Sliding Mode Consensus Control for Multi-Agent Systems

Omar Abdelwaly
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
omar.abdelwaly@student.guc.edu.eg

Waleed Abdelrahman
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
waleed.abdelrahman@student.guc.edu.eg

Abdallah Falah
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
abdallah.falah@student.guc.edu.eg

Mohamed Elattar
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
mohamed.elattar@student.guc.edu.eg

Fouad Elsheikh
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
fouad.elsheikh@student.guc.edu.eg

***Abstract*—Multi-agent systems (MAS) are increasingly being employed for a number of purposes. However, coordinating several agents in a three-dimensional (3D) environment is difficult, particularly in the presence of disturbances and unpredictable environmental conditions. This paper describes a sliding mode consensus control approach for robust coordination among several agents. The control mechanism assures that all of the system's agents successfully achieve the Leader states without diverging, even under varying operational scenarios, ensuring optimal performance and safety.**

I. Introduction

Applications requiring the coordination of several drones, such exploration and surveillance missions, have seen an increase in the use of multi-agent systems (MAS). However, because of uncertainties, reaching consensus in these kinds of systems is difficult. Since sliding mode control (SMC) is well-known for its robustness to disturbances and model uncertainties. this makes it a suitable candidate for achieving consensus in MAS. This project addresses the problem of consensus control for drones, accounting for uncertainties. We propose a sliding mode consensus control approach that ensures robust consensus even in the presence of disturbances and uncertainties.

II. System Description and model

A. System Description

The system consists of two drones: a **leader** drone and a **follower** drone. Each drone's state vector includes acceleration, velocity, and position. The state vector for the follower X_f and the leader X_l are defined as:

$$X_f = \begin{bmatrix} x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix}, \quad X_l = \begin{bmatrix} x_l \\ \dot{x}_l \\ \ddot{x}_l \end{bmatrix}$$

The drone model used in this work is based on the system dynamics presented in [1], which describes the behavior of drones requiring control of position, velocity, and acceleration. This model incorporates a high-order Sliding Mode Control (SMC) framework to ensure robust performance in the presence of disturbances and uncertainties. The high-order SMC approach necessitates controlling not only the position but also the velocity and acceleration to achieve smooth and precise tracking of desired trajectories.

To model the entire system of both drones as a network, the combined state vector is:

$$X = \begin{bmatrix} X_f \\ X_l \end{bmatrix} = \begin{bmatrix} x_f \\ \dot{x}_f \\ \ddot{x}_f \\ x_l \\ \dot{x}_l \\ \ddot{x}_l \end{bmatrix}$$

The combined dynamics are given by:

$$\dot{X} = \begin{bmatrix} A_f & 0 \\ 0 & A_l \end{bmatrix} \begin{bmatrix} X_f \\ X_l \end{bmatrix} + \begin{bmatrix} B_f & 0 \\ 0 & B_l \end{bmatrix} \begin{bmatrix} u_f \\ u_l \end{bmatrix}$$

The error between the follower and the leader is:

$$\bar{\epsilon}(t) = X_f - X_l = \begin{bmatrix} \bar{x} \\ \bar{v} \\ \bar{a} \end{bmatrix}$$

where $\bar{x} = x_f - x_l$, $\bar{v} = \dot{x}_f - \dot{x}_l$, and $\bar{a} = \ddot{x}_f - \ddot{x}_l$ represent the position, velocity, and acceleration errors, respectively.

III. Closed-loop control

A. Controller

The **Sliding Mode Control (SMC)** is a robust control strategy designed to drive the error between the follower and leader drones to zero, ensuring the follower tracks the leader's position, velocity, and acceleration. The error dynamics are defined by the error vector $\bar{\epsilon}(t) = X_f - X_l$, where

$$X_f = \begin{bmatrix} x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix}, \quad X_l = \begin{bmatrix} x_l \\ \dot{x}_l \\ \ddot{x}_l \end{bmatrix}.$$

The **Sliding Surface** $S(t)$ is defined as the error itself:

$$S(t) = \bar{\epsilon}(t) = \begin{bmatrix} \bar{x} \\ \bar{v} \\ \bar{a} \end{bmatrix} = \begin{bmatrix} x_f - x_l \\ \dot{x}_f - \dot{x}_l \\ \ddot{x}_f - \ddot{x}_l \end{bmatrix}.$$

The control input for the follower drone $u_f(t)$ is designed as:

$$u_f(t) = -K\bar{\epsilon}(t) - \mu\|\bar{\epsilon}(t)\| \operatorname{sgn}(S(t)) + x_0(t)$$

where:

- $K = \begin{bmatrix}1.6462 & 0.5522 & 0.5522\end{bmatrix}$ is a gain matrix, ensuring that the error decreases as the system moves towards the sliding surface.
- $\mu = 0.1$ is a positive constant that determines the aggressiveness of the sliding mode, ensuring rapid convergence to the sliding surface.
- $\|\bar{\epsilon}(t)\|$ is the norm of the error vector, quantifying the magnitude of the error.
- $\operatorname{sgn}(S(t))$ is the signum function, which forces the system to converge to the sliding surface by applying corrective control whenever the error deviates.
- $x_0(t)$ is the current state of the leader, which acts as the reference for the follower's motion.

This control law was derived in [2] and was modified by adjusting the gain matrix K and the sliding mode term. Additionally, we introduced a third control term specifically designed to handle the acceleration dynamics of the follower drone. Once the system reaches the sliding surface, the error is driven to zero, ensuring that the follower tracks the leader's state. The sliding mode term $-\mu\|\bar{\epsilon}(t)\| \operatorname{sgn}(S(t))$ forces the system to stay on the sliding surface, eliminating steady-state error in the presence of disturbances or model uncertainties. The term $x_0(t)$ ensures that the follower tracks the leader's trajectory.

This control law guarantees that the follower drone will effectively track the leader's trajectory, even in the presence of uncertainties and disturbances.

B. Hybrid States of the System

The control system for the drones operates in a hybrid framework, transitioning between two distinct operational states: take-off and circular trajectory. Each state has specific control objectives and associated equations, as described below:

1. Take-off State

In the **take-off state**, the drones ascend vertically from their initial position, following a predefined trajectory for both position and velocity. The altitude $z_f(t)$ is defined as:

$$z_f(t) = 0.5t,$$

where t is time in seconds. The velocity during the take-off phase is a constant:

$$\dot{z}_f(t) = 0.5\,\text{m/s}.$$

In this state, the position $z_f(t)$ and velocity $\dot{z}_f(t)$ are controlled independently. The position trajectory ensures the drone reaches a specific altitude smoothly, while the velocity remains constant to maintain a stable ascent.

The input controls during this phase are given directly as:

$$u_z(t) = z_f(t), \quad \text{and} \quad v_z(t) = \dot{z}_f(t),$$

where:

- $u_z(t)$ is the altitude input, directly following the position equation $z_f(t) = 0.5t$,
- $v_z(t)$ is the velocity input, set to $0.5\,\text{m/s}$.

This approach ensures that the drone ascends with a fixed velocity and reaches the desired altitude trajectory smoothly and consistently.

2. Circular Trajectory State

In the **circular trajectory state**, the drones follow a predefined circular trajectory in the horizontal plane. The position of the leader drone is calculated as follows:

$$\begin{aligned} x_l(t) &= x_{\text{start}} + r\cos(t), \\ y_l(t) &= y_{\text{start}} + r\sin(t), \\ z_l(t) &= z_{\text{start}}, \end{aligned}$$

where:

- $x_{\text{start}}, y_{\text{start}}, z_{\text{start}}$ are the position coordinates of the leader drone at the time of state transition,
- r is the radius of the circular trajectory.

The velocity of the leader drone is defined independently as:

$$\begin{aligned} \dot{x}_l(t) &= -r\sin(t), \\ \dot{y}_l(t) &= r\cos(t), \\ \dot{z}_l(t) &= 0. \end{aligned}$$

Summary

The hybrid system transitions between these states based on mission requirements:

- **Take-off:** Drones ascend to a target altitude.
- **Circular Trajectory:** Drones follow a circular path with a specified radius and angular velocity.

This state-based design ensures smooth transitions and coordinated operation of the drones during the mission.

IV. Hardware

A. Component Selection and Initial Prototyping

The development of the drone hardware began with the selection of key components:

- **Microcontroller**: ESP32
- **IMU Sensor**: MPU6050

The first step was assembling these components on a breadboard to test their functionality with the ArduPilot software. This phase ensured that the microcontroller and IMU sensor worked seamlessly together before moving on to the next stage.

B. PCB Design and Circuit Integration

After confirming the circuit's functionality on the breadboard, the design of the PCB began. The circuit included:

- Battery charging module
- Motor driver circuit
- Connections for the ESP32 and MPU6050

In Figure 1 the PCB is mostly empty as all the modules are put on female and male headers to be able to switch out any component if something happens.

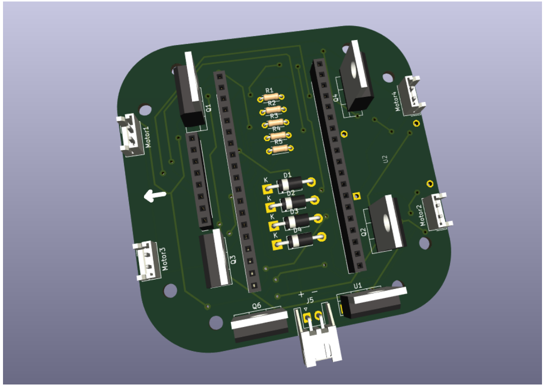

Fig. 1. 3D Render of the PCB Design

C. Drone Frame Design and Fabrication

The drone frame was designed using SolidWorks. The iterative design process involved numerous trials, each followed by 3D printing and testing until a final, optimized frame design was achieved. This included:

- Designing individual components like the arms
- Assembling the complete frame in SolidWorks for structural analysis

The final design resulted in a compact drone as seen in figure 2, with a total size of 13 cm. The 75 mm propellers were chosen to provide a balance between stability and maneuverability.

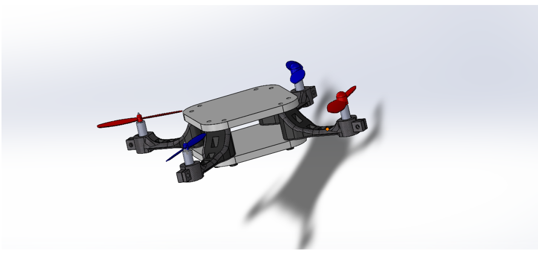

Fig. 2. SolidWorks Model of the Drone

D. Final Assembly and Testing

Once the PCB and frame were ready, the drone was assembled. The components were soldered onto the PCB, and the hardware was integrated into the frame. Initial testing confirmed that the drone was operational, with all systems functioning as expected.

V. Simulations and Results

In this section, we present the results of the proposed system, including the consensus behavior of the drones, the trajectory following, and the output of the hybrid system.

A. Consensus of Two Drones in X and Y

The consensus of the two drones in the X- and Y-directions is illustrated in Figures 3 and 4. This demonstrates the ability of the proposed controller to achieve consensus among agents.

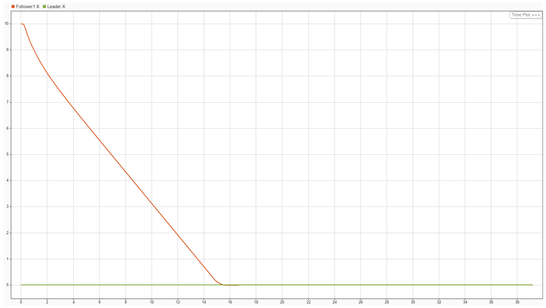

Fig. 3. Consensus behavior of the two drones in the X-direction.

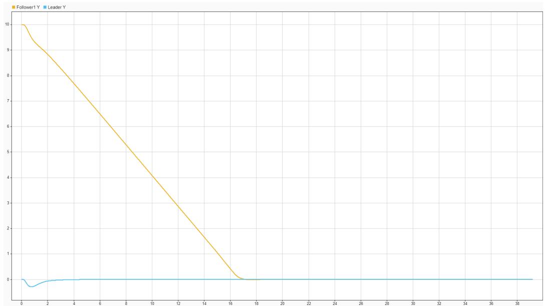

Fig. 4. Consensus behavior of the two drones in the Y-direction.

B. Drone Following a Helical Trajectory

The trajectory of a single drone following a helical path in the X-, Y-, and Z-directions is shown in Figures 5, 6, 7 and 8. These results illustrate the robustness of the control system in maintaining a complex path.

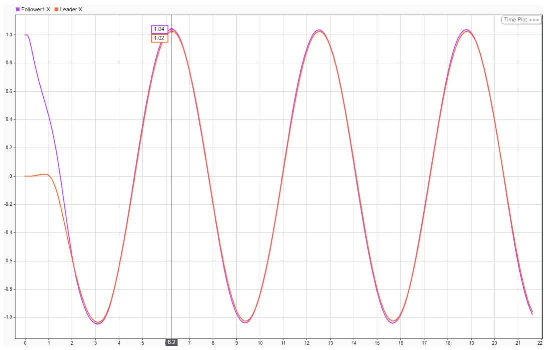

Fig. 5. X- direction of the helical trajectory.

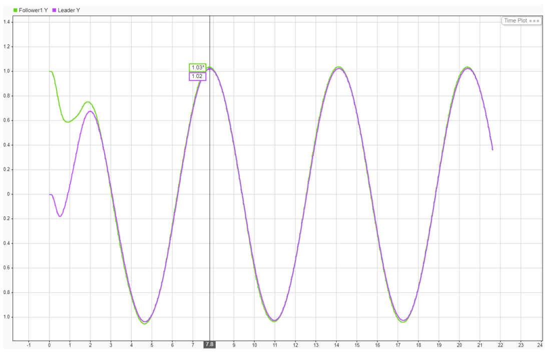

Fig. 6. Y-direction of the helical trajectory.

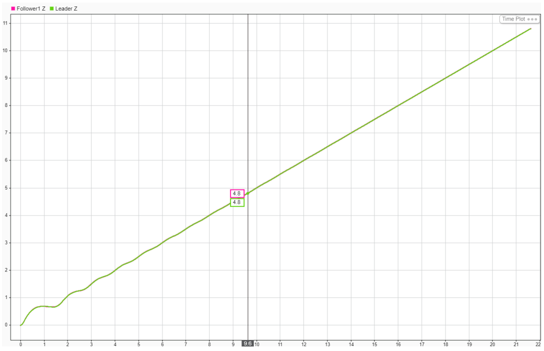

Fig. 7. Z-direction of the helical trajectory.

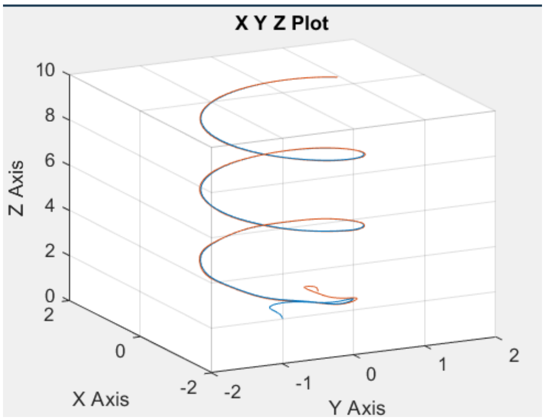

Fig. 8. 3D plot of the helical trajectory.

C. *Helical Trajectory with Distance* d *Between Two Drones*

Figures 9, 10, 11 and 12 show the helical trajectory followed by two drones, with a maintained distance d between them. This demonstrates the system's ability to manage inter-drone distances while following a trajectory.

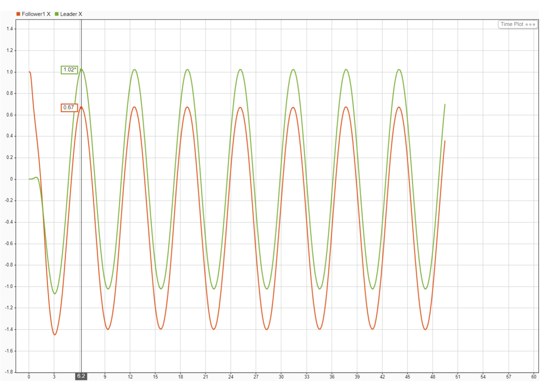

Fig. 9. X- direction of the helical trajectory with a maintained distance d between two drones.

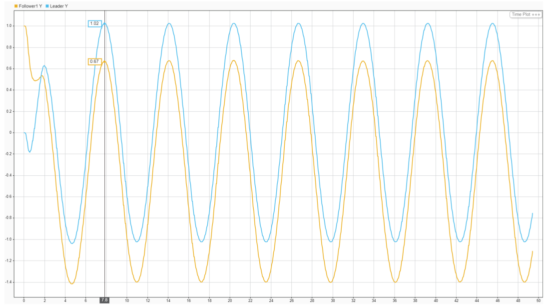

Fig. 10. Y-direction of the helical trajectory with a maintained distance d between two drones.

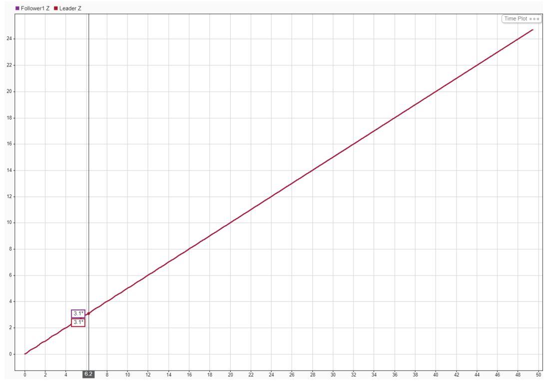

Fig. 11. Z-direction of the helical trajectory with a maintained distance d between two drones.

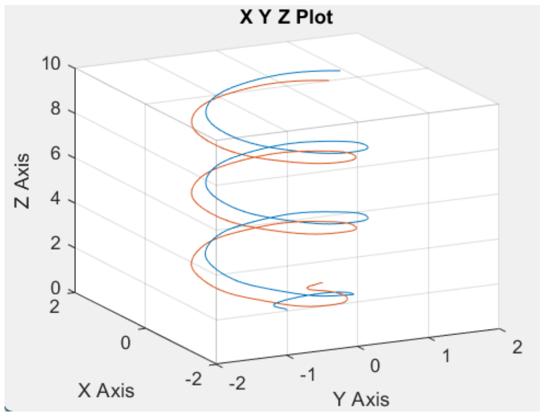

Fig. 12. 3D plot of the helical trajectory with a maintained distance d between two drones.

D. *Hybrid System State Output*

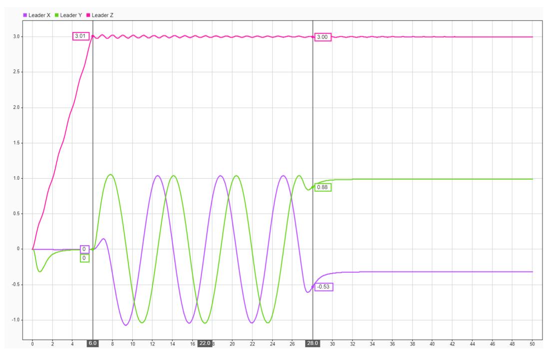

Fig. 13. X- ,Y- and Z- directions of the hybrid system state outputs.

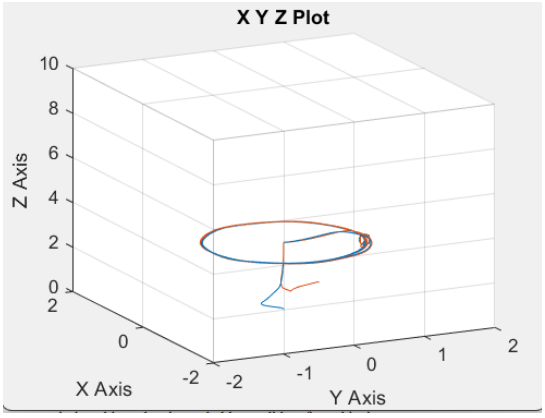

Fig. 14. 3D plot of the hybrid system state outputs.

The hybrid system state outputs in the X-, Y-, and Z-directions are shown in Figure 13 and the corresponding 3D plot is presented in Figure 14. These results validate the performance of the proposed hybrid control system.

REFERENCES

[1] S. Samy, Z. Al Wareh, and A. El-Badawy, "Robust adaptive nonsingular fast terminal sliding mode control for tracking aggressive quadrotor trajectories," in *2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 274–279, 2024.

[2] J. Xu and Y. Niu, "Sliding mode consensus control for multi-agent systems under multi-node round-robin protocol," *Franklin Open*, vol. 5, p. 100052, 2023.