Unmanned Systems, Vol. 0, No. 0 (2013) 1–17
© World Scientific Publishing Company

A Digital Twin-Level Modeling Framework for UAV Control Algorithm Optimization and Sim2real

Xunhua Dai[a] , Wanqi Gong[a], Jinhu Tu[b], Yong Chen[b,*]

[a]School of Computer Science and Engineering, Central South University, Hunan, China

[b]School of Automation, Central South University, Hunan, China
E-mail: chenyong@csu.edu.cn

As multicopter unmanned aerial vehicles (UAVs) are increasingly adopted in industrial and manufacturing applications, there is a growing demand for robust, efficient, and cost-effective control algorithm development. However, traditional trial-and-error approaches relying heavily on real-flight experiments remain time-consuming and resource-intensive, particularly in dynamic or uncertain environments. To address these challenges, this paper presents a digital twin-level modeling framework that enables high-fidelity simulation, accelerated controller design, and seamless sim2real transfer. The framework integrates detailed physical modeling of multicopter dynamics, realistic sensor emulation, and hardware-in-the-loop (HIL) simulation within a modular architecture. Each subsystem—including propulsion, aerodynamics, ground interaction, and onboard sensors—is systematically constructed and verified using experimental data. A representative application is demonstrated by designing a model-based linear quadratic regulator (LQR) controller trained using hybrid data composed of digital simulations and a small amount of real-world flight data. The controller is evaluated under external wind disturbances and benchmarked against baseline PX4 and hand-tuned PID controllers. Quantitative comparisons using standard performance metrics demonstrate that the hybrid-trained LQR outperforms alternatives while significantly reducing tuning time and testing costs. The results confirm the proposed digital twin framework as a promising tool for enhancing the efficiency, reliability, and scalability of UAV control algorithm development.

Keywords: high precision model; HIL; sim2real; UAV; digital twin.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have gained significant traction in the domains of smart manufacturing, industrial automation, and logistics.[1,2] From warehouse inventory monitoring and fault detection in large-scale factories to material transportation and real-time inspection of production lines, UAVs offer a unique advantage in terms of mobility, flexibility, and scalability.[3–5] As manufacturing environments become increasingly intelligent and interconnected, UAVs are expected to serve as mobile robotic agents capable of collaborating with ground systems and adapting to dynamic conditions in real time.

Despite their growing importance, the performance and safety of UAVs remain highly dependent on the quality of their control algorithms.[6] Traditional PID controllers are often insufficient for complex, fast-changing environments where disturbance rejection, trajectory tracking, and system robustness are critical.[7] More advanced model-based control algorithms, such as Linear Quadratic Regulators (LQR)[8] or Model Predictive Control (MPC),[9] offer promising alternatives but require accurate system models to achieve optimal performance.[10,11] In this context, control algorithm development is often hindered by two fundamental issues:

(1) The difficulty in building high-precision dynamic models that capture nonlinear aerodynamics, sensor latency, and actuator behavior;
(2) The lack of an efficient, cost-effective workflow for testing and tuning controllers before deployment.

Accurate dynamic modeling is fundamental to the design and optimization of UAV control systems. Traditional multicopter models are often based on simplified Newton–Euler equations that assume quasi-static aerodynamics and rigid-body dynamics.[12] While such models suffice for basic flight control and academic demonstrations, they fail to capture real-world effects such as motor delay, blade flapping, aerodynamic drag, and sensor latency, which are critical in high-precision applications.[13,14] High-precision modeling efforts have introduced more comprehensive representations, incorporating wind tunnel–derived aerody-

*Corresponding author.

namic coefficients, propulsion system nonlinearities, and hardware-level disturbances.[15,16] However, these models are frequently handcrafted for specific platforms, making them hard to generalize and expensive to scale. Moreover, validation is typically carried out in either purely simulated or purely physical environments, limiting the transferability of the control algorithms across domains.

PID controllers remain the de facto standard in commercial multicopter autopilots (e.g., PX4, ArduPilot), largely due to their simplicity and robustness.[17] However, for demanding tasks such as aggressive maneuvering, wind compensation, or autonomous path tracking, model-based controllers—including LQR, H∞ control, and MPC—offer more consistent performance. Despite their theoretical advantages, these advanced controllers require accurate state-space models and real-time computation capabilities. Many works adopt offline system identification using flight logs[18] or grey-box modeling approaches that blend analytical structures with empirical data.[19–21] Nonetheless, most studies still rely heavily on manual tuning, and few demonstrate how to streamline the control development process in a scalable and reproducible manner.

Digital twin (DT) technology[22] presents a promising solution to these challenges. In manufacturing and robotics, DTs are increasingly used to support predictive maintenance, virtual commissioning, and system optimization.[23,24] A digital twin is more than just a simulation—it is a dynamic, bidirectionally connected model of the physical system that evolves with time and data. DT technology has recently gained momentum in robotics and industrial systems, driven by the need for predictive analytics, fault diagnosis, and virtual commissioning.[25,26] In manufacturing contexts, DTs enable virtual replicas of physical systems to support process optimization, resource planning, and decision-making under uncertainty. For UAV applications, a DT can serve as a platform to continuously test and refine control strategies under realistic physical constraints and environmental disturbances. In UAV research, digital twin concepts have been applied for structural health monitoring,[27] battery state prediction,[28] and fleet management.[29] However, relatively little attention has been paid to their application in real-time control algorithm development. Most existing DT frameworks in aerospace and robotics focus on structural design or maintenance prediction rather than on real-time control algorithm optimization. Furthermore, they often lack the hybrid-fidelity modeling capability and hardware-in-the-loop (HIL) integration needed to simulate complex interactions between UAV hardware, software, and the environment. Most importantly, they seldom support the sim2real[30] workflow that bridges high-fidelity simulation with HIL testing, a crucial step in UAV controller deployment.

Taken together, current practices in UAV modeling and control development face several limitations: simulation environments often lack sufficient fidelity or integration with real-world data; model-based controllers are difficult to scale due to the cost of data acquisition and parameter identification; and digital twin frameworks are underutilized in the context of control algorithm verification and deployment. These gaps highlight the need for an integrated, DT-driven modeling and optimization framework that supports the entire lifecycle of UAV control algorithm development—from model identification and virtual testing to hardware validation and deployment. To address these limitations, this paper proposes a digital twin-level modeling framework designed specifically for UAV control algorithm optimization and sim2real transfer. The framework combines high-fidelity dynamic modeling with real-time simulation and synthetic flight data generation, enabling developers to iteratively design, evaluate, and deploy control algorithms with reduced reliance on time-consuming and risky field experiments. By supporting both low- and high-fidelity simulation modes and integrating physical system feedback, the framework enhances development agility while maintaining modeling accuracy.

Fei[22] discussed the challenges of advancing digital twin maturity from a macro perspective, emphasizing their importance for enabling large-scale industrial applications in the future. Putranto[23] proposed a DT framework that primarily focuses on calibration and communication technologies in digital twin scenarios, addressing the challenges of sensor deployment in enclosed environments. Wang[24] integrates deep learning with DT, where synthetic data from both virtual and real environments are leveraged to improve the efficiency of human–machine collaboration. Zhang[25] presents a DT framework for tunnel fire safety management, driven by dynamic sensors and AIoT technologies, which enables effective fire prediction when coupled with AI models. Zhang[26] develops a DT model that utilizes IoT sensors to transmit real-time data, combined with deep learning methods to achieve online optimization of operational performance. In contrast, the framework proposed in this study is specifically tailored for UAV applications, with a primary focus on dynamic flight scenarios. Unlike the aforementioned works by Alan, Wang, Zhang and Zhang, which are designed for relatively static subjects and environments, our framework incorporates high-precision kinematic modeling to capture the motion characteristics of UAVs. Moreover, while our approach shares some similarity with Wang's framework in leveraging both virtual and physical environments for accelerated data-driven optimization, our design is more open and flexible. The synthetic and real data generated within our DT system can be readily applied by developers to optimize algorithms for diverse UAV control objectives, thereby broadening its applicability and practical value.

In addition, several powerful UAV simulators, such as AirSim,[31] Gazebo,[32] FastSim[33] and IAP,[34] are currently available. However, these simulators are primarily oriented toward applications in perception, planning, and other algorithmic domains where high-fidelity modeling of UAV attitude dynamics is not critical. In contrast, digital twin applications for multirotor UAVs place stricter demands on the accuracy of attitude and motion representation. The digital models embedded in the aforementioned sim-

ulators are therefore insufficient to meet the precision requirements of DT-based multirotor applications. Compared with these simulators, the modeling approach adopted in this study provides high-fidelity models that can be flexibly constructed through system identification experiments and further adapted to digital twin scenarios.

The main contributions of this paper are summarized as follows:

(1) A modular co-simulation architecture that integrates dynamic models of propulsion, aerodynamics, and sensor-actuator systems at multiple fidelity levels.
(2) A digital twin workflow that fuses real-world data with simulation outputs to support robust parameter identification and model verification.
(3) An efficient method for controller optimization using virtual flight testing and hybrid datasets, demonstrated through LQR-based controller design under wind disturbance.
(4) Experimental validation using the FEISI-X150 UAV platform and HIL simulations, highlighting improved performance and reduced development time.

The remainder of this paper is organized as follows. Section 2 reviews related work in UAV modeling, digital twins, and control algorithm development. Section 3 introduces the overall architecture of the proposed digital twin-level modeling framework. Section 4 details the modeling of subsystems and the data-driven identification approach. Section 5 presents the controller optimization workflow and experimental setup. Section 6 provides evaluation results and discussions. Finally, Section 7 concludes the paper and outlines future research directions.

2. Introduction to UAV digital system

In the development of unmanned aerial vehicle (UAV) control algorithms, transitioning from simulation to physical implementation often incurs substantial development costs and time, primarily due to discrepancies between simulated environments and real-world conditions. To address this challenge and enhance the efficiency of control algorithm development, this study builds upon our previous work[35] to develop a UAV digital twin system aimed at accelerating the sim2real transition. The proposed digital twin system is designed to facilitate seamless integration between virtual simulations and physical UAV operations, thereby streamlining the development process. This section presents a comprehensive overview of the digital twin framework, detailing its framework, communication design, data generation processes.

2.1. Digital twin framework

The proposed digital twin-level framework is structured into three primary components: the digital system, the physical system, and the control algorithm development layer, as showed in Fig. 1. The digital system builds upon a high-fidelity hardware-in-the-loop (HIL) simulation platform presented in our previous work[35] , which provides both high hardware reliability and software credibility. This foundation significantly narrows the gap between simulation and physical deployment, enabling realistic behavior emulation and robust control validation.

To facilitate accurate modeling of UAV dynamics, we introduce a generic high-precision modeling methodology that enables efficient mapping of target UAVs from the physical domain into the digital system. Real-time information exchange—including UAV position, velocity, and attitude—is enabled between the digital and physical systems via MAVLink and ROS middleware protocols, ensuring synchronization across platforms.

The digital twin incorporates a simulated UAV equipped with virtual sensors that mirror the configuration of the physical UAV. These virtual sensors, like their physical counterparts, can log sensor data and positional information into flight logs or stream them directly via MAVLink/ROS. This capability allows for the generation of high-quality synthetic data, which is instrumental in developing and iterating control algorithms. The digital twin framework developed in this study is built upon the RflySim simulation platform,[36] which integrates PX4, MATLAB/Simulink, ROS, and related intelligent hardware. This toolchain enables both simulation and practical experiments in areas such as autonomous agent control, swarm coordination, and vision-based applications. It supports essential research processes including unmanned system modeling, controller design, SIL and HIL testing. The proposed digital twin framework was deployed on a high-performance computing host equipped with a 12th Gen Intel(R) Core(TM) i7-12700F processor and 32GB of memory. In the latest version of RflySim, the overall simulation frequency reaches 800Hz, with sensor response at 248Hz, PWM output at 232Hz, and position updates at 52Hz. These performance metrics ensure that the HIL simulations can meet the real-time requirements of high-speed UAV maneuvers. Further details on the parameter specifications of the RflySim simulation environment can be found at the following website:

`https://rflysim.com/doc/zh/`

Once optimized, control algorithms can be deployed uniformly to both the physical UAV and the HIL-based virtual UAV using the same autopilot hardware, ensuring consistency across the entire development and testing pipeline. The proposed framework offers a cost-effective and efficient solution for generating development-critical datasets in the digital domain, thus accelerating the overall control algorithm lifecycle.

By integrating high-accuracy modeling techniques and HIL simulation, the framework minimizes the sim2real gap more effectively than conventional UAV simulation platforms. As a result, control algorithms developed within this digital twin framework demonstrate superior transferability

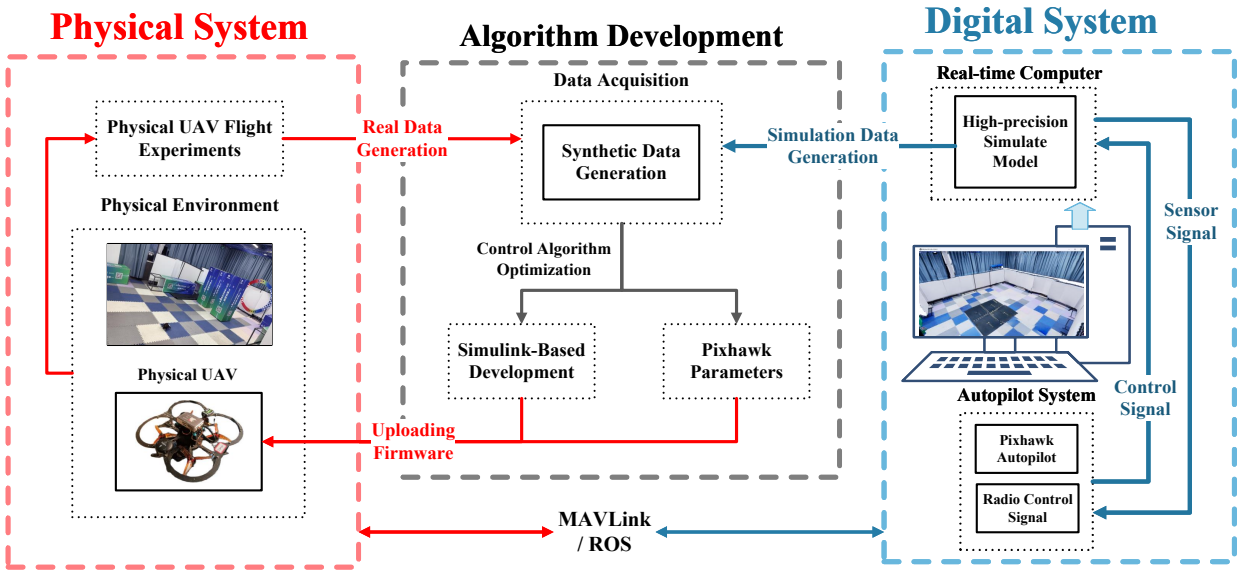


Fig. 1. Theoretical framework for UAV control algorithm optimization and sim2real using digital twin technology.

and reliability when deployed in real-world environments.

The framework emphasizes the integration of high-fidelity digital modeling with sim2real iterative data refinement. Although multicopter UAVs are selected as the primary focus in this study due to their prevalence and practicality, the framework is designed to be generalizable across different unmanned aerial vehicle types. For example, virtual simulations of fixed-wing and VTOL hybrid UAVs have already been realized in our previous work.[36] With the development of high-precision models for other platforms, the proposed digital twin framework can be readily extended to a broader range of unmanned vehicle systems.

2.2. Communication design of the digital twin

In conventional UAV simulation platforms such as Gazebo and AirSim, the entire process of design, simulation, and testing is typically confined to a closed virtual environment, without interaction with external physical systems. In contrast, a digital twin system requires not only simulation capabilities but also real-time interaction with the physical domain. To support this cyber-physical integration, the data transmission architecture of our digital twin framework is depicted in Fig. 2.

ROS is a distributed middleware framework that enables synchronized data transmission between the digital and physical systems when long-range communication is not required. The MAVLink communication protocol, which is widely supported by mainstream flight control systems such as PX4 and ArduPilot, features a lightweight and efficient binary transmission format. Its capability for real-time control and in-flight debugging makes it particularly suitable for ensuring synchronized data exchange between the digital and physical components in a digital twin system. As shown in Fig. 2, the digital system is implemented with a hardware-in-the-loop (HIL) simulation backend, enabling direct control signal input from a remote controller or autopilot hardware. In more advanced experimental scenarios, such as when the physical UAV is tracked using a motion capture system, rigid-body pose data are transmitted to the host system via Ethernet. Through the combination of ROS and MAVLink, both the digital and physical UAVs can access synchronized control commands and telemetry feedback. This allows the two systems to operate in coordination, maintaining consistent states such as position and orientation, thus enabling real-time, closed-loop co-execution and validation of control strategies.

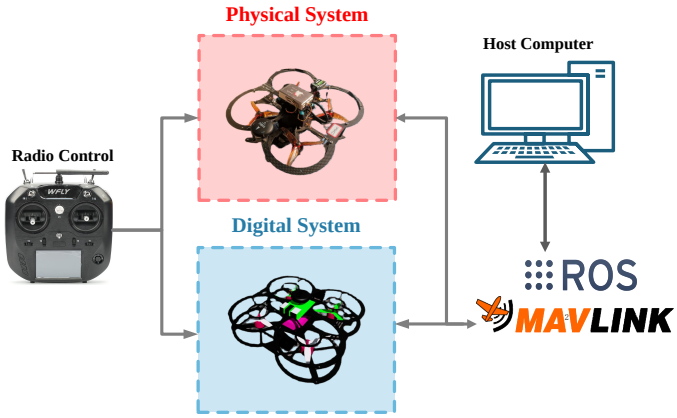


Fig. 2. Communication design of the digital twin.

2.3. Generate data and code

Model-Based Design (MBD)[37] has emerged as an effective methodology for addressing credibility and reliability challenges in the development of complex autonomous systems.

By leveraging modular visual programming and automatic code generation, MBD standardizes the modeling, development, and testing processes, thereby reducing the impact of human-induced programming errors and inconsistencies in development workflows. Tools such as MATLAB/Simulink, one of the most widely adopted platforms for MBD, ensure that the automatically generated code complies with stringent industrial standards and certification guidelines, such as DO-178C.

In the MBD approach, the simulation system is decomposed into modular and reusable subsystems—such as kinematic models, GPS modules, ground interaction components, and propulsion units—which aligns well with the high-fidelity, hierarchical modeling philosophy adopted in the proposed digital twin-level framework. This modularity enables efficient scalability and reuse across different types of UAVs and autonomous platforms, significantly enhancing the efficiency of the design, verification, qualification, and certification processes within the unmanned systems domain.

Within the proposed digital twin system, the MBD methodology is utilized to streamline the development and iterative refinement of the UAV digital model. Automatic code generation not only accelerates the deployment of updated system models in the simulation environment but also facilitates direct firmware generation for real-world embedded hardware, thereby minimizing the overall development cycle for control algorithms.

Moreover, the integration of the digital twin framework with MBD significantly simplifies the acquisition of test data. While physical flight testing remains valuable, it is often time-consuming and costly. The digital twin enables large-scale, high-fidelity simulation experiments to be conducted efficiently, generating extensive synthetic datasets. When combined with selected real-world flight data, these simulated datasets enhance the robustness, optimization, and verification of control algorithms—ultimately improving the reliability and efficiency of the UAV control system development pipeline.

3. The simulation UAV model for sim2real

A reliable and generalizable digital system forms the foundation for efficient sim2real development of UAV control algorithms. Therefore, building a high-fidelity digital model of the UAV is a critical first step. Our previous work [35] established a modular, system-level modeling framework for UAVs. In this paper, we focus specifically on multicopter UAVs. As showed in Fig. 3, a model-based design (MBD) approach is adopted to decompose the multicopter system into several functional subsystems. The following sections provide a detailed description of the modeling methodology.

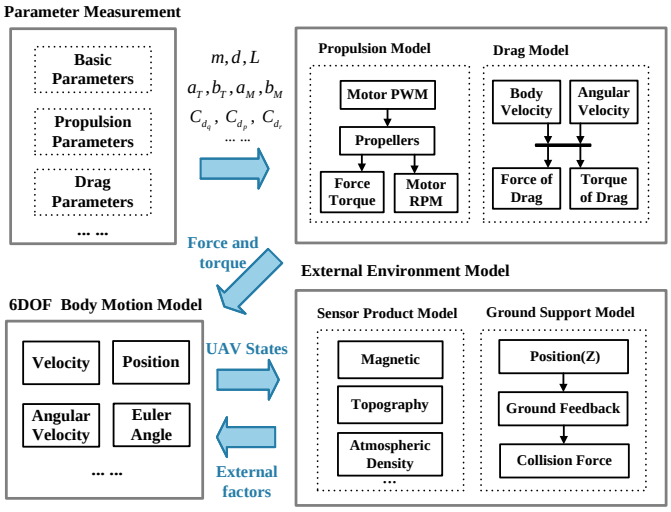


Fig. 3. The modular system-level modeling framework of multicopter high-precision models.

3.1. High precision modeling

3.1.1. 6DOF Body Motion Model

This paper establishes two coordinate frames. The first frame, denoted as "e", denotes the North-East-Down (NED) world frame, with its origin fixed at the vehicle's center of gravity. The second frame, referred to as "b", is the Head-Right-Down (HRD) body-fixed frame. The orientations of the $x_e - y_e - z_e$ axis and $x_b - y_b - z_b$ axis illustrated in Fig.4. Additionally, $\Theta = [\phi, \gamma, \chi]^T$ denotes the Euler angles that characterize the body attitude.

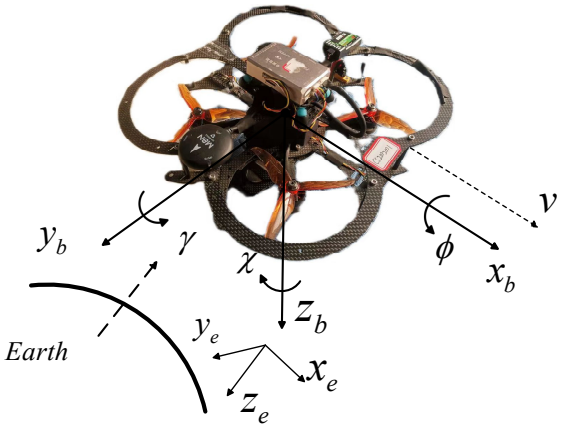


Fig. 4. The world and body coordinate frames for multicopter.

By Newton-Euler equations, the 6DOF dynamics of

the multicopter model is formulated as follows :

$$\begin{cases} {}^{e}\dot{p} = {}^{e}v \\ m{}^{e}\dot{v} = R_b^{e\,b}F + mg \\ \dot{\Theta} = C_b^e \cdot {}^{b}\omega \\ J^b\dot{\omega} = -{}^{b}\dot{\omega} \times \left(J^b\dot{\omega}\right) + {}^{b}M \end{cases} \tag{1}$$

where m denotes the mass of the multicopter, g denotes the gravity acceleration, $J \in \mathbb{R}^{3\times3}$ denotes the torque of inertia matrix, ${}^{e}p \in \mathbb{R}^3$ denotes the current position of the UAV, R_b^e is the rotation matrix to transform a vector from the body frame to the world frame, C_b^e is the rotation matrix from the angular velocity in the body frame to the Eulerian angular velocity, ${}^{e}v \in \mathbb{R}^3$ is the velocity vectors in the world frame, ${}^{b}\omega \in \mathbb{R}^3$ are the angular velocity vectors in the body frame, ${}^{b}F \in \mathbb{R}^{\Lambda}$ and ${}^{b}M \in \mathbb{R}^{\Lambda}$ denote the force vector and torque vector in the body frame, respectively.

The 6DOF model describes the UAV's motion under applied forces and torques. However, its general formulation may omit critical details, limiting its accuracy within a digital twin framework. This study focuses on modeling the individual forces and torques acting on the system, employing a modular design approach to analyze their composition. Building on our previous work [1], the total force vector ${}^{b}F \in u_{body}$ is expressed as the sum of multiple contributing components

$${}^{b}F = {}^{b}F_{drag} + {}^{b}F_{coll} + \sum {}^{b}F_{lift,i} \tag{2}$$

Similarly, the total moment vector ${}^{b}M \in u_{body}$ can be represented as a superposition of individual torque sources

$${}^{b}M = {}^{b}M_{drag} + {}^{b}M_{grav} + {}^{b}M_{coll} + \sum {}^{b}M_{lift,i} \tag{3}$$

where ${}^{b}F_{drag} \in \mathbb{R}^3$ denotes the drag vector, ${}^{b}F_{coll} \in \mathbb{R}^3$ denotes the physical collision force vector originating from the ground support, $\sum {}^{b}F_{lift,i} \in \mathbb{R}^3$ denotes the lift vector of one of the multicopter's propellers, ${}^{b}M_{drag} \in \mathbb{R}^3$ denotes the drag torque vector, ${}^{b}M_{grav} \in \mathbb{R}^3$ denotes the gravity torque vector, ${}^{b}M_{coll} \in \mathbb{R}^3$ denotes the physical collision torque vector originating from the ground support, and $\sum {}^{b}M_{lift,i} \in \mathbb{R}^3$ denotes the lift torque vector of one of the multicopter's propellers. Since gravity ${}^{e}F_{drag} = mg$ has already been accounted for separately from the resultant force ${}^{b}F$ in (1), it is excluded from the resultant force calculation.

3.1.2. Propulsion Model

Modeling the propulsion system of a multicopter should begin with the dynamics of a single rotor. Based on the specific vehicle configuration, the combined thrust of multiple rotors is then integrated to form a complete propulsion model. Commonly used rotor models take motor speed as the input. However, in this digital system, HIL simulation is implemented via an autopilot, which primarily outputs PWM signals. To ensure consistency, the propulsion model is designed to use PWM input.

The propeller model $S_{prop} = \{S_{lift}, S_{\omega}\}$, which operates based on input PWM signals, is divided into two distinct components: the sub-model S_{lift}, which is responsible for generating output force and torque, and the sub-model S_{ω}, which determines the output motor speed. This can be formulated as follows:

$$\begin{aligned} &y_{lift} = S_{lift}\left(u_{\text{prop}}\right), y_{lift} = \left\{{}^{b}F_{lift}, {}^{b}M_{lift}\right\} \\ &y_{\omega} = S_{\omega}\left(u_{\text{prop}}\right), y_{\omega} = \varpi \\ &u_{\text{prop}} \triangleq \sigma_i \end{aligned} \tag{4}$$

where σ_i denotes the normalized motor PWM signal inputs and ϖ denotes the moter speed.

The multicopter propulsion model can be decomposed into sub-models based on different input–output relationships, such as PWM to motor speed, PWM to thrust, PWM to torque, and motor speed to thrust. These sub-models can be identified or fitted using thrust test data. For example, the PWM-to-motor speed relationship can be represented using a polynomial model as follows:

$$\varpi = w_1\sigma_i + w_2{\sigma_i}^2 + \cdots + w_n{\sigma_i}^n \tag{5}$$

the coefficients denoted as $w = [w_1, w_2, \cdots, w_n]^T$ are derived from the identification process, while the polynomial order n can be determined individually based on the fitting results for different airframes. The identification results indicate that the relationship between PWM, propeller lift, and torque follows a general static pattern. Therefore, it can be assumed that the decomposed models for PWM to propeller lift and PWM to propeller torque exhibit static linear correlations, as shown below:

$$\begin{cases} {}^{b}F_{lift,i} = a_F\sigma_i + b_F \\ {}^{b}M_{lift,i} = a_M\sigma_i + b_M \end{cases} \tag{6}$$

where a_F and b_F denote the slope and intercept of the PWM-propeller lift curve, and a_M and b_M denote the slope and intercept of the PWM-propeller torque curve, respectively.

Upon finalizing the modeling of the propulsion generated by an individual propeller. The input u_{Rf} for the comprehensive propulsion model S_{Rf} of the aircraft is derived from the outputs of each propeller model (the propeller lift $y_{lift,i}$ and motor speed $y_{\omega,i}$), which can be expressed as:

$$\begin{aligned} &y_{Rf} = S_{Rf}(u_{Rf}) = \left\{{}^{b}F_{Rf}, {}^{b}M_{Rf}\right\} \\ &u_{Rf} = \{y_{lift,1}, \cdots, y_{lift,i}, \cdots\} \cup \{y_{\omega,1}, \cdots, y_{\omega,i}, \cdots\} \end{aligned}$$

In the absence of high-maneuverability motion or additional flight conditions, the generated forces and torques can be approximated by the inputs from the n propellers. A complete dynamic model must also account for pitch and roll moments. As the full formulation has been extensively detailed in prior work [2], it is omitted here for brevity.

3.1.3. Drag Model

The high-precision multicopter model presented in this paper is designed to support the efficient development of UAV algorithms and applications relevant to digital twin technologies. This model is intended for standard

flight scenarios and does not account for prolonged high-maneuverability actions or extreme environmental conditions. Based on these assumptions, a drag model is constructed according to the flight state during typical multicopter UAV missions. Thus, the calculation of drag primarily relies on the equivalent relative wind field velocity ${}^{b}_{win}v = [{}^{b}_{win}v_x, {}^{b}_{win}v_y, {}^{b}_{win}v_z]^T$ of the multicopter as a key input. Assuming the multicopter is in a stationary state, the equivalent relative wind field velocity in the body frame can be denoted as:

$$ {}^{b}_{win}v = -{}^{b}v + R^{b}_{e} \cdot {}^{e}W \tag{7} $$

where ${}^{e}W \in \mathbb{R}^3$ is the wind velocity disturbance vector.

The input for the calculation of the drag torque is the angular velocity ${}^{b}\omega$. The drag model S_{drag} is formulated as:

$$ \begin{aligned} y_{drag} &= S_{drag}(u_{drag}) \\ y_{drag} &= \left\{ {}^{b}F_{drag}, {}^{b}M_{drag} \right\} \\ u_{drag} &= \left\{ {}^{b}v, {}^{b}\omega \right\} \end{aligned} \tag{8} $$

In typical flight missions, UAVs experience gradual changes in attitude, with relatively small attitude angles, making them less susceptible to wind disturbance effects. Many current studies set the drag torque coefficients directly to zero. However, this paper considers the influence of both drag and drag torque. Based on experimental test results, we simplify the model by approximating the drag and drag torque of the multicopter as linear relationships:

$$ \begin{aligned} {}^{b}F_{drag} &= -C_{df} \cdot {}^{b}_{win}v \\ {}^{b}M_{drag} &= -C_{dm} \cdot {}^{b}_{win}\omega \\ C_{df} &= \{C_{df_x}, C_{df_y}, C_{df_z}\} \\ C_{dm} &= \{C_{dm_p}, C_{dm_q}, C_{dm_r}\} \end{aligned} \tag{9} $$

where $C_{df_x}, C_{df_y}, C_{df_z}$ are the drag coefficients for forward/backward, sideways flight and vertical flight direction. $C_{dm_p}, C_{dm_q}, C_{dm_r}$ are the drag torque coefficients in roll, pitch, and yaw directions.

The simplified drag and drag torque model is streamlined and intuitive, designed to meet the requirements of digital twin applications and to allow for the repeated use of a general dynamic model. The key aspect of this model lies in the accurate identification of drag and drag torque coefficients. To achieve this, multiple experiments were conducted to gather data, followed by data processing and coefficient identification. Detailed procedures are provided in Section III.

3.1.4. External Environment Model

To ensure accurate simulation across the full flight envelope—from takeoff to landing—this section addresses disturbances encountered both on the ground and in flight. Two models are presented: the collision model S_{coll} and the environmental model S_{env}. Due to the complexity of environmental effects, these sub-models are simplified to enable efficient and practical modeling.

The collision model ${}^{b}F_{coll}$is applied during takeoff and landing phases, with its output represented as the ground contact force ${}^{b}F_{coll}$. The UAV is approximated using simplified geometric shapes, and surface contact is modeled through a spring-damper system. Different ground types are characterized by distinct stiffness coefficients, as shown below.

$$ {}^{b}F_{coll} = \begin{cases} 0, & \Delta z > 0 \\ -k_1 \Delta z - k_2 \Delta z, & \Delta z \leq 0 \end{cases} \tag{10} $$

where k_1, k_2 denotes the elasticity coefficients, with varying values indicating different ground environments, Δz denotes the distance from the ground surface along the z-axis, $\Delta z > 0$ indicates that the body is positioned above the ground, $\Delta z < 0$ signifies that a support force is acting on the contact surface. Therefore, the input-output formulation of the collision model S_{coll} can be summarized as follows:

$$ \begin{aligned} y_{coll} &= S_{coll}(u_{coll}) \\ y_{coll} &= \left\{ {}^{b}F_{coll} \right\}, u_{coll} = \{\Delta z\} \end{aligned} \tag{11} $$

The flight environment model S_{env} integrates established, well-validated models to determine the UAV's environmental conditions based on its current position and orientation data. The mathematical expression for the flight environment model S_{env} can be represented as follows:

$$ \begin{aligned} y_{env} &= S_{env}\left(u_{env}\right) \\ y_{env} &= \{g, \rho, T, {}^{e}_{\text{win}}v, {}^{e}_{\text{win}}\omega, \cdots\} \\ u_{env} &= \left\{ {}^{e}p, {}^{b}v, \cdots \right\} \end{aligned} \tag{12} $$

where ρ and T are the air density and temperature at the current position, ${}^{e}_{win}\omega \in \mathbb{R}^3$ denotes the angular velocity of airflow rotation.

The flight environment model incorporates several established models. The World Geodetic System (WGS)[38] is used to calculates the UAV's latitude, longitude, and altitude (LLA) global positioning coordinates, denoted as ${}^{e}p_g = (\lambda, \varphi, r)^{\mathrm{T}}$, which then enables computation of the local gravitational acceleration. Here, λ denotes the latitude relative to the Earth's prime meridian, and r represents the elevation relative to the benchmark horizontal plane. Additionally, the actual gravitational potential of the earth can be obtained using the EGM2008 model;[39] the air density and temperature at the UAV's current location are derived from the International Standard Atmosphere (ISA) model.[40] The wind disturbance model follows the U.S. military standard MIL-F-8785C,[41] decomposing the wind disturbance vector into a summation of multiple wind fields. These wind fields are typically defined in the Earth coordinate system.

3.1.5. Sensor Model

Accurate sensor data is critical for UAV algorithm development. In this study, a digital twin-level sensor model for multicopter UAVs is developed, based on our previous

work .[42] The model consists of two parts: raw data generation and bias characterization.

Ideal raw sensor data are derived from UAV flight states and the 3D environment. For image sensors, data can be directly retrieved from the 3D engine. For most other sensors, appropriate transformations are required. For instance, IMU data—crucial for multicopter control—are computed by combining actual and gravitational accelerations to produce the measured acceleration $a \in \mathbb{R}^3$. Magnetometer readings are generated using the World Magnetic Model (WMM) ,[43] which maps the UAV's global position ${}^{e}P_g$ to a local magnetic field vector. Similarly, GPS LLA coordinates are synthesized by correlating UAV position with satellite data.

Ideal sensor outputs, generated based on underlying physical principles, often differ from real sensor measurements, reducing the portability and generalization of algorithms during development. To address this, the proposed sensor model incorporates noise and drift into the measured data y_{data} . Following the approach in ,[42] a given ideal measurement $x_m \in y_{data}$ is perturbed by stochastic noise and systematic drift. The resulting sensor output $x_m{}'$ can thus be expressed as:

$$\begin{cases} x_m{}' = x_m + b_a + n_a \\ \dot{b_a} = n_b \end{cases} \tag{13}$$

where $n_a \sim N(0, \sigma_a^2)$ and $n_b \sim N(0, \sigma_b^2)$ are zero-mean Gaussian noise vectors for inertial sensors.

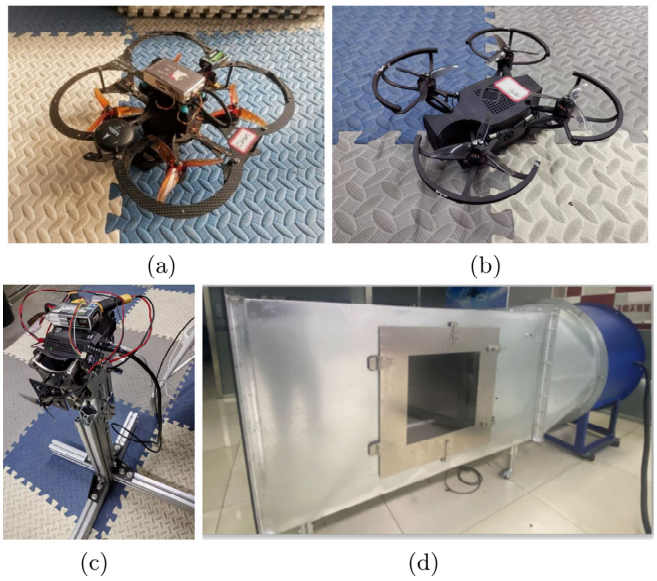


Fig. 5. The experimental equipment for parameter identification.(a)FEISI-X200 multicopter. (b)FEISI-X150 multicopter. (c)Thrust test stand. (d)The wind tunnel.

Table 1. Hardware Configuration Comparison Between FEISI-X200 and FEISI-X150

	FEISI-X200	FEISI-X150
Motors	T-Motor F40 KV255	FEISI 1505-KV4500
ESC	T-Motor 55A 4-in-1	HAKRC 25A 4-in-1
Propellers	T-Motor T5146	Genfan 3016R
Battery	4s 5300mAh Li-Po	3s 1300mAh Li-Po
Autopilot	ZY-H7	ZY-Racer

3.2. Model parameter measurement

This section presents two multicopter models as examples to outline the experimental setup required for parameter determination. It further explains the parameter-solving method and details the associated procedure.

3.2.1. Hardware Configuration

This section details the parameter identification approach for multicopter models, using the FEISI-X200 and FEISI-X150 multicopters as examples, designed for outdoor and indoor applications, respectively. The FEISI-X200 has a wheelbase of 200mm, while the FEISI-X150 has a wheelbase of 150mm. Fig 5 displays the hardware structures and configurations, with additional specifications provided in Table 1.

Once the structure and hardware configurations are established, parameter identification experiments can be conducted on the multicopter. Parameters such as body mass m, arm length d_i, and installation angle φ_i can be directly obtained through measurement without requiring experimental procedures. The parameters requiring identification are primarily within the propulsion and drag sub-models, Table.2. For the environment and sensor sub-models, existing data from established models is used. Thus, in the parameter identification experiments, thrust data generated by the propellers and drag data under wind resistance conditions are the key target measurements. Core tools for this include a DET G10/L10-10KGF thrust test stand, a three-dimensional force sensor, and a custom-built wind tunnel. The arrangement of the thrust test stand, force sensor, and wind tunnel is shown in Fig.5.

Table 2. Hardware Configuration Comparison Between FEISI-X200 and FEISI-X150

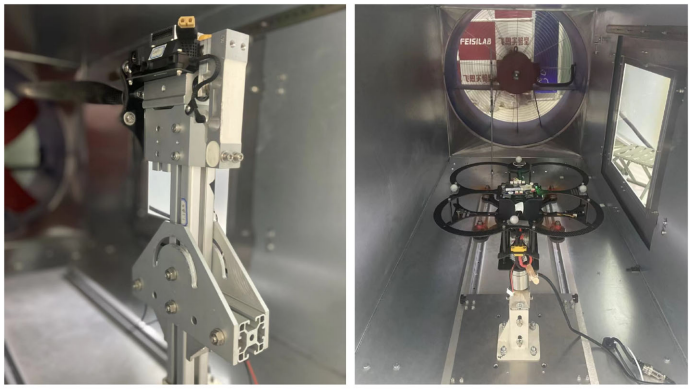
	Parameter Symbol
Basic Parameters	m, d_i, ϕ_i, J
Propulsion Parameters	w_i, a_{F}, b_{F}, a_{M}, b_{M}
Drag Parameters	C_{df_x}, C_{df_y}, C_{df_z}, C_{dm_p}, C_{dm_q}, C_{dm_r}

3.2.2. Propulsion Model Parameters

The propulsion sub-model uses PWM signals as inputs and outputs motor speed, thrust, and torque. During parameter identification, PWM signals are treated as inputs, while a thrust test stand measures the resulting forces and torques. Motor speed is obtained by analyzing three-phase current commutation. The collected data are accurate and suitable for analysis using identification tools. This study employs the System Identification Toolbox in MATLAB$^{\mathrm{TM}}$/Simulink® to facilitate model development.

The propulsion model parameters are identified through static thrust measurements, conducted under varying PWM inputs. The thrust test stand records the step response of thrust, torque, and motor speed. The collected data are then fitted to derive static relationships. Motor speed is modeled using a polynomial function to determine coefficients w_i, while thrust and torque are fitted with linear models to obtain coefficients $a_{\mathrm{F}}, b_{\mathrm{F}}$ and $a_{\mathrm{M}}, b_{\mathrm{M}}$, respectively. The resulting parameters are summarized in Table 3.

Dynamic force testing is conducted using the wind tunnel and the thrust test stand to assess propeller performance under varying inflow velocities (Fig. 6). This setup enables the measurement of thrust and torque at different wind speeds. The objective is to evaluate the robustness of the static propulsion model under dynamic conditions. As shown in Fig. 7, the model exhibits minimal deviation across wind speeds, supporting the validity of a linear approximation.



(a) (b)

Fig. 6. The experimental equipment for parameter identification.(a)FEISI-X200 multicopter. (b)FEISI-X150 multicopter. (c)Thrust test stand. (d)The wind tunnel.

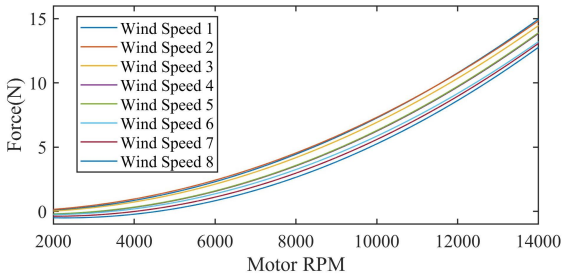


Fig. 7. The test result of verifying the adaptability and offset of the static model in a complex environment under the influence of different wind velocity.

3.2.3. Drag Model Parameters

To identify drag model parameters, steady-flight experiments are conducted using real-flight data. This approach focuses on low-frequency dynamics to estimate aerodynamic drag and corresponding coefficients. By leveraging the previously identified propulsion model, the relationship between steady-state velocity ${}^{e}v$ and thrust force u_x is established, enabling drag to be modeled as a function of flight speed. For steady motion along the UAV's x-axis, Newton's second law is applied as follows:

$$ma_x = u_x - f_{\mathrm{d_x}} \tag{14}$$

where u_x is the force in the x-direction in body frame. Due to the UAV is in uniform motion, the condition $f_{\mathrm{d_x}} = u_x$ holds, meaning that the flight drag along a given axis is equal to the flight force in that direction. As shown in Fig. 8, using the flight drag along the x-axis as an example, the relationship between the flight drag and the flight force can be expressed as follows:

$$F_{\mathrm{d_x}} = -F_{\mathrm{lift}} \cdot [0,0,1]^{\mathrm{T}} \cdot \sin\gamma \tag{15}$$

$$f_{\mathrm{d_x}} = u_x = F_{\mathrm{d_x}} \cdot \cos\gamma \tag{16}$$

where $F_{\mathrm{d_x}}$ denotes the drag in the world frame, $f_{\mathrm{d_x}}$ denotes the drag in the body frame. For example, in the forward flight direction, combined with (9) the drag coefficient is given as:

$$C_{\mathrm{df_x}} = \frac{f_{\mathrm{d_x}}}{{}^{\mathrm{b}}_{\mathrm{win}}v_{\mathrm{x}}} = \frac{F_{\mathrm{lift}} \cdot [0,0,1]^{\mathrm{T}} \cdot \sin\gamma \cdot \cos\gamma}{{}^{\mathrm{b}}_{\mathrm{win}}v_{\mathrm{x}}} \tag{17}$$

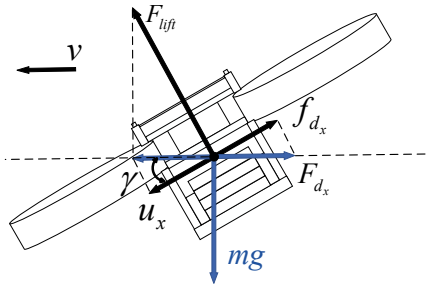


Fig. 8. Force analysis of multicopter in uniform forward flight along x-axis.

It is assumed that the angular velocity during rotation follows a viscous drag model (using roll angular velocity as an example): $\dot{p} = -C_{\mathrm{dm_p}} p$, then the solution for p can then be expressed as:

$$p = C_1 e^{-C_{\mathrm{dm_p}} t} \tag{18}$$

where $C_{\mathrm{dm_p}}$ is the drag torque coefficient, C_1 is a positive constant, and t denotes the time. The identification of the drag torque coefficient is thus transformed into the problem of identifying the natural damping coefficient of the angular velocity.

This paper uses the bifilar pendulum method[42] to determine the drag torque coefficient. Through corresponding experiments, test data for the roll, pitch, and yaw angular velocities of the FEISI-X200 are obtained. As an example, the calculation formula for the roll torque of inertia, measured using the double pendulum method, is as follows:

$$J_{\mathrm{xx}} = \frac{mgd^2}{16\pi^2 L} T_{\mathrm{n}}^2 \tag{19}$$

where d is the distance between the two ropes, L is the length of the rope, T_{n} is the period of simple harmonic oscillation. After conducting multiple double pendulum experiments, the angular velocity decay curves are extracted. By applying a logarithmic transformation to p based on (18), the derivative of each segment is computed to obtain the damping coefficients, that correspond to the drag coefficients. Combining (18) with (9) yields the following formula:

$$\frac{\mathrm{d}\ln p}{\mathrm{d}t} = -C_{\mathrm{dm_p}}, \frac{\mathrm{d}\ln q}{\mathrm{d}t} = -C_{\mathrm{dm_q}}, \frac{\mathrm{d}\ln r}{\mathrm{d}t} = -C_{\mathrm{dm_r}} \tag{20}$$

The results for all coefficients are shown in Table 3.

4. Digital twin model evaluation

4.1. Evaluation method

This study adopts the simulation credibility assessment method for multicopter UAVs proposed in our previous work[36] to evaluate the reliability of the digital model developed in this digital twin framework. The assessment further supports the overall effectiveness of the proposed digital twin system.

In,[36] a quantitative assessment method for evaluating the credibility of simulation models was introduced. This method defines two key metrics: the average amplitude curve error e_{amp} and the phase curve error e_{pha} as:

$$e_{amp} = \sqrt{\frac{1}{n_f}\sum_{1}^{n_f}\left(A_e(f_i) - A_s(f_i)\right)^2} \tag{21}$$

$$e_{pha} = \sqrt{\frac{1}{n_f}\sum_{1}^{n_f}\left(P_e(f_i) - P_s(f_i)\right)^2} \tag{22}$$

where n_f is the number of sample points within the frequency interval $[f_a, f_b]$, $A_e(f)$ and $P_e(f)$ denote the experimental amplitude and phase curves obtained from the actual system, while $A_s(f)$ and $P_s(f)$ refer to the simulated amplitude and phase curves generated by the simulation system. The thresholds for the average errors in amplitude and phase, e_{amp} and e_{pha}, are denoted as $\varepsilon_{amp} \in \mathbb{R}^+$and $\varepsilon_{pha} \in \mathbb{R}^+$, respectively. Thus, the model's credibility in terms of amplitude and phase, denoted as $\eta_{amp} \in (0,1)$ and $\eta_{pha} \in (0,1)$, η_{amp} can be expressed as follows:

$$\eta_{amp} = f_{norm}(e_{amp}, \varepsilon_{amp}) = \frac{K_e \cdot \varepsilon_{amp}}{\sqrt{(K_e \cdot \varepsilon_{amp})^2 + {e_{amp}}^2}} \tag{23}$$

where the coefficient $K_e \in \mathbb{R}^+$ is a scale factor. The calculation of η_{pha} follows the same form as (23), the overall frequency-domain credibility index, denoted as $\eta_f \in (0,1)$, is defined as follows:

$$\eta_f = \sqrt{\frac{1}{2}(\eta_{amp}^2 + \eta_{pha}^2)} \tag{24}$$

The evaluation of the digital model includes both time-domain and frequency-domain analyses. To further assess model fidelity, identical experiments are conducted in HIL simulations and on physical UAV platforms, enabling a direct comparison between the high-precision digital system and the physical system.

4.2. Time domain

To evaluate the key performance characteristics of the multicopter model in the time-domain, outdoor flight experiments and HIL simulations under identical conditions were designed. In the experiments, the FEISI-X200 will perform forward, lateral, and rotational movements controlled by a remote controller. The simulation model will replicate these tasks in the HIL, also controlled by the remote. Fig. 9 shows the experimental setup and scenario for these tests.



(a) (b)

Fig. 9. Force analysis of multicopter in uniform forward flight along x-axis.(a)Real flight experiment. (b)HIL experiment.

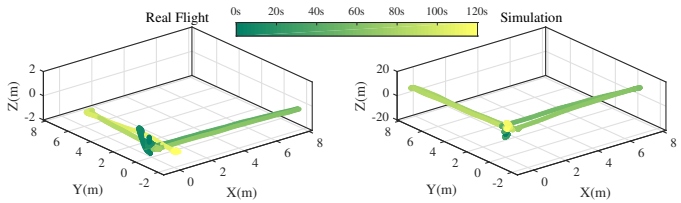
Fig. 10(a) compares the flight trajectories of the FEISI-X200 in real-world and simulated environments under manual control. The trajectories align well overall, though discrepancies appear later in the flight due to GPS noise and

Table 3. Multicopter model parameters identification results

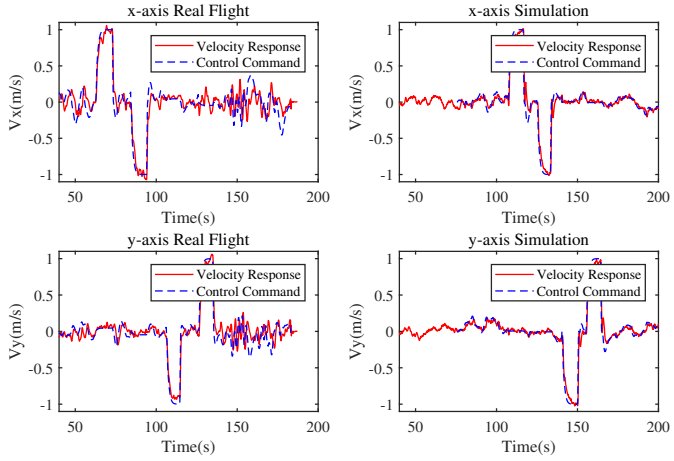
	FEISI-X200 Parameters	FEISI-X150 Parameters
Basic	$m = 1.0192kg$, $d_i = 0.1m$, $\phi_i = \left[\frac{\pi}{2}, \frac{3\pi}{2}, 0, \pi\right]$ $J = \left[4.05 \times 10^{-3}, 0, 0; 0, 5.69 \times 10^{-3}, 0; 0, 0, 6.35 \times 10^{-3}\right]$	$m = 0.3175kg$, $d_i = 0.075m$, $\phi_i = \left[\frac{\pi}{2}, \frac{3\pi}{2}, 0, \pi\right]$ $J = \left[5.24 \times 10^{-4}, 0, 0; 0, 6.93 \times 10^{-4}, 0; 0, 0, 8.82 \times 10^{-4}\right]$
Propulsion	$a_{\mathrm{F}} = 4.0568$, $b_{\mathrm{F}} = 1.3730$, $a_{\mathrm{M}} = 0.1094$, $b_{\mathrm{M}} = -0.0215$ $w_1 = -0.954 \times 10^3$,$w_2 = -0.204 \times 10^3$,$w_3 = 3.768 \times 10^3$	$a_{\mathrm{F}} = 1.4874$, $b_{\mathrm{F}} = -0.0147$, $a_{\mathrm{M}} = 0.03$, $b_{\mathrm{M}} = 0$ $w_1 = 0.548 \times 10^3$,$w_2 = -1.80 \times 10^3$,$w_3 = 3.608 \times 10^3$
Drag	$C_{\mathrm{df}_x} = 0.3684$, $C_{\mathrm{df}_y} = 0.3274$, $C_{\mathrm{df}_z} = 0.5$ $C_{\mathrm{dm}_p} = 0.0062$, $C_{\mathrm{dm}_q} = 0.0048$, $C_{\mathrm{dm}_r} = 0.0051$	$C_{\mathrm{df}_x} = 0.6023$, $C_{\mathrm{df}_y} = 0.6023$, $C_{\mathrm{df}_z} = 0.9$ $C_{\mathrm{dm}_p} = 0.0123$, $C_{\mathrm{dm}_q} = 0.0067$, $C_{\mathrm{dm}_r} = 0.0121$

manual control variability. Fig. 10(b) presents the velocity tracking in the x and y directions. The HIL simulation closely matches the real flight in time-domain response, with improved stability, reduced overshoot, and faster response—likely due to reduced modeling errors and the absence of outdoor airflow disturbances. Since both the real-world and simulation experiments in Fig. 10(b) were conducted under remote control, it is not straightforward to directly compare velocity errors. To enable a more quantitative assessment of velocity responses, we calculated the RMSE between the actual and expected velocities within the control command intervals. The results show that in the x-direction, the RMSE values are 0.1484 for the real-world test and 0.1632 for the simulation, while in the y-direction, they are 0.1311 and 0.1622, respectively. These results indicate that the velocity responses in both environments are highly consistent, with only minor discrepancies. Fig. 10(c) compares PWM outputs from both scenarios. Although similar in range, the real flight data is more dispersed, whereas the simulation shows a tighter distribution. This variation may result from motor inconsistencies, UAV asymmetries, and flight controller mounting angles, which induce PWM adjustments to maintain torque balance. To provide a more rigorous quantitative validation of the consistency between the real UAV and its virtual counterpart, we conducted comparative experiments in both real-world and simulated environments. The experiments involved repeated tasks of forward acceleration followed immediately by reverse acceleration along the x and y axes, as illustrated in Fig. 11. The overall RMSE was calculated to be 0.1090, which falls within the expected error range. This result further substantiates the credibility of the proposed digital model in the time domain.

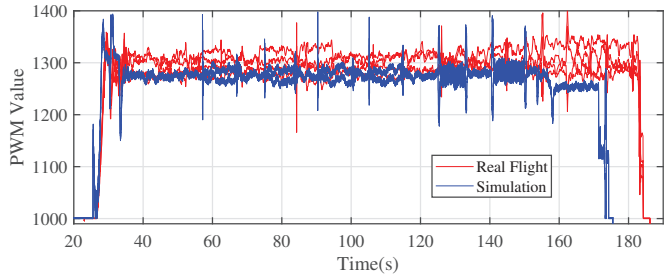
The credibility of sensor data has been successfully demonstrated in.[36] From a qualitative analysis perspective, Fig. 10 confirms that the high-precision multicopter model exhibits a high degree of consistency with the real aircraft in the time-domain. Even when controlled via remote signals, the model maintains a high level of realism.



(a)Trajectories response comparison.



(b)Speed response comparison.



(c)Motor PWM outputs comparison.

Fig. 10. Comparison of simulation and real flight in FX200 time domain experiments.

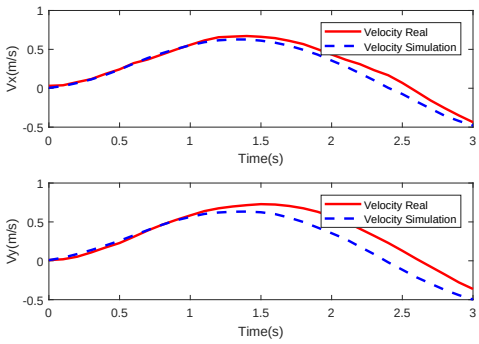
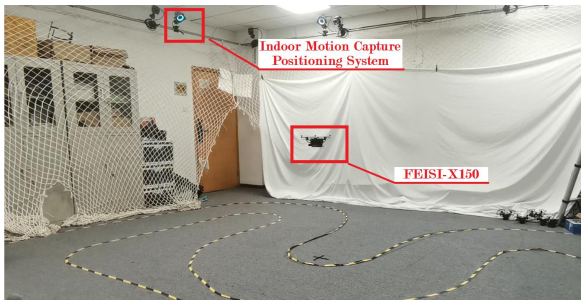
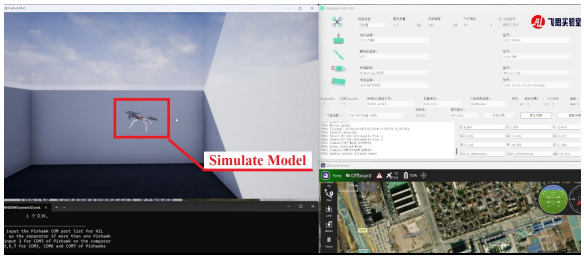


Fig. 11. Speed response comparison of real and virtual FEISI-X200.



(a)FEISI-X150 realflight experiment.



(b)HIL simulation experiment of FEISI-X150.

Fig. 12. Comparison of experimental scenarios for frequency-domain validation.

4.3. Frequency domain

To validating the frequency-domain credibility of the multicopter model, the sweep frequency method was employed. As shown in Fig. 12, sweep frequency experiments were conducted for both the real and simulated scenarios using the FEISI-X150 as an example. The input sweep signal was consistent across both experiments. Fig. 13(a) shows the input sweep signal and the corresponding response output, with the frequency interval of interest set at $[f_a, f_b] = [0.25rad/s, 31rad/s]$. Fig. 13(b) shows the Bode amplitude and phase plots for the x-axis, derived from the original time-domain data through the fast fourier transform (FFT) algorithm. Fig. 13(b) shows that the discrepancies between the experimental and simulated curves are minimal in both amplitude and phase. This is particularly evident in the lower frequency range, where noise levels are moderate. These observations confirm the plausibility of the simulation platform from a qualitative analysis perspective.

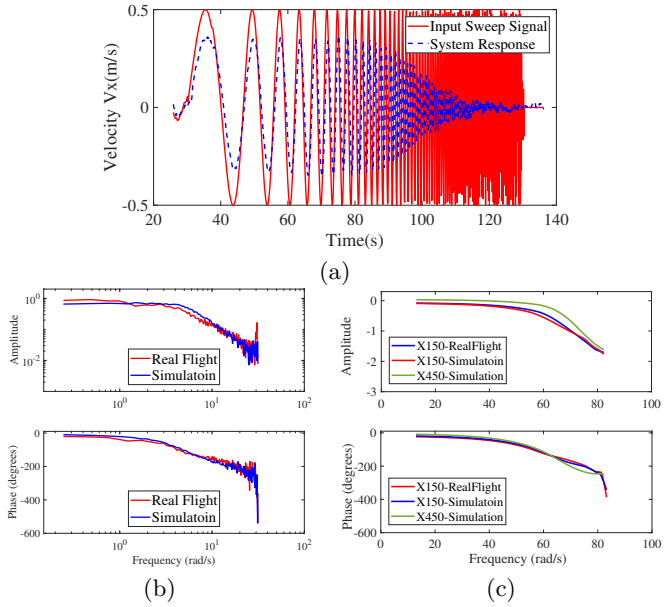


Fig. 13. Frequency-domain assessment for the simulation fidelity. (a)x-channel frequency response test. (b)Bode amplitude plot and Bode phase plot of x-channel. (c)Bode plot of the FEISI-X150 and FEISI-X450 after data processing.

Before assessing model credibility, the raw data in Fig. 13(b), which contains significant noise, must be smoothed and denoised, as shown in Fig. 13(c). The frequency-domain assessment index η_f from (24) was then applied to evaluate the simulation results. For the simulation and experimental curves depicted in Fig. 13(c), the frequency domain fidelity index for the x-channel is $\eta_{xf} = 93.37\%$ (the amplitude credibility index is $\eta_{amp} = 90.63\%$ and the phase credibility index is $\eta_{pha} = 96.04\%$). The experimental data for the y-channel yielded results similar to those of the x-channel, so the corresponding figures are not shown. However, credibility evaluation was still conducted, the frequency domain fidelity index is $\eta_{yf} = 92.14\%$ (the amplitude credibility index is $\eta_{amp} = 90.16\%$ and the phase credibility index is $\eta_{pha} = 94.07\%$).

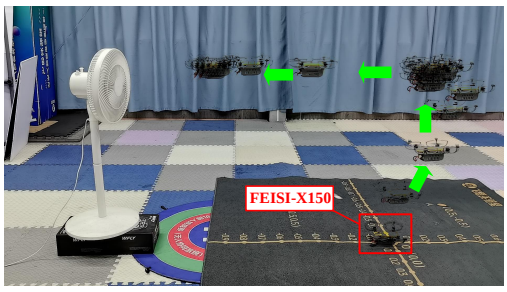
Since the experimental data are raw and subject to random variations, these evaluation results demonstrate that the simulation model developed in this study achieves high credibility in the frequency domain when compared to an actual multicopter UAV. Additionally, as shown in Fig. 13(c), the FEISI-X450 was tested in a simulation experiment, with credibility assessed using real flight data from the FEISI-X150. The frequency domain fidelity index for this comparison is $\eta_f = 61.97\%$ (the amplitude credibility index is $\eta_{amp} = 47.19\%$ and the phase credibility index is $\eta_{pha} = 73.86\%$), highlighting the significant differences between the simulation models of various aircraft.

5. Validation of sim2real performance

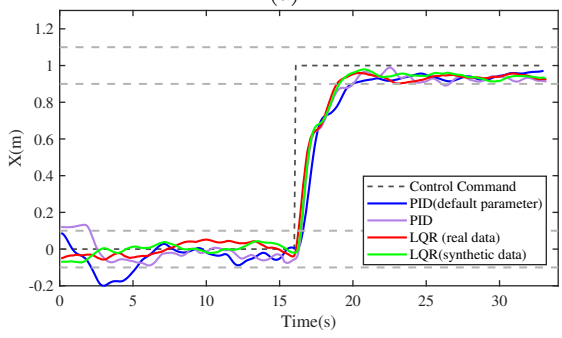
5.1. Control algorithm sim2real verification

To demonstrate the effectiveness of high-fidelity modeling in improving the development efficiency and reducing the cost of multicopter control algorithms, this paper presents a representative application case. A common requirement in multicopter controller optimization is to enhance the system's robustness against disturbances. Accordingly, this study investigates a model-based control strategy. Among various model-based methods, the Linear Quadratic Regulator (LQR) offers a relatively straightforward implementation. The core concept of LQR is to treat the UAV as a linear system, identify a suitable state-space model from input–output data, and compute the optimal state feedback gain based on desired pole placement.

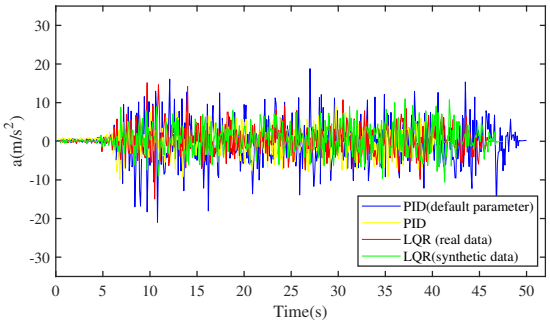
The design process of LQR through system identification involves three main steps: defining the system's input and output variables; conducting frequency sweep experiments to collect input–output data; and deriving a linear state-space model suitable for LQR synthesis. In this case, the digital twin system is employed to perform frequency sweep simulations and generate synthetic datasets, which are combined with a limited set of experimental data obtained through extensive physical testing on the FEISI-X150 platform. This hybrid approach significantly reduces the need for prolonged real-world testing and manual parameter tuning.



(a)



(b)



(c)

Fig. 14. Frequency-domain assessment for the simulation fidelity. (a)(a)The wind perturbation scenario. (b)The closed-loop performance of each controller. (c)The raw IMU acceleration data of each controller.

Table 4. The Closed-Loop Performance of Each Controller.

	IAE	ISE	RT	ST	SSE
PID(default parameter)	3.4484	1.2157	8.60	12.8	0.0664
PID(manually tuned)	3.2124	1.0169	4.02	11.8	0.0788
LQR(real data)	2.7486	0.8807	2.9	9.2	0.0641
LQR(synthetic data)	2.4505	0.8665	3.08	5.7	0.0588

Table 5. The Raw IMU Acceleration Data of Each Controller.

	RMS	Variance	Peak-to-Peak
PID(default parameter)	5.7373	32.9404	39.7718
PID(manually tuned)	3.5310	12.3859	19.9029
LQR(real data)	3.7532	14.0231	30.1148
LQR(synthetic data)	3.5967	12.5586	21.5962

To assess the disturbance rejection performance of the proposed control strategies, a wind perturbation scenario was introduced, as showed in Fig.14(a). The LQR controller designed using synthetic data was compared against three benchmarks: (1) an LQR controller derived from a limited set of experimental flight data; (2) a PID controller with default parameters as implemented in the PX4 autopilot; and (3) a manually tuned PID controller optimized through extensive parameter adjustment. As shown in Fig.14(b), the closed-loop performance of each controller was evaluated under identical wind disturbance conditions. To further highlight robustness characteristics, Fig.14(c) presents the unfiltered and unfused raw IMU acceleration data recorded during task execution. This raw sensor data more clearly captures the immediate response to external disturbances. Quantitative evaluation was conducted using multiple key

performance indicators, including Integral of Absolute Error (IAE), Integral of Squared Error (ISE), Rise Time (RT), Settling Time (ST), Steady-State Error (SSE), Root Mean Square (RMS), Variance, and Peak-to-Peak amplitude.

As summarized in Table.4 and Table.5, the LQR controller trained using synthetic data achieved the best overall performance across most metrics. Both the synthetic-data LQR and the manually tuned PID controller outperformed the default PID controller. Although the manually tuned PID exhibited slightly better raw IMU performance compared to the synthetic-data LQR (Table.5), this came at the expense of significantly degraded transient response, as shown in Table.4. From a holistic perspective, the manually tuned PID remained inferior to the synthetic-data-based LQR controller in terms of both effectiveness and development efficiency.

While manual PID tuning improved robustness compared to the default configuration, it required substantial time and human effort. Similarly, the LQR controller trained on limited experimental data exhibited acceptable disturbance rejection but was constrained by long development cycles and reduced reliability due to insufficient data for accurate model identification. In contrast, the LQR controller designed using synthetic data—generated within the proposed digital twin framework—offered a balanced trade-off between performance, modeling accuracy, and development efficiency.

In contrast, the synthesized-data-driven LQR controller effectively mitigated these limitations, offering enhanced reliability and development efficiency. These findings underscore the potential of the proposed digital twin framework to accelerate the development of UAV control algorithms while reducing the engineering overhead associated with real-world multicopter systems.

5.2. UAV swarm development digital twin case

To demonstrate the scalability and extensibility of the proposed high-presicion multicopter model and digital twin framework, a hybrid physical–virtual application scenario involving a multicopter swarm is presented. As showed in Fig. 15, a cooperative formation flight task is executed simultaneously in both physical and virtual environments. The swarm consists of four quadrotors—two physical UAVs and two virtual UAVs modeled using the high-precision digital twin.

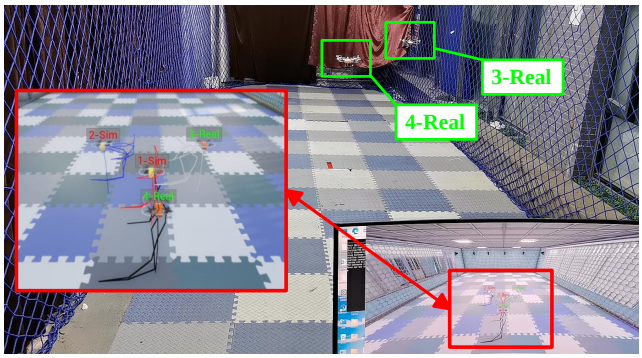


Fig. 15. The application of high-precision models in digital twin scenarios.

During the task, the two physical quadrotors are dynamically mapped into the virtual environment in real time, where they interact and coordinate with two simulated UAVs to complete the formation mission. Throughout the operation, the physical UAVs' states are continuously synchronized with their virtual counterparts, and all four agents—both real and simulated—engage in a shared control loop. Their mutual interactions, including inter-vehicle disturbances, are faithfully captured and reflected across both environments. In the hybrid physical–virtual scenario, the local host was employed as an intermediate base station, where the simulation of the virtual UAV was executed. Since the host directly forwarded messages and data to the virtual UAV, the associated latency can be considered negligible. Experimental measurements indicated that the average response delay for the real UAV receiving commands from the host was approximately $3ms$, which satisfies the performance requirements for virtual–real collaborative operations. To further assess the positional synchronization accuracy between the virtual and real UAVs, we recorded their trajectories during the experiment, as showed in Fig. 16. Excluding the influence of significant ground effect observed during the landing phase of the real UAV, the relative position RMSE was calculated to be 0.1056. This error level is within an acceptable range for hybrid physical–virtual scenarios, demonstrating the framework's capability to ensure reliable synchronization.

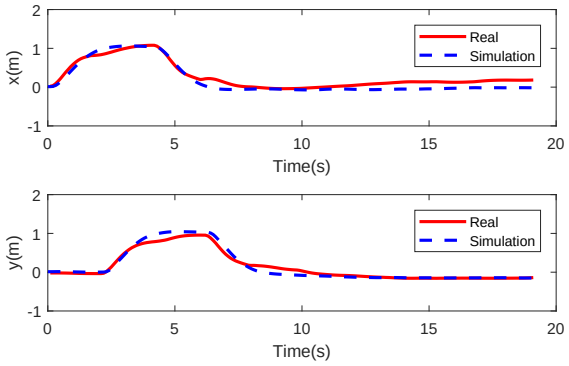


Fig. 16. Comparison of real UAV and digital UAV positions in the hybrid physical–virtual scenario.

This case study validates the accuracy, feasibility, and scalability of the proposed digital twin system for real-time, cooperative UAV operations, and highlights its potential for enabling mixed-reality swarm coordination and testing in complex scenarios.

5.3. Video and Source Code

This paper posted a video, which includes an introduction to the high-precision multicopter modeling method, model credibility evaluation, and application case demonstrations. The video link is as follows:

`https://youtu.be/gYjO5VS_ZUg`

The source code for the high-precision simulation model in MATLAB/Simulink is available on GitHub. The link is as follows:

`https://github.com/RflySim/RflySimModel`

6. Conclusions

This paper presents a digital twin-level modeling framework designed to improve the development, optimization, and deployment of control algorithms for multicopter UAVs. By integrating high-precision physical modeling with real-time co-simulation and hybrid virtual–physical testing environments, the proposed framework bridges the gap between simulation and real-world deployment (sim2real). The methodology supports both component-level accuracy and system-level scalability, enabling reliable identification, tuning, and evaluation of UAV controllers with significantly reduced physical testing overhead.

Through a series of case studies—including model-based LQR control design under wind disturbances and a mixed-reality multicopter swarm formation task—the framework has demonstrated its effectiveness in improving control performance while reducing development cost and effort. Experimental results validate that synthetic datasets generated from the digital twin can accelerate controller design and outperform those based solely on limited physical testing or manual tuning.

The proposed approach offers a scalable and extensible foundation for UAV control research in smart manufacturing, cooperative robotics, and industrial automation. The digital framework is developed on the RflySim platform, which supports multi-UAV SIL, HIL, hybrid virtual–real cluster simulations, and distributed execution across multiple computers within a LAN (see Section 2.1 for the official website). While this work emphasizes high-fidelity digital twin modeling and sim2real optimization for a single UAV, it can also be extended to swarm scenarios and reinforcement learning by leveraging RflySim's distributed multi-simulator capabilities and support for real–sim data integration. We also conducted preliminary attempts to apply reinforcement learning within this simulation environment, and the corresponding implementations have been included in our open-source code (see Section 5.3 for the link).

Future work will explore adaptive digital twin refinement, integration with reinforcement learning pipelines, and application to heterogeneous robotic systems operating in uncertain or dynamic environments.

References

[1] A. Toma, N. Cecchinato, G. Ferrin, I. Scagnetto, C. Drioli and G. L. Foresti, MAV-Link-Based control and coordination of a multi-drone cluster for intelligence, surveillance and reconnaissance tasks, Unmanned Systems 13(04) (2025) 1027–1040.

[2] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif and M. A. Khan, Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends, Intelligent service robotics 16(1) (2023) 109–137.

[3] Z. Hu, Y. Cao, X. Li, X. Zhang, C. Zhang and Z. Liu, A real-time uav delivery system considering dock selection and spatial conflict, Expert Systems with Applications 281 (2025) p. 127498.

[4] F. Kong, F. Du and D. Zhao, Station-viewpoint joint coverage path planning towards mobile visual inspection, Robotics and Computer-Integrated Manufacturing 91 (2025) p. 102821.

[5] M. Baha and F. Saghafi, Precise dynamic modeling and control of a net-throwing aerial vehicle in capturing flying objects, Unmanned Systems (2025) 1–18.

[6] Z. Zuo, C. Liu, Q.-L. Han and J. Song, Unmanned Aerial Vehicles: Control methods and future challenges, IEEE/CAA Journal of Automatica Sinica 9(4) (2022) 601–614.

[7] G. Xu, Z. Sun, H. Liu, Y. Zhou, X. Gong and S. Gong, Identification of ground effect and intelligent control of unmanned aerial vehicles, Aerospace Science and Technology 131 (2022) p. 107976.

[8] A. Sir Elkhatem and S. Naci Engin, Robust lqr and lqr-pi control strategies based on adaptive weighting matrix selection for a UAV position and attitude tracking control, Alexandria Engineering Journal 61(8) (2022) 6275–6292.

[9] H. Yu, L. Han, X. Li, X. Dong and Z. Ren, Online updating data-driven model predictive control for quadrotors in close formation based on gaussian process regression (2023) 84–96.

[10] M. Liu, F. Zhang and S. Lang, The quadrotor position control based on MPC with adaptation (2021) 2639–2644.

[11] Y. Dai and D. Wang, A tube model predictive control method for autonomous lateral vehicle control based on sliding mode control, Sensors 23(8) (2023).

[12] A. R. A. Majid, Quadrotor UAV kinematics and dynamics: A comprehensive model with suggested control techniques (2024) 335–340.
[13] A. Peringal, M. Chehadeh, I. Boiko and Y. Zweiri, Relay-based identification of aerodynamic and delay sensor dynamics with applications for unmanned aerial vehicles, IEEE Sensors Journal 24(8) (2024) 13085–13094.
[14] L. Wälti and A. Martinoli, Lumped drag model identification and real-time external force detection for rotary-wing micro aerial vehicles (2024) 3853–3860.
[15] G. Pan, X. Cui, P. Sun and B. Wang, A brief review of the actuation systems of the morphing systems in wind tunnel models and a case study, Aerospace 11(8) (2024).
[16] M. Zhang, J. Jiao, J. Zhang and Z. Zhang, High-efficiency data fusion aerodynamic performance modeling method for high-altitude propellers, Drones 8(6) (2024) p. 229.
[17] S. Zimao, Y. Hongan, W. Jiakang, S. Peng and Z. Tong, Robust convergency indicator using high-dimension pid controller in the presence of disturbance, arXiv preprint arXiv:2411.13140 (2024).
[18] S. Kusmirek, V. Socha, T. Malich, L. Socha, K. Hylmar and L. Hanakova, Dynamic flight tracking: Designing system for multirotor uavs with pixhawk autopilot data verification, IEEE Access 12 (2024) 109806–109821.
[19] K. Niki Maleki, S. Karimi, S. Mohammadi and K. Ashenayi, System identification and modeling of a multirotor UAV: A comparative study (2024) p. 0567.
[20] S. Wisbacher, D. Ossmann, S. Schubert, J. Frey and H. Pfifer, Wind tunnel based system identification of a small unmanned aircraft system (2025) p. 0857.
[21] O. H. Zekry, M. M. Ashry, A. T. Hafez and T. Attia, CANFIS PSO-Based modeling of the crazyflie quadrotor, Unmanned Systems (2025) 1–20.
[22] F. Tao, H. Zhang and C. Zhang, Advancements and challenges of digital twins in industry, Nature Computational Science 4(3) (2024) 169–177.
[23] A. Putranto, T.-H. Lin and P.-T. Tsai, Digital twin-enabled robotics for smart tag deployment and sensing in confined space, Robotics and Computer-Integrated Manufacturing 95 (2025) p. 102993.
[24] S. Wang, J. Zhang, P. Wang, J. Law, R. Calinescu and L. Mihaylova, A deep learning-enhanced digital twin framework for improving safety and reliability in human–robot collaborative manufacturing, Robotics and Computer-Integrated Manufacturing 85 (2024) p. 102608.
[25] X. Zhang, Y. Jiang, X. Wu, Z. Nan, Y. Jiang, J. Shi, Y. Zhang, X. Huang and G. G. Huang, AIoT-enabled digital twin system for smart tunnel fire safety management, Developments in the Built Environment 18 (2024) p. 100381.
[26] L. Zhang, J. Guo, X. Fu, R. L. K. Tiong and P. Zhang, Digital twin enabled real-time advanced control of tbm operation using deep learning methods, Automation in Construction 158 (2024) p. 105240.
[27] X. Lai, L. Yang, X. He, Y. Pang, X. Song and W. Sun, Digital twin-based structural health monitoring by combining measurement and computational data: An aircraft wing example, Journal of Manufacturing Systems 69 (2023) 76–90.
[28] J. Liu, J. Vatn and S. Yin, Advanced battery health management in electric uavs through digital twins (2024) 1–5.
[29] A. Sadeghi, P. Bellavista, W. Song and M. Yazdani-Asrami, Digital twins for condition and fleet monitoring of aircraft: Toward more-intelligent electrified aviation systems, IEEE Access 12 (2024) 99806–99832.
[30] W. Zhang, J. Tong, F. Liao and Y. Zhang, Simulation-to-reality quadrotor propeller damage detection in windy environments, Unmanned Systems (2025) 1–12.
[31] S. Shah, D. Dey, C. Lovett and A. Kapoor, Airsim: High-fidelity visual and physical simulation for autonomous vehicles (2018) 621–635.
[32] N. Koenig and A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 3 (2004), pp. 2149–2154.
[33] C. Cui, X. Zhou, M. Wang, F. Gao and C. Xu, FastSim: a modular and plug-and-play simulator for aerial robots, IEEE Robotics and Automation Letters 9(6) (2024) 5823–5830.
[34] J. Du, K. Wang, Y. Fan, G. Lai and Y. Yu, High-fidelity integrated aerial platform simulation for control, perception, and learning, IEEE Transactions on Automation Science and Engineering 22 (2025) 13662–13683.
[35] X. Dai, C. Ke, Q. Quan and K. Cai, RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations, Aerospace Science and Technology 114(106727) (2021) 1–15.
[36] X. Dai, C. Ke, Q. Quan and K. Cai, Simulation credibility assessment methodology with FPGA-based hardware-in-the-loop platform, IEEE Transactions on Industrial Electronics 68(4) (2021) 3282–3291.
[37] I. MathWorks, Model-based design-MATLAB & Simulink (2019).
[38] M. J. Merrigan, E. R. Swift, R. F. Wong and J. T. Saffel, A refinement to the World Geodetic System 1984 reference frame (Sept. 2002) 1519–1529.
[39] N. K. Pavlis, S. A. Holmes, S. C. Kenyon and J. K. Factor, The development and evaluation of the earth gravitational model 2008 (EGM2008), Journal of geophysical research: solid earth 117(B4) (2012).
[40] M. Cavcar, The international standard atmosphere (ISA), Anadolu University, Turkey 30(9) (2000) 1–6.
[41] D. J. Moorhouse and R. J. Woodcock, Background Information and User Guide for via MIL-F-8785C, Military Specification - Flying Qualities of Piloted Airplanes (Flight Dynamics Laboratory, 1982).
[42] Q. Quan, Introduction to Multicopter Design and

Control, 1 edn. (Springer Singapore, July 2017).
[43] A. Chulliat, W. J. Brown, P. Alken, C. D. Beggan, M. Nair, G. A. Cox, A. Woods, S. Macmillan, B. Meyer and M. Paniccia, The US/UK World Magnetic Model for 2020-2025 : Technical Report, tech. rep., National Centers for Environmental Information (U.S.) (2020).



Xunhua Dai received the B.S., M.S., and Ph.D. degrees in the control science and engineering from Beihang University, Beijing, China, in 2013, 2016, and 2020, respectively. Since 2020, he has been an associate professor at Central South University in the School of Computer Science and Engineering. His main research interests include reliable intelligent control, safety assessment, and design optimization of unmanned aerial robotics.



Wanqi Gong received the B.S. degree from the School of Automation, Central South University in 2023. He is currently pursuing the M.S. degree at the School of Computer Science, Central South University. His main research interests include UAV digital twins and high-precision modeling of UAVs.



Jinhu Tu received the B.S. degree from the School of Software Engineering, Jiangxi University of Science and Technology in 2021 and the M.S. degree from the School of Computer Science, Central South University in 2024. He is currently pursuing a doctorate degree at the School of Automation at Central South University. His main research interests include UAV digital twins, health assessment and diagnosis, and self-safety enhancement.



Yong Chen received the Ph.D. degree in mechanical engineering from The University of Hong Kong, Hong Kong, in 2016. From 2016 to 2018, he was a Research Assistant with the Department of Mechanical Engineering, the University of Hong Kong. Since 2018, he has been a Lecture with the College of Automation, Central South University, Changsha, China. His current research interests include reachable set estimation for dynamic systems, distributed optimization of multiagent systems, and applications in aerialrobotics.