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:

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

(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.

\[
p = p_0 + \Delta p
\]  

\( p_0 \) represents the nominal parameters and \( \Delta p \) 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:

![DNM Training Process Diagram](image)

Fig. 1: DNM training process

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

\[
y_m(k+1) = \hat{F}(y(k),...,y(k-n+1),u(k),...)
\]  

\( y_m \) 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:

\[
J = \frac{1}{2}e^*e^*
\]  

\( e^* = y(k+1) - y_m(k+1) \) 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 in 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]:

\[
S^w_j(k) = X^w_{zi}(k) + C^w_jX^w_j(k)
\]  

With \( j = 1 \) (5)

The index \( m \) refers to DNM’s parameters, \( j \) is the output node and \( C_0 > 0 \)

\[
X^w_i(k) = \left[ y(k) - y_m(k) \right] f^w \left[ V^w_j(k) \right]
\]  

\( X^w_i(k) = X^w_i(k) - X^w_i(k-1) \)

Where \( f^w \) denotes the derivative of the output activation function, \( V^w_j \) is the global input of the output node \( j \).

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

\[
S^w_h(k) = X^w_{zh}(k) + C^w_hX^w_h(k) \quad h = 1,2...,N^w
\]  

Where \( h \) is the hidden node, \( C_{0h} > 0 \) and:

\[
X^w_{zh}(k) = X^w_h(k)W^w_h(k)f^w \left[ R^w_h(k) \right]
\]  

\[
X^w_{zh}(k) = X^w_{zh}(k) - X^w_{zh}(k-1)
\]

\( W^w \) represents the weight between the output node \( j \) and the hidden node \( h \), \( f^w \) denotes the derivative of the hidden activation function, \( R^w_h \) is the global input of the hidden node \( h \) and \( N^w \) represents the number of neurons in the hidden layer.

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

\[
\Delta W^w_{hm}(k) = \alpha_w \text{sgn}[S^w_h(k)] \left[ X^w_{hi}(k) \right] Y^w_m(k)
\]  

\[
\Delta Z^w_{wm}(k) = \beta_w \text{sgn}[S^w_m(k)] \left[ X^w_{hz}(k) \right] T^w_m(k)
\]

With, \( i = 1,...,(n+m+1) \)

Where \( Y^w_m \) is the output of the hidden node \( h \), \( T^w_m \) is the input of the input node \( i \), \( Z^w_m \) represents the weight between the hidden node \( h_m \) and the input node \( i \). \( \alpha_w > 0 \) and \( \beta_w > 0 \).

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

\[
S \frac{ds}{dt} < 0
\]  

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 \( \alpha_w \) and \( \beta_w \) 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 \( y_n \approx y \) after satisfactory training, the INM is trained.

![Fig. 2 : Specialized method for INM training](image)

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

\[
J_e = \frac{1}{2} \| e \|^2
\]

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

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:

\[
S_j^c(k) = X_j^c(k) + C_j X_j^c(k)
\]

\( C_j > 0 \)

\( X_j^c(k) = (y(k+1) - y^*(k+1)) Y_j^c(k) \cdot f^c(V_j^c(k)) \)

\( f^c \) 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:

\[
S_h^c(k) = X_h^c(k) + C_h X_h^c(k)
\]

\( C_h > 0 \)

\( X_h^c(k) = X_h^c(k) \cdot f_h^c(R_h^c(k)) \)

\( R_h^c \) represents the weight between the hidden node \( h \) and the input node \( i \). \( Y_m^c(k) \) represents the output of the hidden node \( h \). \( T_i^c \) represents the input of the input node \( i \) of the INM.

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:

\[
u(k) = u_{OM}(k) + u_{PI}(k)
\]

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

\[
u_{PI}(k) = k_p e(k) + k_i e(k-1)
\]

\( k_p \) and \( k_i \) are proportional and integral gains, respectively.

\( e(k) \) is the tracking error defined as:
\[ e(k) = y^*(k) - y(k) \]  
(25)

With \( y^*(k) \) represents the desired output and \( y(k) \) is the actual system output.

\[ u(k) = u_{\text{eq}}(k) + u_c(k) \]  
(26)

Where \( u_{\text{eq}}(k) \) 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.

\[ y^*(k+1) + e(k+1) \]  
(27)

Where \( u_{\text{eq}}(k) \) is the control action provided by the INM and \( u_c(k) \) represents the second order sliding mode corrective control.

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

\[ S(k) = \Delta e(k) + \lambda e(k) \]  
(28)

\[ e(k) = y^*(k) - y(k) \]  
(29)

\[ \Delta e(k) = e(k) - e(k-1) \]  

With \( y^*(k) \) and \( \lambda \) donate respectively the desired output and a positive constant that determines the slope of the sliding surface.

\[ y^*(k+1) + u_{\text{eq}}(k) \]  
(30)

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]:

\[ \sigma(k) = S(k) + \beta S(k-1) \]  
(31)

\[ K \] is a constant and \( \text{sign}(\cdot) \) is the signum function defined as:

\[ \text{sign}(\sigma(k)) = \begin{cases} 1 & \sigma(k) > 0 \\ 0 & \sigma(k) = 0 \\ -1 & \sigma(k) < 0 \end{cases} \]  
(32)

5. SIMULATION RESULTS

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

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

\[ y(k+1) = \frac{a(k)y(k)}{1+b(k)y^2(k)} + c(k)u^*(k) \]  
(33)

The variables \( u(k) \) and \( y(k) \) indicate respectively the input and the output of the system at the instant \( k \).
and \( c \) are bonded uncertain parameters such as:
\[
\begin{align*}
\alpha(k) &\in [0.25;1.75] \\
\beta(k) &\in [0.5;1.5] \\
\gamma(k) &\in [0.95;1.05] \\
\end{align*}
\]
Assume that the variations of the parameters \( \alpha, \beta \) and \( \gamma \) can be given by the following figure:

Fig. 7: Variation of \( \alpha(1) \), variation of \( \beta(2) \), and variation of \( \gamma(3) \)

### 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: \( N^e = 5 \) 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 \( \alpha_n \) and \( \beta_n \) 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.

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

![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 \( \alpha, \beta \) and \( \gamma \)](image)

#### Table 1. DNM training parameters

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
</tr>
<tr>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
</tr>
<tr>
<td>( \alpha_n = 0.48, \beta_n = 4.6, C = C_H = 1 )</td>
<td>( \alpha_n = 9, \beta_n = 5, C = C_H = 5 )</td>
<td>( \alpha_n = 6.5, \beta_n = 1, C = C_H = 3.5 )</td>
</tr>
</tbody>
</table>

#### Table 2. INM training parameters

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
<td>( \alpha ) ( \beta ) ( \gamma )</td>
</tr>
<tr>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
<td>( [0.25;1.75] ) ( [0.95;1.05] ) ( [0.8;1.2] )</td>
</tr>
<tr>
<td>( \alpha_n = 3, \beta_n = 9.8, C = C_H = 1 )</td>
<td>( \alpha_n = 2, \beta_n = 13, C = C_H = 2 )</td>
<td>( \alpha_n = 3, \beta_n = 15, C = C_H = 5 )</td>
</tr>
</tbody>
</table>
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.

\[ E = \frac{1}{100} \sum_{k=1}^{100} (y(k) - y_d(k))^2 \]  

(34)

**Table 3. Comparative results**

<table>
<thead>
<tr>
<th>Parametric variations</th>
<th>( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.0038</td>
</tr>
<tr>
<td>Case 2</td>
<td>0.0114</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.2012</td>
</tr>
</tbody>
</table>

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:

\[ (a(k) \in [0.25;1.75], b(k) \in [0.5;1.5] \text{ and } c(k) \in [0.8;1.2]) \]

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:  
\[ a(k) \in [0.25;1.75], b(k) \in [0.5;1.5] \text{ and } c(k) \in [0.8;1.2] \]

in the following parts.

- **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 integral gains are chosen 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. 10: Variation of \( a \) (1), variation of \( b \) (2) and variation \( c \) (3)**

**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 \( \text{sign}(.) \) function is replaced by the following function:

\[ \text{sat}(\sigma(k)) = \begin{cases} \frac{\sigma(k)}{\phi} & \text{if } \sigma(k) < 1 \\ \text{sign}(\sigma(k)) & \text{if } \sigma(k) > 1 \end{cases} \]

(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)](image)

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

<table>
<thead>
<tr>
<th>Control Strategy</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INM control</td>
<td>0.2012</td>
</tr>
<tr>
<td>Hybrid control</td>
<td>0.0062</td>
</tr>
<tr>
<td>Neuro-sliding mode control</td>
<td>0.0057</td>
</tr>
</tbody>
</table>

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


