

## Adaptive Neural Network Control for a Simple Pendulum Using Backstepping with Uncertainties

Messaoud Mokhtari<sup>1</sup>, Kheireddine Chara<sup>2</sup> and Nouredine Golea<sup>3</sup>

<sup>1</sup>*Department of Electronics, Faculty of Technology, Batna University, Algeria*

<sup>2</sup>*Department of electronics, Faculty of Technology, Setif University, Algeria*

<sup>3</sup>*EE Institute, Oum El-Bouaghi University, Algeria*

*messauod.mokhtari@yahoo.fr<sup>1</sup>, chara.kheireddine@hotmail.fr<sup>2</sup>, nour\_golea@yahoo.fr*

### Abstract

*An adaptive neural network backstepping control for a class of uncertain nonlinear systems is presented in this paper. Three main issues will be treated: (1) unknown nonlinearities; (2) unknown system parameters; (3) external or internal disturbances. The proposed technique is applied to a simple pendulum. This latter is an unstable system which is perfectly described by a nonlinear model obtained by applying physics laws. A solution has to be found to stabilize pendulum in a desired position. A specific type of artificial neural networks (ANN) called "Multilayer Perceptron (MLP)" is used and simulation results clearly demonstrate the power of this extension.*

**Keywords:** *Adaptive, backstepping, disturbances, Multilayer Perceptron (MLP)*

### 1. Introduction

The problems encountered in practice are rarely those which concern approximation of a known function. In the vast majority of cases, it seeks to establish a model of a physical action from the real system, or, in other words, to find a function that imitates conveniently the system behaviour.

When the model structure is known, the detection and location of faults can be performed using identification techniques. The basic idea is to estimate system parameters in real time and compare them with nominal values. For that, a mathematical model of the system to diagnose has to be established and all links between physical parameters and those of the model have to be described, then the identification of the parameters is operated via an adaptive law.

On a theoretical level, [1, 2] explained the development of a direct adaptive controller based on multilayer neural network (MNNs) for a class of nonlinear systems. This work goes on to focus on an adaptive control of strict feedback nonlinear systems using MNNs and backstepping. The application was expanded in [3] where a procedure is developed for the design of an adaptive neural network controller for a class of SISO uncertain nonlinear systems using a pure feedback form. The design procedure is a combination of an adaptive backstepping and neural network-based techniques. In [4], a comparative study between an adaptive backstepping control and an adaptive backstepping sliding mode control for an uncertain system is presented; and parameters variations in the presence of disturbances cases are considered to test the robustness of the two control techniques.

The aforementioned theoretical developments can be applied to physical processes. Here, simple and inverted pendulums represent typical examples for control studies. In literature, a classical backstepping method was applied to the stabilisation of the inverted pendulum [5, 6]. Furthermore, in [7], a design methodology for a novel robust adaptive backstepping controller for the stabilization control of an inverted pendulum on a movable cart has been presented in a systematic manner.

The principal aim of this work is to control a simple pendulum for trajectory tracking in respect of global stability and taking into account certain imposed contextual constraints and conditions.

## 2. Adaptive Neural Network Control

In this work, an adaptive backstepping law is combined to neural network technique in order to control a simple pendulum subject to disturbances. The following scheme depicts the proposed technique and shows the connections between the different parts.

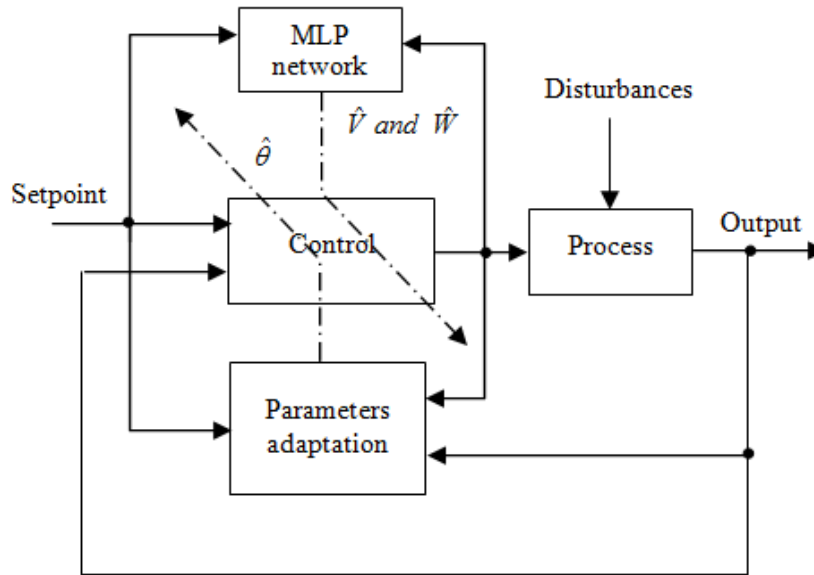


Figure 1. Neural adaptive control diagram

In order to analyze the neural adaptive backstepping control, we consider the following assumptions:

- Nonlinear functions are unknown;
- The system states are measurable;
- System parameters are unknown;
- The system is disturbed.

These considerations allow the system to take the following form:

$$\begin{aligned}
 \dot{x}_1 &= x_2 + \varphi_1(x_1)^T \cdot \theta + \psi_1(x_1)^T + \eta_1(x, t) \\
 \dot{x}_2 &= x_3 + \varphi_2(x_1, x_2)^T \cdot \theta + \psi_2(x_1, x_2)^T + \eta_2(x, t) \\
 &\cdot \\
 &\cdot \\
 \dot{x}_{p-1} &= x_p + \varphi_{p-1}(x_1, \dots, x_{p-1})^T \cdot \theta + \psi_{p-1}(x_1, \dots, x_{p-1})^T + \eta_{p-1}(x, t) \\
 \dot{x}_p &= x_{p+1} + \varphi_p(x_1, \dots, x_p)^T \cdot \theta + \psi_p(x_1, \dots, x_p)^T + \eta_p(x, t) + b_m \cdot u \\
 &\cdot \\
 &\cdot \\
 \dot{x}_{n-1} &= x_n + \varphi_{n-1}(x_1, \dots, x_{n-1})^T \cdot \theta + \psi_{n-1}(x_1, \dots, x_{n-1})^T + \eta_{n-1}(x, t) + b_1 \cdot u \\
 \dot{x}_n &= \beta(x) \cdot u + \varphi_n(x_1, \dots, x_n)^T \cdot \theta + \psi_n(x_1, \dots, x_n)^T + \eta_n(x, t) + b_0 \cdot u \\
 y &= x_1
 \end{aligned} \tag{1}$$

where:  $x_1, x_2, \dots, x_n, y$  and  $u$  denote respectively the states of input and the output of the system,  $\theta \in \mathbb{R}^P$  is a vector of unknown parameters,  $\varphi_i(x): \mathbb{R} \rightarrow \mathbb{R}^P$  is a vector of unknown nonlinear functions that must be approximated by neural networks,  $\eta_i(x, t)$  are the disturbances and  $b_m, b_{m-1}, \dots, b_0$  are unknown constants.

### 3. Control Development of a Simple Pendulum

We can assimilate a manipulator arm to a simple pendulum as shown in Figure 2 below:

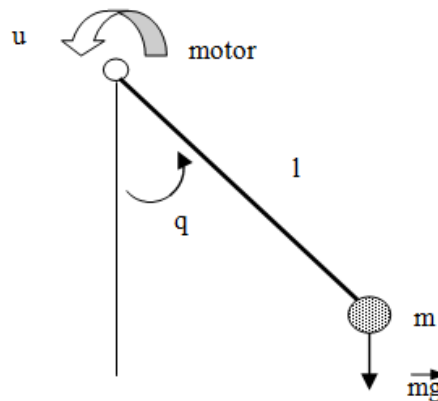


Figure 2. Simple pendulum scheme

The Lagrangian is given by:

$$\begin{aligned}L &= E_C - E_p \\ E_C &= \frac{1}{2} m.l^2.\dot{q}^2 \\ E_p &= m.g.l.(1 - \cos q)\end{aligned}\tag{2}$$

which implies :

$$L = \frac{1}{2} m.l^2.\dot{q}^2 - m.g.l.(1 - \cos q)\tag{3}$$

The differential equations are:

$$\begin{aligned}\frac{\partial L}{\partial \dot{q}} &= m.l^2.\dot{q} \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) &= m.l^2.\ddot{q} \\ \frac{\partial L}{\partial q} &= -m.g.l.\sin q\end{aligned}\tag{4}$$

According to the Lagrange expression, the equation of the system will be expressed by:

$$m.l^2.\ddot{q} + m.g.l.\sin q = u\tag{5}$$

## 4. Development and Procedure Control

### 4.1. Model

To simplify the state representation and to apply the algorithm of backstepping, equation (5) can be written as:

$$\ddot{q} = -\frac{g}{l}.\sin q + \frac{1}{m.l^2} u\tag{6}$$

The following state variables are chosen:

$x_1 = q$  : represent the angular position,

$x_2 = \dot{q}$  : represent the angular velocity.

then the general model is as follows:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l} \cdot \sin x_1 + \frac{1}{m \cdot l^2} \cdot u \\ y &= x_1\end{aligned}\tag{7}$$

The following parameters are used:

$$\theta_1 = -\frac{g}{l} ; \theta_u = \frac{1}{m \cdot l^2}\tag{8}$$

These transformations lead to:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \theta_1 \cdot \varphi(x_1) + \theta_u \cdot u \\ y &= x_1\end{aligned}\tag{9}$$

where the nonlinear function  $\varphi(x_1) = \sin x_1$ .

In the following development, the simple pendulum is supposed to be affected by disturbances. So, equation (9) becomes:

$$\begin{aligned}\dot{x}_1 &= x_2 + \eta_1 \\ \dot{x}_2 &= \theta_1 \cdot \varphi(x_1) + \theta_u \cdot u + \eta_2 \\ y &= x_1\end{aligned}\tag{10}$$

## 4.2. Backstepping procedure

### • Step 1

The following transformations are adopted:

$$z_1 = y - y_r\tag{11}$$

$$z_2 = x_2 - \dot{y}_r - \alpha_1\tag{12}$$

where  $\alpha_1$  is the virtual control law and is yet to be defined.

The first Lyapunov function is defined by:

$$V_1 = \frac{1}{2} z_1^2\tag{13}$$

Its derivative can be written as follows:

$$\begin{aligned}\dot{V}_1 &= z_1 \cdot \dot{z}_1 \\ &= z_1 \cdot (x_2 + \eta_1 - \dot{y}_r)\end{aligned}\quad (14)$$

This leads to:

$$\dot{V}_1 = -c_1 z_1^2 + z_1 \cdot (c_1 z_1 + x_2 + \eta_1 - \dot{y}_r) \quad (15)$$

The first virtual control law (stabilizing function) can be defined by the following expression:

$$\alpha_1 = -c_1 z_1 - h_1 \quad (16)$$

with :  $|\eta_1| \leq h_1$

Then, the dynamics of the error is expressed as:

$$\dot{z}_1 = \dot{x}_1 - \dot{y}_r = z_2 - c_1 z_1 + (\eta_1 - h_1) \quad (17)$$

which gives:

$$\dot{V}_1 = -c_1 z_1^2 + z_1 \cdot z_2 + z_1 \cdot (\eta_1 - h_1) \quad (18)$$

### • Step 2

The Lyapunov function is defined by the following expression:

$$V_2 = V_1 + \frac{1}{2} z_2^2 + \frac{1}{2 \cdot g_1} \tilde{\theta}_1^2 + \frac{1}{2 \cdot g_u} \tilde{\theta}_u^T \cdot \tilde{\theta}_u \quad (19)$$

Its derivative is written as:

$$\dot{V}_2 = -c_1 z_1^2 + z_1 \cdot z_2 + z_1 \cdot (\eta_1 - h_1) + z_2 \cdot \dot{z}_2 + \tilde{\theta}_1 \left( -\frac{1}{g_1} \dot{\hat{\theta}}_1 \right) + \tilde{\theta}_u^T \left( -\frac{1}{g_u} \dot{\hat{\theta}}_u \right) \quad (20)$$

The term  $(z_1 + \dot{z}_2)$  can be developed as follows:

$$(z_1 + \dot{z}_2) = z_1 + \frac{d(\dot{x}_2 - \dot{y}_r - \alpha_1)}{dt} \quad (21)$$

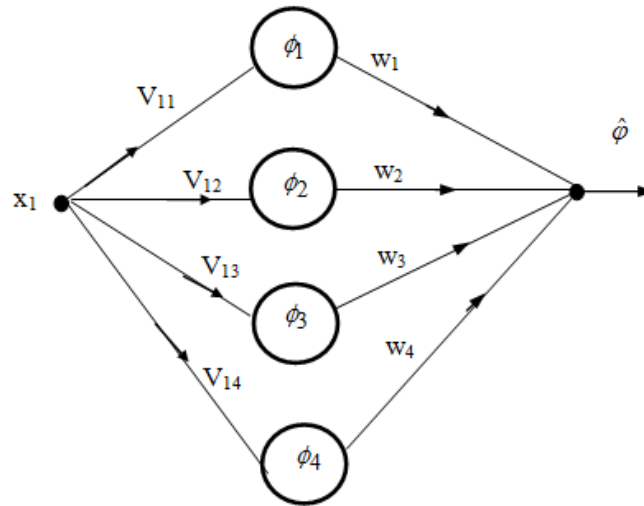
which gives:

$$\begin{aligned} \dot{V}_2 = & -c_1 z_1^2 - c_2 z_2^2 + z_2 [z_1 + c_2 z_2 + \hat{\theta}_1 \varphi(x_1) + \hat{\theta}_u \cdot u + \eta_2 - \dot{\alpha}_1 - \ddot{y}_r] \\ & + \tilde{\theta}_1 \left( -\frac{1}{g_1} \dot{\hat{\theta}}_1 + z_2 \cdot \varphi(x_1) \right) + \tilde{\theta}_u^T \left( z_2 \cdot u - \frac{1}{g_u} \dot{\hat{\theta}}_u \right) + z_1 \cdot (\eta_1 - h_1) \end{aligned} \quad (22)$$

• **Step 3**

The neural network (MLP) in this example is composed of an input layer, a hidden layer (with four neurons) and an output layer. The structure of the MLP network is shown in Figure 3, where  $x_1$  represents the input state,  $\hat{\phi}$  the approximating of nonlinear unknown function,  $\phi_i$  the sigmoidal activation function of the  $i^{\text{th}}$  neuron ( $i = 1, \dots, 4$ ),  $V_{1i}$  the input vector related to the neuron  $i$  and  $x_1$  state,  $W$  the weight vector of the output weight.

where:  $V_i = [V_{1i}, b_i]$ ;  $W = [w_1, w_2, w_3, w_4]^T$ .



**Figure 3. Structure of the MLP network**

Based on the network structure shown in Figure 3, the corresponding expression to the input link I/O can be described by:

$$\hat{\phi} = \sum_{i=1}^4 w_i \cdot \phi_i(V_{1i} \cdot x_1 + b_i) \quad (23)$$

where  $b_i$  is the bias related to the neuron  $i$ . The sigmoidal activation function used in this case is given by the expression:

$$\phi(y) = 1 / [1 + \exp(-\lambda y)], \quad \lambda > 0 \quad (24)$$

and its derivative is defined by the following expression :

$$\dot{\phi}(y) = \lambda\phi(y)[1 - \phi(y)] \quad (25)$$

The nonlinear function which is approximated by the neural network is defined by the expression:

$$\varphi(x_1) = W^T \cdot \phi(V^T \bar{x}_1) + e(x) \quad (26)$$

where:  $e(x)$  is the approximation error,  $\bar{x}_1 = [x_1 \ 1]$  the vector state increased by one.

To overcome the problems caused by the main drawbacks already mentioned in the method of control design "standard backstepping", we propose to change the form given by equation (10) of the system as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 + \eta_1 \\ \dot{x}_2 &= \theta_1 \cdot (W^T \cdot \phi(V^T \bar{x}_1) + e(x)) + \theta_u \cdot u + \eta_2 \\ y &= x_1 \end{aligned} \quad (27)$$

This last step will enable to find the control law. First, we define the relationship between an actual value, its estimation and the estimation error:

$$W = \hat{W} + \tilde{W} \quad \text{and} \quad V = \hat{V} + \tilde{V} \quad (28)$$

The third Lyapunov function is:

$$V_3 = V_2 + \frac{1}{2} \tilde{W}^T A^{-1} \tilde{W} + \frac{1}{2} \tilde{V}^T B^{-1} \tilde{V} \quad (29)$$

A and B are two symmetrical constant matrices  $A=A^T > 0$ ,  $B=B^T > 0$ ,

From equations (26), (27), (28) and (29), the derivative of the Lyapunov function is given by:

$$\begin{aligned} \dot{V}_3 &= -c_1 z_1^2 - c_2 z_2^2 + z_2 \left[ z_1 + c_2 z_2 + \hat{\theta}_1 (W^T \cdot \phi(V^T \bar{x}_1) + e(x)) + \hat{\theta}_u \cdot u + \eta_2 - \dot{\alpha}_1 - \ddot{y}_r \right] \\ &+ \tilde{\theta}_1 \left( -\frac{1}{g_1} \dot{\hat{\theta}}_1 \right) + \tilde{\theta}_1 \cdot z_2 \cdot (W^T \cdot \phi(V^T \bar{x}_1) + e(x)) + \tilde{\theta}_u^T \left( z_2 \cdot u - \frac{1}{g_u} \dot{\hat{\theta}}_u \right) \\ &+ z_1 \cdot (\eta_1 - h_1) - \tilde{W}^T A^{-1} \dot{\hat{W}} - \tilde{V}^T B^{-1} \dot{\hat{V}} \end{aligned} \quad (30)$$



Using the Taylor development, we can write:

$$\phi(\mathbf{V}^T \bar{\mathbf{x}}_1) = \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \quad (31)$$

where  $\mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2$  is the second order error that can be bounded.

The term  $\mathbf{W}^T \cdot \phi(\mathbf{V}^T \bar{\mathbf{x}}_1)$  can be written as follows:

$$\begin{aligned} \mathbf{W}^T \phi(\mathbf{V}^T \bar{\mathbf{x}}_1) &= \mathbf{W}^T \left[ \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \right] \\ &= \hat{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \tilde{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \hat{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \\ &= \hat{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \tilde{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \hat{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 - \tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \hat{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \end{aligned} \quad (32)$$

Using equation (31), the term  $\tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2$  can be written:

$$\begin{aligned} &\tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \\ &= \tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \phi(\mathbf{V}^T \bar{\mathbf{x}}_1) - \mathbf{W}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) - \mathbf{W}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 \\ &= \mathbf{W}^T \left[ \phi(\mathbf{V}^T \bar{\mathbf{x}}_1) - \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \right] - \mathbf{W}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 \\ &= \mathbf{W}^T \left[ \phi(\mathbf{V}^T \bar{\mathbf{x}}_1) - \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \right] + \mathbf{W}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \hat{\mathbf{V}}^T \bar{\mathbf{x}}_1 - \hat{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 \end{aligned} \quad (33)$$

For the sigmoidal activation function, which is continuous and derivable in an interval, we can perform the following constraint:

$$\tilde{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1 + \mathbf{W}^T \mathcal{O}(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}_1)^2 \leq \|\mathbf{W}^T\| \|\dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1)\| + \|\mathbf{V}\|_{\mathbb{F}} \|\bar{\mathbf{x}}_1\| \cdot \hat{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \Big|_{\mathbb{F}} + \|\mathbf{W}^T\|_1 \quad (34)$$

Thus, the following expression is obtained:

$$\begin{aligned} \dot{\mathbf{V}}_3 &\leq -c_1 z_1^2 - c_2 z_2^2 + z_2 [z_1 + c_2 z_2 + \eta_2 + \hat{\theta}_u \cdot \mathbf{u} + \hat{\theta}_1 \cdot \hat{\mathbf{W}}^T \cdot \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + v(t) - \dot{\alpha}_1 - \ddot{y}_r] \\ &+ \tilde{\theta}_1 \left( -\frac{1}{g_1} \dot{\hat{\theta}}_1 + z_2 \hat{\mathbf{W}}^T \cdot \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \right) + \tilde{\theta}_u^T \left( -\frac{1}{g_u} \dot{\hat{\theta}}_u + z_2 \cdot \mathbf{u} \right) \\ &+ \tilde{\mathbf{W}}^T \left( -\mathbf{A}^{-1} \dot{\hat{\mathbf{W}}} + \hat{\theta}_1 \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) z_2 - \hat{\theta}_1 \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \hat{\mathbf{V}}^T \bar{\mathbf{x}}_1 z_2 - \delta \cdot \hat{\mathbf{V}} \right) \\ &+ \tilde{\mathbf{V}}^T \left( -\mathbf{B}^{-1} \dot{\hat{\mathbf{V}}} + \bar{\mathbf{x}}_1 \cdot \hat{\theta}_1 \hat{\mathbf{W}}^T \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) z_2 - \delta \cdot \hat{\mathbf{V}} \right) + z_1 \cdot (\eta_1 - h_1) \end{aligned} \quad (35)$$

where  $\delta$  is a constant positive value.

Using equations (30), (32) and (34), we can deduce the following control law:

$$\mathbf{u} = -\frac{1}{\hat{\theta}_u} \left[ z_1 + c_2 z_2 + \hat{\theta}_u \cdot \hat{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) + \mathbf{h}_2 + v(t) - \dot{\alpha}_1 - y_r^{(2)} \right] \quad (36)$$

where:

$$v(t) = +k \left( \frac{1}{2} + \|\bar{\mathbf{x}}_1\| \cdot \hat{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \right)^2 + \|\phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \hat{\mathbf{V}}^T \bar{\mathbf{x}}_1\|^2 \cdot z_2 + e_M \quad (37)$$

$k$  is a positive value,  $|\eta_2| \leq h_2$  and  $|e(x)| \leq e_M$ .

The update laws are defined by:

$$\begin{aligned} \dot{\hat{\theta}}_1 &= g_1 \cdot z_2 \cdot \hat{\mathbf{W}}^T \cdot \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \\ \dot{\hat{\theta}}_u &= g_u \cdot z_2 \cdot \mathbf{u} \\ \dot{\hat{\mathbf{W}}} &= \hat{\theta}_1 \mathbf{A} \cdot \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) z_2 - \hat{\theta}_1 \mathbf{A} \cdot \dot{\phi}(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) \hat{\mathbf{V}}^T \bar{\mathbf{x}}_1 z_2 - \delta \cdot \mathbf{A} \hat{\mathbf{W}} \\ \dot{\hat{\mathbf{V}}} &= \hat{\theta}_1 \mathbf{B} \cdot \bar{\mathbf{x}}_1 \cdot \hat{\mathbf{W}}^T \phi(\hat{\mathbf{V}}^T \bar{\mathbf{x}}_1) z_2 - \delta \cdot \mathbf{B} \cdot \hat{\mathbf{V}} \end{aligned} \quad (38)$$

Finally, the derived Lyapunov function takes the following form:

$$\dot{V}_3 \leq -\sum_{j=1}^2 c_j z_j^2 + z_1 \cdot (\eta_1 - h_1) \quad (39)$$

## 5. Simulation Results

### 5.1. Adaptive Neural Network Control - Regulation-( $\gamma_r=\pi/6$ )

The aim is to adjust the angle of the rod. In this neural-based test, the selected parameters of the pendulum are:  $m=0.1$  kg,  $l=0.23$ m.  $g=9.81$  m/s<sup>2</sup>. The system is perturbed by sinusoidal signals  $\eta_1$  and  $\eta_2$  respectively limited by  $h_1 =0.3$  and  $h_2 =0.2$ . In this approach, the reference angle of the rod is equal to  $\pi/6$ . The selected parameters of the pendulum are: Adaptation gains are  $c_1=2$ ,  $c_2=10$ . Spline parameter adaptation gain is  $g_1=20$  and load parameter adaptation gain is  $g_u=10$ .

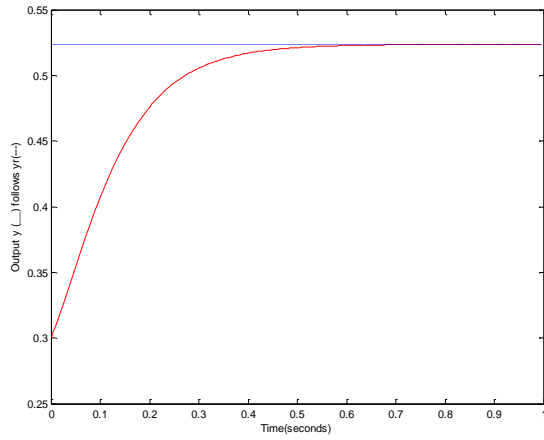


Figure 4. Desired output tracking

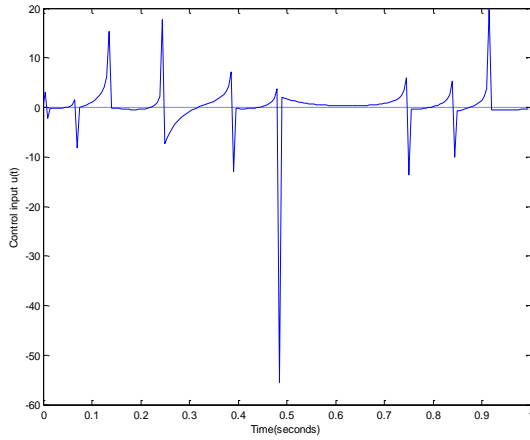


Figure 5. The control law

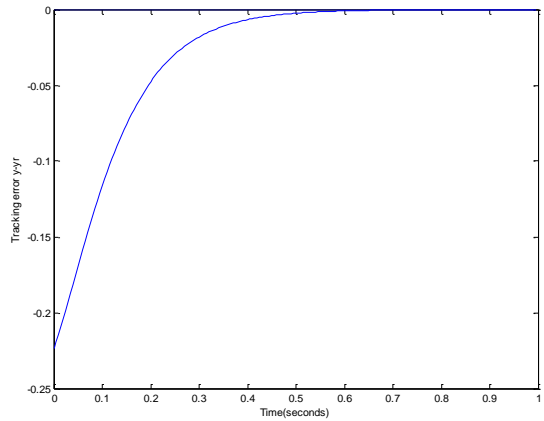


Figure 6. Tracking error

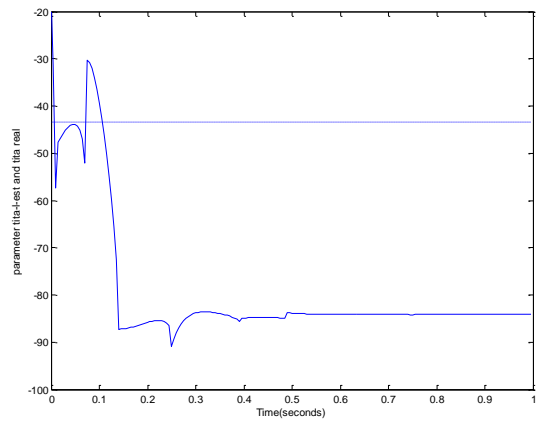


Figure 7. Estimation of the parameter  $\theta_1$

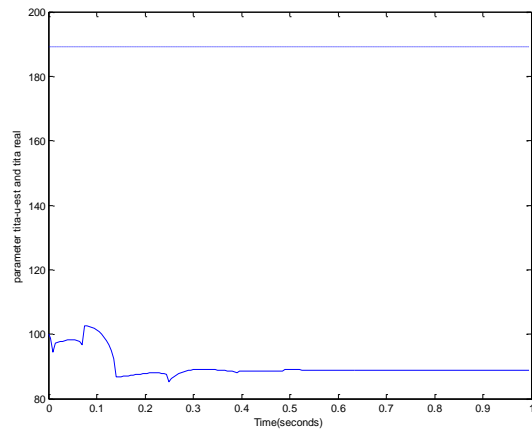


Figure 8. Estimation of the parameter  $\theta_u$

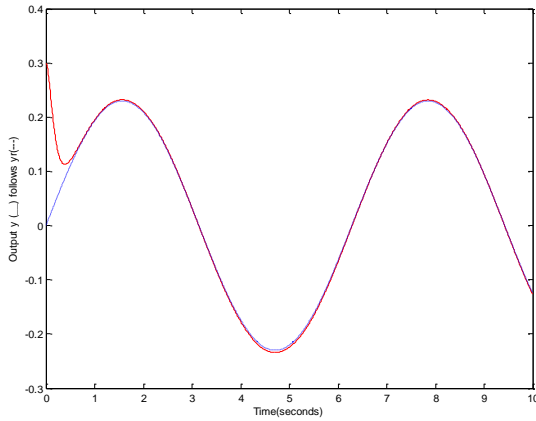
It is clearly seen that the error tends to zero and the output “y” follows perfectly the reference trajectory “y<sub>r</sub>”. These results reflect the efficiency of the proposed technique.

**5.2. Adaptive Neural Network Control -Variable reference-(y<sub>r</sub>=0.23.sin(2π.t))**

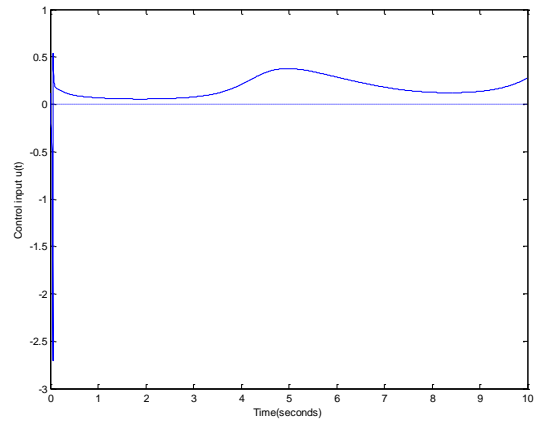
In order to test the robustness of the proposed control technique, a variable setpoint is imposed. Sinusoid signals  $\eta_1$  and  $\eta_2$  are respectively limited by  $h_1 = 0.0089$  and  $h_2 = 0.0059$ . The selected parameters of the pendulum are: Adaptation gains are  $c_1=c_2=10$ . Spline parameter adaptation gain is  $g_r=20$  and load parameter adaptation gain is  $g_u=10$ .

The network used contains four neurons in hidden layer and all the biases are also estimated with weights by neural backstepping techniques.

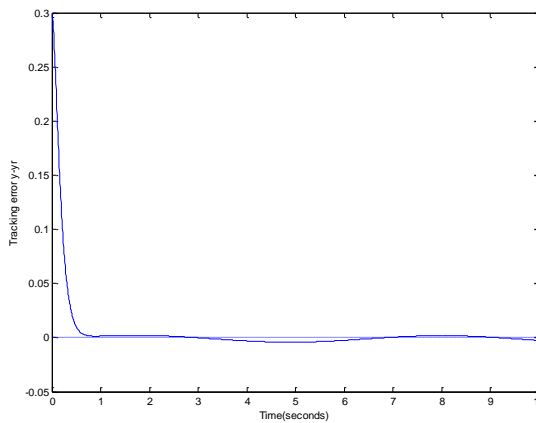
The performance of the neural adaptive control law has been verified by assuming that the nonlinear functions are also unknown. The results represented by the following figures are obtained.



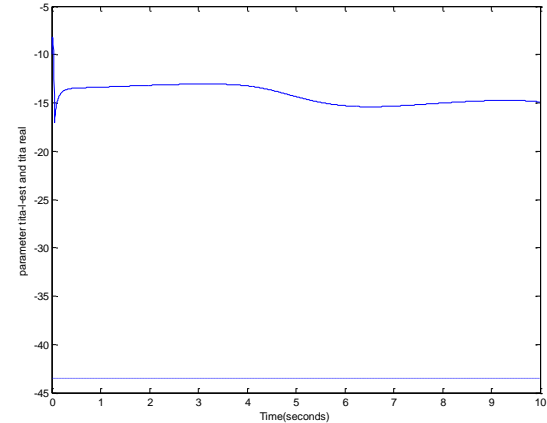
**Figure 9. Desired variable output Tracking**



**Figure 10. The control law**



**Figure 11. Tracking error**



**Figure 12. Estimation of the parameter  $\theta_1$**

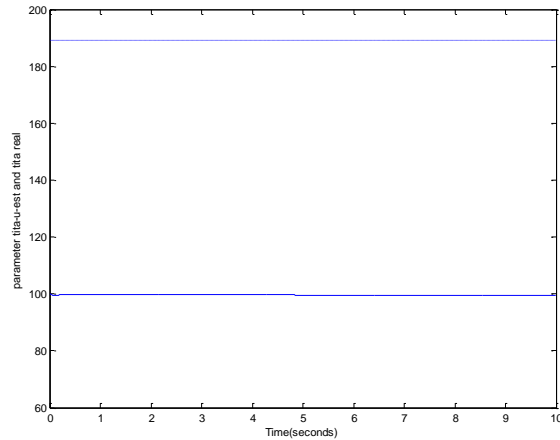


Figure 13. Estimation of the parameter  $\theta_u$

## 6. Conclusion

In this paper, an adaptive neural network control based on backstepping technique is synthesized and applied to a simple pendulum in the presence of model uncertainties and disturbances.

In addition, the proposed technique permits to obtain a robust control law that permits to achieve variable setpoint tracking case in respect of a good time response. The control law behaviour is smooth and shows reasonable amplitude.

The error convergence and the total stability of the controller system express well the advantage of this control strategy. Besides, the estimation of the unknown parameters of the considered plant is well accepted.

## References

- [1] T. Zhang, S. S. Ge and C. C. Hang, "Adaptive Neural Network Control for Strict-feedback Nonlinear Systems Using Backstepping Design", *Automatica*, vol. 36, (2000) May, pp. 1835-1846.
- [2] T. Zhang, S. S. Ge and C. C. Hang, "Design and Performance Analysis of a Direct Adaptive Controller for Nonlinear Systems", *Automatica*, vol. 35, (1999), pp. 1809-1817.
- [3] D. Wang and J. Huang, "Adaptive Neural Network Control for a Class of Uncertain nonlinear Systems in Pure-feedback Form", *Automatica*, vol. 38, (2002), pp. 1365-1372.
- [4] D. Elleuch and T. Damak, "Robust Adaptive Controller for Uncertain Nonlinear Systems", *International Journal of Information Technology, Control and Automation (IJITCA)* vol. 2, no. 2, (2012).
- [5] A. Benaskeur and A. Desbiens, "Application of the Inverted Pendulum", *Electrical and Computer Engineering IEEE, Canadian Conference*, vol. 1, (1998) May 24-28, pp. 113-116.
- [6] Ö. T. Altınöz, "Adaptive Integral Backstepping Motion Control for Inverted Pendulum", *International Journal of Mechanical, Industrial Science and Engineering (IJMISE)* vol. 1, no. 5, (2007).
- [7] S. Rudra and R. K. Barai, "Robust Adaptive Backstepping Control of Inverted Pendulum on Cart System", *International Journal of Control and Automation*, vol. 5, no. 1, (2012).
- [8] M. Kristić, I. Kanellakopoulos and P.V. Kokotović, "Nonlinear and adaptive control design", John Wiley and Sons, New York, (1995).
- [9] T. Knoho and H. Unbehauen, "ANNNAC-extension of adaptive backstepping algorithm with artificial neural networks", *IEE Proc., Control Theory and Application*, vol. 147, (2000), pp.177-183.
- [10] I. Kanellakopoulos, P. V. Kokotović, and R. H. Middleton, "Indirect adaptive output-feedback control of a class of nonlinear systems", *Proceedings of IEEE Conference on Decision and Control, Honolulu, Hawaii*, (1990), pp. 2714-2719.
- [11] J. Zhou and C. Wen, "Adaptive Backstepping Control of Uncertain Systems", Springer-Verlag, Berlin, Heidelberg, (2008).

