https://doi.org/10.1016/j.mechatronics.2018.10.001

Design and Implementation of a Real-Time Hardware-in-the-Loop Testing Platform for a Dual-Rotor Tail-Sitter Unmanned Aerial Vehicle

Jingxuan Sun [1], Boyang Li [1], Chih-Yung Wen [1] and Chih-Keng Chen [2,*]

[1] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong; jingxuan.j.sun@connect.polyu.hk(J.S.); boyang.li@connect.polyu.hk(B.L.); chihyung.wen@polyu.edu.hk (C.-Y.W)

[2] Department of Vehicle Engineering, National Taipei University of Technology, Taipei, Taiwan;

* Correspondence: ckchen@ntut.edu.tw; Tel.: +886-2-2771-2171(ext. 3625)

Abstract: Tail-sitter vertical take-off and landing (VTOL) vehicle is a promising airframe among all the unmanned aerial vehicles (UAV), although challenges regarding the control strategies remain for civil applications, such as high susceptibility to wind disturbance while hovering. A real-time hardware-in-the-loop (HIL) simulation method for a tail-sitter UAV, an efficient tool for developing the control system, is presented in this paper. A nonlinear six-degrees-of-freedom dynamic model that covers the full angle-of-attack range is derived using the component breakdown approach. The environmental model is further introduced in the real-time simulation application to provide prevailing and gust wind conditions. A commonly used open-source flight controller was embedded in the proposed HIL framework. This HIL testbed can help researchers minimise the time spent debugging the controller program and moving from the simulation control system to the practical control one. The HIL simulation system was validated with the typical complete flight scenarios of a tail-sitter, including hovering, forward transition, cruise, and back transition. The results demonstrate that the HIL system can be an efficient tool for verifying the performance of hardware and software designs of the control system at the development stage for tail-sitter UAVs.

Keywords: Hardware-in-the-loop; Unmanned Aerial Vehicle; Tail-sitter; Vertical take-off and land;

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have proliferated in various applications due to their ability to perform a wide range of missions in complex and hazardous environments. Among the various types of UAVs, vertical take-off and landing (VTOL) UAVs have the unique features of VTOL and efficient lift generation with wings, and they therefore offer both flight endurance and agility. Their superior flight performance makes VTOL UAVs particularly suitable for applications in goods delivery, reconnaissance, search and rescue missions [1], mapping [2], especially in urban areas. The tail-sitter, one type of VTOL UAV, has attracted the research interest of scientists and engineers due to its simple mechanism and high reliability [3-5] , and many related development projects have been established.

However, the advantages of the tail-sitter result in great challenges regarding control algorithm design and development due to its particular flight scenario [4,6-11]. An effective approach to examine the reliability of a developed control system and to accelerate the development of control strategies is the use of simulation technologies before real flight tests, including pure simulations [6-8], software-in-the-loop simulations (SIL), or hardware-in-the-loop (HIL) simulations [9]. The HIL framework embeds the physical system in the simulation environment, which enables the detection and prevention of hardware and software malfunctions [10]. The HIL framework can also provide a more accurate evaluation of the flight control system (FCS) and allows a quick transition from the simulation to the experiment. Notably, the time-varying delay of any practical FCS, which is primarily caused by the sampling delay, processing time and data transfer time, can be considered in the HIL framework [11, 12]. In the authors' earlier work on developing the hover control of the tail-sitter [9, 11], the time-varying delay due to the processing time, which caused degradation in the control performance, was identified via HIL simulation technologies. Thus, an HIL testbed for the tail-sitter was necessary to examine the control strategies and performance before flight testing.

HIL simulation technologies are common and rapidly developed for conventional aircrafts such as helicopters [10, 13] and fixed-wing aircrafts in the aerospace industry. With the development of the small UAVs, the corresponding HIL testing approaches were discussed and developed. In such studies, it is common to integrate the commercial flight simulation software, such as X-Plane, to provide aerodynamic prediction [14, 15]. The highly developed aerodynamics of the aircraft and the comprehensive environmental model in X-Plane provide a user-friendly environment for control system development. However, X-Plane is not equipped with the highly nonlinear aerodynamics model of the tail-sitter vehicles, and it thus cannot be directly applied to the tail-sitter vehicles for the whole flight envelope. Therefore, to model the aircraft dynamics more accurately, a more appropriate approach appears to be the separation of the simulator and the flight control computer in an HIL system [10, 16]. In 2007, Muller [16] presented presented an HIL testbed in which both linear and nonlinear models of a small fixed-wing aircraft were investigated and the HIL simulation environment was developed. To avoid interface problems between the avionics and the simulation computer, this HIL simulation environment was coded on the avionics computer. However, although this approach is useful for development and simulation of the autopilot system, the computation power of the on-board autopilot cannot be fully utilised in real flight because when testing in this HIL system, most of the computational power will be used to simulate the nonlinear dynamic model in the avionics computer, and only limited computational power will be available for controller development. Jung and Tsiotras [17] built a 6-degrees-of-freedom (DOF) nonlinear model into an HIL system and used a serial method to establish communication between the simulator and the autopilot. Notwithstanding the current sensor’s update frequency of 1000 Hz, the sensor data were only processed at 20 Hz in this study.

The HIL testbed in aerospace industry requires further development for application in tail-sitter vehicles, particularly in terms of real-time aerodynamic prediction. In the HIL framework developed for power systems in [18], the Simulink toolbox of Real-Time Workshop was used to accelerate the computation and realise the real-time simulation, and it can also be applied to UAVs. At the same time, the development of high-powered embedded computers and open-source flight controller such as Pixhawk and PX4 [19] is progressing rapidly. With these commercial autopilots, researchers can focus on designing their own control strategies without worrying about autopilot system development at the beginning. Open-source autopilot boards have begun to be used as deeply embedded systems that offer a robot operating system (ROS) interface for feedback and control, which enables researchers already familiar with ROS to easily adopt the embedded codebase without learning an application program interface (API) [19]. Open-source autopilots and ROS have been combined and used in many studies of tail-sitter vehicles [8, 20]. The corresponding interface for open-source autopilots and ROS should be introduced and supported. In the authors’ previous work [11], the HIL simulation testbed which supports the ROS network was discussed. The presented method is limited in hover flight due to an incomplete aerodynamic model. Only the flight performance at low angle of attack region (from -10° to 18°) can be predicted. What’s more, the induced flow by the propellers was assumed covering the wing area completely in this study. Due to the small freestream velocity and aerodynamic effect in the hover phase, this assumption is acceptable. However, the model presented in reference [11] will definitely result in a large error in cruise and transition flights.

Therefore, this study aims at designing a complete HIL testbed to simulate and test the control system of a tail-sitter UAV. The aerodynamic database of the tail-sitter vehicle was constructed based on wind tunnel experiments and further expanded to the full angle of attack range. A more comprehensive environment model that contains wind models was introduced to simulate outdoor flight tests. The HIL testbed was examined with the flight tests, and its reliability was demonstrated.

The paper is organised as follows. The tail-sitter UAV system is described in Section 2, and the mathematical model and aerodynamic databases are discussed in Section 3. Section 4 presents the implementation of the real-time HIL simulation testbed. The simulation and flight tests results are presented in Section 5, and Section 6 concludes the paper.

2. Tail-sitter UAV

Before discussing the development of the HIL testbed, the tail-sitter UAV system, including the airframe and navigation system, are described in this section.

2.1 System Architecture

The tail-sitter UAV system consists of 4 main components as shown in Figure 1: 1) airframe, 2) navigation system, 3) remote control unit and 4) data link. The airframe accommodates the relevant actuators and the navigation system sends the control signals to actuators for engaging missions. The remote control unit and the data link enable necessary intervention from the pilot or ground control staff. The details of each component will be introduced in following subsections.

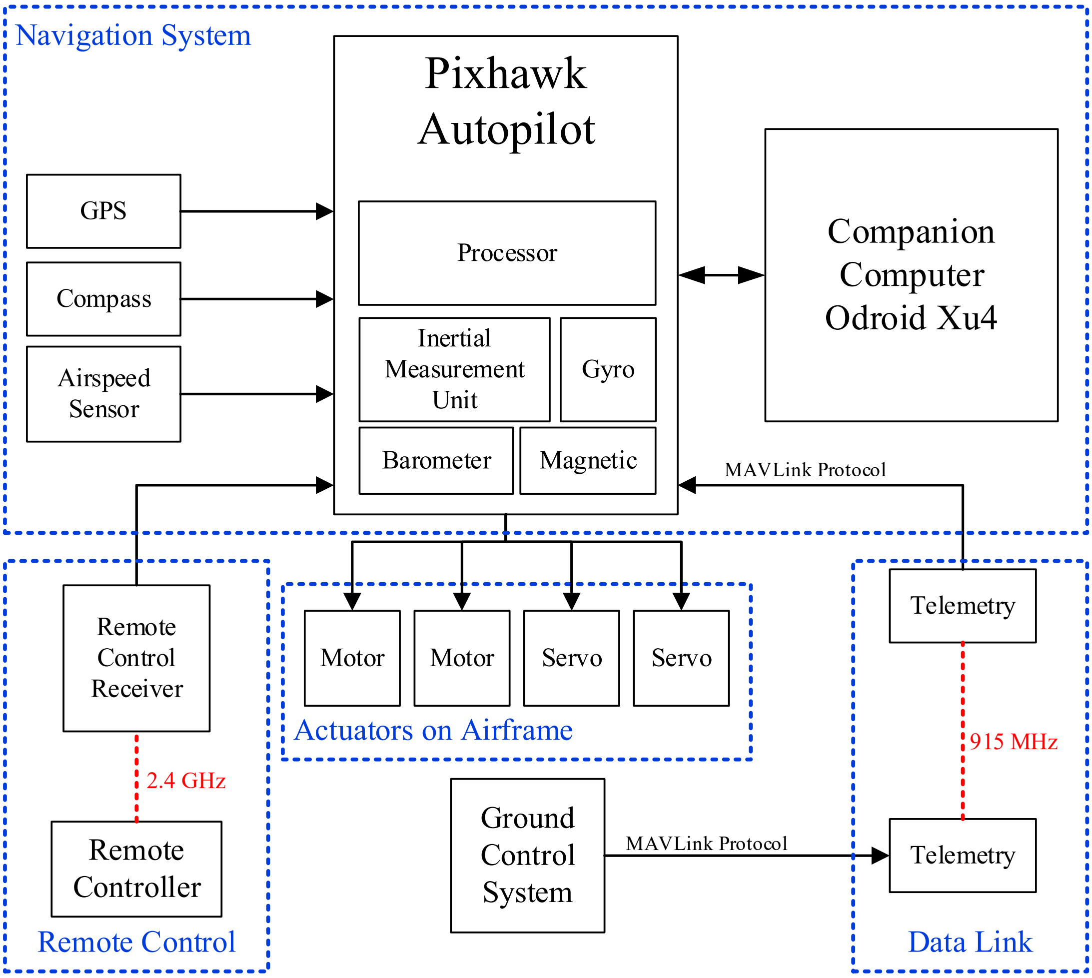


Figure 1. Schematic diagram of the tail-sitter UAV system.

2.2 Airframe of the UAV system

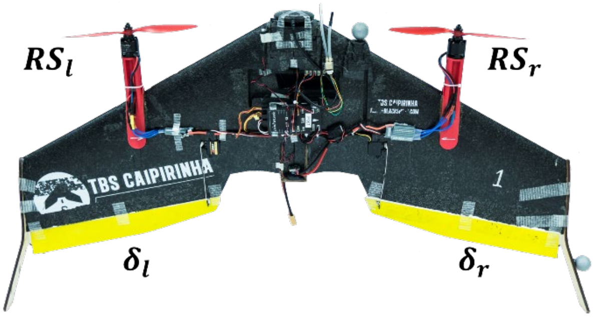


Figure 2. Airframe and control inputs of the modified Caipirinha.

The airframe is the platform of the UAV system, and its configuration and material have to exhibit good aerodynamic performance and reliable structure strength. To eliminate motor torque, the tail-sitter vehicle was proposed in a counter-rotating dual-rotors configuration with a pair of elevons. The preliminary airframe in this study was modified from the Caipirinha, a first-person-view flying wing drone from Team Blacksheep, to achieve VTOL capability. The original Caipirinha is a single-propeller push-type drone with a wingspan of 880 mm. The roll and pitch motions were controlled by a pair of elevons. As shown in Figure 2, the modified Caipirinha airframe is equipped with two brushless motors in conjunction with a pair of 8 × 4.5-inch propellers. The propulsion system and avionics are powered by a 3,000 mAh Lipo 3-cell battery.

2.3 Navigation System

The main component of the navigation system is the flight control unit (FCU). In this study, the Pixhawk flight controller worked with the companion computer Odroid XU4 were used as FCU. The Pixhawk equipped with two processors, one 32-bit STM32F427 works as the main processor and the other 32-bit STM32F103 co-processor as a fail-safe. The Pixhawk is embedded with a gyroscope, an accelerometer/magnetometer, a 3-axis accelerometer/gyroscope, and a barometer. The Odroid XU4 is a small, light, powerful computing device equipped with a 2-GHz core CPU and a 2-GB LPDDR3 RAM. The Pixhawk runs open-source PX4 firmware, and the Odroid XU4 runs Ubuntu 16.04. The deeply embedded interfaces run on Pixhawk. The navigation controller and the attitude controller were developed on the Odroid XU4. The navigation system was also equipped with a GPS/Compass kit and a digital airspeed sensor. Although many small fixed-wing UAVs can fly well without the need for airspeed measurements, tail-sitter vehicles require an airspeed sensor to estimate the point between transition and cruise stages in this study.

2.4 GCS and Data Link

The GCS works via a data link that enables the researcher to monitor or interfere with the UAV during the flight. The open-source software QGroundControl was used as the GCS software. A 3DR 915 Mhz 100 mW radio transmitter and receiver was connected to the GCS laptop, along with a Pixhawk flight controller. The messages between the vehicle and GCS were packed into MAVLink micro air vehicle communication protocol over serial channels and these packets were sent to the radio transmitter.

3. Dynamic Modeling of Tail-sitter UAV

The 6-DOF equations of motion derived in our early work [11] were modified. Notably, the Flat Earth approximation was made when deriving these equations of motion, which is considered acceptable because the tail-sitter is designed with a range of 10 km. In addition, the vehicle is assumed to be a rigid body since the weight is less than 1 kg with a wingspan of 0.88 m.

The tail-sitter vehicle has a large flight envelope and requires a global attitude reference without any attitude singularity. The attitude of the tail-sitter is described in quaternion instead of Euler angles to avoid the singularity in the pitch motion caused by hover flight. Figure 3 shows the body and world coordinate systems, with the x-axis extending toward the nose tip, the y-axis toward the right wing, and the z-axis toward the back of the aircraft.

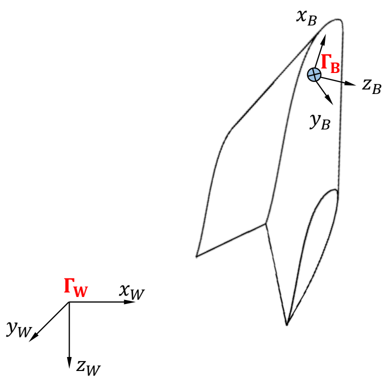


Figure 3. Definitions of body $\mathbf{\Gamma_B}$ and world $\mathbf{\Gamma_W}$ coordinate systems.

The basic rigid-body equations of motion are presented as:

$$\begin{aligned}
\dot{\mathbf{p}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \frac{\mathbf{R}^{\mathbf{T}}(\mathbf{F}^{\mathbf{B}} + \mathbf{F}_{\mathbf{d}}^{\mathbf{B}})}{m} \\
\mathbf{J}\dot{\mathbf{\Omega}} &= -\mathbf{\Omega}\mathbf{J}\mathbf{\Omega} + \mathbf{M}^{\mathbf{B}} + \mathbf{M}_{\mathbf{d}}^{\mathbf{B}} \\
\dot{\mathbf{e}} &= \frac{1}{2}(\mathbf{e}\mathbf{\Omega} + e_0\mathbf{\Omega}) \\
e_0 &= -\tfrac{1}{2}\mathbf{\Omega}^{\mathbf{T}}\mathbf{e},
\end{aligned} \tag{1}$$

where $\mathbf{p}$ denotes the position and $\mathbf{v}$ is the velocity of the UAV in the $\mathbf{\Gamma_W}$. $\mathbf{R}$ is the rotation matrix, which transforms vectors in the World frame $\mathbf{\Gamma_W}$ to the Body frame $\mathbf{\Gamma_B}$ and is determined by the UAV attitude. A quaternion $\mathbf{Q} = [e_0\ \mathbf{e}]^{\mathbf{T}}$ is adopted to described the UAV attitude in turn. $\mathbf{F}_{\mathbf{d}}^{\mathbf{B}}$ and $\mathbf{M}_{\mathbf{d}}^{\mathbf{B}}$ are the disturbing forces and moments in the $\mathbf{\Gamma_B}$, respectively. $\mathbf{J}$ is the inertial matrix and $\mathbf{\Omega}$ is the angular velocity. $\mathbf{F}^{\mathbf{B}}$and $\mathbf{M}^{\mathbf{B}}$ represent the components of the external forces and moments in the $\mathbf{\Gamma_B}$, respectively, and can be written as

$$\begin{aligned}
\mathbf{F}^{\mathbf{B}} &= \mathbf{F_T} + \mathbf{F_{aero}} + \mathbf{F_G} \\
\mathbf{M}^{\mathbf{B}} &= \mathbf{M_T} + \mathbf{M_{aero}},
\end{aligned} \tag{2}$$

where the terms $\mathbf{F_T}$ and $\mathbf{M_T}$ represent the thrust vector and the moments generated by the motors and propellers, respectively. The aerodynamic forces and moments are described by $\mathbf{F_{aero}}$ and $\mathbf{M_{aero}}$, respectively. $\mathbf{F_G}$ is the gravitational force described in the $\mathbf{\Gamma_B}$. The effects of the control surfaces are introduced in aerodynamic forces and moments. Each component is defined and discussed below.

3.1 Propulsion

The thrust vector, $\mathbf{F_T}$, represents of the propulsive forces generated by the two propellers,

$$\mathbf{F_T} = \begin{bmatrix} F_{t,r} + F_{t,l} \\ 0 \\ 0 \end{bmatrix}. \tag{3}$$

As shown in Figure 4, the torque of the two propellers and moments caused by force difference between the two propellers result in the moment vector, $\mathbf{M_T}$:

$$\mathbf{M_T} = \begin{bmatrix} M_{t,r} + M_{t,l} \\ (F_{t,r} + F_{t,r})l_z^m \\ (F_{t,r} - F_{t,l})l_y^m \end{bmatrix}, \tag{4}$$

where the thrust $\mathbf{F_T}$ and the torque $\mathbf{M_T}$ are associated with the airspeed AS, the rotation speeds of the propellers RS, and the angle of attack α. l_y^m and l_z^m are the distances from the propeller disk to the center of gravity along the y-axis and z-axis of the Body frame, respectively. The values of $\mathbf{F_T}$ and $\mathbf{M_T}$ were determined using a three-dimensional look-up table. Wind tunnel experiments were conducted to collect the primary data for the look-up table.

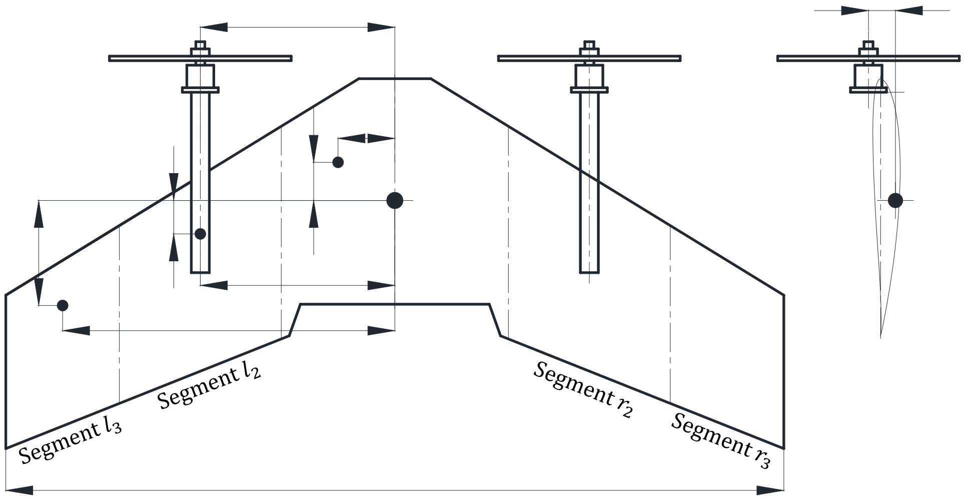


Figure 4. Geometric parameters for the aerodynamic calculation of the Caipirinha tail-sitter.

3.2 Aerodynamics

Unlike the conventional aircraft, the tail-sitter vehicle demands control effectiveness with zero flight speed condition. Thus, its main lift component, the wing, was designed to be inside the region of prop wash during hovering flight. The partial flow conditions over aerodynamic surfaces should be considered in this type of vehicles, and it was modeled based on a component breakdown approach. The aerodynamic surfaces were decomposed into a number of segments accordingly, each of which produced lift, drag, and moments on its aerodynamic center.

The rotation speeds of the propellers change in response to the control commands, which affects the induced velocity and the aerodynamics. Therefore, the main wing of the aircraft was divided into six segments, as shown in Figure 4: Region l_2 and Region r_2 cover the slipstream, and the other regions are exposed to the environment. In following discussion, the subscript *l* or *r* denotes vectors and parameters on the left and right wings, respectively, and the number denotes the segment. For simplicity, two assumptions were made in the aerodynamic model: (1) the speed of the airflow is uniform over each side of the wing and (2) there is no cross coupling between the left and right wings. The aerodynamic force and moment on each side were calculated independently;

The flow condition of each segment was determined by the environment, the flight state, and the prop wash. The airspeed at the aerodynamic center of each segment was calculated based on the wind speed, the flying speed (ground speed), and the contribution of the prop wash, as shown in Eq. (5).

$$\mathbf{V^B} = \mathbf{R_W^B}(\mathbf{V_{wind}^W} + \mathbf{V_{gs}^W}) + \mathbf{V_{induce}^B}, \tag{5}$$

where the wind speed and ground speed were described in the World frame and consist of north, east, and down (NED) components. The momentum theory of propellers and the continuity equation were used to estimate the velocity due to the propeller slipstream, $\mathbf{V_{induce}^B}$, as follows:

$$\mathbf{V}^{\mathbf{B}}_{\mathbf{induce}} = \begin{bmatrix} -u_0 \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

and

$$u_0 = \frac{v_0 cos\alpha + \sqrt{v_0^2 \cos^2 \alpha + 2T_p/\rho A}}{2}\left[1 + \frac{l_p/R_p}{\sqrt{1 + (l_p/R_p)^2}}\right], \tag{7}$$

where α is the angle of attack of the aircraft and T_p is the thrust force generated by the propeller, A is the area of the propeller disk, l_p is the distance between the position of interest and the propeller disk, and R_p is the radius of the propeller disk. The radius of the slipstream tube can also be calculated based on the momentum theory. The second wing segments (l_2 and r_2 in Figure 4) were enveloped in the propeller slipstream tube. v_0 is the velocity of the freestream and is defined as

$$v_0 = \left|\mathbf{V}^{\mathbf{W}}_{\mathbf{wind}} + \mathbf{V}^{\mathbf{W}}_{\mathbf{gs}}\right|. \tag{8}$$

The angles of attack and sideslip angle in the slipstream region are defined as

$$\begin{aligned} \alpha_{eff} &= tan^{-1}(\frac{V_z^B}{V_x^B}) \\ \beta_{eff} &= tan^{-1}(\frac{V_y^B}{V_x^B}) \end{aligned} \tag{9}$$

and the airspeed in the slipstream region is

$$v_{eff} = \sqrt{{V_x^B}^2 + {V_y^B}^2 + {V_z^B}^2}. \tag{10}$$

The aerodynamic coefficients were determined by the calculated angles of attack of both the right-side wing and left-side wing. Considering the large α during the transition, a database of lift coefficient C_L, drag coefficient C_D, and moment coefficient C_M at different α from the wind tunnel experiments was used to estimate the lift, drag, and aerodynamic moment, without linearization to reduce the error of the aerodynamic model. The sideslip angle was assumed to be small, and thus the side force coefficient C_{y_β} was treated as a constant number.

Once these coefficients are determined, the aerodynamic forces and moments of each segment can be estimated as follows:

$$\begin{aligned} L_l &= \sum_{n=1}^{n=3}(\frac{1}{2}C_{L_{l_n}}\rho v_{eff-l_n}^2 S_n) \\ L_r &= \sum_{n=1}^{n=3}(\frac{1}{2}C_{L_{r_n}}\rho v_{eff-r_n}^2 S_n) \\ D_l &= \sum_{n=1}^{n=3}(\frac{1}{2}C_{D_{l_n}}\rho v_{eff-l_n}^2 S_n) \\ D_r &= \sum_{n=1}^{n=3}(\frac{1}{2}C_{D_{r_n}}\rho v_{eff-r_n}^2 S_n) \\ Y_l &= \frac{1}{2}C_{y_\beta}\beta_{eff-l_3}\rho v_{eff-l_3}^2 S_{side} \\ Y_r &= \frac{1}{2}C_{y_\beta}\beta_{eff-r_3}\rho v_{eff-r_3}^2 S_{side}. \end{aligned} \tag{11}$$

The lift, drag, and side force in the Body frame can be expressed as

$$\mathbf{L} = \begin{bmatrix} (L_l + L_r)sin\alpha_{eff} \\ 0 \\ -(L_l + L_r)cos\alpha_{eff} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} -(D_l + D_r)cos\alpha_{eff} \\ 0 \\ -(D_l + D_r)sin\alpha_{eff} \end{bmatrix} \tag{12}$$

$$\mathbf{Y} = \begin{bmatrix} (Y_l + Y_r)sin\beta_{eff} \\ -(Y_l + Y_r)cos\beta_{eff} \\ 0 \end{bmatrix}.$$

Therefore, the aerodynamic forces and moments can be expressed in the Body frame as

$$\mathbf{F_{aero}} = \mathbf{L} + \mathbf{D} + \mathbf{Y} \tag{13}$$

The moment of each segment can be estimated as follows:

$$\mathbf{M_{seg}} = \begin{bmatrix} F_{aero,z}^{seg} l_y^{seg} \\ \frac{1}{2} C_{M_{seg}} \rho v_{seg,eff}^2 S_{seg} \bar{c}_{seg} + F_{aero,z}^{seg} l_x^{seg} \\ F_{aero,x}^{seg} l_y^{seg} \end{bmatrix}, \tag{14}$$

and the total aerodynamic moment contributed from every aerodynamic component can be calculated by

$$\mathbf{M_{aero}} = \sum \mathbf{M_{seg}} \tag{15}$$

3.3 Gravity

The gravitational force of the vehicle of total mass m is expressed as follows:

$$\mathbf{F_G} = \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{16}$$

where g is the acceleration due to gravity.

3.4 Wind Tunnel Experiments

As mentioned above, the tail-sitter vehicle has a large flight envelop. A tail-sitter vehicle can reach large angles of attack during the flight and have significant fight speed changes from the hover phase to the cruise phase. Therefore, the effect on thrust and torque due to the change of airspeed cannot be neglected. The aerodynamic properties at large angles of attack should be carefully estimated for simulating the dynamics of the tail-sitter UAV. In this study, the wind tunnel experiments were conducted to construct the database for the HIL simulation testbed.

3.4.1 Propulsion experiments

The propulsion system of the vehicle and the propulsion database used in this study is the same as that of Sun et al. [11]. To clarify the context of this paper, the experiments setup and data processing methods are summarized in the simplest case here.

Motor and propeller experiments were conducted in a closed-loop low-speed wind tunnel at the Hong Kong Polytechnic University (L × W × H = 2.4 m × 0.6 m × 0.6 m; maximum wind speed, 30 m/s). The thrusts and torques of the propeller for airspeeds ranging from $0\, m/s$ to $15\, m/s$ and motor rotation speeds ranging from 0 to 10,000 RPM were measured using an ATI Mini40E (SI-20N-1Nm) 6-axis force/torque transducer (Figure 5), with a sensing range of $20\ \mathrm{N}$ and resolution of $1/200\ \mathrm{N}$ in the x and y directions and the moment measurement capacity of $1\ \mathrm{Nm}$ and a resolution of $1/8000\ \mathrm{Nm}$. The data were recorded by an NI-9220 data acquisition card at the sampling frequency of 2000 Hz.

One hundred and fifty experiments at various airspeeds and motor speeds were conducted. To create a database that is comprehensive enough to meet the flight situations, the piecewise cubic Hermite interpolation was used [20]. The interpolated data for the force and the moment are plotted in Figure 6, showing a good fit to the experimental data. The whole database is presented in Figure 7, consisting of 3000 sets of the thrust and the torque generated by the piecewise cubic Hermite interpolation. During the following HIL simulations, the data used were calculated linearly based on the surrounding data sets in this database. Therefore, the simulations can make compromises between the efficiency and the accuracy.

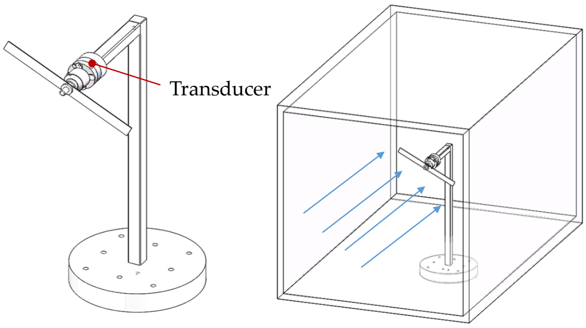


Figure 5. Schematic of thrust experiment [11].

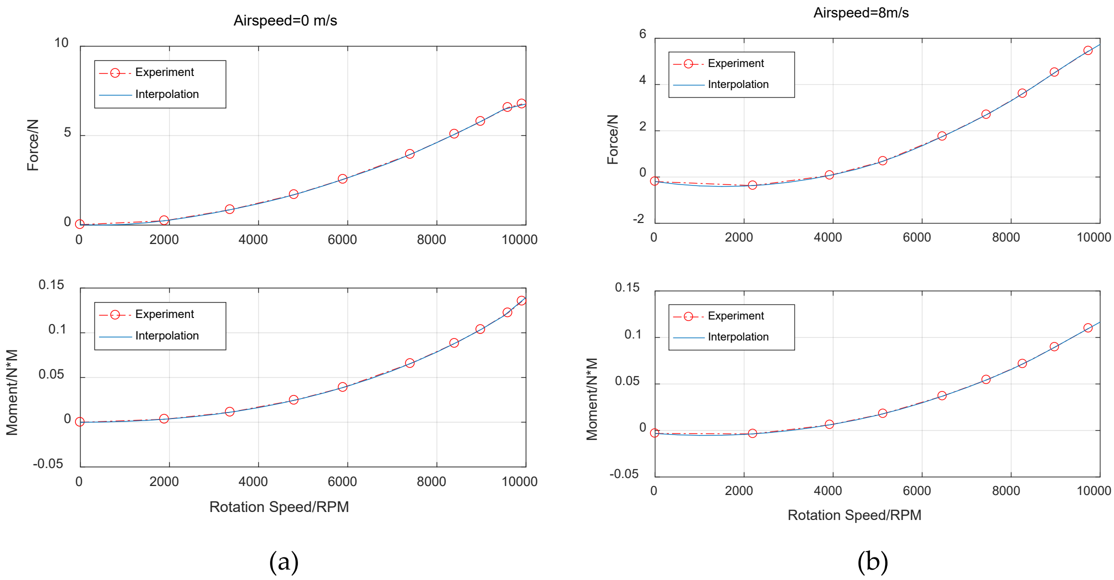


Figure 6. Piecewise cubic Hermite interpolation of force and the moment at airspeeds, a) 0 m/s and b) 8 m/s [11].

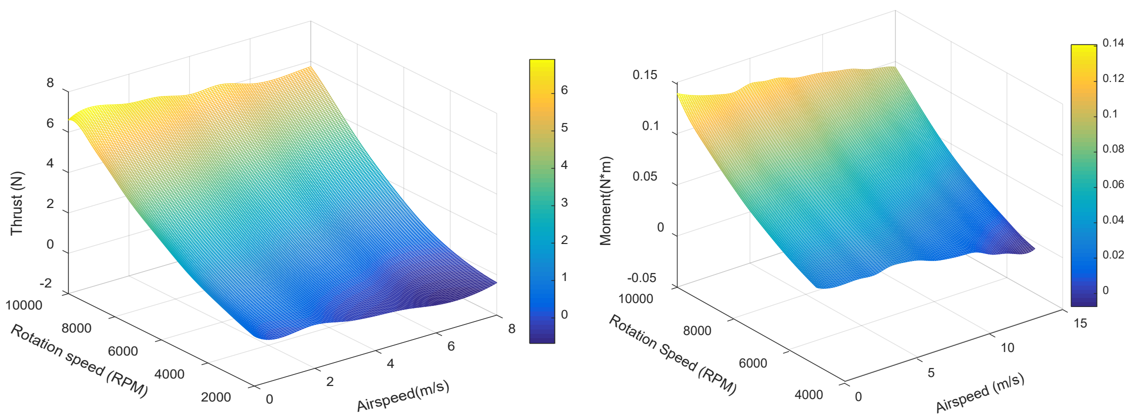


Figure 7. Thrust and moment of motor/propeller at an angle of attack of $0°$ [11].

3.4.2 Aerodynamic experiments

Wind tunnel experiments to measure the aerodynamic forces and moments of the Caipirinha flying wing were conducted in the same wind tunnel using the same ATI Mini40-E 6-axis force/torque transducer, as in the thrust experiments above. The MITSUBISHI HF-KP13 servomotor in conjunction with the MR-J3-10A servo amplifier were applied to change the angle of attack of the experimental model, as shown in Figure 8. The half-aircraft model of photosensitive resin was manufactured with a 3D printer. Limited by the wind tunnel test section size, the model was scaled down (1:2.5) with a maximum blockage ratio of 4.3% [21]. An end plate was applied to control the boundary layer. The model was rotated on a rotating platform to change its angle of attack.

The wind tunnel experiments were conducted with the angle of attack ranging from $-20°$ to $110°$, flap deflection from $-20°$ to $20°$, and Reynolds number from 2.9×10^4 to 1×10^5. For each case, the data were acquired for 60 seconds at a sampling frequency of 1 kHz. As discussed in Section 3.2, segments 1 and 3 are directly exposed to the environment and cause the possibility of an arbitrary angle of attack values. Under this condition, the aerodynamic coefficients were further expanded to $-180°$ to $180°$ based on the experimental data. Interpolation was further conducted to increase data density. Figure 9 shows the database of aerodynamic coefficients when the aircraft applied zero flap deflection. Similar to the thrust database, the data used in the HIL simulations were calculated linearly based on the surrounding data sets in this database to achieve a balance between the efficiency and the accuracy.

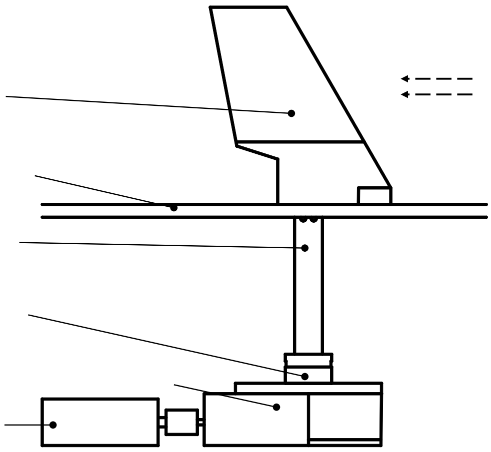
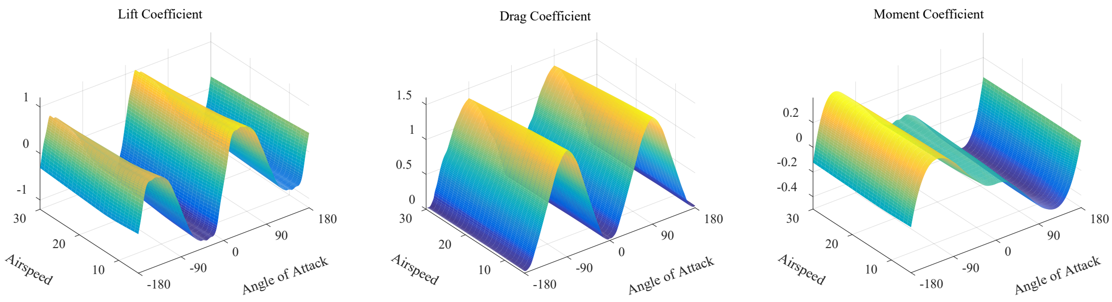


Figure 8. Schematic of aerodynamic experiments.

Figure 10 (a) shows the change of the moment (ΔC_M) caused by the flap variation when the angle of attack increases. Notably, the maximum deflection angle of the flap is $\pm 20°$. Furthermore, at high angles of attack ($\alpha > 20°$), the flap cannot generate positive moment even at the maximum deflection. Consequently, it should be carefully handled when designing a control system under windy conditions. In addition, the control system may need to restrict the vehicle to fly within a certain angle of attack range, especially during the transition, because the flap effectiveness, η, decreases quickly after $40°$ angle of attack, as illustrated in Figure 10 (b). The flap effectiveness is defined as follows:

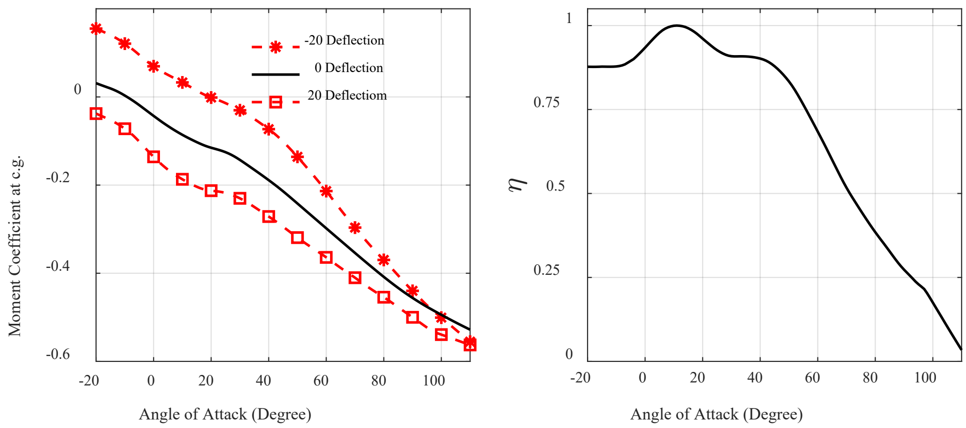
$$\eta \equiv \frac{|C_{M,-20°} - C_{M,20°}|}{\max(|C_{M,-20°} - C_{M,20°}|)}, \tag{17}$$

where $C_{M,20°}$ and $C_{M,-20°}$ are the moment coefficients with $\pm 20°$ flap deflection, respectively.



(a) (b) (c)

Figure 9. **(a)** Lift, **(b)** drag, and **(c)** moment coefficients for 0 flap deflection.



(a) (b)

Figure 10. Variation of **(a)** moment coefficient with flap deflection and **(b)** flap effectiveness with the angle of attack.

4. Real-Time Hardware-in-the-Loop Simulation

The tail-sitter vehicle is a complicated nonlinear system, and wind disturbance while hovering becomes an inevitable issue due to the tail-sitter's specific flight configuration. The control system of the tail-sitter is required to respond quickly and robustly to large disturbances and to have good performance in all flight phases. Given these requirements, a real-time HIL simulation environment is a reliable approach to assess the stability of the control system, the error characteristics and speed of response of the tail-sitter vehicle. In this section, the development of the real-time HIL simulation environment and the implementation of this HIL testbed are demonstrated.

4.1 Hardware-in-the-Loop Architecture

The HIL simulation environment is built in two independent computing units, as shown in Figure 11: a desktop computer, serving as the simulator runs the Simulink [22] real-time dynamics, and the FCU composed of Pixhawk and Odroid XU4 runs the control system.

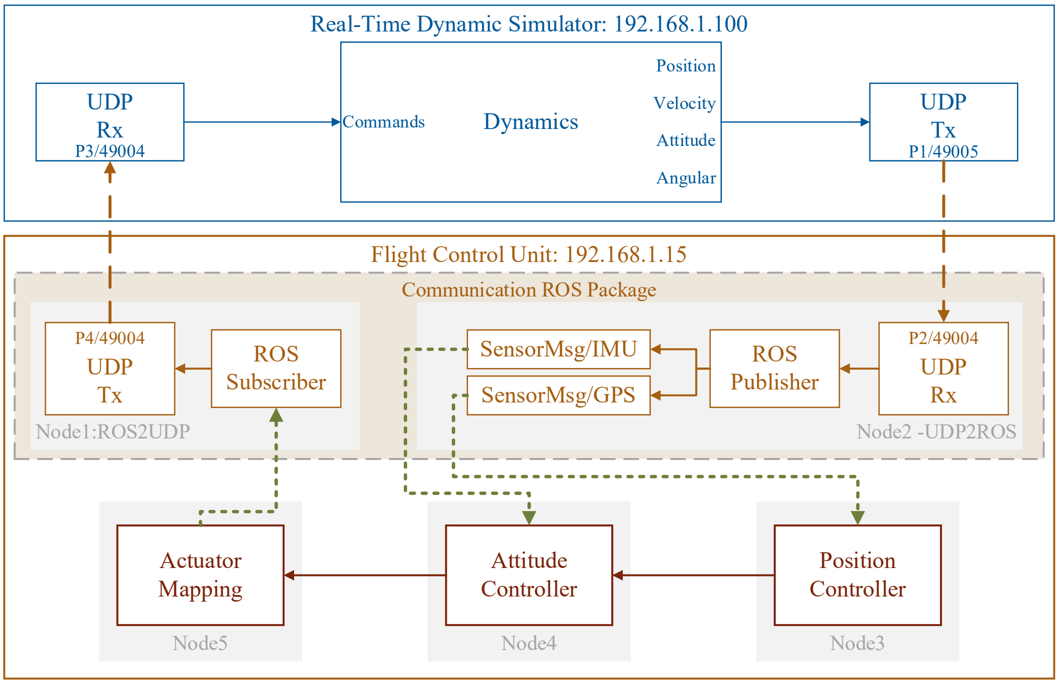


Figure 11. Hardware-in-the-loop simulation environment.

In this HIL architecture, the dynamics and motion of the vehicle are predicted by simulator, while the actual hardware of the FCU is embedded. The real-time dynamic simulator, shown as the upper component in Figure 11, predicted the tail-sitter's aerodynamics and motion. Relative information sends to the autopilot hardware to calculate the control commands. The control commands then feed back to the simulator. There are two aspects to be solved to achieve this HIL architecture: the real-time dynamic simulator and the communication between the simulator and FCU. The details of these problems will be discussed in following subsections.

4.2 Real-Time Dynamic Simulator

The dynamic simulation was developed in Simulink, and the real-time application was developed based on the Simulink Desktop Real-Time Kernel, which supports real-time performance at sampling rates up to 20 kHz. The workflow of the vehicle's dynamic simulation is shown in Figure 12. The control commands, which in this study are the control surface deflections and rotation speed of the propellers, are sent from the controller. The thrust and torque generated by the two propellers are estimated from the rotation speeds and current environmental conditions. The induced flow speed is then calculated with the model derived in Section 3 as well as the effective angle of attack (AOA), sideslip angle and airspeed. The overall aerodynamic forces and moments can be estimated from the database. The contribution of the propellers, aerodynamics and gravity are summed up and fed to the 6-DOF equations of motion (EOM) to integrate the states of the vehicle. During flight, aircraft are exposed to various wind conditions of changing magnitudes and directions. The environment condition is updated with the aircraft's location, velocity and orientation for the next step of the loop. The initial condition was declared in EOM.

As shown in Figure 13, the simulator used Aerospace Blockset to build the aerodynamics, equations of motion and environment models. The aerodynamic database and the thrust database were preloaded in the look-up table format with measured and interpolated data to improve the simulator's efficiency and accuracy. The Environment Models subsystem in Simulink introduces a prevailing wind model, a gust wind model, and a wind shear model. The prevailing wind is described by wind speed and wind direction in the World frame. The gust wind model and the wind shear model are implemented with Military Specification MIL-F-8785C. The resultant mean wind speed is transformed to the Body frame based on the vehicle's current attitude. The attitude described in quaternion is transformed using the direct cosine matrix (DCM) from body frame to earth frame

(Output port 4 in Figure 13). The Simulink model in Figure 13 is ready for real-time execution, and it is compiled to C-MEX functions. The model is synchronized with a real-time clock using Simulink Desktop Real-Time I/O blocks which are discussed as follows.

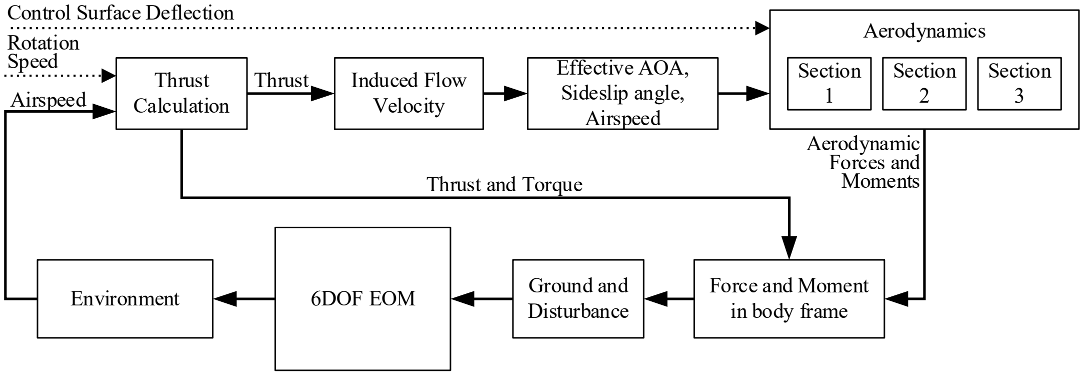


Figure 12. Flowchart of dynamic simulation.

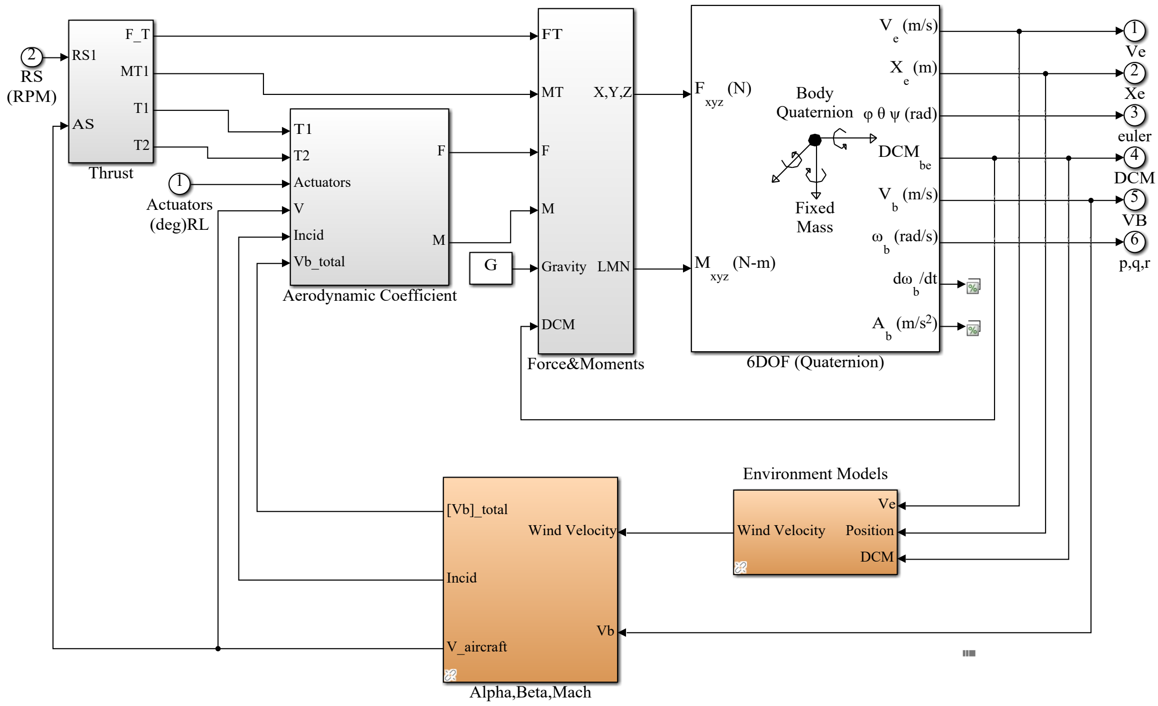


Figure 13. Real-time aerodynamic model in Simulink simulator.

4.3 Flight Control Unit (FCU)

The flight controller receives and processes the data from various sensors and calculates the appropriate control commands to achieve the desired flight mission. Pixhawk is a commonly used node-based platform, conducting flight control via position control, attitude control, and an actuator mixer. The messages such as statuses of the FCU, measured data from sensors, etc., were packed into MAVLink Protocol V2.0 and swapped over the serial port. The Pixhawk introduces three serial ports: TELEM1, TELEM2 and SERIAL 4/5. In this study, TELEM1 is connected to a 915 MHz telemetry transmitter with a baud rate of 57600 and TELEM2 is connected to the companion computer Odroid XU4 with a baud rate of 921600. The ROS package MAVROS, the MAVLink extendable communication node for ROS, is launched on the companion computer when the system starts up to pack and unpack the MAVLink protocol messages.

The flight control of the vehicle was developed with a cascade PID controller structure because the objective of this study is to develop the HIL environment rather than advanced control strategies. The control strategies of PX4 were adopted. The control system is composed of three nodes: 1) a position controller (Node 3 shown in Figure 11), 2) an attitude controller (Node 4) and 3) an actuator map (Node 5). The general control structure is presented in Figure 14-(a). Each flight phase - hover, transition, and cruise - are adopts a similar control structure. In the position controller, a commanded acceleration is calculated from the feedback of the position and velocity errors with loops C_4 and C_3 of position controller, respectively, as shown in Figure 14-(b). Further calculation is proceeded for the desired attitude of the vehicle following the desired acceleration. The attitude controller then calculates the control forces and moments from the PID feedback of the attitude and angular velocity (Figure 14-(c) attitude controller, loops C_2 and C_1). Finally, the conversion of the control forces and moments to actuator commands (the rotation speeds of the two propellers and elevon deflection angles) is accomplished by using The ROS Node5 Actuator Mapping, as illustrated in Figure 11. The rationale for the separation of the attitude control and position control is to allow the IMU and GPS data streams to have different updating rates.

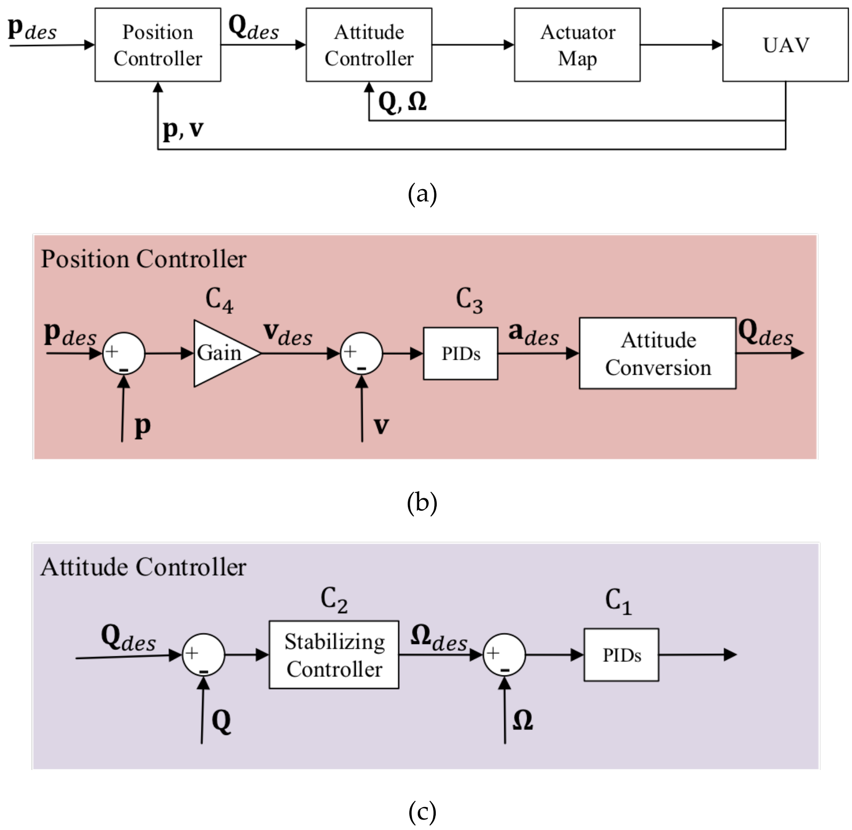


Figure 14. System block diagrams of (a) FCU's general control structure, (b) position controller and (c) attitude controller.

For the validation flight tests that will be discussed in Section 5, the actually engaged control loops are as follows:

(1). In the hover tests, all three controllers are engaged, including loops C_1 to C_4 and the actuator map;

(2). In the transition tests, the attitude controller and actuator map are engaged, including loops C_1 and C_2;

(3). In the cruise tests, the attitude controller and actuator map were engaged, including loops C_1 and C_2.

4.4 Data Communication between Simulator and FCU

The simulator and the flight controller exchange sensor data and control commands, as shown in Figure 11. As mentioned, the simulator runs in the Windows environment while the FCU runs ROS in the Linux environment. In this HIL simulation platform, three environments are included: 1) MatLab/Simulink based simulator, 2) Nuttx-based flight controller and 3) ROS based companion computer. Different message types, including both vehicle states and control commands, are required for these three environments. Several approaches can be used to establish communication between these units and to connect them to the same local network. The most straightforward way is to apply the MatLab/Simulink ROS Support Toolbox to publish and subscribe to the sensor data, and the control commands directly. However, the Simulink real-time kernel is not supported in this toolbox. In this study, the real-time simulation environment is a prerequisite. Another popular method in the HIL simulation is to utilize the I/O devices of the computers, such as the controller area network bus, RS232 serial link, and user-defined protocol (UDP) [14, 16]. The MAVLink protocol, which is used extensively in Nuttx-based flight controller, is based on serial communication. Notably, the length of the message in MAVLink protocol changes according to the message type. Although Simulink Desktop Real-Time supports the serial port driver, it cannot declare variable length messages in the Real-Time kennel. The most common protocol in ROS is based on TCP/IP sockets. Another transport layer in ROS is UDPROS, which uses standard UDP datagram packets. In this situation, UDP protocol is adopted to set up communication between the simulator and flight controller using two IP addresses 192.168.1.100 and 192.168.1.15, respectively. As shown in Figure 11, two access ports are used for Simulink (P1 and P3) and two are used for the flight controller (P2 and P4). P1 transmits the dynamic data packets to P2, and P4 returns the control commands to P3 (Simulink). The dynamic data packets include information on the position, velocity, orientation and angular velocity of the vehicle, whilst the control command data packets include data on the rotation speed of the two propellers and the deflection angles of the two control surfaces.

The data reading and writing in the real-time simulator was implemented with Packet Input and Packet Output block. Packet Output installed the UDP protocol with the setting P1, and the Packet Input was set with P3. The control commands are 4-by-1 vectors, and the states sent out are 13-by-1 vectors with the orientation described in quaternion form. The data package patterns are shown in Table 1. The solver is set to be fixed-step with a time step of 0.005 seconds.

Table 1. UDP data package pattern.

Control commands data package pattern												
RS_L			RS_R			$Flap_L$			$Flap_R$			
States data package pattern												
Position			Velocity			Attitude				Angular		
p_x	p_y	p_z	v_x	v_y	v_z	q_0	q_1	q_2	q_3	ω_x	ω_y	ω_z

In this paper, an additional separate ROS package was developed in C++ to handle the UDP data exchange. Two nodes, Node1 (ROS2UDP) and Node2 (UDP2ROS), were developed in this package to convert the data between UDP data and ROS messages (Figure 11). The dynamic data packets from Simulink are published as IMU and GPS message types, and the publisher (Node2) in this communication package publishes the simulated data at the actual sensor update rates of 200 Hz for IMU and 5 Hz for GPS. These data communication features minimise the difference between the flight control systems in the HIL environment and in the real flight to ensure the reliability of the flight controller and enable quick transitions from simulations to real flight experiments.

5. Simulation and Flight Results

In this section, the proposed HIL testbed is compared and validated with flight tests for the following cases: (i) hover flight, (ii) transition flight, including both forward and back transition, (iii) level flight and (iv) environment and wind disturbance.

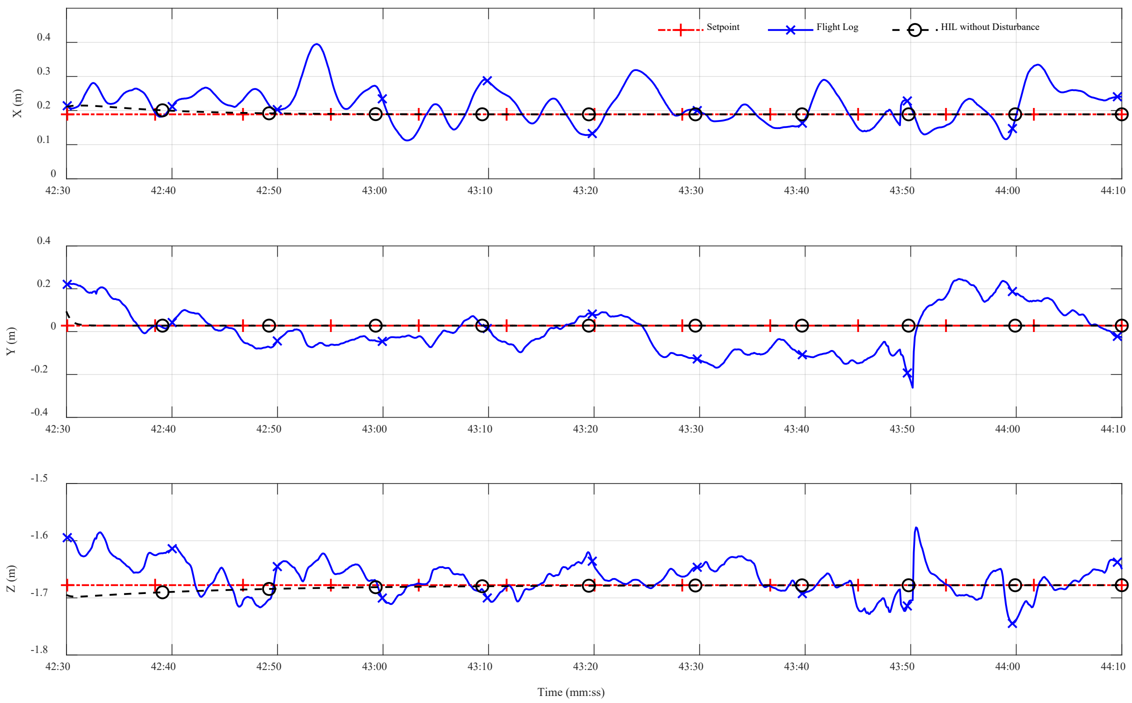
The hover flight tests were first conducted indoors to eliminate complex environmental influences. The indoor flight tests were conducted in the aviation laboratory of The Hong Kong Polytechnic University, Hong Kong and the outdoor flight tests were conducted at the Southern University of Science and Technology in Nanshan District, Shenzhen, China. The control algorithm used in the flight tests is presented in Section 4.3.

5.1 Basic Hover Flight Test

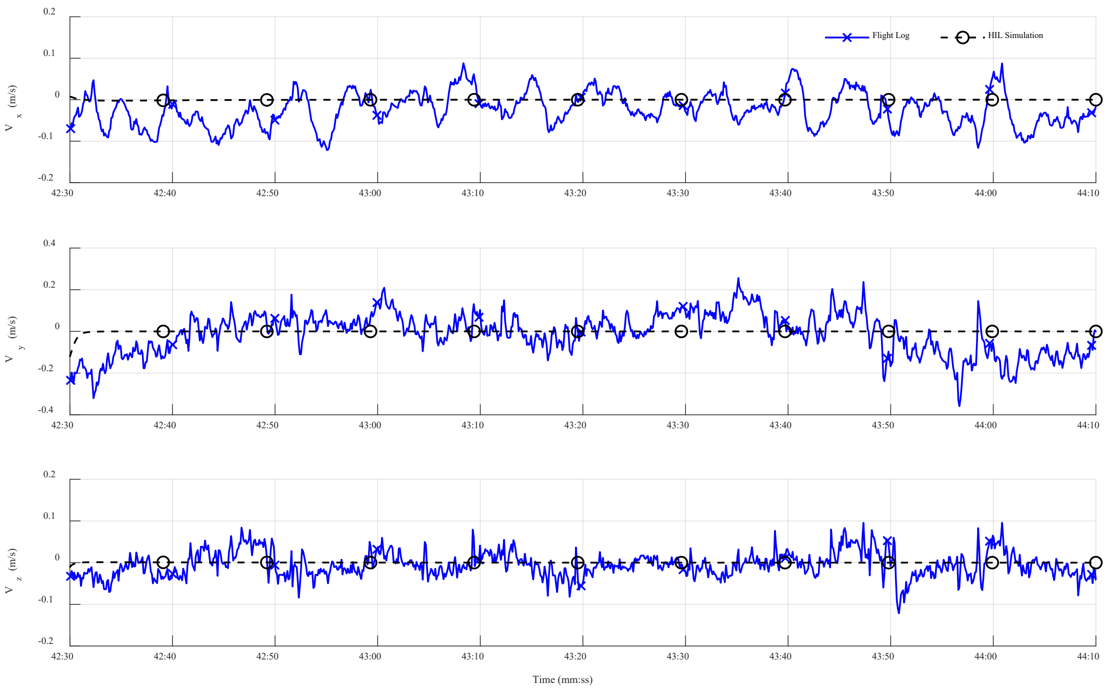
The hover phase is a basic element of the tail-sitter's complete flight scenario. Ensuring stability and control performance for hovering flight is necessary before any further flight tests can be conducted of the tail-sitter.

Given the concerns regarding limited information and complex weather conditions, the indoor hovering flight tests were conducted in the aviation laboratory to preliminary evaluate the aerodynamics and the vehicle dynamic model. During the flight test, the position controller and attitude controller were engaged. The position of the vehicle was measured using a VICON motion capture system, and the data were sent to the flight controller through the ROS network. After taking off, the vehicle was commanded to hold position until other position commands were received. A 140-second segment of hover flight data was retrieved for comparison with the HIL simulation results to exclude the ground effect and unmeasured disturbance during take-off and landing. The first states of the flight segments were set to be the initial state of the HIL simulation. The results of the flight segment, and the HIL simulation are presented in Figure 15. In Figure 15 (a), the HIL simulation results show that the control system can command the vehicle to hover at set points smoothly. As shown in Figure 15 (c), in the HIL simulation results, the vehicle converges around 91° rather than 90° to hold position, which matches the flight data.

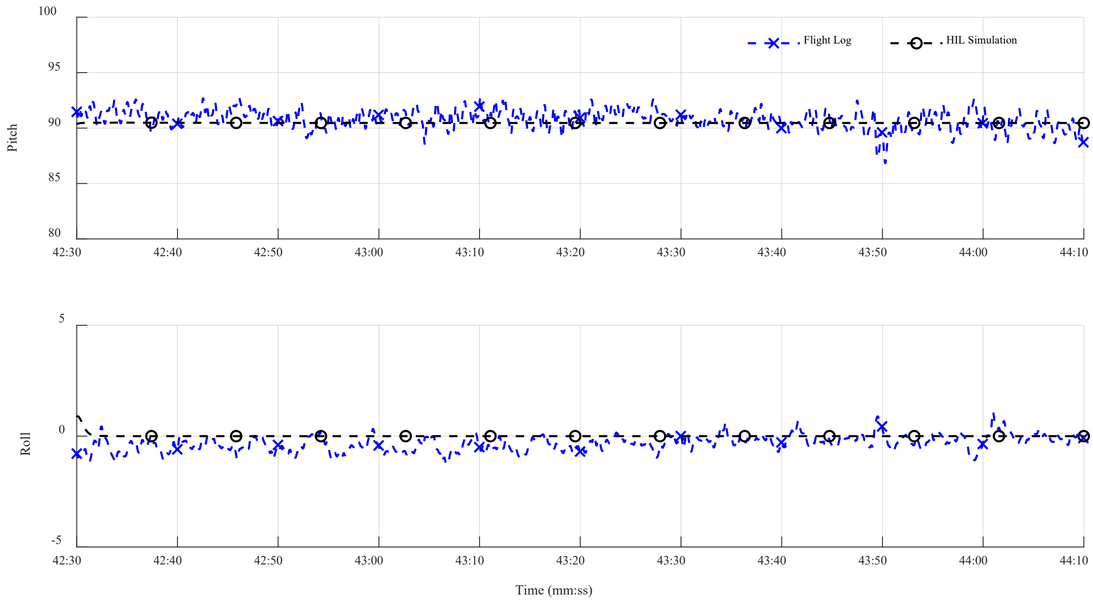
There are slight oscillations in position flight data, bounded within 0.1 m range. The oscillations are supposed to be caused by disturbance, which results in changes in the speed and the attitude of the vehicle (shown in Figure 15 (b) and (c), respectively). The HIL simulation can reproduce similar oscillations when introducing the disturbing force, as shown in Figure 16. The disturbing forces are introduced in 'Ground and Disturbance' subsystem of the real-time dynamic simulator, as shown in Figure 12. The disturbing forces were modelled as white noise with a maximum disturbing force of around 10% of the weight of the vehicle. The HIL simulation system can be further applied to test and discuss the disturbance bound while developing the control system.



(a)



(b)



(c)

Figure 15. -HIL results compared with the corresponding hover flight test results for (a) position, (b) velocity and (c) attitude.

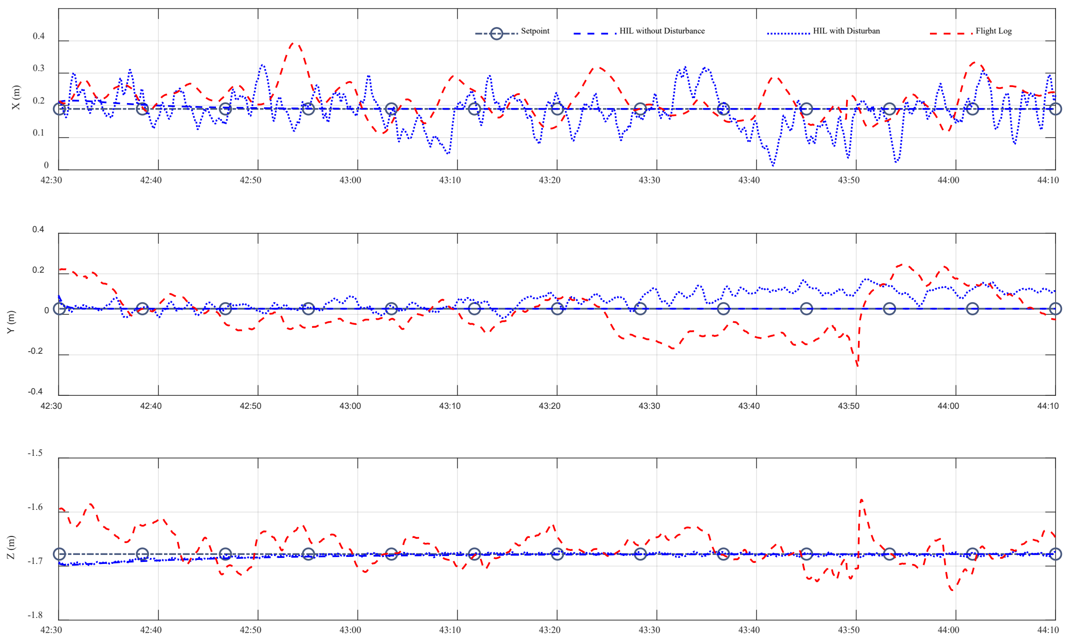
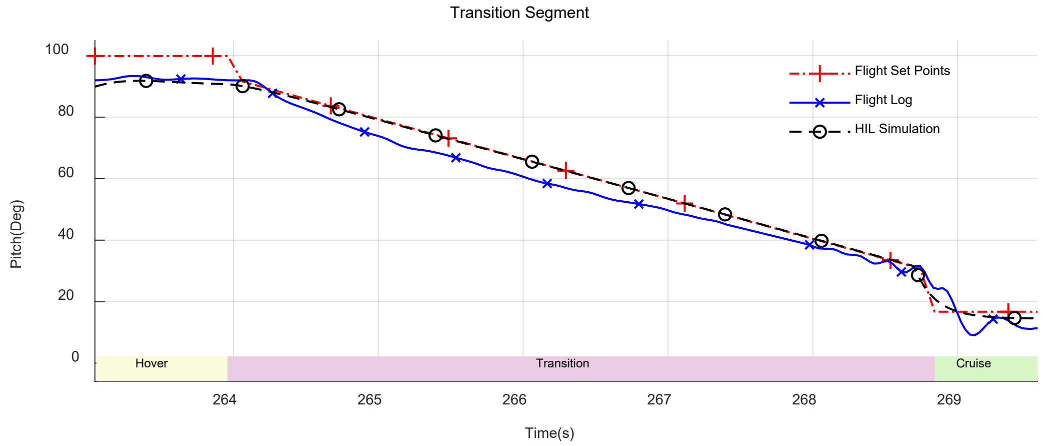


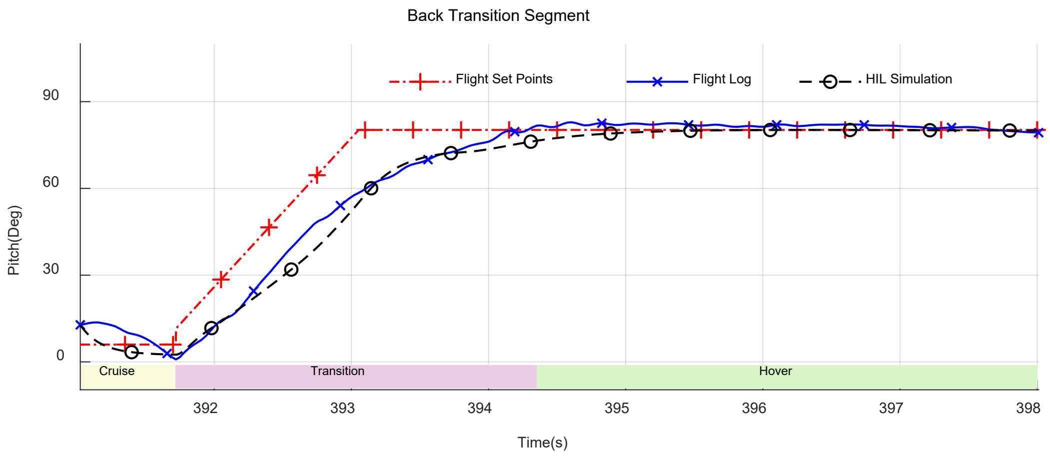
Figure 16. HIL results with disturbing forces compared with flight data.

5.2 Transition Flight test

The transition phase is a challenging task in a regular flight of the tail-sitter. The control algorithm is described in Section 4.3. The comparison of the transition flight in the HIL simulation and in the flight tests is shown in Figure 17. During the forward transition, the time-dependent set points are generated. As shown in Figure 17 (a), in both the HIL simulation and the experiment, the vehicle followed the control commands to pitch down and finally cruise with a pitch angle of around 15°. Similarly, in Figure 17 (b), the HIL simulation results of the back transition matches the flight results quite well. It can be noticed that the performance and response of the back transition controller were not as fast as the forward transition one, which may be either caused by the limited size of the control surface or insufficient control algorithm. However, once again, the HIL simulation provide an accurate evaluation of the control system.



(a)



(b)

Figure 17. HIL results compared with the corresponding transition flight tests of (a) forward transition and (b) back transition.

5.3 Level Flight Test

During the cruise phase of the developed tail-sitter in this study, the vehicle became a typical and conventional tailless aircraft. As discussed in Section I, the HIL simulation technologies for such fixed-wing aircraft have been developed over many years. In addition, the control methods and algorithms for fixed-wing vehicles, from stability augmentation to navigation control, are a traditional research area. However, because the aim of this study is to develop an HIL simulation testbed. The control performance of the fixed-wing vehicle is beyond the scope of this paper. The level flight data were collected to validate the accuracy of aerodynamic prediction in the HIL simulation system. Thus, only the attitude stability augmentation and airspeed controller were engaged in level flight tests. The targets of the attitude were commanded by the pilot through the remote control. During the corresponding HIL simulation, the pilot's commands were loaded and published as attitude set points. The cruise airspeed was set at 15 m/s.

The level flight tests were conducted on the campus of the Southern University of Science and Technology, Shenzhen, China. Due to limited flying space, the vehicle was manually controlled to fly in a rectangular path. An approximately 20-second segment with a relatively straight flight path was retrieved out of the flight data for comparison with the HIL simulation. The results are presented in Figure 18. Due to the low airspeed, the vehicle cruised with a relatively high pitch angle (around 18°). Similar to the results in the hover flight tests, the flight data showed slight oscillations on the pitch, which are considered to be caused by disturbance. The airspeed of a fixed-wing aircraft is always varying due to the outdoor gusty wind environment. Nevertheless, the airspeed in flight data reach an average value of ~ 15 m/s and can still be considered converged to the preset number, under this situation.

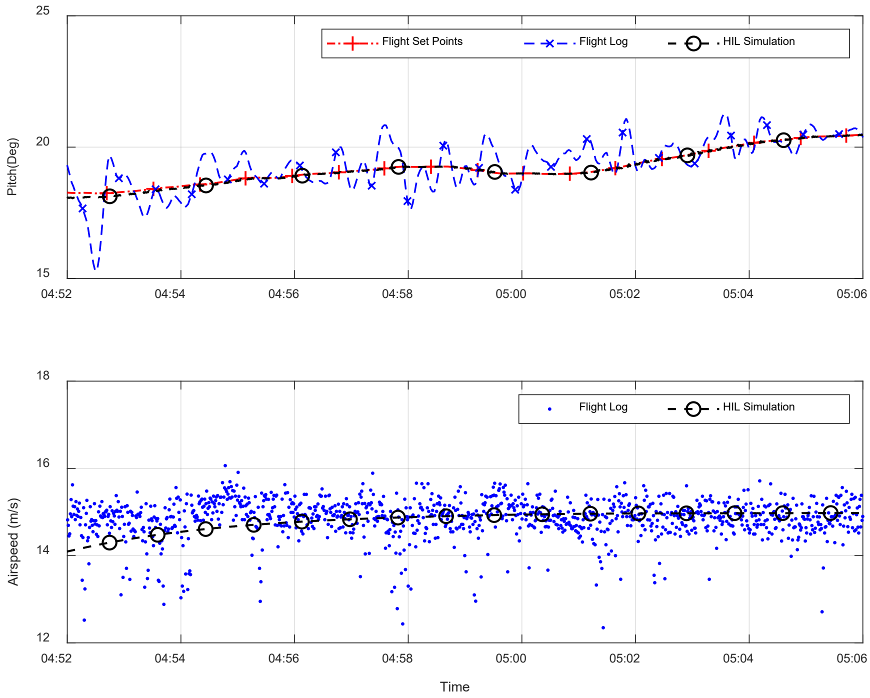


Figure 18. Comparison of the HIL simulation and the flight test in the cruise phase.

5.4 Environment and wind disturbance

Notably, the tail-sitter vehicle was not robust to wind disturbance because of its large windward surface, especially during hovering and transition phases. A set of simulations of the vehicle hovering in the prevailing wind was conducted. The corresponding results are shown in Figure 19, in which the z-axis is set to point north, and the wind comes from the south to simulate the worst case. It is clearly seen that the vehicle cannot maintain its attitude if the wind reaches $1.9\,m/s$. The vehicle eventually lost control after 3 s. In $1.9\,m/s$ wind, the control system pushed the flaps to their limits to maintain attitude. Because the large wing area in high prevailing wind introduces a significant force along the z-axis in the body frame (see the upper right inset in Figure 19), the control forces and moments generated by the control surface, which is of limited area, are not enough to balance the vehicle. Any increased differences in the aerodynamic forces beyond the control limit between the left and right segments will increase the vehicle's roll rate continuously, yielding the large variation and later divergence of the pitch angel.

Also, it is observed that the pitch angle under the steady state increases when the prevailing wind speed increases. The flaps were controlled to adjust the aerodynamic lift and moments to maintain the hovering attitude and location, and the z-component force of the thrust also helps to balance the disturbance.

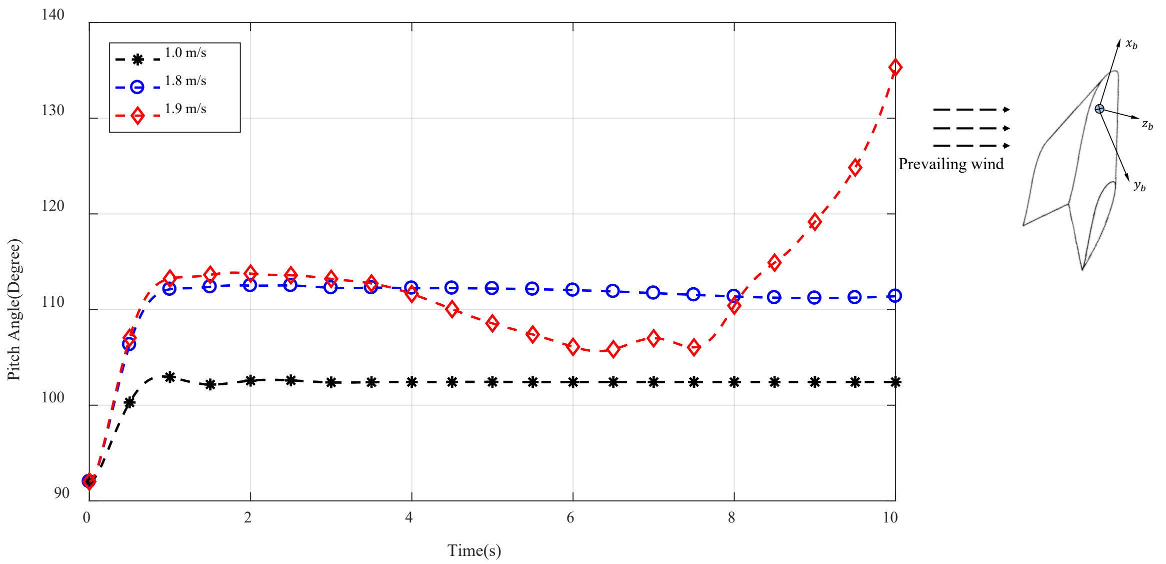


Figure 19. Simulation results of pitch angle for different wind conditions, with the wind direction setting illustrated in the top inset.

A real flight test was conducted to examining the actual flight performance in such extreme conditions. The tail-sitter vehicle continued to hover for only around 20 s before pitching up due to the increasing wind. Unable to maintain its attitude, and the vehicle crashed. Nevertheless, the flight logs still provide important data for comparison with the HIL simulation. The flight log segment that started at 19.2 s was retrieved. The flight state at 19.2 s was used as the initial state of the simulation, and the z-axis was assumed to point to the north in the World frame simulation. The simulation started with a 1 m/s southerly wind (180°). At 19.6 s, a 5.6-m/s gust of wind from the south lasting for 0.5 s was introduced. As shown in Figure 20., the flaps were controlled to increase the moment to maintain the attitude, but the pitch angle of the vehicle nevertheless diverged to 180°, and the vehicle finally crashed. The simulation is in good agreement with the flight log and successfully reproduces the divergence process. After 20.2 s, the vehicle oscillated and crashed in both the flight test and the simulation.

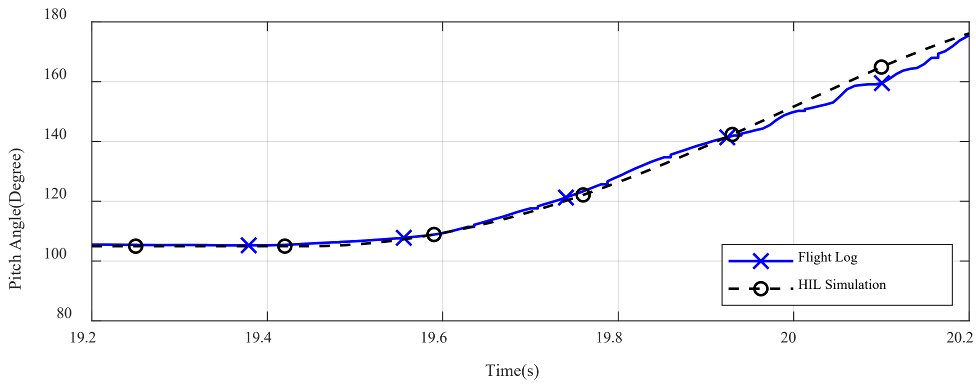


Figure 20. Attitude results from the outdoor flight test segment and HIL simulation.

In fact, the HIL simulation system is of great help and instrumental for researchers before conducting the actual flight tests. Once the limitation of the tail-sitter vehicle against the cross-direction wind is noticed by the HIL simulation, and the real flight tests should be conducted with great care.

6. Conclusions

With the increasing interest in and rapid development of the tail-sitter vehicle, an appropriate testbed is necessary. In this paper, we presented the development and tests of an HIL testbed for a duo-rotors tail-sitter that was supported by the open source FCU. With the component breakdown approach, the aerodynamics of the vehicle could be predicted appropriately even in the large angle of attack. The resulting mathematical model was then providing a real-time application for this HIL testbed, which also included the environmental model. It can be further applied to the discussion of aspects of the environmental influence of the vehicle, such as wind field estimation and wind disturbance control.

To embed the FCU in the HIL environment to mimic the real flight, an additional ROS communication package was developed to convert the UDP data packets to ROS sensor messages, and these messages were published at the same updating rate as that of real sensors. This approach can facilitate researchers in revising the control algorithm and simulating real flight conditions during controller development. Flight tests were then conducted and compared with the HIL simulations, and the results demonstrated the accuracy and reliability of the current HIL system. This HIL simulation system also demonstrated its capability for testing and discussing the environment disturbance bound while developing the control system

In a future study, the HIL system developed in this study will be applied to the control system design. A control system for hovering in windy conditions and transition flight will be developed. The effects on aerodynamics are also under study to improve the accuracy of the HIL simulation.

Acknowledgments: This work is sponsored by Innovation and Technology Commission, Hong Kong under Contract No. ITS/334/15FP.

References

[1] B. Li, Y. Jiang, J. Sun, L. Cai, C.-Y. Wen, Development and Testing of a Two-UAV Communication Relay System, Sensors, 16 (2016) 1696.

[2] J. Sun, B. Li, Y. Jiang, C.-Y. Wen, A Camera-Based Target Detection and Positioning UAV System for Search and Rescue (SAR) Purposes, Sensors, 16 (2016) 1778.

[3] R.H. Stone, Aerodynamic modeling of the wing-propeller interaction for a tail-sitter unmanned air vehicle, J. Aircr., 45 (2008) 198-210.

[4] R.H. Stone, P. Anderson, C. Hutchison, A. Tsai, P. Gibbens, K. Wong, Flight testing of the t-wing tail-sitter unmanned air vehicle, J. Aircr., 45 (2008) 673-685.

[5] D. Kubo, S. Suzuki, Tail-sitter vertical takeoff and landing unmanned aerial vehicle: transitional flight analysis, J. Aircr., 45 (2008) 292-297.

[6] X. Lyu, H. Gu, Y. Wang, Z. Li, S. Shen, F. Zhang, Design and implementation of a quadrotor tail-sitter VTOL UAV, 2017 IEEE International Conference on Robotics and Automation (ICRA), , IEEE, 2017, pp. 3924-3930.

[7] S. Verling, T. Stastny, G. Bättig, K. Alexis, R. Siegwart, Model-based transition optimization for a VTOL tailsitter, 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 3939-3944.

[8] R. Bapst, R. Ritz, L. Meier, M. Pollefeys, Design and Implementation of an Unmanned Tail-sitter, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Congress Center, Hamburg, 2015, pp. 1885-1890.

[9] B. Li, W. Zhou, J. Sun, C.-Y. Wen, C.-K. Chen, Model Predictive Control for Path Tracking of a VTOL Tailsitter UAV in an HIL Simulation Environment, 2018 AIAA Modeling and Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, 2018.

[10] G. Cai, B.M. Chen, T.H. Lee, M. Dong, Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters, Mechatronics, 19 (2009) 1057-1066.

[11] J. Sun, B. Li, L. Shen, C.-K. Chen, C.-Y. Wen, Dynamic Modeling and Hardware-In-Loop Simulation for a Tail-Sitter Unmanned Aerial Vehicle in Hovering Flight, AIAA Modeling and Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, 2017.

[12] Q. He, J. Liu, An observer for a velocity-sensorless VTOL aircraft with time-varying measurement delay, Int. J. Syst. Sci., 47 (2016) 652-661.

[13] M.A. Dehghani, M.B. Menhaj, H. Ghaderi, A hardware in the loop simulation testbed for vision-based leader-follower formation flight, Mechatronics, 47 (2017) 223-232.

[14] M. Abdelkader, Simulink-Xplane10 Communication Via UDP, Mohamed Abdelkader, 2014.

[15] L.R. Ribeiro, N.M.F. Oliveira, UAV autopilot controllers test platform using Matlab/Simulink and X-Plane, 40th ASEE/IEEE Frontiers in Education Conference, DC, USA, 2010.

[16] E.R. Mueller, Hardware-in-the-loop simulation design for evaluation of unmanned aerial vehicle control systems, AIAA Modeling and Simulation Technologies Conference and Exhibit, Hilton Head, South Carolina, 2007, pp. 20-23.

[17] D. Jung, P. Tsiotras, Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle, AIAA Infotech at Aerospace 2007 Conference and Exhibit Rohnert Park, California, 2007, pp. 07-2763.

[18] M. Montazeri-Gh, M. Nasiri, S. Jafari, Real-time multi-rate HIL simulation platform for evaluation of a jet engine fuel controller, Simulation Modelling Practice and Theory, 19 (2011) 996-1006.

[19] L. Meier, D. Honegger, M. Pollefeys, PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms, 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Washington State Convention Center, WA, 2015, pp. 6235-6240.

[20] R. Ritz, D'Andrea, Raffaello, A Global Strategy for Tailsitter Hover Control, International Symposium on Robotics Research (ISRR) 2015, Sestri Levante, Italy, 2015.

[21] R. Howell, N. Qin, J. Edwards, N. Durrani, Wind tunnel and numerical study of a small vertical axis wind turbine, Renewable energy, 35 (2010) 412-422.

[22] Simulink, MathWorks, Inc., Natick, Massachusetts, USA, 2016.