

# Stable Adaptive Neural Control of a Robot Arm

Salem Zerkaoui, Saeed M. Badran

<sup>1</sup>Université Le Havre—GREAH, Lebon, Le Havre, France

<sup>2</sup>Al Baha University, Al Baha, Saudi Arabia

Email: salem.zerkaoui@univ-lehavre.fr, sbadran@bu.edu.sa

Received March 7, 2012; revised April 9, 2012; accepted April 16, 2012

## ABSTRACT

In this paper, stable indirect adaptive control with recurrent neural networks (RNN) is presented for square multivariable non-linear plants with unknown dynamics. The control scheme is made of an adaptive instantaneous neural model, a neural controller based on fully connected “Real-Time Recurrent Learning” (RTRL) networks and an online parameters updating law. Closed-loop performances as well as sufficient conditions for asymptotic stability are derived from the Lyapunov approach according to the adaptive updating rate parameter. Robustness is also considered in terms of sensor noise and model uncertainties. This control scheme is applied to the manipulator robot process in order to illustrate the efficiency of the proposed method for real-world control problems.

**Keywords:** Adaptive Control; Neural Networks; Multivariable Systems; Stability; Robustness; Lyapunov Function; Manipulator Robot

## 1. Introduction

Research in non-linear control theory has been motivated by the inherent characteristics of the dynamical systems to control. Many systems are non-linear, their dynamics are not perfectly known and therefore not exactly modelled. Control engineers have hardly worked to improve the usual control methods as PID in order to guarantee closed loop stability in the presence of unmodeled dynamics and external disturbances. But, despite these efforts, conventional linear control techniques cannot meet all requirements to satisfy system performances, and adaptive control seems today an efficient strategy to study the stabilization and tracking of highly uncertain dynamical systems.

Since neural networks (NNs) have the advantages of inherent approximation capabilities and learning ability, they have been successfully implemented for identification and control of non-linear systems [1-10]. Especially, RNNs are suitable for dynamic mappings and lead to good control performance in the presence of unmodeled dynamics [10-12].

It is well known that FFN and RNN could be used as components in feedback systems [13]. The control system must satisfy three main conditions: boundedness of the NN weights, boundedness of the tracking error and stability of the global system under control. In an attempt to guarantee these criterions, a considerable research effort has concerned the design of neural networks control with high accurate tracking performances and strong

robustness [14]. A major design technique has emerged with the use of the Lyapunov theory. The main advantage of this approach is that the parameter adaptation laws ensure the asymptotic stability of the closed-loop system.

The adaptive neural controllers can be classified according to three structures, such as, inverse model [15], direct [1,5,14] and indirect [9,10,13] control design. Neural control schemes can also be divided into “pure” neural controllers [1,11,13] and neural controllers combined with other conventional control strategies such as back stepping and sliding mode [5,14,16]. In that case, the role of neural networks is to approximate the non-linear part of the input-output relation.

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 [8,9] 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.

The preceding works suffer from one or more of the following drawbacks: 1) the NNs used are linearly parameterised, whereas a non-linear model is necessary in order to control non-linear systems. 2) The system to be controlled is of a special known structure, (e.g., a train-

gular structure). Such results cannot be directly applied to an entirely unknown system. 3) The performances, for example stability and robustness, are locally guaranteed. 4) The model and/or the controller are fixed at the end of the training phase and the operational phase performances can be strongly degraded. It is often difficult to obtain the data which cover the whole range of the process excitation (problem of sufficient excitation).

In contrast to the above, the result presented in this paper avoids these drawbacks by establishing a stable adaptive control scheme based on the use of recurrent neural networks. In order to prove the industrial interest of the method for engineers, the application and validation of *IDNC* have been applied to robot manipulator which is a nonlinear system, multivariable with time-varying and/or inaccurately-known parameters. The controlled process is considered as a “black box” whose operating model is completely unknown. The main advantage of the proposed method is that models of the systems do not have to be known. Starting from zero values, weights, updating rates and time parameters of both adaptive instantaneous neural model and controller one, adapt themselves in order to track continuously the plant dynamics.

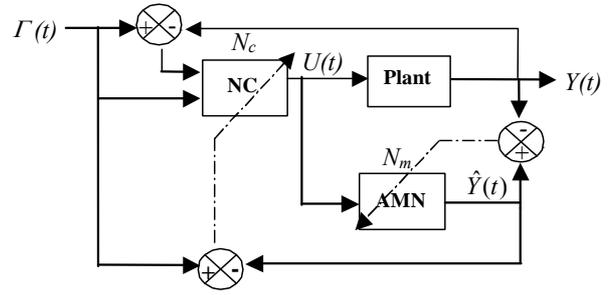
This paper is organized as follows. Section 2 describes *IDNC* structure, adaptation algorithm, stability and robustness conditions. Section 3 deals with the manipulator robot definition. In Section 4 the experiment results are presented. Finally Section 5 concerns our conclusions and perspectives.

## 2. Adaptive NN Control

The proposed control scheme is an Indirect Neural network Controller (*IDNC*) composed of two separate fully connected recurrent neural networks: the Neural Controller (NC) and the Adaptive instantaneous Monitoring Network (*AMN*) [13]. The aim of *AMN* is to provide an estimation of the process output(s) during a short time window in order to drive NC. The subscripts m and c are used to distinguish the *AMN* and NC respectively. The updating of NC and *AMN* is synchronous (**Figure 1**, the dash lines show the RTRL paths to update the parameters of *AMN* and NC). Discrete time is considered and for simplicity, let us refer to instant  $t = k\Delta T$  by using the integer  $k$ , with  $\Delta T$  the sampling period.

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 input and output indexes.

When the process is running, the neural networks continuously adapt. Comparison of the physical measurements with the neural network inputs and outputs is realised at each time. All the measured and manipulated



**Figure 1. Structure of the indirect neural network control, the dimension of  $Y(t)$ ,  $\hat{Y}(t)$  and  $\Gamma(t)$  is  $N_{OUT}$ , the dimension of  $U(t)$  is  $N_{IN}$ .**

variables  $X$  correspond to physical signals (angular positions, spherical-coordinates, etc.) and have positive values constrained into an interval  $[X_{\min} : X_{\max}]$ .

The activation function of the neurons are hyperbolic functions, therefore the network outputs evolve into the interval  $[-1 : 1]$ .

The algebraic expression of this transformation is given by:

$$[X_{\min} : X_{\max}] \rightarrow [X_{\min} : X_{\max}] \quad (1)$$

Let us notice that there is no pre or post-training phase but only an on-line updating of the weights, time parameters and adaptation parameters.

To avoid the problem of persistent excitation and to provide an instantaneous model that adapts itself when plant or environment changes, *AMN* parameters are updated in real time. Since adaptation continues as long as the controller drives the process (it is not our intention to memorize the dynamics of the controlled system). The idea is to compute an instantaneous behavioural model from input and output data of the plant. This instantaneous model is used to automatically update the controller parameters in order to track the process variations. The real time adaptation provides an efficient compensation of the unpredictable process disturbances and sensor.

The autonomous evolution of *AMN* and NC starts from zero values. It results in a compact structure with a small number of nodes ( $N_m = N_{IN} + N_{OUT}$  neurons for *AMN* and  $N_c = 2 \times N_{OUT}$  neurons for NC) [13,17]. The dynamics of  $N_m$  and  $N_c$  neurons are given according to “Equations (2) and (3)” respectively.

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

$$U(k+1) = e^{|\tau(k)|\Delta T} U(k) - \left[ e^{|\tau(k)|\Delta T} - I \right] D(k) \quad (3)$$

where  $B(k) = (B_i(k)) \in \mathbb{R}^{N_m}$  and  $D(k) = (D_i(k)) \in \mathbb{R}^{N_c}$  are considered as constant during each sampling period with:

$$B_i(k) = \tanh \left( \sum_{j=1}^{N_M} W_{ij}(k) \hat{Y}_j(k) + X_i(k) \right) \quad (4)$$

$$D_i(k) = \tanh \left( \sum_{j=1}^{N_c} \Phi_{ij}(k) U_j(k) + Z_i(k) \right) \quad (5)$$

$$X_i(k) = U_i(k) \text{ if } i \in IN \text{ and } X_i(k) = 0 \text{ if } i \in OUT$$

$$Z_i(k) = \begin{cases} \Gamma_i(k) & \text{if } i \in [1, N_{OUT}] \\ \Gamma_{i-N_{OUT}}(k) - Y_{i-N_{OUT}}(k) & \text{if } i \in [N_{OUT} + 1, 2N_{OUT}] \end{cases}$$

$\Gamma_i(k)$  corresponds to the  $i^{\text{th}}$  plant desired output to be tracked.  $\hat{Y}_i(k)$ ,  $W_{ij}(k)$  (resp.  $U_i(k)$ ,  $\phi_{ij}(k)$ ) and  $1/\tau$  represent respectively the  $i^{\text{th}}$  neuron state, weights value from  $j^{\text{th}}$  to  $i^{\text{th}}$  neuron and the adaptive time parameter of *NC* (resp. *AMN*).

The development of autonomous adaptation algorithm parameters are given by “Equations (6), (7) and (8)”.

$$\Delta W_{ij}(k) = -|\eta| \Delta T [Hm^T(k) P_{-ij}(k)] \quad (6)$$

$$\Delta \Phi_{ij}(k) = -|\eta| \Delta T (J_{-} Q_{-ij})^T Hc(k) \quad (7)$$

$$\eta(k) = \eta(k-1) + \Delta T [Hm^T(k) V(k)], \quad (8)$$

$$\tau(k) = \tau(k-1) + \Delta T |\eta| [Hm^T(k) V(k)]$$

where  $Hm(k) = \hat{Y}(k) - Y(k)$  and  $Hc(k) = Y(k) - \Gamma(k)$ .

The notation  $P_{-ij}^T = [P_{1ij}, P_{2ij}, \dots]$  indicates a vector of the 3D matrix  $P$ . The parameters  $|\eta|$  represent the learning rate of both neural networks.  $P_{ij} = \partial \hat{Y}_i / \partial W_{ij}$ ,  $Q_{dij} = \partial U_d / \partial \Phi_{ij}$ ,  $J_{id} = \partial \hat{Y}_i / \partial U_d$  and  $V_i = \partial \hat{Y}_i / \partial \eta$  are the network sensitivity functions.

Stability and robustness properties of the closed loop system are important issues to be addressed. Indeed, small parameter uncertainties and external disturbances can have an unfavourable impact on performance as well as stability. In addition, the dynamic behaviour of the networks can lead to instability of the plant.

The application of the following theorems [11,17-20], guarantee the stability and robustness properties of the closed loop system.

#### Theorem 1

Let  $|\eta|$  be the adaptive updating rate of both the model and control networks, suppose the Lyapunov function is defined by:

$$E(k) = E_m(k) + E_c(k) = \frac{1}{2} \sum_{i \in OUT} (Hm_i^2(k) + Hc_i^2(k))$$

Then the variation of the Lyapunov function could be expressed as  $\Delta E(k) = a|\eta|^2 - 2b|\eta| - c \leq 0$ , where the three parameters  $a$ ,  $b$  and  $c$  depend on the process dynamics.

The sufficient stability condition for the *IDNC* in the sense of Lyapunov should satisfy the following updating rate bounds:

$$0 \leq \frac{b - \sqrt{\Delta'}}{a} \leq |\eta| \leq \frac{b + \sqrt{\Delta'}}{a} \quad (9)$$

where  $\Delta' = b^2 + ac$  depends on the updating procedure and must be non-negative.

#### Theorem 2

Let  $|\eta|$  be the adaptive learning rate of both the monitoring and control networks. Suppose that the sensor noise quickly varies compared to the process dynamics. The *IDNC* compensates the sensor noise, if the following sufficient condition holds:

$$|\eta| \leq \frac{\alpha}{\Delta T \left( \|V(k-1)\| \|V(k)\| + \sum_{i,j} \|P_{-ij}(k-1)\| \|P_{-ij}(k)\| \right)} \quad (10)$$

where  $\alpha$  is an arbitrary threshold depends on the tolerate noise ratio and  $\|\cdot\|$  the usual Euclidean norm.

#### Theorem 3

A sufficient condition on the uncertainty parameters  $(\Delta W, \Delta \tau)$  for robust stability is given by:

$$\max(\|\Delta W\|_{\infty}, \|\Delta W^T\|_{\infty}, |\Delta \tau|) \leq K \quad (11)$$

where  $\|\cdot\|_{\infty}$  is the infinite norm of a matrix and  $K$  a confidence band of uncertainties which depends on the nominal parameters of *AMN*.

**Figure 2** sums up different conditions on the networks parameters that perform the robustness and the stability of the control system. **Theorems 1** and **2** constrain explicitly the evolution of the adaptation parameter  $\eta$  and indirectly the evolution of other parameters. **Theorem 3** describe robustness stability performance

### 3. Control Design for the Manipulator Robot

Our proposed control algorithm is applied on a medical robot designed for dental implantation (**Figure 3**) [21]. The robot is a semi-active mechanical device. It has a passive arm and a motorised wrist with three degrees of freedom that are not convergent. The base of the medical robot is passive, *i.e.* it is not motorised and can be manipulated by the surgeon in any direction.

The aim of the controller is to guide the surgeon so that it will respect the scheduled orientation.

The difficulty of the robot control lies in the calculation of the inverse model that connects the geometric variables to joint variables. Indeed, for a given direction of the robot end-effector, joint variables can be calculated by solving the inverse model equations which permits multiple solutions. According to the actual position of the actuators and the calculated joint variables, the controller must select the optimal solution which prevents large rotation of the actuators. The *IDNC* approximates both the direct and inverse model of the robot by

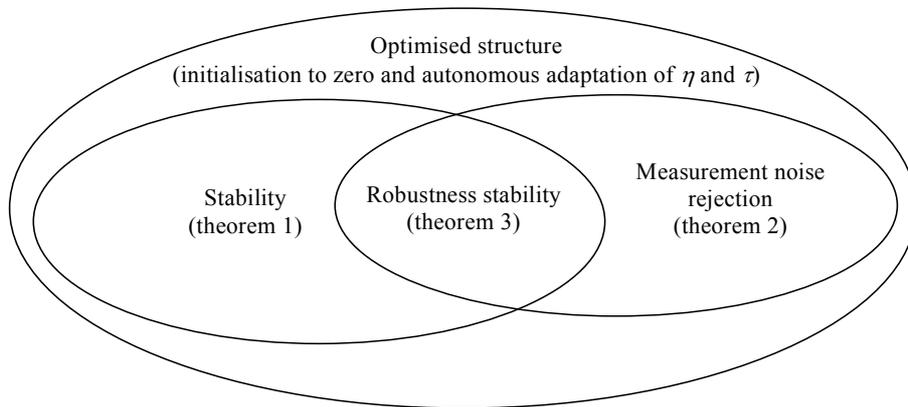


Figure 2. Updating including the different conditions in the network parameters space evolution.



Figure 3. Medical robot.

*AMN* and *NC*. The *IDNC* parameters are adapted in real time in order to minimise a quadratic cost function. This cost function is defined according to the difference between the orientation of the “robot end-effector” vector and the orientation of the “patient” vector and also its variation. Maintain the same orientation between the two vectors are considered as the control objective.

The angular positions of both actuators are considered as plant inputs. The process outputs and desired ones of the “robot end effector” and “patient” vectors are given by the spherical-coordinates ( $\theta$  and  $\beta$ ) (Figure 4).

The nonlinear process is completely unknown for our algorithm which only needs the input and output data and does not require any knowledge about the process model. The autonomous evolution starts from zero initial conditions. It results in a compact structure: the *NC* has six neurons and the *AMN* has four neurons [20].

Figure 5 depicts the tracking performance and angular positions of both actuators obtained with our adaptation algorithm. It can be observed from (Figure 5(d)) that, after a short adaptation stage, due to  $|\eta|$  and  $|\tau|$  parameters start from zero values, the proposed *IDNC* achieves good results ( $NMSE = 3.2 \cdot 10^{-4}$ ). This effect resulted in smoothing action of the learning rate stabilisation constraint, which assures that the closed loop system would

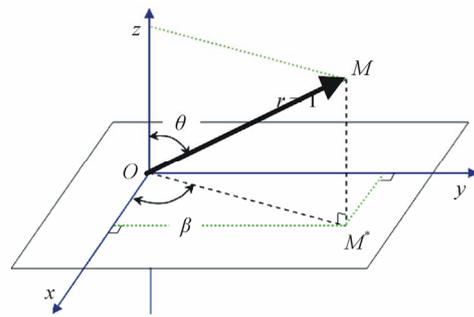


Figure 4. Spherical-coordinates.

converge asymptotically to the reference signal.

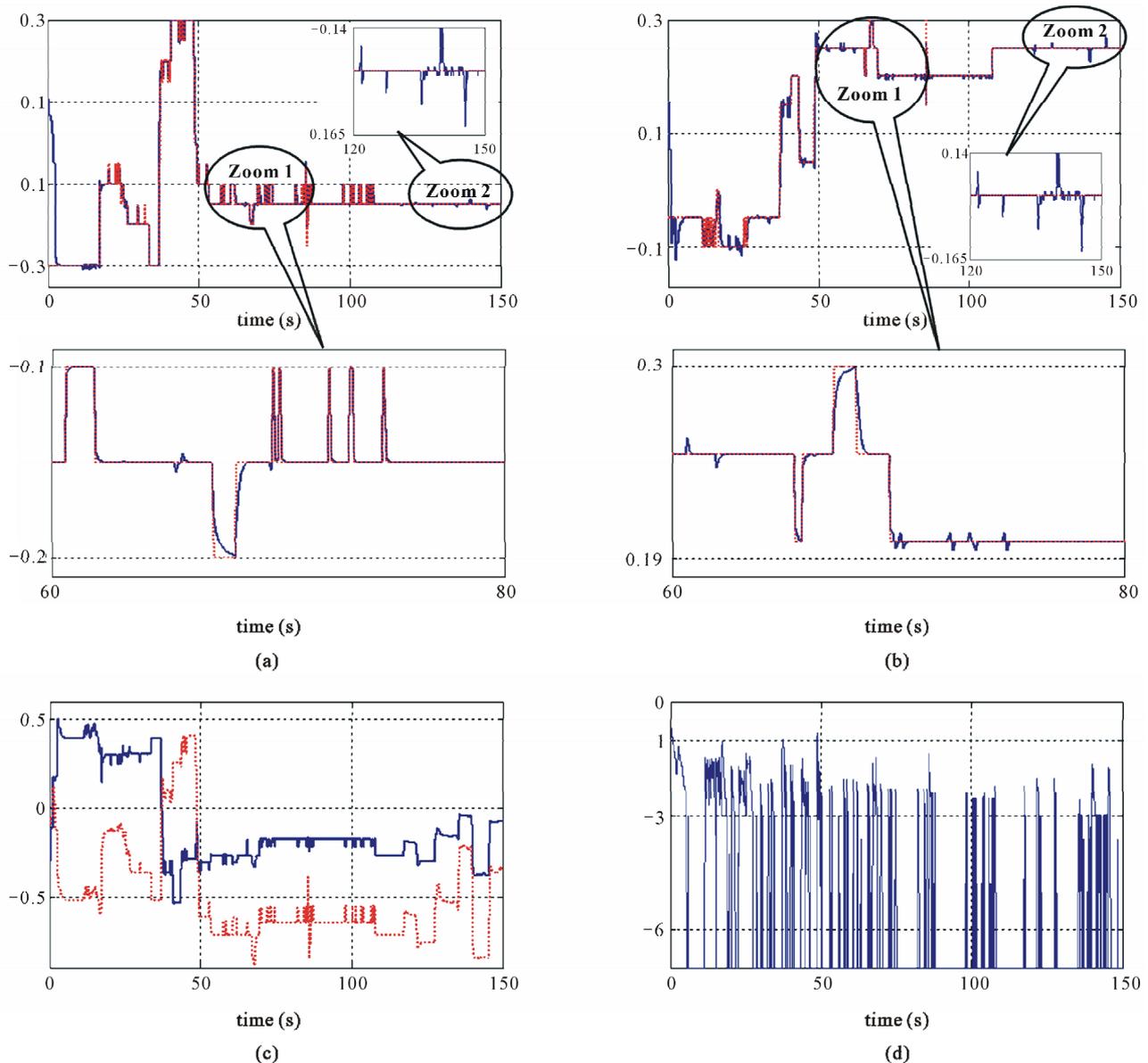
Note that, the angular positions of both actuators, resulting from *NC*, have a smooth form (Figure 5(c)).

We observed fast variations of the  $\theta$  and  $\beta$  spherical-coordinate of the robot end-effector due to measurement noise from the stereo vision cameras (Figure 5(a) and (b): zoom 1). This noise can not be absorbed without filtering.

In order to test the rejection of external disturbances, during the time  $t = 120$  s and  $t = 150$  s, we imposed sudden movement on the end-effector of the robot around the passive SCARA (Figure 3). These movements are considered additive external disturbances on the angular positions of the end-effector. As shown in (Figure 5(a) and (b)) (zoom 2) we can see the *IDNC* can reflect disturbances. Indeed, the perturbations are completely absorbed after a very short time ( $\sim 1$  second).

## 4. Conclusions

Robust stable indirect adaptive control with fully connected neural networks is developed for nonlinear systems with unknown dynamics. A self tuning of the weights, adaptation rates and time parameters of *AMN* and *NC* networks are proposed. The *IDNC* parameters start at zero initial conditions which ensure that its performance does not depend on the initialization phase.



**Figure 5. Experiment results of closed loop system with the adaptation algorithm: (a) and (b)  $\theta$  and  $\beta$  spherical-coordinate of the patient (dashed) and references robot end-effector vectors (solid); (c) angular positions of the first (solid) and second (dashed) actuators; (d) Tracking error in log scale between the angular position of the patient and robot end-effector vectors. Zoom 1: Tracking of the disturbed reference. Zoom 2: Compensation effect of externs perturbations. Range of vertical axis is [-1:1] which reflects angular movements.**

Robustness analysis for closed loop systems is performed and sufficient conditions are obtained. This is due to adaptation rate for stability, noise rejection and robust stability under parameter uncertainties.

Experiment results show that our approach results in good performance, simple structure and self tuning of the parameters.

The method is easy to implement. However, the new stability criterion does not always fulfil global control objectives (such as economic or environmental cost functions) that will be considered in our further works.

## REFERENCES

- [1] W. D. Chang, "Robust Adaptive Single Neural Control for a Class of Uncertain Nonlinear Systems with Input Nonlinearity," *Information Sciences*, Vol. 171, No. 1-3, 2005, pp. 261-271. [doi:10.1016/j.ins.2004.05.001](https://doi.org/10.1016/j.ins.2004.05.001)
- [2] J. Wang, "Sensitivity Identification Enhanced Control Strategy for Nonlinear Process Systems," *Computers & Chemical Engineering*, Vol. 27, No. 11, 2003, pp. 1631-1640. [doi:10.1016/S0098-1354\(03\)00117-0](https://doi.org/10.1016/S0098-1354(03)00117-0)
- [3] F. Fourati, M. Chtourou and M. Kamoun, "Stabilization of Unknown Nonlinear Systems Using Neural Networks,"

- Applied Soft Computing*, Vol. 8, No. 2, 2008, pp. 1121-1130. [doi:10.1016/j.asoc.2007.04.002](https://doi.org/10.1016/j.asoc.2007.04.002)
- [4] D. Wang and P. Bao, "Enhancing the Estimation of Plant Jacobian for Adaptive Neural Inverse Control," *Neurocomputing*, Vol. 34, No. 1-4, 2000, pp. 99-115. [doi:10.1016/S0925-2312\(00\)00319-2](https://doi.org/10.1016/S0925-2312(00)00319-2)
- [5] S. S. Ge and C. Wang, "Direct Adaptive NN Control of a Class of Nonlinear Systems," *IEEE Transactions on Neural Networks*, Vol. 13, No. 1, 2002, pp. 214-221.
- [6] W. Gao and R. R. Selmic, "Neural Network Control of a Class of Nonlinear Systems With Actuator Saturation," *IEEE Transactions on Neural Networks*, Vol. 17, No. 1, 2003, pp. 147-156.
- [7] J. M. Renders, M. Saerens and H. Bersini, "Adaptive Neurocontrol of MIMO Systems Based on Stability Theory," *IEEE Colloquium on Advances in Neural Networks for Control and Systems*, Orlando, 25-27 May 1994, pp. 2476-2481.
- [8] S. S. Ge, C. C. Hang and T. Zhang, "Stable Adaptive Control for Multivariable Nonlinear Systems with a Triangular Control Structure," *IEEE Transactions on Automatic Control*, Vol. 45, No. 6, 2000, pp. 1221-1225.
- [9] S. S. Ge, C. Wang and Y. H. Tan, "Adaptive Control of Partially Known Nonlinear Multivariable Systems Using Neural Networks," *IEEE International Symposium on Proceedings of the Intelligent Control*, Mexico City, 5-7 September 2001, pp. 292-297.
- [10] L. Tian and C. Collins, "A Dynamic Recurrent Neural Network-Based Controller for a Rigid-Flexible Manipulator System," *Mechatronics*, 2004, Vol. 14, No. 5, pp. 471-490. [doi:10.1016/j.mechatronics.2003.10.002](https://doi.org/10.1016/j.mechatronics.2003.10.002)
- [11] S. Zerkaoui, F. Druaux, E. Leclercq and D. Lefebvre, "Indirect Neural Control for Plant-wide Systems: Application to the Tennessee Eastman Challenge Process," *Computers and Chemical Engineering*, Vol. 34, No. 2, 2009, pp. 232-243. [doi:10.1016/j.compchemeng.2009.08.003](https://doi.org/10.1016/j.compchemeng.2009.08.003)
- [12] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, Vol. 1, No. 2, 1989, pp. 270-280. [doi:10.1162/neco.1989.1.2.270](https://doi.org/10.1162/neco.1989.1.2.270)
- [13] E. Leclercq, F. Druaux, D. Lefebvre and S. Zerkaoui, "Autonomous Learning Algorithm for Fully Connected Recurrent Networks," *Neurocomputing*, Vol. 63, 2005, pp. 25-44. [doi:10.1016/j.neucom.2004.04.007](https://doi.org/10.1016/j.neucom.2004.04.007)
- [14] R. M., Sanner and J.-J. E., Slotine, "Gaussian Networks for Direct Adaptive Control," *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, 1992, pp. 837-863.
- [15] D. Wang and P. Bao, "Enhancing the Estimation of Plant Jacobian for Adaptive Neural Inverse Control," *Neurocomputing*, Vol. 34, No. 1-4, 2000, pp. 99-115. [doi:10.1016/S0925-2312\(00\)00319-2](https://doi.org/10.1016/S0925-2312(00)00319-2)
- [16] H. R. Wu and M. Palaniswami, "An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems," *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, 1998, pp. 947-955. [doi:10.1109/72.712168](https://doi.org/10.1109/72.712168)
- [17] S. Zerkaoui, F. Druaux, E. Leclercq and D. Lefebvre, "Commande Adaptative par Réseau de Neurones Hyper-Connectés: Etude de la Stabilité et de la Robustesse," *Journées Doctorales en Modélisation, Analyse et Conduite des Systèmes Dynamiques*, Lyon, 5-7 September 2005, Article ID: 024.
- [18] S. Zerkaoui, F. Druaux, E. Leclercq and D. Lefebvre, "Stable Adaptive Control with Recurrent Neural Networks," *International Federation of Automatic Control*, Prague, 4-8 July 2005, Article ID: 02103.
- [19] S. Zerkaoui, F. Druaux, E. Leclercq and D. Lefebvre, "Robust Stability Analysis for Indirect Neural Adaptive Control," *International Control Conference*, Glasgow, 30 August-1 September 2006, Article ID: 105.
- [20] S. Zerkaoui, F. Druaux, E. Leclercq and D. Lefebvre, "Multivariable Adaptive Control for Non-Linear Systems: Application to the Tennessee Eastman Challenge Process," *European Control Conference*, Kos, 2007.
- [21] R. Chaumont, E. Vasselin, M. Gorka and D. Lefebvre, "Forward Kinematics and Geometric Control of a Medical Robot: Application to Dental Implants," *International Conference on Informatics in Control, Automation and Robotics*, Angers, 9-11 May 2007, pp. 110-115.