



Development of a Robust Interval Type-2 TSK Fuzzy Logic Controlled UAV Platform

Abel Hailemichael¹ · Ali Karimoddini¹

Received: 15 July 2021 / Accepted: 16 December 2022 / Published online: 8 February 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract

Type-2 Fuzzy Logic Controllers (FLCs) are capable of effectively capturing and accommodating uncertainties and disturbances. However, these controllers generally suffer from high computation costs. This paper develops a robust quadcopter UAV platform, equipped with a new interval type-2 (IT2) Takagi-Sugeno-Kang (TSK) fuzzy logic controller. The advantage of the developed controller is to enhance the robustness of the control structure, while managing the computation costs, making it appropriate for real-time control developments. The developed controller is applied to the attitude control of a UAV, which is relatively a fast dynamical system. The effectiveness of the proposed IT2 TSK FLC is verified through a developed software-in-the-loop (SITL) simulator for a quadcopter UAV. Then, actual flight experiments are conducted. The performance of the UAV when using the developed IT2 TSK FLC is compared with its performance when using a classical PID controller.

Keywords Unmanned aerial vehicles · Fuzzy logic controller · Interval type-2 · Flight control

Mathematics Subject Classification (2010) Categories (5) · (3)

1 Introduction

Unmanned Aerial Vehicles (UAVs) are usually controlled remotely from a ground station or autonomously from decisions generated by an on-board computer. Eliminating the risk to a pilots' life, UAVs are particularly ideal choices for hard-to-reach and dangerous places. Therefore, UAVs have been increasingly used for a wide range of applications including combat [13, 46], surveillance [20, 39], transportation [1, 27], agriculture [5, 44], inspection [17, 18], disaster management [2, 14], network coverage [22, 36], etc. Regardless of the targeted application, it is important to develop a reliable controller for the flight

control of deployed UAVs. Controlling a UAV commonly involves stabilizing its attitude at a desired reference value in the inner-loop and then driving the UAV to follow a desired trajectory in the outer-loop (higher level) controller. With such a control strategy, a challenging problem is to control the attitude of the UAV which has a fast dynamics. Failing to properly control the UAV attitude may quickly end with a crash. Once a proper attitude control is achieved, a simple controller can be used for driving the UAV to a desired position. Following this strategy, in this paper, we develop an advanced and robust interval type-2 TSK fuzzy logic system for the attitude control of a quadcopter UAV and employ a PID controller for the control of its altitude and position.

Many classical UAV control techniques require an accurate model of the UAV and/or significant tuning efforts, in the end, the controller may not be robust against model uncertainties, external disturbances or noises. This challenges the UAV flight control systems in many real-world situations, which involve uncertainties rise from sensor-reading inaccuracy and noises as well as external disturbances and changes in environmental conditions.

Along with the ability of FLCs to express the behavior of complex systems without knowing much about their

✉ Ali Karimoddini
akarimod@ncat.edu

Abel Hailemichael
athailam@aggies.ncat.edu

¹ Department of Electrical and Computer Engineering,
North Carolina Agricultural and Technical State
University, Greensboro, NC, 27411, USA

mathematical model, FLCs are capable of effectively capturing and accommodating uncertainties [16, 33, 47]. Fuzzy logic controllers are exceptionally powerful in mimicking human decision-making by using membership functions and rule based inference mechanisms [33, 41, 48]. These strong advantages of FLCs can potentially improve UAVs' performance in the presence of uncertainties which arise from noises, sensor measurement errors, and external disturbances which commonly exist in almost any real-world environment. Therefore, FLCs have been previously employed for several position and trajectory control applications of UAVs [6, 11, 42].

The robustness of FLCs can be improved using Type-2 memberships [8, 34]. This is due to the fact that Type-2 FLCs (T2 FLCs) can handle uncertainties better than Type-1 FLCs by introducing secondary memberships to quantify the level of uncertainty in the degree of primary memberships [10, 37]. Our work in [15] shows that a more robust performance would be achieved if T2 FLCs are used compared to T1 FLC. As such, T2 FLCs have been applied to several application domains such as aerospace [9], voice recognition [32], communication and signal processing [26], and time-series analysis [19]. In [3] and [4], a T2 FLC controller is applied to the altitude control of a UAV and evaluated via MATLAB simulations, noting that the altitude control of a UAV is less complex than its attitude control. In [23], a multi-layer flight controller is used for the control of a quadcopter, whose outerloop controller is an IT2 FLC for reference tracking, noting that the outerloop control design is less complex and sensitive than the innerloop. Similarly, in [7] and [40], T2 FLCs are used for position and trajectory tracking control of UAVs, assuming that an innerloop controller takes care of the innerloop dynamics. However, to the best of our knowledge, T2 FLCs have not been applied to the attitude control of UAVs. This is due to the fact that T2 FLCs are computationally expensive, making them very difficult to be used for real-time control applications.

To overcome this problem, in this paper, we use Interval type-2 fuzzy sets [24, 35], in which the secondary membership values of their elements are always unity. Such Interval type-2 FLCs have to manage much lesser computation costs, while maintaining major advantages of type-2 FLCs. Further, instead of using a computationally expensive Mamdani fuzzy inference mechanism [28], we use Takagi-Sugeno-Kang (TSK) fuzzy inference mechanisms [25, 43]. TSK FLCs can greatly reduce the computation costs of FLCs by easily expressing fuzzy rule outputs and allowing parallel processing of the inputs and outputs [43]. Moreover, we have adopted the uncertainty bounds output processing method [12, 33] for further reduction of the computation

cost of the controller output. The developed flight control system is then implemented on a quadcopter UAV. For enabling efficient information sharing between components of the UAV and its ground control station (GCS) as well as easily incorporating additional hardware and software components, Robot Operating System (ROS) [21, 38] and PX4 autopilot architecture [29] are integrated with the developed UAV platform. The performance of the developed control system is then verified through a software in the loop (SITL) simulation. Further, using this SITL simulator, the developed IT2 TSK FLC, the performance of the UAV controlled by an IT2 TSK FLC is compared with that of the UAV controlled by a PID controller. Finally, the performance of the developed control structure is verified through actual flight experiments. The systematic construction procedure, modeling, control design, and implementation details are provided and made publicly available for possible future enhancement of the control performance and expansion of the application horizon.

Therefore, the contributions of this paper are as follows:

- Development and integration of a computationally effective Interval Type-2 FLC, which is implementable in a resource-constrained on-board flight control system. The developed IT2 FLC controller uses a TSK inference technique concatenated with the uncertainty bound output processing enabling parallel processing of rule outputs.
- Development of a modular UAV control structure with integrated IT2 FLCs for orientation (inner-loop) control of a UAV. To the best of our knowledge, this is the first work that implements and integrates an IT2 TSK FLC for the control of a UAV.
- Modification and enhancement of the PX4 autopilot architecture to support IT2 TSK FLC-based control of UAVs.
- Development of a robust and scalable research quadcopter UAV platform, and experimental validation of the performance of the developed IT2 FLC flight controller via SITL simulations as well as actual flight tests.

The rest of this paper is organized as follows. In Section 2, the hardware components and software architecture of the developed UAV are explained. Section 3 discusses the mathematical model of a quadcopter UAV. Then, the flight control architecture of the developed quadcopter UAV is discussed in Section 4. An IT2 TSK FLC is then designed for the developed UAV as detailed in Section 5. Section 6 presents the simulation and real-time flight test results for the attitude control of the quadcopter UAV. The paper is concluded in Section 7.

2 Structure of the Developed Quadcopter UAV

2.1 Hardware Structure of the Developed UAV

The developed UAV is a quadcopter with a *Cross Configuration (X)*, whose mechanical and electrical components are explained here. Its basic mechanical components include the body frame and propellers. The frame is used to hold and carry the UAV's hardware components while the propellers are used for creating a lift force to move horizontally and generating torque to rotate the UAV by varying the motors' speeds. Additional hardware components of the developed UAV include sensors, autopilot hardware, voltage regulators, motors, electronic motor speed controllers (ESCs), battery, and battery status sensors. The block-diagram for the hardware structure of the developed UAV is shown in Fig. 1. The UAV communicates with the ground control station (GCS) using WiFi and radio links.

An autopilot hardware receives information from an inertial measurement unit (IMU), a motion capture system, GPS, Camera, LIDAR, and other sensors, to estimate the state of the UAV and control it. As shown in

Fig. 1, the autopilot hardware uses Pixhawk [30] as the main flight control system and Raspberry Pi-3 as a companion computer. With a redundant design architecture, Pixhawk system performs the state estimation and control the UAV. The Pixhawk system communicates with the companion computer using the Micro Aerial Vehicle Link (MAVLink) protocol. The companion computer assists the Pixhawk flight controller with high level tasks such as communicating with the motion capture unit, communicating with the GCS, receiving sensor data, image processing, and path planning.

Remark 1 The focus of this paper is on the control design for a UAV to robustly follow a given trajectory, which is assumed to be collision-free. Therefore, Lidar and camera sensors, which primarily are used for collision avoidance are not employed during the experimentation for this paper. We included these sensors as a part of the developed platform for demonstrating a UAV platform for relatively complete list of sensors. In this paper, the test flights are performed in both simulated and experimental flight test environments. A software generated GPS as well as sonar sensors are used for the localization of UAV in the simulated environment.

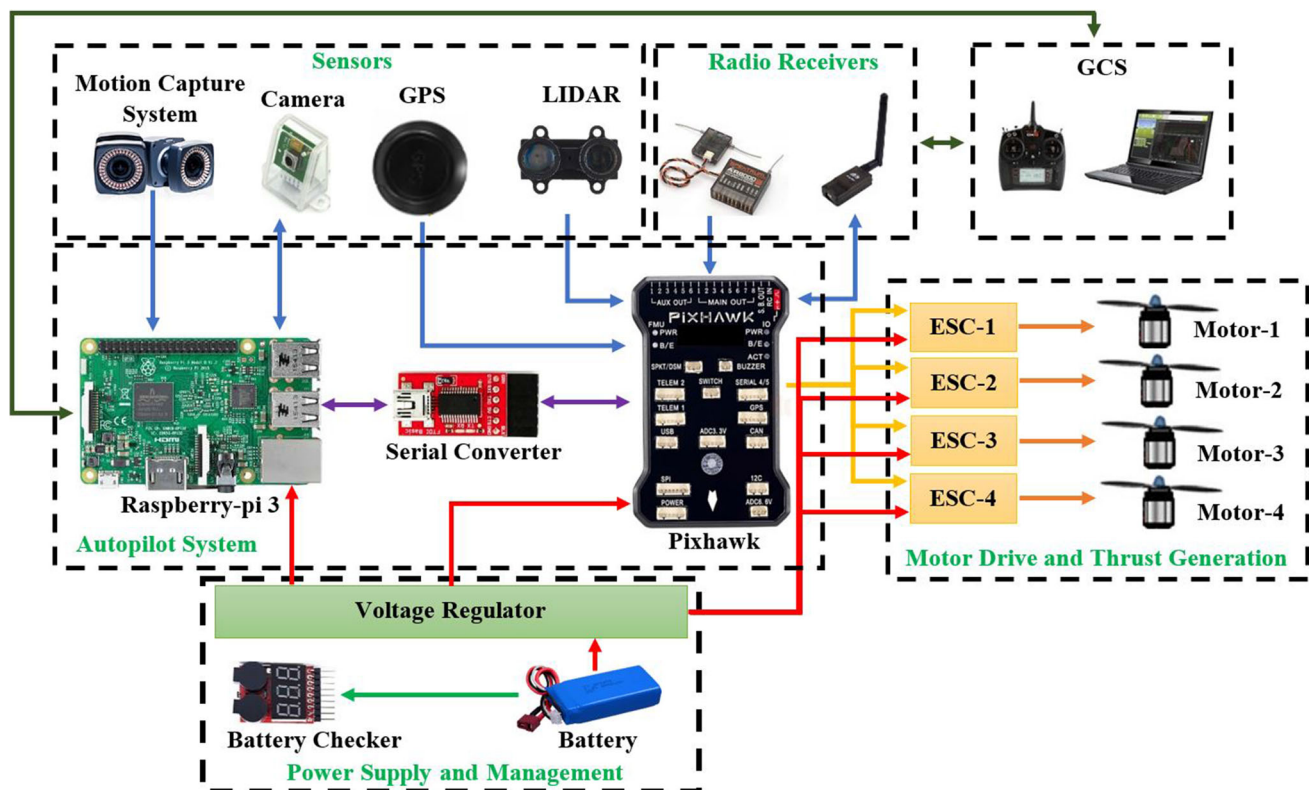


Fig. 1 The hardware structure of the Developed UAV

On the other hand, for the real-time experimental flight tests, a motion capture system is used for mimicking the GPS signals and estimating the location of the UAV.

2.2 Software Structure of the Developed UAV

The developed UAV adopts Robot Operating System (ROS) for high level operating system functionalities, PX4 firmware for flight control, and QgroundControl [31] for ground mission control. ROS has a powerful architecture which enables publish-subscribe data sharing features. Multiple applications on the companion computer can use ROS publish-subscribe bus to share data and communicate with the Pixhawk autopilot. MAVROS package is used for sending and receiving data using MAVLink protocol in a timely manner. Additionally, ROS also serves as a hub for receiving the UAV's sensor readings and sharing it with the GCS. The use of ROS in the developed UAV platforms allows for expanding the application horizon by easily adding application software, sensors, and communication modules. The PX4 firmware flight-stack architecture is used as a major autopilot software. The flight-stack supports effective inter-thread and inter-UAV communications, conducts state estimations, and generates control signals. The developed IT2 TSK FLC for controlling the attitude of the UAV is incorporated in the attitude controller block of the flight-stack. QgroundControl is used as a GCS software for programming the Pixhawk autopilot, remote commanding of the UAV, and data logging. The developed UAV along with its GCS is shown in Fig. 2.

3 Mathematical Model of a Quadcopter UAV

Consider two reference frames for the UAV, the body frame ($F^B = X_b, Y_b, Z_b$) and inertial frame ($F^I = X_i, Y_i, Z_i$),



Fig. 2 The developed UAV and its ground station

as shown in Fig. 3. The body frame is a coordinate frame located at the center of gravity of the UAV, its X-axis points in a forward direction, its Y-axis points towards the right side of the UAV, and its Z-axis points downside of the UAV. The inertial coordinate system's origin is fixed at a certain point (usually the home location of the UAV), and its X-axis, Y-axis, and Z-axis point North direction, East direction, and down toward the center of the earth, respectively. The position of the UAV in F^I is given as $\mathbf{p} = [x, y, z]^T$ and its attitude (the UAV's orientation with respect to the inertial frame) is captured by the Euler angles $\mathbf{o} = [\phi, \theta, \psi]^T$, which are named roll, pitch, and yaw, respectively. The linear velocities of the UAV, $[u, v, w]^T$, and its angular velocities, $[p, q, r]^T$, are defined as linear and angular velocities of F^B with respect to F^I .

Each motor of the UAV is controlled using a pulse width modulation (PWM) signal. Based on the width of the applied PWM signal, the revolutions per minute (RPM) of the motors creating a controlled thrust and torque. Equation 1 describes the relationship between the motor input PWM and the generated RPM.

$$w_i = K_m u_i + b \quad (1)$$

where w_i is the RPM output of i th motor, u_i is its PWM input, and K_m and b are motor coefficients. The motors are assumed to be identical whose coefficient b usually has a small value and can be ignored.

Attached to each motor of the UAV is a propeller. The rotation of the propellers creates the thrust and the torque to lift the UAV or rotate it in a particular direction. The generated total thrust and torque is related to the RPM of the motors as:

$$\begin{bmatrix} f_b \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = C \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \quad (2)$$

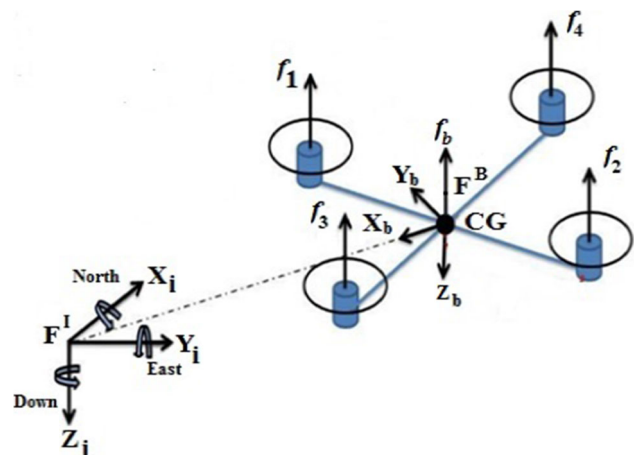


Fig. 3 The body and inertial frames for a UAV

where f_b is the total generated thrust applied at the center of gravity (CG) for lifting the UAV; τ_ϕ , τ_θ , and τ_ψ are the total torque generated around z , y , and x axes, and

$$C = \begin{bmatrix} C_T & C_T & C_T & C_T \\ -dC_T & dC_T & dC_T & -dC_T \\ dC_T & -dC_T & dC_T & -dC_T \\ C_Q & C_Q & -C_Q & -C_Q \end{bmatrix} \quad (3)$$

where C_T and C_Q are thrust coefficients, $d = \sqrt{2}/2\ell$, and ℓ is the length of the UAV arm.

The generated torque and trust result in linear and angular accelerations of the UAV, which can be mathematically driven in F^B as expressed in Eqs. 4 and 5:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -gc_\theta \\ gc_\theta s_\phi \\ gc_\theta c_\phi \end{bmatrix} + 1/m \begin{bmatrix} 0 \\ 0 \\ -f_b \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_\phi/j_{xx} \\ \tau_\theta/j_{yy} \\ \tau_\psi/j_{zz} \end{bmatrix} + \begin{bmatrix} rq(j_{yy} - j_{zz})/j_{xx} \\ pr(j_{zz} - j_{xx})/j_{yy} \\ pq(j_{xx} - j_{yy})/j_{zz} \end{bmatrix} \quad (5)$$

where j_{xx} , j_{yy} , and j_{zz} are respectively the moments of inertia along with the x , y , and z axes of the UAV's body frame, g is the gravitational acceleration, and m is the mass of the UAV.

The UAV's rates of changes in its Euler angles can be described as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (6)$$

and the UAV's linear velocities in F^I can be found by transforming its linear velocities from F^B as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (7)$$

Table 1 UAV's model parameters

Parameter	Measured Value
m	0.876 Kg
d	0.28 m
g	9.80665m/s ²
C_T	0.0065
C_Q	0.055
K_m, b	$K_m = 38, b = 87.23$
j_{xx}, j_{yy}, j_{zz}	$j_{xx} = 0.00512, j_{yy} = 0.00628, j_{zz} = 0.00612$

where c , s , and t are the abbreviations for cos, sin, and tan functions, and R is the transformation matrix, which can be expressed as:

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (8)$$

Figure 4 summarizes the mathematical equations of the UAV model and the relationships between different components of the model. The quadcopter UAV's model parameters are provided in Table 1.

4 Flight Control Architecture

Controlling a UAV includes controlling its position and attitude. The most important and most difficult of the two is controlling the attitude. Furthermore, the acceptable error margin for attitude control is very small compared to that of position control. This makes attitude control more challenging particularly in noisy or uncertain environments. Once a reliable attitude control is designed, the UAV's position can be controlled in an outer control loop. As

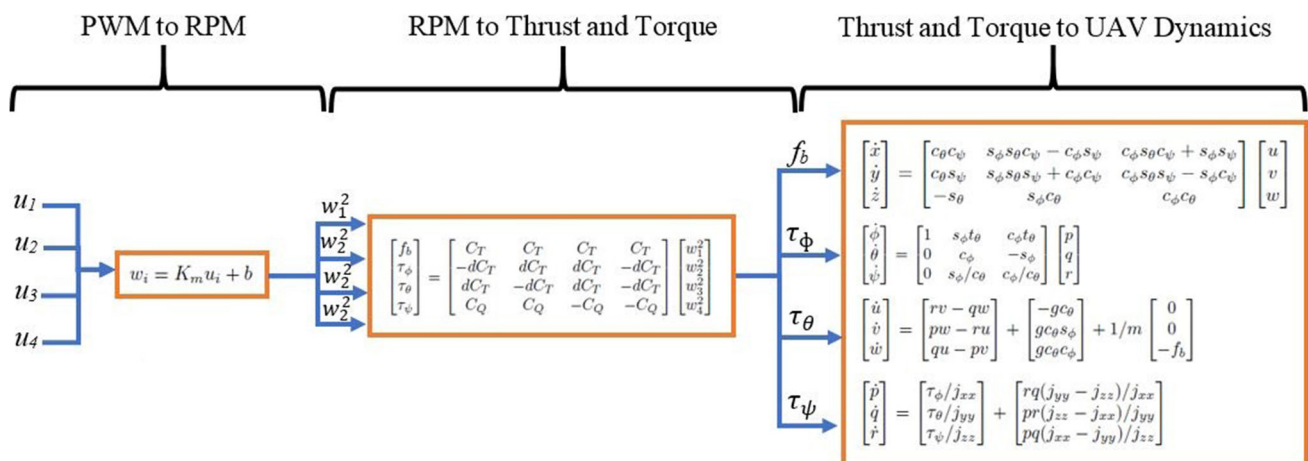


Fig. 4 Components of the mathematical model of a quadcopter UAV

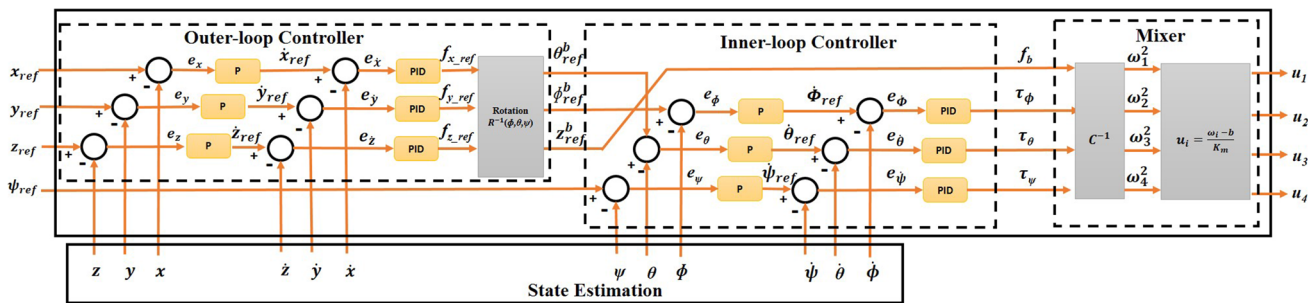


Fig. 5 PID attitude and position control structure of a UAV

presented in Fig. 5, the basic control structure of a UAV is composed of a position controller, commonly referred to as an outer-loop controller cascaded with an attitude controller, commonly referred to as an inner-loop controller.

The inputs to the outer-loop controller are the position reference-points (x_{ref} , y_{ref} , z_{ref}) as well as the sensor readings including the current position (x , y , z) and velocity of the UAV ($\dot{x}$, $\dot{y}$, $\dot{z}$). The outer-loop has two sets of controller: a P controller for the position control, and a PID controller for the velocities. The outputs of the position controllers are inputs for the velocity controllers ($\dot{x}_{ref}$, $\dot{y}_{ref}$, $\dot{z}_{ref}$). The outputs of the velocity controllers (The thrust outputs: $f_{x,ref}$, $f_{y,ref}$, $f_{z,ref}$) are then transformed to the body frame to serve as the attitude reference-points (ϕ_{ref} , θ_{ref} , ψ_{ref}) for the inner-loop attitude controller.

With a similar structure, the inputs to the inner-loop controller are the angle reference-points (ϕ_{ref} , θ_{ref} , ψ_{ref}) as well as the sensor readings including the current angles (ϕ , θ , ψ) and angular velocity of the UAV ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$). Further, the inner-loop has two sets of controllers: P controllers for angular controls, and a PID controllers for the angular velocities. the output of the angular controller are inputs for the angular velocity controllers ($\dot{\phi}_{ref}$, $\dot{\theta}_{ref}$, $\dot{\psi}_{ref}$).

The outputs of the angular velocity controllers (f_b , τ_ϕ , τ_θ , τ_ψ) are then passed to the *Mixer* to be converted back to control signals u_i , $i = 1, \dots, 4$, using Eqs. 1 and 2. Employing this multi-layer control structure, based on the desired reference-points as well as sensor inputs indicating

the UAV's position, attitude, velocity, and angular velocity, the outer-loop and inner-loop controllers together generate control outputs (f_b , τ_ϕ , τ_θ and τ_ψ) which are converted to control signals u_i , $i = 1, \dots, 4$, to command the actuator motors for rotating at a desired speed forcing the UAV to achieve the desired reference-points.

As stated, the attitude dynamics of UAVs are faster and more sensitive. If the attitude control of a UAVs is designed well, simple outer-loop controllers, e.g., PIDs, may be used for controlling their position. Therefore, by improving the attitude controller, it is possible to enhance the holistic performance of a UAV, which is highly challenged by uncertainties arising from noise, inaccuracy of the mathematical model, loss and delay of data, etc. To overcome these challenges, it is demanding to implement robust attitude controllers capable of accommodating such uncertainties and generating proper control actions.

5 Designing an IT2 TSK FLC for UAV Flight Control

In this paper, to enhance the flight performance of the developed UAV, we design an IT2 TSK FLC for the attitude controller. The control of the yaw angle is relatively straightforward and can be done with a control structure similar to what we have for the position control in the outer-loop. The proposed control structure using the IT2 TSK fuzzy logic controller for roll (ϕ) and pitch (θ) angles is

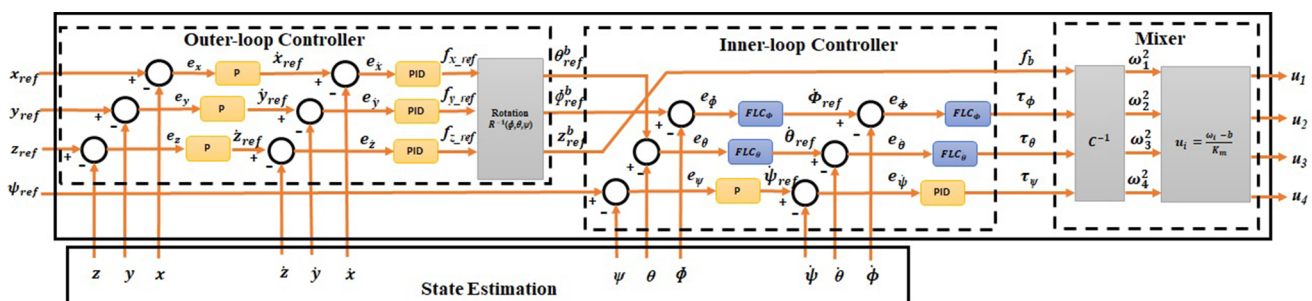
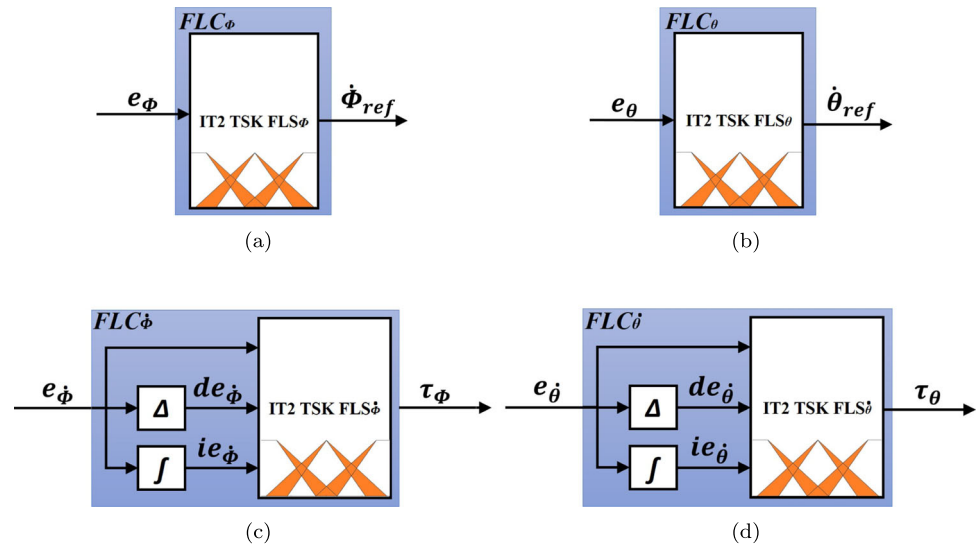


Fig. 6 Developed IT2 TSK fuzzy logic UAV attitude control structure of a UAV

Fig. 7 (a) IT2 FLC controller for ϕ angle, (b) IT2 FLC controller for θ angle, (c) IT2 FLC controller for $\dot{\phi}$, and (d) IT2 FLC controller for $\dot{\theta}$



presented in Fig. 6 to control the attitude of the UAV. As shown in Fig. 7, the inputs to the attitude FLCs are the roll error, e_ϕ , and pitch error, e_θ . Similarly, the inputs to the angular velocity FLCs are the roll rate error, $e_{\dot{\phi}}$, and pitch rate error, $e_{\dot{\theta}}$. Within the angular velocity FLCs the integral terms (integral of roll velocity error, $ie_{\dot{\phi}}$, and integral of pitch velocity error, $ie_{\dot{\theta}}$) and derivative terms (derivative of the roll velocity error, $de_{\dot{\phi}}$, and derivative of the pitch velocity error, $de_{\dot{\theta}}$) are constructed and fed to the IT2 TSK FLSs to generate the control signals.

The structure of an IT2 FLS system is shown in Fig. 8. Given the crisp inputs, the Fuzzifier block converts the crisp inputs into fuzzy inputs. The Rule Base block consists of a

set of rules in the form of fuzzy If-Then statements, which relate the fuzzy inputs to fuzzy outputs. To apply these fuzzy rules, having the fuzzified values of the inputs, two process should be done in parallel: (1) the Firing Level block calculates the firing level of each rule based on the structure of the rule's antecedent part, and (2) the Rule Output block calculates rule outputs, based on the structure of rule's consequent part and output membership functions. The Aggregator block combines the firing levels and rule outputs into an aggregated type-1 fuzzy set. Finally, the Defuzzifier block converts the obtained type-1 fuzzy set into a crisp output value. The details of the implementation of this control process are provided in the following sections.

In the proposed control structure, the inputs are captured and fuzzified using IT2 fuzzy memberships. An interval type-2 fuzzy set, F , is a type-2 fuzzy set in which the secondary membership values of its elements, $\mu_F(x, u)$, are always unity, formally defined as:

$$F = \{((x, u), \mu_F(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1], \mu_F(x, u) = 1\} = \sum_{u \in J_x} \sum_{x \in X} ((x, u), 1) \quad (9)$$

where X is the input domain, u is the primary membership value, J_x is the range of the primary membership function, μ_F is the secondary membership function. Since the secondary membership function of an IT fuzzy set is always set to unity, it can be characterized by its footprint. The footprint of an IT2 fuzzy set can be captured by its lower upper and lower membership functions, $\bar{F}$ and $\underline{F}$. Figure 9 shows an example of an IT2 fuzzy set and its footprint.

The footprint of IT2 fuzzy sets for e_ϕ and e_θ are shown in Fig. 10a. Similarly, the footprint of IT2 fuzzy sets for fuzzifying the errors in the proportional, integrals and

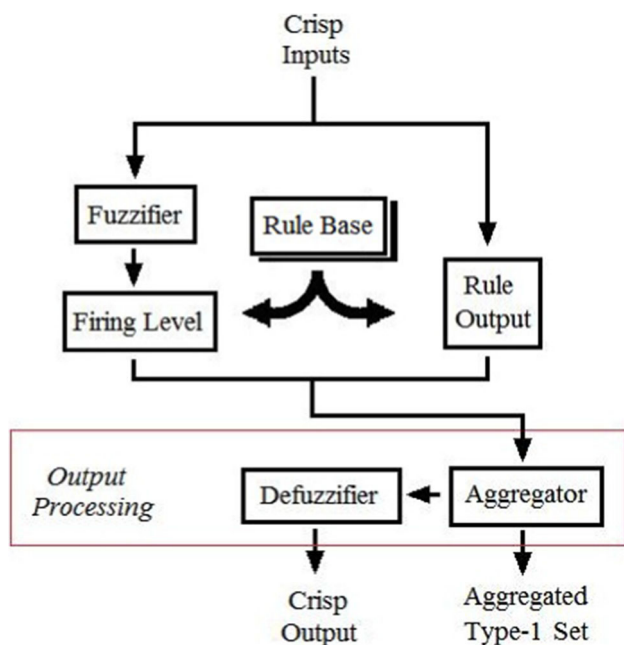


Fig. 8 The general structure of an IT2 FLS

Fig. 9 (a) An IT2 fuzzy set, (b) The footprint of the IT2 fuzzy set, and its lower and upper membership functions

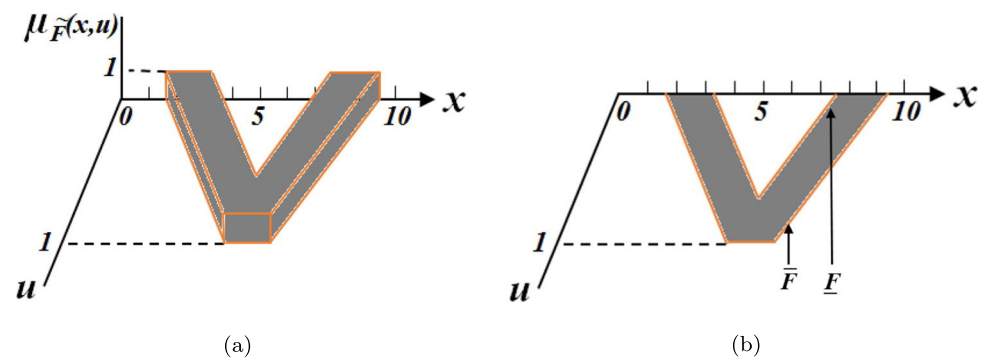


Fig. 10 Footprints of IT2 MFs for attitude and angular velocity errors

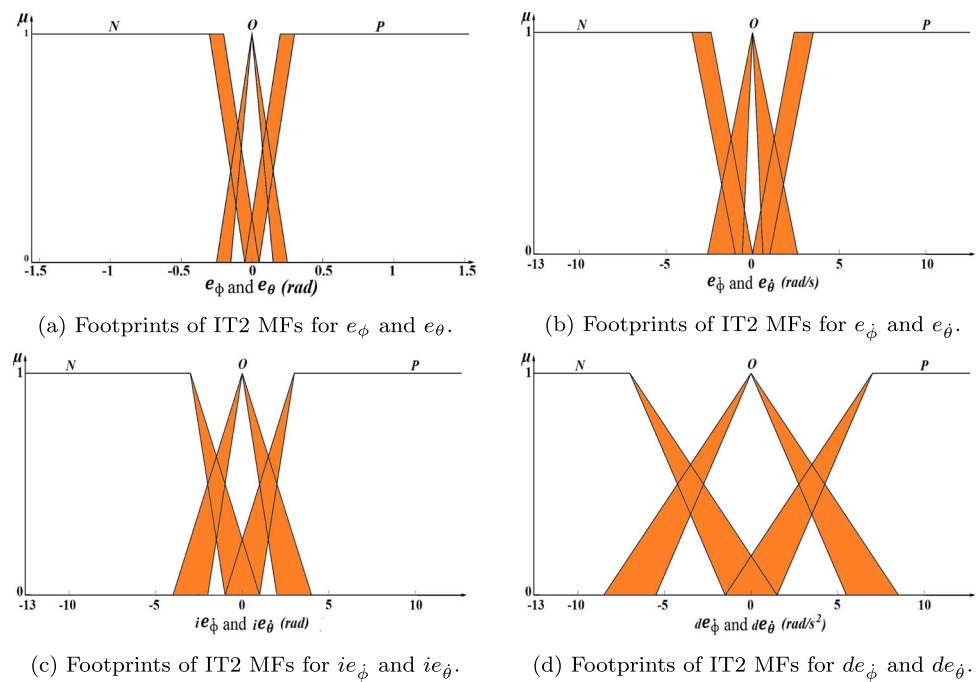
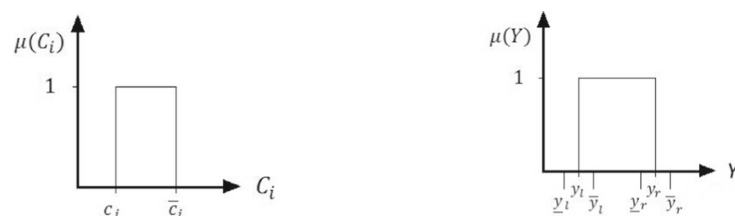


Fig. 11 IT2 TSK consequent coefficient bound and output fuzzy sets



(a) A Type-1 fuzzy set for output coefficient C_i^ℓ . (b) The output of the inference process of an IT2 FLS.

Table 2 Rule base for $\dot{\phi}$ and $\dot{\theta}$ control

ℓ	$e_{\dot{\phi}}, e_{\dot{\theta}}$	$ie_{\dot{\phi}}, ie_{\dot{\theta}}$	$de_{\dot{\phi}}, de_{\dot{\theta}}$	$\tau_{\phi}, \tau_{\theta}$	ℓ	$ie_{\dot{\phi}}, ie_{\dot{\theta}}$	$ie_{\dot{\phi}}, ie_{\dot{\theta}}$	$ie_{\dot{\phi}}, ie_{\dot{\theta}}$	$\tau_{\phi}, \tau_{\theta}$
1	P	P	P	L	15	O	O	N	S
2	P	P	O	L	16	O	N	P	MS
3	P	P	N	ML	17	O	N	O	MS
4	P	O	P	L	18	O	N	N	MS
5	P	O	O	L	19	N	P	P	MS
6	P	O	N	ML	20	N	P	O	ML
7	P	N	P	ML	21	N	P	N	ML
8	P	N	O	ML	22	N	O	P	ML
9	P	N	N	MS	23	N	O	O	L
10	O	P	P	MS	24	N	O	N	L
11	O	P	O	MS	25	N	N	P	ML
12	O	P	N	MS	26	N	N	O	L
13	O	O	P	S	27	N	N	N	L
14	O	O	O	S	—	—	—	—	—

derivatives of the angular velocities are shown in Fig. 10. The symbols N, O, and P represent the labels for the membership functions, which indeed are the positions of the errors in linguistic form as Negative, Zero, and Positive, respectively.

Using the fuzzy inference process, these fuzzified input errors are then mapped to TSK output functions based on rules predefined in the Rule Base. For example, the ℓ th rule, with p inputs and one output, can be described as

$$R^{\ell}: \text{IF } x_1 \text{ is } F_1^{\ell}, \text{ and } x_2 \text{ is } F_2^{\ell} \text{ and } \dots \text{ and } x_p \text{ is } F_p^{\ell} \\ \text{THEN } y^{\ell} = c_0^{\ell} + c_1^{\ell}x_1 + \dots + c_p^{\ell}x_p$$

where, F_p^{ℓ} is the activated antecedent fuzzy set for input channel x_p and $c_0, c_1 \dots c_p$ are TSK output coefficients for the output signal. An example rule for IT2 TSK roll control can be described as,

$$R^1: \text{IF } e_{\dot{\phi}} \text{ is } S \text{ THEN } \dot{\phi}_{ref} = c_1^1 e_{\dot{\phi}}$$

Similarly, an example rule for IT2 TSK roll velocity controller can be described as

$$R^1: \text{IF } e_{\dot{\phi}} \text{ is } N \text{ and } ie_{\dot{\phi}} \text{ is } O \text{ and } de_{\dot{\phi}} \text{ is } P \text{ THEN} \\ R^1 = c_1^1 e_{\dot{\phi}} + c_2^1 ie_{\dot{\phi}} + c_3^1 de_{\dot{\phi}}$$

Table 3 Rule coefficient bounds for roll and pitch velocity control

Rule Output Coefficient Label	$\underline{c}_1$	$\overline{c}_1$	$\underline{c}_2$	$\overline{c}_2$	$\underline{c}_3$	$\overline{c}_3$
S	0.1	0.12	0.015	0.02	0.0005	0.0008
MS	0.12	0.15	0.023	0.03	0.0005	0.0008
ML	0.12	0.15	0.03	0.04	0.0008	0.001
L	0.15	0.18	0.04	0.06	0.0015	0.003

The above rule consequent structure is due to the fact that the IT2 TSK rule outputs of the attitude controller resemble a proportional (P) controller, while the IT2 TSK rule outputs of the angular velocity controller resemble a PID controller. In both cases, we set c_0 to zero. To capture the uncertainties in the outputs, we consider the coefficients C_i^{ℓ} as interval Type-1 fuzzy sets, bounded by $\underline{C}_i^{\ell}$ and $\overline{C}_i^{\ell}$ as shown in Fig. 11a.

The fuzzy rule base and the output uncertainty coefficient bounds for the roll and pitch velocity controls are presented in Tables 2 and 3. The fuzzy rule base and the output uncertainty coefficient bounds, $\underline{c}$ and $\overline{c}$, for the attitude controller are presented in Tables 4 and 5, respectively. For all rules, the TSK output coefficient c_0 is assumed to be zero. As some of the rules have the same output coefficients, rather than repeating the coefficients for each individual rule, we have grouped them and labeled them as S, MS, ML, and L (Tables 3 and 5).

For obtaining the final crisp control output to apply to actuators, an output fuzzy set has to be inferred, type reduced, and then defuzzified, which is a computationally expensive process. In [12], we have developed a computationally effective for calculating the output signal. As shown in [12], the output of the inference process of an IT2 FLS is an interval type-1 fuzzy set, depicted in Fig. 11b, which

Table 4 Rule base for ϕ and θ control

ℓ	Membership Function Label	$\dot{\phi}_{ref}, \dot{\theta}_{ref}$
1	N	L
2	O	S
3	P	L

Table 5 Rule coefficient bounds for ϕ and θ control

Rule Output Coefficient Label	$\underline{c}_1$	$\overline{c}_1$
S	6.0	6.9
L	6.5	7.6

is bounded by y_r and y_l . These bounds can be calculated based on the lower and upper bound membership functions of footprints of input IT2 fuzzy sets as well as the lower and upper bounds of fuzzy sets of the output coefficients. Adopting the uncertainty bound technique [45], the proposed technique estimates y_l and y_r by calculating and averaging their upper and lower bounds values, $\underline{y}_l$, $\overline{y}_l$, $\underline{y}_r$, and $\overline{y}_r$.

Following the fuzzification process, the lower and upper bounds of the firing level for the i th rule can be computed as

$$\underline{f}^i = \underline{F}_1^i(x_1) * \underline{F}_2^i(x_2) * \dots * \underline{F}_p^i(x_p) \quad (10)$$

$$\overline{f}^i = \overline{F}_1^i(x_1) * \overline{F}_2^i(x_2) * \dots * \overline{F}_p^i(x_p) \quad (11)$$

where f^i is the firing level of the i th rule and $F_1, F_2, \dots, F_p$ are the activated antecedent fuzzy sets for input channels $x_1, x_2, \dots, x_p$.

Employing the uncertainty bound technique [45], the inner upper and lower bounds, $\overline{y}_l$ and $\underline{y}_r$, of y_l and y_r are calculated as follows:

$$\overline{y}_l = \min\{y_{ll}, y_{ul}\}, \quad (12)$$

$$y_r = \max\{y_{lr}, y_{ur}\} \quad (13)$$

where,

$$y_{ll} = \frac{\underline{f}^1 y_l^1 + \dots + \underline{f}^M y_l^M}{\underline{f}^1 + \dots + \underline{f}^M} \quad (14)$$



Fig. 12 Snapshot of SITL simulation in Gazebo simulation environment

$$y_{ul} = \frac{\overline{f}^1 y_l^1 + \dots + \overline{f}^M y_l^M}{\overline{f}^1 + \dots + \overline{f}^M} \quad (15)$$

$$y_{lr} = \frac{\underline{f}^1 y_r^1 + \dots + \underline{f}^M y_r^M}{\underline{f}^1 + \dots + \underline{f}^M} \quad (16)$$

$$y_{ur} = \frac{\overline{f}^1 y_r^1 + \dots + \overline{f}^M y_r^M}{\overline{f}^1 + \dots + \overline{f}^M} \quad (17)$$

With the inner bounds calculated, the outer bounds, $\underline{y}_l$ and $\overline{y}_r$, can then be found using Eqs. 18 and 19 as

$$\underline{y}_l = \overline{y}_l - \left[\frac{\sum_{i=1}^M (\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^M \overline{f}^i \times \sum_{i=1}^M \underline{f}^i} \times \frac{\sum_{i=1}^M \underline{f}^i (y_l^i - y_l^1) \times \sum_{i=1}^M \overline{f}^i (y_l^M - y_l^i)}{\sum_{i=1}^M \underline{f}^i (y_l^i - y_l^1) + \sum_{i=1}^M \overline{f}^i (y_l^M - y_l^i)} \right] \quad (18)$$

$$\overline{y}_r = \underline{y}_r + \left[\frac{\sum_{i=1}^M (\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^M \overline{f}^i \times \sum_{i=1}^M \underline{f}^i} \times \frac{\sum_{i=1}^M \overline{f}^i (y_r^i - y_r^1) \times \sum_{i=1}^M \underline{f}^i (y_r^M - y_r^i)}{\sum_{i=1}^M \overline{f}^i (y_r^i - y_r^1) + \sum_{i=1}^M \underline{f}^i (y_r^M - y_r^i)} \right] \quad (19)$$

The lower and upper bounds of y_l and y_r can then be estimated using Eqs. 20 and 21.

$$y_l = \frac{\underline{y}_l + \overline{y}_l}{2} \quad (20)$$

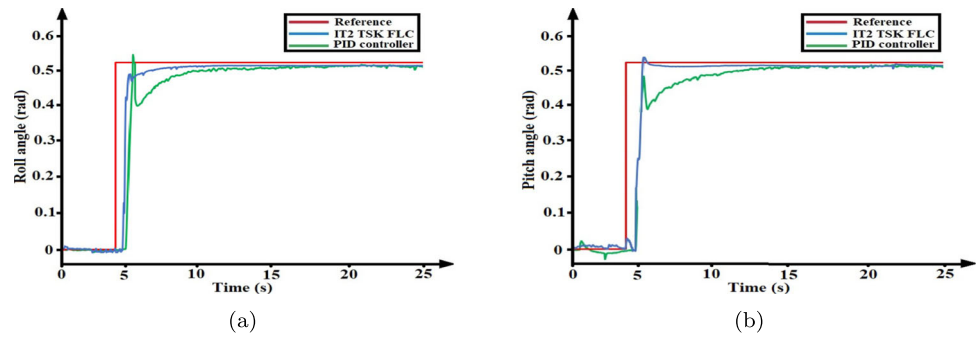
$$y_r = \frac{\underline{y}_r + \overline{y}_r}{2} \quad (21)$$

Finally, the final crisp output can be obtained using Eq. 22:

$$y = \frac{y_l + y_r}{2} \quad (22)$$

With this method, $y = \dot{\phi}_{ref}$ and $y = \dot{\theta}_{ref}$ can be calculated as the outputs of FLC_ϕ and FLC_θ , which are applied to $FLC_{\dot{\phi}}$ and $FLC_{\dot{\theta}}$. Then, $y = \tau_\phi$ and $y = \tau_\theta$, as the outputs of $FLC_{\dot{\phi}}$ and $FLC_{\dot{\theta}}$ along with other outputs of the inner-loop including τ_ψ and f_b will be provided to the the Mixer to calculate the control signals u_i , $i = 1, \dots, 4$, as shown in Fig. 6.

Fig. 13 UAV SITL attitude flight control responses when using the developed IT2 TSK FLC and PID controllers: (a) Roll angle control response, (b) Pitch angle control response



6 Simulation and Flight Test Results

In this section, we evaluate the effectiveness of the developed IT2 TSK fuzzy logic flight controller using both SITL simulation and actual flight tests. The SITL simulation platform and the quadcopter UAV hardware and software architectures were discussed in Section 2. A normal noise, with mean of 0 rads and standard deviation, 0.05 was applied to the UAV attitude sensor readings. Similarly, a normal noise, with mean of 0 meters and standard deviation, 0.05 was applied to the position sensor readings. The simulation and flight test results of the developed IT2 TSK FLC were then compared with a classical PID controller. To make the comparison fair, the gains of the PID controller were set at the centers of membership functions for the output coefficients of the implemented IT2 TSK FLC. The attitude control performance of the UAV directly affects the position control performance. Therefore, both attitude and position control response data are provided.

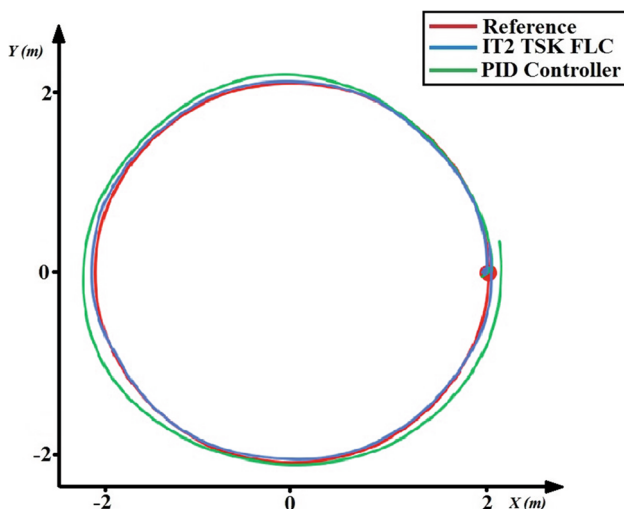


Fig. 14 UAV SITL for circular trajectory tracking when using the developed IT2 TSK FLC and PID controller

6.1 SITL Simulation Tests

The SITL simulation tests were conducted for both attitude and position control in Gazebo simulation environment, shown in Fig. 12. First, to evaluate the performance of the developed IT2 TSK FLC for the attitude control of the UAV, we changed the desired reference signal for the UAV in roll and pitch channels for 0.5 rad, resulting in UAV horizontal moves. The step responses of the attitude controller of the UAV in roll and pitch channels are shown in Fig. 13. From the attitude control responses of the UAV, it can be seen that both the IT2 TSK FLC and the PID controllers were able to reach the desired set point. However, it has been found that the IT2 TSK FLC was able to control the attitude of the UAV with a smaller overshoot and faster settling time.

Then, to evaluate the performance of the developed IT2 TSK FLC for the position control of the UAV and its capability of trajectory tracking, a circular trajectory with the radius of 2 meters was provided to the UAV. The UAV path tracking results for both the IT2 TSK FLC and the PID controllers are shown in X-Y plane in Fig. 14. From these simulation results, it can be seen again that the developed IT2 TSK FLC has successfully controlled the UAV to follow the desired trajectory better than the PID controller, with smaller tracking error. To statistically quantify the performance differences, Tables 6 and 7 present the computed root mean square (RMS) errors when the two controllers are employed. From Table 6, it can be seen that the IT2 TSK FLC demonstrated 18.7% and 27.4% performance improvements in roll (ψ) and pitch (θ) angle controls, respectively. Similarly, as presented in Table 7, compared to the PID controller, 24.1% and 20.1% improvements are achieved during circular path position control in the X and Y axes, respectively when employing the IT2 TSK FLC.

6.2 Flight Test Results

The IT2 TSK FLC was implemented and tested through actual real-time autonomous indoor flight control of the

Table 6 SITL simulation RMS errors results for the attitude control

	Controller Type	$RMSE_{\phi}$	$RMSE_{\theta}$
Attitude	IT2 TSK FLC	0.0425	0.0396
Control	PID	0.0513	0.0522

Table 7 SITL simulation RMS error results for the circular path tracking

	Controller Type	$RMSE_X$	$RMSE_Y$
Circular Path	IT2 TSK FLC	0.2650	0.2499
Tracking	PID	0.3378	0.3097

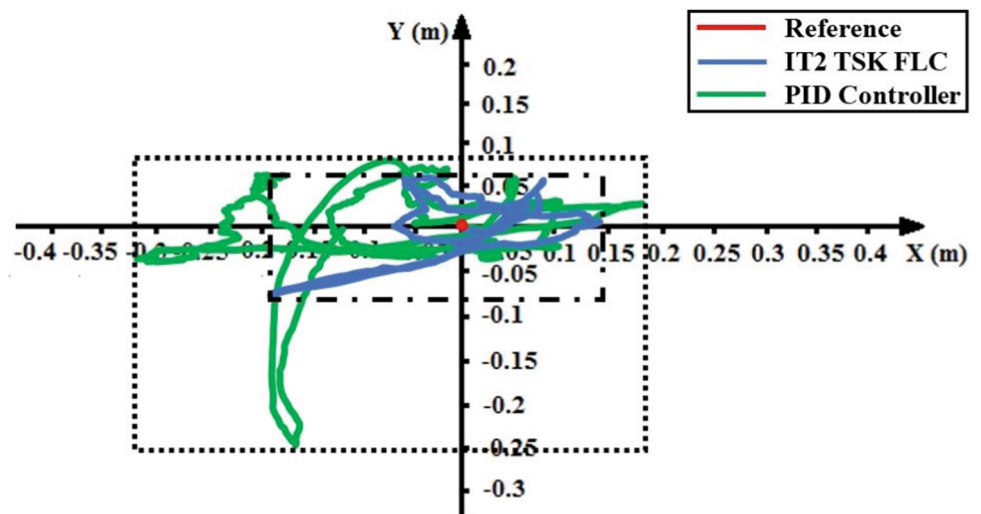
Fig. 15 Hovering flight test results in $X - Y$ plane

Fig. 16 (a) Hovering position control flight test at $(x = 0, y = 0)$ followed by the change of the hovering set point to $(x = 2, y = 0)$, (b) Hovering position control flight test at $(x = 0, y = 0)$, followed by the change of the hovering set point to $(x = 0, y = 2)$

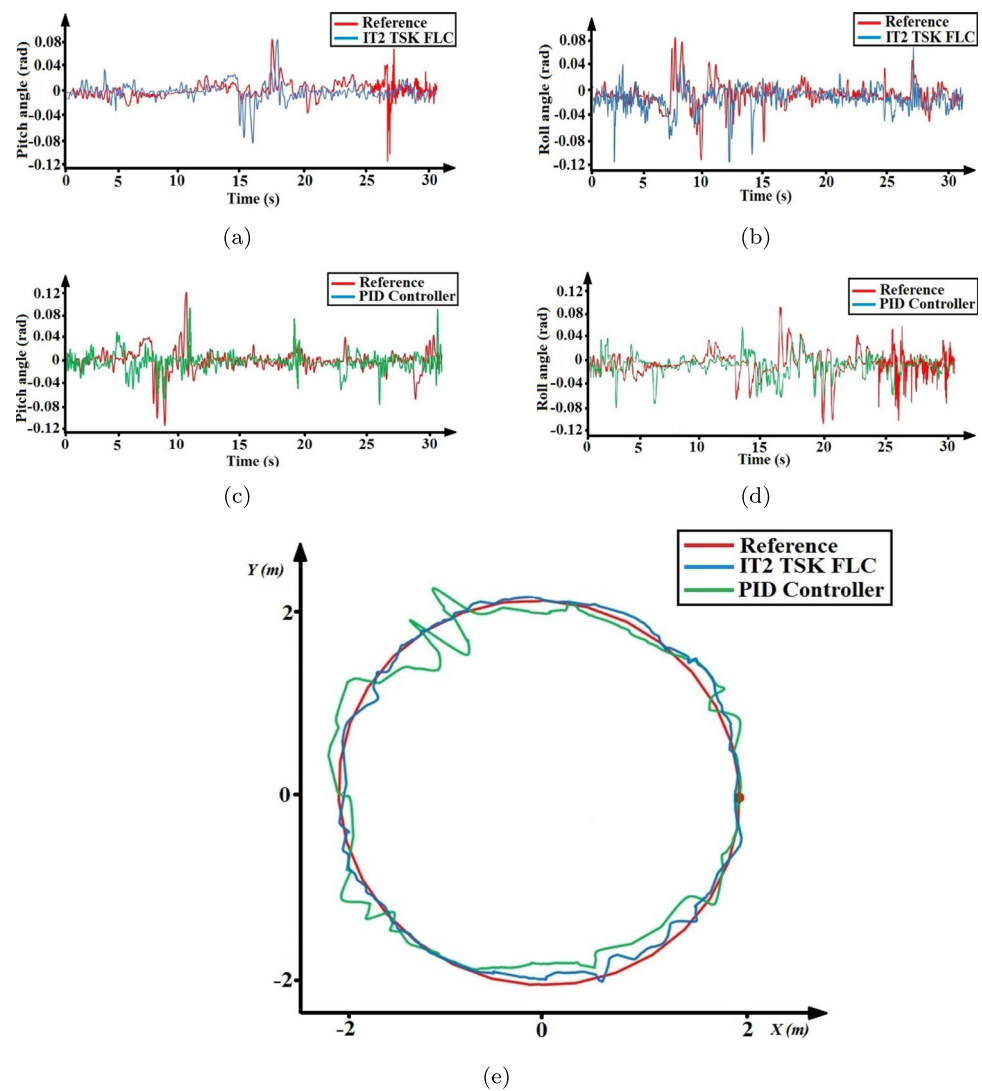
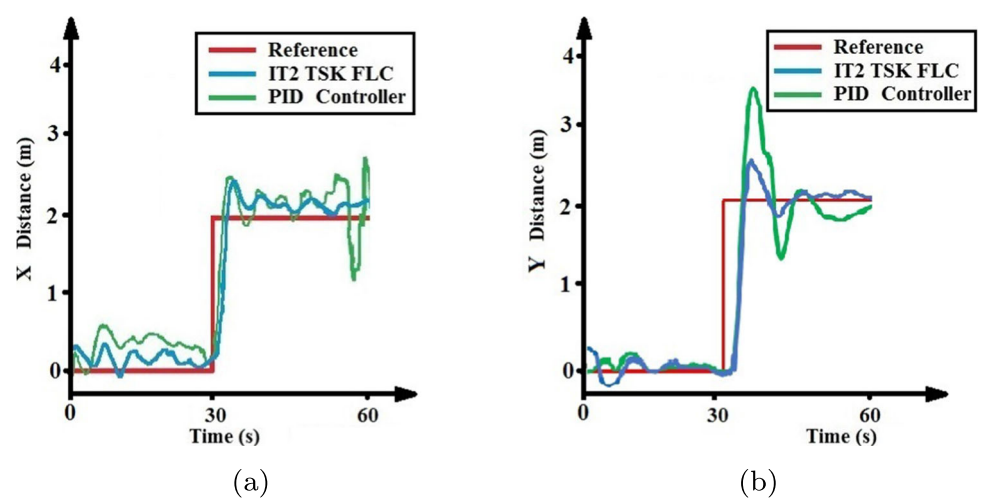


Fig. 17 UAV real-time circular trajectory flight responses when using the developed IT2 TSK FLC and PID controller



developed UAV, where the UAV was exposed to uncertain environmental disturbances arising from sensor limitations, electromagnetic fields, etc. The UAV position data was provided by a motion capture system.

In the first experiment, the UAV hovers at the hovering point at $(x = 0, y = 0)$ for 32 seconds, and then, the hovering set point is changed to $(x = 2, y = 0)$, driving the UAV for 2 meters in X direction. The test results for this experiment is shown in Fig. 16.(a). Further, for the first hovering part at $(x = 0, y = 0)$ for 32 sec, the UAV position is shown in $X - Y$ plane in Fig. 15. In the second experiment, again the UAV hovers at the hovering point at $(x = 0, y = 0)$ for 32 sec, and then, the hovering set point is changed to $(x = 0, y = 2)$, driving the UAV for 2 meters in Y direction. The test results for this experiment is shown in Fig. 16.(b).

The videos of these two experiments for moving in X and Y directions are available at <https://youtu.be/2Cx90DRT1OQ> and <https://youtu.be/5GsPMNG0ptI>, respectively. Note that these sudden changes of hovering point resemble a disturbance that suddenly push away the UAV from its hovering point, requiring the UAV to compensate the impact of the disturbance and braining the UAV to the hovering point.

In the third flight experiment, the UAV is driven to track a circular trajectory with the radius of 2 meters, for which the flight test results are provided in Fig. 17. The video of this experiment is available at <https://youtu.be/jdlhbx-ef5Q>.

Similar to the simulation results, the actual flight tests also confirm that the IT2 TSK FLC performs better than the PID controller. The RMS errors for hovering flight test and circular trajectory tracking are presented in Table 8. From these errors, it can be seen that the IT2 TSK FLC demonstrated 27.0% and 16.25% performance improvements in X and Y hover position controls, respectively. Similarly, compared to the PID controller, 23.7% and 18.9% performance improvements are achieved during circular path position control in the X and Y axes, respectively when employing the IT2 TSK FLC.

Table 8 RMS errors for the flight test results for the hovering and circular path tracking

	Controller Type	$RMSE_X$	$RMSE_Y$
Hovering	IT2 TSK FLC	0.1165	0.0955
	PID	0.1529	0.1254
Circular Path Tracking	IT2 TSK FLC	0.3977	0.3563
	PID	0.5050	0.4307

7 Conclusion

In this paper, we developed a robust quadcopter UAV platform. The hardware and software architectures and system components of the UAV were presented. To enhance the UAV's control performance in uncertain environments, IT2 TSK fuzzy logic attitude and angular velocity controller architectures were developed and implemented. Uncertainty bounds output processing method was used for making the IT2 TSK FLC computationally effective. The technique expresses antecedent parts of rules using IT2 MFs and consequent using first-order linear functions of inputs. Output processing of the input fuzzy set was performed by using only the upper and lower bounds of membership functions and TSK rule output coefficients. These arrangements enhanced the robustness of the control structure, while reducing the computation costs, making it appropriate for real-time control developments. The effectiveness of the developed flight control system was then demonstrated through a SITL simulation and real-time flight tests. Additionally, the performance of the developed IT2 TSK fuzzy logic attitude controller was compared with that of a PID controller in the presence of uncertainties. From the flight test results, it was found that the developed UAV performed better with smaller overshoot, faster settling time, less tracking error, and less RMS error when the IT2 TSK fuzzy logic attitude controller was employed.

Acknowledgements The second author would like to acknowledge the support from the National Science Foundation under the award number 1832110 and Sandia National Laboratories under the contract number 2086781.

Author Contributions A. Karimoddini contributed to the conceptualization and supervision of the research, writing (review & editing) of the paper, and funding acquisition. A. Hailemichael contributed to the development of methodology and software as well as writing the original draft. The manuscript was read and approved by all authors.

Funding This research is supported by Air Force Research Laboratory and Office of the National Science Foundation under the award number 1832110 and Sandia National Laboratories under the contract number 2086781. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of OSD, NSF, or the U.S. Government.

Data Availability The data used to support the findings of this study are available from the corresponding author upon request.

Code Availability The software used to support the findings of this study are available from the corresponding author upon request.

Declarations

Ethics approval Not applicable (this article does not contain any studies with human participants or animals performed by any of the authors).

Consent to participate Not applicable (this article does not human participants performed by any of the authors).

Consent for Publication Not applicable (this article does not contain any studies with human participants performed by any of the authors).

Conflict of Interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Aghdam, A.S., Menhaj, M.B., Barazandeh, F., Abdollahi, F.: Cooperative load transport with movable load center of mass using multiple Quadrotor Uavs. In: 2016 4Th International Conference on Control, Instrumentation, and Automation (ICCIA), pp. 23–27 (2016)
2. Akram, T., Awais, M., Naqvi, R., Ahmed, A., Naeem, M.: Multicriteria uav base stations placement for disaster management. *IEEE Syst. J.* **14**(3), 3475–3482 (2020)
3. Al-Mahturi, A., Santoso, F., Garratt, M.A., Anavatti, S.G.: Nonlinear altitude control of a Quadcopter drone using interval type-2 fuzzy logic. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 236–241. IEEE (2018)
4. Al-Mahturi, A., Santoso, F., Garratt, M.A., Anavatti, S.G.: Chapter 2 - Modeling and control of a Quadrotor unmanned aerial vehicle using type-2 fuzzy systems. In: Koubaa, A., Azar, A.T. (eds.) *Unmanned Aerial Systems, Advances in Nonlinear Dynamics and Chaos (ANDC)*, pp. 25–46. Academic Press (2021)
5. Basso, M., de Freitas, E.P.: A uav guidance system using crop row detection and line follower algorithms. *J. Intell. Robot. Syst.* **97**(3), 605–621 (2020)
6. Camci, E., Kayacan, E.: Game of Drones: Uav Pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 618–625. IEEE (2016)
7. Canbek, K.O., Oniz, Y.: Real-time implementation of an interval Type-2 fuzzy logic controller for the trajectory tracking of an Uav. In: 2021 5Th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 418–423 (2021). <https://doi.org/10.1109/ISMSIT52890.2021.9604539>
8. Castillo, O., Amador-Angulo, L., Castro, J.R., Garcia-Valdez, M.: A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Inform. Sci.* **354**, 257–274 (2016)
9. Castillo, O., Cervantes, L., Soria, J., Sanchez, M., Castro, J.R.: A generalized type-2 fuzzy granular approach with applications to aerospace. *Inform. Sci.* **354**, 165–177 (2016)
10. Chaoui, H., Gueaieb, W.: Type-2 fuzzy logic control of a flexible-joint manipulator. *J. Intell. Robot. Syst.* **51**(2), 159–186 (2008)
11. Doitsidis, L., Valavanis, K.P., Tsourveloudis, N.C., Kontitsis, M.: A framework for fuzzy logic based Uav navigation and control. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference On*, vol. 4, pp. 4041–4046 vol.4. IEEE (2004)
12. Enyinna, N., Karimodini, A., Opoku, D., Homaifar, A., Arnold, S.: Developing an interval type-2 tsk fuzzy logic controller. *Fuzzy Information Processing Society NAFIPS* **5**(4), 1–6 (2015)
13. Fan, J., Li, D., Li, R., Yang, T., Wang, Q.: Analysis for cooperative combat system of manned-unmanned aerial vehicles and combat simulation. In: 2017 IEEE International Conference on Unmanned Systems (ICUS), pp. 204–209 (2017)
14. Girma, A., Bahadori, N., Sarkar, M., Tadewos, T.G., Behnia, M.R., Mahmoud, M.N., Karimodini, A., Homaifar, A.: Iot-enabled autonomous system collaboration for disaster-area management. *IEEE/CAA Journal of Automatica Sinica* **7**(5), 1249–1262 (2020)
15. Hailemichael, A., Salaken, S.M., Karimodini, A., Homaifar, A., Abbas, K., Nahavandi, S.: Developing a computationally effective interval type-2 tsk fuzzy logic controller. *Journal of Intelligent & Fuzzy Systems* **38**(2), 1915–1928 (2020)
16. Hamrawi, H., Coupland, S.: Measures of uncertainty for type-2 fuzzy sets. *Computational Intelligence (UKCI), 2010 UK Workshop* 1–7 (2010)
17. HekmatiAthar, S., Goudarzi, N., Karimodini, A., Homaifar, A., Divakaran, D.: A systematic evaluation and selection of Uas-Enabled solutions for bridge inspection practices. In: 2020 IEEE Aerospace Conference, pp. 1–11 (2020)
18. Hoang, V.T., Phung, M.D., Dinh, T.H., Ha, Q.P.: System architecture for real-time surface inspection using multiple uavs. *IEEE Syst. J.* **14**(2), 2925–2936 (2020)
19. Karnik, N.N., Mendel, J.M.: Applications of type-2 fuzzy logic systems to forecasting of time-series. *Inform. Sci.* **120**(1), 89–111 (1999)
20. Kim, H., Ben-Othman, J.: A collision-free surveillance system using smart uavs in multi domain iot. *IEEE Commun. Lett.* **22**(12), 2587–2590 (2018)
21. Koubaa, A.: *Robot Operating System (ROS) The Complete Reference*, vol. 2. Springer, Berlin (2017)
22. Kwon, Y., Baek, H., Lim, J.: Uplink noma using power allocation for uav-aided csma/ca networks. *IEEE Syst. J.* 1–4 (2020)
23. Le, T.L., Quynh, N.V., Long, N.K., Hong, S.K.: Multilayer interval type-2 fuzzy controller design for quadcopter unmanned aerial vehicles using jaya algorithm. *IEEE Access* **8**, 181246–181257 (2020)
24. Li, J., John, R., S.C., Kendall, G.: On nie-tan operator and type-reduction of interval type-2 fuzzy sets. *IEEE Trans. Fuzzy Syst.* **PP**(99), 1–13 (2017)
25. Liang, Q., Mendel, J.M.: An introduction to type-2 tsk fuzzy logic systems, pp. 1534–1539 (1999)
26. Liang, Q., Mendel, J.M.: Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *Fuzzy Systems IEEE Transactions on* **8**(5), 551–563 (2000)
27. Liu, X., Zhong-Ren, P., Li-Ye, Z.: Real-time uav rerouting for traffic monitoring with decomposition based multi-objective optimization. *Journal of Intelligent & Robotic Systems* **94**(2), 491–501 (2019)
28. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Stud.* **7**(1), 1–13 (1975)
29. Meier, L., Honegger, D., Pollefeys, M.: Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6235–6240. IEEE (2015)
30. Meier, L., Tanskanen, P., Fraundorfer, F., Pollefeys, M.: Pixhawk: A system for autonomous flight using onboard computer vision. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2992–2997. IEEE (2011)

31. Meier, L., Tanskanen, P., Heng, L., Lee, G.H., Fraundorfer, F., Pollefeys, M.: Pixhawk: A Micro Aerial vehicle design for autonomous flight using onboard computer vision. In: Masters Thesis. ETH Zurich (2013)
32. Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O.: Voice recognition with neural networks, type-2 fuzzy logic and genetic algorithms. *Engineering Letters* 13(3) (2006)
33. Mendel, J.M.: Uncertain rule-based fuzzy logic system: introduction and new directions. Prentice-Hall PTR (2001)
34. Mendel, J.M., John, R.B.: Type-2 Fuzzy Sets Made Simple, pp. 117–127. IEEE, New York (2002)
35. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. *IEEE Trans. Fuzzy Syst.* **14**(6), 808–821 (2006)
36. Namvar, N., Homaifar, A., Karimoddini, A., Maham, B.: Heterogeneous uav cells: An effective resource allocation scheme for maximum coverage performance. *IEEE Access* **7**, 164708–164719 (2019)
37. Ontiveros-Robles, E., Melin, P., Castillo, O.: Comparative analysis of noise robustness of type 2 fuzzy logic controllers. *Kybernetika* **54**(1), 175–201 (2018)
38. Quigley, M., Gerkey, B., Conley, K., Fausty, J., Foote, T., Leibs, J., Bergery, E., Wheeler, R., Ng, A.: Ros: an Open-Source Robot Operating System. In: Masters Thesis. Stanford University and University of Southern California (2013)
39. Ramasamy, M., Ghose, D.: A heuristic learning algorithm for preferential area surveillance by unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems* **88**(2–4), 655 (2017)
40. Sarabakha, A., Fu, C., Kayacan, E., Kumbasar, T.: Type-2 fuzzy logic controllers made even simpler: From design to deployment for uavs. *IEEE Trans. Ind. Electron.* **65**(6), 5069–5077 (2017)
41. Stellakis, H.M., Valavanis, K.P.: Fuzzy logic-based formulation of the organizer of intelligent robotic systems. *J. Intell. Robot. Syst.* **4**(1), 1–24 (1991)
42. Stojcsics, D.: Fuzzy controller for small size unmanned aerial vehicles. In: 2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMII), pp. 91–95. IEEE (2012)
43. Sugeno, M.: Industrial Applications of Fuzzy Control. Elsevier Science Inc (1985)
44. Vinh, K., Gebreyohannes, S., Karimoddini, A.: An area-decomposition based approach for cooperative tasking and coordination of uavs in a search and coverage mission. In: 2019 IEEE Aerospace Conference, pp. 1–8 (2019)
45. Wu, H., Mendel, J.M.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* **10**(5), 622–639 (2002)
46. Xia, C., Yongtai, L., Liyuan, Y., Lijie, Q.: Cooperative task assignment and track planning for multi-uav attack mobile targets. *Journal of Intelligent & Robotic Systems* **100**(3), 1383–1400 (2020)
47. Yerukayev, A.: An approach for the formation of the type-2 fuzzy sets, pp. 460–462 (2016)
48. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning ii. *Inf. Sci.* **8**(4), 301–357 (1975)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Abel Hailemichael received his Bachelor of Electrical and Electronics Engineering from the Addis Ababa University in 2009. He later received his Master of Science in Electrical Engineering from Polytechnico di Torino in 2015. He then joined the North Carolina A&T State University and received his Ph.D. degree in 2019. His research interests include “Functional Safety of ADAS/ADS Systems,” “Testing and Evaluation of Autonomous Vehicles,” “Fuzzy Logic Systems,” “Aerial and Ground Robotics”. He was a member of Autonomous Cooperative Control of Emergent Systems of Systems (ACCESS) Lab and Autonomous Control and Information Technology (ACIT) Institute.

Ali Karimoddini received the Bachelor of Electrical and Electronics Engineering from the Amirkabir University of Technology, Iran, in 2003, the Master of Science in Instrumentation and Automation Engineering from Petroleum University of Technology in 2007, and the Ph.D. degree in Electrical Engineering from the National University of Singapore, Singapore, in 2013. He is currently an Associate Professor in the ECE Department at N.C. A&T State University, NC, US. He is also the Director of the ACCESS Laboratory, Director of the NC-CAV Center of Excellence on Advanced Transportation, and the Deputy Director of the TECHLAV DoD Center of Excellence at N.C. A&T State University, NC, US. His research interests include Control and Robotics, Autonomy, Resilient Control Systems, Smart Transportation, Connected and Autonomous Vehicles, Human-machine Interactions, Cyber-physical Systems, Flight Control Systems, and Multi-agent Systems.