2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Inter-Agent Collision Avoidance for Decentralized 3D Fixed-Wing Flocking

Archit Gupta, Ajay Tak, Shashi Ranjan Kumar, and Leena Vachhani

Abstract— **UAV swarms have gained prominence due to their reliability, agility, and cooperative capabilities. This paper addresses the challenges of devising algorithms for fixed-wing UAVs posed by non-holonomic kinematics and actuator constraints. While existing works explored optimal control, artificial potential fields, and biologically inspired methods, these often overlook 3-dimensional non-holonomic constraints and actuator saturation. This work contributes to a decentralized 3D fixed-wing flocking algorithm that ensures inter-agent collision avoidance and effective flock formation. Moreover, using a leader-follower model, the agents track a virtual leader while communicating only with neighbors, keeping the motion constraints of fixed-wings. We establish safe initial conditions to guarantee collision avoidance. Numerical simulations validate the proposed algorithm's robustness, demonstrating its effectiveness in solving the flocking problem for non-holonomic fixed-wing agents and advancing UAV swarm coordination.**

Decentralized Algorithm, Fixed-Wing UAV, Collision Avoidance, Non-holonomic Constraints.

I. Introduction

Rapid advancements in autonomous technology and unmanned aerial vehicles (UAVs) have garnered significant attention due to their reliability, agility, and user-friendly operation. These UAVs have found applications across diverse fields, from autonomous quadrotors to fixed-wing UAVs. Although quadrotors and fixed-wing UAVs have distinct advantages and disadvantages, they are noteworthy for their superior endurance and extended-range capability. This endurance is achieved through aerodynamic efficiency, which allows fixed-wing UAVs to conserve energy better than their multi-rotor counterparts. Their ability to cover vast distances and remain airborne for extended periods makes them especially valuable for applications requiring sustained flight, such as environmental monitoring, surveillance, and long-distance delivery.

This shift underscores the growing recognition of the unique capabilities and applications of a swarm of fixed-wing UAVs in the evolving landscape of autonomous aerial technology. That said, the swarm formation of UAVs can be exceptionally beneficial for many tasks due to enhanced coordination and mutual support. Extensive research has been dedicated to developing sophisticated flocking swarm algorithms for UAVs. A decentralized flocking algorithm [1] has been designed specifically for quadrotor UAVs. Considering that fixed-wing UAVs have distinct advantages over quadrotors, developing practical flocking algorithms for swarms of fixed-wing UAVs remains a significant challenge. The fixed-wing UAVs present their own set of challenges. Their non-holonomic kinematics complicate the analysis and implementation of flocking algorithms for fixed-wing UAVs, making the process tedious and cumbersome. Unlike holonomic systems, where movement is unrestricted in multiple directions, fixed-wing UAVs are constrained by their forward motion and turning capabilities, which must be meticulously managed. Additionally, for these algorithms to be practical, it is crucial to account for actuator saturation in the control inputs to the agents. The physical limitations of the UAVs' actuators, which can only exert a finite amount of force or torque, must be integrated into the algorithm design. Ignoring these constraints could lead to unrealistic or unachievable control commands, reducing the effectiveness and reliability of the swarm's coordination and performance. Considerable effort is required to develop algorithms that can navigate these complexities and optimize the performance of fixed-wing UAV swarms.

The decentralized flocking algorithms for fixed-wing agents [2]–[12] typically adhere to two primary methodologies. One approach involves artificial potential-based algorithms, where virtual forces influence the movement and formation of the UAVs. These potential fields guide the UAVs toward desired positions while avoiding collisions and maintaining formation cohesion. The other approach frames the swarm control problem as an optimal control problem. This methodology aims to derive a control strategy that optimizes a specific performance criterion, such as minimizing energy consumption or maximizing coverage. The optimal solution provides a systematic control strategy for achieving flocking behavior. Both methodologies aim to address the inherent challenges of fixed-wing dynamics, seeking to enhance the coordination, efficiency, and robustness of UAV swarms in complex operational environments [2]–[4], [6]–[8].

The optimal control [2] formulates formation control, trajectory tracking, and obstacle/collision avoidance into a unified optimal control framework. Similar methodologies have been adopted in other significant studies [3]–[5]. However, these works predominantly considered 2D non-holonomic agents. For the 3D, the focus has been often limited to formation or reconfiguration control algorithms rather than comprehensive flocking algorithms. Additionally, none of these studies adequately accounted for actuator saturation.

Many other works have been based on artificial potential fields, with several drawing inspiration from Olfati-Saber's

A. Gupta is with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India. (e-mail: archit.gupta1006@gmail.com).

A. Tak and S. R. Kumar are with the Department of Aerospace Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India. (e-mails: ajytak@gmail.com, srk@aero.iitb.ac.in).

L. Vachhani is with the System & Control Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India. (e-mail: leena.vachhani@iitb.ac.in).

formulation [1]. For instance, an optimized potential field-based algorithm [6] inspired by Olfati-Saber's approach has been specifically tailored for collaborative trajectory planning and collision avoidance. Similar algorithms [7]–[9] have been designed for the 2D scenarios but ignoring actuator saturation. The 3D non-holonomic kinematics [7] has been considered but the algorithm assumed a constant velocity magnitude for each agent, rendering it a formation control algorithm rather than an accurate flocking. In particular, the group of agents forms formation and does not have a dynamic group objective. Furthermore, there have been biologically inspired approaches [10], which draw inspiration from pigeon flock formation. A hybrid approach that merges biologically inspired techniques with potential field-based methods has also been presented in [11]. However, biologically inspired approaches do not consider 3D kinematics, and actuator saturation is often neglected.

It is imperative to address the fixed-wing flocking problem to enhance practicality and scalability by fully incorporating the 3D non-holonomic structure and actuator saturation. A small case study on fixed-wing UAVs [12] explores the trade-off between communication range and turning radius. The objective of this work is to develop a 3D algorithm for fixed-wing UAVs considering actuator saturation. The proposed algorithm with the mentioned objective is decentralized providing safety guarantees with specific safe initial conditions, ensuring robust and reliable performance in real-world applications. This approach addresses the current limitations and paves the way for more efficient and effective coordination of fixed-wing UAV swarms.

The rest of this paper is organized as follows. Section II presents a formulation of the problem and preliminary information and theories that aid the algorithm's development. The proposed decentralized algorithm is detailed in Section III. Simulation results and analysis are presented in Section IV, along with concluding remarks in Section V.

II. Preliminaries and Problem Formulation

This section presents some preliminaries and prior works that have inspired the development of our proposed 3D flocking algorithm. Before delving into the relevant theoretical background, we introduce the kinematic model and the problem statement for the fixed-wing flocking agents. This foundational model serves as the basis for our algorithm, incorporating essential aspects of fixed-wing kinematics and constraints to ensure realistic and practical flocking behavior.

A. Problem formulation

The system representation of the fixed-wing UAV uses kinematic model taken from the work in [14]. The state variables include the position $\mathbf{p} = [x, y, z]^T$ in the inertial reference frame, the speed in the air V, and the orientation represented by Euler angles $[\phi, \theta, \psi]^T$, where ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle, depicted in Fig 1. For the fixed-wing UAV operations, the pitch and roll angles are small and can be decoupled. The control inputs used to govern the UAV's motion are the acceleration a,

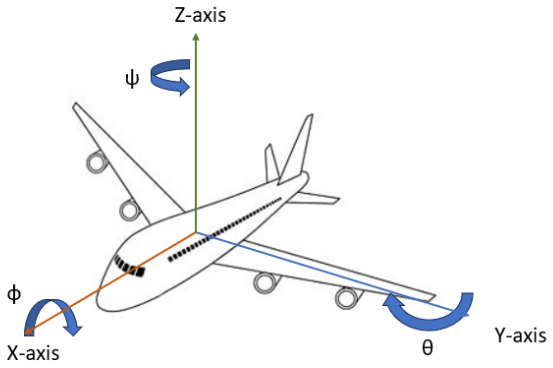


Fig. 1. Euler angles are shown in the Inertial frame of the aircraft for (roll, pitch, yaw)

which affects the magnitude of the velocity along the current direction of motion, the roll ϕ, and the pitch θ. Additionally, the yaw rate $\dot{\psi}$ is derived from the roll angle ϕ and the airspeed, ensuring coordinated turns through the banking. The system description for a fixed-wing UAV agent is given by

$$\dot{x} = V\cos\theta\cos\psi, \quad \dot{y} = V\cos\theta\sin\psi \tag{1a}$$

$$\dot{z} = V\sin\theta, \qquad \dot{V} = a \tag{1b}$$

$$\dot{\psi} = \frac{g}{V}\tan\phi, \tag{1c}$$

Next, we also have the actuator constraints on each of these control inputs as follows:

$$V_{\text{stall}} \le V \le V_{\max}, \quad a_{\min} \le a \le a_{\max} \tag{2a}$$

$$|\phi| \le \phi_{\max}, \qquad |\theta| \le \theta_{\max} \tag{2b}$$

The problem is to find the optimal control input (a, ϕ, θ) for each fixed-wing UAV agent that satisfies the given constraints, forms a flock without inter-agent collisions, and ensures the flock tracks the trajectory of a virtual leader. In the following subsections, we will explain the various concepts that have influenced or been used in developing our flocking algorithm.

B. Dubin's Path

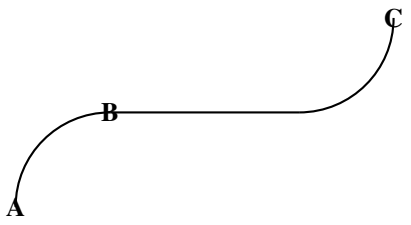


Fig. 2. An example of Dubin's Path for lateral shift

Dubin's path for fixed-wing agents [13] offers a robust solution for lateral shifts. By leveraging the principles of

Pontryagin's maximum principle, this method optimally balances the constraints of turning radius and non-holonomic system dynamics. Through a combination of circular arcs and straight lines, it navigates lateral shifts with remarkable efficiency and precision. A lateral shift from Pose A to C using Dubin's path is illustrated in in Fig 2.

C. Sigmoid Function

Effective 3D path following and successfully implementing Dubins' paths necessitate switching control laws. In this context, we will employ a sigmoid activation function. This approach is a common strategy in nonlinear control systems. The sigmoid function used in this work is given by

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

Fig 3 shows the graph of the sigmoid function. This

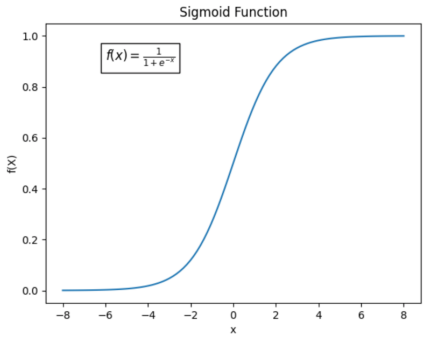


Fig. 3. Sigmoid Function

function maps any real-valued number to the range (0, 1), making it suitable for smooth transitions between control laws. Consider two control laws C_1 and C_2. We define a composite control law C as a weighted sum of C_1 and C_2, with the weights determined by the sigmoid function:

$$C = \sigma(x)C_1 + (1-\sigma(x))C_2.$$

Here, x can be a function of a state variable or an error term.

D. Consensus Algorithm

1) Distributed Agent Count Algorithm : The distributed algorithm for counting the total number of agents allows individual agents within a network to collectively determine the overall count of agents present. Each agent begins by initializing its unique identifier (ID) and communicates this information with its neighboring agents. Through iterative communication and updating, each agent incorporates the received IDs from its neighbors into its own ID set. Upon achieving convergence, each agent can ascertain the total count by evaluating the cardinality of its final ID set.

Let $S_i(k)$ be the set of IDs known to the agent i at iteration k. The update rule for the agent i is written as

$$S_i(k+1) = S_i(k) \cup \left(\bigcup_{j \in N_i} S_j(k) \right), \tag{3}$$

where N_i is the set of neighbors of the agent i. The algorithm converges when

$$S_i(k+1) = S_i(k) \quad \forall i$$

At convergence, each S_i contains the IDs of all agents in the network, and the size of S_i gives the total number of agents.

2) Continuous Fixed Time Average Consensus Algorithm : The continuous fixed-time average consensus algorithm, [15], is designed to enable a network of agents to reach consensus on the average of their initial values within a fixed time, regardless of initial conditions. Each agent i starts with an initial value $x_i(0)$ and updates its state continuously. Given the estimate of the leader's state x_j by agent j, the updated law for each agent's estimate of the leader's state $x_i(t)$:

$$\dot{x}_i = -\sum_{j \in N_i} [\text{sgn}^{\mu_1}(x_i - x_j) + \text{sgn}^{\mu_2}(x_i - x_j)], \tag{4}$$

where

$$\text{sgn}^{\mu}(x) := x||x||^{\mu-1}, \mu_1 < 1 \text{ , and } \mu_2 > 1.$$

III. Proposed Flocking Approach

We first describe the flock geometry given by the platonic solid considered in this work. The virtual leader is positioned at the center of the mass of the flock, with the agents arranged at the vertices of a platonic solid. This configuration allows for effective coordination and minimizes the risk of collision. Given the virtual leader's trajectory and current state, the agents calculate how the corners of a platonic solid evolve, with the virtual leader at the center of the solid. The corners of the platonic solid move along with the virtual leader, following the same non-holonomic trajectory as the virtual leader.

Given the solid's lattice structure, each agent occupies an n-part partition. Each agent is required to track the corresponding corner. If it is ensured that each corner's trajectory remains within its n-part partition and each agent tracks its leader trajectory asymptotically, then collision avoidance is guaranteed. The proposed method aims to establish a decentralized control algorithm for a swarm of fixed-wing agents. The key steps in Fig 4 outline the components and methodology employed to achieve efficient, collision-free flocking behavior.

The process begins with n agents, each spawned randomly in a small cube around its assigned corner of the platonic solid, ensuring safe spacing. Agents perform a distributed agent count algorithm to determine n. Once the value of n is known, each agent estimates the virtual leader's trajectory by communicating only with their neighbors. This is achieved through a continuous fixed-time consensus algorithm, where all agents initialize their estimates to their initial positions and velocities. The Fixed-Time Consensus algorithm ensures that all agents reach a consensus in a fixed time, aligning their understanding of the virtual leader's trajectory. With the leader's trajectory known, each agent knows its assigned corner's trajectory. Now, each agent needs to track this corner trajectory to flock efficiently.

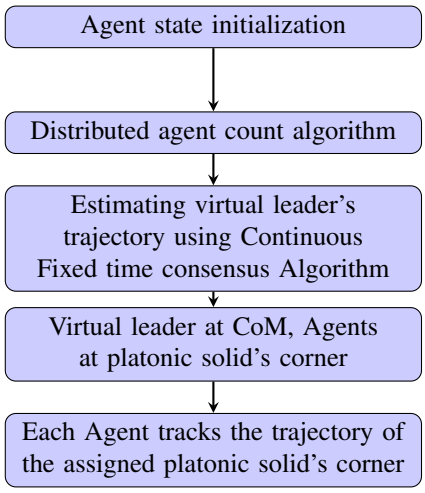


Fig. 4. Block diagram of the decentralized Leader-Follower Flocking pipeline.

The agents track its corner trajectory by employing the following control strategy

- A control strategy incorporating proportional-integral-derivative (PID) control and sigmoid functions ensures smooth and continuous adjustments.
- Speed constraints are managed through feedback mechanisms that restrict acceleration based on maximum and minimum velocities as per the constraints.

Now, we present the mathematical formulation for our tracking law. Before we present the mathematical formulation, we describe some variables that will be used further. The leader and agent positions in the inertial frame of reference are $[x_{\text{leader}}, y_{\text{leader}}, z_{\text{leader}}]$ and $[x_{\text{agent}}, y_{\text{agent}}, z_{\text{agent}}]$ respectively. The relative positions of an agent with respect to the leader in the inertial and leader frame are $[\Delta x, \Delta y, \Delta z]$ and $[\Delta x_l, \Delta y_l, \Delta z_l]$ respectively. The leader's yaw angle and speed are ψ_{leader} and v_{leader} respectively, while $[x_{\text{corner}}, y_{\text{corner}}, z_{\text{corner}}]$ is the position of a corner with respect to the leader in the leader's frame of reference. Next, we present various steps through the mathematical formulation:

Now, the position difference between the virtual leader and the agent, given by $\Delta x = x_{\text{leader}} - x_{\text{agent}},\ \Delta y = y_{\text{leader}} - y_{\text{agent}},\ \Delta z = z_{\text{leader}} - z_{\text{agent}}$ The position difference is changed into the frame of the virtual leader, and each agent should have the same yaw as the leaders.

$$\Delta x_l = \cos(\psi_{\text{leader}})\Delta x + \sin(\psi_{\text{leader}})\Delta y \tag{5a}$$
$$\Delta y_l = -\sin(\psi_{\text{leader}})\Delta x + \cos(\psi_{\text{leader}})\Delta y \tag{5b}$$
$$\Delta z_l = \Delta z. \tag{5c}$$

The control objective for each agent is to reach the assigned corner of the agent and match the leader's yaw. Thus, we have relative variables of an agent with respect to its corner as

$$\delta x = \Delta x_l - x_{\text{corner}},\ \delta y = \Delta y_l - y_{\text{corner}} \tag{6a}$$
$$\delta z = \Delta z_l - z_{\text{corner}},\ \delta\psi = \psi_{\text{leader}} - \psi_{\text{agent}}, \tag{6b}$$
$$\delta v = v_{\text{leader}} - v_{\text{agent}} \tag{6c}$$

When tracking corner trajectory, the agent needs to decide whether to minimize δy first or $\delta\psi$ first. The proposed strategy is to use sigmoid function for the smooth transition. The sigmoid function gets activated when the difference between the yaw angle is increased by more than $\pm\ \frac{\pi}{6}$. This number is just a hyper-parameter and can be changed as required. The Sigmoid function is defined as

$$\sigma_1 = \frac{1}{1 + e^{-1000((\delta\psi)^2 - (\pi/6)^2)}}. \tag{7}$$

This means that the first $|\delta\psi|$ of each agent is decreased to $\frac{\pi}{6}$ degrees, and once that is done, another control law is used to decrease $\delta y, \delta\psi$ to zero simultaneously. When $|\delta\psi| \geq \frac{\pi}{6}$, max angular velocity is applied in the opposite direction until $|\delta\psi| \leq \frac{\pi}{6}$. This switching is smooth and is facilitated by the sigmoid function.

Now, the angular velocity control when $|\delta\psi| \leq \frac{\pi}{6}$) is designed using a ψ-matching term. A ψ-matching term ensures that motion decreases the difference in ψ. It is given by

$$\psi_{\text{matching}} = -\text{sign}(\delta\psi)(1 + \text{sign}(\delta\psi\delta y))\left(\frac{\omega_{\text{max}}}{4}\right). \tag{8}$$

From (8), the ψ_{matching} term becomes 0 if motion is in the direction opposite to that of reducing δy (when both $\delta\psi, \delta y$ are of opposite signs). Further, to decrease δy, we use the Dubin's path. The calculated $d_{\min}$ is used to decide on the minimum distance before switching to angular velocity to match the path.

$$\delta y_{\text{matching}} = \alpha\beta,$$
where
$$\alpha = (1 - \text{sign}(\delta\psi\delta y))\text{sign}(\delta y)$$
$$\beta = (1 - 2\text{sign}((\delta y)^2 - d_{\text{min}}^2))\left(\frac{\omega_{\text{max}}}{2}\right),$$

where α ensures that the direction of Dubin's path is correct while decreasing only δy the opposite of that of $\delta\psi$. While β ensures that the switchin of arc happens once we cross d_{min}.

The final computation uses the sigmoid function for smooth switching

$$\omega' = (\psi_{\text{matching}} + \delta y_{\text{matching}})(1 - \sigma_1) + \sigma_1(\omega_{\text{max}}\text{sign}(-\delta\psi)). \tag{10}$$

to ensure that we first bring the $|\delta\psi|$ less than $\frac{\pi}{6}$ degrees and then simultaneously decrease $\delta y, \delta\psi$ to zero.

For matching speeds, we use a decay function on δx and δv individually to control the acceleration independently, and they are given by

$$\delta x_{\text{matching}} = -0.5\,\text{sign}(\delta x)\left(1 - e^{-\frac{(\delta x)^2}{10}}\right)a_{\text{max}} \tag{11a}$$
$$\delta v_{\text{matching}} = -0.3\,\text{sign}(\delta v)\left(1 - e^{-10(\delta v)^2}\right)a_{\text{max}}. \tag{11b}$$

This choice satisfies acceleration constraints naturally and $\delta x, \delta v$ go to zero simultaneously.

Hence, the final control inputs in an unconstrained manner are given by

$$\phi_{\text{new}} = \arctan\left(\frac{v_{\text{agent}}\omega'}{g}\right), \tag{12a}$$

$$\theta_{\text{new}} = -\text{sign}(\delta z)(\theta_{\text{max}}) \tag{12b}$$

$$a_{\text{inter}} = \delta x_{\text{matching}} + \delta v_{\text{matching}}, \tag{12c}$$

where a_{inter} is an intermediate acceleration does not account for speed constraints being violated. Now, to satisfy speed constraints, we use another activation function given by

$$A_{\text{max}} = -a_{\text{max}}\left(1 - \frac{1}{1+e^{1000(v_{\text{agent}}-v_{\text{max}}+2)}}\right)\Lambda,$$

$$A_{\text{min}} = a_{\text{max}}\left(1 - \frac{1}{1+e^{-1000(v_{\text{agent}}-v_{\text{min}}-2)}}\right)\Lambda,$$

$$\Lambda = \left(1 - \frac{1}{1+e^{-10a_{\text{inter}}}}\right). \tag{13}$$

These terms guarantee the following:

- When, $v \geq (v_{max} - 2)$ then, A_{max} is non-zero, facilitating negative acceleration and thereby preventing a further increase in speed. A_{min} is zero in this case.
- When, $v \leq (v_{min} + 2)$ then, A_{min} is non-zero, facilitating positive acceleration and thereby preventing a further decrease in speed. A_{max} is zero in this case.
- When $v_{\text{min}} + 2 \leq v \leq v_{\text{max}} - 2$, then both $A_{\text{max}}, A_{\text{min}}$ are zero.

The term Λ is multiplied so that acceleration constraints are not violated. Also, the safety factor of 2 m/s is used since speed cannot be increased or decreased instantaneously. The final acceleration is obtained as the summation of all terms and is given by

$$a_{\text{new}} = a_{\text{inter}} + A_{\text{max}} + A_{\text{min}}. \tag{14}$$

IV. Performance Validations

In this section, we evaluate the performance of the proposed algorithm using numerical simulations conducted using a fixed-wing UAV with known specifications to ensure realistic conditions. The UAV's maximum speed ranged from 50 m/s to 150 m/s. It had a maximum acceleration capacity of 1 m/s^2 in both forward and reverse directions. The maximum bound on the UAV's roll and pitch angles was assumed to be 20 degrees and 10 degrees, respectively. The control system operated at a reaction rate of 20 ms, without any communication delay, and under a gravitational acceleration of 9.81 m/s^2. The kinematic model assumed instantaneous roll and pitch adjustments. A safety distance of 5 m is taken with a collision distance of 1 m.

A. Setting 1

To explain the concept optimally, we consider 8 agents positioned within a cuboid of dimensions 400×400×200. A virtual leader is assumed to be located at the center of the flock. Initially, the agents are randomly distributed within a 20×20×20 cube around its assigned corner of

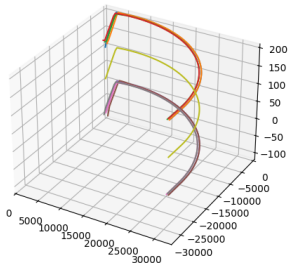


Fig. 5. Path followed

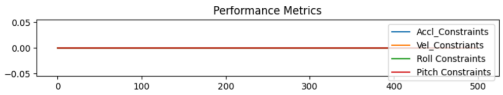


Fig. 6. A plot showing number of times the actuator constraints were violated during one of the Monte-Carlo simulations

the 400×400×200 cuboid. The yaw angle of each agent is randomly initialized between −10 degrees and +10 degrees. Now, the flock needs to head in a particular direction and with a specific speed to complete its group task; these heading and speed are called target yaw and target speed. The target speed and yaw are randomly initialized between their permissible limits. The speed of each agent is initialized to be within ±10% of the target speed. The goal of the flock is to track its assigned corner while moving with the target speed and heading.

This scenario mimics a practical one, where fixed-wing UAVs are often launched in no preset formation with a random target. This setup reflects practical conditions where the initial agent positions and orientations may vary, and the target's speed and yaw are not predetermined.

A Monte Carlo simulation of the described scenario demonstrated no collisions or safety breaches, validating the system's robustness. Each agent's speed, initialized within ± 10% of the target speed, converged toward the target, minimizing deviations. The agents aligned their yaw angles with the targets, starting with yaw angles between -10 and +10 degrees. The agents successfully tracked the corners of the 400×400×200 cuboid while moving with the target yaw

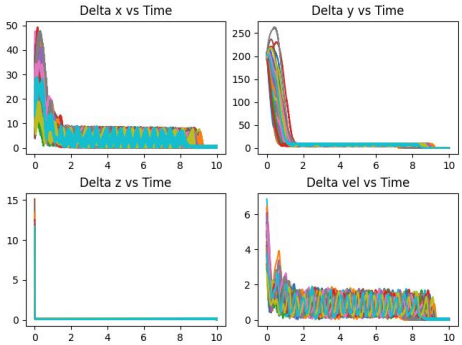


Fig. 7. Time evolution of $\delta x, \delta y, \delta z, \delta v$ vs t averaged over 8 agents. 200 such trajectories are displayed for each randomized simulation.

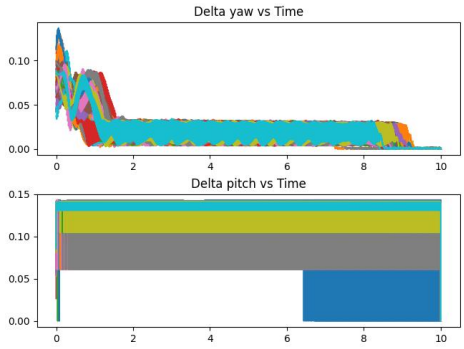


Fig. 8. Time evolution of $\delta\psi, \delta\theta$ vs t averaged over 8 agents. 200 such trajectories are displayed for each randomized simulation.

and speed. Fig 5 shows the trajectories of the agents for one of the randomized simulations. Fig 6 shows the number of times actuator constraints were violated during one of the Monte-Carlo simulations. Fig 7,8 presents the time evolution of different state variables averaged over 8 agents over 200 randomized test cases. As shown, each agent's trajectory converges to its assigned corner trajectory without violating actuator constraints and without any inter-agent collisions. The agents reformed into the desired configuration with the virtual leader at the center, maintaining formation integrity and achieving consensus on key parameters.

V. Conclusion

The current landscape of decentralized algorithms for 3D fixed-wing flocking needs a solution that adequately considers the actuator saturation. Despite the advancements in multi-agent systems and flocking behaviors, existing algorithms must address the limitations imposed by actuator constraints. The research presented in this paper detailed the development and implementation of a decentralized flocking algorithm for 3D fixed-wing UAVs that addressed actuator saturation. Agents were positioned at the vertices of a cuboid with the virtual leader at the center, minimizing collision risks. The algorithm integrated PID control and sigmoid functions to manage positional and velocity tracking, ensuring smooth and safe adjustments. Simulation results showcased the algorithm's robustness, with agents maintaining formation and achieving consensus without collisions. The work presented can be extended to accommodate significant numbers of agents. This study considered a fixed number of eight agents arranged in a cuboid formation. Future work would explore the algorithm's scalability by considering formations based on platonic solids for any number of agents. This approach will ensure the algorithm's scalability and effectiveness in more enormous UAV swarms. We can enhance decentralized flocking algorithms' practical applications and robustness in more complex and varied operational environments by addressing these challenges.

VI. Acknowledgment

The authors would like to thank Rohan Waghmare, Anushka Sinha, and Aakash Chaphalkar of L&T Defense for funding this research.

References

[1] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," in *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401-420, March 2006.

[2] J. Wang and M. Xin, "Integrated Optimal Formation Control of Multiple Unmanned Aerial Vehicles," in *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1731-1744, Sept. 2013.

[3] H. Duan, Q. Luo, Y. Shi and G. Ma, "?Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAV Formation Reconfiguration," in *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 16-27, Aug. 2013.

[4] Wei Lin, "Distributed UAV formation control using differential game approach," in *Aerospace Science and Technology*, Volume 35, pp 54-62, 2014.

[5] S. A. P. Quintero, G. E. Collins and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," *2013 American Control Conference*, pp 2025-2031, 2013.

[6] J. Sun, J. Tang and S. Lao, "Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm," in *IEEE Access*, vol. 5, pp. 18382-18390, 2017.

[7] N. Moshtagh and A. Jadbabaie, "Distributed Geodesic Control Laws for Flocking of Nonholonomic Agents," in *IEEE Transactions on Automatic Control*, vol. 52, no. 4, pp. 681-686, April 2007.

[8] Tanner, H.G., Jadbabaie, A., Pappas, G.J. "Flocking in Teams of Nonholonomic Agents," in *Kumar, V., Leonard, N., Morse, A.S. (eds) Cooperative Control Lecture Notes in Control and Information Science*, vol 309.

[9] Y. Liang and H.H. Lee, "Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints," *2006 American Control Conference*, pp. 6, 2006.

[10] H. Qiu, H. Duan, "Pigeon interaction mode switch-based UAV distributed flocking control under obstacle environments," in *ISA Transactions*, Volume 71, Part 1, pp. 93-102, 2017.

[11] Q. Luo, H. Duan, "Distributed UAV flocking control based on homing pigeon hierarchical strategies," in *Aerospace Science and Technology,* Volume 70, pp. 257-264, 2017

[12] S. Hauert et al., "Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5015-5020, 2011.

[13] I. Lugo-Cárdenas, G. Flores, S. Salazar and R. Lozano, "Dubins path generation for a fixed wing UAV," in *2014 International Conference on Unmanned Aircraft Systems*, pp. 339-346, 2014.

[14] V. Anand, R. Waghmare, "Structured Flocking for Fixed Wing Multiple Agents," in *IFAC-PapersOnLine*, Volume 55, Issue 22, pp. 352-357, 2022.

[15] M. Baranwal, K. Garg, D. Panagou and A. O. Hero, "Robust Distributed Fixed-Time Economic Dispatch Under Time-Varying Topology," in *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1183-1188, Oct. 2021.