Micro Single Rotor Unmanned Aerial Vehicle Modeling and Simulation Under Turbulent Wind Perturbation

Ali Mehrabi[1]*, Hesam Jamali[2]

[1] Flight and Engineering Department, IAMU, Tehran, Iran
[2]Aerospace Engineering Department, Amirkabir University of Technology, Tehran, Iran

Abstract:

This study presents the modeling, simulation, and control of a novel Micro Single-Rotor Unmanned Aerial Vehicle with an onboard camera for real-time video transmission. The proposed design addresses the high manufacturing costs and mechanical complexity of conventional multirotors by employing a single brushless motor and four servo-actuated fins for attitude and position control. A comprehensive six-degree-of-freedom flight dynamics model is developed, incorporating translational and rotational equations and aerodynamic drag. A cascaded proportional-integral-derivative–proportional–integral architecture serves as the baseline controller. To enhance disturbance rejection in turbulent winds, a Fuzzy Inference System is integrated into the position loop for real-time gain scheduling. Numerical simulations are run in indoor and outdoor (turbulent) environments with step and helical trajectory references. Indoor results show the Fuzzy Inference System controller achieves 14.2% lower root-mean-square error and faster settling than the conventional cascaded controller. Under outdoor 5-knot mean winds, the fuzzy inference controller shows superior vertical tracking while the conventional one exhibits better horizontal rejection. Monte Carlo simulations with 200 random wind conditions confirm the improved robustness of the fuzzy inference cascaded controller, maintaining stable tracking up to mean wind speeds of ~11.6 knots. Overall, fuzzy-enhanced control offers an effective solution for improving Micro Single-Rotor Unmanned Aerial Vehicle position control under perturbations, suitable for cost-sensitive surveillance, inspection, and emergency response.

Keywords:

Micro UAV, Single Rotor UAV, Cascaded Control, Turbulent Wind

[1]* Mehrabi@iamu.ac.ir

1- Introduction

Micro UAVs are a subclass of unmanned aerial vehicles (UAVs) characterized by a Maximum Take-Off Weight (MTOW) of less than 5 kg, an operational range below 10 km, a flight endurance shorter than 1 hour, and a flight ceiling under 250 m, according to the Van Blyenburgh classification system. These vehicles have been widely employed in various applications, including surveillance, search-and-rescue, emergency response, traffic monitoring, infrastructure inspection, and aerial photography, among others.

Multirotor UAVs, a major family within the vertical take-off and landing (VTOL) category, are widely recognized for their ability to hover and perform rapid maneuvers. Among them, quadcopters are the most extensively studied and commercially available platforms. These systems typically employ four brushless electric motors, each driving an individual propeller. However, quadcopter design faces two notable challenges: first, brushless electric motors are costly to manufacture and require precise production infrastructure; and second, they demand significant electrical power, which increases the overall system cost due to the need for larger batteries and electronic speed controllers (ESCs).

The Micro Single-Rotor UAV (MSRUAV) represents an alternative design that utilizes only a single motor and propeller, potentially mitigating several complexities associated with multirotor platforms. SRUAVs may therefore provide a more cost-effective solution. Numerous SRUAV configurations have been introduced in the literature, each exhibiting distinct structural and control characteristics.

In recent years many types of MSRUAV are designed. Win, et al. Proposed a micro SRUAV design that is composed of one motor/propeller pair and one fin/servo pair. Their design uses a cyclic control scheme that essentially lacks yaw stability which is crucial for a lot of applications; Although it might be advantageous to unique applications like omni-directional telecommunication relay application where yaw stability is not needed. Their design was proved feasible both numerically and experimentally.[1] Moon, et al. proposed a micro SRUAV design that was intended to fulfill portability requirements and be easy-to-assemble. Their design actuators include one motor/propeller pair and eight fin/servo pairs. Their modeling of dynamics was linearized about hover state and consequently, their SRUAV platform was controlled by a cascaded PID-PID control scheme. Feasibility of their SRUAV design was demonstrated both numerically and experimentally.[2] Bernardes, et al. proposed a MSRUAV with a swashplateless motor/propeller design. In their design, anti-torque is countered by a fixed fin structure. This design is particularly useful for trading weight and power of servomotors to other components of the design, However, complexity of design and control is increased due to its control method. In swashpleless design, brushless motor makes osculatory RPM changes which actuates its (swashplateless) propeller pitch mechanism and tilts its thrust vector.[3]

Carholt, et al. proposed a micro SRUAV design that its actuators include one motor/propeller pair and four fins with each fin driven by a servo motor. They developed a cascaded PID-P-PI control scheme to drive their SRUAV platform which is nonlinearly stable and has demonstrated smooth trajectory tracking for the proposed combination of SRUAV and control scheme; however, their modeling for flight dynamics did not include environmental perturbations, namely, turbulent winds.[4]

Traditionally, UAVs' control is accomplished using algorithms based on PID controllers.[5] In recent years, other novel controller algorithms are developed for UAVs, such as nonlinear H∞ [6], super twisting [7] and reinforcement learning [8] controllers.

Portillo, et al. developed two nonlinear controllers for MSRUAV of [4], namely, Adaptive Sliding Mode (ASM) and Super Twisting (ST). they simulated these controllers under turbulent wind perturbation and compared their performance agains cascaded PID-P-PI controller. Their two nonlinear controllers had better performance than cascaded controller against perturbations, while having more actuator (servo/fins) abrupt changes.[9]

Although nonlinear controllers are disturbance resistant, however their design and implementation is complicated. Madebo developed a PID control scheme for a quadrotor UAV that obtained its gains from Fuzzy Inference System (FIS). He even used genetics algorithm to optimize parameters of fuzzy inference system. His results showed smoother trajectory tracking and faster settling times for fuzzy enhanced controller against conventional PID controller.[10]

In this research, a novel MSRUAV concept based on [4] is presented that incorporates a fins under propellers design with its center of gravity being lower than the fins. A fuzzy inference system is presented for gain tuning of position controller in cascaded control scheme to improve MSRUAV control in presence of external turbulent wind disturbance and investigate its behaviour and up/downsides against previous cascaded controller design. Finally the design robustness is evaluated against turbulent winds.

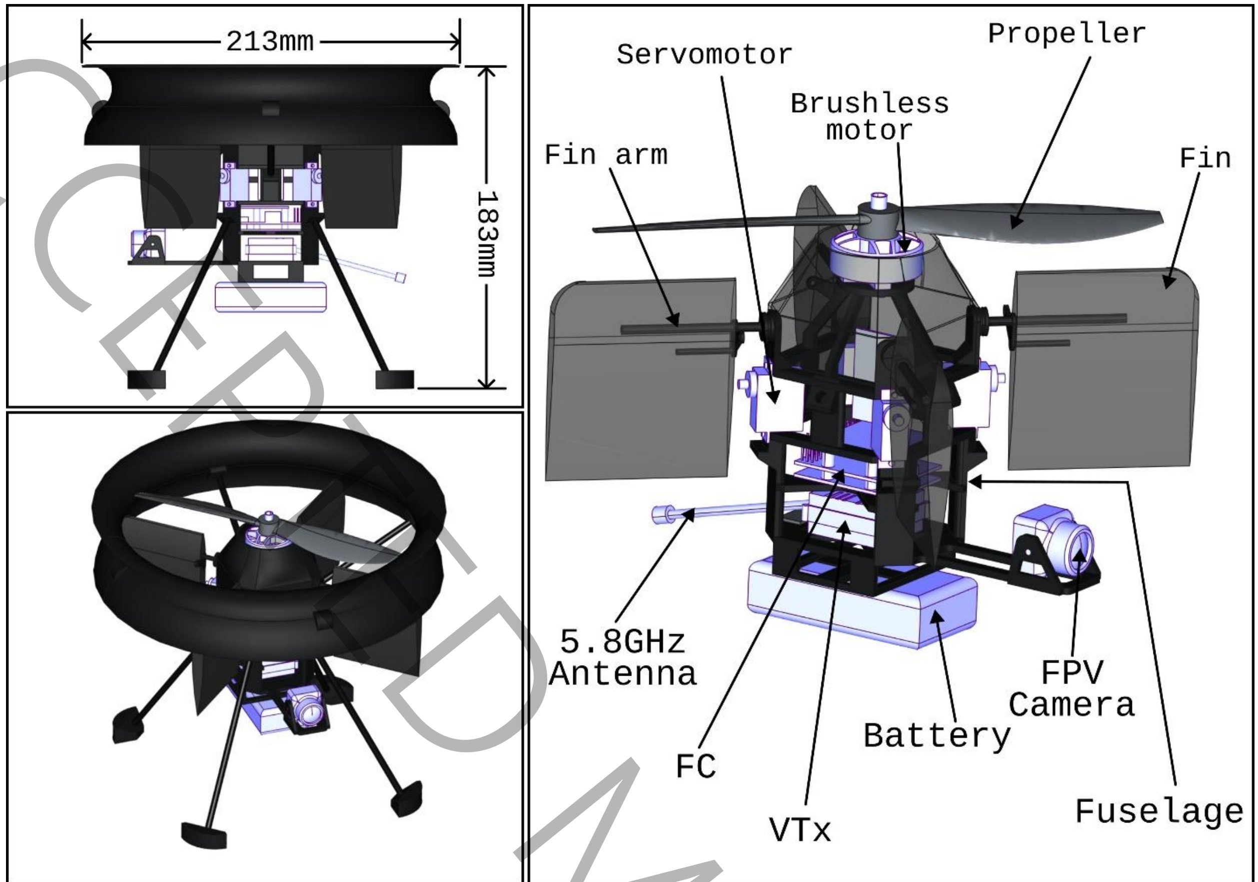

Fig. 1 Micro Single Rotor UAV concept of this research and its components

2- SRUAV Concept

The SRUAV platform proposed in this study is shown in Fig. 1, with the components labeled. The Flight Controller (FC) is responsible for measuring the UAV's position and attitude, and for processing actuator signals based on deviations from the commanded data. The FC receives commands from the drone operator, transmitted via a Radio Frequency (RF) receiver on the drone and a transmitter on the ground. The Video Transmitter (V-Tx) processes video data from the FPV camera and transmits it via a 5.8 GHz frequency antenna.

The Electronic Speed Controller (ESC) serves as an electric inverter, converting DC input to AC output for the Brushless Motor (BM). Power is supplied by LiPo batteries and distributed by the FC to the RF receiver, V-Tx, camera, ESC, and servo motors (SM). These components are mounted on an airframe, or fuselage. A duct is designed to protect the propeller and enhance its aerodynamic performance. The landing gear ensures protection for the fuselage and components during landings and may also slightly improve propeller efficiency by reducing the ground effect. The fin assembly consists of foam fins and fin arms that attach to the fuselage, providing a rotational joint connected to the servo motors (SM). Design

MTOW of SRUAV is 288 g. Design endurance at hovering flight and max throttle are 8 and 2 minutes, respectively. mass and power budget partitioning by components is presented in Table 1.

Table 1 Mass and power allocation for each component of SRUAV

Component	Mass (g)	Power (A×V)
FC	19	2.0A×5V
FPV (camera+VTx+5.8GHz antenna)	31	2.0A×5V
Propeller	8	-
BM+ESC	39	35.0A×15V (Max)
SM ×4	19	1.0A×5V (Max)
RF	3	1.0A×5V
Battery	64	-
Structure (PA12CF + carbon fiber tubes)	98	-
Fins (EPP foam) × 4	7	-

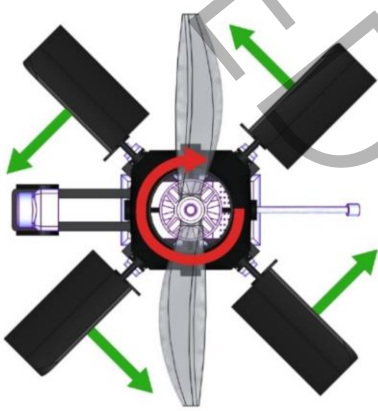

Fig. 2. Anti-torque (red arrow) and fin forces (green arrows) that act to counter

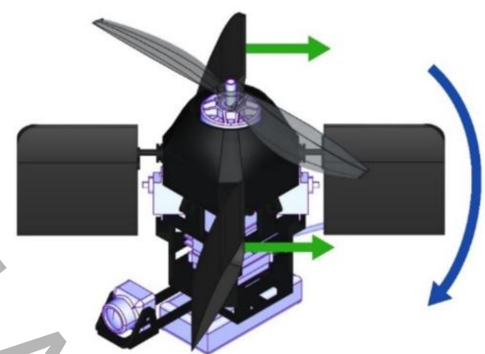

Fig. 3. Codirectional forces of two opposite fins (green) that generates a tilt moment (blue) about SRUAV's center of mass

When SRUAV is in hovering flight, as shown in **Fig. 2**, BM generates an axial torque to power the propeller that generates required thrust force to counter weight force of SRUAV. However, an equal and opposite torque is imposed on SRUAV itself, commonly known in literature as "anti-torque" which causes SRUAV to spin about the propeller axis undesirably. Propeller also blows air at a certain speed downwards passing through fins. SMs rotate fins by a certain angle that by aerodynamic interaction with blown air from propeller, can generate force. Hence, four fin forces over an arm length ℓ_h, result in the required moment to counter anti-torque.

The idea for position control of SRUAV, is similar to other VTOL drones in the sense that horizontal displacement is achieved by thrust vectoring. For deviating thrust vector from local vertical axis, horizontal moments must be generated to tilt SRUAV and deviate z_B from z_ε. To generate a horizontal moment as shown in Fig. 3, two opposing side fins rotate by the same angle to align forces of both fins in the same direction to generate a horizontal moment over an arm length ℓ_v.

3- Flight Dynamics Model

Three orthogonal axes $x_{\mathcal{E}}, y_{\mathcal{E}}, z_{\mathcal{E}}$ make up the earth-fixed frame $\mathcal{E}$ as the inertial coordinate frame. Flat-earth gravitational model with gravitational acceleration $g = 9.81\, m/s^2$ is assumed. Body-fixed frame $\mathcal{B}$ made by three orthogonal axes $x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}}$ is the frame attached to SRUAV and rotates with it. Axes x, y, z have unit vectors $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \mathbf{k}$, respectively. One can describe orientation of body-fixed frame $\mathcal{B}$ with respect to earth-fixed frame $\mathcal{E}$ by means of Euler rotation angles φ, θ, and ψ which are named roll, pitch and yaw, respectively.

Rotation Direct Cosine Matrix (DCM) ${}^{\mathcal{E}}\mathbf{R}_{\mathcal{B}}$ for a Z-X-Y rotation sequence from $\mathcal{B}$ to $\mathcal{E}$ is

$$
{}^{\mathcal{E}}\mathbf{R}_{\mathcal{B}} = \begin{bmatrix} \cos\psi\cos\theta - \sin\varphi\sin\psi\sin\theta & -\cos\varphi\sin\psi & \cos\psi\sin\theta + \cos\theta\sin\varphi\sin\psi \\ \cos\theta\sin\psi + \cos\psi\sin\varphi\sin\theta & \cos\varphi\cos\psi & \sin\psi\sin\theta - \cos\psi\cos\theta\sin\varphi \\ -\cos\varphi\sin\theta & \sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \tag{1}
$$

SRUAV angular rate w.r.t inertial frame $\mathcal{E}$ measured in $\mathcal{B}$ frame is

$$
{}^{\mathcal{E}}\boldsymbol{\omega}_{\mathcal{B}} = \omega_x \hat{\mathbf{i}}_{\mathcal{B}} + \omega_y \hat{\mathbf{j}}_{\mathcal{B}} + \omega_z \hat{\mathbf{k}}_{\mathcal{B}} \tag{2}
$$

euler angles' rates however, are obtained with transformation of ${}^{\mathcal{E}}\boldsymbol{\omega}_{\mathcal{B}}$ as[11]

$$
\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\cos\varphi sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\varphi\cos\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3}
$$

3-1- Translation and Rotation Dynamics

governing equations for translational displacement are

$$
m\begin{bmatrix} \ddot{x}_{\mathcal{E}} \\ \ddot{y}_{\mathcal{E}} \\ \ddot{z}_{\mathcal{E}} \end{bmatrix} = \begin{bmatrix} D_x^{\mathcal{E}} \\ D_y^{\mathcal{E}} \\ D_z^{\mathcal{E}} - mg \end{bmatrix} + {}^{\mathcal{E}}\mathbf{R}_{\mathcal{B}} \begin{bmatrix} \Sigma F_x^{\mathcal{B}} \\ \Sigma F_y^{\mathcal{B}} \\ \Sigma F_z^{\mathcal{B}} \end{bmatrix} \tag{4}
$$

where m is MTOW (unit: kg), $\Sigma\mathbf{F}^{\mathcal{B}} = \left[\Sigma F_x^{\mathcal{B}} \quad \Sigma F_y^{\mathcal{B}} \quad \Sigma F_z^{\mathcal{B}}\right]^T$ is the resultant force (excluding drag force) acting on SRUAV in $\mathcal{B}$ frame, and $\mathbf{D}^{\mathcal{E}} = \left[D_x^{\mathcal{E}} \quad D_y^{\mathcal{E}} \quad D_z^{\mathcal{E}}\right]^T$ is the aerodynamic drag force directly related to SRUAV's true airspeed. Assuming $\text{sgn}(.)$ as the signum function, $\mathbf{D}^{\mathcal{E}}$ is formulated as

$$\mathbf{D}^{\mathcal{E}} = K_D \begin{bmatrix} (V_x^a)^2 \,\text{sgn}(V_x^a) \\ (V_y^a)^2 \,\text{sgn}(V_y^a) \\ (V_z^a)^2 \,\text{sgn}(V_z^a) \end{bmatrix} \tag{5}$$

Where K_D is the drag coefficient of this SRUAV and $\mathbf{V}^a = \left[V_x^a \quad V_y^a \quad V_z^a\right]^T$ is SRUAV's true airspeed. Drag coefficient is directly dependent on SRUAV geometry, dimensions and air density. True airspeed is the difference of instantaneous wind speed and SRUAV's velocity in $\mathcal{E}$ frame. True airspeed is obtained as

$$\mathbf{V}^a = \begin{bmatrix} V_x^a \\ V_y^a \\ V_z^a \end{bmatrix} = \begin{bmatrix} -\dot{x}_{\mathcal{E}} + V_x^w \\ -\dot{y}_{\mathcal{E}} + V_y^w \\ -\dot{z}_{\mathcal{E}} + V_z^w \end{bmatrix} \tag{6}$$

Where $\mathbf{V}^w = [V_x^w \quad V_y^w \quad V_z^w]^T$ is the instantaneous wind speed. if $\mathbf{V}^w = [0 \quad 0 \quad 0]^T$ for all $t > 0$ then drag force exerted onto SRUAV entirely arises from its velocity, i.e. the environment is representative of indoors flight.

Governing equations for attitude dynamics are

$$[MOI]_{\mathcal{B}} \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \Sigma M_x^{\mathcal{B}} \\ \Sigma M_y^{\mathcal{B}} \\ \Sigma M_z^{\mathcal{B}} \end{bmatrix} - \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times [MOI]_{\mathcal{B}} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{7}$$

where $\Sigma\mathbf{M}^{\mathcal{B}} = \left[\Sigma M_x^{\mathcal{B}} \quad \Sigma M_y^{\mathcal{B}} \quad \Sigma M_z^{\mathcal{B}}\right]^T$ is the resultant moment acting on SRUAV in $\mathcal{B}$ frame and [MOI] is UAV's moment of inertia matrix which is

$$[MOI]_{\mathcal{B}} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \tag{8}$$

MOI mainly depends on geometry of SRUAV and is diagonal for a fairly symmetric design, i.e. all elements other than J_{xx}, J_{yy}, J_{zz} are assumed to be zero.[12]

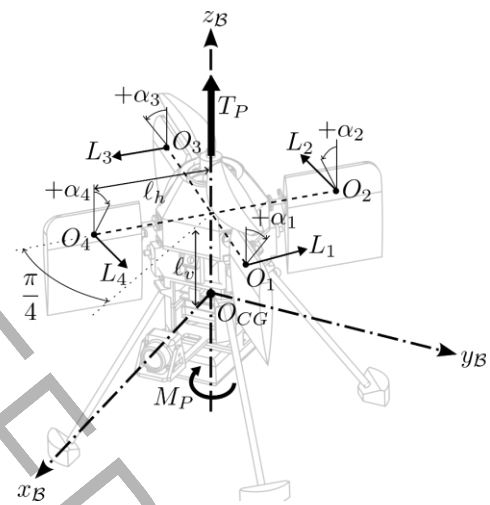

Fig. 4. Free-body diagram of MSRUAV

3-1- SRUAV Model

Free-body diagram of proposed design is shown in Fig. 4. O_1, O_2, O_3 and O_4 are aerodynamic centers of fins 1, 2, 3, and 4, respectively. Origin of body-fixed frame $\mathcal{B}$ is coincident with SRUAV's center of mass O_{CG}. forward-flight axis $x_{\mathcal{B}}$ is located at an counter-clockwise angle of $\pi / 4$ from fin 4 axis of rotation. L_1, L_2, L_3 and L_4 are fin aerodynamic forces exerted onto O_1, O_2, O_3 and O_4, respectively. Each fin makes an incidence angle with blown air from propeller denoted as α. Notice that positive direction for $\alpha_i, \quad i = 1, 2, 3, 4$ is assumed as a clockwise rotation about fin's axis looking from fin's outer side. T_P and M_P are generated thrust and anti-torque of propeller, respectively. ℓ_h and ℓ_v are vertical and horizontal moment arms of fins, respectively.

Resultant force components in body-fixed frame B are

$$\begin{aligned} \Sigma F_x^{\mathcal{B}} &= \cos(\pi / 4)(L_4 + L_3 - L_1 - L_2) \\ \Sigma F_y^{\mathcal{B}} &= \cos(\pi / 4)(L_4 + L_1 - L_3 - L_2) \\ \Sigma F_z^{\mathcal{B}} &= T_P \end{aligned} \tag{9}$$

Resultant moments over each axis of body-fixed frame B are

$$\Sigma M_x^{\mathcal{B}} = \ell_v \cos(\pi/4)(L_2 + L_3 - L_1 - L_4)$$
$$\Sigma M_y^{\mathcal{B}} = \ell_v \cos(\pi/4)(L_4 + L_3 - L_1 - L_2) \tag{10}$$
$$\Sigma M_z^{\mathcal{B}} = \ell_h (L_1 + L_2 + L_3 + L_4) - M_P$$

each fin force L, thrust T_P and anti-torque M_P are modelled as

$$L_j = K_F (2\pi\alpha_j)(N/60)^2, \quad j = 1,2,3,4$$
$$F_P = K_T (N/60)^2 \tag{11}$$
$$M_P = K_M (N/60)^2$$

where N is the propeller rate (RPM).[4, 9] K_F, K_T and K_M are constants resulting from SRUAV's design that can be estimated whether theoretically or experimentally. Sizing of K_F, K_T and K_M is done in a previous dedicated research and are presented as design constants in this research. $0 < N < N_{\max}$ and $-\alpha_{\max} < \alpha_j < \alpha_{\max}, \quad j = 1,2,3,4$ form control inputs of the dynamical system. Notice that both N and α have upper and lower limits related to equipment safety or prevention of performance loss.

3-2- Turbulent Wind Model

The most accurate method of modeling wind turbulence could be Computational Fluid Dynamics (CFD) analysis of a predetermined scenario as in[13]. However, this method is computationally-intensive. Dryden and Von Karman wind models, are two most used models for turbulent winds over land which both models are approved by specifications such as MIL-F-8785C. Turbulent wind models based on Von Karman model yield higher accuracy for simulation but are formulated in frequency domain and have complicated algorithmic implementation. [14] An approximation of Dryden's method developed in [15] is used for this research, because this model can be easily programmed in time domain.

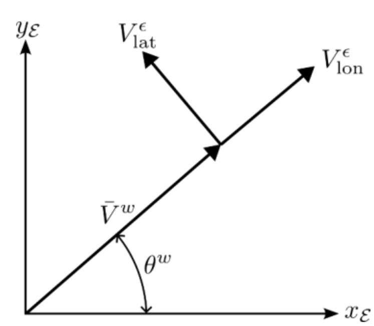

Fig. 5. Mean wind speed and planar components of stochastic turbulence

$\bar{V}^w$ is the mean wind speed (non-turbulent constant wind speed), θ^w is the mean wind direction deviation measured counter clockwise from $x_{\mathcal{E}}$ and $\mathbf{V}^\epsilon$ is the stochastic turbulence of wind speed. Mean wind speed is a two dimensional speed field, as shown in Fig. 5. However, $\mathbf{V}^\epsilon$ is a three dimensional velocity vector which its components are longitudinal turbulence V^ϵ_{lon}, latitudinal turbulence V^ϵ_{lat}, and vertical turbulence V^ϵ_{vert}. According to the presented definitions, $\mathbf{V}^w$ in frame $\mathcal{E}$ can be reformulated as[16]

$$\mathbf{V}^w = \begin{bmatrix} \bar{V}^w \cos\theta^w \\ \bar{V}^w \sin\theta^w \\ 0 \end{bmatrix} + \begin{bmatrix} V^\epsilon_{lon} \cos\theta^w + V^\epsilon_{lat} \sin\theta^w \\ V^\epsilon_{lon} \sin\theta^w + V^\epsilon_{lat} \cos\theta^w \\ V^\epsilon_{vert} \end{bmatrix} \tag{12}$$

Stochastic turbulence is applied with a Δt_w period, usually within 1000-125 milliseconds (1-8Hz) range because natural wind turbulence in open fields has a value in that range.[16]

Turbulence speed $\mathbf{V}^\epsilon$ at every timestep index $i+1$ is

$$\begin{aligned} (V^\epsilon_{lon})^{i+1} &= (1-\Delta t_w / \tau_{lon})(V^\epsilon_{lon})^i + \sigma_{lon}\sqrt{2\Delta t_w / \tau_{lon}}\, r^{i+1}_{lon} \\ (V^\epsilon_{lat})^{i+1} &= (1-\Delta t_w / \tau_{lat})(V^\epsilon_{lat})^i + \sigma_{lat}\sqrt{2\Delta t_w / \tau_{lat}}\, r^{i+1}_{lat} \\ (V^\epsilon_{ver})^{i+1} &= (1-\Delta t_w / \tau_{ver})(V^\epsilon_{ver})^i + \sigma_{ver}\sqrt{2\Delta t_w / \tau_{ver}}\, r^{i+1}_{ver} \end{aligned} \tag{13}$$

where $-1 < r < 1$ is the white noise and is generated using a uniformly distributed random number generator algorithm. Two parameters τ and σ corresponding to each component of $(V^\epsilon)^{i+1}$ which depend on a given height above ground h (unit:ft) are presented in Table 2. W_{20} .is wind speed at $h = 6\,ft$ and is used as turbulence criteria. If $W_{20} = 15kt$ turbulence is considered light, 30kt is moderate, and 45kt is severe turbulence.[14] For this research although mean wind speed is used to generate turbulence.

Table 2. Stochastic turbulence parameters

	τ	σ
lon	$\tau_{lon} = L_u / V^a, \quad L_u = \dfrac{h}{(0.177+0.000823h)^{1.2}}$	$\dfrac{0.1W_{20}}{(0.177+0.000823h)^{0.4}}$
lat	$\tau_{lat} = L_v / V^a, \quad L_v = \dfrac{h}{(0.177+0.000823h)^{1.2}}$	$\dfrac{0.1W_{20}}{(0.177+0.000823h)^{0.4}}$
vert	$\tau_{ver} = L_w / V^a, \quad L_w = h$	$0.1W_{20}$

4- Control Scheme

Cascaded PID-P-PI control algorithm of [4] is chosen to control the SRUAV of this research. Where its first loop $(PID)_i$ is the position controller and the next two loops (P-PI) collectively are the attitude controller.

4-1- Cascaded Controller

Cascaded controller block diagram is shown in Fig. 6. x_c, y_c, z_c and ψ_c are four control references (operator's commands) to be tracked. Controller inputs to be measured in real-time are six system states $x, y, z, \varphi, \theta, \psi$ and three angular rates $\omega_x^{\mathcal{B}}, \omega_y^{\mathcal{B}}, \omega_z^{\mathcal{B}}$.

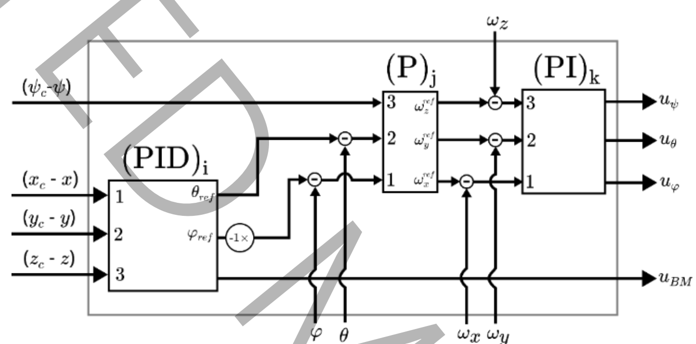

Fig. 6. Cascaded control scheme block diagram

First loop of controller outputs position control signals

$$\begin{aligned}
\theta_{ref} &= P_{i,1}(x_c - x) + I_{i,1}\int_0^t (x_c - x)dt + D_{i,1}\frac{d(x_c - x)}{dt} \\
\varphi_{ref} &= P_{i,2}(y_c - y) + I_{i,2}\int_0^t (y_c - y)dt + D_{i,2}\frac{d(y_c - y)}{dt} \\
u_{BM} &= P_{i,3}(z_c - z) + I_{i,3}\int_0^t (z_c - z)dt + D_{i,3}\frac{d(z_c - z)}{dt}
\end{aligned} \tag{14}$$

These three outputs of PID must be saturated that weill be discussed later in this section. $0 \le u_{BM} \le 1$ is the RPM signal and $u_{BM} \times N_{max} = N$. Two other signals φ_{ref} and θ_{ref} are inputs to the second loop of control which result in

$$\begin{aligned}
\omega_x^{ref} &= P_{j,1}(-\varphi_{ref}^{sat} - \varphi) \\
\omega_y^{ref} &= P_{j,2}(\theta_{ref}^{sat} - \theta) \\
\omega_z^{ref} &= P_{j,3}(\psi_c - \psi)
\end{aligned} \tag{15}$$

last control loop outputs desired control moments for attitude control[4]

$$
\begin{aligned}
u_{\varphi} &= P_{k,1}(\omega_x^{ref} - \omega_x) + I_{k,1}\int_0^t (\omega_x^{ref} - \omega_x)dt \\
u_{\theta} &= P_{k,2}(\omega_y^{ref} - \omega_y) + I_{k,2}\int_0^t (\omega_y^{ref} - \omega_y)dt \\
u_{\psi} &= P_{k,3}(\omega_z^{ref} - \omega_z) + I_{k,3}\int_0^t (\omega_z^{ref} - \omega_z)dt
\end{aligned} \tag{16}
$$

However, signals needed for attitude control in the dynamical system are incidence angles $\alpha_1, \alpha_2, \alpha_3$ and α_4 which are obtained as

$$
\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & +1 & +1 \\ -1 & +1 & +1 \end{bmatrix} \begin{bmatrix} u_{\varphi} \\ u_{\theta} \\ u_{\psi} \end{bmatrix} \tag{17}
$$

One might notice integral and differential terms in Eq.s 14 to 16 are continuous in nature, but due to computational limitations onboard SRUAV, these terms must be discretely computed at a time period of Δt_c. Assuming $[.]$ as integral part function, discretized integral and differential terms are

$$
\begin{aligned}
&\int_0^t e\, dt \approx \sum_0^n e_n \Delta t_c, \quad n = [t / \Delta t_c] \\
&\frac{de}{dt} \approx \frac{e_n - e_{n-1}}{\Delta t_c}, \quad n = [t / \Delta t_c] + 1
\end{aligned} \tag{18}
$$

Integral controllers in PID schemes have a well known undesirable characteristic known as *windup* effect. For countering windup, backtracking method is used as an anti-reset antiwindup mechanism.[17]

Assuming *SAT(input,lowerlim,upperlim)* as saturation or clamp function, outputs of position controller (PID) are

$$
\begin{aligned}
\theta_{ref}^{sat} &= SAT(\theta_{ref}, \theta_{ref}^{low}, \theta_{ref}^{up}) \\
\varphi_{ref}^{sat} &= SAT(\varphi_{ref}, \varphi_{ref}^{low}, \varphi_{ref}^{up}) \\
u_{BM}^{sat} &= SAT(u_{BM}, u_{BM}^{low}, u_{BM}^{up})
\end{aligned} \tag{19}
$$

These saturated signals are inputs to (15). assuming e as the error signal corresponding to each component of displacement vector, namely, x, y and z; the discretized memory of integral controller is readjusted

$$\begin{aligned}
&\sum_{1}^{n} e^{n}\Delta t = (\sum_{1}^{n-1} e^{n-1}\Delta t) + e^{n}\Delta t, && if\ u = u^{sat} \\
&\sum_{1}^{n} e^{n}\Delta t = (\sum_{1}^{n-1} e^{n-1}\Delta t) + (u^{sat} - u)K_{AW}, && else
\end{aligned} \tag{20}$$

Where u is one of control signals $\varphi_{ref}, \theta_{ref}$ or u_{BM}. The anti-windup parameter K_{AW} must be tuned experimentally.

4-1- Fuzzy Inference System

Fuzzy logic is a form of many-valued logic in which the truth value of variables may be any real number between 0 and 1, while, in computer logic (Boolean logic) values are either 0 or 1. Fuzzy logic forms the foundation for Fuzzy Inference System (FIS) which in control applications, can compute control signal from input error signal(s) using linguistic rules.[18]

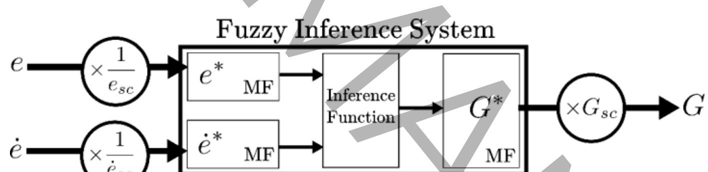

Fig. 7. Fuzzy inference system for gain adjustment of cascaded controller

FIS in this research is used to calculate gains of position controller (PID) loop in the cascaded controller, namely P_i, I_i, D_i and its flow block diagram is shown in Fig. 7. Inputs to FIS are unscaled error e^* and its unscaled time derivative $\dot{e}^*$. Output of FIS is an unscaled gain G^*. Scaled gains are obtained as

$$\begin{aligned}
&e^* = \frac{e}{e_{sc}},\ \dot{e}^* = \frac{\dot{e}}{\dot{e}_{sc}} \\
&P_i = FIS_P(e^*, \dot{e}^*) \times P_{sc} \\
&I_i = FIS_I(e^*, \dot{e}^*) \times I_{sc} \\
&D_i = FIS_D(e^*, \dot{e}^*) \times D_{sc}
\end{aligned} \tag{21}$$

Where $e_{sc}, \dot{e}_{sc}, P_{sc}, I_{sc}, D_{sc}$ are 5 parameters of FIS which need to be tuned. FIS itself is composed of input and output Membership Functions (MF). An inference function, computes membership score of each input (fuzzification), then decides on the output fuzzy set determined by a linguistic rule set and finally computes corresponding output value using an inference function (defuzzification).

FIS in this research is a Mamdani type-1 system. Triangular MF and Gaussian MF are used for input and output membership functions, respectively. Centeroid inference method is used as inference function.

Rulesets must be defined linguistically for output gains. For inputs three fuzzy sets are defined as Negative (NE), Zero (ZE) and Positive (PO). For outputs three fuzzy sets are defined as Low (L), Medium (M) and High (H). Now with memberships defined, ruleset for P^* , I^* are defined in Table 3 and ruleset for D^* is defined in Table 4.

Table 3. inference ruleset for P* and I*

$\dot{e}$ \ e	NE	ZE	PO
NE	M	L	M
ZE	H	M	H
PO	M	L	M

Table 4. Inference ruleset for D*

$\dot{e}$ \ e	NE	ZE	PO
NE	M	H	M
ZE	L	M	L
PO	M	H	M

4-1- Performance Evaluation Criterea

Deviations of control algorithm with respect to references must be quantified using a metric. Root Mean Squared Error (RMSE) is one such a metric. As stated above, control signals are computed in discrete pulses. Assuming Δt_c as discretization time period, index of any arbitrary pulse is $n=[t/\Delta t_c]$ and index of the last pulse is $n_f=[t_f/\Delta t_c]$. Tracking error at any given pulse n is

$$e(n)=\sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2} \tag{22}$$

Now, by having all discrete errors computed over $n=1,2,3,...,n_f$, total RMSE is obtained as

$$\text{RMSE}=\sqrt{\frac{\sum_{n=1}^{n_f}\{e(n)\}^2}{n_f}} \tag{23}$$

RMSE for each component of trajectory can also be obtained by reformulating (22) for one component

$$e=\sqrt{(x-x_c)^2} \quad or \quad e=\sqrt{(y-y_c)^2} \quad or \quad e=\sqrt{(z-z_c)^2} \tag{24}$$

5- Parameter Estimation Methodology

Parameter estimation block diagram is shown in Fig. 8. Dynamical system is consisted of Eq.s (4) and (7). Derivatives of state vector $\mathbf{s}(t) = [x\ y\ z\ \varphi\ \theta\ \psi\ \dot{x}\ \dot{y}\ \dot{z}\ \omega_x\ \omega_y\ \omega_z]^T$ are computed in dynamical system, then integrated by Initial Value Problem (IVP) solver algorithm with $\mathbf{s}_0 = \mathbf{s}(t_0)$. Control pulses frequency and sensors data sampling frequency might have very different values in a real flight scenario and must be synchronized using data fusion process.[19] However, these frequencies are assumed synchronized and equal for the purposes of this research. Simulations are developed with Julia programming language[20] and Differential Equations.jl is used as a parameter estimation framework.[21] Tsit5() is chosen as the IVP solver algorithm. Gain tunings for controller algorithm may be done using LQR optimization problem which is a methodology requiring optimal control techniques.[11] Gain tunings in this research were done using trial-and-error simulations as programs developed in Julia language are well known to be fast in runtime. FuzzyLogic.jl is utilized to create FIS structures for MSRUAV controller in (21).[22]

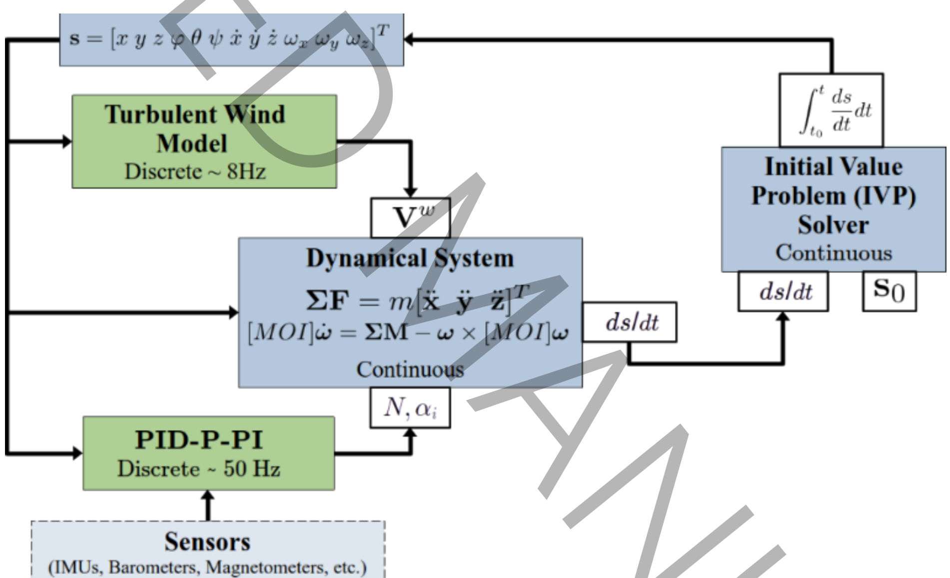

Fig. 8. Simulation block diagram with continuous and discrete terms marked in blue and green, respectively.

6- Numerical Simulations

Parameters commonly used in all simulations are presented in Table 5. Control discretization period is Δt_c = 0.02s = 50Hz.

Table 5. Common parameters for all simulations

$\ell_v = 0.036m$	$\ell_h = 0.073m$	$m = 0.288kg$
$K_F = 2.4114 \times 10^{-5}$	$K_T = 1.6412 \times 10^{-4}$	$K_F = 2.1564 \times 10^{-6}$
$K_D = 0.084987$	$\alpha_{\max} = 0.1745\ rad$	$N_{\max} = 17208\ RPM$

$\varphi_{ref}^{up} = 1.4\ rad = -\varphi_{ref}^{low}$	$\theta_{ref}^{up} = 1.4\ rad = -\theta_{ref}^{low}$	$u_{BM}^{up} = 1, \quad u_{BM}^{low} = 0$

Two reference trajectories are utilized in simulations, namely step response and helix. Step response reference is used for gain tunings and is defined as

$$
\begin{aligned}
&x_c(t) = \begin{cases} 0 & t < 20s \\ 2m & t \geq 20s \end{cases} \\
&y_c(t) = \begin{cases} 0 & t < 10s \\ 2m & t \geq 10s \end{cases} \\
&z_c(t) = 2m \\
&\psi_c(t) = \begin{cases} 0 & t < 40s \\ 20^{\circ} & t \geq 40s \end{cases}
\end{aligned}
\tag{25}
$$

helix reference is used for RMSE evaluation as it is continuous through simulation and is set to start from SRUAV's initial position. Helix reference is defined as

$$
\begin{aligned}
&x_c(t) = 15\cos\left(\frac{2\pi}{30}t\right) \\
&y_c(t) = 15\sin\left(\frac{2\pi}{30}t\right) \\
&z_c(t) = t \\
&\psi_c(t) = 0
\end{aligned}
\tag{26}
$$

Matrix Of Inertia (MOI) for proposed SRUAV concept is directly obtained from CAD software as

$$
[MOI] = \begin{bmatrix} 1.14719\times10^{-3} & 1.774\times10^{-5} & 4.66\times10^{-6} \\ 1.774\times10^{-5} & 1.14719\times10^{-3} & -4.45\times10^{-6} \\ 4.66\times10^{-6} & -4.45\times10^{-6} & 7.7517\times10^{-4} \end{bmatrix} (kg.m^2)
\tag{27}
$$

6-1- Parameter Tuning

Cascaded controller gains tuning in this research are done by trial-and-error simulation experiments. Position controller (PID) of cascaded scheme has the most impact in overall trajectory tracking and control. Integral term in PID is very effective for reducing mean wind speed disturbance and with under or overvalued anti-windup parameter K_{AW}, it might impose steady-state error on control. Derivative term of PID is effective against fastly growing errors like discontinuous reference value changes or sudden disturbances. Attitude controller (P-PI) has little impact on trajectory control and P_j controller mainly affects attitude osculation reduction. $(PI)_k$ are set once as in Table 6 and be used in all simulations.

Table 6. Attitude controller gains used in this research

	1	2	3
P_j	3.30	3.30	4.00
P_k	0.06	0.06	0.06
I_k	0.08	0.08	0.08

Creating a FIS and tuning its parameters needs intuitive understanding of PID controller which is described before. Fig. 9 and Fig. 10 shows FIS structures for unscaled gains P^* and I^*, respectively. Error and its derivative have symmetric MFs for both P^* and I^* but output MF of I^* has larger distribution for L (Low) values and smaller distribution for H (High) values. By comparing input-output inference 3D surface plots, it is evident that peak values of I^* happen at the very edges of error input range to avoid downsides of integral controller unless wind push is sufficiently strong for integral term to act in with highest gains. Also lowest unscaled value for I^* is set to 0.1 because for the same inputs, $P^* = 0$. FIS structure of unscaled gain D^* is shown in Fig. 11. similar to I^*, highest values of D^* are needed for the highest error derivatives to counter turbulence or discontinuous reference changes.

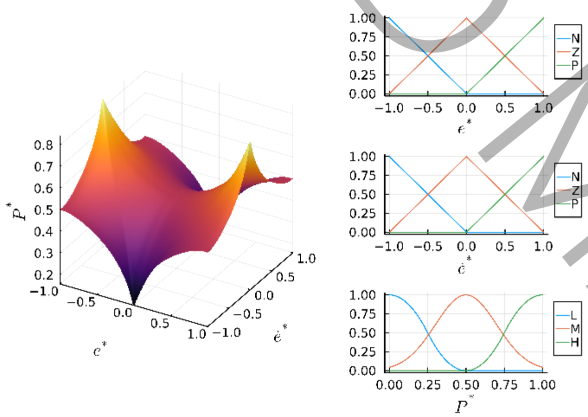

Fig. 9. FIS structure for P* unscaled gain

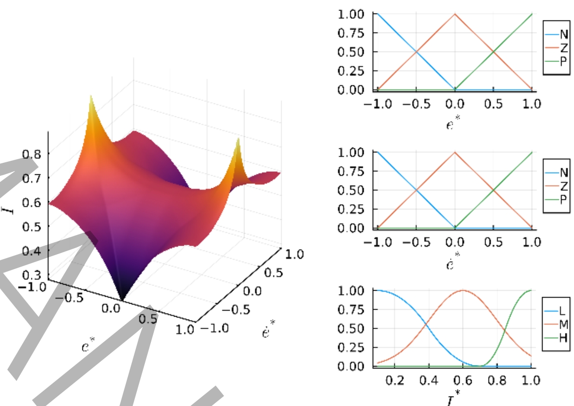

Fig. 10. FIS structure for I* unscaled gain

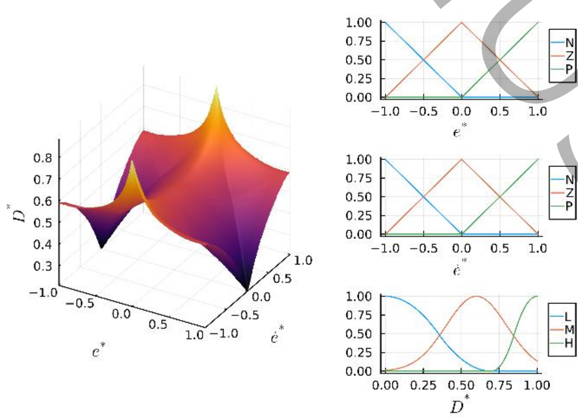

Fig. 11. FIS structure for D* unscaled gain

6-2- Indoors Flight

Indoors flight is the first scenario to be simulated. Cascaded controller gains and FIS parameters are presented in Table 7 and Table 8, respectively. Step response simulation integral curves are presented in Fig. 12. FIS-cascaded controller can evidently settle on reference sooner than cascaded controller. However, pitch and roll angle oscilations in FIS-cascaded trajectory have more amplitude than cascaded controller trajectory. Controller outputs (actuators operating profile) for cascaded and FIS-cascaded simulations are presented in Fig. 14 and Fig. 15, respectively. One example of overvalued anti-windup parameter K_{AW}^{3} is occured in FIS-cascaded trajectory of Fig. 15 which causes a reset in integral memory. Initially, RPM signal shoots up to 100% and immediately drops down to 10% and then again RPM signal grows to reach the level of hovering flight. This in consequence leads to a slightly longer settling time of z component of FIS-cascaded trajectory in Fig. 12.

Helix path simulation is presented in Fig. 13. One observation in FIS-cascaded control system, was that although system has lower settling times compared to cascaded controller, However, its steady-state errors in continuous trajectories are hardly mitigated by tuning its parameters and might need structural modifications of membership functions or inference methods. RMSEs for each reference and control scheme along each axis of translation are presented in Table 9 and in both cases of step and helix reference simulations, FIS-cascaded controller is performing better.

Table 7. Cascaded gains for indoors simulation case

	1	**2**	**3**
P_i	0.1	0.1	1.8
I_i	0.0001	0.0001	2.0
D_i	0.25	0.25	0.5
K_{AW}	0.02	0.02	0.002

Table 8. FIS parameters for indoors simulation case

	1	**2**	**3**
P_{sc}	0.34	0.34	3.45
I_{sc}	0.003	0.003	1.5
D_{sc}	0.40	0.40	0.70
K_{AW}	0.1	0.1	0.01
e_{sc}	1 m		
$\dot{e}_{sc}$	20 m/s		

Table 9. Components RMSE recorded in indoors simulation case

RMSE	**x**	**y**	**z**
Step, Cascaded	0.4441	0.4471	0.2708
Step, FIS-Cascaded	0.3810	0.3865	0.2333
Helix, Cascaded	1.2083	0.6796	0.0325
Helix, FIS-Cascaded	1.0420	0.5536	0.0479

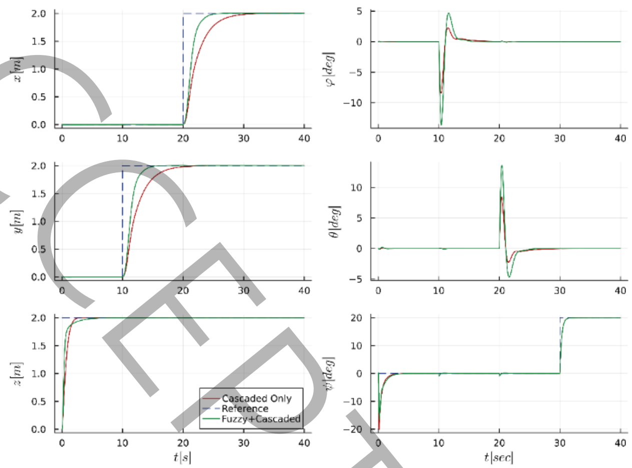

Fig. 12. Step response simulation for indoors flight

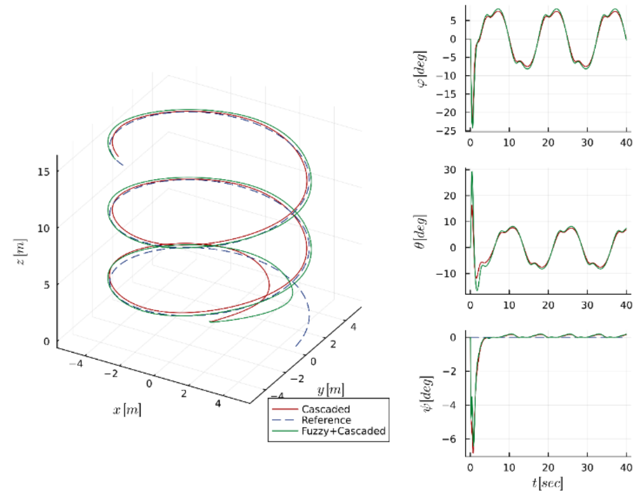

Fig. 13. Helix simulation for indoors flight

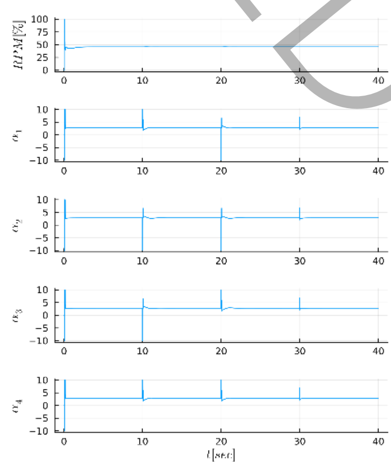

Fig. 14. Actuators inputs for indoors step cascaded case

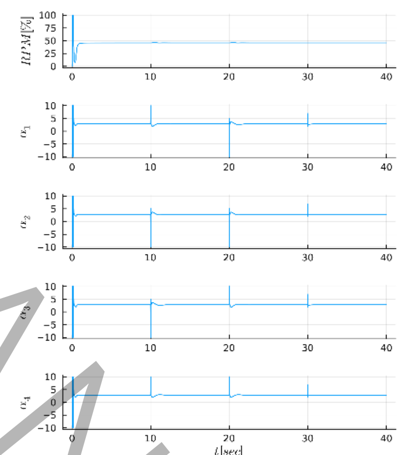

Fig. 15. Actuators inputs for indoors step FIS-cascaded case

6-3- Outdoors Flight

Outdoors flight is the next scenario of simulations, for which a mean wind speed of 5 knots at $\theta^{w} = 30^{\circ}$ is present. Wind speed experienced in step and helix simulations are presented in Fig. 16 and Fig. 17, respectively. Cascaded controller gains and FIS parameters are presented in Table 10 and Table 11, respectively. Step response simulation integral curves are shown in Fig. 18. FIS-Cascaded controller still has lower settling time than cascaded only controller, However, Cascaded controller can counter stochastic turbulence better than FIS-cascaded. Both controllers suffer from initial overshoot due to disturbance pushback, but for FIS-cascaded controller, this overshoot is difficultly (if at all) affected by FIS parameters. Helix path simulation in presence of turbulent wind is presented in **Fig. 19** which shows FIS-cascaded controller has performed better in settling times, However, component RMSE of trajectories in Table 12. show tracking errors of cascaded controller was lower. Actuator input profiles of helix simulation using cascaded controller and FIS-cascaded controller which are shown in Fig. 20 and Fig. 21, respectively, are fairly similar although actuation frequency of cascaded controller is slightly more than the FIS-cascaded controller.

Table 10. Cascaded gains for outdoors simulation case

	1	2	3
P_i	0.7	0.7	1.5
I_i	0.18	0.18	0.5
D_i	0.5	0.5	0.5
K_{AW}	0.1	0.1	0.01

Table 11. FIS parameters for outdoors simulation case

	1	2	3
P_{sc}	0.34	0.34	3.45
I_{sc}	0.003	0.003	1.5
D_{sc}	0.40	0.40	0.70
K_{AW}	0.08	0.08	0.008
e_{sc}	1 m		
$\dot{e}_{sc}$	1 m/s		

Table 12. Components RMSE recorded in outdoors simulation case

RMSE	x	y	z
Step, Cascaded	0.3670	0.3642	0.4178
Step, FIS-Cascaded	0.3931	0.3817	0.2093
Helix, Cascaded	0.9381	0.1933	0.0783
Helix, FIS-Cascaded	1.3035	0.1815	0.0447

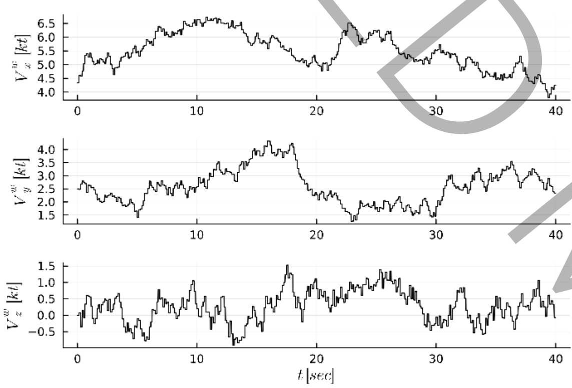

Fig. 16. Wind speed components in step response simulation

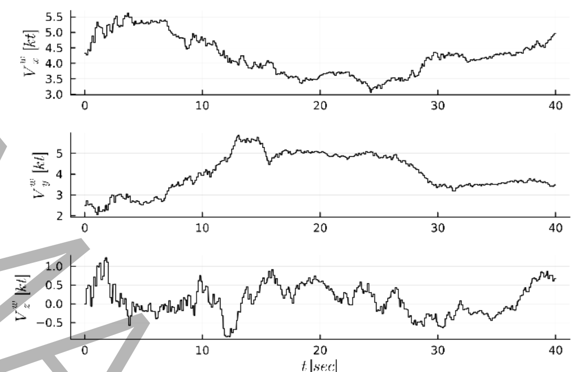

Fig. 17. Wind speed components in helix simulation

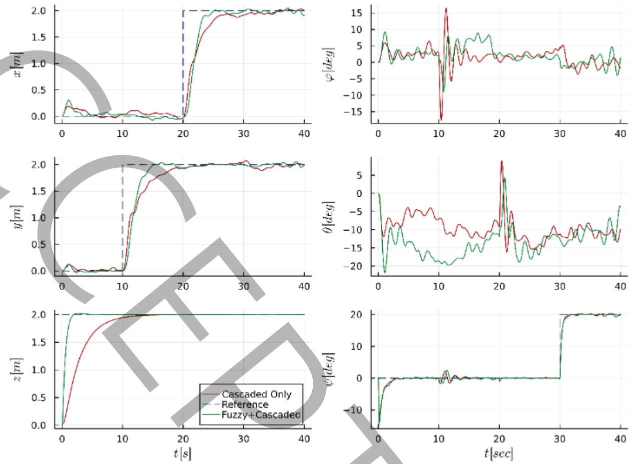

Fig. 18. Step response simulation for outdoors flight

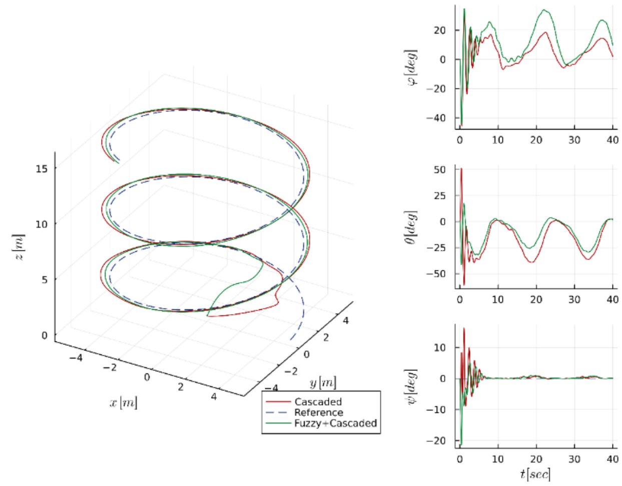

Fig. 19. Helix simulation for outdoors flight

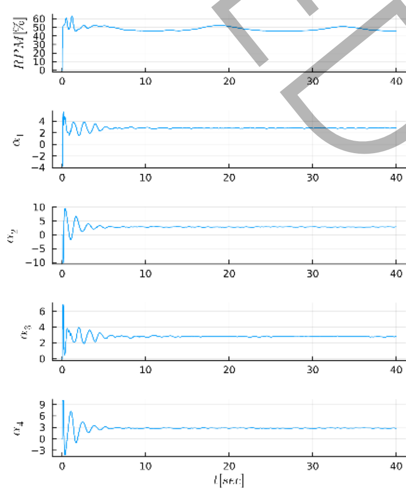

Fig. 20. Actuators inputs for outdoors helix cascaded case

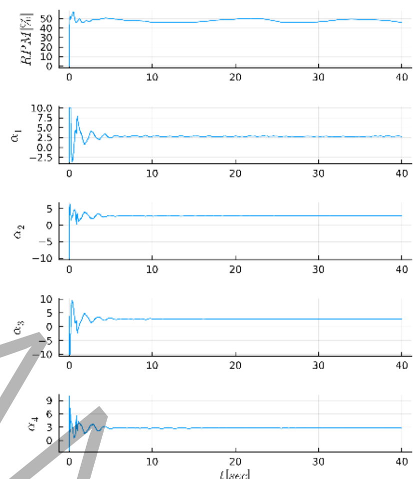

Fig. 21. Actuators inputs for outdoors helix FIS-cascaded case

6-4- Design Robustness Evaluation

In previous simulations, FIS-cascaded controller was observed to operate smoothly in major wind speed changes, however, manually tuned cascaded controller needs readjustment for even minor mean wind speed changes. For demonstrating these behaviors, a Monte Carlo (MC) simulation with 200 randomly sampled $0 \le \bar{V}^w \le 8kt$ and $0 \le \theta^w \le 90^o$ is executed using same controller parameter set for every sample. MC plots for step response trajectories of cascaded only and FIS-cascaded controllers are presented in Fig. 22 and Fig. 23, respectively.

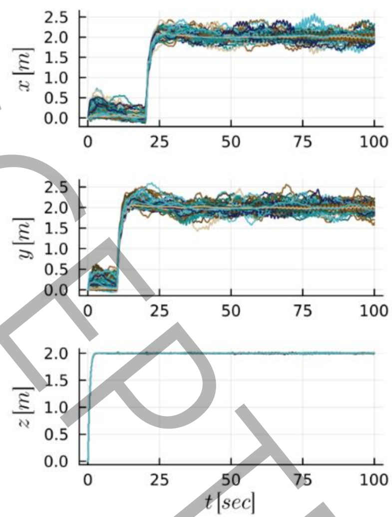

Fig. 22. MC plots of cascaded only controller with step reference – 200 samples

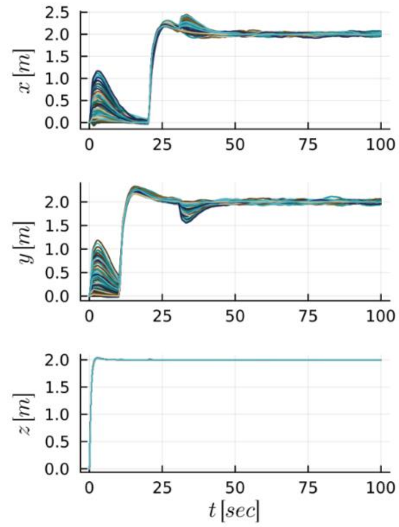

Fig. 23. MC plots of FIS-cascaded controller with step reference – 200 samples

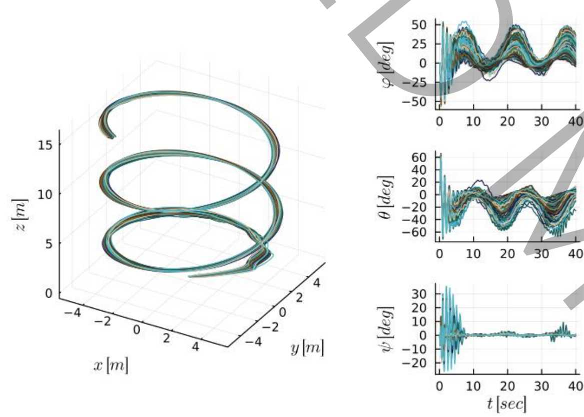

Fig. 24. Helix MC simulation for cascaded controller

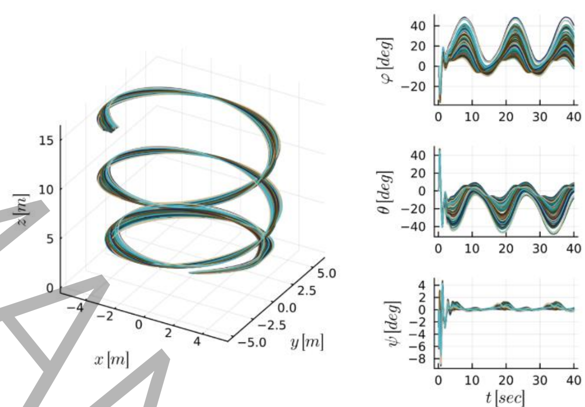

Fig. 25. Helix MC simulation for FIS-cascaded controller

The FIS-cascaded controller robustness against disturbance and parameter variations is evident but cascaded controller cannot mitigate undesireable osculations in most of the trajectories. Results of helix MC simulations for cascaded and FIS-cascaded controllers are presented in Fig. 24 and Fig. 25, respectively. The same smoothness of FIS-cascaded controlled trajectories is seen, However, steady-state errors of cascaded controller are more converged toward reference.

Finally, a test is designed to evaluate robustness of FIS control scheme against turbulent wind speed. $V^w \in [0.0, 20.0]kt$ is sampled 30 times in equal distances and in each sample point, a MC simulation with 500 samples in a neighborhood of $V^w \pm 2kt$ and $\theta^w = 0$ is executed and average of RMSEs is obtained as a criteria for controller robustness which is shown in Fig. 26. Up until 11.6 kt of mean wind speed, average RMSE is fairly constant, However, after 15kt average RMSE grows sharply because many trajectories separate from reference or cannot settle on reference at all.

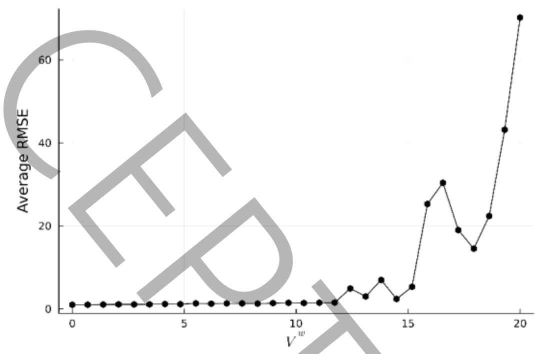

Fig. 26. Controller robustness against turbulent wind evaluation

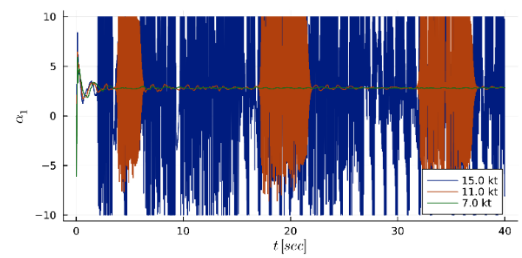

Fig. 27. Fins and servo motors deflection angles timeline in 15, 11, and 7 kt mean wind speeds

Discussion

While the FIS-enhanced cascaded controller demonstrates commendable capability in tracking reference trajectories even under elevated wind speeds, a critical operational concern emerges when analyzing the actuator demand profiles across different turbulence intensities. As illustrated in the fluctuation data of Fig. 27, the fins and their corresponding servo motors experience increasingly pronounced travel demands as mean wind speed escalates from 7.0 knots to 15.0 knots. At moderate wind conditions of 7.0 knots, the servo actuators exhibit relatively restrained and periodic deflections, suggesting that the control system can maintain trajectory tracking without imposing excessive mechanical stress on the fin assembly. However, as mean wind speed intensifies to 11.0 knots, the fin deflection patterns reveal significantly larger amplitudes and higher-frequency oscillations, indicating that the servos are compelled to execute rapid and extensive travel cycles to counteract the growing aerodynamic disturbances (see Fig. 27).

These prolonged and fast actuation requirements present substantial mechanical challenges, as servo motors operating under such demanding conditions are prone to accelerated wear, increased heat generation, and potential premature failure.

A significant contribution of this research lies in the computational efficiency achieved through the implementation of all modeling, simulation, and control algorithms in the Julia programming language. The cascaded controller simulations demonstrated exceptional runtime performance, with step response and helix trajectory simulations completing in approximately 1.95 milliseconds, requiring only 8–9 thousand memory allocations and approximately 3 MiB of memory. The integration of the Fuzzy Inference System into the cascaded controller, while naturally increasing computational complexity due to the additional fuzzification, inference, and defuzzification operations at each time step, still achieved remarkably efficient execution times of approximately 45 milliseconds per simulation, with memory allocations of approximately 19 MiB. Furthermore, the comprehensive Monte Carlo simulation combining both FIS-cascaded and conventional cascaded controllers for helix trajectory tracking across 200 random wind conditions completed in approximately 48 seconds, demonstrating Julia's capability to handle extensive parametric studies without prohibitive computational costs. Although efficient, development of fast programs in Julia requires code optimization techniques which demand a programming habit unique to Julia. Some examples of code optimizations include avoiding vector and matrix computions, wrapping most if not all of the code in functions, etc. Interested reader is refered to [21] for detailed descriptions of inner workings and methods to optimize Julia codes.

Ecosystem of Julia has feature-rich packages for scientific purposes but compared to traditional softwares like Python and MATLAB, many tools are not developed as of the time of writing. Specific example of Julia lacking behind MATLAB; is the turbulent wind models toolset which was implemented by authors in Julia code for the purposes of this research.

Conclusions

This research has successfully developed and evaluated a novel Micro Single-Rotor Unmanned Aerial Vehicle concept incorporating an integrated FPV camera system, along with a fuzzy-enhanced cascaded position controller designed to improve disturbance rejection capabilities under turbulent wind conditions. Through comprehensive modeling, numerical simulation, and performance analysis, several significant findings and contributions have emerged from this study.

The integration of a Fuzzy Inference System for real-time gain scheduling of the cascaded position controller has yielded measurable performance improvements in trajectory tracking. Under indoor flight conditions without wind disturbances, the FIS-cascaded controller achieved root mean square error reductions of 14.2 percent and 13.6 percent for the horizontal axes in step response simulations, while helical trajectory tracking showed improvements of 13.8 percent and 18.5 percent along the x and y axes, respectively. The fuzzy-enhanced controller consistently demonstrated faster settling times and smoother convergence toward reference trajectories compared to the manually tuned conventional cascaded controller, indicating the effectiveness of intelligent gain adaptation in improving transient response characteristics.

In outdoor simulations incorporating mean wind speeds of 5 knots with stochastic turbulence modeled using Dryden's approximation, the performance comparison revealed more nuanced outcomes. The FIS-cascaded controller maintained superior performance in vertical position control, achieving substantial RMSE reductions of approximately 50 percent and 43 percent for step and helix references, respectively.

However, the conventional cascaded controller exhibited better horizontal tracking performance under these turbulent conditions, suggesting that the current fuzzy gain scheduling implementation requires further optimization for multi-axis disturbance rejection in complex wind environments. This finding highlights the inherent trade-offs in controller design and the need for balanced tuning approaches.

The robustness evaluation against increasing mean wind speeds revealed that the FIS-cascaded controller maintains consistent performance up to approximately 11.6 knots, beyond which significant performance deterioration occurs, establishing a practical operational envelope for the proposed control system.

Several limitations were identified that warrant future investigation. The steady-state errors observed in continuous trajectory tracking proved difficult to eliminate through parameter tuning alone, indicating the need for structural modifications to membership functions or inference methods. The controller's performance degraded substantially at mean wind speeds exceeding 15 knots, suggesting the necessity of adaptive or switching control strategies for extreme weather conditions. Additionally, instances of integral windup were observed in the FIS-controlled system during initial transient phases, emphasizing the critical importance of proper anti-windup parameter calibration.

Future works on this subject include development of a novel controller algorithm to overcome the deficiencies in FIS-cascaded controller and full quaternion based controller to overcome inherent singularities (like gimbal-lock phenomena) in eulers method of rotations.

Nomenclature

F	Force, N
M	Moment, N.m
V	Velocity, m/s
m	Mass, kg
N	Propeller/motor shaft rate, RPM
MOI	Matrix Of Inertia, kg.m^2
K	Design Constants, various dimensions used
P	Proportional gain term in control scheme
I	Integral gain term in control scheme
D	Derivative gain term in control scheme / Drag force

Greek symbols

ρ	Density, kg/m^3
α	Incidence angle, radians

Subscript

$_x$	Vector component along x axis
$_y$	Vector component along y axis
$_z$	Vector component along z axis
$_D$	Drag
$_F$	Fin
$_T$	Thrust
$_M$	Anti-torque
$_{lon}$	Longitudinal component of wind turbulence vector
$_{lat}$	Latitudinal component of wind turbulence vector
$_{vert}$	Vertical component of wind turbulence vector
$_{sc}$	parameters used for scaling FIS inputs and outputs

Superscript

$^{\mathcal{E}}$	related to inertial earth-fixed frame
$^{\mathcal{B}}$	related to body-fixed frame
*	unscaled inputs and outputs of FIS
^{sat}	saturated value

References

[1] L.S.T. Win, S.K.H. Win, D. Sufiyan, G.S. Soh, S. Foong, Achieving efficient controlled flight with a single actuator, in: 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2020, pp. 1625-1631.

[2] J.-S. Moon, C. Kim, Y. Youm, J. Bae, UNI-Copter: A portable single-rotor-powered spherical unmanned aerial vehicle (UAV) with an easy-to-assemble and flexible structure, Journal of Mechanical Science and Technology, 32(5) (2018) 2289-2298.

[3] E. Bernardes, F. Boyer, S. Viollet, Modelling, control and simulation of a single rotor UAV with swashplateless torque modulation, Aerospace Science and Technology, 140 (2023) 108433.

[4] O.C. Carholt, E. Fresk, G. Andrikopoulos, G. Nikolakopoulos, Design, modelling and control of a single rotor UAV, in: 2016 24th Mediterranean Conference on Control and Automation (MED), IEEE, 2016, pp. 840-845.

[5] T. Susanto, M.B. Setiawan, A. Jayadi, F. Rossi, A. Hamdhi, J.P. Sembiring, Application of unmanned aircraft PID control system for roll, pitch and yaw stability on fixed wings, in: 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), IEEE, 2021, pp. 186-190.

[6] F. Rekabi, F. A Shirazi, M.J. Sadigh, Robust Performance Analysis for a Cascade Nonlinear H∞ Control Algorithm in Quadrotor Position Tracking, AUT Journal of Mechanical Engineering, 4(2) (2020) 151-168.
[7] M. Kahouadji, M.R. Mokhtari, A. Choukchou-Braham, B. Cherki, Real-time attitude control of 3 DOF quadrotor UAV using modified super twisting algorithm, Journal of the Franklin Institute, 357(5) (2020) 2681-2695.
[8] B. Ma, Z. Liu, Q. Dang, W. Zhao, J. Wang, Y. Cheng, Z. Yuan, Deep reinforcement learning of UAV tracking control under wind disturbances environments, IEEE Transactions on Instrumentation and Measurement, 72 (2023) 1-13.
[9] P. Portillo, L.E. Garza-Castañón, L.I. Minchala-Avila, A. Vargas-Martínez, V. Puig Cayuela, P. Payeur, Robust nonlinear trajectory controllers for a single-rotor UAV with particle swarm optimization tuning, Machines, 11(9) (2023) 870.
[10] N.W. Madebo, Enhancing intelligent control strategies for UAVs: A comparative analysis of fuzzy logic, fuzzy PID, and GA-optimized fuzzy PID controllers, IEEE Access, 13 (2025) 16548-16563.
[11] C. Powers, D. Mellinger, V. Kumar, Quadrotor Kinematics and Dynamics, in: K.P. Valavanis, G.J. Vachtsevanos (Eds.) Handbook of Unmanned Aerial Vehicles, Springer Netherlands, Dordrecht, 2015, pp. 307-328.
[12] A. Tewari, Atmospheric and space flight dynamics: modeling and simulation with MATLAB® and Simulink®, Springer, 2007.
[13] S.A. Raza, Autonomous UAV control for low-altitude flight in an urban gust environment, Carleton University, 2015.
[14] X. Liu, A. Abà, P. Capone, L. Manfriani, Y. Fu, Atmospheric disturbance modelling for a piloted flight simulation study of airplane safety envelope over complex terrain, Aerospace, 9(2) (2022) 103.
[15] J.C. Yeager, Implementation and testing of turbulence models for the F18-HARV simulation, 1998.
[16] K. Cole, A. Wickenheiser, Spatio-temporal wind modeling for uav simulations, arXiv preprint arXiv:1905.09954, (2019).
[17] L. Rundqwist, Anti-reset windup for PID controllers, IFAC Proceedings Volumes, 23(8) (1990) 453-458.
[18] J. Jantzen, Foundations of fuzzy control: a practical approach, John Wiley & Sons, 2013.
[19] M. Bryson, S. Sukkarieh, Inertial sensor-based simultaneous localization and mapping for UAVs, in: Handbook of Unmanned Aerial Vehicles, Springer, 2015, pp. 401-431.
[20] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: A fresh approach to numerical computing, SIAM review, 59(1) (2017) 65-98.
[21] C. Rackauckas, Q. Nie, Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia, Journal of open research software, 5(1) (2017) 15-15.
[22] L. Ferranti, J. Boutellier, Fuzzylogic. jl: A flexible library for efficient and productive fuzzy inference, in: 2023 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2023, pp. 1-5.