3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)
Proceedings of Engineering & Technology (PET)
pp. 682-688

Comparative Study of Adaptive and Recurrent Neural IMC controller for DC Motor

Ben Mabrouk Zaineb[#1], Abid Aicha[*2], Ben Hamed Mouna[#3], Sbita Lassad[#4]

[#]*Department of electrical & automatic, National engineering school of Gabes*
Avenue Omar Ibn El Khattab Zrig Gabes6029 Tunisia
[1]benmabroukza@gmail.com
[3]benhamed2209@ gmail.com

Avenue Omar Ibn El Khattab Zrig Gabes6029 Tunisia Tunisia
[2]aicha.abid@gmail.com

***Abstract*— This paper presents the design and implementation of IMC controller for speed control of DC Motor using artificial neural network (ANN). The neural IMC approach comprises two parts. One is the neural identifier, which is used to identify the model of a DC Motor. The other is the neural controller, which is used to generate a control signal. We apply adaptive and recurrent neural IMC controllers. Then, we compare the speed control of DC Motor with the two controllers. The simulation results show the effectiveness and an advantage of the recurrent neural IMC controller than an adaptive neural IMC controller.**

***Keywords*— Internal Model Controller (IMC), Artificial Neural Networks (ANN), DC motor, Speed control.**

I. Introduction

The increasing complexity of industrial systems requires the exploitation of powerful and efficient tools and techniques for modeling and control. Among these emerging techniques, the artificial neural network (A.N.N.) is developed for robust control [1].

Neural applications require only a set of input/output database collected from the real system. Then a neural network is obtained after a learning procedure via an adequate learning algorithm. The back-propagation algorithm is the most used [2].

Neural network, that has high competence to reproduce the dynamic behavior of a complex system, can be exploited simultaneously to represent an identifier or a controller. Thus, the combination of the internal model controller (IMC) technique that is known as a robust controller and the ANN technique appears as a powerful modeling and control tool [3].

Based on its architecture, there are several types of neural networks. In this paper two types are exploited to control the DC motor. The first is the adaptive feed forward, while the second is the recurrent neural network. To test the effectiveness of these two types, two neural IMC controllers are developed and implemented to control the DC motor speed. Then the two controllers are compared [4].

The organization of this paper is as follows. The second part presents the classic model of the DC motor. Part three is devoted to the neural IMC controller design. Part four is dedicated to the discussion of the simulation results and the comparison between the two neural IMC controllers.

II. Modeling of DC Motor

We use in this paper the DC Motor for our analysis of neural control. It is widely used in industrial applications. It is described by the following equations [5].

$$u_a = R_a i_a + L_a \frac{di_a}{dt} + E \tag{1}$$

$$j \frac{d\Omega}{dt} = C_{em} - C_r - C_v \tag{2}$$

where

u_a: the armature winding input voltage
E: the back-electromotive-force (EMF) voltage
L_a: the armature winding inductance
i_a : the armature winding current
R_a: the armature winding resistance
Ω: the rotor angular speed
j : the system moment of inertia
C_v: the load torque
C_r : the torque constant and C_{em}

The Laplace transform of the equations (1) and (2) is:

$$\begin{cases} U_a(p) = R_a I_a(p) + L_a p I_a(p) + K\Omega(p) \\ jp\Omega(p) = KI_a(p) - f\Omega(p) \end{cases} \tag{3}$$

The transfer function of the DC motor is given by the equation (4):

$$y(p) = \frac{K}{(R_a + L_a(p))(f + Jp) + K^2} U(p) \tag{4}$$

Table 1 shows the physical parameters of DC motor.

TABLE I
DC MOTOR PARAMETERS

R_a	Resistance	8 Ω
L_a	Inductance	0.129 H
R_f	Resistance	100 Ω
L_f	Inductance	0.2 H

ISSN: 2356-5608

J	Moment of Inertia	0.02 Kg-m^2
f	viscous friction coefficient	0.0218 Nm-sec/rad

Substitute the values of the DC motor parameters in equation (4) gives.

$$y(p) = \frac{0.7745}{0.00258p^2 + 0.1628p + 0.7743} U(p) \quad (5)$$

In the next section we use recurrent equation (6).

$$y(k) = 1.51y(k-1) - 0.532y(k-2) + 0.0126U(k-1) + 0.009942U(k-2) \quad (6)$$

III. Neural IMC Controller

The artificial neural networks have two phases. One is the learning off line. The other is the phase of implemented in the process. The adaptive neural network needs learning off line and it updates his weights and biases online. On the other hand the recurrent neural network need only learning online. It doesn't need off line training. It is a real-time adaptive controller. The neural IMC controller used the artificial neural networks to replace the model of the system by direct neural model and the controller also replaced by the inverse neural model.

A. Adaptive Neural IMC Design

1) Adaptive Neural Network Identifier (ANNI)

The direct neural model as shown in fig. 1 attained after simulation testing. We used the input vector shown by equation (7) for excitation of the process.

$$U = [u_0\ u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7\ u_8\ u_9\ u_{10}\ u_{11}] \quad (7)$$

The direct neural model has three layers, the input layer has four neurons, one hidden layer has seven neurons and an output layer has one neuron.

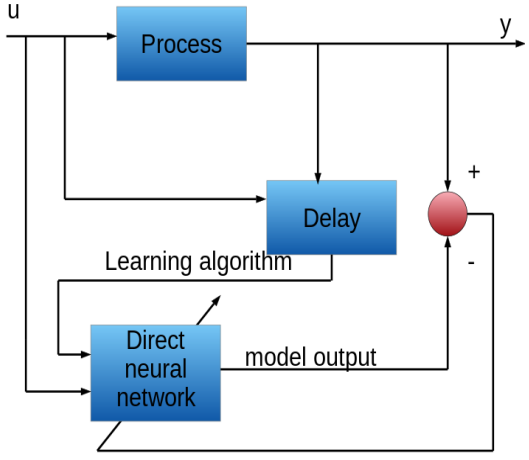


Fig. 1. The direct neural model of process.

2) Adaptive Neural Network Controller (ANNC)

The inverse neural model is described by the fig. 2. It contains three layers, the input layer has six neurons, one hidden layer has seven neurons and an output layer has one neuron. We find after several iterations the best learning coefficient η=0.02.

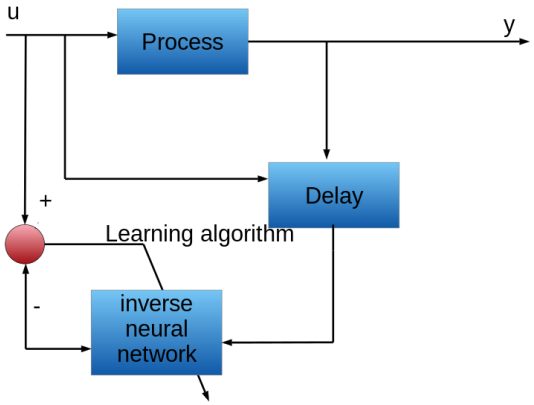


Fig. 2. The inverse neural model of process.

The mathematical equations of the ANN are given by the equation (8)-(10).

$$h\_in = Wx + w0 \quad (8)$$

$$h = f(h\_in) = f(Wx + w0) \quad (9)$$

$$o = g(Zh + z0) \quad (10)$$

h_in: the hidden layer inputs.
h: a transfer function .
o: the response of the output layer

Or *W* and *w0*, denote respectively, weight matrix (L, I) and biases vectors. *x* the input vector (J, 1). *Z* and *z0*, denote respectively, weight matrix and biases vectors.

The gradient back propagation algorithm is used to compute the error gradient for neurons of the last layer then to the first layer as defined by equations (11)-(14).

$$o_k = g\left[Z(f(Wx_k + w0))\right] \quad (11)$$

$$e_k = T_k - o_k \quad (12)$$

$$E_k = e_k^T e_k = \frac{1}{2}(T_k^T T_k + o_k^T o_k - 2o_k^T T_k) \quad (13)$$

$$\nabla E_{k/z} = \frac{1}{2}\nabla[o_k^T o_k - 2o_k^T T_k]_{/z} \quad (14)$$

If $f \equiv g$

$$o_k = f[Zh_k + z0] \quad (15)$$

$$\nabla E_{k/z} = \frac{\partial E_k}{\partial Z} = \frac{\partial E_k}{\partial o_k}\frac{\partial o_k}{\partial (Zh_k + z0)}\frac{\partial (Zh_k + z0)}{\partial Z} \quad (16)$$

$$\frac{\partial E_k}{\partial o_k} = o_k - T_k \quad (17)$$

$$\frac{\partial o_k}{\partial (Zh_k + z0)} = f(Zh_k + z0).[1 - f(Zh_k + z0)] = o_k.(1 - o_k) \quad (18)$$

$$\frac{\partial(Zh_k + z0)}{\partial Z} = h_k^T \quad (19)$$

The adjusting of the weight value *Z* and biases *z0* using equations (16), (17), (18) and (19) is:

$$Z(k+1) = Z(k) - \eta \nabla E_{k/z} = Z(k) + \eta (T_k - o_k).o_k.(1-o_k)h_k^T \quad (20)$$
$$= Z(k) + \eta\ \partial s\ h_k^T$$

$$z0(k+1) = z0(k) - \eta \nabla E_{k/z0} = z0(k) + \eta (T_k - o_k).o_k.(1-o_k) \quad (21)$$
$$= z0(k) + \eta\ \partial s$$

The adjusting of the weights *W* and *w0* is:

$$W(k+1) = W(k) - \eta \nabla E_{k/W} = W(k) + \eta Z(k)^T \partial s.h_k.(1-h_k)x_k^T \quad (22)$$
$$= W(k) + \eta\ \partial h\ x^T$$

$$w0(k+1) = w0(k) - \eta \nabla E_{k/w0} = w0(k) + \eta Z(k)^T \partial s.h_k.(1-h_k) \quad (23)$$
$$= w0(k) + \partial h$$

T_k is the target.

The propagation algorithm can be summarized by the diagram presented by the fig. 3.

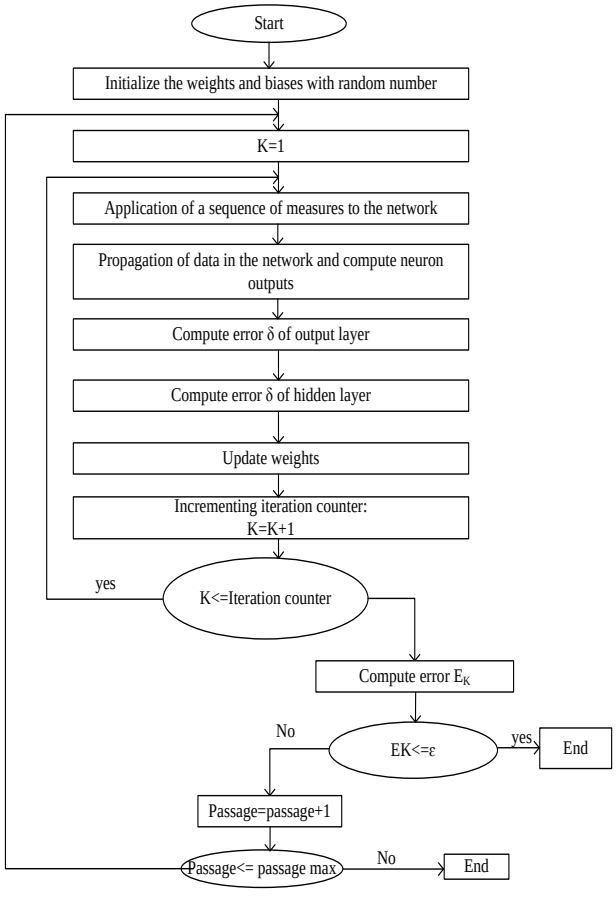


Fig. 3 Example of an unacceptable low-resolution image

The schema of the Adaptive neural controller of DC Motor is shown in the fig. 4

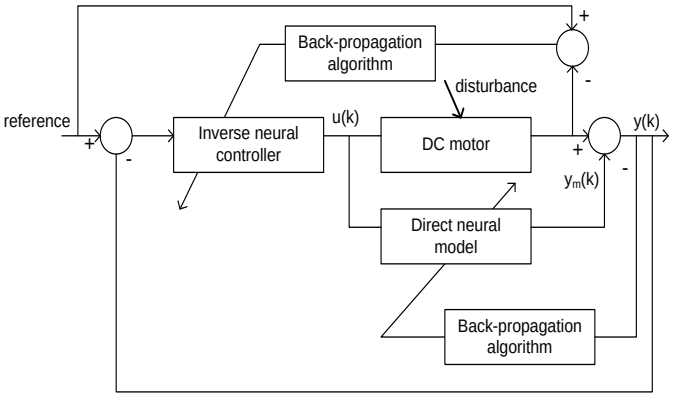


Fig. 4. The Adaptive internal model control (IMC) scheme.

B. Recurrent Neural IMC controller Design

The fig. 5 presents the scheme of the recurrent neural controller of a DC Motor.

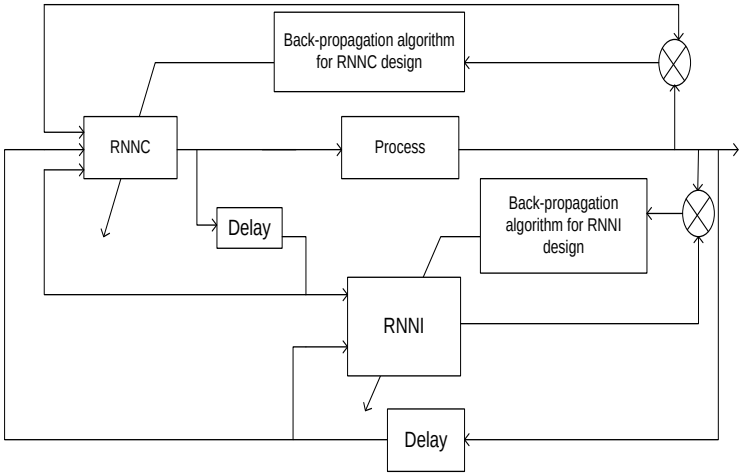


Fig. 5. The Recurrent internal model control (IMC) scheme.

1) Recurrent Neural Network Identifier (RNNI)

The RNNI goal is to identify the direct model of DC machine. Fig. 6 presents the RNNI architecture composed of 3 layers: Input layer (I: input), A hidden layer (D, L) and an output layer (O: output).

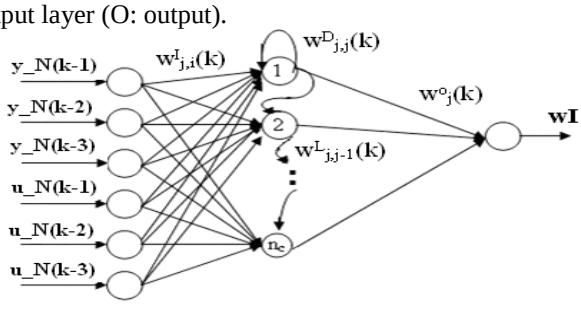


Fig. 6 RNNI structure.

y_N (k); y_N (k-1); y_N (k-2) are the normalized outputs of the process and u_N (k-1); u_N (k-2) are the normalized control law. The RNNI is composed of three neurons in the hidden layer and a unique neuron in the output layer.

The mathematical equations of the RNNI are given by the equations (25)-(27).

$$S_j(k) = w^D_{j,j}(k)X_j(k-1) + \sum_{i=1}^{m} w^I_{j,i}(k)I_i(k) + w^L_{j,j-1}(k)X_{j-1}(k-1), \quad j=1,...,n \quad (25)$$

$$X_j(k) = f(s_j(k)) = \frac{1}{1+\exp(-s_j(k))}, \quad j=1,...,n \quad (26)$$

$$w_I(k) = \sum_{j=1}^{n} w^o_j X_j(k) \quad (27)$$

with:
$(I_1,\ldots I_m)$: The network inputs.
$(S_1,\ldots,S_n)$:The hidden layer inputs.
$(X_1,\ldots,X_n)$: The hidden layer outputs.
$w_I(k)$:The network output.
$w^I_{j,i}(k)$: the weights that link the inputs and the hidden layer.
$w^D_{j,j}(k)$, $w^L_{j,j-1}(k)$: The hidden layer weights.
$w^o_j(k)$: the weight that connect the hidden layer to the output layer.
n : number of the hidden layer neurons.
Remark:

$w^L_{1,0}(k) = w^L_{1,n}(k)$ and $X_0(k) = X_n(k)$.

The back propagation algorithm is presented by the equation (28).

$$E_I(k) = \frac{1}{2}(y(k) - w_I(k))^2 = \frac{1}{2}e_I^2(k) \quad (28)$$

The gradient method is used to update the RNNI weights described by equations (29)-(32).

$$w^o_j(k+1) = w^o_j(k) - n^o_I \frac{\partial E_I(k)}{\partial w^o_j(k)} \quad (29)$$

$$w^I_{j,i}(k+1) = w^I_{j,i}(k) - n^I_I \frac{\partial E_I(k)}{\partial w^I_{j,i}(k)} \quad (30)$$

$$w^D_{j,j}(k+1) = w^D_{j,j}(k) - n^D_I \frac{\partial E_I(k)}{\partial w^D_{j,j}(k)} \quad (31)$$

$$w^L_{j,j-1}(k+1) = w^L_{j,j-1}(k) - n^L_I \frac{\partial E_I(k)}{\partial w^L_{j,j-1}(k)} \quad (32)$$

Where n^o_I, n^D_I, n^L_I and n^I_I are the learning rates for the RNNI. The error gradients are described by the equations (34)-(37):

$$\frac{\partial E_I(k)}{\partial w^o_j(k)} = -e_I(k)X_j(k) \quad (34)$$

$$\frac{\partial E_I(k)}{\partial w^D_{j,j}(k)} = -e_I(k)w^o_j(k)P_{j,j}(k) \quad (35)$$

$$\frac{\partial E_I(k)}{\partial w^I_{j,i}(k)} = -e_I(k)w^o_j(k)Q_{j,i}(k) \quad (36)$$

$$\frac{\partial E_I(k)}{\partial w^L_{j,j-1}(k)} = -e_I(k)w^o_j(k)R_{j,j-1}(k) \quad (37)$$

With:

$$P_{j,j}(k) = X_j(k)(1-X_j(k))(X_j(k-1) + w^D_{j,j}(k)P_{j,j}(k-1)) \quad (38)$$

$$Q_{j,i}(k) = X_j(k)(1-X_j(k))(I_i(k) + w^D_{j,j}(k)Q_{j,i}(k-1)) \quad (39)$$

$$R_{j,j-1}(k) = X_j(k)(1-X_j(k))(X_{j-1}(k-1) + w^D_{j,j}(k)R_{j,j-1}(k-1)) \quad (40)$$

The RNNI algorithm is expressed as follows:
Step 1: initialize the weights $w^I_{j,i}(k)$, $w^D_{j,j}(k)$, $w^L_{j,j-1}(k)$, $w^o_j(k)$ and $P_{j,j}(k-1)$,$Q_{j,i}(k-1)$, $R_{j,j-1}(k-1)$ with k=1, and the learning rates n_I^o, n_I^D, n_I^L and n_I^I with random values.
Step 2: compute $S_j(k)$, $X_j(k)$ and $w_I(k)$
Step 3: compute $P_{j,j}(k)$, $Q_{j,i}(k)$, $R_{j,j-1}(k)$
Step 4: compute the gradients of the error
Step 5: compute the new weights $w^I_{j,i}(k+1)$, $w^D_{j,j}(k+1)$, $w^L_{j,j-1}(k+1)$, $w^o_j(k+1)$.
Then, return to step 2.

2) Recurrent Neural Network Controller RNNC

The RNNC is the inverse model of the DC motor. The main is to ensure the speed control using the back propagation algorithm. The fig. 7 presents the network structure composed of three layers: a layer input, a hidden layer and an output layer.

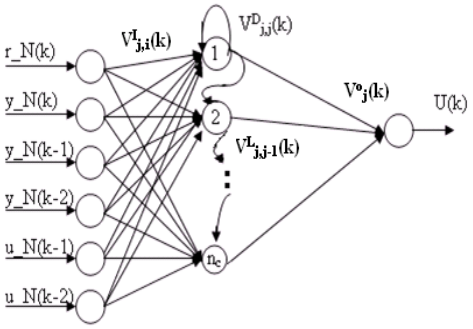


Fig. 7 RNNC structure.

The principle of the algorithm used for the learning of the RNNC network is similar to that used by RNNI.
$(T_1,\ldots,T_{nc})$: The hidden layer inputs.
$(Z_1,\ldots,Z_{nc})$: The hidden layer outputs.
$V_I(k)$: The network output.
$V^I_{j,i}(k)$: the weights that link the inputs and the hidden layer.
$V^D_{j,j}(k)$, $V^L_{j,j-1}(k)$: The hidden layer weights.
$V^o_j(k)$: the weight that connect the hidden layer to the output layer.
n_c : number of the hidden layer neurons.
Remark:
$V^L_{1,0}(k) = V^L_{1,n}(k)$ and $Z_0(k) = Z_n(k)$ (25)

IV. SIMULATION RESULTS

We used Matlab/Simulink to simulate the DC Motor control system. The output responses of the system are obtained for controllers used in this paper. A load is applied at t =10s. Fig. 8 describes the control signal evolution. Fig. 9

presents the speed responses of the DC motor using the adaptive neural controller. Fig. (10.a, 10b, 10.c, 10.d) show the adjustment of different weights and biases of the network of the adaptive neural controller under the effect of the disturbance and the variation of the input signal. W and w0 are the weights and biases between the input layer and the hidden layer of the corrector. Z and z0 are the weights and biases between the hidden layer and the output layer of the corrector.

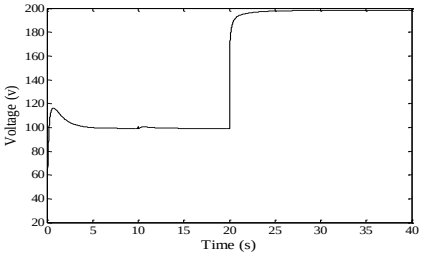


Fig. 8 Adaptive neural control: control signal

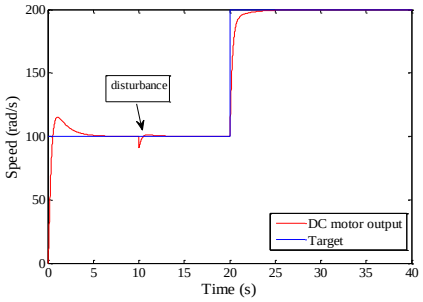
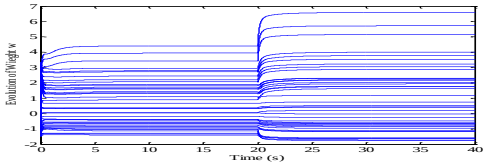
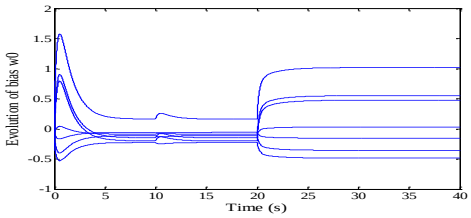


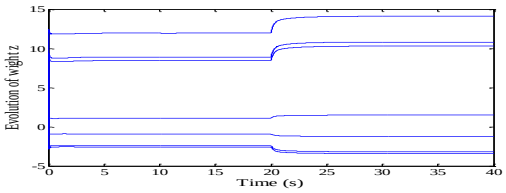
Fig. 9 Adaptive neural control: target and Dc motor outputs



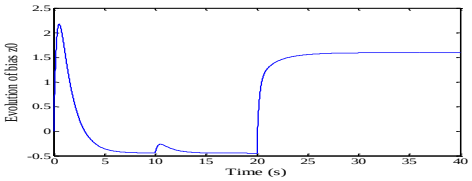
(a)



(b)



(c)



(d)

Fig. 10 Evolutions neural weights and biases of neural inverse model (w, w 0, z, z 0)

Fig. 11 illustrates the control signal growth of the recurrent neural controller. The evolution of target signal and the output signal of a DC machine using a recurrent neural controller are shown in fig. 12. Figures (13.a, 13b, 13.c, 13.d) display the adjustment of different weights of the network of recurrent neural controller under the influence of the disturbance and the variation of the target signal.

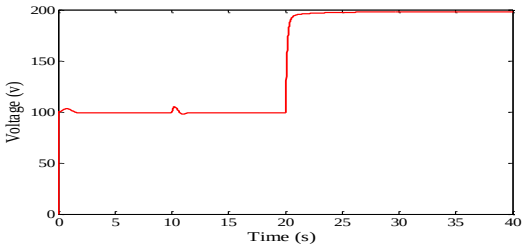


Fig. 11 Recurrent neural control: control signal

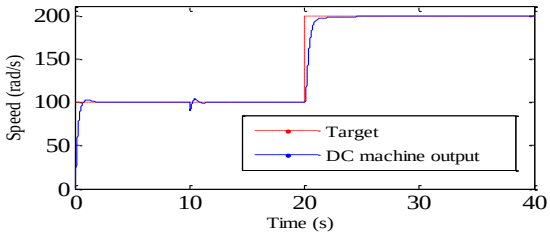
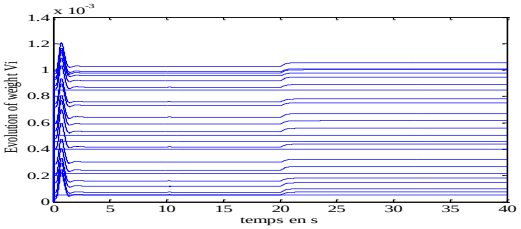
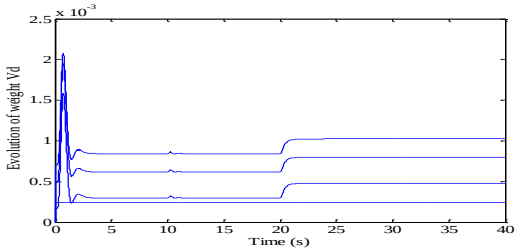


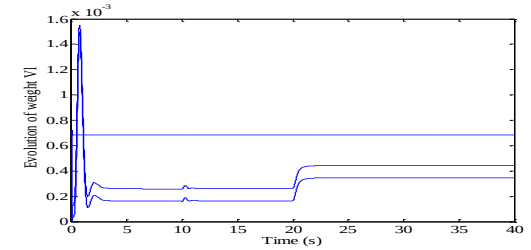
Fig. 12 Recurrent neural control: target and Dc motor outputs



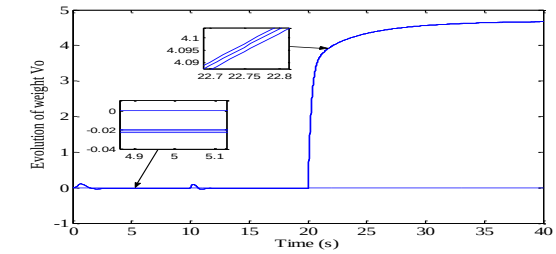
(a) Weights Vi



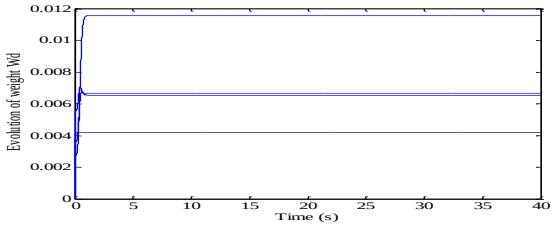
(b) Weights Vd



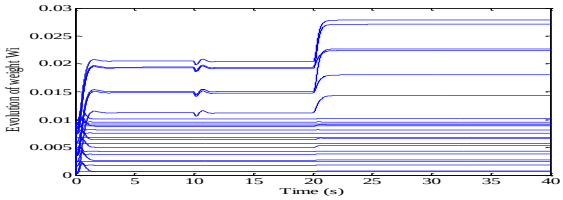
(c) Weights Vl



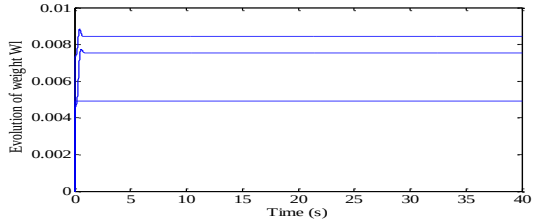
(d) Weights Vo



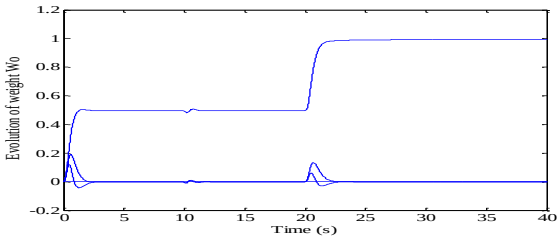
(e) Weights Wd



(f) Weights Wi



(g) Weights Wl



(h) Weights Wo

Fig. 13 Evolutions neural weights of neural inverse model

Figure 14 illustrates the comparison of speed evolution of DC Motor using an adaptive neural IMC controller and the recurrent neural controller.

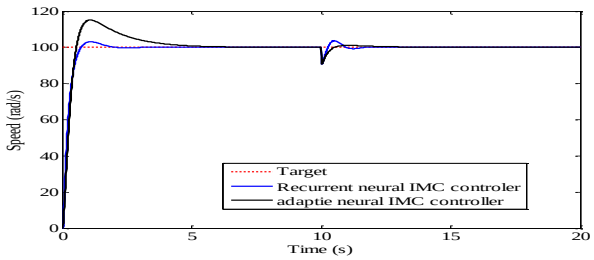


Fig. 14 Speed evolution by adaptive neural IMC control and recurrent neural control under disturbances

Table 2. illustrates the comparison of performance of speed control of DC Motor using the recurrent neural and an adaptive IMC control.

TABLE III
COMPARISON OF ADAPTIVE AND RECURRENT NEURAL CONTROLLER OF DC MOTOR

	Static error	**Response time (s)**	**overtaking**
Adaptive neural IMC	0	2.78	15%
Recurrent neural IMC	0	0.73	3%

The two controllers provide disturbance rejection and have static error equal to zero, but the recurrent neural IMC has the less response time and the less overtaking. Also, it doesn't need training off line. The simulation results show the effectiveness and an advantage of the recurrent neural IMC controller than an adaptive neural IMC controller.

V. CONCLUSION

In this work, two neural IMC controllers are presented and applied for speed control of DC motor. The neural IMC controller has two functions: estimating and controlling the speed of DC motor. The simulation results have shown an appreciable advantage of control system using the recurrent neural IMC controller than an adaptive neural IMC controller.

To improve the performance of the neural control approach an experimental set up is to be performed in the future.

REFERENCES

[1] T. J. Ren and T. C. Chen, "Robust speed-controlled induction motor drive based on recurrent neural network '', *Electric Power Systems Research 76*, pp. 1064-1074, 2006.

[2] B. Pryymak, J. M. Moreno-Eguilaz and J. Peracaula, "Neural network flux optimization using a model of losses in induction motor drives", *Mathematics and Computers in Simulation*, pp. 290–298, 2006

[3] F. Zouari, K. Ben Saad and M. Benrejeb, "Adaptive Internal Model Control of a DC Motor Drive System Using Dynamic Neural Network", *Journal of Software Engineering and Applications*, vol. 5, pp. 168-189, 2012.

[4] EA. Feilat, EK. Ma'aitab, "RBF Neural Network Approach for Identification and Control of DC Motors", *TJER*, vol. 9, no. 2, pp. 80-89, 2012.

[5] Sh. Bhushan Kumar, M. Hasmat Ali and A. Sinha, "Design and Simulation of Speed Control of DC Motor by Artificial Neural Network Techniqu ", *International Journal of Scientific and Research Publications*, vol. 4, no. 7, July. 2014.