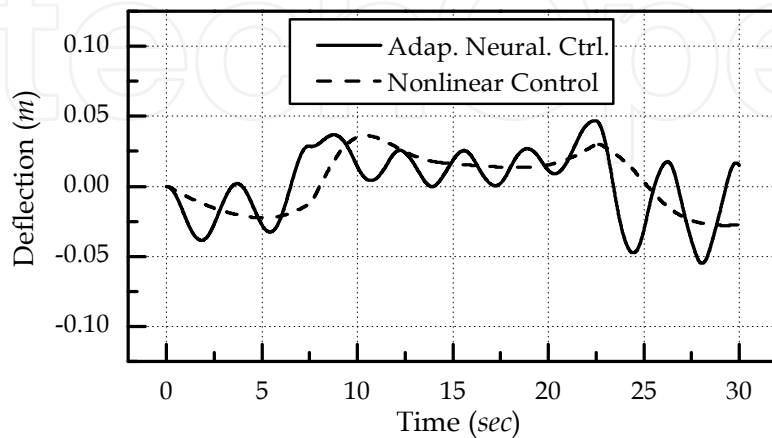Fig. 4. Evolution of the angular position error $\tilde{\theta}_1$ (*rad*)

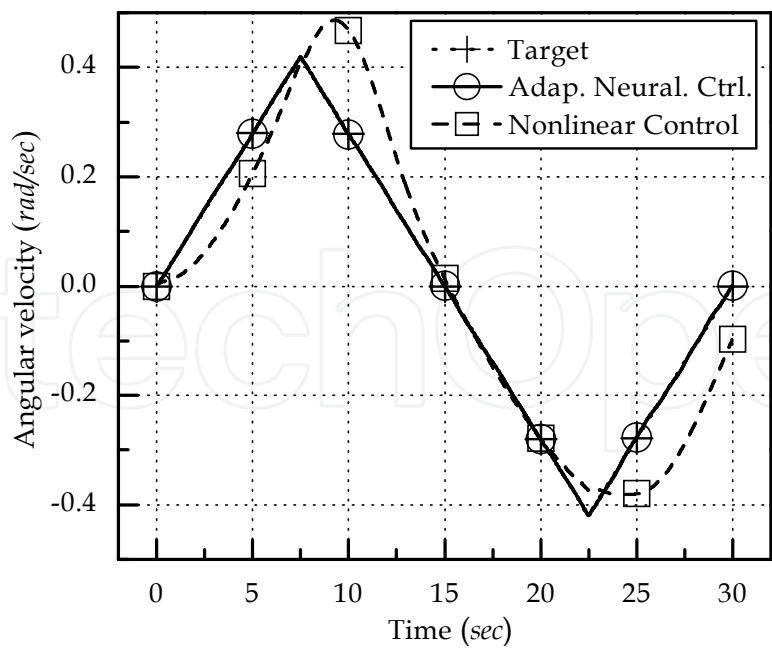Fig. 5. Evolution of the deflection f_1 (*m*)

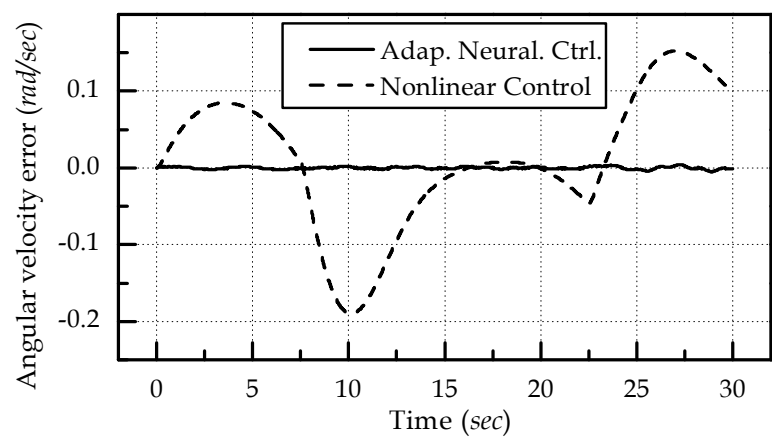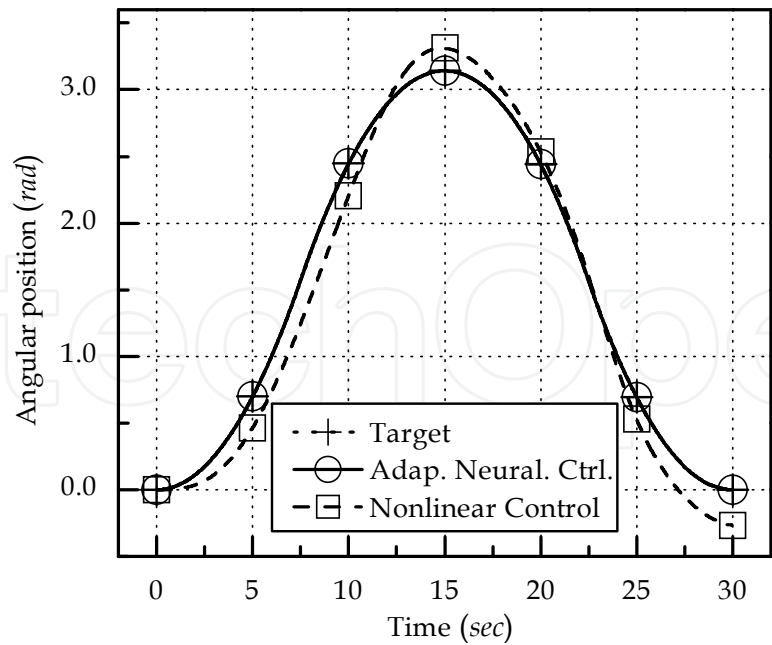Fig. 6. Evolution of the angular velocity $\dot{\theta}_1$ (*rad/sec*)

Fig. 7. Evolution of the angular velocity error $\dot{\tilde{\theta}}_1$ (*rad/sec*)

	Maximum Error		Root Mean Square Error	
Variable	θ_1 (*rad*)	$\dot{\theta}_1$ (*rad/sec*)	θ_1 (*rad*)	$\dot{\theta}_1$ (*rad/sec*)
Adap. Neural Control	3.07×10^{-3}	5.19×10^{-3}	1.14×10^{-3}	1.60×10^{-3}
Nonlinear Control	4.31×10^{-1}	1.90×10^{-1}	2.98×10^{-1}	8.92×10^{-2}

Table 2. Error on joint 1

Fig. 8. Evolution of the angular position θ_2 (*rad*)
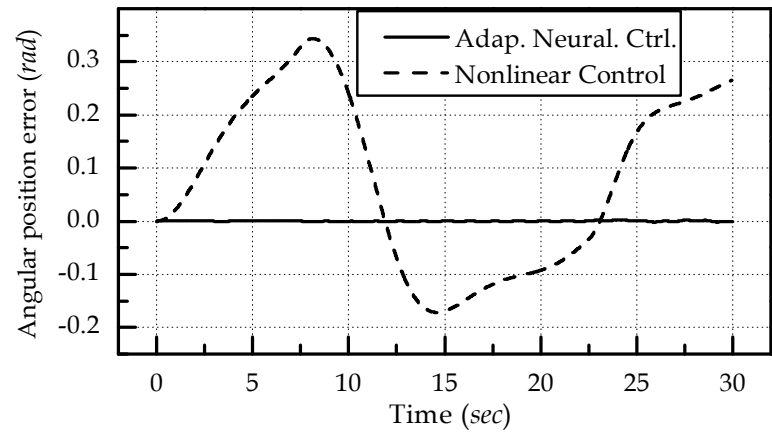

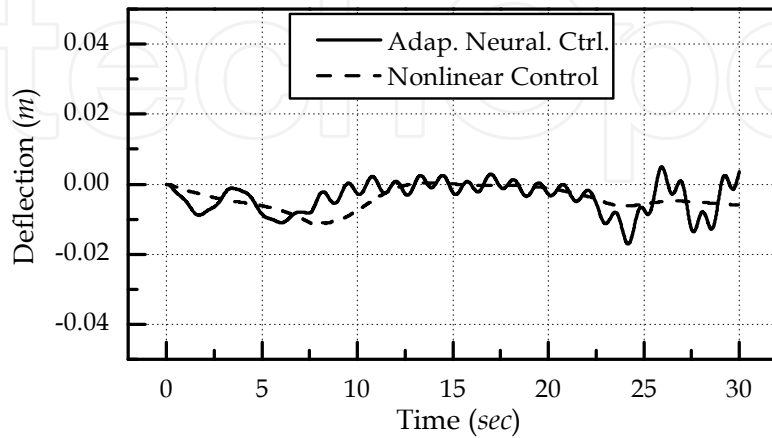Fig. 9. Evolution of the angular position error $\tilde{\theta}_2$ (*rad*)

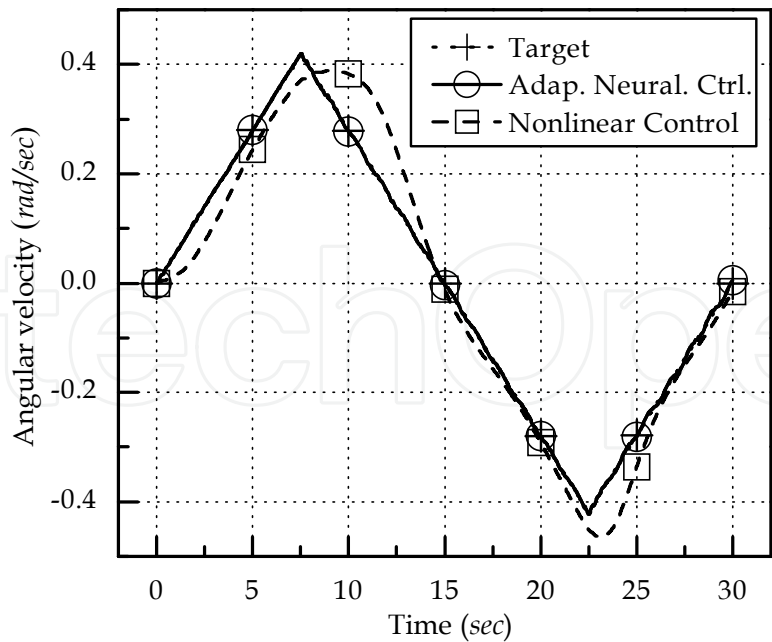
Fig. 10. Evolution of the deflection f_2 (*m*)

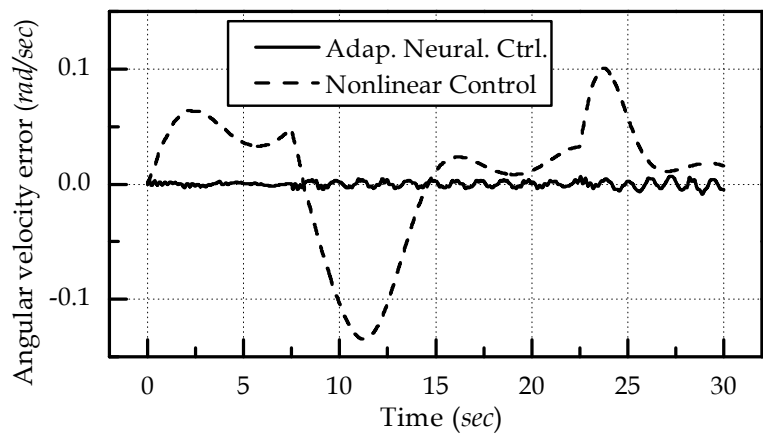Fig. 11. Evolution of the angular velocity $\dot{\theta}_2$ (*rad/sec*)

Fig. 12. Evolution of the angular velocity error $\dot{\tilde{\theta}}_2$ (*rad/sec*)

Variable	Maximum Error		Root Mean Square Error	
	θ_2 (*rad*)	$\dot{\theta}_2$ (*rad/sec*)	θ_2 (*rad*)	$\dot{\theta}_2$ (*rad/sec*)
Adap. Neural Control	2.85×10^{-3}	8.67×10^{-3}	8.43×10^{-4}	2.55×10^{-3}
Nonlinear Control	3.44×10^{-1}	1.35×10^{-1}	1.83×10^{-1}	5.53×10^{-2}

Table 3. Error on joint 2

7. Conclusion

Our goal is to search for intelligent control techniques that improve the performance of the controller and reduce the computation burden. The main idea here is to combine two control techniques, nonlinear control and neural network control.

The new adaptive neural control strategy presented is a combination of two controllers.

The first controller is based on the approximation with neural networks of the robot dynamic equation of the motion. Its aim is to provide a stable and fast control based on the dynamic model of the system. Using artificial neural network in the place of the nonlinear model allow to simplify the structure of the controller reducing its computation time and enhancing its reactivity.

The second controller is based on neural networks that are trained online. Its objective is to ensure that the actual trajectory matches the desired one by compensating errors due to structured and unstructured uncertainty, increasing the precision of the control.

Simulation results on a robot manipulator with two flexible arms have shown the robustness in performance of this control design scheme against adverse effects such as model parameter variations.

In summary, this article provides a novel control structure, to overcome the robotic manipulator control difficulties faced by conventional control schemes when uncertainties (e.g., friction, changing payload, time-varying friction, disturbances) cannot be ignored.

8. Appendix A: Stability analysis

By subtracting (12) from (11a), we obtain the error equation:

$$\mathbf{A_r}\ddot{\tilde{\mathbf{q}}}_{\mathbf{r}} + \mathbf{A_{re}}\ddot{\tilde{\mathbf{q}}}_{\mathbf{e}} + \mathbf{h_r}\dot{\tilde{\mathbf{q}}}_{\mathbf{r}} + \mathbf{h_{re}}\dot{\tilde{\mathbf{q}}}_{\mathbf{e}} + \mathbf{K_{pr}}\tilde{\mathbf{q}}_{\mathbf{r}} + \mathbf{K_{vr}}\dot{\tilde{\mathbf{q}}}_{\mathbf{r}} = 0 \tag{A.1}$$

with, $\tilde{\mathbf{q}}_{\mathbf{e}} = 0 - \mathbf{q}_{\mathbf{e}} = -\mathbf{q}_{\mathbf{e}}$ and $\dot{\tilde{\mathbf{q}}}_{\mathbf{e}} = 0 - \dot{\mathbf{q}}_{\mathbf{e}} = -\dot{\mathbf{q}}_{\mathbf{e}}$ representing the elastic stabilization errors. In addition, rewriting the coupling equation (11b) according to the trajectory and the elastic stabilization error variables ($\tilde{\mathbf{q}}_{\mathbf{r}}$ and $\tilde{\mathbf{q}}_{\mathbf{e}}$) gives:

$$\mathbf{A_{er}}\ddot{\tilde{\mathbf{q}}}_{\mathbf{r}} + \mathbf{A_e}\ddot{\tilde{\mathbf{q}}}_{\mathbf{e}} + \mathbf{h_{er}}\dot{\tilde{\mathbf{q}}}_{\mathbf{r}} + \mathbf{h_e}\dot{\tilde{\mathbf{q}}}_{\mathbf{e}} + \mathbf{K_e}\tilde{\mathbf{q}}_{\mathbf{e}} = \mathbf{A_{er}}\ddot{\mathbf{q}}_{\mathbf{r}}^{\mathbf{d}} + \mathbf{h_{er}}\dot{\mathbf{q}}_{\mathbf{r}}^{\mathbf{d}} \tag{A.2}$$

Using (A.1) and (A.2), the global error equation becomes:

$$\mathbf{A}\ddot{\tilde{\mathbf{q}}} + \mathbf{h}\dot{\tilde{\mathbf{q}}} + \mathbf{K_p}\tilde{\mathbf{q}} + \mathbf{K_v}\dot{\tilde{\mathbf{q}}} + \mathbf{s_1} = 0 \tag{A.3}$$

where the positive constant matrices $\mathbf{K_p}$, $\mathbf{K_v}$ are respectively $\begin{bmatrix} \mathbf{K_{pr}} & 0 \\ 0 & \mathbf{K_e} \end{bmatrix}$, $\begin{bmatrix} \mathbf{K_{vr}} & 0 \\ 0 & 0 \end{bmatrix}$, and

$\mathbf{s_1} = -\begin{bmatrix} 0 \\ \mathbf{A_{er}}\ddot{\mathbf{q}}_{\mathbf{rd}} + \mathbf{h_{er}}\dot{\mathbf{q}}_{\mathbf{rd}} \end{bmatrix}$.

To study the stability of the global system, the following Lyapunov function is considered:

$$\mathbf{V} = (1/2)\dot{\tilde{\mathbf{q}}}^{\mathrm{T}}\mathbf{A}\dot{\tilde{\mathbf{q}}} + (1/2)\tilde{\mathbf{q}}^{\mathrm{T}}\mathbf{K_p}\tilde{\mathbf{q}} \tag{A.4}$$

by differentiating V and using (A.3), and the fact that $\mathbf{A}$ is a symmetric positive-definite matrix (Kurfess, 2005), we obtain:

$$\dot{\mathbf{V}} = \dot{\tilde{\mathbf{q}}}^{\mathrm{T}}((1/2)\dot{\mathbf{A}} - \mathbf{h})\dot{\tilde{\mathbf{q}}} - \dot{\tilde{\mathbf{q}}}^{\mathrm{T}}(\mathbf{K_v}\dot{\tilde{\mathbf{q}}} + \mathbf{s_1}) \tag{A.5}$$

The property of passivity of the flexible manipulator implies that $(1/2)\dot{\mathbf{A}} - \mathbf{h}$ is skew symmetric (Lewis, 1999), finally we have:

$$\dot{\mathbf{V}} = -\dot{\tilde{\mathbf{q}}}_{\mathbf{r}}^{\mathrm{T}}\mathbf{K_{vr}}\dot{\tilde{\mathbf{q}}}_{\mathbf{r}} + \dot{\tilde{\mathbf{q}}}_{\mathbf{e}}^{\mathrm{T}}(\mathbf{A_{er}}\ddot{\mathbf{q}}_{\mathbf{rd}} + \mathbf{h_{er}}\dot{\mathbf{q}}_{\mathbf{rd}}) \tag{A.6}$$

The Lyapunov second method provides that the asymptotic stability of the control is assured if the following conditions are met. $\mathbf{V}$ is strictly positive everywhere except in $\tilde{\mathbf{q}} = \mathbf{0}$ where it is 0 and $\dot{\mathbf{V}}$ is strictly negative everywhere except in $\tilde{\mathbf{q}} = \mathbf{0}$ where it is 0.

These conditions are always met if the desired angular velocities and accelerations are not too significant for a given tuning of $\mathbf{K_{vr}}$, so that $\dot{\mathbf{V}}$ remains essentially negative to ensure the control stability.

9. References

Azad, A. K. M.; Tokhi, M. O. & Anand, N. (2003). Teaching of control for complex systems trough simulation. *Proceedings of the 2003 ASEE/WFEO International Colloquium*, American Society for Engineering Education.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, New York.

Book, W. J. (1990). Modeling, design, and control of flexible manipulator arms: A tutorial review. *IEEE Proceedings of the 29th Conference on Decision and Control*, pp. 500-506, Honolulu, Haway.

Book, W. J.; Maizza-Neto, O. & Whitney, D. E. (1975). Feedback control of two beam, two joint systems with distributed flexibility. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 97, No. 4, pp. 424-431.

Chedmail, P. & Khalil, W. (1989). Non linear decoupling control of flexible robots. *4th International Conference on Advanced Robotics*, pp. 138-145, Columbus.

Cheng, X. P. & Patel, R. V. (2003). Neural network based tracking control of a flexible macro–micro manipulator system. *Neural Networks*, Vol. 16, pp. 271-286.

De Shutter, J.; Van Brussels, H.; Adam, M.; Froment, A. & Faillot, J. L. (1988). Control of flexible robots using generalized non-linear decoupling. *Proceedings of IFAC*, pp. 113-118.

Dreyfus, G. (2005). *Neural Networks Methodology and Applications*. Springer, ISBN-10 3-540-22980-9, Germany.

Elahinia, M. H. & Ashrafiuon, H. (2001). Nonlinear control of a shape memory alloy actuated manipulator. *ASME Journal Of Vibration Acoustic*, Vol. 123, No. 4, pp. 487-495.

Er, M. J. & Zhou, Y. (2009). *Theory and Novel Applications of Machine Learning*. In-Tech, ISBN: 978-3-902613-55-4.

Giurgiutiu, V.; Jichi, F.; Berman, J. & Kamphaus, J. M. (2001). Theoretical and experimental investigation of magnetostrictive composite beams. *Smart Material Structure*, Vol. 10, No 5, pp. 934-935.

Gupta, M. M.; Jin, L. & Homma, N. (2003). *Static and Dynamic Neural Networks*. Wiley-Interscience, ISBN: 0-471-21948-7.

Hassan, M.; Dubay, R.; Li, C. & Wang, R. (2007). Active vibration control of a flexible one-link manipulator using a multivariable predictive. *Mechatronics*, Vol. 17, pp. 311-323.

Hillsley, K. L. & Yurkovitch, S. (1991). Vibration control of two link flexible robot arm. *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 2121-2126.

Kecman, V. (2001). *Learning and soft computing: support vector machines, neural networks and fuzzy logic models*. The MIT press, ISBN: 0-262-11255-8.

Khorrami, F.; Jain, S. & Tzes, A. (1995). Experimental results on adaptive nonlinear control and input preshaping for multi-link flexible manipulators. *Automatica*, Vol. 31, pp. 83-97.

Krose, B. & Smagt, P. (1996). *An Introduction to Neural Networks*. Amsterdam University Press.

Kuo, C. F. J. & Lee C. J. (2001). Neural network control of a rotating elastic manipulator. *Computers and Mathematics with applications*, Vol. 42, pp. 1009-1023.

Kurfess, T. R. (2005). *Robotics and Automation Handbook*. p. 81, CRC Press, ISBN: 0-8493-1804-1.

Leng, J. & Asundi, A. (1999). Active vibration control system of smart structures based on FOS and ER actuator. Smart Material Structure, Vol. 8, No. 2, pp. 252-256.

Lewis, F. L.; Fitzgerald, J. M.; Walker, I. D.; Cutkosky, M. R.; Lee, K. M.; Bailey, R.; Zhou, C.; Priest, J. W.; Stevens, G. T. Jr. & Liu, K. (1999). *Robotics. Mechanical engineering handbook*. p. 105, CRC Press LLC.

Lin, Y. J. & Lee, T. S. (1992). Comprehensive dynamic modeling and motion/force control of flexible manipulators. *Mechatronics*, Vol. 2, pp. 129-148.

Maouche, A. R. & Attari, M. (2008a). Neural network-based adaptive control of a two-link flexible manipulator. *The Mediterranean Journal of Measurement and Control*, Vol. 4, No 2, pp. 66-75, SoftMotor Ltd., United Kingdom. ISSN: 1743-9310.

Maouche, A. R. & Attari, M. (2008b). Adaptive neural controller of a rotating flexible manipulator. *19th IEEE Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM 2008)*, pp. 517-522, June 2008, Ischia, Italy, ISBN: 978-1-4244-1663-9.

Maouche, A. R. & Attari, M. (2007). Hybrid control strategy for flexible manipulators. *17th IEEE International Symposium on Industrial Electronics (ISIE 2007)*, pp. 50-55, June 2007, Vigo, Spain, ISBN: 978-1-4244-0755-2.

Mohamed, Z.; Martins, J. M.; Tokhi, M. O.; Da Costa, J. & Botto, M. A. (2005). Vibration control of a very flexible manipulator system. *Control Engineering Practice*, Vol. 13, pp. 267-277.

Ower, J. C. & Van de Vegt, J. (1987). Classical control design for a flexible manipulator: modeling and control system design. *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, pp. 485-489.

Park, N. C.; Yang, H. S.; Park, H. W. & Park, Y. P. (2002). Position/vibration Control of two-degree-of-freedom arms having one flexible link with artificial pneumatic muscle actuators. *Robotic and Autonomous Systems*, Vol. 40, pp. 239-253.

Pham, C. M. (1992). Identification and control of flexible robots. *Doctorate Thesis of University of Nantes*, France.

Pham, C. M.; Chedmail, P. & Gaultier, M. (1991). Determination of base parameters of flexible links manipulators. *Proc. IMACS Symposium on Modeling and Control of Technological Systems*, pp. 524-529.

Ryu, J. H.; Kwon, D. S. & Hannaford, B. (2004). Control of a flexible manipulator with non collocated feedback: Time-domain passivity approach. *IEEE Transaction on Robotics*, Vol. 20, No. 4, pp. 776-780.

Shan, J.; Liu, H. T. & Sun, D. (2005). Slewing and vibration control of a single-link flexible manipulator by positive feedback (PPF). *Mechatronics*, Vol. 15, pp. 487-503.

Shin, H. C. & Choi, S. B. (2001). Position control of a two-link flexible manipulator featuring piezoelectric actuators and sensors. *Mechatronics*, Vol. 11, No. 6, pp. 707-729.

Slotine, J. J. E. & Lie, W. (1987). On the adaptive control of robot manipulators. *International Newspaper of Robotics Research*, Vol.6, pp. 49-59.

Siciliano, B. & Book, W. J. (1988). A singular perturbation approach to control of lightweight flexible manipulators. *The International Journal of Robotics Research*, Vol. 7, pp. 79-90.

Spong, M. W. (1995). Adaptive Control of flexible joint manipulators: comments on two papers. *Automatica*, Vol. 31, pp. 585-590.

Sun, D.; Mills, J. K.; Shan, J. J. & Tso, S. K. (2004). A PZT actuator control of a single-link flexible manipulator based on linear velocity feedback and actuator placement. *Mechatronics*, Vol. 14, No. 4, pp. 381-401.

Tang, Y.; Sun, F. & Sun, F. (2006). Neural network control of flexible-link manipulators using sliding mode. *Neurocomputing,* Vol. 70, pp. 288-295.

Tian, L. & Collins, C. (2004). A dynamic recurrent neural network-based controller for a rigid-flexible manipulator system. *Mechatronics*, Vol. 14, pp. 471-490.

Ushiyama, M. & Konno, A. (1991). Computed acceleration control for the vibration supression of flexible robotic manipulators. *5th International Conference on Advanced Robotics*, pp. 126-131, Pisa, Italy.

Motion Control

Edited by Federico Casolo

The book reveals many different aspects of motion control and a wide multiplicity of approaches to the problem as well. Despite the number of examples, however, this volume is not meant to be exhaustive: it intends to offer some original insights for all researchers who will hopefully make their experience available for a forthcoming publication on the subject.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Maouche Amin Riad (2010). Intelligent Control, Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8, InTech, Available from: http://www.intechopen.com/books/motion-control/intelligent-control

INTECH

open science | open minds