A PSO Based Decentralized Neuro-Fuzzy-Sliding Mode Controller for the Twin Rotor MIMO system

Fouad Allouani, Djamel Boukhetela and Fares Boudjema

Abstract-In this paper, a decentralized neuro-fuzzy-sliding mode controller (NFSMC) with particle swarm optimization algorithm (PSO) is proposed to position the beam of the Twin Rotor MIMO System (TRMS) at the desired positions quickly and accurately. Moreover, a synergistic combination of neural networks (NNs) with sliding mode control (SMC) methodology is presented. In the present approach, in each sub-NFSMC (the first sub-controller is used to control the vertical sub-TRMS and the second used for the horizontal sub-TRMS) a feed-forward NN is used to compute the equivalent controller. In addition, a Fuzzy logic controller (FLC) with a Mamdani fuzzy inference system is used as an approximation mechanism to compute the corrective term in the SMC. The weights of each sub-net are updated such that the corrective control term of each sub-NFSMC goes to zero. Moreover, the PSO algorithm is adopted to tune some parameters of each sub-NFSMC which are: the learning rate parameter of the backpropagation algorithm (BPA) used to establish the multilayer feed-forward networks, the slopes of the switching surfaces used as inputs of each sub-FLC and output NN gains. Finally, simulation results demonstrate that the controller designed can greatly alleviate the chattering effect and provide a very good accuracy and a very good response time, it also indicates that better performance can be achieved with this controller compared to the conventional SMC in the aspect of accuracy and control signals energy.

Keywords: Fuzzy control, neural networks, particle swarm optimization (PSO), sliding mode, TRMS.

1. INTRODUCTION

It is all well known that, because of the great robustness to system parameter perturbations and external disturbances, variable structure systems (VSS) are widely applied in many engineering problems. SMC as a special class of variable structure control has received more attention among control research communities. Recently, this method has found several applications in various important fields such as: robot manipulators [1], building structure [2], underwater vehicles [3] and complex systems [4]. Basically, the SMC design is composed of two stages. The first stage is to define a sliding surface on which the controlled system dynamics is restricted with accordance to some performance criterion. Then, the second stage is to design a discontinuous feedback control law such that any system trajectory outside the sliding surface is driven to reach the surface in a finite time and keep on it. When sliding mode is realized, the closed-loop SMC system becomes robust to matched uncertainties and external disturbances. However, in applications of practical control, SMC suffers from two main disadvantages. The first one is high frequency oscillations of the state trajectories around the sliding manifold known as chattering phenomenon. The second disadvantage is that it is difficult to obtain parameters of the
system and consequently the difficulty encountered in the calculation of the equivalent control.

In the literature, a number of methods have been proposed to reliably prevent chattering. The most popular technique for the elimination of this phenomenon consists in replacing the sign function with smooth ones such as saturation functions, hysteresis and hysteresis with saturation [5]. This approach, however, provide no guarantee of convergence to the sliding mode and involve a trade-off between chattering and robustness. Elimination of chattering may be achieved without decreasing robust performance by combining SMC with the attractive features of fuzzy control [6].

In general, a NN controller with the learning rule based on sliding mode algorithm, is used to assure calculation of unknown part of the equivalent control in the presence of plant uncertainties [7],[12]. The nonlinear mapping and learning properties of NNs are key factors that make this controller possesses the features of robustness under parameter variation and external disturbance.

This paper presents an approach of cooperative control based on the combination of fuzzy control, NNs, the methodology of sliding mode control and a heuristic optimization algorithm. In order to eliminate the chattering phenomenon, a fuzzy logic controller (FLC) is used to approximate the corrective control term. In this context, the FLC is adapted to work in a fixed boundary layer. Outside the boundary, the SMC is applied to drive the system states into the boundary layer. To compute the equivalent control, a layer feed-forward NN is designed and its weights are adapted to minimize the square of the corrective control [7].

PSO first introduced by Kennedy and Eberhart [8], is one of the modern heuristic algorithms. It mimics the behavior of individuals in a swarm to maximize the survival of the species. The PSO technique can generate a high-quality solution within short calculation time and stable convergence characteristic, it is characterized also by the following main advantages: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with other mathematical algorithms and other heuristic optimization techniques [8]. In this study, the PSO algorithm is used to adjust the slopes of the switching surfaces, the learning rates and the output NN gains to improve firstly the rate of convergence on the sliding surfaces and secondly the learning capability in the NNs and to control the maximum values of the equivalent controls generated by NNs.

Because of its highly nonlinear behavior and elevate coupling effect between the two propellers vertical and horizontal subsystems, the control problem of the TRMS has been considered as a challenging research subject [13],[14],[15]. Moreover, the control of the TRMS has attracted a lot of attention because the dynamic of this system and a real helicopter are similar in certain aspects. All these characteristics qualify this aero-dynamical system to be a very good example to validate our approach.

The rest of the paper is organized as follows: Section 2 describes the TRMS; a decentralized NFSMC based on the PSO algorithm is designed in Section 3; Simulation analysis and results are provided in Section 4; Finally, Section 5 is devoted to conclusion.
2. TRMS DESCRIPTION

The TRMS is a laboratory setup resembling a flight control system as shown in figure 1 [9]. It is characterized by complex and highly nonlinear functions with some inaccessible parameters for measurements.

![Figure 1 Twin Rotor Multi-Input Multi-Output System.](image)

The TRMS consists of a beam pivoted on its base in such a way that it can rotate freely both in the horizontal and vertical planes. At both ends of the beam, the rotors (the main and tail rotors) are driven by dc motors. A counterbalance arm with a weight at its end is fixed to the beam at the pivot to make the TRMS stabilizable. The state of the beam is described by four process variables: horizontal and vertical angles ($\alpha_h$ and $\alpha_v$) measured by position sensors fitted at the pivot, and two corresponding angular velocities. In a real helicopter, the aerodynamic force is controlled by changing the angle of attack of the blades. However, in the TRMS the pitch angle of the blades is fixed and the aerodynamic force is controlled by varying the speed of the motors. Therefore, the control inputs are the supply voltages to the dc motors. A change in the voltage value results in a change in the rotation speed of the propeller. This further results in a change of the corresponding position of the beam.

From the control point of view, TRMS exemplifies a high order nonlinear system with significant cross coupling effects. The control objective is to stabilize the system in a coupled condition and make its beam move quickly and accurately to track a trajectory or to reach specified positions. Since the TRMS permits both 1- degree of freedom (1-DOF) and 2-DOF motion, it would be convenient to design a controller for this system after its decomposition into subsystems ($s_h$ horizontal subsystem and $s_v$ vertical subsystem). Moreover, the cross coupling effects between these two subsystems are considered as the uncertainties and disturbances.

Approximate decoupled mathematical model of the TRMS is obtained by using Newton’s second law of motion and is converted into the state space form as given below [14], [15].

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{J_h} \left[ l_c S_f F_h(\omega_r) \cos(x_4) - k_h x_2 - x_2 x_4 (D - E) \sin(2x_4) - J_m \omega_m x_5 \sin(x_4) + \frac{J_m}{T_{mr}} (u_2 - x_6) \frac{d \omega_m(x_6)}{dx_6} \cos(x_4) \right] \\
\dot{x}_3 &= \frac{1}{T_{tr}} u_1 - \frac{1}{T_{tr}} x_3
\end{align*}
\] (1)
\[
\begin{align*}
X_4 &= x_5 \\
\dot{x}_5 &= 9.1[l_m S_f F_p(\omega_m) - g(0.0099\cos(x_4) + 0.0168\sin(x_4) - k_v x_5] \\
&\quad -0.0252 x_2^2 \sin(2 x_4) + \frac{\omega_r}{t_{mr}} (u_1 - x_3) \frac{d\omega_r(x_3)}{dx_5}
\end{align*}
\]

Where \( x_1 = \alpha_v \) the yaw angle, \( x_2 = \alpha_p \) is the pitch angle, \( x_3 = \Omega_h \) is the angular velocity around the horizontal axis, \( x_4 = \Omega_v \) is the angular velocity around the vertical axis, \( x_5 = i_h \) is the armature current of the tail propeller subsystem and \( x_6 = \Omega_v \) is the armature current of the main propeller subsystem.

We have also:

\[
\begin{align*}
j_h &= D \sin^2(x_2) + E \sin^2(x_2) + G \\
\omega_m(x_4) &= 90.99x_4^5 + 599.73x_4^5 - 129.26x_4^4 - 1238.64x_4^3 + 63.45x_4^2 + 1283.41x_4 \\
\omega_1(x_3) &= 2020x_3^5 + 194.69x_3^4 - 4283.15x_3^3 - 262.27x_3^2 + 3769.83x_3 \\
F_h(\omega_r) &= -3 \times 10^{-14} \omega_r^5 + 1.595 \times 10^{-11} \omega_r^4 - 2.511 \times 10^{-7} \omega_r^3 - 1.808 \times 10^{-4} \omega_r^2 \\
&\quad + 8.01 \times 10^{-2} \omega_r \\
F_p(\omega_m) &= -3.48 \times 10^{-12} \omega_m^5 + 1.09 \times 10^{-9} \omega_m^4 + 4.123 \times 10^{-6} \omega_m^3 - 1.632 \times 10^{-4} \omega_m^2 \\
&\quad + 9.544 \times 10^{-2} \omega_m
\end{align*}
\]

Table 1 shows the list of physical parameters and their values:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_m )</td>
<td>Length of the main part of the beam</td>
<td>0.236m</td>
</tr>
<tr>
<td>( l_t )</td>
<td>Length of the tail part of the beam</td>
<td>0.25m</td>
</tr>
<tr>
<td>( k_v )</td>
<td>Friction coefficient of the vertical axis</td>
<td>0.0095</td>
</tr>
<tr>
<td>( k_h )</td>
<td>Friction coefficient of the horizontal axis</td>
<td>0.0054</td>
</tr>
<tr>
<td>( J_{mr} )</td>
<td>Inertia of the main propeller dc motor</td>
<td>( 1.6543 \times 10^{-5} \text{kgm}^2 )</td>
</tr>
<tr>
<td>( J_{tr} )</td>
<td>Inertia of the tail propeller dc motor</td>
<td>( 2.65 \times 10^{-5} \text{kgm}^2 )</td>
</tr>
<tr>
<td>( T_{mr} )</td>
<td>Time constant of the main rotor</td>
<td>1.432</td>
</tr>
<tr>
<td>( T_{tr} )</td>
<td>Time constant of the tail rotor</td>
<td>0.3842</td>
</tr>
<tr>
<td>( D )</td>
<td>Mechanical related constant</td>
<td>( 1.6065 \times 10^{-3} \text{kgm}^2 )</td>
</tr>
<tr>
<td>( E )</td>
<td>Mechanical related constant</td>
<td>( 4.90092 \times 10^{-2} \text{kgm}^2 )</td>
</tr>
<tr>
<td>( G )</td>
<td>Mechanical related constant</td>
<td>( 6.3306 \times 10^{-3} \text{kgm}^2 )</td>
</tr>
<tr>
<td>( S_f )</td>
<td>Balance scale</td>
<td>( 8.43318 \times 10^{-4} )</td>
</tr>
<tr>
<td>( g )</td>
<td>Gravitational constant</td>
<td>( 9.81 \text{m/s}^2 )</td>
</tr>
</tbody>
</table>

Table 1 Physical parameters of the TRMS [15]

3. DESIGN OF THE DECENTRALIZED NEURO-FUZZY-SLIDING MODE CONTROLLER BASED ON THE PSO ALGORITHM FOR THE TRMS

3.1 Design of SMC

SMC is a method derived from Variable Structure Control (VSC) which was originally studied by [10]. The controller designed using this technique is particularly appealing due to its ability to deal with nonlinear systems and time varying systems. SMC proved their high accuracy and robustness with respect to various internal and external disturbances. The basic idea of this method is to force the system, after a finite time reaching phase, to a sliding surface \( s(x,t) \) containing the system operating point and defined to represent a
desired global behavior for instance stability and tracking performance. The $s(x, t)$ selected in this work is presented by [11]

$$s(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} e(x, t)$$  \hspace{1cm} (3)

Where $\lambda$ is a positive constant.

It is clear from (3) that keeping the states of the system on the sliding surface will guarantee the tracking error vector $e(x, t)$ asymptotically reaching to zero. The corresponding sliding condition [11] is

$$\dot{V}(t) = \frac{1}{2} \frac{d}{dt} s^2(x, t) = \dot{s}(x, t)s(x, t) < \rho |s(x, t)|$$  \hspace{1cm} (4)

Where $V(t)$ indicates the Lyapunov function and $\rho$ is a strictly positive constant. So, this condition implies that the squared “distance” from the sliding surface $s^2(x, t)$ decreases along all system trajectories. Thus, the output of the designed controller $u$ that is called equivalent control law $u_{eqv}$ which satisfies the sliding condition and makes the sliding surface an invariant set. However, in order to cancel out the uncertainties and disturbances, a discontinuous term must be added to $u_{eqv}$.

$$u = u_{eqv} - k \times sgn(s(x, t))$$  \hspace{1cm} (5)

Where $u_{eqv}$ is $sgn()$ is the sign function and $k$ is a constant parameter.

However, a discontinuous part of the control law results in an undesirable phenomenon, known as chattering. When the implementation of the associated control switching is imperfect, chattering will be seen in output of the system (i.e., measured variable).

Thus, as the relative degree is $n = 3$ for the TRMS vertical subsystem and for the TRMS horizontal subsystem, we obtain the following two sliding surfaces:

$$s_v(x, t) = \lambda_v^2 e_v + 2\lambda_v \frac{de_v}{dt} + \frac{d^2 e_v}{dt^2}$$  \hspace{1cm} (6)

$$s_h(x, t) = \lambda_h^2 e_h + 2\lambda_h \frac{de_h}{dt} + \frac{d^2 e_h}{dt^2}$$  \hspace{1cm} (7)

Where, $e_v(t)$ and $e_h(t)$ is the pitch angle error and the yaw angle error respectively.

3.2 Structure of the proposed decentralized NFSMC

The global decentralized controller includes two sub-NFSMC each of them comprises of two parts, the first one computes the corrective term in the SMC using a FLC as an approximation mechanism, and the second one calculates the equivalent control term in SMC by a NN [7], [12]. The corrective control term generated by the FLC is summed with the output of NN to construct the control signal. The corrective control is used as a measure of the error to update the weights of the NN. The proposed adaptation scheme directly results in chattering free control action for the corrective control. Furthermore, the PSO is used to fined the following sub-NFSMC parameters : the learning rate parameter of the BPA used to establish the multilayer feedforward network, the output NN gains and the
switching surface coefficients (slopes); to improve firstly the network learning ability and increase the speed of network convergence, to control the maximum values of the equivalent controls generated by each NN, and secondly to enhance the rate of convergence on the sliding surface. The overall system with the proposed controller and the internal configuration of each sub-NFSMC are illustrated in figure 2(a) and figure 2(b).

Figure 2 (a) Overall structure of the decentralized NFSMC used to control TRMS and the internal configuration of each sub-NFSMC

3.3 Computation of the corrective control term

The SMC depends on the sign of the switching function. The ideal sliding mode exists in high frequency oscillations of the control signal between two limit values. In practice, it would be impossible to carry out commutations at infinite frequencies, which induces a non-ideal sliding mode that generates the undesirable chattering phenomenon. Therefore, to eradicate chattering, two single-input-single-output (SISO) FLC (vertical and horizontal sub-controllers) are used to approximate the corrective control term in SMC. The design of the fuzzy controller begins with extending the crisp sliding surface $\tilde{s} = 0$ to the fuzzy sliding surface defined by a linguistic expression [16]:

\[
\tilde{s} \text{ is ZERO}
\]

Where $\tilde{s}$ is the linguistic variable for $s$ and ZERO is one of its fuzzy set. In order to partition the universe of discourse of $s$, the following fuzzy sets of $s$ are introduced:

\[
T(\tilde{s}) = \{NB, NM, ZR, PM, PB\} = \{F_1^s, \ldots, F_5^s\}
\]

Where $T(\tilde{s})$ is the term set of $\tilde{s}$, and $NB$, $NM$, $ZR$, $PM$ and $PB$ are labels of fuzzy sets, which are negative big, negative medium, zero, positive medium, and positive big, respectively. For the control output $u_{Fuzzy}$, its term set and labels of the fuzzy sets are defined similarly by:

\[
T(\tilde{u}_{Fuzzy}) = \{NB, NM, ZR, PM, PB\} = \{F_1^{u_{Fuzzy}}, \ldots, F_5^{u_{Fuzzy}}\}
\]
The Gaussian membership functions were used for both input and output of FLC. Gaussian
membership function is defined as:

\[ \mu_{\text{Gaussian}} (x) = \exp \left( \frac{-(x-a)^2}{2b^2} \right) \]  

(11)

Where subscript \( Nfs \) represents the total number of fuzzy sets, \( a_p \) and \( b_p \) are respectively,
the center and the width of the Gaussian membership function, they are determined by the
PSO. Moreover, in figure 3 the term \( \phi \) indicates the boundary layer around the switch surface.

From these two term sets, we can build the following fuzzy rules [16]:

- **R1**: If \( s \) is \( NB \), then \( u_{\text{Fuzzy}} \) is \( PB \).
- **R2**: If \( s \) is \( NS \), then \( u_{\text{Fuzzy}} \) is \( PS \).
- **R3**: If \( s \) is \( ZR \), then \( u_{\text{Fuzzy}} \) is \( ZR \).
- **R4**: If \( s \) is \( PS \), then \( u_{\text{Fuzzy}} \) is \( NS \).
- **R5**: If \( s \) is \( PB \), then \( u_{\text{Fuzzy}} \) is \( NB \).

The result of the inference for every \( \sigma \) can be written as follows:

\[ u_{\text{Fuzzy}} = -K \times \text{sig} \left( \frac{s}{\phi} \right) \]  

(12)

Where

\[ \text{sig}(z) = \begin{cases} 
-1 & z < -1 \\
1 & z > A 
\end{cases} \]  

(13)

The figure 4, show the results of the inference of the fuzzy rules. From this figure, we can
see easily that the shape of this function is a saturation function.
3.3 Computation of the equivalent control term

The structure of each NN (vertical and horizontal NN) used to compute the equivalent controls $u_{eqv}$ and $u_{eqh}$ is selected as a three-layer feed-forward NN, with one hidden layer, one input layer and one output layer. The inputs and outputs of the network are dictated by the equivalent control equation. The structure of NN used to generate $u_{eqv}$ is presented in figure 5.

The symbols used in figure 5 are defined as follows: Let $Z_i$ be the input to the $i$th node in the input layer, $Y_{net,j}$ be the input to the $j$th node in the hidden layer, and the output of the hidden layer be $Y_{out,j}$. Similarly, the input and output of the output layer are designated as $U_{net}$ and $U_{out}$, respectively. Moreover, $W_{z_{ij}}$ means the weight between the input layer and the hidden layer, $W_{y_{ij}}$ means the weight between the hidden layer and the output layer. So, the values of $u_{eq}$ can be computed as [7], [12]:

$$Y_{net,j} = \sum_{i=1}^{N_i} W_{z_{ij}} Z_i, \quad i = 1, ..., N_L$$

(14)
The activation function $g(x)$ of the net is selected as a sigmoid transfer function, defined by (19). The number $N_L$ represents the total number of the input layer, and $M_H$ represents the total number of the hidden neurons. $k_{eq}$ is a constant that represents the maximum available value of the equivalent control (output of the network). In our case, there are two $k_{eq}$ parameters one for the vertical NN and one for the horizontal NN. Hence, $\hat{u}_{eq}$ represents the estimated value of the equivalent control. Generally, in the NNs the backpropagation method uses the iterative gradient descent algorithm to establish the multilayer feed-forward network. The weight adaptation is based on a minimization of the mean square error as a cost function that is selected as the difference between the desired and the estimated equivalent control. Therefore, a simple cost function is defined as follows:

$$E = \frac{1}{2} [u_{eq}(t) - \hat{u}_{eq}(t)]^2$$

(20)

The gradient descent method is used to update the weights of NN. The weights are then updated by using the following two equations:

$$W_{yj}(t) = W_{yj}(t - 1) - \eta \frac{\partial E}{\partial W_{yj}}$$

(21)

$$W_{zlj}(t) = W_{zlj}(t - 1) - \eta \frac{\partial E}{\partial W_{zlj}}$$

(22)

Where $\eta$ is the learning rate parameter of the BPA and it is a constant. Furthermore, the update terms in (21) and (22) can be derived as follows:

$$\frac{\partial E}{\partial W_{yj}} = -\frac{1}{2} (u_{eq} - \hat{u}_{eq}) k_{eq} (1 - U_{out}^2) Y_{outj}$$

(23)

$$\frac{\partial E}{\partial W_{zlj}} = -\frac{1}{4} (u_{eq} - \hat{u}_{eq}) k_{eq} (1 - U_{out}^2) W_{yj} (1 - Y_{outj}^2) Z_i$$

(24)

From (23) and (24), we find that the actual equivalent control is unknown. Hence, (23) and (24) cannot be calculated. This problem can be solved by using the value of corrective control to replace the error between desired and estimated equivalent control. The reason is that the characteristics of this error and corrective control are similar [7][12].

$$\frac{\partial E}{\partial W_{yj}} = -\frac{1}{2} u_{Fuzzy} k_{eq} (1 - U_{out}^2) Y_{outj}$$

(25)
\[ \frac{\partial E}{\partial W_{i,j}} = -\frac{1}{4} u_{\text{Fuzzy}} k_{eq} (1 - u_{\text{out}}^2) W_{y_j} \left( 1 - Y_{\text{out}}^2 \right) Z_i \]  

\[ (26) \]

### 3.4 Tuning parameters of the Decentralized NFSMC by PSO

Similar to GA, PSO is a population based optimization tool, it is inspired by social behavior among individuals. This algorithm has been found to be robust in solving problems featuring nonlinearity and non-differentiability, multiple optima, and high dimensionality through adaptation. PSO is basically developed through simulation of bird flocking and fish schooling (simulating animals' social activities). It attempts to mimic the natural process of group communication to share individual knowledge when such swarms flock, migrate, or hunt. If one member sees a desirable path to go, the rest of this swarm will follow quickly. In this optimization method, this behavior of animals is imitated by particles with certain positions and velocities in a searching space, wherein the population is called a swarm, and each member of the swarm is called a particle.

Starting with a randomly initialized population and searches for optima by continually updating this population, each particle (potential solution) in PSO moves through the searching space and remembers the best position it has seen. Members of a swarm communicate good positions to each other and dynamically adjust their own position and velocity based on these good positions. The velocity adjustment is based upon the historical behaviors of the particles themselves as well as their neighbors. In this way, the particles tend to fly towards better and better searching areas over the searching process.

The vectors and operators for PSO algorithm is represented as follows:

- **Particle –** \( X(t) \): In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each particle in the swarm flies in the problem space looking for the optimal position to land. A particle, as time passes through, adjusts its position according to its own “experience”, as well as according to the experience of neighboring particles. A particle represents a candidate solution for the problem. Each particle is treated as a point in the \( D \)-dimensional problem space. The particle is represented as:

  \[ X(t) = (x_{j1}(t), x_{j2}(t), \ldots, x_{jd}(t)) \]

  where \( D \) represents also the number of optimized parameters.

- **Swarm –** \( S(t) \): It is a set of \( N_p \) particles. \( S(t) = \{ X_1(t), X_2(t), \ldots, X_{N_p}(t) \} \).

- **Particle best –** \( pbest(t) \): The best position that is associated with the best fitness encountered so far is called the particle best \( pbest(t) \).

- **Global best –** \( gbest(t) \): It is the best position among all of the particle best positions achieved so far.

- **Particle velocity –** \( V(t) \): It is the velocity of the moving particles represented by \( D \)-dimensional real-valued vector \( V(t) = [v_{j1}(t), v_{j2}(t), \ldots, v_{jd}(t)] \).

The current velocity of the \( dth \) dimension of the \( jth \) particle at time \( t \) is

\[ v_{jd}(t + 1) = w(t) \times v_{jd}(t) + c_1 r_1 (pbest_{jd}(t) - x_{jd}(t)) + c_2 r_2 (gbest_d(t) - x_{jd}(t)) \]

\[ (27) \]

Where, \( c_1 \) and \( c_2 \) are positive constants, called the acceleration constant, \( r_1, r_2 \) are random numbers, uniformly distributed within the interval \([0, 1]\), and \( t \) is indicates the iteration index (generation).
• **Inertia weight** – \( w(t) \): It is a tuning parameter that is used to control the exploration and exploitation abilities of the swarm and as mechanism to eliminate the need for velocity clamping. We allow for changes of the weight over the course of the optimization process; these changes are governed by the following expression (here \( w_{\text{max}} \) and \( w_{\text{min}} \) are the predetermined boundary values the inertia weight can assume and \( \text{iter}_{\text{max}} \) is the maximum number of generations) [17]:

\[
w(t) = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{\text{iter}_{\text{max}}} \times t
\]

• **Maximum velocity** – \( v_{\text{max}} \): The velocity is restricted to the \([-v_{\text{max}}, +v_{\text{max}}]\) range in which \( v_{\text{max}} \) is a predefined boundary value. The value of \( v_{\text{max}} \) determines the resolution of the search regions between the present and target position. Eberhart and Shi [18] suggested that \( v_{\text{max}} \) be set at about 10 – 20% of the dynamic range of the variable in each dimension.

• **Stopping criteria**: These are the conditions under which the search process terminates.

In this paper, the PSO is used to find appropriate values of each-NFSMC (vertical or horizontal) parameters which are respectively: the output NN gain \( k_{\text{eqv}} \) and \( k_{\text{eqh}} \), the learning rate parameters \( \eta_{v} \) and \( \eta_{h} \) and the slopes of the switching surfaces \( \lambda_{v} \) and \( \lambda_{h} \). Then, we have in sum six parameters to be optimized.

The process of optimization decentralized NFSMC parameters utilizing PSO is depicted as following:

**Step (1) Initialization** – Specify the lower and upper bounds of all NFSMC parameters to be optimized (see Table 1), the number of particle (swarm size \( N_{p} \)) and number of iteration as termination criterion (in here, \( N_{p} = 5 \) particles and \( \text{iter}_{\text{max}} = 50 \) iterations are applied). Randomly generate an initial population that contains \( N_{p} \) particles (population size) from certain search interval. Similarly, generate randomly initial velocities of all particles in \([-v_{\text{max}}, +v_{\text{max}}]\). Each particle in the initial population is evaluated using the following objective function:

\[
F = \int [ |e_{h}(t)| + |e_{v}(t)| + |u_{h}(t)| + |u_{v}(t)| ] dt
\]

Search for the best value of the objective function \( p_{\text{best}} \). In initial \( p_{\text{best}} \), \( g_{\text{best}} \) is set as the best value of the objective function.

**Step (2) Inertia weight and velocity updating** – Calculate the inertia weight and \( j \) th Particle velocity using (28) and (27), respectively. Check for the maximum velocity.

**Step (3) Position updating** – In PSO algorithm (general case), based on the updated velocities, each particle changes its position according to the following expression:

\[
x_{jd}(t) = v_{jd}(t) + x_{jd}(t - 1)
\]

Where, \( x_{d}^{\text{min}}(t) \leq x_{jd}(t) \leq x_{d}^{\text{max}}(t) \)

**Step (4) Individual & global best updating** – Each particle \( j \) is evaluated according to the updated position. Reset the \( p_{\text{best}} \) in comparison with previous \( p_{\text{best}} \) through fitness of objective function. Update the \( g_{\text{best}} \) in comparison with the best \( p_{\text{best}} \).
4. SIMULATION RESULTS

The control objective is to force the pitch angle $\alpha_v$ and the yaw angle $\alpha_h$ to follow a given reference trajectory $\alpha_{v\text{ref}}$ and $\alpha_{h\text{ref}}$ respectively. Therefore, the inputs of each NN (designated as Z) consisted of the desired states and actual states, as $Z_{v/h} = [\alpha_{v\text{ref}/h\text{ref}} \alpha_{v/h}]$ and all the network weights for the two NNs were initialized to small random values between $[-0.001, 0.001]$. In order to control TRMS to track a specified trajectory quickly and accurately, the parameters of the each sub-NFSMC are tuned on-line using the PSO algorithm. Simulations of tracking control with a desired reference wave in vertical and horizontal planes are shown in figure 7. The expressions of these reference signals are given as follows:

$$\alpha_{v\text{ref}}(t) = 0.3[\cos(0.23\pi t - 0.015\pi) + 0.3 \sin (0.45\pi t)] \text{ (rad)}$$ (31)

$$\alpha_{h\text{ref}}(t) = 0.2[\cos(0.55\pi t - 0.015\pi) + 0.2 \sin (0.45\pi t)] \text{ (rad)}$$ (32)

The control signals $u_v$ and $u_h$ and the tracking errors $e_v$ and $e_h$ are also illustrated in figure 8 and figure 9. The figure 6 illustrates the best cost function (for the best particle, which is particle number $10/i = 10$) evolution in term of the number of iterations, the final value of the cost function $F$ to which converge the PSO algorithm is 0.8001. Simulation experiments using SMC had been also studied to demonstrate the effectiveness of the proposed method. The system responses $\alpha_v$ and $\alpha_h$ and the output control signals $u_v$ and $u_h$ obtained using SMC with the same tracking control tests used before are illustrated in figure 10 and figure 11. The SMC parameters $\lambda_v$, $\lambda_h$, $k_{SMCv}$ and $k_{SMCh}$ using in this test are set respectively to: 16.3936, 9.9605, 1 and 1. The coefficients of PSO algorithm are chosen as follows: the constants $c_1 = 0.1$, $c_2 = 0.2$, $w_{\max} = 0.9$, $w_{\min} = 0.4$ according to our simulation tests. The limit values of the controller parameters to be optimized and the learned parameters values are given successively in Table 2 and Table 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{eqv}$ and $k_{eqh}$</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>$\eta_v$ and $\eta_h$</td>
<td>0.001</td>
<td>0.1</td>
</tr>
<tr>
<td>$\lambda_v$ and $\lambda_h$</td>
<td>7</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 2 Type limit values of the controller parameters to be optimized

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Final learned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{eqv}$</td>
<td>59.3628</td>
</tr>
<tr>
<td>$k_{eqh}$</td>
<td>15.5392</td>
</tr>
<tr>
<td>$\eta_v$</td>
<td>0.0022</td>
</tr>
<tr>
<td>$\eta_h$</td>
<td>0.0023</td>
</tr>
<tr>
<td>$\lambda_v$</td>
<td>6.7295</td>
</tr>
<tr>
<td>$\lambda_h$</td>
<td>5.5631</td>
</tr>
</tbody>
</table>

Table 3 Learned controller parameters values obtained with the PSO
Figure 6 The value of the cost function $F$ in successive iterations for the best particle.

Figure 7 Yaw angle $\alpha_h$ and pitch angle $\alpha_r$ responses (red line: actual state, blue line: desired state $\alpha_{r,ref}$) of the TRMS controlled by the proposed decentralized NFSMC after optimization.

Figure 8 Tail and main rotor applied voltages $u_t$ and $u_m$ generated by the proposed decentralized NFSMC after optimization.
From the simulation results, we can find that the control result of the conventional SMC produces a serious chattering phenomenon, as in figure 11. On the contrary, the chattering phenomenon of the controlled system was suppressed in the proposed controller, as shown.
In figure 8. Moreover, in the decentralized NFSMC, we did not need to compute the dynamical equation of the system and the equivalent control was estimated by the NN.

It should be noted that, applying PSO to tune a real process plant is rather impractical, as the algorithm needs to perform a large amount of repetitive iterations and evaluation on the plant, which may be impracticable and time consuming. Moreover, the risk of instability is high during the initial stages of the PSO iterations. Usually, a plant model can be used to simulate the controllability and evaluation of the controller. Also, the optimized parameters bounds are must be determined (as in Table 2). In this study, theses bounds are fixed according to the controlled system available information’s.

5. CONCLUDING REMARKS

In this paper, a decentralized NFSMC was proposed for a TRMS, and simulation results were illustrated. First, the TRMS was presented. Second, the SMC using a linear sliding surface was designed for each TRMS subsystem (vertical and horizontal subsystems). The design yielded an equivalent control term plus an addition control term for each TRMS subsystem. Third, the FLC structure used to compute the corrective control was presented, it was surveyed also how a NN was used to compute the equivalent control. In addition, a PSO algorithm has been adopted to tune some key parameters of each sub-NFSMC. The corrective control was accepted as a measure of error to update the weights of the NN. As a result, the decentralized NFSMC designed can achieve good accuracy and can eliminate completely chattering phenomenon without a degradation of the tracking performance compared to SMC used with the same control problem. Furthermore, this method presents also the advantage that there is no need to know the dynamical equation of a system to compute the equivalent control. Moreover, the learning process is online. Learning and calculation of the equivalent control signal are carried out simultaneously. The simulation results presented in this paper indicate that the suggested approach has considerable advantages compared to the classical sliding mode control. These characteristics make it a promising approach for control applications.

REFERENCES


