# STABLE ADAPTIVE CONTROL WITH RECURRENT NEURAL NETWORKS

**Salem Zerkaoui, Fabrice Druaux, Edouard Leclercq, Dimitri Lefebvre**

*Université Le Havre – GREAH, 25 rue P. Lebon, 76063 Le Havre, France*
*{salem.zerkaoui ;fabrice.druaux ;edouard.leclercq ;dimitri.lefebvre}@univ-lehavre.fr*

Abstract: In this paper, stable indirect adaptive control with recurrent neural networks is presented for multi-input multi-output (MIMO) square non linear plants with unknown dynamics. The control scheme is made of a neural model and a neural controller based on fully connected RTRL networks. On-line weights updating law, closed loop performance, and boundedness of the neural network weights are derived from the Lyapunov approach. Sufficient conditions for stability are obtained according to the adaptive learning rate parameter. 

Keywords: Adaptive control, fully connected recurrent neural networks, Lyapunov function, multivariable systems, stability analysis.

## 1. INTRODUCTION

Research in non linear control theory has been motivated by the inherent characteristics of the dynamical systems to control. Concerning the non linear nature of the dynamics: if we consider the fact that most systems are not perfectly known, therefore the choice of a model is not an easy task. Adaptive control seems today an efficient strategy to tackle the stabilization and tracking of highly uncertain dynamical systems.

Since neural networks (NN) have general properties like learning ability, approximation capability and generalization, they have been successfully applied to the identification and control of non linear systems. In closed loop control using NN the problems associated with implementation are complicated, so that the control system must satisfy three conditions: boundedness of the NN weights, boundedness of the tracking errors and stability of the global system under control. In an attempt to guarantee these criterions, a considerable research effort has concerned advanced neural control systems with high accurate tracking performance and strong robustness. A major design technique has emerged, namely, neural controller design using Lyapunov stability theory. One main advantage of this approach is that the parameter adaptation laws are based on Lyapunov synthesis in order to satisfy the stability of the closed loop system. In most of the above works, the neural control schemes are divided into two groups: "pure" neural controllers (Druaux *et al.*, 2004; Wang, 2003) and neural controllers combined with other conventional control strategies such as back stepping and sliding mode (Sanner and Slotine, 1992; Zhihong *et al*., 1998; Ge *et al*., 2002). In that case, the role of neural networks is to approximate the non linear input-output relations. Recently, many works concerning the stability of NN controllers have also been published. In (Tian and Collins, 2004; Ge *et al*., 2002), direct and indirect adaptive control schemes based on a recurrent NN and radial basis functions models of the unknown system are proposed. Lyapunov methods were also investigated to provide answers to the problems of stability, convergence and robustness (Wang, 2003). In (Tian and Collins, 2004), stability of the whole control system is

governed by the characteristics of the dynamic recurrent neural network, which can be established and analyzed according to the learning rate choice. For multivariable nonlinear systems, due to the couplings among various inputs and outputs, the control problem is more complex and few results are available in the literature. In (Ge *et al*., 2000; Ge *et al*., 2001), the authors presented a stable adaptive control scheme for a multivariable nonlinear systems with a triangular structure using multilayer neural networks. The control design is based on integral type Lyapunov function and the block-triangular structure properties. These control schemes, however, cannot be extended to the general class of MIMO nonlinear systems.

In this paper, we further investigate stable indirect recurrent NN control for unknown non linear multivariable systems. (i.e. no model is drawn of the process because of a lack of information about it) (Druaux et al., 2004; Leclercq et al., 2005). The process is considered as a black box, known as series of input and output measurements. The proposed algorithm does not depend on any preliminary knowledge about dynamics. We consider the case of square MIMO (i.e. input-output numbers are assumed to be equal) controllable systems. The proposed adaptation algorithm is inspired by the real time recurrent learning (RTRL) which was proposed by R.J. William and D. Zipser (Williams and Zipser, 1989). The main advantage of this method is that adaptation is obtained whatever the process evolution is. Starting from zero initial conditions, the proposed controller adapts itself, so as the learning rate and the time parameter, in order to track dynamical behaviors. Analyzing the error functions in term of Lyapunov criterion, we show that the stability of the closed loop can be reinforced. As a consequence, sufficient conditions for asymptotic stability are obtained according to the learning rate parameter.

The paper is organized as follows: design of the indirect adaptive neural network controller is described in section 2; self adaptation algorithm for the parameters of the controllers is also presented; in section 3, the convergence and stability of the on-line RTRL control algorithm based on discrete Lyapunov function is studied; simulations are proposed in section 4.

## 2. ADAPTIVE NN CONTROL

Indirect neural network controller (IDNC) consists of two separate neural networks, namely the neural controller (NC) and the neural model (NM). The complete scheme of IDNC is shown in Fig. 1. The updating of NC and NM is synchronous. Let us define $N_{IN}$ and $N_{OUT}$ respectively as the number of plant inputs and outputs and assume that $N_{IN} = N_{OUT}$, where *IN* and *OUT* represent the set of inputs and outputs index . For NM, the total number of neurons $N_{m,}$ is chosen equal to $N_{IN}+N_{OUT}$ , so that any node is either an input node or an output node but not both at same time, in order to avoid to perturb any output signal with input ones. For NC, the total number of neurons $N_{c,}$ is chosen equal to $N_{OUT}$ , so that each neuron is simultaneously an input and an output.

Inputs and outputs signals of the NM, NC and plant are normalized in the range of [-1,1]. Let us define $t = k\Delta T$ where $\Delta T$ is a sampling period and $k$ an integer. For the sake of simplicity, let us refer to instant $t$ by using the integer $k$. The subscripts $m$ and $c$ are used to distinguish the NM and NC respectively.

### *2.1. Neural model*

The NM is developed with fully connected recurrent neural networks (Leclercq *et al.*, 2005). The dynamics of the $N_m$ neurons take the following form in continuous time:

$$\frac{1}{|\tau_m(t)|}\frac{d\hat{Y}_i(t)}{dt} = -\hat{Y}_i(t) + \tanh\left(\sum_{j=1}^{N_m} W_{ij}(t)\hat{Y}_j(t) + X_i(t)\right) \quad (1)$$

with $X_i(t) = U_i(t)$ if $i \in IN$ and $X_i(t) = 0$ if $i \in OUT$. $\hat{Y}_i(t)$ , $W_{ij}$ , $1/\tau_m(t)$ and $U_i(t)$ represent respectively the $i^{th}$ neuron state, the NM weight from $j^{th}$ neuron to $i^{th}$ neuron, the NM adaptive time parameter and the NC output. Let us define $\hat{Y}(k) = [\hat{Y}_i(k)]^T$ , $i \in OUT$, as the estimated output vector of the plant at time $t = k\Delta T$. A matrix representation of (1) is given in sampled time with (2):

$$\hat{Y}(k+1) = e^{-|\tau_m(k)|\Delta T}\hat{Y}(k) - \left[e^{-|\tau_m(k)|\Delta T} - I\right]B(k) \quad (2)$$

with $B(k) = (B_i(k)) \in IR^{Nm}$ and:

$$B_i(k) = \tanh\left(\sum_{j=1}^{Nm} W_{ij}(k)\hat{Y}_j(k) + X_i(k)\right) \quad (3)$$

The vector $B$ is considered constant during each sampling period. The autonomous adaptation algorithm is inspired by the principle of RTRL. Let us consider the instantaneous square error between the estimated output vector and the plant output vector defined as in (4):

$$E_m(k) = \frac{1}{2}\sum_{l\in OUT}\left[Y_l(k) - \hat{Y}_l(k)\right]^2 \quad (4)$$

The objective is to minimize $E_m(k)$ with respect to the parameters: weights are updated according to (5):

$$\Delta W_{ij}(k) = W_{ij}(k) - W_{ij}(k-1) = -|\eta_m|\Delta T\left[Hm^{\mathrm{T}}(k)P_{-ij}(k)\right] \quad (5)$$

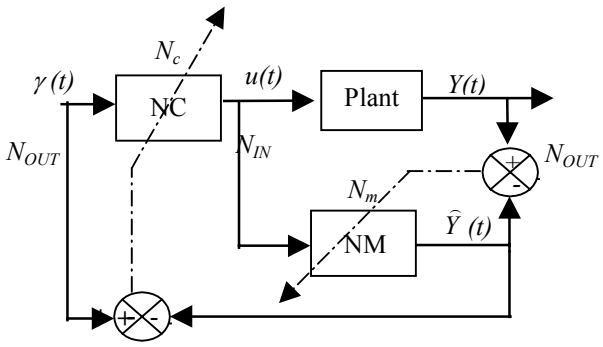


Fig. 1. Structure of the indirect neural network controller

The subscript "_" in the notation $P_{-ij}^{\mathrm{T}} = \left[P_{1ij}, P_{2ij}, \ldots\right]$ is used to indicate a vector of the 3D matrix $P$. The notations $\eta_m, P_{lij} = \partial \hat{Y}_l / \partial W_{ij}$ and $Hm = \hat{Y}(k) - Y(k)$ represent respectively the learning rate, the network sensitivity with respect to weights and the model error vector. $P_{lij}$ is updated according to (6):

$$P_{\_ij}(k+1) = e^{|\tau_m(k)|.A(k).\Delta T} P_{\_ij}(k) + A^{-1}(k)\left[e^{|\tau_m(k)|.A(k).\Delta T} - I\right] B'(k) \tag{6}$$

where $B'(k) = (0,\ldots,0,B_i',0,\ldots,0)^T \in IR^{Nm}$, with $B_i'(k) = H_i(k)\hat{Y}_j(k)$ and :

$$A(k) = \begin{bmatrix} -1+H_1W_{11} & H_1W_{12} & \ldots & H_1W_{1N_m} \\ H_2W_{21} & -1+H_2W_{22} & \ldots & H_2W_{2N_m} \\ . & . & & . \\ . & . & \ldots & . \\ . & . & & . \\ H_{N_m}W_{N_m1} & H_{N_m}W_{N_m2} & \ldots & -1+H_{N_m}W_{N_mN_m} \end{bmatrix}$$

$$H_i(k) = \tanh'\left(\sum_{h=1}^{N_m} W_{ih}(k)\hat{Y}_h(k) + X_i(k)\right) \tag{7}$$

The vector $B'$ is considered constant during each sampling period

*2.2 Neural controller*

The NC is also a fully connected recurrent neural network based on autonomous RTRL algorithm. The NC input is the plant desired output vector γ(t), and the NC output is the control signal vector U(t). The control signals are calculated by comparing the NM output with the desired system responses and according to the neurons dynamic activation given by (8) in continuous time:

$$\frac{1}{|\tau_C(t)|}\frac{dU_i(t)}{dt} = -U_i(t) + \tanh\left(\sum_{j=1}^{N_c}\Phi_{ij}(t)U_j(t) + \gamma_i(t)\right) \tag{8}$$

where $\Phi_{ij}$ and $1/\tau_c(t)$ represent respectively the NC weights value from $j^{th}$ neuron to $i^{th}$ neuron and the NC adaptive time parameter. A matrix representation of (8) is given in sampled time by (9):

$$U(k+1) = e^{-|\tau_C(k)|.\Delta T}U(k) - \left[e^{-|\tau_C(k)|.\Delta T} - I\right]D(k) \tag{9}$$

with $D(k) = (D_i(k)) \in IR^{Nc}$ and:

$$D_i(k) = \tanh\left(\sum_{j=1}^{N_c}\Phi_{ij}(k).U_j(k) + \gamma_i(k)\right) \tag{10}$$

The vector $D$ is considered constant during each sampling period. The adaptation of the NC weights is also obtained with an optimization algorithm inspired by the gradient descent. The cost function for training the NC is given by (11):

$$E_c(k) = \frac{1}{2}\sum_{l\in OUT}\left[\gamma_l(k) - \hat{Y}_l(k)\right]^2 \tag{11}$$

The weights are then updated using (12):

$$\Delta\Phi_{ij}(k) = -|\eta_c|\Delta T\left(J_{\_}Q_{\_ij}\right)^{\mathrm{T}} Hc(k) \tag{12}$$

where $Hc(k) = \hat{Y}(k) - \gamma(k)$ represents the tracking error vector and:

$$\frac{\partial \hat{Y}_l}{\partial \phi_{ij}} = \sum_{d\in IN}\frac{\partial \hat{Y}_l}{\partial U_d}\frac{\partial U_d}{\partial \phi_{ij}} = \sum_{d\in IN} J_{ld}Q_{dij} = J_{l\_}Q_{\_ij}$$

$Q_{\_ij}$ and $J_{\_d}$ are defined in the same way as $P_{\_ij}$ and are computed as follows:

$$Q_{\_ij}(k+1) = e^{|\tau_c(k)|.C(k).\Delta T}Q_{\_ij}(k) + C^{-1}(k)\left[e^{|\tau_c(k)|.C(k).\Delta T} - I\right]D'(k) \tag{13}$$

$$J_{\_d}(k+1) = e^{|\tau_m(k)|.A(k).\Delta T}J_{\_d}(k) + A^{-1}(k)\left[e^{|\tau_m(k)|.A(k).\Delta T} - I\right]D''(k) \tag{14}$$

$D'(k) = (0,\ldots,0,D_i',0,\ldots,0)^T \in IR^{Nc}$ with $D_i'(k) = S_i(k).U_j(k)$, $D''(k) = (0,\ldots,0,D_d'',0,\ldots,0)^T \in IR^{Nm}$ with $D_d''(k) = H_d(k)$, and:

$$C(k) = \begin{bmatrix} -1+S_1\Phi_{11} & S_1\Phi_{12} & \ldots & S_1\Phi_{1N_c} \\ S_2\Phi_{21} & -1+S_2\Phi_{22} & \ldots & S_2\Phi_{2N_c} \\ . & . & & . \\ . & . & \ldots & . \\ . & . & & . \\ S_{N_c}\Phi_{N_c1} & S_{N_c}\Phi_{N_c2} & \ldots & -1+S_{N_c}\Phi_{N_cN_c} \end{bmatrix}$$

$$S_i(k) = \tanh'\left(\sum_{h\in IN}\Phi_{ih}(k).U_h(k) + \gamma_i(k)\right) \tag{15}$$

*2.3. Automatic adaptation of the parameters $\tau$ and $\eta$*

In order to obtain an autonomous control able to adapt itself to a large variety of uncertain or unknown processes with few initial constraints, we consider updating rates $\eta_m(k)$ and $\eta_c(k)$ and time parameters $1/\tau_m(k)$ and $1/\tau_c(k)$ as time varying functions. The dynamics of $\eta_c(k)$ and $\tau_c(k)$ are worked out in order to get the dynamics of the NC close to the NM one.

$$\tau_c(k) = \tau_m(k) = \tau(k), \qquad \eta_c(k) = \eta_m(k) = \eta(k) \quad (16)$$

$\tau(0) = \tau_0$ and $\eta(0) = \eta_0$ are very small constants necessary for the control process to start. Using the instantaneous square model error $E_m(k)$, parameters $\tau(k)$ and $\eta(k)$ are updated using (17):

$$\begin{aligned} \eta(k) &= \eta(k-1) + \Delta T\left[Hm^{\mathrm{T}}(k) V^{\eta}(k)\right], \\ \tau(k) &= \tau(k-1) + \Delta T|\eta|\left[Hm^{\mathrm{T}}(k) V^{\tau}(k)\right] \end{aligned} \quad (17)$$

with $V_l^{\eta} = \dfrac{\partial \hat{Y}_l}{\partial \eta}$ and $V_l^{\tau} = \dfrac{\partial \hat{Y}_l}{\partial \tau}$.

Our objective is to design a stable self adaptive neuronal controller, as simple as possible in term of parameters and calculations to be carried out. To achieve this objective we suppose that $V_l^{\eta} = V_l^{\tau} = V_l$. Considering the analogy between $V_l(k)$ and $P_{lij}(k)$, we can write:

$$V(k+1) = e^{|\tau(k)|A(k).\Delta T} V(k) + A^{-1}(k)\left[e^{|\tau(k)|A(k).\Delta T} - I\right].\frac{\varepsilon}{|\tau(k)|} \quad (18)$$

with $\varepsilon = (dV_l / dt)_{t=0} > 0$ in order to start the adaptation process. As a consequence, we notice that the network evolves in an autonomous way starting from zero initial conditions.

## 3. STABILITY ANALYSIS

Stability properties of the closed loop system are important issues to be addressed. Abrupt variations of the reference signal make the process to move away from the equilibrium state. To reach a new equilibrium, the adaptation procedure can cause oscillations or divergences: instabilities may arise as a consequence.

We propose to study the stability of the IDNC with the well-known Lyapunov approach. The purpose of control is to force the output of the controlled process to track a desired trajectory. From this point of view, Lyapunov function candidate is investigated according to the model-tracking error; sufficient condition for stability is derived as a consequence. The Lyapunov function candidate is defined as:

$$E_{mc}(k) = E_m(k) + E_c(k) = \frac{1}{2}\sum_{l \in OUT}\left(Hm_l^2(k) + Hc_l^2(k)\right) \quad (19)$$

The output of the closed loop system will accurately track the desired output and remains near the desired trajectory if $\Delta E_{mc}(k) \le 0$, with

$$\begin{aligned} \Delta E_{mc}(k) &= \frac{1}{2}\sum_{l \in OUT}\left(Hm_l^2(k+1) - Hm_l^2(k) + Hc_l^2(k+1) - Hc_l^2(k)\right) \\ &= \left[\Delta Hm\right]^{\mathrm{T}}\left\{\left[Hm\right] + \frac{1}{2}\left[\Delta Hm\right]\right\} + \left[\Delta Hc\right]^{\mathrm{T}}\left\{\left[Hc\right] + \frac{1}{2}\left[\Delta Hc\right]\right\} \end{aligned} \quad (20)$$

Let us introduce the following notations:

$$\begin{aligned} \Phi_1 &= \Delta T \sum_{i=1}^{Nm}\sum_{j=1}^{Nm} P_{-ij} P_{-ij}^{T} \\ \Phi_2 &= \Delta T . V_{-} V_{-}^{\mathrm{T}} \\ \Phi_3 &= \Delta T \sum_{i=1}^{Nc}\sum_{j=1}^{Nc} J_{--} Q_{-ij} Q_{-ij}^{\mathrm{T}} J_{--}^{\mathrm{T}} \end{aligned} \quad (21)$$

So, (20) can be rewritten as:

$$\begin{aligned} \Delta E_{mc}(k) = & -Hm^{\mathrm{T}}\left(|\eta|\Phi_1 + (|\eta|+1)\Phi_2\right)^{\mathrm{T}} . \\ & \left\{I - \frac{1}{2}\left(|\eta|\Phi_1 + (|\eta|+1)\Phi_2\right)\right\} Hm \\ & -\left\{Hm^{\mathrm{T}}\left(|\eta|\Phi_1 + (|\eta|+1)\Phi_2\right)^{\mathrm{T}} + Hc^{\mathrm{T}}|\eta|\Phi_3^{\mathrm{T}}\right\} . \\ & \left\{Hc - \frac{1}{2}\left(|\eta|\Phi_1 + (|\eta|+1)\Phi_2\right) Hm - \frac{1}{2}|\eta|\Phi_3 Hc\right\} \end{aligned} \quad (22)$$

where $I$ is the identity matrix. A sufficient condition to ensure the close loop asymptotic stability is expressed by the following quadratic form:

$$\Delta E_{mc} = a|\eta|^2 - 2b|\eta| - c \le 0 \quad (23)$$

with:

$$\begin{cases} a = \left[\left(\Phi_1 + \Phi_2\right) Hm + \Phi_3 Hc\right]^{\mathrm{T}}\left[\left(\Phi_1 + \Phi_2\right) Hm + \Phi_3 Hc\right] \\ \quad + Hm^{\mathrm{T}}\left(\Phi_1 + \Phi_2\right)^{\mathrm{T}}\left(\Phi_1 + \Phi_2\right) Hm \ge 0 \\ b = \left(Hm^{\mathrm{T}}\left(\Phi_1 + \Phi_2\right)^{\mathrm{T}} Hc + Hc^{\mathrm{T}}\Phi_3^{\mathrm{T}} Hc + Hm^{\mathrm{T}}\left(\Phi_1 + \Phi_2\right)^{\mathrm{T}}\right. \\ \quad \left.\left(I - 2\Phi_2\right) Hm - Hc^{\mathrm{T}}\Phi_3^{\mathrm{T}}\Phi_2 Hm\right) \\ c = 2Hm^{\mathrm{T}}\left(\Phi_2^{\mathrm{T}} - \Phi_2^{\mathrm{T}}\Phi_2\right) Hm + 2Hm^{\mathrm{T}}\Phi_2^{\mathrm{T}} Hc \end{cases}$$

Hence concerning $|\eta|$ condition for the stability of the indirect neural network controller is given by (24):

$$0 \le \frac{b - \sqrt{\Delta'}}{a} \le |\eta| \le \frac{b + \sqrt{\Delta'}}{a} \quad (24)$$

with $\Delta' = b^2 + a.c$ .When the sufficient condition (24) is not verified, the process is stabilized by replacing the current value of parameter $|\eta|$ with the nearest limit value (i.e. $(b+\sqrt{\Delta'})/a$ or $(b-\sqrt{\Delta'})/a$).

## 4. SIMULATION RESULTS

Let us consider the dynamical model of a planar robotic manipulator with 2 links. The dynamics of this system are non linear and coupled according to the freedom degrees of both arms. The equations of motion in terms of the angular positions of arms 1 and 2 are given by:

$$\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} \begin{pmatrix} \Gamma_1 - G_1 - G_2 - f_1\dot{q}_1 - 2Q\dot{q}_2\dot{q}_2 - Q\dot{q}_2^2 \\ \Gamma_2 - G_2 - f_2\dot{q}_2 \end{pmatrix}$$

with:

$$j_1 = \left((m_1 + m_2)L_1^2 + m_2 L_2^2\right)/2$$
$$j_2 = m_2 L_1 L_2 \quad \text{and} \quad j_3 = m_2 L_2^2$$
$$G_1 = (m_1 + m_2) g L_1 \cos(q_1)$$
$$G_2 = m_2 g L_2 \cos(q_1 + q_2)$$
$$Q = j_2\left(\sin(q_1)\cos(q_1 + q_2) + \cos(q_1)\sin(q_1 + q_2)\right)$$
$$H_{11} = 2\left(j_1 + j_2\left(\sin(q_1)\sin(q_1 + q_2) + \cos(q_1)\cos(q_1 + q_2)\right)\right)$$
$$H_{12} = H_{21} = j_3 + j_2\left(\sin(q_1)\sin(q_1 + q_2) + \cos(q_1)\cos(q_1 + q_2)\right)$$
$$H_{22} = 2 j_3$$

where $q_1$, $q_2$ are the angular positions; $m_1$, $m_2$ are the masses; $L_1$ ,$L_2$ are the lengths of both arms. The plant parameters are chosen as $m_1$ = 3Kg , $m_2$ = 2.3 Kg, $L_1$ = 1.1 m , $L_2$=1 m, $g$ = 9.81 m/s$^2$ and $f_1 = f_2$= 10 kg.m$^2$.s$^{-1}$.rad$^{-1}$. For this purpose the NC has 2 neurons and the NM has 4 neurons (Leclercq *et al*., 2005). The neural networks parameters are starting from zero initial conditions. For our algorithm, this plant is supposed to be completely unknown and can be considered like a black box.

Simulations are obtained using a four-order Runge–Kutta method with an integral step of 0.1s (to solve the plant non-linear differentials equations numerically), a controller sampling interval $\Delta T$ = 0.1 s and $\varepsilon$ = 1. Fig 2(a)-(d) show respectively the system outputs, the control inputs and the $\eta$ and $\tau$ parameters. Without stability constraint, closed loop system diverges. Instabilities are due to the descent gradient. We observe such instabilities because the network state does not reach the global minima and jumps from a local minima to another one (Leclercq *et al.*, 2005). Both outputs present a divergence when the learning rate parameter grows infinitely. But concerning the control signals, they are still bounded.

Fig 3(a) -(d) show the first and second arm positions, the control input and the $\eta$ and $\tau$ parameters with the stability sufficient condition index. This index is shown on the bottom of Fig. 3(d) and could take three values, {-1, -3, -5}: (-1): the stability criterion is not verified and the learning rate is updated; (-3): the stability criterion is verified; (-5): stability criterion is not verified but the upper bound of the criterion is negative. On Fig. 3(a) transient behaviours arise, due to some steps input signals. These transients not arise on the second arm because the corresponding reference signal has no switch.

The comparison between both series of simulations, proves firstly that the network divergences can be related to the variation of a single parameter, namely the time parameter, and secondly that these instabilities are avoided by application of the sufficient stability criterion.

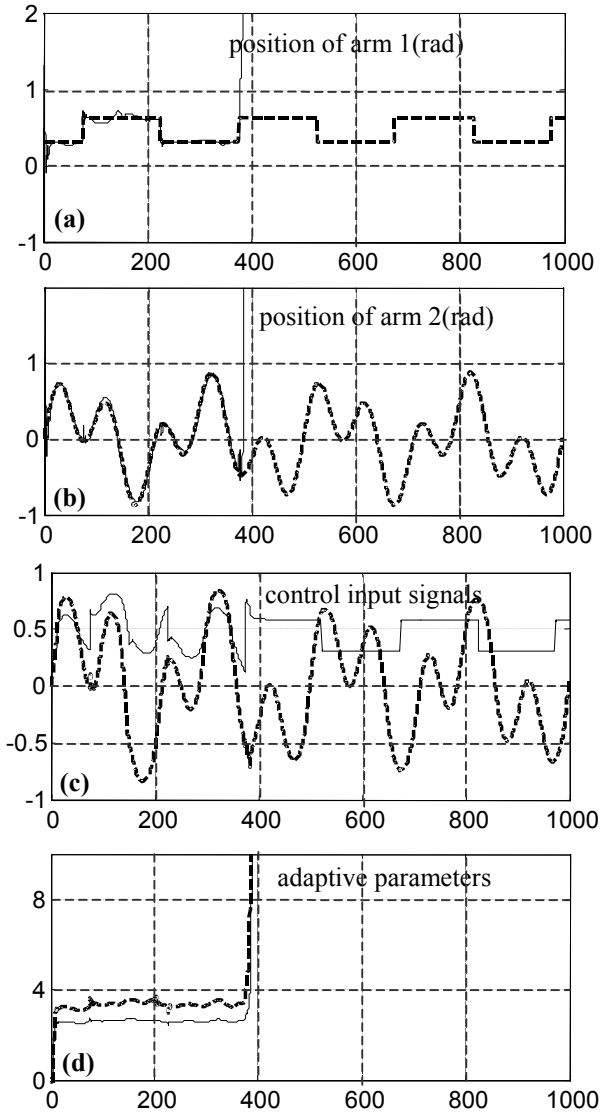


Fig. 2. Simulation results without stability criterion: (a) normalized output (solid line) and reference (dashed line) for arm 1; (b) normalized output (solid line) and reference (dashed line) for arm 2;(c) control input; (d) parameters $\tau(t)$ (solid line), $\eta(t)$ (dashed line).

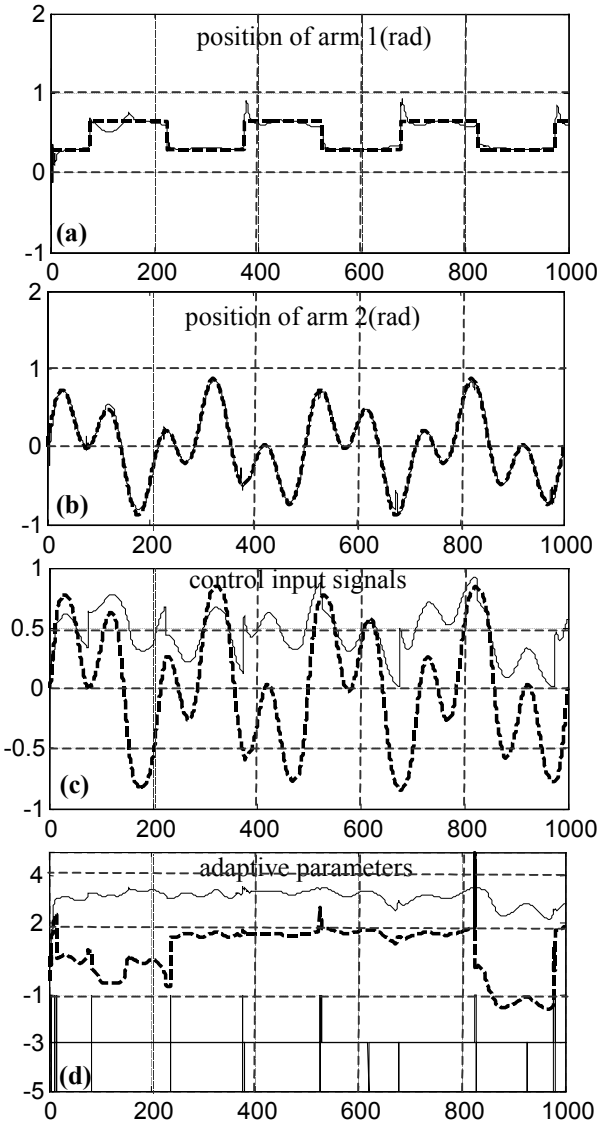


Fig.3. Simulation results with the stability criterion: (a) normalized output (solid line) and reference (dashed line) for arm 1; (b) normalized output (solid line) and reference (dashed line) for arm 2; (c) control input; (d) parameters $\tau(t)$ (solid line), $\eta(t)$ (dashed line) and stability sufficient condition index (lowest curve).

However, the apparent poor performance of the controller for arm 1 can be explained by the strong interaction between both arms and the continuous large swing of arm 2 which causes deviation of arm1 from the target position.

## 5. CONLUSIONS

A stable indirect adaptive control with recurrent neural networks is developed for MIMO square non linear plants with unknown dynamics. Stability analysis of the on-line RTRL learning algorithm for the closed loop system has been given, based on Lyapunov theory. It results a sufficient conditions. To guarantee stability, the learning rate must be bounded with two functions that are explained. Simulation results show that our approach results in good performance, simple structure and self tuning of the adaptation rates and time parameters. The method is easy to implement because stability is obtained thanks to the evaluation of a single parameter. Our further works concern the investigation of noise sensitivity in order to design robust control.

## ACKNOWLEDGEMENTS

This work is supported by the Industrial Development Department of the Region Haute-Normandie, France.

## REFERENCES

Druaux, F., Leclercq, E. and Lefebvre, D. (2004). Adaptive Neural Network Control for Uncertain or Unknown Non-linear Systems, *IEEE – MMAR*, Poland. pp. 1309-1314.

Ge, S.S., Hang, C.C. and Zhang, T. (2000). Stable adaptive control for multivariable non linear systems with a triangular control structure. *IEEE TAC*, vol. 45, no. 6, pp. 1221-1225.

Ge, S.S., Wang, C. and Tan, Y.H. (2001). Adaptive Control of Partially Known Non linear Multivariable Systems Using Neural Networks. *IEEE International Symposium on Intelligent Control*, September 5-7, México City, México. pp. 292-297.

Ge, S.S. and Wang, C. (2002). Direct Adaptive NN Control of a Class of Nonlinear Systems. *IEEE Trans. Neural Networks*, vol. 13, no.1. pp. 214-221.

Leclercq, E., Druaux, F., Lefebvre, D. and Zerkaoui. S. (2005). Autonomous learning algorithm for fully connected recurrent networks. *Neurocomputing,* vol. 63, pp. 25-44.

Sanner, R.M. and Slotine, J-J.E. (1992). Gaussian networks for direct adaptive control. *IEEE Trans. Neural Networks,* vol 3, pp.837-863.

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

Wang, J. (2003). Sensitivity identification enhanced control strategy for nonlinear process systems, *Computers and Chemical Engineering*, vol. 27, pp. 1631-1640.

Williams, R.J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comp*, vol. 1, no. 2, pp. 270–280.

Zhihong, M., Wu, H.R. and Palaniswami, M. (1998). An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems, *IEEE TNN*, vol. 9, no.5. pp. 947-955.