Neural network based adaptive sliding mode control of uncertain nonlinear systems

Ghania Debbache and Noureddine Goléa*
Electrical Engineering Institute, Oum El Bouaghi University, 04000 Oum El Bouaghi, Algeria

Abstract: The purpose of this paper is the design of neural network-based adaptive sliding mode controller for uncertain unknown nonlinear systems. A special architecture adaptive neural network, with hyperbolic tangent activation functions, is used to emulate the equivalent and switching control terms of the classic sliding mode control (SMC). Lyapunov stability theory is used to guarantee a uniform ultimate boundedness property for the tracking error, as well as of all other signals in the closed loop. In addition to keeping the stability and robustness properties of the SMC, the neural network-based adaptive sliding mode controller exhibits perfect rejection of faults arising during the system operating. Simulation studies are used to illustrate and clarify the theoretical results.

Keywords: nonlinear system, neural network, sliding mode control (SMC), adaptive control, stability, robustness.

DOI: 10.1109/JSEE.2012.00015

1. Introduction

Sliding mode control (SMC) is a powerful scheme for nonlinear systems with uncertainties [1−3]. However, this control scheme suffers from some problems. The first problem is the chattering phenomenon caused by the finite switching frequency on the sliding surface, and therefore deteriorates the system performance. The second problem is that, in order to guarantee the stability of the sliding mode system, the uncertainties bounds are to be known, which is not the case for the uncertainties untaken into consideration, such as unmodeled dynamic or faults arising during the system operating. Several approaches for reducing the chattering have been proposed, among which the well known one is to apply a saturation function or continuous signum function to the control gain when the sliding surface is within a boundary of the sliding hyper-plane [2,3]. An alternative way to solve the chattering problem is the application of the fuzzy logic in the construction of the control input (fuzzy SMC) [4,5]. To solve the problem of the unknown uncertainties estimation, intelligent techniques such as neural networks or fuzzy systems are combined with the basic SMC, where the former is used as nonlinear approximators of the nonlinearities, and the latter is used to ensure stability and robustness properties (for an early survey, see [6−8] and references therein). In [9] neural networks are used to estimate the uncertainties bounds. The outputs of the neural networks adaptively adjust the gain of the sliding mode controller so that the effects of system uncertainties can be eliminated. In [10,11] the SMC and the fuzzy neural networks (FNN) are integrated by a smooth transformation. It is proposed to let the SMC force the state tracking error to slide into the boundary layer, then to use the FNN in the boundary layer to let the tracking error converge asymptotically to the neighbor of zero. In [12] a single auto-tuning neuron is activated to replace the SMC when the state trajectory of system goes into the boundary layer in order to eliminate the chattering effect. In [13] the SMC is used to control the system nominal part, while the neural network compensates for the uncertainties terms. In [14], a neural network is used to minimize the cost function that is selected to depend on the distance from the sliding-mode manifold, thus providing that the neural network controller enforces sliding-mode motion in the closed-loop system. In [15,16], wavelets and radial basis function (RBF) networks are used to estimate the bound of uncertainties terms on line. In [17] the cerebellar model articulation controller (CMAC) network is used to reproduce the equivalent control term, and an $H_\infty$ supervisory term is used to attenuate the approximation error effect. In [18,19] experimental implementations are proposed to validate SMC-neural network hybrid approaches.

In this paper, we propose a new neural network-based adaptive SMC (NASMC) scheme for a class of nonlinear systems. It is shown that, unlike other neural network-based control schemes, neural network is not directly used to learn the system nonlinearities, but it is used to adap-
tively learn the SMC dynamic in a compact set of the system space. The output of the neural network then adaptively emulates the SMC so that the effects of system uncertainties can be eliminated and the output tracking error between the plant output and the desired reference signal can asymptotically converge to bounded value. The proposed neural control scheme behaves with the strong robustness with respect to unknown dynamics and nonlinearities. Unlike previous neural network-SMC hybrid works, this is not a combination of neural networks and SMC approaches, but a new implementation of adaptive SMC using a single neural network, with a special architecture, whose parameters are trained on line using the sliding surface information. The NASMC demonstrates that, in addition to keeping the stability and robustness properties of classical SMC, it is robust against unmodeled dynamics and faults arising during the system operating. Simulation results for a single input single output (SISO) nonlinear uncertain system and the multiple input multiple output (MIMO) chaotic Chua circuit show the effectiveness of the proposed approach.

2. SMC

In this section, we introduce the design of SMC for a class of SISO uncertain nonlinear systems whose dynamic equations can be expressed in the following form:

\[ x^{(n)} = f(x, t) + g(x, t)u(t) + \eta(x, t) \]  

(1)

where \( x^T = [x(t) \quad \dot{x}(t) \quad \ldots \quad x^{(n-1)}(t)] \in \mathbb{R}^n \) is the system state vector, \( u \) is the system input, \( f(x, t) \) and \( g(x, t) \) are smooth nonlinear functions, and \( \eta(x, t) \) is a bounded uncertainty.

The desired bounded state trajectory is defined by \( x^{\text{d}} = [x_r(t) \quad \dot{x}_r(t) \quad \ldots \quad x_r^{(n-1)}(t)] \in \mathbb{R}^n \), then, the tracking error is defined by \( \varepsilon = x - x^{\text{d}} \). The control objective is to design the control input \( u(t) \), such that \( \varepsilon \) tracks \( x^{\text{d}} \), with all closed loop signals remaining bounded.

For the SMC design, the following assumptions are used:

S1) The nonlinear functions \( f(x, t) > 0, g(x, t) > 0 \) and \( \dot{g}(x, t) \) are known.

S2) The uncertain term is bounded by \( |\eta(x, t)| \leq \eta_0(x, t) \), where \( \eta_0(x, t) \) is a known positive function.

In general, SMC design follows two standard steps: (i) a sliding surface is designed such that the closed-loop system exhibits desired dynamic behavior during sliding mode; and (ii) a robust control law is employed to force the system states to remain on the sliding surface \([1-3]\). This results in a surface-dependent sliding motion insensitive to perturbations or matched uncertainties.

As first step, define the sliding surface as

\[ s = [K_1 \quad 1]e \]  

(2)

with \( K = [k_1 \quad \ldots \quad k_{n-1}] \in \mathbb{R}^{n-1} \) chosen such that the zeros of the polynomial \( [K_1 \quad 1]e \) are in the left half of the complex plane. \( s = 0 \) is called the switching plane variable and the sliding mode in sliding mode control literature \([2-3]\).

Considering the system (1) and definition (2), the time derivative of \( s \) may be written as

\[ \dot{s} = v - f(x, t) - g(x, t)u - \eta(x, t) \]  

(3)

where

\[ v = [0 \quad \dot{K}]e + x_r^{(n)}. \]  

(4)

To derive the control input \( u(t) \) and verify the closed loop stability, we define the Lyapunov function

\[ V = \frac{1}{2g(x, t)} s^2. \]  

(5)

Then, the differentiation of (5) along (3) yields

\[ \dot{V} = s \left(-\frac{\dot{g}(x, t)}{2g^2(x, t)} s + \frac{v - f(x, t) - u - \eta(x, t)}{g(x, t)} \right). \]  

(6)

Further, define the SMC control input as

\[ u(t) = u_{eq} + u_s^* \]  

(7)

with the equivalent control term defined as

\[ u_{eq}^* = \lambda s - \frac{\dot{g}(x, t)}{2g^2(x, t)} s + \frac{v - f(x, t)}{g(x, t)}. \]  

(8)

for some \( \lambda > 0 \).

Then, introducing (8) in (6) becomes

\[ \dot{V} = -\lambda s^2 - s \left(u_{eq}^* + \frac{\eta(x, t)}{g(x, t)}\right). \]  

(9)

Now, the switching control term is defined as

\[ u_s^* = \frac{\eta_0(x, t)}{g(x, t)} \text{sign}(s) \]  

(10)

where sign \((s)\) is the signum function.

Finally, introducing (10) in (9) yields

\[ \dot{V} \leq -\lambda s^2 \]  

(11)

which implies that \( \lim_{t \to \infty} s = 0 \), and by the definition (2) also \( \lim_{t \to \infty} \varepsilon = 0 \).

Remark 1  In practice, the signum function is not used since it causes the undesirable chattering phenomenon. Thus, it is replaced by the saturation function \( \text{sat}(s/\varepsilon) \) or the continuous signum \( s/(|s| + \varepsilon) \) for some small boundary parameter \( \varepsilon > 0 \). Then, the tracking error converges to a neighborhood of the origin defined by the magnitude of the parameter \( \varepsilon \).
3. NASMC

In this section, we consider the design of NASMC for the class of systems defined by (1). For the NASMC design, the assumptions required about the nonlinear system (1) are sensibly different. This is due to the fact that neural approaches consider the controlled system nonlinearities as unknown. Then, the assumptions to be used are as follows:

NS1) \( f(\mathbf{x}, t) \) and \( g(\mathbf{x}, t) \) are smooth unknown nonlinear functions. Without loss of generality we assume that \( g(\mathbf{x}, t) > 0 \).

NS2) The uncertain term in (1) represents the external disturbances, and is bounded by \( |\eta(\mathbf{x}, t)| \leq \eta_0 \), where \( \eta_0 \) is a known positive constant.

3.1 NASMC design

The structure of the proposed NASMC is depicted in Fig. 1, where we distinguish a single neural network, which will emulate adaptively the SMC dynamic. The update algorithm uses principally the sliding surface \( s \) information to adjust the neural network parameters.

The detailed structure of the used neural network is depicted in Fig. 2. As can be seen, the network, with one hidden layer, has a special structure and is composed by two parts. The first part is composed by \( q \) neurons, and \((n+2)\) inputs are used to emulate the equivalent control term in the SMC. The second part, composed by a single neuron with single input \( s \), represents the robust control term. The NASMC action is formed by adding the two parts contributions.

Hence, using the above described network architecture, the control action is given by

\[
 u = \sum_{i=1}^{q} w_i \phi_i \left( L \mathbf{z} \right) + \beta \phi_{q+1} \left( \alpha s \right) \tag{12}
\]

where \( \mathbf{z}^T = [x^T \, v \, s] \in \mathbb{R}^{n+2} \) is the neural network input vector, \( \phi_i \left( L \mathbf{z} \right) \) \((i = 1, \ldots, q)\) are the equivalent term activation functions with \( L_i = [l_{i1} \, \ldots \, l_{i(n+2)}] \in \mathbb{R}^{n+2} \) as the input layer weighting parameters, and \( w_i \) \((i = 1, \ldots, \) \footnote{Equation number} \) are the output weights. Finally, the robustifying term in (12) is represented by the single neuron \( \phi_{q+1} \) with the input parameter \( \alpha \) and the output weight \( \beta \).

In more compact form, (12) can be written as

\[
 u = \mathbf{w} \psi \left( L, \phi \mathbf{z} \right) + \beta \phi_{q+1} \left( \alpha s \right) \tag{13}
\]

where \( \mathbf{w} = [w_1 \, \ldots \, w_q] \in \mathbb{R}^{q}, L = [L_1 \, \ldots \, L_q] \) and \( \psi^T \left( L, \phi \mathbf{z} \right) = \left[ \phi_1 \left( L_1 \mathbf{z} \right) \, \ldots \, \phi_q \left( L_q \mathbf{z} \right) \right] \in \mathbb{R}^{q} \).

In the SMC framework, the first term in the right side of (13) is denoted as \( u_{eq} (\mathbf{z}, \theta) \) with \( \theta = [\mathbf{w}, L]^T \) and the second term is denoted as \( u_s (s, \alpha, \beta) \). Hence, the output of the adaptive neural network in Fig. 2, can be rewritten as

\[
 u = u_{eq} (\mathbf{z}, \theta) + u_s (s, \alpha, \beta). \tag{14}
\]

The activation functions \( \phi_i (\alpha y) \) \((i = 1, \ldots, q + 1)\) are hyperbolic tangent functions defined by

\[
 \phi_i (\alpha y) = \frac{e^{\alpha y} - e^{-\alpha y}}{e^{\alpha y} + e^{-\alpha y}}. \tag{15}
\]

The following properties of \( \phi_i (\alpha y) \) are necessary to the subsequent analysis:

P1) For any \( y_1, y_2 \in \mathbb{R} \), we have

\[
 |\phi_i (\alpha y_1) - \phi_i (\alpha y_2)| \leq \alpha_0 |y_1 - y_2| \tag{16}
\]

for some \( \alpha_0 > 0 \). Observing that \( \phi_i (\alpha y) \) is uniformly Lipschitz, property (16) can be easily proved.

P2) For any \( \alpha > 0 \) and for any \( y \in \mathbb{R} \), the following inequality holds

\[
 |y| - y \phi_i (\alpha y) \leq \frac{\kappa}{\alpha} \tag{17}
\]

where \( \kappa \) is a constant that satisfies \( \kappa = e^{-(\kappa+1)} \), that is \( \kappa = 0.2785 \).

According to the neural networks approximation results [20, 21], there exists an optimal parameter vector \( \theta^* \) such that

\[
 u_{eq}^* = u_{eq} (\mathbf{z}, \theta^*) + \varepsilon (\mathbf{z}) \tag{18}
\]

where \( \varepsilon (\mathbf{z}) \) represents the neural network approximation error. The universal approximation theory indicates that,
if \( q \) is sufficiently large, it is possible to made \( \varepsilon (x) \) arbitrarily small on the compact state space region \( \Omega_x \). The optimal vector \( \boldsymbol{\theta}^* \) is an arbitrary quantity required for analytical purpose only, and is typically defined by

\[
\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^{n+1}} \left\{ \sup_{\boldsymbol{x} \in \Omega_x} \left| u^*_q - u_q (\boldsymbol{x}, \boldsymbol{\theta}) \right| \right\}.
\]

The following assumptions on the used neural network are required.

\[
\begin{align*}
\dot{\theta} &= \begin{cases} 
\gamma_1 s \frac{\partial u_q}{\partial \boldsymbol{\theta}}^T, & |\theta| \leq \theta_0 \text{ or } |\theta| = \theta_0 \text{ and } s \theta^T \left[ \frac{\partial u_q}{\partial \theta} \right]^T \leq 0 \\
\gamma_1 s \frac{\partial u_q}{\partial \theta}^T - \gamma_2 s \frac{\partial \theta}{\theta^2} \left[ \frac{\partial u_q}{\partial \theta} \right]^T, & |\theta| = \theta_0 \text{ and } s \theta^T \left[ \frac{\partial u_q}{\partial \theta} \right]^T > 0
\end{cases}
\end{align*}
\]

\[
\dot{\beta} = \begin{cases} 
\gamma_2 s \frac{\partial u_q}{\partial \beta}, & \beta < \beta_h \\
0, & \beta = \beta_h
\end{cases}
\]

\[
\dot{\alpha} = \begin{cases} 
\gamma_3 s \frac{\partial u_q}{\partial \alpha}, & \alpha < \alpha_h \\
0, & \alpha = \alpha_h
\end{cases}
\]

where \( \gamma_i > 0 \) \((i = 1, 2, 3)\) are design parameters.

It can be shown that the update laws (19)–(21) guarantee the boundedness of the adjustable parameters, i.e. \( |\theta| \leq \theta_0, \beta \leq \beta_h \) and \( a \leq a_h \), \( \forall t \geq 0 \), provided that their initial values \( \theta(0), \beta(0), \alpha(0) \) are set correctly.

**Remark 2** Update laws (19)–(21) are used to ensure parameters boundedness, which is needed for subsequent stability proof. On the other hand, (21) guarantees that \( a > 0 \), which is necessary for the property P2) to hold.

### 3.2 Stability analysis

For the stability analysis, we consider the Lyapunov function

\[
V = \frac{1}{2g(x, t)} s^2 + \frac{1}{2\gamma_2} \beta^2
\]

with the parameter error \( \tilde{\beta} = \beta^* - \beta \), and the optimal parameter \( \beta^* \) that will be defined later.

Then, differentiating (22) along (3) and introducing (14) yields

\[
\dot{V} = s \left( -\frac{\hat{g}(x, t)}{2g^2(x, t)} s + \frac{v - f(x, t)}{g(x, t)} - u_q (x, \boldsymbol{\theta}) - u_s (s, \alpha, \beta) - \frac{\eta(x, t)}{g(x, t)} \right) - \frac{1}{\gamma_2} \tilde{\beta} \tilde{\beta}.
\]

Further, adding and subtracting \( u^*_q \) from right side of (23) gives

\[
\dot{V} = -\lambda s^2 + s \left( u^*_q - u_q (x, \boldsymbol{\theta}) - u_s (s, \alpha, \beta) - \frac{\eta(x, t)}{g(x, t)} \right) - \frac{1}{\gamma_2} \tilde{\beta} \tilde{\beta}.
\]

Hence, using the property (18) in (24) yields

\[
\dot{V} \leq -\lambda s^2 + |s| \left( |\varepsilon(x)| + |u_q (x, \theta^*) - u_q (x, \theta)| \right) - s u_s (s, \alpha, \beta) - \frac{1}{\gamma_2} \tilde{\beta} \tilde{\beta}.
\]

Moreover, (25) can be upper bounded by

\[
\dot{V} \leq -\lambda s^2 + |s| \left( |\varepsilon(x)| + |u_q (x, \theta^*) - u_q (x, \theta)| \right) - s u_s (s, \alpha, \beta) - \frac{1}{\gamma_2} \tilde{\beta} \tilde{\beta}.
\]

Now, exploiting the property P1) (16), we have \( |u_q (x, \theta^*) - u_q (x, \theta)| \leq \alpha |\theta| \), where \( \theta = \theta^* - \theta \) is the parameters error, and \( \alpha \) is a given positive constant. Therefore, using the above result in (26) yields

\[
\dot{V} \leq -\lambda s^2 + |s| \left( |\varepsilon(x)| + \alpha |\theta| + \frac{\eta(x, t)}{g(x, t)} \right) - s u_s (s, \alpha, \beta) - \frac{1}{\gamma_2} \tilde{\beta} \tilde{\beta}.
\]

Furthet, by exploiting assumptions NS1), NS2), NN1) and bounding property of the update law (19) (i.e., \( |\theta| \leq 2\theta_0 \)), one can suppose that there exists a positive constant \( \beta^\star \) such that

\[
|\varepsilon(x)| + \alpha |\theta| + \frac{\eta(x, t)}{g(x, t)} \leq \beta^\star.
\]

Then, introducing (28) in (27) gives
Using the fact that \( \beta = \beta^* - \bar{\beta} \), (29) can be arranged as
\[
\dot{V} \leq -\lambda s^2 + |s| \beta^* - s\beta \varphi_{q+1}(as) + \frac{1}{\gamma_2} \bar{\beta} (\gamma_2 s \varphi_{q+1}(as) - \bar{\beta}) .
\] (30)
Hence, using the update law (20) in (30) yields
\[
\dot{V} \leq -\lambda s^2 + \beta^* (|s| - \varphi_{q+1}(as)) .
\] (31)
Finally, using the property P2) (17) in (31) gives
\[
\dot{V} \leq -\lambda s^2 + \beta^* \frac{K}{\alpha} .
\] (32)

The above result shows that \( \dot{V} \) is negative whenever \( s \) is outside the bounded region defined by
\[
s \leq \sqrt{\frac{\beta^* K}{\lambda \alpha}}
\] (33)
which implies that \( s \) will enter the above region, i.e., it is bounded. Hence, using (33) and the bounded input bounded output (BIBO) stability property of (2), the boundedness of the tracking error \( e \) can be concluded.

**Remark 3** Assumption (28) is reasonable since all terms in (28) are bounded. The optimal value \( \beta^* \) needs not to be exactly known, but it can be reasonably situated as \( \beta^* < \beta_h \).

**Remark 4** Unlike the classical SMC or other direct neural network adaptive approaches [9], \( \dot{y}(x, t) \) is not necessary to be known, which simplify the implementation requirements.

### 4. Extension to MIMO nonlinear systems

The described NASMC can be easily extended to a certain class of MIMO nonlinear systems, which expands its practical applications possibilities. To this end, we consider the MIMO nonlinear systems class given by
\[
x^{(r)} = F(x, t) + G(x, t) u + \eta(x, t)
\] (34)
where \( x^{(r)} = [x_1^{(r)} \; x_2^{(r)} \; \cdots \; x_p^{(r)}] \in \mathbb{R}^p \), \( X^T = [X_1^T \; X_2^T \; \cdots \; X_p^T] \in \mathbb{R}^n \) is the state vector, \( X_i^T = [x_i \; \dot{x}_i \; \cdots \; x_i^{(r_i-1)}] \in \mathbb{R}^{r_i} \), for \( i = 1, \ldots, p \), and \( \sum_{i=1}^{p} r_i = n \), are the state subvectors. \( F^T(x, t) = [f_1(x, t) \; f_2(x, t) \; \cdots \; f_p(x, t)] \in \mathbb{R}^p \) is a vector field, and \( G(x, t) = [g_{ij}(x, t)] \in \mathbb{R}^{p \times p} \) is the control gain matrix. \( u^T = [u_1 \; u_2 \; \cdots \; u_p] \in \mathbb{R}^p \) is the control input vector, and \( \eta^T(x, t) = [\eta_1(x, t) \; \eta_2(x, t) \; \cdots \; \eta_p(x, t)] \in \mathbb{R}^p \) is the uncertainties vector.

For MIMO NASMC design, it is required that:

**NSM1** \( F(X, t) \) and \( G(X, t) \) are unknown but smooth, \( G(X, t) = G^T(X, t) > 0 \) (symmetric positive definite).

**NSM2** The uncertain vector elements are bounded by \( |\eta_i(x, t)| \leq \eta_i (i = 1, \ldots, p) \).

**Remark 5** The assumption on \( G(x, t) \) is not very restrictive since several real systems verify this conditions, e.g., robot manipulators.

Each control input \( u_i \) is formed by a neural network, of the same structure as in Fig. 1, composed by \( (q_i + 1) \) neurons, and satisfies the same assumptions NN1 and NN2), and yields the control inputs
\[
u_i = u_{eqi}(z_i, \theta_i) + u_{si}(s_i, a_i, \beta_i), \quad i = 1, \ldots, p.
\] (35)
For the given reference trajectories \( \bar{X}_i \in \mathbb{R}^n \), \( i = 1, \ldots, p \), the sliding surfaces are defined as
\[
s_i = [K_{ei}]_1 \bar{E}_i, \quad i = 1, \ldots, p
\] (36)
with \( K_{ei} = [k_{i1} \; \cdots \; k_{i(r_i-1)}] \in \mathbb{R}^{r_i-1} \), \( E_i = \bar{X}_i - \bar{X}_1 \), and \( \bar{z} = [\bar{S} \; \bar{v} \; \bar{X}] \in \mathbb{R}^{u+2p} \), with \( \bar{S}^T = [s_1 \; s_2 \; \cdots \; s_p] \in \mathbb{R}^p \) is the sliding surfaces vector, and \( \bar{w}^T = [v_1 \; v_2 \; \cdots \; v_p] \in \mathbb{R}^p \) with \( v_i = [0 \; K_{ei}] \bar{E}_i \).

The parameters of the NASMC (35) are adjusted by the same update laws (19)–(21) with the appropriate bounds for each control input.

Then, introducing (36) in (34) yields
\[
\dot{S} = \bar{v} - F(X, t) - G(X, t) u - \eta(X, t).
\] (37)
To show the closed loop stability, consider the Lyapunov function
\[
V = \frac{1}{2} S^T G^{-1}(X, t) S + \sum_{i=1}^{p} \frac{1}{2} \gamma_{2i} \beta_i \bar{\beta}_i^2 .
\] (38)
The differentiation of (38) along (37) yields
\[
\dot{V} = S^T \left[ \frac{1}{2} G^{-1}(X, t) S + G^{-1}(X, t) [\bar{v} - F(X, t)] - u - \zeta(X, t) \right] - \sum_{i=1}^{p} \frac{1}{2} \gamma_{2i} \beta_i \bar{\beta}_i
\] (39)
where \( \zeta(X, t) = G^{-1}(X, t) \eta(X, t) \in \mathbb{R}^p \).

The ideal SMC equivalent control term is given by
\[
\dot{w}_{eq} = S^T \left[ \frac{1}{2} G^{-1}(X, t) S + G^{-1}(X, t) [\bar{v} - F(X, t)] \right] \dot{\zeta}(X, t)
\] (40)
where \( \dot{w}_{eq} = [u_{eq1} \; u_{eq2} \; \cdots \; u_{eqp}] \in \mathbb{R}^p \) and \( \Lambda = \text{diag} \{ \lambda_1 \; \lambda_2 \; \cdots \; \lambda_p \} \in \mathbb{R}^{p \times p} \), with \( \lambda_i > 0 \ (i = \cdots \).
Then, adding and subtracting (40) from (39) yields

\[ \dot{V} = -\tilde{S}^T A \tilde{S} + \tilde{S}^T [\tilde{u}_e^* - u - \zeta_i(\mathbf{x}, t)] - \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \tilde{\beta}_i. \]  

(41)

Hence, (41) can be written element wise as

\[ \dot{V} = -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} s_i [u_{eq} - u - \zeta_i(\mathbf{x}, t)] - \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \tilde{\beta}_i. \]  

(42)

Further, introducing (35) in (42) yields

\[ \dot{V} = -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} s_i \left[u_{eqi} - u_{eqj}(\mathbf{x}, \mathbf{\theta})\right] - \varepsilon_i(s_i, a_i, \beta_i) - \zeta_i(\mathbf{x}, t) - \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \tilde{\beta}_i. \]  

(43)

Then, using the property (18) in (43) yields

\[ \dot{V} = -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} s_i \left[u_{eq} - u_{eqj}(\mathbf{x}, \mathbf{\theta})\right] + \varepsilon_i(\mathbf{x}, t) - u_{eqi}(s_i, a_i, \beta_i) - \zeta_i(\mathbf{x}, t) - \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \tilde{\beta}_i \]  

(44)

where \( \varepsilon_i(\mathbf{x}, t) \) is the \( i \)th neural network reconstruction error.

Hence, using the facts that \( |u_{eq} - u_{eqj}(\mathbf{x}, \mathbf{\theta})| \leq |u_{eq} - u_{eqj}(\mathbf{x}, \mathbf{\theta})| \leq \alpha_i |\mathbf{\theta}| \) and \( |\mathbf{x}| + |\varepsilon_i(\mathbf{x}, t)| + |\zeta_i(\mathbf{x}, t)| \leq \beta^*_i \), for some \( \alpha_i, \beta^*_i \) (\( i = 1, \ldots, p \)), (44) becomes

\[ \dot{V} \leq -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} s_i \beta^*_i - \sum_{i=1}^{p} s_i \beta_i \phi_{q+1}(a_i s_i) - \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \tilde{\beta}_i. \]  

(45)

Using the fact that \( \beta_i = \beta^*_i - \tilde{\beta}_i \) and \( \beta^*_i \) (\( i = 1, \ldots, p \)), (45) can be arranged as

\[ \dot{V} \leq -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} \beta^*_i (|s_i| - s_i \phi_{q+1}(a_i s_i)) + \sum_{i=1}^{p} \frac{1}{\gamma_{2i}} \tilde{\beta}_i \left( s_i \phi_{q+1}(a_i s_i) - \tilde{\beta}_i \right). \]  

(46)

Finally, using the update law (20) and the property P2) (17) in (46) yields

\[ \dot{V} \leq -\sum_{i=1}^{p} \lambda_i s_i^2 + \sum_{i=1}^{p} \frac{\beta^*_i}{a_i} \]  

(47)

From (47), one can note that \( \dot{V} \) is negative whenever the sliding surfaces \( s_i \) are outside the bounded regions defined by

\[ s_i \leq \sqrt{\frac{\beta^*_i}{\lambda_i a_i}}, \quad i = 1, \ldots, p \]  

(48)

which implies that \( s_i \) (\( i = 1, \ldots, p \)) will enter the above regions, i.e., they are bounded. Hence, using (48) and the BIBO stability property of (36), the boundedness of the tracking errors \( \epsilon_i(\mathbf{x}, t) \) (\( i = 1, \ldots, p \)) can be established.

5. Simulation

To illustrate the NASMC approach, we consider two examples. The first one, used to compare NASMC and SMC performances, concerns the control of a nonlinear system with uncertainties and faults arising during the system operating. The second example, used to compare the NASMC and the CMAC-based control scheme proposed in [17], concerns the control of the chaotic MIMO Chua circuit subject to variable parameters uncertainties and external disturbances.

Example 1: We consider the simple but illustrative nonlinear system:

\[ \begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f(x, t) + g(x, t)u + \eta(x, t)
\end{aligned} \]  

(49)

with \( f(x, t) = f_0(x, t) + \Delta f(x, t) \) and \( g(x, t) = g_0(x, t) + \Delta g(x, t) \), where \( f_0(x, t) = x_2^2 - 1.5x_2 + 2 \cos(t)x_2, \Delta f(x, t) = \cos(x_1) \) are the nominal (known) parts of \( f(x, t) \) and \( g(x, t) \), respectively, and \( \Delta f(x, t) = 0.3\sin(t)x_2^2 + 2 \cos(t)x_2, \Delta g(x, t) = \cos(x_1) \) are fault terms arising at a certain moment of the system operating. The external disturbance term is \( \eta(x, t) = 0.2\sin(x_2) + x_1\sin(2t) \) with a known upper bound \( \eta_0(x, t) = 0.2 + |x_1| \).

For the SMC design the control input is defined by (7) with

\[ u_{eq} = \frac{s - \hat{g}_0(x, t)}{2g_0(x, t)} + \frac{v - f_0(x, t)}{g_0(x, t)} \]  

(50)

\[ u_{eq} = \frac{\eta_0(x, t)}{g_0(x, t)} + \frac{s}{|s| + \varepsilon}. \]  

(51)

The SMC design parameters are: \( s = e_2 + k_1e_1, \quad v = k_1e_2 + \dot{x}_r \) with \( e_1 = x_r - x_1, e_2 = x_r - x_2, k_1 = 2, \) and \( x_r(t) = 2\sin(2t) \) is the reference state; \( \lambda = 4 \) and \( \varepsilon = 0.005 \). The state vector initial value is \( x^0(0) = [2, 0] \).

For the NASMC design a neural network, with 5 neurons and hyperbolic tangent activation functions, is used. All network parameters are initially set to 1. The update laws (19)–(21) are used to adjust the adaptive parameters with \( \gamma_i = 10 \) (\( i = 1, 2, 3 \)). The required bounds are \( \theta_0 = 8, \beta_1 = 3.5 \) and \( a_0 = 2.5 \). The sliding surface is the same as for the SMC design.

The first simulation test concerns the nominal case, i.e., \( \Delta f(x, t) = 0 \) and \( \Delta g(x, t) = 0 \). As depicted in Fig. 3,
the SMC performs very well for the known bound uncertainty. After a transient training phase the NASMC learns rapidly the SMC control strategy, as illustrated by the depicted control inputs. As can be seen from the tracking errors plots, the SMC performs better, which is not surprising since we are in the ideal context, and the NASMC is still in the learning phase.

At the instant $t = 21$ s, a fault operating is introduced, such that $f(x, t) = f_0(x, t) + \Delta f(x, t)$ and $g(x, t) = g_0(x, t) + \Delta g(x, t)$. It is clear from Fig. 4 that SMC cannot cope with this new uncertainty, and its performance is largely degraded as can be remarked from the tracking errors magnitudes. On the other hand, the NASMC is not affected by the fault introduction. This is due certainly to the NASMC ability to absorb unknown nonlinearities. The adaptive parameters of the NASMC converge to bounded values (Fig. 5) as predicted by the theory.
Example 2 We consider the Chua chaotic circuit [17] depicted in Fig. 6, whose dynamic equations are

\[
\begin{align*}
\dot{x}_1 &= \frac{1}{C_1} \left( \frac{1}{R} (x_2 - x_1) - g(x_1) + u_1 \right) + d_1 \\
\dot{x}_2 &= \frac{1}{C_2} \left( \frac{1}{R} (x_1 - x_2) + x_3 + u_2 \right) + d_2 \\
\dot{x}_3 &= \frac{1}{L} (-x_3 + u_3) + d_3
\end{align*}
\]

where \( u_i, d_i \ (i = 1, 2, 3) \) are the control inputs and the external disturbances, respectively. The circuit components are modeled as \( C_i = C_{10} + \Delta C_i, C_2 = C_{20} + \Delta C_2, R = R_0 + \Delta R, L = L_0 + \Delta L \) and \( g(x_1) = g_0(x_1) + \Delta g(x_1) \), where \( C_{10} = 1, C_{20} = 0.5, R_0 = 5, L_0 = 1, g_0(x_1) = -x_1 + 0.02x_1^3 \) are the components nominal values, and \( \Delta C_1 = 0.1 + 0.1 \cos(t/2), \Delta C_2 = 0.1, \Delta R = \sin(t/2), \Delta L = 0.15, \Delta g(x_1) = 0.2 \sin(t)x_1 \) are the components unknown uncertainties.

![Chua chaotic circuit](image)

For the simulation purpose the initial values are \( x^T(0) = [-1 \ 0 \ 0] \) and the reference trajectories are defined by \( x^T(t) = [\sin t \ \cos t \ 1 + \sin t] \). The sliding surfaces are defined as \( s_i = x_{ri} - x_i \ (i = 1, 2, 3) \). The NASMC is composed by three identical neural networks, with 5 neurons in the hidden layer, yielding the three control inputs for the system. The input vectors for the three neural networks are given by \( \mathbf{z}^T(t) = [x^T \ \dot{x}_i \ s_i] \ (i = 1, 2, 3) \). All neural networks parameters are initially set to 1. The update laws (19)–(21) are used to adjust the three neural networks parameters with \( \gamma_{ij} = 10 \ (i, j = 1, 2, 3) \). The required bounds are \( \theta_{0i} = 6, \beta_{hi} = 3 \) and \( a_{hi} = 2 \ (i = 1, 2, 3) \).

As in [17], two simulation tests are performed.

(i) For the first test, the parameters uncertainties are as defined above and the external disturbances are given by

\[
\begin{align*}
d_1 &= \sin(2t) \exp(-0.2t) + 0.3, \\
d_2 &= \cos(2t) \exp(-0.2t) - 0.5, \\
d_3 &= \sin(3t) \exp(-0.2t) + 0.2.
\end{align*}
\]

The test results are depicted in Fig. 7.

(ii) The second test is broken in two steps: For the interval \( 0 \leq t \leq 10 \) s, the parameters uncertainties are as defined above and the external disturbances are given by
\(d_1 = \sin(2t) + 0.3, \ d_2 = \cos(2t) - 0.5, \ d_3 = \sin(3t) + 0.2.\)

For the interval \(t > 10\) s, the parameters uncertainties and the external disturbances magnitudes are doubled. This test results are shown in Fig. 8.

![Fig. 8 Second test NASMC performance](image)

These results illustrate clearly the NASMC tracking performance; the transient learning phase is very short and the tracking errors are rapidly reduced. The second test results indicate perfect rejection of the uncertainties and external disturbances effects. Compared with the results obtained in [17], the two approaches exhibit similar tracking performances; however, the NASMC neural networks architecture is far from the simpler and of reduced complexity which is of a great importance in a practical context. Further, the NASMC generated control inputs are of smaller magnitudes and smoother (especially in the second test) which indicate low solicitations of the control organs. Note that the approach proposed in [17] was shown to be of superior performance compared with the SMC and adaptive PID control approaches.

### 6. Conclusion

In this paper, we propose that adaptive SMC strategy can be emulated by a single neural network, with a special structure. The stability and convergence properties of the NASMC are carried out using the Lyapunov stability tools. Unlike other neural-based strategies, in the NASMC only the adaptive neural network is used as a feedback tool, and no combination with other kind of controllers is needed to ensure the stability or to enhance the control performance. Simulation results show the NASMC’s superiority, as compared with the SMC, in handling unknown uncertainties or faults arising during the plant operating. Further, the NASMC is shown to be applicable to complex chaotic MIMO systems with reduced complexity and moderate control inputs.

### References


**Biographies**

Ghania Debbache was born in Constantine, Algeria, in 1974. She received her B.S. and M.S. degrees in electronics from the Constantine University, Algeria, in 1997 and 2000, respectively. She is currently pursuing the Ph.D. degree on the intelligent control at the University Constantine. Currently, she is a teaching assistant at the Technologic Sciences Institute of Oum El Bouaghi University, Algeria.

E-mail: gdebbache@yahoo.fr

Noureddine Goléa was born in Batna, Algeria, in 1967. He received his B.E. and M.E. degrees from Sétif University, Algeria, and the Doctor from Batna University, Algeria, all in industrial control, in 1991, 1994, and 2000, respectively. From 1991 to 1994, he was with Electronics Institute at Sétif University, and from 1994 to 1996, he was with the Electronics Institute at Batna University. Currently, he is a professor of electrical engineering in the Technologic Sciences Institute at Oum El Bouaghi University, Algeria. His research interests are nonlinear and adaptive control, and intelligent control applied to motion control.

E-mail: nour-golea@yahoo.fr