<table>
<thead>
<tr>
<th>Title</th>
<th>Learning Control of Fixed-Wing Unmanned Aerial Vehicles Using Fuzzy Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Kayacan, Erdal; Khanesar, Mojtaba Ahmadieh; Rubio-Hervas, Jaime; Reyhanoglu, Mahmut</td>
</tr>
<tr>
<td>Date</td>
<td>2017</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/43648">http://hdl.handle.net/10220/43648</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2017 Erdal Kayacan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.</td>
</tr>
</tbody>
</table>
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 Reyhanoglu4

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

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

Received 21 August 2016; Revised 25 December 2016; Accepted 26 December 2016; Published 9 February 2017

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 \((\gamma_1, \gamma_2, \gamma_3)\) 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 \(V_w\) (acting in a horizontal plane along a heading angle \(\psi_w\)) and gust components \(\dot{u}_w, \dot{v}_w, \dot{w}_w\). Then, for a fixed-wing UAV of mass \(m\), the translational EOMs are given by

\[
\begin{align*}
\dot{x}^o &= V_a \gamma_2 \gamma_3 + V_{m_w} \psi_w + u^o_w \quad (1) \\
\dot{y}^o &= V_a \gamma_2 \gamma_3 + V_{m_w} \psi_w + v^o_w \quad (2) \\
\dot{h}^o &= V_a \gamma_2 - w^o_w \quad (3) \\
m \dot{V}_a &= -D + T \alpha_a \beta_a - mg \gamma_2 + P_1 \quad (4) \\
m \dot{V}_a \gamma_2 &= Y_c \gamma_1 + L \gamma_1 \\
&- T (\alpha_a \beta_a \gamma_1 - s \alpha_a \gamma_1) + P_2 \quad (5) \\
m \dot{V}_a \gamma_3 &= -Y_c \gamma_1 + L \gamma_3 + mg \gamma_3 \\
&+ T (\alpha_a \beta_a \gamma_3 + s \alpha_a \gamma_3) + P_3 \quad (6)
\end{align*}
\]

where \(g\) is the gravitational acceleration, \(V_a\) is the air velocity, and \(\alpha_a\) and \(\beta_a\) 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 \(\alpha_a\) and \(\beta_a\) are expressed as

\[
\begin{align*}
\alpha_a &= t^{-1} \frac{\dot{u}_w}{V_a} \\
\beta_a &= s^{-1} \frac{\dot{v}_w}{V_a}
\end{align*}
\]

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

\[
\begin{bmatrix}
\dot{u}_a \\
\dot{v}_a \\
\dot{w}_a
\end{bmatrix}
= V_a T_{bo}
\begin{bmatrix}
\cos \gamma_1 \\
\sin \gamma_1 \cos \gamma_3 \\
-\sin \gamma_1 \sin \gamma_3
\end{bmatrix}
\]

where \(T_{bo}\) denotes the rotation matrix from the Earth-fixed inertial frame to the UAV-fixed body frame, which is given by

\[
T_{bo} =
\begin{bmatrix}
 c\theta \psi & c\theta \phi & -s\theta \\
 s\phi \theta \psi + c\phi \psi s\theta & s\phi \theta \phi + c\phi \psi c\theta & s\phi \psi c\theta \\
 s\phi \theta \psi + c\phi \psi s\theta & s\phi \theta \phi + c\phi \psi c\theta & s\phi \psi c\theta
\end{bmatrix}.
\]

### 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

\[
I = \begin{bmatrix}
I_x & 0 & -I_{xz} \\
0 & I_y & 0 \\
-I_{xz} & 0 & I_z
\end{bmatrix},
\]

where \(I_x, I_y, I_z\) denote the principal moments of inertia and \(I_{xz}\) is the product of inertia. Let \((\phi, \theta, \psi)\) denote the roll, pitch, and yaw angles, respectively, and let \((P_o, q_o, r_o)\) be the aerodynamic angular velocity vector. Then, the rotational equations of motion can be expressed as

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix}
= \begin{bmatrix}
1 & s \phi t \theta & c \phi t \theta \\
0 & c \phi & -s \phi \\
0 & s \phi & c \phi
\end{bmatrix}
\begin{bmatrix}
\dot{\phi}_a \\
\dot{\theta}_a \\
\dot{\psi}_a
\end{bmatrix}
+ \begin{bmatrix}
P_w \psi V_r + q_o w^o_y + r_o w^o_z \\
-P_w \phi V_r + q_o w^o_z - r_o v^o_y \\
-p_o \phi \psi + s_o V_w + \phi \psi \frac{q_o}{c \phi} + \frac{r_o}{c \phi}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{\phi}_a \\
\dot{\theta}_a \\
\dot{\psi}_a
\end{bmatrix}
= \Gamma_1
\begin{bmatrix}
L - \dot{h}_a h_z + \dot{r}_a h_y - P_4 \\
M + T \dot{h}_a h_z - \dot{r}_a h_x + \dot{p}_a h_z - P_5 \\
N - \dot{r}_a h_y + \dot{p}_a h_x - P_6
\end{bmatrix}
+ \Gamma_1
\begin{bmatrix}
-(I_z - I_x) \dot{q}_a h_z + I_x z_a \dot{p}_a \dot{q}_a \\
(I_z - I_x) \dot{p}_a h_z - I_z \dot{r}_a - r^2_a \\
(I_y - I_z) \dot{p}_a h_y - I_y \dot{r}_a - r^2_a
\end{bmatrix},
\]

where \(\Gamma_1\) is the moment vector (the vector of rolling, pitching, and yawing moments, resp.); \((P_o, q_o, r_o)\) is the wind angular velocity vector; \(z_a\) is the distance between the point of application of the thrust and the UAV’s center of mass along the \(z_b\) axis; \((h_3, h_y, h_z)\) is the angular momentum vector of all rotors about the UAV-fixed \(x_b, y_b, z_b\) axes; and \((P_a, P_3, P_6)\) is the wind perturbation vector (see [24]). Throughout the paper, 312 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:

\[
L = QS \left[ C_{L_0} + C_{L_a} \alpha_a \right]
\]
\[
D = QS \left[ C_{D_0} + C_{D_a} \alpha_a + C_{D_m} \delta_x + \frac{c}{2V_a} C_n q_b \right]
\]
\[
\mathcal{M} = QSc \left[ C_{m_0} + C_{m_a} \alpha_a + C_{m_h} \delta_x + \frac{c}{2V_a} C_m q_b \right]
\]
\[
Y = QS \left[ C_Y \beta_a + C_{Y_e} \delta_e + \frac{b}{2V_a} \left( C_{Y_r} \beta_a + C_{Y_r} \beta_b \right) \right]
\]
\[
\mathcal{X} = QSb \left[ C_i \beta_a + C_{l_a} \delta_a + C_{l_h} \delta_r \right]
\]
\[
+ \frac{b}{2V_a} \left( C_i \beta_a + C_{l_a} \beta_b \right)
\]
\[
\mathcal{N} = QSb \left[ C_{n_1} \beta_a + C_{n_a} \delta_a + C_{n_b} \delta_r \right]
\]
\[
+ \frac{b}{2V_a} \left( C_{n_1} \beta_a + C_{n_a} \beta_b \right)
\]
\[
T = k_m \frac{\rho V_a^2}{a} \eta,
\]

where \(Q = \rho V_a^2 / 2\) is the dynamic pressure; \((\rho, S, c, b)\) 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:

\[
V_a = \sqrt{\left( x_a^o - V_m \psi \right)^2 + \left( y_a^o - V_m \psi \right)^2 + \psi^2}
\]
\[
y_{2_a} = s^{-1} \frac{h_a^o}{V_a}
\]
\[
y_{3_a} = t^{-1} \frac{y_a^o - V_m \psi}{x_a^o - V_m \psi}
\]

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

\[
V_{a_{ref}} = \sqrt{x_{a_{ref}}^2 + y_{a_{ref}}^2 + h_{a_{ref}}^2}
\]
\[
y_{2_{ref}} = s^{-1} \frac{h_{a_{ref}}}{V_{a_{ref}}}
\]
\[
y_{3_{ref}} = t^{-1} \frac{y_{a_{ref}} - V_m \psi}{x_{a_{ref}} - V_m \psi}
\]

where

\[
x_{a_{ref}}^o = x_d^o - V_m \psi \omega + k_x \theta (e_x \psi)
\]
\[
y_{a_{ref}}^o = y_d^o - V_m \psi \omega + k_y \theta (e_y \psi)
\]
\[
h_{a_{ref}}^o = h_d^o + k_h \theta (e_h \psi)
\]

and \(\theta\) is the hyperbolic tangent; \(e_x = x^o - x_a^o\), \(e_y = y^o - y_a^o\), and \(e_h = h^o - h_a^o\) are the position errors in the inertial \(x, y, \) and \(h\) axes, respectively; the parameters \(k_x, k_y, k_h\) are controller gains and \(k_m, k_n, k_l\) are saturation constants; and \((x_d^o, y_d^o, h_d^o)\) stands for the desired inertial coordinates. The parameters \(V_{a_{ref}}, y_{2_{ref}}\) and \(y_{3_{ref}}\) are the generated references for the controllers.

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

\[
y_{1_{ref}} = t^{-1} \left( \frac{V}{g} y_{3_{ref}} \right).
\]

Defining the trim angle of attack as

\[
\alpha_{a_d} = 2mg/\rho V_{a_{ref}}^2 S - C_{L_a}.
\]

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

\[
\phi_{ref} = y_{1_{ref}}
\]
\[
\theta_{ref} = \alpha_{a_d} + y_{2_{ref}}
\]
\[
\psi_{ref} = y_{3_{ref}} - \beta
\]

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

\[
\eta = -k_\psi \epsilon_{\psi}
\]
\[
\delta_x = -k_{\delta} \epsilon_{x}
\]
\[
\delta_e = -k_{e} \epsilon_{e}
\]
\[
\delta_r = -k_{\delta} \epsilon_{\delta}
\]

where \(\epsilon_{v} = V_a - V_{a_{ref}}, e_\phi = \phi - \phi_{ref}, e_\theta = \theta - \theta_{ref}, \) and \(e_\psi = \psi - \psi_{ref}\) are the air velocity, roll, pitch, and yaw errors, respectively; and the parameters \(k_\psi, k_\delta, k_\theta, \) and \(k_\psi\) are the gains of the P controller.
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 $\mu_{1i}(x_1)$ and $\mu_{2j}(x_2)$ 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:

$$W_{ij} = \mu_{1i}(x_1) \mu_{2j}(x_2). \quad (21)$$

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

$$\mu_{1i}(x_1) = \exp \left[ -\frac{(x_1 - c_{1i})^2}{\sigma_{1i}^2} \right], \quad (22)$$

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

\[
W_{ij} = \exp \left[ -\frac{(x_1 - c_{ij})^2}{\sigma_{i1}^2} - \frac{(x_2 - c_{ij})^2}{\sigma_{i2}^2} \right].
\]  

(23)

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

\[
\tau_n(t) = \sum_{i=1}^{l} \sum_{j=1}^{I} f_{ij} \overline{W}_{ij},
\]

(24)

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

\[
\overline{W}_{ij} = \frac{W_{ij}}{\sum_{i=1}^{l} \sum_{j=1}^{I} W_{ij}}.
\]

(25)

The control signal \( \tau \) of the system is calculated as the difference between the conventional P controller and the FNN:

\[
\tau = \tau_e - \tau_n.
\]

(26)

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

\[
\overline{W}(t) = [\overline{W}_{11}(t) \overline{W}_{12}(t) \cdots \overline{W}_{1I}(t) \cdots \overline{W}_{l1}(t) \cdots \overline{W}_{lI}(t)]^T
\]

is vector of the normalized output signals of the neurons from the second hidden layer; \( \sigma = [\sigma_1 \cdots \sigma_1 \cdots \sigma_1]^T, \sigma_2 = [\sigma_2 \cdots \sigma_2 \cdots \sigma_2]^T, c_1 = [c_1 \cdots c_1 \cdots c_1]^T \), and \( c_2 = [c_2 \cdots c_2 \cdots c_2]^T \) are vectors of the tuning parameters \( \sigma \) and \( c \) of the Gaussian MFs relevant to the fuzzification of the signals supplied to the first and second input of the FNN, respectively; \( f(t) = [f_{11}(t) f_{12}(t) \cdots f_{1I}(t) f_{21}(t) \cdots f_{2I}(t) \cdots f_{lI}(t)] \) 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

\[
\begin{align*}
|x_1(t)| &\leq B_x, \\
|x_2(t)| &\leq B_x, \\
|\dot{x}_1(t)| &\leq B_x, \\
|\dot{x}_2(t)| &\leq B_x, \\
\forall t,
\end{align*}
\]

(27)

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

\[
\begin{align*}
\|\sigma_1\| &\leq B_\sigma, \\
\|\sigma_2\| &\leq B_\sigma, \\
\|c_1\| &\leq B_c, \\
\|c_2\| &\leq B_c,
\end{align*}
\]

(28)

where \( B_\sigma \) 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,

\[
|f_{ij}| \leq B_f \forall t,
\]

(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 \( \sum_{i=1}^{l} \sum_{j=1}^{I} \overline{W}_{ij} = 1 \).

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

\[
\begin{align*}
|\tau(t)| &\leq B_\tau, \\
|\dot{\tau}(t)| &\leq B_\dot{\tau},
\end{align*}
\]

(30)

\[
\forall t,
\]

where \( B_\tau \) and \( B_\dot{\tau} \) are known positive constant upper bounds for \( \tau \) 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 \( \tau_n(t) \) can be defined as time-varying sliding surface; that is,

\[
S_c(\tau_n, \tau) = \tau_n(t) + \tau(t) = 0
\]

(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 \( S_c(\tau_n, \tau) \equiv \tau_n(t) = 0 \) after a time \( t_h \), if the condition \( S_c(\tau_n) = \tau_n(t) \dot{\tau}(t) < 0 \) 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

\[
\dot{c}_i = \dot{x}_1 (32)
\]

\[
\dot{c}_j = \dot{x}_2 (33)
\]

\[
\dot{\sigma}_i = - \frac{(\sigma_i)^3}{(x_i - c_i)^2} \alpha \text{sgn} (\tau_c) (34)
\]

\[
\dot{\sigma}_j = - \frac{(\sigma_j)^3}{(x_j - c_j)^2} \alpha \text{sgn} (\tau_c) (35)
\]

\[
\dot{f}_{ij} = - \frac{\overline{W}_{ij}}{W} \alpha \text{sgn} (\tau_c) (36)
\]

\[
\dot{\alpha} = \gamma |\tau_c| - \gamma \alpha, (37)
\]

where \( \alpha \) 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 \( \tau_c \) as a learning error for the FNN together with the adaptation laws (32)–(37) enforce 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 the Lambda Unmanned Research Vehicle [25]. We set the saturation limits as

\[
|\delta_d| \leq 30^\circ, \\
|\delta_e| \leq 30^\circ, \\
|\delta_i| \leq 30^\circ, (38)
\]

\[
0 \leq \eta \leq 1.
\]

The initial conditions are taken as

\[
(x^0, y^0, h^0)_0 = (0, 25, 150) \text{ m}, \]

\[
V_{ao} = 27.44 \text{ m/s} \]

\[
(\phi, \theta, \psi)_0 = (5^\circ, 2^\circ, -5^\circ).
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho )</td>
<td>1.225 kg/m³</td>
</tr>
<tr>
<td>( m )</td>
<td>92.10 kg</td>
</tr>
<tr>
<td>( I_y )</td>
<td>137.43 kg·m²</td>
</tr>
<tr>
<td>( I_{xz} )</td>
<td>3.05 kg·m²</td>
</tr>
<tr>
<td>( b )</td>
<td>4.29 m</td>
</tr>
<tr>
<td>( C_{l0} )</td>
<td>0.7939</td>
</tr>
<tr>
<td>( C_{r0} )</td>
<td>0.0290</td>
</tr>
<tr>
<td>( k_2 )</td>
<td>0.0363</td>
</tr>
<tr>
<td>( C_{ml} )</td>
<td>-1.1010</td>
</tr>
<tr>
<td>( C_{mr} )</td>
<td>-15.4000</td>
</tr>
<tr>
<td>( C_{\gamma h} )</td>
<td>0.2865</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>0.2601</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>0.2608</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.5538</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>0.0600</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.0943</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.1650</td>
</tr>
<tr>
<td>( g )</td>
<td>9.81 m/s²</td>
</tr>
<tr>
<td>( I_x )</td>
<td>83.75 kg·m²</td>
</tr>
<tr>
<td>( I_z )</td>
<td>810.99 kg·m²</td>
</tr>
<tr>
<td>( S )</td>
<td>1.96 m²</td>
</tr>
<tr>
<td>( c )</td>
<td>0.46 m</td>
</tr>
<tr>
<td>( C_{l0} )</td>
<td>5.8200</td>
</tr>
<tr>
<td>( C_{r0} )</td>
<td>0</td>
</tr>
<tr>
<td>( C_{ml} )</td>
<td>0</td>
</tr>
<tr>
<td>( C_{mr} )</td>
<td>-0.8449</td>
</tr>
<tr>
<td>( C_{\gamma h} )</td>
<td>-0.4372</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.0016</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.0145</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>0.0022</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>0.0876</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.0137</td>
</tr>
<tr>
<td>( C_{\gamma c} )</td>
<td>-0.0360</td>
</tr>
</tbody>
</table>
The control objective is to track a sinusoidal reference trajectory at a constant altitude defined by

\[
\begin{align*}
    x^o_d &= 22.5t \text{ m} \\
    y^o_d &= 30 \sin \left( \frac{2\pi}{30} t \right) \text{ m} \\
    h^o_d &= 90 - 2.5 \tanh(-5 + 0.05t) \\
    &\quad - 2.5 \tanh(-15 + 0.05t) \text{ m.}
\end{align*}
\]

(40)

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

\[
\begin{align*}
    k_v &= 300, \\
    k_\phi &= 0.5, \\
    k_\theta &= -10, \\
    k_\psi &= -0.5.
\end{align*}
\]

(41)

After 200 s, the FNNs are turned on. The mean wind velocity has a magnitude of \( V_{w_\infty} = 5 \text{ m/s} \) along the inertial lateral axis (i.e., \( \psi_w = 0 \)) for the first 100 s; then its orientation changes to \( \psi_w = -180^\circ \) 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 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.
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 alone 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:

\[
\dot{\mu}_{1i}(x_1) = -2A_{1i}(A_{1i})'\mu_{1i}(x_1),
\]

\[
\dot{\mu}_{2j}(x_2) = -2A_{2j}(A_{2j})'\mu_{2j}(x_2),
\]

(A.1)

where

\[
A_{1i} = \left(\begin{array}{c} x_1 - c_{1i} \\ \sigma_{1i} \end{array}\right),
\]

\[
A_{2j} = \left(\begin{array}{c} x_2 - c_{2j} \\ \sigma_{2j} \end{array}\right).
\]

(A.2)

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

\[
\bar{W}_{ij} = -\bar{W}_{ij}\dot{K}_{ij} + \bar{W}_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}(\bar{W}_{ij}\dot{K}_{ij}),
\]

(A.3)

where

\[
\dot{K}_{ij} = 2\left(A_{1i}(A_{1i})' + A_{2j}(A_{2j})'\right).
\]

(A.4)

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

\[
V_c = \frac{1}{2}e^2(t) + \frac{1}{2y}(\alpha - \alpha^*)^2.
\]

(A.5)
Figure 6: Time responses of the thrust ($T$), aileron ($\delta_a$), elevator ($\delta_e$), and rudder ($\delta_r$) 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
\[ V_c = \tau_c \dot{\tau}_c = \tau_c (\dot{\tau}_n + \dot{\tau}) + \frac{1}{y} \alpha (\alpha - \alpha^*), \] (A.6)
where
\[ \dot{\tau}_n = \sum_{i=1}^{I} \sum_{j=1}^{J} \left( f_{ij} \dot{w}_{ij} + f_g \dot{w}_{ij} \right). \] (A.7)
By replacing (A.7) to the (A.6), the following equation is obtained:
\[ V_c = \tau_c \left[ \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \dot{w}_{ij} - 2 \tau_c \sum_{i=1}^{I} \sum_{j=1}^{J} \left( A_{1i}(A_{1i})' + A_{2j}(A_{2j}) \right) f_{ij} \right. \\
\left. + 2 \sum_{i=1}^{I} \sum_{j=1}^{J} \left( \dot{w}_{ij} f_{ij} + \sum_{i=1}^{I} \sum_{j=1}^{J} (A_{1i}(A_{1i})' + A_{2j}(A_{2j}) \right) \right] \\
+ \dot{\tau} + \frac{1}{y} \alpha (\alpha - \alpha^*), \]
where
\begin{align*}
\dot{A}_{1i} &= \frac{(\dot{x}_1 - \dot{c}_{1i}) \sigma_{1i} - (\dot{x}_1 - \dot{c}_{1i}) \sigma_{1i}^2}{\sigma_{1i}^2}, \\
\dot{A}_{2j} &= \frac{(\dot{x}_2 - \dot{c}_{2j}) \sigma_{2j} - (\dot{x}_2 - \dot{c}_{2j}) \sigma_{2j}^2}{\sigma_{2j}^2}.
\end{align*}
(A.9)
Equation (A.10) can be obtained by using (32)–(35);
\[ A_{1i} \dot{A}_{1i} = A_{2j} \dot{A}_{2j} = \alpha \text{sgn} (\tau_c). \] (A.10)
Considering (A.10) and the fact that $\sum_{i=1}^{I} \sum_{j=1}^{J} \dot{w}_{ij} = 1$, we have the following equation:
\[ \dot{V}_c = \tau_c \left[ \sum_{i=1}^{I} \sum_{j=1}^{J} \dot{f}_{ij} \dot{w}_{ij} + \dot{\tau} \right] + \frac{1}{y} \alpha (\alpha - \alpha^*), \] (A.11)
where
\[ \dot{f}_{ij} = -\frac{\dot{w}_{ij}}{\dot{w}_{ij}} \alpha \text{sgn} (\tau_c). \] (A.12)
Considering the adaptation law of $\alpha$ as in (37) and the adaptation law of $f_{ij}$ as in (A.12), the following equation is obtained:
\[ \dot{V}_c \leq -\frac{\alpha}{2} |\tau_c| + \frac{y}{4} \alpha^2, \] (A.13)
where $\alpha^*$ as in (37) and the adaptation law of $f_{ij}$ as in (A.12), the following equation is obtained:
\[ \dot{V}_c \leq -\frac{\alpha}{2} |\tau_c| + \frac{y}{4} \alpha^2. \] The relation between the sliding function (it is a point in this investigation) $S_p$ and the zero adaptive learning error level $S_p$ is as follows:
\[ S_p = \tau_c = k_p e = k_p S_p, \] (A.14)
The tracking performance of the feedback control system can be analyzed by introducing the following Lyapunov function candidate:
\[ V_p = \frac{1}{2} S_p^2. \] (A.15)
\begin{proof}
Evaluating the time derivative of the Lyapunov function in (A.15) yields
\[ \dot{V}_c \leq -\frac{\alpha}{2} |\tau_c| + \frac{y}{4} \alpha^2 \forall S_p, S_p \neq 0. \] (A.16)
This equation implies that $V_c$ converges until $|\tau_c| < \alpha^*/2$ and $|\tau_c|$ remains bounded.
\end{proof}

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

Acknowledgments
This research was supported by Nanyang Technological Internal Start-Up Grant and Ministry of Education with the project titles “Learning Control Algorithms for Unmanned Aerial Vehicles” and “Model Predictive Control-Moving Horizon Evaluation Framework as Applied to Tilt Rotor UAVs and Its Experimental Evaluation,” respectively.

References


