Robust Neural Control Strategies for Discrete-Time Uncertain Nonlinear Systems

Imen Zaidi
Electrical Engineering Department
National School of Engineers, Sfax, Tunisia

Mohamed Chtourou
Electrical Engineering Department
National School of Engineers, Sfax, Tunisia

Mohamed Djemel
Electrical Engineering Department
National School of Engineers, Sfax, Tunisia

ABSTRACT

In this paper, three neural control strategies are addressed to a class of single input-single output (SISO) discrete-time nonlinear systems affected by parametric variations. According to the control scheme, in a first step, a direct neural model (DNM) is developed to emulate the behavior of the system, then an inverse neural model (INM) is synthesized using specialized learning technique and cascaded to the system as a controller. The sliding mode backpropagation algorithm (SM-BP), which presents in a previous study robustness and high speed learning, is adopted for the training of the neural models. However, in the presence of strong parametric variations, the synthesized (INM) shows limitations to present satisfactory tracking performances. In fact, in order to improve the control results, two neural control strategies such as hybrid control and neuro-sliding mode control are proposed in this work. A simulation example is treated to show the effectiveness of the proposed control strategies

Keywords

SISO Discrete-time uncertain nonlinear systems, neural modelling, sliding mode, backpropagation algorithm, INM control, hybrid control, neuro-sliding mode control.

1. INTRODUCTION

In practice, a large number of systems are strongly nonlinear and uncertain. Thus, in recent years, several studies dealing with modeling and control of uncertain nonlinear systems have been developed [1-2-3]. The first step of the control of an uncertain nonlinear system is to find a mathematical model able to reproduce the dynamic of this system with a required accuracy. However, conventional modeling methods have shown limitations to approximate correctly nonlinear systems affected by parametric uncertainties. In fact, the last decade has witnessed an ever increasing research in Non conventional modeling methods as fuzzy system [4-5] and neural networks [6] since they have been considered as positional solutions to overcome these difficulties of modeling owing to their universal approximation property. Topalov and kaynak presented in [7] a robust neural identification of robotic manipulators using learning algorithm based on sliding mode control technique. In [8], a problem of identification and control of uncertain nonlinear system was investigated based on fuzzy neural networks. Reference [9] proposed an adaptive robust control based on neural network approximation for a class of uncertain strict-feedback discrete-time nonlinear systems.

Moreover, control techniques using classical controllers present performance indexes degradation in case of uncertain nonlinear system. Indeed, it is important to develop effective robust control techniques [10]-[13] to guarantee stability, robustness and satisfactory tracking performances. In many studies, neural networks have been proven useful and effective for controlling a wide class of uncertain nonlinear system. In fact, Tellez et al, proposed in [14] a neural inverse optimal controller to achieve stabilization for discrete time uncertain nonlinear systems. In [15] a new approach for the calibration and the control of spark ignition engines using a combination of neural networks and sliding mode control technique was presented. Internal model control (IMC) is also considered as a robust control technique. Indeed, Alzohairy proposed in [16] a neural internal model control approach for the tracking of unknown nonaffine nonlinear discrete time systems subject to external disturbances.

This paper suggests robust neural control strategies for a class of single input-single output (SISO) discrete-time uncertain nonlinear systems. Indeed, in a first step, a direct neural model (DNM) is elaborated to reproduce the dynamic of the system, then, in a second step, an inverse neural model (INM) is developed. After satisfactory training, the synthesized (INM) is applied as a controller for the uncertain nonlinear system. The most popular algorithm for the training of feedforward neural network (FNN) is backpropagation (BP) algorithm [17]. However, this training method is not completely robust face to disturbance and parameter variations. The sliding mode backpropagation (SM-BP) [18]-[20] has been adopted in a specialized learning technique of the (INM). The training of both (DNM) and (INM) is accomplished through this algorithm which has been proven as the best configuration in previous study [19]. In order to improve the robustness and the tracking performance of the above neural control strategy, in the presence of strong parametric variations, two control strategies are proposed in this work such as: hybrid control and neuro-sliding mode control. Thus, a proportional-integral controller (PI) and a second order neuro-sliding mode corrective controller are added to operate with the synthesized (INM) for the case of the hybrid control and the neuro-sliding mode control respectively. The rest of paper is organized as follows. Section 2 introduces problem statement. Neural modeling is presented in section 3. In section 4, the neural control strategies are described in order to develop a robust neural controller for the discrete-time nonlinear affected by small and strong parametric uncertainties. A simulation example is treated in section 5 to show the effectiveness of the proposed control strategies. Finally, in section 6 conclusions are given.

2. PROBLEM STATEMENT

Consider the SISO uncertain nonlinear system described by the following equation:

$$y(k+1) = \left[y(k),..., y(k-n+1), u(k),..., u(k-m+1), p \right] \quad (1)$$

y and u are, respectively, the output and the input of the system, n is the order of $y(k)$, m is the order of $u(k)$, F is an unknown nonlinear function to be estimated by a neural network and p is an uncertain parameters vector. In this work, an additive uncertainty is considered.

$$p = p_0 + \Delta p \tag{2}$$

p_0 represents the nominal parameters and Δp is uncertain vector affecting the system.

3. NEURAL MODELLING: DNM

In order to reproduce the dynamic of system (1) a DNM is used. Indeed, it estimates the output of the system through old data of its inputs and outputs. Two approaches often discussed in the literature are the series parallel model and the parallel one [6]. In this work, we are interested in series parallel model.

The block diagram of the DNM training process is presented by Fig.1:

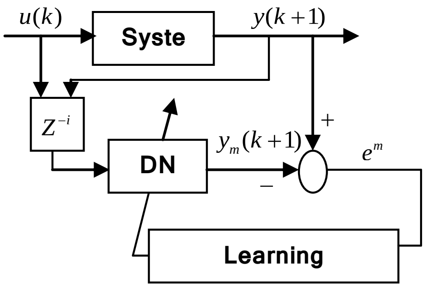

Fig. 1: DNM training process

The output of the DNM is given by the following equation:

$$y_m(k+1) = \hat{F}\left[y(k),....,y(k-n+1),u(k),.... ,u(k-m+1),p\right] \tag{3}$$

y_m and $\hat{F}$ denote respectively the output of the DNM and the estimate of F .

The weights of the DNM are adjusted to minimize the cost function defined by:

$$J = \frac{1}{2}\left[e^m\right]^2 \tag{4}$$

$e^m = y(k+1) - y_m(k+1)$ is the error between the output of the system $y(k+1)$ and the one of the DNM $y_m(k+1)$.

The learning algorithm adopted is this work is SM-BP algorithm which combines gradient descent method and sliding mode theory [18-19-20]. In fact, the SM-BP equations are presented by the following equations.

For the node j from the output layer, sliding surface is defined as [21]:

$$S_j^m(k) = X_{2j}^m(k) + C_0.X_{1j}^m(k) \qquad With\ j = 1 \tag{5}$$

The index m refers to DNM's parameters, j is the output node and $C_0 > 0$

$$X_{1j}^m(k) = \left[y(k) - y_m(k)\right].f^{m'}\left[V_j^m(k)\right] \tag{6}$$

$$X_{2j}^m(k) = X_{1j}^m(k) - X_{1j}^m(k-1) \tag{7}$$

Where $f^{m'}$ denotes the derivative of the output activation function, V_j^m is the global input of the output node j .

Let the sliding surface for each node of the hidden layer be such as:

$$S_{Hh}^m(k) = X_{2Hh}^m(k) + C_{0H}.X_{1Hh}^m(k) \qquad h = 1,2\ldots,N_c^m \tag{8}$$

Where h is the hidden node, $C_{0H} > 0$ and:

$$X_{1Hh}^m(k) = X_{1j}^m(k).W_{jh}^m(k).f_H^{m'}\left[R_h^m(k)\right] \tag{9}$$

$$X_{2Hh}^m(k) = X_{1Hh}^m(k) - X_{1Hh}^m(k-1) \tag{10}$$

W_{jh}^m represents the weight between the output node j and the hidden node h , $f_H^{m'}$ denotes the derivative of the hidden activation function, R_h^m is the global input of the hidden node h and N_c^m represents the number of neurons in the hidden layer.

Thus, the weights update equations based on SM-BP are given by:

$$\Delta W_{jh}^m(k) = \alpha_m.\mathrm{sgn}\left[S_j^m(k)\right].\left|X_{1j}^m(k)\right|.Y_{Hh}^m(k) \tag{11}$$

$$\Delta Z_{hi}^m(k) = \beta_m.\mathrm{sgn}\left[S_{Hh}^m(k)\right].\left|X_{1Hh}^m(k)\right|.T_i^m(k) \tag{12}$$

With: $i = 1,..,(n+m+1)$

Where Y_{Hh}^m is the output of the hidden node h , T_i^m is the input of the input node i , Z_{hi}^m represents the weight between the hidden node h and the input node i , $\alpha_m > 0$ and $\beta_m > 0$.

According to Utkin [21], the condition for existence of sliding mode and system stability is defined by the following equation:

$$S\frac{ds}{dt} < 0 \tag{13}$$

For discrete time, Sarpturk et al. [22] defined the equation $|S(k)| < |S(k-1)|$ instead of equation (13) as the necessary and sufficient condition to guarantee the sliding manifold. The computing of the limits of α_m and β_m is presented in [18-19].

4. NEURAL CONTROL

In this section, the design of the neural controller for uncertain nonlinear system through different control strategies is presented.

4.1 Inverse Neural Model Controller

The INM is trained to provide a control action that allows the behavior of the uncertain system to be as close as possible to the desired one. The specialized learning technique shown in Fig.2 is considered in this work for training the INM [23]. Thus, based on the DNM presented in section 3, which gives good representation of the system $y_m \approx y$ after satisfactory training, the INM is trained.

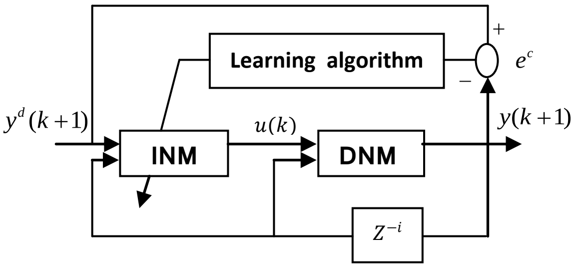

Fig. 2 : Specialized method for INM training

The cost function to be minimized in the training step is expressed as follows:

$$J_c = \frac{1}{2}\left[e_c\right]^2 \tag{14}$$

$e^c = y(k+1) - y^d(k+1)$ is the error between the output of the DNM $y(k+1)$ and the desired one $y^d(k+1)$.

Based on the SM-BP algorithm, updating rules for adjusting the weights of the INM are expressed by the following equations:

Let's the sliding surface for the node j from the output layer be defined as:

$$S_j^c(k) = X_{2j}^c(k) + C_1.X_{1j}^c(k) \tag{15}$$

Where j is the output node, $C_1 > 0$

$$X_{1j}^c(k) = [y(k+1) - y^d(k+1)].\gamma_h^m(k).f^{c'}[V_j^c(k)] \tag{16}$$

With
$$\gamma_h^m(k) = \sum_{h=1}^{N_c^m-1} f^{m'}\left[V_j^m(k)\right].W_{jh}^m(k).f_{Hh}^{m'}\left[R_h^m(k)\right].Z_{h1}^m(k)$$

$$X_{2j}^c(k) = X_{1j}^c(k) - X_{1j}^c(k-1) \tag{17}$$

The index c refers to INM's parameters, $f^{c'}$ denotes the derivative of the output activation function and V_j^c is the global input of the output node j.

For the node h from the hidden layer, the sliding surface is expressed as follows:

$$S_{Hh}^c(k) = X_{2Hh}^c(k) + C_{1H}.X_{1Hh}^c(k) \tag{18}$$

Where h is the hidden node, $C_{1H} > 0$

$$X_{1Hh}^c(k) = X_{1j}^c(k).W_{jh}^c(k).f_H^{c'}\left[R_h^c(k)\right] \tag{19}$$

$$X_{2Hh}^c(k) = X_{1Hh}^c(k) - X_{1Hh}^c(k-1) \tag{20}$$

Where W_{jh}^c is the weight between the output node j and the hidden node h of the INM, $f_H^{c'}$ is the derivative of the hidden activation function and R_h^c denotes the global input of the hidden node h.

Thus, the weights update equations of the INM based on the SM-BP algorithm are given by equations (21) and (22):

$$\Delta W_{jh}^c(k) = \alpha_c.\text{sgn}\left[S_j^c(k)\right].\left|X_{1j}^c(k)\right|.Y_{Hh}^c(k) \tag{21}$$

$$\Delta Z_{hi}^c(k) = \beta_c.\text{sgn}\left[S_{Hh}^c(k)\right].\left|X_{1Hh}^c(k)\right|.T_i^c(k) \tag{22}$$

Z_{hi}^c is the INM's weight between the hidden node h and the input node i, Y_{Hh}^c represents the output of the hidden node h T_i^c represents the input of the input node i of the INM, $\alpha_c > 0$ and $\beta_c > 0$ further information of the limits for the gain α_c and β_c can be found in reference [19].

After satisfactory training, the synthesized INM is simply cascaded with the plant as a neural controller of the uncertain nonlinear system [23] as given in Fig.3. Clearly, this approach relies heavily on the fidelity of the inverse neural model used as controller. For general purpose use serious questions arise regarding the robustness of the INM controller. This lack of robustness can be attributed primarily to the absence of feedback. This problem can be overcome to some extent by using on-line learning: the parameters of the inverse model can be adjusted on-line [24]. In this work, in order to improve the performance of the developed INM controller, in presence of strong parametric variations, other control strategies such that: hybrid control and neuro-sliding mode control are adopted.

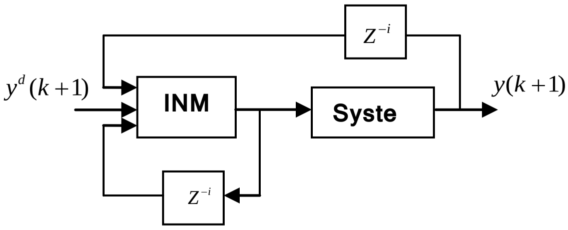

Fig. 3 : INM control structure

4.2 Hybrid Control

This approach has been proposed in [25-26]. It consists on operating simultaneously a conventional controller and a connectionist model to improve the control performance. In fact, the INM developed in section 4.1, trained using SM-BP, is used with a proportional-integral controller PI to generate a control action given as:

$$u(k) = u_{INM}(k) + u_{PI}(k) \tag{23}$$

With $u_{INM}(k)$ and $u_{PI}(k)$ represent the output of the INM controller and the output of the PI controller respectively.

$$u_{PI}(k) = u_{PI}(k-1) + k_p\, e(k) + k_i\, e(k-1) \tag{24}$$

k_p and k_i are proportional and integral gains, respectively.

$e(k)$ is the tracking error defined as:

$$e(k) = y^d(k) - y(k) \tag{25}$$

With $y^d(k)$ represents the desired output and $y(k)$ is the actual system output.

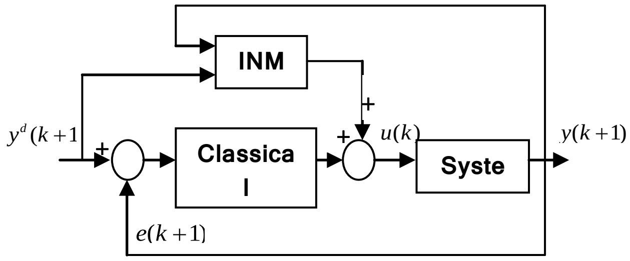

Fig. 4 : Hybrid control structure

4.3 Neuro-sliding mode control

Sliding-mode has been widely used to control uncertain nonlinear systems. In fact many studies have been proposed towards finding a controller that guarantees robustness and satisfactory tracking performances [27-28-29].

In general, the sliding mode control law based on Lyapunov stability theory is given by:

$$u(k) = u_{eq}(k) + u_c(k) \tag{26}$$

Where $u_{eq}(k)$ is the equivalent control law and $u_c(k)$ is the corrective term added to ensure robustness.

The classical SMC suffers mainly from two disadvantages. The first one is the high frequency oscillations of the controller output, termed “chattering”. The second is that a complete knowledge of the plant dynamics is needed in the computation of the equivalent control [30]. In the literature, many works adopt the neuro-sliding mode control as a structure to solve these problems [31-32-33]. In fact, two parallel neural networks are used to realize the equivalent control and the corrective control as in Fig.5.

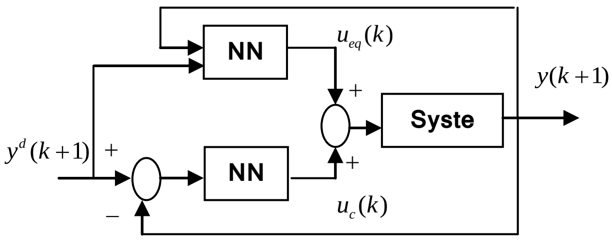

Fig. 5 : Neuro-sliding mode control structure

NN1 and NN2 are two neural networks used, respectively, to estimate the equivalent control and to generate the corrective control to estimate the chattering effects. The sum of $u_{eq}(k)$ and $u_c(k)$ forms the control signal to be applied to the controlled system.

In this work, the adopted neuro-sliding control structure is shown in Fig.6. The INM as presented in section 4.1 is used with a second order sliding mode corrective control to generate the control signal to be applied to the uncertain system.

The control action is computed as follows:

$$u(k) = u_{INM}(k) + u_{C_2}(k) \tag{27}$$

Where $u_{INM}(k)$ is the control action provided by the INM and $u_{C_2}(k)$ represents the second order sliding mode corrective control.

For the system defined by equation (1), the following sliding surface is selected:

$$S(k) = \Delta e(k) + \lambda e(k) \tag{28}$$

$$e(k) = y^d(k) - y(k)$$
$$\Delta e(k) = e(k) - e(k-1) \tag{29}$$

With $y^d(k)$ and λ donate respectively the desired output and a positive constant that determines the slope of the sliding surface.

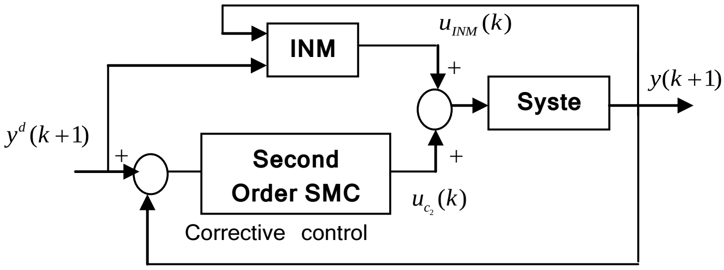

Fig. 6 : The adopted neuro-sliding mode control structure

In case of second order sliding mode control, the sliding surface and the sliding mode corrective term are given by equations (30) and (31), respectively [34]:

$$\sigma(k) = S(k) + \beta S(k-1) \tag{30}$$

With $\beta \in [0,1[$ to ensure the stability of $\sigma(k)$.

The associated control action is given by the following equation:

$$u_{c_2}(k) = u_{c_2}(k-1) + K\,sign(\sigma(k)) \tag{31}$$

K is a constant and $sign(.)$ is the signum function defined as:

$$sign(\sigma(k)) = \begin{cases} 1 & \sigma(k) > 0 \\ 0 & \sigma(k) = 0 \\ -1 & \sigma(k) < 0 \end{cases} \tag{32}$$

5. SIMULATION RESULTS

In this section, the different proposed neural control strategies are evaluated through a numerical example described by a recurrent nonlinear equation inspired from [6]:

Consider the nonlinear uncertain system given by equation (33) which is a modified version of the one presented in [6]:

$$y(k+1) = \frac{a(k)y(k)}{1 + b(k)y^2(k)} + c(k)u^3(k) \tag{33}$$

The variables $u(k)$ and $y(k)$ indicate respectively the input and the output of the system at the instant k.

a, b and c are bonded uncertain parameters such as: $a(k) \in [0.25; 1.75]$, $b(k) \in [0.5; 1.5]$

$$c(k)\begin{cases} case\ 1 & c(k) \in [0.95; 1.05] \\ case\ 2 & c(k) \in [0.9; 1.1] \\ case\ 3 & c(k) \in [0.8; 1.2] \end{cases}$$

Assume that the variations of the parameters a, b and c can be given by the following figure:

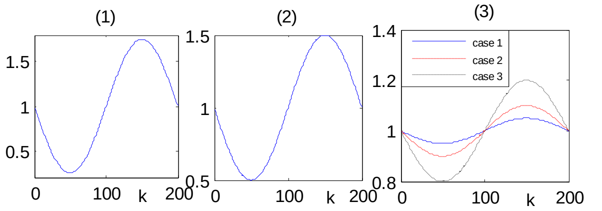

Fig. 7 : Variation of a (1), variation of b (2), and variation of c (3)

5.1 Neural Modeling: DNM

According to the control structure, a DNM has to be developed to emulate the behavior of the uncertain nonlinear system (33). The proposed DNM has two inputs $u(k)$ and $y(k)$, $y_m(k+1)$ is the output of this model. The input is a signal with amplitude distributed over the interval $[0, 2]$. In order to ensure compromise between the quality of modeling and the time of convergence, the choice of the DNM's parameters has been done after several simulations. In fact, neural model parameters are chosen such that: $N_c^m = 5$ and a total of training sets $N = 200$. In this work, the SM-BP algorithm considers fixed learning rates. Thus several simulation results were carried out in order to find the best values such that the sliding surface be smooth enough, avoiding thus chattering problems. The deduction of the boundaries of α_m and β_m is not shown here, nevertheless, the values chosen are within these boundaries. Fig. 8 illustrates the behavior of the DNM on a test data set for the parametric variations given by Fig.7.

Table 1. DNM training parameters

Case 1	Case 2	Case 3
$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.95, 1.05]$	$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.9, 1.1]$	$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.8, 1.2]$
$\alpha_m = 0.48, \beta_m = 4.6$ $C = C_H = 1$	$\alpha_m = 9, \beta_m = 5$ $C = C_H = 5$	$\alpha_m = 6.5, \beta_m = 1$ $C = C_H = 3.5$

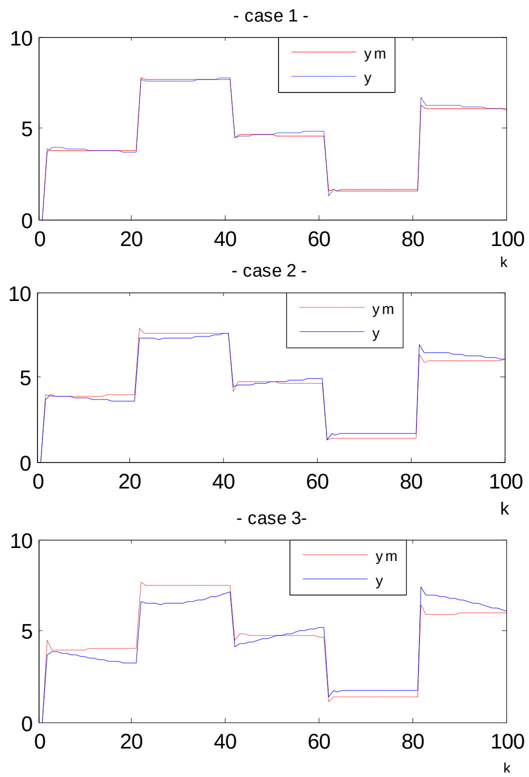

Fig. 8 : Evolution of the system output $y(k)$, the DNM output $y_m(k)$ for the validation set for different cases of variation of parameters a, b and c

According to the simulation results given by Fig.8, the DNM presents acceptable accuracy for the representation of the dynamic of system (33) for small variation of the parameter c $(c \in [0.95, 1.05], c \in [0.9, 1.1])$. However, for more important variation $(c \in [0.8, 1.2])$ the DNM is less accurate.

5.2 Neural Control

The design of the robust neural controller for uncertain nonlinear systems through three control strategies is presented in this section.

- INM controller

After satisfactory training of the DNM, it is used to train the INM. The input vector of the INM is composed by the desired output $y^d(k+1)$ and the output of the neural model $y(k)$, one hidden layer with five hidden neurons $N_c^c = 5$ and $u(k)$ as output. The training set is composed by $N = 200$. INM training parameters are given by table 2.

Table 2. INM training parameters

Case 1	Case 2	Case 3
$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.95, 1.05]$	$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.9, 1.1]$	$a(k) \in [0.25, 1.75]$ $b(k) \in [0.5, 1.5]$ $c(k) \in [0.8, 1.2]$
$\alpha_c = 3, \beta_c = 9.8$ $C = C_H = 1$	$\alpha_c = 2, \beta_c = 13$ $C = C_H = 2$	$\alpha_c = 3, \beta_c = 15$ $C = C_H = 5$

After a satisfactory training, the synthesis INM is placed in cascade with the plant to be controlled, thus it is used as a neural controller for the uncertain nonlinear system. The evolution of the system output for the parametric variations given by Fig. 7, the desired output and different control signal are illustrated by Fig.9.

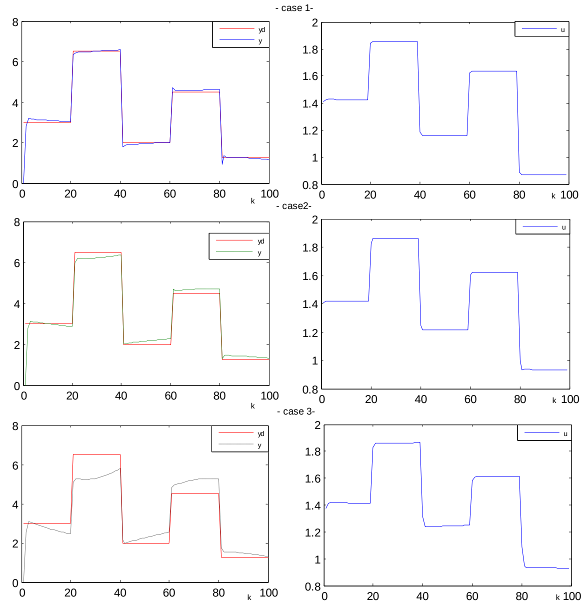

Fig. 9 : Evolution of the system output $y(k)$, the desired output $y^d(k)$ and the control signal $u(k)$ for different cases of the variation of parameters a,b and c

To show the robustness performances of the synthesized INM controller against parametric variations given by Fig.7, the error between the desired output and the system one is computed as follows:

$$E = \frac{1}{100}\sum_{k=1}^{100}\left(y^d(k) - y(k)\right)^2 \tag{34}$$

Table 3. Comparative results

Parametric variations	E
Case 1	0.0038
Case 2	0.0114
Case 3	0.2012

It is noted from Fig.9 and table 3 that the INM controller is not able to present satisfactory tracking performances for the parametric variations of case 3 where:

$\left(a(k) \in [0.25\,;1.75],\ b(k) \in [0.5\,;1.5]\ and\ c(k) \in [0.8\,;1.2]\right).$

Thus it is recommended to propose others control strategies in order to improve control results of the system (33) affected by strong parametric uncertainties.

Assume that: $a(k) \in [0.25;1.75]$, $b(k) \in [0.5;1.5]$ and $c(k) \in [0.8;1.2]$ in the following parts.

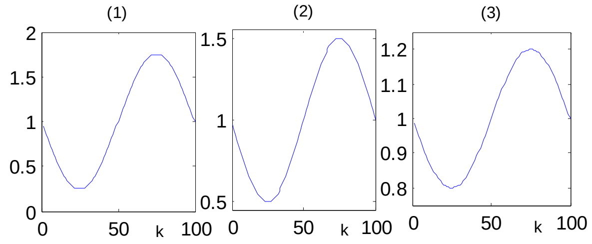

Fig. 10 : Variation of a (1), variation of b (2) and variation c (3)

- Hybrid control

According to the hybrid control structure, a proportional integral controller is used to operate simultaneously with the INM developed in the previous section. The proportional and the integral gains are chosen such as: $k_p = 0.1$ and $k_i = 0.01$. The evolution of the system output, the desired one and the control signal are shown in Fig.11.

The performance result of the hybrid control is compared with the INM one.

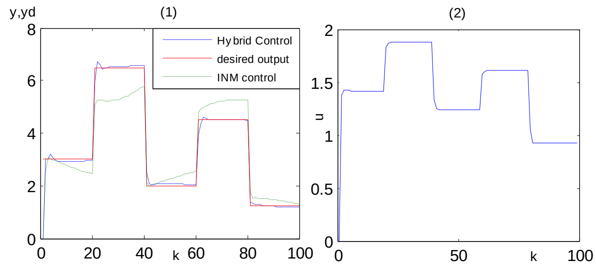

Fig. 11: Evolution of the system output $y(k)$ controlled by hybrid control strategy and INM controller, the desired output $y^d(k)$ (1) and control signal $u(k)$ associated to the hybrid approach (2)

Fig.11 shows that the performance of the hybrid control strategy is satisfactory when the system to be controlled is affected by parametric variations given by Fig.10.

- Neuro-sliding mode control

For this proposed neuro-sliding mode control approach, a second order sliding mode corrective controller is used to operate with the INM synthesized previously in order to compensate the effect of the parametric variations on the system to be controlled.

The simulations parameters are chosen as:

$\lambda = 2.5,\ \beta = 0.2,\ K = 0.03, \phi = 0.47$

In order to avoid chattering problems, the $sign(.)$ function is replaced by the following function:

$$sat(\sigma(k)) = \begin{cases} \dfrac{\sigma(k)}{\phi} & \left|\dfrac{\sigma(k)}{\phi}\right| < 1 \\ sign(\sigma(k) & \left|\dfrac{\sigma(k)}{\phi}\right| > 1 \end{cases} \tag{35}$$

Fig.12. illustrates the evolution of the system output, the desired one and the control signal.

The performance result of neuro-sliding control is compared with the INM ones.

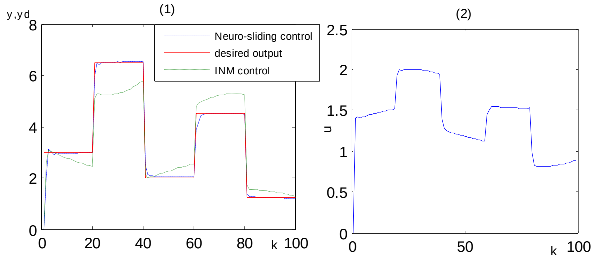

Fig. 12 : Evolution of the system output $y(k)$ controlled by neuro-sliding control strategy and INM controller, the desired output $y^d(k)$ (1) and control signal $u(k)$ associated to the neuro-sliding mode approach (2)

It is noted from the simulation results given by Fig.12 that the adding of the second order neuro-sliding mode controller to the INM has improved the system tracking performances. The performances of the different control strategies presented above are recapitulated in table 4.

Table 4. Comparative results

Control Strategy	*E*
INM control	0.2012
Hybrid control	0.0062
Neuro-sliding mode control	0.0057

6. CONCLUSION

Tree neural control strategies of a class of SISO nonlinear discrete time system affected by parametric variations were proposed in this paper. The dynamic of this system was approximated by a DNM, in a first step, then based on the specialized learning technique, an INM was synthesized. The SM-BP algorithm was the used training algorithm adopted in this work. In order to improve the tracking performances of the INM controller in case of important parametric variations, hybrid control and neuro-sliding mode control strategies were proposed. A simulation example was employed to illustrate the effectiveness of the proposed control strategies. As future work, others neural control strategies will be studied.

7. REFERENCES

[1] Y-W. Chen, J-B. Yang, C-C. Pan, D-L. Xu, Z-J. Zhou “Identification of uncertain nonlinear systems: Constructing belief rule-based models” Knowledge Based System, vol. 73, pp. 124-133, 2015.

[2] Y. Lin, Y. Shi, R. Burton “Modeling and Robust Discrete-Time Sliding-Mode Control Design for a Fluid Power Electrohydraulic Actuator (EHA) System” IEEE/ASME Transactions on Mechatronics, vol. 18, no. 1, pp. 1-10, 2013

[3] A.aydi, M. Djemal, M. Chtourou “Robust pole assignement for control uncertain nonlinear discrete-time systems” 12th International Multi-Conference on Systems, Signals & Devices, pp. 1-5, 2015.

[4] L.A. Zadeh “Fuzzy Sets” Information and control, vol. 8, pp. 338-353, 1965

[5] C.C. Chuang, J-T. Jeng, C-W. Tao “Hybrid robust approach for TSK fuzzy modeling with outliers” Expert Systems with Applications, vol.36, no. 5, pp. 8925-8931, 2009.

[6] K.S. Narendra, K. Parthasarathy “Identification and control of dynamical systems using neural networks” IEEE Transaction on Neural Networks, vol. 1, no.1, pp. 4–27, 1990.

[7] A.V. Topalov, O. Kaynak “Robust neural identification of robotic manipulators using discrete time adaptive sliding mode learning” In Proceeding of International Federation of Automatic Control World Congress, vol. 38, no.1, pp. 336-341, 2005.

[8] X-H. Ji “Fuzzy neural network control and identification for uncertain nonlinear systems” Chinese Control and Decision Conference, pp.4237-4242, 2010

[9] X, Wang, T.Li, C.L.Philip Chen, B. Lin “Adaptive robust control based on single neural network approximation for a class of uncertain strict-feedback discrete-time nonlinear systems” Neurocomputing, vol. 138, pp. 325-331, 2014.

[10] Z. Wang, D. W. C. Ho, Y. Liu, X. Liu “Robust control for a class of nonlinear discrete delay stochastic systems with missing measurements” Automatica, vol.45, no. 3, pp. 684-691, 2009.

[11] S. Sam Ge, J. Wang “Robust adaptive tracking for time-varying uncertain nonlinear systems with unknown control coefficients” IEEE Transaction on Automatic Control, vol. 48, no.8, pp. 1463-1469, 2003.

[12] Z. Yan, J. Wang “Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks” IEEE Transactions on Neural Networks and Learning system, vol. 25, no. 3, pp. 457-469, 2014.

[13] H. Abid, M. Chtourou, A. Toumi “Robust fuzzy sliding mode controller for discrete nonlinear systems” International Journal of Computers, Communications & Control vol. 3, no. 1, pp. 6-20, 2008.

[14] F O. Tellez, E N. Sanchez, R G. Hernandez, J A. Ruz-Hernandez, J L. Rullan-Lara “Neural inverse optimal control for discrete time uncertain nonlinear systems stabilization” The International Joint Conference on Neural Networks (IJCNN), pp. 1-6, 2012.

[15] T. Huang “Nonlinear torque and air-to-fuel ratio control of spark ignition engines using neuro-sliding mode techniques” International Journal of Neural Systems, vol. 21, no. 3, pp. 213–224, 2011.

[16] T.A.A. Alzohairy. “Neural internal model control for tracking unknown nonaffine nonlinear discrete-time systems under external disturbances” International Journal of Computer Applications, vol. 40, no.6, pp. 19-26, 2012.

[17] D.E. Rumelhart, G.E. Hinton, R.J. Williams “Learning internal representation by error propagation. Parallel distributed processing: explorations in the microstructure of cognition, MIT Press Cambridge, MA, vol.1, pp. 318-362, 1986.

[18] G.G. Parma, B.R. Menezes, A.P. Barga “Improving backpropagation with sliding mode control” Proceedings of the Vth Brazilian Symposium on Neural Networks.

Belo Horizonte, Brazil: IEEE Computer Society Press, pp. 8-13, 1998

[19] Zaidi, M, Chtourou, M, Djemel "Robust neural control of discrete time uncertain nonlinear systems using sliding mode backpropagation training algorithm" submitted to International Journal of Automation and Computing.

[20] G. Parma, B.R. Menezes, A.P. Barga "Neural networks learning with sliding mode control: the sliding mode backpropagation algorithm" International Journal of Neural Networks, vol. 9, pp.187-193, 1999

[21] V.I Utkin. "Sliding modes and their application in variable structure systems" MIR, Moscow, 1978.

[22] S.Z. Sarpturk, Y. Istefanopulos, O. Kaynak "On the stability of discrete time sliding mode control system" IEEE Transaction on Automatic Control, vol. 32, no. 10, pp. 930-932, 1987

[23] D. Psaltis, A.Sideris, A.A. Yamamura "A multilayered neural network controller" IEEE Control Systems Magazine, pp. 17-21, 1988.

[24] K J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop "Neural Networks for Control Systems -A Survey," Automatica, vol. 28, no. 6, pp. 1083-1112, 1992.

[25] W.T. van Luenen "Neural networks for control: on Knowledge Representation and Learning," Ph.D Thesis, Control Laboratory of Electrical Engineering, University of Twente,

[26] Enschede, the Netherlands. 1993.

[27] O. Sørensen "Neural networks in control applications" Ph.D. Thesis. Aalborg University, Department of Control Engineering, 1994

[28] Y. Lin, Y. Shi, R. Burton "Modeling and robust discrete-time sliding-mode control design for a fluid power electrohydraulic actuator (EHA) System" IEEE/ASME Transactions on Mechatronics, vol.18, no.1, pp. 1-10, February 2013.

[29] D.M. Tuan, Z. Man, C. Zhang, J. Jin, H. Wang "Robust sliding mode learning control for uncertain discrete-time multi-input multi-output systems" IET Control Theory and Applications, vol. 8, no. 12, pp. 1045-1053, 2014.

[30] F.Huang, Y. Jing, G.M. Dimirovski " Sliding mode feedback control for uncertain discrete-time Markov Jump systems" Proceedings of the 18th World Congress The International Federation of Automatic Control (IFAC), vol. 4, no. 1, pp. 2419-2423, 2011.

[31] V.I Utkin. "Sliding mode in Control and Optimization". Springer- Verlag, 1981.

[32] M. Ertugrul, O. Kaynak "Neuro sliding mode control of robotic manipulators". Mechatronics, vol. 10, no. 12, pp. 243-267, 2000

[33] Ch-H. Tsai, H-Y. Chung, F-M. Yu "Neuro-Sliding mode control with its applications to seesaw systems" IEEE transactions on neural networks, vol. 15, no. 1, pp. 124-134, January 2004.

[34] T. Huang, "Nonlinear torque and air-to fuel ratio control of spark ignition engines using neuro-sliding mode techniques" International Journal of Neural Systems, Vol. 21, No. 3, pp. 213-224, 2011

[35] M. Mihoub, A.S. Nouri, R. B. Abdennour "Real time application of discrete second order sliding mode control to a chemical reactor" Control Engineering Practice, Vol. 17, pp. 1089-1095, 2009.