Chapter 4
Neural-Fuzzy Sliding Mode Control for Uncertain and Nonlinear Systems

In previous Chapters, FSMC and ExFSMC design methods are discussed to minimize chattering and reaching time to the sliding surface. However, controller design for complex nonlinear systems by conventional method is very difficult. To overcome this problem, in this Chapter NFSMC design method is discussed.

4.1 Introduction

In SMC the discontinuous control action moves system dynamics to the desired surface \( s(x) \) from initial state, whereas the continuous control action maintains on it [48]. However, SMC suffers from chattering problem and mathematical computation of continuous control law for complex and not exactly known systems. The FSMC was successful in minimizing the chattering problem but need of precise mathematical model for the design of controller is still remains.

The most of the real world systems exhibits nonlinear dynamics, that induce complexity due to non availability of model structure, exact mathematical model to define the behavior of nonlinearity. For such systems designing a suitable controller to obtain desired performance and specification is very complex and challenging.

The NNs are one of the computational intelligence method. They have inherent ability to learn and approximate a nonlinear function to arbitrary accuracy. This ability of NNs has injected a new driving force into the “control engineering” literature. The NSMC design method exploits the ability of NN to overcome the computational burden of SMC [38–42, 77, 78].
Recently, many research have done fusion of FLC and NNs as fuzzy neural network (FNN) systems. These systems have advantages of both in the field of control systems. According to the learning methods, the FNN can be classified into offline learning and online learning. The offline learning method requires tremendous amount of data to connect weights and membership functions of FNN. The offline trained networks are not adaptable in real time control [79, 80]. The online training can overcome such problems. In the design FNNs, the activation function of NN are realized by fuzzy membership function. Recently, in [78], an adaptive fuzzy wavelet neural controller was proposed to obtain equivalent control term and an adaptive PI controller for implementing switching term. These methods increases difficulty level of controller design and execution time of the control systems. In this Chapter the NNC and FSMC are combined to obtain neural fuzzy sliding mode control for nonlinear system. In this approach, NNC and FLC methods are used to approximate either continuous control or discontinuous control term of SMC to make it mathematical computation free controller design method [11–13]. The proposed controller design method NNC is designed by an output feedback type NNC also called as DNNC. In DNNC method the inputs to the NN are state variables of the plant, control signal, error signal and the reference values are considered. The error signal is the difference between reference value and state variables of the system and the weights of the neural network are updated online using error back propagation algorithm based on gradient algorithm to minimize the cost function of the error. The FSMC design method is used to approximate discontinuous control term by fuzzification of crisp sliding surface and SMC based rule base. The performance of the proposed NFSMC is demonstrated by simulation results using a nonlinear system as compared to ExFSMC, FSMC and modified SMC.

4.2 Basic Idea of Neural network Control

The direct DNNC method is used to design NNC for approximating the value of continuous or equivalent control term $u_{eq}$ of SMC. The architecture of NNC is a feed forward NN with single hidden layer and the output layer having single neuron. The design method of NNC based on DNNC is illustrated in Figure (4.1). The network has particular type of input and output model, which maps input vector $X$ to the output $u_{neq}$. The output of NNC is given as,

$$u_{neq} = \sum_{j=1}^{m} V_{o,j} Y_{out,j}$$

where, $V_{o,j}$ is weight of the connection between output neuron $o$ and $j^{th}$ hidden layer neuron and $Y_{out,j}$ is output of hidden layer neuron and becomes input to the output
Figure 4.1: Direct Neural Network Control

neuron, given as

\[ Y_{out_j} = g(Y_{net_j}) \]

where, \( g(Y_{net_j}) \) is activation function of hidden layer neuron and \( Y_{net_j} \) is net function of inputs to the \( j^{th} \) neuron represented as,

\[ Y_{net_j} = \sum_{i=1}^{n} W_{x_{ji}} X_i \]

where, \( X_i \) is vector of inputs \( \{x_i, r_i, u_i, E_i\} \) and \( x_i \) is state variable vector, \( r_i \) is reference value, \( u_i \) is output of the controller and \( E_i \) is a vector of error signals i.e. difference between state variables \( x_i \) and reference values \( r_i \), and \( W_{x_{ji}} \) is weight of connection between \( j^{th} \) hidden layer neuron and \( i^{th} \) input neuron. The activation function considered here is bipolar binary activation function given as,

\[ g(\sigma) = \frac{2}{1 + e^{-\sigma}} - 1 \]

where, \( \sigma = Y_{net_j} \). The overall output of the neural network is a transformation of its input is given as,

\[ u_{neg} = \Gamma (V (\Gamma (W X_i))) \]

where, \( u_{neg} \) is value of NFSMC and \( \Gamma \) is a nonlinear operator. \( u_{neg} \) is obtained by approximation of its inputs. The approximation property discussed
in [81] [82], is a smooth function of \( u_n \) on a compact set of \( \Omega \). There exists \( n \) hidden layer neurons with weight matrices as \( W \) and \( V \) such that,

\[
u_n(X) = V^T \, g(\, W^T \, X) + \varepsilon(X)
\]

where, \( \varepsilon(X) \) denotes minimum reconstructed error vector satisfying \( ||\varepsilon(X)|| < \varepsilon_n \) for some \( \varepsilon_n > 0 \) and \( V \) and \( W \) are given as

\[
W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\
           & w_{21} & w_{22} & \cdots & w_{2n} \\
           &       & \vdots & \ddots &           \\
           & w_{j1} & w_{j2} & \cdots & w_{jn} \end{bmatrix}
\]

and

\[
V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\
v_{21} & v_{22} & \cdots & v_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
v_{y1} & v_{y2} & \cdots & v_{yn} \end{bmatrix}
\]

Then, an estimate of \( u_n \) can be given by

\[
\hat{u}_n(X) = \hat{V}^T \, g(\, \hat{W}^T \, X)
\]

where, \( \hat{V}^T \) and \( \hat{W}^T \) are the estimations of \( V \) and \( W \), respectively obtained by on-line weight tuning algorithm.

The weights of the neural network are tuned as following,

\[
\dot{\hat{V}} = F_v \, g(\, \sigma \, x) \tag{4.1}
\]

and

\[
\dot{\hat{W}} = G_w \, X(\, \hat{g}^{'}, \, \hat{V}, \, x)^T \tag{4.2}
\]

where, \( F_v \) and \( G_w \) are scalar positive symmetric matrices and \( \hat{g}' = \frac{dg(\sigma)}{d\sigma} \) is the second term of Taylor's series expansion of a function \( g(\sigma) \) [81], [82].

The network is trained by error back propagation algorithm based on gradient descent algorithm to minimize the error \( E \). The error function is

\[
E(w) = \frac{1}{2} \sum_{k=1} c (r_k - x_k)^2,
\]

where, \( x_k \) is state variable, \( r_k \) is reference signal and \( c \) is output vector length of weight \( w \) in the network. The weights are initialized with random values and changed in a direction that will reduce the error \( E \). The weights are updated online from output-to-hidden layer as follows,

\[
\Delta V_{y_j}(t) = \nabla E \frac{\partial E}{\partial V_{y_j}}
\]
where, $\eta$ is a learning rate constant that denotes the learning rate of the back propagation algorithm.

In multilayer NN, the error is an indirect function of the weights in the hidden layer, thus we use chain rule of calculus to calculate the derivatives,

$$\frac{\partial E}{\partial V_{y_j}} = \frac{\partial E}{\partial \text{o}_\text{net}} \times \frac{\partial \text{o}_\text{net}}{\partial V_{y_j}}$$

and

$$\frac{\partial E}{\partial W_{x_{ji}}} = \frac{\partial E}{\partial \text{Y}_{\text{net}_j}} \times \frac{\partial \text{Y}_{\text{net}_j}}{\partial W_{x_{ji}}}$$

where, $\partial E/\partial \text{o}_\text{net} = -\delta_o$, and $\partial E/\partial \text{Y}_{\text{net}_j} = -\delta_y$ can be derived as follows:

$$\delta_o = \frac{1}{2}(r - x_i)(1 - x_i^2)Y_{\text{out}_j}$$

$$\delta_y = V_{y_j}\delta_o g'(\sigma)$$

where,

$$g'(\sigma) = Y_{\text{out}_j}(1 - Y_{\text{out}_j})$$

and $\partial Y_{\text{net}_j}/\partial W_{x_{ji}} = X_i$, $\partial \text{o}_\text{net}/\partial V_{y_j} = 1$.

The weights of hidden and output layer of a NN are updated as follows,

$$W'_{x_{ji}}(t) = W_{x_{ji}}(t - 1) + \eta \delta_y X_i$$

$$V'_{y_j}(t) = V_{y_j}(t - 1) + \eta \delta_o$$

4.2.1 Neural-Fuzzy Sliding Mode Control for Uncertain Non-linear Systems

The control logic of NFSMC is shown in Figure (4.2). In this approach, the NNC is designed based on the DNNC method to approximate the equivalent or continuous control term is called as neural sliding mode control, and the discontinuous term is approximated using FSMC method. Combined output of NSMC and FSMC is called as NFSMC. The design steps are based on methods discussed in Section 2.2 and 4.2.

The control structure of NFSMC has the form,

$$u_{nf} = \text{uneq} - u_{fn}$$

The $\text{uneq}$ is obtained by NNC,

$$u_{\text{uneq}} = \sum_{j=1}^{m} V_{y_j} Y_{\text{out}_j}$$

(4.3)
$un_{eq}$ is a function of its input vector $X_i = \{x_i, r_i, u, E_i\}$, where $x_i$ is vector of state variables of a system, $r_i = 0$ is reference value for regulation problem, $u$ is controller output and $E_i$ is an error vector of the system. The network is trained to minimize the cost function $E_i$, by Error back propagation algorithm based on gradient descent algorithm to minimize the error function $E_i(w)$.

The $un_f$ is obtained using design method of FSMC as given in Section 2.2.4. The FSMC output $un_f$ is obtained by considering the sliding surface $s$ as input to the FLC and the output membership function are defined on $un_f = un_{eq} \pm K$. The general fuzzy control rule for NFSMC is,

$$R^i : \text{if } s \text{ is } F^i, \text{ then } unf \text{ is } F^i_{un_f}$$

where, $uf$ is control input. The sup-min compositional rule of inference is,

$$\mu_{sof}(un_f) = \sup_{x} \{ \min \left[ \mu_{F^i_x}(s), \min \left[ \mu_{F^i_y}(s), \mu_{un_f}(un_f) \right] \right] \}.$$ 

The NFSMC output has the form,

$$un_f = NFSMC(s) = -K \text{ sat}(s/\Phi)$$

### 4.2.2 Design steps of Neural Fuzzy Sliding mode Control

1. Design sliding surface $s$ for the systems.

2. The equivalent control $un_{eq}$ for the given system is approximated by NNC.

3. Design input and output fuzzy membership functions on the universe of discourse of $s$ and $un_{eq} \pm K$.

4. Define the inference engine as (2.13).

5. Define the rule base as,

   (a) if $s$ is NB then $un_f$ is Bigger.

---

Figure 4.2: Neural-Fuzzy Sliding Mode Control strategy
(b) if $s$ is NM then $u_{nf}$ is Big.
(c) if $s$ is ZE then $u_{nf}$ is Medium.
(d) if $s$ is PM then $u_{nf}$ is Small.
(e) if $s$ is PB then $u_{nf}$ is Smaller.

6. Obtain $u_{nf}$ using (2.14) and overall control law using has the form

$$u = u_{nf} = u_{neq} - K \text{sat}(s/\Phi)$$

The stability of system for proposed control input can be proved using Theorem 2.2.1.

### 4.3 Simulation studies

The performance of the NFSMC is illustrated by simulation of two nonlinear systems.

#### 4.3.1 Example 1

To test the performance of proposed control methodology, simulation test has been carried out on a nonlinear system (2.3.1) in section 2.3. The sliding surface and equivalent control and initial values are consider same as in section 2.3. The NNC is design by considering, the input vector as $\{x_1, x_2, r, E_1, E_2 \text{ and } u\}$, where, $r = 0$ is reference value, $E_1 = r - x_1$ and $E_2 = r - x_2$, are errors and $u$ is out of the NFSMC.

The simulation is carried out by considering the design parameters for the NFSMC as $k_{eq} = 0.002$, $K = 12$, $\Phi = 3$. The dynamic behavior of state variable $x_1(t)$, control input $u$ and sliding surface $s$ are obtained using NFSMC are shown in Figures (4.3) and (4.5).

![Figure 4.3: Response of state variable of uncertain system to NFSMC.](image)

From Figures (4.3-4.5) it can be seen that, the proposed control method was successful in stabilizing the system with smaller overshoot as compared to modified SMC, FSMC.
and ExFSMC. The NFSMC method shows better performance.

### 4.3.2 Example 2

To test the performance of proposed control methodology, simulation test has been carried out on a nonlinear inverted pendulum example 2.3.2 discussed in section 2.3. The initial values are consider same as in section 2.3. The NNC is design by considering, the input vector as \{x_1, x_2, r, E_1, E_2\} and \(u\), where, \(r = 0\) is reference value, \(E_1 = r - x_1\) and \(E_2 = r - x_2\), are errors and \(u\) is out of the NFSMC.

The simulation is carried out by considering the design parameters for the NFSMC as \(k_{eq} = 0.002\), \(K = 8\), \(\Phi = 3\). The results are summarized as follows: The simulation results of NFSMC, ExFSMC, FSMC and modified SMC are shown in Figures (4.6–4.8), of state variable, control input and sliding surface dynamics of the system. From figures it is observed that, the NFSMC shows faster response than other three methods. The NFSMC method has high control effort compared to other methods.

The qualitative analysis of proposed method compared to ExFSMC, FSMC and modified SMC based on \(ISE\), second norm of error and reaching time is shown in Table (4.1). From the analysis results, it is observed that, the NFSMC has faster reaching
time and smaller $ISE$.

## 4.4 Conclusion

In this Chapter, a computational intelligence based SMC design method is discussed for uncertain nonlinear systems. In the controller design approach, the two terms of SMC, that is, continuous control (equivalent control) and discontinuous control are
approximated by NNC and FLC respectively. The NNC is designed to approximate continuous control using DNNC method. In DNNC, the network inputs will be the state variables of the plant, reference values, errors and the output of the controller. Considering, error as the difference between reference value and state variable of the system, the weights of the NN are updated on line using the error back propagation algorithm based on gradient algorithm. The discontinuous control is approximated using FSMC method. The overall controller is called as NFSMC. The performance of the proposed controller is investigated on two nonlinear systems and compared with ExFSMC, FSMC and modified SMC methods. From the results, its concluded that, the NFSMC approximates the conventional control method and holds the good control property. Looking at the controller design complexity and program execution time required for single input single output nonlinear systems, the ExFSMC is simple in design and requires less time for program execution.