

Neural Networks-based Adaptive State Feedback Control of Robot Manipulators

Ghania Debbache, Abdelhak Bennia, Noureddine Goléa

Abstract: This paper proposes an adaptive control suitable for motion control of robot manipulators with structured and unstructured uncertainties. In order to design an adaptive robust controller, with the ability to compensate these uncertainties, we use neural networks (NN) that have the capability to approximate any nonlinear function over a compact space. In the proposed control scheme, we need not derive the linear formulation of robot dynamic equation and tune the parameters. To reduce the NNs complexity, we consider the properties of robot dynamics and the decomposition of the uncertainties terms. The proposed controller is robust against uncertainties and external disturbance. The validity of the control scheme is demonstrated by computer simulations on a two-link robot manipulator.

Keywords: Robot manipulator, neural networks, adaptive control, stability.

1 Introduction

Robot manipulators are multivariable nonlinear systems and are frequently subjected to structured and unstructured uncertainties even in a well-structured setting for industrial use. Structured uncertainties are mainly caused by imprecision in the manipulator link properties, unknown loads, and so on. Unstructured uncertainties are caused by unmodeled dynamics, such as, nonlinear friction, disturbances, and high-frequency dynamics. As a result, it is difficult to obtain an accurate mathematical model so that computed torque controllers [1-6] or other model-based controllers [7-8] can be accurately applied. Although adaptive controllers [1-5, 7] can achieve fine control and compensate for partially unknown manipulator dynamics (i.e., structured uncertainties), they often suffer from incapacity to deal with unstructured uncertainties. Hence, there is a need for model-free adaptive control strategies.

The application of neural networks to robots dynamic control is not new [9-11]. Though the proposed methods have been practically successful, it has proved extremely difficult to develop a general analysis and design theory for early NNs control systems. During the last few years, a number of papers have been presented to deal with the problem of robot adaptive control [12-14]. The basic idea of these methods is to design the feedback controller based on the computed torque principle, and to use an adaptive NN to approximate the robot nonlinearities needed in the control input design. However, most of the above designs present the drawback that, robot dynamic model is presented as single nonlinearity approximated by a single NN with the robot real and desired positions and velocities as inputs, which results in large NN with lot of parameters to be tuned.

In this paper, our goal is to develop a method for designing an adaptive NN control for rigid robot manipulators. A structured or partitioned NN structure, that simplifies the controller design and makes for faster weight tuning online, is designed to ensure the closed loop stability. Robust update laws are used to tune the NNs parameters, and to ensure their boundedness. Lyapunov stability theory is used to drive the stability conditions, and to show the robustness against uncertainties and disturbances. Simulation tests for a two-link robot, under uncertainties, disturbance and parameters variations, show the accuracy and the robustness of the proposed adaptive scheme.

2 Robot control problem

The Lagrange–Euler formulation, the dynamic equation of an n -joint robot arm can be expressed as

$$M(q)\ddot{q} + c(q, \dot{q}) + g(q) + \tau_c(q, \dot{q}) + \tau_d(q, \dot{q}) = u \quad (1)$$

where $M(q) \in R^{n \times n}$ bounded positive definite inertia matrix; $c(q, \dot{q}) \in R^n$ vector representing centrifugal and Coriolis effects; $g(q) \in R^n$ vector representing gravitational torques; $\tau_c(q, \dot{q}) \in R^n$, $\tau_d(q, \dot{q}) \in R^n$ vectors representing the dynamic effects as nonlinear frictions, small joint and link elasticities, backlash and bounded torque disturbances. Here the uncertainties effect is decomposed as continuous part $\tau_c(q, \dot{q})$ and discontinuous part $\tau_d(q, \dot{q})$. $u \in R^n$ vector of joint torques supplied by the actuators; $q \in R^n$ vector of joint positions; $\dot{q} \in R^n$ vector of joint velocities and $\ddot{q} \in R^n$ vector of joint accelerations.

Taking as state vector $x^T = [x_1^T \dots x_n^T]$ with $x_i^T = [q_i \ \dot{q}_i]$, the robot model (1) can be rewritten as

$$\dot{x} = Ax + B [F(q, \dot{q}) + G(q)u + d(q, \dot{q})] \quad (2)$$

where

$$F(q, \dot{q}) = \begin{bmatrix} f_1(q, \dot{q}) \\ \vdots \\ f_n(q, \dot{q}) \end{bmatrix} := -M^{-1}(q) [c(q, \dot{q}) + g(q) + \tau_c(q, \dot{q})]$$

$$G(q) = \begin{bmatrix} g_{11}(q) & \dots & g_{1n}(q) \\ \vdots & \ddots & \vdots \\ g_{n1}(q) & \dots & g_{nn}(q) \end{bmatrix} := M^{-1}(q)$$

$$d(q, \dot{q}) = \begin{bmatrix} d_1(q, \dot{q}) \\ \vdots \\ d_n(q, \dot{q}) \end{bmatrix} := -M^{-1}(q)\tau_d(q, \dot{q})$$

and $A = \text{diag}[A_1, \dots, A_n]$, $B = \text{diag}[b_1, \dots, b_n]$ with

$$A_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, i = 1..n$$

The control problem can be stated as: for a given bounded reference trajectories q_r , \dot{q}_r and $\ddot{q}_r \in R^n$ design the control input torques u such as the robot's states follow their references, with all involved signals in closed loop remain bounded.

3 Neural networks

The general function of one hidden layer feedforward neural network can be described as in (3) as the weighted combination of N activation functions. Here the input vector x and $\varphi_i(\cdot)$ represents the i th activation function (with its parameters vector θ_i) connected to the output by weight w_i .

$$y = \sum_{i=1}^N \varphi_i(x, \theta_i) w_i \quad (3)$$

The numbers of the input and output layers coincide with the dimension of the input vector and output information number, respectively. Since the above neural networks will be trained on line to achieve the

control task, and in order to reduce computation load, we will assume that the activation functions parameters θ_i are fixed, i.e., their number and shape is a priori determined. The only adjustable parameters are the weights w_i . Then, (3) can be rewritten in the compact form

$$y = w^T \phi(x) \quad (4)$$

where $\phi^T(x) = [\phi_1(x) \dots \phi_N(x)]$ and $w^T = [w_1 \dots w_N]$.

It is known from NNs approximation theory [15-19] that the modeling error can be reduced arbitrarily by increasing the number N , i.e., the number of the linear independent activation functions in the network. That is, a smooth function $f(x)$, $x \in \Omega_x \subset R^n$ can be written as

$$f(x) = w^{*T} \phi(x) + \varepsilon(x) \quad (5)$$

where $\varepsilon(x)$ is the network inherent approximation error, and w^* is an optimal weight vector.

Various well-known results, see e.g. [16-19], for various activation functions $\phi_i(\cdot)$, based, e.g. on the Stone-Weierstrass theorem, say that any sufficiently smooth function can be approximated by a suitably large NN [5-8]. The functional range of NN (4) is said to be dense, if for any $f(x)$ and a constant $\varepsilon^* > 0$ there exist finite N and w^* such that (5) holds with $|\varepsilon(x)| < \varepsilon^*$. The range of activation functions include for instance the step, the ramp, the sigmoid and radial basis functions. Several algorithms are proposed in the literature to select the structure and parameters for those kind of NNs, see e.g. [20-21].

4 Neural state feedback

Due to approximation property (5), we can assume that the nonlinear terms in (2) can be approximated as

$$\begin{aligned} f_i(q, \dot{q}) &= \theta_{f_i}^{*T} \phi_i(q, \dot{q}) + \varepsilon_i(q, \dot{q}) \\ g_{ii}(q) &= \theta_{g_i}^{*T} \psi_i(q) + \varepsilon_i(q) \end{aligned} \quad i = 1..n \quad (6)$$

where $\theta_{f_i}^{*T} \phi_i(q, \dot{q})$ and $\theta_{g_i}^{*T} \psi_i(q)$ are NNs of the form (4), and $\varepsilon_i(q, \dot{q})$, $\varepsilon_i(q)$ are the inherent approximation errors due to the finite size of the NNs. The optimal weights $\theta_{f_i}^*$ and $\theta_{g_i}^*$ defined above are quantities required only for analytical purpose. Typically $\theta_{f_i}^*$ and $\theta_{g_i}^*$ are chosen to minimize $\varepsilon_i(q, \dot{q})$ and $\varepsilon_i(q)$ over the compact regions Ω_f and Ω_g respectively, that is

$$\begin{aligned} \theta_{f_i}^* &= \arg \min_{\theta_{f_i}} \left\{ \sup_{q, \dot{q} \in \Omega_f} |f_i(q, \dot{q}) - \theta_{f_i}^T \phi_i(q, \dot{q})| \right\} \\ \theta_{g_i}^* &= \arg \min_{\theta_{g_i}} \left\{ \sup_{q \in \Omega_g} |g_{ii}(q) - \theta_{g_i}^T \psi_i(q)| \right\} \end{aligned}$$

Assumption 1: The neural networks approximation errors are bounded by $|\varepsilon_i(q, \dot{q})| \leq \varepsilon_{0i}$ and $|\varepsilon_i(q)| \leq \varepsilon_{0i}$, $i = 1..n$, for some constants ε_{0i} and ε_{0i} .

Assumption 1 results from the universal approximation property of neural networks, that can approximate any well-defined function over a compact space with finite approximation error.

Using (6) in (2), the robot dynamic can be written as

$$\dot{x} = Ax + B [\Theta_f^* \Phi(q, \dot{q}) + \Theta_g^* \Psi(q)u + H(q)u + \omega(q, \dot{q})] \quad (7)$$

where $\Phi(q, \dot{q}) = \text{block-diag}[\phi_1(q, \dot{q}), \dots, \phi_n(q, \dot{q})]$, $\Psi(q, \dot{q}) = \text{block-diag}[\psi_1(q, \dot{q}), \dots, \psi_n(q, \dot{q})]$, $\Theta_f^* = \text{block-}$

$\text{diag}[\theta_{f_1}^{*T}, \dots, \theta_{f_n}^{*T}]$, $\Theta_g^* = \text{block-diag}[\theta_{g_1}^{*T}, \dots, \theta_{g_n}^{*T}]$, $\omega(q, \dot{q}) = \varepsilon + d(q, \dot{q})$, with $\varepsilon^T = [\varepsilon_1 \ \dots \ \varepsilon_n]$, and

$$H(q) = \begin{bmatrix} \varepsilon_1 & g_{12}(q) & \dots & g_{1n}(q) \\ g_{21}(q) & \ddots & & \vdots \\ \vdots & & \ddots & g_{(n-1)n}(q) \\ g_{n1}(q) & \dots & g_{n(n-1)}(q) & \varepsilon_n \end{bmatrix}$$

Based on (7), the control inputs are defined as

$$u = [\Theta_g \Psi(q)]^{-1} [-\Theta_f \Phi(q, \dot{q}) + \ddot{q}_r + Ke] \quad (8)$$

where $e^T = [(q_r - q)^T \ (\dot{q}_r - \dot{q})^T]$ is the tracking error vector, Θ_g , Θ_f are the estimated neural networks parameters, and $K = \text{diag}[K_1, \dots, K_n]$ with $K_i \in R^2$ is PD gain vector, chosen such as the matrix $A_c = A - BK$ is Hurwitz.

Then, introducing the control input (8) in (7) yields

$$\dot{e} = A_c e - B [\tilde{\Theta}_f \Phi(q, \dot{q}) + \tilde{\Theta}_g \Psi(q)u + H(q)u + \omega(q, \dot{q})] \quad (9)$$

where $\tilde{\Theta}_f = \Theta_f^* - \Theta_f$ and $\tilde{\Theta}_g = \Theta_g^* - \Theta_g$ are the parameters estimation errors.

From (9), it can be seen that the tracking error vector is driven by the coupling terms and the finite approximation accuracy effects reflected by $H(q)$ and the uncertainty term $\omega(q, \dot{q})$.

To design the neural networks parameters update laws and to ensure boundedness of the involved signals in the closed loop robot control, the following assumptions are used:

Assumption 2: The diagonal elements of $G(q)$ are bounded such as $g_m \leq \text{diag}[g_{11}(q), \dots, g_{nn}(q)] \leq g_M$, the matrix $H(q)$ is bounded by $|H(q)| \leq h_0$, and the disturbance term $\omega(q, \dot{q})$ is bounded by $|\omega(q, \dot{q})| \leq \omega_0$.

Assumption 3: The neural networks parameters are bounded by the constraint sets Ω_f and Ω_g such that: $\Omega_f = \{\Theta_f \mid |\Theta_f| \leq f_M\}$ and $\Omega_g = \{\Theta_g \mid g_m \leq |\Theta_g| \leq g_M\}$, respectively, where f_M , g_m , and g_M are some known constants.

The first part of assumption 2 follows from the fact that $M(q)$ is bounded positive definite matrix, the second part follows from the boundedness of $M(q)$ and $\varepsilon_i(q)$. Finally, the third part follows from boundedness of $M(q)$, τ_d and $\varepsilon_i(q, \dot{q})$. The bounds used in assumption 3 result from the assumptions 1-2 and are used to ensure the boundedness of the neural networks outputs.

In order to constraint the parameters Θ_f and Θ_g within the sets Ω_f and Ω_g , respectively, we use the following parameter projection algorithm [22]:

$$\dot{\Theta}_f = \begin{cases} -\gamma_1 B^T P e \Phi^T(q, \dot{q}) & \text{if } |\Theta_f| < f_M \text{ or } (|\Theta_f| = f_M \\ & \text{and } \text{tr}[B^T P e \Phi^T(q, \dot{q}) \Theta_f^T] \geq 0) \\ -\gamma_1 B^T P e \Phi^T(q, \dot{q}) & \text{if } |\Theta_f| = f_M \text{ and} \\ +\gamma_1 \text{tr}[B^T P e \Phi^T(q, \dot{q}) \hat{\Theta}_f^T] \left(\frac{1+|\Theta_f|}{f_M}\right)^2 \Theta_f & \text{tr}[B^T P e \Phi^T(q, \dot{q}) \Theta_f^T] < 0 \end{cases} \quad (10)$$

and

$$\dot{\Theta}_g = \begin{cases} -\gamma_2 B^T P e u^T \Psi^T(q) & \text{if } |\hat{\Theta}_g| < g_M \text{ or } (|\hat{\Theta}_g| = g_M \\ & \text{and } \text{tr}[B^T P e u^T \Psi^T(q) \hat{\Theta}_g^T] \geq 0) \\ -\gamma_2 B^T P e u^T \Psi^T(q) & \text{if } |\Theta_g| = g_M \text{ and} \\ +\gamma_2 \text{tr}[B^T P e u^T \Psi^T(q) \Theta_g^T] \left(\frac{1+|\Theta_g|}{g_M}\right)^2 \Theta_g & \text{tr}[B^T P e u^T \Psi^T(q) \Theta_g^T] \leq 0 \end{cases} \quad (11)$$

where $\gamma_1, \gamma_2 > 0$ are design parameters, and $P = P^T > 0$ is the solution, for a given $Q = Q^T > 0$, of the Lyapunov equation

$$A_c^T P + P A_c = -Q \quad (12)$$

Moreover, in order to guarantee $|\Theta_g| \geq g_m$ such that an inverse of $\Theta_g \Psi(q)$ always exists, we use the following law to adjust the parameter Θ_g .

1. Whenever any element $[\Theta_g]_{ij} = g_m$ use

$$[\dot{\Theta}_g]_{ij} = \begin{cases} -\gamma_2 [B^T P e u^T \Psi^T(q)]_{ij} & \text{if } [B^T P e u^T \Psi^T(q)]_{ij} < 0 \\ 0 & \text{if } [B^T P e u^T \Psi^T(q)]_{ij} \geq 0 \end{cases} \quad (13)$$

2. Otherwise, use (11).

where $[A]_{ij}$ stands for the ij th element of the matrix A .

The stability properties of the proposed NNs adaptive state feedback are summarized by the following theorem.

Theorem 1: The robot adaptive control composed by the robot dynamic (2), the control input (8), the update laws (10)-(11) and (13) verifying assumptions 1-3, guarantees the following:

1. $|\Theta_f| \leq f_M$ and $g_m \leq |\Theta_g| \leq g_M$
2. $|e| \in L_\infty$
3. $|u| \in L_\infty$

Proof:

1. To prove $|\Theta_g| \leq g_M$, let $V_g = \frac{1}{2\gamma_2} \text{tr} [\Theta_g^T \Theta_g]$, then $\dot{V}_g = \frac{1}{\gamma_2} \text{tr} [\dot{\Theta}_g^T \Theta_g]$. If the first line of (11) is true, we have either $|\Theta_g| < g_M$ or $\dot{V}_g = -\text{tr} [(B^T P e u^T \Psi^T(q))^T \Theta_g] \leq 0$ when $|\Theta_g| = g_M$, that is, we get always $|\Theta_g| \leq g_M$. If the second line of (11) is true, we have $|\Theta_g| = g_M$ and

$$\begin{aligned} \dot{V}_g &= \text{tr} \left[- (B^T P e u^T \Psi^T(q))^T \Theta_g + \text{tr} [B^T P e u^T \Psi^T(q) \Theta_g^T] \left(\frac{1 + |\Theta_g|}{g_M} \right)^2 \Theta_g^T \Theta_g \right] \\ &= -\text{tr} \left[(B^T P e u^T \Psi^T(q))^T \Theta_g \right] + \text{tr} [B^T P e u^T \Psi^T(q) \Theta_g^T] \left(\frac{1 + |\Theta_g|}{g_M} \right)^2 \text{tr} [\Theta_g^T \Theta_g] \\ &= -\text{tr} \left[(B^T P e u^T \Psi^T(q))^T \Theta_g \right] + \text{tr} [B^T P e u^T \Psi^T(q) \Theta_g^T] \left(\frac{1 + |\Theta_g|}{g_M} \right) |\Theta_g|^2 \end{aligned} \quad (14)$$

since $|\Theta_g| = g_M$, we get

$$\dot{V}_g = \text{tr} [B^T P e u^T \Psi^T(q) \Theta_g^T] g_M^2 \leq 0 \quad (15)$$

that is, $|\Theta_g| \leq g_M$. Therefore, we have $|\Theta_g| \leq g_M, \forall t \geq 0$.

From (13) we see that if $|\Theta_g|_{ij} = g_m$, then $[\dot{\Theta}_g]_{ij} \geq 0$; that is, we have that $|\Theta_g|_{ij} \geq g_m$.

Using the same analysis, we can prove that $|\Theta_f| \leq f_M, \forall t \geq 0$.

2. Consider the Lyapunov function

$$V = \frac{1}{2}e^T P e + \frac{1}{2\gamma_1} \text{tr} \left[\tilde{\Theta}_f^T \tilde{\Theta}_f \right] + \frac{1}{2\gamma_2} \text{tr} \left[\tilde{\Theta}_g^T \tilde{\Theta}_g \right] \quad (16)$$

The differentiation of (16) along (9) yields

$$\begin{aligned} \dot{V} = & -\frac{1}{2}e^T Q e - e^T P B \left[\tilde{\Theta}_f \Phi(q, \dot{q}) + \tilde{\Theta}_g \Psi(q) u + H(q) u + \omega(q, \dot{q}) \right] \\ & + \frac{1}{\gamma_1} \text{tr} \left[\dot{\tilde{\Theta}}_f^T \tilde{\Theta}_f \right] + \frac{1}{\gamma_2} \text{tr} \left[\dot{\tilde{\Theta}}_g^T \tilde{\Theta}_g \right] \end{aligned} \quad (17)$$

which can be arranged as

$$\begin{aligned} \dot{V} = & -\frac{1}{2}e^T Q e - e^T P B \left[H(q) u + \omega(q, \dot{q}) \right] \\ & + \frac{1}{\gamma_1} \text{tr} \left[\left(\dot{\tilde{\Theta}}_f^T - \gamma_1 \Phi(q, \dot{q}) e^T P B \right) \tilde{\Theta}_f \right] + \frac{1}{\gamma_2} \text{tr} \left[\left(\dot{\tilde{\Theta}}_g^T - \gamma_2 \Psi(q) u e^T P B \right) \tilde{\Theta}_g \right] \end{aligned} \quad (18)$$

Then, using (10)-(11) and the fact that $\dot{\tilde{\Theta}}_f = -\dot{\Theta}_f$ ($\dot{\tilde{\Theta}}_g = -\dot{\Theta}_g$), one can show that the third and fourth terms in (18) are always ≤ 0 . Therefore, (18) can be written as

$$\dot{V} \leq -\frac{1}{2}e^T Q e - e^T P B \left[H(q) u + \omega(q, \dot{q}) \right] \quad (19)$$

Further, (19) can be upper bounded by

$$\dot{V} \leq -\frac{1}{2} \lambda_{\min}(Q) |e|^2 + |e| |PB| \left[|H(q)| |u| + |\omega(q, \dot{q})| \right] \quad (20)$$

Then, using (8) and the result in 1, the input torques can be upper bounded by

$$\begin{aligned} |u| \leq & \left| [\Theta_g \Psi(q)]^{-1} \right| \left[|\Theta_f \Phi(q, \dot{q})| + |\ddot{q}_r| + |K| |e| \right] \\ \leq & \frac{1}{g_m} (f_M + q_0 + |K| |e|) \end{aligned} \quad (21)$$

where q_0 is an upper bound on the desired accelerations \ddot{q}_r .

Then, using (21) and the assumptions 1-3 in (20), becomes

$$\dot{V} \leq -\frac{1}{2} \lambda_{\min}(Q) |e|^2 + \kappa_1 |e|^2 + \kappa_2 |e| \quad (22)$$

where $\kappa_1 = \frac{h_0}{g_m} |PBK|$ and $\kappa_2 = |PB| \left(\frac{h_0}{g_m} (f_M + q_0) + \omega_0 \right)$.

Hence, (22) can be arranged as

$$\dot{V} \leq -\frac{1}{2} (\lambda_{\min}(Q) - 2\kappa_1) |e|^2 + \kappa_2 |e| \quad (23)$$

If the matrix Q is chosen such as $\lambda_{\min}(Q) > 2\kappa_1$, then $\dot{V} \leq 0$ whenever the tracking error is outside the region given by

$$|e| \leq \frac{2\kappa_2}{(\lambda_{\min}(Q) - 2\kappa_1)} \quad (24)$$

which implies that $|e| \in L_\infty$.

3. Using the result (24) in (21) yields

$$|u| \leq \frac{1}{g_m} \left(f_M + q_0 + |K| \frac{2\kappa_2}{(\lambda_{\min}(Q) - 2\kappa_1)} \right) \quad (25)$$

this implies that $|u| \in L_\infty$.

Remarks:

1. Only the diagonal elements of $G(q)$ are estimated and used in the control inputs design. By doing this, we avoid the estimation of the coupling terms (considered here as disturbances) and the need to compute the inverse of the estimation of $G(q)$.
2. Although, the control torques (8) are presented in vector form, they can be, in practice, computed independently, since $\Theta_g \Psi(q)$ and K are diagonal matrices and no information is needed from the other input torques.
3. From (25), it can be seen that constraints on the control inputs, i.e., $|u_i| \leq u_{i\max}$ can be met by tuning the the PD gain K_i and the desired accelerations magnitudes q_0 .

5 Simulation Results

To test the proposed adaptive neural control, we consider the two-link manipulator (fig. 1) whose dynamic equations of motion (1) are:

$$M(q) = \begin{bmatrix} (m_1 + m_2)l_1^2 & m_2l_1l_2(s_1s_2 + c_1c_2) \\ m_2l_1l_2(s_1s_2 + c_1c_2) & m_2l_2^2 \end{bmatrix}$$

$$c(q, \dot{q}) = \begin{bmatrix} 0 & -m_2l_1l_2(c_1s_2 - s_1c_2)\dot{q}_2 \\ -m_2l_1l_2(c_1s_2 - s_1c_2)\dot{q}_1 & 0 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} -g(m_1 + m_2)s_1l_1 \\ -gm_2l_2s_2 \end{bmatrix}$$

where $c_1 = \cos(q_1)$, $c_2 = \cos(q_2)$, $s_1 = \sin(q_1)$ and $s_2 = \sin(q_2)$. The robot physical parameters are: $l_1 = l_2 = 1\text{m}$, $m_1 = m_2 = 1\text{kg}$, and $g = 9.81 \text{ m/s}^2$.

The uncertainties terms in (1) are given by:

$$\tau_c = \begin{bmatrix} \dot{q}_1 + \sin(3q_1) \\ 1.2\dot{q}_2 + 0.5\sin(2q_2) \end{bmatrix}, \quad \tau_d = \begin{bmatrix} 0.2\text{sign}(\dot{q}_1) \\ 0.1\text{sign}(\dot{q}_2) \end{bmatrix}$$

The NNs adaptive control design for the two-link robot is as follows:

1. In order to construct the NNs approximators, each variable $q_1, q_2 \in [-\pi, \pi]$ and $\dot{q}_1, \dot{q}_2 \in [-2\pi, 2\pi]$ range is devised into 3 sub domains, which yields four RBF networks to approximate $f_1(q, \dot{q}_2)$, $f_2(q, \dot{q}_1)$, $g_{11}(q)$ and $g_{22}(q)$, with $q^T = [q_1 \quad q_2]$, with, respectively, 27, 27, 9 and 9 RBFs. The designed RBF networks take the following compact forms:

$$\hat{f}_1(q, \dot{q}_2) = \theta_{f_1}^T \phi_1(q, \dot{q}_2) \quad (26)$$

$$\hat{f}_2(q, \dot{q}_1) = \theta_{f_2}^T \phi_2(q, \dot{q}_1) \quad (27)$$

$$\hat{g}_{11}(q) = \theta_{g_1}^T \psi_1(q) \quad (28)$$

$$\hat{g}_{22}(q) = \theta_{g_2}^T \psi_2(q) \quad (29)$$

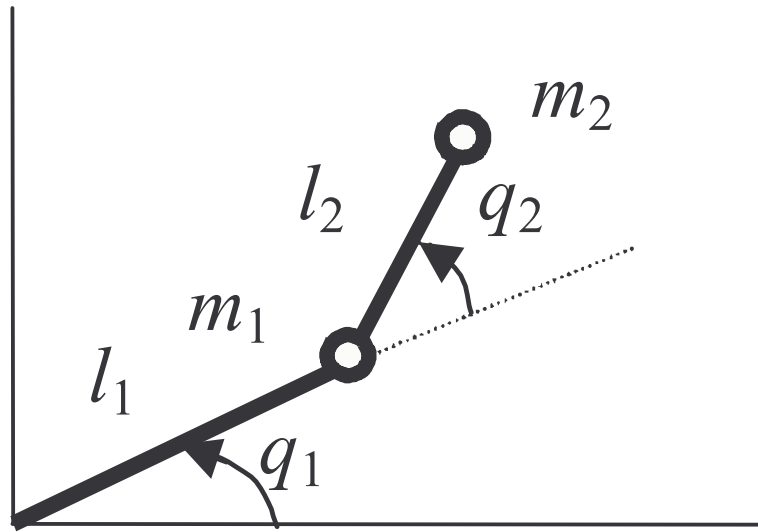


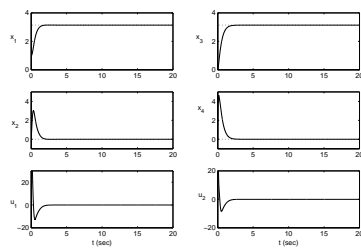
Figure 1: Two link robot

where $\theta_{f_1}, \theta_{f_2} \in R^{27 \times 1}$ and $\theta_{g_1}, \theta_{g_2} \in R^{9 \times 1}$ are the NNs adjustable parameters. This approach is far from being optimal, but it has the merit to reduce the number of parameters to be learned and thus to reduce the update algorithm complexity and execution time.

2. The control input is designed as in (8) with the adaptive NNs defined in (26)-(29). The PD gain is defined as $K = \text{diag}[K_1, K_2]$ with $K_1 = K_2 = \begin{bmatrix} 16 & 8 \end{bmatrix}$.
3. For the choice of $Q = 100I_4$ and the solutions of (12) we get P .
4. By analyzing the dynamic of the robot, the following bounds are fixed $g_M = 2.5$, $g_m = 0.5$ and $f_M = 20$. The adjustable parameters are updated using (10)-(11) with $\gamma_1 = 0.01$ and $\gamma_2 = 0.001$.

In the simulation, the NNs parameters are initialized as $|\theta_{g_1}(0)| = |\theta_{g_2}(0)| = 0.5$ and $|\theta_{f_1}(0)| = |\theta_{f_2}(0)| = 0$. The initial states, in all simulations, are $x^T(0) = \begin{bmatrix} 0.5 & 2 & -0.5 & 1 \end{bmatrix}$.

The first simulation test concerns the regulation of the joint positions under nominal conditions, i.e., no parameters changes and no disturbances. As depicted in figure 2.a, the joint positions exhibit a good transient performance, and no error is remarked in steady state regime. Figure 2.b shows the regulation performance under uncertainties effects. From this figure it is seen that these uncertainties affect little the regulation performance and small steady state error is introduced and the NNs adaptive control achieves good compensation of the uncertainties effects. In figure 2.c, in addition to the uncertainties effects, a payload change is introduced at $t = 15$ s when m_2 passes from 1kg to 3kg. This situation is rapidly taken in account by the NNs control and it's effect is compensated.



a) nominal case

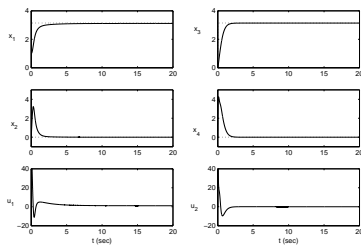
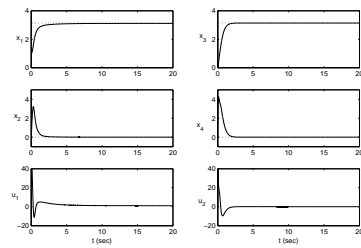
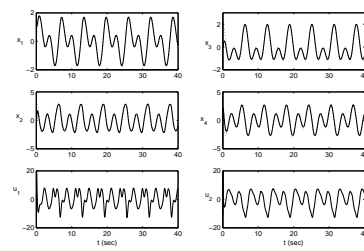
b) $\tau_c + \tau_d$ uncertainties effectsc) $\tau_c + \tau_d + m_2$ variation effects

Figure 2: Regulation performance



a) nominal case

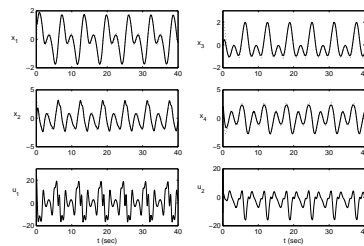
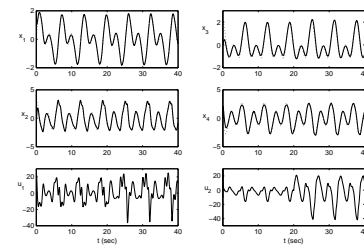
b) $\tau_c + \tau_d$ uncertainties effectsc) $\tau_c + \tau_d + m_2$ variation effects

Figure 3: Tracking performance

The second simulation test concerns the tracking performance for the desired trajectories $q_{r1} = \sin(t) + \sin(2t)$ and $q_{r2} = \cos(t) + \cos(2t)$. Figure 3.a presents the tracking performance in the nominal case. As depicted, the joint positions exhibit a good transient performance, and small error is remarked in steady state regime. Figure 3.b shows the tracking performance under uncertainties effects. It is clear that these uncertainties introduce acceptable tracking error, and the NNs control inputs compensate the uncertainties with a little effort. In figure 3.c, we add to the uncertainties effects, a payload change is introduced at $t = 20s$ when m_2 passes from 1kg to 3kg. This variation affects essentially the developed torques to compensate the additional mass, and small error is remarked.

6 Conclusion

In this paper, an adaptive NNs control scheme for rigid robot control, was proposed. The adaptive capability of handling modeling errors and external disturbances was demonstrated. The error convergence rate with the NNs adaptive approach was found to be fast. Asymptotic stability of the control system is established using the Lyapunov approach. Simulation studies for a two-link robot verify the flexibility, adaptation and tracking performance of the proposed approach. The major contributions of the paper are as follows: reduction of the NNs complexity and no robustifying control is required to achieve the stability or to enhance the control performance.

Bibliography

- [1] J. Y. Han, H. Hemami, and S. Yurkovich, Nonlinear adaptive control of a N-link robot with unknown load, *Int. J. Robotics Res.*, vol. 6, no. 3, pp. 71–86, 1987.
- [2] J. E. Slotine and W. Li, On the adaptive control of robot manipulators, *Int. J. Robotics Res.*, vol. 6, no. 3, pp. 49–59, 1987.
- [3] R. Ortega and M. Spong, Adaptive motion control of rigid robots: A tutorial, *Proc. 27th Conf. on Decision and Control CDC*, Austin, Texas, , pp. 1575-1584, Dec 1988.
- [4] J. J. Graig, *Adaptive Control of Mechanical Manipulators*, New York, Addison-Wesley, 1988.
- [5] Z. Qu, J. F. Dorsey and D. M. Dawson, Robust Control of robots by computed torque law, *Systems and Control Letter*, Vol.16, pp.25-32, 1991.
- [6] G. Jumarie, Tracking control of mechanical systems via sliding Lagrangian, *J. Intell. Robotic Syst.*, vol. 13, pp. 181–199, 1995.
- [7] S. Pandian and M. Hanmandlu, Adaptive generalized model-based control of robot manipulators, *Int. J. Contr.*, vol. 58, no. 4, pp. 835-853, 1993.
- [8] W. C. Ham, Adaptive control based on explicit model of robot manipulators, *IEEE Trans. Automat. Contr.*, vol. 38, pp. 654–658, 1993.
- [9] M. B. Leahy, M. A. Johnson, S. K. Rogers, Neural network payload estimation for adaptive robot control, *IEEE Transactions on Neural Networks*, Vol. 2, No. 1, pp. 93-100, Jan. 1991.
- [10] M. Saad, P. Bigras, L.-A. Dessaint, K. Al-Haddad, Adaptive robot control using neural networks, *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 2, pp. 173-181, Apr 1994.
- [11] L. Behera, M.Gopal and S.Chaudhury, Inversion of RBF networks and applications to adaptive control of nonlinear systems, *IEE Proc.-Control Theory Appl.*, Vol. 142, No. 6, Nov. 1995.
- [12] F. L. Lewis, A. Yesildirak , S. Jagannathan, *Neural network control of robot manipulators and nonlinear systems*, Taylor & Francis, Inc. Bristol, PA, USA 1998.
- [13] M. Quoy, S. Moga, P. Gaussier, Dynamical neural networks for planning and low-level robot control, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 33, No. 4, pp. 523-532, July 2003.
- [14] C. Torras, Natural inspiration for artificial adaptivity: some neurocomputing experiences in robotics, *Proc. 4th Intl. Conf. on Unconventional Computation (UC'05)*, Sevilla, Spain, Oct. 2005.
- [15] M. J. D. Powell, “Radial basis functions for multivariable interpolation: A review,” in *Algorithms for Approximation of Functions and Data*, J. C. Mason, M. G. Cox, Eds. Oxford, U.K.: Oxford Univ. Press, 1987, pp. 143-167.
- [16] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, “Layered neural networks with gaussian hidden units as universal approximations,” *Neural Comp.*, vol. 2, no. 2, pp. 210-215, 1990.
- [17] T. Poggio and F. Girosi, “Networks for approximation and learning,” *Proc. IEEE*, vol. 78, pp. 1481-1497, Sept. 1990.
- [18] J. Park and I. W. Sandberg, Universal approximation using radial-basis-function networks, *Neural Comp.* 3, 246-257, 1991.

- [19] B. Liu and J. Si, The best approximation to C^2 function and its error bounds using regular-center gaussian networks, IEEE Trans. Neural Networks, vol. 5, pp.848-857, 1994.
- [20] R.J. Schilling, J.J. Jr. Carroll, A.F Al-Ajlouni, Approximation of nonlinear systems with radial basis function neural networks, IEEE Transactions on Neural Networks, vol. 12, pp. 1-15, 2001.
- [21] H. Guang-Bin, P. Saratchandran, N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks, vol. 16, pp. 57- 67, 2005
- [22] S. S. Sastry and M. Bosdon, Adaptive control: Stability, Convergence and Robustness, Printice-Hall, 1989.

Ghania Debbache
Electrical Engineering Institute, Oum El-Bouaghi University, 04000 Oum
El-Bouaghi, Algeria
gdebbache@yahoo.fr

Abdelhak Bennis
Electronic Departement, Constantine University
25000 Constantine, Algeria
abdelhak.bennis@laposte.net

Noureddine Goléa
Electrical Engineering Institute, Oum El-Bouaghi University, 04000 Oum
El-Bouaghi, Algeria
n.golea@lycos.com

Received: March 15, 2007



Ghania Debbache was born in Constantine, Algeria, in 1974. She received the B.Sc. and M.Sc. degrees in electronics from the Constantine University, Algeria, in 1997 and 2000, respectively. She is currently pursuing the Ph.D. degree on the intelligent control at the University Constantine. Currently, she is a Teaching Assistant at the Technologic Sciences Institute of Oum El Bouaghi University, Algeria.



Abdelhak Bennia received his D.E.S. degree in 1983 in physics from the University of Constantine, Algeria. From 1984 to 1990 he attended the graduate school at Virginia Tech, majoring in electrical engineering. He received the M.Sc. degree in 1986 and the Ph.D. degree in 1990. Since 1990, he has been with the electronics department of the University of Constantine. His current research interests are deconvolution, system identification and neural networks applied to character recognition, control systems, and signal processing.



Nouredine Goléa was born in Batna, Algeria, in 1967. He received the Engineer and Master grades from Sétif University, Algeria, and the Doctorat from Batna University, Algeria, all in industrial control, in 1991, 1994, and 2000, respectively. From 1991 to 1994, he was with Electronics Institute at Sétif University, and from 1994 to 1996, he was with the Electronics Institute at Batna University. Currently, he is Professor of electrical engineering in the Technologic Sciences Institute at Oum El-Bouaghi University, Algeria. His research interests are nonlinear and adaptive control, and intelligent control applied to motion control.