

EXPERIMENTAL FLIGHT DYNAMICS AND CONTROL OF AN AGILE FIXED-WING UAV USING HYBRID ONBOARD AND OFFBOARD CONTROL

Yuchen Huang[1], KC Wong[1]

[1]School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, NSW, Australia

Abstract

This paper presents a hybrid onboard-offboard control architecture for indoor flight testing of an agile fixed-wing unmanned aerial vehicle. Agile indoor fixed-wing flight is challenging because the aircraft must operate within a confined GNSS-denied environment, where flight volume, speed margin, and recovery distance are highly limited. The proposed architecture separates the control task into two layers: the onboard subsystem provides high-rate inertial sensing, stabilisation, and actuator control, while the offboard ROS 2 subsystem integrates motion-capture feedback, reference generation, telemetry logging, and higher-level flight supervision.

The architecture is demonstrated using two representative indoor flight cases: a prop-hang-to-forward-flight-to-prop-hang transition and a circular trajectory tracking manoeuvre. These cases evaluate flight-regime transition capability, reference-command generation, trajectory tracking, and repeatable data collection within a confined indoor test volume. Wind-tunnel aerodynamic data are used as prior information for a simplified flight-dynamics model, while output-error analysis is introduced to support selected aerodynamic and propulsion-related parameter refinement from closed-loop flight data.

Keywords: Agile fixed-wing UAV; Hybrid onboard-offboard control; Motion-capture flight testing; Flight-dynamics identification

1. Introduction

Indoor flight testing provides a controlled and repeatable environment for developing and validating unmanned aerial vehicle (UAV) flight control systems, particularly for agile manoeuvres that are difficult to evaluate safely outdoors. However, indoor fixed-wing flight introduces challenges that are fundamentally different from those of hovering platforms. Unlike multirotor aircraft, fixed-wing vehicles must continuously manage forward speed, attitude, and flight-path curvature within a confined flight volume. These requirements become more demanding for small agile fixed-wing UAVs operating at low Reynolds numbers and high angles of attack [1,2], where aerodynamic nonlinearities, strong coupling between translational and rotational motion, and limited control margins can dominate the flight response. Similar fixed-wing hover, transition, and perching manoeuvres have been investigated for small agile aircraft, demonstrating both the potential and the control difficulty of operating fixed-wing platforms near or beyond the conventional low-angle-of-attack flight regime [3, 4, 5, 6].

In indoor environments, the absence of global navigation satellite system (GNSS) measurements further complicates the control problem. Onboard inertial measurement units (IMUs) are well suited to high-rate attitude and angular-rate stabilisation, and they remain essential for safety-critical control. However, inertial sensing alone provides limited long-term observability of position, velocity, and global trajectory error. Dead-reckoning drift, sensor bias, and the lack of direct global position feedback make it difficult to conduct repeatable trajectory-tracking experiments using onboard sensing alone. For agile fixed-wing vehicles, where small errors in speed, attitude, or turn radius can

DOI: 10.71945/icas2026_0259

quickly lead to large position deviations, accurate external-state feedback becomes particularly valuable [6, 7].

Hybrid onboard–offboard control architectures offer a practical solution for this class of indoor flight experiment. In such systems, the onboard controller remains responsible for high-rate stabilisation and actuator control, while the offboard system provides global state feedback, reference generation, experiment supervision, and data logging. This separation of responsibilities allows the aircraft to retain local stabilisation authority even when offboard sensing or communication is delayed, while still benefiting from high-accuracy motion-capture measurements for trajectory-level control. For fixed-wing indoor flight testing, this architecture is especially attractive because it supports both real-time control and repeatable collection of flight data within a limited test volume.

Although motion-capture-assisted control has been widely used in indoor UAV experiments, the experimental architecture is often treated as a supporting implementation detail rather than as a central contribution. For agile fixed-wing platforms, however, the architecture strongly influences what manoeuvres can be performed, how safely they can be tested, and whether the resulting data are suitable for flight-dynamics analysis. A useful experimental framework must therefore integrate onboard stabilisation, offboard global-state feedback, reliable communication, fail-safe behaviour, reference-command generation, and synchronised logging of state, command, and actuator data.

Accurate flight-dynamics modelling is also important for analysing and improving agile fixed-wing control performance. Wind-tunnel measurements can provide physically meaningful prior aerodynamic information, particularly for small aircraft operating in low-Reynolds-number and high-angle-of-attack regimes. However, wind-tunnel data alone may not fully represent the dynamic response of the aircraft in free flight, especially when propulsion effects, unsteady motion, actuator behaviour, and closed-loop control are present. Flight-test data can therefore be used to refine selected model parameters. In this work, output-error analysis is introduced [8] as a preliminary method for refining selected aerodynamic and propulsion-related parameters using recorded actuator commands and measured flight states, rather than as a complete aerodynamic database identification from flight data alone.

This paper presents a hybrid onboard-offboard experimental control architecture for an agile indoor fixed-wing UAV. The architecture is evaluated using two representative indoor flight cases: a prop-hang-to-forward-flight-to-prop-hang transition, which tests the system across a large change in pitch attitude, speed, and flight regime; and circular trajectory tracking, which tests reference-command generation and repeatable path following in a confined flight volume.

The main contributions of this paper are: A hybrid onboard–offboard control architecture for agile indoor fixed-wing UAV experiments.

1. A ROS 2 and MAVLink-based implementation that integrates motion-capture feedback, onboard inertial stabilisation, predefined reference tracking, telemetry logging, and experiment supervision.
2. Two representative indoor flight demonstrations covering flight-regime transition and circular trajectory tracking.
3. A preliminary flight-dynamics analysis framework that uses wind-tunnel aerodynamic data as prior information and supports selected aerodynamic and propulsion-related parameter refinement from closed-loop flight data.

2. Experimental Platform Hardware

2.1 Agile Fixed-Wing UAV Platform

The experimental aircraft is a custom-built agile indoor fixed-wing UAV based on a compact F3P biplane configuration. The platform was developed for low-Reynolds-number flight experiments, high-angle-of-attack manoeuvres, prop-hang flight, and confined indoor flight testing. The biplane layout was selected because it provides a relatively large lifting area within a small wingspan, allowing the aircraft to fly at low speed while maintaining sufficient control authority in a limited flight volume.

The aircraft has a wingspan of approximately 420 mm and is constructed primarily from lightweight Depron foam. Carbon-fibre rods and thin carbon-fibre sheets are used at critical load-bearing locations to improve stiffness while keeping the structural mass low. The bare airframe mass is approximately 24 g, and the total take-off mass for the experiments is kept below 80 g. This low mass reduces inertia

and allows the aircraft to respond quickly to attitude and throttle commands, which is important for transition manoeuvres and indoor trajectory tracking. Figure 1 shows the test platform.

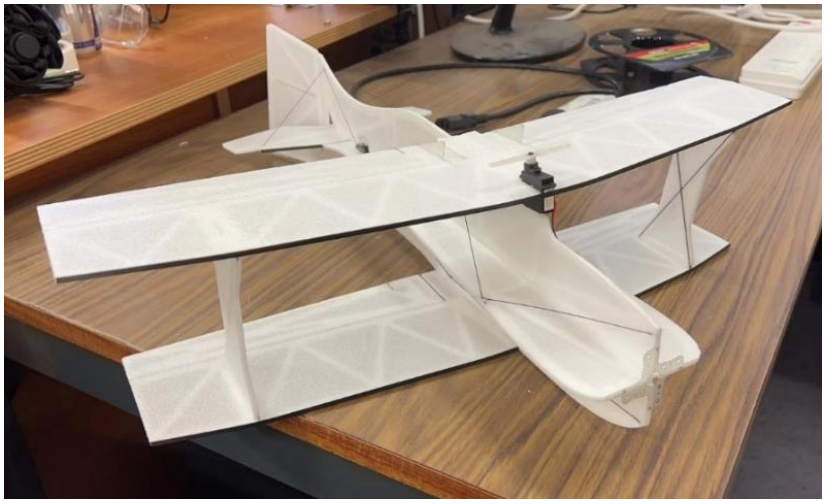


Figure 1 – Test platform.

The propulsion system consists of a T-Motor AM20 Pro electric motor driving an 8041 carbon-fibre propeller, powered by a 2S 300 mAh lithium-polymer battery. This motor-propeller combination provides a high thrust-to-weight ratio, enabling prop-hang operation, rapid acceleration, and recovery from low-speed or high-angle-of-attack flight conditions. In addition to producing thrust, the propeller generates a strong slipstream over the downstream lifting surfaces and control surfaces.

Large control surfaces are used on the aircraft to improve control authority at low forward speed. A significant portion of the elevator, rudder, and aileron surfaces is exposed to the propeller slipstream. As a result, the aircraft can still generate aerodynamic control moments even when the freestream velocity of the airframe is very low or close to zero, such as during prop-hang flight. This feature is essential for maintaining attitude control during hovering-like manoeuvres, transition from prop-hang to forward flight, and recovery back to prop-hang. The platform therefore relies not only on conventional forward-flight aerodynamic control, but also on propeller-slipstream-assisted control.

The platform was designed as an experimental research aircraft rather than an endurance or payload-carrying vehicle. Its main requirements are low flight speed, high thrust-to-weight ratio, high control authority, low structural mass, and compatibility with indoor motion-capture testing. These characteristics make it suitable for studying agile fixed-wing flight dynamics, hybrid onboard-offboard control, and preliminary flight-test-based model refinement.

The geometric, mass, and baseline aerodynamic parameters in Tables 1 and 2 are taken from the previous wind-tunnel and modelling study reported in [9].

Parameter	Symbol	Value	Unit
Upper wingspan	b_u	420	mm
Lower wingspan	b_l	380	mm
Upper aspect ratio	AR_u	3.50	-
Lower aspect ratio	AR_l	3.60	-
Reference wing area	S	0.0905	m^2
Total length	L	510	mm
Bare airframe mass	M_{empty}	24	g
Normal take-off mass	M	81	g
Centre of gravity location	X_{CG}	145	mm from nose
Thrust-to-weight ratio	T_{Max}/W	2.10	-

Table 1 – Aircraft geometric and mass parameters [9].

The baseline aerodynamic parameters used in this study are summarised in Table 2. These parameters were obtained from previous wind-tunnel testing of the aircraft, including force and moment measurements, control-surface sweeps, and propulsion/control-surface interaction tests,

following standard low-speed wind-tunnel testing practice [15]. In the present work, these aerodynamic data are used as prior information for flight-dynamics modelling and output-error-based parameter refinement. The detailed wind-tunnel setup, calibration process, data-reduction method, and coefficient fitting procedure are reported in [9] and are therefore not repeated here.

Lift coefficient		Drag coefficient		Side-force coefficient	
CL_0	-4.160E-03	Cd_0	1.225E-01	Cy_0	3.000E-04
CL_a	4.946E-02	Cd_a	1.615E-04	Cy_b	-6.850E-02
Cl_{a2}	4.560E-05	Cd_{a2}	6.101E-04	Cy_p	-3.400E-02
CL_q	-9.730E-02			Cy_r	6.500E-02
Rolling moment coefficient		**Pitching moment coefficient**		**Yawing moment coefficient**	
Cl_0	5.000E-04	Cm_0	5.966E-02	Cn_0	-2.000E-05
Cl_b	1.420E-03	Cm_a	-2.760E-02	Cn_b	1.880E-03
Cl_p	6.981E-02	Cm_{a2}	-1.298E-04	Cn_p	-4.014E-03
Cl_r	5.585E-03	Cm_q	-1.379E-01	Cn_r	-6.195E-03

Table 2 – Baseline aerodynamic parameters per radian obtained from wind-tunnel testing [9].

2.2 Onboard Electronics and Actuation

The onboard hardware is designed to provide high-rate inertial sensing, deterministic attitude stabilisation, reliable actuator output, wireless communication, and onboard data recording. The system is built around a Teensy 4.1 microcontroller, which serves as the primary onboard flight controller. It was selected for its high processing capability, small form factor, low mass, and multiple hardware interfaces, including SPI, I2C, UART, PWM, DShot-compatible output, and high-speed SD card support.

An ISM330DHCX inertial measurement unit is used as the primary onboard motion sensor. The IMU is connected to the Teensy 4.1 through SPI to support high-rate angular-rate and acceleration measurement. This high-rate inertial feedback is essential for inner-loop stabilisation, since external motion-capture feedback has higher latency and may be affected by marker occlusion or communication delay. The Teensy 4.1 also provides onboard data recording through its high-speed SDIO microSD interface, allowing high-frequency IMU, command, actuator, and telemetry data to be stored locally for post-flight analysis.

Wireless communication with the ground station is handled by an ESP32-based module. The ESP32 acts as a lightweight MAVLink bridge between the offboard ROS 2 system and the Teensy 4.1. This hardware separation allows the Teensy to focus on time-critical sensing, control, logging, and actuator output, while the ESP32 handles command and telemetry communication. The control surfaces are driven by lightweight servos using PWM signals, while the propulsion system is controlled through an electronic speed controller using DShot600. A compact battery eliminator circuit provides regulated power for the onboard electronics and servos.

The main onboard electronic and actuation components are summarised in Table 3.

Component	Model / Type	Interface	Function
Primary flight controller	Teensy 4.1	SPI, UART, PWM, SDIO	Onboard MCU for control, logging, and actuator output
IMU	ISM330DHCX	SPI	High-rate angular-rate and acceleration measurement
Communication module	Seeed Studio XIAO ESP32 S3	UART, Wi-Fi	Bridge between onboard controller and ground station
Onboard storage	microSD card	SDIO	High-rate local flight-data recording
Control-surface actuators (X3)	Lightweight servos	PWM	Aileron, elevator, and rudder actuation
Motor controller	BLHeli-S 7A	DShot600	Brushless motor control
Power regulation	Matek BEC	-	7.2V to 5V power regulation
Propulsion motor	T-Motor AM20 Pro	-	Main propulsion
Battery	Gaoneng 2S 300 mAh	-	Power supply

Table 3 – Onboard electronics and actuation hardware.

2.3 Motion-Capture Environment and Ground Station

Indoor global-position and attitude measurement is provided by the OptiTrack motion-capture system installed in the W.A. Bennett Structures Laboratory at the University of Sydney. The laboratory provides a dedicated indoor motion-capture environment for small UAV and robotics experiments. The system consists of 26 OptiTrack cameras arranged around the indoor test volume, including 16 Prime 17 cameras and 10 Prime 41 cameras. The Prime 17 cameras provide wider-volume coverage, while the Prime 41 cameras provide higher-resolution tracking for targets at longer range. A schematic of the OptiTrack tracking and data-transfer principle is shown in Figure 2.

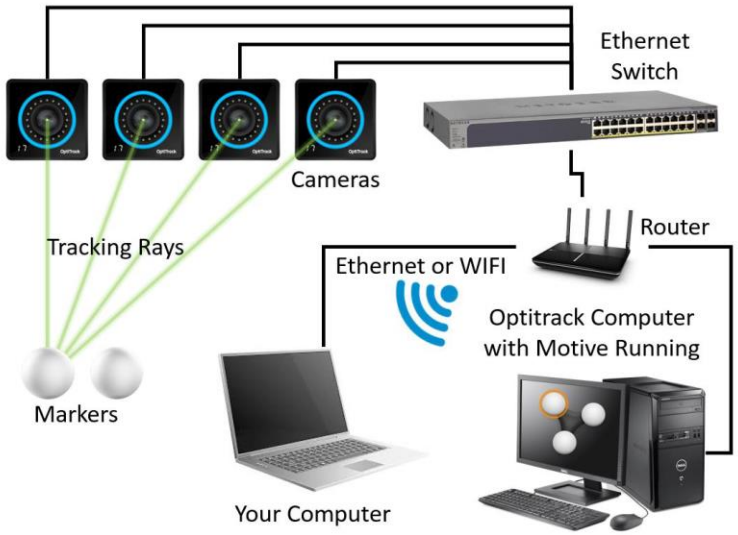


Figure 2 – Schematic of the OptiTrack tracking and data-transfer principle.

The flight experiments use the available indoor capture volume of the Bennett Lab as a single test environment. This provides a controlled GNSS-denied space for evaluating agile fixed-wing manoeuvres while allowing the aircraft to use as much of the laboratory volume as possible for transition and circular flight tests. Compared with outdoor flight testing, the indoor environment also reduces atmospheric disturbances and provides near-still-air conditions, with minimal turbulence during the experiments.

For the experiments in this work, the aircraft is fitted with a set of retroreflective markers arranged as a rigid body. The marker layout is designed to be asymmetric and rigidly attached to the airframe so that the OptiTrack system can measure the aircraft position and attitude in the laboratory coordinate frame during rapid attitude changes and transition manoeuvres.

The motion-capture system is supported by dedicated laboratory hardware, including the OptiTrack camera network, camera power system, OptiTrack computer, network router, and calibration equipment. The cameras detect the retroreflective markers, and the OptiTrack computer reconstructs the rigid-body pose using Motive. A separate ground-station computer is connected to the OptiTrack network for experiment monitoring, data reception, command generation, and logging. This hardware arrangement provides external global-state measurements for the offboard system, while the onboard IMU and flight controller remain responsible for high-rate inertial feedback, local stabilisation, and actuator control.

3. Hybrid Onboard–Offboard Control Architecture

3.1 Architecture Overview

proposed control system uses a hybrid onboard-offboard architecture to combine high-rate local stabilisation with accurate external state feedback. The architecture is designed for indoor agile fixed-wing flight, where the flight volume is limited and the aircraft must remain stable during rapid attitude changes, prop-hang, and transition manoeuvres. The main design principle is to keep safety-critical stabilisation and actuator output onboard, while using the offboard system for global-state feedback, reference generation, experiment supervision, and data logging.

The architecture is divided into three functional layers: the aircraft onboard layer, the communication layer, and the ground-station offboard layer. The onboard layer is implemented on the aircraft and

includes the IMU, onboard estimator, attitude controller, actuator output, motor control, and onboard data logging. The communication layer is implemented using an ESP32-based bridge, which transfers reference commands from the offboard system to the onboard flight controller and sends onboard telemetry back to the ground station. It also supports synchronisation between onboard flight data and OptiTrack measurements for post-flight analysis. The offboard layer is implemented on a ground-station computer running ROS 2, where external motion-capture measurements from the OptiTrack system are received and used to generate reference commands for target-trajectory following.

A key feature of the architecture is that computationally heavier tasks, such as trajectory generation, motion-capture-based state processing, supervisory control, and future model-based control algorithms, can be executed on the ground-station computer rather than on the lightweight onboard flight controller. This reduces the onboard computational burden and allows control algorithms to be modified and tested more flexibly during experiments. However, the offboard system does not directly drive the aircraft actuators. Instead, it generates reference-level commands, such as desired attitude, angular-rate, velocity, or throttle-related commands, which are transmitted to the onboard controller.

The onboard controller converts the received references into actuator commands using local inertial feedback. This separation keeps the safety-critical inner-loop stabilisation onboard, while allowing the offboard computer to provide flexible high-level guidance. It also reduces the sensitivity of actuator-level control to motion-capture latency, communication delay, or temporary tracking loss.

The overall information flow is shown in Figure 3. Motion-capture measurements are processed by the ground-station system and used to generate reference commands. These commands are sent to the onboard flight controller through the communication bridge. The onboard controller combines the received references with high-rate IMU feedback to compute actuator commands for the control surfaces and motor. Telemetry and onboard log data are then used for experiment monitoring, controller evaluation, and later flight-dynamics analysis.

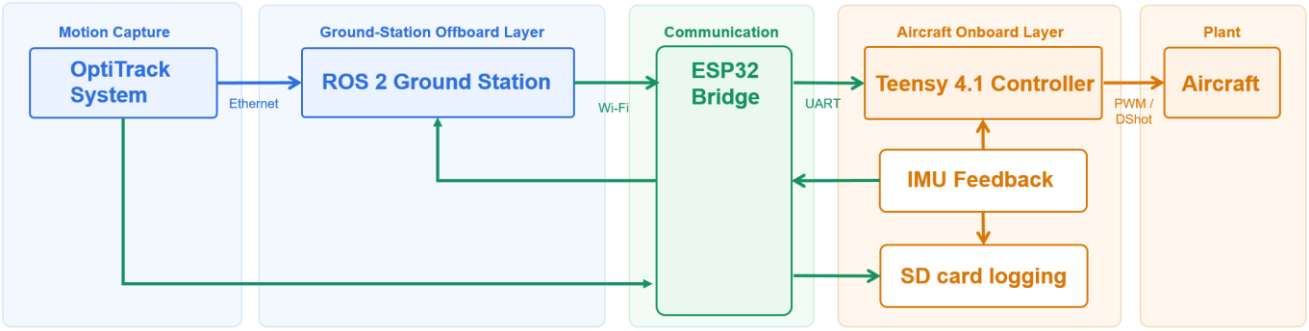


Figure 3 – Hybrid onboard-offboard control architecture and information flow.

Basic fail-safe handling is implemented at the reference-command level. The onboard controller monitors the age of the most recently received offboard command. If the command timeout threshold is exceeded, the controller stops accepting stale offboard references and maintains local inertial stabilisation using the most recent valid command or a conservative fall-back command selected before the experiment. The offboard ROS 2 system also monitors the validity and update rate of the OptiTrack rigid-body solution. If marker tracking is lost or the motion-capture update becomes invalid, the offboard system suspends trajectory-reference updates and flags the event to the operator. A manual override channel is retained so that the operator can command the aircraft directly or terminate the experiment if required. These mechanisms prevent communication delay or motion-capture dropout from directly propagating to actuator-level commands.

3.2 Onboard Stabilization and Actuator Control

The onboard layer is responsible for high-rate state estimation, attitude stabilisation, actuator command generation, and local data recording. It remains the safety-critical control layer of the aircraft because actuator-level control cannot depend entirely on external motion-capture measurements or wireless communication. The onboard controller therefore uses local IMU feedback as the primary input for stabilisation, while commands from the offboard system are treated as reference inputs rather than direct actuator commands.

The onboard estimator processes angular-rate and acceleration measurements from the IMU to obtain the attitude and angular-rate feedback required by the inner-loop controller. This local inertial feedback is used at a higher rate than the offboard motion-capture update rate, allowing the aircraft to respond rapidly to attitude disturbances and control-surface effects. This is particularly important during prop-hang and transition manoeuvres, where the aircraft experiences rapid pitch changes, strong propeller-slipstream effects, and very low forward speed.

Because the aircraft operates across prop-hang, transition, and forward-flight regimes, the onboard reference command is defined in a mode-dependent form rather than as a conventional fixed-wing autopilot command. In general, the onboard controller receives attitude/rate and thrust-related references,

$$[R_{att}^{T}, T_c]^T$$

where R_{att}^{T} denotes the attitude or body-rate reference, and T_c is the throttle or thrust-related command. During prop-hang operation, $r_{att}{}^{T}$ is interpreted primarily as an attitude-stabilization reference, with the pitch command close to the vertical attitude and the thrust command providing weight support. During forward flight, the same interface can be interpreted as roll, pitch, yaw-rate, and throttle commands for trajectory tracking. During transition, the reference commands are varied continuously between these two regimes.

The actuator command vector generated by the onboard controller is expressed as:

$$[\delta_{\mathrm{a}}, \delta_{\mathrm{e}}, \delta_{\mathrm{r}}, \delta_{\mathrm{t}}]^T$$

where $\delta_{\mathrm{a}}, \delta_{\mathrm{e}}$, and δ_{r}, are the aileron, elevator, and rudder commands, and δ_{t} is the throttle command. The actuator-output module converts these commands into PWM signals for the servos and a digital throttle command for the electronic speed controller.

A single fixed PID tuning is not sufficient across the full manoeuvre envelope, because the control characteristics in prop-hang flight and forward flight are fundamentally different. In prop-hang flight, the freestream velocity is very low and the control moments are dominated by propeller slipstream over the control surfaces. In forward flight, the control surfaces operate in a more conventional aerodynamic regime. Therefore, two PID gain sets are used: one tuned for prop-hang or near-vertical flight, and one tuned for forward flight.

To avoid a discontinuous controller switch during transition, the active PID gains are continuously blended using a mixing parameter,

$$\lambda \in [0,1]$$

where λ = 0 corresponds to the prop-hang tuning and λ = 1 corresponds to the forward-flight tuning. The active gain vector is computed as

$$K = (1-\lambda)\mathrm{K_h} + \lambda \mathrm{K_f}$$

where $\mathrm{K_h}$ and $\mathrm{K_f}$ are the PID gain vectors for prop-hang and forward-flight operation, respectively. The same interpolation is applied to the relevant proportional, integral, and derivative gains of each attitude-control axis.

The mixing parameter is computed from the current flight condition using flight speed and angle of attack. Two normalised indicators are first defined:

$$\lambda_{\mathrm{v}} = \mathrm{sat}_{[0,1]}\left(\frac{V - V_{\mathrm{h}}}{V_f - V_{\mathrm{h}}}\right)$$

$$\lambda_{\mathrm{a}} = \mathrm{sat}_{[0,1]}\left(\frac{a_{\mathrm{h}} - a}{a_{\mathrm{h}} - a_{\mathrm{f}}}\right)$$

where Vh and Vf define the approximate speed range between prop-hang and forward-flight operation, and ah and af define the corresponding angle-of-attack range. The final mixing parameter is then computed as:

$$\lambda = \mathrm{sat}_{[0,1]}(\omega_{\mathrm{v}}\lambda_{\mathrm{v}} + \omega_{\mathrm{a}}\lambda_{\mathrm{a}})$$

$$\omega_{\mathrm{v}} + \omega_{\mathrm{a}} = 1$$

Here, $\omega_{\mathrm{v}} = 0.6$ and $\omega_{\mathrm{a}} = 0.4$ are weighting factors that determine the relative influence of speed and angle of attack in identifying the current flight regime. The saturation operator ensures that λ remains between 0 and 1. Therefore, low speed and high angle of attack bias the controller toward the prop-hang tuning, while higher speed and lower angle of attack bias it toward the forward-flight tuning.

In the present indoor experiments, V can be approximated from the motion-capture-derived ground speed because the indoor environment provides near-still-air conditions. When direct onboard

airspeed measurement is not available, the angle of attack can be estimated from the aircraft pitch attitude and the motion-capture-derived flight-path angle. In implementation, the computed value of λ can also be passed through a smoothing function to reduce abrupt gain changes. This gain-blending strategy provides a continuous transition between the prop-hang and forward-flight controllers during prop-hang-to-forward-flight and forward-flight-to-prop-hang manoeuvres. The blending law is used as an implementation-level scheduling strategy rather than as an optimised gain-scheduling design.

3.3 Software Modules and Multi-Rate Implementation

The onboard and communication software is organised into modular tasks, as shown in Figure 4. The Teensy 4.1 runs the main real-time flight-control functions, including sensor fusion, attitude control, actuator output, telemetry handling, and flight logging. The ESP32 provides the communication interface between the onboard controller and the ground-station system. This separation allows time-critical sensing, stabilisation, and actuator-output functions to remain on the Teensy, while command reception, telemetry forwarding, and communication management are handled by the ESP32.

Figure 4 shows the data flow between the ESP32 communication layer and the Teensy 4.1 flight-control layer. The ESP32 receives reference commands from the ground-station system and forwards them to the Teensy through a MAVLink-based serial interface. It also receives telemetry from the Teensy and sends it back to the ground station for monitoring and logging. Optional supervisory input can also be converted into an SBUS-style command interface if required. On the Teensy side, the sensor-fusion module provides filtered attitude and angular-rate estimates to the attitude-control module. The attitude-control module computes the actuator command, which is then converted by the actuator-output module into servo and motor commands. The telemetry and flight-log modules record system state, command, actuator, and status information for real-time monitoring and post-flight analysis.

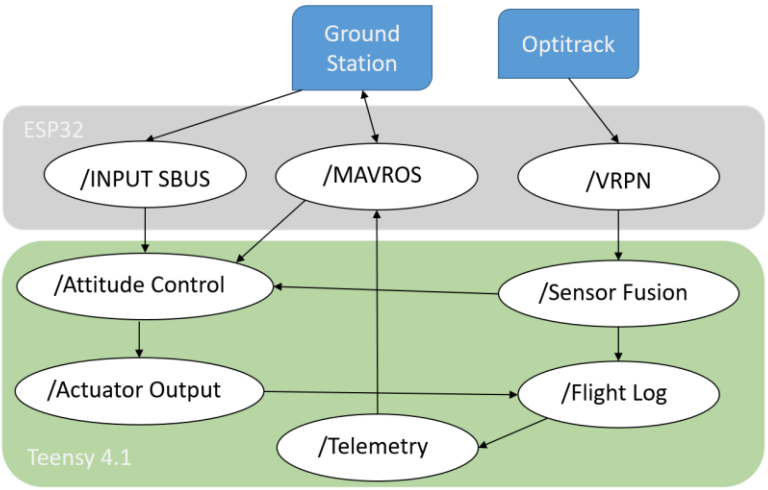


Figure 4 – Onboard software modules and communication interfaces

Different software functions require different update rates. Attitude stabilisation and state estimation must run at high rate, while telemetry transmission, command reception, and logging can operate at lower or task-specific rates. To support this multi-rate structure, FreeRTOS is used on both the Teensy 4.1 and the ESP32. On the Teensy, FreeRTOS schedules sensing, control, actuator output, telemetry, and flight-logging tasks. On the ESP32, FreeRTOS separates MAVLink communication, command handling, and optional input-conversion tasks. The main onboard and communication software modules are summarised in Table 4.

Software module / task	Main function	Platform	Running rate

ESP32 MAVLink Bridge	Receives offboard reference commands and forwards them to the Teensy 4.1 through MAVLink over UART; sends onboard telemetry back to the ground station	ESP32	50 Hz
SBUS input converter	Optional interface for converting supervisory or manual input into SBUS-style commands if required	ESP32	50 Hz
Sensor fusion	Fuses onboard IMU measurements with available OptiTrack external-state updates using a Kalman-filter-based estimator; publishes filtered attitude, angular-rate, and motion states for control and logging	Teensy 4.1	180 Hz
Attitude control	Runs PID-based attitude/rate control using the current reference command and blended gain set; computes control-surface and throttle commands	Teensy 4.1	180 Hz
Actuator output	Converts actuator commands into PWM outputs for servos and digital throttle commands for the ESC	Teensy 4.1	PWM 50 Hz; DShot600
Telemetry	Collects attitude, command, actuator, battery, and system-status data and sends telemetry to the ESP32	Teensy 4.1	20 Hz
Flight logging	Records high-rate IMU, reference, actuator, telemetry, and mode information for post-flight analysis	Teensy 4.1	500 Hz local logging

Table 4 – Onboard and communication software modules.

The MAVLink interface is used because it provides a standard and extensible message structure [11] for command and telemetry exchange. In the present system, MAVLink connects the ROS 2 ground station, ESP32 Bridge, and Teensy-based onboard controller. This design also provides a path toward future compatibility with ArduPilot-based UAVs. For example, the same ESP32-based MAVLink interface could be used to transmit real-time motion-capture position and attitude data to an ArduPilot vehicle using external-vision or odometry-style MAVLink messages. That broader autopilot-integration capability is outside the scope of the present paper and will be reported separately.

The present section focuses on the onboard and communication-side implementation. The ground-station processing of OptiTrack measurements, predefined target-point tracking, and future real-time trajectory generation are described in Section 3.4.

3.4 Offboard Motion-Capture Feedback and Reference Tracking

The offboard layer is implemented on a ground-station computer running ROS 2. Its main role is to process external motion-capture measurements, generate reference commands for the aircraft, monitor the experiment, and record flight-test data. In contrast to the onboard layer, the offboard layer does not generate direct actuator commands. Instead, it operates at the reference-command level and sends desired motion or attitude-related commands to the onboard controller through the ESP32 MAVLink Bridge.

The offboard software structure is shown in Figure 5. The OptiTrack system provides rigid-body position and attitude measurements of the aircraft through the VRPN interface. These measurements are received by the VRPN receiver node and converted into a ROS 2 odometry message, which provides the global position, attitude, and, when required, velocity estimates for feedback control and post-flight analysis. The odometry information is then combined with onboard inertial and telemetry data through a Kalman-filter-based state-estimation process [16]. The resulting fused state estimate provides smoother and more reliable position, velocity, attitude, and angular-rate information for the offboard reference-tracking controller, which computes command references for the onboard controller.

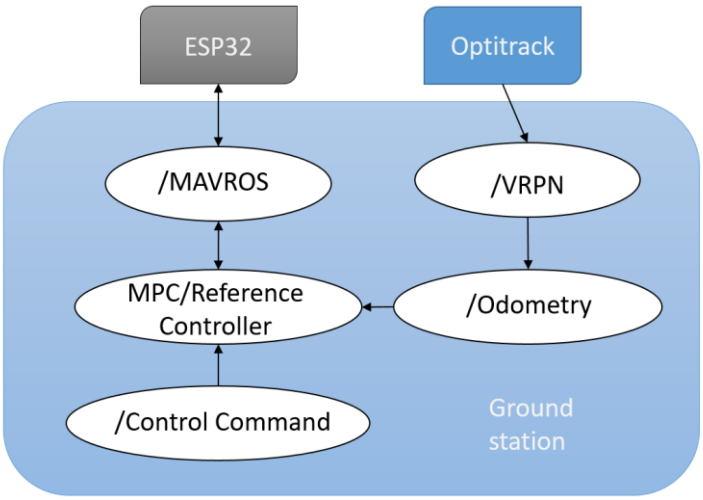


Figure 5 – Offboard ROS 2 software architecture.

In the present implementation, the flight trajectory is defined using pre-generated target points rather than online path generation. For the circular tracking case, these target points define the desired circular path within the indoor flight volume. For the transition case, the reference sequence defines the main flight phases, including prop-hang, transition to forward flight, level-flight segment, and recovery back to prop-hang. The offboard controller compares the measured aircraft state with the current target reference and generates reference-level commands such as desired attitude, body-rate, velocity-related, or throttle-related commands.

The command interface between the offboard and onboard systems is implemented through MAVROS and MAVLink. MAVROS provides the ROS 2-side interface for transmitting command messages and receiving telemetry, while the ESP32 Bridge handles the communication link to the onboard Teensy 4.1 flight controller. This structure keeps the offboard software compatible with standard MAVLink-based workflows and also provides a path for future integration with autopilot platforms such as Ardupilot. In future work, the same interface can be used to transmit real-time external-position or odometry information to MAVLink-compatible UAVs for motion-capture-based indoor flight testing. That broader autopilot-integration capability will be reported separately.

The current offboard implementation focuses on repeatable experimental flight testing rather than fully autonomous online trajectory planning. Since the target points are generated before flight, the same reference sequence can be repeated across multiple experiments, which is useful for controller tuning, performance comparison, and flight-dynamics data collection. Future work will extend the offboard layer to include real-time trajectory generation, allowing the reference path to be updated online according to the measured aircraft state, available indoor flight volume, and manoeuvre objectives.

Table 5 summarises the main offboard ROS 2 software modules used in the present system.

Software module / task	Main function	Running rate
VRPN receiver	Receives OptiTrack rigid-body position and attitude measurements through VRPN and publishes the measured aircraft pose	180 Hz
Odometry conversion	Converts VRPN pose measurements into ROS 2 odometry format and estimates velocity when required	180 Hz
Target/reference manager	Provides pre-generated target points or manoeuvre references for the selected flight case	50 Hz
Reference-tracking controller	Computes reference-level commands from the target reference and measured aircraft state	50 Hz
MAVROS interface	Transmits reference commands to the ESP32 MAVLink Bridge and receives telemetry from the aircraft	50 Hz

Table 5 – Offboard ROS 2 software modules

3.5 Data Logging and Synchronization

Flight-test data are recorded using a timestamp-based software synchronization strategy. The Teensy 4.1 records high-rate onboard measurements using its local microsecond-resolution clock, while the

ROS 2 ground-station computer records OptiTrack measurements, reference commands, and telemetry using the ROS 2 time source [12]. This arrangement avoids the need to stream all high-rate onboard data to the offboard computer during flight, while still allowing the onboard and offboard datasets to be aligned during post-processing.

The primary flight log is stored on the onboard microSD card. This log includes IMU angular-rate and acceleration measurements, estimated attitude states, received reference commands, controller outputs, actuator commands, throttle commands, battery information, and selected telemetry variables. Local SD-card logging is used because the IMU data rate is significantly higher than the command and telemetry rate, and continuously transmitting the complete IMU stream to the ground station would consume excessive wireless bandwidth. Instead, the wireless link is reserved for reference commands, selected telemetry, and synchronization messages.

Clock synchronization between the onboard and offboard systems is performed using timestamped message exchange through the ESP32 MAVLink Bridge. The method follows the same principle as two-way network time transfer [13], where clock offset and round-trip delay are estimated from timestamped request and response packets. A similar mechanism is used in the MAVLink TIMESYNC protocol [14] for estimating clock offsets between MAVLink components. During each experiment, synchronization packets are exchanged between the ROS 2 ground station and the Teensy 4.1. These packets are used to estimate the offset, and when necessary the drift, between the Teensy clock and the ROS 2 ground-station clock. The onboard log is then mapped onto the ROS 2 time axis during post-processing.

OptiTrack measurements are received by the ROS 2 ground station through the VRPN interface. The VRPN/ROS interface supports timestamping tracking data using either the VRPN server timestamp or the local ROS time [10]. In this work, OptiTrack rigid-body position and attitude measurements are continuously broadcast to the ROS 2 offboard layer for feedback control and are recorded together with reference commands and telemetry. Where available, the original motion-capture frame timestamp is preserved; otherwise, the ROS 2 reception timestamp is used. This provides a common time basis for comparing external trajectory measurements with onboard sensor and actuator data.

After each experiment, the onboard SD-card log, OptiTrack measurements, ROS 2 reference commands, and telemetry records are aligned on the ground-station time axis. The synchronized dataset provides the measured aircraft motion, high-rate inertial response, reference commands, actuator outputs, and external trajectory measurements required for controller evaluation, flight-performance analysis, and output-error-based model refinement. The present implementation provides software-level synchronization rather than hardware-triggered camera-IMU synchronization.

4. Flight-Dynamics Analysis Framework

The flight-dynamics analysis in this paper is intended to demonstrate the data-analysis capability enabled by the proposed experimental architecture, rather than to identify a complete aerodynamic database from flight data alone. The baseline aerodynamic model is taken from the previous wind-tunnel and modelling study [9], while the present flight data are used to illustrate how time-synchronised state, command, and actuator measurements can support selected aerodynamic and propulsion-parameter refinement. This analysis relies on the synchronised logging architecture described in Section 3.5, which provides aligned actuator commands, onboard measurements, OptiTrack states, and reference-command records.

4.1 Baseline Model and Wind-Tunnel Prior

The baseline flight-dynamics model used in this work is derived from the aerodynamic and propulsion characterisation reported in [9]. That study combined component-level modelling, wind-tunnel force and moment measurements, control-surface sweeps, propulsion tests, and preliminary output-error analysis to obtain a simplified aerodynamic model for the compact F3P biplane platform. The present paper does not repeat the wind-tunnel setup, calibration procedure, data-reduction process, or coefficient-fitting method. Instead, the resulting geometric, mass, aerodynamic, and propulsion parameters are used as prior information for the flight-dynamics analysis.

The aircraft is represented using a six-degree-of-freedom rigid-body model. The aerodynamic force and moment coefficients are expressed as functions of the relevant flight variables and control inputs, including angle of attack, sideslip angle, angular rates, control-surface deflections, and throttle-dependent propulsion effects. The reference area, reference span, mean aerodynamic chord, and

baseline aerodynamic coefficients are taken from the prior wind-tunnel model summarised in Tables 1 and 2. This provides a physically meaningful initial model before flight-test-based refinement is applied.

The use of wind-tunnel data as prior information is important for the present aircraft because the available flight-test data cover only selected manoeuvres and do not provide uniform excitation across the full aerodynamic envelope. In particular, the transition flight mainly excites the longitudinal dynamics, thrust response, and high-angle-of-attack behaviour, while the circular trajectory case provides stronger excitation of lateral-directional tracking and turning behaviour. The wind-tunnel prior therefore constrains the model to remain physically reasonable while allowing selected parameters to be adjusted using flight data.

In the present analysis, the wind-tunnel model is not treated as a final truth model. Differences may arise between wind-tunnel measurements and free-flight behaviour because of unsteady motion, propeller-slipstream interaction, structural flexibility, actuator dynamics, and closed-loop control effects. The purpose of the flight-data analysis is therefore to refine selected model parameters around the wind-tunnel prior, rather than to replace the prior aerodynamic database entirely.

4.2 Output-Error Model Refinement

The flight-test-based refinement is formulated using an output-error approach [8]. In this method, the aircraft model is simulated using the recorded initial conditions and measured actuator commands, and the simulated outputs are compared with the measured flight states. The parameters are then adjusted to minimise the difference between the measured and simulated responses.

The workflow used to apply the output-error method to the present closed-loop flight-test dataset is shown in Figure 6.

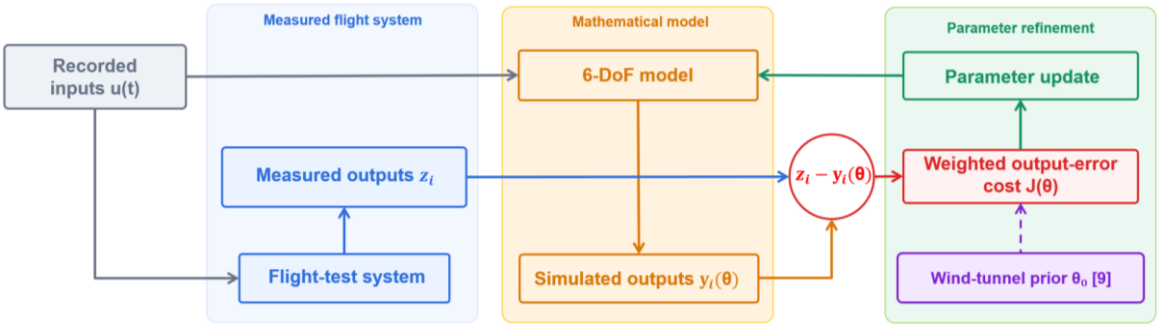


Figure 6 – Output-error model-refinement workflow adapted from the standard flight-vehicle system-identification structure in [8] and applied to the present closed-loop flight-test dataset.

Figure 6 illustrates how recorded closed-loop flight data are used for model refinement. The recorded actuator commands are treated as the known model inputs, denoted by u(t), and are applied both to the real flight-test system and to the mathematical model. The measured flight response, denoted by z(t), is obtained from the synchronised OptiTrack, onboard IMU, telemetry, and actuator-log data. In parallel, the six-degree-of-freedom model is integrated using the same input history to generate the simulated output response, denoted by y(t).

The response error is formed as z(t) - y(t). This error is used in a weighted output-error cost function, J(θ), where θ represents the aerodynamic and propulsion parameters being refined. The parameter update step adjusts selected elements of θ to reduce the output error. The wind-tunnel prior, θ_0, obtained from [9], is used to initialise the model and to constrain weakly excited parameters during the refinement process.

For each flight segment, the recorded actuator commands are treated as known model inputs. These include the aileron, elevator, rudder, and throttle commands generated by the onboard controller. The measured outputs include the aircraft position, velocity, attitude, and angular-rate information obtained from the synchronised onboard and OptiTrack datasets. This formulation is suitable for closed-loop flight data because the controller itself does not need to be identified. Instead, the closed-loop controller is treated as the mechanism that generated the measured actuator inputs.

For a given parameter vector θ, the simulated aircraft response can be written as:

$$\dot{x}(t) = f(x(t), u(t), \theta)$$

$$\hat{y}(t) = h(x(t), u(t), \theta)$$

where x(t) is the aircraft state vector, u(t) is the recorded actuator-input vector, θ contains the aerodynamic and propulsion parameters to be refined, and y(t) is the simulated output vector.

The output-error cost function is expressed as:

$$J(\theta) = \sum_{i=1}^{N} \big(z_i - y_i(\theta)\big)^T R^{-1} \big(z_i - y_i(\theta)\big)$$

where z_i is the measured output at sample i, $y_i(\theta)$ is the corresponding simulated output, and R is a weighting matrix representing the relative confidence in each measured output channel.

Because prior aerodynamic information is available from wind-tunnel testing, an optional prior penalty can be included:

$$J(\theta) = \sum_{i=1}^{N} \big(z_i - y_i(\theta)\big)^T R^{-1} \big(z_i - y_i(\theta)\big) + (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where θ_0 is the wind-tunnel-based prior parameter vector and P_0 represents the assumed uncertainty of the prior. This term prevents poorly excited parameters from drifting to non-physical values and allows the flight data to refine only the parameters that are sufficiently observable from the available manoeuvres.

Compared with the preliminary gliding-test output-error formulation in [9], the present formulation retains the recorded closed-loop actuator commands as known model inputs. This allows transition and circular-flight datasets to be used for selected parameter refinement, rather than requiring the controller itself to be included as part of the identified model.

The synchronised logging architecture described in Section 3.5 is essential for this analysis. The actuator commands, onboard IMU data, OptiTrack measurements, reference commands, and telemetry records must be aligned on a common time axis before the output-error calculation is performed. This allows the measured closed-loop response to be compared directly with the simulated aircraft response under the same input history.

4.3 Scope and Identifiable Parameters

The present output-error analysis is limited to selected parameter refinement. It is not intended to identify a complete aerodynamic database or to replace the wind-tunnel model. This limitation is deliberate, because the available flight cases provide strong excitation only in certain parts of the flight envelope.

The transition manoeuvre is most useful for refining longitudinal and propulsion-related parameters. During the prop-hang-to-forward-flight and forward-flight-to-prop-hang phases, the aircraft experiences large changes in pitch attitude, flight speed, throttle demand, and elevator activity. This makes the dataset suitable for examining selected pitch-moment, drag, elevator-effectiveness, and thrust-scale parameters. Candidate longitudinal parameters include:

$$C_{m0}, \quad C_{m\alpha}, \quad C_{mq}, \quad C_{m\delta_e}$$

as well as drag correction terms and a propulsion scale factor.

The circular trajectory case provides more information about lateral-directional behaviour and turning performance. The repeated turning motion excites roll response, aileron activity, yaw-rate response, and turn-induced drag effects. Candidate lateral-directional parameters include:

$$C_{l\delta_a}, \quad C_{lp}, \quad C_{nr}$$

as well as possible turn-drag or thrust-correction terms. However, because the circular flight does not independently excite all lateral-directional states over a broad envelope, parameters such as:

$$C_{Y\beta}, \quad C_{l\beta}, \quad C_{n\beta}$$

are retained from the wind-tunnel prior unless additional manoeuvres become available.

The identifiable parameter set is therefore selected based on manoeuvre excitation, sensor availability, and physical interpretability. Parameters with weak excitation or strong correlation with other parameters are fixed at their wind-tunnel prior values. This approach keeps the estimation problem well-conditioned and avoids overfitting the limited flight-test dataset.

The final output of the analysis is a refined flight-dynamics model that remains anchored to the wind-tunnel aerodynamic database while incorporating corrections suggested by closed-loop flight data. The refined model can then be evaluated by comparing measured and simulated flight responses, including attitude histories, angular rates, speed trends, throttle response, and trajectory reconstruction.

5. Results and Discussion

This section presents representative indoor flight-test results obtained using the proposed hybrid onboard-offboard architecture. The available dataset consists of four prop-hang–to–level-flight–to–prop-hang transition flights and one circular trajectory-tracking flight. These tests are used to evaluate the feasibility of the control architecture, the quality of the synchronised onboard-offboard logging system, and the suitability of the resulting data for the flight-dynamics analysis described in Section 4. The results should therefore be interpreted as a representative experimental validation rather than a statistical performance assessment over a complete flight envelope.

5.1 Transition Results

Four prop-hang–to–forward-flight–to–prop-hang transition tests were conducted to evaluate the ability of the proposed platform to perform aggressive mode changes in a confined indoor environment. The four trials were designed to compare two transition objectives and two transition strategies. Trials 1 and 3 were performed with the objective of completing the transition within the shortest possible distance, whereas Trials 2 and 4 were conducted over a fixed transition distance of 8 m with a nominal forward-flight speed command of 2 m/s. Trials 1 and 2 used a pre-dive strategy before the recovery phase, while Trials 3 and 4 used a direct transition strategy without a deliberate pre-dive.

The altitude responses of the four transition trials are compared in Fig. 7. The vertical position is used as the main comparison metric because the transition between prop-hang and forward flight is strongly constrained by the available indoor flight volume. A successful manoeuvre must not only complete the attitude and speed change, but also limit altitude loss and recover to a controlled prop-hang condition. This is particularly important for indoor fixed-wing flight, where even small altitude deviations can reduce the available recovery margin.

By contrast, the no-pre-dive trials show that the aircraft can transition more directly without first descending to gain speed. Trial 3 demonstrates that a minimum-distance transition can be achieved without a deliberate pre-dive. Trial 4 is particularly important because it uses the same 8 m transition distance and 2 m/s nominal speed command as Trial 2, but avoids the large altitude loss associated with the pre-dive strategy. This indicates that, for the present platform, direct transition is feasible and can better preserve altitude during the manoeuvre.

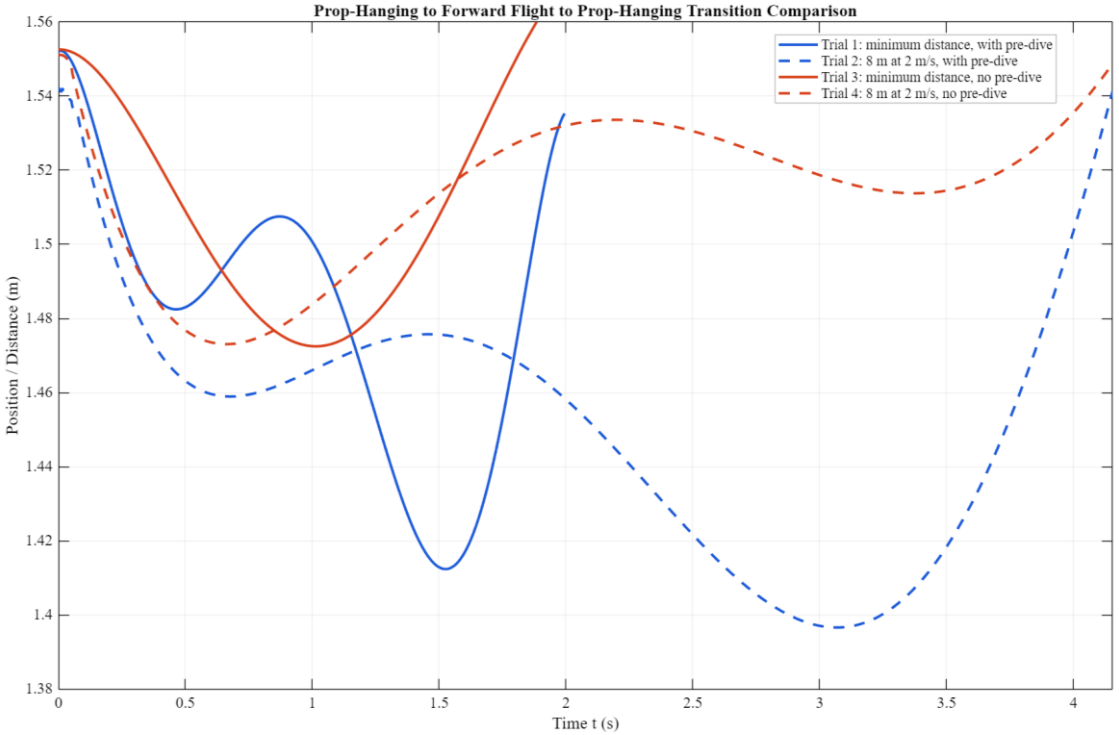


Figure 7 – Altitude response comparison for four transition trials. Trials 1 and 3 target minimum-distance transitions, while Trials 2 and 4 use an 8 m transition distance with a nominal forward-flight speed command of 2 m/s. Trials 1 and 2 use a pre-dive strategy, whereas Trials 3 and 4 use a direct no-pre-dive strategy.

The comparison also highlights the trade-off between transition compactness, forward-speed margin, and altitude preservation. The minimum-distance cases require faster attitude and thrust changes, which can lead to sharper transient responses. The fixed-distance 8 m cases allow the manoeuvre to be distributed over a longer flight path and are therefore more suitable for controlled forward-flight entry and recovery at a prescribed speed. However, the difference between Trials 2 and 4 shows that the transition strategy remains important even when the commanded distance and speed are fixed.

Overall, the results indicate that a deliberate pre-dive is not necessary for completing the prop-hang–to–forward-flight–to–prop-hang transition on this lightweight F3P platform. While pre-diving can provide additional forward-speed margin, it produces larger altitude loss and is therefore less desirable for indoor flight. The direct no-pre-dive strategy is more compact and better preserves altitude, suggesting that the strong propeller slipstream over the control surfaces provides sufficient control authority for direct transition between prop-hang and forward flight.

5.2 Circular Trajectory Tracking Demonstration

A circular trajectory-tracking flight was conducted to demonstrate the ability of the proposed hybrid onboard-offboard architecture to support aggressive indoor path-following control. In this test, the offboard controller commanded a circular trajectory with a radius of 1 m and a nominal flight speed of 1.5 m/s. The aircraft was commanded to complete three counter-clockwise laps before returning to the hover region. The target trajectory was generated offboard as a sequence of predefined target points, while the OptiTrack system provided the external position and attitude measurements used by the ROS 2 control framework. The resulting commands were transmitted to the onboard system through the ESP32 MAVLink interface and executed by the Teensy-based attitude-control and actuator-output loops.

The measured XY trajectory is shown in Figure 8. The flight is divided into five phases: initial hover from 0 to 5 s, transition into circular flight from 5 to 6.8 s, circular flight from 6.8 to 22 s, transition back to hover from 22 to 22.7 s, and final hover from 22.7 to 28.6 s. The transition into circular flight was performed within the small central region of the trajectory, rather than by first descending or diving to build up forward speed. In other words, the aircraft was commanded to change directly from a prop-hang-like condition into level circular flight. This is a demanding manoeuvre for a fixed-wing aircraft, because the aircraft must generate forward speed, establish lift, and enter a coordinated turning motion within a very limited space.

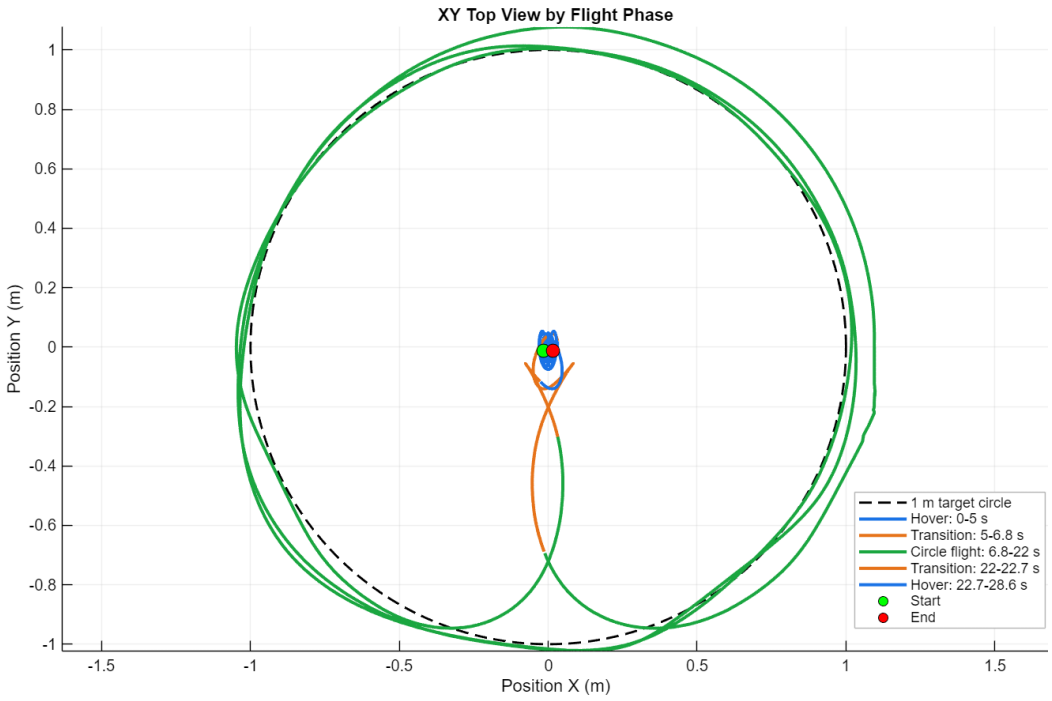


Figure 8 – XY top-view trajectory of the circular flight test. The aircraft transitions from the central hover region into a 1 m-radius counter-clockwise circular trajectory, completes approximately three laps, and then returns towards the hover region.

During the circular-flight phase, the aircraft completed approximately three counter-clockwise laps around the commanded circular path. Although the measured trajectory is not perfectly circular and the centre and radius vary during the manoeuvre, the result demonstrates that the aircraft can sustain repeated turning flight inside a confined indoor test volume. For a small fixed-wing aircraft operating at low Reynolds number and near the boundary between prop-hang and forward flight, maintaining a 1 m-radius circular trajectory at a commanded speed of 1.5 m/s is challenging. The observed deviations are therefore expected and should be interpreted in the context of the limited flight volume, strong attitude-translation coupling, and rapidly changing propwash and aerodynamic effects.

The corresponding vertical position history is shown in Figure 9. During the initial hover phase, the aircraft remains close to a steady height. A noticeable altitude loss occurs during the first transition phase, which indicates the difficulty of preserving vertical position while the aircraft changes directly from prop-hang to forward circular flight without a deliberate diving acceleration phase. During the circular-flight phase, the altitude varies as the aircraft completes the counter-clockwise laps. This variation reflects the coupled effects of low-speed aerodynamics, thrust demand, roll angle, pitch attitude, and limited control authority during tight turning flight. After the circular-flight segment, the aircraft transitions back towards the hover region and recovers to a more stable height.

This circular trajectory test is used as a representative demonstration rather than a statistical evaluation of tracking accuracy. The result demonstrates the integration of the offboard trajectory-command layer, OptiTrack feedback, MAVLink communication, onboard attitude control, and actuator-output generation. It also provides a dataset with lateral-directional excitation that complements the prop-hang–level-flight–prop-hang transition tests discussed in Section 5.1. While additional repeated circular flights would be required to quantify tracking repeatability and disturbance sensitivity, this result shows that the proposed architecture can support a demanding indoor fixed-wing manoeuvre involving direct transition, tight-radius turning, and recovery to hover.

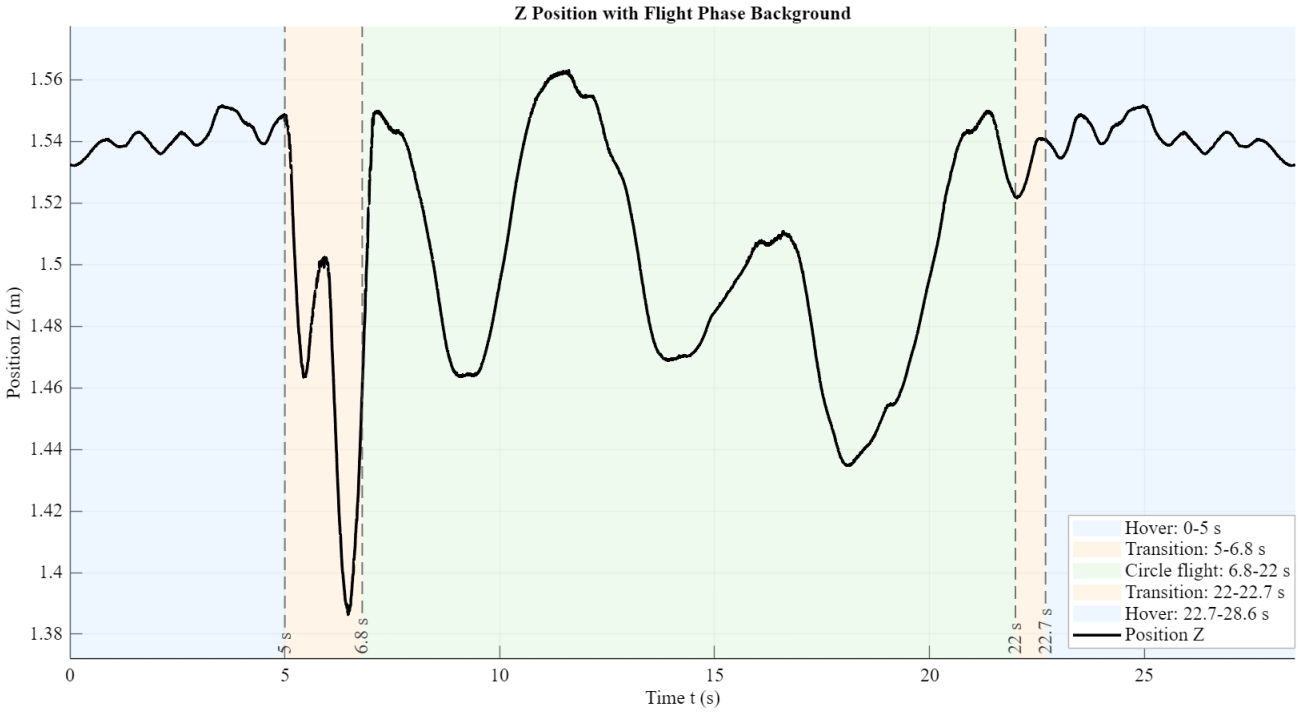


Figure 9 – Vertical position history during the circular trajectory test, showing altitude variation during initial hover, direct transition into circular flight, three counter-clockwise laps, recovery transition, and final hover.

5.3 Discussion: Control Performance and Model-Refinement Usefulness

The transition and circular-flight results demonstrate different aspects of the proposed experimental system. The transition manoeuvres primarily evaluate the ability of the onboard controller to stabilise

the aircraft across a large change in flight condition. During prop-hang, the aerodynamic control moments are strongly influenced by the propeller slipstream, while during level flight the aircraft relies more on conventional forward-flight aerodynamic forces. Successful recovery from level flight back to prop-hang therefore indicates that the controller can operate across a wide range of attitude, speed, and control-effectiveness conditions.

The circular trajectory test, by contrast, demonstrates the role of the offboard control and motion-capture feedback loop. The aircraft follows a commanded path generated offboard, while the onboard system executes the attitude and actuator commands at higher rate. This separation of functions is consistent with the architecture described in Section 3: computationally heavier guidance and trajectory-level control are handled offboard, while time-critical attitude control and actuator output remain onboard.

The flight-test logs provide the data structure required for the model-analysis framework described in Section 4. For both the transition and circular trajectory cases, the recorded actuator commands can be used as known model inputs, while the synchronised OptiTrack and onboard measurements provide the measured aircraft response for comparison with the simulated model output. This is particularly important for output-error analysis, where timing errors between the input command, actuator response, and measured state can introduce artificial phase errors into the response residual.

From a flight-dynamics perspective, the two test types provide complementary information. The transition manoeuvres are most relevant to longitudinal dynamics, including pitch response, elevator effectiveness, throttle response, and propwash-influenced control behaviour. The circular trajectory test provides information about lateral-directional response, roll activity, yaw-rate behaviour, and turn-induced drag effects. The combined dataset is therefore suitable for selected parameter refinement around the wind-tunnel prior introduced in Section 4.

However, the present dataset is not sufficient for identifying a complete aerodynamic database. The flights are closed-loop manoeuvres, and the controller commands are correlated with the aircraft response. In addition, the available manoeuvres do not uniformly excite all aerodynamic modes across the flight envelope. For this reason, the wind-tunnel model remains necessary as prior information, while the flight-test data are used to examine selected corrections and to evaluate whether the baseline model reproduces the measured response trends.

Overall, the results support the main objective of this paper: to demonstrate a practical hybrid onboard-offboard experimental architecture for agile indoor fixed-wing flight testing. The architecture enables closed-loop control, external-state feedback, high-rate onboard data recording, and synchronised post-flight analysis. These capabilities provide the basis for future work on more extensive trajectory generation, broader manoeuvre design, and higher-confidence flight-dynamics model refinement.

5.4 Limitations

The results presented in this section should be interpreted as a representative experimental validation rather than a complete statistical performance assessment. The available dataset consists of four transition flights and one circular trajectory-tracking flight. This is sufficient to demonstrate feasibility of the architecture and to provide example datasets for analysis, but it is not sufficient to establish general tracking-performance statistics over the full flight envelope.

The circular trajectory result is also limited to a single representative flight. Although it demonstrates the integration of the offboard trajectory-command layer, OptiTrack feedback, MAVLink communication, and onboard attitude-control system, additional repeated runs would be required to quantify repeatability, disturbance sensitivity, and tracking accuracy under different speeds, radii, and command profiles.

The experiments also depend on the indoor OptiTrack system for external state feedback. This provides accurate position and attitude measurements inside the test volume, but it limits the current implementation to instrumented indoor environments. Future work will investigate how the same MAVLink-based ESP32 interface can be used to transmit real-time external-position information to other UAV autopilot systems, such as Ardupilot-based platforms. That extension is outside the scope of the present paper and will be reported separately.

Another limitation is that the current offboard trajectory is generated as a predefined sequence of target points. The present implementation is therefore suitable for testing planned manoeuvres and controlled indoor trajectories, but it does not yet include real-time trajectory generation or online replanning. Future work will incorporate real-time trajectory generation so that the offboard controller can respond more flexibly to tracking errors, constraints, and changes in the flight environment.
Finally, the flight-dynamics analysis is limited by the excitation content of the available manoeuvres. The transition flights mainly excite longitudinal and propulsion-related dynamics, while the single circular flight provides limited lateral-directional excitation. As a result, only selected parameters can be meaningfully refined from the current dataset. A broader set of designed excitation manoeuvres, combined with repeated flight trials, would be required for more complete aerodynamic model identification.

6. Conclusions and Future Work

This paper presented a hybrid onboard-offboard architecture for agile indoor fixed-wing UAV flight testing. The system combines onboard high-rate sensing, attitude control, actuator output, and SD-card logging with offboard ROS 2-based trajectory command, OptiTrack feedback, and MAVLink communication. Representative transition and circular trajectory flights demonstrated that the architecture can support aggressive indoor manoeuvres while producing synchronised state-command-actuator datasets for post-flight flight-dynamics analysis. The results show the feasibility of the proposed platform as both a closed-loop control testbed and a data-generation system for selected model refinement around a wind-tunnel-based prior.

Future work will extend the flight-test dataset with richer excitation manoeuvres, incorporate real-time trajectory generation and online replanning, and further develop the ESP32 MAVLink interface for transmitting external-position information to autopilot-based UAV systems such as Ardupilot.

7. Contact Author Email Address

The contact author email address Yuchen Huang: yhua0329@uni.sydney.edu.au, KC Wong: kc.wong@sydney.edu.au.

8. Copyright Statement

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

References

[1] Mueller, T. J., and DeLaurier, J. D., “Aerodynamics of Small Vehicles,” Annual Review of Fluid Mechanics, Vol. 35, 2003, pp. 89–111. https://doi.org/10.1146/annurev.fluid.35.101101.161102

[2] Selig, M. S., “Modeling Full-Envelope Aerodynamics of Small UAVs in Realtime,” AIAA Atmospheric Flight Mechanics Conference, AIAA Paper 2010-7635, Toronto, Ontario, Canada, 2010. https://doi.org/10.2514/6.2010-7635

[3] Frank, A., McGrew, J. S., Valenti, M., Levine, D., and How, J. P., “Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane,” AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA Paper 2007-6318, Hilton Head, SC, 2007. https://doi.org/10.2514/6.2007-6318

[4] Johnson, E. N., Wu, A. D., Neidhoefer, J. C., Kannan, S. K., and Turbe, M. A., “Flight-Test Results of Autonomous Airplane Transitions Between Steady-Level and Hovering Flight,” Journal of Guidance, Control, and Dynamics, Vol. 31, No. 2, 2008, pp. 358–370. https://doi.org/10.2514/1.29261

[5] Cory, R., and Tedrake, R., “Experiments in Fixed-Wing UAV Perching,” AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA Paper 2008-7256, Honolulu, HI, 2008. https://doi.org/10.2514/6.2008-7256

[6] Green, W. E., and Oh, P. Y., “A MAV That Flies Like an Airplane and Hovers Like a Helicopter,” Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, CA, 2005, pp. 693–698.

[7] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y., “An Application of Reinforcement Learning to Aerobatic Helicopter Flight,” Advances in Neural Information Processing Systems, Vol. 19, 2007, pp. 1–8.

[8] Jategaonkar, R. V., Flight Vehicle System Identification: A Time-Domain Methodology, 2nd ed., AIAA Education Series, American Institute of Aeronautics and Astronautics, Reston, VA, 2015. https://doi.org/10.2514/4.102790

[9] Huang, Y., and Wong, K. C., “A Comprehensive Approach to Dynamic Modeling of a Compact Agile Fixed-Wing UAV: Bridging Theory and Experimentation,” The University of Sydney, 2024.

[10] ROS Wiki, “vrpn_client_ros,” online documentation. Available: https://wiki.ros.org/vrpn_client_ros

[11] MAVLink Developer Guide, “MAVLink Protocol,” online documentation. Available: https://mavlink.io/en/

[12] Foote, T., “Clock and Time,” ROS 2 Design, Open Robotics, online documentation. Available: https://design.ros2.org/articles/clock_and_time.html

[13] Mills, D., Martin, J., Burbank, J., and Kasch, W., “Network Time Protocol Version 4: Protocol and Algorithms Specification,” RFC 5905, Internet Engineering Task Force, 2010. https://doi.org/10.17487/RFC5905

[14] MAVLink Developer Guide, “Time Synchronization Protocol v2,” online documentation. Available: https://mavlink.io/en/services/timesync.html

[15] Barlow, J. B., Rae, W. H., and Pope, A., Low-Speed Wind Tunnel Testing, 3rd ed., Wiley, New York, 1999.

[16] Grewal, M. S., and Andrews, A. P., Kalman Filtering: Theory and Practice with MATLAB, 4th ed., Wiley, Hoboken, NJ, 2015.