Research Article

Learning Control of Fixed-Wing Unmanned Aerial Vehicles Using Fuzzy Neural Networks

Erdal Kayacan,[1] Mojtaba Ahmadieh Khanesar,[2] Jaime Rubio-Hervas,[3] and Mahmut Reyhanoglu[4]

[1]*School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798*
[2]*Faculty of Electrical and Computer Engineering, Semnan University, Semnan 35131, Iran*
[3]*Infinium Robotics Pte Ltd., Singapore 128381*
[4]*Physical Sciences Department, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA*

Correspondence should be addressed to Erdal Kayacan; erdal@ntu.edu.sg

A learning control strategy is preferred for the control and guidance of a fixed-wing unmanned aerial vehicle to deal with lack of modeling and flight uncertainties. For learning the plant model as well as changing working conditions online, a fuzzy neural network (FNN) is used in parallel with a conventional P (proportional) controller. Among the learning algorithms in the literature, a derivative-free one, sliding mode control (SMC) theory-based learning algorithm, is preferred as it has been proved to be computationally efficient in real-time applications. Its proven robustness and finite time converging nature make the learning algorithm appropriate for controlling an unmanned aerial vehicle as the computational power is always limited in unmanned aerial vehicles (UAVs). The parameter update rules and stability conditions of the learning are derived, and the proof of the stability of the learning algorithm is shown by using a candidate Lyapunov function. Intensive simulations are performed to illustrate the applicability of the proposed controller which includes the tracking of a three-dimensional trajectory by the UAV subject to time-varying wind conditions. The simulation results show the efficiency of the proposed control algorithm, especially in real-time control systems because of its computational efficiency.

1. Introduction

Over the past several decades, unmanned aerial vehicles (UAVs) have proved their potential in several applications by using their several capabilities, *inter alia*, continuous and persistent surveillance, eliminating the need of aircrew, image processing capabilities by using relatively cheap sensors, and decreasing the size and weight of the aerial vehicle when compared to a conventional aircraft. UAVs have been used in a variety of civilian applications; some of which are disaster rescue [1], agricultural monitoring [2], wildlife protection [3], infrastructure inspection [4], 3D environment reconstruction [5], and person following [6].

UAVs can be classified into two groups: rotary wing and fixed wing. While the former has the capability of having aggressive maneuvers and being able to land and take off in small areas, the latter offers long flight endurance due to its flight characteristics about their gliding capabilities with no power. Among the gigantic number of fixed-wing UAVs applications, surveillance seems to be the most common application while benefitting from advanced computer vision techniques [7]. The most common path for a fixed-wing UAV is a combination of straight lines and circular orbits on a constant altitude [8].

For having a full autonomy of the aircraft, model-based controllers require a precise dynamic model of the aircraft. The controller must also be robust to wind and gust disturbances. However, under the time-varying parameters of an aircraft as well as time-varying working conditions and several stochastic disturbances, a learning control strategy is preferred in this paper. The proposed control algorithm does not need an accurate model of the aircraft. Instead, the intelligent structure of the controller learns the system dynamics online throughout the flight and optimizes its

performance for any arbitrary trajectory including both straight lines and circular orbits. For this purpose, the fusion of fuzzy logic and artificial neural networks, namely, FNNs, is preferred [9–12].

To eliminate all uncertainties in a control system and design a sophisticated model-based controller based on an accurate model of the system seems to be convincing. The reason is that, in the absence of model uncertainty, nonlinearity, and computational constraints, it is a well-known fact that linear-quadratic regulator (LQR) and linear-quadratic-Gaussian (LQG) control laws give reasonably satisfactory performance. However, eliminating all the uncertainties seems to be neither realistic nor a novel idea. For instance, till the beginning of the 20th century, it had been a big dream to eliminate all the uncertainties and to be able to achieve a fully predictable world. In 1814, P.S. Laplace formulated the predictability of the universe as follows:

> *"Given, for one instant, an intelligence which could comprehend all the forces by which nature is animated and the respective situation of the beings who compose it an intelligence sufficiently vast to submit these data to analysis it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atom. For it, nothing would be uncertain and the future, as the past, would be present to its eyes."* [13]. P. S. Laplace

On the other hand, quantum mechanics and the theory of relativity, which both appeared in the beginning of the 20th century, showed that our universe is quite random, and it is almost impossible to model or predict everything. In other words, our universe, at least on the level of subatomic particles, is not working like a "giant clock" which was claimed by P. S. Laplace. Even in a deterministic system, that is, a chaotic system, inevitable uncertainties in the initial conditions lead to huge differences in the future states of the system. In a similar manner, estimation and prediction of all changes during a fixed-wing UAV flight cannot be foreseen and considered in advance. All the aforementioned facts force us to propose some intelligent control algorithms which have learning capabilities throughout the operation.

Fuzzy logic theory and probability theory are the most widely used approaches to deal with the aforementioned inevitable phenomena: uncertainty. Although the concept of fuzzy logic and the concept of probability seem to be similar, they are quite different. While probability makes guesses about a certain reality, fuzzy logic does not make probability statements but represents membership in vaguely defined sets. For instance, if 0.5 is defined as a probability value for the oldness of a person, it can be said that there is a chance that he/she can be old. It is not known whether he/she is old or young. However in fuzzy logic, if 0.5 is defined as the degree of membership in the set of young and old people, we have some knowledge about his/him and he/she is positioned in the middle of young and old people. Since fuzzy logic contained vagueness, it was not appreciated by researchers when it was proposed for the first time in 1960s. However, since the 1970s,

this approach to set theory has been widely applied to control systems.

While the most significant feature of a fuzzy logic controller is its capability to inject expert knowledge into the controller design, the well-known capability of an artificial neural network is to be able to learn from input-output data. The fusion of fuzzy logic controllers and artificial neural networks results in FNNs [14, 15]. In any FNN architecture, the use of a learning algorithm is a must. In literature, there are three types of learning algorithms: derivative-based ones (backpropagation [16], Levenberg-Marquardt [17, 18], and least square), derivative-free ones (genetic algorithm, particle swarm optimization [19], and sliding mode control (SMC) theory-based), and hybrid algorithms (Levenberg-Marquardt-particle swarm optimization [20], backpropagation-Kalman filter, gradient descent-Kalman filter, and genetic algorithm-Kalman filter). The main problem with the derivative-based learning algorithms is that they need the calculation of the partial derivatives of the outputs of the FNN with respect to the antecedent parameters. Another problem worth mentioning is that derivative-based algorithms have always a possibility of getting trapped in local minima. In order to eliminate the aforementioned disadvantages of the derivative-based methods, derivative-free methods are proposed. As a derivative-free method, the disadvantage of the genetic algorithms is that their update formula is entirely random, and there is no mathematical guarantee that the cost function will decrease over time. Moreover, these algorithms are computationally expensive. On the other hand, as a derivative-free method, SMC based algorithms are computationally efficient and they provide robustness to the control of the system [21]. A detailed survey on the optimal tuning of FNNs can be found in [22].

Despite the fact that UAVs are being more and more visible in our daily life, their control is still a challenging task as they are open loop unstable, multi-input multi-output, and highly nonlinear systems in which there are significant intercouplings. What is more, they are always subjected to noise and disturbances because of the uncertainties in their navigation systems as well as wind and gust conditions. One way of controlling them is to use model-based control techniques. However, they need an accurate model of both the system and disturbances which is a challenging task in real life. A requirement is the use of sophisticated system identification methods to obtain the model of the aerial vehicle which is time-consuming task. The detailed steps and several methods for system identification and parameter estimation of aerial vehicles are discussed in [23]. What is more, the working conditions are always changing resulting in a fact that adaptability is a must. Motivated by the aforementioned drawbacks of the model-based controllers, a model free controller, the combination of a P controller and an FNN, is preferred in this paper. In order to be able to design a practical controller in real time in which computational power is always limited, we prefer one of the fastest learning algorithms in literature which is an SMC theory-based algorithm.

This paper presents a novel SMC theory-based learning algorithm with an adaptive learning rate and the evaluation

of the algorithm performance for a UAV flying in changing wind conditions. The paper is organized as follows: Section 2 introduces a fully nonlinear aircraft dynamic model in the presence of wind. In Section 3, the overall control scheme is described. In Section 4, a fuzzy neural control approach is introduced. Furthermore, the proposed training method, based on SMC theory, for the parameters of the FNN is proposed for the case of Gaussian membership functions. In Section 5, the proposed method is used to control a fixed-wing UAV. Finally, the concluding marks are presented in Section 6.

2. Mathematical Description of the UAV

This section briefly introduces the translational and rotational equations of motion (EOMs) for a fixed-wing UAV in the presence of wind.

2.1. Translational Dynamics. Let (x^o, y^o, h^o) denote the inertial coordinates of the UAV's center of mass and let (D, T, L, Y) be the drag, thrust, lift, and side forces, respectively. Denote by $(\gamma_{1_a}, \gamma_{2_a}, \gamma_{3_a})$ the aerodynamic bank, climb, and track angles, respectively, and let (P_1, P_2, P_3) be the wind perturbation vector (full expressions for these perturbations can be found in [24]). The wind is characterized by the sum of a mean speed V_{m_w} (acting in a horizontal plane along a heading angle ψ_w) and gust components $(u_{g_w}, v_{g_w}, w_{g_w})$. Then, for a fixed-wing UAV of mass m, the translational EOMs are given by

$$\dot{x}^o = V_a c\gamma_{2_a} c\gamma_{3_a} + V_{m_w} s\psi_w + u^o_{g_w} \tag{1}$$

$$\dot{y}^o = V_a c\gamma_{2_a} s\gamma_{3_a} + V_{m_w} c\psi_w + v^o_{g_w} \tag{2}$$

$$\dot{h}^o = V_a s\gamma_{2_a} - w^o_{g_w} \tag{3}$$

$$m\dot{V}_a = -D + Tc\alpha_a c\beta_a - mgs\gamma_{2_a} + P_1 \tag{4}$$

$$m V_a \dot{\gamma}_{3_a} c\gamma_{2_a} = Yc\gamma_{1_a} + Ls\gamma_{1_a}$$
$$- T\left(c\alpha_a s\beta_a c\gamma_{1_a} - s\alpha_a s\gamma_{1_a}\right) + P_2 \tag{5}$$

$$m V_a \dot{\gamma}_{2_a} = -Ys\gamma_{1_a} + Lc\gamma_{1_a} - mgc\gamma_{2_a}$$
$$+ T\left(c\alpha_a s\beta_a s\gamma_{1_a} + s\alpha_a c\gamma_{1_a}\right) + P_3, \tag{6}$$

where g is the gravitational acceleration, V_a is the air velocity, and α_a and β_a denote the aerodynamic angle of attack and sideslip angle, respectively. A superscript refers to the frame used within the formulations, and the abbreviations $s(\cdot) = \sin(\cdot)$, $c(\cdot) = \cos(\cdot)$, and $t(\cdot) = \tan(\cdot)$ are used throughout the paper. Note that α_a and β_a are expressed as

$$\alpha_a = t^{-1} \frac{w^b_a}{u^b_a},$$
$$\beta_a = s^{-1} \frac{v^b_a}{V_a}, \tag{7}$$

where (u_a, v_a, w_a) is the vector of linear velocities which is given, in body frame, as

$$\begin{bmatrix} u^b_a \\ v^b_a \\ w^b_a \end{bmatrix} = V_a \mathbf{T}_{bo} \begin{bmatrix} c\gamma_{2_a} c\gamma_{3_a} \\ c\gamma_{2_a} s\gamma_{3_a} \\ -s\gamma_{2_a} \end{bmatrix}, \tag{8}$$

where $\mathbf{T}_{bo}$ denotes the rotation matrix from the Earth-fixed inertial frame to the UAV-fixed body frame, which is given by

$$\mathbf{T}_{bo} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}. \tag{9}$$

2.2. Rotational Dynamics. Let (x_b, y_b, z_b) denote the UAV-fixed longitudinal, lateral, and directional axes, respectively. Assume that (x_b, y_b, z_b) are principal axes and x_b-z_b is the symmetry plane so that the inertia tensor is given by

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}, \tag{10}$$

where (I_x, I_y, I_z) denote the principal moments of inertia and I_{xz} is the product of inertia. Let (ϕ, θ, ψ) denote the roll, pitch, and yaw angles, respectively, and let (p_a, q_a, r_a) be the aerodynamic angular velocity vector. Then, the rotational equations of motion can be expressed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \dfrac{s\phi}{c\theta} & \dfrac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} p^b_a \\ q^b_a \\ r^b_a \end{bmatrix}$$
$$+ \begin{bmatrix} p^o_w \dfrac{c\psi}{c\theta} + q^o_w \dfrac{s\psi}{c\theta} \\ -p^o_w s\psi + q^o_w c\phi \\ p^o_w c\psi t\theta + q^o_w s\psi t\theta + r^o_w \end{bmatrix}$$

$$\begin{bmatrix} \dot{p}^b_a \\ \dot{q}^b_a \\ \dot{r}^b_a \end{bmatrix} = \mathbf{I}^{-1} \begin{bmatrix} \mathcal{L} - q^b_a h'_z + r^b_a h'_y - P_4 \\ \mathcal{M} + Tz^b_M - r^b_a h'_x + p^b_a h'_z - P_5 \\ \mathcal{N} - p^b_a h'_y + q^b_a h'_x - P_6 \end{bmatrix}$$
$$+ \mathbf{I}^{-1} \begin{bmatrix} -\left(I_z - I_y\right) q^b_a r^b_a + I_{xz} p^b_a q^b_a \\ \left(I_z - I_x\right) p^b_a r^b_a - I_{xz}\left(p^{b2}_a - r^{b2}_a\right) \\ -\left(I_y - I_x\right) p^b_a q^b_a - I_{xz} q^b_a r^b_a \end{bmatrix}, \tag{11}$$

where $(\mathcal{L}, \mathcal{M}, \mathcal{N})$ is the moment vector (the vector of rolling, pitching, and yawing moments, resp.); (p_w, q_w, r_w) is the wind angular velocity vector; z^b_M is the distance between the point of application of the thrust and the UAV's center of mass along the z_b axis; (h'_x, h'_y, h'_z) is the angular momentum vector of all rotors about the UAV-fixed x_b, y_b, z_b axes; and (P_4, P_5, P_6) is the wind perturbation vector (see [24]). Throughout the paper, 321 Euler angle sequence is used, so that $|\theta| < \pi/2$.

2.3. Forces and Moments Expressions. The following expressions of forces and moments are used in this paper:

$$
\begin{aligned}
L &= QS\left[C_{L_0} + C_{L_\alpha}\alpha_a\right] \\[4pt]
D &= QS\left[C_{D_0} + k_1 C_L + k_2 C_L^2\right] \\[4pt]
\mathcal{M} &= QSc\left[C_{m_0} + C_{m_\alpha}\alpha_a + C_{m_{\delta_e}}\delta_e + \frac{c}{2V_a}C_{m_q}q_a^b\right] \\[4pt]
Y &= QS\left[C_{Y_\beta}\beta_a + C_{Y_{\delta_r}}\delta_r + \frac{b}{2V_a}\left(C_{Y_p}p_a^b + C_{Y_r}r_a^b\right)\right] \\[4pt]
\mathcal{L} &= QSb\left[C_{l_\beta}\beta_a + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \right. \\[4pt]
&\quad \left. + \frac{b}{2V_a}\left(C_{l_p}p_a^b + C_{l_r}r_a^b\right)\right] \\[4pt]
\mathcal{N} &= QSb\left[C_{n_\beta}\beta_a + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \right. \\[4pt]
&\quad \left. + \frac{b}{2V_a}\left(C_{n_p}p_a^b + C_{n_r}r_a^b\right)\right] \\[4pt]
T &= \frac{k_m \rho}{V_a}\eta,
\end{aligned}
\tag{12}
$$

where $Q = \rho V_a^2/2$ is the dynamic pressure; (ρ, S, c, b) denote the air density, the wing surface, the mean aerodynamic chord, and the wing span, respectively; k_m is a constant; $(\delta_a, \delta_e, \delta_r)$ denote the aileron, elevator, and rudder deflections, respectively; and $\eta \in (0, 1]$ represents the throttle position.

3. The Proposed Control Structure

3.1. Kinematic Controller. The kinematic model is given by (1)–(3), from which an inverse kinematic model can be obtained which allows to calculate the reference air velocity, air-path and air-track angles. It is written as follows:

$$
\begin{aligned}
V_a &= \sqrt{\left(\dot{x}^o - V_{m_w}s\psi_w\right)^2 + \left(\dot{y}^o - V_{m_w}c\psi_w\right)^2 + \dot{h}^{o^2}} \\[6pt]
\gamma_{2_a} &= s^{-1}\frac{\dot{h}^o}{V_a} \\[6pt]
\gamma_{3_a} &= t^{-1}\frac{\dot{y}^o - V_{m_w}c\psi_w}{\dot{x}^o - V_{m_w}s\psi_w}.
\end{aligned}
\tag{13}
$$

Therefore, assuming the existence of a suitable state estimator scheme that estimates the mean wind vector, the kinematic control law to be applied to the UAV for trajectory tracking control is written as

$$
\begin{aligned}
V_{a_{\text{ref}}} &= \sqrt{\dot{x}_{\text{ref}}^{a^2} + \dot{y}_{\text{ref}}^{a^2} + \dot{h}_{\text{ref}}^{a^2}} \\[6pt]
\gamma_{2_{a_{\text{ref}}}} &= s^{-1}\frac{\dot{h}_{\text{ref}}^a}{V_{a_{\text{ref}}}} \\[6pt]
\gamma_{3_{a_{\text{ref}}}} &= t^{-1}\frac{\dot{y}_{\text{ref}}^a}{\dot{x}_{\text{ref}}^a},
\end{aligned}
\tag{14}
$$

where

$$
\begin{aligned}
\dot{x}_{\text{ref}}^a &= \dot{x}_d^o - V_{m_w}s\psi_w + k_{s_x}\mathrm{th}\left(k_{e_x}e_x\right) \\[6pt]
\dot{y}_{\text{ref}}^a &= \dot{y}_d^o - V_{m_w}c\psi_w + k_{s_y}\mathrm{th}\left(k_{e_y}e_y\right) \\[6pt]
\dot{h}_{\text{ref}}^a &= \dot{h}_d^o + k_{s_h}\mathrm{th}\left(k_{e_h}e_h\right)
\end{aligned}
\tag{15}
$$

and th is the hyperbolic tangent; $e_x = x^o - x_d^o$, $e_y = y^o - y_d^o$, and $e_h = h^o - h_d^o$ are the position errors in the inertial x, y, and h axes, respectively; the parameters k_{e_x}, k_{e_y}, and k_{e_h} are controller gains and k_{s_x}, k_{s_y}, and k_{s_h} are saturation constants; and (x_d^o, y_d^o, h_d^o) stands for the desired inertial coordinates. The parameters $V_{a_{\text{ref}}}$, $\gamma_{2_{a_{\text{ref}}}}$ and $\gamma_{3_{a_{\text{ref}}}}$ are the generated references for the controllers.

The following reference value is defined (i.e., coordinated turn conditions):

$$
\gamma_{1_{a_{\text{ref}}}} = t^{-1}\left(\frac{V}{g}\dot{\gamma}_{3_{a_{\text{ref}}}}\right).
\tag{16}
$$

Defining the trim angle of attack as

$$
\alpha_{a_d} = \frac{2mg/\rho V_{a_{\text{ref}}}^2 S - C_{L_0}}{C_{L_\alpha}}.
\tag{17}
$$

the generated references for the roll, pitch, and yaw angles can be linearly approximated by

$$
\begin{aligned}
\phi_{\text{ref}} &= \gamma_{1_{a_{\text{ref}}}} \\[4pt]
\theta_{\text{ref}} &= \alpha_{a_d} + \gamma_{2_{a_{\text{ref}}}} \\[4pt]
\psi_{\text{ref}} &= \gamma_{3_{a_{\text{ref}}}} - \beta
\end{aligned}
\tag{18}
$$

3.2. Proportional Controller Design. The proportional (P) controller can be implemented as follows:

$$
\begin{aligned}
\eta &= -k_V e_V \\[4pt]
\delta_a &= -k_\phi e_\phi \\[4pt]
\delta_e &= -k_\theta e_\theta \\[4pt]
\delta_r &= -k_\psi e_\psi,
\end{aligned}
\tag{19}
$$

where $e_V = V_a - V_{a_{\text{ref}}}$, $e_\phi = \phi - \phi_{\text{ref}}$, $e_\theta = \theta - \theta_{\text{ref}}$, and $e_\psi = \psi - \psi_{\text{ref}}$ are the air velocity, roll, pitch, and yaw errors, respectively; and the parameters k_V, k_ϕ, k_θ, and k_ψ are the gains of the P controller.
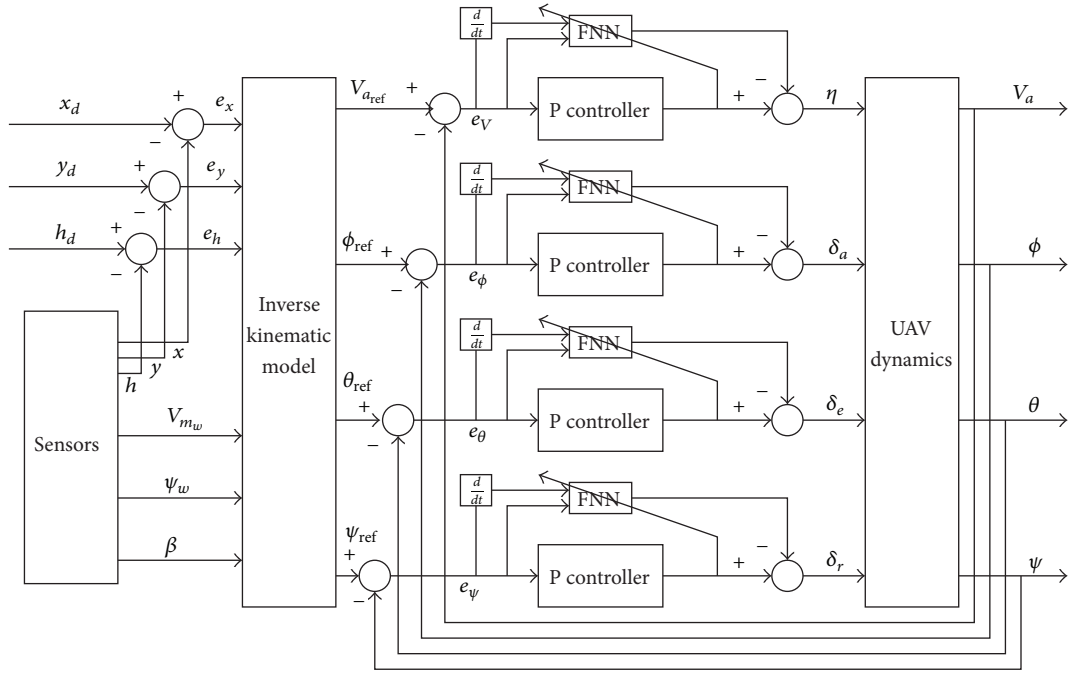
FIGURE 1: Block diagram of the proposed adaptive FNN control structure.

4. Fuzzy Neural Control Approach

Even if Mamdani-type fuzzy logic controllers were firstly proposed in the literature, a Takagi Sugeno Kang (TSK) fuzzy structure is preferred in this paper benefitting from its capability to be adapted over time. In the proposed method, as shown in Figure 1, P controllers work in parallel with FNNs. The task of the conventional P controllers is to provide some time for the FNN to learn the system dynamics online without going into the unstable working region. On the other hand, there is no need for the conventional P controllers to be tuned precisely.

4.1. The Structure of Fuzzy Neural Network. Figure 2 shows the internal structure of the proposed FNN. Even if the related figure shows the structure for m inputs, only two inputs are used in this study, namely, $x_1(t) = e(t)$ and $x_2(t) = \dot{e}(t)$. As a membership function (MF), Gaussian MFs are preferred. The fuzzified inputs are indicated as $\mu_{1_i}(x_1)$ and $\mu_{2_j}(x_2)$ for $i = 1, \ldots, I$ and $j = 1, \ldots, J$.

The fuzzy *if-then* rule R_{ij} of a zero-order TSK model can be defined as follows in which the consequent parameters consist of only crisp numbers:

$$R_{ij}: \text{If } x_1 \text{ is } M_{1i}, \ x_2 \text{ is } M_{2j}, \text{ then } \tau_n = f_{ij}. \quad (20)$$

For the calculation of the firing strength, a product-*T*-norm of the MFs is preferred as follows:

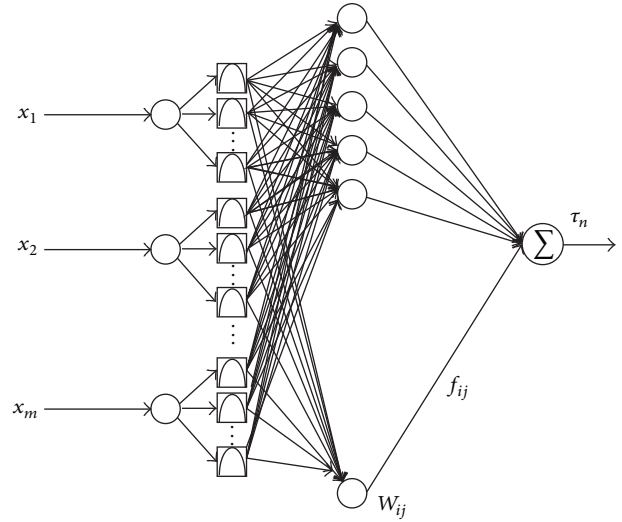$$W_{ij} = \mu_{1_i}(x_1)\,\mu_{2_j}(x_2). \quad (21)$$

FIGURE 2: Structure of the proposed FNN.

The Gaussian MFs $\mu_{1_i}(x_1)$ and $\mu_{2_j}(x_2)$ of the inputs x_1 and x_2 in (21) are represented by the following equation:

$$\mu_{1_i}(x_1) = \exp\left[-\frac{\left(x_1 - c_{1_i}\right)^2}{\sigma_{1_i}^2}\right]$$

$$\mu_{2_j}(x_2) = \exp\left[-\frac{\left(x_2 - c_{2_j}\right)^2}{\sigma_{2_j}^2}\right], \quad (22)$$

where the tunable parameters $c, \sigma > 0$ are the parameters of the MFs of the TSK FNN model.

Hence, (21) can be modified as follows:

$$W_{ij} = \exp\left[-\frac{\left(x_1 - c_{1_i}\right)^2}{\sigma_{1_i}^2} - \frac{\left(x_2 - c_{2_j}\right)^2}{\sigma_{2_j}^2} \right]. \quad (23)$$

The output of the proposed FNN can be calculated as the weighted average of the output of each rule:

$$\tau_n(t) = \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \overline{W}_{ij}, \quad (24)$$

where $\overline{W}_{ij}$ is the normalized value of the output signal of the neuron ij from the hidden layer of the network:

$$\overline{W}_{ij} = \frac{W_{ij}}{\sum_{i=1}^{I} \sum_{j=1}^{J} W_{ij}}. \quad (25)$$

The control signal τ of the system is calculated as the difference between the conventional P controller and the FNN:

$$\tau = \tau_c - \tau_n. \quad (26)$$

In order to ease the notation and make some of the equations vectorial, the following definitions are made.

$\overline{W}(t) = [\overline{W}_{11}(t)\ \overline{W}_{12}(t)\cdots\overline{W}_{21}(t)\cdots\overline{W}_{ij}(t)\cdots\overline{W}_{IJ}(t)]^T$ is vector of the normalized output signals of the neurons from the second hidden layer; $\sigma_1 = [\sigma_{1_1}\cdots\sigma_{1_i}\cdots\sigma_{1_I}]^T$, $\sigma_2 = [\sigma_{2_1}\cdots\sigma_{2_j}\cdots\sigma_{2_J}]^T$, $c_1 = [c_{1_1}\cdots c_{1_i}\cdots c_{1_I}]^T$, and $c_2 = [c_{2_1}\cdots c_{2_j}\cdots c_{2_J}]^T$ are vectors of the tuning parameters σ and c of the Gaussian MFs relevant to the fuzzification of the signals supplied to the first and second input of the FNN, respectively; $f(t) = [f_{11}(t)\ f_{12}(t)\cdots f_{21}(t)\ f_{22}(t)\cdots f_{ij}(t)\cdots f_{IJ}(t)]$ is vector of the time variable weight coefficients of the connections between the neurons from the second hidden layer and the output neuron of the FNN.

The following assumptions have been used in this paper.

The presence of the classical control system in the control scheme which adopted (Figure 1) the global asymptotic stability of the feedback system in compact space is guaranteed and we have

$$|x_1(t)| \le B_x,$$
$$|x_2(t)| \le B_x,$$
$$|\dot{x}_1(t)| \le B_{\dot{x}}, \quad (27)$$
$$|\dot{x}_2(t)| \le B_{\dot{x}}$$
$$\forall t,$$

where B_x and $B_{\dot{x}}$ are known upper bounds of the states of the system and their time derivatives, respectively.

The adaptation laws for the parameters of the MFs are made bounded which guarantees that

$$\|\sigma_1\| \le B_\sigma,$$
$$\|\sigma_2\| \le B_\sigma,$$
$$\|c_1\| \le B_c, \quad (28)$$
$$\|c_2\| \le B_c,$$

where B_σ and B_c are known bounds considered for the parameters of MFs.

It is also assumed that the time-varying parameters of the consequent part of the TSK FNN are bounded; that is,

$$|f_{ij}| \le B_f \quad \forall t, \quad (29)$$

where B_f is the known positive constant upper bound of the parameters f_{ij}.

From (23) and (25) it follows that $0 < \overline{W}_{ij} < 1$. Furthermore, it can be easily seen from (25) that $\sum_{i=1}^{I} \sum_{j=1}^{J} \overline{W}_{ij} = 1$.

Constraints (27) to (29) for the state variables of the system and their derivatives and the parameters of the zero-order TSK FNN make τ and $\dot{\tau}$ bounded:

$$|\tau(t)| \le B_\tau,$$
$$|\dot{\tau}(t)| \le B_{\dot{\tau}} \quad (30)$$
$$\forall t,$$

where B_τ and $B_{\dot{\tau}}$ are known positive constant upper bounds for τ and $\dot{\tau}$, respectively.

4.2. The Sliding Mode Learning Algorithm. Using the SMC theory principles [26] the zero value of the learning error coordinate $\tau_c(t)$ can be defined as time-varying sliding surface; that is,

$$S_c(\tau_n, \tau) = \tau_c(t) = \tau_n(t) + \tau(t) = 0 \quad (31)$$

which is the required condition for the training of the FNN. If this condition is satisfied, the FNN becomes a nonlinear regulator to learn the inverse dynamic of the system and obtains the desired performance of the system. This in turn guarantees the tracking of the reference signal.

The sliding surface for the nonlinear system under control $S_p(e)$ is defined as $S_p(e) = e$.

Definition 1. A sliding motion will appear on the sliding manifold $S_c(\tau_n, \tau) = \tau_c(t) = 0$ after a time t_h, if the condition $S_c(t)\dot{S}_c(t) = \tau_c(t)\dot{\tau}_c(t) < 0$ is satisfied for all t in some nontrivial semiopen subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$.

4.2.1. The Parameter Update Rules for the FNN.

Theorem 1. *If the adaptation law for the parameters of the considered FNN is chosen, respectively, as*

$$\dot{c}_{1i} = \dot{x}_1 \tag{32}$$

$$\dot{c}_{2j} = \dot{x}_2 \tag{33}$$

$$\dot{\sigma}_{1i} = -\frac{(\sigma_{1i})^3}{(x_1 - c_{1i})^2} \alpha \, \text{sgn}(\tau_c) \tag{34}$$

$$\dot{\sigma}_{2j} = -\frac{(\sigma_{2j})^3}{(x_2 - c_{2j})^2} \alpha \, \text{sgn}(\tau_c) \tag{35}$$

$$\dot{f}_{ij} = -\frac{\overline{W}_{ij}}{W^T W} \alpha \, \text{sgn}(\tau_c) \tag{36}$$

$$\dot{\alpha} = \gamma |\tau_c| - \gamma \nu \alpha, \tag{37}$$

where α is the adaptive learning rate and has a positive value.

Then, given an arbitrary initial condition $\tau_c(0)$, the learning error $\tau_c(t)$ will converge to a small neighborhood of zero during a finite time t_h.

Proof. See Appendix. □

Theorem 2. *If the adaptation strategy for the adjustable parameters of the FNN is chosen as in (32)–(37), then the negative definiteness of the time derivative of the Lyapunov function in (A.15) is ensured.*

Proof. See Appendix. □

Remark 1. The obtained result means that, assuming the SMC task is achievable, using τ_c as a learning error for the FNN together with the adaptation laws (32)–(37) enforces the desired reaching mode followed by a sliding regime for the system under control.

Remark 2. The reason behind using continuous time instead of using discrete time is that the stability proof in discrete time is very challenging. That is why we have decided to use the continuous time. This selection does not play a very critical role as we keep the sampling frequency of the system very high. Therefore, the system behaves like a continuous time system in its implementation. The explained framework is preferred by many researchers in literature.

5. Simulation Studies

The physical parameters used in the simulations can be found in Table 1, which correspond to the actual values of

TABLE 1: Parameters for the Lambda UAV landing at sea level [25].

Parameter	Value
ρ	$1.225 \, \text{kg/m}^3$
m	$92.10 \, \text{kg}$
I_y	$137.43 \, \text{kg·m}^2$
I_{xz}	$3.05 \, \text{kg·m}^2$
b	$4.29 \, \text{m}$
C_{L_0}	0.7939
C_{D_0}	0.0290
k_2	0.0363
C_{m_α}	-1.1010
C_{m_q}	-15.4000
$C_{Y_{\delta_r}}$	0.2865
C_{Y_r}	0.2601
$C_{l_{\delta_a}}$	0.2608
C_{l_p}	-0.5538
C_{n_β}	0.0600
$C_{n_{\delta_r}}$	-0.0943
C_{n_r}	-0.1650
g	$9.81 \, \text{m/s}^2$
I_x	$83.75 \, \text{kg·m}^2$
I_z	$210.99 \, \text{kg·m}^2$
S	$1.96 \, \text{m}^2$
c	$0.46 \, \text{m}$
C_{L_α}	5.8200
k_1	0
C_{m_0}	0
$C_{m_{\delta_e}}$	-0.8449
C_{Y_β}	-0.4372
C_{Y_p}	-0.0016
C_{l_β}	-0.0145
$C_{l_{\delta_r}}$	0.0022
C_{l_r}	0.0876
$C_{n_{\delta_a}}$	-0.0137
C_{n_p}	-0.0360

the Lambda Unmanned Research Vehicle [25]. We set the saturation limits as

$$|\delta_a| \leq 30^o,$$
$$|\delta_e| \leq 30^o,$$
$$|\delta_r| \leq 30^o, \tag{38}$$
$$0 \leq \eta \leq 1.$$

The initial conditions are taken as

$$(x^o, y^o, h^o)_0 = (0, 25, 150) \, \text{m},$$
$$V_{a_0} = 27.44 \, \text{m/s} \tag{39}$$
$$(\phi, \theta, \psi)_0 = (5^o, 2^o, -5^o).$$
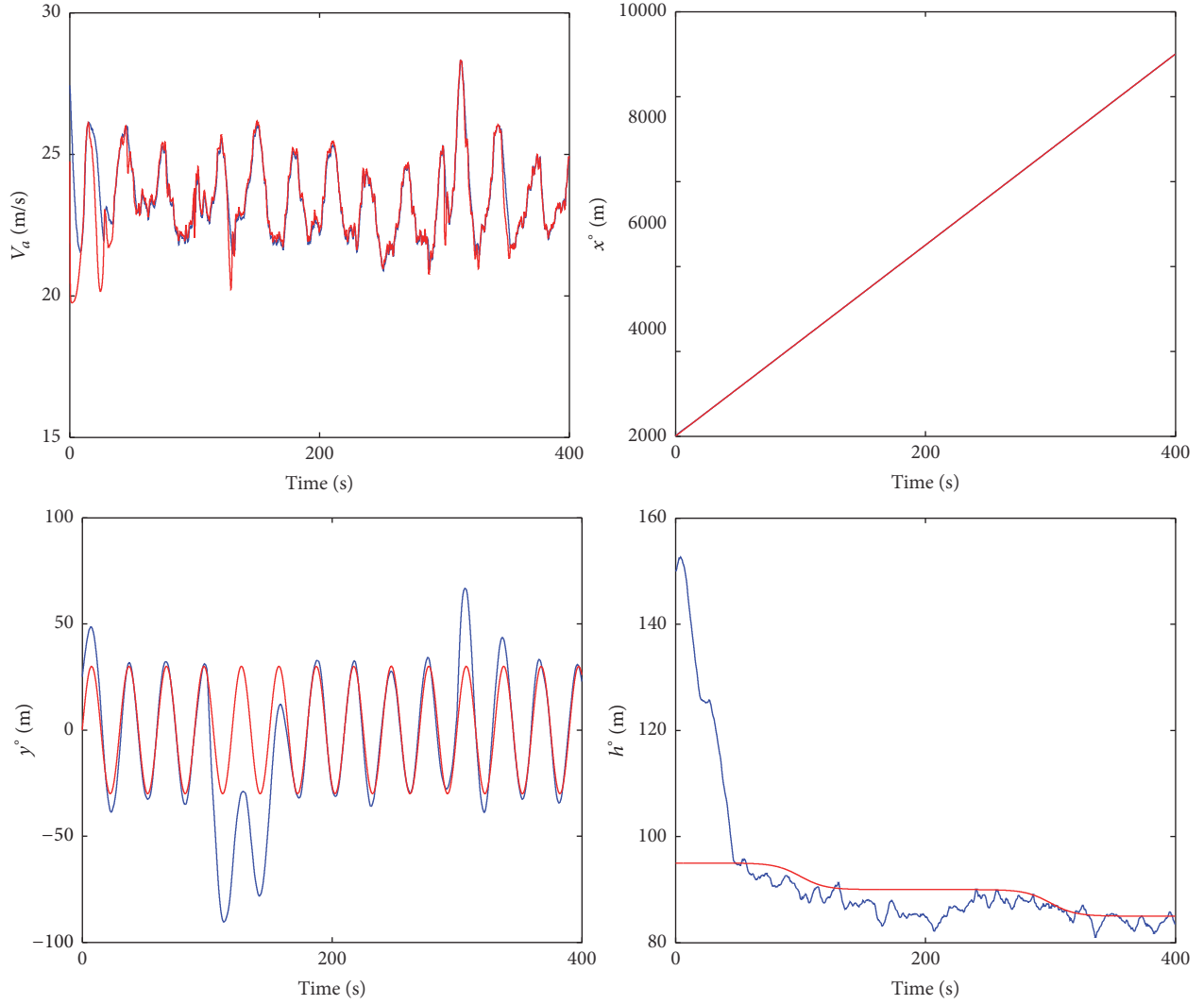
FIGURE 3: Time responses of the air velocity (V_a), range (x^o), inertial lateral displacement (y^o), and altitude (h^o) (blue line) compared with their reference (red) values.

The control objective is to track a sinusoidal reference trajectory at a constant altitude defined by

$$x_d^o = 22.5t \text{ m}$$

$$y_d^o = 30 \sin\left(\frac{2\pi}{30}t\right) \text{ m}$$

$$h_d^o = 90 - 2.5 \tanh(-5 + 0.05t) \quad (40)$$

$$- 2.5 \tanh(-15 + 0.05t) \text{ m}.$$

The P controller (19) is applied with the following control parameters:

$$k_V = 300,$$

$$k_\phi = 0.5,$$

$$k_\theta = -10, \quad (41)$$

$$k_\psi = -0.5.$$

After 200 s, the FNNs are turned on. The mean wind velocity has a magnitude of $V_{m_\psi} = 5$ m/s along the inertial lateral axis (i.e., $\psi_w = 0$) for the first 100 s; then its orientation changes to $\psi_w = -180^o$ to change back to $\psi_w = 0$ after 300 s. The wind gusts have been accordingly incorporated via the "Dryden Wind Turbulence Model (Continuous)" MATLAB toolbox.

It can be seen in Figure 3 that the P controller (first 200 s) results in tracking errors for the variables of interest. This is due to the inherent difficulties which arise from the tuning of this kind of controllers. As the FNN is switched on, these errors are significantly reduced and the actual trajectory (blue line) clearly tracks the reference trajectory (red line) as shown in Figures 3–5. As can be seen from these figures, the tracking is still accurate in the presence of time-varying wind conditions. These results show that the control scheme consisting of an FNN working in parallel with a P controller gives a better trajectory following accuracy than the one where only a P controller acts alone.
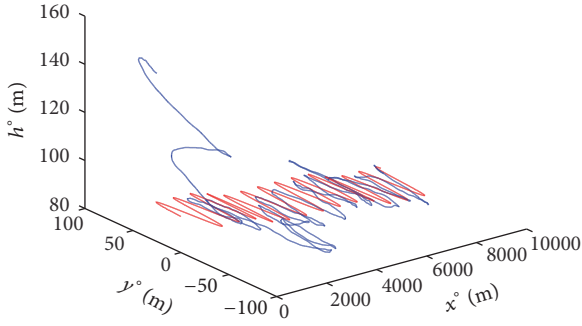
FIGURE 4: Three-dimensional representation of the UAV trajectory (blue line) compared with its reference (red line) trajectory.
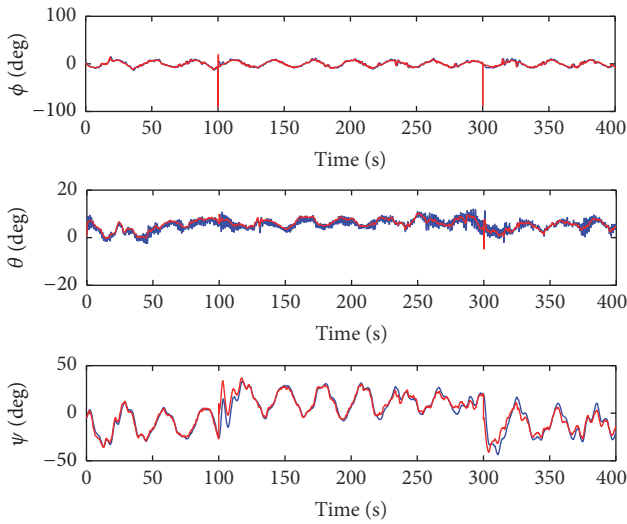

FIGURE 5: Time responses of the roll (ϕ), pitch (θ) and yaw (ψ) angles (blue line) compared with their reference (red) values.

Although the performance of the P controller could be improved by better tuning, this is a challenging task in real life. Thus, the proposed control structure, consisting of an intelligent controller and a conventional controller, is preferable.

The control action (i.e., deflection of the different control surfaces) is shown as the corresponding contribution of the P controller (blue line) and the FNN (red line) in Figure 6. It can be seen that after the 200th second, the FNN takes over control, whereas the output of the P controller comes to approximately zero as is expected in this kind of intelligent controllers.

Figure 7 shows the Euclidean error between the actual and desired trajectory. It can be seen that, as the FNN is turned on after the 200th second, the accuracy improves. It can also be seen that even when the wind conditions change (i.e., 100th second and 300th second for the P controller acting along and the FNN working in parallel with the P controller, resp.), the FNN improves the system response.

It is to be noted that although there exist four independent subsystem controllers in the UAV control system, the FNN works in parallel with a P controller only for the control of

the thrust, pitch, and yaw, given the good performance of the roll channel being actuated only by the P controller.

6. Conclusion

In this paper, an SMC theory-based learning algorithm has been introduced for the control of a fixed-wing UAV in the presence of wind. The adaptation laws for the parameters of the FNN are proposed and their learning stability conditions are investigated for a structure with two inputs, each being modeled by Gaussian MFs. The proposed control structure consists of a P controller and an FNN which can learn the inverse dynamics of the plant model online rather than a need for an accurate predefined dynamic model of the system. The obtained simulation results illustrate that using the proposed learning laws for the parameters of the FNN makes it possible to reach and maintain the predefined sliding manifold. It is further observed that not only is the proposed method robust but also another prominent feature of it is its ease of implementation. The effectiveness of the proposed algorithm has been demonstrated through computer simulations, which include the tracking of a three-dimensional trajectory by the UAV in the presence of time-varying wind conditions. As a future extension to this paper, authors would like to implement state estimation methods for the available on board information.

Appendix

Proof of Theorem 1. The time derivatives of (22) are as follows:

$$\dot{\mu}_{1i}(x_1) = -2A_{1i}(A_{1i})' \mu_{1i}(x_1)$$
$$\dot{\mu}_{2j}(x_2) = -2A_{2j}(A_{2j})' \mu_{2j}(x_2), \tag{A.1}$$

where

$$A_{1i} = \left(\frac{x_1 - c_{1i}}{\sigma_{1i}} \right),$$
$$A_{2j} = \left(\frac{x_2 - c_{2j}}{\sigma_{2j}} \right). \tag{A.2}$$

The time derivative of (25) can be obtained easily as follows:

$$\overline{W}_{ij} = -\overline{W}_{ij}\dot{K}_{ij} + \overline{W}_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\overline{W}_{ij}\dot{K}_{ij}\right), \tag{A.3}$$

where

$$\dot{K}_{ij} = 2\left(A_{1i}(A_{1i})' + A_{2j}(A_{2j})' \right). \tag{A.4}$$

The stability analysis of the system is done using the following Lyapunov function:

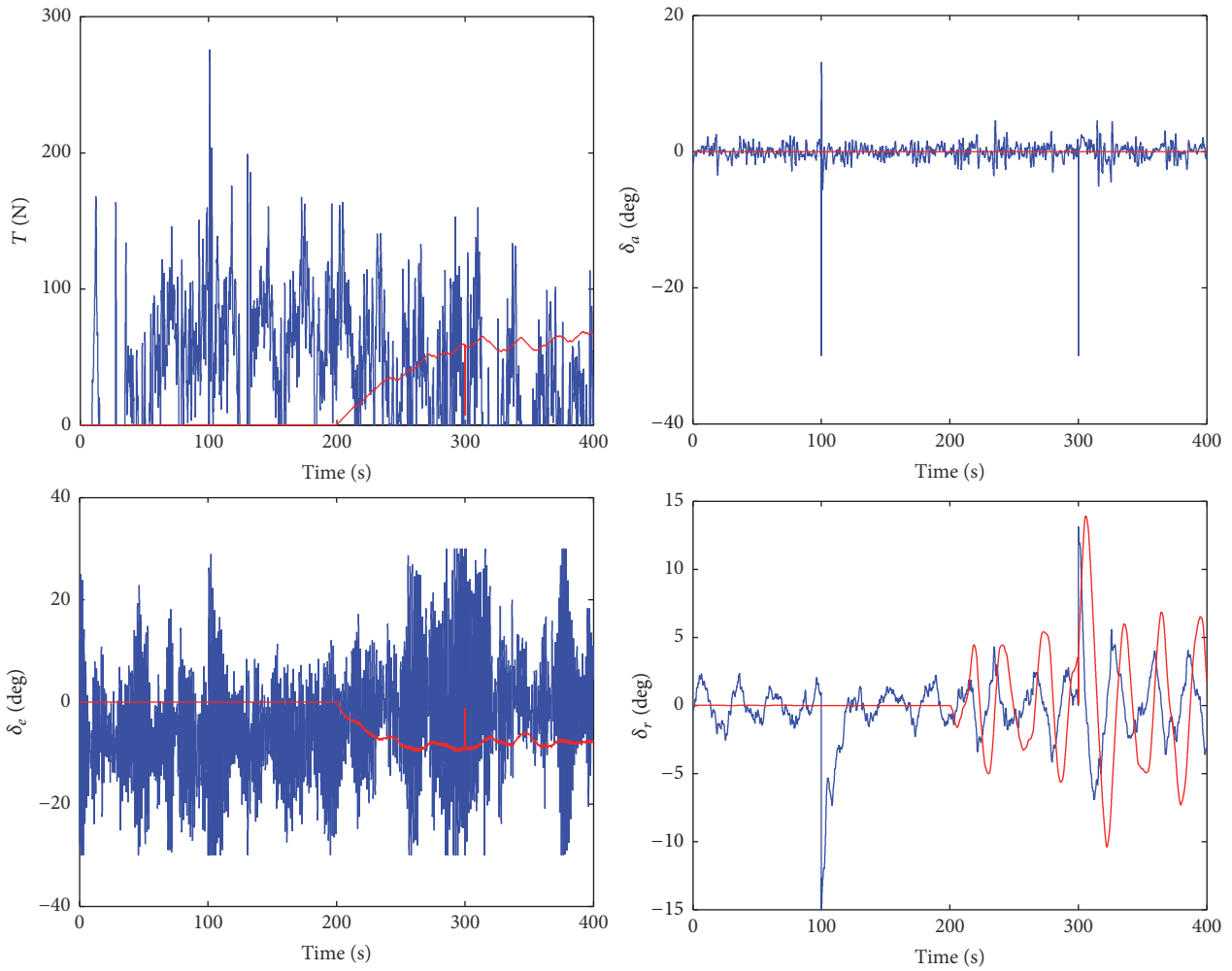$$V_c = \frac{1}{2}\tau_c^2(t) + \frac{1}{2\gamma}(\alpha - \alpha^*)^2. \tag{A.5}$$

FIGURE 6: Time responses of the thrust (T), aileron (δ_a), elevator (δ_e), and rudder (δ_r) deflections for the P controller (blue line) and the FNN (red line).
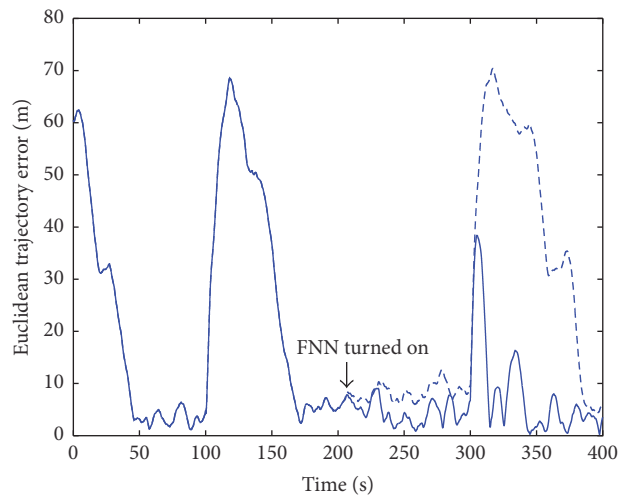

FIGURE 7: Time response of the Euclidean error between the actual and reference trajectory for the P controller acting alone (dashed line) and the FNN in parallel with the P controller (solid line).

The time derivative of V_c is given by

$$\dot{V}_c = \tau_c \dot{\tau}_c = \tau_c \left(\dot{\tau}_n + \dot{\tau} \right) + \frac{1}{\gamma} \dot{\alpha} \left(\alpha - \alpha^* \right), \quad \text{(A.6)}$$

where

$$\dot{\tau}_n = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(\dot{f}_{ij} \overline{W}_{ij} + f_{ij} \dot{\overline{W}}_{ij} \right). \quad \text{(A.7)}$$

By replacing (A.7) to the (A.6), the following equation is obtained:

$$\dot{V}_c = \tau_c \left[\sum_{i=1}^{I} \sum_{j=1}^{J} \dot{f}_{ij} \overline{W}_{ij} \right.$$

$$- 2 \sum_{i=1}^{I} \sum_{j=1}^{J} \overline{W}_{ij} \left(A_{1i} \left(A_{1i} \right)' + A_{2j} \left(A_{2j} \right)' \right) f_{ij}$$

$$+ 2 \sum_{i=1}^{I} \sum_{j=1}^{J} \left(\overline{W}_{ij} f_{ij} \times \sum_{i=1}^{I} \sum_{j=1}^{J} \overline{W}_{ij} \left(A_{1i} \left(A_{1i} \right)' + A_{2j} \left(A_{2j} \right)' \right) \right)$$

$$\left. + \dot{\tau} \right] + \frac{1}{\gamma} \dot{\alpha} \left(\alpha - \alpha^* \right), \quad \text{(A.8)}$$

where

$$\dot{A}_{1i} = \frac{ \left(\dot{x}_1 - \dot{c}_{1i} \right) \sigma_{1i} - \left(x_1 - c_{1i} \right) \dot{\sigma}_{1i} }{ \sigma_{1i}^2 }$$

$$\dot{A}_{2j} = \frac{ \left(\dot{x}_2 - \dot{c}_{2j} \right) \sigma_{2j} - \left(x_2 - c_{2j} \right) \dot{\sigma}_{2j} }{ \sigma_{2j}^2 }. \quad \text{(A.9)}$$

Equation (A.10) can be obtained by using (32)–(35);

$$A_{1i} \dot{A}_{1i} = A_{2j} \dot{A}_{2j} = \alpha \, \text{sgn} \left(\tau_c \right). \quad \text{(A.10)}$$

Considering (A.10) and the fact that $\sum_{i=1}^{I} \sum_{j=1}^{J} \overline{\overline{W}}_{ij} = 1$, we have the following equation:

$$\dot{V}_c = \tau_c \left[\sum_{i=1}^{I} \sum_{j=1}^{J} \dot{f}_{ij} \overline{W}_{ij} + \dot{\tau} \right] + \frac{1}{\gamma} \dot{\alpha} \left(\alpha - \alpha^* \right), \quad \text{(A.11)}$$

where

$$\dot{f}_{ij} = -\frac{\overline{W}_{ij}}{\overline{W}^T \overline{W}} \alpha \, \text{sgn} \left(\tau_c \right). \quad \text{(A.12)}$$

Considering the adaptation law of α as in (37) and the adaptation law of f_{ij} as in (A.12), the following equation is obtained:

$$\dot{V}_c \leq -\frac{\alpha^*}{2} \left| \tau_c \right| + \frac{\nu}{4} \alpha^{*2}, \quad \text{(A.13)}$$

where α^* as $B_{\dot{\tau}} \leq \alpha^*/2$. The equation implies that the Lyapunov function decreases until $\left| \tau_c \right| < \nu \alpha^*/2$. So that τ_c will stay bounded. Furthermore ν is a design parameter and it is possible to take this value as small as desired.

The relation between the sliding function (it is a point in this investigation) S_p and the zero adaptive learning error level S_c is as follows:

$$S_c = \tau_c = k_P e = k_p S_p. \quad \text{(A.14)}$$

The tracking performance of the feedback control system can be analyzed by introducing the following Lyapunov function candidate:

$$V_p = \frac{1}{2} S_p^2. \quad \text{(A.15)}$$

$\square$

Proof of Theorem 2. Evaluating the time derivative of the Lyapunov function in (A.15) yields

$$\dot{V}_c \leq -\frac{\alpha^*}{2} \left| \tau_c \right| + \frac{\nu}{4} \alpha^{*2} \quad \forall S_c, S_p \neq 0. \quad \text{(A.16)}$$

This equation implies that V_c converges until $\left| \tau_c \right| < \nu \alpha^*/2$ and $\left| \tau_c \right|$ remains bounded. $\square$

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] N. Michael, S. Shen, K. Mohta et al., "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[2] V. H. Andaluz, E. López, D. Manobanda et al., "Nonlinear controller of quadcopters for agricultural monitoring," in *Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14–16, 2015, Proceedings, Part I*, vol. 9474 of *Lecture Notes in Computer Science*, pp. 476–487, Springer, Berlin, Germany, 2015.

[3] M. A. Olivares-Mendez, C. Fu, P. Ludivig et al., "Towards an autonomous vision-based unmanned aerial system against wildlife poachers," *Sensors (Switzerland)*, vol. 15, no. 12, pp. 31362–31391, 2015.

[4] I. Sa and P. Corke, "Vertical infrastructure inspection using a quadcopter and shared autonomy control," in *Field and Service Robotics*, pp. 219–232, Springer, Berlin, Germany, 2014.

[5] C. Fu, A. Carrio, and P. Campoy, "Efficient visual odometry and mapping for unmanned aerial vehicle using ARM-based stereo vision pre-processing system," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '15)*, pp. 957–962, IEEE, Denver, Colo, USA, June 2015.

[6] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a Quadrotor MAV," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '15)*, pp. 2182–2189, May 2015.

[7] S. A. P. Quintero and J. P. Hespanha, "Vision-based target tracking with a small UAV: optimization-based control strategies," *Control Engineering Practice*, vol. 32, pp. 28–42, 2014.

[8] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: a survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, 2014.

[9] M. T. Long and W. Y. Nan, "Adaptive position tracking system and force control strategy for mobile robot manipulators using fuzzy wavelet neural networks," *Journal of Intelligent & Robotic Systems*, vol. 79, no. 2, pp. 175–195, 2015.

[10] R.-J. Wai and P.-C. Chen, "Robust neural-fuzzy-network control for robot manipulator including actuator dynamics," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1328–1349, 2006.

[11] S. Rezazadeh, M. A. Ardestani, and P. S. Sadeghi, "Optimal attitude control of a quadrotor UAV using Adaptive Neuro-Fuzzy Inference System (ANFIS)," in *Proceedings of the 3rd International Conference on Control, Instrumentation, and Automation (ICCIA '13)*, Tehran, Iran, December 2013.

[12] I. D. M. Esper and P. F. F. Rosa, "Heading controller for a fixed wing UAV with reduced control surfaces based on ANFIS," in *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W '15)*, pp. 118–123, June 2015.

[13] P. De Laplace, F. Truscott, and F. Emory, A Philosophical Essay on Probabilities. Cosimo, Incorporated, 2007.

[14] E. Kayacan, W. Saeys, E. Kayacan, H. Ramon, and O. Kaynak, "Intelligent control of a tractor-implement system using type-2 fuzzy neural networks," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '12)*, pp. 1–8, Brisbane, Australia, June 2012.

[15] A. V. Topalov, E. Kayacan, Y. Oniz, and O. Kaynak, "Adaptive neuro-fuzzy control with sliding mode learning algorithm: application to antilock braking system," in *Proceedings of the 7th Asian Control Conference (ASCC '09)*, Hong Kong, China, August 2009.

[16] E. Kayacan, Y. Oniz, A. C. Aras, O. Kaynak, and R. Abiyev, "A servo system control with time-varying and nonlinear load conditions using type-2 TSK fuzzy neural system," *Applied Soft Computing*, vol. 11, no. 8, pp. 5735–5744, 2011.

[17] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[18] G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, A. A. Bandala, and E. P. Dadios, "A neural network approach to a cooperative balancing problem in quadrotor-unmanned aerial vehicles (QUAVs)," in *Proceedings of the 8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM '15)*, pp. 1–5, Cebu, Philippines, December 2015.

[19] Y. Wang, Y. Chenxie, J. Tan, C. Wang, Y. Wang, and Y. Zhang, "Fuzzy radial basis function neural network PID control system for a quadrotor UAV based on particle swarm optimization," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '15)*, pp. 2580–2585, August 2015.

[20] A. I. Abubakar, A. Khan, N. M. Nawi et al., "Studying the effect of training levenberg marquardt neural network by using hybrid meta-heuristic algorithms," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 1, pp. 450–460, 2016.

[21] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[22] S. Hassan, M. A. Khanesar, E. Kayacan, J. Jaafar, and A. Khosravi, "Optimal design of adaptive type-2 neuro-fuzzy systems: a review," *Applied Soft Computing Journal*, vol. 44, pp. 134–143, 2016.

[23] N. V. Hoffer, C. Coopmans, A. M. Jensen, and Y. Chen, "A survey and categorization of small low-cost unmanned aerial vehicle system identification," *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1, pp. 129–145, 2014.

[24] J. Rubio-Hervas, E. Kayacan, M. Reyhanoglu, and H. Tang, "Nonlinear control of fixed-wing UAVs in presence of stochastic winds," *Communications in Nonlinear Science and Numerical Simulation*, vol. 33, pp. 57–69, 2016.

[25] G. A. Swift, *Model identification and control design for the Lambda unmanned research vehicle [M.S. thesis]*, Air Force Institute of Technology, 1991.

[26] V. I. Utkin, *Sliding Modes in Control and Optimization*, Springer, Berlin, Germany, 1992.