

SOFTWARE-IN-THE-LOOP SIMULATION OF DISTRIBUTED 3D INTEGRATED SWARM GUIDANCE FOR FIXED-WING UAVS

Yechan Yang[1], Junho Jeong[2], Seungkeun Kim[1] & Jinyoung Suk[1,†]

[1]Department of Aerospace Engineering, Chungnam National University, Daejeon, Republic of Korea
[2]Department of Aerospace Artificial Intelligence Software, Hanseo University, Seosan, Republic of Korea

Abstract

This paper proposes a distributed three-dimensional integrated guidance that simultaneously achieves path following, flocking, and collision avoidance for fixed-wing unmanned aerial vehicles (FWUAVs) in swarm flight. Fixed-wing UAVs are limited by their kinematics: they cannot hover and require bank-angle control for turning flight. Therefore, balancing the conflicting control objectives of precise path following and safe swarm maintenance poses a significant challenge. To overcome this problem, this study presents a regularization-based integrated method that combines the outputs of vector field guidance, augmented Cucker-Smale model, and potential field guidance within the physical limits of the UAV. The proposed algorithm is implemented in a Software-In-the-Loop (SILS) environment utilizing PX4, ROS2, and Gazebo, validating the theoretical derivations in a practical flight software stack. The simulation demonstrates the feasibility of cooperative flight for multiple UAVs to simultaneously perform diverse missions.

Keywords: Fixed-Wing UAVs, Three-Dimensional Swarm Flight, Distributed Integrated Guidance, Software-In-the-Loop Simulation

1. Introduction

Swarm flight using multiple unmanned aerial vehicles (UAVs) has been actively studied because it can improve mission scalability, spatial coverage, and operational robustness compared with single-UAV operation. In particular, distributed swarm control has attracted considerable attention because each UAV can generate collective behavior through local information exchange without relying on a central node. This structure is advantageous for scalable and robust swarm operation, especially when communication links or individual agents are partially degraded.

Early studies on distributed swarm control introduced behavior-based rules, flocking models, and consensus-based cooperative control methods [1–3]. Among them, the Cucker-Smale model provides a representative velocity-alignment mechanism for multi-agent systems. The augmented Cucker-Smale (ACS) model further incorporates inter-particle bonding forces, enabling velocity alignment, cohesion, and spatial formation maintenance [4]. Therefore, ACS-based flocking can serve as a useful basis for distributed swarm formation of fixed-wing UAVs (FWUAVs).

However, flocking control alone is not sufficient for mission-level FWUAV swarm flight. Unlike rotary-wing UAVs, FWUAVs cannot hover and must maintain continuous forward motion with limited turning capability. Therefore, swarm guidance commands should simultaneously consider path following, formation maintenance, collision avoidance, and the physical maneuvering limits of fixed-wing aircraft. Although vector field guidance has been widely used for FWUAV path following because of its smooth

†Corresponding author: jsuk@cnu.ac.kr

convergence characteristics for straight and circular paths [5], direct application of single-vehicle path following guidance to multiple UAVs may reduce inter-agent distances and increase collision risk.
Hybrid flocking approaches have recently been proposed by combining path following guidance, flocking control, and collision avoidance for FWUAV swarms [6, 7]. These studies demonstrated the feasibility of cooperative fixed-wing swarm flight. Nevertheless, when heterogeneous guidance commands are integrated using optimization-based weighting, the computational burden may become significant for onboard systems that must update guidance commands at a fixed rate for multiple UAVs. Thus, a lightweight integrated guidance framework is required for practical FWUAV swarm flight.
This paper proposes a distributed three-dimensional integrated guidance method for FWUAV swarm flight. The proposed framework combines vector field (VF) guidance, ACS based flocking guidance, and potential field (PF) based collision avoidance. The VF component generates mission-dependent path-following commands, the ACS component maintains three-dimensional swarm cohesion and velocity alignment, and the PF component provides local inter-agent collision avoidance. To combine these heterogeneous commands, a normalization-based weighted integration method is introduced so that the final guidance command remains compatible with the physical maneuvering limits of the FWUAV without requiring online optimization.
The main contributions of this paper are summarized as follows:

- **Distributed integrated guidance framework:** A distributed guidance framework is developed to simultaneously address path following, swarm formation maintenance, and collision avoidance for FWUAV swarms.
- **Three-dimensional swarm formation:** The ACS-based flocking guidance is extended to three-dimensional space to improve spatial utilization and inter-agent separation within a compact mission region.
- **SILS implementation and validation:** The proposed algorithm is implemented in a PX4–ROS2–Gazebo based Software-In-the-Loop Simulation (SILS) environment and validated through cooperative waypoint and loiter missions.

The remainder of this paper is organized as follows. Section II presents the proposed distributed integrated guidance algorithm. Section III describes the SILS environment and mission scenario. Section IV presents the simulation results, and Section V concludes the paper.

2. Distributed Integrated Guidance Law

This section presents the distributed integrated guidance law for cooperative FWUAV swarm flight. The objective is to allow multiple UAVs to follow mission paths while maintaining swarm cohesion and inter-agent safety. To this end, vector field guidance, augmented Cucker-Smale based flocking guidance, and potential field based collision avoidance are formulated and integrated through a normalization based weighted scheme.

2.1 Path Following Guidance

A vector field guidance law is used to generate path following commands for FWUAVs. Because the proposed mission includes both waypoint transition and circular loitering, a three-dimensional VF is used for straight-path following, whereas a two-dimensional VF is used for horizontal loitering.

2.1.1 Three-Dimensional Vector Field

For straight-path following, a local cylindrical coordinate frame is defined along the desired path, as shown in Figure 1a. Following the vector field formulation in [8], the relationship between the cylindrical and local Cartesian velocity components is given by

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{bmatrix} = \begin{bmatrix} \cos\lambda & -\sin\lambda & 0 \\ \sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{r} \\ r\dot{\lambda} \\ \dot{z}' \end{bmatrix}, \tag{1}$$

where x', y', and z' denote the local Cartesian coordinates in the path-aligned frame, r is the cross-track error, λ is the azimuth angle around the path axis, and z' denotes the distance along the path direction. The approach angle (ρ) and the desired cylindrical velocity are defined as

$$\rho(r) = \alpha \arctan(\beta r), \tag{2}$$

$$\dot{r} = -V_g \sin\rho(r), \quad r\dot{\lambda} = 0, \quad \dot{z}' = V_g \cos\rho(r), \tag{3}$$

where α determines the maximum approach angle, β adjusts the convergence rate, and V_g is the ground speed. Substituting Eq. (3) into Eq. (1) yields

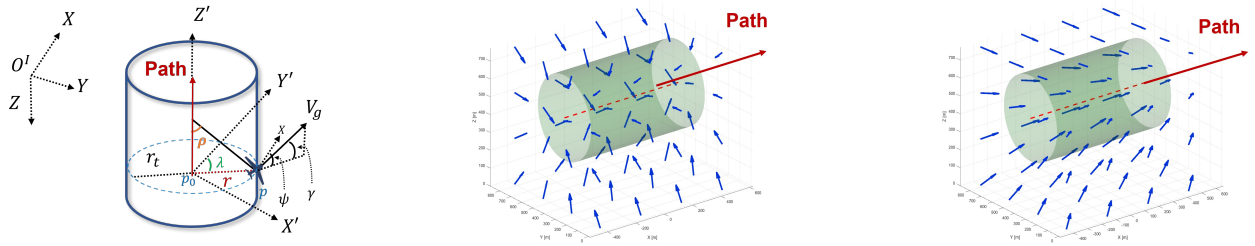
$$\begin{bmatrix} \dot{x}'_d \\ \dot{y}'_d \\ \dot{z}'_d \end{bmatrix} = \begin{bmatrix} -V_g \sin\rho(r)\cos\lambda \\ -V_g \sin\rho(r)\sin\lambda \\ V_g \cos\rho(r) \end{bmatrix}. \tag{4}$$

Here, $\dot{x}'_d$, $\dot{y}'_d$, and $\dot{z}'_d$ are the desired velocity components in the local path-aligned frame. The desired velocity $(\dot{p}_d)$ in the inertial frame and the corresponding guidance angles(ψ_d, γ_d) are obtained as

$$\dot{p}_d = R^T \begin{bmatrix} \dot{x}'_d \\ \dot{y}'_d \\ \dot{z}'_d \end{bmatrix}, \tag{5}$$

$$\psi_d = \text{atan2}(\dot{y}_d, \dot{x}_d), \quad \gamma_d = \text{atan2}\left(-\dot{z}_d, \sqrt{\dot{x}_d^2 + \dot{y}_d^2}\right), \tag{6}$$

where R is the rotation matrix from the inertial frame to the local path frame.



(a) Cylindrical frame (b) $\alpha = 0.9,\ \beta = 0.2$ (c) $\alpha = 0.9,\ \beta = 0.002$

Figure 1 – Three-dimensional vector field for straight-path following.

2.1.2 *Two-Dimensional Vector Field*

For loitering, a two-dimensional vector field is designed in the horizontal plane to maintain a prescribed circular orbit around the target region. Adapting the formulation in [9], a polar coordinate system (r, λ) is defined at the loiter center, where r_t is the target radius. The position and velocity relationship is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\lambda & -\sin\lambda \\ \sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \dot{r} \\ r\dot{\lambda} \end{bmatrix}. \tag{7}$$

The circular loiter vector field is defined as

$$\begin{bmatrix} \dot{r} \\ r\dot{\lambda} \end{bmatrix} = \frac{V_g}{\sqrt{(r-r_t)^2 + p_c^2 r^2}} \begin{bmatrix} -(r-r_t) \\ p_c r \end{bmatrix}, \tag{8}$$

where p_c is the convergence gain. Transforming Eq. (8) into the inertial frame gives

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \frac{V_g}{\sqrt{(r-r_t)^2 + p_c^2 r^2}} \begin{bmatrix} -(r-r_t)\cos\lambda - p_c r \sin\lambda \\ -(r-r_t)\sin\lambda + p_c r \cos\lambda \end{bmatrix}. \tag{9}$$

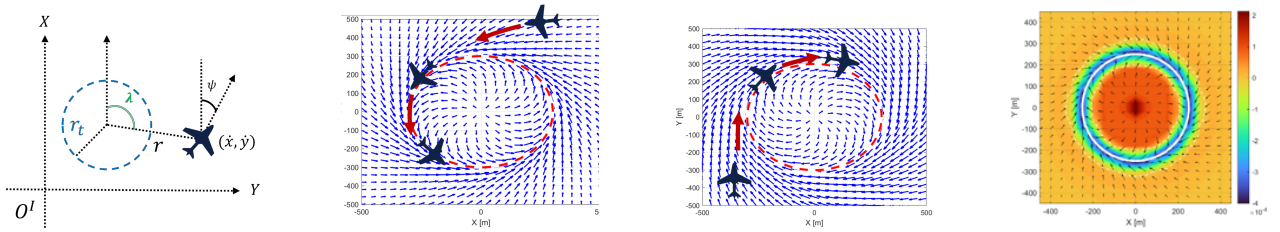
The desired heading angle (ψ_d) is then obtained as

$$\psi_d = \text{atan2}(\dot{y}_d, \dot{x}_d). \tag{10}$$

Finally, the VF guidance commands are generated from the angle errors:

$$u_{\psi,\text{VF}} = \dot{\psi}_{cmd} = k_\psi(\psi_d - \psi), \quad u_{\gamma,\text{VF}} = \dot{\gamma}_{cmd} = k_\gamma(\gamma_d - \gamma), \tag{11}$$

where k_ψ and k_γ are positive guidance gains.



(a) Polar frame (b) $p_c = 0.4$ (c) $p_c = -0.8$ (d) Intensity

Figure 2 – Two-dimensional vector field for loiter guidance.

2.2 ACS-based Flocking Guidance

For FWUAV swarms, rigid position-constrained formation control may become restrictive because fixed-wing aircraft must maintain forward motion and satisfy nonholonomic constraints [10]. Therefore, this study adopts ACS-based flocking guidance, which provides velocity alignment, cohesion, and separation through distributed relative-state interactions [4].

Let $\mathbf{x}_i, \mathbf{v}_i \in \mathbb{R}^3$ denote the position and velocity of the i-th UAV. The ACS acceleration command is defined as

$$\frac{d\mathbf{v}_i}{dt} = \frac{\lambda}{N}\sum_{j=1}^{N}\eta(r_{ij})(\mathbf{v}_j-\mathbf{v}_i) + \frac{\sigma}{N}\sum_{j=1}^{N}\frac{K_1}{2r_{ij}^2}\langle \mathbf{v}_i-\mathbf{v}_j, \mathbf{x}_i-\mathbf{x}_j\rangle(\mathbf{x}_j-\mathbf{x}_i) + \frac{\sigma}{N}\sum_{j=1}^{N}\frac{K_2}{2r_{ij}}(r_{ij}-2R)(\mathbf{x}_j-\mathbf{x}_i), \tag{12}$$

where $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the relative distance, N is the number of UAVs, λ, σ, K_1, and K_2 denote the ACS gain parameters, R is the desired spacing parameter, and the communication weight is

$$\eta(r_{ij}) = \frac{1}{(1+r_{ij}^2)^{\beta}}. \tag{13}$$

To convert the ACS acceleration into FWUAV guidance commands, the inertial velocity of the i-th UAV is expressed as

$$\mathbf{v}_i^I = \begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{z}_i\end{bmatrix} = \begin{bmatrix}V_{g,i}\cos\psi_i\cos\gamma_i\\ V_{g,i}\sin\psi_i\cos\gamma_i\\ -V_{g,i}\sin\gamma_i\end{bmatrix}, \tag{14}$$

where the inertial frame follows the NED convention. The inertial ACS acceleration is transformed into the velocity-aligned body frame as

$$\mathbf{a}_{i,\mathrm{ACS}}^B = R_\gamma R_\psi \mathbf{a}_{i,\mathrm{ACS}}^I = \begin{bmatrix}a_{i,\mathrm{ACS},x}^B\\ a_{i,\mathrm{ACS},y}^B\\ a_{i,\mathrm{ACS},z}^B\end{bmatrix}. \tag{15}$$

Using the fixed-wing kinematic relation

$$\mathbf{a}_{i,\mathrm{ACS}}^B = \begin{bmatrix}\dot{V}_{g,i}\\ V_{g,i}\cos\gamma_i\dot{\psi}_i\\ -V_{g,i}\dot{\gamma}_i\end{bmatrix}, \tag{16}$$

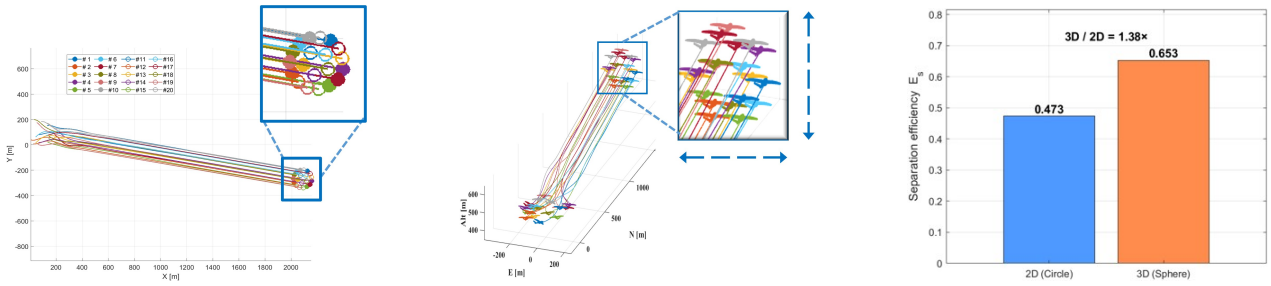
the ACS-based guidance commands are obtained as

$$\begin{aligned} u_{i,V,\mathrm{ACS}} &= \dot{V}_{i,\mathrm{cmd,ACS}} = a_{i,\mathrm{ACS},x}^B,\\ u_{i,\psi,\mathrm{ACS}} &= \dot{\psi}_{i,\mathrm{cmd,ACS}} = \frac{a_{i,\mathrm{ACS},y}^B}{V_{g,i}\cos\gamma_i},\\ u_{i,\gamma,\mathrm{ACS}} &= \dot{\gamma}_{i,\mathrm{cmd,ACS}} = -\frac{a_{i,\mathrm{ACS},z}^B}{V_{g,i}}. \end{aligned} \tag{17}$$

Because the ACS interaction is formulated in three-dimensional space, the swarm can form a volumetric configuration rather than a planar formation. To quantify this spatial advantage, the separation efficiency (E_s) is defined as

$$E_s = \frac{\bar{d}_{nn}}{R_{\mathrm{rms}}}, \tag{18}$$

where $\bar{d}_{nn}$ is the average nearest-neighbor distance and R_{rms} is the root-mean-square swarm radius. As shown in Figure 3, the three-dimensional formation increases E_s from 0.473 to 0.653, corresponding to an improvement of approximately 1.38 times.



(a) 2D swarm formation (b) 3D swarm formation (c) Separation efficiency

Figure 3 – Comparison of two-dimensional and three-dimensional swarm formations generated by the ACS model.

2.3 Collision Avoidance Guidance

Although the ACS model regulates swarm-level cohesion, local collision avoidance is still required in dense swarm flight. This study employs a potential field based repulsive guidance law that generates an avoidance command when neighboring UAVs approach the safety region.

Let $\rho_{ij} = \mathbf{x}_j - \mathbf{x}_i$ be the relative position vector from the i-th UAV to the j-th UAV and $r_{ij} = \|\rho_{ij}\|$. The unit relative position vector is

$$\hat{\rho}_{ij} = \frac{\rho_{ij}}{\|\rho_{ij}\|}. \tag{19}$$

The PF-based repulsive acceleration is formulated as

$$\mathbf{a}_{i,\mathrm{PF}} = -\sum_{j=1}^{N} B_s(r_{ij})\hat{\rho}_{ij}, \tag{20}$$

where $B_s(r_{ij})$ is a bell-shaped repulsive magnitude:

$$B_s(r_{ij}) = \frac{B_d}{1 + \left|\dfrac{r_{ij} - B_c}{B_a}\right|^{2B_b}}. \tag{21}$$

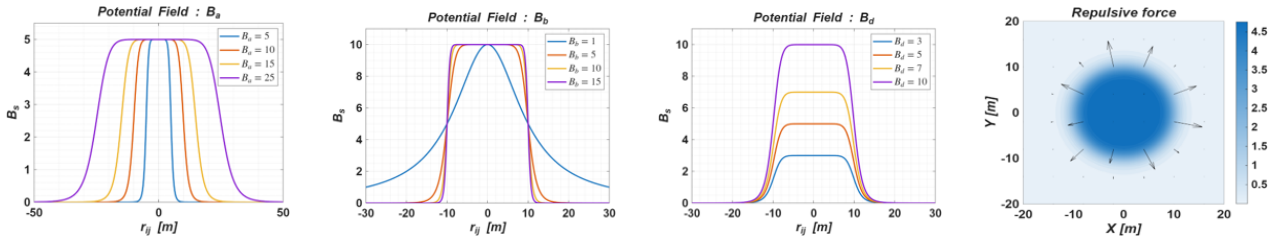
Here, B_a and B_c are distance parameters, B_b determines the shape of the function, and B_d specifies the maximum repulsive acceleration.

Using the same acceleration-to-guidance conversion as in the ACS guidance, the PF acceleration is transformed into the body frame:

$$\mathbf{a}^B_{i,\mathrm{PF}} = R_\gamma R_\psi \mathbf{a}_{i,\mathrm{PF}} = \begin{bmatrix} a^B_{i,\mathrm{PF},x} \\ a^B_{i,\mathrm{PF},y} \\ a^B_{i,\mathrm{PF},z} \end{bmatrix}. \tag{22}$$

The corresponding guidance commands are

$$\begin{aligned} u_{i,V,\mathrm{PF}} &= \dot{V}_{i,\mathrm{PF,cmd}} = a^B_{i,\mathrm{PF},x}, \\ u_{i,\psi,\mathrm{PF}} &= \dot{\psi}_{i,\mathrm{PF,cmd}} = \frac{a^B_{i,\mathrm{PF},y}}{V_i \cos\gamma_i}, \\ u_{i,\gamma,\mathrm{PF}} &= \dot{\gamma}_{i,\mathrm{PF,cmd}} = -\frac{a^B_{i,\mathrm{PF},z}}{V_i}. \end{aligned} \tag{23}$$



(a) B_a (b) B_b (c) B_d (d) Repulsive field

Figure 4 – Design parameter analysis of the bell-shaped potential field.

2.4 Integrated Swarm Guidance

The VF, ACS, and PF guidance laws produce commands with different physical meanings and magnitudes. Direct summation may therefore cause command dominance or actuator saturation. To address this problem, each guidance command is normalized with respect to its physical limit, combined using weighted summation, and denormalized into the physical command domain.

Each raw command u_k is normalized as

$$s(u_k) = \begin{cases} \tanh\left(k_s \dfrac{u_k}{|u_{\max}|}\right), & u_k \geq 0, \\ -\tanh\left(k_s \dfrac{u_k}{|u_{\min}|}\right), & u_k < 0, \end{cases} \tag{24}$$

where $u_{\max}$ and $u_{\min}$ are the upper and lower physical limits of the corresponding command. The scaling gain is selected as

$$k_s = \operatorname{atanh}(q), \quad 0 < q < 1, \tag{25}$$

where q is a saturation margin in the normalized domain.

Let S_{VF}, S_{ACS}, and S_{PF} denote the normalized outputs of the VF, ACS, and PF guidance laws. The integrated normalized command is computed as

$$S_{\text{mix}} = \omega_{VF} S_{VF} + \omega_{ACS} S_{ACS} + \omega_{PF} S_{PF}, \tag{26}$$

subject to

$$\omega_{VF} + \omega_{ACS} + \omega_{PF} = 1, \quad \omega_k \geq 0. \tag{27}$$

Finally, the physical command is recovered through the inverse scaling function:

$$U_{\text{cmd}} = \begin{cases} u_{\max} \dfrac{\operatorname{atanh}(S_{\text{mix}})}{k_s}, & S_{\text{mix}} \geq 0, \\ -|u_{\min}| \dfrac{\operatorname{atanh}(-S_{\text{mix}})}{k_s}, & S_{\text{mix}} < 0. \end{cases} \tag{28}$$

Because the proposed method requires only algebraic operations, it can be implemented in real time without solving an online optimization problem.

Algorithm 1 Integrated Swarm Guidance

Input: UAV state X_i, neighbor states X_j, waypoints, physical limits, weights ω_k
Output: Final guidance command $U_{i,\text{cmd}}$

1: **for** each UAV i **do**
2: Compute $u_{i,VF}$, $u_{i,ACS}$, and $u_{i,PF}$
3: Normalize each command using Eq. (24)
4: Compute $S_{i,\text{mix}}$ using Eq. (26)
5: Apply inverse scaling to obtain $U_{i,\text{cmd}}$ using Eq. (28)
6: Send $U_{i,\text{cmd}}$ to the flight controller
7: **end for**

3. System Architecture

SILS validation is performed to evaluate whether the proposed guidance algorithm can operate within an autopilot based fixed-wing software loop before flight tests.

3.1 SILS Environment

The SILS environment is constructed using PX4, ROS2(Robot Operating System 2), and Gazebo, as shown in Figure 5. The proposed swarm guidance algorithm is implemented as an external ROS2 module, while PX4 handles the low-level flight control and Gazebo simulates the multi-FWUAV dynamics. PX4 and ROS2 are connected through uXRCE-DDS, allowing PX4 uORB messages to be exchanged as ROS2 topics.

Each UAV instance publishes its position, velocity, and attitude states to the ROS2 environment. The Swarm Flight Node computes the integrated guidance command for each vehicle using the shared swarm states and sends the corresponding command to PX4. Inter-agent state sharing is assumed to be available through the ROS2 network, and each UAV instance is assigned an independent namespace and system identifier.

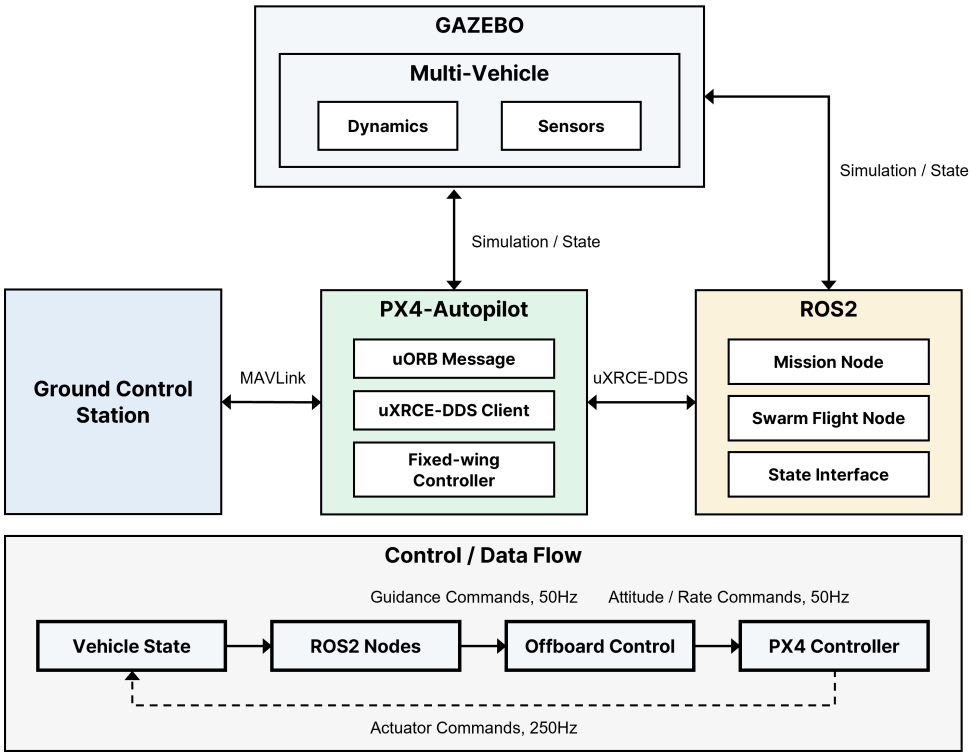


Figure 5 – Swarm Flight SILS Architecture

The guidance commands generated by the Swarm Flight Node are converted into PX4 compatible fixed-wing control commands. Because FWUAV heading motion is achieved through bank-to-turn dynamics, the heading-rate command is converted into a roll angle setpoint using the coordinated-turn relationship:

$$\phi_{sp,i} = \tan^{-1}\left(\frac{V_{g,i}\dot{\psi}_{cmd,i}}{g}\right), \tag{29}$$

where $\phi_{sp,i}$ is the roll angle setpoint of the i-th UAV, $V_{g,i}$ is the ground speed, $\dot{\psi}_{cmd,i}$ is the heading-rate command, and g is the gravitational acceleration.

For longitudinal guidance, the flight path angle-rate command derived from the proposed guidance law is injected into the PX4 pitch rate command $q_{cmd,i}$ in a feedforward type by partially modifying the pitch controller. Thus, the lateral command is applied through the roll angle setpoint, while the vertical command is reflected through the pitch rate channel. The guidance commands are published at 50 Hz, and PX4 converts them into actuator commands through its attitude and angular rate controllers.

3.2 Mission Scenario

The mission scenario is designed to evaluate an individual-to-swarm transition rather than a prearranged compact formation. Each UAV first performs autopilot-managed takeoff and individual loitering, and then switches to the offboard swarm-guidance mode after a swarm mission trigger, as shown in Figure 6.

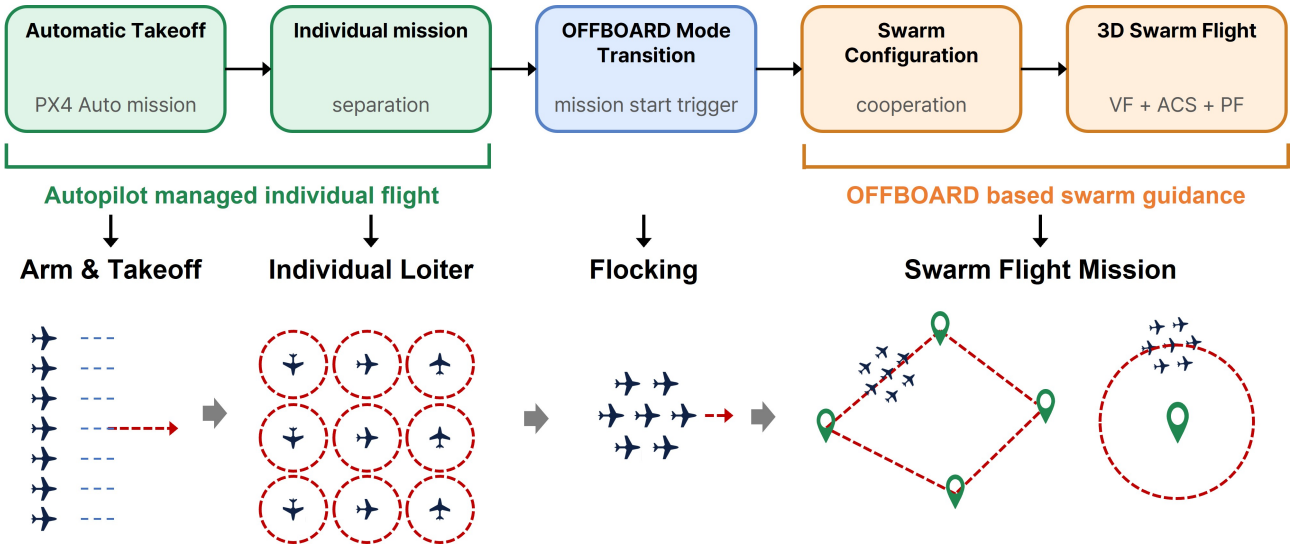


Figure 6 – Mission scenario and mode-transition sequence for SILS-based swarm guidance evaluation.

At the transition time, the UAVs start from spatially separated loiter regions with different positions and flight directions. This scenario verifies whether the proposed guidance law can merge initially dispersed FWUAVs into a coherent three-dimensional swarm while maintaining path following performance and inter-agent safety.

4. Simulation Results and Discussion

This section validates the proposed distributed three-dimensional integrated swarm guidance algorithm in the PX4–ROS2–Gazebo-based SILS environment. Following the mission scenario shown in Figure 6, the SILS simulation is conducted with 12 FWUAVs. The guidance commands are updated at 50 Hz.

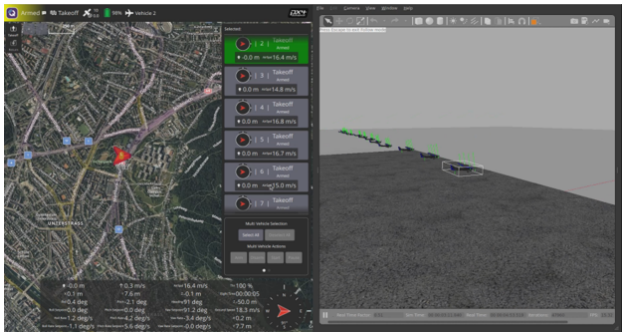
The mission-dependent weighting factors are listed in Table 1. Here, ω_{VF}, ω_{ACS}, and ω_{PF} denote the weights for path following, flocking, and collision avoidance, respectively. In the waypoint mission, ω_{VF} and ω_{ACS} are set equally to balance reference-path following and swarm aggregation. In the loiter mission, ω_{ACS} is increased to emphasize swarm distribution around the target region.

Table 1 – Weight configurations for mission phases

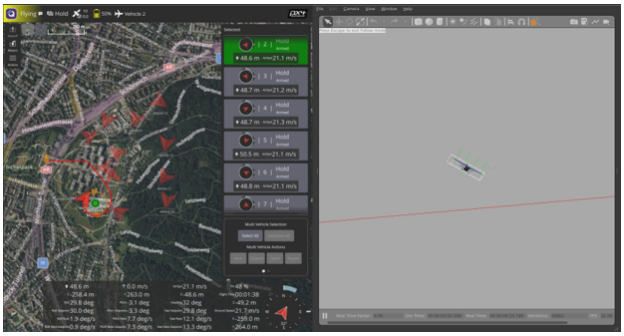
Phase	**Mission**	ω_{VF}	ω_{ACS}	ω_{PF}
Phase 1	Waypoint	0.4	0.4	0.2
Phase 2	Loiter	0.3	0.6	0.1

Figure 7 shows the SILS visualization of the mission sequence. The 12 FWUAVs first perform individual takeoff and loitering, and then switch to the cooperative swarm-guidance mode for waypoint flight and loitering. This result confirms that the proposed guidance algorithm can be executed through the PX4–ROS2–Gazebo software loop during sequential mission transitions.

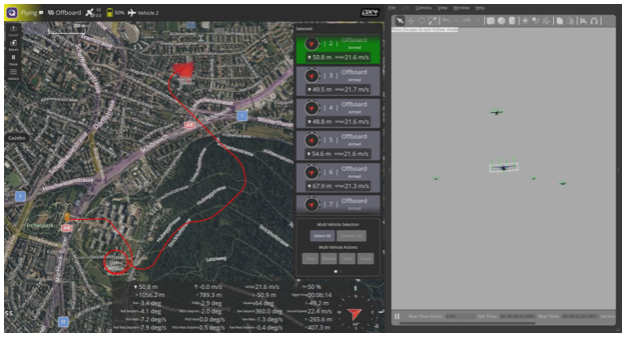
Figure 8 presents the three-dimensional and top-view trajectories of the 12 UAVs. Although the UAVs start from spatially separated loiter regions with different positions and headings, they merge into a cooperative waypoint path after the offboard transition and subsequently perform the loiter mission.



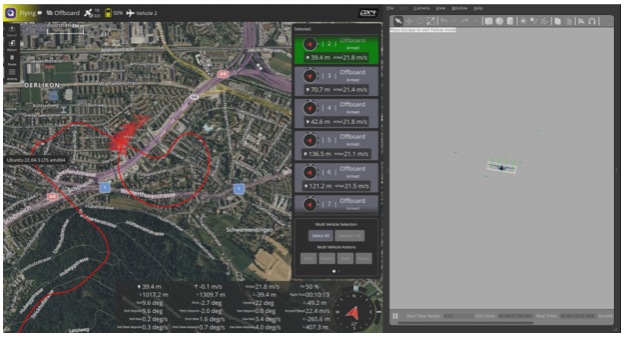
(a) Individual takeoff



(b) Individual mission

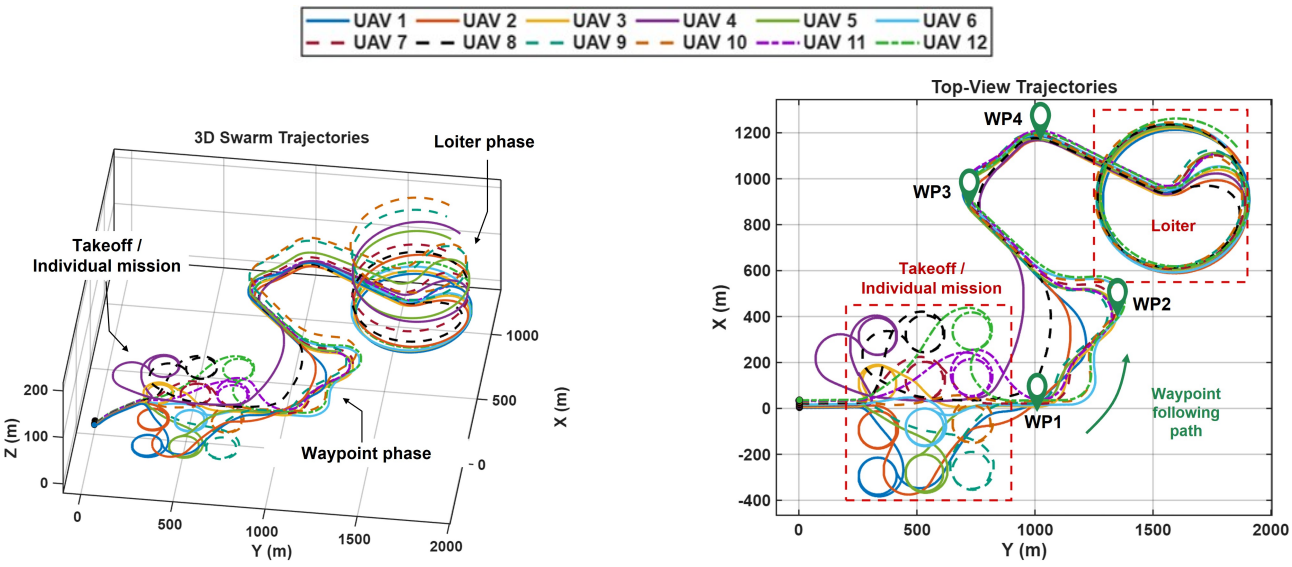


(c) Cooperative waypoint mission



(d) Cooperative loiter mission

Figure 7 – SILS visualization of the FWUAV swarm mission phases.



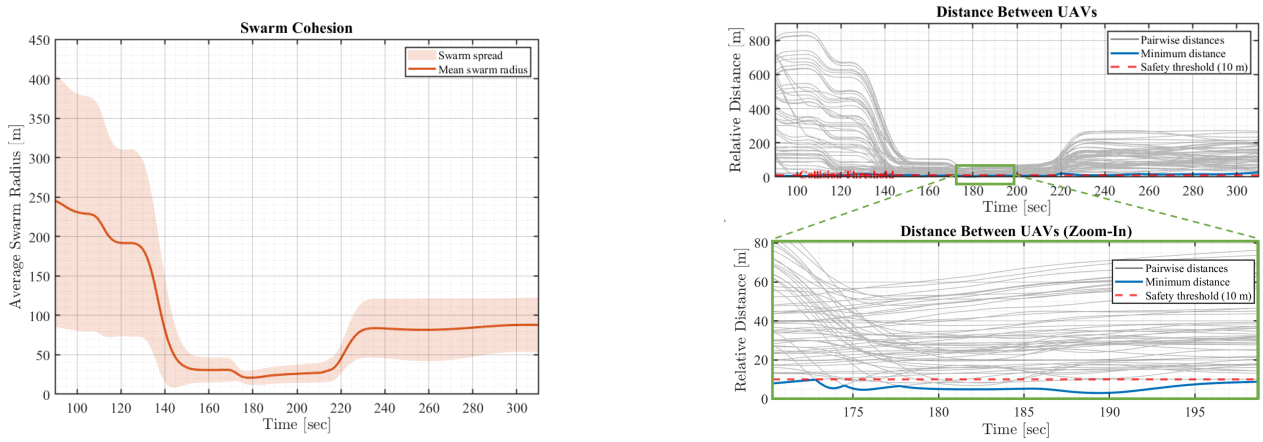
(a) 3D swarm trajectories

(b) Top-view trajectory with annotated waypoints

Figure 8 – Swarm trajectories for the 12-UAV SILS mission.

The three-dimensional trajectory shows that the swarm is not constrained to a planar formation but maintains altitude-wise distribution during the mission. This indicates that the proposed ACS-based flocking and PF-based avoidance components preserve a volumetric swarm configuration while supporting path-following behavior.

To evaluate swarm performance, the average swarm radius and inter-agent relative distances are analyzed, as shown in Figure 9. The average swarm radius is initially large because the UAVs are distributed around individual loiter regions. After the transition to swarm-guidance mode, the radius decreases as the ACS-based flocking guidance aggregates the UAVs into a cohesive group. During the cooperative loiter phase, the radius remains bounded, indicating that the swarm maintains a stable group structure while orbiting around the target region.



(a) Swarm cohesion

(b) Inter-agent relative distance

Figure 9 – Swarm performance results in the SILS environment.

The inter-agent distance results show that some UAV pairs approach each other during the transition and cooperative flight phases. When the inter-agent distance decreases, the PF-based collision avoidance component generates larger repulsive commands, preventing collision between neighboring UAVs.

Overall, the SILS results demonstrate that the proposed distributed three-dimensional integrated guidance algorithm can perform cooperative waypoint and loiter missions from spatially dispersed initial conditions. The results also confirm that the algorithm can be implemented within the PX4–ROS2–Gazebo-based software loop and can maintain both swarm cohesion and inter-agent safety during mission-level transitions.

5. Conclusion

This paper proposed a distributed three-dimensional integrated guidance strategy for FWUAV swarm flight. The proposed method combines vector field guidance, augmented Cucker-Smale based flocking guidance, and potential field based collision avoidance through a normalization based weighted integration framework. By normalizing heterogeneous guidance commands with respect to the physical maneuvering limits of the UAV, the proposed approach generates feasible fixed-wing guidance commands. The proposed algorithm was implemented in a PX4–ROS2–Gazebo based SILS environment and validated using a 12-UAV sequential mission scenario. The simulation results showed that the UAVs could transition from individual missions to cooperative waypoint flight and loitering from spatially dispersed initial conditions. In addition, the trajectory, swarm cohesion, and inter-agent distance results demonstrated that the proposed algorithm maintained a three-dimensional swarm structure while supporting path following performance and inter-agent safety. Future work will consider Hardware-In-the-Loop Simulation and flight tests to further evaluate the robustness of the proposed framework under real operating conditions.

6. Contact Author Email Address

mailto: jsuk@cnu.ac.kr (Corresponding Author)
mailto: diddpcks12@o.cnu.ac.kr

7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT)(No. 2021R1A5A1031868).

References

[1] Reynolds C. W. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[2] Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[3] Cucker F. and Smale S. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.

[4] Park J., Kim H., and Ha S. Cucker-smale flocking with inter-particle bonding forces. *IEEE Transactions on Automatic Control*, 55(11):2617–2623, 2010.

[5] Nelson D. R., Barber D. B., McLain T. W., and Beard R. W. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, 2007.

[6] Song Y., Lim S., Myung H., Lee H., Jeong J., Lim H., and Oh H. Distributed swarm system with hybrid-flocking control for small fixed-wing uavs: Algorithms and flight experiments. *Expert Systems with Applications*, 229:120457, 2023.

[7] Lim S., Song Y., Choi J., Myung H., Lim H., and Oh H. Decentralized hybrid flocking guidance for a swarm of small uavs. In *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, pages 287–296. IEEE, 2019.

[8] Liang Y., Jia Y., Du J., and Zhang J. Vector field guidance for three-dimensional curved path following with fixed-wing uavs. In *2015 American Control Conference (ACC)*, pages 1187–1192. IEEE, 2015.

[9] Jung W., Lim S., Lee D., and Bang H. Unmanned aircraft vector field path following with arrival angle control. *Journal of Intelligent & Robotic Systems*, 84(1):311–325, 2016.

[10] Kownacki C. and Ambroziak L. Local and asymmetrical potential field approach to leader tracking problem in rigid formations of fixed-wing uavs. *Aerospace Science and Technology*, 68:465–474, 2017.

[11] PX4 Development Team. PX4 Autopilot User Guide. `https://docs.px4.io/`, 2025.