

Article

Sliding Mode Controller with Disturbance Observer for Quadcopters; Experiments with Dynamic Disturbances and in Turbulent Indoor Space

Yutao Jing ^{1,†} , Adam Mirza ^{1,†}, Rifat Sipahi ¹  and Jose Martinez-Lorenzo ^{1,2,*}

¹ Department of Mechanical & Industrial Engineering, Northeastern University, Boston, MA 02115, USA; jing.yut@northeastern.edu (Y.J.); mirza.ad@northeastern.edu (M.A.); r.sipahi@northeastern.edu (R.S.)

² Department of Electrical & Computer Engineering, Northeastern University, Boston, MA 02115, USA

* Correspondence: j.martinez-lorenzo@northeastern.edu

† These authors contributed equally to this work.

Abstract: In this study, a sliding mode surface controller (SMC) designed for a quadcopter is experimentally tested. The SMC was combined with disturbance observers in six degrees of freedom of the quadcopter to effectively reject external disturbances. While respecting stability conditions all control parameters were automatically initialized and tuned using a simulation-based offline particle swarm optimization (PSO) algorithm, followed by onboard manual fine-tuning. To demonstrate its superiority, the SMC was compared with a PSO-optimized PID controller in terms of agility, stability, and the accurate tracking of hover, rectangular, and figure-eight pattern trajectories. To evaluate its robustness, the SMC controller was extensively tested in a small, enclosed, turbulent space while being subjected to a series of external disturbances, such as hanging payloads and lateral wind.

Keywords: sliding mode control; disturbance observer; unmanned aerial vehicles; quadcopter; nonlinear control; particle swarm optimization; experimental validation; payload; trajectory tracking; wind rejection



Citation: Jing, Y.; Mirza, A.; Sipahi, R.; Martinez-Lorenzo, J. Sliding Mode Controller with Disturbance Observer for Quadcopters; Experiments with Dynamic Disturbances and in Turbulent Indoor Space. *Drones* **2023**, *7*, 328. <https://doi.org/10.3390/drones7050328>

Academic Editors: Zhaokui Wang, Xiumin Diao and Lin Zhang

Received: 21 April 2023

Revised: 16 May 2023

Accepted: 18 May 2023

Published: 20 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned aerial vehicles (UAVs), specifically those with multirotors, are being used in a growing number of applications that require robust and reliable performance under a variety of operating conditions. These applications cover a range of civil use cases, such as product delivery [1], structure inspection [2], aerial photography [3], agriculture and search [4], disaster relief [5], and rescue [6,7]. Additionally, UAVs are used in a variety of defense applications, including reconnaissance [8], threat identification [9], and surveillance [10].

Most commercially available multirotor UAVs, such as quadcopters, use PID (proportional integral derivative) controllers [11]. These controllers have a simple and linear structure that can be easily tuned. However, due to their simple nature and low complexity, these controllers struggle under noise, latency, and external disturbances. In response to these shortcomings, researchers have modified the PID controller structure to be more robust to system disturbances and unknown parameters. For example, in [12], Goel et al. provide a numerical investigation of the performance of the adaptive autopilot on a quadcopter and propose an adaptive PID control scheme designed for a UAV with unknown dynamics. Dong and He [13] propose fuzzy logic PID controllers for quadcopters in simulations: a control method combining PID-ILC (iterative learning control) and fuzzy control to achieve robustness to disturbances and model uncertainties. To further improve the robustness of the controller against wind disturbances, Zhou et al. [14] designed a cascade inner-loop PID angular rate control and an outer-loop PID attitude control for the UAV. The PID gain automatically tuned by reinforcement learning neural networks is

also discussed in the field of UAV control [15], where the proposed method demonstrates effectiveness by both software- and hardware-in-the-loop simulations.

Some researchers have moved beyond PID controllers to more complex and robust control schemes. Several adaptive and nonlinear controllers are proposed, including the nonlinear adaptive intelligent controller [16], which incorporates fuzzy logic and neural networks, model predictive controllers (MPC) [17], to provide a nonlinear model predictive control law, and adaptive backstepping controllers [18,19], which can be used for velocity field following and timed trajectory tracking. Sliding mode control offers a robust way to maintain the stability of unknown UAV dynamics [20]. Indeed, sliding mode surface controller (SMC) can overcome uncertainties in the system and provide good control on nonlinear systems. Importantly, one can incorporate explicitly into SMC the upper bounds of uncertainties in the dynamics. Lenaick et al. [21] theoretically show SMC's characteristics of a fast response, simple operation, and robustness to external environmental disturbances.

The basic idea of SMC is to design a switching hyperplane (sliding surface) according to the dynamic characteristics of the system and then drive the tracking error from outside the hyperplane to this hyperplane by sliding mode controllers. Once the states reach the hyperplane, the controller then guides the error dynamics to zero. R. Guruganesh [22] verifies a second-order sliding mode approach and implements it on a fixed-wing micro aerial vehicle (MAV). Instead of multicopters, they experimentally show that the performance of the sliding mode controller exceeds a benchmark classical controller with a fixed-wing UAV. Another work by En-Hui in [23] demonstrates the effectiveness and robustness of a proposed SMC control method through simulations. Currently, most of the research related to sliding-mode-controlled quadcopters relies on computer simulations to validate the proposed methods and theories. Although some attempts built real-world UAV prototypes to achieve trajectory tracking tasks, such as a rectangular trajectory [24], the prototypes were implemented in combinations of PID and SMC, i.e., partial SMC attitude controllers without disturbance observers; meanwhile, the position is controlled by a traditional PID, which leads to potential performance degradation in flight dynamics when exposed to disturbances.

In contrast to PID controllers, however, tuning SMC parameters is less intuitive. Particle swarm optimization (PSO) has been used to aid in the design and tuning of various types of controllers, including PID [25], SMC [26], and fuzzy logic controllers [27]. PSO is a computationally inexpensive optimization algorithm when dealing with nonlinear functions. It works by tracking a set of particles through the parameter space of interest, searching for the best solution parameters that optimize closed-loop performance as quantified by certain metrics. The swarming behavior of the particles promotes exploring regions adjacent to both a particle's local best solution and the swarm's current global best solution [28].

While there are a number of options for control design, one should also consider whether to design attitude controllers based on quaternions (see, e.g., [29]) or on Euler angles (see, e.g., [30]). Quaternions are particularly useful for describing rotations in three dimensions [31], with the main advantage of avoiding the problem of gimbal lock that may otherwise occur with Euler angles when two of the rotation axes align [32], causing a loss of one degree of freedom. However, Euler angles are more intuitive, representing rotations around the principal axes (X, Y, and Z) or roll, pitch, and yaw, respectively. Most sensors directly provide the measurements in Euler angles, making their integration into control systems straightforward. Moreover, Euler angles can be more computationally efficient in certain situations [32]. Quaternion-based controllers are not short of advantages; for example, previous work demonstrates their promise in situations where aggressive drone maneuvers might be preferred [33], including experimental results [34] and combining them with SMC [35]. Given the importance of continuity from our previous work [26], we utilize here the SMC controller from the cited study with Euler angles.

Based on our previous research [26] regarding a PSO-optimized SMC with disturbance observers (Figure 1: phases (1) to (3)), we have already developed the control theory

and validated the controller using a simulated PX4-conducted (an open-source autopilot) quadcopter. To summarize the work, we firstly designed the SMC controllers as well as corresponding disturbance observers in MATLAB/SIMULINK for the theory validation and then optimized SMC parameters using an offline PSO method. We also built a PX4-conducted quadcopter simulator to validate the optimized SMC. In simulations, we compared this SMC with several PSO-optimized PID controllers and showed its superiority. The PSO-optimized SMC with disturbance observers facilitated more accurate and faster UAV adaptation against disturbances in the environment.

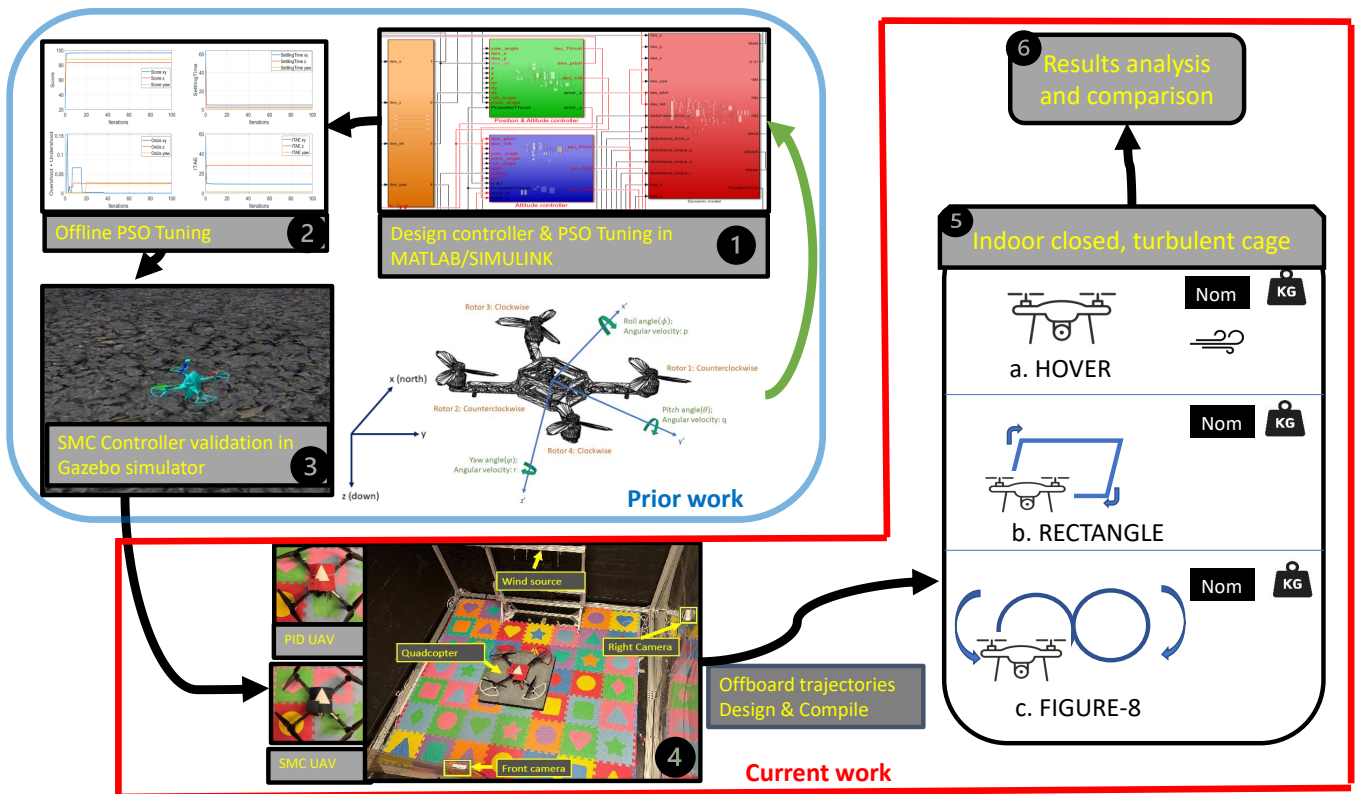


Figure 1. Overall structure of this paper.

To further validate our proposed SMC controller, in this work, we designed a series of indoor experiments for an actual quadcopter as shown in Figure 1 phases (4) to (6). In order to be compatible with the actual UAV as well as its autopilot, we upgraded the proposed control theory, improved the initialization and tuning of the PSO algorithm, optimized the control signal mixing algorithm, and tested the fine-tuned SMC parameters. Unlike most of the current theoretical research, here we validate not only the feasibility and stability of SMC through nominal flights but also the reliability of SMC against dynamic disturbances. These disturbances are designed as a series of imbalanced force/torque loads and turbulence in narrow and closed spaces as shown in Figure 2. These disturbances were applied to the UAV to mimic the most likely future indoor scenarios and to evaluate how well SMC performs.

To the best of our knowledge, most studies on SMC rely on computer simulations and remain in the theoretical phase [36–43], while implementation of SMC on actual quadcopters is rare and limited [44], let alone extensive experiments in extreme environments. This is likely because the hardware implementation is quite involved and bridging theory with practice is not straightforward. First, the frequency of the control signal must not be too low to ensure the stability of the control, so the signal synchronization process becomes important. In this study, we use the PX4 open-source autopilot, which provides fixed-rate and continuous feedback of sensor signals and actuator signal outputs. However, due to

hardware limitations, the control frequency cannot be too high, so the controller frequency and the fundamental sampling frequency are fixed at 250 Hz. Second, we need to filter sensor data within a reasonable bandwidth due to unavoidable sensor uncertainties, such as noise and signal gaps. Here, we use an extended Kalman filter (EKF) estimator with a cutoff frequency of 30 Hz for the IMU/GYRO sensor to reduce computational consumption and UAV vibration. Third, due to the delay and response time of the actuator, which can be negligible in simulations but critical when dealing with real UAVs, we choose a reasonable PWM update rate, which is 400 Hz, based on the type of motors utilized. At the same time, the mixing process of the actuator signals is unavoidable for the hardware implementation, without which the PWM signals can easily exceed or fall below the physical limit values, causing instability. This can be also an issue for the sensitivity of disturbance observers, which estimate dynamic disturbances and feedback these estimates to the main control loop for disturbance rejection. Therefore, including a reasonable control signal mixing is critical to ensure the aerial stability of the UAV in the real world. Fourth, the PSO algorithm is not implemented in real time because of its high computation cost and time-consuming process. Instead, the PSO tuning results are based on a purely mathematical model and take into account the theoretical stability of the SMC. In more detail, the PSO is run and iterated in conjunction with SIMULINK, after which we encode and embed PSO-tuned parameters onboard the UAV. Note that not all PSO-tuned parameters are applicable to an actual UAV; thus, a further manual tuning process is required. We modify the parameters judiciously via an initial calibration phase first, according to different orientations (pitch, roll, and yaw) and different directions (X , Y , and Z) of the quadcopter. Although we have simultaneously developed an in-flight remote SMC parameter adjustment system that relies on a lightweight wireless communication protocol called MAVLink, we do not recommend adjusting parameters in-flight because of the possibility of improper adjustment leading to an accidental crash. Instead, we start a brand new flight after the hardware coding and updating of each parameter and finally evaluate the flight performance in terms of settling time, overshoot, and rise time based on the flight log data. Finally, after each iteration of parameter tuning or controller modification, we use the MATLAB code generator to automatically generate the C code and upload the new firmware to the actual UAV microcontroller board.

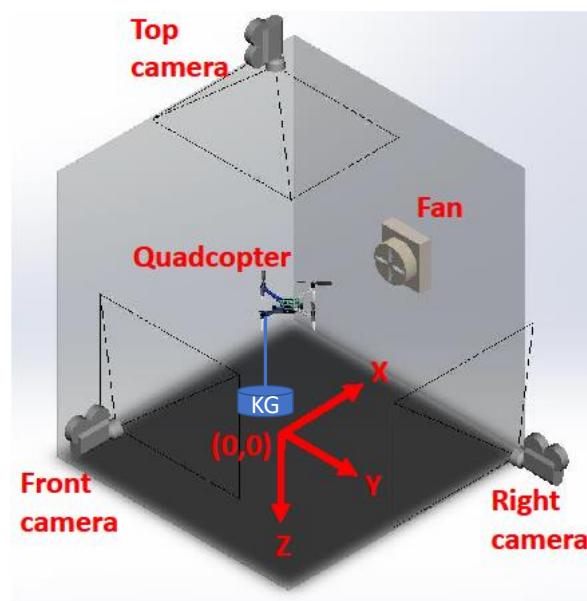


Figure 2. Experiments in a narrow, closed, turbulent indoor space.

In view of the above discussions, the novelty of our work can be summarized as follows: We first demonstrate what additional steps are needed when transitioning from [26]

the particle swarm optimization (PSO)-tuned sliding mode control (SMC) with a disturbance observer (DOB) to real-time control, hardware implementation, and experiments. This is a topic of strong interest but with little guidance in the open literature. Next, we demonstrate how the drone performs at the face of a set of complex disturbances, such as wind and unbalanced weights under SMC, and in particular show how a combination of these disturbances can still be addressed by SMC—an experimental condition that does not exist in the literature to the best of our knowledge. Furthermore, we complement the SMC experiments with those by PID controllers as industry-standard benchmarks, for comparisons with respect to SMC. Lastly, with the use of the DOB, we demonstrate how to estimate wind direction and wind forces in real time—a critical capability in monitoring the environment and atmospheric conditions. These results add to the body of literature rapidly growing and will help support future research into the robust control of drones in challenging environments.

In the next section, the system is defined and the equations of motion for the quadcopter are summarized. In Section 3, the SMC control laws are developed for the attitude and position dynamics, a disturbance observer is designed, improvements in the control signal mixing process are explained, and the tuning process on the physical UAV is outlined. Next, in Section 4, the experimental setup is described, the step response of the SMC is evaluated, and the trajectory following capabilities of the SMC are compared to a commercial PID controller. Each controller is subject to a variety of disturbance conditions in a narrow, closed, and turbulent chamber.

2. System Definition

Equations of Motion

In this manuscript, the north–east–down (NED) coordinate frame is used for the UAV positioning, and Euler angles are used for the attitude control. The coordinate system relative to the S500-frame UAV is shown in Figure 3, including $X, Y, Z, p, q, r, \phi, \theta,$ and ψ .

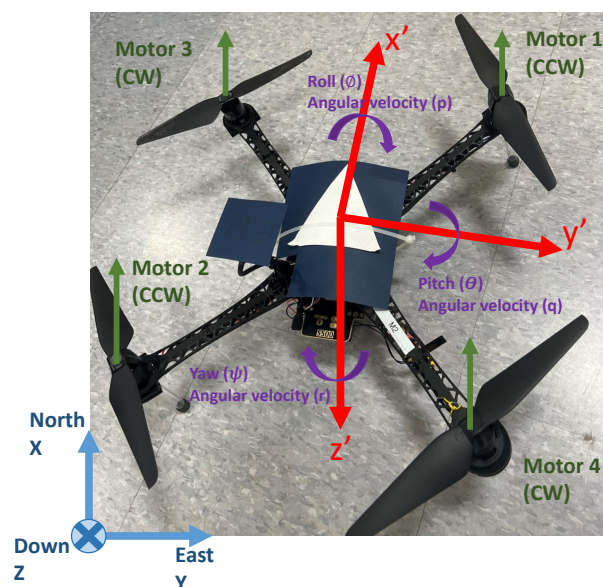


Figure 3. Euler angles and coordinate system of the S500-frame quadcopter used in the experiments.

As seen from Figure 3, coordinates $X, Y,$ and Z define the global NED coordinate system; $x', y',$ and z' are in the UAV’s body coordinate system; $p, q,$ and r are angular velocities in the UAV body frame about $x', y',$ and z' , respectively; and $\phi, \theta,$ and ψ are Euler angles for roll, pitch, and yaw, respectively.

The equations of motion (see [26,45,46]) are given by the following:

$$\begin{cases}
 \dot{x} = -\frac{1}{m}[(\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \cdot (T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{x} \cdot |\dot{x}|] \\
 \dot{y} = -\frac{1}{m}[(-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \cdot (T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{y} \cdot |\dot{y}|] \\
 \dot{z} = -\frac{1}{m}[(\cos \phi \cos \theta)(T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{z} \cdot |\dot{z}|] + g \\
 \dot{p} = \frac{1}{J_x}[\frac{\sqrt{2}}{2}l(T_2 + T_3 - T_1 - T_4) - q \cdot r(J_z - J_y) - K_2 \cdot p \cdot |p|] \\
 \dot{q} = \frac{1}{J_y}[\frac{\sqrt{2}}{2}l(T_1 + T_3 - T_2 - T_4) - p \cdot r(J_x - J_z) - K_2 \cdot q \cdot |q|] \\
 \dot{r} = \frac{1}{J_z}[(Q_1 + Q_2 - Q_3 - Q_4) - p \cdot q(J_y - J_x) - K_2 \cdot r \cdot |r|] \\
 \dot{\phi} = p + q \cdot \tan \theta \sin \phi + r \cdot \tan \theta \cos \phi \\
 \dot{\theta} = q \cdot \cos \phi - r \cdot \sin \phi \\
 \dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}
 \end{cases} \tag{1}$$

where model parameters are defined in Table 1, based on an evaluation of the UAV CAD model and the specifications provided by the manufacturer. Here, T_{1-4} are the thrusts of each propeller and Q_{1-4} are anti-torques at each motor caused by the propellers. From the UAV model in Figure 3, each propeller generates a thrust in the $-z'$ direction and an anti-torque around $-z'$ direction given by

$$\begin{cases}
 T_i = T_{max} \cdot v_i \\
 Q_i = Q_{max} \cdot v_i
 \end{cases} \tag{2}$$

where v_i are the normalized actuator signals v_{1-4} , T_{max} is the full thrust of one motor, and Q_{max} is an anti-torque at the full thrust of one motor.

Table 1. Dynamics model parameters.

Parameter	Value	Unit	Definition
m	1.300	kg	Mass of UAV
g	9.810	m/s ²	Gravity
l	0.250	m	Distance between the center of the motor and the center of the UAV
J_x	0.015	kg · m ²	Moment of inertia in x
J_y	0.016	kg · m ²	Moment of inertia in y
J_z	0.029	kg · m ²	Moment of inertia in z
K_1	1.00	$\frac{N}{(m/s)}$	The air drag force coefficient
K_2	0.025	$\frac{N \cdot m}{(rad/s)}$	The air drag torque coefficient
T_{max}	9.600	N	The full thrust of one motor
Q_{max}	0.150	N · m	The anti-torque at the full thrust of one motor

Utilizing the thrusts T_i and anti-torques Q_i , we can define the normalized control signals τ_T , τ_P , τ_R , and τ_Y that affect the thrust, pitch motion, roll motion, and yaw motion of the UAV, respectively:

$$\begin{aligned}\tau_T &= \frac{T_1 + T_2 + T_3 + T_4}{4T_{max}} \\ \tau_R &= \frac{T_2 + T_3 - T_1 - T_4}{4T_{max}} \\ \tau_P &= \frac{T_1 + T_3 - T_2 - T_4}{4T_{max}} \\ \tau_Y &= \frac{Q_1 + Q_2 - Q_3 - Q_4}{4Q_{max}}\end{aligned}\quad (3)$$

where the thrust signal τ_T ranges from 0 to 1, while the other signals all range from -1 to 1 . It is important to note that the control signals τ_T , τ_P , τ_R , and τ_Y are prescribed by SMC (see Section 3) and are defined in the “body” frame of the drone. These signals appear in (p, q, r) expressions in Equation (1) and are converted to the inertial frame through trigonometric identities that can be found in the expressions of $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ in Equation (1).

The next step is to utilize the equations of motion of the UAV to design the control laws such that the UAV can track trajectories in a stable manner. This is discussed in the next subsection.

3. Control Design

3.1. Sliding Mode Control and Disturbance Observer

Sliding mode control (SMC) is a type of variable structure control with a nonlinear control law. In this section, we first present a simple example without disturbances to show the controller derivation process and then provide the details of SMC design for the UAV in Appendix A.

Consider that we have a nonlinear system of the form $\ddot{w} = F(w) + \beta u$, where $w \in [x, y, z, \phi, \theta, \psi]$ is the continuous UAV state, $F(w)$ is a nonlinear function with continuous derivatives with respect to each state, and β is a known constant. First, the controller u is defined as $u = u_{sw} + u_o$, where u_o is the nominal part and u_{sw} is the switching part. The goal is to track a desired, continuous, sufficiently smooth signal w_{des} , i.e., the tracking error should satisfy $e_w = w - w_{des} \rightarrow 0$ as $t \rightarrow \infty$.

In SMC, a sliding surface is defined with the error dynamics of the system. While this surface can be defined in various ways, here, the sliding surface is taken to be in the form $s_w = \alpha_w e_w + \dot{e}_w$, where $\alpha_w > 0$ can be tuned. To perform the stability analysis, we choose the Lyapunov function $V(s) = \frac{1}{2}s_w^2 > 0$, and therefore $\dot{V}(s) = s_w \dot{s}_w < 0$ must hold as the sufficient condition for the system to be stable. Given that $\dot{s}_w = \alpha_w \dot{e}_w + \ddot{e}_w$, we can write

$$\begin{aligned}\dot{V}(s) &= s_w \dot{s}_w = s_w (\alpha_w \dot{e}_w + \ddot{e}_w) \\ &= s_w (\alpha_w \dot{e}_w + \ddot{w} - \ddot{w}_{des}) \\ &= s_w (\alpha_w \dot{e}_w + F(w) + \beta u - \ddot{w}_{des}) \\ &= s_w (\alpha_w \dot{e}_w + F(w) + \beta u_o + \beta u_{sw} - \ddot{w}_{des})\end{aligned}\quad (4)$$

Next, the nominal terms in the last line above can be cancelled out by proposing the nominal control law as

$$u_o = \frac{1}{\beta} (-F(w) + \ddot{w}_{des}) \quad (5)$$

Hence, with the following switching control law,

$$u_{sw} = \frac{1}{\beta}(-E_1 \operatorname{sgn}(s) - E_2 s - E_3 \dot{e}_w) \tag{6}$$

where $E_1, E_2,$ and E_3 are positive values, and \dot{V} becomes

$$\dot{V}(s) = s[(\alpha_w - E_3) \cdot \dot{e}_w - E_1 \cdot \operatorname{sgn}(s) - E_2 \cdot s] \tag{7}$$

It is easy to show that by selecting $E_1, E_2,$ and E_3 as

$$\begin{cases} E_3 - \alpha_w = 0 \\ E_1 > 0 \\ E_2 \geq 0 \end{cases} \tag{8}$$

we can prove that $\dot{V} < 0$ always holds. This means the controller $u = u_{sw} + u_o$ will promote the tracking error $e_w = w - w_{des} \rightarrow 0$ as $t \rightarrow \infty$.

To avoid actuator chattering, we next replace the sgn operation with a saturation function as is the practice in the literature [47,48]. Specifically,

$$\operatorname{sat}\left(\frac{s}{\epsilon}\right) = \begin{cases} \frac{s}{\epsilon}, & |s| < \epsilon \\ \operatorname{sgn}\left(\frac{s}{\epsilon}\right) = \begin{cases} 1, & s \geq \epsilon \\ -1, & s \leq -\epsilon \end{cases}, & |s| \geq \epsilon \end{cases} \tag{9}$$

where $\epsilon > 0$ is a small user-defined constant value. Thus, the final form of the control law reads

$$u = \frac{1}{\beta}(-E_1 \operatorname{sat}\left(\frac{s}{\epsilon}\right) - E_2 s - E_3 \dot{e}_w - F(w) + \ddot{w}_{des}) \tag{10}$$

To enhance the surface reaching performance, we further define E_1 as a positive time-varying variable related to the instantaneous position and velocity errors in the X–Y–Z coordinates. That is, a larger position error and a larger velocity error together contribute to a larger E_1 and hence the action on the UAV toward achieving $e \rightarrow 0$. However, to smooth high-frequency system oscillations near the sliding surfaces and ensure system stability, it is useful to define the minimum positive threshold k_1 for horizontal XY movement and k_3 for vertical Z movement. In attitude control, however, we fix E_1 as a positive constant k_7 to achieve a more straightforward angular tracking. To summarize,

$$E_1 = \begin{cases} \max_{w \in x,y} (k_1, b_1 \cdot |e_w| + b_2 \cdot |\dot{e}_w|) \\ \max_{w \in z} (k_3, b_3 \cdot |e_w| + b_4 \cdot |\dot{e}_w|) \\ k_7, w \in \phi, \theta, \psi \end{cases} \tag{11}$$

where $k_1, k_3, k_7, b_1, b_2, b_3,$ and $b_4 > 0$ are tunable control gains for the sliding reaching laws. When tuning these gains, stability conditions rendered from (8) must be strictly enforced. We next define E_2 as

$$E_2 = \begin{cases} c, & w \in x, y \\ 0, & w \in z \\ 0, & w \in \phi, \theta, \psi \end{cases} \tag{12}$$

where E_2 is a positive tunable constant c only for XY, such that the $E_2 \cdot s$ term in (7) generates an exponentially decaying term to promote tracking efficiency. With the benefits of this exponential term, the reaching rate in XY has a large value at the beginning of the UAV motion, and then it decreases to zero gradually when reaching XY setpoints. Again, for altitude and attitude control, E_2 has been set to zero to reduce oscillations.

Consequently, the above general SMC design approach can be implemented one by one on X, Y, Z , roll (ϕ), pitch (θ), and yaw (ψ) motions of the UAV, paying attention to the couplings among the six degrees of freedom (see Appendix A). This effort leads to six baseline SMC control laws ($u_1, u_2, \tau_{Thrust} (\tau_T), \tau_{Roll} (\tau_R), \tau_{Pitch} (\tau_P), \tau_{yaw} (\tau_Y)$) for each DoF, respectively. The control structure of the overall flight controller is shown in Figure 4.

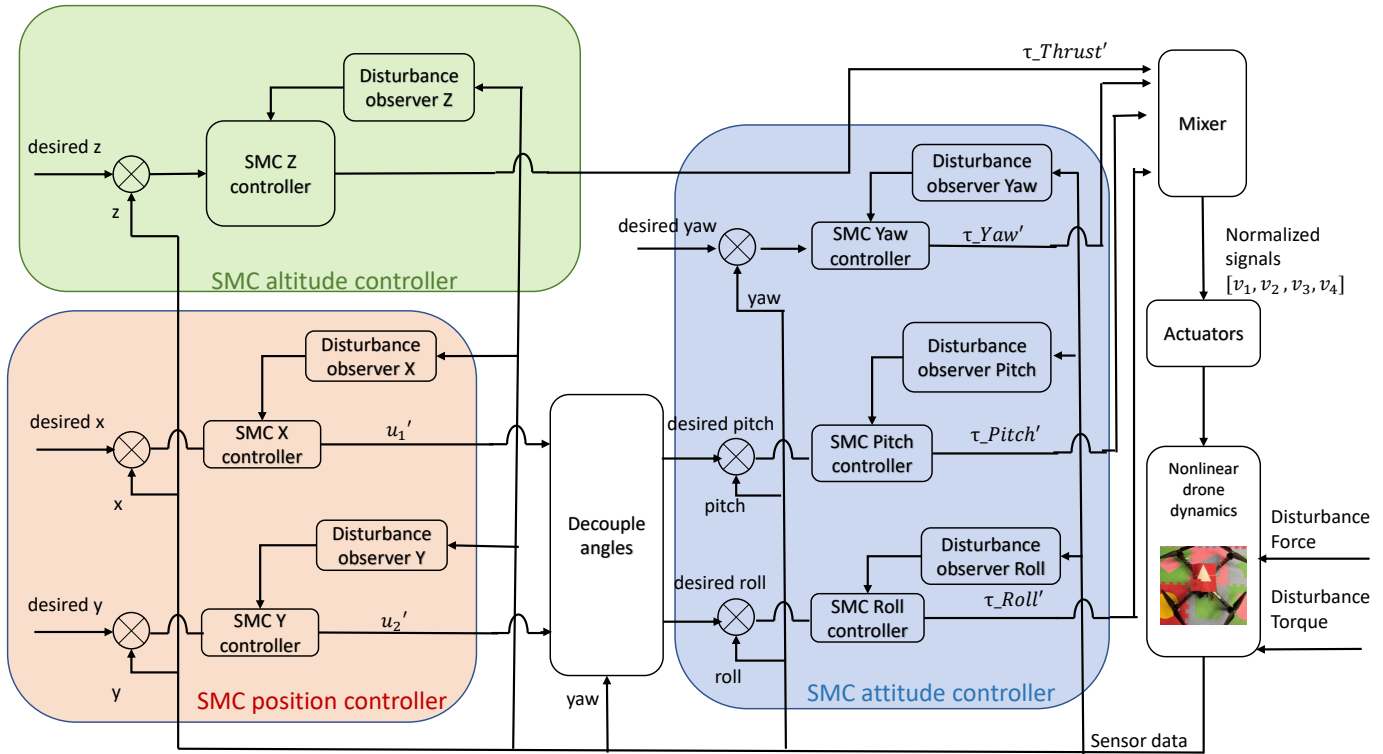


Figure 4. The quadcopter sliding mode control structure diagram.

Furthermore, each SMC is equipped with a disturbance observer to correct for external disturbances. When continuous force and torque disturbances are applied to the system dynamics, these observers actively estimate the disturbances and reject them by adding the magnitude of these disturbances in opposite polarity to the baseline control actions. Hence, thrust, roll, pitch, and yaw control signals are resized as $\tau'_T, \tau'_R, \tau'_P,$ and τ'_Y , respectively, as explained in Appendix A.

3.2. Control Signal Mixing

Control signal mixing is an unavoidable part of driving an actual quadcopter. In practice, the controller calculates the saturated control actions $\tau'_T, \tau'_R, \tau'_P,$ and τ'_Y , as determined by SMC. These actions are a composite of the four motor speeds. Hence, they must be mapped back to each motor. This is achieved by sending these control actions to the mixer, which scales and converts them to actuators' native units such as pulse-width modulation (PWM); that is, this process converts the control output signals τ into PWM signals ranging from 1000 to 2000, which is finally fed into the electronic speed controllers (ESCs). As a result, the individual motors can be actuated accordingly using a mixer (see Figure 4).

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = (1000 \cdot \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \tau'_T \\ \tau'_R \\ \tau'_P \\ \tau'_Y \end{bmatrix} - P_{min}) \cdot \frac{1}{P_{max} - P_{min}}, \quad (13)$$

where $v_1, v_2, v_3,$ and v_4 are normalized PWM signals.

As an introduction, a simple mixer architecture could be adopted, which has a straightforward diagram as shown in Figure 5. Moreover, τ_{Roll} and τ_{Pitch} inputs are saturated to the range from -0.05 to 0.05 , τ_{Yaw} ranges from -0.2 to 0.2 , and τ_{Thrust} ranges from 0.0 to 1.0 . Based on our previous experimental results, this architecture was feasible for taking off and even tracking some specific trajectories if ignoring its attendant over-saturation problems. To remedy these issues and the potential crash risk at the face of uncertainty in the environment and due to the sudden appearance of disturbances, a safer mixing approach was adopted.

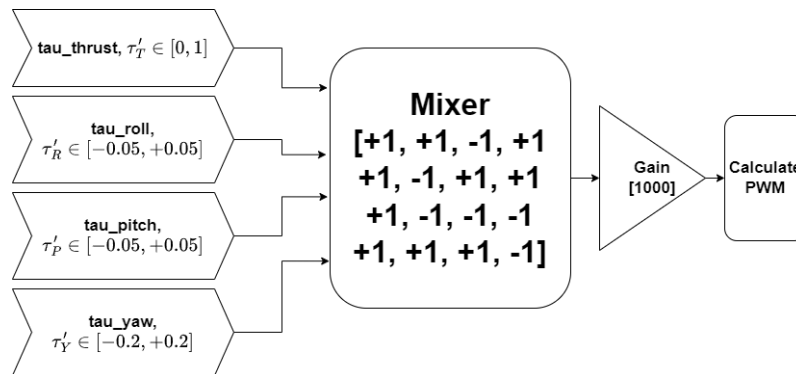


Figure 5. A simple mixer structure.

In the proposed method, the τ_{Roll} , τ_{Pitch} , and τ_{Yaw} inputs were limited to the range from -1.0 to 1.0 , and τ_{Thrust} ranged from 0.0 to 1.0 as shown in Figure 6. Compared to the previous simple mixer, although this new mixer can provide a larger control range and thus be more flexible, in practice it may saturate a motor when the motor command becomes negative. In other words, motor commands exceed the lower bound. This would happen when a low thrust τ'_T and large negative roll command τ'_R are demanded, which is physically impossible to execute unless the motors are reversible. In such a case, we can unsaturate the τ values by increasing τ'_T to boost the motors and in the meanwhile reducing the attitude control commands τ'_R , τ'_P , and τ'_Y such that none of motor commands goes below zero. Our experiments also show scenarios where motor commands exceed the upper boundary, e.g., for a high thrust τ'_T and large positive roll control factor τ'_R . In this case, we can easily unsaturate the control signals by reducing the thrust τ'_T to ensure that no motor is commanded to exceed its maximum limitation.

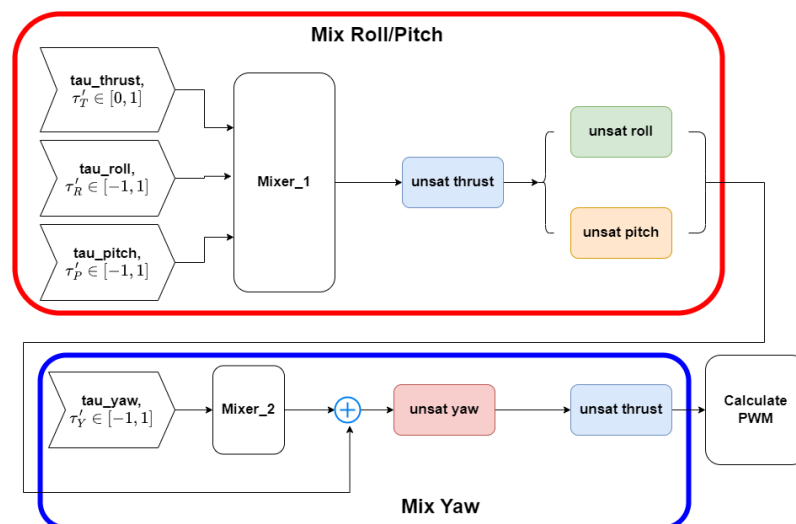


Figure 6. Improved mixer structure.

The top section of Figure 6 suggests mixing the pitch and roll but without yaw; that is because the primary task is to make sure of the UAV's position stability regardless of its heading accuracy. In this procedure, the roll, pitch and thrust are mixed first:

$$\mathbf{Mixer\ Thrust/Roll/Pitch} = \begin{bmatrix} 1 & 0.707107 & -0.707107 \\ 1 & -0.707107 & 0.707107 \\ 1 & -0.707107 & -0.707107 \\ 1 & 0.707107 & 0.707107 \end{bmatrix} \quad (14)$$

where they are scaled by gains of $1/\sqrt{2}$ to smooth the pitch and roll control effects for the symmetric quadcopter frame. After this mixer is applied, the thrust control input is unsaturated if needed. During this process, the τ_{Thrust} term will be adjusted slightly by a certain gain to meet the requirement of the maximum or the minimum actuator limitation. After tuning the thrust, the roll and pitch terms are ready to be unsaturated if necessary to reduce their angular acceleration effects.

The lower section in Figure 6 is proposed to mix the yaw using the yaw mixer matrix as follows:

$$\mathbf{Mixer\ Yaw} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \quad (15)$$

This mixer remains unchanged from Figure 5. In this study, the diagonal arms of the symmetric UAV have the same rotation direction, so the first and second rows in the yaw mixer matrix have the same values and are both positive. Similarly, the third and fourth terms have the same value but opposite directions with respect to the first two terms to ensure their total torque in the Z direction is zero. After mixing the yaw for each control output, another unsaturation function will be applied to the yaw control input to modify yaw acceleration if needed. Finally, we adjust the τ_{Thrust} again before feeding these control inputs into the PWM calculator box.

Inside the PWM calculator box, the idle-speed terms and certain offsets will be added to control signals to ensure actuators' arming states, and baseline PWM will be applied to respective motors, normally 1000; hence, PWM signals are constrained in the range of 1000 to 2000.

The necessity of improving and extending the mixer structure is that they tend to easily overflow actuators' limitations after adding the disturbance terms onto the control inputs without the mixer in Figure 6, which leads to poor UAV tracking in attitude and position, and even crashes, although the controllers are performing as desired. As shown in Figure 7, when using the simple mixer the UAV loaded with a hanging weight could not finish the pre-configured figure-eight pattern trajectory, and it crashed at the halfway point of the flight. However, after improving the mixer, the UAV could finish this complex flight pattern easily and successfully. Furthermore, it can be observed that the UAV motion becomes much smoother with fewer oscillations in PWM as shown in Figure 7. In short, the SMC control inputs are adjusted and modified through the proposed mixing algorithm to avoid accidents caused by over-saturation.

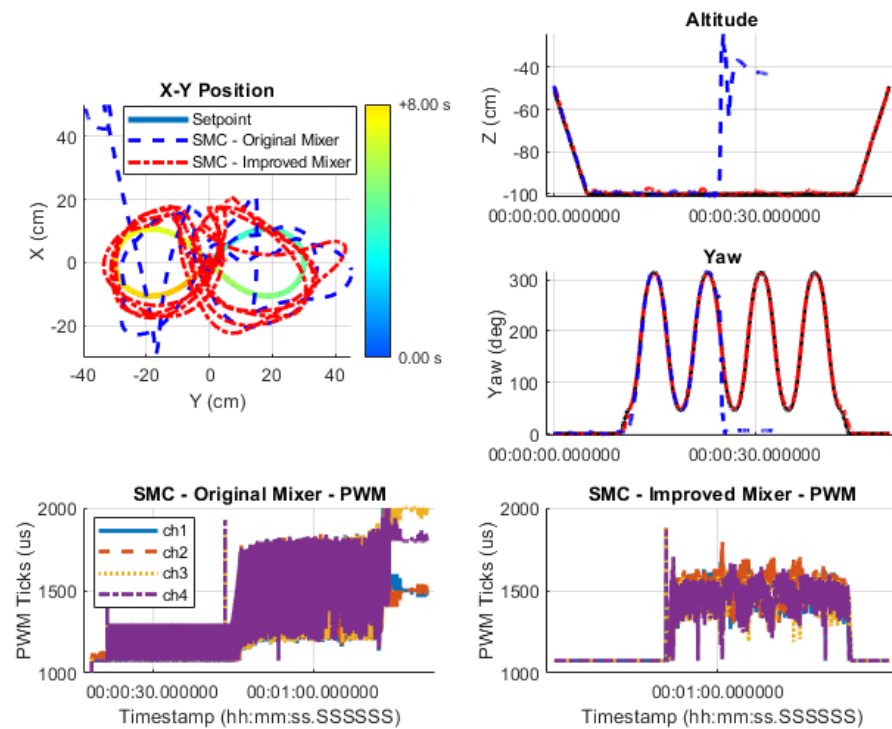


Figure 7. Figure-eight trajectory tracking experimental results: before and after mixer improvement.

3.3. Controller Tuning

The SMC designed above has over 30 tunable parameters, each having complex nonlinear interactions with the other parameters. Tuning a controller of this complexity manually would be extremely difficult and would likely result in a sub-optimal result. Conversely, this controller is sensitive to model parameters and unmodeled actuator and sensor dynamics, so a simulation-only optimization of the controller would not perform well under real-world conditions. To this end, a hybrid tuning approach was adopted. First, a PSO simulation-based tuning was performed as a coarse controller tuning approach to bring the UAV as close as possible to the desired performance. Then, a manual fine-tuning was performed through a series of experiments on the real UAV to optimize the controller performance.

PSO was selected to tune the SMC since it is simple to implement and is effective at optimizing high dimensional parameters spaces [49]. The tuning was performed only in time simulations, using step inputs along the X , Y , Z , and yaw for training and using a complex trajectory for validating. The scoring was based on an equal weight combination of sub-scores based on ITAE (integral time absolute error), overshoot, undershoot, and settling time. The scheme used for scoring was designed to minimize steady-state errors and provide a fast, smooth response to inputs. The number of particles chosen was 50 and the optimization was run for 100 time iterations, though the scores leveled out within 20 iterations. The simulated results of the SMC before and after PSO tuning are in Figure 8.

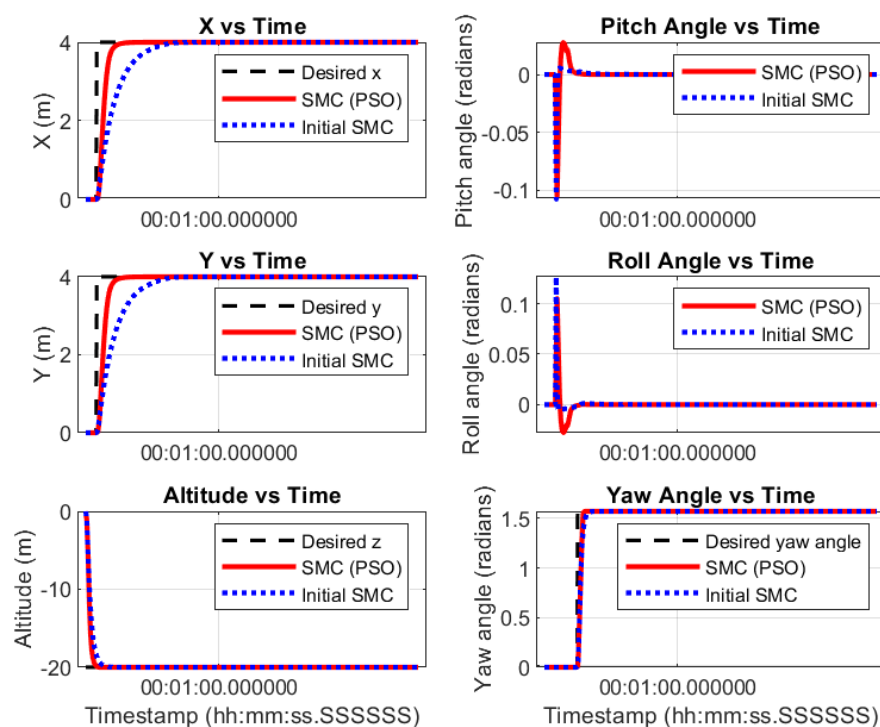


Figure 8. Simulated step response: before and after 100 iterations of PSO–tuning.

It is important to note that PSO is implemented here under the constraints imposed by stability theory. Therefore, only stabilizing PSO parameters were iterated. Ultimately, the output of the PSO tuning provided SMC parameters that outperformed benchmark PID controllers in most trajectory tracking experiments in time simulations. This was comprehensively documented in our recent paper [26]. However, SMC suffered from chattering issues and caused oscillations in the attitude in real-world experiments. The chatter in the configured PWM values going to the motors was significant, having a deviation of over 20% of the entire PWM range. The response of the UAV with SMC after the PSO tuning but before any manual fine-tuning is presented in Figure 9.

The manual fine-tuning adjusted the a_i in the sliding surface while adjusting the \dot{e} gain k_6 for roll in the control law to maintain the stability condition. Then, the gain k_5 on the saturation function for roll was adjusted. The intuition for tuning the sliding surface and related gains follows along the lines of tuning a PD controller. The inner attitude loop is tuned first, then the outer loop. A comparison of the SMC performance on the real UAV before and after the fine-tuning is provided in Figure 9.

The fine-tuning of the parameters improved the performance of the SMC in every axis of motion. There was approximately a $2\times$ reduction in position error in the Y axis, and the maximum error in yaw was reduced from ± 5 degrees to ± 2 degrees. The chatter in the PWM output was reduced, by a factor of 4, to a magnitude of 50 μ s in pulse width. The fine-tuning on the physical hardware was essential to obtaining a high performance on the real UAV due to model inaccuracies and overall less noise in the simulation versus the real world.

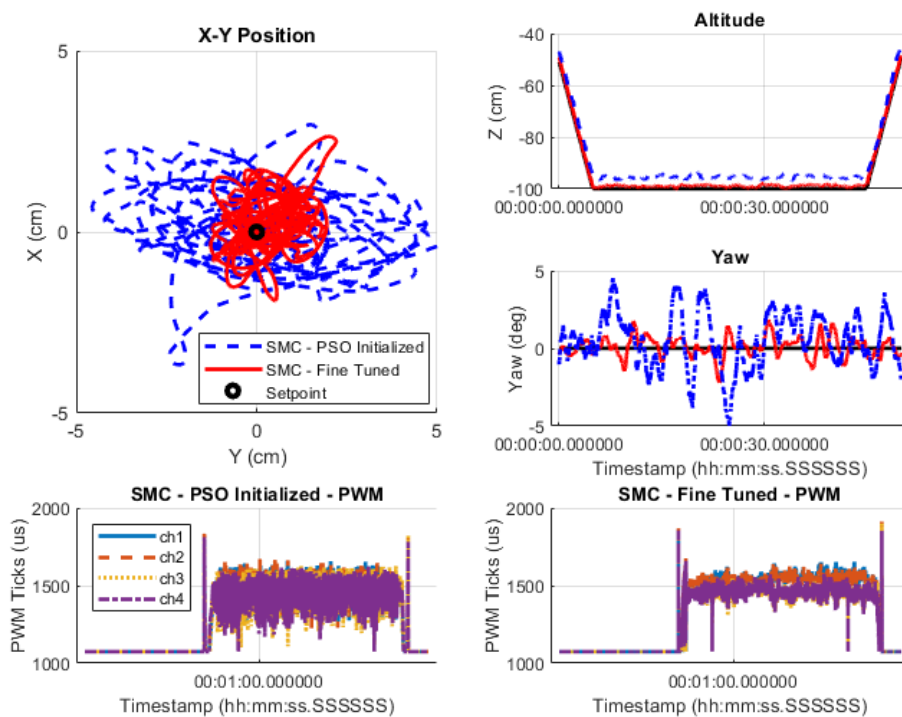


Figure 9. Hover experimental results: before and after manual fine-tuning.

4. Experimental Results

4.1. Experimental Setup

The experiments were conducted in a narrow, closed, and turbulent space, as shown in Figures 2 and 10. The testing field was built out of transparent acrylic sheets. The chamber had a length of 2.15 m, width of 2.20 m, and height of 2.30 m. Three cameras were located along each of the NED axes. The stationary wind source (location marked in Figure 10) was turned on before the flight and turned off after the flight. The thrust produced by the fan could be adjusted by a remote controller (RC). Rag strips were attached around the edges of the wind source in order to illustrate the wind. The indoor navigation was achieved by the onboard realistic states measurement system consisting of a visual inertial odometry (VIO) system and three redundant inertial measurement units (IMU). In addition, to validate the data from the onboard sensors, the position and trajectory of the UAV was also recorded by three external, fixed high-resolution depth cameras.

As the UAV changes its attitude frequently during the flight, winds with dynamic amplitude and changing direction generated by UAV motors combined with a stationary wind source tend to create turbulent airflow in a small, enclosed room. The inertial wind forces dominate over the viscous forces, especially at relatively low altitudes (under 1 m) or near the walls. Based on our measurements, the ratio of the length of the UAV to the length of the cage is around 0.25, which means even when the UAV is located at the center it is still easily affected by the airflow near the walls.

Five representative cases of experiments were designed, as shown in Figure 11. The first case was nominal flight without intentionally added disturbances, which means the disturbances perceived by the flight controller were mainly induced by the UAV itself (physical characteristics) as well as the diffused turbulent airflow within the small testing field, for example, the imbalance caused by varying battery positions, sensor distribution, the vibration from the actuators, and the changing turbulent flow, which depends on the UAV's position and speed. The second scenario was flying with a suspended spherical weight that was attached at the center of mass (CM) of the UAV. The length of the string was 0.35 m and its weight was negligible, and the mass of the metal ball was 131.7 g. As

shown in Figure 11b, when flying, disturbance force would be decomposed into X - Y - Z directions and the force magnitude in each axis could change due to the back and forth swinging of the metal ball. The third scenario, which is shown in Figure 11c, was similar to the second case except the suspended weight was attached to the end of the right-rear landing gear; thus, this time not only were dynamic disturbance forces in X - Y - Z induced, but also variable disturbance torques in pitch roll and yaw were generated.

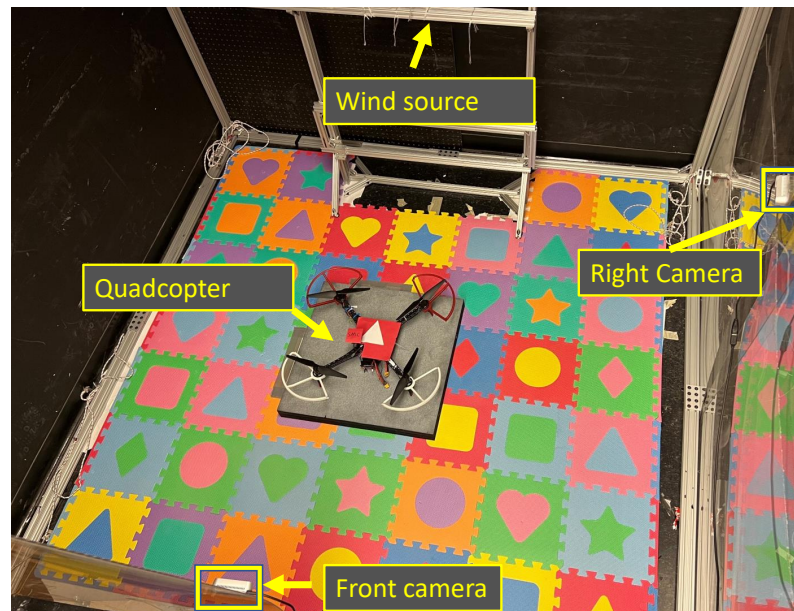


Figure 10. Testing field located in a narrow, closed, transparent space, with three fixed cameras (top, right, front) and a stationary point wind source.

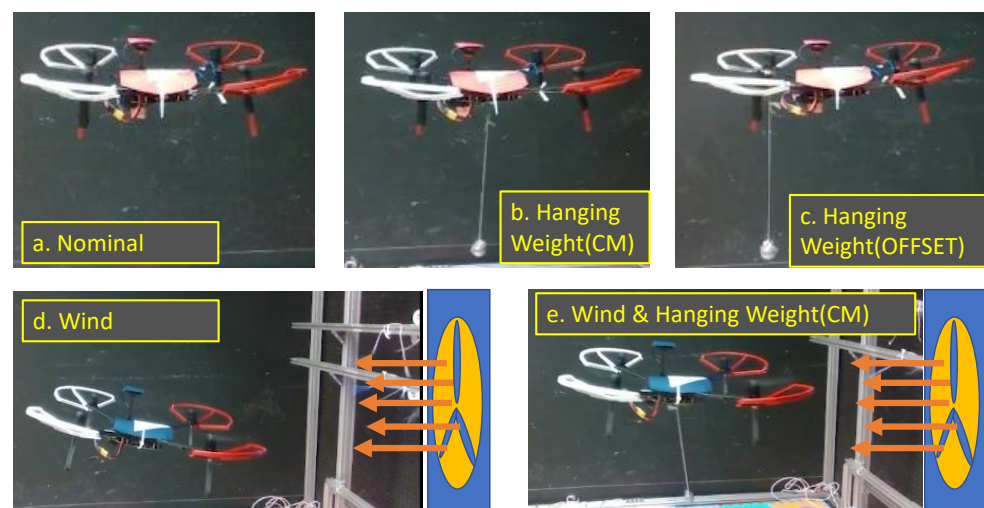


Figure 11. Experimental scenarios: (a) nominal flight; (b) mounting a hanging weight (CM) at the center of mass of the UAV; (c) mounting a hanging weight (OFFSET) at the end of the right-rear leg of the quadcopter; (d) adding wind in front of the UAV; (e) adding wind together with the hanging weight (CM).

Due to the uncertainty of the real-world environment, such as unpredictable wind gusts, variable weather conditions, and so on, a series of comprehensive wind experiments are necessary in developing a new, reliable, robust flight controller, as shown in Figure 11d,e. The fourth scenario was to hold the position under a unidirectional wind. This was also regarded as the wind-only case since no extra disturbance was added except the wind.

From the 3D environment diagram in Figure 2, the wind was produced directly in front of the UAV. The maximum wind speed at the position of the UAV was 9 m/s. Since the wind tends to spread and disperse over a range, the effective distance of the lateral wind would be limited, and the magnitude and direction of the wind depends on the flight position of the UAV. The fifth case of experiments, shown on the bottom-right of Figure 11e, was an extension of the wind-only experiment, and the UAV was loaded with a spherical suspended weight.

4.2. SMC Performance Analysis

The SMC was subject to step inputs in the X , Y , Z , and Yaw axes in four separate experiments to evaluate the error dynamics of the controller. Plots of the velocity error \dot{e} against the position error e with the $s = 0$ hypersurface included on the graph are described throughout this section. In the X axis, a step input of 0.5 m was applied. The response can be seen in Figure 12.

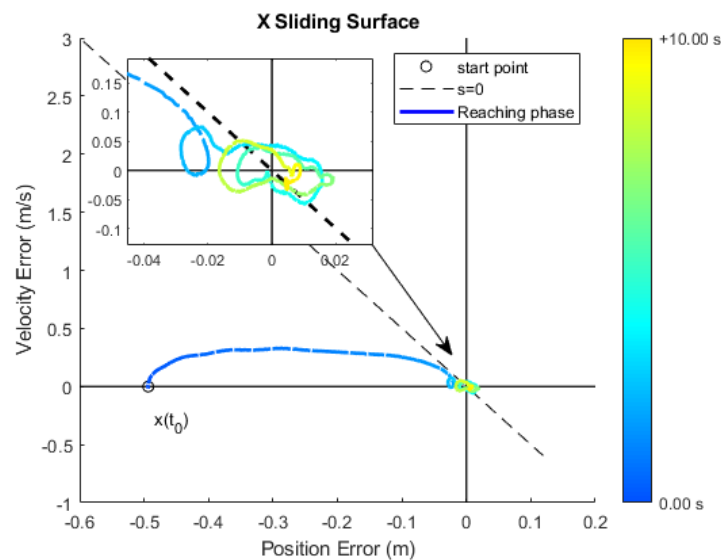


Figure 12. Sliding surface for X –trajectory step tracking. Color code indicates time (s).

The error exhibits both the reaching and sliding phases, as expected. In the reaching phase, the velocity only reaches 0.5 m/s, producing a flat reaching mode response, and the sliding mode is achieved as the controller drives the error to 0 (see Figure 12). The Y axis was also subject to a 0.5 m step input. The error dynamics performed very similarly to the X controller, as shown in Figure 13. The slow reaching mode dynamics of the X and Y controllers is because of the nested-loop controller structure. That is, the X and Y responses are subject to the pitch and roll dynamics. Figures 14 and 15 show the pitch and roll sliding modes during the respective X and Y step inputs. The pitch and roll error dynamics very quickly pass through the reaching mode and enter the sliding mode, which is what causes the flat reaching mode curves in X and Y .

The Z axis was subject to a 0.3 m altitude step input. As shown in Figure 16, the reaching mode overshoots the sliding surface but then smoothly re-approaches it while entering the sliding mode. This relatively smooth oscillation about the sliding surface is allowed by the saturation function. If the saturation function were not used, the sliding mode would be achieved faster, but there would almost certainly be chatter about the sliding surface at the transition from the reaching mode to the sliding mode.

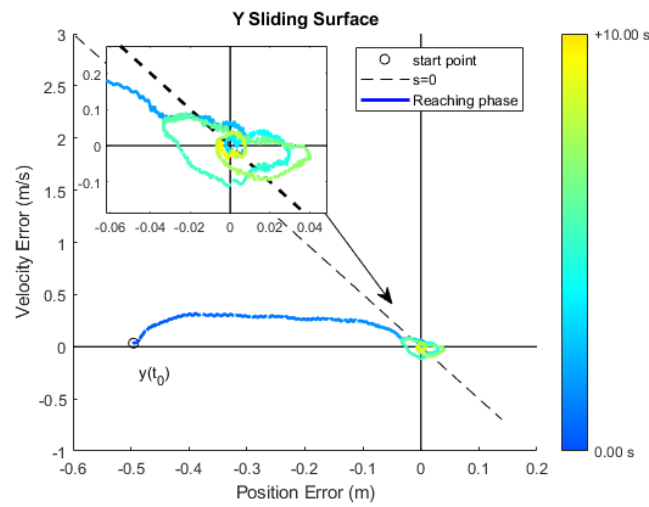


Figure 13. Sliding surface for Y–trajectory step tracking. Color code indicates time (s).

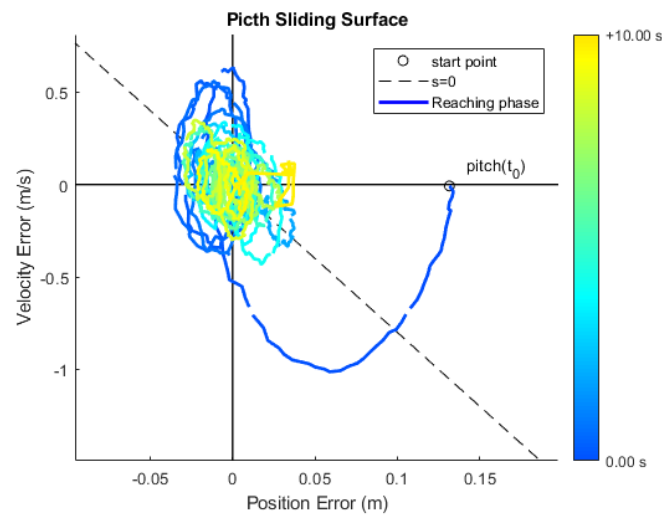


Figure 14. Sliding surface for pitch during the X–step response. Color code indicates time (s).

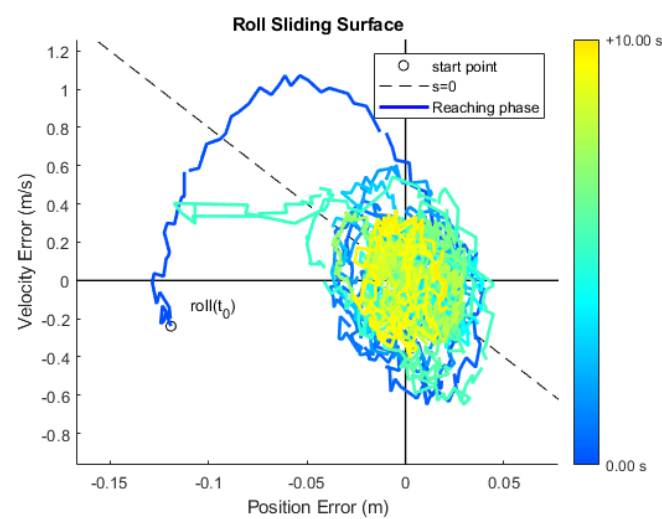


Figure 15. Sliding surface for roll during Y–step response. Color code indicates time (s).

Finally, the yaw was subject to a 90 degrees clockwise step rotation (Figure 17). Similarly to the Z error dynamics, without the saturation function the transition from reaching mode to sliding mode would occur sooner. Instead, there is some undershoot of the sliding surface initially, then a smooth convergence towards the sliding surface. The yaw motion here may be a bit conservative, but due to the coupling of the attitude dynamics, this is required for a smooth system response. More aggressive yaw control introduces chatter into all of the attitude control variable outputs, which propagates through the system.

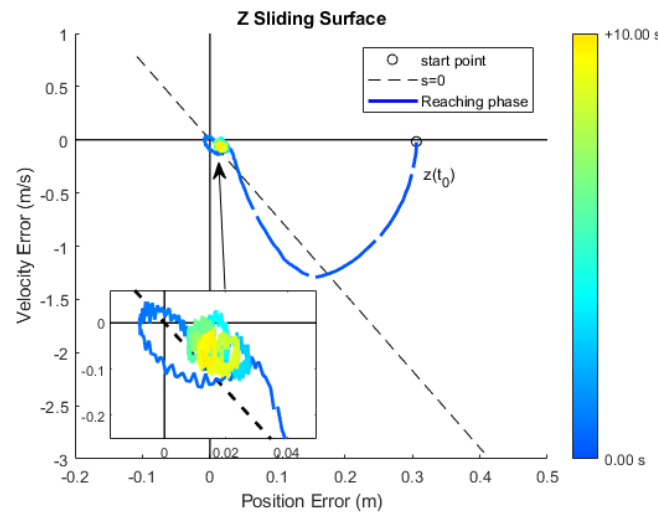


Figure 16. Sliding surface for Z–trajectory step tracking. Color code indicates time (s).

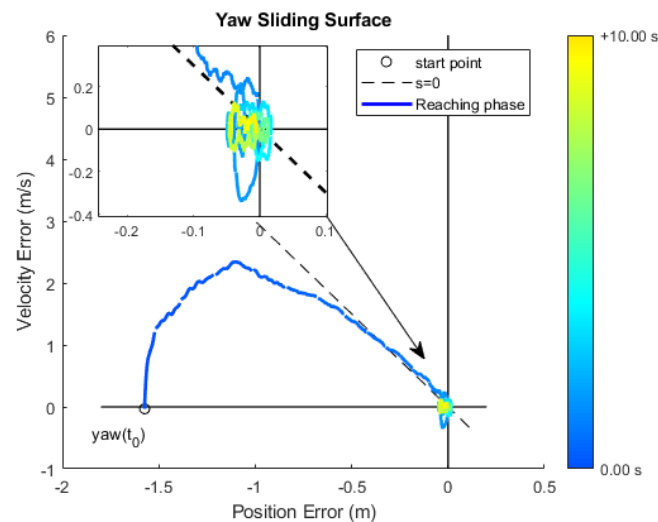


Figure 17. Sliding surface for Yaw–trajectory step tracking. Color code indicates time (s).

All of the axes subject to step inputs following the conventional reaching mode to sliding mode error dynamics, as expected. The X and Y response is limited by the attitude dynamics. The Z controller overshoots the sliding surface slightly when reaching the sliding mode, while the yaw controller undershoots the sliding surface. The overshoot and undershoot of the sliding surface are allowed by the use of the saturation functions on the switching portion of the control law, instead of a more aggressive sign function.

4.3. Trajectory Tracking

The tracking performance of the SMC was tested on three different trajectories: hovering, a rectangle, and a figure eight. In each trajectory, the UAV ascended to an altitude of 1 m at 0.1 m/s then performed the desired trajectory and returned to the ground at 0.1 m/s.

For the hovering trajectory, the setpoint (0 cm, 0 cm, 100 cm, 0 deg), in X , Y , Z , and Yaw was sent to the UAV for the entirety of the trajectory.

The rectangle was comprised of four sides with a length of 30 cm, which were traversed in 3 s each. There was a 1 s hover at each corner of the rectangle. Yaw of 0 degrees was maintained throughout the trajectory. The pattern was repeated twice before landing.

The figure-eight pattern had a radius of 30 cm and the UAV was commanded to face in the direction of travel for the entirety of the trajectory. A total of four loops of the figure eight, each 8 s in duration, were completed prior to landing.

Each of the trajectories was completed with no disturbance applied to the UAV (the nominal case); with a weight hanging from the center of mass of the UAV (weight CM case); and with a hanging weight offset under one of the motors (weight offset case). In addition to these disturbances, for the hover trajectory, the UAV was subject to a wind disturbance along the $-X$ axis. This wind disturbance was applied with no weight added to the UAV (wind case) and with the hanging weight at the center of mass (wind and weight case).

The results are organized by trajectories through the remainder of this subsection in the following order: hover, rectangle, figure eight. The primary quantitative metric for comparing the performance of the controllers is the root mean square (RMS) error in each axis. The RMS error is reported in centimeters for the translational axes and degrees for the rotational axes.

4.3.1. Hover

The RMS error along each axis for the hover trajectory subject to each of the types of disturbances is summarized in Table 2. As a general trend, the RMS error increased with each experiment to the right along the table. This suggests that, as intended, the difficulty of the control problem increased as more severe disturbances were added.

Table 2. Root mean square position and angular error for hover trajectory.

		Nominal	Weight CM	Weight Offset	Wind	Wind and Weight
X (cm)	PID	2.20	2.46	3.56	9.85	9.51
	SMC	0.75	0.73	0.76	1.90	2.27
Y (cm)	PID	2.01	2.05	2.02	7.81	5.11
	SMC	0.87	0.91	0.97	2.30	1.85
Z (cm)	PID	13.38	11.57	12.70	15.20	15.05
	SMC	1.24	1.76	1.82	1.25	2.17
Roll (deg)	PID	1.09	1.24	1.20	1.88	1.92
	SMC	1.04	1.15	1.22	2.31	1.86
Pitch (deg)	PID	0.85	1.05	1.27	2.24	1.99
	SMC	0.98	1.06	1.05	1.32	1.61
Yaw (deg)	PID	2.04	2.16	2.09	6.70	4.90
	SMC	0.90	0.94	0.85	2.06	1.87

In the nominal case, the SMC had an over $2\times$ reduction in RMS error compared to the PID for X and Y tracking. That improvement increased up to a $5\text{--}10\times$ error reduction when disturbances, especially wind disturbances, were added. The Z error for the SMC was less than or around 2 cm for all experiments, regardless of the disturbance applied. Conversely, the PID had a Z error larger than 15 cm and a higher tracking delay for all experiments. The graphs of the experiments revealed that there was a steady-state error of $10\text{--}20$ cm present for all of the PID results. The PID controller was unable to maintain the desired altitude especially under disturbances. The yaw error was $2\times$ better with the SMC and relatively improved when the wind was added. However, the pitch and roll errors were comparable between the two controllers.

Graphs comparing the X , Y , Z , and heading tracking of the PID and SMC controllers for the nominal case are presented in Figure 18 based on the flight logs, along with overlays of the trajectories captured by the top and side cameras in top of the figure. Similar results could be obtained from the camera recording images or the onboard flight logs that the drift area of SMC was significantly smaller than that of PID during hovering. From the sensors' data, the SMC clearly had less variance in X , Y , and yaw than the PID controller. Additionally, there was a steady-state error of over 15 cm for the PID controller in the Z axis. Similar trends can be seen in Figures 19 and 20, when the hanging weights were added. The PID had a larger fluctuation in yaw angle of over 5 degrees, while the SMC fluctuated less than 2 degrees.

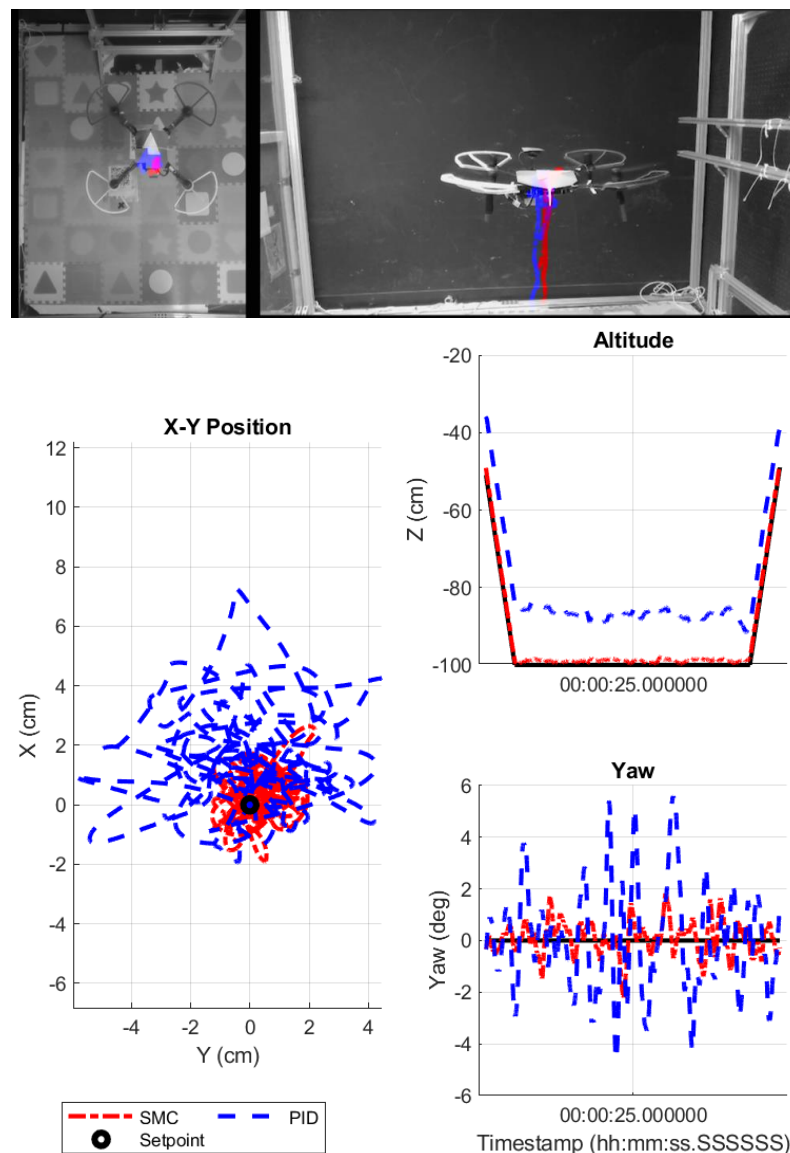


Figure 18. Hover pattern—nominal cases (top: cameras; bottom: onboard flight logs).

In Figure 19, hanging weights were attached to the center of mass or to the end of the landing gear. The SMC was able to maintain its setpoints with or without the payloads. In contrast, the PID had a relatively larger position and heading error, and its drift area in the XY plane increased when an offset hanging weight was added. In addition, the suspended payload had a significant effect on the tracking of the PID in the Z direction. From the results, the PID pulled up and down the UAV when the weights swung from side to side.

Finally, the plots and overlays for the case when the wind was added are in Figure 20. The PID was blown in the direction of the wind on several occasions (green and blue), while the SMC was able to resist the wind disturbance and more accurately maintain the setpoint position (red and pink). There was an oscillation in the Z axis of the PID-controlled UAV caused by it being pushed up and down in the wind. Furthermore, the SMC-controlled UAV fluctuated less than the PID-controlled UAV in terms of heading.

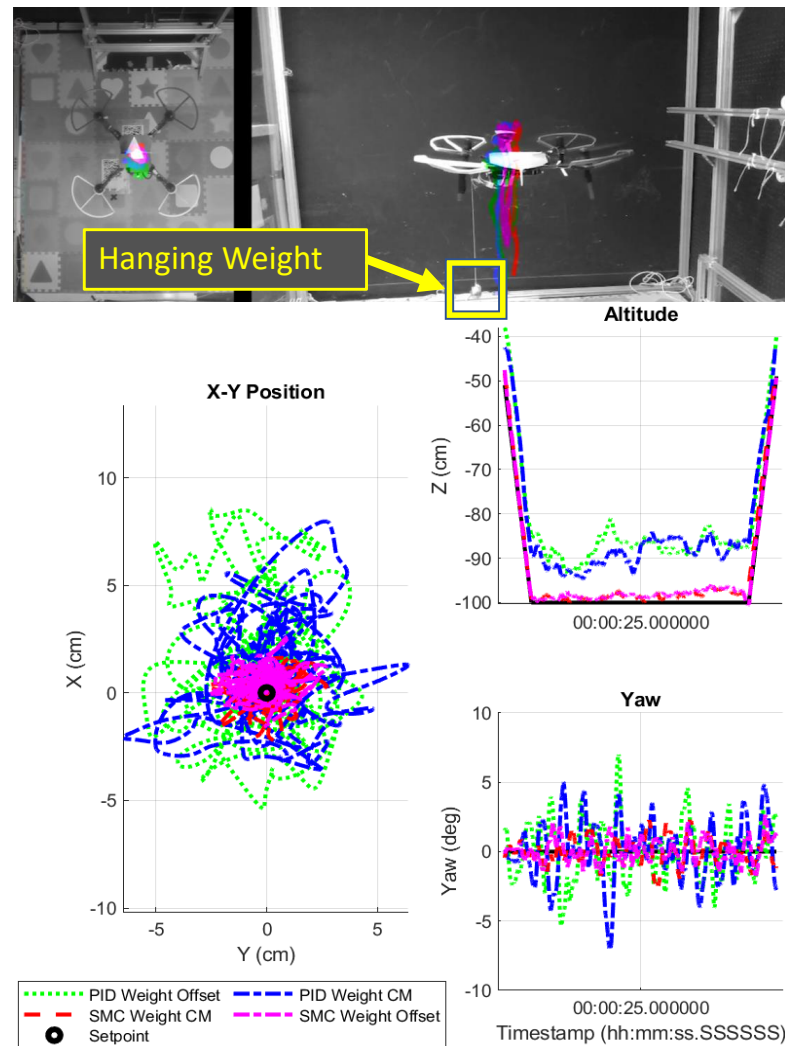


Figure 19. Hover pattern—hanging weight cases (top: cameras; bottom: onboard flight logs).

When hovering, the UAV should maintain its current position, heading, and altitude, and it was designed to actively reject any externally applied disturbances, such as forces and torques, with the help of the omnidirectional disturbance observers. In the nominal case, Figure 21i shows nonzero disturbances of 0.75 N in the X direction, -0.2 N in the Y direction, and -3 N in the Z direction, which remained constant throughout the experiment. These baseline disturbances might be caused by small space turbulence, i.e., airflow synthesized from forward, leftward, and downward winds. In other words, ideally, these values should be 0 N if the UAV was located in a large open and undisturbed space. When the UAV was on the ground, as shown in the beginning of experiments in Figure 21i(c), the force in the Z direction shows the same magnitude as the UAV's weight of 12.75 N but in the opposite direction, which indicated a disarmed status without any propeller speed.

The force disturbances in the hanging weight cases should have no difference from the nominal case except an additional downward force in the Z direction. By analyzing the average differences between the nominal and weighted cases, we can estimate the weight of

the payload to be ≈ 1 N, while the actual weight of the payload was 131.5 g (1.29 N). When the hanging weight was mounted on the left-rear leg of the UAV, i.e., the weight-offset case in the figure, the observer-estimated forces increased due to the additional thrust produced to balance the torque.

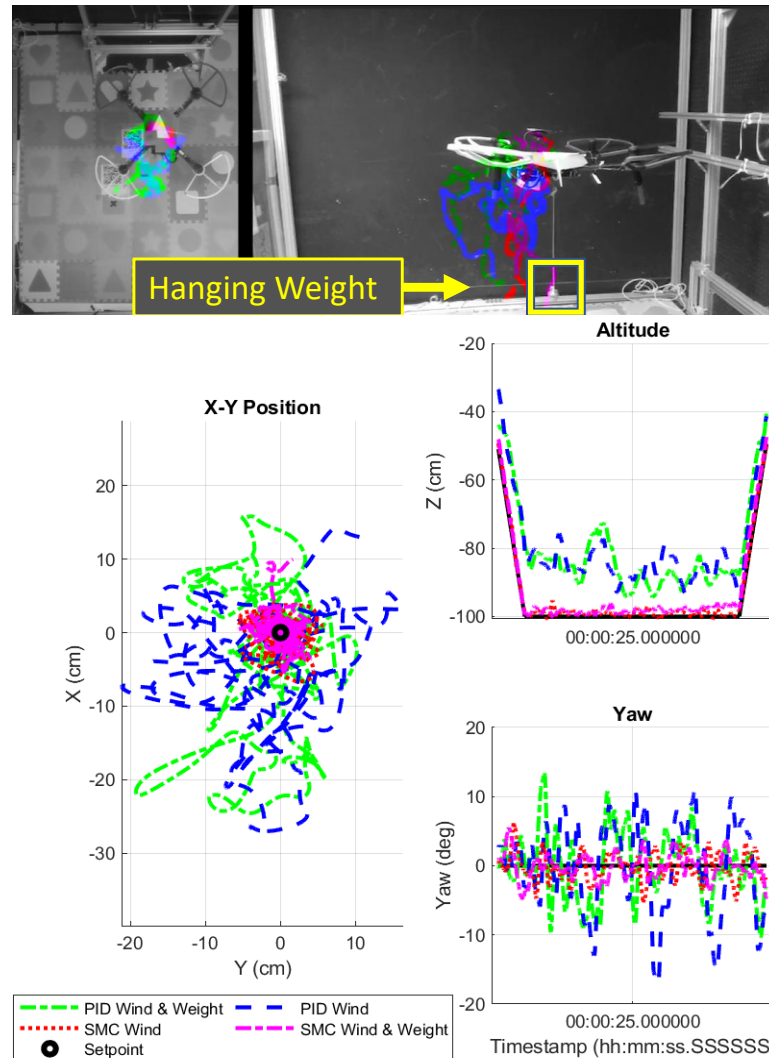


Figure 20. Hover pattern—wind and weight cases (**top**: cameras; **bottom**: onboard flight logs).

Finally, the magnitude of forces increased significantly in $-X$ and $-Y$ after applying the lateral wind. The force in X surged in the beginning and gradually stabilized at 0.25 N. We estimated the wind force in X as ≈ -0.5 N by calculating the steady state offsets between the nominal case and the wind cases, and the wind force in Y was estimated as ≈ -0.2 N. Figure 21i shows the airflow inside the cage was blowing from north-east to south-west with an approximated angle of ≈ 21 degrees. The disturbance peaks in the initial and final phases represented the transition into and out of the main wind field, where the forces may change abruptly.

The torque disturbances estimated by the observers were plotted in Figure 21ii. The roll torque means the torque around the X axis, the pitch torque represents the torque around the Y axis, and the yaw torque means the torque around the Z axis, respectively. In the case of a weight offset, by comparing the differences between nominal and weighted cases, it could be seen that the torques in pitch and roll were both ≈ 0.2 Nm, compared to the actual torque (0.2278 Nm). In addition, because the hanging weight was mounted to the right-rear leg of the UAV, both disturbances in pitch and roll should be positive in the

NED coordinates, which was validated by the blue curves in Figure 21ii(d,e). Both nominal and weighted cases show small yaw torques that wobbled around zero.

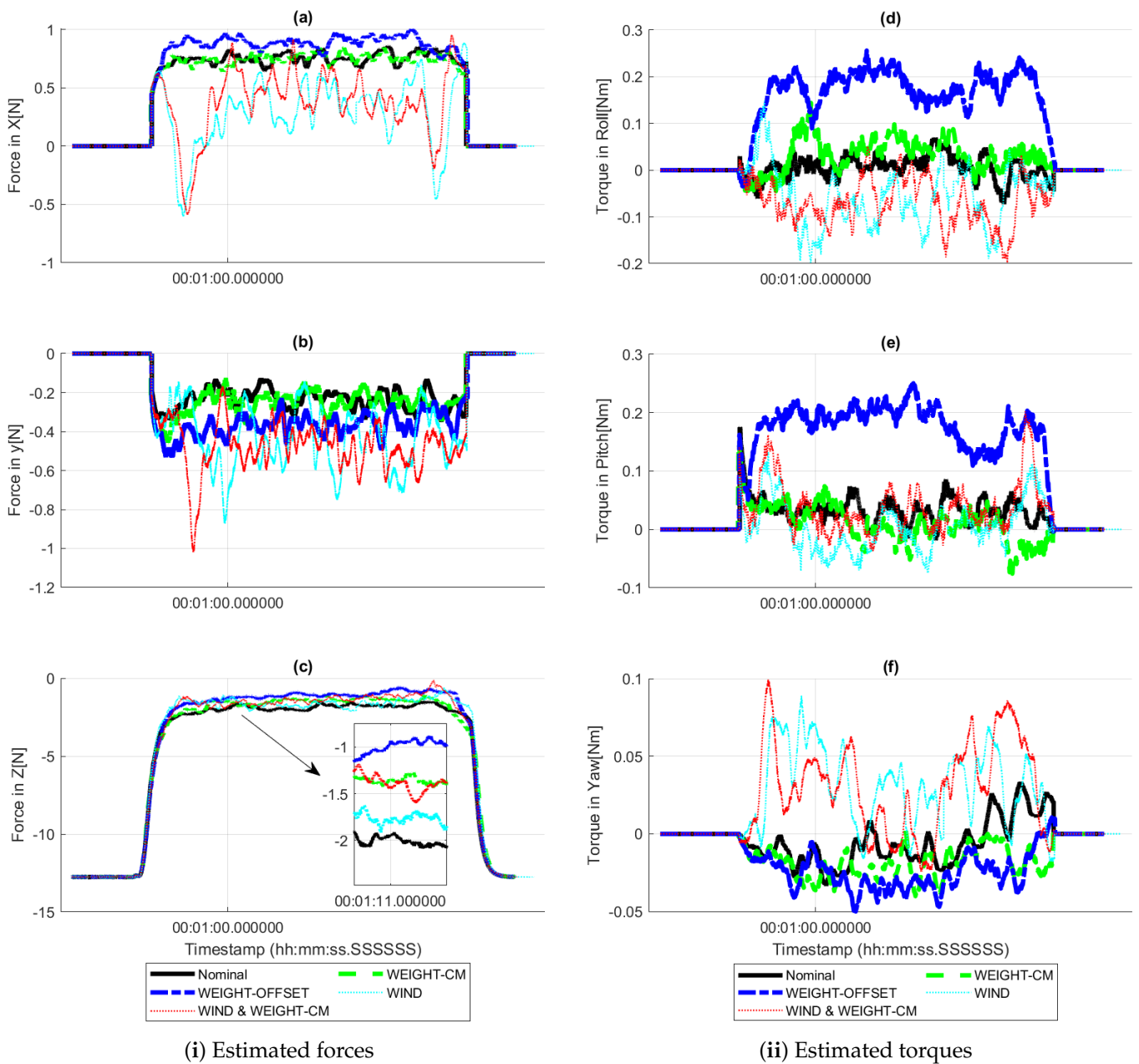


Figure 21. The SMC-based actual UAV flying in hover mode: (i) estimated forces (N) in X, Y, and Z by the disturbance observer and (ii) estimated torque (Nm) in roll, pitch, and yaw by the disturbance observer.

In the wind cases, torque disturbances had more intense oscillations in pitch and roll due to a more frequent need of attitude correction, as shown in Figure 21ii(d,e). From Figure 21ii(f), compared with the nominal case, a clear increase in yaw could be observed, which indicated an increase in positive torque around the Z axis. From the force estimation results in Figure 21i, we already know the fan was located in the north-east direction relative to the UAV. However, the wind generated by the fan was still to the south, which would produce a positive yaw torque around the Z axis. In fact, a misalignment of the center of the UAV and the center curve of the wind tunnel is a common cause of yaw torque

for the indoor aerial vehicle. This misalignment could also be verified by the north-east wind in Figure 22.

Even though we did not have the real-time information on the actual turbulence profile around the UAV, from the disturbance observer results in Figure 21i,ii, the magnitude and direction of the airflow could be estimated. As a practical application, the disturbance observer can also be used to estimate the real-time wind direction and magnitude and sketch the wind profile. As an example, in the wind case, the horizontal force applied to the UAV was shown on the polar plot in Figure 22. Based on the NED coordinate frame, the UAV was located at the origin, with 0 degree corresponding to the +X and 90 degrees representing the +Y. The wind generated by the fan was blowing in the $-X$ direction. The color bar is associated with the time-series of the experiment, starting from blue and landing at yellow, where the maximum magnitude of the wind and its corresponding direction could be retrieved after the flight. From this polar map, the wind was ≈ 0.5 N, blowing from the north-east towards the UAV during most of the flight, with the maximum wind occurring at the beginning and at the end.

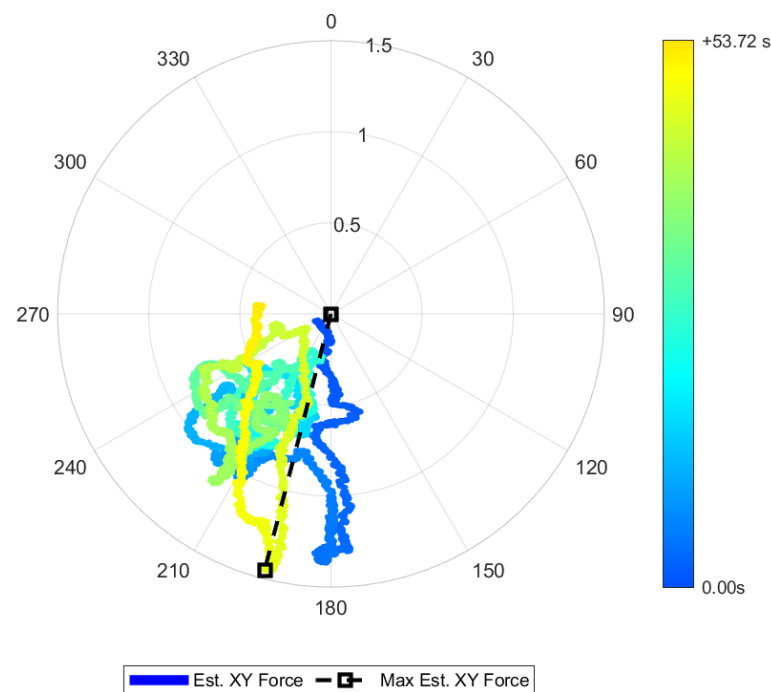


Figure 22. Disturbance-observer-estimated wind force direction (deg) and magnitude (N) in the XY plane.

As a short summary of the hover flight experiment, the total error was calculated according to Table 2 and by

$$\begin{cases} \|e_{pos}\|_2 = \sqrt{\sum_{i=x,y,z} \|e_i\|^2} \\ \|e_{att}\|_2 = \sqrt{\sum_{j=r,p,yaw} \|e_j\|^2} \end{cases} \quad (16)$$

where e_{pos} denotes the total position error, e_{att} denotes the total attitude error, i and j represent each degree of freedom, and x , y , z , r , p , and yaw correspond to X , Y , Z , $roll$, $pitch$, and yaw , respectively. The RMS error distribution for a particular controller in each case was scattered and formed a region in Figure 23, with the horizontal and vertical coordinates indicating the position error and attitude error, respectively. In addition, to distinguish the differences between SMC and PID, their results were displayed in different colors and textures.

In Figure 23, each marker denotes a different experiment. For example, comparing the SMC with the PID under the nominal condition, the blue circle was farther from the origin than the red circle, which indicated the PID has a larger position and attitude error than the SMC. The colored region enclosed by each experiment has the meaning of deviation under uncertainty, and a smaller area means better adaptation to external disturbances; in other words, the controller adapts faster when uncertainty occurs. When a hanging weight was mounted, the UAV tended to track poorly, with larger position and attitude errors compared to the nominal case, although some exceptions occur; for example, when looking at the PID weight case in Figure 23 the PID had a smaller attitude error than the SMC. In the presence of the wind, there was no doubt that both SMC and PID had larger tracking errors compared to their own nominal cases. In short, it was clear that SMC showed a better tracking accuracy as well as a better disturbance rejection capacity than the PID controller.

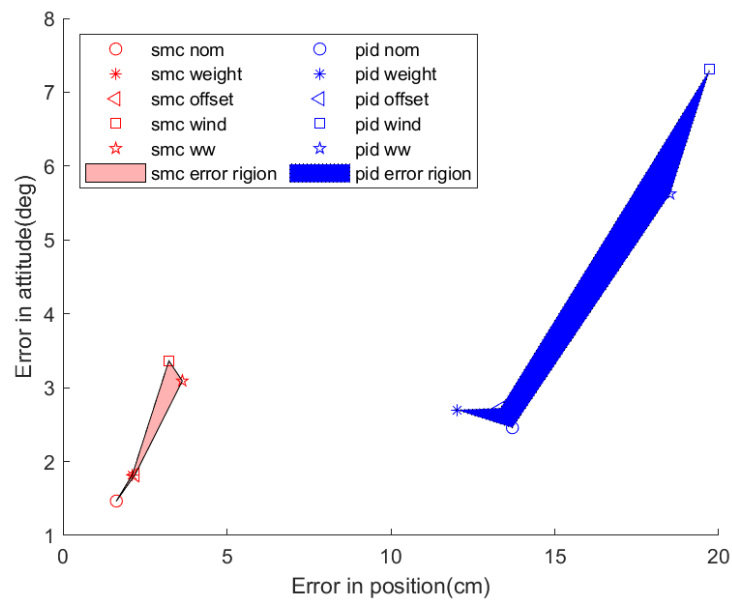


Figure 23. Hover: position and attitude error (*_nom*: nominal condition; *_weight*: centered hanging weight condition; *_offset*: offset hanging weight; *_ww*: centered hanging weight together with the wind; *_wind*: wind-only condition).

4.3.2. Rectangle

The rectangle trajectory was completed for the nominal, center of mass weight, and offset weight cases. The RMS error from each of these experiments is summarized in Table 3 for the PID and SMC.

The SMC offered more than a 4× improvement in X and Y position tracking. Additionally, the Z tracking was improved by over 5× with the SMC versus the PID controller. The pitch and roll performance of the SMC was slightly worse than that of the PID. Finally, the yaw error was reduced by a factor of 2. In the yaw error plots, it can be seen that the oscillations for the PID controller are approximately double the amplitude of the SMC yaw oscillations (Figure 24). This 2 to 1 relationship holds regardless of the applied disturbance. However, when a weight disturbance is added there are some spikes in the PID yaw angle error larger than ±5 degrees, which are not present in the nominal case.

Table 3. Root mean square position and angular error for rectangle trajectory.

		Nominal	Weight CM	Weight Offset
X (cm)	PID	4.81	5.08	5.43
	SMC	0.91	0.89	0.90
Y (cm)	PID	4.31	5.14	4.28
	SMC	1.16	1.09	1.23
Z (cm)	PID	10.79	11.54	9.07
	SMC	1.61	2.13	1.55
Roll (deg)	PID	1.28	1.36	1.43
	SMC	1.78	1.74	1.82
Pitch (deg)	PID	1.07	1.25	1.17
	SMC	1.96	1.97	1.99
Yaw (deg)	PID	2.03	2.54	2.63
	SMC	0.99	1.07	0.91

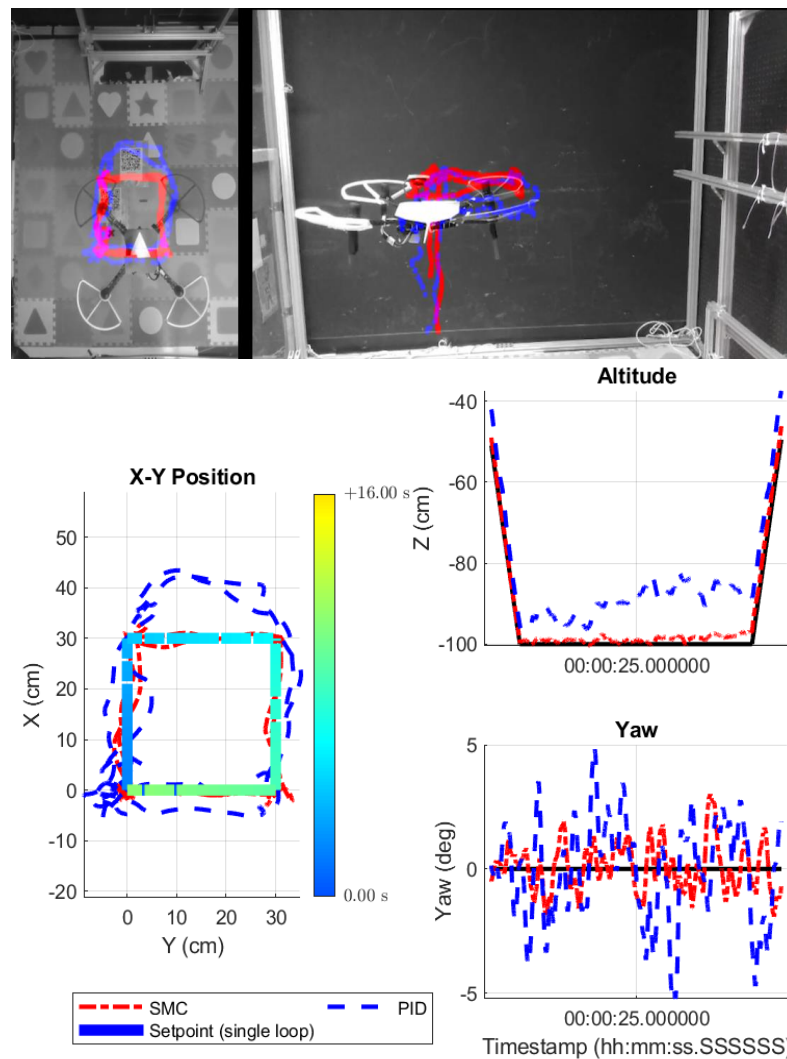


Figure 24. Rectangle pattern—nominal weight cases (top: cameras; bottom: onboard flight logs).

The XY position plots in Figure 24, as well as the captured trajectories from the tracking cameras, show a consistent overshoot in +X with the PID after the first side of the rectangle is traced. This overshoot was present both with and without added disturbances (see Figure 25). Interestingly, despite this large and consistent overshoot, the RMS error

values in X and Y are comparable for the PID controller, suggesting that the overshoot did not single-handedly degrade the X tracking performance.

The Z plots show a consistent undershoot in the altitude for the PID, regardless of the applied disturbance. This result is consistent with the 10 cm undershoot seen in the hover-trajectory tracking as well (see Figures 24 and 25).

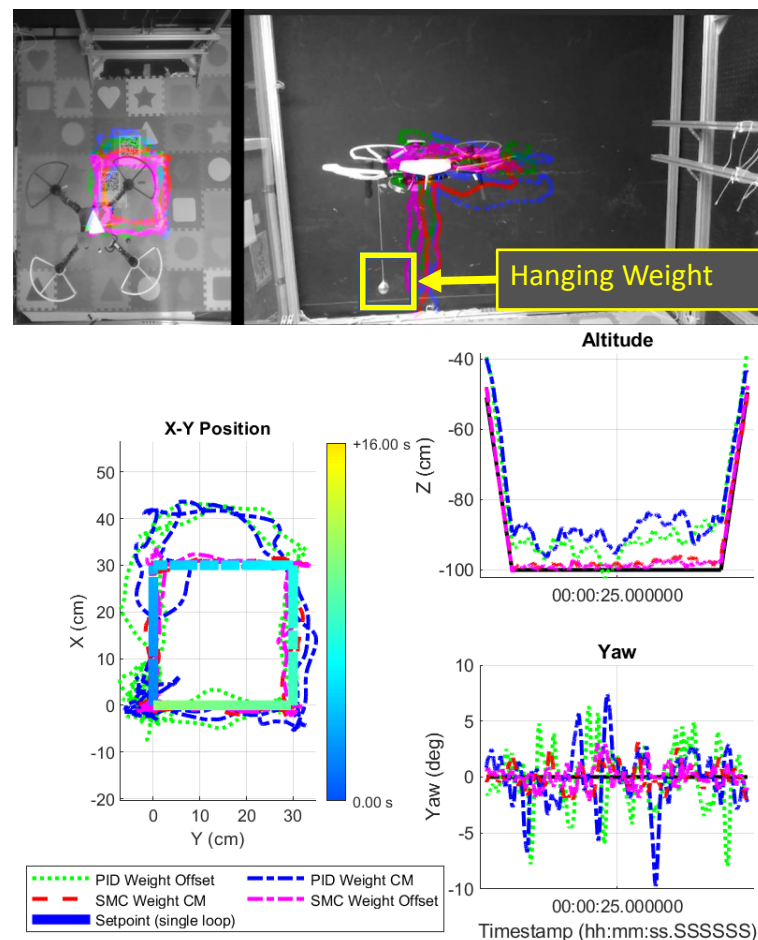
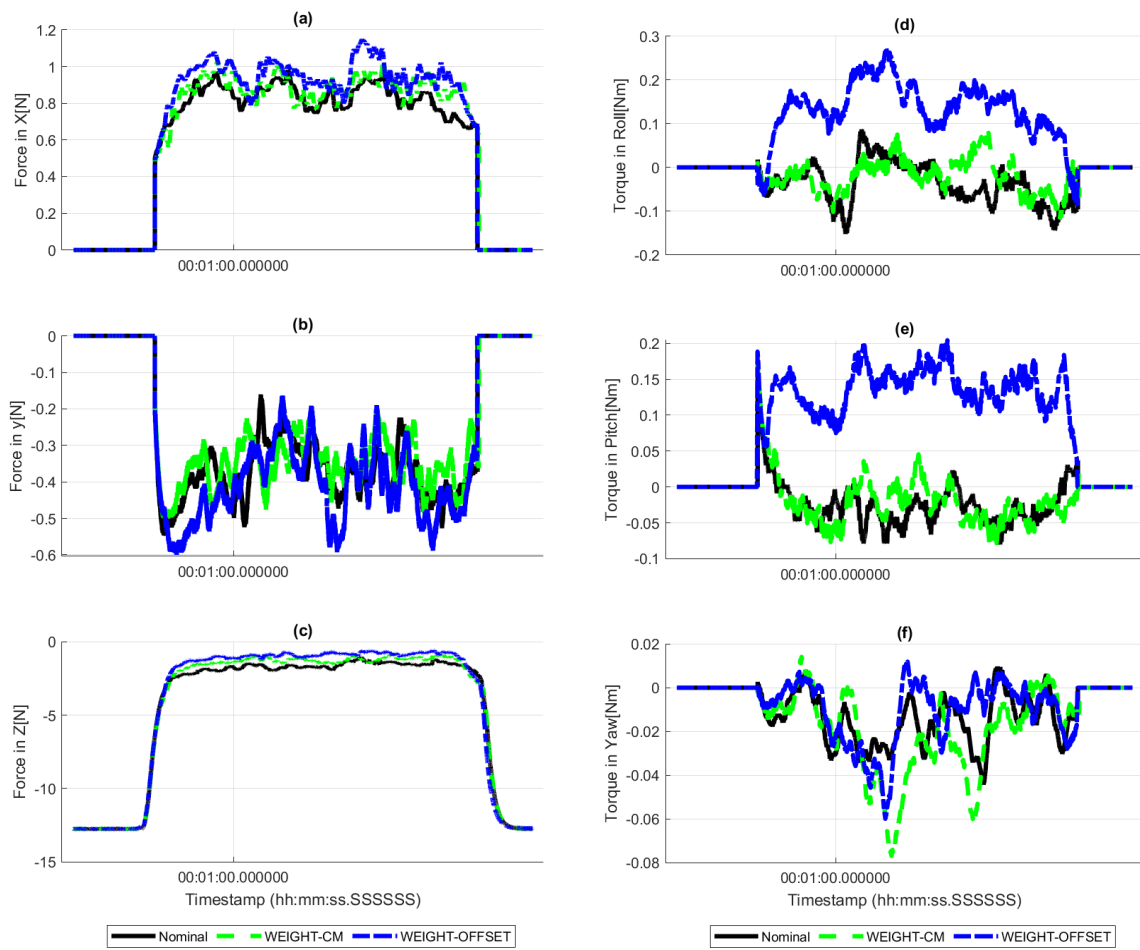


Figure 25. Rectangle pattern—hanging weight cases (**top:** cameras; **bottom:** onboard flight logs).

Figure 26i shows the estimated linear forces on the UAV, calculated by the SMC disturbance observer. Unlike the hover case, due to the swinging pendulum motion, there were small disturbance fluctuations with respect to XY . However, the Z disturbances were slightly larger than the nominal case due to the attached weight.

In Figure 26ii, the estimated torque disturbances are plotted. With the offset weight, constant torque disturbances of ≈ 0.2 Nm can be seen in pitch and roll. There were minimal differences in yaw disturbance between the test with the center weight and the test with the offset weight; however, their fluctuations became larger compared to the nominal case due to the swinging pendulum.

During the rectangular trajectory tracking experiments, the PID has a larger tracking error than the SMC, as shown in Figure 27. The overall position error reached over $6\times$ of the SMC, and the average attitude error was about the same; however, the deviation of attitude error of PID was much larger than the one of SMC (about $30\times$). The large blue region enclosed by three standalone experiments implies the behaviors of the PID varying under different uncertainties. Although it had a better angle tracking in the nominal scenario, PID's performance reduced dramatically in the other disturbance scenarios. Again, from the rectangular experiment results, the SMC overwhelms the PID in both tracking accuracy and disturbance rejection.



(i) Estimated forces

(ii) Estimated torques

Figure 26. The SMC-based actual UAV flying in rectangle mode: (i): estimated forces (N) in X, Y, and Z by the disturbance observer. (ii): estimated torque (Nm) in roll, pitch, and yaw by the disturbance observer.

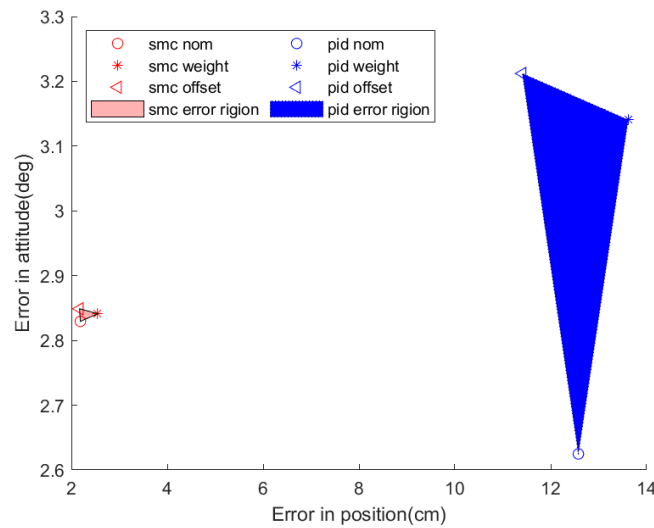


Figure 27. Rectangle: position and attitude error (*_nom*: nominal condition; *_weight*: centered hanging weight condition; *_offset*: offset hanging weight; *_ww*: centered hanging weight together with the wind; *_wind*: wind-only condition).

4.3.3. Trajectory of a Figure Eight

The figure-eight trajectory was completed for the nominal case and both hanging weight cases. The RMS error from each of these experiments is summarized in Table 4 for the PID and SMC (Figure 28).

Table 4. Root mean square position and angular error for a figure-eight trajectory.

		Nominal	Weight CM	Weight Offset
X (cm)	PID	7.89	7.61	8.24
	SMC	4.70	4.53	5.61
Y (cm)	PID	5.64	5.60	6.24
	SMC	4.77	4.09	4.38
Z (cm)	PID	3.92	6.17	5.47
	SMC	1.75	1.79	1.71
Roll (deg)	PID	1.57	1.68	1.56
	SMC	1.44	1.71	1.49
Pitch (deg)	PID	1.46	1.69	1.89
	SMC	1.27	1.31	1.44
Yaw (deg)	PID	5.85	5.69	7.53
	SMC	4.78	4.67	4.57

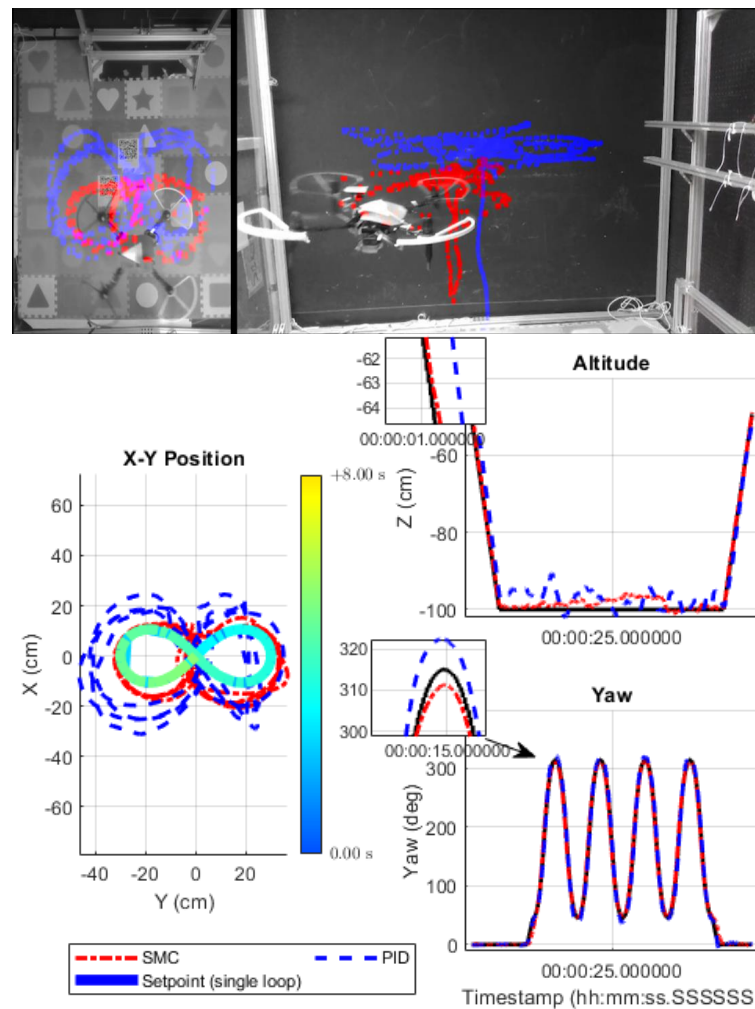


Figure 28. Figure-eight pattern—nominal cases (top: cameras; bottom: onboard flight logs).

In the XY position tracking, the PID had around 1.2–1.5 times the RMS error of that of SMC. The Z tracking was a little more than 2× better with the SMC. In pitch and roll, the SMC error did not change compared to the rectangular trajectory; however, the PID error increased slightly. The difference between the two controllers for pitch and roll tracking was, again, not very significant. In yaw, the SMC outperformed the PID by a small margin with the exception of the offset weight. The PID had around 2× more error in yaw tracking with the offset weight, where the SMC’s performance did not change relative to the other cases.

In the XY position plots (Figure 29), the SMC follows a more symmetric path than the PID. The PID has a larger circle on the second half of the trajectory, the left side, than on the first half. This difference is a result of the PID not returning to the origin between the first loop and the second loop. It overshoots the origin on the first loop and is unable to compensate in the second loop.

The Z tracking steady-state offset was not present in the PID controller for the figure-eight trajectory, unlike with the other trajectories, though there was an initial height error after takeoff that was reduced within the first 10 s of the trajectory following. The Z regulation was still poor compared to the SMC. The PID yaw tracking had a slight delay and some overshoot as the direction changed.

Figure 30i shows the estimated linear force disturbances, from the SMC disturbance observer. The reciprocal oscillations of disturbances in XY showed the aerodynamic forces opposing the motion. As the UAV would yaw and turn, the weight would swing out accordingly. Thus, for the offset weight case, due to a larger centrifugal force, the X and Y disturbances grew larger, with sharper oscillations; meanwhile, Z had a larger downward force. Because of the UAV’s aggressive flight motion and rapidly changing attitude, the offset weight could usually swing at an angle of about 30–45 deg.

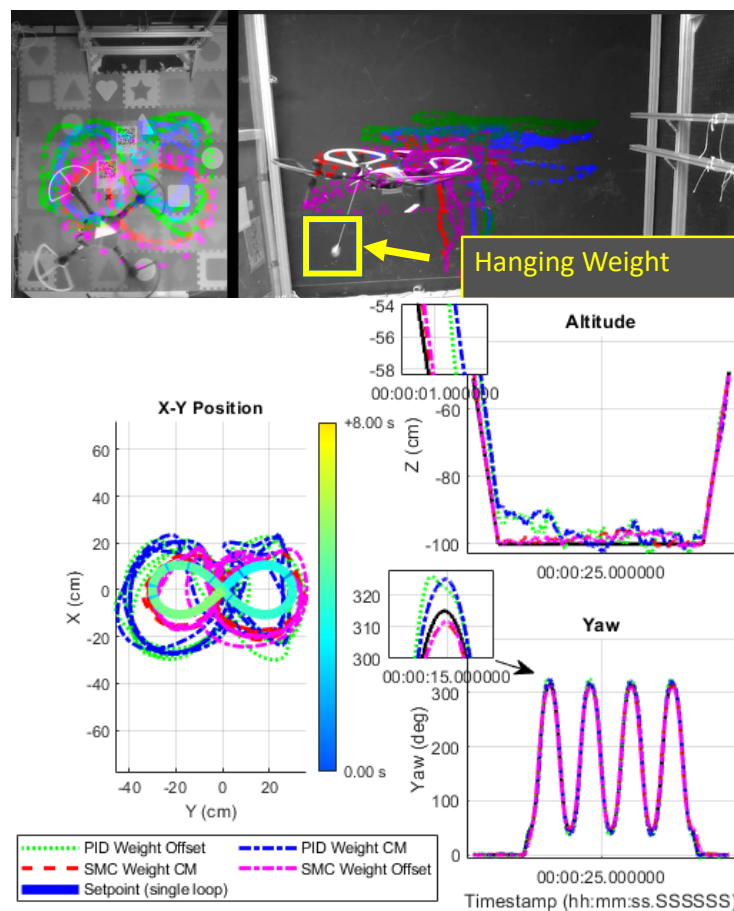
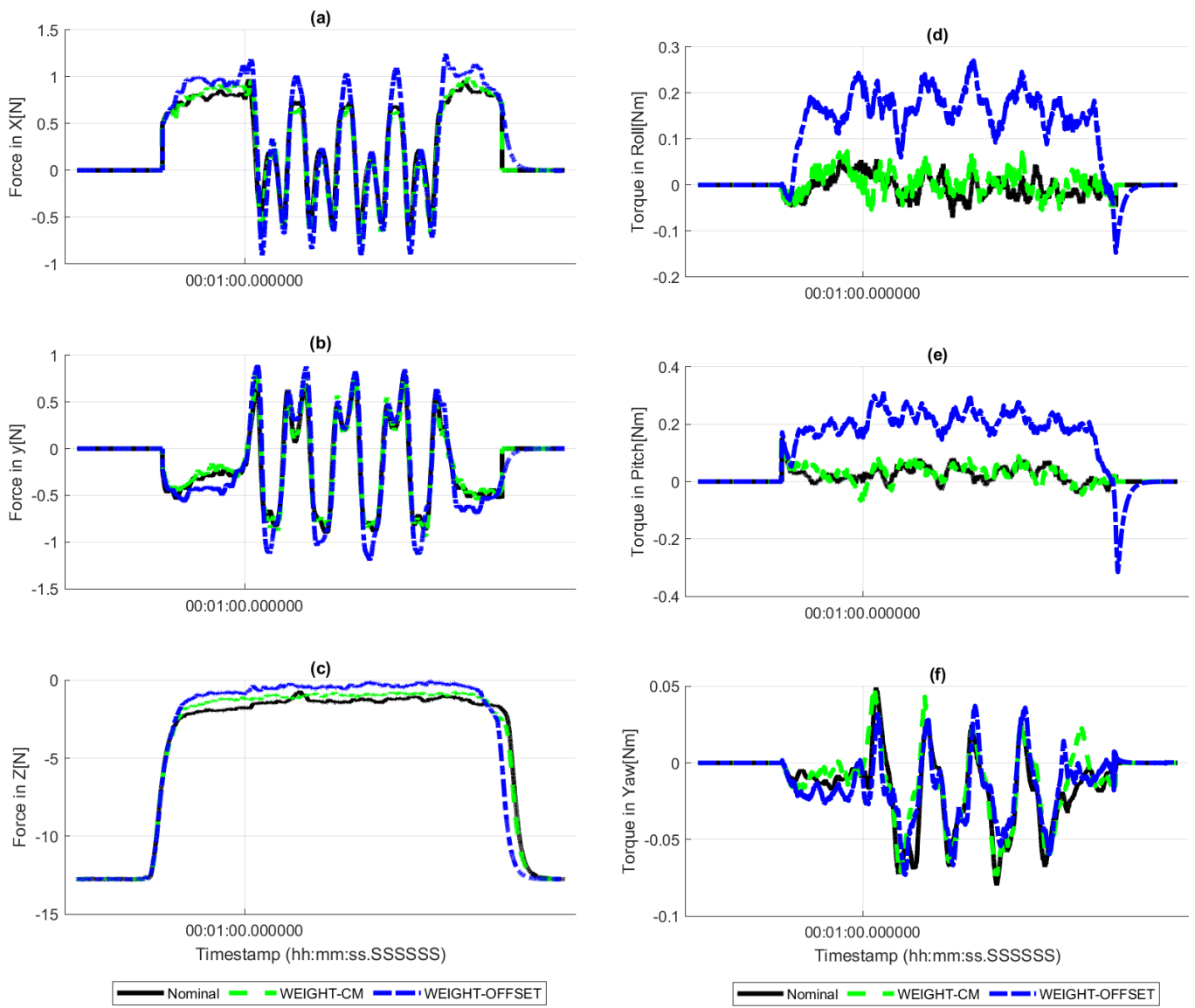


Figure 29. Figure-eight pattern—hanging weight cases (top: cameras; bottom: onboard flight logs).



(i) Estimated forces

(ii) Estimated torques

Figure 30. The SMC-based actual UAV flying in figure-eight mode: (i) estimated forces (N) in X, Y, and Z by the disturbance observer; (ii) estimated torque (Nm) in roll, pitch, and yaw by the disturbance observer.

The pitch and roll disturbances in Figure 30ii show constant torque offsets of ≈ 0.2 Nm for the offset weight case, similarly to the rectangle and hover trajectories (Figures 21ii and 26ii). The yaw disturbance had oscillations, which were caused by a larger aerodynamic moment as the UAV followed the figure-eight yaw setpoints (Figure 30ii(f)).

As for the complex figure-eight pattern tracking test, the UAV was commanded to change position and attitude all the time. Both the SMC and the PID have larger position and attitude errors in this case. For the figure-eight nominal case, the PID had $1.5\times$ position error and more than $1.2\times$ attitude error than that of SMC (Figure 31). For the CM weight case, the PID had $1.8\times$ position error and more than $1.2\times$ attitude error than that of SMC. For the offset weight case, the PID had $1.6\times$ position error and more than $1.5\times$ attitude error than that of SMC. Overall, SMC had a better performance than the PID in this case.

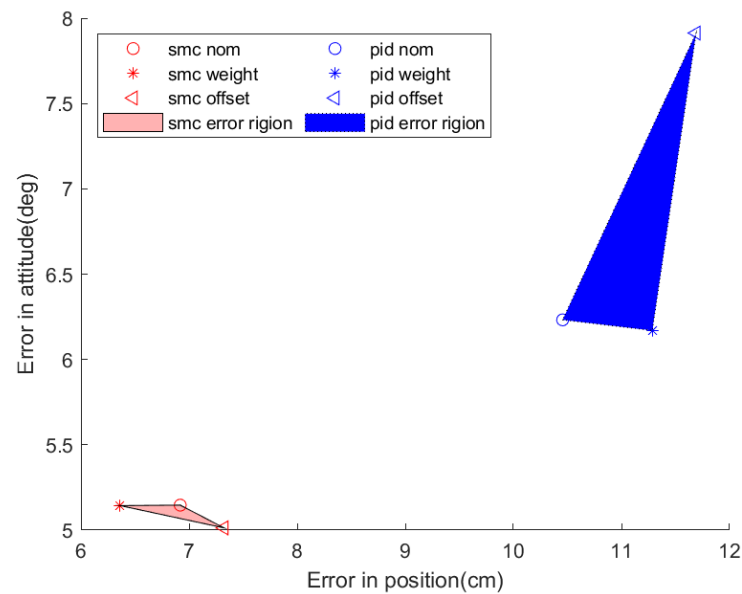


Figure 31. Figure eight: position and attitude error (*_nom*: nominal condition; *_weight*: centered hanging weight condition; *_offset*: offset hanging weight; *_ww*: centered hanging weight together with the wind; *_wind*: wind-only condition).

5. Conclusions and Future Work

In this work, a 6-degrees-of-freedom sliding mode controller was designed for a UAV to operate in real-world scenarios. A simulated UAV was developed first for roughly tuning the control parameters using a PSO algorithm, followed by a manual fine-tuning process on the actual UAV. The proposed controller was tested in an enclosed and turbulent space, where it showed a better tracking accuracy and higher robustness than PID, especially under severe disturbances, such as loads, torques, and wind.

The SMC controller outperformed an industry-standard PID controller across a variety of tests. When subject to disturbances while following complex trajectories, the SMC responded faster and more precisely, thanks to its disturbance observers, and always maintained an RMS positional error below 10 cm. The SMC had more than a $2\times$ improvement over a PID controller in position tracking across a variety of trajectories under nominal conditions, with more than a $5\times$ improvement under severe disturbances.

In future work, a second-order sliding mode controller could be used to reduce chattering from the controller and reduce the susceptibility to noise. This would especially improve the attitude control. Another possible direction would be to create an online tuning process, potentially using neural networks, instead of relying on a manual fine-tuning process. These results can further be combined with obstacle avoidance algorithms that can serve to generate real-time safe trajectories along which the drone should navigate.

Author Contributions: Conceptualization, R.S. and J.M.-L.; methodology, Y.J., A.M., R.S. and J.M.-L.; software, Y.J. and A.M.; validation, Y.J. and A.M.; formal analysis, Y.J. and A.M.; investigation, Y.J. and A.M.; resources, R.S. and J.M.-L.; data curation, Y.J. and A.M.; writing—original draft preparation, Y.J. and A.M.; writing—review and editing, R.S. and J.M.-L.; visualization, Y.J.; supervision, R.S. and J.M.-L.; project administration, R.S. and J.M.-L.; funding acquisition, R.S. and J.M.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would also like to thank Weite Zhang, Chang, Liu, Xu Mao, and Yi Huang for their help in the simulations and experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. X and Y Controllers

For the completeness of the work, here we summarize from [26] the SMC design supported by stability theory. The designs of the X and Y controllers are very similar, so the derivation of the X controller is presented in detail and the Y controller result will be shown by analogy. We have the following definition for the UAV:

$$\begin{cases} u_1 = \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ u_2 = -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \end{cases} \quad (\text{A1})$$

According to (1), (3) and (A1),

$$\ddot{x} = -\frac{1}{m}[u_1 \cdot 4T_{max} \cdot \tau_T + K_1 \cdot \dot{x} \cdot |\dot{x}|] \quad (\text{A2})$$

where u_1 is the control signal. Next, consider the sliding surface:

$$s_x = a_1 e_x + \dot{e}_x \quad (\text{A3})$$

where $e_x = x - x_d$, with x_d as the X setpoint. Then, use Equations (A2) and (A3) and apply the SMC design summarized in Section 3. A Lyapunov stability analysis is conducted, using the Lyapunov function $V(s_x) = \frac{1}{2}s_x^2$, whose time derivative reads as follows:

$$\begin{aligned} \dot{V}(s_x) &= s_x \dot{s}_x = s_x(a_1 \dot{e}_x + \ddot{e}_x) = s_x(a_1 \dot{e}_x + \ddot{x} - \ddot{x}_d) \\ &= s_x \left[a_1 \dot{e}_x - \frac{1}{m}(u_1 \cdot 4T_{max} \tau_T + K_1 \cdot \dot{x} \cdot |\dot{x}|) - \ddot{x}_d \right] \\ &= s_x \left[a_1 \dot{e}_x - \left(k'_1 \text{sat}\left(\frac{s_x}{\epsilon}\right) + k_2 \dot{e}_x + cs_x - \ddot{x}_d \right) \right. \\ &\quad \left. - K_1 \cdot \dot{x} \cdot |\dot{x}| + K_1 \cdot \dot{x} \cdot |\dot{x}| - \ddot{x}_d \right] \\ &= s_x \left[(a_1 - k_2) \dot{e}_x - \left(k'_1 \text{sat}\left(\frac{s_x}{\epsilon}\right) + cs_x \right) \right] \end{aligned} \quad (\text{A4})$$

From Equation (A4),

$$\begin{cases} a_1 - k_2 = 0 \\ k'_1 > 0 \\ c > 0 \end{cases} \quad (\text{A5})$$

are sufficient conditions to ensure stability with $\dot{V}(s_x) < 0$, where k'_1 , c , and k_2 represent E_1 , E_2 , and E_3 in Section 3, respectively. Following the same procedures in Section 3, we obtain the control law as follows:

$$u_1 = \frac{1}{4T_{max} \tau_T} \left[m \left(k'_1 \text{sat}\left(\frac{s_x}{\epsilon}\right) + k_2 \dot{e}_x + cs_x - \ddot{x}_d \right) - K_1 \cdot \dot{x} \cdot |\dot{x}| \right] \quad (\text{A6})$$

where $+cs$ is an additional control term used on the X and Y controllers, and k'_1 is a variable gain that reduces large position errors quickly without causing instability for small position errors:

$$k'_1 = \begin{cases} k_1, & (b_1 \cdot |e| + b_2 \cdot |\dot{e}|) < k_1 \\ (b_1 \cdot |e| + b_2 \cdot |\dot{e}|), & (b_1 \cdot |e| + b_2 \cdot |\dot{e}|) \geq k_1 \end{cases}$$

where b_1 and b_2 are tunable constants.

Following by analogy, and considering the same stability conditions given by (A5), the Y control law is given by the following:

$$u_2 = \frac{1}{4T_{max}\tau_T} [m \left(k'_1 \text{sat}\left(\frac{s_y}{\epsilon}\right) + k_2 \dot{e}_y + c s_y - \ddot{y}_d \right) - K_1 \cdot \dot{y} \cdot |\dot{y}|] \tag{A7}$$

With u_1 and u_2 given in (A6) and (A7), recall (A1) we can solve for ϕ and θ to retrieve the roll and pitch setpoints for the attitude controller:

$$\begin{cases} \phi_{des} &= \arcsin(u_1 \sin \psi - u_2 \cos \psi) \\ \theta_{des} &= \arcsin\left(\frac{u_1 \cos \psi + u_2 \sin \psi}{\cos \phi}\right) \end{cases} \tag{A8}$$

Appendix A.2. Altitude Controller

For the Z controller, start with the Z equation from (1), (3)

$$\ddot{z} = -\frac{1}{m} [(\cos \phi \cos \theta) \cdot 4T_{max} \cdot \tau_T + K_1 \cdot \dot{z} \cdot |\dot{z}|] + g \tag{A9}$$

and let the sliding surface be the following:

$$s_z = a_2 e + \dot{e} \tag{A10}$$

The stability analysis follows similarly to X with a Lyapunov function of $V(s_z) = \frac{1}{2}s_z^2$, whose time derivative reads as follows:

$$\begin{aligned} \dot{V}(s_z) &= s_z \dot{s}_z = s_z [a_2 \dot{e}_z + \ddot{e}_z] \\ &= s_z \left[\left(a_2 - \frac{4T_{max}}{m} k_4 \right) \dot{e}_z - \left(k'_3 \frac{4T_{max} (\cos \phi \cos \theta)}{m} \text{sat}\left(\frac{s_z}{\epsilon}\right) \right) \right] \end{aligned} \tag{A11}$$

Then, the sufficient conditions for achieving $\dot{V}(s_z) < 0$ become the following:

$$\begin{cases} a_2 - \frac{4T_{max}}{m} k_4 = 0 \\ k'_3 > 0 \end{cases} \tag{A12}$$

where k'_3 and k_4 are in the form of E_1 and E_3 in Section 3, respectively, and k'_3 is defined similar to k'_1 in the X and Y controllers:

$$k'_3 = \begin{cases} k_3, & (b_3 \cdot |e| + b_4 \cdot |\dot{e}|) < k_3 \\ (b_3 \cdot |e| + b_4 \cdot |\dot{e}|), & (b_3 \cdot |e| + b_4 \cdot |\dot{e}|) \geq k_3 \end{cases}$$

Then, following the SMC design summarized in Section 3, rewrite (A9) and (A10) in the form of τ_T :

$$\begin{aligned} \tau_T &= k'_3 \text{sat}\left(\frac{s_z}{\epsilon}\right) + \frac{k_4}{\cos \theta \cos \phi} \dot{e}_z \\ &\quad + \frac{-K_1 \cdot \dot{z} \cdot |\dot{z}| + mg - m\ddot{z}_d}{4T_{max} \cos \phi \cos \theta} \end{aligned} \tag{A13}$$

Appendix A.3. Attitude Controllers

The design of the attitude control laws was similar to the design of the altitude controller. The primary difference is that there is a coupling between the τ variables. For

example, τ_{Roll} controller depends on τ_{Pitch} and τ_{Yaw} . This coupling will be resolved at the end of Appendix A after the attitude control laws are derived.

In the design of the roll controller, an expression for $\ddot{\phi}$ needs to be obtained. By differentiating the $\dot{\phi}$ equation from (1), the following is obtained:

$$\begin{aligned} \ddot{\phi} = & -\frac{2\sqrt{2} \cdot l \cdot T_{max}}{J_x} \tau_R - \frac{2\sqrt{2} \cdot l \cdot T_{max} \cdot \tan \theta \sin \phi}{J_y} \tau_P \\ & + \frac{4Q_{max} \tan \theta \cos \phi}{J_z} \tau_Y + f_\phi \end{aligned} \tag{A14}$$

where f_ϕ

$$\begin{aligned} f_\phi = & q \cdot \left(\frac{1}{\cos^2 \theta} \cdot \dot{\theta} \cdot \sin \phi + \tan \theta \cos \phi \cdot \dot{\phi} \right) \\ & + r \cdot \left(\frac{1}{\cos^2 \theta} \cdot \dot{\theta} \cdot \cos \phi - \tan \theta \sin \phi \cdot \dot{\phi} \right) \\ & - \frac{J_z - J_y}{J_x} \cdot qr - \frac{K_2}{J_x} \cdot p \cdot |p| - \frac{J_x - J_z}{J_y} \cdot \\ & pr \cdot \tan \theta \sin \phi - \frac{K_2}{J_y} \cdot q \cdot |q| \cdot \tan \theta \sin \phi \\ & - \frac{J_y - J_x}{J_z} \cdot pq \cdot \tan \theta \cos \phi \\ & - \frac{K_2}{J_z} \cdot r \cdot |r| \cdot \tan \theta \cos \phi \end{aligned} \tag{A15}$$

Next, the sliding surface is selected:

$$s_\phi = a_3 e_\phi + \dot{e}_\phi \tag{A16}$$

Similarly to the altitude controller, the control law becomes

$$\begin{aligned} \tau_R = & k_5 \cdot \text{sat}\left(\frac{s_\phi}{\epsilon}\right) + k_6 \cdot \dot{e}_\phi - \frac{J_x}{J_y} \cdot \tan \theta \sin \phi \cdot \tau_P \\ & + \frac{4J_x \cdot Q_{max} \cdot \tan \theta \cos \phi}{2\sqrt{2}J_z \cdot l \cdot T_{max}} \cdot \tau_Y \\ & + \frac{J_x}{2\sqrt{2} \cdot l \cdot T_{max}} \cdot (f_\phi - \ddot{\phi}_{des}) \end{aligned} \tag{A17}$$

Then, the Lyapunov stability condition can be calculated using the Lyapunov function $V(s_\phi) = s_\phi^2$, and, imposing $\dot{V}(s_\phi) < 0$, we have the following:

$$\begin{aligned} \dot{V}(s_\phi) = & s_\phi \dot{s}_\phi = s_\phi (a_3 \dot{e}_\phi + \ddot{\phi} - \ddot{\phi}_{des}) \\ = & s_\phi \left[\left(a_3 - \frac{2\sqrt{2} \cdot l \cdot T_{max}}{J_x} \cdot k_6 \right) \dot{e}_\phi \right. \\ & \left. - \frac{2\sqrt{2} \cdot l \cdot T_{max}}{J_x} \cdot k_5 \cdot \text{sat}\left(\frac{s_\phi}{\epsilon}\right) \right] \end{aligned} \tag{A18}$$

Sufficient conditions for roll stability become the following:

$$\begin{cases} a_3 - \frac{2\sqrt{2}T_{max}l}{J_x}k_6 = 0 \\ k_5 > 0 \end{cases} \tag{A19}$$

Through analogy with the roll controller, the pitch controller is found to be

$$\begin{aligned} \tau_P = & \frac{1}{\cos \phi} \cdot k_5 \cdot \text{sat}\left(\frac{s_\theta}{\epsilon}\right) + \frac{k_6}{\cos \phi} \cdot \dot{e}_\theta - \frac{4J_y \cdot Q_{max} \cdot \tan \phi}{2\sqrt{2}J_z \cdot l \cdot T_{max}} \\ & \cdot \tau_Y + \frac{J_y}{\cos \phi \cdot 2\sqrt{2} \cdot l \cdot T_{max}} \cdot (f_\theta - \ddot{\theta}_{des}) \end{aligned} \quad (\text{A20})$$

where f_θ

$$\begin{aligned} f_\theta = & - (q \cdot \sin \phi \cdot \dot{\phi} + r \cdot \cos \phi \cdot \dot{\phi} \\ & + \frac{J_x - J_z}{J_y} \cdot pr \cdot \cos \phi + \frac{K_2}{J_y} \cdot q \cdot |q| \cdot \cos \phi) \\ & + \frac{J_y - J_x}{J_z} \cdot pq \cdot \sin \phi + \frac{K_2}{J_z} \cdot r \cdot |r| \cdot \sin \phi \end{aligned} \quad (\text{A21})$$

and the stability condition is also given by (A19). The yaw sliding surface is selected next, as in (A22).

$$s_\psi = a_4 e_\psi + \dot{e}_\psi \quad (\text{A22})$$

Then, through the same method as for the pitch and roll controller derivation, the yaw control law becomes

$$\begin{aligned} \tau_Y = & -\frac{\cos \theta}{\cos \phi} \cdot k_7 \cdot \text{sat}\left(\frac{s_\psi}{\epsilon}\right) - k_8 \cdot \frac{\cos \theta}{\cos \phi} \cdot \dot{e}_\psi \\ & + \frac{2\sqrt{2}J_z \cdot l \cdot T_{max} \cdot \tan \phi}{4J_y \cdot Q_{max}} \cdot \tau_P \\ & - \frac{J_z \cdot \cos \theta}{4 \cdot Q_{max} \cdot \cos \phi} \cdot (f_\psi - \ddot{\psi}_{des}) \end{aligned} \quad (\text{A23})$$

where

$$\begin{aligned} f_\psi = & q \cdot \frac{\cos \phi \cos \theta \cdot \dot{\phi} + \sin \phi \sin \theta \cdot \dot{\theta}}{\cos^2 \theta} \\ & + r \cdot \frac{-\sin \phi \cos \theta \cdot \dot{\phi} + \cos \phi \sin \theta \cdot \dot{\theta}}{\cos^2 \theta} \\ & - \frac{\cos \phi (J_y - J_x)}{\cos \theta \cdot J_z} \cdot pq - \frac{\cos \phi \cdot K_2}{\cos \theta \cdot J_z} \cdot r \cdot |r| \\ & - \frac{\sin \phi (J_x - J_z)}{\cos \theta \cdot J_y} \cdot pr - \frac{\sin \phi \cdot K_2}{\cos \theta \cdot J_y} \cdot q \cdot |q| \end{aligned} \quad (\text{A24})$$

Finally, using a Lyapunov function of $V(s_\psi) = s_\psi^2$ and obtaining its time derivative,

$$\begin{aligned} \dot{V}(s) = & s_\psi \dot{s}_\psi = s_\psi (a_4 \dot{e}_\psi + \ddot{\psi} - \ddot{\psi}_{des}) \\ = & s_\psi \left[\left(a_4 - \frac{4Q_{max}}{J_z} \cdot k_8 \right) \dot{e}_\psi \right. \\ & \left. - \frac{\cos \phi \cdot 4Q_{max}}{\cos \theta \cdot J_z} \cdot k_7 \cdot \text{sat}\left(\frac{s_\psi}{\epsilon}\right) \right] \end{aligned} \quad (\text{A25})$$

The yaw stability conditions are found to be the following:

$$\begin{cases} a_4 - \frac{4Q_{max}}{J_z} \cdot k_8 = 0 \\ k_7 > 0 \end{cases} \quad (\text{A26})$$

satisfying $\dot{V}(s_\phi) < 0$.

As mentioned, the attitude control laws (A17), (A20), (A23) are all coupled. The coupling of these control terms is linear, so these equations form a linear system of equations $A\tau = B$. Then, the control variables are given by the solution $\tau = A^{-1}B$. The expressions for A , B , and τ are as follows:

$$A = \begin{bmatrix} 1 & \frac{J_x}{J_y} \cdot \tan \theta \sin \phi & -\frac{4J_x \cdot Q_{max} \cdot \tan \theta \cos \phi}{2\sqrt{2}J_z \cdot l \cdot T_{max}} \\ 0 & 1 & \frac{4J_y \cdot Q_{max} \cdot \tan \phi}{2\sqrt{2}J_z \cdot l \cdot T_{max}} \\ 0 & -\frac{2\sqrt{2}J_z \cdot l \cdot T_{max} \cdot \tan \phi}{4J_y \cdot Q_{max}} & 1 \end{bmatrix} \tag{A27}$$

$$B = \begin{bmatrix} k_5 \cdot \text{sat}\left(\frac{s_\phi}{\epsilon}\right) + k_6 \cdot \dot{\epsilon}_\phi + \frac{J_x}{2\sqrt{2} \cdot l \cdot T_{max}} \cdot (f_\phi - \ddot{\phi}_{des}) \\ \frac{1}{\cos \phi} \cdot k_5 \cdot \text{sat}\left(\frac{s_\theta}{\epsilon}\right) + \frac{k_6}{\cos \phi} \cdot \dot{\epsilon}_\theta + \frac{J_y}{\cos \phi \cdot 2\sqrt{2} \cdot l \cdot T_{max}} \cdot (f_\theta - \ddot{\theta}_{des}) \\ \frac{\cos \theta}{\cos \phi} \cdot -k_7 \cdot \text{sat}\left(\frac{s_\psi}{\epsilon}\right) - k_8 \cdot \frac{\cos \theta}{\cos \phi} \cdot \dot{\epsilon}_\psi \\ -\frac{J_z \cdot \cos \theta}{4 \cdot Q_{max} \cdot \cos \phi} \cdot (f_\psi - \ddot{\psi}_{des}) \end{bmatrix} \tag{A28}$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_P \\ \tau_Y \end{bmatrix} \tag{A29}$$

Appendix A.4. Disturbance Observer

From the UAV model in Figure 3, each propeller generates thrust in the $-z'$ direction. The real thrust factor for each propeller is denoted as $u_{r_i} = (PWM_{r_i} - PWM_{min}) / (PWM_{max} - PWM_{min})$, and $u_{r_i} \in [0, 1]$, which is used to calculate the actual τ_r :

$$\begin{aligned} \tau_{T_r} &= \frac{u_{r_1} + u_{r_2} + u_{r_3} + u_{r_4}}{4} \\ \tau_{R_r} &= \frac{u_{r_2} + u_{r_3} - u_{r_1} - u_{r_4}}{4} \\ \tau_{P_r} &= \frac{u_{r_1} + u_{r_3} - u_{r_2} - u_{r_4}}{4} \\ \tau_{Y_r} &= \frac{u_{r_1} + u_{r_2} - u_{r_3} - u_{r_4}}{4} \end{aligned} \tag{A30}$$

These τ_r values are then fed back into the disturbance observers to construct the final control laws, as explained next. From [50], the disturbance observer is designed to drive the state tracking error asymptotically to zero in the presence of constant force and torque disturbances. The constant force disturbances are represented by $d_f = [d_{f_x}, d_{f_y}, d_{f_z}]$ in X-Y-Z directions, respectively, and constant torque disturbances $d_t = [d_{t_x}, d_{t_y}, d_{t_z}]$ around X-Y-Z axes, respectively.

Unlike the design in [50], which only concerns a three-dimensional force disturbance observer and was only tested in simulations, the proposed method uses a six-dimensional disturbance observer including forces and torques and is tested in UAV experiments. Similar to [50], each observer error dynamic is decoupled from the rest of the loop.

First, the state variables v and w are defined as $[\dot{x}, \dot{y}, \dot{z}]$ and $[p, q, r]$, respectively, where

$$\dot{v} = \begin{bmatrix} -\frac{1}{m} [u_1 \cdot 4T_{max} \cdot \tau_{T_r} + K_1 \cdot \dot{x} \cdot |\dot{x}|] \\ -\frac{1}{m} [u_2 \cdot 4T_{max} \cdot \tau_{T_r} + K_1 \cdot \dot{y} \cdot |\dot{y}|] \\ -\frac{1}{m} [(\cos \phi \cos \theta) \cdot 4T_{max} \cdot \tau_{T_r} + K_1 \cdot \dot{z} \cdot |\dot{z}|] + g \end{bmatrix} \tag{A31}$$

$$\dot{w} = \begin{bmatrix} \frac{1}{J_x} [2\sqrt{2}l \cdot T_{max} \cdot \tau_{R_r} - q \cdot r (J_z - J_y) - K_2 \cdot p \cdot |p|] \\ \frac{1}{J_y} [2\sqrt{2}l \cdot T_{max} \cdot \tau_{P_r} - p \cdot r (J_x - J_z) - K_2 \cdot q \cdot |q|] \\ \frac{1}{J_z} [4Q_{max} \cdot \tau_{Y_r} - p \cdot q (J_y - J_x) - K_2 \cdot r \cdot |r|] \end{bmatrix} \quad (A32)$$

where τ_{T_r} , τ_{P_r} , τ_{R_r} , and τ_{Y_r} are the actual feedback thrust factors calculated by the actuator speed measurement factors u_r . To estimate disturbance acceleration d_f in X–Y–Z directions, we consider the observer structure

$$\begin{cases} \dot{\hat{d}}_f = z_f + k_f \cdot v_r \\ \dot{z}_f = -k_f \cdot \hat{d}_f - k_f \cdot v_r \end{cases} \quad (A33)$$

where v_r is the actual velocity after disturbances $\dot{v}_r = \dot{v} + d_f$. With the error of disturbance defined as $e = d_f - \hat{d}_f$, the error dynamics become

$$\begin{aligned} \dot{e} &= \dot{d}_f - \dot{\hat{d}}_f = \dot{d}_f - z_f - k_f v_r \\ &= -k_f \cdot e + \dot{d}_f \end{aligned} \quad (A34)$$

Assuming that d_f changes slowly compared to the tracking dynamics, then at any instant it can be approximated by a constant, meaning $\dot{d}_f \approx 0$. Under this assumption, the error dynamics become

$$\dot{e} = -k_f \cdot e \quad (A35)$$

which is globally exponentially stable for $k_f > 0$. Here, $k_f = 5$ is selected in the experiments.

To estimate the disturbance in angular acceleration d_t around X–Y–Z axes, a similar observer structure is considered:

$$\begin{cases} \dot{\hat{d}}_t = z_t + k_t \cdot w_r \\ \dot{z}_t = -k_t \cdot \hat{d}_t - k_t \cdot w_r \end{cases} \quad (A36)$$

where w_r is the actual angular velocity after applying disturbances and $\dot{w}_r = \dot{w} + d_t$. With the error of disturbance defined as $e^* = d_t - \hat{d}_t$, the error dynamics become

$$\begin{aligned} \dot{e}^* &= \dot{d}_t - \dot{\hat{d}}_t = \dot{d}_t - z_t - k_t w_r \\ &= -k_t \cdot e^* + \dot{d}_t \end{aligned} \quad (A37)$$

Again, assuming d_t changes slowly, which means $\dot{d}_t \approx 0$, the error dynamics become

$$\dot{e}^* = -k_t \cdot e^* \quad (A38)$$

which is globally exponentially stable for $k_t > 0$. The larger $k_t > 0$ is, the faster the disturbances observer converges. In this study, we chose $k_t = 5$. The disturbance terms (\hat{d}_f, \hat{d}_t) can thus be added into the system dynamics to compensate the constant disturbances.

Referring to Equations (A6), (A7), (A13), (A17), (A20), (A23), (A33), and (A36), we add the estimated disturbances terms in the control laws, which ultimately read as follows:

$$\begin{cases} u'_1 = u_1 + \hat{d}_{f_x} \cdot \frac{m}{4T_{max} \cdot \tau_{T_r}} \\ u'_2 = u_2 + \hat{d}_{f_y} \cdot \frac{m}{4T_{max} \cdot \tau_{T_r}} \\ \tau'_T = \tau_T + \hat{d}_{f_z} \cdot \frac{m}{4 \cdot T_{max} \cos \theta \cos \phi} \\ \tau'_R = \tau_R + \hat{d}_{t_x} \cdot \frac{J_x}{T_{max} \cdot 2\sqrt{2} \cdot l} \\ \tau'_P = \tau_P + \hat{d}_{t_y} \cdot \frac{J_y}{T_{max} \cdot 2\sqrt{2} \cdot l} \\ \tau'_Y = \tau_Y + \hat{d}_{t_z} \cdot \frac{J_z}{4 \cdot Q_{max}} \end{cases} \quad (A39)$$

References

- Samouh, F.; Gluza, V.; Djavadian, S.; Meshkani, S.; Farooq, B. Multimodal Autonomous Last-Mile Delivery System Design and Application. In Proceedings of the 2020 IEEE International Smart Cities Conference (ISC2), Piscataway, NJ, USA, 28 September–1 October 2020; pp. 1–7.
- Guerrero, J.A.; Bestaoui, Y. UAV path planning for structure inspection in windy environments. *J. Intell. Robot. Syst.* **2013**, *69*, 297–311. [\[CrossRef\]](#)
- Flener, C.; Vaaja, M.; Jaakkola, A.; Krooks, A.; Kaartinen, H.; Kukko, A.; Kasvi, E.; Hyypä, H.; Hyypä, J.; Alho, P. Seamless mapping of river channels at high resolution using mobile LiDAR and UAV-photography. *Remote Sens.* **2013**, *5*, 6382–6407. [\[CrossRef\]](#)
- Adão, T.; Hruška, J.; Pádua, L.; Bessa, J.; Peres, E.; Morais, R.; Sousa, J.J. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sens.* **2017**, *9*, 1110. [\[CrossRef\]](#)
- Mohd Daud, S.M.S.; Mohd Yusof, M.Y.P.; Heo, C.C.; Khoo, L.S.; Chainchel Singh, M.K.; Mahmood, M.S.; Nawawi, H. Applications of drone in disaster management: A scoping review. *Sci. Justice* **2022**, *62*, 30–42. [\[PubMed\]](#)
- Erdos, D.; Erdos, A.; Watkins, S.E. An experimental UAV system for search and rescue challenge. *IEEE Aerosp. Electron. Syst. Mag.* **2013**, *28*, 32–37. [\[CrossRef\]](#)
- Naidoo, Y.; Stopforth, R.; Bright, G. Development of an UAV for search rescue applications. In Proceedings of the IEEE Africon '11, Victoria Falls, Zambia, 13–15 September 2011; pp. 1–6. [\[CrossRef\]](#)
- Obermeyer, K.; Oberlin, P.; Darbha, S. Sampling-based roadmap methods for a visual reconnaissance UAV. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010; p. 7568.
- Samaras, S.; Diamantidou, E.; Ataloglou, D.; Sakellariou, N.; Vafeiadis, A.; Magoulianitis, V.; Lalas, A.; Dimou, A.; Zarpalas, D.; Votis, K.; et al. Deep learning on multi sensor data for counter UAV applications—A systematic review. *Sensors* **2019**, *19*, 4837. [\[CrossRef\]](#)
- Agbeyangi, A.O.; Odieta, J.O.; Olorunlome, A.B. Review on UAVs used for aerial surveillance. *J. Multidiscip. Eng. Sci. Technol.* **2016**, *3*, 5713–5719.
- Lim, H.; Park, J.M.; Lee, D.; Kim, H.J. Build Your Own Quadrotor: Open-Source Projects on Unmanned Aerial Vehicles. *IEEE Robot. Autom. Mag.* **2012**, *19*, 33–45. [\[CrossRef\]](#)
- Goel, A.; Salim, A.M.; Ansari, A.; Ravela, S.; Bernstein, D. Adaptive digital PID control of a quadcopter with unknown dynamics. *arXiv* **2020**, arXiv:2006.00416.
- Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors* **2019**, *19*, 24. [\[CrossRef\]](#)
- Zhou, J.; Deng, R.; Shi, Z.; Zhong, Y. Robust cascade PID attitude control of quadrotor helicopters subject to wind disturbance. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 6558–6563.
- Park, D.; Yu, H.; Xuan-Mung, N.; Lee, J.; Hong, S.K. Multicopter PID Attitude Controller Gain Auto-tuning through Reinforcement Learning Neural Networks. In Proceedings of the 2019 2nd International Conference on Control and Robot Technology, Jeju, Republic of Korea, 12–14 December 2019; pp. 80–84.
- Mo, H.; Farid, G. Nonlinear and Adaptive Intelligent Control Techniques for Quadrotor UAV—A Survey. *Asian J. Control* **2017**, *21*, 989–1008. [\[CrossRef\]](#)
- Ru, P.; Subbarao, K. Nonlinear model predictive control for unmanned aerial vehicles. *Aerospace* **2017**, *4*, 31. [\[CrossRef\]](#)
- Ha, C.; Zuo, Z.; Choi, F.B.; Lee, D. Passivity-based adaptive backstepping control of quadrotor-type UAVs. *Robot. Auton. Syst.* **2014**, *62*, 1305–1315. [\[CrossRef\]](#)
- Fang, Z.; Gao, W. Adaptive backstepping control of an indoor micro-quadrotor. *Res. J. Appl. Sci. Eng. Technol.* **2012**, *4*, 4216–4226.
- Zhao, B.; Xian, B.; Zhang, Y.; Zhang, X. Nonlinear robust sliding mode control of a quadrotor unmanned aerial vehicle based on immersion and invariance method. *Int. J. Robust Nonlinear Control* **2015**, *25*, 3714–3731. [\[CrossRef\]](#)
- Besnard, L.; Shtessel, Y.B.; Landrum, B. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *J. Frankl. Inst.* **2012**, *349*, 658–684. [\[CrossRef\]](#)

22. Guruganesh, R.; Bandyopadhyay, B.; Arya, H.; Singh, G. Design and hardware implementation of autopilot control laws for MAV using super twisting control. *J. Intell. Robot. Syst.* **2018**, *90*, 455–471. [[CrossRef](#)]
23. Zheng, E.H.; Xiong, J.J.; Luo, J.L. Second order sliding mode control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 1350–1356. [[CrossRef](#)]
24. Gong, X.; Hou, Z.C.; Zhao, C.J.; Bai, Y.; Tian, Y.T. Adaptive backstepping sliding mode trajectory tracking control for a quad-rotor. *Int. J. Autom. Comput.* **2012**, *9*, 555–560. [[CrossRef](#)]
25. Noordin, A.; Basri, A.; Mohamed, Z.; Zainal Abidin, A.F. Modelling and PSO fine-tuned PID control of quadrotor UAV. *Int. J. Adv. Sci. Eng. Technol.* **2017**, *7*, 1367. [[CrossRef](#)]
26. Jing, Y.; Wang, X.; Heredia-Juesas, J.; Fortner, C.; Giacomo, C.; Sipahi, R.; Martinez-Lorenzo, J. PX4 Simulation Results of a Quadcopter with a Disturbance-Observer-Based and PSO-Optimized Sliding Mode Surface Controller. *Drones* **2022**, *6*, 261. [[CrossRef](#)]
27. Boubertakh, H. Optimal Stabilization of A Quadrotor UAV by a Constrained Fuzzy Control and PSO. *MATEC Web Conf.* **2017**, *99*, 03001. [[CrossRef](#)]
28. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95–International Conference on Neural Networks, Perth, WA, Australia, 27 November– 1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
29. Spitzer, A.; Michael, N. Rotational Error Metrics for Quadrotor Control. *arXiv* **2020**, arXiv:cs.RO/2011.11909.
30. Abdelmaksoud, S.I.; Mailah, M.; Abdallah, A.M. Control strategies and novel techniques for autonomous rotorcraft unmanned aerial vehicles: A review. *IEEE Access* **2020**, *8*, 195142–195169. [[CrossRef](#)]
31. Mueller, M.W. Multicopter attitude control for recovery from large disturbances. *arXiv* **2018**, arXiv:cs.RO/1802.09143.
32. Alaimo, A.; Artale, V.; Milazzo, C.; Ricciardello, A. Comparison between Euler and quaternion parametrization in UAV dynamics. *Aip Conf. Proc.* **2013**, *1558*, 1228–1231.
33. Jing, Y.; Zhihao, C.; Wang, Y. Modeling of The Quadrotor UAV Based on Screw Theory via Dual Quaternion. In Proceedings of the AIAA Modeling and Simulation Technologies (MST) Conference, Boston, MA, USA, 19–22 August 2013; p. 4594.
34. Kehlenbeck, A. Quaternion-based control for aggressive trajectory tracking with a micro-quadrotor UAV. Ph.D. Thesis, University of Maryland, College Park, MD, USA, 2014.
35. Kahouadji, M.; Choukchou-Braham, A.; Mokhtari, M.; Cherki, B. Super twisting control for attitude tracking using quaternion. *PET* **2018**, *36*, 35–39.
36. Chen, F.; Jiang, R.; Zhang, K.; Jiang, B.; Tao, G. Robust Backstepping Sliding-Mode Control and Observer-Based Fault Estimation for a Quadrotor UAV. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5044–5056. [[CrossRef](#)]
37. Xiong, J.J.; Zheng, E.H. Position and attitude tracking control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 725–731. [[CrossRef](#)]
38. Shi, X.; Cheng, Y.; Yin, C.; Shi, H.; Huang, X. Actuator fault tolerant controlling using adaptive radical basis function neural network SMC for quadrotor UAV. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5409–5414. [[CrossRef](#)]
39. Merheb, A.R.; Noura, H.; Bateman, F. Passive fault tolerant control of quadrotor UAV using regular and cascaded Sliding Mode Control. In Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France, 9–11 October 2013; pp. 330–335. [[CrossRef](#)]
40. Mofid, O.; Mobayen, S.; Wong, W.K. Adaptive Terminal Sliding Mode Control for Attitude and Position Tracking Control of Quadrotor UAVs in the Existence of External Disturbance. *IEEE Access* **2021**, *9*, 3428–3440. [[CrossRef](#)]
41. Rosales, C.; Gimenez, J.; Rossomando, F.; Soria, C.; Sarcinelli-Filho, M.; Carelli, R. UAVs Formation Control With Dynamic Compensation Using Neuro Adaptive SMC. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 93–99. [[CrossRef](#)]
42. Zhou, F.; Zhou, Y.J.; Jiang, G.P.; Cao, N. Adaptive tracking control of quadrotor UAV system with input constraints. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 5774–5779. [[CrossRef](#)]
43. Eltayeb, A.; Rahmat, M.F.; Basri, M.A.M.; Eltoum, M.A.M.; El-Ferik, S. An Improved Design of an Adaptive Sliding Mode Controller for Chattering Attenuation and Trajectory Tracking of the Quadcopter UAV. *IEEE Access* **2020**, *8*, 205968–205979. [[CrossRef](#)]
44. Abro, G.E.M.; Zulkifli, S.A.B.; Asirvadam, V.S.; Ali, Z.A. Model-free-based single-dimension fuzzy SMC design for underactuated quadrotor UAV. *Actuators* **2021**, *10*, 191. [[CrossRef](#)]
45. Zheng, E.; Xiong, J. Quad-rotor unmanned helicopter control via novel robust terminal sliding mode controller and under-actuated system sliding mode controller. *Optik* **2014**, *125*, 2817–2825. [[CrossRef](#)]
46. Beard, R.W. Quadrotor dynamics and control. *Brigh. Young Univ.* **2008**, *19*, 46–56.
47. Suleiman, H.U.; Mu'azu, M.B.; Zarma, T.A.; Salawudeen, A.T.; Thomas, S.; Galadima, A.A. Methods of Chattering Reduction in Sliding Mode Control: A Case Study of Ball and Plate System. In Proceedings of the 2018 IEEE 7th International Conference on Adaptive Science Technology (ICAST), Accra, Ghana, 22–24 August 2018; pp. 1–8. [[CrossRef](#)]
48. Guldner, J.; Utkin, V. The chattering problem in sliding mode systems. In *MTNS2000 Mathematical Theory of Networks and Systems*; Universite de Perpignan: Perpignan, France, 2000.

49. Nobile, M.S.; Cazzaniga, P.; Besozzi, D.; Colombo, R.; Mauri, G.; Pasi, G. Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm Evol. Comput.* **2018**, *39*, 70–85. [[CrossRef](#)]
50. Moeini, A.; Rafique, M.A.; Xue, Z.; Lynch, A.F.; Zhao, Q. Disturbance Observer-Based Integral Backstepping Control for UAVs. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 382–388.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.