

# A Fast Adaptive Artificial Neural Network Controller for Flexible Link Manipulators

Amin Riad MAOUCHE <sup>(1)\*</sup> and Hosna MEDDAHI <sup>(2)</sup>

<sup>(1), (2)</sup> LIMOSE, Department of computer science

M'Hamed Bougara University of Boumerdes

35000, Independence Avenue, Boumerdes

ALGERIA

<sup>(1)</sup> armaouche@gmail.com , <sup>(2)</sup> meddahi.informatique@gmail.com

*Abstract:* - This paper describes a hybrid approach to the problem of controlling flexible link manipulators in the dynamic phase of the trajectory. A flexible beam/arm is an appealing option for civil and military applications, such as space-based robot manipulators. However, flexibility brings with it unwanted oscillations and severe chattering which may even lead to an unstable system. To tackle these challenges, a novel control architecture scheme is presented. First, a neural network controller based on the robot's dynamic equation of motion is elaborated. Its aim is to produce a fast and stable control of the joint position and velocity and damp the vibration of each arm. Then, an adaptive Cerebellar Model Articulation Controller (CMAC) is implemented to balance unmodeled dynamics, enhancing the precision of the control. Efficiency of the new controller obtained is tested on a two-link flexible manipulator. Simulation results on a dynamic trajectory with a sinusoidal form show the effectiveness of the proposed control strategy.

*Key-Words:* - Adaptive control, CMAC neural network, artificial neural network, nonlinear control, flexible-link manipulator, dynamic motion equation.

## 1 Introduction

The trajectory control of a manipulator robot can be decomposed into two parts. Tracking the desired trajectory on the dynamic phase of the movement and positioning the tip of the link on the final phase of the movement. While the majority of the existing researches on the control of flexible link manipulators concentrate on the positioning phase of the movement, very few of them deal with the dynamic phase of the movement.

This paper presents a new control system structure to deal with the tracking control problem of flexible link manipulators on the dynamic phase of the trajectory.

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 present in this paper a controller based on Artificial Neural Networks (ANNs) that approximate the dynamic model of the robot.

Using ANNs, replacing nonlinear modeling, may simplify the structure of the controller, reduce its

computation time and enhance its reactivity without a loss in the accuracy of the tracking control. 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.

To reduce the modeling error between the actual system and its representation, we propose to add an adaptive Cerebellar Model Articulation Controller (CMAC).

The advantages of using a CMAC in the adaptive controller are as follows. The CMAC is fast in terms of convergence speed and computation time. Because of the associative and local generalization properties of the CMAC, the number of training cycles to converge is orders of magnitude smaller with the CMAC than with other neural networks [1]. The learning law and the output function of the CMAC are simple, so the CMAC needs fewer computations and less time to make adjustments and produce outputs than other neural networks, in which complex update laws and nonlinear sigmoid output functions are involved [2].

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

Trajectory following control of flexible-link manipulator system has been an important research area in the last three decades. A non-rigid link bears resemblance to a flexible (cantilever) beam often used as a starting point in modeling the dynamics of a flexible link [3].

Well-known approaches such as Euler–Lagrange’s equation and Hamilton’s principle commonly used in modeling the motion of rigid-link manipulator have been applied to derive the general equation of motion for flexible link manipulator.

Infinite-dimensional manipulator system is commonly approximated by a finite-dimensional model for controller design. Finite element method is used to derive the dynamic model leading to a computationally attractive form for the displacement bending [4].

However, most of the control techniques for non-rigid manipulators are inspired by classical controls. A bibliographical study allowed us to note some of them.

A multi-step control strategy was used in [5-11], which consists of superimposing to the control of the rigid body, the techniques of shaping or correction of the elastic effects. Other algorithms use the techniques of decoupling [12, 13], others are based on the method of the singular perturbation approach [14-16] or use non-collocated feedback [17]. Sliding mode control algorithms were successfully applied for maneuvering planar flexible manipulators while suppressing vibrations [18, 19].

On other hand, much research effort has been put into the design of artificial neural network and fuzzy logic-based controllers as they reduce the complexity and allow a faster computation of the command [20-32].

With recent developments in sensor/actuator technologies, many researchers have concentrated on control methods for suppressing vibrations of flexible structures using smart materials such as shape memory alloys (SMA) [33], magnetorheological (MR) materials [34],

electrorheological (ER) materials [35], piezoelectric transducers (PZT) [36-39], or using positive position feedback (PPF) to suppress the vibration amplitude of nonlinear systems [40], etc.

New techniques, based on swarm intelligence have also been used to elaborate optimal controllers for vibration suppression [41].

The majority of the existing non-linear controls nevertheless are subjected to constraints such as: frequency and damping of the mode shapes known exactly a priori or complex online computing like matrix inversion or computation of the dynamics of the manipulator. Other control techniques suffer from the lack of robustness facing significant variation of the dynamic parameters of the manipulator, particularly the payload.

The presented control law has several distinguished advantages. It is easy to compute since it does not require the computation of all or part of the dynamic model. This robust controller design method maximizes the control performance guaranteeing good precision when regulating the tip position of the flexible arm in the presence of large structured and unstructured uncertainties.

The reminder of this article is organized as follows. In Section 2, a two-link planar flexible manipulator is modeled according to Euler–Lagrange’s formulation and finite element method for the discretization. Section 3 presents the non-linear control. Stability analysis of this control method is carried out in Section 4. Section 5 presents the approach used to reduce the complexity of the controller. In Section 6 an adaptive controller is added to enhance the precision of the control. Section 7 describes simulation results. Tests have been carried out for the hybrid controller and compared to those obtained with the non-linear controller used solely. Results show the efficiency of the hybrid control strategy facing an important variation of the dynamic parameters. Finally, conclusion is presented in Section 8.

## 2 Dynamic Modeling

The system considered here consists of two links connected with a rotating joint moving in a horizontal plane as shown in Fig. 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 Fig. 1,  $\{O_0 \bar{x}_0 \bar{y}_0\}$  represents the stationary frame,  $\{O_1 \bar{x}_1 \bar{y}_1\}$  and  $\{O_2 \bar{x}_2 \bar{y}_2\}$  are the moving coordinate frames with origin at the hubs of links 1 and 2, respectively.  $\bar{y}_1$  and  $\bar{y}_2$  are omitted to simplify the figure.  $\theta_1$  and  $\theta_2$  are the revolving angles at the hub of the two links with respect to their frames.  $f_1, \alpha_1, f_2$  and  $\alpha_2$  are the elastic displacements, they describe deflection and section rotation of the tip for the first and the second arm, respectively.

Motion of each manipulator's arm is described by one rigid and one elastic variable [1]:

$$\mathbf{q} = [\mathbf{q}_r \ \mathbf{q}_e]^T \quad (1)$$

where  $\mathbf{q}_r = [\theta_1 \ \theta_2]^T$  and  $\mathbf{q}_e = [f_1 \ f_2]^T$ .

Torques applied to the manipulator joints are given by:

$$\mathbf{\Gamma} = [\Gamma_1 \ \Gamma_2]^T \quad (2)$$

Let us consider an arbitrary point  $M_1$  on the link 1. The expression of  $M_1$  in the moving coordinate frame  $\{O_1 \bar{x}_1 \bar{y}_1\}$  is given by:

$$\mathbf{O}_1 \mathbf{M}_1 = (x - y f_{,x}) \bar{x}_1 + (y + f) \bar{y}_1 \quad (3)$$

with,  $(x, y)$  the coordinates of  $M_1$  in a non deformed link,  $f$  the deflection at the abscissa  $x$  and  $f_{,x} = \frac{\partial f}{\partial x}$ .

Velocity of the point  $M_1$  is given by the equation:

$$\mathbf{V}(M_1) = \frac{d}{dt} \mathbf{O}_1 \mathbf{M}_1 = -[(y + f)\dot{\theta}_1 + y\dot{f}_{,x}] \bar{x}_1 + [(x - y f_{,x})\dot{\theta}_1 + \dot{f}] \bar{y}_1 \quad (4)$$

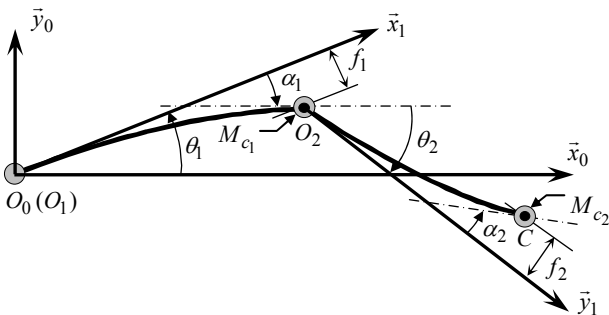


Fig. 1 Two-link manipulator with flexible arms

Finite-element theory allowed us to write the approximation [4]:

$$f(x, t) = \left( \frac{3x^2}{L_1^2} - \frac{2x^3}{L_1^3} \right) f_1(t) + L_1 \left( -\frac{x^2}{L_1^2} + \frac{x^3}{L_1^3} \right) \alpha_1(t) \quad (5)$$

with,  $L_1$  the length of the first link,  $f_1$  and  $\alpha_1$  are the deflection and the section rotation of the tip of the first link, respectively.

Bernoulli-Euler beam theory states that for a link with small elasticity, deflection and section rotation can be considered linearly dependent.

Therefore we have [42]:

$$\alpha_1 \approx \tan(\alpha_1) = \frac{3f_1}{2L_1} \quad (6)$$

with,  $\tan(\cdot)$  is the tangent function. The value of  $\alpha_1$  is considered here in radians.

The same reasoning can be made for an arbitrary point  $M_2$  on the link 2.

The kinetic energy  $T_i$  of the link  $i$  (with  $i=1, 2$ ) is then given by:

$$T_i = \frac{1}{2} \rho_i \int_0^{L_i} \int_0^{S_i} [\mathbf{V}(M_i)]^2 ds dl \quad (7)$$

where  $\mathbf{V}(M_i)$  is the velocity of  $M_i$  on the flexible link  $i$ .  $L_i$ ,  $S_i$  and  $\rho_i$  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 [43]:

$$T = T_1 + T_2 + \frac{1}{2} J_{A_1} \dot{\theta}_1^2 + \frac{1}{2} J_{B_1} (\dot{\theta}_1 + \dot{\alpha}_1)^2 + \frac{1}{2} J_{A_2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2)^2 + \frac{1}{2} J_{B_2} (\dot{\theta}_1 + \dot{\alpha}_1 + \dot{\theta}_2 + \dot{\alpha}_2)^2 + \frac{1}{2} M_{C_1} \mathbf{V}(O_2)^2 + \frac{1}{2} M_{C_2} \mathbf{V}(C)^2 \quad (8)$$

where  $J_{A_i}$  and  $J_{B_i}$  are, the mass moment of inertia at the origin and at the end of link  $i$ , respectively. Note that the first and the second terms on the right-hand side in (8) 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 [27]:

$$U = \sum_{i=1}^2 \frac{1}{2} E_i I_i \int_0^{L_i} \left[ \frac{\partial^2 f_i}{\partial x_i^2} \right]^2 dx_i \quad (9)$$

the term on the right-hand side in (9) describes the potential energy due to elastic deformation of the links. The term relative to the gravity is not present here as the manipulator moves on a horizontal plane.  $f_i$  is the deflection at the abscissa  $x_i$  of the link  $i$ .  $E_i$  is the Young's modulus and  $I_i$  the second area moment of inertia of the considered link.

Then the potential energy can be reduced to the more condensed form:

$$U = \frac{1}{2} \mathbf{q}^T \mathbf{K} \mathbf{q} \quad (10)$$

where,  $\mathbf{K}$  is the symmetric stiffness matrix. The first two rows and columns of  $\mathbf{K}$  are zeros as  $U$  does not depend on  $\mathbf{q}_r$  so it can be written  $\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_e \end{bmatrix}$ .

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

$$\frac{d}{dt} \left[ \frac{\partial \ell}{\partial \dot{q}_r(i)} \right] - \left[ \frac{\partial \ell}{\partial q_r(i)} \right] = \Gamma_i \quad (i = 1, 2) \quad (11a)$$

$$\frac{d}{dt} \left[ \frac{\partial \ell}{\partial \dot{q}_e(j)} \right] - \left[ \frac{\partial \ell}{\partial q_e(j)} \right] = 0 \quad (j = 1, 2) \quad (11b)$$

where  $\ell$  is the Lagrangian function and  $\ell = T - U$ .

Substituting (8) and (10) into (11a) and (11b) yields to [20]:

$$\mathbf{L}_r \Gamma = \mathbf{A}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{K} \mathbf{q} \quad (12)$$

where  $\mathbf{L}_r \Gamma$  is the torque vector  $[\Gamma_1, \Gamma_2, 0, 0]^T$  applied to the joint,  $\mathbf{A}(\mathbf{q})$  is the (4×4) inertia matrix,  $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$  is the (4×4) matrix of Coriolis and centrifugal terms,  $\mathbf{K}$  is the (4×4) stiffness matrix.

### 3 Nonlinear Control

This control law consists of a proportional and derivative (PD) part completed by a reduced model which contains only the rigid part of the whole nonlinear dynamic model of the flexible manipulator. It is a generalization of the classically known 'computed torque' used to control rigid manipulator [44, 45].

Dynamic equation of motion (12) can be rewritten as:

$$\begin{bmatrix} \Gamma \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_r & \mathbf{A}_{re} \\ \mathbf{A}_{er} & \mathbf{A}_e \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_r \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{B}_r & \mathbf{B}_{re} \\ \mathbf{B}_{er} & \mathbf{B}_e \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_e \end{bmatrix} \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_e \end{bmatrix} \quad (13)$$

We deduce from (13) that:

$$\Gamma = \mathbf{A}_r \ddot{\mathbf{q}}_r + \mathbf{B}_r \dot{\mathbf{q}}_r + \mathbf{A}_{re} \ddot{\mathbf{q}}_e + \mathbf{B}_{re} \dot{\mathbf{q}}_e \quad (13a)$$

$$\mathbf{0} = \mathbf{A}_{er} \ddot{\mathbf{q}}_r + \mathbf{B}_{er} \dot{\mathbf{q}}_r + \mathbf{A}_e \ddot{\mathbf{q}}_e + \mathbf{B}_e \dot{\mathbf{q}}_e + \mathbf{K}_e \mathbf{q}_e \quad (13b)$$

Therefore, we propose to use the following control law:

$$\Gamma_{NL} = \mathbf{A}_r \ddot{\mathbf{q}}_r^d + \mathbf{B}_r \dot{\mathbf{q}}_r^d + \mathbf{K}_{pr} \tilde{\mathbf{q}}_r + \mathbf{K}_{vr} \dot{\tilde{\mathbf{q}}}_r \quad (14)$$

where  $\mathbf{q}_r^d$ ,  $\dot{\mathbf{q}}_r^d$  and  $\ddot{\mathbf{q}}_r^d$  define the desired angular trajectory.  $\tilde{\mathbf{q}}_r = \mathbf{q}_r^d - \mathbf{q}_r$ ,  $\dot{\tilde{\mathbf{q}}}_r = \dot{\mathbf{q}}_r^d - \dot{\mathbf{q}}_r$  are angular position and velocity errors, respectively.  $\mathbf{K}_{pr}$  and  $\mathbf{K}_{vr}$  are positive gain constants.

### 4 Stability Analysis

In this section a Lyapunov stability analysis of the nonlinear control given in (14) is presented.

By subtracting the control law (14) from the dynamic motion equation (13a), we obtain the error equation:

$$\mathbf{0} = \mathbf{A}_r \ddot{\tilde{\mathbf{q}}}_r + \mathbf{A}_{re} \ddot{\tilde{\mathbf{q}}}_e + \mathbf{B}_r \dot{\tilde{\mathbf{q}}}_r + \mathbf{B}_{re} \dot{\tilde{\mathbf{q}}}_e + \mathbf{K}_{pr} \tilde{\mathbf{q}}_r + \mathbf{K}_{vr} \dot{\tilde{\mathbf{q}}}_r \quad (15)$$

with  $\tilde{\mathbf{q}}_e = \mathbf{0} - \mathbf{q}_e = -\mathbf{q}_e$  and  $\dot{\tilde{\mathbf{q}}}_e = \mathbf{0} - \dot{\mathbf{q}}_e = -\dot{\mathbf{q}}_e$  representing the elastic stabilization errors.

In addition, rewriting the coupling equation (13b) according to the trajectory and the elastic stabilization error variables gives:

$$\mathbf{A}_{er} \ddot{\tilde{\mathbf{q}}}_r + \mathbf{B}_{er} \dot{\tilde{\mathbf{q}}}_r = \mathbf{A}_{er} \ddot{\tilde{\mathbf{q}}}_r + \mathbf{A}_e \ddot{\tilde{\mathbf{q}}}_e + \mathbf{B}_{er} \dot{\tilde{\mathbf{q}}}_r + \mathbf{B}_e \dot{\tilde{\mathbf{q}}}_e + \mathbf{K}_e \tilde{\mathbf{q}}_e \quad (16)$$

Equations (15) and (16) allow us to write the global error equation:

$$\mathbf{A} \ddot{\tilde{\mathbf{q}}} + \mathbf{B} \dot{\tilde{\mathbf{q}}} + \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}} + \mathbf{s}_1 = \mathbf{0} \quad (17)$$

where the positive constant matrices  $\mathbf{K}_p$  and  $\mathbf{K}_v$

are  $\begin{bmatrix} \mathbf{K}_{pr} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_e \end{bmatrix}$  and  $\begin{bmatrix} \mathbf{K}_{vr} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ , respectively,

$$\text{and } \mathbf{s}_1 = - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_{er} \ddot{\tilde{\mathbf{q}}}_r + \mathbf{B}_{er} \dot{\tilde{\mathbf{q}}}_r \end{bmatrix}.$$

We consider the following Lyapunov function to study the stability of the global system,

$$\mathbf{V} = \frac{1}{2} \dot{\tilde{\mathbf{q}}}^T \mathbf{A} \dot{\tilde{\mathbf{q}}} + \frac{1}{2} \tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}} \quad (18)$$

By differentiating  $\mathbf{V}$ , using (17) and the fact that  $\mathbf{A}$  is symmetric positive-definite [46], we obtain:

$$\dot{\mathbf{V}} = \dot{\tilde{\mathbf{q}}}^T \left( \frac{1}{2} \dot{\mathbf{A}} - \mathbf{B} \right) \dot{\tilde{\mathbf{q}}} - \dot{\tilde{\mathbf{q}}}^T (\mathbf{K}_v \dot{\tilde{\mathbf{q}}} + \mathbf{s}_1) \quad (19)$$

The property of passivity of the flexible manipulator provides that  $(1/2)\dot{\mathbf{A}} - \mathbf{B}$  is skew symmetric [47], finally we have:

$$\dot{\mathbf{V}} = -\dot{\tilde{\mathbf{q}}}_r^T \mathbf{K}_{vr} \dot{\tilde{\mathbf{q}}}_r + \dot{\tilde{\mathbf{q}}}_e^T (\mathbf{A}_{er} \ddot{\tilde{\mathbf{q}}}_r^d + \mathbf{B}_{er} \dot{\tilde{\mathbf{q}}}_r^d) \quad (20)$$

The Lyapunov second method provides that the asymptotic stability of the control is borne out if the following conditions are met.

$\mathbf{V}$  is strictly positive everywhere except in  $\tilde{\mathbf{q}} = \mathbf{0}$  where it is  $\mathbf{0}$  and  $\dot{\mathbf{V}}$  is strictly negative everywhere except in  $\tilde{\mathbf{q}} = \mathbf{0}$  where it is  $\mathbf{0}$ .

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.

## 5 Reducing the Computation Burden with Artificial Neural Networks

The nonlinear law presented in (14) has some major advantages as it uses information extracted from the dynamic motion equations of the system to control the manipulator. Physical characteristics like the passivity of the system can then be used to elaborate a stable controller [27].

The drawback is that, using the model of the system in the construction of the controller can lead to a complex controller [1]. 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 parts of the model (which will be used in the controller) with ANNs.

The main feature that makes ANNs 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.

Hence, we propose to approximate the functions  $\mathbf{A}_r$  and  $\mathbf{B}_r$  with the ANNs  $\mathbf{A}_r\text{NN}$  and  $\mathbf{B}_r\text{NN}$ , respectively. We will then use their output in

addition to the PD part of (14) to elaborate the first controller:

$$\begin{aligned} \Gamma_{\text{NN}} = & \mathbf{A}_r\text{NN}(\mathbf{q}_r, \mathbf{q}_e) \ddot{\mathbf{q}}_r^d \\ & + \mathbf{B}_r\text{NN}(\mathbf{q}_r, \mathbf{q}_e, \dot{\mathbf{q}}_r, \dot{\mathbf{q}}_e) \dot{\mathbf{q}}_r^d \\ & + \mathbf{K}_{pr} \tilde{\mathbf{q}}_r + \mathbf{K}_{vr} \dot{\tilde{\mathbf{q}}}_r \end{aligned} \quad (21)$$

Multi Layer Perceptron (MLP) model is adopted in the neural network design scheme of  $\mathbf{A}_r\text{NN}$  and  $\mathbf{B}_r\text{NN}$ , with three-layered networks consisting of input, hidden and output layers. We used a sigmoid function in the hidden layer and a linear function in the output layer. There are 8 neurons in the hidden layer of  $\mathbf{A}_r\text{NN}$  and 12 neurons in the hidden layer of  $\mathbf{B}_r\text{NN}$ .

Back-propagation algorithm is adopted to perform supervised learning. 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 a response at the output of the network.

In this control scheme, the input signals of the input layer for  $\mathbf{A}_r\text{NN}$  are the rigid and elastic position of the two links:  $[\theta_1, \theta_2, f_1, f_2]^T$ . For  $\mathbf{B}_r\text{NN}$  the inputs are rigid and elastic position and velocity of the two links:  $[\theta_1, \theta_2, f_1, f_2, \dot{\theta}_1, \dot{\theta}_2, \dot{f}_1, \dot{f}_2]^T$ .

The actual response of  $\mathbf{A}_r\text{NN}$  and  $\mathbf{B}_r\text{NN}$  so produced are then compared with the desired response of  $\mathbf{A}_r$  and  $\mathbf{B}_r$  respectively, generating error signals that 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 the weights and reduce the error. The training is made off line so that it does not disturb the real time control.

## 6 Adaptive CMAC Neural Control

To reduce the modeling error between the actual system and its representation, we propose to add an adaptive Cerebellar Model Articulation Controller (CMAC).

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

$$\Gamma = \Gamma_{\text{NN}} + \Gamma_{\text{CMAC}} \quad (22)$$

where  $\Gamma$  is the overall controller's output (torque),  $\Gamma_{NN}$  is the neural network controller's output as defined in (21) and  $\Gamma_{CMAC}$  is the CMAC adaptive neural controller's output.

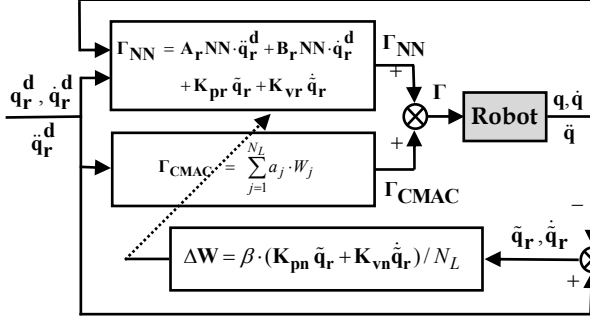


Fig. 2 The overall control system

The CMAC offers the potential of parallel computation with high flexibility; it can improve the controller's response time, important for robotic dynamic tracking. The fast convergence of its algorithm is essential for online adaptation and motivates us to use it in the adaptive control part of the proposed control strategy.

CMAC is an associative memory neural network in that each of inputs maps to a subset of weights whose values are summed to produce outputs.

An important concept used here is generalization that assumes that similar states require similar control efforts. The use of generalization speeds up learning because a group of memory cells that are close is updated in each control cycle. Generally, all the memory cells in a hypercubic region are updated in each control cycle. On the other hand, input vectors that are far away from each other will generate independent outputs [1].

The Cerebellar Model Articulation Controller (CMAC) network was first developed by Albus [48] to approximate the information processing characteristics of the human cerebellum. Miller [49] later developed a practical implementation of the CMAC neural network that could be applied to real-time control applications.

On a typical Albus CMAC neurons are called receptive fields and are organized as follows. The total collection of receptive fields is divided into  $N_L$  layers. The layers represent parallel  $N$ -dimensional hyperspaces for a network with  $N$  inputs. The receptive fields in each of the layers have rectangular boundaries and are organized so as to span the input space without overlap. Any input

vector excites one receptive field from each layer, for a total of  $N_L$  excited receptive fields for any input. Each of the layers of receptive fields is identical in organization, but each layer is offset by a quantity  $Q_L$  relative to the others in the input hyperspace.

The width of the receptive fields produces input generalization, while the offset of the adjacent layers input quantization. The ratio of the width of each receptive field to the offset between adjacent layers must be equal to  $N_L$  for all dimensions of the input space. The integer parameter  $N_L$  is referred to as the generalization parameter.

Let us take a two-dimensional CMAC (2D CMAC), as an example. The structure of a 2D CMAC is shown in Fig. 3. The two-dimensional input vector is defined by two input variables,  $x_1$  and  $x_2$ , quantized into three discrete regions, called receptive fields.

It is noted that the width of receptive fields affects the generalization capability of the CMAC network. For the first layer, the variable  $x_1$  is divided into receptive fields A, B, and C and the variable  $x_2$  is divided into receptive fields a, b, and c. The areas Aa, Ab, Ac, Ba, Bb, Bc, Ca, Cb, and Cc formed by quantized regions are called hypercubes.

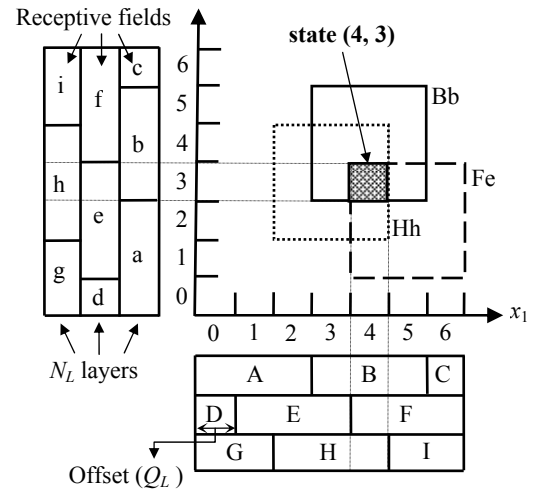


Fig. 3 Structure of a two-dimensional CMAC

By shifting each block a small interval  $Q_L$ , different hypercubes can be obtained. In Fig. 3, there are 27 hypercubes used to distinguish 49 different states in the 2D CMAC. For example, let the hypercubes Bb, Fe, and Hh be addressed by the state  $(x_1, x_2) = (4, 3)$ , only these three hypercubes are

one, and the others are zero. The trained data are stored into these regions.

The CMAC network is a local network. For a given input vector, only a few of the network's nodes (or hypercube cells) will be active and will effectively contribute to the corresponding network output. The general architecture of the CMAC network is shown in Fig. 4.

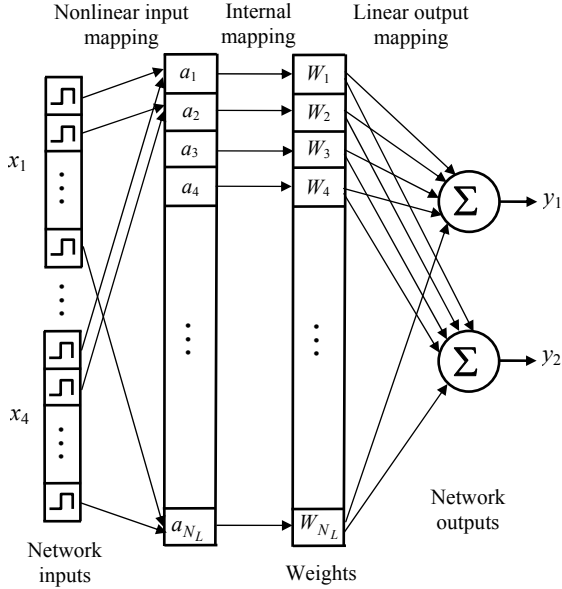


Fig. 4 Architecture of the CMAC network

The basic idea of the CMAC network is to store learned data into overlapping regions in a way that the data can easily be recalled but use less storage space. Furthermore, the action of storing weight information in the CMAC network is similar to that of the cerebellum in humans.

In the CMAC network used, each nonlinear output function  $y_i$  of the CMAC network corresponds to one CMAC controller's output (torque)  $\Gamma_{\text{CMAC}_i}$  (with  $i=1,2$ ).

The vector of nonlinear output functions  $\mathbf{y} = f(\mathbf{x})$  is approximated using two primary mappings,  $S: Q \rightarrow A$  and  $P: A \rightarrow D$ . Here,  $Q$  is a 4-dimensional input space corresponding to the angular position and velocity of joint 1 and joint 2,  $A$  is a  $N_A$ -dimensional association space, and  $D$  is a 2-dimensional output space corresponding to the CMAC adaptive neural controller's torque  $\Gamma_{\text{CMAC}}$ . The function  $S(\mathbf{x})$  maps each point  $\mathbf{x}$  in the input space onto an association vector  $\mathbf{a} = S(\mathbf{x}) \in A$  that has  $N_L$  nonzero elements, with  $N_L < N_A$ .

For a conventional CMAC, the association matrix contains only binary elements, either zero or one. The function  $P(\mathbf{a})$  computes an output vector  $\mathbf{y}$  by projecting the association vector onto a matrix  $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2]^T$  of adjustable weights so that each output  $y_i$  can be obtained by evaluating the inner product of the two vectors  $\mathbf{a}$  and  $\mathbf{W}_i$  with  $i=1,2$ .

Finally, each actual output  $y_i$  is derived as follows:

$$y_i = \Gamma_{\text{CMAC}_i} = \mathbf{a}^T \mathbf{W}_i = \sum_{j=1}^{N_L} a_j \cdot W_{i,j} \quad (23)$$

where  $a_j$  represents the  $j^{\text{th}}$  element of the association vector  $\mathbf{a}$  and  $W_{i,j}$  the  $j^{\text{th}}$  element of the weight vector  $\mathbf{W}_i$  with  $i=1,2$ .

The basic concept of the adaptive CMAC 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 the manipulator joint to track the desired trajectory.

Errors between the joint's desired and actual position/velocity values are then used to train the CMAC neural controller. Training is made online and the weight adjustment  $\Delta \mathbf{W}$  is given by:

$$\begin{aligned} \Delta \mathbf{W} &= \frac{\beta \cdot (\mathbf{y}^d - \mathbf{y})}{N_L} \\ &= \frac{\beta \cdot (\mathbf{K}_{\text{pn}} \tilde{\mathbf{q}}_r + \mathbf{K}_{\text{vn}} \dot{\tilde{\mathbf{q}}}_r)}{N_L} \end{aligned} \quad (24)$$

where  $\mathbf{y}^d$  and  $\mathbf{y}$  are the vectors of the desired and actual outputs of the CMAC network, respectively.  $\mathbf{K}_{\text{pn}}$  and  $\mathbf{K}_{\text{vn}}$  are positive gain constants.  $\beta$  is the learning rate.

## 7 Simulation Analysis

Performance of the hybrid controller given in (22) is tested using a dynamic trajectory having a sinusoidal form:

$$\theta_1^d(t) = \theta_2^d(t) = \pi \sin\left(\frac{2\pi}{T}t\right) \quad (25)$$

We choose  $T = 20$  s, to avoid the destabilization of the control law induced by fast dynamics. Parameters of the first controller as follows:

$$\mathbf{K}_{\text{pr}} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \text{ and } \mathbf{K}_{\text{vr}} = \begin{bmatrix} 4 & 0 \\ 0 & 0.8 \end{bmatrix}.$$

Parameters of the second controller are as follows:  $\mathbf{K}_{\mathbf{pn}} = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.4 \end{bmatrix}$ ,  $\mathbf{K}_{\mathbf{vn}} = \begin{bmatrix} 4.8 & 0 \\ 0 & 1.5 \end{bmatrix}$ ,  $N_L = 32$ ,  $Q_L = 0.5$ ,  $\beta = 0.9$ .

Let suppose that the actual values of the parameters of the robot are such as specified in Table 1. To test the robustness of the control strategy proposed we consider the extreme case where the estimated values of the dynamic parameters of the robot:  $J_{A_i}$ ,  $J_{B_i}$ ,  $M_i$ ,  $M_{C_i}$ ,  $\rho_i$ ,  $E_i$  and  $I_i$  (with  $i=1,2$ ) are the tenth of their actual values given in Table 1.

We will then, use these estimated values to compute  $\mathbf{A}_r$  and  $\mathbf{B}_r$  and therefore  $\mathbf{A}_r\mathbf{NN}$  and  $\mathbf{B}_r\mathbf{NN}$ .

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.

In order to better appreciate the effectiveness of the hybrid controller given in (6), we will compare its results with the nonlinear controller given in (4). 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. Even if it is not always the case on practice, this will not affect the adaptive CMAC controller, which is in charge of compensating these errors, because this controller considers the global error (the resultant of the sum of all errors).

**Table 1** Manipulator characteristics.

Physical parameters	Link 1	Link 2
Length (m)	$L_1 = 0.80$	$L_2 = 0.50$
Moment of inertia at the origin of the link ( $\text{kg m}^2$ )	$J_{A_1} = 1.80 \cdot 10^{-3}$	$J_{A_2} = 1.85 \cdot 10^{-4}$
Moment of inertia at the end of the link ( $\text{kg m}^2$ )	$J_{B_1} = 4.70 \cdot 10^{-2}$	$J_{B_2} = 0.62$
Mass of the link (kg)	$M_1 = 1.89$	$M_2 = 1.18$
Mass at the tip (kg)	$M_{C_1} = 1.0$	$M_{C_2} = 0.5$
Mass density ( $\text{kg/m}^3$ )	$\rho_1 = 7860$	$\rho_2 = 7860$
Young modulus	$E_1 = 1.98 \cdot 10^{11}$	$E_2 = 1.98 \cdot 10^{11}$
Second area moment of inertia ( $\text{m}^4$ )	$I_1 = 2.25 \cdot 10^{-10}$	$I_2 = 2.25 \cdot 10^{-10}$

Fig. 5 to Fig. 14 illustrate the results obtained with the hybrid controller applied to the two-link flexible manipulator. These figures describe the evolution of: angular position, error on position, deflection, angular velocity and error on the angular velocity, for the joints 1-2, respectively.

Table 2 and Table 3 present the maximum error and the root mean square (RMS) error on the angular position and velocity obtained with the two types of control strategy used for the joints 1-2, respectively.

Results of the nonlinear control are reported in dashed line for comparison. The desired trajectory (target) is reported on Fig. 5, Fig. 8, Fig. 10 and Fig. 13 in dotted line.

For the position control (see Fig. 5 and Fig. 10), we notice that the angular trajectory obtained with the hybrid controller matches perfectly the target, with an error of no more than 0.012 rad (0.7 deg) for the first and the second links (see Fig. 6 and Fig. 11), whereas it reaches 1.2 rad (69 deg) with the nonlinear controller (see Tables 2-3).

Velocity tracking with the hybrid controller is good (see Fig. 8 and Fig. 13) and the error induced is acceptable, whereas the nonlinear controller strongly deviates from the target.

Velocity error obtained with the hybrid controller is lower than 0.024 rad/s (1.35 deg/s) for the first and the second links (see Fig. 9 and Fig. 14), whereas it reaches 0.696 rad/s (20.9 deg/s) with the nonlinear controller (see Tables 2-3).

The proposed controller deals well with the elasticity of the links (see Fig. 7 and Fig. 12). As the manipulator is on the dynamic phase of the movement, with an imposed position and velocity we can't avoid the deflection of the links. Our purpose here is rather to dampen vibrations. We can see from Fig. 7 and Fig. 12 that whenever vibrations appear they are lessened.

With the nonlinear controller we notice the presence of high frequency oscillations at the tip of the links and in the angular velocity tracking.

Using the nonlinear controller with faster trajectory and/or more flexible links lead to the destabilization of the trajectory control due to the presence of these oscillations. Here a better behavior of the trajectory tracking system is achieved with the hybrid controller.

Results obtained demonstrate the good performance of the proposed control strategy.



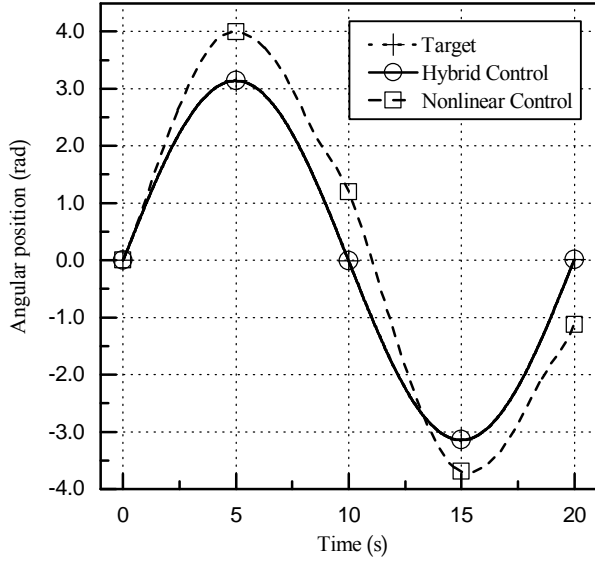


Fig. 5 Evolution of the angular position  $\theta_1$

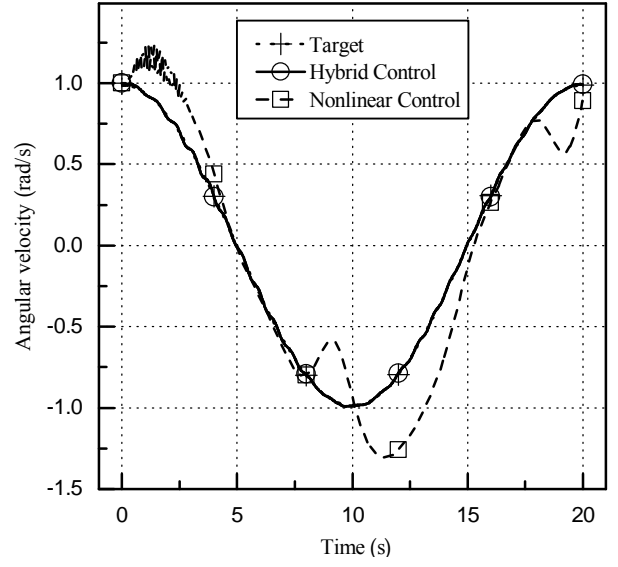


Fig. 8 Evolution of the angular velocity  $\dot{\theta}_1$

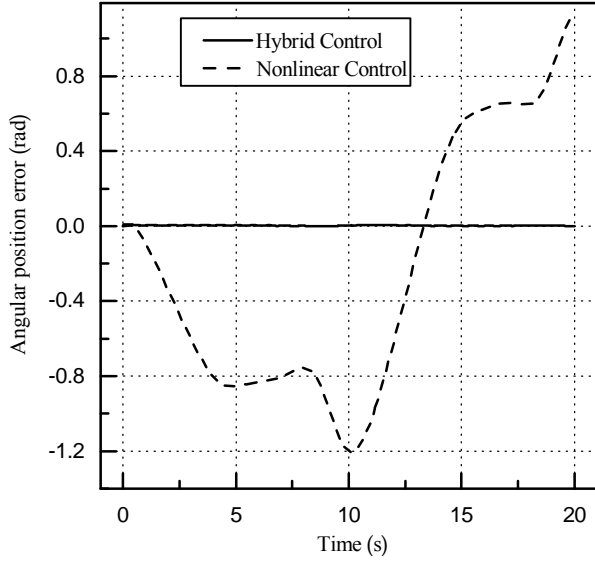


Fig. 6 Evolution of the angular position error  $\tilde{\theta}_1$

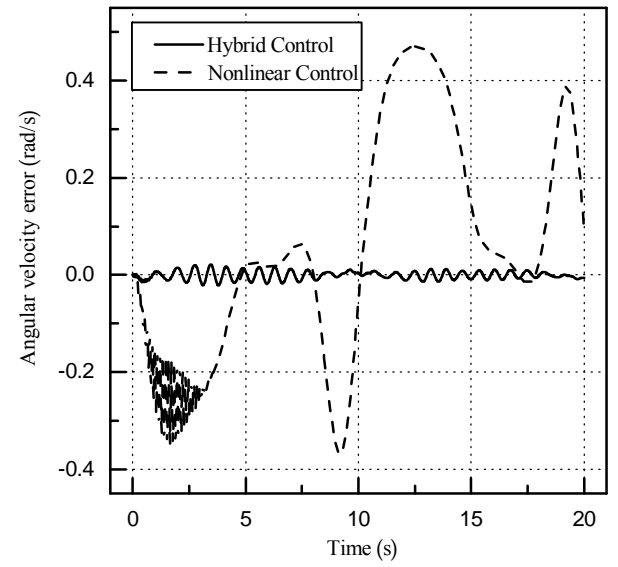


Fig. 9 Evolution of the angular velocity error  $\dot{\tilde{\theta}}_1$

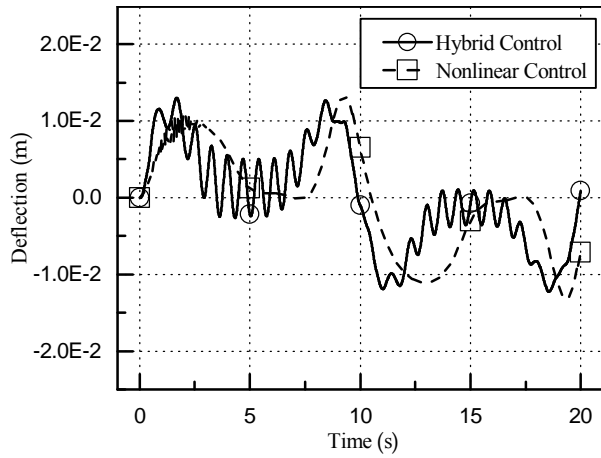


Fig. 7 Evolution of the deflection  $f_1$

**Table 2** Trajectory error on joint 1

Variable	Max. Error		Root Mean Square	
	$\theta_1$ (rad)	$\dot{\theta}_1$ (rad/s)	$\theta_1$ (rad)	$\dot{\theta}_1$ (rad/s)
Hybrid Control	9.86 $\times 10^{-3}$	2.19 $\times 10^{-2}$	3.98 $\times 10^{-3}$	8.65 $\times 10^{-3}$
Nonlinear Control	1.21	4.70 $\times 10^{-1}$	7.25 $\times 10^{-1}$	2.43 $\times 10^{-1}$

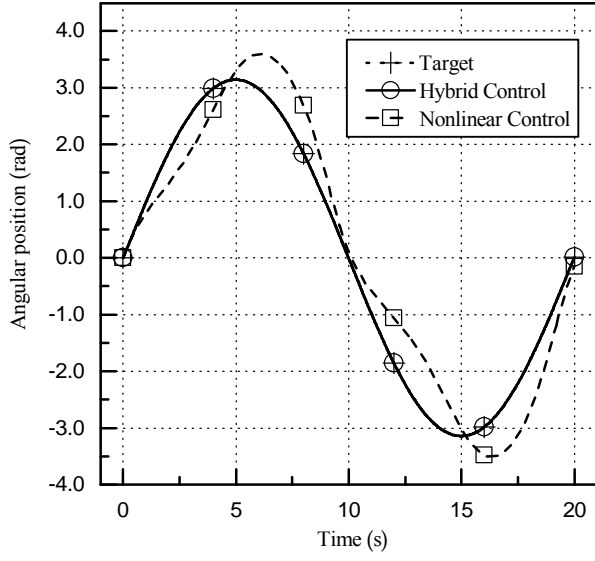


Fig. 10 Evolution of the angular position  $\theta_2$

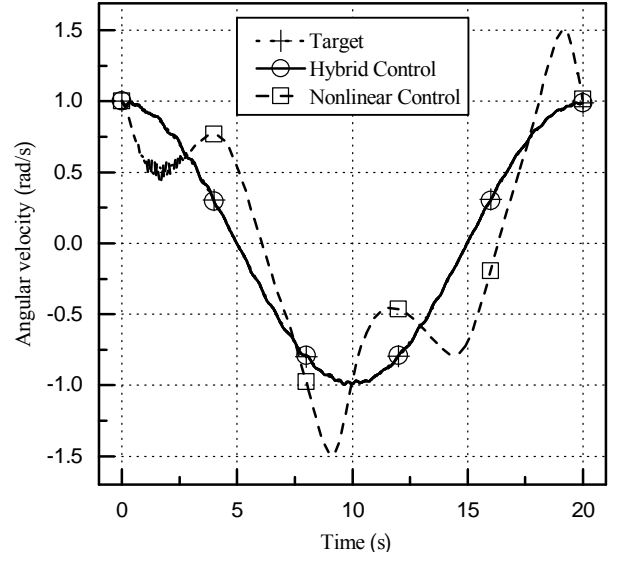


Fig. 13 Evolution of the angular velocity  $\dot{\theta}_2$

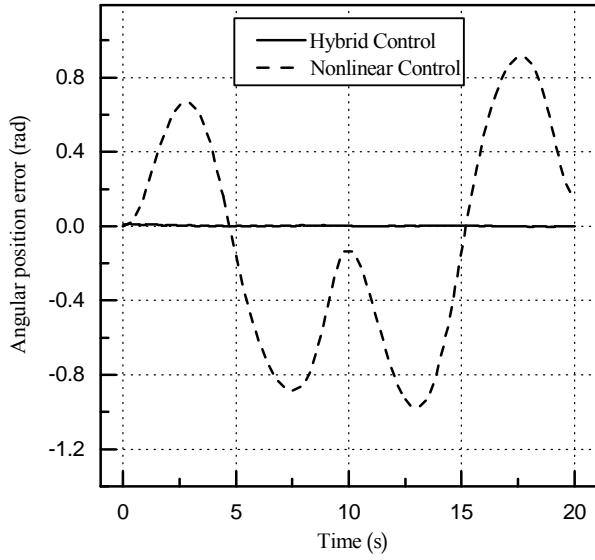


Fig. 11 Evolution of the angular position error  $\tilde{\theta}_2$

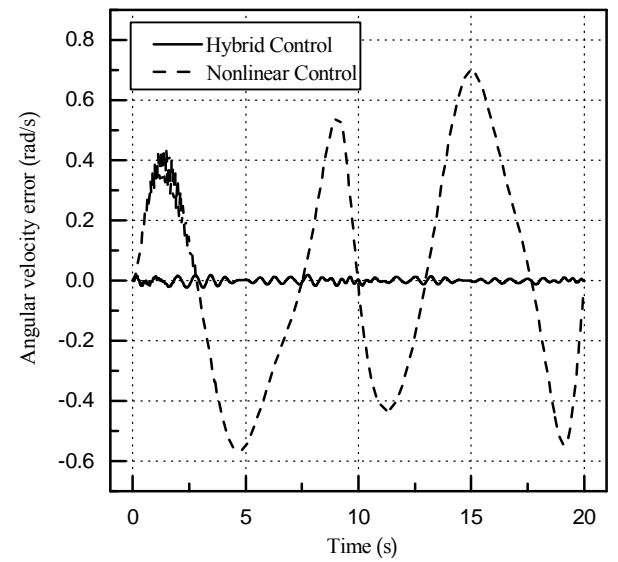


Fig. 14 Evolution of the angular velocity error  $\dot{\tilde{\theta}}_2$

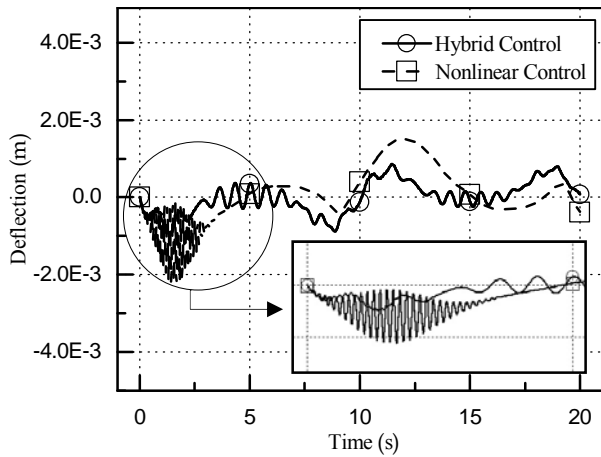


Fig. 12 Evolution of the deflection  $f_2$

**Table 3** Trajectory error on joint 2

Variable	Max. Error		Root Mean Square	
	$\theta_2$ (rad)	$\dot{\theta}_2$ (rad/s)	$\theta_2$ (rad)	$\dot{\theta}_2$ (rad/s)
Hybrid Control	1.17 $\times 10^{-2}$	2.35 $\times 10^{-2}$	3.43 $\times 10^{-3}$	8.40 $\times 10^{-3}$
Nonlinear Control	9.77 $\times 10^{-1}$	6.96 $\times 10^{-1}$	6.07 $\times 10^{-1}$	3.71 $\times 10^{-1}$

## 8 Conclusion

Our goal is to search for control techniques that improve the performance of flexible manipulators on the dynamic phase of the trajectory and reduce the computation burden.

The main idea here is to combine two control techniques, nonlinear control and neural network control.

The new control strategy presented is composed of two controllers. A static feed forward artificial neural network controller and an adaptive CMAC neural network controller.

The first controller is elaborated by approximating the robot's dynamic equation of motion with an artificial neural network and adding a proportional and derivative term. The aim of the first controller is to provide a stable and fast control, based on the dynamic model of the system.

While the first controller provides the main of the control, the adaptive CMAC neural network strategy ensure that the real trajectory matches the desired one by compensating errors due to structured and unstructured uncertainties, increasing the precision of the control.

Using ANNs in the place of the nonlinear model allowed us to simplify the structure of the controller reducing its computation time and enhancing its reactivity.

Simulation results on a dynamic trajectory with a sinusoidal form show the effectiveness of the proposed control strategy.

### References:

- [1] A. R. Maouche and M. Attari, "Adaptive Cerebellar Model Articulation Controller-Nonlinear Control System for Flexible Link Manipulator," *Journal of Vibration and Control*, vol. 19(14), pp. 2109-2123, Sage publications, August 2012.
- [2] A. R. Maouche and M. Attari, "CMAC Based Adaptive Control of a Flexible Link Manipulator," *IEEE Proceedings of the 35th International Conference on Industrial Electronics CONTROL (IECON)*, pp. 1480-1485, Porto, Portugal, November 2009.
- [3] W. J. Book, "Modeling, design, and control of flexible manipulator arms: A tutorial review," *IEEE Proceedings of the 29th Conference on Decision and Control*, pp. 500-506, 1990.
- [4] P. Chedmail, "Synthèse de robots et de sites robotisés. Modélisation de robots souples," *PhD Dissertation*, ENSM Nantes, France, 1990.
- [5] Z. Mohamed, J. M. Martins, M. O. Tokhi, J. Sa Da Costa, M. A. Botto, "Vibration control of a very flexible manipulator system," *Control Engineering Practice*, vol. 13, pp. 267-277, 2005.
- [6] A. K. M. Azad, M. O. Tokhi, N. Anand, "Teaching of control for complex systems through simulation," *Proceedings of the 2003 ASEE/WFEO International Colloquium, American Society for Engineering Education*, 2003.
- [7] F. Khorrarni, S. Jain and A. Tzes, "Experimental results on adaptive nonlinear control and input preshaping for multi-link flexible manipulators," *Automatica*, vol. 31, pp. 83-97, 1995.
- [8] Y. J. Lin and T. S. Lee, "Comprehensive dynamic modeling and motion/force control of flexible manipulators," *Mechatronics*, vol. 2, pp. 129-148, 1992.
- [9] K. L. Hillsley and S. Yurkovitch, "Vibration control of two link flexible robot arm," *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 2121-2126, 1991.
- [10] M. Ushiyama and A. Konno, "Computed acceleration control for the vibration suppression of flexible robotic manipulators," *Proceedings of the 5<sup>th</sup> International Conference on Advanced Robotics*, pp. 126-131, 1991.
- [11] W. J. Book, O. Maizza-Neto and D. E. Whitney, "Feedback control of two beam, two joint systems with distributed flexibility," *Journal of Dynamic Systems, Measurement, and Control*, vol. 97(4), pp. 424-431, 1975.
- [12] P. Chedmail and W. Khalil, "Non linear decoupling control of flexible robots," *Proceedings of the 4<sup>th</sup> International Conference on Advanced Robotics*, pp. 138-145, 1989.
- [13] J. De Shutter, H. Van Brussels, M. Adam, A. Froment and J. L. Faillot, "Control of flexible robots using generalized non-linear decoupling," *Proceedings of IFAC*, pp. 113-118, 1988.
- [14] A. A. El-Badawy, M. W. Mehrez and A. R. Ali, "Nonlinear modeling and control of flexible-link manipulators subjected to parametric excitation," *Nonlinear Dynamics*, vol. 62, pp. 769-779, Springer, 2010.
- [15] M. W. Spong, "Adaptive control of flexible joint manipulators: comments on two papers," *Automatica*, vol. 31, pp. 585-590, 1995.

- [16] B. Siciliano and W. J. Book, "A singular perturbation approach to control of lightweight flexible manipulators," *The International Journal of Robotics Research*, vol. 7, pp. 79-90, 1988.
- [17] J.H. Ryu, D.S. Kwon, and B. Hannaford, "Control of a flexible manipulator with non collocated feedback: Time-domain passivity approach," *IEEE Transaction on Robotics*, vol. 20(4), pp. 776-780, 2004.
- [18] Y. Xu and E. Ritz, "Vision based flexible beam tip point control," *IEEE Transaction on Control Systems Technology*, vol. 17(5), pp. 1220-1227, 2011.
- [19] N. G. Chalhoub and G. A. Kfoury, "Development of a robust nonlinear observer for a single-link flexible manipulator," *Nonlinear Dynamics*, vol. 39, pp. 217-233, Springer, 2005.
- [20] A. R. Maouche and H. Meddahi, "Adaptive neural control of a two link flexible manipulator", *WSEAS proceedings of the 9<sup>th</sup> International Conference on Computer Engineering and Applications (CEA'15)*, pp 247-254, February 2015.
- [21] P. Kuo, A. Hosein and M. S. Farmanborda, "Nonlinear Output Feedback Control of a Flexible Link Using Adaptive Neural Network: Controller Design," *Journal of Vibration and Control*, vol. 19(11), pp. 1690-1708, Sage publications, 2013.
- [22] T. M. Mansour and S. Kazi, "Fuzzy logic modified proportional-integral-derivative (MPID) control for flexible manipulator," *NAUN International Journal of Circuits, Systems and Signal Processing*, vol. 7(5), pp. 285-293, 2013.
- [23] A. R. Maouche, "Intelligent Control," In: Casolo, F. (eds.) *Motion Control*, pp 31-50, In-Teh Publication, Croatia, 2010. ISBN 978-953-7619-55-8.
- [24] A. R. Maouche and M. Attari, "Nonlinear adaptive RBFNN control of a one-link flexible manipulator," *IEEE Proceedings of the 1<sup>st</sup> International Conference on Machine and Web Intelligence*, pp. 144-149, 2010.
- [25] G. G. Rigatos, "Model-Based and Model-Free Control of Flexible-Link Robots: A Comparison Between Representative Methods," *Applied Mathematical Modelling*, pp. 3906-3925, 2009.
- [26] A.R. Maouche and M. Attari, "Hybrid adaptive neural control for flexible manipulators," *International Journal of Intelligent Systems Technologies and Applications*, vol. 7(4), pp. 396-413. Inderscience Ltd, Switzerland, 2009.
- [27] M. O. Tokhi and A. K. M. Azad, "Flexible Robot Manipulators: Modelling, Simulation and Control," *Institution of Engineering and Technology*, UK, 2008. ISBN: 978-0-86341-448-0.
- [28] S. Kamalasadan and A. A. Ghandakly, "A neural network parallel adaptive controller for dynamic system control," *IEEE Transaction on Instrumentation and Measurement*, vol. 56(5), pp. 1786-1796, 2007.
- [29] F. M. Caswara and H. Unbehauen, "A neurofuzzy approach to the control of a flexible-link manipulator," *IEEE Transaction on Robotics and Automation*, vol. 34(3), pp. 406-420, 2004.
- [30] L. Tian and C. Collins, "A dynamic recurrent neural network-based controller for a rigid-flexible manipulator system," *Mechatronics*, vol. 14, pp. 471-490, 2004.
- [31] Y. Yang, G. Feng and J. Ren, "A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 34(3), pp. 406-420, 2004.
- [32] X. P. Cheng and R. V. Patel, "Neural network based tracking control of a flexible macro-micro manipulator system," *Neural Networks*, vol. 16, pp. 271-286, 2003.
- [33] M.H. Elahinia, and H. Ashrafioun, "Nonlinear control of a shape memory alloy actuated manipulator," *ASME Journal Of Vibration Acoustic*, vol. 123(4), pp. 487-495, 2001.
- [34] V. Giurgiutiu, F. Jichi, J. Berman and J. M. Kamphaus, "Theoretical and experimental investigation of magnetostrictive composite beams," *Smart Material Structure*, vol. 10(5), pp. 934-935, 2001.
- [35] J. Leng and A. Asundi, "Active vibration control system of smart structures based on FOS and ER actuator," *Smart Material Structure*, vol. 8(2), pp. 252-256, 1999.
- [36] J.C.P. Reis and J. Sa da Costa, "Design and Experimental Control of an Active-Flexible Tubular Manipulator Link," *Intelligent Material Systems and Structures*, pp. 1741-1757, 2012.
- [37] J. Shan, H. T. Liu, D. Sun, "Slewing and vibration control of a single-link flexible manipulator by positive feedback (PPF)," *Mechatronics*, vol. 15, pp.487-503, 2005.
- [38] D. Sun, J. K. Mills, J. J. Shan and S. K. Tso, "A PZT actuator control of a single-link flexible

- manipulator based on linear velocity feedback and actuator placement," *Mechatronics*, vol. 14(4), pp. 381-401, 2004.
- [39] H. C. Shin and S. B. Choi, "Position control of a two-link flexible manipulator featuring piezoelectric actuators and sensors," *Mechatronics*, vol. 11(6), pp. 707-729, 2001.
- [40] W. A. El-Ganaini, N. A. Saeed and M. Eissa, "Positive position feedback (PPF) controller for suppression of nonlinear system vibration," *Nonlinear Dynamics*, vol. 72, pp. 517-537, Springer, 2013.
- [41] H. Mohd Yatim and I. Z. Mat Darus, "Self-tuning active vibration controller using particle swarm optimisation for flexible manipulator system," *WSEAS Transactions on Systems and Control*, vol. 9, pp. 55-66, 2014.
- [42] R. C. Hibbeler, "Structural Analysis," *Prentice Hall*, New Jersey, USA, 2002. ISBN: 0-13-041825-0.
- [43] J. C. Ower and J. Van de Vegt, "Classical control design for a flexible manipulator: modeling and control system design," *IEEE Journal of Robotics and Automation*, vol. 3(5), pp. 485-489, 1987.
- [44] J.J.E. Slotine, and W. Lie, "On the adaptive control of robot manipulators," *International Newspaper of Robotics Research*, vol. 6, pp. 49-59, 1990.
- [45] C.M. Pham, W. Khalil, and C. Chevallereau, "A Nonlinear Model-Based Control of Flexible Robots," *Robotica*, vol. 11(1), pp. 73-82, 1993.
- [46] T. R. Kurfess, "Robotics and Automation Handbook," *CRC Press LLC*, Boca Raton USA, 2005. ISBN 0-8493-1804-1.
- [47] F. Kreith, "Mechanical engineering handbook," *CRC Press LLC*, USA, 1999.
- [48] J.S. Albus, "A new approach to manipulator control: the Cerebellar Model Articulation Controller (CMAC)," *ASME Journal of Dynamic System, Measurement and Control*, vol. 97, pp. 220-227, 1975.
- [49] W.T. Miller, R.P. Hewes, F.H. Glanz, and L.G. Kraft, "CMAC: an associative neural network alternative to backpropagation," *Proceedings of the IEEE, Special Issue on Neural Networks*, II, vol. 78, pp. 1561-1567, 1990.