CMAC Based Adaptive Control of a Flexible Link Manipulator

Amin Riad Maouche*, and Mokhtar Attari**
* ** Laboratory of Instrumentation, Faculty of Electronic and Computer Science, Houari Boumediene University, Algiers, Algeria
* armaouche@umbb.dz, ** attari.mo@gmail.com

Abstract—This paper describes a hybrid approach to the problem of controlling flexible link manipulators in the face of both structured and unstructured uncertainties. First, a nonlinear controller based on the equations of motion of the robot is elaborated. Its aim is to produce a stable control. Then, an adaptive CMAC neural controller is implemented to compensate structured and unstructured uncertainties. Efficiency of the new controller obtained by combining the two control laws is tested facing an important variation of the dynamic parameters of the flexible manipulator and compared to a classical nonlinear controller. Simulation results show the effectiveness of the proposed control strategy.

Index Terms—Adaptive control, CMAC neural network, nonlinear control, flexible-link manipulator.

I. INTRODUCTION

This article presents a novel control system structure which combines two types of control. A nonlinear control law based on the dynamic motion equation of the robot and an adaptive controller based on The Cerebellar Model Articulation Controller (CMAC) neural network.

We use here the concept of semi-physical modeling or “gray-box”. This technique is intended to combine the best of two worlds: knowledge-based modeling, whereby mathematical equations are derived in order to describe a process, based on a physical analysis, and black-box modeling, whereby a parameterized model is designed, whose parameters are estimated solely from measurements made on the process.

The 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 [1].

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 [2]. 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 sigmoidal output functions are involved. These properties motivate us to use a CMAC based controller in real-time control applications and especially adaptive control.

The demand for increased productivity in industry has led to the use of lighter robots with faster response and lower energy consumption. Flexible-link manipulator systems have relatively smaller actuators, higher payload to weight ratio and, in general, 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. However, most of the control techniques for flexible-link manipulators are inspired by classical controls [3]-[7]. On other hand, much research effort has been put into the design of neural network application for robotic control [8]-[10]. 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) [11], Magnetorheological (MR) materials, Electrorheological (ER) materials, Piezoelectric transducers (PZT) [12], etc.

The presented control law has several distinguished advantages. It is easy to compute and maximizes the control performance guaranteeing good precision when regulating the tip position of the flexible arm in the presence of large time-varying payload and parameter uncertainties.

A flexible manipulator is a system with distributed parameters and is 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 performances of the control. Choosing an adaptive control, 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.

The reminder of this paper is organized as follows. In Section 2, a planar flexible-link manipulator is modeled according to Euler-Lagrange’s formulation and finite element method for the discretisation. Section 3 presents the nonlinear controller. Stability and robustness analysis of this control method is carried out which ends in the development of the adaptive CMAC controller presented in Section 4. Section 5 describes simulation results and shows the efficiency of the proposed control strategy facing an important variation of the dynamic parameters. Finally, conclusion is presented in Section 6.
II. DYNAMIC MODELING

The system considered here consists of a one link flexible manipulator constrained to move in the horizontal plane. The link is composed of a flexible beam cantilevered onto a rigid rotating joint. The manipulator is actuated by a DC motor with a torque input at the hub as shown in Fig. 1. It is assumed that the link can be bent freely in the horizontal plane but is stiff in the vertical bending and torsion (no gravity effect). Thus, the Euler-Bernoulli beam theory is sufficient to describe the flexural motion of the link. The 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, \( \{O_1 \bar{x}_1 \bar{y}_1\} \) is the moving coordinate frame with origin at the hub of the link. \( \theta \) is the revolving angle at the hub of the link. \( f \) and \( \alpha \) are the elastic displacements, they describe the deflection and the section rotation at the tip of the link.

Motion of the manipulator’s arm is described by one rigid and two elastic variables:

\[
q = [q_f \ q_\alpha]^T
\]

where \( q_f = [\theta] \) and \( q_\alpha = [f, \alpha]^T \).

Torque applied to the manipulator’s joint is given by:

\[
\Gamma = [\Gamma]
\]

Let us consider an arbitrary point \( M \) on the link. The kinetic energy of the link \( T_L \) is then given by:

\[
T_L = \frac{1}{2} \rho \int_0^L \int_0^S V(M)^2 \, ds \, dx
\]

where, \( V(M) \) is the velocity of \( M \) on the flexible link. \( L \), \( S \) and \( \rho \) are the length, the section and the mass density of the link, respectively.

Now, the total kinetic energy \( T \) can be written as [13]:

\[
T = \frac{1}{2} J_A \dot{\theta}^2 + \frac{1}{2} J_B (\dot{\theta} + \dot{\alpha})^2 + \frac{1}{2} \rho c V(C)^2 + T_L
\]

where, \( J_A \) and \( J_B \) are, mass moment of inertia at the origin and at the end of the link, respectively.

Note that the first term on the right-hand side in (4) is due to moment of inertia of the portion of the mass of the actuator relative to the link. The second term is due to moment of inertia of the mass at the end of the link. The third term is the kinetic energy of the mass at \( C \) (payload). The forth term is the kinetic energy of the flexible link.

The potential energy \( U \) can be written as:

\[
U = \frac{1}{2} q^T K q
\]

with, \( K = \begin{bmatrix} 0 & 0 \\ 0 & K_e \end{bmatrix} \) and \( K_e = \begin{bmatrix} 12EI/L^3 & -6EI/L^2 \\ -6EI/L^2 & 4EI/L \end{bmatrix} \).

The term on the right-hand side in (5) describes the potential energy due to elastic deformation of the link. Note that the term relative to the gravity is not present here as the manipulator moves on a horizontal plane. \( K \) is the stiffness matrix. The first row and column of \( K \) is zero as \( U \) does not depend on \( q_f \). \( E \) is the Young modulus and \( I \) the quadratic moment of section of the link.

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

\[
\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} = \Gamma_i \quad (i = 1, 2)
\]

Substituting (4) and (5) into (6a) and (6b) yields to:

\[
L_q \Gamma = A(q) \ddot{q} + B(q) [\dot{q}] + C(q) [\dot{q}^2] + K q
\]

where \( A(q) \) is the \((n \times n)\) inertia matrix, \( B(q) \) is the \((n \times (n^2 - n)/2)\) matrix of Coriolis terms and \([\dot{q}]\) is an \((n \times n)\) vector of joint velocity products given by:

\[
[q_1 \dot{q_2}, \ q_2 \dot{q_3}, \ q_3 \dot{q_4}, \ldots, \ q_{n-1} \dot{q_n}]^T
\]

\( C(q) \) is the \((n \times n)\) matrix of centrifugal terms and \([\dot{q}^2]\) is an \((n \times 1)\) vector given by:

\[
[q_1^2, \ q_2^2, \ldots, \ q_n^2]^T
\]

where \( \Gamma \) is the Lagrangian function and \( \ell = T - U \).

If we suppose known the length \( L \) of the link, we define [13]:

\[
X = [J_A + J_B, J_B, \rho c I, M, E I]^T
\]

where \( X \) represents the vector of dynamic parameters of the flexible-link manipulator.
III. NONLINEAR CONTROL

This control is a generalization of the classically known 'computed torque' used to control rigid manipulator [14], [15]. 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-link manipulator. Let:

\[
h(q, \dot{q}) = B(q)\dot{q} + C(q)\dot{q}^2
\]

then, the dynamic equation of motion (7) can be reduced to the following:

\[
L_r\Gamma = A(q)\dot{q} + h(q, \dot{q})\dot{q} + Kq
\]

or even,

\[
\begin{bmatrix}
\dot{q}_r \\
\dot{q}_v \\
\end{bmatrix} = 
\begin{bmatrix}
A_r & A_{re} \\
A_{er} & A_e
\end{bmatrix}
\begin{bmatrix}
\dot{q}_r \\
\dot{q}_e
\end{bmatrix} + 
\begin{bmatrix}
h_r & h_{re} \\
h_{er} & h_e
\end{bmatrix}
\begin{bmatrix}
q_r \\
q_e
\end{bmatrix} + 
\begin{bmatrix}
0 & 0 \\
0 & K_e
\end{bmatrix}
\begin{bmatrix}
q_r \\
q_e
\end{bmatrix}
\]

(11)

We deduce from (11) that:

\[
\Gamma = A_r \dot{q}_r + h_r \dot{q}_r + A_{re} \dot{q}_e + h_{re} \dot{q}_e
\]

(11a)

\[
0 = A_{er} \dot{q}_r + h_{er} \dot{q}_r + A_e \dot{q}_e + h_e \dot{q}_e + K_e q_e
\]

(11b)

We propose then to use the following control law:

\[
\Gamma_{NL} = A_r \dot{q}_r + h_r \dot{q}_r + A_{pr} \dot{q}_r + K_{pr} \dot{q}_r + K_{vr} \dot{q}_r
\]

(12)

where \( \dot{q}_r, \dot{q}_v \) and \( \dot{q}_d \) define the desired angular trajectory. \( \dot{q}_r = q_r \) and \( \dot{q}_d = q_d \) are angular position and velocity errors. \( K_{pr} \) and \( K_{vr} \) are positive gain constants.

If we consider the ideal case where no errors are made while evaluating the dynamic parameters \( \dot{X} \), a Lyapunov stability analysis of the control law is presented on Appendix.

IV. ADAPTIVE CMAC NEURAL CONTROL

Let us consider now the case where the estimated parameters \( \dot{\dot{X}} \) used in the nonlinear control are different from the actual parameters \( \dot{X} \) 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.

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

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

\[
\Gamma = \Gamma_{NL} + \Gamma_{NN}
\]

(13)

where \( \Gamma \) is the overall controller’s output (torque); \( \Gamma_{NL} \) is the nonlinear controller’s output as defined in (12); \( \Gamma_{NN} \) is the CMAC adaptive neural controller’s output.

The Cerebellar Model Articulation Controller (CMAC) network was first developed by Albus [16] to approximate the information processing characteristics of the human cerebellum. Miller [2] later developed a practical implementation of the CMAC neural network that could be applied to real-time control applications.

The artificial neural network and especially 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.

The 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 effort. The use of generalization speeds up learning because a group of memory cells that are close is updated in each control cycle. In general, 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.

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.

![Fig. 2. The overall control system](image-url)
In a CMAC network a nonlinear function \( y = f(x) \) is approximated by using two primary mappings: \( S: Q \rightarrow A \) and \( P: A \rightarrow D \). Here, \( Q \) is a 2-dimensional input space corresponding to the angular position and velocity, \( A \) is a \( N_L \)-dimensional association space, and \( D \) is a 1-dimensional output space corresponding to the CMAC adaptive neural controller’s torque \( \Gamma_{NN} \).

The function \( S(x) \) maps each point \( x \) in the input space onto an association vector \( a = S(x) \in A \) that has \( N_L \) nonzero elements (\( N_L < N_A \)).

For a conventional CMAC, the association vector contains only binary elements, either zero or one. The function \( P(a) \) computes a scalar output \( y \) by projecting the association vector onto a vector \( W \) of adjustable weights so that the scalar output \( y \) can be obtained by evaluating the inner product of the two vectors \( a \) and \( W \). Then the actual output \( y \) is derived as follows:

\[
y = \Gamma_{NN} = P(a) = a^T W = \sum_{j=1}^{N_L} a_j \cdot W_j
\] (14)

where \( a_j \) represents the \( j \)th element of the association vector and \( W_j \) the \( j \)th element of the weight vector.

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. The errors between the joint’s desired and actual position/velocity values are then used to train online the CMAC neural controller. Training is made online and the weight adjustment \( \Delta W \) is given by:

\[
\Delta W = \beta \cdot (y^d - y) / N_L = \beta \cdot (K_{pp} \hat{u}_r + K_{vn} \hat{v}_r) / N_L
\] (15)

where \( y^d \) and \( y \) are the desired and actual output of the CMAC neural network. \( K_{pp} \) and \( K_{vn} \) are positive gain constants. \( \beta \) is the learning rate.

V. SIMULATION ANALYSIS

Performance of the adaptive controller is tested using a dynamic trajectory having the following form:

\[
\theta^d(t) = \begin{cases} 
16\pi / T^2 & \text{for } t \in [0, T / 4] \\
-16\pi / T^2 & \text{for } t \in [T / 4, 3T / 4] \\
16\pi / T^2 & \text{for } t \in [3T / 4, T] 
\end{cases}
\] (16)

with \( \theta^d(0) = \theta^d(0) = 0 \). To avoid the destabilization of the control law induced by fast dynamics, we choose \( T = 30 \) s. CMAC parameters are as follows: \( N_L = 32, Q_L = 0.5, \beta = 0.9, K_{pp} = 0.8 \) and \( K_{vn} = 4.8 \). The gain constants in the nonlinear control are \( K_{pr} = 1 \) and \( K_{vr} = 4 \).

Let suppose that the actual values of the parameters of the robot are such as specified in Table I.

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 such that:

\[
\dot{\mathbf{X}} = \mathbf{X} / 100
\] (17)

Then, we use these values \( \dot{\mathbf{X}} \) to compute \( \mathbf{A}_r \) and \( \mathbf{h}_r \). This will drive the first controller to produce an incorrect error. 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 overall adaptive CMAC controller we will compare its results with the nonlinear controller given in (12).

Fig. 3 to Fig. 7 illustrate the results obtained with the adaptive CMAC neural controller applied to the flexible-link manipulator. These figures describe the evolution of: angular position, error on position, deflection, angular velocity and error on the angular velocity, respectively. Results of the nonlinear controller are reported in dashed line for comparison. The desired trajectory (target) is reported on Fig. 3 and Fig. 6, in dotted line.

Table II presents 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.

The desired trajectory imposes a fast change of acceleration on the moment \( t = T/4 = 7.5 \) s and \( t = 3T/4 = 22.5 \) s. This radical change from a positive to a negative acceleration for the first moment and from a negative to a positive acceleration for the second one stresses the control. We can see its impact on the control of the angular velocity in Fig. 6, Fig. 7 and Table II. The velocity tracking with the adaptive neural controller is good and the error induced is acceptable, whereas the nonlinear controller strongly deviates from the target.

The proposed controller deals well with the elasticity of the link as the deflection is contained (see Fig. 5).

For the position control (see Fig. 3 and Fig. 4) we notice that the angular trajectory obtained with the adaptive CMAC controller matches perfectly the target, with an error lower than 0.023 rad, whereas it reaches 0.19 rad with the nonlinear controller (see Table II).

This demonstrates the good performance of the proposed adaptive CMAC neural control.

<table>
<thead>
<tr>
<th>MANIPULATOR CHARACTERISTICS</th>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical parameters</strong></td>
<td><strong>Link</strong></td>
</tr>
<tr>
<td>Length (m)</td>
<td>( L = 1.0 )</td>
</tr>
<tr>
<td>Moment of inertia at the origin of the link (kg m²)</td>
<td>( J_z = 1.80 \times 10^{-3} )</td>
</tr>
<tr>
<td>Moment of inertia at the end of the link (kg m²)</td>
<td>( J_x = 4.70 \times 10^{-2} )</td>
</tr>
<tr>
<td>Mass of the link (kg)</td>
<td>( m = 1.26 )</td>
</tr>
<tr>
<td>Mass at the tip (kg)</td>
<td>( m_r = 5.5 )</td>
</tr>
<tr>
<td>Mass density (kg/m³)</td>
<td>( \rho = 7860 )</td>
</tr>
<tr>
<td>Young modulus</td>
<td>( E = 1.98 \times 10^{11} )</td>
</tr>
<tr>
<td>Quadratic moment of section (m³)</td>
<td>( I = 3.41 \times 10^{21} )</td>
</tr>
</tbody>
</table>
Fig. 3. Evolution of the angular position $\theta$ (rad)

Fig. 4. Evolution of the angular position error $\theta$ (rad)

Fig. 5. Evolution of the deflection $f$ (m)

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

Fig. 7. Evolution of the angular velocity error $\dot{\theta}$ (rad/s)

TABLE II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Max. Error</th>
<th>Root Mean Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive CMAC Control</td>
<td>2.21 $\times 10^{-2}$</td>
<td>1.06 $\times 10^{-2}$</td>
</tr>
<tr>
<td>Nonlinear Control</td>
<td>4.31 $\times 10^{-1}$</td>
<td>2.98 $\times 10^{-1}$</td>
</tr>
</tbody>
</table>
VI. CONCLUSION

In this paper, a new adaptive CMAC neural control strategy was presented. This controller is a combination of two controllers. A nonlinear controller and an adaptive CMAC neural network based controller.

The nonlinear control used has a better generalization than the classic neural network control or other statistical learning based control, as it can assume different types of trajectory without the need of training the controller for it. This control uses knowledge-based modeling, whereby mathematical equations are derived from physical analysis in order to describe the behavior of the manipulator. The use of this physical information in the controller, guarantees in our sense that the actual trajectory is, virtually, always close to the target.

While the nonlinear law 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 uncertainty, increasing the precision of the control.

Simulation results have shown the robustness in performance of this control design scheme against adverse effects such as model parameter variations.

Our future work will include experimental tests and the exploitation of sliding mode control to improve stability and performances of the proposed control strategy.

APPENDIX

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

\[
A_r \ddot{q}_e + A_e \ddot{q}_e + b_r \dot{q}_e + h_r \dot{q}_e + K_{pr} \dot{q}_e + K_{vr} \dot{q}_e = 0 \quad (A.1)
\]

with, \(q_e = 0 - \dot{q}_e = -\dot{q}_e\) and \(\ddot{q}_e = 0 - \ddot{q}_e = -\ddot{q}_e\) representing the elastic stabilization errors.

In addition, rewriting the coupling equation (11b) according to the trajectory and the elastic stabilization error variables (\(q_e\), \(\dot{q}_e\), \(\ddot{q}_e\)) gives:

\[
A_e \dddot{q}_e + A_e \ddot{q}_e + b_e \dot{q}_e + h_e \dot{q}_e + K_e \dddot{q}_e = 0 \quad (A.2)
\]

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

\[
A \dddot{q} + b \dddot{q} + K_p \dddot{q} + K_v \dddot{q} + s_1 = 0 \quad (A.3)
\]

where the positive constant matrices \(K_p\), \(K_v\) are

\[
K_p = \begin{bmatrix} K_{pr} & 0 \\ 0 & K_e \end{bmatrix}, \quad K_v = \begin{bmatrix} K_{vr} & 0 \\ 0 & 0 \end{bmatrix},
\]

respectively, and

\[
s_1 = \begin{bmatrix} A_e \dot{q}_e \dot{q}_e + b_e \dot{q}_e \dot{q}_e + h_e \dot{q}_e \dot{q}_e + K_e \dot{q}_e \dot{q}_e \end{bmatrix}.
\]

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

\[
V = \frac{1}{2} \dddot{q}^T A \dddot{q} + \frac{1}{2} \dddot{q}^T K_p \dddot{q} \quad (A.4)
\]

By differentiating \(V\) and using (A.3), we obtain:

\[
V = \dot{\dddot{q}}^T \left( \frac{1}{2} A - h \right) \dddot{q} - \frac{1}{2} \dddot{q}^T \left( K_e \dddot{q} + s_1 \right) \quad (A.5)
\]

The property of passivity of the flexible robot implies that \(\frac{1}{2} A - h\) is skew symmetric, finally we have:

\[
V = -\dddot{q}_r^T K_v \dddot{q}_r + \dddot{q}_r^T \left( A_e \dot{q}_e + h_e \dot{q}_e \right) \quad (A.6)
\]

The Lyapunov second method provides that the asymptotic stability of the control is assured if the following conditions are met. \(V\) is strictly positive everywhere except in \(\dddot{q} = 0\) where it is \(0\) and \(V\) is strictly negative everywhere except in \(\dddot{q} = 0\) where it is \(0\). These conditions are always met if the desired angular velocity and acceleration are not too significant for a given tuning of \(K_{pr}\), so that \(V\) remains essentially negative to ensure the control stability.

REFERENCES


