International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 4, N° 2, December 2010, pp. 1298–1319.

Neural emulator and controller with decoupled adaptive rates for nonlinear systems: application to chemical reactors

ATIG Asma*§, DRUAUX Fabrice*, LEFEBVRE Dimitri*, ABDERRAHIM Kamel§ and BEN ABDENNOUR Ridha §

* Groupe de Recherche en Electrotechnique et Automatique du Havre (GREAH)
Université du havre
25, Rue Philippe Lebon, 76063 La Havre, France.
atigasmatica@yahoo.fr fabrice.druaux@univ-lehavre.fr
dimitri.lefebvre@univ-lehavre.fr
§Numerical Control of Industrial Processes (CONPRI)
Université de Gabès, École Nationale d'Ingénieurs de Gabès,
Route de Medenine, 6029 Gabès, Tunisie
kamelabderrahim@yahoo.fr Ridha.benabdennour@enig.rnu.tn

Abstract. In this paper, we develop an indirect adaptive control structure based on recurrent neural networks. An adaptive emulator inspired from the Real Time recurrent Learning algorithm is presented. Neural network does not learn the plant dynamics but emulates the input-output mapping with a small time window. Thereafter, a controller with a structure similar to neural emulator is described. Both emulator and controller adapt their parameters using an online adaptation algorithm in order to track the process variations. Independent adaptation of networks parameters improve controller performances. Regulation and tracking problems are investigated according to nonlinear system simulations. The satisfactory obtained results show a very good performances in terms of neural emulation and control of nonlinear systems. The contributions of this paper are the validation of our emulator with experimental data from the batch reactor of National Engineering of Gabes, Tunisia and the application of the real time control algorithm with decoupled adaptive rates to large scale process: Tennessee Eastman Challenge Process (TECP).

Keywords: Adaptive control, Recurrent neural networks, Neural emulator, RTRL algorithm, Parameters adaptation, Chemical reactors.

1 Introduction

Most industrial processes, particularly chemical systems, are large dimensional, nonlinear, do not have a known mathematical representation and are constantly responding to disturbances that are unmeasurable and occurring at unknown

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

times [11]. A plant wide control system has received increasing attention in the last few years. Since neural networks can approximate any nonlinear functions with arbitrary accuracy [9, 18, 10], they have been successfully used for a number of chemical engineering applications [24, 29, 28, 1, 2]. Adaptive control by means of neural networks for nonlinear plant wide dynamical systems is an open issue [3, 8, 19, 30]. Multilayer networks require, therefore, a large number of nodes for catching the dynamics of the plant [21].

Recurrent neural networks, which contain internal feedback loop, seem an efficient tool to provide system control with high performance, according to their ability to capture input-output mapping [4, 5, 20]. R.J. William and D. Zipser have proposed in 1989 [22] a real time recurrent learning based on the gradient backpropagation learning algorithm. An indirect adaptive control scheme appears as an alternative solution to control nonlinear plant wide systems thanks to its low sensitivity to the noise and better disturbance rejection [10, 18]. The main advantage of this indirect real time recurrent learning-based control scheme is that it requires none dynamical model of the process, none particular initialization and none a priori training of the neural networks [6]. The neural emulator and the neural controller parameters are self adapted starting from zero initial conditions [14].

The indirect neural network control scheme is composed of Neural Emulator (NE) and Neural Controller (NC). The subscripts e and c are used to distinguish the NE and NC respectively. The NE and NC adapt themselves, as well as for the adapting rate and the time parameter of both networks (Fig. 1) [2].

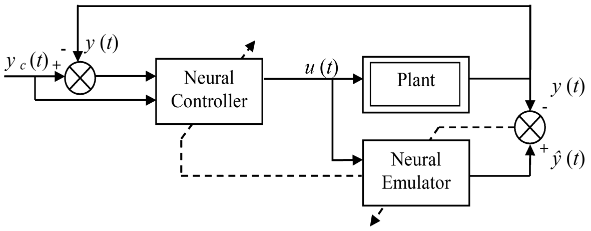

Fig. 1. Indirect neural control structure.

Let us define N_{IN} and N_{OUT}, respectively as the number of plant inputs and outputs where IN and OUT represent the set of inputs and outputs indexes [26]. In the following, we investigate square MIMO systems (ie. $N = N_{IN} = N_{OUT}$, where, $IN = \{1, \dots, N\}$ and $OUT = \{1, \dots, N\}$). The total number of neurons N_e of NE is chosen equal to $2N$ so that any node is either an input node or an output node but not both at same time in order to avoid perturbing any output signal with input ones. Additional nodes are useless [13]. For NC, the total number of neurons N_c is chosen equal to $2N$. The NC inputs are the N output error functions and eventually the N desired outputs. Inputs and outputs signals for NE, NC and plant are normalized in the range [-1, 1].

The contributions of this paper are the validation of our emulator with experimental data from the batch reactor of National Engineering of Gabes, Tunisia and the application of the real time control algorithm with decoupled adaptive rates to large scale process: Tennessee Eastman Challenge Process (TECP).
The paper is planned as follows. In section two, neural emulation is described. Efficiency of network-based algorithm is established according to switching system simulation. An experimental validation of such method to chemical reactor is also provided in this section. Neural controller and independent adaptation of networks parameters are provided in section three. Potential of neural control with decoupled rates is illustrated thanks to the simulation of a nonlinear system. A real time application of the proposed control structure to the TECP is presented.

2 Neural emulation and real time adaptation

2.1 Autonomous adaptation algorithm

The dynamics of the N_e neurons of neural emulator are defined by (1), for $i = 1, \ldots, N_e$, in continuous time:

$$\frac{1}{|\tau_e(t)|}\frac{d(s_i(t))}{dt} + s_i(t) = \tanh\left(\sum_{j=1}^{N_e} w_{ij}(t)s_j(t) + x_i(t)\right) \tag{1}$$

where, $x_i(t) = u_i(t)$ if $i \in IN$ and $x_{i+N}(t) = 0$ if $i \in OUT$, $s_i(t)$, w_{ij} and $1/|\tau_e|$ represent respectively the ith neuron state, the weights from jth neuron to ith neuron and the NE adaptive time parameter [6]. $\hat{y_i}(t) = s_{i+N}(t)$ if $i \in OUT$, represents the estimation of plant output $y_i(t)$.

In this work, we are not interested to memorize the dynamics of the system but just to estimate the instantaneous output. Consequently, the size of such network depends only on the number of inputs and outputs [13, 1]. The real time recurrent learning algorithm is used for the adaptation of neural emulator parameters and weights. The major advantage of such method is that doesn't depend on any preliminary knowledge about dynamics [23].

The instantaneous square output estimation error $E_e(t)$ is defined by (2):

$$E_e(t) = \frac{1}{2}\sum_{l=1}^{N} (\hat{y}_l(t) - y_l(t))^2 \tag{2}$$

where, $\hat{y_l}$ is the neural emulator output and y_l is the plant output. The adaptation of the weight matrix is based on the gradient of the instantaneous error:

$$\frac{dw_{ij}(t)}{dt} = -|\eta_e(t)| \sum_{l=1}^{N} e_l(t)\, P_{lij}(t) \tag{3}$$

where, η_e is the learning rate, $e_l(t) = (\hat{y}_l(t) - y_l(t))$. $P_{lij} = \partial \hat{y}_l / \partial w_{ij}$ is the network sensitivity function updating according to equation (4):

$$\begin{aligned} \frac{1}{|\tau_e(t)|} \frac{dP_{lij}(t)}{dt} &= \tanh'\left(\sum_{m=1}^{N_e} w_{lm}(t) s_m(t) + x_l(t)\right) \\ &\times \left(\delta_i^l \hat{y}_j(t) + \sum_{m=1}^{N_e} w_{lm}(t) P_{mij}(t)\right) - P_{lij}(t) \end{aligned} \tag{4}$$

δ_i^l is the Kronecker symbol and tanh' is the derivative of tanh [6, 13].

2.2 Autonomous Adaptation of parameters η_e and τ_e

Based on the same method as the one used for the weights matrix adaptation, an algorithm is defined in order to adapt the parameters η_e and τ_e:

$$\frac{d\eta_e(t)}{dt} = -\sum_{l=1}^{N} (\hat{y}_l(t) - y_l(t)) V_l^{\eta}(t) \tag{5}$$

$$\frac{d\tau_e(t)}{dt} = -|\eta(t)| \sum_{l=1}^{N} (\hat{y}_l(t) - y_l(t)) V_l^{\tau}(t) \tag{6}$$

with, $V_l^{\eta_e} = \partial \hat{y}_l / \partial \eta_e$ and $V_l^{\tau_e} = \partial \hat{y}_l / \partial \tau_e$. $V_l^{\eta_e}$ and $V_l^{\tau_e}$ are considered as small perturbations added to the lth neuron state consequently to small variations $\partial \eta_e$ and $\partial \tau_e$ of respectively η_e and τ_e [13]. For simplification, we set $V_l^{\eta_e} = V_l^{\tau_e} = V_l$. Using the dynamic behavior of the network given by equation (1), V_l is computed as:

$$\begin{aligned} \frac{1}{|\tau_e(t)|} \frac{dV_l(t)}{dt} &= \tanh'\left(\sum_{m=1}^{N_e} w_{lm}(t) s_m(t) + x_l(t)\right) \\ &\times \left(\sum_{m=1}^{N_e} w_{lm}(t) V_m(t)\right) - V_l(t) + \frac{\varepsilon_e}{|\tau_e(t)|} \end{aligned} \tag{7}$$

where, $\varepsilon_e = \left.\frac{dV_l}{dt}\right|_{t=0} > 0$ ensure the starting and the autonomous evolution of the algorithm starting from zero initial conditions [26].

2.3 Simulation results for switching system emulation

In this subsection, a witching system is investigated to illustrate performance of our algorithm-based emulator. The process dynamics is supposed to switch suddenly from the linear system (S_1) given by (8) to the nonlinear one (S_2) given by (9) at time $t_c = 600$s. The term ε_e is chosen equal to 1 to ensure the starting of the system with zero initial conditions of parameters. The sampling time is chosen equal to 0.1s.

$$y(k) = 1.918y(k-1) - 0.945y(k-2) + 0.11x(k-1) - 0.0838x(k-2) \tag{8}$$

$$y(k) = 0.9y(k-1) - 0.001y(k-2)^2 + x(k-1) + \sin(x(k-2)) \tag{9}$$

Obtained results, given in Fig. 2, 3 and 4, show that each variation of the input signal leads to transient behavior while neural emulator adapts itself to the variation of the plant output.

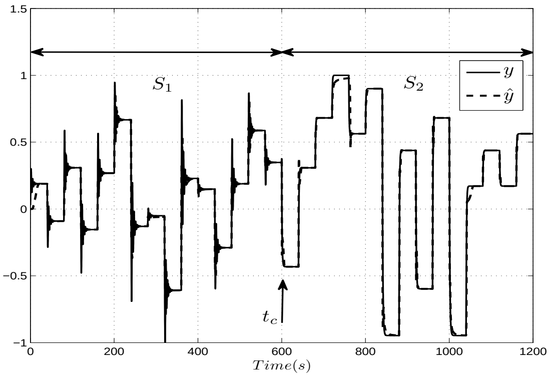

Fig. 2. Actual (full line) and estimated outputs (dotted line).

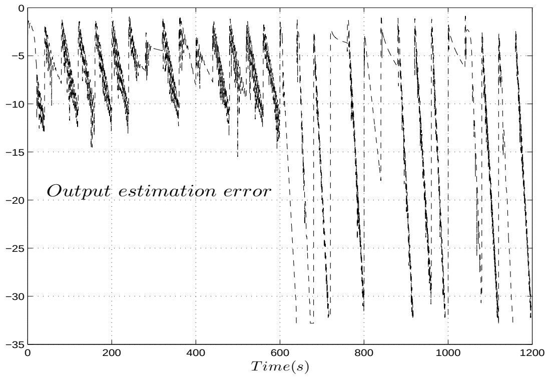

Fig. 3. Output estimation error (in logarithmic scale).

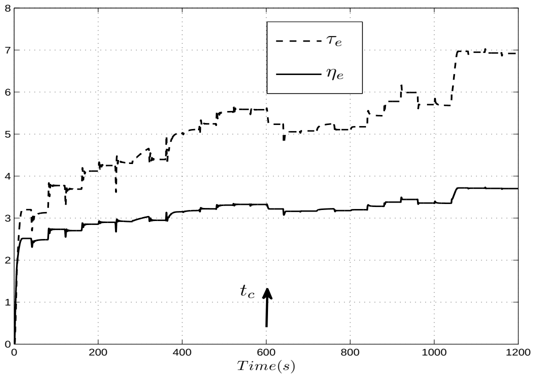

Fig. 4. Evolution of Emulator parameters: τ_e (dotted line) and η_e (full line).

We note also that a new adaptation of neural emulator occur at time t_c. The output estimation error plotted in logarithmic scale indicate the satisfied emulation performances as shown in Fig. 3. Figure 4 illustrates well adaptation of emulator parameters. Then, the efficiency of the proposed approach to emulate unknown non linear processes as well as linear ones is proved.

2.4 Application to chemical reactors

Process description In this application, we consider a chemical process (Fig. 5), used to esterify olive oil. The reaction carried out in this reactor is a chemical esterification of the crude acid of olive oil by an alcohol, such as butane, in order to extract an ester with high quality. The produced ester is widely used for the manufacture of cosmetic products [15, 1]. The esterification reaction is given as follow:

$$\text{Acid} + \text{Alcohol} \longleftrightarrow \text{Ester} + \text{Water}$$

The ester's proportion can be increased by vaporization of water. We set that the acid and the ester ebullition temperatures are approximately 300 °C. The butane is characterized by an ebullition temperature of 118 °C. Consequently, the reactor's temperature is over 100°C to get rid of water only [16]. The process temperature is regulated by means of a fluid circulating through the reactor jacket. This fluid is heated by three resistors whose electric power can be varied from 0 to 2500 Watts and located in the heat exchanger (Fig. 5). It is also cooled in a tubular cooler whose cooling rate is changed by varying the external water [15].

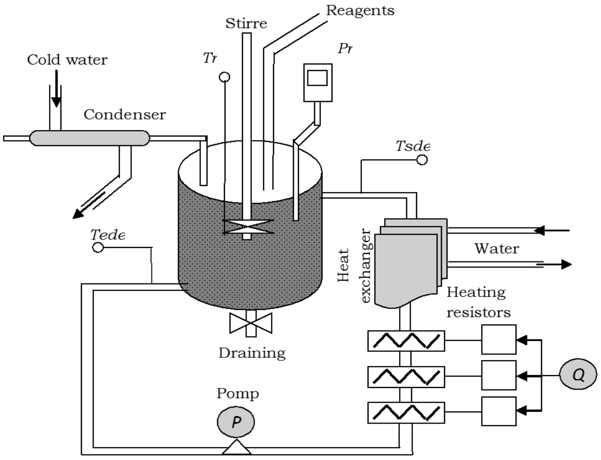

Fig. 5. Synoptic scheme of the reactor.

The variables $Tede$ and $Tsde$ are respectively input and output temperature of double envelope, T_r is the reactor's temperature and Q is the heating power. Then, temperature must follow a specific trajectory. This trajectory (Fig. 6) consists on three stages:

- Heating stage: the reactor's temperature T_r is increased to 105°C.
- Reaction stage: the reactor's temperature T_r is maintained constant during the reaction (when no more water is dripping out of the condenser).
- Cooling stage: the reactor's temperature is decreased.

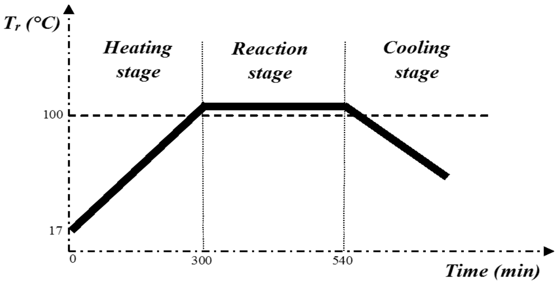

Fig. 6. Reactor's temperature profile.

Reactor emulation Experimental studies show that the process is non linear [17]. The static characteristic of the system (Fig.7) illustrates that it can be

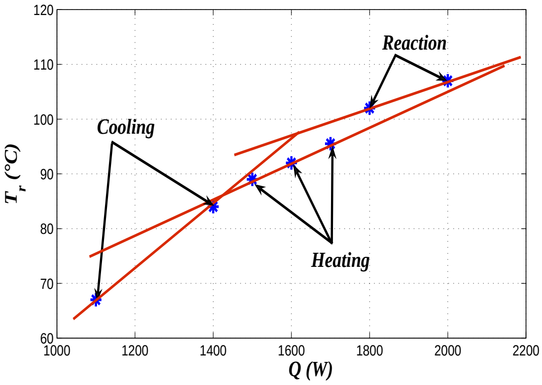

Fig. 7. Static characteristic of the reactor.

considered as piecewise linear in every operating phase: the heating phase, the reaction phase and the cooling phase.

The reactor is considered as a single input single output system. The output is the reactor's temperature T_r. The input is the heating power Q. Then, a two-neuron neural network is capable to emulate the chemical reactor input-output mapping. This number of neurons depends only on the input and the output numbers. The proposed network does not learn the plant dynamics, but only adapts its parameters. The process sampling time ΔT is chosen equal to $3min$ according to the step response of the system. The starting term ε_e is chosen equal to 100 so that it ensure the starting and the autonomous evolution of the algorithm initially zero conditions.
A pseudo-random binary input signal (PRBS) is applied to the real system (Fig. 8). The amplitudes of the signal are chosen so that they focus on the three reaction stages [15, 16]. Results of neural emulation of the reactor temperature are given by Fig. 9. The neural emulator provides a satisfactory estimation of the process output. We notice that in the heating and cooling stages, the estimated reactor's temperature is fluctuated consequently to the temperature variation (depending on the magnitude of input signal). While, during the reaction stage (when the temperature is maintained stable), the neural emulation is efficient and capable to estimate the reactor output.

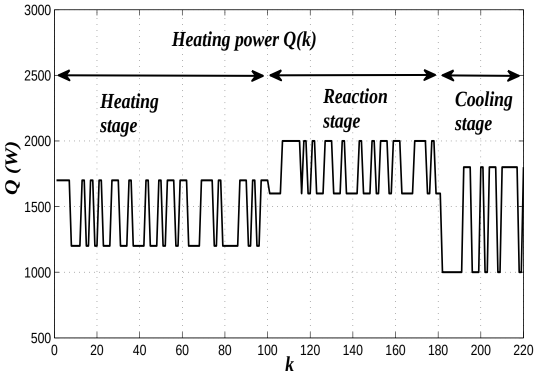

Fig. 8. Input signal.

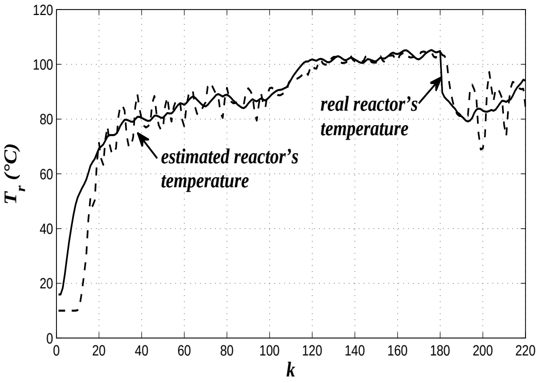

Fig. 9. Real (full line) and estimated (dotted line) reactor's temperatures.

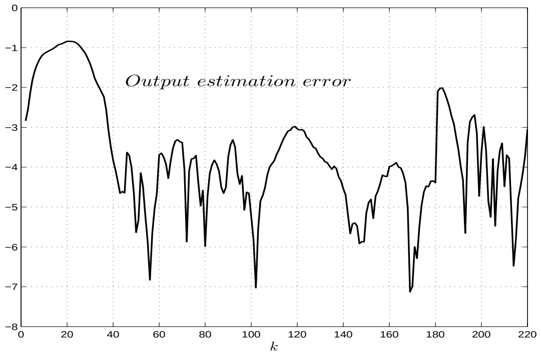

Fig. 10. Square output estimation error (in logarithmic scale).

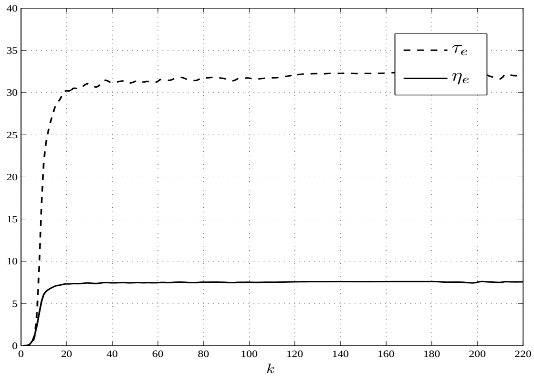

Fig. 11. Evolution of emulator parameters η_e (full line) and τ_e (dotted line).

Figure 10 illustrates the square output estimation error plotted in logarithmic scale. This error prove the efficiency of the neural emulation approach. The evolution of neural emulator parameters η_e and τ_e is given by Fig. 11. This figure illustrates a satisfactory adaptation of the parameters τ_e, η_e. The obtained results prove the good performance of the neural emulator method adapted for the real time emulation of chemical reactor.

3 Neural controller

3.1 Autonomous adaptation algorithm

Neural controller is a fully connected recurrent network similar to NE. This structure is formed by $2N$ neurons. The NC inputs are the N plant desired outputs y_{c_i} and the N output error functions $(y_{c_i} - y_i)$. The N control inputs u_i are the NC outputs. This NC structure is suitable for the control of square MIMO systems [27, 2].

According to the dynamic activation of neurons, the control signals are calculated, in continuous time, by the following equation:

$$\frac{1}{|\tau_c(t)|} \frac{d\left(o_i(t)\right)}{dt} + o_i(t) = \tanh\left(\sum_{j=1}^{N_c} \phi_{ij}(t)\, o_j(t) + z_i(t)\right) \tag{10}$$

where, $o_i(t)$ is the i^{th} neuron state. $\phi_{ij}(t)$ and $1/|\tau_c(t)|$ are respectively the NC weights from j^{th} neuron to i^{th} neuron and the NC adaptive time parameter. $z_i(t) = y_{c_i}(t) - y_i(t)$ if $i \in OUT$. In addition, $z_i(t) = y_{c_i}(t)$ if $i \in OUT$. $u_i(t) = o_i(t)$, if $i \in IN$, represents the i^{th} input signal.

Let us consider the instantaneous square error between desired output and measured output defined as in:

$$E_c(t) = \frac{1}{2}\sum_{l=1}^{N} (y_{c_l}(t) - y_l(t))^2 \tag{11}$$

The minimization of $E_c(t)$ with respect to the parameters is used for NC weights adaptation:

$$\frac{d\phi_{ij}(t)}{dt} = |\eta_c(t)| \sum_{l=1}^{N} (y_{c_l}(t) - y_l(t)) \frac{\partial y_l(t)}{\partial \phi_{ij}(t)} \tag{12}$$

where, $|\eta_c(t)|$ is the NC adapting rate. $\partial y_l / \partial \phi_{ij}$ is approximated by $\partial \hat{y}_l / \partial \phi_{ij}$ which is given by:

$$\frac{\partial \hat{y}_l}{\partial \phi_{ij}} = \sum_{k \in IN} \frac{\partial \hat{y}_l}{\partial u_k} \frac{\partial u_k}{\partial \phi_{ij}} = \sum_{k \in IN} J_{lk}\, Q_{kij} \tag{13}$$

According to the NC dynamic behavior given by equation (10), the sensitivity functions Q_{kij} are calculated as follows:

$$\begin{aligned} \frac{1}{|\tau_c(t)|}\frac{dQ_{kij}(t)}{dt} &= \tanh'\left(\sum_{h=1}^{N_c} \phi_{kh}(t)\, o_h(t) + z_k(t)\right) \\ \times \left(\delta_i^k\, o_j(t) + \sum_{h=1}^{N_c} \phi_{kh}(t)\, Q_{hij}(t) + \frac{\partial z_k(t)}{\partial \phi_{ij}(t)}\right) &- Q_{kij}(t) \end{aligned} \tag{14}$$

For $k \in IN$, $z_k(t) = y_{c_k}(t)$. So, Q_{kij} are given by

$$\begin{aligned} \frac{1}{|\tau_c(t)|}\frac{dQ_{kij}(t)}{dt} &= \tanh'\left(\sum_{h=1}^{N_c} \phi_{kh}(t)\, o_h(t) + z_k(t)\right) \\ \times \left(\delta_i^k\, o_j(t) + \sum_{h=1}^{N_c} \phi_{kh}(t)\, Q_{hij}(t)\right) &- Q_{kij}(t) \end{aligned} \tag{15}$$

For $k \in OUT$, $z_k(t) = y_{c_k}(t) - y_k(t)$. Consequently, Q_{kij} are computed as in

$$\begin{aligned} \frac{1}{|\tau_c(t)|}\frac{dQ_{kij}(t)}{dt} &= \tanh'\left(\sum_{h=1}^{N_c} \phi_{kh}(t)\, o_h(t) + z_k(t)\right) \\ \times \left(\delta_i^k\, o_j(t) + \sum_{h=1}^{N_c} \phi_{kh}(t)\, Q_{hij}(t) - \sum_{\mu=1}^{N} J_{k\mu}\, Q_{\mu ij}\right) \\ -Q_{kij}(t) \end{aligned} \tag{16}$$

The sensitivity functions J_{lk} are computed as follows, using the NE dynamic behavior given by equation (2):

$$
\begin{aligned}
\frac{1}{|\tau_e(t)|}\frac{dJ_{lk}(t)}{dt} &= \tanh'\left(\sum_{h=1}^{N_e} w_{lh}(t)\, s_h(t) + x_l(t)\right) \\
&\times\left(\delta_k^l \;+ \sum_{h=1}^{N_e} w_{lh}(t)\, J_{hk}(t)\right) - J_{lk}(t)
\end{aligned}
\tag{17}
$$

3.2 Autonomous adaptation of NC parameters η_c and τ_c

Based on the same method used for η_e and τ_e, an algorithm is defined in order to adapt NC parameters η_c and τ_c according to the following equations:

$$
\frac{d\eta_c(t)}{dt} = \sum_{l=1}^{N}(y_{c_l}(t) - y_l(t))\, R_l^{\eta_c}(t) \tag{18}
$$

$$
\frac{d\tau_c(t)}{dt} = |\eta_c(t)| \sum_{l=1}^{N}(y_{c_l}(t) - y_l(t))\, R_l^{\tau_c}(t) \tag{19}
$$

$R_l^{\eta_c}$ and $R_l^{\tau_c}$ are considered as small perturbations added to the l^{th} neuron state consequently to small variations $\partial\eta_c$ and $\partial\tau_c$ of respectively η_c and τ_c . We also consider $R_l^{\eta_c}$=$R_l^{\tau_c}$=R_l for simplification.

Functions R_l are then computed using the NE dynamic behavior (2) as follows:

$$
\begin{aligned}
\frac{1}{|\tau_e(t)|}\frac{dR_l(t)}{dt} &= \tanh'\left(\sum_{m=1}^{N_e} w_{lm}(t)\; s_m(t) + x_l(t)\right) \\
&\times\left(\sum_{m=1}^{N_e} w_{lm}(t)\, R_m(t)\right) - R_l(t) + \frac{\varepsilon_c}{|\tau_e(t)|}
\end{aligned}
\tag{20}
$$

where, $\varepsilon_c = \left.\frac{dR_l}{dt}\right|_{t=0} > 0$ guarantee the algorithm starting and accelerates the controller adaptation from zero initial conditions.

3.3 Simulation results for nonlinear system control

The objective of this simulation is to show the effectiveness of the proposed neural control structure with decoupled adaptive rates to control nonlinear systems. For this aim, a non linear system given by (21), is considered [6].

$$
y(k) = 0.4u(k-1) + 0.3y(k-1) + \frac{u(k-1)y(k-2)(u(k-1)+2.5)}{1+u(k-1)^2+6.25y(k-2)^2} \tag{21}
$$

Both tracking and regulation problems are studied according to this simulation example. The proposed simulation include a tracking phase (0-120s) and also a regulation phase (120-400s). During the regulation phase, at time t=250s, a disturbance (during 10s) in the form of a step input, with a magnitude of 10 % of the control output, was injected to the system. The terms ε_e and ε_c are chosen equal to 10 and 30 respectively in order to ensure the starting of the system with zero initial conditions of parameters. The sampling time is chosen equal to 0.1s.

Fig. 12 illustrates results obtained for control using the same adaptive rates for emulator and controller compared to control with decoupled adaptive rates. Independent adaptation of networks parameters ensure better and faster convergence of controller to the desired trajectory. Then, we notice that neural structure with decoupled rates ensure better perturbation rejection with comparison to the one with the same values of rates for both emulator and controller. Neural networks parameters are adapted independently, their adaptation is illustrated with Fig. 13 for emulator and Fig. 14 for controller. Values of controller parameters are not similar to emulator ones. Controller with decoupled rates is more performing in terms of convergence. Mean square error has been calculated to show the performance of the proposed method. For neural control with decoupled rates, mean square error is equal to 5.58 10^{-7}. Therefore, it is equal to 5.14 10^{-4} for control with same rates. Satisfactory results obtained confirm the good efficiency of this structure for tracking and regulation problems.

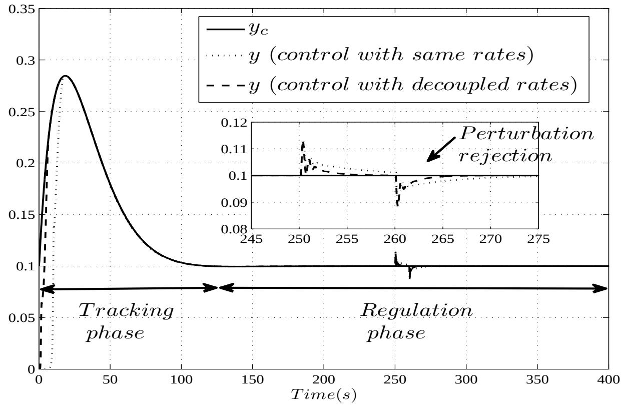

Fig. 12. Real and desired outputs.

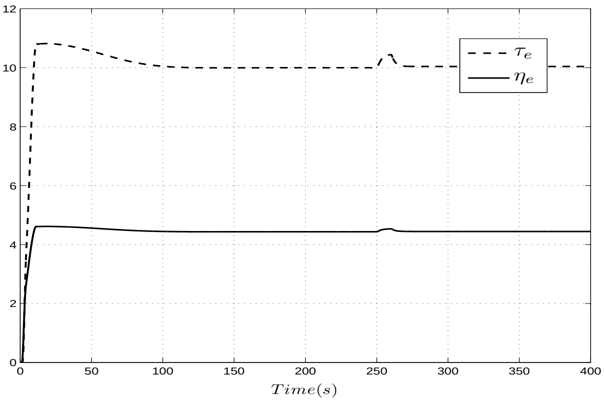

Fig. 13. Evolution of Emulator parameters: τ_e (dotted line) and η_e (full line).

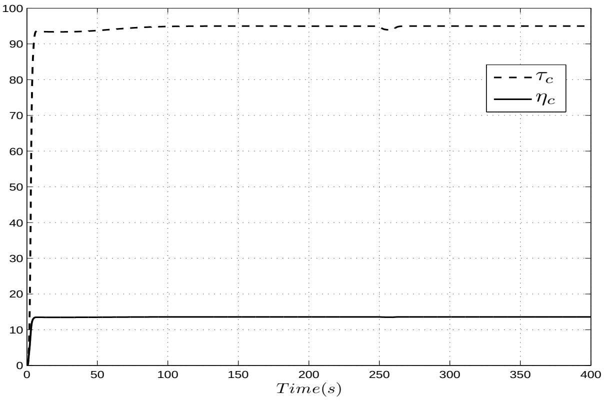

Fig. 14. Evolution of controller parameters: τ_c (dotted line) and η_c (full line).

In conclusion, real time adaptation of networks with decoupled rates improve controller performances. The obtained results prove the performance of the considered approach for control of nonlinear systems in terms of regulation and tracking with better disturbance rejection.

3.4 Application to TECP

Process description The Tennessee Eastman challenge process (Fig. 15), published by the Tennessee Eastman Company [7], is a real time simulation of a chemical process that has been used for academic research [11, 12]. The process

consists of five major units: an exothermic two-phase reactor, condenser, compressor, separator, and stripper. The Tennessee Eastman process involves the production of two products and an undesired one from four reactants, A, C, D, and E. In addition, there are two side reactions that occur.
The Tennessee Eastman (TE) problem is a large-scale, continuous, nonlinear process with 50 states, 41 measurements and 12 manipulated variables [25]. In addition to the process description, the problem statement defines process constraints, 20 types of process disturbances, and 7 operating modes corresponding to different production rates and G/H mass ratios. A complete description of these variables and disturbances is presented in [7, 25].

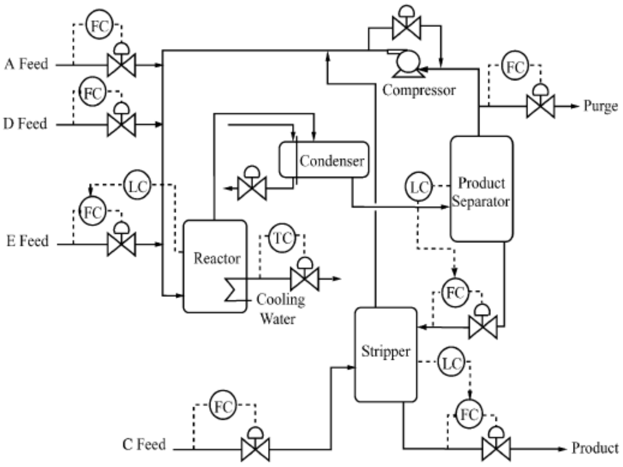

Fig. 15. Tennessee Eastman Challenge Process

Fig. 15 represents the Tennessee Eastman (TE) process taken from Downs and Vogel. The gaseous products from the reactor pass through a condenser and from there to the vapor/liquid separator. The noncondensed components recycle back through the compressor to the reactor feed, and the condensed components go to the product stripping column to remove the remaining reactants [24].

Process control and disturbance rejection In this work, we are limited to four measured variables and four controlled ones [25]. The measured variables are the reactor temperature (RT), the reactor pressure (RP), the separator level (SepL) and the stripper one (StrL). Then, the manipulated variables are the purge valve (Purge Valve), the separator valve (Sep Valve), the reactor cooling water valve (RCW Valve) and the condenser cooling water valve (CCW Valve). The sampling period for measurements and for the purge and reactor cooling water valves is 72 seconds and the one for the stripper and condenser cooling water valves is 180 seconds. The terms ε_e and ε_c are chosen equal to 10 to ensure the process starting with a minimum normalized mean square error (NMSE) and in the same time these values ensure the best disturbance rejection. Neural emulator and neural controller networks, both of them formed with eight neurons, are used for the TECP control in the base case. From zero initial conditions of the all parameters of these networks, process control is ensured with real time adaptation of these networks.

To show the efficiency of the proposed approach, many simulations in presence of different perturbations are carried out [2]. In this paper, wi will present results obtained from simulation in presence of perturbation IDV(4) (Reactor cooling water temperature) during 5 hours from t=5h to t=10h. The evolutions of the reactor temperature, pressure and purge valve are illustrated respectively by Fig. 16, 17 and 18. We note that control with decoupled adaptive rates accelerates controller adaptation. Independent adapting of networks parameters ensure better regulation of temperature and pressure in reactor, and the levels in separator and stripper particularly during the presence of perturbation and after dynamic changes. This structure with decoupled rates guarantees also disturbance rejection. Then, networks parameters adapted independently are shown by Fig. 19 and 20. We note that adaptation with decoupled rates ensure a perfect plant stabilization with constant parameters. Obtained results prove the efficiency of this approach.

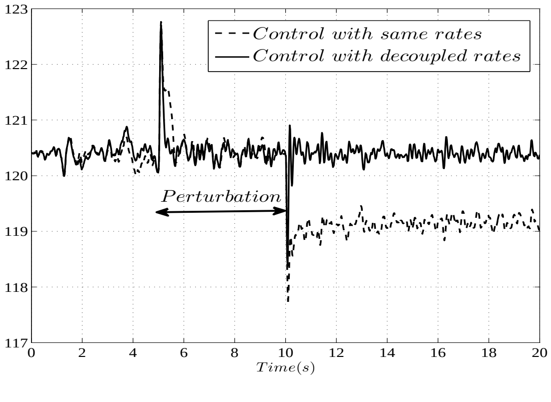

Fig. 16. Evolution of reactor temperature (RT).

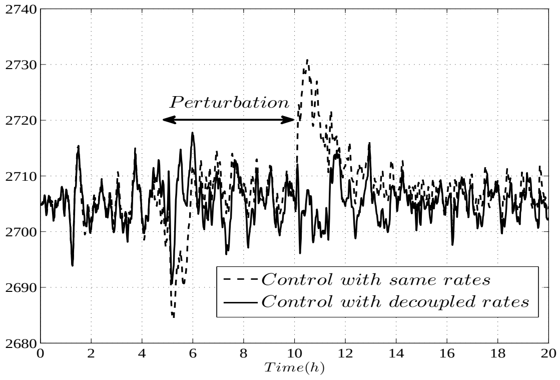

Fig. 17. Evolution of reactor pressure (RP).

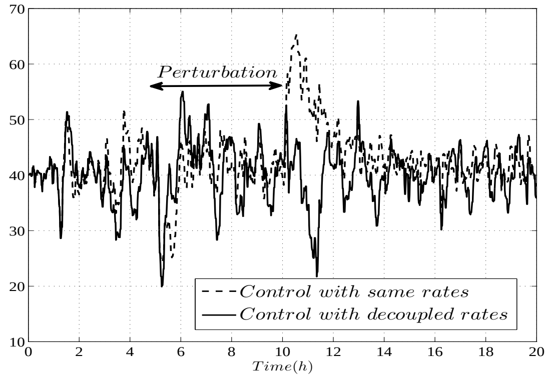

Fig. 18. Evolution of Purge valve.

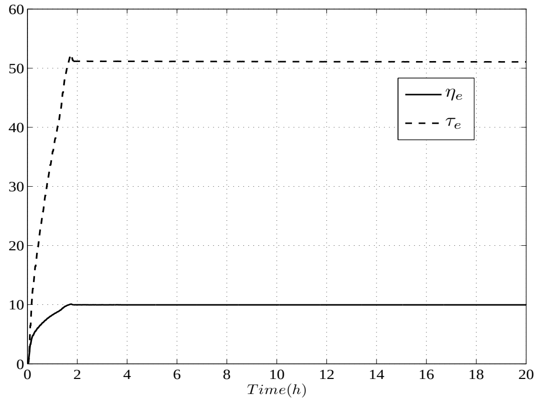

Fig. 19. Evolution of emulator parameters: τ_e (dotted line) and η_e (full line).

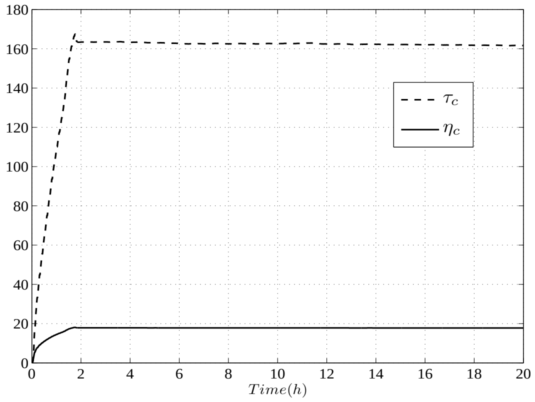

Fig. 20. Evolution of controller parameters: τ_c (dotted line) and η_c (full line).

The global performance of this method is illustrated with the calculation of the $NMSE$ (Normalized Mean Square Error) criteria given by (22):

$$NMSE = \frac{\sum_{t=1}^{T_s} \left(y_c(t) - y(t)\right)^2}{\sum_{t=1}^{T_s} \left(y_c(t)\right)^2} \tag{22}$$

where, T_s is the simulation time. The following table illustrates NMSE calculated for control with same rates and with decoupled rates in presence of perturbation. These NMSE are measured during the simulation (20h) and only during perturbation (5h-10h). We note that NMSE is more important for the control with same rates compared to the one with decoupled rates. Due to perturbation, controller does not regulate temperature in the reactor which affect process control performances. Results show that independent updating of neural emulator and controller parameters improve controller performances. A diminution of 22 % of the NMSE, calculated for the reactor temperature during perturbation which affect particularly RT, is noticed in comparison with control with decoupled rates. Also a diminution of 53 % of the global NMSE is observed.

NMSE (10^{-4})	Control with same rates		Control with decoupled rates	
	(20h)	(5h-10h)	(20h)	(5h-10h)
Reactor temperature	0.61	0.13	0.035	0.077
Reactor pressure	0.058	0.056	0.019	0.031
Separator level	20	13	6.78	9.06
Stripper level	5.042	5.25	4.97	5.38
Global variables	6.34	4.66	2.95	3.63

Control with decoupled rates provides efficient plant stabilization with well adaptation of networks parameters. To show efficiency of our proposed approach to reject various perturbations, many simulations with other disturbances are considered. The table below illustrates results obtained for control structure with decoupled rates in presence of perturbations IDV(3) (D feed temperature) and IDV(8) (A, B and C feed composition).

Simulation (40 hours)		NMSE (10^{-4})	NMSE (10^{-4})
Perturbation (5h-20h)		(40h)	(5h-20h)
Perturbation (3)	RT	0.009	0.0089
	RP	0.014	0.012
	SepL	7.92	7.46
	StrL	4.95	4.97
	Global	3.22	3.11
Perturbation (8)	RT	0.018	0.024
	RP	0.051	0.097
	SepL	29	38
	StrL	5.09	5.27
	Global	8.61	11

In conclusion, control with decoupled adaptive rates increases the number of freedom degrees of the adaptive algorithm. Particularly, real time adaptation of networks improve controller performances and ensure better disturbances rejection of the multivariable non linear chemical plant TECP with only four measurements and four actuators.

4 Conclusion

In this paper, an indirect control structure based on neural emulator is proposed for nonlinear systems. This structure is inspired from RTRL algorithm. Neural emulator and neural controller are adapted independently using an online-adaptation algorithm. Efficiency of the proposed structure with decoupled adaptive rates has been established for tracking and regulation problems according to nonlinear systems simulations. The neural control structure proposed has been validated on chemical processes. Satisfactory results obtained prove the potential of the method considered for real plant wide processes.

In our future works, we will consider economic and environmental objectives in control of complex and large scale square systems in order to minimize variability of product rate and product quality during perturbations. Extension to non square multivariables systems will be studied. Supervision of starting terms (ε_e and ε_c) will be also considered.

References

1. Atig A., Druaux F., Lefebvre D., Abderrahim K. and Ben Abdennour R., Neural Emulation Applied to Chemical Reactors. *" 7th IEEE International Multi-Conference on Systems, Signals and Devices (SSD'10)"*. Amman, Jordan, 2010.
2. Atig A., Druaux F., Lefebvre D., Abderrahim K. and Ben Abdennour R., Neural Network Control for Large Scale Systems with Faults and Perturbations. *" 10th IEEE International Conference on Control and Fault-Tolerant Systems (Sys-Tol'10)"*. Nice, France, 2010.
3. Ben Abdennour R., Borne P., Ksouri M. and M'sahli F., Identification et commande numérique des procédés industriels. *"Technip"*. Paris, France, 2001.
4. Budik D. and Elhanany I., TRTRL: a localized resource-efficient learning algorithm for recurrent neural networks. *" Proceedings of IEEE Midwest Symp. Circuits Syst"*. Puerto Rico, Aug, pp.371-374, 2006.
5. Chow T.W.S. and Fang Y., A Recurrent neural-network based real time learning control strategy applying to nonlinear systems with unknown dynamics. *"IEEE Transactions on Industrial Electronics"*. vol.45, no.1, pp.151-161, 1998.
6. Druaux F., Leclercq E. and Lefebvre D., Adaptive neural network control for uncertain or unknown non linear systems. *"Proceedings of IEEE MMAR"*. Poland, pp.1309-1314, 2004.
7. Downs J.J. and Vogel E.F.A, Plant-wide industrial process control problem. *"Comput. Chem. Eng."*. vol.17, no.3, pp. 245-255, 1993.
8. Ge S.S., Wang C. and Tan Y.H., Adaptive control of partially known nonlinear multivariable systems using neural networks. *"Proceedings of IEEE International Symposium on Intelligent Control"*. Mexico, pp.292-297, 2001.
9. Ge S.S., Zhang J. and Lee T.H., Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time. *"IEEE Trans. Syst., Man, Cybern. B"*. vol. 34, no. 4, pp. 1630-1645, 2004.
10. Jeon G.J. and Lee I., Neural network indirect adaptive control with fast learning algorithm. *"Neurocomputing"*. vol.2, pp.185-199, 1996.
11. Juricek B., Seborg D. and Larimore W., Identification of the Tennessee Eastman Challenge Process Using Subspace Methods. *" Control Engineering Practice"*. vol. 9, pp. 1337-1351, 2001.
12. Juricek B., Seborg D. and Larimore W., Process Control Applications of Subspace and Regression-based Identification and Monitoring Methods. *" Proceeding of American Control Conference"*. Portland, OR, USA, 2005.
13. Leclercq E., Druaux F., Lefebvre D. and Zerkaoui S., Autonomous learning algorithm for fully connected recurrent networks. *"Neurocomputing"*. vol.63, pp.25-44, 2005.
14. Lefebvre D., Zerkaoui S., Druaux F. and Leclercq E., Adaptive control with stability and robustness analysis for nonlinear plant wide systems by means of neural networks. *"In Progress in Nonlinear Analysis Research"*. chapter. 8, Nova Science Publishers, pp.187-212, 2008.
15. Messaoud A., Ltaief M. and Ben Abdennour R., Supervision based on partial predictors for a multimodal generalized predictive control: Experimental validation on a semi-batch reactor. *"International Journal of Modeling, Identification and Control"*. vol. 6, no. 4, pp. 333–340, 2009.

16. Mihoub M., Nouri A.S. and Ben Abdennour R., Real-time application of discrete second order sliding mode control to a chemical reactor. *" Control Engineering Practice"*. vol. 17, no. 9, pp. 1089–1095, 2009.
17. M'sahli F., Ben Abdennour R. and Ksouri M., Application of adaptive controllers for the temperature control of a semi batch reactor. *" International Journal of Computational Engineering Science"*. vol. 2, no. 2, pp. 287–307, 2001.
18. Narendra K.S. and Parthasarathy K., Identification and control of dynamical systems using neural networks. *"IEEE Transactions on Neural Networks"*. vol.1, no.1, pp.4-27, 1991.
19. Nelles O., Nonlinear system identification. *"Springer"*, 2001.
20. Tian L. and Collins C., A dynamic recurrent neural network based controller for a rigid flexible manipulator system. *"Mechatronics"*. vol.14, pp.3187-3202, 2004.
21. Wai R.J., Linn H.H. and Lin F.J., Hybrid controller using fuzzy neural networks for identification and control of induction servo motor drive. *"Neurocomputing"*. vol.35, pp.91-112, 2000.
22. Williams R.J. and Zipser D., A learning algorithm for continually running fully recurrent neural networks. *"Neural Computation"*. vol.1, pp.270-280, 1989.
23. Williams R.J. Adaptive state representation and estimation using recurrent connectionist networks. *"Neural Networks for Control"*. chapter, Mit Press, Cambridge, 1990.
24. Yeh T.M., Huang M.C. and Huang, C.T., Estimate of process compositions and plantwide control from multiple secondary measurements using artificial neural networks. *"Computers and Chemical Engineering"*. vol. 27, pp. 55-72, 2003.
25. Yan M. and Ricker N.L., Multi-Objective Control of the Tennessee Eastman Challenge Process. *" Proceeding of American Control Conference"*. Sealtle, Washington, 1995.
26. Zerkaoui S., Druaux F., Leclercq E. and Lefebvre D., Stable adaptive control with recurrent neural networks for square MIMO nonlinear systems. *"IFAC WC, International Federation of Automatic Control"*. Prague, Tcheque, 2005.
27. Zerkaoui S., Druaux F., Leclercq E. and Lefebvre D., Stable adaptive control with recurrent neural networks for square MIMO nonlinear systems. *"Engineering Applications of Artificial Intelligence"*. vol.12, no.4-5, pp.702-717, 2009.
28. Zerkaoui S., Druaux F., Leclercq E. and Lefebvre D., Indirect neural control for plant-wide systems: Application to the Tennessee Eastman Challenge Process. *" Computers and Chemical Engineering"*. vol. 34, pp. 232-243, 2010.
29. Zhang J., Batch to batch optimal control of a batch polymerization process based on stacked neural network models. *" Chemical Engineering Science"*. vol. 63, pp. 1273-1281, 2008.
30. Zhiong M., Wu H.R. and Palaniswami M., An adaptive tracking controller using neural network for a class of nonlinear systems. *"IEEE Transactions on Neural Networks"*. vol.9, no.5, pp.947-955, 2008.