International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Multi UAV and Geometrical Shape Trajectory Planning Using PD Controller

[1]Kadali Sri Sai Teja, [2]S. Rooban

Submitted: 02/10/2023 **Revised**: 21/11/2023 **Accepted**: 02/12/2023

Abstract: The field of developing multiple UAV trajectories for surveillance and task orientation presents a captivating area of inquiry. A comprehensive study is made here for developing groups of geometrical trajectories for quadrotor, which are tailored to the unique capabilities of micro unmanned aerial vehicles (mUAVs). The meticulously designed routes are randomly selected to overcome the chance of encounters. A quadrotor's behaviour is derived from Newton's laws and Euler's equations and simulations performed in MATLAB. Multiple trajectories for multiple quadrotors in simulation customised the flight paths for clusters of geometric shapes. A code is developed for the designed routes, which are selected randomly to eliminate collisions. The mUAVs successfully navigated the designated paths and effortlessly revert to the original locations utilizing a PD (Proportional-Derivative) controller. The intricate dynamics of a solitary mUAV are considered to replicate other mUAVs in a proficient manner ensuring the specific optimal distance.

Keywords**:** *micro unmanned aerial vehicle(muav), group trajectories, chance selected, PD controller, MATLAB.*

1. Introduction: -

Quadcopters are highly versatile and agile aerial vehicles that have found quadrotors have numerous applications in a variety of industries, including military operations, surveillance missions, search and rescue operations, industrial applications, and construction projects. They can contribute to ecological studies and wildlife preservation by exploring undiscovered locations. Quadcopters are technological marvels that have changed several industries, advancing our society. These vehicles' propellers rotate, and their wings cut through the air, offering many future possibilities. Dynamic machines drive path planning research to improve navigation. The specification of the trajectory for an initial Unmanned Aerial Vehicle (UAV) is established with precise temporal parameters and is contingent upon the intended launch location of the UAV. Furthermore, the trajectory plan for the entire cohort of UAVs serves as a point of reference for this determination. The achievement of precise tracking control by multi-degree-of-freedom cooperative robots is facilitated through the utilisation of an improved adaptive proportional-integral-derivative feedback control approach[1]. A model predictive control algorithm is used to generate a real-time continuous-time path for a group of robots. To make a path with no collisions, separate safe regions are made in time, and the robots can only move within these safe regions to avoid running into each other [2]. Within a predetermined area, the robots work together while keeping a set distance between one another. The desired configuration of the area can change by choosing suitable cost function. Robots in the group communicate with their close neighbours; they do not need to do so with the rest of the neighbourhood [3]. while coordinating its movement with that of other robots to keep the formation's relative kinematics. The position synchronisation error is used to measure the formation error. This error is defined by the translation of each robot to make sure that both the positional and synchronisation errors converge to zero [4]. A virtual velocity rigid body is used to make the model for planning the paths of multiple UAVs[5]. Collective path is determined by desired path of a leader or virtual robot within group, while shape vectors indicate the relative distances and orientations among the robots. Every single robot plans its own path based on local information from neighbouring ones, such as their planned trajectories and estimated states. Decentralised trajectory planners have been shown to achieve consensus on the planned trajectory for predefined shapes and to ensure safe reconfiguration when transitioning between different shapes[6]. The grouping space framework utilized for the formation flying of three quadrotors. The grouping space framework utilized for the formation flying of three quadrotors. Its dynamics model separates the horizontal and vertical channels. Also, a formation controller employing a fuzzy proportional-derivative (PD) approach is made to make the desired alignment happen [7]. comparable approach has been proposed as a means of achieving Coordination of multiple autonomous underwater vehicles for

[1]*M.tech Student, Department of ECE, Koneru Lakshmaiah Education Foundation, Guntur, India-522302, Email: srisaitejakadali4798@gmail.com*
[2]*Associate Professor, Department of ECE, Koneru Lakshmaiah Education Foundation, Guntur, India-522302, Email id: rooban@kluniversity.in*

manoeuvring and control through the utilisation of the region control concept [8]. The study involves the use of genetic algorithms to improve precision in the application and utilisation of the given task and convergence speed of the PID controller in an autonomous nonholonomic wheeled mobile robot [9]. Six NLPID controllers are designed to correspond to the roll, pitch, yaw, altitude, and position subsystems. A genetic algorithm (GA) is tuned with parameters of the controllers in order to minimise a multi-objective output performance index. Under certain conditions, the Hurwitz stability theorem is also applied to the NLPID controller gains [10]. Lyapunov-like functions are used to encode local decision-making processes among a network of connected agents, allowing them to choose goal assignments based on the shortest total distance to the goals. Even as number of agents increase, the proposed adaptive tactics produce feedback control policies that are computationally efficient and scalable[11]. Pd controller is being considered for altitude stabilisation and there is an significant change after tuning [12]. The controller's design is simplified by selecting visual features based on perspective image moments and projecting them onto a rotated image plane. The flow of these features is used to obtain information about linear velocity, while the controller assumes that feedback about angular velocity and attitude is available [13]. It is possible to increase the size of basin of attraction by optimising gains to achieve smooth response. Quadrotor is taken out of static state and response of the static controller is measured to experimentally characterise this basin [14]. An algorithm that takes an infeasible trajectory designed by a user and returns a feasible trajectory as close as possible to the user's input. Dataset of infeasible quadrotor camera trajectories serves as performance benchmarks. We discovered that our method is anywhere between 25 and 180 times faster than spacetime constraints [15]. The proposed solution includes the first real-time Time-Based One-Time Password (TOTP) mechanism designed specifically for rotorcrafts. It takes into account velocity and nonlinear input constraints. Key concept is to use the differential flatness of the rotorcraft to convert thrust and attitude constraints into inequality constraints. The constraints are formulated using a quadratic nonlinear function that incorporates the acceleration of the path parameter, as well as the squared velocity and acceleration of the path. Backwards and forwards acceleration profiles can be obtained in this manner [16]. Lévy and Gaussian distributions were designed for single-objective optimisation problems and are therefore unsuitable for Multi-Objective Optimisation Problems (MaOPs). The HMOCS algorithm, which uses levy and exponential distributions, has proven to be superior to other methods. This conclusion is based on a thorough comparison and analysis of DTLZ and WFG test suite results, which include 2, 3, 4, 6, 8, and 10 objectives[17]. Waypoint planning improves data transmission reliability by optimising the 3-D position and transmit power of UAVs and user equipment. Furthermore, trajectory planning maximises each UAV's acceleration, allowing it to travel between each waypoint as quickly as possible. Because of this efficient traversal, more time can be allocated to data collection activities [18]. The problem of maximising energy efficiency through recharging scheduling is addressed in order to reduce the total length of UAV trajectories while maintaining quality of service (QoS). A distributed reinforcement learning algorithm is proposed to address this. This algorithm decouples the two games involved in the process effectively [19]. The Gemological Digital Analyser (GDA) is a cutting-edge system for automated opal grading. This grading method employs statistical machine learning techniques and multiple extracted opal image characteristics [20]. The shape vectors represent the robots' relative distances and directions, whereas the group trajectory represents the desired path taken by the leader or virtual robot within group. Each robot plans its trajectory using local information from neighbouring robots, that include planned trajectory of its neighbours as well as an estimate of their current states [21].

Traditional PD and PID controllers have been studied in a number of state-of-the-art research projects, which has made it more important to use their abilities for accurate path following in complex multi-UAV systems.

In section (I) modelling and control of quadcopter. section (II) pd controller for the model. section (III) working structure section (IV) simulation results. Section (V)conclusion and future

2. Modelling and Control of Quadcopter

Modelling of the quadrotor has been describing in [5], [12], [14]

Figure 1 depicts the quadcopter's structure, while Figure 2 depicts the coordinate frames that account for angular velocities and Euler's translation. Figure(a) 1 depicts the quadcopter's structure, while Figure (b)2 depicts the coordinate frames that account for angular velocities and Euler's translation.

The quadrotor's dynamical model

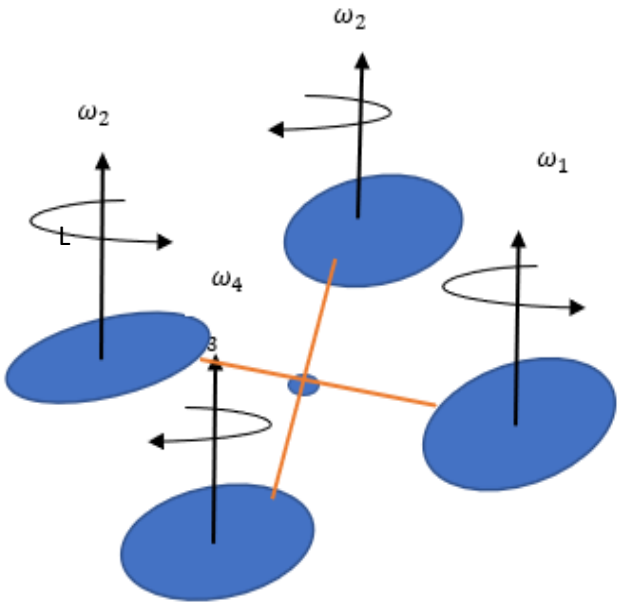


Fig (a) Represents the Body frame of the UAV where ω_i is angular speed and where i= 1,2,3,4 for four rotors.

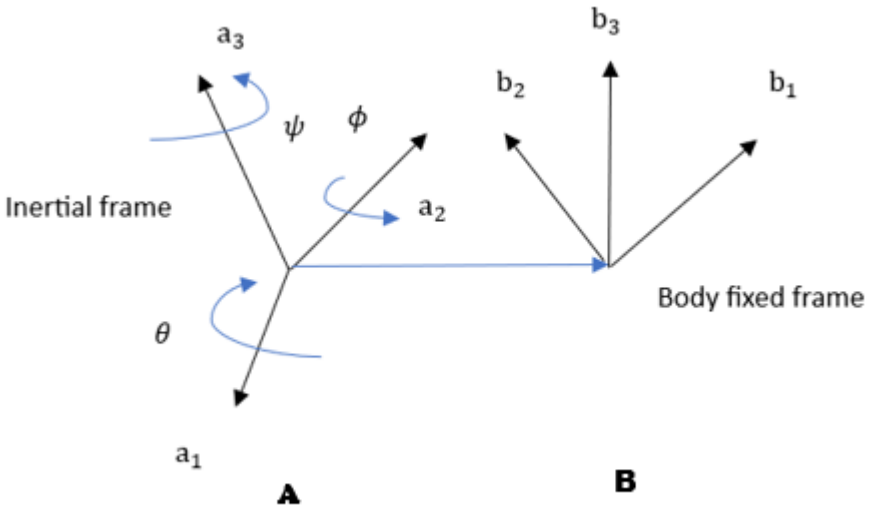


Fig (b) Represent the Euler Z-X-Y transformation of inertial frame to the body-fixed frame. Initially, focusing on the rotational aspect of ψ around axis a_3 then roll by ϕ around a_1 axis and lastly a pitch by θ around the a2 axis.

The quadrotor's coordinate with the free body diagram represents are depicted in the representation Figure (1,2). The frame of inertia, denoted as A, is characterized by the triad a_1 ,a_2, and a_3 by means of a_3 upward facing. Body fixed frame B. Indicates the location of quadrotor centre of gravity is, where b_1 aligns along desired forwards motion. b_3 is aligned perpendicular to the rotor plane, pointing directly upward during ideal hover shown in figure (b). These lines are perpendicular to the main axes.so from Euler angles we can compute the matrix of rotation (R) that transforms points on a map A to B is provided as follows

$$R = \begin{bmatrix} C\psi C\theta - S\phi S\psi S\theta & -C\phi S\psi & C\psi S\theta + C\theta S\phi S\psi \\ C\theta s\psi + C\psi S\phi S\theta & C\psi C\theta & C\theta S\psi - C\psi C\,\theta S\phi \\ -C\phi S\theta & S\phi & C\phi C\theta \end{bmatrix}$$

Here cos(θ) as Cθ and sin(θ) as Sθ and similar with ϕ and ψ.

MOTOR MODEL

Each rotor is characterized by its angular speed $w_i$and the rotor generates an upward push F_i consistent with

$$F_i = k_F w_i^2 \quad (2)$$

In accordance with, the rotors produce a moment when

$$M_i = k_m w_i^2 \quad (3)$$

The equations that control how fast the quadrotor's centre of mass moves is given by

$$\ddot{M}\boldsymbol{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathrm{R}\begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \quad (4)$$

Angular velocity of quadrotor is represented as s, t, and u and are associated with derivatives of ϕ, θ, and ψ angles respectively and given as

$$\begin{bmatrix} s \\ t \\ r \end{bmatrix} = \begin{bmatrix} \mathrm{C}\theta & O & -\mathrm{C}\phi\mathrm{S}\theta \\ 0 & 1 & \mathrm{S}\phi \\ \mathrm{S}\theta & 0 & \mathrm{C}\phi\mathrm{C}\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (5)$$

In Euler's equations of motion, it is observed that each rotor produces not only forces but also a moment that is orthogonal oriented orthogonal to plane of rotation of blades M_i. Propellers in diagonal are motioned at same $-b_3$ and $+b_3$ way. Since quadrotor's moment of inertia is in the opposite direction of how the blades turn, M_1 and M_3 act in a b_3 direction, even as M_2 and M_4 act towards $-b_3$ way. L represents length between the propellers axis of rotation and quadrotor's centre of mass.

Here the motor gain was tuned with respective to real micro quadrotor in simulation and angular velocities within the limits.

ROBOT CONTROLLERS

The derivation of controllers involves the linearization of motion and motor model equations that are mentioned above are kept in an operating point that results correspond to the standard hovering condition, and the roll, pitch angles are negligible and considered as zero.

At static state in air lift force from the motor blades, where roll, pitch angles are zero and yaw angle is constant

$$F = \frac{mg}{4} \quad (6)$$

F_1, F_2, F_3, F_4 are lift forces for four propellers. ,

As governed by the Euler equations, the angular acceleration given by

$$I\begin{bmatrix} \dot{s} \\ \dot{t} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} s \\ t \\ u \end{bmatrix} \times I\begin{bmatrix} s \\ t \\ u \end{bmatrix} \quad (7)$$

The standard regular values for the inputs during at still in air is equal to weight of the quadrotor

Converting equation (4) to a linear form we get

$$\ddot{r}_1 = g(\Delta\theta\cos\psi_0 + \Delta\phi\sin\psi_0) \quad (8\ a)$$

$$\ddot{r}_2 = \mathrm{g}\,(\Delta\,\theta\sin\psi_0 - \Delta\,\phi\cos\psi_0) \quad (8\ b)$$

$$\ddot{r}_3 = \frac{1}{m}u_1 - g$$

Converting equation (7) to a linear form we get

$$\begin{bmatrix} \dot{s} \\ \dot{t} \\ \dot{u} \end{bmatrix} = I^{-1}\begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix}\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \quad (9)$$

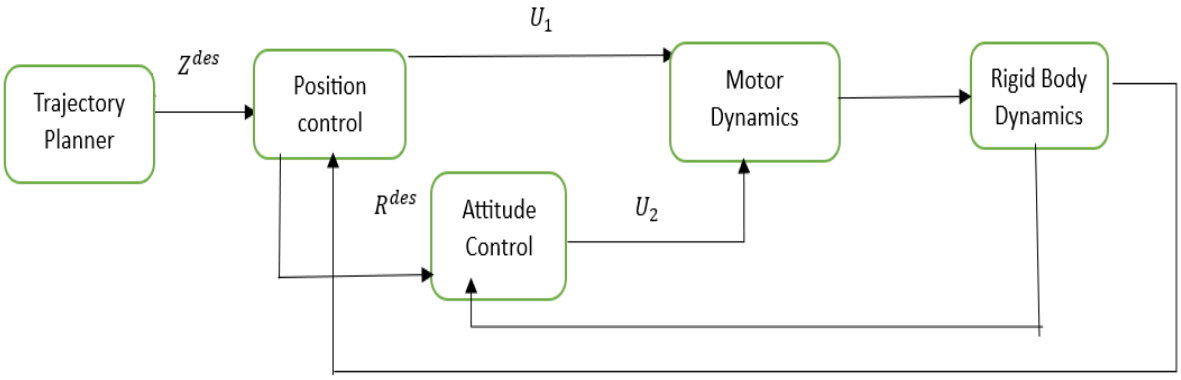


Fig (3) The hierarchical control approach is a fundamental flow that is widely utilised in quadrotors.

Assuming the rotorcraft is symmetric so $I_{xx} = I_{yy}$we get:

$$\dot{s} = \frac{U_{2,x}}{I_{xx}} = \frac{L}{I_{xx}} = (F_2 - F_4) \quad (10)$$

$$\dot{t} = = (-= \frac{U_{2,y}}{I_{yy}} = \frac{L}{I_{yy}} (F_3 - F_1) \quad (11)$$

$$\dot{u} = \frac{u_{2,z}}{I_{zz}} = \frac{\gamma}{I_{zz}} (F_1 - F_2 + F_3 - F_4) \quad (12)$$

To put it another way, separated from one another the equations of motion for angular accelerations. The only thing that determines which component of angular acceleration one has is which constitute of U_2.

3. PD Controller Design

Regulation of position and attitude

Determining the four inputs that are required for control is the challenge that must be solved $\{u_1, u_2\}$ the need for manoeuvres such as hovering or following a predetermined path as Z^{des}. As shown in Figure 3, we will use the positional mistakes produced by the robot to direct the operation of a position controller that is derived from equation (8). This position controller makes the decisive call directly u_1. The model presented in (8) also permits the derivation of a desired orientation. The model presented in equation (9), which describes this orientation, is used to generate attitude regulator for this orientation. The transformation of the inputs into the output. $\{u_1, u_2\}$ was brought in procedure for the conversion is simple and may be found described in reference [14]. The roll, pitch, and yaw are brought under control by the inner attitude control loop, which is comprised of onboard accelerometers and gyroscopes.

Attitude Control

Now, we are going to implement a combination proportional and derivative (PD) attitude controller (PD controller). This controller is going to be built to follow a trajectory in SO (3) that this controller is going to be defined by a desired ϕ, θ, ψ angles. Given that the controller is developed because the creation of our controller is based on linearized motion equations, it is crucial to observe that the attitude should closely resemble the close approximation to minimal static state. θ and ψ angles are minimised to at their tiniest possible values in this state. In the nominal hover state PD control law assume the following control law structure. The control law structure is adopted by proposed pd controller laws when the nominal hover state is close by.:

$$U_2 = I \begin{bmatrix} k_{p,\phi}(\phi^{des} - \phi) + & k_{d,\phi}(s^{des} - s) \\ k_{p,\theta}(\theta^{des} - \theta) + & k_{d,\theta}(t^{des} - t) \\ k_{p,\psi}(\psi^{des} - \psi) + & k_{d,\psi}(u^{des} - u) \end{bmatrix} \quad (13)$$

Here the $k_{p,\phi}$, $k_{p,\theta}$, $k_{p,\psi}$, $k_{d,\phi}$, $k_{d,\theta}$, $k_{d,\psi}$ are the proportional and derivative gains with respective to roll and pitch angles.

Position control

In the following section, we will investigate and discuss two pose control strategies methods that θ and ψ angles are taken as inputs for controlling and regulating the position of the quadrotor. These angles play a crucial role in determining the quadrotor's roll and pitch. The position control algorithm takes responsibility for computing the necessary roll adjustments and pitch angles in each of these approaches $\theta^{des}$and ψ^{des} using equation (13), one is able to determine the appropriate velocities by employing these angles as inputs. Station-keeping, also known as holding the position at a certain position vector, may be accomplished with the help of the first technique, which is known as the hover controller, r_0. The second approach follows a path that has been predetermined r_T(t), in three dimensions. Both of these situations need the desired yaw angle to be provided in their own requirement. It is possible for it to have a fixed value, or it may be the result of a distinct control method ψ_0 with a time varying quantity, ψ_T(t). The intended trajectory:

$$z^{des} = \begin{bmatrix} r_{T(t)} \\ \psi_T(t) \end{bmatrix} \quad (14)$$

Here z^{des} represents the desired trajectory the quadrotor need to follow

As input, the trajectory coordinator receives the intended trajectory, which specifies the position vector and yaw angle that must be monitored.

For maintaining the quadrotor idle in air, the subsequent factors need to be considered to establish control when hovering r_T (t) and yaw angle will be constant. The command accelerations $\ddot{r}_i^{des}$are determined using a PD controller. Using the standard reference triad, you can say the following about the position error in terms of the position of the components:

$$e_i = (r_{i,T} - r_i) \quad (15)$$

To ensure the guarantee that this is accomplished with certainty inaccuracy error will eventually reach zero via an exponential descent, we consider

$$(\ddot{r}_{i,T} - \ddot{r}_i^{des}) + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i) + k_{p,i}(r_{i,T} - r_i) = 0 \quad \text{-(16)}$$

Here at hover situation desired position are considered

as zero.

At the sudden tuns of trajectory the drones are controlled by using hover controller

By establishing a correlation among intended acceleration and θ and ψ angles from equation (8), it can effectively attain their desired outcome. Knowing that change in roll and pitch is the same and we can write it as will allow you to achieve your goals.

$$\ddot{r}_1^{des} = g(\theta^{des}\cos\psi_T + \phi^{des}\phi\sin\psi_T) \quad (17a)$$

$$\ddot{r}_2^{des} = g(\theta^{des}\sin\psi_T - \phi^{des}\phi\cos\psi_T) \quad (17b)$$

$$\ddot{r}_3^{des} = \frac{1}{m}u_1 - g \quad (17c)$$

For the attitude controller calculating the optimal roll and pitch angles may be done with the (17 a, b) remaining two equations:

$$\phi^{des} = \frac{1}{g}(\ddot{r}_1^{des}\sin\psi_T - \ddot{r}_1^{des}\cos\psi_T) - \quad (18\ a)$$

$$\theta^{des} = \frac{1}{g}(\ddot{r}_1^{des}\cos\psi_T + \ddot{r}_1^{des}\sin\psi_T) - \quad (18\ b)$$

Since the yaw, ψ_T(t) the trajectory generator is responsible for autonomously prescribing the yaw angle, which results in:

$$\psi^{des} = = \psi_T(t) \quad (19a)$$

$$r^{des} = \dot{\psi}_T(t) \quad (19b)$$

The above attitude controller's setpoints are provided by equations in equation (18) .

The 3-D Trajectory Controller is employed in this instance to track three-dimensional trajectories that demonstrate moderate accelerations and comply with the near-hovering assumptions. To derive this controller, similarly the similar stages are repeated as in equation (16) leaving that $\dot{r}_{i,T}$ and $\ddot{r}_{i,T}$ as they are nonzero but obtained from the properties of the trajectory. Hover assumption holds and linear the dynamics are linear with no input saturation on the inputs, a controller can be designed to produce the desired acceleration $\ddot{r}_i^{des}$ according to equation (16) which ensures that the error will converge exponentially to zero.

Let r_t be planned trajectory point that is geographically closer to the present place and is denoted by the letter r. After differentiating the trajectory that was given to us, we were able to determine the required desired velocity and acceleration, which are represented by $\dot{r}_T$ and $\ddot{r}_T$. Tangent vector of the trajectory represented as $\hat{t}$. Orthogonal vector that is perpendicular to the trajectory, $\hat{n}$, the unit binormal vector can be obtained through the cross product of derivation of unit tangent vector is obtained by deriving $\hat{t}$ with time. Additionally, the direction of a curve at a particular location is determined through the product

$$\hat{b} = \hat{n}, \text{x } \hat{t} \quad (20)$$

Position and velocity errors calculated from:

$$e_p = ((r_T - r).\hat{n})\hat{n} + ((r_T - r).\hat{b})\hat{b} \quad (21a)$$

$$e_v = \dot{r}_T - \dot{r} \quad (22b)$$

It should be noted that in this context, the positional error in the tangential direction is disregarded. Instead, the focus is solely on the positional error of the plane which is perpendicular to at its nearest point. Again, we compute instructed acceleration $\ddot{r}_i^{des}$. Once more, we compute the acceleration that has been instructed $\ddot{r}_i^{des}$, based on the PD feedback, as illustrated in (16). Expressed in, as illustrated in (16) in vector notation, this is:

$$(\ddot{r}_T - r^{\ddot{des}}) + k_d e_v + k_p e_p = 0 \quad (23)$$

For desired roll and pitch values the equations (18 – 19 (a)(b)) are consider and in detail explanation was given in [14].

4. Working Structure

Here the motor gain was tuned with respective to reality micro quadrotor in simulation and angular velocities within the limits. The above explanation were for the single quadrotor and remaining quadrotor mimic the model for multiple muav's and are being controlled.

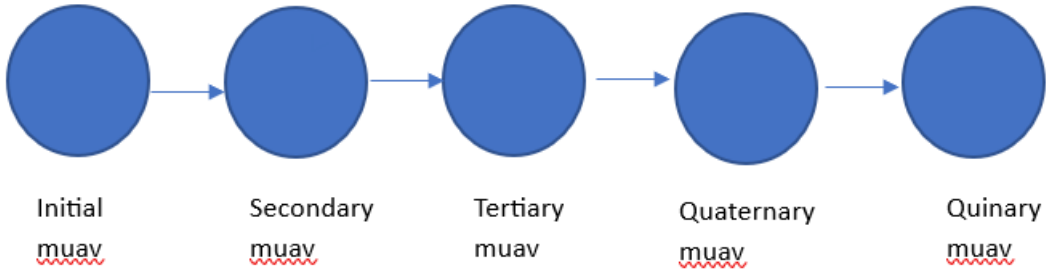


Fig (4) explains the main rule followed by the mUAV's (blue circle represents mUAV)

Every subsequent Unmanned Aerial Vehicle (UAV) in this work imitates the characteristics of the preceding UAV, resulting in a series of imitated characteristics. While the flight paths UAVs are like differ, the launch of each miniature UAV varies during simulation, allowing for the incorporation of randomness into their individual trajectories. In contrast to the leader-following theory, the position and dynamics of the initial quadrotor are considered here.

The total time given for iterations is 28 seconds and the number of iterations done are 560.

The velocity profile depends on the number of quadrotors.

5. Simulation

Here we have considered five uav's which follow unique geometrical trajectories without collision and for each simulation the uav's are randomly opt the trajectories with in. The current simulation is performed in DELL G15 5511 model with 16 GB RAM,11th Gen Intel(R) Core (TM) i7-11800H @ 2.30GHz 2.30 GHz.

Blue colour is coded to desired trajectory and red colour for actual trajectory drone followed.

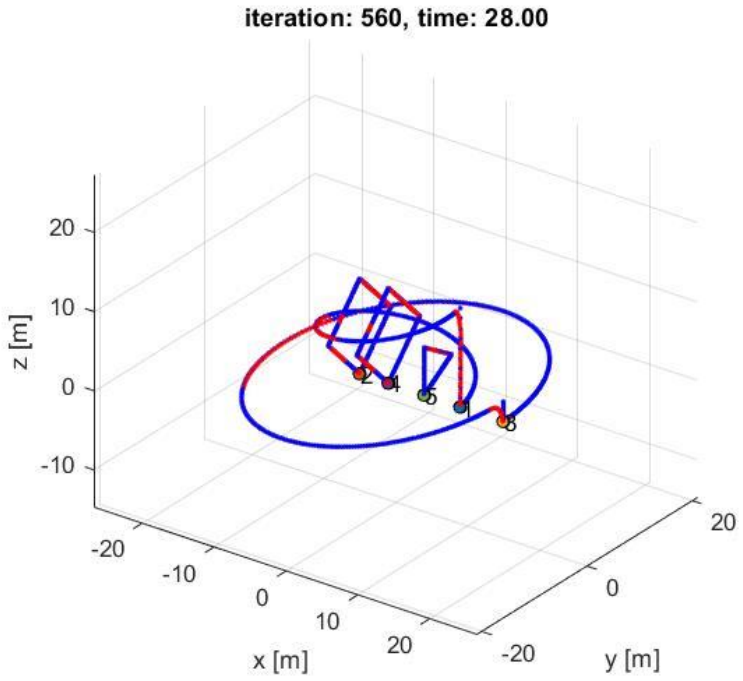


Fig (5(a)) mUAV's are in follow up of 2, 4, 5, 1, 3.

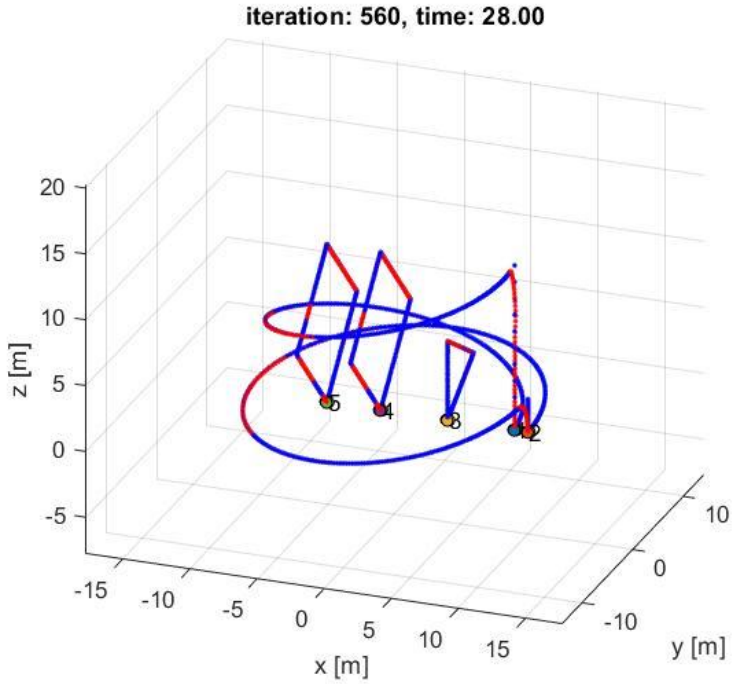


Fig (5(b)) mUAV's are in follow up of 5, 4, 3, 2.

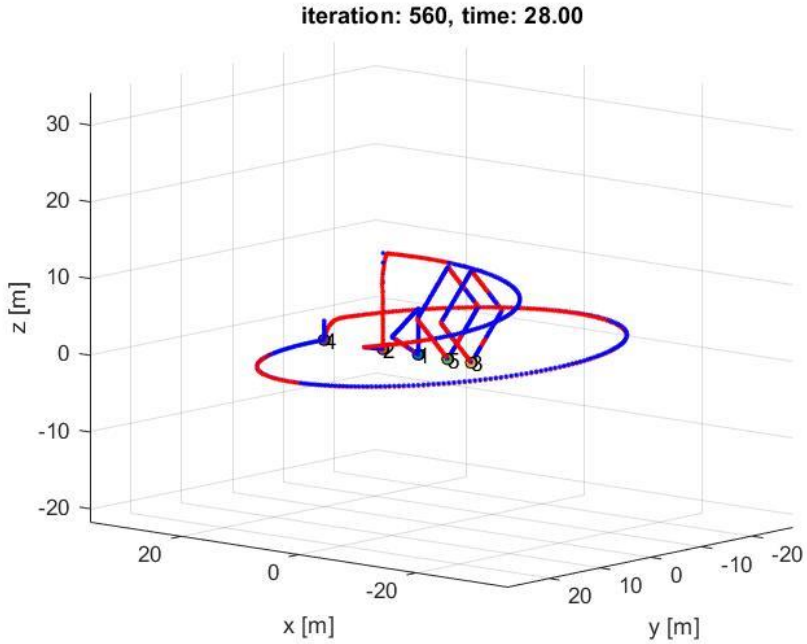
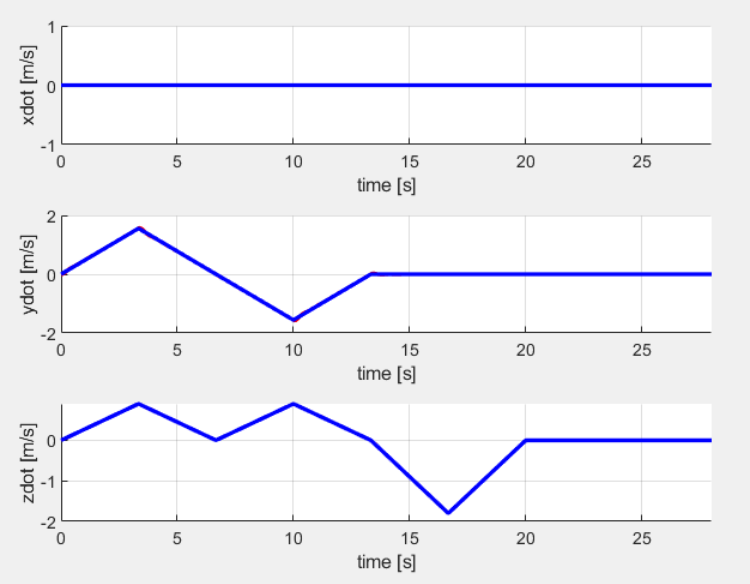


Fig (5(c)) mUAV's are in follow up of 4, 2, 1, 5, 3.

Fig (5(a), 5(b), 5(c)) The group of micro uav's following the desired geometrical trajectories without colliding and getting back to their initial positions. Here the uav's are formulated to change their initial positions.

As the position and velocities of individual micro uav's



Fig(6) Drone 1 following the triangular trajectory and it's velocity profile**.**

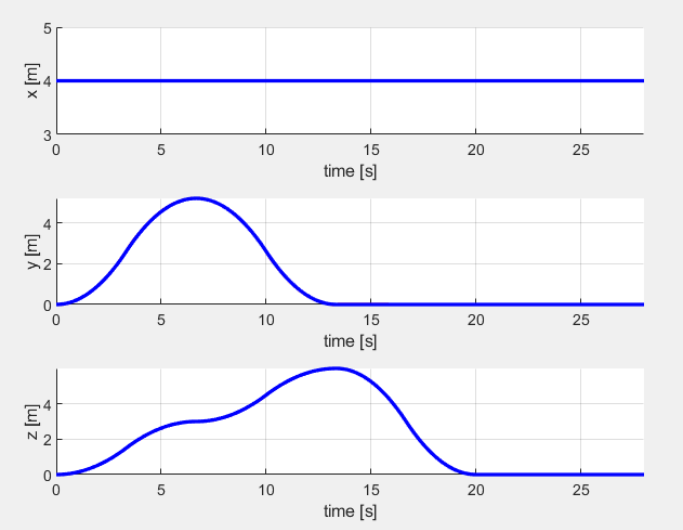
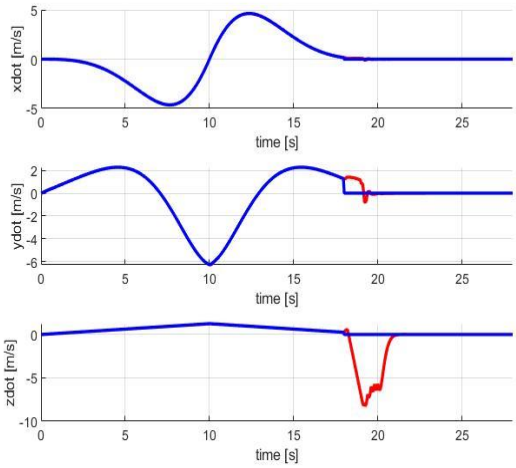
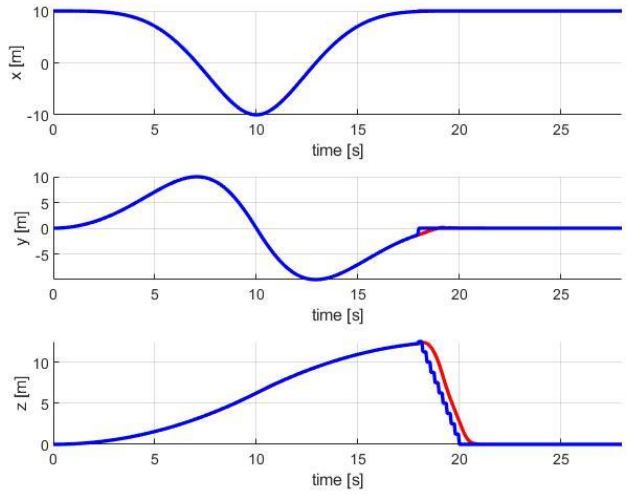


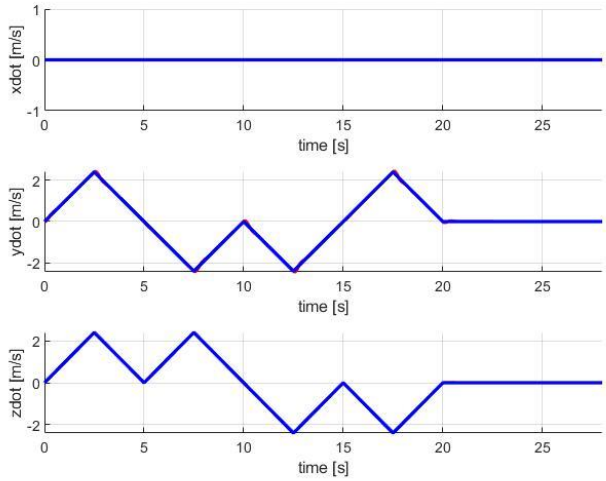
Fig (7) Drone 1 following the triangular trajectory and its position profile.



Fig(8) Drone 2 following the circular trajectory and it's velocity profile where, and here at the y axis and the z axis profile there is a change in a bit of scale which signals the muav lowered its velocity for the quick shift in trajectory which you can see in the position profile.



Fig(9) Drone 2 following the circular trajectory and it's position profile where the drone has gradually dropping further and further away from its position to minimize the error from previous location.



Fig(10) Drone 3 following Rhombus trajectory and it's velocity profile.

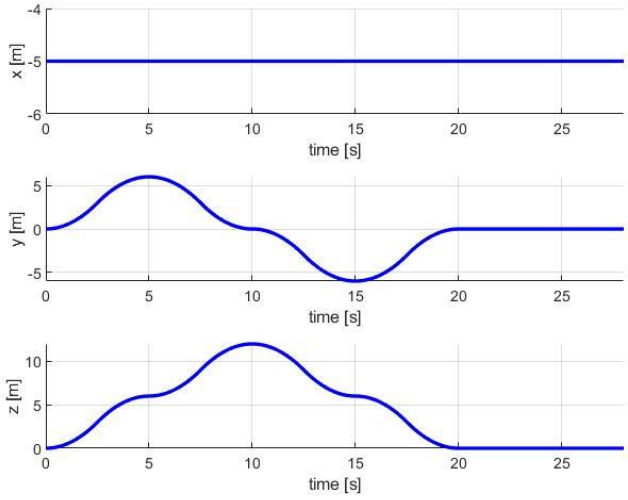
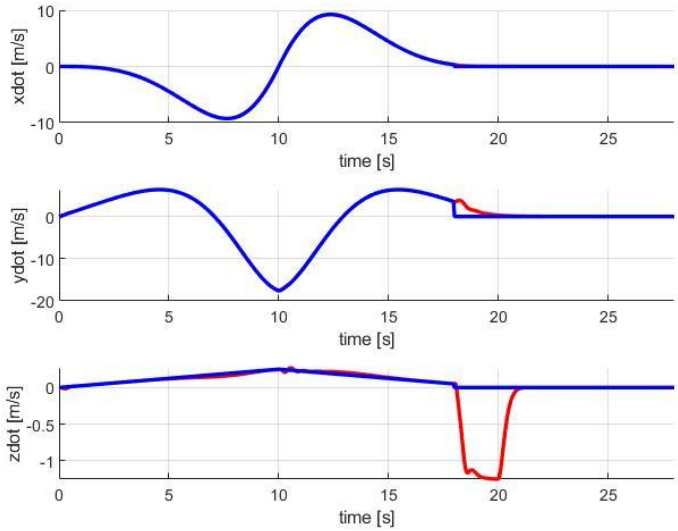
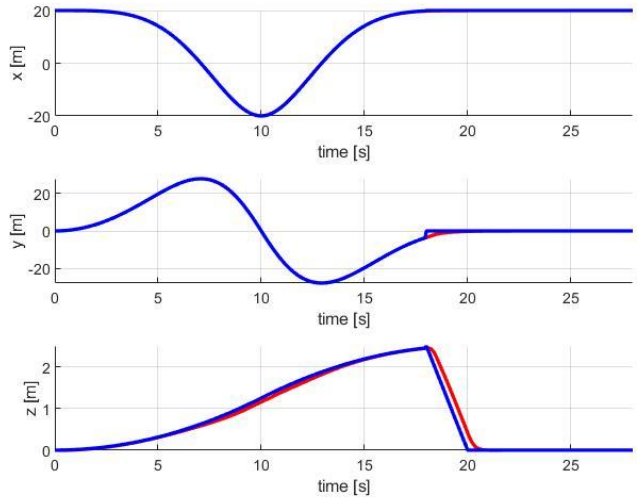


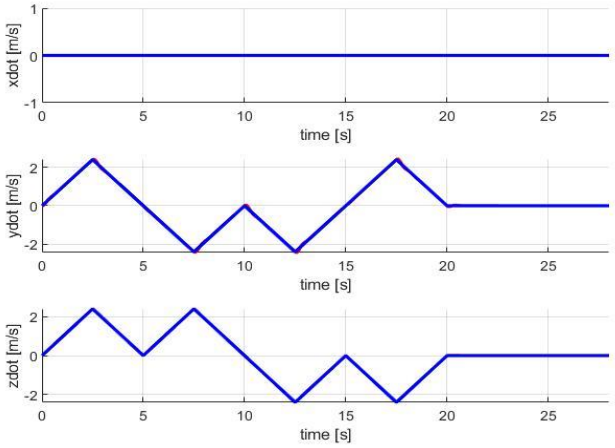
Fig (11) Drone3 following Rhombus trajectory and it's position profile.



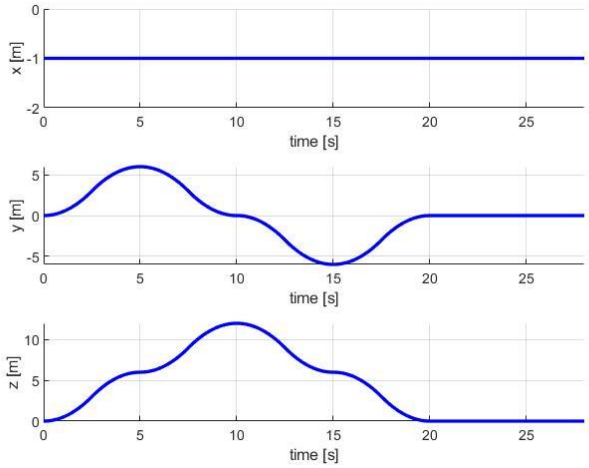
Fig(12) Drone 4 following the Ellipse trajectory and it's velocity profile where there is about sudden variation in velocity profile at Z axis.



Fig(13) Drone 4 following the ellipse trajectory and it's position profile



Fig(14) Drone 5 following trajectory and it's velocity profile.



Fig(15) Drone 5 following rhombus trajectory and it's position profile.

6. Conclusion: -

A systematic development process for multiple mUAV's and trajectories is summarised here. Multiple trajectories for multiple quadrotors in simulation customised the flight paths for clusters of geometric shapes. These clusters of geometrical shapes are optimised for micro unmanned aerial vehicles. The routes are designed and selected at random for eliminating chance encounters (if any). The multiple mUAV's geometrical trajectory planning made possible using the developed code of collision avoidance for micro quadrotors. The mUAVs followed the predetermined trajectories and seamlessly retrace the initial positions by implementing a PD controller. The controller exhibits remarkable agility in responding to changing conditions and effectively managing the distances between micro-UAVs to ensure safety and efficiency.

References: -

[1] Zaidi et al., "Adaptive Active Disturbance Rejection Control for Rendezvous of a Swarm of Drones," IEEE Access, vol. 10, pp. 90355–90368, 2022, doi: 10.1109/ACCESS.2022.3201845.

[2] S. Kandhasamy, V. B. Kuppusamy, and S. Krishnan, "Scalable decentralized multi-robot trajectory optimization in continuous-time," IEEE Access, vol. 8, pp. 173308–173322, 2020, doi: 10.1109/ACCESS.2020.3024685.

[3] C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," Automatica, vol. 45, no. 10, pp. 2406–2411, Oct. 2009, doi: 10.1016/j.automatica.2009.06.026.

[4] Sun, C. Wang, W. Shang, and G. Feng, "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations," IEEE Transactions on Robotics, vol. 25, no. 5, pp. 1074–1086, 2009, doi: 10.1109/TRO.2009.2027384.

[5] Y. B. Chen, J. Q. Yu, X. L. Su, and G. C. Luo, "Path Planning for Multi-UAV Formation," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 77, no. 1, pp. 229–246, Jan. 2015, doi: 10.1007/s10846-014-0077-y.

[6] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," Auton Robots, vol. 33, no. 1–2, pp. 143–156, Aug. 2012, doi: 10.1007/s10514-012-9279-y.

[7] Wang, L. Bischof, R. Lagerstrom, V. Hilsenstein, A. Hornabrook, and G. Hornabrook, "Automated Opal Grading by Imaging and Statistical Learning," IEEE Trans Syst Man Cybern Syst, vol. 46, no. 2, pp. 185–201, Feb. 2016, doi: 10.1109/TSMC.2015.2427776.

[8] 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics : Suntec Convention and Exhibition Center, Singapore, July 14-17, 2009. IEEE, 2009.

[9] R. R. Slavescu, Universitatea Tehnică din Cluj-Napoca. Computer Science Department, IEEE Romania Section, and Institute of Electrical and Electronics Engineers, Proceedings, 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP) : Cluj-Napoca, Romania, September 6-8, 2018.

[10] A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," Engineering Science and Technology, an International Journal, vol. 22, no. 4, pp. 1087–1097, Aug. 2019, doi: 10.1016/j.jestch.2019.02.005.

[11] Panagou, M. Turpin, and V. Kumar, "Decentralized Goal Assignment and Safe Trajectory Generation in Multirobot Networks via Multiple Lyapunov Functions," IEEE Trans Automat Contr, vol. 65, no. 8, pp. 3365–3380, Aug. 2020, doi: 10.1109/TAC.2019.2946333.

[12] SSN College of Engineering, Institute of Electrical and Electronics Engineers. Madras Section., and Institute of Electrical and Electronics Engineers, Proceedings of the 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC) : 21-23 August 2019, SSN College of Engineering, Kalavakkam, Chennai, India.

[13] H. J. Asl and J. Yoon, "Bounded-Input Control of the Quadrotor Unmanned Aerial Vehicle: A Vision-Based Approach," Asian J Control, vol. 19, no. 3, pp. 840–855, May 2017, doi: 10.1002/asjc.1420.

[14] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," IEEE Robot Autom Mag, vol. 17, no. 3, pp. 56–65, Sep. 2010, doi: 10.1109/MRA.2010.937855.

[15] M. Roberts and P. Hanrahan, "Generating dynamically feasible trajectories for quadrotor cameras," in ACM Transactions on Graphics, Association for Computing Machinery, Jul. 2016. doi: 10.1145/2897824.2925980.

[16] X. Zhang, Y. Fang, X. Zhang, P. Shen, J. Jiang, and X. Chen, "Attitude-Constrained Time-Optimal Trajectory Planning for Rotorcrafts: Theory and Application to Visual Servoing," IEEE/ASME Transactions on Mechatronics, vol. 25, no. 4, pp. 1912–1921, Aug. 2020, doi: 10.1109/TMECH.2020.2993617.

[17] Z. Cui, M. Zhang, H. Wang, X. Cai, W. Zhang, and J. Chen, "Hybrid many-objective cuckoo search algorithm with Lévy and exponential distributions," Memet Comput, vol. 12, no. 3, pp. 251–265, Sep. 2020, doi: 10.1007/s12293-020-00308-3.

[18] Z. Huang, C. Chen, and M. Pan, "Multiobjective UAV Path Planning for Emergency Information Collection and Transmission," IEEE Internet Things J, vol. 7, no. 8, pp. 6993–7009, Aug. 2020, doi: 10.1109/JIOT.2020.2979521.

[19] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, and J. Li, "Multi-UAV Trajectory Planning for Energy-Efficient Content Coverage: A Decentralized Learning-Based Approach," IEEE Journal on Selected Areas in Communications, vol. 39, no. 10, pp. 3193–3207, Oct. 2021, doi: 10.1109/JSAC.2021.3088669.

[20] D. Wang, L. Bischof, R. Lagerstrom, V. Hilsenstein, A. Hornabrook, and G. Hornabrook, "Automated Opal Grading by Imaging and Statistical Learning," IEEE Trans Syst Man Cybern Syst, vol. 46, no. 2, pp. 185–201, Feb. 2016, doi: 10.1109/TSMC.2015.2427776.

[21] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," Auton Robots, vol. 33, no. 1–2, pp. 143–156, Aug. 2012, doi: 10.1007/s10514-012-9279-y.