

Adaptive Internal Model Control of a DC Motor Drive System Using Dynamic Neural Network

Farouk Zouari, Kamel Ben Saad, Mohamed Benrejeb

Unité de Recherche LARA Automatique, Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunis, Tunisia.
Email: zouari.farouk@gmail.com, {[kamel.bensaad](mailto:kamel.bensaad@enit.rnu.tn), [mohamed.benrejeb](mailto:mohamed.benrejeb@enit.rnu.tn)}@enit.rnu.tn

Received December 13th, 2011; revised January 20th, 2012; accepted February 10th, 2012

ABSTRACT

This work concerns the study of problems relating to the adaptive internal model control of DC motor in both cases conventional and neural. The most important aspects of design building blocks of adaptive internal model control are the choice of architectures, learning algorithms, and examples of learning. The choice of parametric adaptation algorithm for updating elements of the conventional adaptive internal model control shows limitations. To overcome these limitations, we chose the architectures of neural networks deduced from the conventional models and the Levenberg-marquardt during the adjustment of system parameters of the adaptive neural internal model control. The results of this latest control showed compensation for disturbance, good trajectory tracking performance and system stability.

Keywords: Adaptive Internal Model Control; Recurrent Neural Network; DC Motor; Parametric Adaptation Algorithm; Levenberg-Marquardt

1. Introduction

During recent decades, adaptive internal model control has been studied in several research works of which we mention [1-11]. It has been exploited in several industrial fields. It is usually interesting for its performances in servo and control where systems to be controlled are dynamic, complex, finite dimensional, open-loop stable and in addition if they have numerous delays and disruptions.

Another advantage of the structure of this control lies in its simple construction and easy interpretation of the roles of its building blocks. It includes an internal model which is an explicit process model to be controlled, a controller which can be chosen the inverse of this model and, if necessary, robustness filters.

The modeling process is to find a model whose dynamic behavior of the process approach based either on theoretical analysis, either on an experimental analysis, or on theoretical and experimental analysis. This model will be used to make predictions of the output of the process for learning the controller, and even to simulate the processes within the control system [12-16].

The inversion of model is one of the main problems of the approach of adaptive internal model control, since the direct inversion of the model for most physical systems provides an unrealizable structure (systems characterized by a function of transfer with the order of numerator less than the order of the denominator, the systems of non-minimum phase systems with delay, etc.). In this context,

our work presents solutions that enable the design of a controller who comes to the best of the inverse model [17-23].

In the case of a model which is not perfect, a robust filter is useful to avoid destabilization of the command structure in the presence of modeling errors and/or major disturbances [24]. The robustness filter is usually synthesized based on the Nyquist criterion [25]. The method of synthesis of this filter is based on small gain theorem [26, 27]. By cons, this filter does not affect system stability but slows down the system in the case of perfect model [28].

Thus neural network based control systems which have the desirable proprieties of nonlinear mapping, generalization and learning can offered as candidates for solution to high performance electrical drives.

Our objective is to apply a neural network adaptive control scheme to a Drive Motor system. The proposed scheme can control the speed of the considered Motor Drive system to tack the reference speed with fast and damped response.

The rest of the paper is organized as follows: In Section 2, we present a description discrete time models of the considered DC Motor Drive. The adaptive conventional internal model control scheme application of the considered system is developed in Section 3. The Section 4, provide the neural adaptive internal model control scheme to our system. The stability analysis of these two

adaptive internal model control systems is developed in Section 5. A comparative study between these two control schemes is illustrated in Section 6 and a conclusion is drawn in Section 7.

2. Description of DC Motor

A DC motor can be used in a variety of industrial applications [29]. The type DC motor is characterized by the following equations [30]:

The electrical equation:

$$U(t) = E(t) + Ri(t) + L \frac{di(t)}{dt} \quad (1)$$

The mechanical equation:

$$C_m(t) = C_r(t) + J \frac{d\omega(t)}{dt} + f\omega(t) \quad (2)$$

The electromechanical coupling equations:

$$E(t) = K_e \omega(t) \quad (3)$$

$$C_m(t) = K_m i(t) \quad (4)$$

with:

$U(t)$ is the armature voltage (V), $E(t)$ the back EMF (V), R the armature resistance (Ω), $i(t)$ the armature current (A), L the armature inductance (H), $C_m(t)$ the motor torque (N.m), $C_r(t)$ the load torque (N.m), J the rotor inertia ($\text{kg}\cdot\text{m}^2$), f the viscous friction coefficient ($\text{N}\cdot\text{m}\cdot\text{s}\cdot\text{rad}^{-1}$), $\omega(t)$ the rotor speed ($\text{rad}\cdot\text{s}^{-1}$), K_e the back EMF ($\text{V}\cdot\text{s}\cdot\text{rad}^{-1}$), K_m is the back torque ($\text{V}\cdot\text{s}\cdot\text{rad}^{-1}$).

From the above equations, the DC motor is schematically as follows (**Figure 1**):

with: Laplace transforms of $U(t)$, $C_m(t)$, $C_r(t)$ and $\omega(t)$ are respectively $U(s)$, $C_m(s)$, $C_r(s)$ and $\Omega(s)$; s is a Laplace variable.

The transfer function between the input and output of the DC motor can be written as follows:

$$\Omega(s) = \frac{K_m U(s) - C_r(Ls + R)}{LJs^2 + (Lf + RJ)s + (Rf + K_e K_m)} \quad (5)$$

The study of the stability of the system can be made by the Routh criterion (or Routh-Hurwitz) which is to form the following table (**Table 1**).

Whether $D(s) = LJs^2 + (Lf + RJ)s + (Rf + K_e K_m)$

with:

$$d_0 = - \frac{1}{(Lf + RJ)} \begin{vmatrix} LJ & (Rf + K_e K_m) \\ (Lf + RJ) & 0 \end{vmatrix} \quad (6)$$

The terms LJ , $(Lf + RJ)$ and d_0 are strictly positive so this system is stable.

From the Equation (5), the load torque can be considered as a perturbation.

The discrete time model system can be calculated by

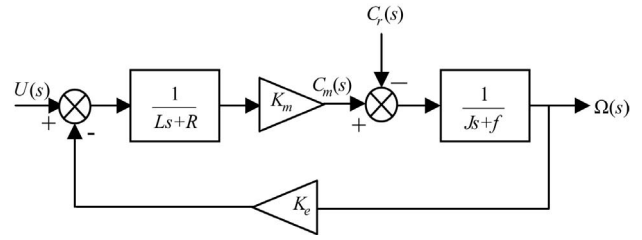


Figure 1. Block diagram of a DC motor.

Table 1. Table of coefficients.

	LJ	$(Rf + K_e K_m)$
s^2		
s^1	$(Lf + RJ)$	0
s^0	d_0	

replacing in Equation (5) s by the approximate expression z :

$$s = \frac{z-1}{zT_s} \quad (7)$$

Hence the expression of discrete system:

$$\Omega(z) = \frac{(zT_s)^2 K_m U(z) - C_r(L(z-1)zT_s + R(zT_s)^2)}{LJ(z-1)^2 + (Lf + RJ)(z-1)zT_s + (Rf + K_e K_m)(zT_s)^2} \quad (8)$$

3. Adaptive Conventional Internal Model Control System of DC

The basic structure of the adaptive conventional internal model control with robustness filter of DC motor is shown in figure 2 [31].

where: $M_{n_a, n_b, n_c, n_k}(z)$ a model parameters of the process (n_a, n_b, n_c, n_k), $r(k)$ the reference signal at instant k , $y(k)$ the system output, $C(z)$ the controller, $\hat{y}(k)$ the model output, $\varepsilon_1(k)$ the difference between the reference signal and the output system, $\varepsilon(k)$ the difference between the output of the system and that of the model, $u(k-1)$ the command, $F(z)$ the filter robustness, $r^*(k)$ the input of the controller, $v(k)$ the disturbance affecting the output system such as $|v(k)| \leq 10^{-1}$.

The **Figure 2** can be redrawn as follows (**Figure 3**).

For having $\varepsilon_1(k) = 0$ and robustification in the sense of stability margin, the controller, the model and the filter robustness must satisfy the following relations:

$$F(1)C(1)M_{n_a, n_b, n_c, n_k}(1) = 1 \quad (9)$$

$$F(1) = 1 \quad (10)$$

The robustness filter is often taken as first order:

$$F(z) = \frac{(1-\alpha)}{1-\alpha z^{-1}} \quad (11)$$

with $0 < \alpha < 1$

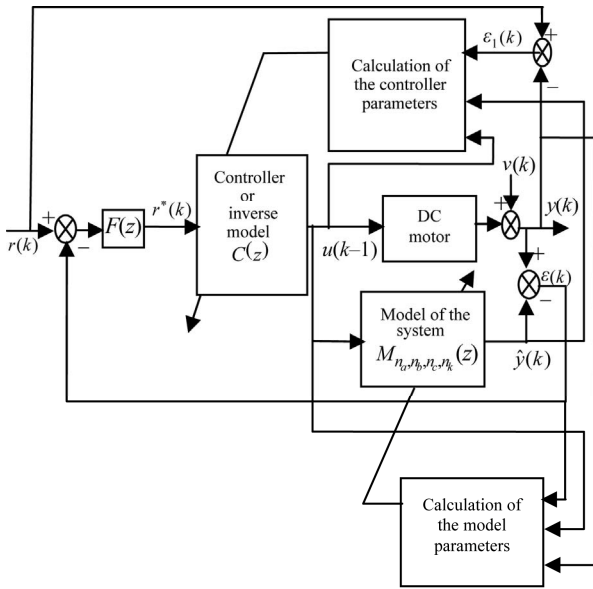


Figure 2. Basic structure of the adaptive conventional internal model control of DC motor with robustness filter.

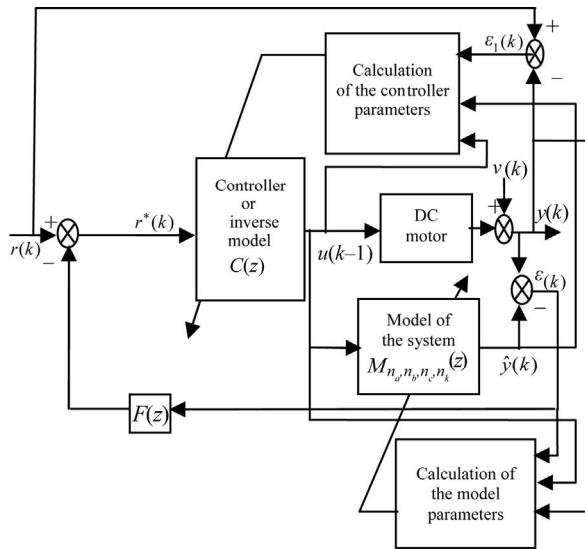


Figure 3. Structure of adaptive conventional internal model control with robustness filter in the feedback loop.

In our work, we fix $\alpha = 10^{-3}$.

The disturbance affects both the output and system status. The model of DC motor which is of type ARMAX (Autoregressive moving average model with exogenous inputs model), is therefore given by the following equation:

$$\hat{y}(k) = \sum_{i=1}^{n_a} A_i y(k-i) + \sum_{i=1}^{n_b} b_i u(k-n_k-i) + \sum_{i=1}^{n_c} c_i \rho(k-i) \quad (12)$$

$\rho(k)$ is a noise. It can be replaced by their estimated

value a priori or a posteriori:

$$\rho(k) = y(k) - \hat{y}(k) \quad (13)$$

From Equation (12), we obtain:

$$\hat{y}(k) = \sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} b_i u(k-n_k-i) - \sum_{i=1}^{n_c} c_i \hat{y}(k-i) \quad (14)$$

with: $a_i = A_i + c_i$

The model of the motor is as follows:

$$\hat{y}(z) = \frac{\left(\sum_{i=1}^{n_a} a_i z^{-i} \right) y(z) + \left(\sum_{i=1}^{n_b} b_i z^{-n_k-i} \right) u(z)}{1 + \sum_{i=1}^{n_c} c_i z^{-i}} \quad (15)$$

The relations $1 + \sum_{i=1}^{n_c} c_i z^{-i}$ and $\left(\sum_{i=1}^{n_b} b_i z^{-n_k-i} \right)$ can be written:

$$1 + \sum_{i=1}^{n_c} c_i z^{-i} = z^{-n_c} \prod_{i=1}^{n_c} (z - Zc_i) \quad (16)$$

$$\left(\sum_{i=1}^{n_b} b_i z^{-n_k-i} \right) = b_1 z^{-n_b-n_k} \prod_{i=1}^{n_b-1} (z - Zb_i) \quad (17)$$

with: $\{Zb_1, \dots, Zb_{n_b-1}\}$ and $\{Zc_1, \dots, Zc_{n_c}\}$ are respectively the zeros and poles of the model.

If $|Zb_1|, \dots, |Zb_{n_b-1}|$ are less than or equal to 1, the inverse model of the system expressed by:

$$u(k-1) = -\sum_{i=1}^{n_a} \frac{a_i}{b_1} y(k+n_k-i) - \sum_{i=2}^{n_b} \frac{b_i}{b_1} u(k-i) + \frac{1}{b_1} \hat{y}(k+n_k) + \sum_{i=1}^{n_c} \frac{c_i}{b_1} \hat{y}(k+n_k-i) \quad (18)$$

This inverse model is used in adaptive conventional internal model control system replacing the $\hat{y}(k)$ by $r^*(k)$. In this case, the control applied to DC motor is given by:

$$u(k-1) = -\sum_{i=1}^{n_a} \frac{a_i}{b_1} y(k+n_k-i) - \sum_{i=2}^{n_b} \frac{b_i}{b_1} u(k-i) + \frac{1}{b_1} r^*(k+n_k) + \sum_{i=1}^{n_c} \frac{c_i}{b_1} r^*(k+n_k-i) \quad (19)$$

From the above equation, we can deduce:

$$u(z) = \frac{\left(-\sum_{i=1}^{n_a} a_i z^{n_k-i} \right) y(z) + \left(1 + \sum_{i=1}^{n_c} c_i z^{n_k-i} \right) r^*(z)}{\sum_{i=1}^{n_b} b_i z^{-i}} \quad (20)$$

If $|Zb_1|, \dots, |Zb_{n_b}|$ are strictly greater than 1 such as

$n_0 \leq n_b - 1$, the system is unstable. One method of inversion that proposed in [32] is to partition the model into two parts: one part contains the stable zeros and the other contains the unstable zeros. In this case, the controller is chosen as the inverse of the stable part. In addition, Frank imposes a transfer equal to unity for the unstable part.

Such as:

$$\lim_{z \rightarrow 1} \frac{\prod_{i=1}^{n_0} (z - Zb_i)}{\prod_{i=1}^{n_0} (1 - Zb_i) z^{n_0}} = 1 \quad (21)$$

So we multiply the relation (20) by $\frac{\prod_{i=1}^{n_0} (z - Zb_i)}{\prod_{i=1}^{n_0} (1 - Zb_i) z^{n_0}}$

to ensure stability. Equation (20) becomes:

$$u(z) = \frac{\left(-\sum_{i=1}^{n_a} a_i z^{n_k - i} \right) y(z) + \left(1 + \sum_{i=1}^{n_c} c_i z^{n_k - i} \right) r^*(z)}{b_1 \left(\prod_{i=1}^{n_0} (1 - Zb_i) \right) \left(\prod_{i=n_0+1}^{n_b-1} (z - Zb_i) \right) z^{-n_b + n_0}} \quad (22)$$

For direct and inverse models are physically feasible, the parameters n_a, n_b, n_c and n_k are chosen:

$$\begin{cases} n_c = n_b \\ n_a \leq n_b \\ n_k \geq 0 \end{cases} \quad (23)$$

It is possible to rewrite the Equations (14) and (19) under the following matrix forms:

$$\hat{y}(k) = \theta^T \psi(k) \quad (24)$$

$$u(k-1) = \theta c^T \varphi(k) \quad (25)$$

In which the vectors of parameters and observations are defined by:

$$\theta^T = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}, -c_1, \dots, -c_{n_c}] \quad (26)$$

$$\theta c^T = \frac{1}{b_1} [-a_1, \dots, -a_{n_a}, -b_2, \dots, -b_{n_b}, 1, c_1, \dots, c_{n_c}] \quad (27)$$

$$\begin{aligned} \psi(k) &= [y(k-1), \dots, y(k-n_a), u(k-n_k-1), \\ &\quad \dots, u(k-n_k-n_b), \hat{y}(k-1), \dots, \hat{y}(k-n_c)]^T \\ \varphi(k) &= [y(k+n_k-1), \dots, y(k+n_k-n_a), u(k-2), \\ &\quad \dots, u(k-n_b), r^*(k+n_k), \dots, r^*(k+n_k-n_c)]^T \end{aligned} \quad (28)$$

The vectors of observations written yet in the forms:

$$\psi^T(k) = [\psi_1(k), \dots, \psi_{n_r}(k)] \quad (29)$$

$$\phi^T(k) = [\phi_1(k), \dots, \phi_{n_r}(k)] \quad (30)$$

with $n_r = n_a + n_b + n_c$

The number of parameters to adjust in this command structure $n_\beta = 2 * n_r$.

The adjustment procedure of the vector of model parameters θ at each sampling period can be made by the parametric adaptation algorithm to minimize the following quadratic criterion:

$$J_r = \frac{1}{2} (y(k) - \hat{y}(k))^2 \quad (31)$$

This algorithm which is part of the simple gradient methods can be expressed in terms of criterion (32) as follows:

$$\hat{\theta}(k) = \hat{\theta}(k-1) - P(k) \frac{\partial J_r(k)}{\partial \hat{\theta}(k-1)} \quad (32)$$

According to [33-35], this procedure is given by the system of equations:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \psi^0(k) e(k) \quad (33)$$

$$\lambda_1(k) = \lambda_0 \lambda_1(k-1) + (1 - \lambda_0) \quad (34)$$

$$\lambda_2(k) = 1 \quad (35)$$

$$\begin{aligned} P(k) &= \frac{1}{\lambda_1(k)} \left[P(k-1) - \frac{P(k-1) \psi(k) \psi^T(k) P(k-1)}{\lambda_1(k) + \psi^T(k) P(k-1) \psi(k)} \right] \end{aligned} \quad (36)$$

$$\hat{y}(k/k-1) = \hat{y}^0(k) = \hat{\theta}^T(k-1) \psi^0(k) \quad (37)$$

$$e(k) = y(k) - \hat{y}^0(k) \quad (38)$$

with:

$$\hat{\theta}(k) = \begin{bmatrix} \hat{a}_1(k) \\ \vdots \\ \hat{a}_{n_a}(k) \\ \hat{b}_1(k) \\ \vdots \\ \hat{b}_{n_b}(k) \\ -\hat{c}_1(k) \\ \vdots \\ -\hat{c}_{n_c}(k) \end{bmatrix}, \psi^0(k) = \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-n_a) \\ u(k-n_k-1) \\ \vdots \\ u(k-n_k-n_b) \\ \hat{y}^0(k-1) \\ \vdots \\ \hat{y}^0(k-n_c) \end{bmatrix} \quad (39)$$

Initialization for the matrix $P(k)$, the parameter vector $\hat{\theta}(k)$, the growth factor $\lambda_1(k)$ and the decay factor $\lambda_2(k)$ is often as follows:

$$\hat{\theta}(0) = [0, \dots, 0]^T \quad (40)$$

$$P(0) = \alpha_1 I \quad (41)$$

I being the identity matrix of dimensions (n_r, n_r) and α_1 can be chosen equal to $\approx 10^3$ according to Landau.

$$\lambda_0 = \lambda_1(0) = 0.95 \quad (42)$$

$$\lambda_2(0) = 1 \quad (43)$$

The model output can be written as follows:

$$\hat{y}(k) = \hat{\theta}^T(k) \psi(k) \quad (44)$$

The difference between the system output and the model output can be expressed by the Equation (45):

$$\varepsilon(k) = y(k) - \hat{y}(k) \quad (45)$$

The input of the equalizer is given by the following equation:

$$r^*(k) = (1 - \alpha)(r(k) - \varepsilon(k)) + \alpha r^*(k-1) \quad (46)$$

From Equation (27), the vector of the controller parameters $\hat{\theta}_c(k)$ estimated at time k is defined by:

$$\hat{\theta}_c(k) = \frac{1}{\hat{b}_1(k)} [-\hat{a}_1(k), \dots, -\hat{a}_{n_a}(k), -\hat{b}_2(k), \dots, -\hat{b}_{n_b}(k), 1, \hat{c}_1(k), \dots, \hat{c}_{n_c}(k)]^T \quad (47)$$

The command applied to the DC motor is given by:

$$u(k-1) = \hat{\theta}_c^T(k) \varphi(k) \quad (48)$$

The difference between the reference signal and the system output is determined by:

$$\varepsilon_1(k) = r(k) - y(k) \quad (49)$$

4. Adaptive Neural Internal Model Control System of DC

The structures of internal model control using nonlinear neural networks have been proposed in [36-38]. The **Figure 4** illustrates the structure of an adaptive neural internal model control of DC motor, where the model and the controller are replaced by recurrent neural networks with multilayer internal and external closure.

The choice of the structures of neural networks takes into account the knowledge and assumptions about the behavior of the system [39-43]. The **Figures 5** and **6** present respectively the architecture of the model and the neuronal controller.

where:

$f_1(x) = \frac{e^{-x} - 1}{1 + e^{-x}}$ and $f_2(x)$ are the activation functions of neurons,

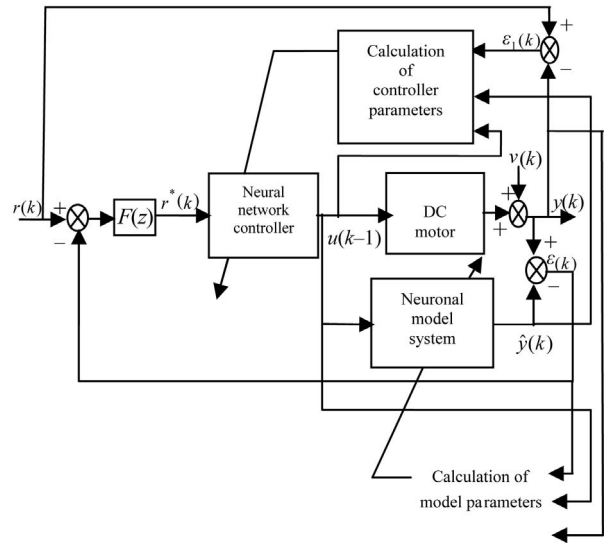


Figure 4. Basic structure of adaptive neural internal model control with filter robustness of DC motor.

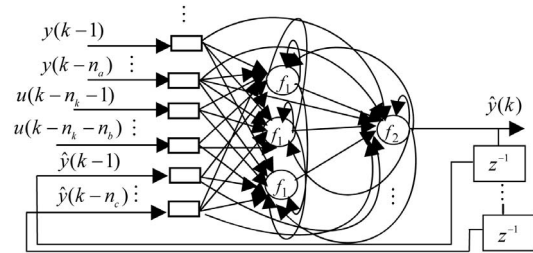


Figure 5. Neural model of the DC motor.

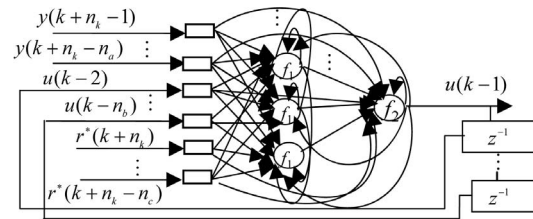


Figure 6. Neural controller of the DC motor.

n_h the number of neurons in the hidden layer respectively of the model and controller,

The coefficients of the vector of parameters of the neural model w are decomposed into seven groups, formed respectively by:

$$w^1 = \begin{bmatrix} w_{11}^1 & \dots & w_{1n_r}^1 \\ \vdots & \ddots & \vdots \\ w_{n_h 1}^1 & \dots & w_{n_h n_r}^1 \end{bmatrix} \quad \text{the weights between neurons}$$

in the input layer and neurons in the hidden layer,

$$w^2 = \begin{bmatrix} w_{11}^2 \\ \vdots \\ w_{n_h 1}^2 \end{bmatrix} \quad \text{the biases of neurons in the hidden layer,}$$

$w^3 = [w_{11}^3 \cdots w_{1n_h}^3]$ the weights between neurons in the hidden layer neurons and output layer,

$w^4 = [w_{11}^4]$ the bias of neuron in the output layer,

$w^5 = [w_{11}^5 \cdots w_{1n_r}^5]$ the weights between neurons of input layer neurons and output layer,

$w^6 = \begin{bmatrix} w_{11}^6 & \cdots & w_{1n_h}^6 \\ \vdots & \ddots & \vdots \\ w_{n_h 1}^6 & \cdots & w_{n_h n_h}^6 \end{bmatrix}$ the weights between neurons

in the hidden layer, $w^7 = [w_{11}^7]$ back weight of neuron in the output layer,

$x^h = \begin{bmatrix} x_{11}^h \\ \vdots \\ x_{n_h 1}^h \end{bmatrix}$ the outputs of the hidden layer of neural

model,

The parameters of the vector of parameters of the controller neural Wc are also broken down into seven groups, formed respectively by:

$Wc^1 = \begin{bmatrix} Wc_{11}^1 & \cdots & Wc_{1n_r}^1 \\ \vdots & \ddots & \vdots \\ Wc_{n_h 1}^1 & \cdots & Wc_{n_h n_r}^1 \end{bmatrix}$ the weights between neurons in the input layer and neurons in the hidden layer,

$Wc^2 = \begin{bmatrix} Wc_{11}^2 \\ \vdots \\ Wc_{n_h 1}^2 \end{bmatrix}$ the biases of neurons in the hidden

layer,

$Wc^3 = [Wc_{11}^3 \cdots Wc_{1n_h}^3]$ the Weights between neurons in the hidden layer neurons and output layer,

$Wc^4 = [Wc_{11}^4]$ the bias of neuron in the output layer,

$Wc^5 = [Wc_{11}^5 \cdots Wc_{1n_r}^5]$ the weights between neurons of input layer neurons and output layer,

$Wc^6 = \begin{bmatrix} Wc_{11}^6 & \cdots & Wc_{1n_h}^6 \\ \vdots & \ddots & \vdots \\ Wc_{n_h 1}^6 & \cdots & Wc_{n_h n_h}^6 \end{bmatrix}$ the weights between neurons in the hidden layer,

$Wc^7 = [Wc_{11}^7]$ back weight of neuron in the output

layer,

$\chi^h = \begin{bmatrix} \chi_{11}^h \\ \vdots \\ \chi_{n_h 1}^h \end{bmatrix}$ the outputs of the hidden layer of neural controller.

The two vectors of model parameters and the neural controller are respectively the following forms:

$$w = [w_{11}^1, \cdots, w_{n_h n_r}^1, w_{11}^2, \cdots, w_{n_h 1}^2, w_{11}^3, \cdots, w_{1n_h}^3, w_{11}^4, w_{11}^5, \cdots, w_{1n_r}^5, w_{11}^6, \cdots, w_{n_h n_h}^6, w_{11}^7]^T$$

$$Wc = [Wc_{11}^1, \cdots, Wc_{n_h n_r}^1, Wc_{11}^2, \cdots, Wc_{n_h 1}^2, Wc_{11}^3, \cdots, Wc_{1n_h}^3, Wc_{11}^4, Wc_{11}^5, \cdots, Wc_{1n_r}^5, Wc_{11}^6, \cdots, Wc_{n_h n_h}^6, Wc_{11}^7]^T \quad (50)$$

These two vectors are of size $(n_m, 1)$, such as $n_m = n_h * n_r + 2 * n_h + 2 + n_r + (n_h)^2$.

In this section, the number of fitting parameters $n_r = 2 * n_m$.

The output of neural model of DC motor $\hat{y}(k)$ and the output of the controller neural $u(k-1)$ are respectively the following forms:

$$\hat{y}(k) = w_{11}^7 \hat{y}(k-1) + \sum_{j=1}^{n_h} w_{1j}^3 x_j^h(k) + \sum_{i=1}^{n_r} w_{1i}^5 \psi_i(k) + w_{11}^4 \quad (51)$$

$$u(k-1) = Wc_{11}^7 u(k-2) + \sum_{j=1}^{n_h} Wc_{1j}^3 \chi_j^h(k) + \sum_{i=1}^{n_r} Wc_{1i}^5 \varphi_i(k) + Wc_{11}^4 \quad (52)$$

with:

$$s_j(k) = \sum_{m=1}^{n_r} w_{jm}^1 \psi_m(k) + \sum_{m=1}^{n_h} w_{jm}^6 x_m^h(k-1) + w_{j1}^2 \quad (53)$$

$$x_j^h(k) = f_1(s_j(k)) \quad (54)$$

$$S_j(k) = \sum_{i=1}^{n_r} Wc_{ji}^1 \varphi_i(k) + \sum_{i=1}^{n_h} Wc_{ji}^6 \chi_i^h(k-1) + Wc_{j1}^2 \quad (55)$$

$$\chi_j^h(k) = f_1(S_j(k)) \quad (56)$$

Note by $g(\cdot)$ nonlinear function representing the neuronal controller and $h(\cdot)$ nonlinear function representing the neural model of the controlled system. We can then express respectively $u(k-1)$ and $\hat{y}(k)$ by the following expressions:

$$u(k-1) = g(y(k+n_k-1), \cdots, y(k+n_k-n_a), u(k-2), \cdots, u(k-n_b), r^*(k+n_k), \cdots, r^*(k+n_k-n_c), Wc) = g(\varphi(k), Wc) \quad (57)$$

$$\hat{y}(k) = h(y(k-1), \dots, y(k-n_a), u(k-n_k-1), \dots, u(k-n_k-n_b), \hat{y}(k-1), \dots, \hat{y}(k-n_c), w) = h(\psi(k), w) \quad (58)$$

The neural model is updated to minimize the error function $J_r(k)$ defined by:

$$J_r(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 \quad (59)$$

The adjustment of the online parameter vector w of this model by Levenberg-Marquardt, which belongs to the class of gradient methods of second order, is performed as follows [44-48]:

$$\hat{w}(k) = \hat{w}(k-1) + (F_r(k))^{-1} H(k)e(k) \quad (60)$$

such as:

$$F_r(k) = G(k) + \lambda(k-1)I \quad (61)$$

$$G(k) \cong H(k)H^T(k) \quad (62)$$

$$H(k) = \left(\frac{d\hat{y}(k)}{dw} \right)_{w=\hat{w}(k-1)} \quad (63)$$

$$\hat{y}(k) = h(\psi(k), \hat{w}(k-1)) \quad (64)$$

$$e(k) = y(k) - \hat{y}(k) \quad (65)$$

$$J_r(k) = \frac{1}{2}(e(k))^2 \quad (66)$$

$$\lambda(k) = \begin{cases} \mu\lambda(k-1), & \text{if } \lambda(k-1) > \delta_0 \\ \delta_0, & \text{if } \lambda(k-1) \leq \delta_0 \end{cases} \quad (67)$$

$$\mu = \begin{cases} 10, & \text{if } J_r(k) > J_r(k-1) \\ 0.1, & \text{if } J_r(k) \leq J_r(k-1) \end{cases} \quad (68)$$

The update of parameters depends on the desired performance indices (accuracy, stability). She is stopped if the number of iterations is reached or $J_r(k) \leq \delta_0$ or $\|H(k)\| \leq \delta_0$ or $\lambda(k) \geq \delta_1$.

With $\| \cdot \|$ is the Euclidean norm.

In our work, we fix $\delta_0 = 10^{-10}$ and $\delta_1 = 10^{10}$.

The neural model parameters $\hat{w}(k)$ and the learning rate $\lambda(k)$ must be initialized as follows:

The matrix elements $\hat{w}^1(0)$, $\hat{w}^3(0)$, $\hat{w}^5(0)$, $\hat{w}^6(0)$ and $\hat{w}^7(0)$ are in interval $[-0.5, 0.5]$ [48].

$\hat{w}^2(0) = [0 \dots 0]^T$ and $\hat{w}^4(0) = 0$.

$\lambda(0) = 0.1$ according to Bishop [49].

where:

I the identity matrix of size (n_m, n_m) .

$$\frac{d\hat{y}(k)}{dw} = \left[\begin{array}{c} \frac{\partial \hat{y}(k)}{\partial w_{11}^1}, \dots, \frac{\partial \hat{y}(k)}{\partial w_{n_b n_r}^1}, \frac{\partial \hat{y}(k)}{\partial w_{11}^2}, \dots, \frac{\partial \hat{y}(k)}{\partial w_{n_b 1}^2}, \\ \frac{\partial \hat{y}(k)}{\partial w_{11}^3}, \dots, \frac{\partial \hat{y}(k)}{\partial w_{1 n_h}^3}, \frac{\partial \hat{y}(k)}{\partial w_{11}^4}, \frac{\partial \hat{y}(k)}{\partial w_{11}^5}, \\ \dots, \frac{\partial \hat{y}(k)}{\partial w_{1 n_r}^5}, \frac{\partial \hat{y}(k)}{\partial w_{11}^6}, \dots, \frac{\partial \hat{y}(k)}{\partial w_{n_h n_h}^6}, \frac{\partial \hat{y}(k)}{\partial w_{11}^7} \end{array} \right]^T \quad (69)$$

$F_r(k)$ is a positive definite matrix and can therefore be invertible using the Cholesky method [50,51].

If all conditions are met a good identification, the model output $\hat{y}(k)$ is then a good approximation of the output system $y(k)$, which allows writing:

$$y(k) \cong \hat{y}(k) \quad (70)$$

$$r(k) \cong r^*(k) \quad (71)$$

The learning of neural controller is implemented so as to minimize the following quadratic criterion:

$$J_c(k) = \frac{1}{2}(r(k) - y(k))^2 \quad (72)$$

Using the Levenberg-Marquardt method, the vector of parameters of the controller W_c is adjusted online by the following equations:

$$\widehat{W}_c(k) = \widehat{W}_c(k-1) + (G_c(k) + \lambda(k-1)I)^{-1} D(k)e_1(k) \quad (73)$$

$$D(k) \cong \left(\frac{dy(k)}{dW_c} \right)_{W_c=\widehat{W}_c(k-1)} \cong \left(\frac{d\hat{y}(k)}{dW_c} \right)_{W_c=\widehat{W}_c(k-1)} \quad (74)$$

$$G_c(k) = D(k)D^T(k) \quad (75)$$

$$e_1(k) = r(k) - y(k) \cong r(k) - \hat{y}(k) \quad (76)$$

$$\hat{y}(k) = h(\psi(k), \hat{w}(k)) \quad (77)$$

$$\psi(k) = [\hat{y}(k-1), \dots, \hat{y}(k-n_a), u(k-n_k-1), \dots, u(k-n_k-n_b), \hat{y}(k-1), \dots, \hat{y}(k-n_c)]^T \quad (78)$$

$$u(k-1) = g(\varphi(k), \widehat{W}_c(k-1)) \quad (79)$$

$$\varphi(k) = [\hat{y}(k+n_k-1), \dots, \hat{y}(k+n_k-n_a), u(k-2), \dots, u(k-n_b), r(k+n_k), \dots, r(k+n_k-n_c)]^T \quad (80)$$

$$J_c(k) = \frac{1}{2}(e_1(k))^2 \quad (81)$$

$$\lambda(k) = \begin{cases} \mu\lambda(k-1), & \text{if } \lambda(k-1) > \delta_0 \\ \delta_0, & \text{if } \lambda(k-1) \leq \delta_0 \end{cases} \quad (82)$$

$$\mu = \begin{cases} 10, & \text{if } J_c(k) > J_c(k-1) \\ 0.1, & \text{if } J_c(k) \leq J_c(k-1) \end{cases} \quad (83)$$

The online adjustment of the parameters must be repeated until such time as the number of iterations is reached or $J_c(k) \leq \delta_0$ or $\|D(k)\| \leq \delta_0$ or $\lambda(k) \geq \delta_1$.

The neural controller parameters $\widehat{W}_c(k)$ and the positive definite matrix $\lambda(k)$ must be initialized as follows:

The elements of matrices

$\widehat{W}_c^1(0), \widehat{W}_c^3(0), \widehat{W}_c^5(0), \widehat{W}_c^6(0)$ and $\widehat{W}_c^7(0)$ are in the interval $[-0.5, 0.5]$.

$$\widehat{W}_c^2(0) = [0 \cdots 0]^T \quad \text{and} \quad \widehat{W}_c^4(0) = 0, \quad \lambda(0) = 0.1.$$

5. Stability Analysis

If the system to be controller is stable, and if its model is perfect, then the system controlled by the adaptive internal model control structure is stable if and only if the controller is stable [1,28,52]. The parametric adaptation algorithm and the Levenberg-Marquardt algorithm are algorithms stable. The neural model, neural controller and the conventional model are stable. We can therefore conclude that the control systems mentioned previously of the DC motor are stable.

6. Comparative Study

In our work, we will assume that $C_r(t)$, K_e and K_m are variable during the time (Figure 7).

The considered DC motor having the following characteristics:

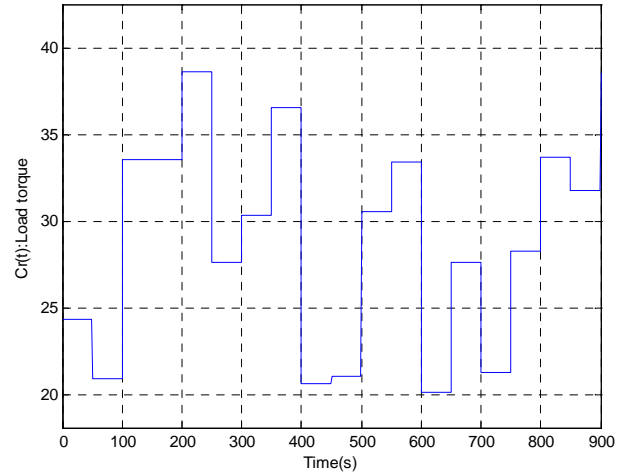
$$R = 12 \Omega, \quad J = 24 \times 10^{-4} \text{ kg} \cdot \text{m}^{-2}, \quad L = 0.18 \text{ H},$$

$$f = 6 \times 10^{-3} \text{ N} \cdot \text{s} \cdot \text{m} \cdot \text{rad}^{-1}.$$

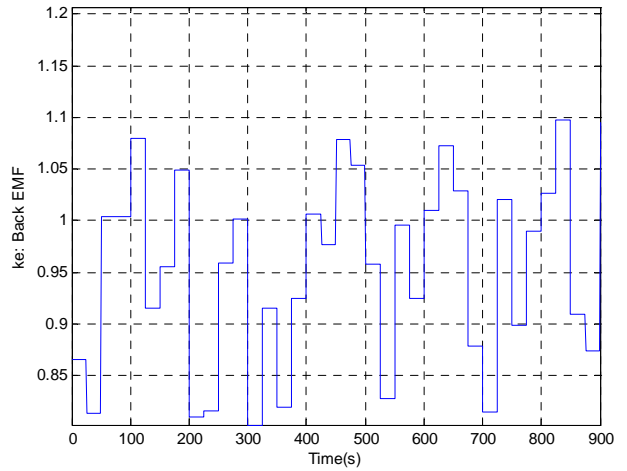
The choice of the sampling period can have a dramatic effect on the results of the identification and control. If the sampling period is chosen large enough, this causes a bad description of the dynamics of the system and can lead to failure. By cons, if the sampling period is low, then the values of some parameters may become very small, so it is difficult to estimate with good accuracy. According to [53], the sampling period T_s can be determined from the step response of the process. In this sense, Isermann proposed choose T_s satisfying:

$$\frac{1}{15} T_{95} \leq T_s \leq \frac{1}{4} T_{95} \quad (84)$$

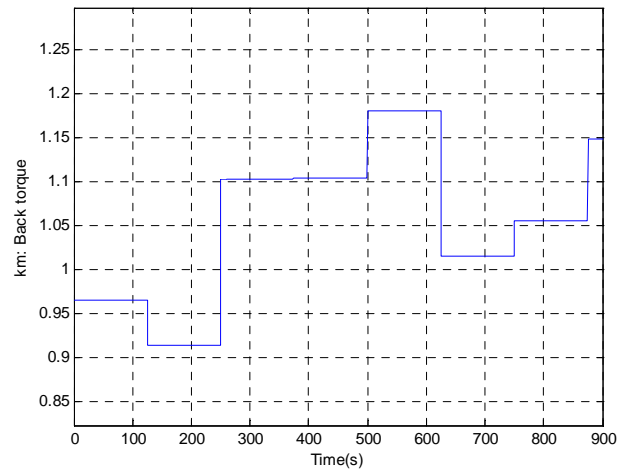
where T_{95} represents the response time to 95% of the system.



(a)



(b)



(c)

Figure 7. DC motor parameters: (a) Load torque; (b) Back EMF; (c) Back torque.

The **Figure 8** shows that:

$$0.5 \text{ s} \leq T_{95} \leq 1.6 \text{ s} \quad (85)$$

So:

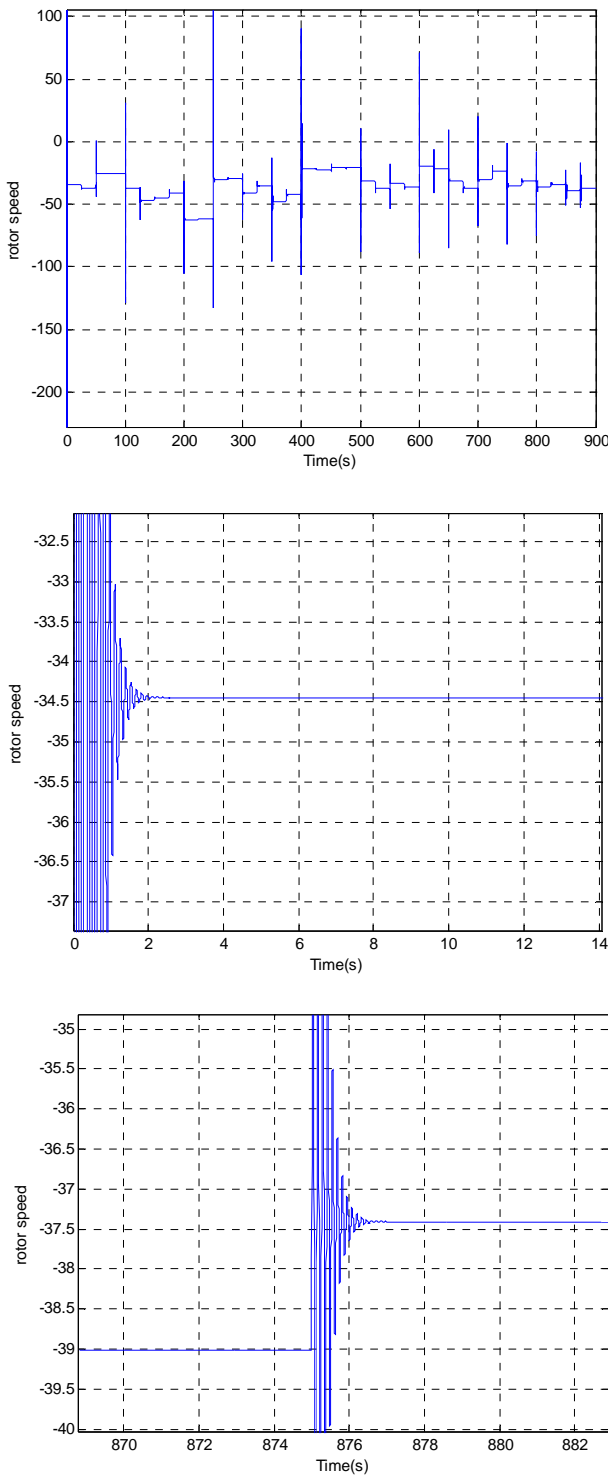


Figure 8. Step response of DC motor: determined of the sampling period T_s .

$$0.03 \text{ s} \leq T_s \leq 0.4 \text{ s} \quad (86)$$

We fix $T_s = 0.4 \text{ s}$

The Figure 9 shows the sequence input and output used to calculate the model parameters. These sequences

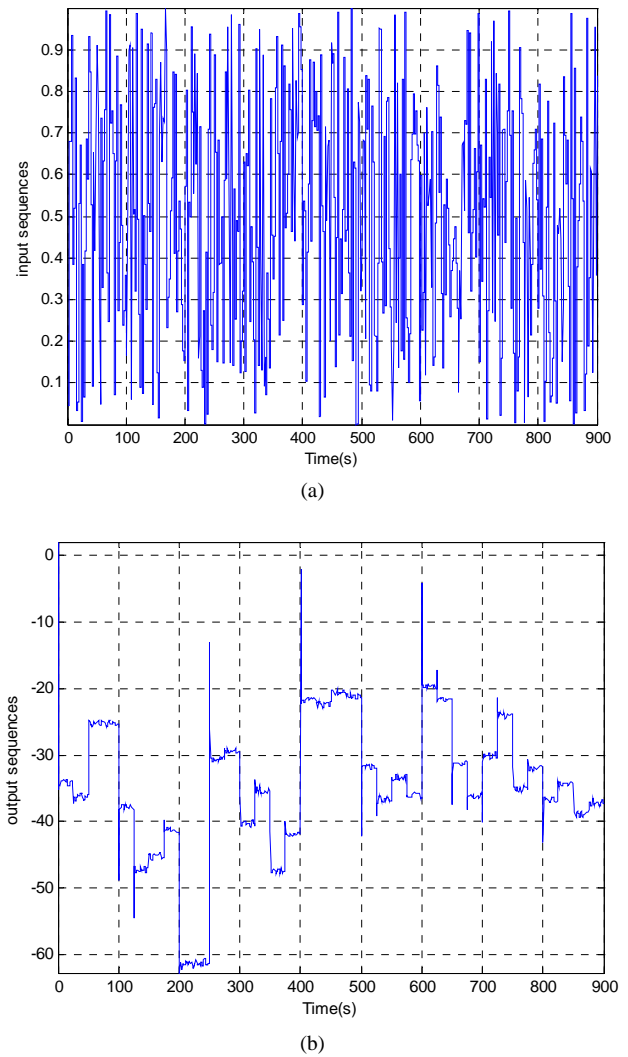


Figure 9. Sequences Data: (a) Input sequence; (b) Output sequence.

represent the system response to a random signal of zero mean and variance 1.

From Table 2, the best model to correctly approach the system dynamics model is $M_{2,2,2,0}$.

We fix $n_a = 2, n_b = 2, n_c = 2$ and $n_k = 0$.

To validate the model chosen ($M_{2,2,2,0}$), we plot the autocorrelation functions of residuals (Figure 10) and cross-correlation between input and residuals (Figure 11). We note that these two functions are almost within the confidence intervals, thus validating the use of the resulting network as a model system studied.

The curves representing the evolution of the model parameters are plotted in Figure 12.

The Figures 13 and 14 represent the evolutions of zeros and poles of the model. The modules of the poles and zeros of the models are less than 1, thus validating the stability of this control system.

To build the neural model of the engine, we used the

Table 2. Nash criterion values of the different candidate models.

Different candidate models	$Q(\%)$
$M_{1,2,2,0}$	75.53
$M_{1,2,2,1}$	83.91
$M_{2,2,2,0}$	91.87
$M_{2,2,2,1}$	89.12
$M_{2,3,3,0}$	88.86
$M_{2,3,3,1}$	89.32

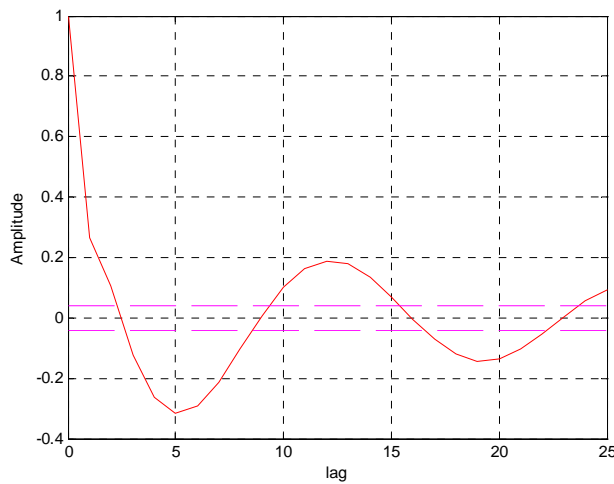


Figure 10. Auto-correlation function of residuals.

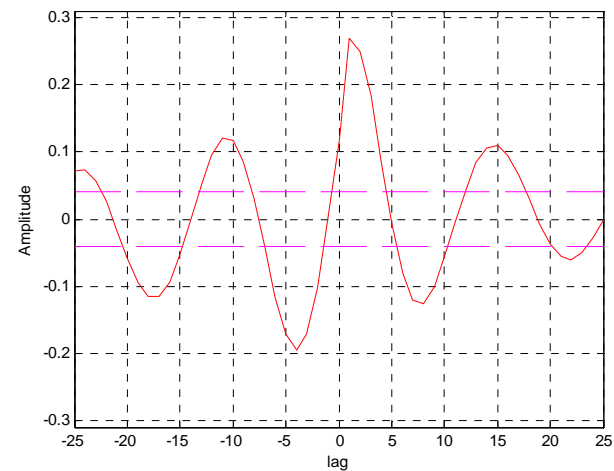


Figure 11. Intercorrelation function between input and residuals.

data sequences that are represented in **Figure 9**. The evolution of the Nash criterion of different candidate models of the DC motor (**Table 3**) leads to the conclusion that 5 neurons in the hidden layer are necessary and sufficient for a neuronal model with satisfactory accuracy. It therefore sets $n_h = 5$.

The autocorrelation functions of residuals and cross-

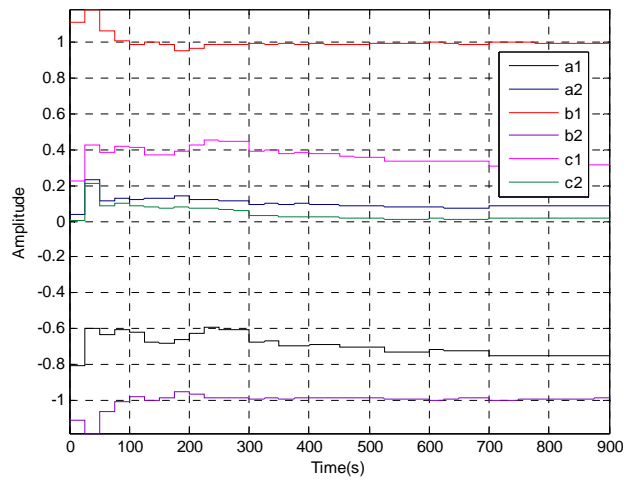


Figure 12. Evolution of model parameters.

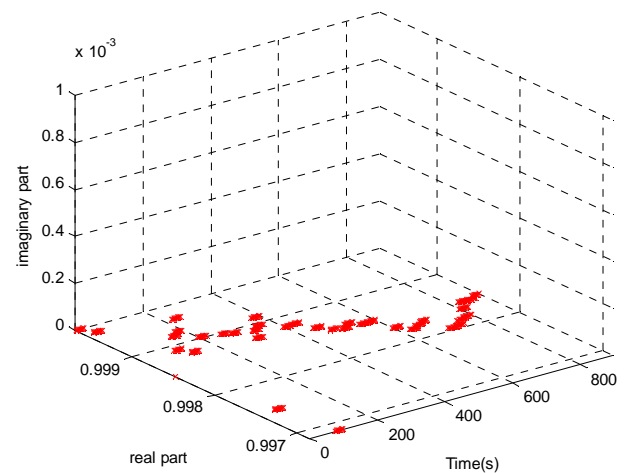


Figure 13. Evolution of the zeros of the model.

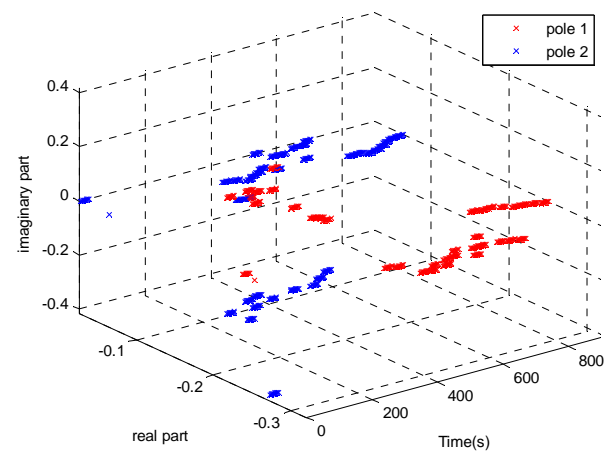


Figure 14. Evolution of the poles of the model.

correlation between input and residuals (**Figure 15**) are within the confidence intervals, thus validating the use of the network chosen as the model system studied.

Table 3. Evolution of the Nash criterion of different neural models.

Number of neurons in the hidden layer	$Q(\%)$
1	75.15
2	78.27
3	88.34
4	85.26
5	99.62
6	89.21
7	79.57
8	77.23
9	75.45

The performance of the adaptive conventional internal model control law applied to the DC motor is illustrated in **Figures 16, 17 and 18**. On the other side, the **Figures 19, 20 and 21** show the results of adaptive neural internal model control of the DC motor. By comparing the results of two control systems of the DC motor mentioned previously, we see clearly:

- $n_\gamma > n_\beta$ therefore the adjustment of adaptive neural internal model control system parameters requires more time than the conventional adaptive internal model control system.

- A sudden change of system parameters implies a sudden change in the amplitude of the model output and the controlled system.

- The system output and the neural model output are almost identical, which confirms that the neural network is synthesized with more high precision than the output

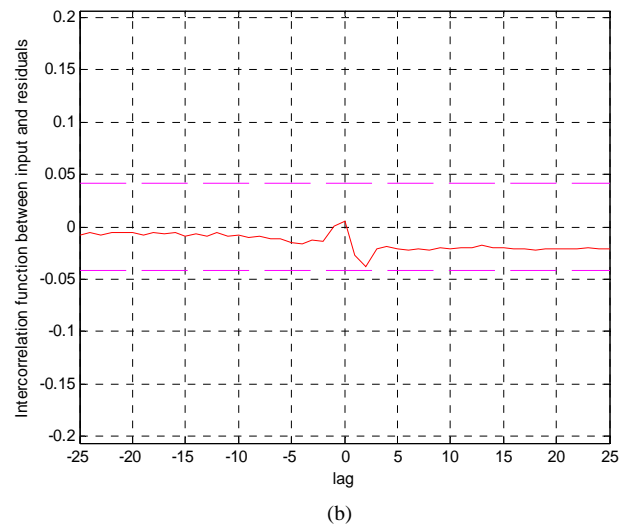
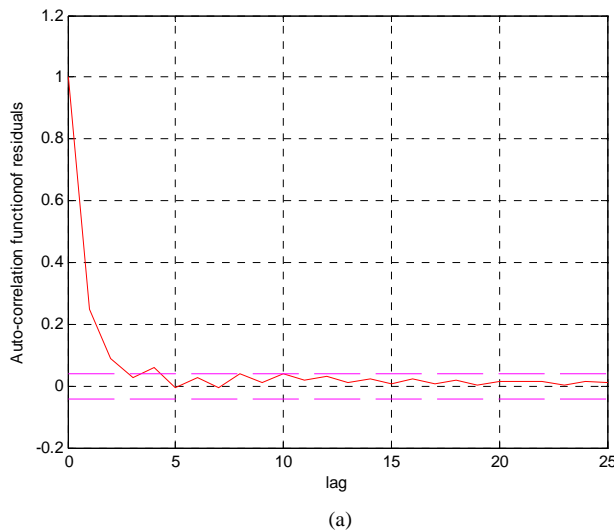
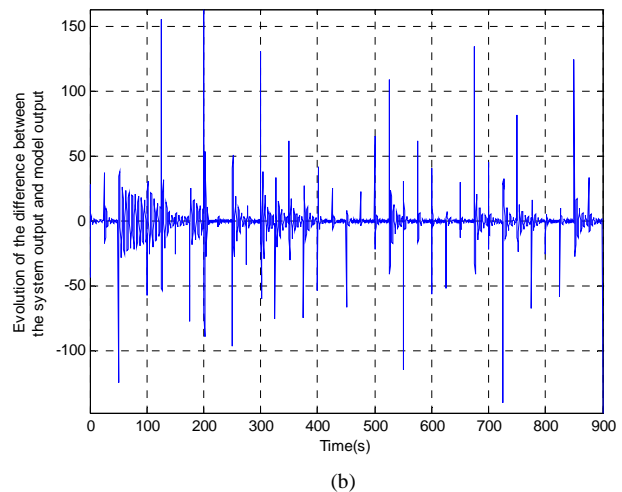
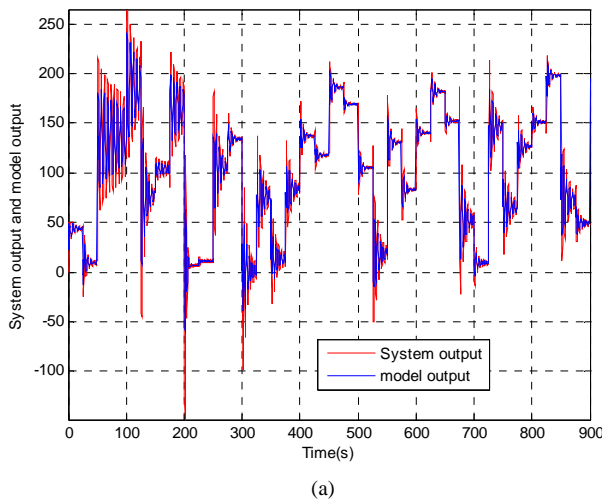


Figure 15. Tests of model validation: (a) Autocorrelation function of residuals; (b) Intercorrelation function between input and output residuals.



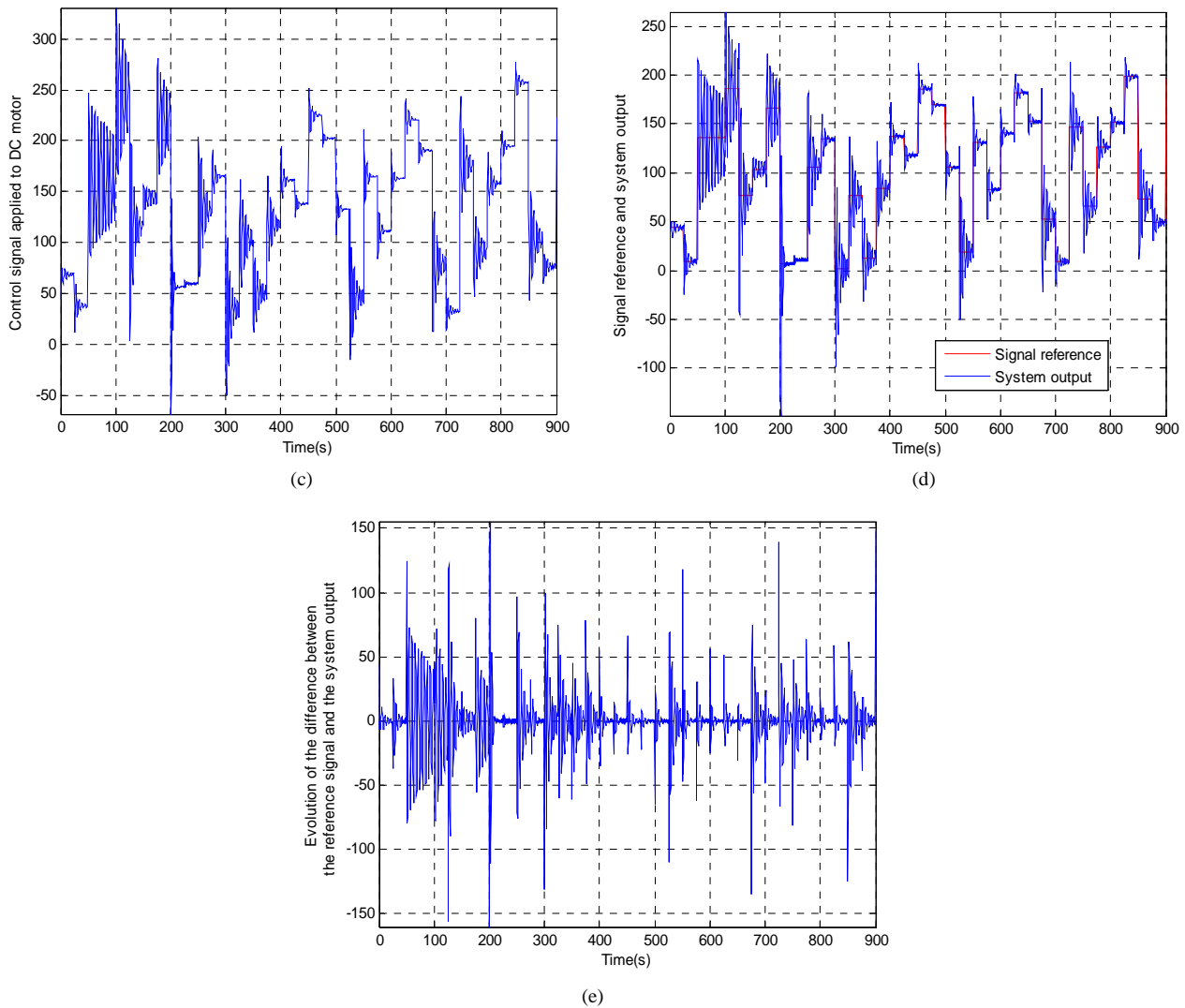
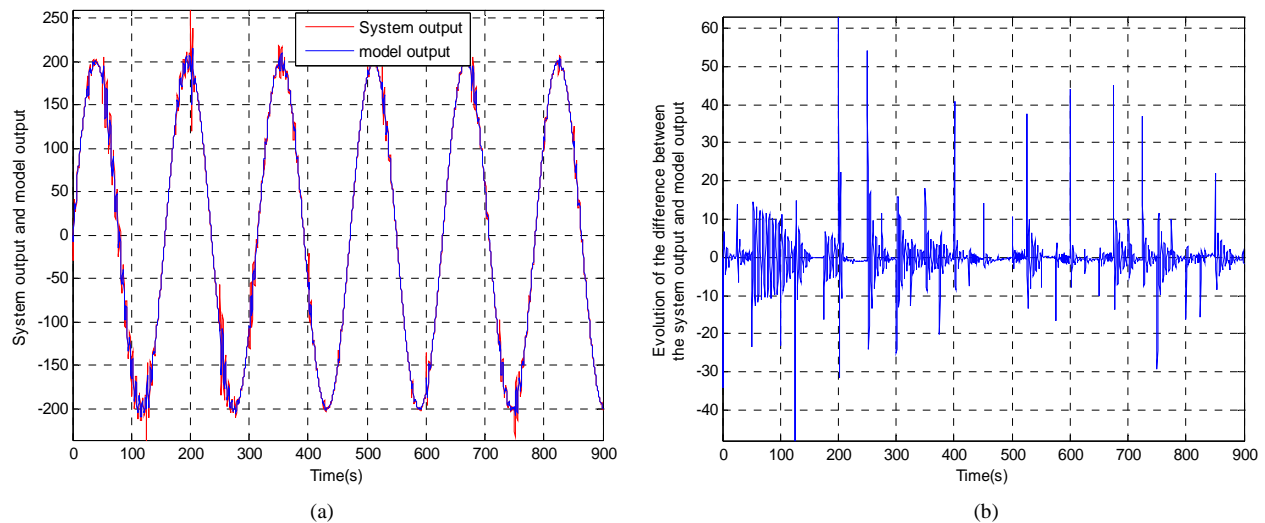


Figure 16. Results of the adaptive conventional internal model control of DC motor in the case of a reference signal amplitude random uniform distribution: (a) Output system and output model; (b) Evolution of the criterion ε ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ε_1 .



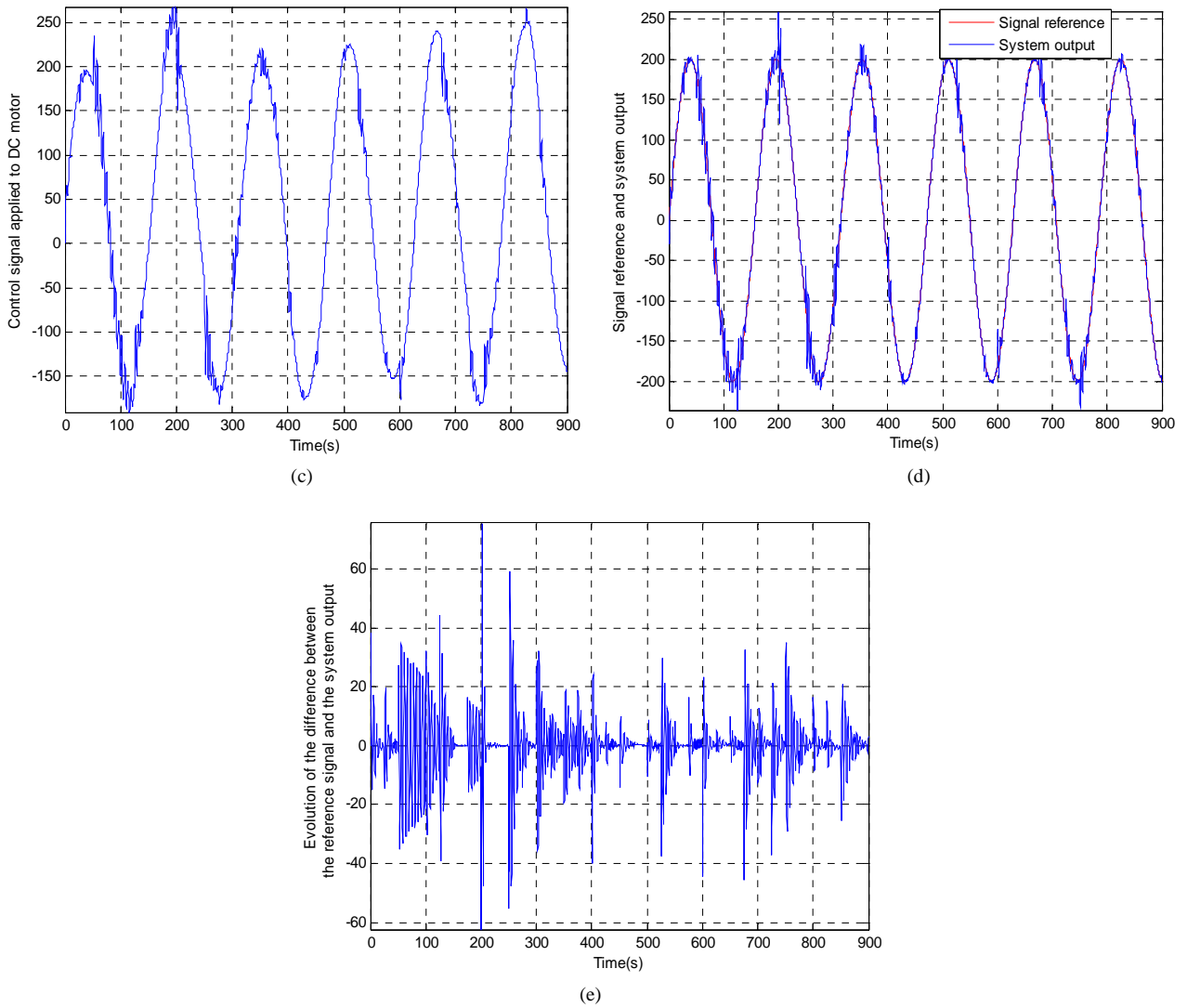
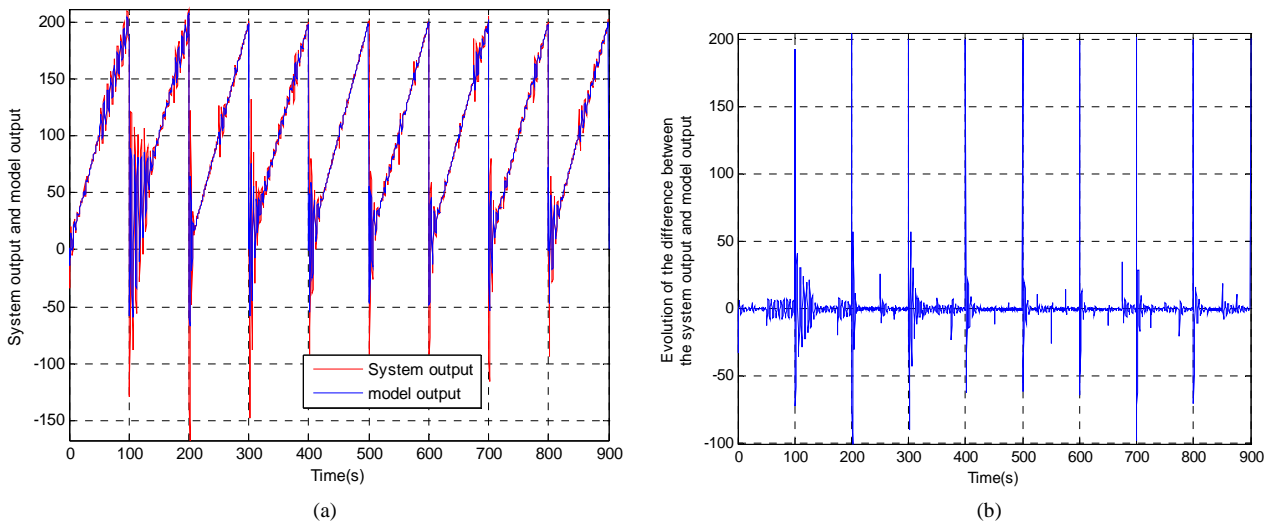
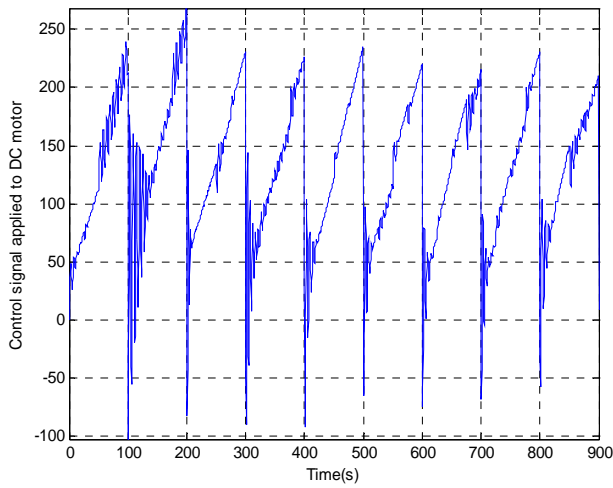
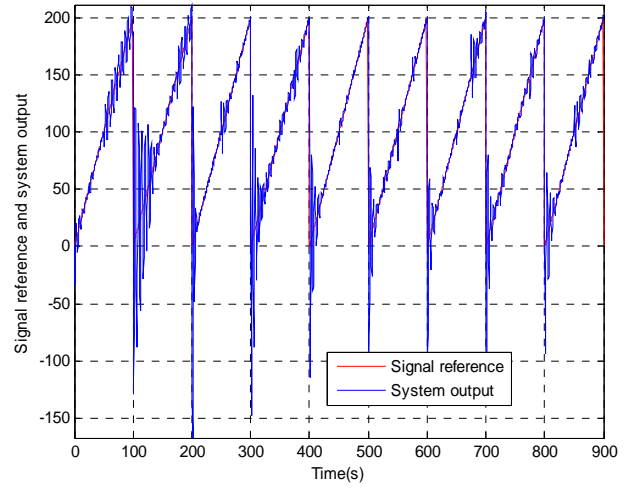


Figure 17. Results of adaptive conventional internal model control of DC motor in the case of a sinusoidal reference signal: (a) Output system and output model; (b) Evolution of the criterion ε ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ε_1 .

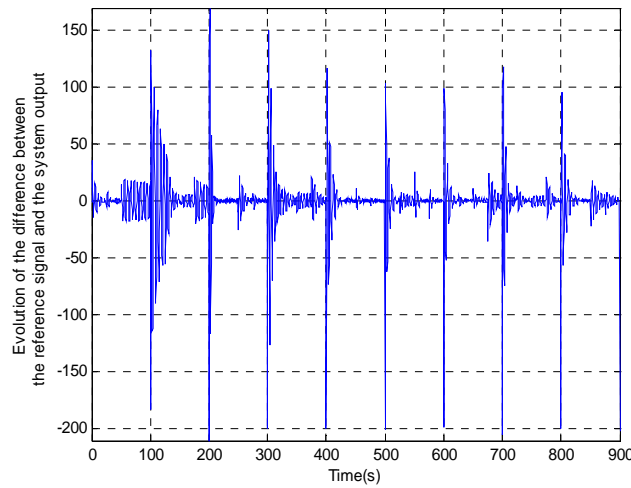




(c)

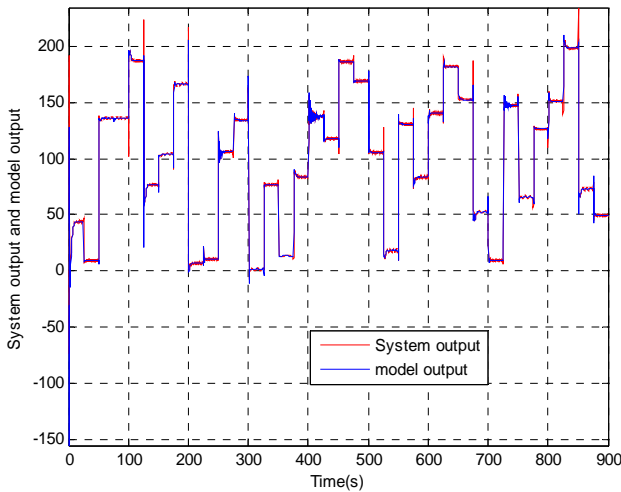


(d)

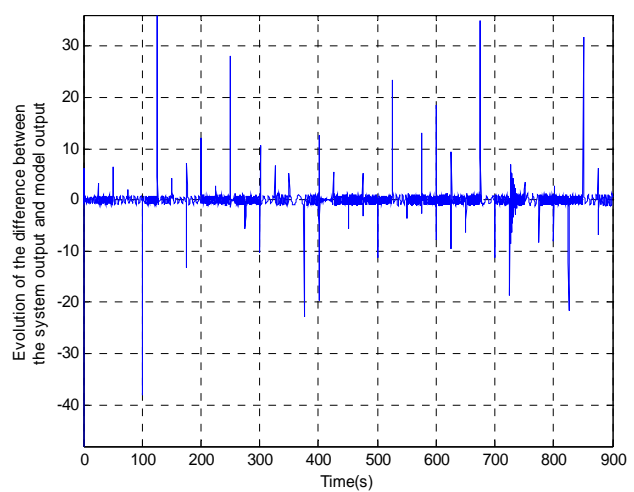


(e)

Figure 18. Results of the adaptive conventional internal model control of DC motor in the case of a triangular: (a) Output system and output model; (b) Evolution of the criterion ε ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ε_1 .



(a)



(b)

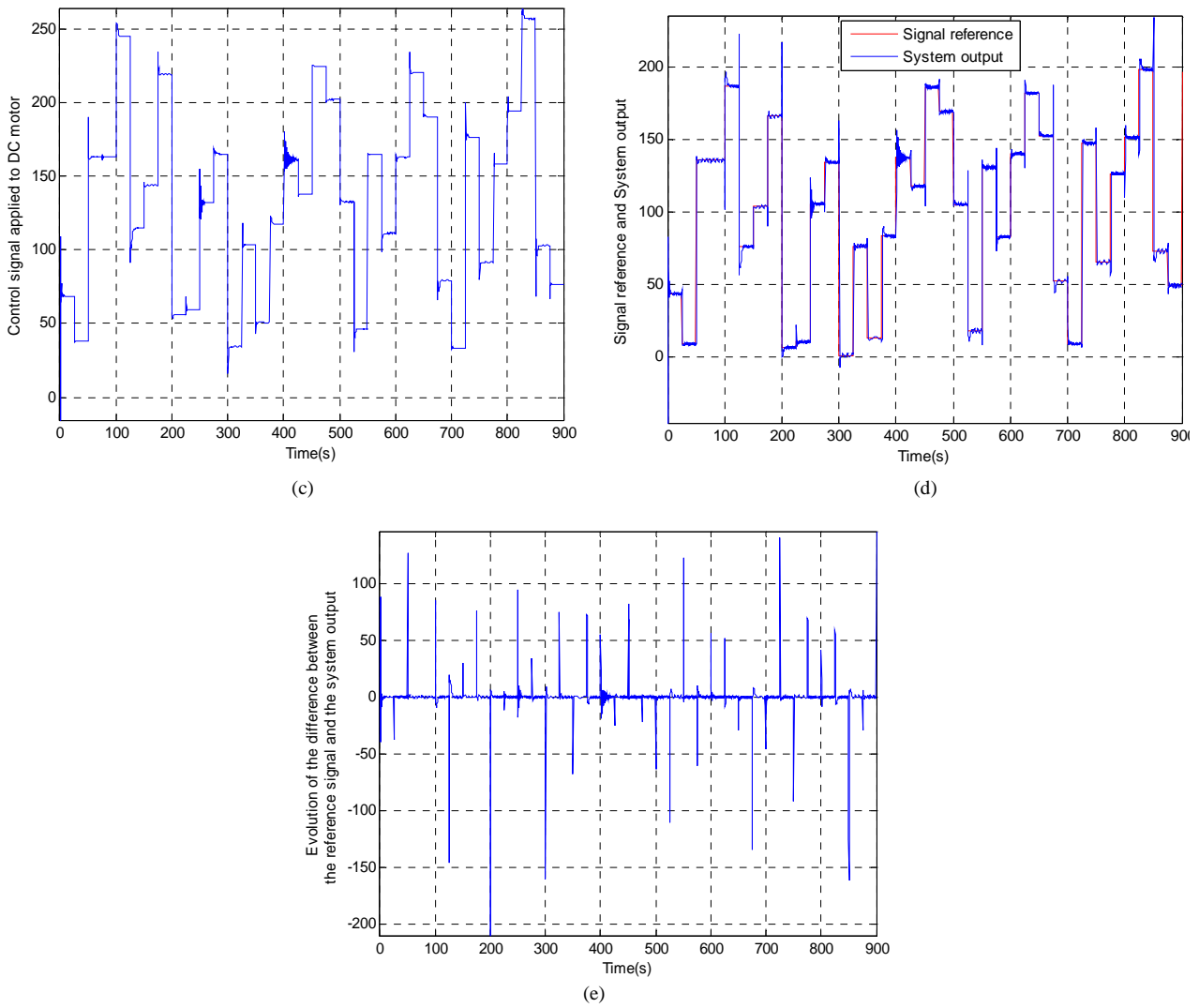
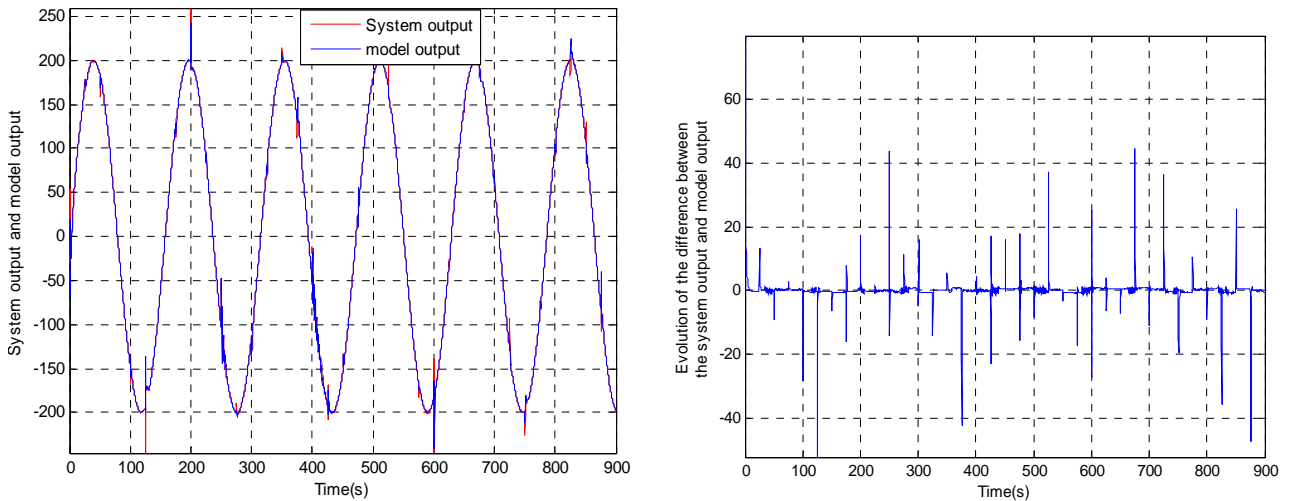


Figure 19. Performance of adaptive neural internal model control of the DC motor in the case of a reference signal amplitude random uniform distribution: (a) Output system and output model; (b) Evolution of the criterion ε ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ε_1 .



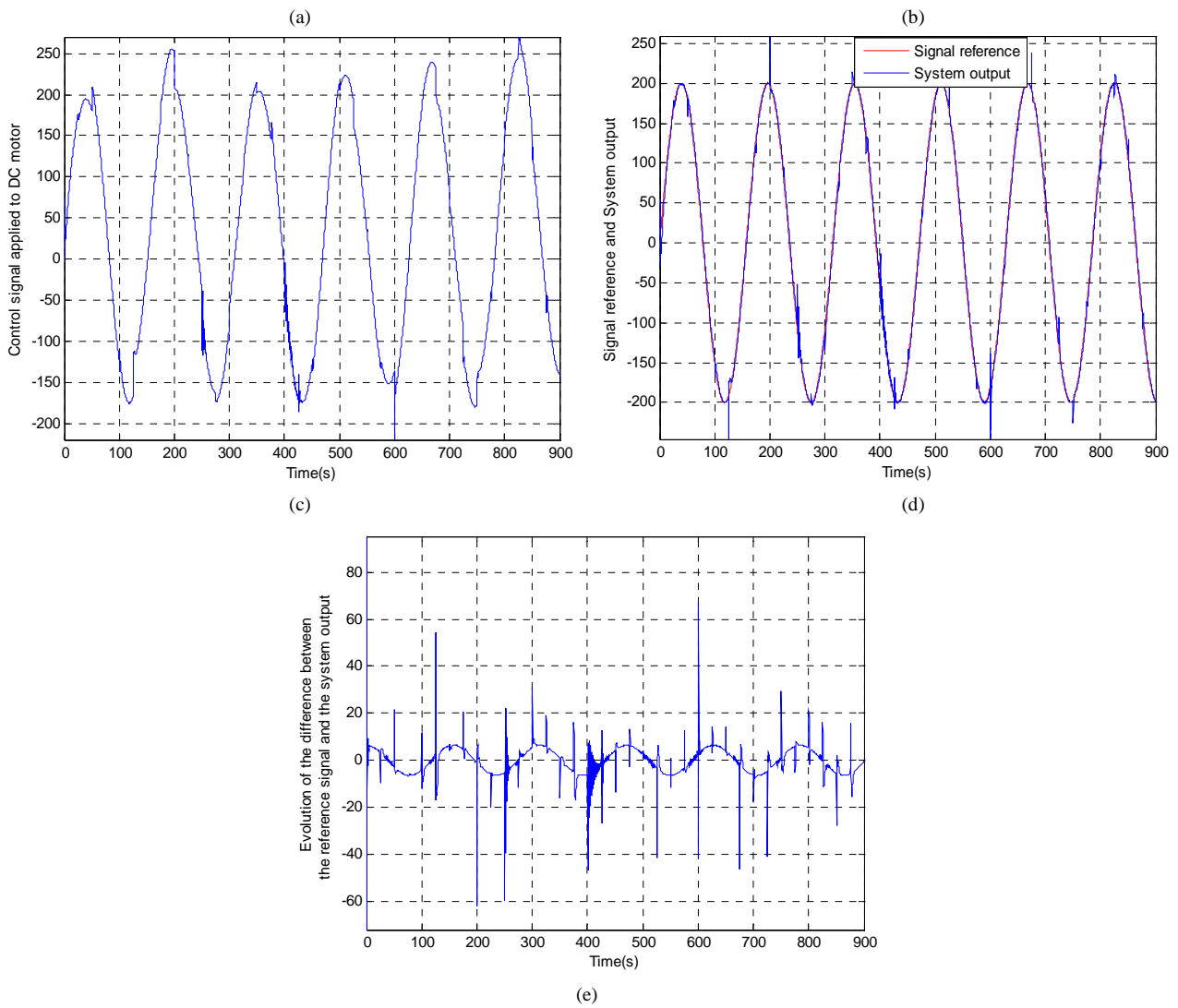
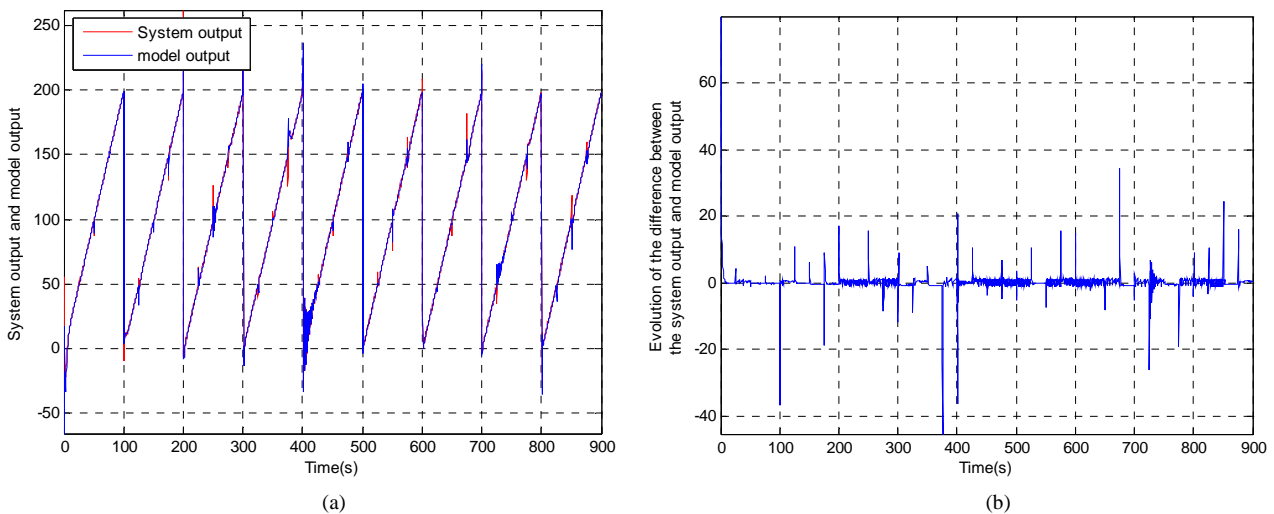


Figure 20. Performance of adaptive neural internal model control of the DC motor in the case of a sinusoidal reference signal: (a) Output system and output model; (b) Evolution of the criterion ϵ_1 ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ϵ_1 .



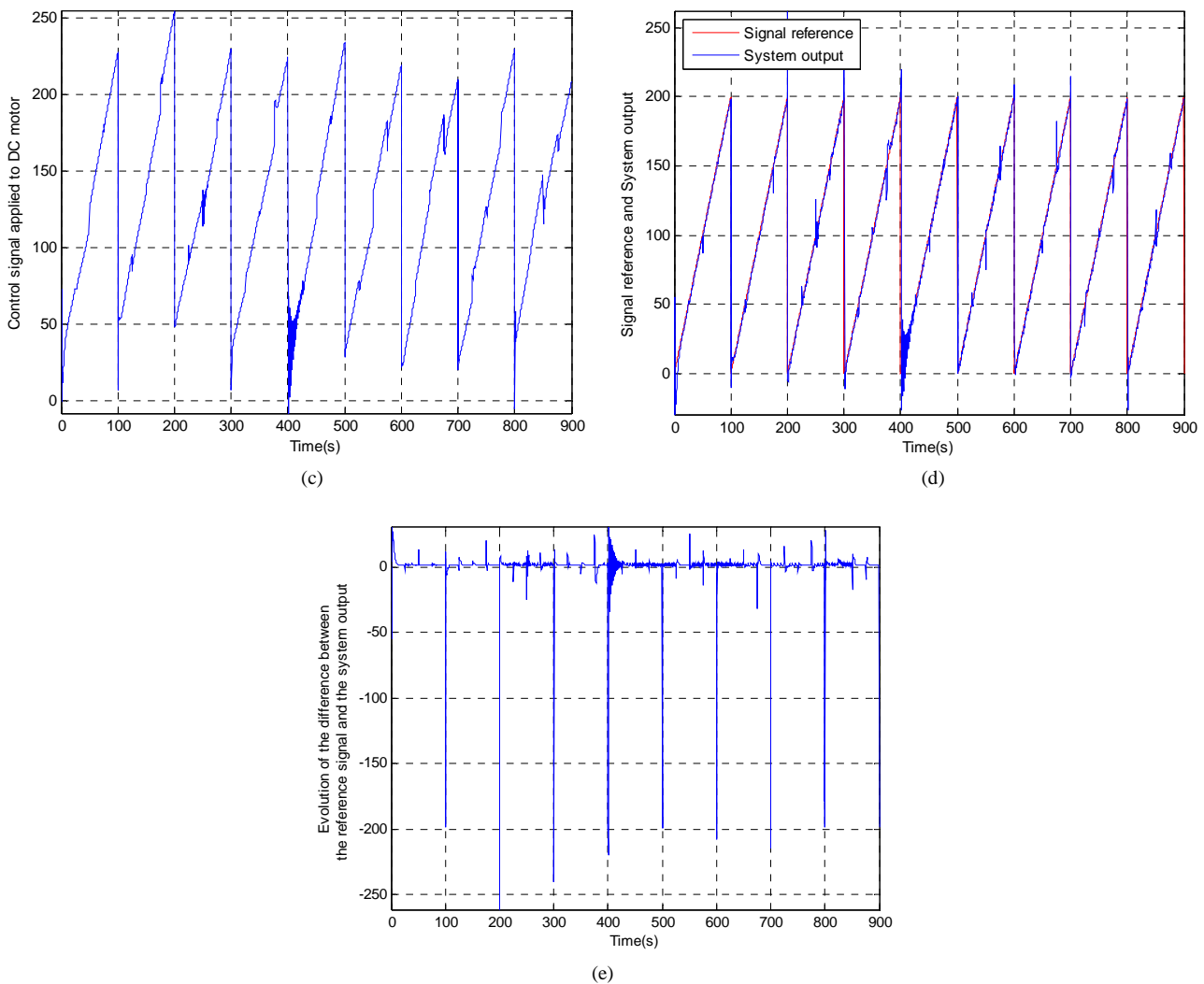


Figure 21. Performance of adaptive neural internal model control of the DC motor in the case of a triangular reference signal reference signal: (a) Output system and output model; (b) Evolution of the criterion ε ; (c) Control signal applied to DC motor; (d) Response system; (e) Evolution of the criterion ε_1 .

model obtained by the conventional identification method.

-The control signal shows fluctuations in the case of adaptive conventional internal model control. By cons in the case of adaptive neural internal model control, fluctuations are greatly diminished.

-The output of the adaptive neural internal model control system adequately follows the reference signal compared to the output of conventional adaptive internal model control system.

-The evolutions of the criterion ε_1 clearly show that the adaptive conventional internal model control system cannot fully offset the effect of disturbance compared to the adaptive neural internal model control system.

7. Conclusion

In this paper, two adaptive internal model control struc-

tures are used so that the speed of DC motor follows the given trajectories.

The comparative study showed the effectiveness of the adaptive neural internal model control system compared to conventional adaptive internal model control system. Indeed, it was found that the adaptive control system by neural internal model meets the desired objectives: the powerful regulation and robustness of the speed, the disturbance rejection and system stability.

REFERENCES

- [1] A. Datta and J. Ochoa, "Adaptive Internal Model Control: Design and Stability Analysis," *Automatica*, Vol. 32, No. 2, 1996, pp. 261-266.
[doi:10.1016/0005-1098\(96\)85557-9](https://doi.org/10.1016/0005-1098(96)85557-9)
- [2] C.-H. Choi and H.-C. Kim, "A robust internal model adap-

- tive controller,” *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, 10-12 December 1997, pp. 2758-2761.
- [3] L. Xing and A. Datta, “Decentralized Adaptive Internal Model Control for Multi-Input Multi-Output Systems,” *American Control Conference*, San Diego, 2-4 June 1999, pp. 39-46.
- [4] Q. P. Hu and G. P. Rangaiah, “Adaptive Internal Model Control of Nonlinear Processes,” *Chemical Engineering Science*, Vol. 54, No. 9, 1999, pp. 1205-1220. doi:10.1016/S0009-2509(98)00543-0
- [5] G. J. Silva and A. Datta, “Adaptive Internal Model Control: The Discrete-Time Case,” *International Journal of Adaptive Control and Signal Processing*, Vol. 15, No.1, 2001, pp. 15-36. doi:10.1002/1099-1115(200102)15:1<15::AID-ACS625>3.0.CO;2-I
- [6] M. Shafiq and S. H. Riyaz, “Internal Model Control Structure Using Adaptive Inverse Control Strategy,” *4th International Conference on Control and Automation*, Montreal, 12-12 June 2003, pp. 148-152.
- [7] D. Rupp and L. Guzzella, “Adaptive Internal Model Control with Application to Fueling Control,” *Control Engineering Practice*, Vol. 18, No. 8, 2010, pp. 873-881. doi:10.1016/j.conengprac.2010.03.011
- [8] F. X. Xu, G. Xia, M. Zeng, B. K. Sun and X. Z. Zhao, “Improvement of Precision for Pendulous Integrating Gyro Accelerometer via Adaptive Internal Model Control,” *Journal of Systems Engineering and Electronics*, Vol. 18, No. 4, 2007, pp. 841-845. doi:10.1016/S1004-4132(08)60029-8
- [9] C. Cheng and M.-S. Chiu, “Adaptive IMC Controller Design for Nonlinear Process Control,” *Chemical Engineering Research and Design*, Vol. 85, No. 2, 2007, pp. 234-244. doi:10.1205/cherd06071
- [10] Y. Kansha, J. Li and M.-S. Chiu, “Adaptive IMC Controller Design Using Linear Multiple Models,” *Journal of the Taiwan Institute of Chemical Engineers*, Vol. 41, No. 4, 2010, pp. 446-452. doi:10.1016/j.jtice.2010.04.002
- [11] N. R. Lakshmi Narayanan, P. R. Krishnaswamy and G. P. Rangaiah, “An Adaptive Internal Model Control Strategy for PH Neutralization,” *Chemical Engineering Science*, Vol. 52, No. 18, 1997, pp. 3067-3074. doi:10.1016/S0009-2509(97)00130-9
- [12] A. Mehmood, S. Laghrouche and M. El Bagdouri, “Modeling Identification and Simulation of Pneumatic Actuator for VGT System,” *Sensors and Actuators A: Physical*, Vol. 165, No. 2, 2011, pp. 367-378. doi:10.1016/j.sna.2010.11.006
- [13] R.-F. Fung, Y.-S. Kung and G.-C. Wu, “Dynamic Analysis and System Identification of an LCD Glass-Handling Robot Driven by a PMSM,” *Applied Mathematical Modelling*, Vol. 34, No. 5, 2010, pp. 1360-1381. doi:10.1016/j.apm.2009.08.020
- [14] J. R. Rodriguez Vasquez, R. Rivas Perez, J. Sotomayor Moriano and J. R. Peran Gonzalez, “System Identification of Steam Pressure in a Fire-Tube Boiler,” *Computers and Chemical Engineering*, Vol. 32, No. 12, 2008, pp. 2839-2848. doi:10.1016/j.compchemeng.2008.01.010
- [15] M. Modabberifar, Y. Hojjat, A. Abdullah and M. Dadkhah, “Analysis of the Dynamic Characteristics of a Three Phase Panel Type Electrostatic Actuator Using Pem System Identification Method,” *Sensors and Actuators A: Physical*, Vol. 151, No. 2, 2009, pp. 173-178. doi:10.1016/j.sna.2009.01.019
- [16] X. Z. Wei, “Least-Squares Identification of a Class of Multivariable Systems with Correlated Disturbances,” *Journal of the Franklin Institute*, Vol. 336, No. 8, 1999, pp. 1309-1324. doi:10.1016/S0016-0032(99)00038-1
- [17] Q. Z. Zou and S. Devasia, “Precision Preview-Based Stable-Inversion for Nonlinear Nonminimum-Phase Systems: The VTOL Example,” *Automatica*, Vol. 43, No. 1, 2007, pp. 117-127. doi:10.1016/j.automatica.2006.08.007
- [18] L. R. Hunt and G. Meyer, “Stable Inversion for Nonlinear Systems,” *Automatica*, Vol. 33, No. 8, 1997, pp. 1549-1554. doi:10.1016/S0005-1098(97)00064-2
- [19] N. Kazantzis, “A New Approach to The Zero-Dynamics Assignment Problem for Nonlinear Discrete-Time Systems Using Functional Equations,” *Systems & Control Letters*, Vol. 51, No. 3-4, 2004, pp. 311-324. doi:10.1016/j.sysconle.2003.03.001
- [20] A. Piazzzi and A. Visioli, “A Noncausal Approach for PID Control,” *Journal of Process Control*, Vol. 16, No. 8, 2006, pp. 831-843. doi:10.1016/j.jprocont.2006.03.001
- [21] M. T. Nihtilä, “Adaptive Control of a Continuous-Time System with Time-Varying Input Delay,” *Systems & Control Letters*, Vol. 12, No. 4, 1989, pp. 357-364. doi:10.1016/0167-6911(89)90046-7
- [22] C. Kambhampati, R. J. Craddock, M. Tham and K. Warwick, “Inverse Model Control Using Recurrent Networks,” *Mathematics and Computers in Simulation*, Vol. 51, No. 3-4, 2000, pp. 181-199. doi:10.1016/S0378-4754(99)00116-0
- [23] S. Suresh and N. Kannan, “Direct Adaptive Neural Flight Control System for an Unstable Unmanned Aircraft,” *Applied Soft Computing*, Vol. 8, No. 2, 2008, pp. 937-948. doi:10.1016/j.asoc.2007.07.009
- [24] M. Morari and E. Zafirou, “Robust Process Control,” Prentice Hall, Englewood Cliffs, 1989.
- [25] P. Larminat, “Commande des Systèmes Linéaires,” Hermès, Paris, 1993.
- [26] G. Duc, “Robustesse des Systèmes Linéaires Multi-variables,” Cours de l’Ecole Supérieure d’Electricité, France, 1993.
- [27] Scorletti G., “Approche Unifiée de l’Analyse et de la Commande des Systèmes par Formulation LMI,” Thèse de Doctorat, Université Paris XI, Orsay, 1997.
- [28] A. J. Fossard, “La Régulation Industrielle: Régulateur PID, Prédicatifs et Flous,” Hermès, Paris, 1994.
- [29] C. Keyes, “Guide de Référence sur l’Efficacité Energétique de Moteurs,” CEA Technologies Inc., Fyshwick, 2007.
- [30] S. Weerasooriya and M. A. El-Sharkawi, “Identification and Control of a DC Motor Using Back-Propagation Neural Networks,” *IEEE Transactions on Energy Conversion*, Vol. 6, No. 4, 1991, pp. 663-669. doi:10.1109/60.103639

- [31] N. Amuthan and S. N. Singh, "Direct Model Reference Adaptive Internal Model Controller for DFIG Wind Farms," *International Journal of Recent Trends in Engineering*, Vol. 1, No. 1, 2009, pp. 7-11.
- [32] P. M. Frank, "Entwurf von Regelkreisen mit Vorgeschriebenem Verhalten," Paperback, G. Braun Verlag, Karlsruhe, 1974.
- [33] R. Abdennour, P. Borne, M. Ksouri and F. M'sahli, "Méthodes et Pratiques de l'Ingénieur: Identification et Commande Numérique des Procédés Industriels," Editions Technip, Paris, 2001.
- [34] A. V. Silaev, "Parametric Adaptation Algorithm for The Basic Law Used to Control Orientation of a Large Space Structure on the Base of Flexible Modes Envelope Estimation," *Automation and Remote Control*, Vol. 71, No. 5, 2010, pp. 911-919. [doi:10.1134/S000511791005019X](https://doi.org/10.1134/S000511791005019X)
- [35] A. Diala, "Modélisation Dynamique et Supervision Des fours de Réchauffage Continus de la Sidérurgie," Thèse de Doctorat, Ecole des Mines de Paris, 2006.
- [36] K. J. Hunt and D. Sbarbaro, "Neural Networks for Nonlinear Internal Model Control," *IEE Proceedings D of Control Theory and Applications*, Vol. 138, No. 5, 1991, pp. 431-438.
- [37] I. Rivals and L. Personnaz, "Nonlinear Internal Model Control Using Neural Networks: Application to Processes With Delay and Design Issues," *IEEE Transactions on Neural Networks*, Vol. 11, No. 1, 2000, pp. 80-90. [doi:10.1109/72.822512](https://doi.org/10.1109/72.822512)
- [38] S. Bel Hadj Ali, A. El Abed-Abdelkrim and M. Benrejeb, "An Internal Model Control Strategy Using Artificial Neural Networks for a Class of Nonlinear Systems," 2002 *IEEE International Conference on Systems, Man and Cybernetics*, 6-9 October 2002, p. 4.
- [39] K. Nouri, R. Dhaouadi and N. B. Braiek, "Adaptive Control of a Nonlinear DC Motor Drive Using Recurrent Neural Networks," *Applied Soft Computing*, Vol. 8, No. 1, 2008, pp. 371-382. [doi:10.1016/j.asoc.2007.03.002](https://doi.org/10.1016/j.asoc.2007.03.002)
- [40] H. Q. Zhao and J. S. Zhang, "A Novel Nonlinear Adaptive Filter Using a Pipelined Second-Order Volterra Recurrent Neural Network," *Neural Networks*, Vol. 22, No. 10, 2009, pp. 1471-1483. [doi:10.1016/j.neunet.2009.05.010](https://doi.org/10.1016/j.neunet.2009.05.010)
- [41] M. Hermans and B. Schrauwen, "Memory in Linear Recurrent Neural Networks in Continuous Time," *Neural Networks*, Vol. 23, No. 3, 2010, pp. 341-355. [doi:10.1016/j.neunet.2009.08.008](https://doi.org/10.1016/j.neunet.2009.08.008)
- [42] H. Q. Zhao, X. P. Zeng, J. S. Zhang, Y. G. Liu, X. M. Wang and T. R. Li, "A Novel Joint-Processing Adaptive Nonlinear Equalizer Using a Modular Recurrent Neural Network for Chaotic Communication Systems," *Neural Networks*, Vol. 24, No. 1, 2011, pp. 12-18. [doi:10.1016/j.neunet.2010.09.009](https://doi.org/10.1016/j.neunet.2010.09.009)
- [43] M. Syed Ali and M. Marudai, "Stochastic Stability of Discrete-Time Uncertain Recurrent Neural Networks with Markovian Jumping and Time-Varying Delays," *Mathematical and Computer Modelling*, Vol. 54, No. 9-10, 2011, pp. 1979-1988. [doi:10.1016/j.mcm.2011.05.004](https://doi.org/10.1016/j.mcm.2011.05.004)
- [44] F. M. Dias, A. Antunes, J. Vieira and A. Mota, "A Sliding Window Solution for the on-Line Implementation of the Levenberg—Marquardt Algorithm," *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 1, 2006, pp. 1-7. [doi:10.1016/j.engappai.2005.03.005](https://doi.org/10.1016/j.engappai.2005.03.005)
- [45] B. G. Kermani, S. S. Schiffman and H. T. Nagle, "Performance of the Levenberg—Marquardt Neural Network Training Method in Electronic Nose Applications," *Sensors and Actuators B: Chemical*, Vol. 110, No. 1, 2005, pp. 13-22. [doi:10.1016/j.snb.2005.01.008](https://doi.org/10.1016/j.snb.2005.01.008)
- [46] V. Singh, I. Gupta and H. O. Gupta, "ANN-Based Estimator for Distillation Using Levenberg—Marquardt Approach," *Engineering Applications of Artificial Intelligence*, Vol. 20, No. 2, 2007, pp. 249-259. [doi:10.1016/j.engappai.2006.06.017](https://doi.org/10.1016/j.engappai.2006.06.017)
- [47] A. A. Suratgar, M. B. Tavakoli and A. Hoseinabadi, "Modified Levenberg-Marquardt Method for Neural Networks Training," *World Academy of Science, Engineering and Technology*, Vol. 6, 2005, pp. 46-48.
- [48] Y. Tanoto, W. Ongsakul and C. O. P. Marpaung, "Levenberg-Marquardt Recurrent Networks for Long- Term Electricity Peak Load Forecasting," *Telkomnika*, Vol. 9, No. 2, 2011, pp. 257-266.
- [49] C. M. Bishop, "Neural Networks for Pattern Recognition," Oxford University Press, Oxford, 1995.
- [50] F. Rotella and P. Borne, "Théorie et Pratique du Calcul Matriciel," Editions Technip, Paris, 1995.
- [51] Z. Dostál, T. Kozubek, A. Markopoulos and M. Menšík, "Cholesky Decomposition of A Positive Semi Definite Matrix with Known Kernel," *Applied Mathematics and Computation*, Vol. 217, No. 13, 2011, pp. 6067-6077. [doi:10.1016/j.amc.2010.12.069](https://doi.org/10.1016/j.amc.2010.12.069)
- [52] A. A. Adegbege and W. P. Heath, "Stability Conditions for Constrained Two-Stage Internal Model Control," 49th *IEEE Conference on Decision and Control (CDC)*, Atlanta, 15-17 December 2010, pp. 5251-5256.
- [53] R. Isermann, "Practical Aspects of Process Identification," *Automatica*, Vol. 16, No. 5, 1980, pp. 575-585. [doi:10.1016/0005-1098\(80\)90079-5](https://doi.org/10.1016/0005-1098(80)90079-5)
- [54] J. E. Nash and J. V. Sutcliffe, "River Flow Forecasting Through Conceptual Models Part I—A Discussion of Principles," *Journal of Hydrology*, Vol. 10, No. 3, 1970, pp. 282-290. [doi:10.1016/0022-1694\(70\)90255-6](https://doi.org/10.1016/0022-1694(70)90255-6)
- [55] S. A. Billings and Q. M. Zhu, "Nonlinear Model Validation Using Correlation Tests," *International Journal of Control*, Vol. 60, No. 6, 1994, pp. 1107-1120. [doi:10.1080/00207179408921513](https://doi.org/10.1080/00207179408921513)
- [56] S. A. Billings, H. B. Jamaluddin and S. Chen, "Properties of Neural Networks with Applications to Modelling Non-Linear Dynamical Systems," *International Journal of Control*, Vol. 55, No. 1, 1992, pp. 193-224. [doi:10.1080/00207179208934232](https://doi.org/10.1080/00207179208934232)

Appendixs

The model obtained from the estimation of its parameters is valid strictly used for the experiment. So check it is compatible with other forms of input in order to properly represent the system operation to identify. Most static tests of model validation are based on the criterion of Nash, on the auto-correlation of residuals, based on cross-correlation between residues and other inputs to the system. According to [54], the Nash criterion is given by the following equation:

$$Q = 100\% \left(1 - \frac{\sum_{k=1}^N (y(k) - \hat{y}(k))^2}{\sum_{k=1}^N \left(y(k) - \left(\frac{1}{N} \sum_{k=1}^N y(k) \right) \right)^2} \right) \quad (87)$$

In our work, the number of samples N is equal to 2251.

In [55,56], the correlation functions are:

-autocorrelation function of residuals:

$$R_{\varepsilon\varepsilon}(\tau) = \frac{\sum_{k=1}^{N-\tau} \left(\varepsilon(k) - \left(\frac{1}{N} \sum_{k=1}^N \varepsilon(k) \right) \right) \left(\varepsilon(k-\tau) - \left(\frac{1}{N} \sum_{k=1}^N \varepsilon(k) \right) \right)}{\sum_{k=1}^N \left(\varepsilon(k) - \left(\frac{1}{N} \sum_{k=1}^N \varepsilon(k) \right) \right)^2} \quad (88)$$

-crosscorrelation function between the residuals and the previous entries:

$$R_{u\varepsilon}(\tau) = \frac{\sum_{k=1}^{N-\tau} \left(u(k) - \left(\frac{1}{N} \sum_{k=1}^N u(k) \right) \right) \left(\varepsilon(k-\tau) - \left(\frac{1}{N} \sum_{k=1}^N \varepsilon(k) \right) \right)}{\sqrt{\sum_{k=1}^N \left(u(k) - \left(\frac{1}{N} \sum_{k=1}^N u(k) \right) \right)^2} \sqrt{\sum_{k=1}^N \left(\varepsilon(k) - \left(\frac{1}{N} \sum_{k=1}^N \varepsilon(k) \right) \right)^2}} \quad (89)$$

Ideally, if the model is validated, the results of correlation tests and the Nash criterion following results:

$$\hat{R}_{\varepsilon\varepsilon}(\tau) = \begin{cases} 1, \tau = 0 \\ 0, \tau \neq 0 \end{cases}, R_{u\varepsilon}(\tau) = 0 \forall \tau \text{ and } Q = 100\%$$

Typically, we verify that $Q \cong 100\%$ and the functions \hat{R} are null for the interval $\tau \in [-20, 20]$ with a confidence interval 95%, that is to say that:

$$-\frac{1.96}{\sqrt{N}} \leq \hat{R} \leq \frac{1.96}{\sqrt{N}}$$

The calculation of partial derivatives is carried by the following equations:

-for the neuron in the output layer:

$$\frac{\partial \hat{y}(k)}{\partial w_{11}^7} = \hat{y}(k-1) + w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{11}^7} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{11}^7} + \sum_{j=1}^{n_h} w_{1j}^3 \frac{\partial x_j^h(k)}{\partial w_{11}^7} \quad (90)$$

$$\frac{\partial x_j^h(k)}{\partial w_{11}^7} = f_1'(s_j(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{11}^7} + \sum_{i=1}^{n_h} w_{ji}^6 \frac{\partial x_i^h(k-1)}{\partial w_{11}^7} \right) \quad (91)$$

$$\frac{\partial \hat{y}(k)}{\partial w_{1j}^3} = x_j^h(k) + w_{1j}^3 \frac{\partial x_j^h(k)}{\partial w_{1j}^3} + w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{1j}^3} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{1j}^3} \quad (92)$$

$$\frac{\partial x_j^h(k)}{\partial w_{1j}^3} = f_1'(s_j(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{1j}^3} + \sum_{m=1}^{n_h} w_{jm}^6 \frac{\partial x_m^h(k-1)}{\partial w_{1j}^3} \right) \quad (93)$$

$$\frac{\partial \hat{y}(k)}{\partial w_{1m}^5} = \psi_m(k) + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{1m}^5} + \sum_{j=1}^{n_h} w_{1j}^3 \frac{\partial x_j^h(k)}{\partial w_{1m}^5} + w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{1m}^5} \quad (94)$$

$$\frac{\partial x_j^h(k)}{\partial w_{1j}^5} = f_1'(s_j(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{1j}^5} + \sum_{m=1}^{n_h} w_{jm}^6 \frac{\partial x_m^h(k-1)}{\partial w_{1j}^5} \right) \quad (95)$$

$$\frac{\partial \hat{y}(k)}{\partial w_{11}^4} = 1 + w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{11}^4} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{11}^4} + \sum_{j=1}^{n_h} w_{1j}^3 \frac{\partial x_j^h(k)}{\partial w_{11}^4} \quad (96)$$

$$\frac{\partial x_j^h(k)}{\partial w_{11}^4} = f_1'(s_j(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{11}^4} + \sum_{m=1}^{n_h} w_{jm}^6 \frac{\partial x_m^h(k-1)}{\partial w_{11}^4} \right) \quad (97)$$

-for a neuron in the hidden layer:

$$\frac{\partial \hat{y}(k)}{\partial w_{jm}^6} = w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{jm}^6} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^6} + \sum_{n=1}^{n_h} w_{1n}^3 \frac{\partial x_n^h(k)}{\partial w_{jm}^6} \tag{98}$$

if $n = j$

$$i \frac{\partial x_j^h(k)}{\partial w_{jm}^6} = f_1'(s_j(k))(x_m^h(k-1) + \sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^6} + \sum_{i=1}^{n_h} w_{ji}^6 \frac{\partial x_i^h(k-1)}{\partial w_{jm}^6}) \tag{99}$$

if $n \neq j$

$$i \frac{\partial x_n^h(k)}{\partial w_{jm}^6} = f_1'(s_n(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ni}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^6} + \sum_{i=1}^{n_h} w_{ni}^6 \frac{\partial x_i^h(k-1)}{\partial w_{jm}^6} \right) \tag{100}$$

$$\frac{\partial \hat{y}(k)}{\partial w_{j1}^2} = w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{j1}^2} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{j1}^2} + \sum_{n=1}^{n_h} w_{1n}^3 \frac{\partial x_n^h(k)}{\partial w_{j1}^2} \tag{101}$$

if $n = j$

$$\frac{\partial x_n^h(k)}{\partial w_{j1}^2} = f_1'(s_j(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{j1}^2} + \sum_{m=1}^{n_h} w_{jm}^6 \frac{\partial x_m^h(k-1)}{\partial w_{j1}^2} + 1 \right) \tag{102}$$

if $n \neq j$

$$\frac{\partial x_n^h(k)}{\partial w_{j1}^2} = f_1'(s_n(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ni}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{j1}^2} + \sum_{m=1}^{n_h} w_{nm}^6 \frac{\partial x_m^h(k-1)}{\partial w_{j1}^2} \right) \tag{103}$$

$$\frac{\partial \hat{y}(k)}{\partial w_{jm}^1} = w_{11}^7 \frac{\partial \hat{y}(k-1)}{\partial w_{jm}^1} + \sum_{i=n_a+n_b+1}^{n_r} w_{1i}^5 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^1} + \sum_{n=1}^{n_h} w_{1n}^3 \frac{\partial x_n^h(k)}{\partial w_{jm}^1} \tag{104}$$

if $n = j$

$$\frac{\partial x_j^h(k)}{\partial w_{jm}^1} = f_1'(s_j(k)) \left(\psi_m(k) + \sum_{i=n_a+n_b+1}^{n_r} w_{ji}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^1} + \sum_{i=1}^{n_h} w_{ji}^6 \frac{\partial x_i^h(k-1)}{\partial w_{jm}^1} \right) \tag{105}$$

if $n \neq j$

$$\frac{\partial x_n^h(k)}{\partial w_{jm}^1} = f_1'(s_n(k)) \left(\sum_{i=n_a+n_b+1}^{n_r} w_{ni}^1 \frac{\partial \hat{y}(k+n_a+n_b-i)}{\partial w_{jm}^1} + \sum_{i=1}^{n_h} w_{ni}^6 \frac{\partial x_i^h(k-1)}{\partial w_{jm}^1} \right) \tag{106}$$

The calculation of the coefficient matrix of the network D is conducted via the two following relations [39]:

$$\frac{\partial \hat{y}(k)}{\partial Wc} = \sum_{m=1}^{n_b} \left(\frac{\partial \hat{y}(k)}{\partial u(k-n_k-m)} \right) \left(\frac{du(k-n_k-m)}{dWc} \right) + \sum_{m=1}^{n_c} \left(\frac{\partial \hat{y}(k)}{\partial \hat{y}(k-m)} \right) \left(\frac{d\hat{y}(k-m)}{dWc} \right) \tag{107}$$

$$\frac{du(k-1)}{dWc} = \frac{\partial u(k-1)}{\partial Wc} + \sum_{m=2}^{n_b} \left(\frac{\partial u(k-1)}{\partial u(k-m)} \right) \left(\frac{du(k-m)}{dWc} \right) + \sum_{m=1}^{n_a} \left(\frac{\partial u(k-1)}{\partial \hat{y}(k+n_k-m)} \right) \left(\frac{d\hat{y}(k+n_k-m)}{dWc} \right) \tag{108}$$

If we denote:

$$B(k-1) = \frac{du(k-1)}{dWc} \tag{109}$$

We can write the equations (107) and (108) as follows:

$$D(k) = \sum_{m=1}^{n_b} \left(\frac{\partial \hat{y}(k)}{\partial u(k-n_k-m)} \right) B(k-n_k-m) + \sum_{m=1}^{n_c} \left(\frac{\partial \hat{y}(k)}{\partial \hat{y}(k-m)} \right) D(k-m) \tag{110}$$

$$B(k-1) = \frac{\partial u(k-1)}{\partial Wc} + \sum_{m=2}^{n_b} \left(\frac{\partial u(k-1)}{\partial u(k-m)} \right) B(k-m) + \sum_{m=1}^{n_a} \left(\frac{\partial u(k-1)}{\partial \hat{y}(k+n_k-m)} \right) D(k+n_k-m) \tag{111}$$

The terms $\frac{\partial \hat{y}(k)}{\partial \hat{y}(k-m)}$, $\frac{\partial \hat{y}(k)}{\partial u(k-n_k-m)}$, $\frac{\partial u(k-1)}{\partial u(k-m)}$ and

$\frac{\partial u(k-1)}{\partial \hat{y}(k+n_k-m)}$ are called the system Jacobian. They can be determined through the controller considered and the use of numerical approximation of the partial derivative:

$$\frac{\partial \hat{y}(k)}{\partial u(k)} \cong \frac{\hat{y}(k) - \hat{y}(k-1)}{u(k) - u(k-1)} \quad (112)$$

We have then the following approximate relations:

$$\frac{\partial \hat{y}(k)}{\partial u(k-1)} \cong \frac{\hat{y}(k) - \hat{y}(k-1)}{u(k-1) - u(k-2)} \quad (113)$$

$$\frac{\partial \hat{y}(k)}{\partial u(k-n_k-m)} = \left(\frac{\partial \hat{y}(k)}{\partial u(k-1)} \right) \left(\frac{\partial u(k-1)}{\partial u(k-n_k-m)} \right) \quad (114)$$

$$\frac{\partial \hat{y}(k)}{\partial \hat{y}(k-m)} = \left(\frac{\partial \hat{y}(k)}{\partial u(k-1)} \right) \left(\frac{\partial u(k-1)}{\partial \hat{y}(k-m)} \right) \quad (115)$$

$$\frac{\partial u(k-1)}{\partial u(k-m)} = \left(\frac{\partial u(k-1)}{\partial \hat{y}(k)} \right) \left(\frac{\partial \hat{y}(k)}{\partial u(k-m)} \right) \quad (116)$$

The Jacobians $\frac{\partial u(k-1)}{\partial u(k-m)}$ and $\frac{\partial u(k-1)}{\partial \hat{y}(k+n_k-m)}$ are calculated directly by the back-propagation of the partial derivative of output network $u(k-1)$ respectively compared to $u(k-m)$ and $\hat{y}(k-n_k-m)$ through the neuronal controller.