CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (1) (2026), Pages 1234-1250
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Design of Fuzzy Logic Controller Based on Gustafson-Kessel Clustering for Quadcopter Altitude Control

Belgis Ainatul Iza*, Heri Purnawan, Moh. Khoridatul Huda, and Nurul Laili

Department of Data Science, Universitas Negeri Surabaya, Indonesia.

Abstract

Quadcopter altitude control is a critical issue in unmanned aerial vehicle systems due to its nonlinear dynamics and impact on operating conditions. Various fuzzy logic-based approaches have been used to address these nonlinear characteristics. However, in many previous studies, the rule structure and membership function were determined heuristically or based on expert knowledge, resulting in a less systematic design process that did not fully represent the distribution of system dynamics data. This study proposes a data-driven fuzzy control approach using the Gustafson–Kessel clustering algorithm to automatically generate fuzzy rules from system data. This algorithm is used to identify an ellipsoidal cluster structure in the input space formed by the error variable (e) and the error change (Δe). The cluster center parameters are used to construct Gaussian membership functions and a rule base for the fuzzy inference system that generates control signals for the quadcopter's vertical dynamics. Evaluation is conducted through quadcopter altitude control simulations using several error metrics. The simulation results show Integral Squared Error value of 0.363789, Integral Absolute Error of 0.554409, Root Mean Square Error of 0.190637, and Mean Absolute Error of 0.055386, with a maximum error of 0.999600 in the beginning of the system response.

Keywords: Data-Driven Control; Fuzzy Logic Controller; Gustafson-Kessel Clustering; Quadcopter.

1. Introduction

Quadcopters are an Unmanned Aerial Vehicle (UAV) platform widely used in various applications such as aerial mapping, environmental monitoring, infrastructure inspection, and autonomous navigation systems [1–3]. Compared to other types of UAVs, quadcopters have a relatively simple mechanical configuration and vertical take-off and landing (VTOL) capabilities, enabling operation in confined spaces and complex environments [4–6].

Despite their simple mechanical structure, quadcopter dynamics are complex. These systems are nonlinear with multivariable characteristics and strong coupling between variables. Furthermore, system dynamics are also affected by external disturbances such as air turbulence, parameter uncertainty, and changes in operating conditions [1, 2, 7–9]. These characteristics make designing a stable and robust control system a major challenge in UAV control system research [5, 10].

Various control methods have been developed for quadcopter systems. The Proportional Integral Derivative (PID) approach is the most widely used due to its simple algorithmic structure

*Corresponding author. E-mail: belgisiza@unesa.ac.id

Submitted: March 12, 2026 Reviewed: April 28, 2026 Accepted: May 18, 2026
DOI: https://doi.org/10.18860/cauchy.v11i1.41543

and ease of implementation [1, 3, 11]. However, PID controller performance can degrade when the system encounters strong nonlinear dynamics or changes in system parameters due to external disturbances [3, 10]. Therefore, various studies have developed more adaptive control approaches to improve system performance [8, 9].

One widely used approach is intelligent control, such as fuzzy logic control, neural network control, and adaptive control [12–14]. Fuzzy Logic Controllers (FLCs) have the ability to handle nonlinear systems without requiring highly precise mathematical models, making them widely applied in robotics and aerial vehicle systems [15].

However, most FLC implementations in UAV systems still use manually designed fuzzy rules based on expert knowledge [15]. This approach has several limitations, including subjectivity in rule determination and difficulty in adapting the rule structure when system characteristics change. Furthermore, the membership function determination process is often heuristic, so it does not always optimally represent the system's data distribution [16].

To address these limitations, a data-driven fuzzy modeling approach has been developed that utilizes clustering techniques to automatically generate fuzzy rules from system data [16, 17]. Some commonly used clustering algorithms in fuzzy system design are K-Means and Fuzzy C-Means (FCM) [17]. However, both methods assume a spherical cluster shape, making them less capable of representing data structures with complex covariances [16].

The Gustafson-Kessel clustering algorithm is an extension of the FCM method that uses an adaptive covariance matrix for each cluster [18]. This approach allows the formation of ellipsoidal clusters, making it more flexible in representing anisotropic data distributions [19]. These characteristics make the Gustafson–Kessel algorithm more suitable for modeling nonlinear relationships in complex dynamic systems [20].

2. Mathematical Model of Quadcopter

The quadcopter dynamics model can be obtained through the classical mechanics formulation using the Newton–Euler approach. The mathematical model can be explained here.

2.1. Coordinate System

Quadcopters are defined using two main coordinate systems, namely inertial frame dan body frame. Inertial frame $\{X, Y, Z\}$ is a coordinate frame that is fixed to the earth and is used to represent the global position of the system. On the contrary, body frame $\{x, y, z\}$ tied to the center of mass of the quadcopter and moves according to the vehicle's dynamics.

Quadcopter position on inertial frame expressed by the position vector [21]

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{1}$$

which represents the translation coordinates along the axis X, Y, and Z.

The quadcopter's orientation is expressed using Euler angle parameters, which define the rotation of the body frame relative to the inertial frame. The orientation vector is expressed as [21]

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{2}$$

where ϕ, θ, and ψ represent the roll, pitch, and yaw angles. These parameters describe the rotation of the quadcopter about the longitudinal, lateral, and vertical axes of the body frame.

2.2. Rotation Matrix

The coordinate transformation from the body frame to the inertial frame is expressed using a rotation matrix that depends on the Euler angles ϕ, θ, and ψ, expressed in Eq. (2). This rotation matrix represents the orientation of the quadcopter with respect to the inertial coordinate frame and is given as [22, 23]

$$\mathbf{R}(\phi,\theta,\psi) = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3}$$

with short notation trigonometric functions are defined as $c_{(\cdot)} = \cos(\cdot)$ *and* $s_{(\cdot)} = \sin(\cdot)$.

Rotation matrix $\mathbf{R}(\phi,\theta,\psi)$ in Eq. (3) is used to transform the vectors expressed in the body frame into the inertial frame, so that the relationship between the two coordinate frames can be expressed as

$$\mathbf{v}_I = \mathbf{R}(\phi,\theta,\psi)\,\mathbf{v}_B \tag{4}$$

where $\mathbf{v}_B$ and $\mathbf{v}_I$ represent the vectors on the body frame and inertial frame.

2.3. Translation Dynamics

The translational dynamics of a quadcopter are derived based on Newton's second law, which states that the resultant force acting on a system is equal to its mass multiplied by the acceleration of the center of mass. Therefore, the translational dynamics can be expressed as [23]

$$m\ddot{\boldsymbol{\eta}} = \mathbf{F} \tag{5}$$

in which m states the mass of the quadcopter, η is expressed in Eq. (1) and $\mathbf{F}$ is the resultant force acting on the system.

In a quadcopter, the primary forces affecting translational dynamics consist of the total rotor thrust and gravity. The thrust is generated along the z-axis of the body frame and must be transformed into the inertial frame using Eq. (3) and Eq. (4). Taking into account the influence of gravity, the translational dynamics equation in Eq. (5) can be written as

$$m\ddot{\boldsymbol{\eta}} = \mathbf{R}(\phi,\theta,\psi)\begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{6}$$

with U_1 states the total thrust generated by the four rotors and g is the acceleration of gravity.

By substituting Eq. (3) into Eq. (6), the translational dynamics components are obtained on each axis as follows

$$m\ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)U_1 \tag{7}$$

$$m\ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)U_1 \tag{8}$$

$$m\ddot{z} = (\cos\phi\cos\theta)U_1 - mg \tag{9}$$

Eqs. (7)–(9) show that the translational dynamics of the quadcopter are nonlinear and coupled to each other through Euler angles ϕ, θ, and ψ which determines the orientation of the vehicle with respect to the inertial frame.

2.4. Rotational Dynamics

The rotational dynamics of the quadcopter are modeled using Euler's equations for a rigid body, which describe the relationship between the torque of a force and the change in the angular velocity of the system. The rotational dynamics equations are expressed as [23, 24]

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \boldsymbol{\tau} \tag{10}$$

in which $\mathbf{I}$ states the moment of inertia matrix, $\boldsymbol{\omega}$ is the angular velocity vector, and $\boldsymbol{\tau}$ is the torque vector generated by the rotor system. The angular velocity vector of the quadcopter on the body frame is $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ where p, q, and r represent the angular velocities about the roll, pitch, and yaw axes, respectively.

In addition, the kinematic relationship between the angular velocity at the body frame (p, q, r) and the rate of change of the Euler angle $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ is given by the following transformation [23]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{11}$$

Eq. (11) shows that the rate of change of the Euler angle is a nonlinear function of the angular velocity of the body frame. This relationship is necessary to ensure consistency between the representation of kinematics and rotational dynamics in a quadcopter model.

Assuming that the mass distribution of the quadcopter is symmetrical about the principal axis, the moment of inertia matrix can be expressed in diagonal form as

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{12}$$

with I_x, I_y, and I_z representing the moments of inertia about the x, y, and z axes, respectively.

By substituting $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ and Eq. (12) into Euler's equation in Eq. (10), the rotational dynamics on each axis are obtained as follows

$$I_x\dot{p} = \tau_\phi + (I_y - I_z)qr \tag{13}$$

$$I_y\dot{q} = \tau_\theta + (I_z - I_x)pr \tag{14}$$

$$I_z\dot{r} = \tau_\psi + (I_x - I_y)pq \tag{15}$$

Eqs. (13)–(15) show that the rotational dynamics of the quadcopter are nonlinear due to the gyroscopic coupling term involving the multiplication of angular velocities between the axes.

2.5. Rotor Input Model

The quadcopter generates lift and torque through four rotors that rotate at angular velocities ω_i, $i = 1, 2, 3, 4$. The angular velocities of these rotors generate thrust and torque that control the vehicle's translational and rotational motion.

The total thrust generated by the rotor system is expressed as [23]

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{16}$$

where b represents the thrust coefficient that relates the rotor speed to the thrust generated.

In addition to generating thrust, differences in rotor speed also generate torque about the quadcopter's main axes, which are used to control the system's orientation. Torques about the roll, pitch, and yaw axes are each expressed as [23]

$$U_2 = \tau_\phi = bl(\omega_4^2 - \omega_2^2) \tag{17}$$

$$U_3 = \tau_\theta = bl(\omega_3^2 - \omega_1^2) \tag{18}$$

$$U_4 = \tau_\psi = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{19}$$

where l represents the length of the rotor arm relative to the center of mass of the quadcopter, b is the thrust coefficient, and d is the aerodynamic drag coefficient related to the yaw moment generated by the rotor.

To ensure reproducibility of the simulation results, all parameters used in the quadcopter model and numerical implementation are explicitly defined in Table 1.

Table 1: Simulation Parameters

Parameter (Symbol)	Value	Unit	Parameter (Symbol)	Value	Unit
Mass (m)	1	kg	Reference altitude (z_{ref})	1	m
Gravitational acc. (g)	9.81	m/s^2	Integration step size (dt)	0.01	s
Arm length (l)	0.2	m	Simulation time (T)	10	s
Moment of inertia (I_x)	0.02	kg·m^2	No. of data samples (N_{data})	400	-
Moment of inertia (I_y)	0.02	kg·m^2	No. of clusters (c)	3	-
Moment of inertia (I_z)	0.04	kg·m^2	Fuzzy width parameter (σ)	0.5	-
Thrust coefficient (k_t)	1.5×10^{-5}	N/(rad/s)^2	Proportional gain (K_p)	8	-
Drag coefficient (k_d)	2.5×10^{-7}	N·m/(rad/s)^2	Derivative gain (K_d)	4	-
Initial altitude (z_0)	0	m	Min. actuator input ($u_{\min}$)	0	N
Initial velocity (v_0)	0	m/s	Max. actuator input ($u_{\max}$)	20	N

2.6. State-Space Model

For the purposes of control system analysis and design, the quadcopter dynamics model can be represented in state-space form. This representation allows multivariable nonlinear systems to be expressed in a more systematic first-order differential equation form for control analysis.

The system state vector is defined as

$$\mathbf{x} = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ \phi\ \theta\ \psi\ p\ q\ r]^T \tag{20}$$

which represents the translational position, linear velocity, orientation of the quadcopter in Euler angles, and angular velocity about each axis of rotation.

The system control inputs are the total thrust force U_1, roll torque U_2, pitch torque U_3, and yaw torque U_4, which are described in Eqs. (16)–(19). Then, the input vector is defined as

$$\mathbf{u} = \begin{bmatrix} U_1 & U_2 & U_3 & U_4. \end{bmatrix}^T \tag{21}$$

By using Eq. (20) and Eq. (21), the quadcopter dynamics model can be generally expressed as a nonlinear system as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ [25], which represents the translational and rotational dynamics derived from the Newton–Euler equations. This model is then used as the basis for designing a control system based on a fuzzy logic controller using a clustering approach.

In the altitude control simulation, it is assumed that the roll and pitch angles are small ($\phi \approx 0, \theta \approx 0$), so that the influence of rotational coupling on vertical translational dynamics can be significantly neglected. With this assumption, the 6-DOF system is reduced to dominant dynamics along the vertical axis (z-axis) for altitude control design purposes.

3. Methods

This section describes the methodological framework used to design a fuzzy logic-based quadcopter altitude control system. All the stages are used as the basis for evaluating controller performance through system simulation.

3.1. Dataset Generation

The dataset used in the clustering process was obtained through numerical simulations of the quadcopter dynamic model formulated in Section 2. This dataset consists of a pair of input variables, namely error (e) and error change (Δe), which represent the dynamic conditions of the system during the altitude control process.

Mathematically, the error is defined as the difference between the altitude reference signal $z_{ref}(t)$ and the actual altitude of the quadcopter $z(t)$, namely

$$e(t) = z_{ref}(t) - z(t). \tag{22}$$

While the change in error is defined as the discrete derivative of the error with respect to time

$$\Delta e(t) = \frac{e(t) - e(t - \Delta t)}{\Delta t}. \tag{23}$$

To generate a representative dataset, the system is given a reference signal in the form of a step function and varying initial conditions to produce a dynamic response encompassing both transient and steady-state conditions. Furthermore, varying the initial control signal is applied to excite the system, resulting in a data distribution spread across the input space $(e, \Delta e)$.

The simulation was performed using the fourth-order Runge-Kutta (RK4) method with a time step of Δt. At each time step, the values of $e(t)$ and $\Delta e(t)$ are calculated and stored as data pairs. Thus, the dataset obtained can be expressed as

$$\mathcal{D} = \{(e_k, \Delta e_k)\}_{k=1}^{N}. \tag{24}$$

This dataset is then used as input in the clustering process using the Gustafson-Kessel algorithm to automatically build a fuzzy rule structure.

3.2. Cluster Number Selection

Determining the number of clusters is a crucial step in the clustering process because it directly affects the structure of the resulting fuzzy rules. In this study, the number of clusters was systematically selected, taking into account the separation quality and cluster density.

To evaluate clustering quality, the Xie-Beni Index (XB) is used, which is mathematically defined as

$$XB = \frac{\sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^2 \|x_k - v_i\|^2}{N \cdot \min_{i \neq j} \|v_i - v_j\|^2} \tag{25}$$

This index measures the ratio between intra-cluster compactness and inter-cluster separation, with a smaller value indicating better clustering quality. A lower XB value indicates more compact and well-separated clusters, which is desirable for constructing reliable fuzzy rules.

Several candidate numbers of clusters were evaluated using the Xie-Beni index. The optimal number of clusters is selected as the one that minimizes the XB value. Based on this criterion, four clusters were chosen, as they yield the lowest XB value among the tested configurations. This ensures that the selected model achieves an optimal balance between intra-cluster compactness and inter-cluster separation, leading to a representative fuzzy rule structure.

3.3. Gustafson–Kessel Clustering

The Gustafson-Kessel method is a fuzzy clustering algorithm capable of identifying elliptical cluster structures using an adaptive Mahalanobis metric. Unlike the Fuzzy C-Means, this algorithm allows each cluster to have a different covariance matrix, thus representing an anisotropic data distribution. In general, the Gustafson-Kessel algorithm minimizes an objective function defined as

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m D_{ik}^2 \tag{26}$$

where c represents the number of clusters, N is the number of data points, u_{ik} is the degree of membership of the kth data point to the ith cluster, and $m > 1$ is a fuzzification parameter that controls the level of fuzziness of the cluster partition.

The distance between data points x_k and the cluster center v_i is calculated using the Mahalanobis metric, defined as

$$D_{ik}^2 = (x_k - v_i)^T A_i (x_k - v_i) \tag{27}$$

where A_i is a metric matrix corresponding to the covariance structure of the ith cluster.

Cluster centroids in Eq. (27) are calculated using a weighted average of the membership degrees given by

$$v_i = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m}. \tag{28}$$

From Eq. (28), the cluster covariance matrix is calculated as

$$F_i = \frac{\sum_{k=1}^{N} u_{ik}^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{N} u_{ik}^m}. \tag{29}$$

The covariance matrix F_i in Eq. (29) is then used to construct a metric matrix A_i that determines the elliptical shape of each cluster. This approach allows the Gustafson-Kessel algorithm to capture the complex, non-isotropic distribution structure of the data.

Next, the metric matrix A_i is formed from the covariance matrix F_i by determinant normalization to keep the cluster volume constant, which is expressed as

$$A_i = \frac{(\det F_i)^{-1/n}}{F_i} \tag{30}$$

where n is the data dimension. The membership degree of u_{ik} is updated using

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ik}}{D_{jk}}\right)^{\frac{2}{m-1}}}. \tag{31}$$

The iteration process is carried out until convergence, which is determined based on changes in the membership matrix, namely

$$\|U^{(t+1)} - U^{(t)}\| < \varepsilon \tag{32}$$

where ε is a predetermined tolerance.

3.4. Cluster to Fuzzy Rule Conversion

The results of the clustering process are then used to build a rule base in the Fuzzy Logic Controller. Each cluster obtained from the Gustafson-Kessel algorithm is interpreted as a fuzzy rule that represents the local characteristics of the system.

In general, the resulting fuzzy rules can be expressed in the form

$$IF\ e\ is\ A_i\ AND\ \Delta e\ is\ B_i\ THEN\ u\ is\ C_i \tag{33}$$

where e represents the system error, Δe is the error change, and u is the control signal. The fuzzy sets A_i, B_i, and C_i are derived from the cluster centers generated by the Gustafson-Kessel algorithm.

The membership function used to represent the fuzzy sets for each variable is a Gaussian function, which is expressed as

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right) \tag{34}$$

where c represents the center of the membership function obtained from the cluster center, while σ represents the distribution parameter related to the variance of the data distribution in that cluster.

To establish a mathematical relationship between the covariance matrix obtained from the clustering process and the Gaussian membership function parameter σ, a dimensional reduction from the covariance matrix to scalar variance is performed.

Let the input vector be defined as $x = [e, \Delta e]^T$, then the covariance matrix $F_i \in \mathbb{R}^{2\times 2}$ for the i-th cluster can be expressed as:

$$F_i = \begin{bmatrix} [F_i]_{11} & [F_i]_{12} \\ [F_i]_{21} & [F_i]_{22} \end{bmatrix} \tag{35}$$

The variance of each input variable is obtained from the diagonal elements of the covariance matrix:

$$\sigma_{e,i}^2 = [F_i]_{11}, \quad \sigma_{\Delta e,i}^2 = [F_i]_{22} \tag{36}$$

The Gaussian distribution parameters are then defined as the square root of the corresponding variances:

$$\sigma_{e,i} = \sqrt{[F_i]_{11}}, \quad \sigma_{\Delta e,i} = \sqrt{[F_i]_{22}} \tag{37}$$

The clustering process in this study is performed on the input space only, namely the pair $(e, \Delta e)$, without directly involving the output variables. Thus, each cluster represents a local partition of the input space.

The fuzzy rule consequences (u_i) are not obtained from the clustering process but are determined as local control actions associated with each rule. The final control signal is obtained through a weighted aggregation of all rules, namely

$$u = \sum_{i=1}^{c} w_i u_i, \quad \sum_{i=1}^{c} w_i = 1 \tag{38}$$

where w_i is the activation degree of the ith rule calculated from the membership function.

Thus, the Gaussian membership function in Eq. (34) is constructed independently for each input variable using the corresponding center c and standard deviation σ derived from the clustering results. To provide a more rigorous mathematical foundation for the conversion from clustering results to fuzzy inference system parameters, a formal formulation is presented in the following subsection.

3.5. Mathematical Formulation of GK Fuzzy Control

In contrast to conventional fuzzy logic controllers where rule bases and membership functions are determined heuristically, this study proposes a mathematically grounded construction of a fuzzy controller based on the covariance structure of Gustafson–Kessel (GK) clustering.

Let the input vector be defined as

$$x = [e, \Delta e]^T. \tag{39}$$

From the GK clustering process, each cluster i is characterized by a center v_i and a covariance matrix $F_i \in \mathbb{R}^{2\times 2}$. The distance metric used in GK is defined in Eq. (27).

Based on this structure, a multivariate Gaussian membership function can be defined as

$$\mu_i(x) = \exp\left(-\frac{1}{2}(x - v_i)^T F_i^{-1}(x - v_i)\right). \tag{40}$$

To construct a fuzzy inference system with separable inputs, the covariance matrix is reduced into marginal variances by extracting its diagonal elements

$$\sigma_{e,i}^2 = [F_i]_{11}, \quad \sigma_{\Delta e,i}^2 = [F_i]_{22}. \tag{41}$$

Thus, the Gaussian membership functions for each input variable are defined as

$$\mu_{e,i}(e) = \exp\left(-\frac{(e - c_{e,i})^2}{2\sigma_{e,i}^2}\right), \quad \mu_{\Delta e,i}(\Delta e) = \exp\left(-\frac{(\Delta e - c_{\Delta e,i})^2}{2\sigma_{\Delta e,i}^2}\right). \tag{42}$$

This formulation establishes a direct mathematical relationship between the statistical structure of the data (through F_i) and the parameters of the fuzzy membership functions, eliminating the need for heuristic tuning.

Furthermore, each cluster can be interpreted as representing a local approximation of the nonlinear system dynamics. Therefore, the overall control law can be expressed as a weighted combination of local control actions

$$u(x) = \sum_{i=1}^{c} w_i(x) u_i \tag{43}$$

where $w_i(x)$ is the normalized firing strength of the i-th rule.

This interpretation shows that the proposed method constructs a data-driven piecewise nonlinear control system, where the partitioning of the input space is determined by the GK clustering structure.

Compared to conventional Fuzzy C-Means (FCM)-based approaches, which use a Euclidean distance

$$D_{ik}^2 = \|x_k - v_i\|^2 \tag{44}$$

the GK method employs an adaptive Mahalanobis distance that accounts for the covariance structure of each cluster. As a result, the proposed method is capable of capturing anisotropic data distributions and correlations between input variables, leading to a more accurate representation of the system dynamics.

3.6. Control Algorithm

The fuzzy inference system used in this study is a Mamdani type with two inputs $(e, \Delta e)$ and one output u. The logical AND operator is implemented using the minimum operator, while the rule implication also uses the minimum operator. The output aggregation of all rules is performed using the maximum operator. For each ith rule, the activation degree is calculated as

$$w_i = \min\{\mu_{e,i}(e), \mu_{\Delta e,i}(\Delta e)\} \tag{45}$$

which is then normalized to

$$\bar{w}_i = \frac{w_i}{\sum_{j=1}^{c} w_j}. \tag{46}$$

The control signal is obtained through a defuzzification process using the centroid method, which is generally expressed as

$$u = \frac{\int u\, \mu(u)\, du}{\int \mu(u)\, du}. \tag{47}$$

The input variables are normalized before being processed in the fuzzy system to maintain scale consistency, while the output variables are multiplied by a scale factor to meet the system's actuation requirements. The membership function parameters are determined based on the clustering results, so the entire FIS structure is built in a data-driven manner.

The rule base structure is implicitly formed by the clustering results, where each ith cluster generates one fuzzy rule according to the form in Eq. Thus, the number of rules is equal to the number of clusters (c), eliminating the need for manual rule table design.

The membership function parameters for each rule are determined directly from the clustering results. The membership function center is obtained from the center of cluster v_i, while the dispersion parameters are derived from the diagonal elements of the covariance matrix F_i. Thus, all membership function parameters are obtained systematically and data-driven.

The design procedure for a Gustafson-Kessel clustering-based fuzzy controller for a quadcopter system involves several main stages, including system dynamics modeling, dataset formation, clustering, fuzzy rule formation, and controller implementation on the quadcopter model. In summary, the controller design steps proposed in this study are summarized in the algorithm 1.

Algorithm 1: Gustafson-Kessel Based Fuzzy Controller

Formulate the nonlinear dynamic model of the quadcopter;
Generate input-output dataset from the quadcopter model;
Select the number of clusters c;
Initialize the membership matrix U;
repeat
  Compute cluster centers v_i;
  Estimate cluster covariance matrices F_i;
  Update membership degrees u_{ik};
until *objective function* J *converges*;
Extract cluster parameters (v_i, F_i);
Convert clusters into fuzzy rules;
Construct the fuzzy inference system (FIS);
Implement the fuzzy controller in the quadcopter dynamic model;

3.7. Simulation Setup

System simulations were performed using a script-based MATLAB environment (*m-file*) without utilizing the Simulink platform. Numerical integration of the quadcopter's nonlinear dynamics model was performed using the fourth-order Runge-Kutta (RK4) method, which is widely used to solve nonlinear differential equations due to its stability and accuracy.

In general, the fourth-order Runge-Kutta integration scheme can be expressed in Eq. (48).

$$x_{k+1} = x_k + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{48}$$

with

$$k_1 = f(x_k, u_k) \qquad k_3 = f\left(x_k + \frac{\Delta t}{2}k_2, u_k\right)$$
$$k_2 = f\left(x_k + \frac{\Delta t}{2}k_1, u_k\right) \qquad k_4 = f\left(x_k + \Delta t\, k_3, u_k\right)$$

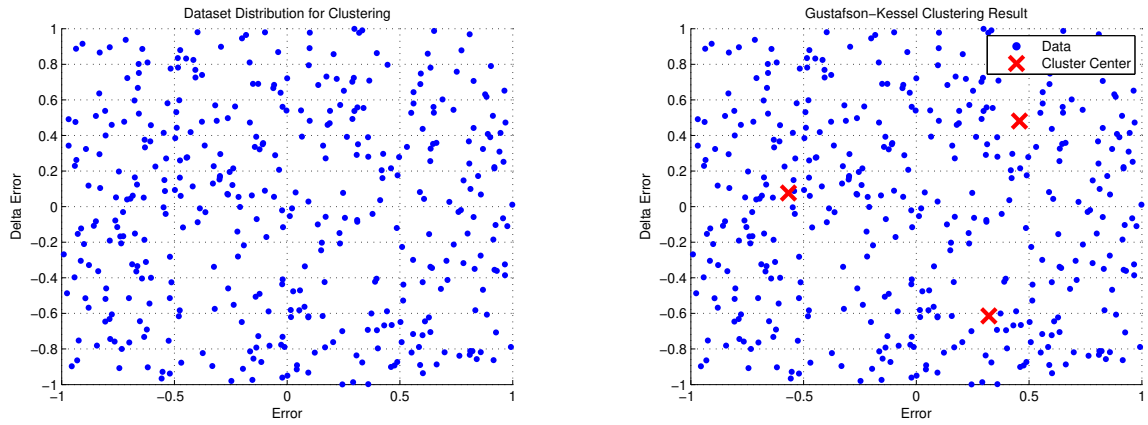
where x_k represents the system state vector at time k, Δt is the integration time step, and $f(\cdot)$ represents the nonlinear dynamics of the quadcopter derived from the Newton-Euler equations. This method is used to simulate the system's response to control signals generated by a clustering-based fuzzy controller.

4. Results and Discussion

This section evaluates the performance and robustness of the proposed fuzzy controllers. The analysis covers clustering dataset evaluation and comparative dynamic responses under both nominal and disturbed conditions.

4.1. Dataset Distribution and Clustering Results

Fig. 1a shows the distribution of the dataset used in the clustering process. The dataset consists of a pair of system error variables (e) and error changes (Δe) that represent the dynamic conditions of the system during the quadcopter altitude control process. The distribution of data points is seen in the positive and negative regions, indicating that the dataset covers conditions when the quadcopter altitude is below and above the reference value. The distribution of data scattered across two-dimensional space forms the basis for the data-driven fuzzy modeling process. The presence of data in various regions of the input space allows the Gustafson–Kessel algorithm to identify several local structures that represent different system operating conditions. Mathematically, the clustering process minimizes the objective function in Eq. (26). Adequate



(a) Dataset Distribution for Clustering **(b)** Gustafson-Kessel Clustering Result

Fig. 1: Clustering Analysis Overview

representation of data distribution is a crucial factor in developing a fuzzy rule base. This is reflected in the Mean Absolute Error value of 0.055386, indicating that the relationship between the system state variables and control signals can be fairly accurately represented by the fuzzy model generated from the dataset.

Fig. 1b shows the results of the clustering process using the Gustafson Kessel algorithm. The dataset is divided into several clusters, each with a different cluster center and covariance matrix. The presence of ellipsoidal clusters allows the algorithm to capture the anisotropic structure of data distribution. This is important because the relationship between error variables and error changes in a quadcopter system is nonlinear. Thus, each cluster represents a different local system condition. In the context of fuzzy controller design, each cluster is interpreted as a single fuzzy rule. This process establishes a direct relationship between the clustering stage and the development of the control system. Conceptually, this relationship can be expressed as

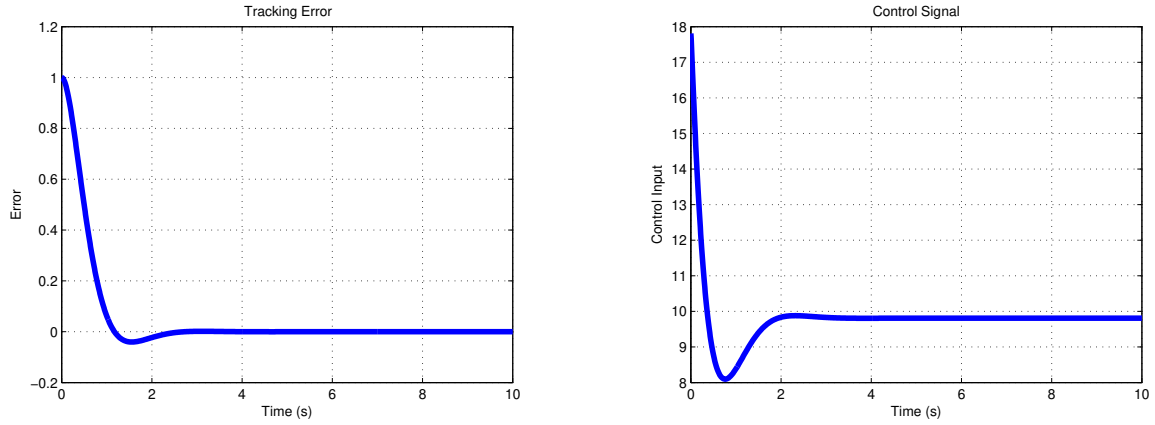
$$Cluster_i \rightarrow Rule_i$$

Thus, the number of clusters generated directly determines the number of fuzzy rules used in the control system. The clustering's ability to represent the system's data distribution contributes to control performance. This can be observed from the Root Mean Square Error value of 0.190637, indicating that the mean squared deviation between the system output and the reference signal remains within relatively small limits throughout the simulation.

4.2. Control Performance and Comparative Analysis

Fig. 2a shows the evolution of the tracking error, defined as the difference between the reference altitude and the quadcopter's actual altitude. At the beginning of the simulation, the error was quite large due to the difference between the system's initial conditions and the reference value. Over time, the error gradually decreased until it approached zero. This decrease indicates that the control system effectively adjusted the control signal to reduce the deviation between the system output and the reference. In stability analysis, a decrease in the error toward zero indicates that the system is converging toward equilibrium. Furthermore, the relatively small error amplitude after the transient phase indicates that the system maintains stability at steady-state conditions. The Mean Absolute Error value of 0.055386 indicates that the average absolute error during the simulation was at a low level. This indicates that the system was able to maintain a response close to the reference throughout the observation period.

Fig. 2b shows the control signal generated by the fuzzy system to regulate the quadcopter's total thrust. At the beginning of the simulation, the control signal has a relatively large value because the system must correct for the difference between the initial condition and the reference



(a) Tracking Error **(b)** Control Signal

Fig. 2: System Performance Evaluation

value. As the quadcopter's altitude approaches the reference value, the control signal amplitude gradually decreases and then stabilizes at a certain value. This condition indicates that the rotor thrust has reached equilibrium with the gravitational force acting on the system. The stable behavior of the control signal in the steady-state phase indicates that the fuzzy controller is capable of producing control actions that do not oscillate excessively. This control signal stability contributes to overall system stability and keeps errors within a small range.

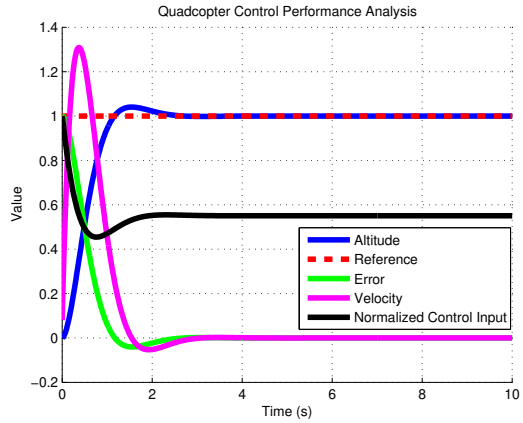


Fig. 3: Quadcopter Control Performance Analysis

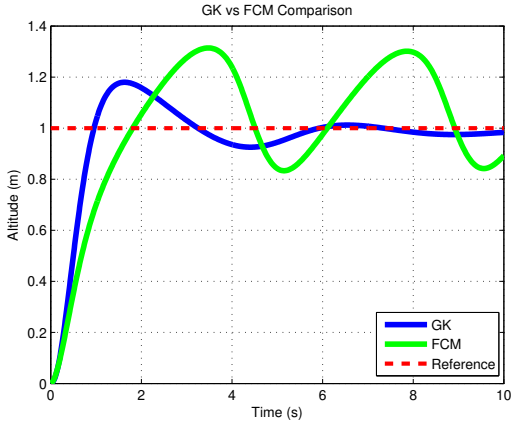


Fig. 4: Comparison of GK and FCM-based controller

Fig. 3 visualizes the dynamic relationships between the system state variables and the generated fuzzy control actions, specifically tracking altitude, reference signal, error, vertical velocity, and the normalized control signal. The close alignment between the altitude response and the reference signal confirms successful tracking, where the gradual error reduction correlates directly with the control signals from the fuzzy inference system. Additionally, the vertical velocity peaks during the transient phase and diminishes as the system reaches a steady state. These coupled variables illustrate the core mechanism of the clustering-based controller, where the fuzzy rule structure derived from the clustering process dictates the control actions based on instantaneous error and its changes. Consequently, the relationship between the methodological stages can be summarized as

$$\textit{Clustering} \rightarrow \textit{Fuzzy Rules} \rightarrow \textit{Control Signal} \rightarrow \textit{System Response}.$$

Fig. 4 show that the GK-based controller achieves lower error values across all performance metrics. The Integral Squared Error (ISE) is 0.457631 for GK compared to 0.738556 for FCM, while the Integral Absolute Error (IAE) is 1.097293 for GK and 1.779432 for FCM. Similarly, the

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are 0.213816 and 0.109620 for GK, respectively, whereas the corresponding values for FCM are 0.271628 and 0.177765. The maximum error values are comparable for both methods. These results indicate that the proposed GK-based controller reduces the overall tracking error more effectively compared to the FCM-based controller under the same simulation conditions.

4.3. Disturbance and Robustness Testing

To evaluate the system's robustness, external disturbances combining a velocity-dependent nonlinear component and random noise were introduced into the quadcopter dynamics

$$d(t) = -0.4, v|v| + 0.1, \text{randn}(t), \tag{49}$$

where v is the vertical velocity and randn(t) represents zero-mean Gaussian noise. These disturbances were incorporated directly into the acceleration equations to better mimic real-world aerodynamic effects and environmental uncertainties. Simulations were conducted under identical conditions for both the Gustafson-Kessel (GK) and Fuzzy C-Means (FCM) methods to ensure a fair comparison. System performance was evaluated using both error-based and time-domain metrics under nominal and disturbed conditions. The comprehensive results are presented in Table 2. Both controllers successfully maintained system stability and convergence to the reference signal without divergence. This demonstrates the robustness of the proposed control methods against nonlinear disturbances and noise within the simulated scenarios.

Table 2: Performance Comparison Between GK and FCM Under Nominal and Disturbed Conditions

	Without Disturbance		With Disturbance	
Metric	GK	FCM	GK	FCM
Error Tracking Metrics				
Integral Squared Error (ISE)	0.457631	0.738556	0.801332	0.805072
Integral Absolute Error (IAE)	1.097293	1.779432	2.356840	2.141813
Root Mean Square Error (RMSE)	0.213816	0.271628	0.282937	0.283596
Mean Absolute Error (MAE)	0.109620	0.177765	0.235449	0.213967
Maximum Error	0.999740	0.999853	0.999760	0.999745
Time-Domain Performance				
Rise Time	0.5000	0.4800	0.5600	0.5400
Settling Time	10.0000	10.0000	10.0000	10.0000
Overshoot (%)	109.9516	110.9739	75.0649	74.8943
Steady-State Error	0.8873	0.4977	0.0741	0.1524

4.4. Stability Analysis

To provide a theoretical justification for the stability of a closed system, a Lyapunov function approach based on tracking error is used. Define error as

$$e(t) = z_{\text{ref}} - z(t), \tag{50}$$

where z_{ref} is a constant reference. The error derivative is given by

$$\dot{e}(t) = -\dot{z}(t) = -v(t). \tag{51}$$

The selected candidate Lyapunov function is of the form

$$V(e) = \frac{1}{2}e^2, \tag{52}$$

which is positive definite for all $e \neq 0$. The time derivative is

$$\dot{V}(e) = e\dot{e}. \tag{53}$$

The control signal generated by a Gustafson-Kessel based fuzzy controller is expressed as a convex combination of local rules

$$u = \sum_{i=1}^{c} w_i u_i, \quad w_i \geq 0, \quad \sum_{i=1}^{c} w_i = 1, \tag{54}$$

with each u_i representing a local control action based on the pair $(e, \Delta e)$.

This structure produces error dynamics that can be expressed in a general form

$$\dot{e}(t) = f(e,t) + d(t), \tag{55}$$

where $f(e,t)$ is the control contribution that depends on the fuzzy rules, and $d(t)$ is the external disturbance assumed to be finite, that is, $|d(t)| \leq d_{\max}$.

If the local control action provides negative feedback to the error around the equilibrium point, then there is a constant $k > 0$ so that the local approximation applies.

$$f(e,t) \approx -ke. \tag{56}$$

Thus, the derivative of the Lyapunov function can be written as

$$\dot{V}(e) = e(-ke + d(t)) = -ke^2 + e\,d(t). \tag{57}$$

Using inequalities

$$|e\,d(t)| \leq |e|\,|d(t)|, \tag{58}$$

upper limit obtained

$$\dot{V}(e) \leq -ke^2 + |e|\,d_{\max}. \tag{59}$$

This inequality indicates that $\dot{V}(e)$ is negative outside a bounded neighborhood around the equilibrium point, so the error $e(t)$ remains bounded at all times. Thus, the closed system satisfies the uniform ultimate boundedness (UUB) criterion under finite perturbations.

It should be noted that this result is a local analysis that relies on the assumption of linear approximation around the equilibrium point. Therefore, global stability is not claimed, and the validity of the system's performance is further confirmed through simulation results.

5. Conclusion

This study implements a quadcopter altitude controller based on a fuzzy system whose parameters are obtained through a Gustafson-Kessel clustering process on the error variable pair (e) and the error change (Δe). The fuzzy rule structure resulting from the clustering process is used to build an inference system that generates control signals for the quadcopter's vertical dynamics.

Simulation results show the Integral Squared Error (ISE) values of 0.363789 and Integral Absolute Error (IAE) of 0.554409, indicating that the accumulated squared and absolute errors remain within a low range throughout the control process. The Root Mean Square Error (RMSE) value of 0.190637 indicates that the mean squared deviation between the system output and the reference is below 0.2 throughout the simulation. Furthermore, the Mean Absolute Error (MAE) value of 0.055386 indicates that the average absolute error of the system is only about 5.5% of the system's output units. The maximum error value observed during the simulation was 0.999600. This value appears at the beginning of the simulation when the system's initial conditions are far from the reference value. After this initial phase, the error amplitude decreases to approximately zero, as reflected in the Mean Absolute Error value of 0.055386.

The consistency between the ISE, IAE, RMSE, and MAE indicates that the deviation of the system output from the reference remains limited throughout the simulation process. Based on the numerical evaluation results, the fuzzy controller built through Gustafson-Kessel clustering produces a system response with a measurable and limited error value throughout the simulation period.

CRediT Authorship Contribution Statement

Belgis Ainatul Iza: Conceptualization, Methodology, Software, Data curation, Formal analysis, Visualization, Writing – original draft. **Heri Purnawan:** Validation, Writing – review and editing. **Moh. Khoridatul Huda:** Methodology, Writing – review and editing. **Nurul Laili:** Investigation, Writing – review and editing.

Declaration of Generative AI and AI-assisted technologies

During the preparation of this work, the author used generative AI tools for language refinement and grammar checking. The author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Declaration of Competing Interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and Code Availability

The datasets generated and the MATLAB codes used in this study are available from the corresponding author upon reasonable request.

References

[1] Samir Bouabdallah. "Design and control of quadrotors with application to autonomous flying". PhD thesis. Epfl, 2007. DOI: 10.5075/epfl-thesis-3727.

[2] Robert Mahony, Vijay Kumar, and Peter Corke. "Multirotor aerial vehicles: Modeling, estimation, and control". In: *IEEE Robotics & Automation Magazine* 19.3 (2012), pp. 20–32. DOI: 10.1109/MRA.2012.2206474.

[3] Farid Kendoul. "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems". In: *Journal of Field Robotics* 29.2 (2012), pp. 315–378. DOI: 10.1002/rob.20414.

[4] Daniel Mellinger and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". In: *The International Journal of Robotics Research* 31.5 (2012), pp. 664–674. DOI: 10.1177/0278364911434236.

[5] Muhammad Maaruf, Magdi Sadek Mahmoud, and Alfian Ma'arif. "A survey of control methods for quadrotor uav". In: *International Journal of Robotics and Control Systems* 2.4 (2022), pp. 652–665. DOI: 10.31763/ijrcs.v2i4.743¯.

[6] Marco Rinaldi, Stefano Primatesta, and Giorgio Guglieri. "A comparative study for control of quadrotor UAVs". In: *Applied Sciences* 13.6 (2023), p. 3464. DOI: 10.3390/app13063464.

[7] Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. "Real-time stabilization and tracking of a four-rotor mini rotorcraft". In: *IEEE Transactions on Control Systems Technology* 12.4 (2005), pp. 510–516. DOI: 10.1109/TCST.2004.825052.

[8] Dengxiu Yu, Shizhuo Ma, Yan-Jun Liu, Zhen Wang, and CL Philip Chen. "Finite-time adaptive fuzzy backstepping control for quadrotor UAV with stochastic disturbance". In: *IEEE Transactions on Automation Science and Engineering* 21.2 (2023), pp. 1335–1345. DOI: 10.1109/TASE.2023.3282661.

[9] Wesam Taha, Ahmed Al-Durra, Rachid Errouissi, and Khaled Al-Wahedi. "Nonlinear disturbance observer-based control for quadrotor UAV". In: (2018), pp. 2589–2595. DOI: 10.1109/IECON.2018.8591597.

[10] Bo Han, Yimin Zhou, Kranthi Kumar Deveerasetty, and Chaofang Hu. "A review of control algorithms for quadrotor". In: (2018), pp. 951–956. DOI: 10.1109/ICInfA.2018.8812437.

[11] Belgis Ainatul Iza, Chairul Imron, and Mardlijah Mardlijah. "Adaptive Trajectory Control for Quadcopter using Extended Kalman Filter-Based Self-Tuning PID under Gaussian Disturbances". In: *Journal of Robotics and Control (JRC)* 6.5 (2025), pp. 2249–2259. https://journal.umy.ac.id/index.php/jrc/article/view/27955.

[12] Weibin Gu, Kimon P Valavanis, Matthew J Rutherford, and Alessandro Rizzo. "UAV model-based flight control with artificial neural networks: A survey". In: *Journal of Intelligent & Robotic Systems* 100.3 (2020), pp. 1469–1491. DOI: 10.1007/s10846-020-01227-8.

[13] Imil Hamda Imran, Rustam Stolkin, and Allahyar Montazeri. "Adaptive control of quadrotor unmanned aerial vehicle with time-varying uncertainties". In: *IEEE Access* 11 (2023), pp. 19710–19724. DOI: 10.1109/ACCESS.2023.3243835.

[14] Yusron Fattahillah Akbar, Belgis Ainatul Iza, et al. "Comparative Analysis of SMC and FSMC on Autonomous Underwater Vehicle (AUV) with Active Ballast". In: *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. IEEE. 2023, pp. 7–12. DOI: 10.1109/IC3INA60834.2023.10285788.

[15] C Ardil. "Type-3 Fuzzy Logic Control for Dynamic Guidance of Uncrewed Stealth Aircraft Under Deep Uncertainty". In: *International Journal of Computer and Information Engineering* 20.4 (2026), pp. 517–522. https://www.researchgate.net/publication/404082605_Type-3_Fuzzy_Logic_Control_for_Dynamic_Guidance_of_Uncrewed_Stealth_Aircraft_Under_Deep_Uncertainty.

[16] Anushka Bhosale and RameshD Jadhav. "Customer Segmentation using K-Means Clustering in Retail Industry". In: *Available at SSRN 6537841* (2026). DOI: 10.2139/ssrn.6537841.

[17] Niteesh Kumar, Anjul Siwch, et al. "A Similarity Measure for Intuitionistic Fuzzy Sets with improved Clustering Accuracy and Applications in Pattern Recognition". In: *Franklin Open* (2026), p. 100518. DOI: 10.1016/j.fraope.2026.100518.

[18] Qiuyi Wu, Zihan Zhu, and Anru R Zhang. "Statistical Inference for Fuzzy Clustering". In: *arXiv preprint arXiv:2601.02656* (2026). DOI: 10.48550/arXiv.2601.02656.

[19] Pooja Sangwan and Rakesh Kumar. "A generalized arbitrary base euclidean space technique for C-Mean clustering simulation". In: *Journal of Ambient Intelligence and Humanized Computing* (2026), pp. 1–29. DOI: 10.1007/s12652-025-05014-x.

[20] Manas Singh Sakrwar, AS Ranadive, and Dragan Pamucar. "A novel Gustafson–Kessel based clustering algorithm using n-Pythagorean fuzzy sets". In: *Systems and Soft Computing* (2025), p. 200345. DOI: 10.1016/j.sasc.2025.200345.

[21] Jinlong Sun, Dong Zhang, Yuhe Chen, Lingzhi Mu, and Guangyu Jia. "Local obstacle avoidance and trajectory tracking control of multi-quadrotor cooperative transportation system". In: *Aerospace Science and Technology* 173 (2026), p. 111670. DOI: 10.1016/j.ast.2026.111670.

[22] Min Duan, Kun Wang, Xiucai Huang, Zhengguo Li, and Jinyu Ni. "Asymptotic attitude tracking control for uncertain quadrotors based on enhanced controllability condition". In: *Journal of Automation and Intelligence* (2026). DOI: 10.1016/j.jai.2026.02.004.

[23] Belgis Ainatul Iza, Chairul Imron, et al. "Trajectory Control of a Delivery Quadcopter Using Self-Tuning Fuzzy-PID and Extended Kalman Filter". In: *Journal of Quality Measurement and Analysis* 22.1 (2026), pp. 141–159. DOI: 10.17576/jqma.2201.2026.08.

[24] R Harisankar, Abhishek Kaushik, and Sishu Shankar Muni. "Path planning for multi-quadrotor 3D boundary surveillance using non-autonomous discrete memristor hyperchaotic system". In: *Franklin Open* 10 (2025), p. 100239. DOI: 10.1016/j.fraope.2025.100239.

[25] Belgis Ainatul Iza, Chairul Imron, and Mardlijah. "Evaluation of robust Kalman-based estimators for quadcopter systems under measurement disturbances". In: *Transactions of the Institute of Measurement and Control* (2026), p. 01423312261424558. DOI: 10.1177/01423312261424558.