NASA/TM–20250000954

Design, Simulation, and Flight Testing of a Multi-Purpose VTOL Flight Control System

Anthony M. Comer
OSTEM Internship Program, Langley Research Center, Hampton, Virginia

Benjamin M. Simmons and Garrett D. Asper
Langley Research Center, Hampton, Virginia

September 2025

NASA/TM–20250000954

Design, Simulation, and Flight Testing of a Multi-Purpose VTOL Flight Control System

Anthony M. Comer
OSTEM Internship Program, Langley Research Center, Hampton, Virginia

Benjamin M. Simmons and Garrett D. Asper
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

September 2025

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

A novel flight control system architecture aimed at vertical takeoff and landing (VTOL) vehicles was designed, simulated, and flight tested. A simulation model was developed using an empirically-derived aerodynamic model and mass properties for a subscale fixed-wing aircraft. The simulation framework was used to design and simulate the proposed flight control architecture through a series of piloted simulations. A simulation model of a VTOL variant of the fixed-wing vehicle was also created to demonstrate the flight control system on a transitioning VTOL aircraft. Flight tests were performed at the NASA Langley Autonomy Lab for Intelligent Flight Technology (ALIFT) and City Environment Range Testing for Autonomous Integrated Navigation (CERTAIN) facilities to validate the proposed control system. Validation flights were performed on a small multirotor vehicle and a subscale fixed-wing vehicle to assess hover and forward flight performance. A Pixhawk® flight computer was used along with the MathWorks® UAV Toolbox to provide rapid control law integration and testing. The presented results aim to validate the feasibility of the proposed control system architecture and simulation-to-hardware process.

Contents

1 Introduction

Over the past few years, ongoing research at NASA Langley Research Center (LaRC) has made use of the MathWorks® UAV Toolbox to support rapid, inexpensive flight testing of custom flight control laws on Pixhawk®-based flight controllers [1, 2]. In recent studies, this process has been used on the Inexpensive Multirotor Platform for Advanced Controls Testing (IMPACT) flight hardware [2]. To expand this research, the Subscale Electric Aircraft for ResearCH (SEARCH) flight hardware was proposed as a subscale fixed-wing aircraft to test novel control law development. Subscale vehicles provide simple, inexpensive testbeds for flight control system development. With the recent surge in vertical takeoff and landing (VTOL) vehicles, there is a need for flight control system development to bridge the gap from rotor-borne to wing-borne flight regimes. To support this work, a simulation model was developed for the fixed-wing SEARCH vehicle and a notional vectored-thrust VTOL variant of the SEARCH vehicle to test the VTOL control laws.

This report discusses the design, simulation, and flight testing of a novel flight control system aimed at multirotor, VTOL, and fixed-wing vehicle applications. This work was completed during a 10-week Summer 2024 internship project in the Flight Dynamics Branch at NASA LaRC. The flight control system employs concepts from the Total Energy Control System (TECS) [3–6] developed by Lambregts in the 1980's and flight tested by NASA researchers [7]. This control architecture also employs concepts from prior simulation work by Comer and Chakraborty, where they extend TECS concepts to a lift-plus-cruise (LPC) [8] and tilt-wing (TW) [9] vehicle. Flight testing was performed on the IMPACT and SEARCH vehicles to test vertical flight and forward flight control laws of the control system. A VTOL platform was not available for flight testing during the internship period. Therefore, simulation results for a VTOL variant of the SEARCH vehicle are presented.

The remainder of this report is organized as follows. Section 2 discusses the aircraft dynamics model common to the SEARCH fixed-wing and SEARCH VTOL simulation models. Section 3 discusses the simulation platform developed to support the control law design and testing of these vehicles. Section 4 provides details on the proposed flight control system architecture. Section 5 describes the SEARCH VTOL aircraft and highlights some simulation results. Section 6 provides insight to the hardware development process used to enable flight testing. Section 7 discusses the SEARCH fixed-wing vehicle, including simulation and flight test results. Section 8 shows the extension and flight testing of the flight control system on the IMPACT multirotor vehicle. Section 9 provides concluding remarks and future research avenues.

2 Aircraft Dynamics Model

The aircraft dynamics model was implemented into a MATLAB®/Simulink® simulation environment to provide a generalized platform for multiple vehicles. This section presents the aircraft equations of motion, as well as the aero-propulsive, actuator, and sensor models.

2.1 Equations of Motion

The vehicle equations of motion follow a standard six degree-of-freedom (6DOF) rigid-body form about the aircraft center of gravity [10, 11]. The states variables are the aircraft body-fixed velocities (u, v, w), angular rates (p, q, r), Euler angles (ϕ, θ, ψ), and Cartesian coordinates (x, y, z). The translational dynamics are modeled as

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \frac{1}{m} \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} + \begin{Bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{Bmatrix} \tag{1}$$

where F_x, F_y, F_z represent the body-fixed external aerodynamic, propulsion, gravity, and ground contact forces acting on the vehicle. The rotational dynamics are modeled under the assumption of lateral symmetry ($I_{yz} = I_{xy} = 0$) and neglecting propulsion gyroscopic effects as:

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix}^{-1} \begin{Bmatrix} M_x + (I_{yy} - I_{zz})qr + I_{xz}pq \\ M_y + (I_{zz} - I_{xx})pr + I_{xz}(r^2 - p^2) \\ M_z + (I_{xx} - I_{yy})pq - I_{xz}qr \end{Bmatrix} \tag{2}$$

Here, M_x, M_y, M_z are the body-fixed external aerodynamic, propulsion, and ground contact moments acting on the vehicle and $I_{xx}, I_{yy}, I_{zz}, I_{xz}$ are the aircraft inertia tensor elements. The rotational kinematics are formed as:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{3}$$

The translational kinematics are modeled from:

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{4}$$

where $c_{(\cdot)}$ and $s_{(\cdot)}$ represent the cosine and sine functions, respectively.

2.2 Aero-Propulsive Model

For this report, the aero-propulsive model for the SEARCH fixed-wing aircraft was primarily adapted from Refs. [12, 13], which developed an aero-propulsive model for a similar aircraft using flight test data. The model consists of a nonlinear multivariate polynomial equation for aircraft nondimensional force and moment coefficients in the body axes

$$C_X = \frac{X}{\bar{q}S}, \quad C_Y = \frac{Y}{\bar{q}S}, \quad C_Z = \frac{Z}{\bar{q}S}, \quad C_l = \frac{L}{\bar{q}Sb}, \quad C_m = \frac{M}{\bar{q}S\bar{c}}, \quad C_n = \frac{N}{\bar{q}Sb} \tag{5}$$

which are normalized by dynamic pressure $\bar{q} = \frac{1}{2}\rho V^2$ and aircraft reference geometry. Here, ρ is the air density, V is the freestream velocity, S is the wing reference area, $\bar{c}$ is the mean aerodynamic chord, and b is the wingspan. The baseline model structure for each force and moment coefficient was

$$C_X = C_{X_\alpha}\alpha + C_{X_{P_{\mathcal{J}_c}}}\mathcal{J}_c + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\beta^2}}\beta^2 + C_{X_{P_{\mathcal{J}_c^2}}}\mathcal{J}_c^2 + C_{X_{A_o}} + C_{X_{P_o}} \tag{6}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_p}\hat{p} + C_{Y_r}\hat{r} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r + C_{Y_o} \tag{7}$$

$$C_Z = C_{Z_\alpha}\alpha + C_{Z_q}\hat{q} + C_{Z_o} \tag{8}$$

$$C_l = C_{l_\beta}\beta + C_{l_p}\hat{p} + C_{l_r}\hat{r} + C_{l_{\delta_a}}\delta_a + C_{l_o} \tag{9}$$

$$C_m = C_{m_\alpha}\alpha + C_{m_q}\hat{q} + C_{m_{\delta_e}}\delta_e + C_{m_{\alpha^2}}\alpha^2 + C_{m_o} \tag{10}$$

$$C_n = C_{n_\beta}\beta + C_{n_r}\hat{r} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r + C_{n_o} \tag{11}$$

where α is the angle of attack in radians; β is the angle of sideslip in radians; $\hat{p} = \frac{pb}{2V}$, $\hat{q} = \frac{q\bar{c}}{2V}$, and $\hat{r} = \frac{rb}{2V}$ are the dimensionless angular rates; and δ_e, δ_a, and δ_r are the elevator, aileron, and rudder control surface deflection angles, respectively, in radians. Equation (6) is from the second modeling approach described in Ref. [12] that developed a separate propulsion and airframe model from flight data. The centered inverse advance ratio $\mathscr{J}_c$ is used to model the propulsive effects. Equations (7)-(11) are from the standard modeling approach described in Ref. [13]. The dimensionless angle of attack rate terms included in Refs. [12, 13] that model unsteady effects were neglected for this study.

To create a SEARCH VTOL simulation model, a simple high-incidence-angle propeller model was created assuming that there were no aero-propulsive interaction effects (this is an unrealistic assumption because aero-propulsive interaction effects for VTOL aircraft are significant, but was deemed appropriate for the preliminary SEARCH VTOL simulation study performed in this report). Propeller performance data from Refs. [14, 15] were used to create nonlinear polynomial models describing the thrust and torque coefficient variation as a function of advance ratio $J = \frac{V}{nD}$. To approximate high-incidence-angle effects, the component of advance ratio normal to the freestream velocity vector ($J_x = J \cos i_p$, where i_p is the propeller incidence angle, which is defined to be zero in axial airflow) was used in the polynomial equations, which has been shown empirically to be a reasonable assumption for a large range of incidence angles about the axial airflow condition [16]. The propeller force and moment components were calculated in the body frame about the aircraft center of gravity and were then superimposed with the airframe aerodynamic forces and moments.

2.3 Actuator and Sensor Dynamics

Control surface servo-actuators are modeled with second-order dynamics of the following form:

$$G_{act}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{12}$$

with $\omega_n = 75$ rad/s and $\zeta = 0.9$ selected to model the response characteristics. These dynamics were selected heuristically, with the natural frequency following actuator metrics reported in [17]. Rate limits of $\pm 300°$/sec are imposed on the servo-actuator model following E-flite manufacturer data.

Motor dynamics are modeled along with an electronic speed controller (ESC). The ESC model takes the form of a proportional-integral (PI) controller, generating a commanded power (in Watts) to the motor as follows:

$$P_{cmd} = \left(K_P + \frac{K_I}{s}\right)(RPM_{cmd} - RPM)\ , \qquad P_{cmd} \in [0, P_{max}] \tag{13}$$

The ESC model uses PI control to track commanded RPM through torque balance. The torque balance equation of motion is modeled as:

$$\dot{\Omega} = \frac{1}{I_{prop}}(Q_{mot} - Q_{prop}) \tag{14}$$

where I_{prop} is the moment of inertia of the propeller (approximated from $I = (1/3)mr^2$), Q_{mot} is the commanded motor torque, and Q_{prop} is the torque from the propeller. When torque data is unavailable, propeller torque is modeled as a second-order polynomial $Q_{prop} = K_Q N^2$, where N is the motor rotational speed. The motor torque gain K_Q can be determined heuristically based on available data. If torque curves are available, motor torque can be input externally based on the flight conditions. The motor torque is also limited such that $Q_{mot} \in [Q_{min}, Q_{max}]$. For the vehicles simulated, the ESC model was tuned to track a second-order transfer function [see Eq. (12)]

with $\omega_n = 40$ and $\zeta = 1.0$ based on similar motor-propeller step test characteristics used in prior works [18–20].

Sensors are modeled using first order and second-order Butterworth filter transfer functions of the following forms:

$$G_{sens}(s) = \frac{1}{\tau s + 1}\ , \qquad G_{sens}(s) = \frac{\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{15}$$

These sensor models serve as both an estimation delay and a low-pass filter, and follow the form of Pixhawk® firmware internal filters [21] for the attitudes, rates, and accelerations. Attitude (ϕ, θ, ψ) and rate (p, q, r) sensors employ a second-order form with a 10 Hz cutoff frequency. Accelerations a_x, a_y, a_z use a 5 Hz cutoff frequency. Velocity and position terms use a first-order form with $\tau = 0.1$. The airspeed sensor is filtered with $\tau = 0.25$.

3 SIMulation PLatform (SIMPL)

SIMPL was developed to provide a generalized simulation framework to allow rapid development of simulation models using a modular framework. SIMPL provides the following vehicle-independent capabilities:

- **Trim Analysis:** trim of the vehicle is determined offline at user-specified flight conditions through a constrained optimization problem. This optimization problem can be modified as desired to minimize specific states, control inputs, or specific measurable quantities.

- **Model Linearization:** linearized models can be developed at trimmed conditions of the vehicle. These linearized models can subsequently be used for control law development and stability analysis of the vehicle.

- **Actuator and Sensor Modeling:** generalized actuator and sensor models are developed for simple modeling and tuning, including first- and second-order models with optional rate limits as desired. (see Sec. 2.3).

- **External Visualization:** external connections, including external joysticks, cross-network multiplayer, and external visualization using third-party simulation software are supported. SIMPL interfaces with X-Plane 11 for external visualization. Though not explicitly shown in this work, X-Plane 11 has native support of cross-network multiplayer.

3.1 Trim Problem Setup

Vehicle trim is solved offline as a constrained-parameter optimization problem, similar to methods described in prior work by Comer and Chakraborty [8, 9, 22–24]. This approach reduces the number of actuators in over-actuated configurations (more control effectors than desired degrees of freedom) to aid in trimming the aircraft. Trim is determined at user-defined airspeeds based on desired (i) altitude and (ii) flight path angle γ assuming non-turning flight. The trim solver has access to angle of attack (α) and sideslip (β) as well as vehicle-specific normalized control inputs. For example, the SEARCH vehicle trims using a combination of three normalized control inputs (lateral control lat_{cmd}, longitudinal control lon_{cmd}, directional control dir_{cmd}), RPM, and flap setting δ_f. The normalized control commands are effectively the control authority of the vehicle expressed as a unit value $[-1, +1]$, where $+1$ would be 100% positive control input on each respective axis. The optimization problem seeks to minimize the vehicle linear and angular accelerations

($\dot{X}_{res} = \dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$) using the MATLAB® gradient-based `fmincon` optimization function [25]. This process is visualized in Fig. 1 for the SEARCH VTOL configuration, showing optimizer inputs (left) with the residual outputs (right). Any scheduled actions (e.g., lift propulsor RPM shutdown, flap scheduling, etc.) are determined within the "Scheduled Controls."

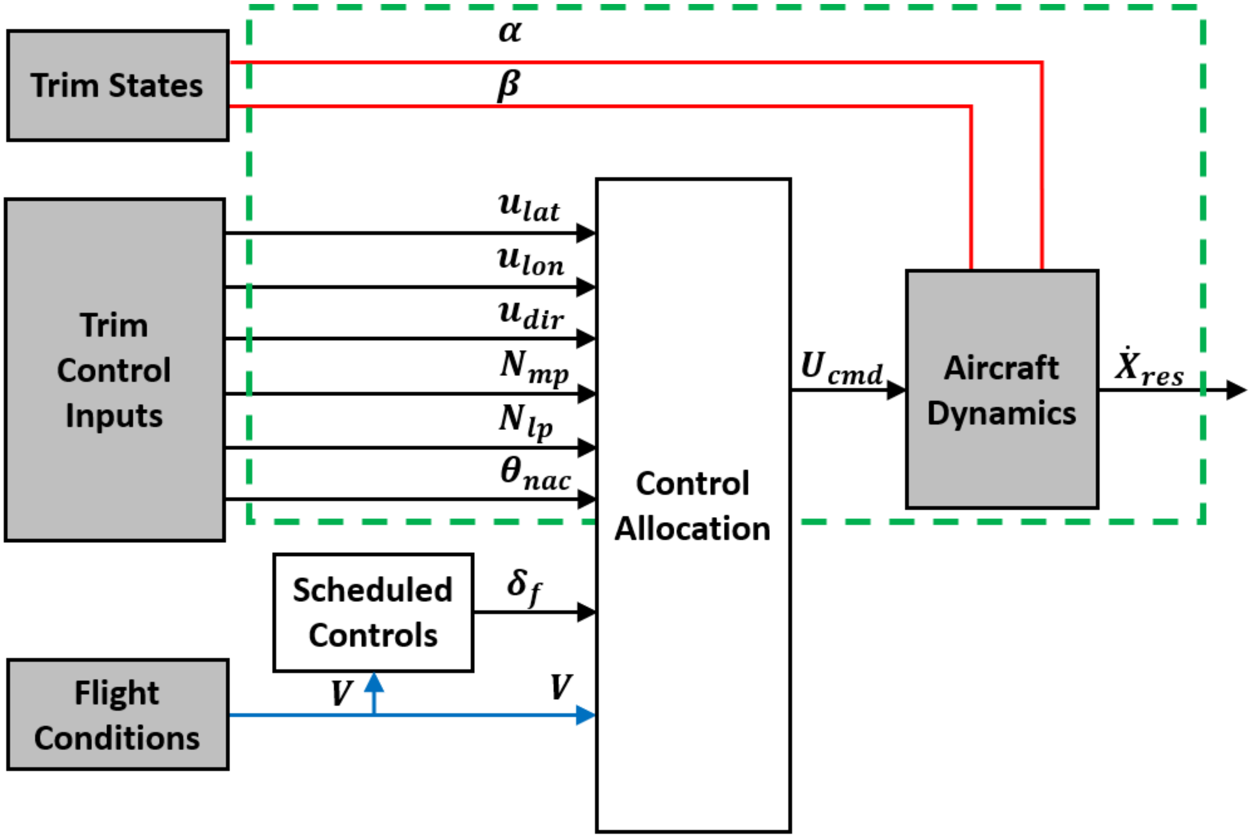


Figure 1: Visualization of the trim problem setup with inputs and outputs passing through the dashed green box.

3.2 Model Linearization

Linearization is performed at each of the user-defined trim points to determine the linearized vehicle dynamics of the form $\dot{X} = AX + BU$ and linearized vehicle outputs of the form $Y = CX + DU$. The linearization process employs central differencing to determine A, B, C, D matrices as follows:

$$A = \frac{f(X + \Delta x, U_0) - f(X - \Delta x, U_0)}{2\Delta x}, \qquad B = \frac{f(X_0, U + \Delta u) - f(X_0, U - \Delta u)}{2\Delta u} \tag{16}$$

$$C = \frac{g(X + \Delta x, U_0) - g(X - \Delta x, U_0)}{2\Delta x}, \qquad D = \frac{g(X_0, U + \Delta u) - g(X_0, U - \Delta u)}{2\Delta u} \tag{17}$$

where $f(X,U)$ represents the vehicle equations of motion and $g(X,U)$ represents the vehicle output equations. The individual vehicle states X and control effectors U are perturbed above and below the trim condition by a specified incremental term $\Delta x/\Delta u$. Vehicle dynamic stability analysis can be performed following linearization and is used to aid in flight control system development and optimization.

3.3 External Visualization

The SIMPL platform connects with X-Plane 11 [26] for external visualization using the X-Plane Blockset (XPB) [27]. By sending the vehicle position and attitude information, X-Plane begins acting as a visualization engine—no longer using X-Plane's internal vehicle physics models. Other X-Plane parameters (referred to as *datarefs*) can be directly written to, providing additional capabilities including (i) vehicle animation, (ii) changing refresh rate, (iii) changing date and time (e.g., morning, noon, evening flight) and (iv) sending system commands. Further details regarding the X-Plane visualization development process are described in Appendix B.

FlightGear [28] was also considered to be used as the SIMPL platform's flight visualization software. The following comparison between FlightGear and X-Plane was created to help select the visualization software used for this work:

1. **MATLAB® Support:** Both X-Plane and FlightGear integrate with MATLAB®/Simulink® using available toolboxes. X-Plane toolboxes support seamless animation and dataref control, whereas FlightGear requires custom communication files to be defined to modify custom parameters. This is noteworthy for vehicle animation, where the X-Plane datarefs are directly controlled from MATLAB®/Simulink® using available toolboxes, streamlining the integration process.

2. **Image Quality:** X-Plane has better out-of-the-box graphics and more realistic aircraft aesthetics compared to FlightGear; however, this comes at the expense of requiring additional computational resources.

3. **Cost:** FlightGear is free, open-source software. X-Plane is commercial software, with the professional license costing $1,000 per machine at the time of writing this report.

4. **Animation:** X-Plane add-ons are supported directly from Blender, which can be used to animate and texture vehicle models. FlightGear animation, in the lead author's experience, is more difficult with less available online help.

Based on this comparison and the lead author's experience using both programs, X-Plane 11 was selected as the flight simulation visualization software for the present application.

4 Flight Control System Architecture

The flight control system (FCS) architecture is shown in Fig. 2, depicting the different subsystems within the control system. The general FCS architecture layout is based on prior works by Comer and Chakraborty [8, 9, 22–24]. Feedback signals directly involved in command generation are also shown with dashed lines. The FCS comprises the following key controller modules:

- **Pilot Command Mappings (PCM):** houses the stick filters, command mode determination, and autopilot/hold functions. The pilot inputs ($p_{ver}, p_{acc}, p_{dir}, p_{lat}$) are mapped to respective longitudinal and lateral axis commands. The longitudinal commands include vertical velocity command VV_{cmd} and longitudinal acceleration command acc_{cmd}. The lateral commands include turn rate command $\dot{\psi}_{cmd}$ and lateral velocity command v_{cmd}.

- **Transition Flight Controller (TFC):** computes the required total thrust T_{cmd} based on a desired vertical velocity and acceleration. In VTOL applications, TFC can also provide nacelle angle commands $\theta_{nac,cmd}$ based on Total Energy principles.

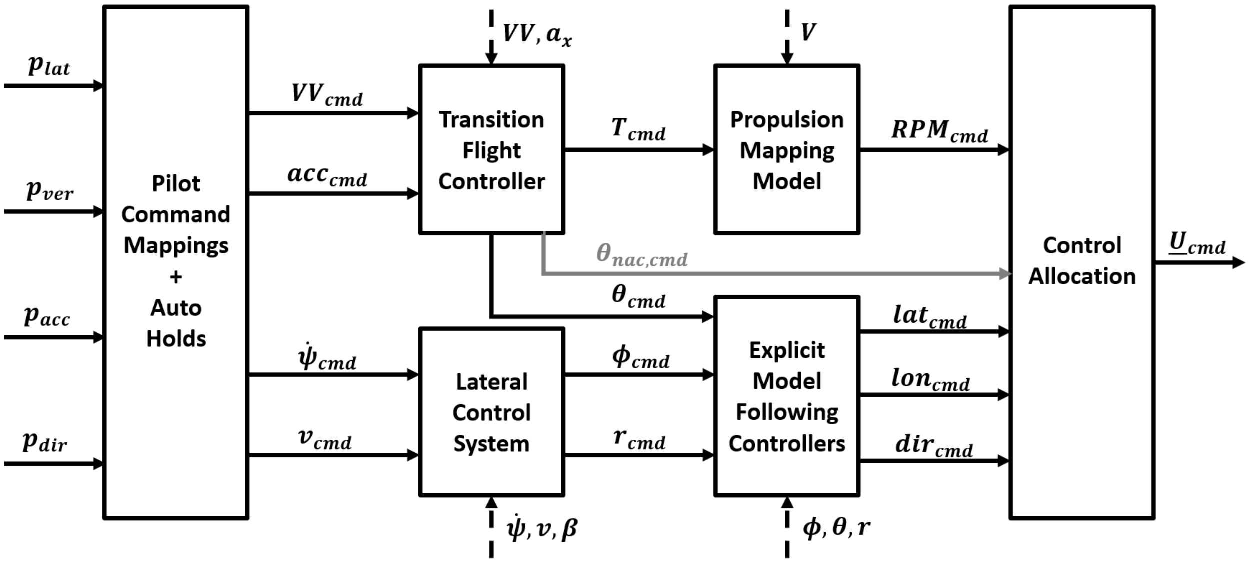


Figure 2: Top-level view of the TFC-based flight control system architecture with commands (solid lines) and feedback states (dashed lines) shown for each subsystem.

- **Lateral Control System (LATCS):** converts incoming turn rate and lateral velocity commands into a corresponding bank angle command ϕ_{cmd} and yaw rate command r_{cmd} based on the flight conditions.
- **Propulsion Mapping Model (PMM):** converts incoming thrust commands to equivalent RPM commands for a given propulsor based on a scheduled, linear modeling approach.
- **Explicit Model-Following (EMF) Inner-Loops:** provides inner-loop attitude and rate stabilization, tracking commanded roll, pitch, and yaw rate commands by commanding normalized lateral lat_{cmd}, longitudinal lon_{cmd}, and directional dir_{cmd} commands to the allocator.
- **Control Allocator:** converts incoming RPMs and normalized control inputs into the control effector commands.

4.1 Pilot Command Mappings

The Pilot Command Mappings (PCM) subsystem maps incoming pilot commands to corresponding commands for each axis system and is responsible for outer-level "holds." Holds are pilot-aiding functions that maintain a flight condition under specific flight conditions when the pilot is not directly commanding the corresponding axis. The commands are first passed through a first-order stick filter of the form

$$\frac{p_{(\cdot),cmd}}{p_{(\cdot),in}}(s) = \frac{1}{\tau s + 1} \tag{18}$$

which relates a raw stick input $p_{(\cdot),in}$ to a filtered command $p_{(\cdot),cmd}$ based on a desired filter time constant τ. The four pilot inputs include the vertical command p_{ver}, acceleration command p_{acc}, directional command p_{dir}, and lateral command p_{lat}. These filtered inputs are then converted into the longitudinal commands

$$VV_{cmd} = VV_{cmd,max} p_{ver}\ , \qquad acc_{cmd} = \frac{acc_{cmd,max}}{g} p_{acc} \tag{19}$$

and lateral commands

$$\dot{\psi}_{cmd} = \dot{\psi}_{cmd,max} p_{dir} \ , \qquad v_{cmd} = v_{cmd,max} p_{lat} \tag{20}$$

displayed in Fig. 2, where each respective axis's maximum allowable command is directly multiplied by the pilot inputs, which span $p_{(\cdot),in} \in [-1, +1]$.

The pilot commands (left and right transmitter sticks) map directly to the commands shown in Fig. 3. These commands are maintained in hovering and forward flight, and the Transition Flight Controller and Lateral Control System (described subsequently), generate the required inner-loop commands to achieve these commands. The sticks, when centered, return the vehicle to wings-level, unaccelerated flight with no vertical velocity.

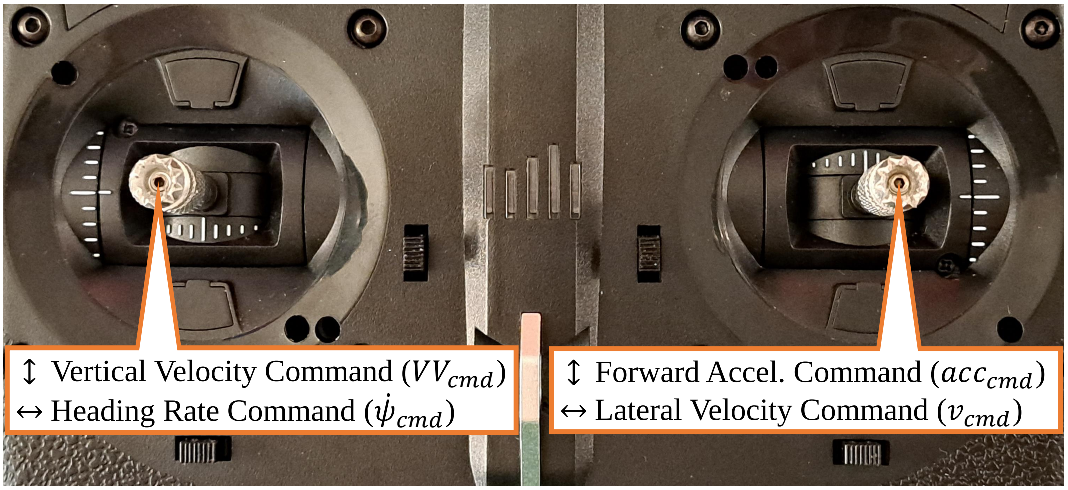

↕ Vertical Velocity Command (VV_{cmd})
↔ Heading Rate Command ($\dot{\psi}_{cmd}$)
↕ Forward Accel. Command (acc_{cmd})
↔ Lateral Velocity Command (v_{cmd})

Figure 3: Pilot command variables assigned to left and right transmitter sticks.

In forward flight mode, the acceleration command is synthesized as $a_{cmd,FFM} = K_V(V_{cmd} - V)$, with the stick commanding velocity rate $\dot{V}_{cmd}$ directly. This commanded velocity rate changes the current velocity command V_{cmd} at the commanded rate. The direct, normalized acceleration command in vertical flight mode blends linearly with the forward flight mode command. In vertical flight mode, a dissipation factor is also included to slowly return the vehicle to a hover; this effect also washes out as the vehicle speeds up. The net acceleration command is synthesized as:

$$\left(\frac{\dot{V}}{g}\right)_{cmd} = \zeta_{acc} a_{cmd,VFM} + (1 - \zeta_{acc}) a_{cmd,FFM} \tag{21}$$

where transition from vertical flight mode (VFM) control commands to forward flight mode (FFM) control commands occurs at a predetermined airspeed range ($V_{acc,1}$, $V_{acc,2}$), providing a linear blending $\zeta_{acc} \in [0, 1]$ between VFM and FFM commands.

4.1.1 Altitude Hold

Altitude hold enables when the pilot is not commanding vertical velocity ($p_{ver} = 0$), the vehicle is above minimum operating altitude ($h > h_{min}$), and the pilot has enabled the hold ($ALT = 1$) in the flight control system parameters. The altitude target is determined with a look-ahead factor LAF to provide smooth crossover. This is achieved by dynamically computing an altitude "anchor" rate as follows:

$$\dot{h}_{cmd} = K\left(h + LAF * VV - h_{cmd}\right) \tag{22}$$

By selecting K to be a large value, this logic essentially acts as a rapid first-order tracker, shadowing the reference value while disengaged. The dynamics in Eq. (22) are held constant $\dot{h} = 0$ when the hold enables, latching on to the reference altitude. Once latched, the vertical velocity command is generated using a proportional controller $VV_{cmd} = K_h(h_{cmd} - h)$.

4.1.2 Heading Hold

The heading hold follows a similar logic to the altitude hold. The hold enables when the pilot is not commanding a turn rate ($p_{dir} = 0$), the turn rate has dropped below a specified threshold ($\dot{\psi} < \dot{\psi}_{lim}$, indicating that the aircraft is not carrying a large rate), and the vehicle is above the minimum operating altitude ($h > h_{min}$). The heading target is once again determined with a look-head factor LAF to smooth the heading hold enabling. The heading target is dynamically computed as:

$$\dot{\psi}_{cmd} = K\left(\psi + LAF * \dot{\psi} - \psi_{cmd}\right) \tag{23}$$

A turn rate command is generated to capture the heading using a proportional controller:

$$\dot{\psi}_{cmd} = K_{\psi}(\psi_{cmd} - \psi)$$

4.1.3 Velocity Hold

Velocity hold enables when the pilot is not commanding acceleration ($p_{acc} = 0$) and the vehicle is not in ground contact ($GC = 0$). The target velocity is determined in the same way as the altitude and heading holds, using a look-ahead factor for smooth capture as follows:

$$\dot{V}_{cmd} = K(V + LAF * a_x - V_{cmd}) \tag{24}$$

The velocity is held by commanding an acceleration proportional to the error as

$$acc_{cmd} = \frac{K_v}{g}(V_{cmd} - V)$$

4.1.4 Position Hold

Position hold engages in hover conditions when the ground speed falls below a specified threshold and the pilot inceptor inputs responsible for repositioning (p_{lat}, p_{acc}) are centered. Longitudinal and lateral position errors (x_e, y_e) are calculated relative to position target (x_{cmd}, y_{cmd}) updated dynamically when the position hold is disengaged as:

$$\begin{aligned} \dot{x}_{cmd} &= K(x + LAF * VFWD - x_{cmd}) && (25) \\ \dot{y}_{cmd} &= K(y + LAF * VLAT - y_{cmd}) && (26) \end{aligned}$$

These errors then generate an acceleration command $(\dot{V}/g)_{cmd}$ for the longitudinal axis and a lateral velocity command $VLAT_{cmd}$ for the lateral axis through PI control with rate damping as:

$$\begin{aligned} \left(\frac{\dot{V}}{g}\right)_{cmd} &= K_{P,x}x_e + K_{I,x}\int x_e \, dt - K_{D,x}VFWD && (27) \\ VLAT_{cmd} &= K_{P,y}y_e + K_{I,y}\int y_e \, dt - K_{D,y}VLAT && (28) \end{aligned}$$

4.1.5 Auto-Takeoff (Fixed-Wing)

Auto-takeoff mode, when enabled, performs a takeoff maneuver by commanding a reference altitude $h_{cmd} = 30$ ft and reference velocity $V_{cmd} = 30$ kts. These commands are acted on using the same logic as the altitude and velocity holds, using proportional controllers to generate the required vertical velocity and acceleration commands. Control allocation schedules ensure that the appropriate control effectors are used during the takeoff roll. The auto-takeoff logic remains active until the altitude is within 2 ft of the target altitude and the velocity is within 2 kts of the target velocity. When these criteria are met, the auto-takeoff mode automatically disengages and remains disengaged unless re-engaged by the pilot.

4.2 Transition Flight Controller

The Transition Flight Controller (TFC) was designed following concepts from the latest version of the Total Energy Control System (TECS) [29, 30], which aims to balance the energy rates (potential and kinetic) on the vehicle through coordinated flight path and velocity control. Prior work by Comer and Chakraborty [8, 9] extended this controller to lift-plus-cruise and tilt-wing VTOL configurations. Some of these concepts were also used to provide a basis for the TFC controller.

The TFC controller acts on incoming vertical velocity VV_{cmd} and acceleration acc_{cmd} commands to generate a thrust command T_{cmd}, pitch attitude command θ_{cmd}, and a nacelle angle command $\theta_{nac,cmd}$. At the core of the controller design, the total energy Hamiltonian $\dot{E}$ and Lagrangian $\dot{L}$ rates are used and defined as follows:

$$\dot{E} = \gamma + \frac{\dot{V}}{g} \tag{29}$$

$$\dot{L} = \frac{\dot{V}}{g} - \gamma \tag{30}$$

For extension to hovering flight, a conversion factor F is first developed following [8, 9] to convert vertical velocity into an equivalent flight path angle from $VV \approx V\sin(\gamma)$. Using small angle approximations, the term $F = 1/V$ is used for computations such that $\gamma \approx VV * F$. The energy-rate terms and errors can now be defined as:

$$\dot{E} = VV * F + \frac{\dot{V}}{g}\ , \qquad \dot{E}_{err} = F\left(VV_{cmd} - VV\right) + \left(\left(\frac{\dot{V}}{g}\right)_{cmd} - \frac{\dot{V}}{g}\right) \tag{31}$$

$$\dot{L} = \frac{\dot{V}}{g} - VV * F\ , \qquad \dot{L}_{err} = \left(\left(\frac{\dot{V}}{g}\right)_{cmd} - \frac{\dot{V}}{g}\right) - F\left(VV_{cmd} - VV\right) \tag{32}$$

Following the latest TECS updates [30], the additional pitch correction term K_{trim} is included to provided pitch attitude compensation due to acceleration and deceleration. Using these terms, the original TECS equations are modified to produce a thrust-to-weight command and pitch attitude command as:

$$\left(\frac{T}{W}\right)_{cmd} = \frac{K_{TI}}{s}\dot{E}_{err} - K_{TP}\dot{E} \tag{33}$$

$$\theta_{cmd} = -\frac{K_{EI}}{s}\left\{\dot{L}_{err} - K_{trim}\frac{\dot{V}}{g}\right\} + K_{EP}\dot{L} \tag{34}$$

The TFC architecture accounts for rotating propulsors through a common "nacelle" term. This term represents the common tilt angle of propulsors on a VTOL aircraft (e.g., the wing angle in a tilt-wing and the rotor tilt angle in a tilt-rotor). The nacelle angle, beginning at hover condition $\theta_{nac,cmd} = 90°$, responds to acceleration and vertical velocity errors as follows:

$$\dot{\delta}_{nac,cmd} = -K_x \left(\frac{\dot{V}}{g}\right)^{+}_{err} + K_z \left(VV_{err} * F\right)^{+} \tag{35}$$

where K_x and K_z are scheduled gains and the x^{+} terms are nacelle-angle corrected error terms. The concept of nacelle-angle correction stems from the differing effectiveness of nacelle movements as the vehicle transitions. In hovering conditions, nacelle angle movements will provide almost no effect on vertical velocity, but will result in acceleration changes. In near-forward flight conditions ($\theta_{nac} < 20°$), the nacelle angle movements will have minor effects on the acceleration of the vehicle, but will impact vertical velocity substantially. To account for these effectiveness changes, the following relationships are developed:

$$\begin{aligned}
(VV_{err})^{+} &= (VV_{cmd} - VV)\left(1 - \sin(\theta_{nac})\right) && (36)\\
(VV_{err})^{*} &= (VV_{cmd} - VV)\sin(\theta_{nac}) && (37)\\
\left(\frac{\dot{V}}{g}\right)^{+}_{err} &= \left(\left(\frac{\dot{V}}{g}\right)_{cmd} - \frac{\dot{V}}{g}\right)\sin(\theta_{nac}) && (38)\\
\left(\frac{\dot{V}}{g}\right)^{*}_{err} &= \left(\left(\frac{\dot{V}}{g}\right)_{cmd} - \frac{\dot{V}}{g}\right)\left(1 - \sin(\theta_{nac})\right) && (39)
\end{aligned}$$

These relationships naturally wash-in and wash-out the nacelle angle movements to respective axis errors. The total thrust command $(T/W)_{cmd}$ can be further subdivided into respective horizontal $(T/W)_{x,cmd}$ and vertical $(T/W)_{z,cmd}$ commands based on the current nacelle angle command $\theta_{nac,cmd}$ as:

$$(T/W)_{x,cmd} = (T/W)_{cmd}\cos(\theta_{nac,cmd}) \;, \qquad (T/W)_{z,cmd} = (T/W)_{cmd}\sin(\theta_{nac,cmd}) \tag{40}$$

These commands provide the necessary horizontal and vertical thrust requirements to achieve the desired trajectory commands. These commands are then allocated and realized by the Propulsion Mapping Model.

Three modes have been considered in the TFC architecture. These modes correspond with fundamental transitioning flight vehicle modes and are described in the subsequent sections.

4.2.1 Hovering Flight (Mode 1)

Hovering flight encompasses flight in and around hover conditions, where acceleration and vertical velocity commands are nearly decoupled between propulsor and attitude responses. This mode typically encompasses nacelle angles in the range $[80°, 90°]$. Multi-copter configurations would operate solely in this mode under the TFC. In this flight regime, the following three observations can be made: (i) nacelle movement is ineffective for meeting vertical velocity commands, (ii) pitch attitude changes are ineffective for meeting vertical velocity commands, and (iii) thrust changes are ineffective for meeting acceleration commands. As such, the thrust-to-weight and pitch attitude commands in hovering flight are formed from:

$$\begin{aligned}\left(\frac{T}{W}\right)_{cmd} &= \frac{K_{TI}}{s}\dot{E}^*_{err} - K_{TP}\dot{E}^* \qquad (41)\\ \theta_{cmd} &= -\frac{K_{EI}}{s}\dot{L}^+_{err} + K_{EP}\dot{L}^+ \qquad (42)\end{aligned}$$

To improve hover responsiveness to acceleration commands, a feed-forward gain K_A^{ff} was proposed by Comer and Chakraborty [8] to the pitch axis. In hovering applications, acceleration commands can map directly to a pitch attitude command following $(\dot{V}/g)_{cmd} = -\theta_{cmd}$ using small angle approximations where $\sin(\theta) \approx \theta$. This term acts directly on the pitch command (Eq. (42)) as:

$$\theta_{cmd} = -\frac{K_{EI}}{s}\dot{L}^+_{err} + K_{EP}\dot{L}^+ - K_A^{ff}\left(\frac{\dot{V}}{g}\right)_{cmd} \qquad (43)$$

4.2.2 Transition Flight (Mode 2)

This mode encompasses any form of flight where the vehicle is producing thrust-borne and wing-borne lift. This mode would typically encompass nacelle angles in the range $[15°, 75°]$. In this flight regime, the following observations can be noted: (i) pitch attitude to achieve flight path would provide inadvertent thrust vectoring, (ii) on-axis vertical and acceleration responses become increasingly more coupled, and (iii) there exists a combination of nacelle angle and thrust to reach a desired flight condition. Based on these observations, it is clear that acceleration and vertical velocity commands can be achieved through a combination of thrust and nacelle angle control. Pitch attitude is held level ($\theta_{cmd} = 0$) via first-order dynamics by discharging the $\dot{L}$ integrator as follows:

$$\dot{L}_{err} = -K_{EI}\int \dot{L}_{err}\ dt\ , \qquad \theta_{cmd} = \int \dot{L}_{err}\ dt \qquad (44)$$

Thrust requirements are met in the same way as in hovering flight (Eq. (41)), with diminishing responsiveness to vertical velocity commands as the nacelle angle reduces. Nacelle angle is commanded based on a combination of horizontal acceleration and vertical velocity commands. To provide smooth blending between modes and no jump-discontinuities, the nacelle angle rate $\dot{\delta}_{nac,cmd}$ is commanded following Eq. (35).

4.2.3 Forward Flight (Mode 3)

As the nacelles tilt further forward ($\delta_{nac} < 15°$), the vehicle can be controlled using fixed-wing concepts, where aerodynamic surfaces are used and lifting propulsors are shut down. When the vehicle enters into this operating mode, the nacelle angle is commanded to a forward-flight orientation $\delta_{nac,fwd}$ using a first-order dynamic as follows:

$$\dot{\delta}_{nac,cmd} = K(\delta_{nac,fwd} - \delta_{nac}) \qquad (45)$$

This provides a smooth final movement of the nacelles into their forward-flight position at a specified rate governed by the gain K. This method of transition was also employed in prior work by Comer and Chakraborty [19, 31] with success.

With the nacelle angle fixed in a forward-flight position, the governing equations for thrust and pitch follow the original TECS equations as:

$$\begin{aligned} \left(\frac{T}{W}\right)_{cmd} &= \frac{K_{TI}}{s}\dot{E}_{err} - K_{TP}\dot{E} && (46) \\ \theta_{cmd} &= -\frac{K_{EI}}{s}\left\{\dot{L}_{err} - K_{trim}\frac{\dot{V}}{g}\right\} + K_{EP}\dot{L} && (47) \end{aligned}$$

The TECS architecture's responsiveness has been criticized by some researchers. This was originally addressed in the first formulations of feed-forward gains [32, 33]. A feed-forward gain was also incorporated into the pitch axis for this work, providing a simple, yet effective, direct pitch command. Under the assumption of small angles, the approximation $VV \approx \gamma$ is valid. In near steady-level, unaccelerated flight conditions, the approximation of $\Delta\theta \approx \Delta\gamma$ holds fairly valid. As a result, the proposed feed-forward pitch attitude command of $\theta^{ff}_{cmd} = VV_{cmd} * F$ was used in Eq. (48) to provide increased responsiveness of the vehicle to vertical velocity commands, forming the pitch attitude command as:

$$\theta_{cmd} = -\frac{K_{EI}}{s}\left\{\dot{L}_{err} - K_{trim}\frac{\dot{V}}{g}\right\} + K_{EP}\dot{L} + VV_{cmd} * F \tag{48}$$

4.2.4 Vertical Takeoff Assist Mode

To support smooth vertical takeoff and landing operations, a takeoff module was incorporated into the TFC controller to provide a smooth spool-up to takeoff thrust. This design concept was originally developed and tested in the Trajectory Control System, developed by Chakraborty and Comer [19, 20, 31]. The thrust command is set to an idle command $(T/W)_{idle}$ when in ground contact and when a vertical velocity command threshold is met such that $VV_{cmd} < VV_{TKO}$, where VV_{TKO} is the minimum pilot vertical velocity command required to initiate takeoff. When a takeoff is commanded, the thrust is set to a specified takeoff value $(T/W)_{TKO}$ at a specified rate K_{spool} plus an additional command $(T/W)_{boost}$ which responds linearly to the magnitude of the incoming vertical velocity command VV_{cmd} from the pilot to modulate takeoff aggressiveness. This logic acts through the $\dot{E}$ channel, directly resulting in a $(T/W)_{cmd}$ to takeoff.

$$\begin{gathered} \frac{d}{dt}\left(\frac{T}{W}\right)_{cmd} = K_{spool}\left\{\left(\frac{T}{W}\right)_{ref} - \left(\frac{T}{W}\right)_{idle}\right\}, \\ \left(\frac{T}{W}\right)_{ref} = \begin{cases} \left(\frac{T}{W}\right)_{idle} & \text{if } VV_{cmd} < VV_{TKO} \\ \left(\frac{T}{W}\right)_{TKO} + VV_{cmd}\left(\frac{T}{W}\right)_{boost} & \text{otherwise} \end{cases} \end{gathered} \tag{49}$$

4.3 Lateral Control System (LATCS)

The lateral control system is a middle-loop lateral/directional controller that computes bank angle ϕ_{cmd} and yaw rate r_{cmd} commands based on turn rate commands $\dot{\psi}_{cmd}$ and lateral velocity commands v_{cmd}. The foundations of this control blending scheme were first investigated in [34]. The controller blends vertical flight commands with forward flight commands through a linear speed blend based on defined control system limits. This blending occurs around a determined speed V_x defined as

$$V_x = \frac{v_{max}}{\sin(\beta_{max})} \tag{50}$$

where v_{max} is the maximum allowable lateral velocity and β_{max} is the maximum allowable sideslip. The computed V_x is used to form a blending factor ζ_{lat} which blends linearly from 0 (vertical flight) to 1 (forward flight) around the crossover speed V_x. The commanded bank angle and yaw rate are generated as follows:

$$\phi_{cmd} = K_v(v_{cmd} - v)(1 - \zeta_{lat}) + \tan^{-1}\left(\frac{V\dot{\psi}_{cmd}}{g}\right)\zeta_{lat} \tag{51}$$

$$r_{cmd} = \dot{\psi}_{cmd}(1 - \zeta_{lat}) + \zeta_{lat}\left[\frac{g}{V}\cos(\gamma)\sin(\phi_{cmd}) - K_\beta\left(\beta_{cmd} - \beta\right)\right] \tag{52}$$

Coordinated turn kinematics are used to determine the resulting bank angle ϕ_{cmd} and yaw rate r_{cmd} commands in forward flight along with a sideslip β correction factor.

4.4 Propulsion Mapping Model

The Propulsion Mapping Model (PMM) is responsible for: (i) allocation of thrust to respective propulsor groups and (ii) mapping propulsor group commands to equivalent RPM commands.

4.4.1 Propulsor Group Mappings

Horizontal and vertical thrust commands generated from the TFC architecture must be mapped to respective propulsor groups. In general, VTOL configurations have a combination of fixed vertical, fixed horizontal, or tilting propulsors. The general mapping of the thrust to these groups can be best realized by considering the following five configuration types in steady, level flight:

- **Fixed-Wing:** features forward-mounted propulsors with no ability to tilt. Horizontal thrust is allocated completely to the propulsors, while vertical thrust requirements are met through attitude changes and are not allocated.

- **Multi-Copter:** features upward-mounted propulsors no with ability to tilt. Vertical thrust is allocated completely to the propulsors. Horizontal thrust requirements are met through attitude changes and are not allocated.

- **Lift-Plus-Cruise:** features both forward- and upward-mounted propulsors with no tilt capabilities. Vertical thrust is naturally mapped to the upward-mounted propulsors and horizontal thrust is naturally mapped to the forward-mounted propulsors.

- **Tilt-Wing:** features a set of propulsors that tilt together and no fixed propulsors. Thrust is allocated completely to the propulsors.

- **Vectored-Thrust:** can feature a combination of fixed and tilting propulsors. Thrust is allocated on a configuration-specific basis, mapping vertical thrust to upward-mounted propulsors and a portion of the remaining requirements to tilting propulsors based on the nacelle angle. Horizontal thrust is mapped to forward-mounted propulsors and a portion of the remaining required horizontal thrust to tilting propulsors based on the nacelle angle.

4.4.2 Inverse Propulsor Model

A thrust command T_{cmd} is mapped to an equivalent RPM command in this module. The mapping method used in this work makes use of linearized thrust models following work from [24, 34], scheduling with airspeed. For a generic fixed or tilting propeller, this inverse model is determined as follows:

$$K_{T2RPM} = \frac{1}{m}\sqrt{\left(\frac{\partial RPM}{\partial \dot{u}}\right)^2 + \left(\frac{\partial RPM}{\partial \dot{w}}\right)^2} \tag{53}$$

This model effectively computes the net change in acceleration (horizontal acceleration $\dot{u}$ and vertical acceleration $\dot{w}$) due to a change in RPM. The partial derivative terms in Eq. (53) are determined using the linearized model **B** matrix, which is scheduled with airspeed along the trim points. The differential RPM about the trim condition is then formed as:

$$\Delta RPM = K_{T2RPM}\Delta T \tag{54}$$

4.5 Explicit Model Following Controllers

The inner-loop attitude and rate controllers employ an explicit model-following (EMF) control architecture. This architecture, used extensively in rotorcraft applications [17, 35, 36], is also applicable to fixed-wing control as described in [17, 37, 38]. The block diagram of the EMF controllers integrated with the actuator, plant, and sensor dynamics is shown in Fig. 4.

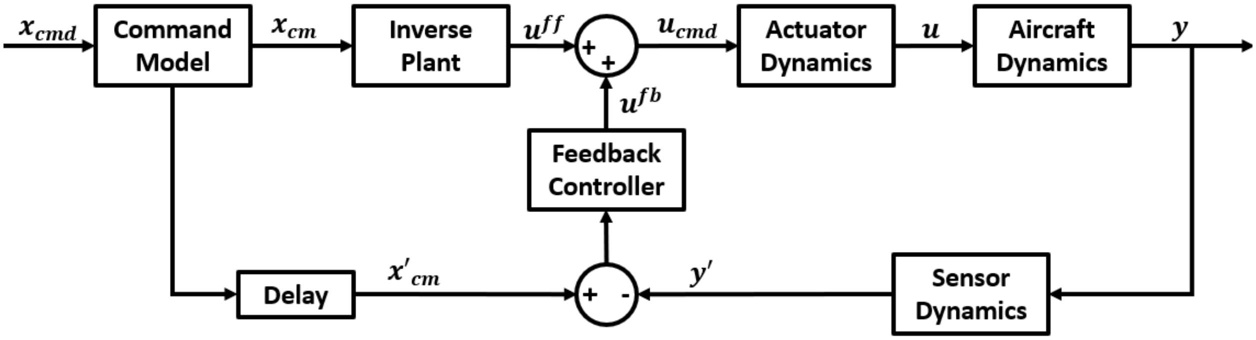


Figure 4: Block diagram of the generalized EMF control architecture with actuator, aircraft, and sensor dynamics.

The incoming command x_{cmd} first passes through a *Command Model*, providing a desired shaping of the command to the EMF controllers. This command model typically takes on a first- or second-order form. All three axes employ a second-order command model of the form

$$\frac{x_{cm}}{x_{cmd}}(s) = \frac{\omega_{n,cm}^2}{s^2 + 2\zeta_{cm}\omega_{n,cm}s + \omega_{n,cm}^2} \tag{55}$$

with a selection of $\omega_{n,cm} = 3$ and $\zeta_{cm} = 1$ to achieve an approximate settling time of 2 seconds.

The shaped output x_{cm} then feeds into an *Inverse Plant* model housing a lower-order equivalent system (LOES) model. The LOES model is used to form an approximate fitting of the command as a feed-forward input to the control. For the roll, pitch, and yaw axes the LOES function is the inverse of a fitting of p/u_{lat}, q/u_{lon}, and r/u_{dir}, respectively. The fitting process is detailed further in Sec. 4.5.1. The output from this is the feed-forward control command u^{ff}.

After passing through an equivalent *Delay*, the shaped command output x_{cm} is compared to the feedback states to develop errors to be acted on by the *Feedback Controller*. The feedback control can take several forms, including conventional proportional-integral-derivative (PID) control and linear quadratic regulator (LQR) control. Conventional PID control is employed in this work. This forms the feedback control element u^{fb} which is then summed with the feed-forward control to form the total control command u_{cmd}.

The control commands then pass through the *Actuator Dynamics* and *Aircraft Dynamics*, ultimately forming the output states. These states finally pass through the *Sensor Dynamics*, forming the sensed feedback.

4.5.1 Lower-Order Equivalent System

The *Inverse Plant* operates using a LOES model fit in a frequency range of interest for each axis. For the SEARCH vehicle, the inverse plant model was fit around a "truth" model, which was the on-axis transfer function from the linearized aircraft dynamics model. This fitting occurred across the speed envelope in increments of 10 knots. The fitting process was performed using a constrained-parameter minimization problem with the objective function following suggestions from [17, 39], as follows:

$$J_{LOES} = \frac{20}{n} \sum_{i=1}^{n} \left\{ (M_{i,truth} - M_{i,LOES})^2 + 0.01745(\phi_{i,truth} - \phi_{i,LOES})^2 \right\} \tag{56}$$

The fitting occurred with a selected $n = 50$ number of points across the fitting range. The LOES models were fit in the following frequency ranges: (i) roll axis fit from $\omega_n \in [3.0, 20.0]$ rad/s, (ii) pitch axis from $\omega_n \in [2.0, 20.0]$ rad/s, and (iii) yaw axis from $\omega_n \in [2.0, 20.0]$ rad/s. These frequency ranges were chosen to isolate the on-axis damping modes (roll mode, short period mode, and Dutch roll mode, respectively) following analysis of the eigenvalues of the vehicle. As a rule of thumb, acceptable LOES fits have $J_{LOES} \leq 100$, while $J_{LOES} \leq 50$ are considered indistinguishable from the truth transfer function [17]. $J_{LOES} \leq 10$ are desirable for modern fixed-wing control systems [17]. Based on these guidelines, the J_{LOES} costs for all fits were desirable within the fitting range.

4.5.2 Roll Axis

The roll axis operates in an attitude-command-attitude-hold (ACAH) control scheme. The reference bank angle command ϕ_{cmd} is generated from the upstream Lateral Control System (LATCS). A first-order LOES model is used to achieve the feed-forward command, relating normalized lateral command lat_{cmd} to roll rate p as:

$$\frac{p}{lat_{cmd}}(s) = \frac{L_{lat_{cmd}}}{s - L_p} \implies lat_{cmd}^{ff} = \frac{1}{L_{lat_{cmd}}} \left(\dot{p}_{cm} - L_p p_{cm} \right) \tag{57}$$

This feed-forward component is then synthesized with the feedback controller, which utilizes a proportional-integral-derivative (PID) control scheme for feedback stabilization and disturbance rejection. The feedback control operates on the command model signal, which tracks the command following:

$$\frac{\phi_{cm}}{\phi_{cmd}}(s) = \frac{\omega_{cm}^2}{s^2 + 2\zeta_{cm}\omega_{cm}s + \omega_{cm}^2} \tag{58}$$

The feedback control is then formed as:

$$lat_{cmd}^{fb} = K_P \left(\phi_{cm} - \phi \right) + \frac{K_I}{s} \left(\phi_{cm} - \phi \right) + K_D \left(p_{cm} - p \right) \tag{59}$$

where p_{cm} is the derivative state of Eq. (58). A roll-due-to-yaw coupling term K_{ry} is also applied on the roll axis, forming the total lateral control command lat_{cmd} as:

$$lat_{cmd} = lat_{cmd}^{ff} + lat_{cmd}^{fb} + K_{ry} dir_{cmd} \tag{60}$$

4.5.3 Pitch Axis

The pitch axis operates in an attitude-command-attitude-hold (ACAH) control scheme. The reference pitch angle command θ_{cmd} is generated from the upstream TFC. A second-order LOES model is used to achieve the feed-forward command, relating normalized longitudinal command lon_{cmd} to pitch rate q as:

$$\begin{aligned} \frac{q}{lon_{cmd}}(s) &= \frac{M_{lon_{cmd}}(s + 1/T_{\theta_2})}{s^2 + 2\zeta_{sp}\omega_{sp}s + \omega_{sp}^2} \implies \\ lon_{cmd}^{ff} &= \frac{1}{M_{lon_{cmd}}}\left(\ddot{q}_{cm} + 2\zeta_{sp}\omega_{sp}\ \dot{q}_{cm} + \omega_{sp}^2\ q_{cm} - \frac{M_{u_{lon}}}{T_{\theta_2}}\ u_{lon}^{ff}\right) \end{aligned} \tag{61}$$

The optimization problem was bounded to avoid the high-frequency zero sometimes encountered when fitting this transfer function [40]. This feed-forward component is then synthesized with the feedback controller, which utilizes a proportional-integral-derivative (PID) control scheme for feedback stabilization and disturbance rejection. The feedback control operates on the command model signal, which tracks the command following:

$$\frac{\theta_{cm}}{\theta_{cmd}}(s) = \frac{\omega_{cm}^2}{s^2 + 2\zeta_{cm}\omega_{cm}s + \omega_{cm}^2} \tag{62}$$

The feedback control is then formed as:

$$lon_{cmd}^{fb} = K_P\left(\theta_{cm} - \theta\right) + \frac{K_I}{s}\left(\theta_{cm} - \theta\right) + K_D\left(q_{cm} - q\right) \tag{63}$$

where q_{cm} is the derivative state of Eq. (62).

4.5.4 Yaw Axis

The yaw axis operates on a rate-command-rate hold (RCRH) architecture, acting on an incoming yaw rate command r_{cmd} to produce a normalized directional command dir_{cmd}. In prior works using EMF architectures, feed-forward elements have been used for rotorcraft, VTOL, and fixed-wing applications. In this application, the feed-forward portion of the EMF controller is not used. The lateral-directional coupling term K_{yr} provides a direct directional command due to lateral command inputs. This term acts as a feed-forward to provide direct uncoupling. The feedback terms then act to compensate for remaining error through proportional-integral (PI) control. The command model on the yaw axis is also second order of the form:

$$\frac{r_{cm}}{r_{cmd}}(s) = \frac{\omega_{cm}^2}{s^2 + 2\zeta_{cm}\omega_{cm}s + \omega_{cm}^2} \tag{64}$$

The feedback control is then formed as:

$$dir_{cmd}^{fb} = K_P\left(r_{cm} - r\right) + \frac{K_I}{s}\left(r_{cm} - r\right) \tag{65}$$

to form the total directional control command:

$$dir_{cmd} = K_{yr}lat_{cmd} + dir_{cmd}^{fb} \tag{66}$$

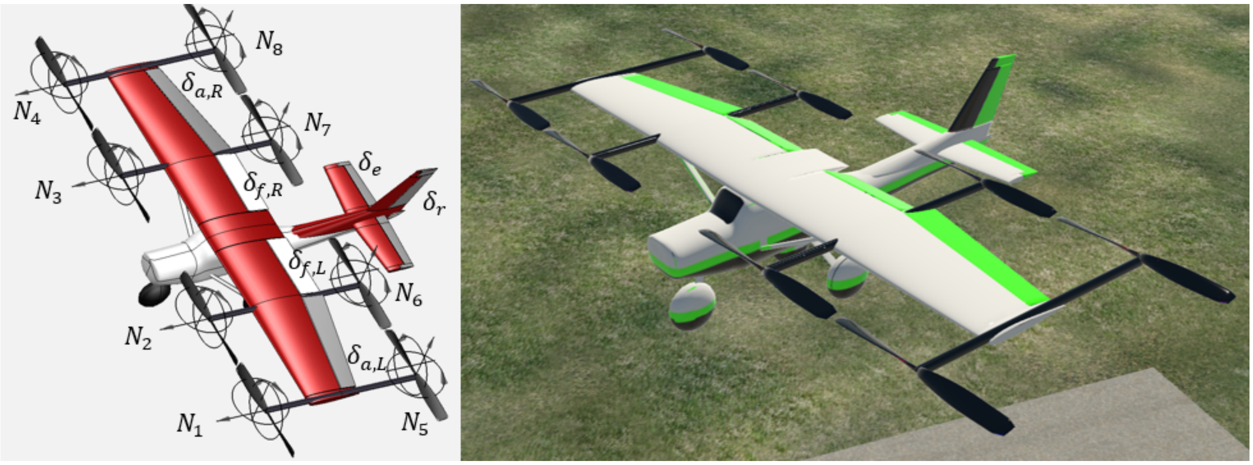


Figure 5: SEARCH VTOL OpenVSP and X-Plane external visualization models.

5 SEARCH VTOL Simulation

The proposed FCS architecture was intended for use on VTOL aircraft configurations. A modified vectored-thrust version of the SEARCH fixed-wing vehicle was proposed as a candidate configuration for testing the control system in a simulation environment. The SEARCH VTOL configuration is shown in Fig. 5 in the OpenVSP [41] and X-Plane [26] environments.

The configuration is based on the SEARCH subscale airframe, modified to incorporate four additional booms on the wing. The booms are positioned at the half-span and wingtips of the wings and feature eight propulsors mounted at the fore and aft tips of the booms. The leading propulsors tilt forward (referred to as main propulsors, MP) while the rear propulsors are fixed in a vertical orientation (referred to as lift propulsors, LP). The rear propulsors are canted 15° about the roll axis in an alternating fashion to aid in yaw control. The full list of control effectors for the SEARCH VTOL vehicle is shown in Table 1.

Table 1: SEARCH VTOL vehicle control effectors.

Effector	Description	Position Limits
$\delta_{a,L}$	Left aileron	$[-30°, +30°]$
$\delta_{f,L}$	Left flap	$[0°, +30°]$
$\delta_{f,R}$	Right flap	$[0°, +30°]$
$\delta_{a,R}$	Right aileron	$[-30°, +30°]$
δ_e	Elevator	$[-30°, +30°]$
δ_r	Rudder	$[-30°, +30°]$
δ_{nac}	Nacelle angle	$[0°, 90°]$
$N_1, N_2, ..., N_8$	Propeller RPM	$[0, 9000]$ RPM

Assumed properties of the SEARCH VTOL vehicle are shown in Table 2. The mass properties were scaled based on the measured SEARCH vehicle mass properties (see Sec. 7). This scaling was performed heuristically to achieve a target mass of 17.6 lb (8 kg) and could be improved using more sophisticated mass moment of inertia calculations given approximate motor and component weight additions.

Table 2: SEARCH VTOL vehicle configuration properties.

Parameter	Name	Value
Weight	W	17.6 lb
Roll Mass Moment of Inertia	I_{xx}	0.657 slug-ft^2
Pitch Mass Moment of Inertia	I_{yy}	0.517 slug-ft^2
Yaw Mass Moment of Inertia	I_{zz}	0.963 slug-ft^2
Mass Product of Inertia	I_{xz}	0.080 slug-ft^2
Wing Area	S	7.32 ft^2
Wingspan	b	6.97 ft
Aspect Ratio	AR	6.64
Wing Loading	W/S	2.41 lb/ft^2
Motor Power	P	400 W (per motor)
Propeller Diameter	d	12.0 in

5.1 Control Allocation

The control deflection conventions for the control surfaces employed on the SEARCH VTOL vehicle follow conventional fixed-wing aircraft control. Positive deflections are considered to be trailing edge down and trailing edge right for control surfaces. The control effectors are deflected based on the incoming control group commands (see Fig. 2) lift propulsor RPM ($LPRPM_{cmd}$), main propulsor RPM ($MPRPM_{cmd}$), $\theta_{nac,cmd}$, lat_{cmd}, lon_{cmd}, and dir_{cmd}. Each on-axis command is a normalized command spanning $[-1, +1]$. The control surfaces are mapped as follows:

$$\delta_{a,L-R} = \pm\,(lat_{cmd})\,\delta_{a,max}\ , \qquad \delta_{a,max} = 30^\circ \tag{67}$$

$$\delta_{e} = \pm\,(lon_{cmd})\,\delta_{e,max}\ , \qquad \delta_{e,max} = 30^\circ \tag{68}$$

$$\delta_{r} = \pm\,(dir_{cmd})\,\delta_{r,max}\ , \qquad \delta_{r,max} = 30^\circ \tag{69}$$

The VTOL configuration also features a set of inboard flaps. These flaps can operate in both manual and autonomous modes. In manual mode, the flap setting is directly commanded such that $\delta_{f,L} = \delta_{f,R} = \delta_{f,cmd}$. This provides direct control of the flaps from the pilot. Autonomous flaps aim to reduce transition speed (V_{TXN}) by deploying when the vehicle is below transition speed on a first-order time constant as follows:

$$\dot{\delta}_{f,cmd} = \frac{1}{\tau_f}\ (\delta_{f,ref} - \delta_f)\ , \qquad \delta_{f,ref} = \begin{cases} 20^\circ & \text{if } V < V_{TXN} \\ 0^\circ & \text{otherwise} \end{cases} \tag{70}$$

This provides a smooth addition or reduction of the flaps as the vehicle transitions between flight modes. This logic has also been employed in prior work by Comer et al. [20] with success during transitions of a lift-plus-cruise subscale vehicle to provide increased aerodynamic lift during transition.

The propulsors are used to provide attitude stabilization during vertical and transition flight using the normalized commands and common RPM elements through a control allocation, or mixing matrix, as:

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{Bmatrix} = \begin{bmatrix} 1.0 & 1.0 & 1.0 & -0.50 \\ 1.0 & 1.0 & 0.50 & 1.0 \\ 1.0 & -1.0 & 0.50 & -1.0 \\ 1.0 & -1.0 & 1.0 & 0.50 \end{bmatrix} \begin{Bmatrix} N_{0,MP} \\ N_{\phi,MP} \\ N_{\theta,MP} \\ N_{\psi,MP} \end{Bmatrix} \tag{71}$$

$$\begin{Bmatrix} N_5 \\ N_6 \\ N_7 \\ N_8 \end{Bmatrix} = \begin{bmatrix} 1.0 & 1.0 & -1.0 & -0.50 \\ 1.0 & 1.0 & -0.50 & 1.0 \\ 1.0 & -1.0 & -0.50 & -1.0 \\ 1.0 & -1.0 & -1.0 & 0.50 \end{bmatrix} \begin{Bmatrix} N_{0,LP} \\ N_{\phi,LP} \\ N_{\theta,LP} \\ N_{\psi,LP} \end{Bmatrix} \tag{72}$$

where N_0 represents common RPM components for the main propulsors (MP) and lift propulsors (LP), N_ϕ provides differential thrust for lateral commands, N_θ provides differential thrust for longitudinal commands, and N_ψ provides differential thrust for directional commands. The elements of the mixing matrices are not designed to be optimal, but ensure no cross-coupling during on-axis control commands.

The differential thrust terms N_ϕ, N_θ, and N_ψ are defined by a differential RPM quantity and a wash-in/wash-out parameter for each term as follows:

$$N_\phi = \Delta N_\phi lat_{cmd} \zeta_\phi \ , \qquad N_\theta = \Delta N_\theta lon_{cmd} \zeta_\theta \ , \qquad N_\psi = \Delta N_\psi dir_{cmd} \zeta_\psi \tag{73}$$

where $\Delta N_\phi = 1000$, $\Delta N_\theta = 2000$, and $\Delta N_\psi = 2000$ for the main and lift propulsors. The wash-in/wash-out parameters are scheduled with nacelle angle θ_{nac} as shown in Table 3. The general control allocation selection rational follows general design concepts outlined in [23].

Table 3: SEARCH VTOL propulsor mappings as a function of nacelle angle (with linear interpolation between breakpoints).

θ_{nac}	90°	80°	70°	60°	50°	20°	0°
$\zeta_{\phi,MP}$	1	1	0	0	0	0	0
$\zeta_{\theta,MP}$	1	1	1	0	0	0	0
$\zeta_{\psi,MP}$	1	1	0	0	0	0	0
$\zeta_{\phi,LP}$	1	1	1	1	1	1	1
$\zeta_{\theta,LP}$	1	1	1	1	1	1	1
$\zeta_{\psi,LP}$	1	1	1	1	1	1	1

5.2 Simulation Model Development

The SIMPL platform was used to develop a simulation model for the SEARCH VTOL configuration. A top-level view of the simulation model is shown in Fig. 6 illustrating core subsystems. The flight control system architecture (housed within the orange *FCS* subsystem) is nearly identical between the SEARCH fixed-wing and SEARCH VTOL models, with differences only in the inverse propulsor model and control allocation subsystems (which are configuration-specific).

5.3 Simulation Results

A series of manually-piloted simulations were performed to verify the FCS architecture. Maneuvers were performed in vertical flight (hover), transition flight (20 kts), and forward flight (45 kts)

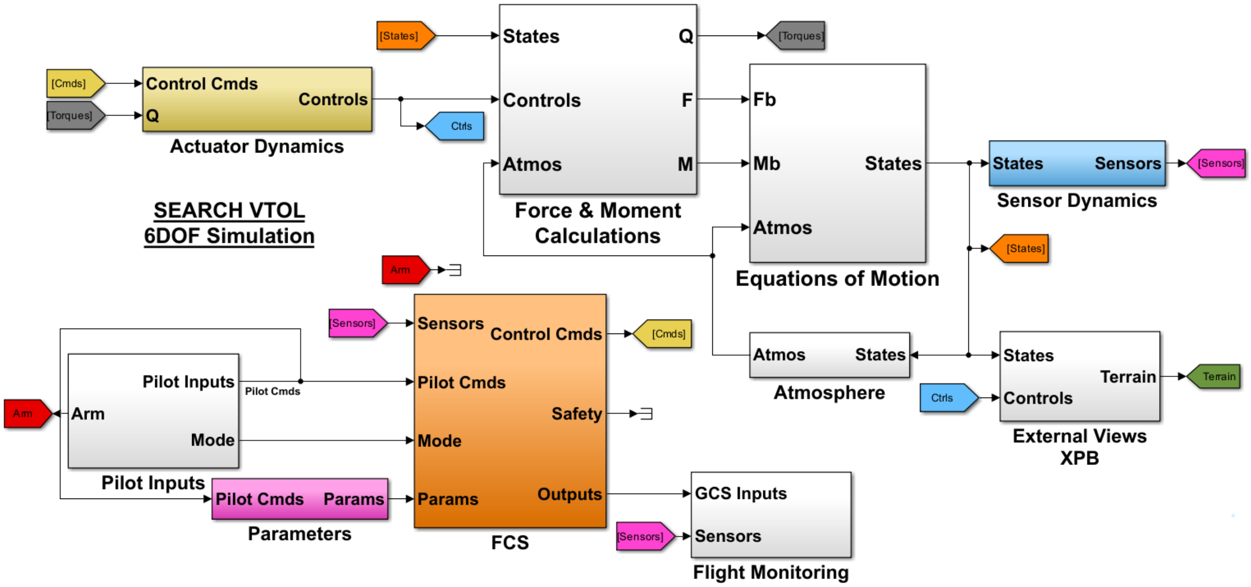


Figure 6: SEARCH VTOL SIMPL Simulink®-based simulation model.

along with full transitions performed during vertiport departures and arrivals. The MATLAB® Aerospace Blockset Von-Karman Wind Turbulence Model (VKTM) [25] was enabled during all maneuvers with the "Continuous Von-Karmen (-q +r)" setting with a 3 m/s wind speed for low-altitude winds. The turbulence was set to severe settings for all maneuvers to monitor disturbance rejection and control activity. The VKTM, in low speed conditions, was noted to provide very little disturbance to the vehicle. As a result, hover conditions appear to have no turbulence. A control-equivalent turbulence input (CETI) model is recommended to be used in future works.

Note that both in the forthcoming simulation and flight test results, the command signals will be seen to "spike" in certain instances. These spikes are a result of autonomous holds engaging and disengaging. The use of integrators to "shadow" states (see Sec. 4.1) adds a fast, but noticeable, dynamic response during the engagement period. This logic, though not meaningfully impacting the stability of the vehicle, could be revisited in future works.

5.3.1 Vertical Flight Maneuvers

A series of piloted maneuvers for each axis were performed in vertical flight (hover). The first maneuver includes a vertical axis multistep input, commanding a 200 fpm climb rate followed by a 200 fpm descent rate. The commands were held for approximately 10 seconds to allow the TFC controller adequate time to settle. The results from the maneuver are shown in Fig. 7. The maneuver begins at around $t = 5$ sec, and the command is met with minimal overshoot. The vehicle picks up a minor forward velocity, which is countered with a slight nose-up pitch command to maintain the position hold of the vehicle. The descent is also met promptly with no major overshoots or oscillations. After the maneuver is complete, the altitude hold re-engages and the vehicle stabilizes at the target altitude.

A forward acceleration pulse was performed to demonstrate the proposed feed-forward acceleration term K_A^{ff}, proposed initially by Comer and Chakraborty [8]. This pulse involved a piloted acceleration command followed by a centering of the stick, resulting in a dissipation of the velocity back to a hover condition, followed by the position hold engaging to maintain the vehicle once the speed reduced below the position hold engagement threshold. The results from this maneuver are

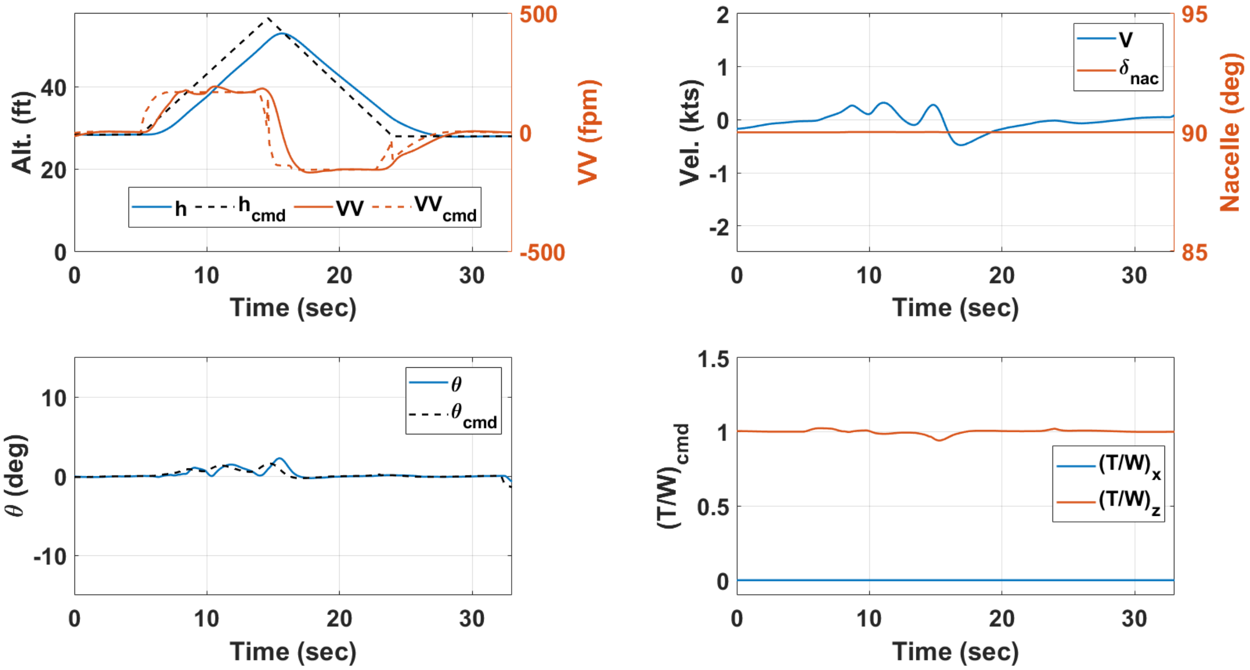


Figure 7: SEARCH VTOL response to a vertical axis multistep input.

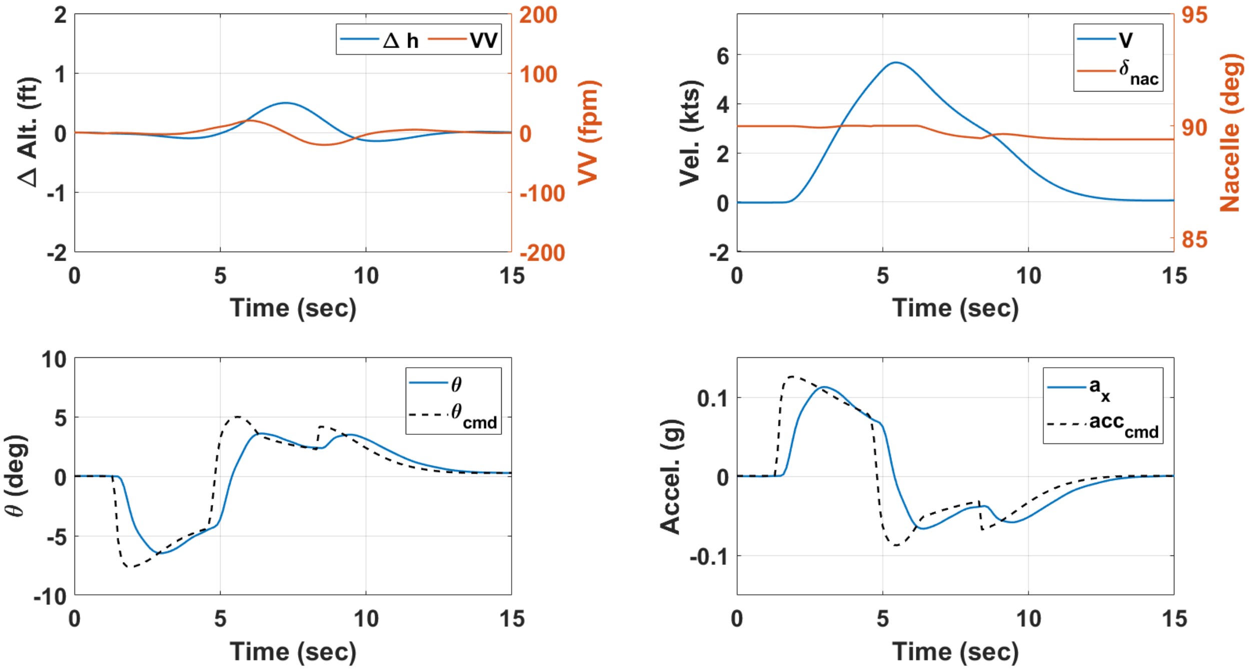


Figure 8: SEARCH VTOL response to an acceleration pulse command.

shown in Fig. 8. Acceleration results are shown with the component of gravitational acceleration removed for all maneuvers. This isolates the vehicle's acceleration to only include the body-axis acceleration values that result in a speed change.

The acceleration pulse is maintained for around 4 seconds with a magnitude of around 0.12 g. The feed-forward gain K_A^{ff} commands an immediate pitch attitude which is met by the inner-loop EMF controllers, resulting in a velocity increase of just over 5 kts. During this time, the nacelles move slightly forward to aid in acceleration, but otherwise maintaining the vertical flight orientation. Once the pulse is removed, the vehicle begins to dissipate the speed, acting in the vertical flight acceleration regime to bring the vehicle back to hover (see Sec. 4.1). Throughout the maneuver, the altitude was held to within ±0.5 ft.

To assess the lateral axis, a lateral velocity multistep command was performed. Lateral inputs from the pilot command lateral velocity throughout the flight envelope, and these commands are met with direct bank angle commands in low-speed flight conditions. The response of the VTOL aircraft to a 2.5 kt lateral velocity multistep input is shown in Fig. 9.

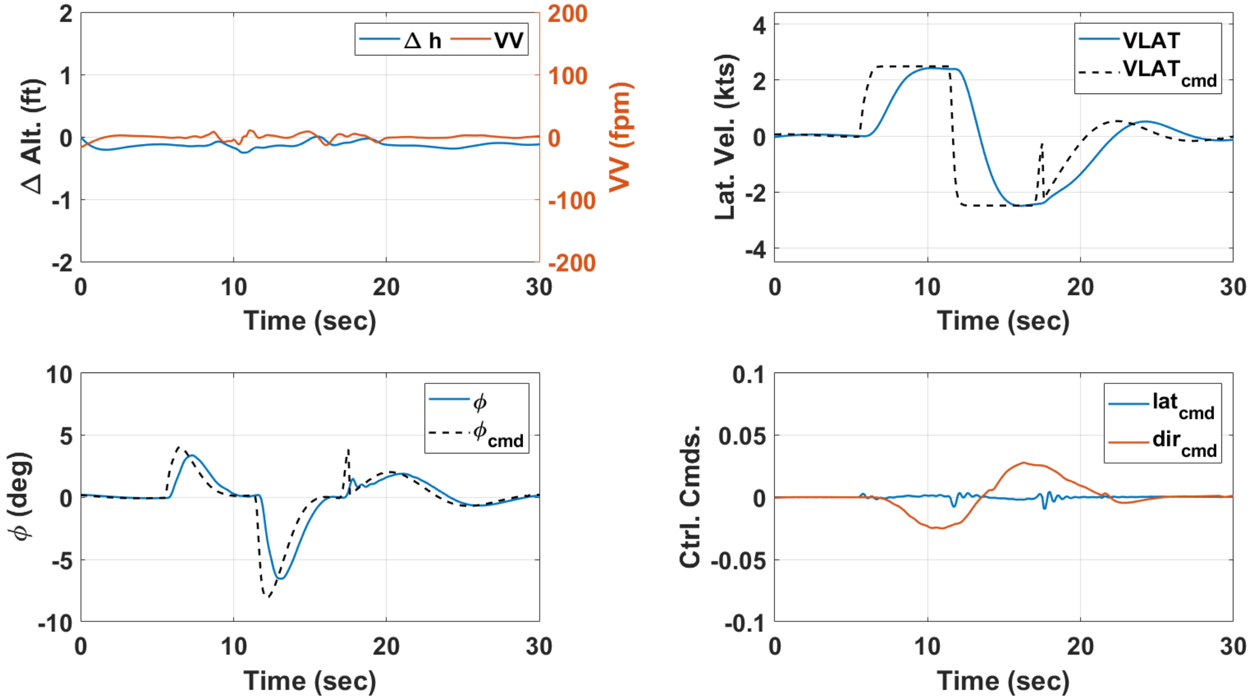


Figure 9: SEARCH VTOL response to a lateral velocity multistep command.

The directional axis responsiveness was assessed through a yaw rate multistep command, shown in Fig. 10. The lack of a feed-forward directional axis command does not show a noticeable degradation in performance in comparison to the pitch and roll axes, meeting commands in a comparable time with favorable performance.

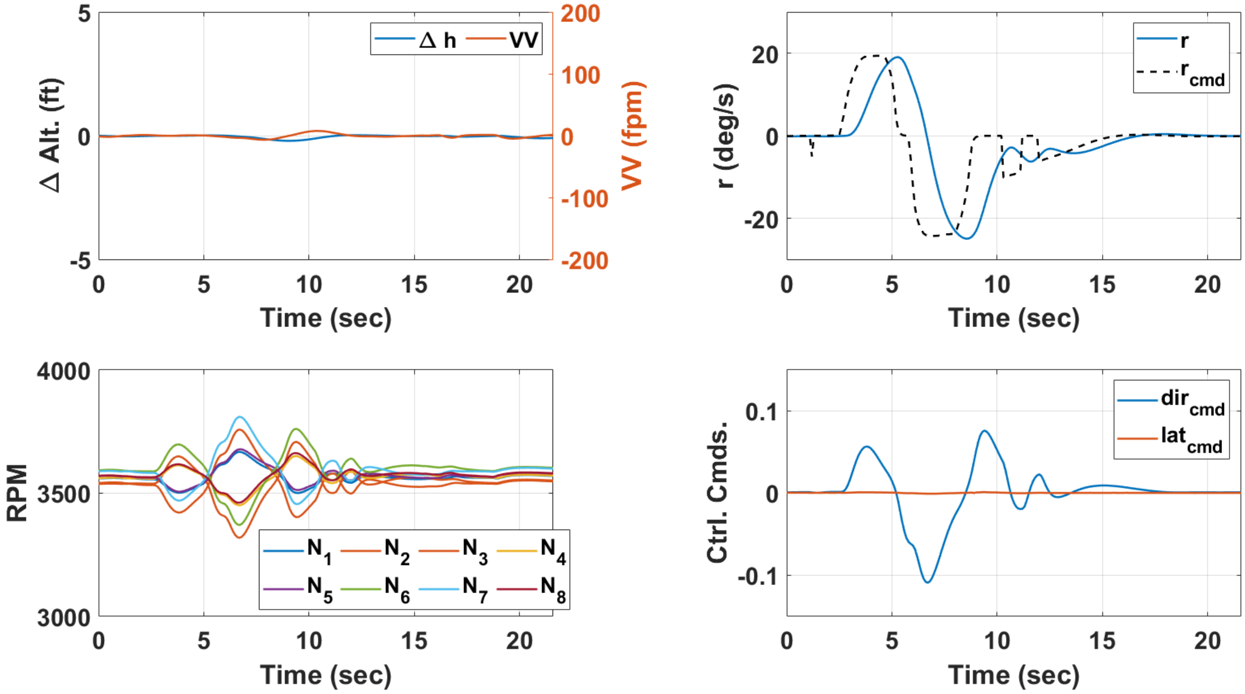


Figure 10: SEARCH VTOL response to a yaw rate multistep command.

5.3.2 Transition Mode Flight Maneuvers

To highlight transition flight maneuverability, on-axis multistep and pulse inputs were performed at 20 kts via pilot inputs. Figure 11 shows the response of the vehicle to a vertical velocity multistep command of 300 fpm. The multistep input begins at $t = 15$ sec and is sustained for around 10 seconds in each axis. The commands are met with minor overshoot, and the turbulence is notable but otherwise rejected successfully throughout the maneuver. The nacelle angle θ_{nac} is seen to rise by around 5° before reducing back to the initial value of around 76°. The longitudinal command lon_{cmd} decreases slightly during the climb (nose-down pitching moment), counteracting the nose-up pitch tendency due to the download of the horizontal stabilizer. The opposite trend is seen during the descent as the longitudinal command increases briefly before returning to the initial value. The speed deviations are maintained below 1 knot throughout the maneuver.

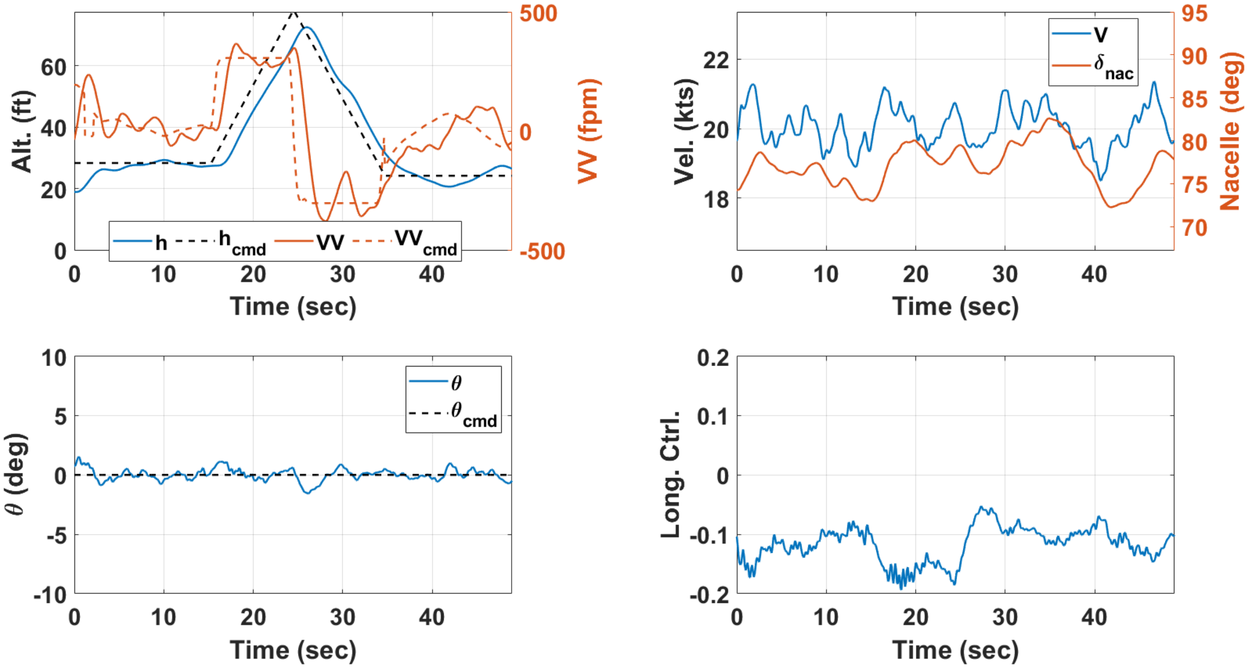


Figure 11: SEARCH VTOL response to a vertical velocity multistep input.

A manual velocity command change was performed from 20 kts to 25 kts to assess the response of the TFC to speed changes during transition flight. The results from this step are shown in Fig. 12. The pilot commands velocity rate in this flight regime, setting a velocity command for the PCM controller to generate an acceleration command from. The maneuver occurs with an altitude hold engaged, and shows minor deviations in altitude (±2 ft) throughout the acceleration. The deck is maintained level (see Sec. 4.2) and the TFC uses a combination of thrust control and nacelle movement to achieve the desired speed. The nacelle angle moves from 75° to around 63° between the 5 kt speed change, meeting the velocity response with minor overshoot and highlighting the success of the TFC to decouple flight path and velocity.

A turn rate multistep input was performed to showcase transition maneuvering performance, as shown in Fig. 13. At this speed, the vehicle is above the crossover threshold V_x described in Sec. 4.3. Above this speed, turn rate commands $\dot{\psi}_{cmd}$ are achieved through coordinated turns and sideslip correction. The coordinated turn commands are met with desirable response characteristics and showed minor altitude deviations, highlighting the success of the turn compensation of the TFC.

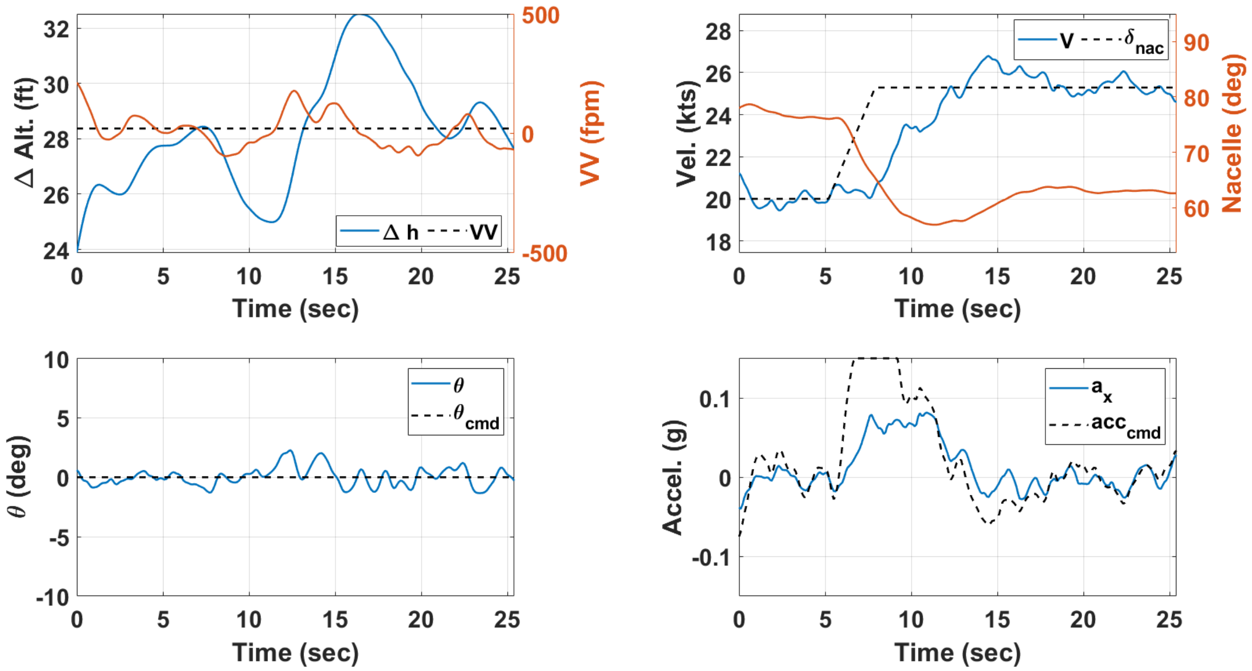


Figure 12: SEARCH VTOL response to a velocity rate pulse.

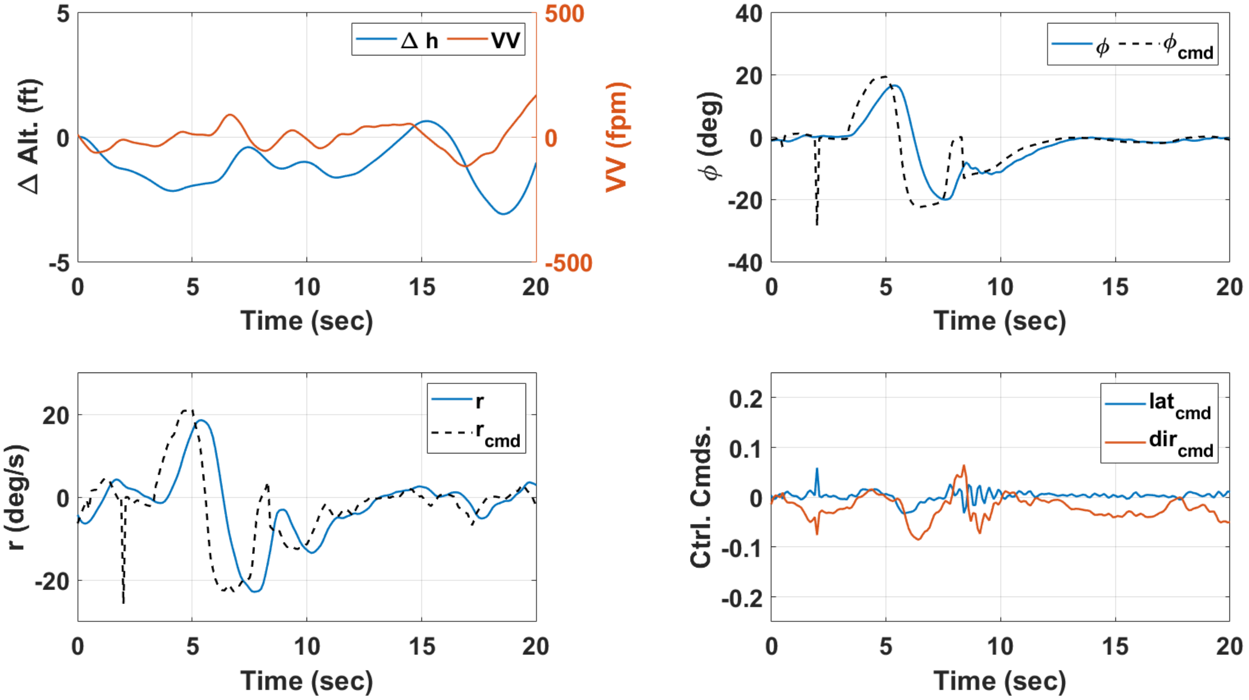


Figure 13: SEARCH VTOL response to a turn rate multistep command.

5.3.3 Forward Flight Maneuvers

Forward flight (lift propulsors stopped, nacelle in forward flight position) maneuvers were performed at 45 kts, showing the modified TECS-based response to a vertical velocity multistep command in Fig. 14. The addition of the feed-forward term shows improved response times when compared to the original TECS formulation, but also introduces additional control activity and small oscillations in pitch and yaw rate. The 300 fpm vertical velocity commands are met with minimal velocity deviations, illustrating that the addition of the feed-forward gain did not introduce undesirable coupling effects reported in some studies [33].

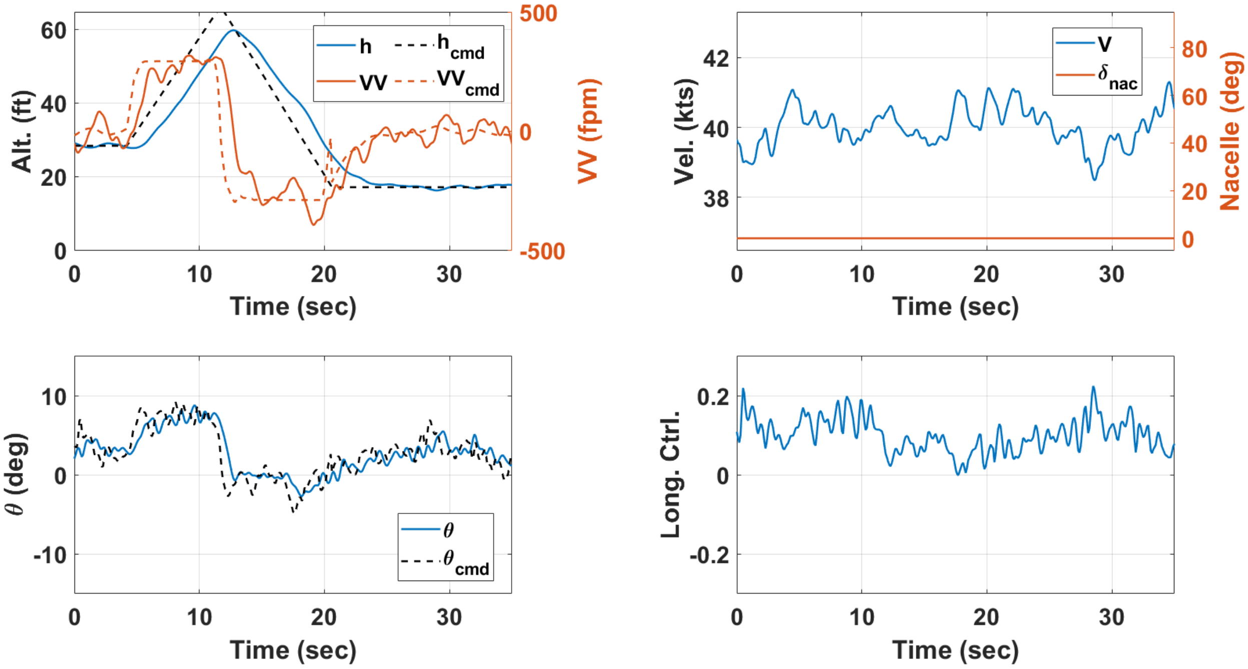


Figure 14: SEARCH VTOL response to a vertical velocity multistep command in forward flight.

The response of the vehicle from a velocity step from 40 kts to 45 kts is shown in Fig. 15. The commanded velocity is achieved with no overshoot, and the compensatory nose-down pitch attitude from K_{trim} is shown to slightly overcompensate for the acceleration, causing a 4 ft altitude reduction which is then recovered by the altitude controller. Throughout the maneuver, a prominent pitch oscillation is observed. This oscillation also appears to couple with the yaw axis and will be investigated in future studies.

A turn rate multistep input was performed in forward flight to assess lateral/directional control responses. The results are shown in Fig. 16, with a set of 30° turns performed at 45 kts. The EMF inner-loop controllers meet the commanded bank angle and yaw rate commands with favorable response characteristics, and the turn compensation within TFC performs well in maintaining minimum altitude deviations throughout the multistep command. The yaw rate oscillations observed in the previous plot remain and will be investigated in future studies.

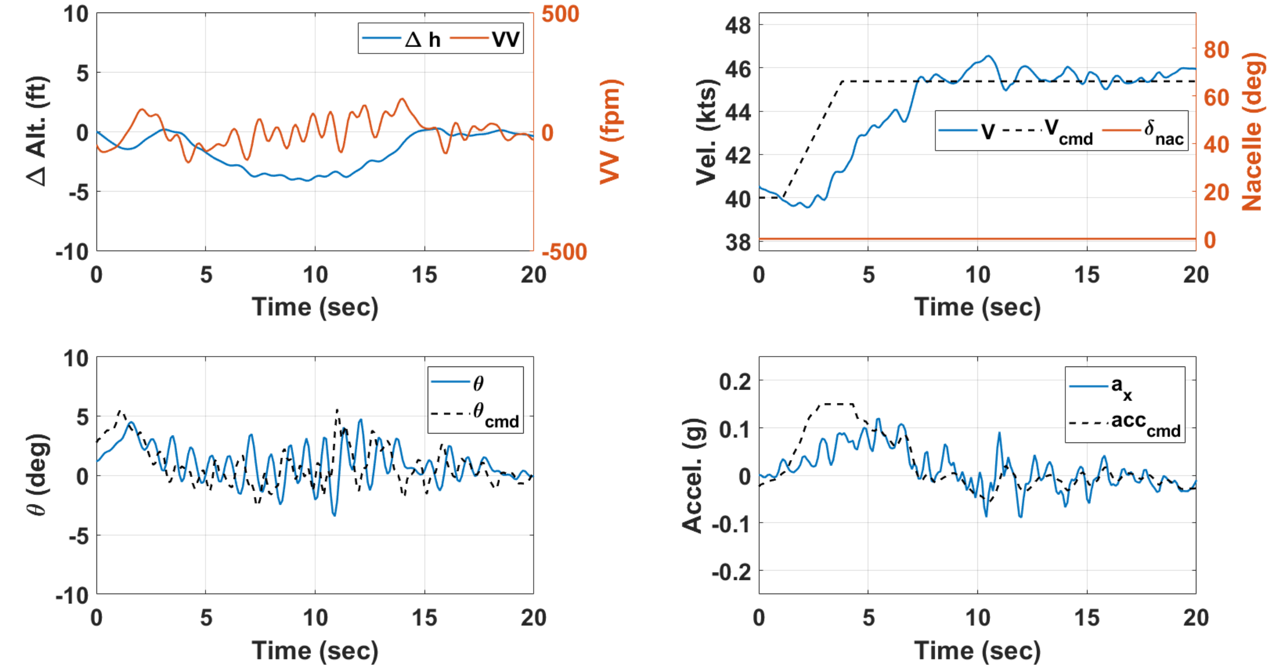


Figure 15: SEARCH VTOL response to a velocity step command in forward flight.

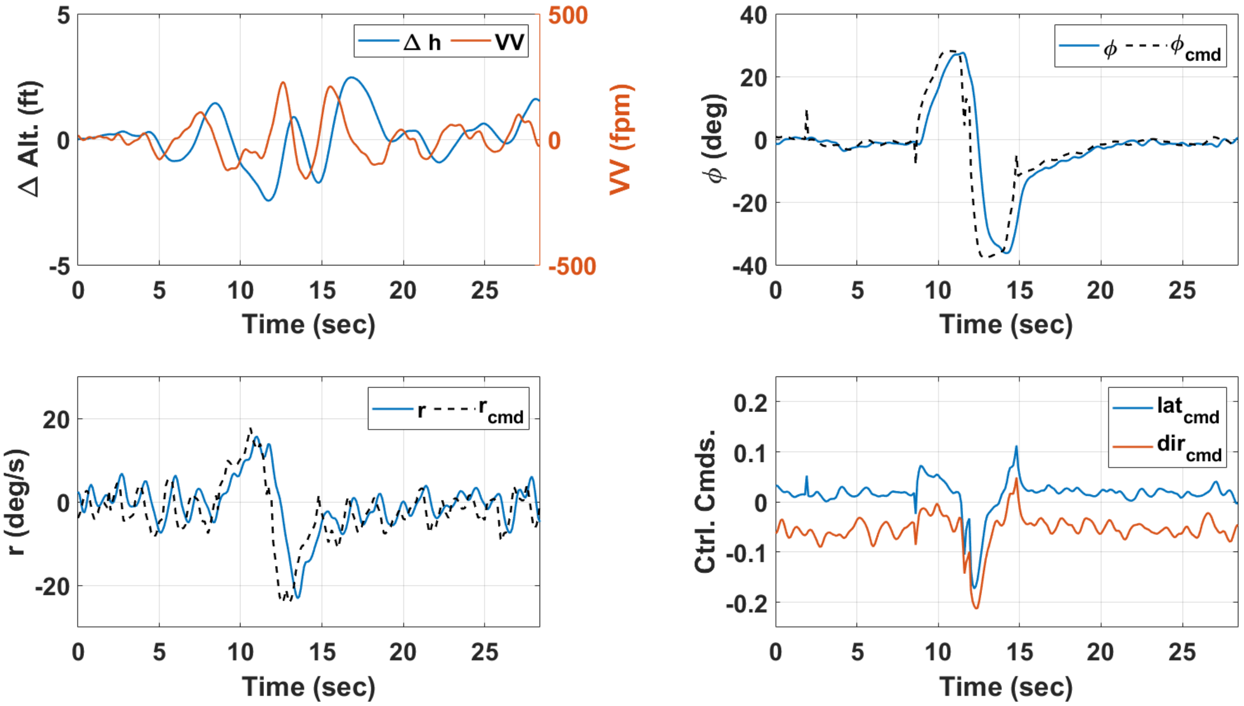


Figure 16: SEARCH VTOL response to a turn rate multistep command in forward flight.

5.3.4 Transition Flights

Outbound (accelerating) and inbound (decelerating) transitions were performed to assess the TFC's ability to transition the SEARCH VTOL in the current design. The transition maneuvers were manually piloted with altitude hold engaged to assess longitudinal transients during transition.

The results from the outbound transition are shown in Fig. 17. An acceleration command is held, eventually transitioning into velocity rate command, until the commanded speed was at 45 kts. Throughout the departure, the heading hold and altitude hold remained engaged. The vehicle begins the acceleration with nose-down pitch based on hover flight operations, transitioning into transition flight at $t = 12$ sec. At this point, the deck-leveling begins, slowly diminishing the pitch command $\theta_{cmd} = 0°$. The nacelles begin moving forward to meet the required acceleration, while modulating the vertical thrust requirements as well. From $t = 20$ sec until $t = 28$ sec, the nacelle angle begins to naturally induce a diminishing thrust to the lift propulsors, shown as the divergence of MPRPM and LPRPM terms. In this same period, the longitudinal control command lon_{cmd} begins increasing in a nose-down command, to further increase the LPRPM and balance the pitching moments before the elevator becomes effective enough. From $t = 28$ sec onward, the vehicle is in forward flight operations, and the nacelle angle is slowly reduced to forward flight position $\theta_{nac,cmd} = 0°$. TFC Mode definitions are described in Sec. 4.2. During the final transition period (from $t = 28$ sec to $t = 33$ sec), the vehicle is seen to rise around 10 ft, indicating a potential need to increase the final transition rate or decrease the transition starting speed.

The results from the inbound transition are shown in Fig. 18. Beginning from 45 kts (forward flight), a negative velocity rate command is held, eventually transitioning into a deceleration command in hover flight, until the commanded speed was at 0 kts. Throughout the flight, the heading hold and altitude hold remained engaged. The vehicle begins the deceleration with nose-up pitch to counter the reducing speed, transitioning into transition flight at $t = 12$ sec. At this point, the deck-leveling begins, slowly diminishing the pitch command $\theta_{cmd} = 0°$. The nacelles quickly begin to respond on account of the descent that begins. From $t = 12$ sec until $t = 20$ sec, the nacelle angle begins to naturally wash-in thrust to the lift propulsors, shown as the convergence of MPRPM and LPRPM terms. In this same period, the vehicle arrests the descent rate and begins to re-capture the commanded altitude. From $t = 28$ sec to $t = 45$ sec, the vehicle slows in transition flight naturally due to drag. From $t = 45$ sec onward, the vehicle is in hover flight operations, and the pitch attitude aids in the final deceleration. During the transition period (from $t = 12$ sec to $t = 20$ sec), the vehicle is seen to sink around 30 ft, indicating a need to potentially implement a spool-up logic or fixed-rate nacelle increase to improve the responsiveness of the vehicle in this flight regime.

The Transition Flight Controller successfully performed inbound and outbound transitions of the vectored-thrust configuration, but has room for improvement during transition segments. The TFC design investigated in this work contains three flight modes and aimed to use geometric allocation of thrust based on the nacelle angle to achieve commands. However, there may be a need to introduce a separate logic during the transition process to account for the nonlinear complexities within this region of flight.

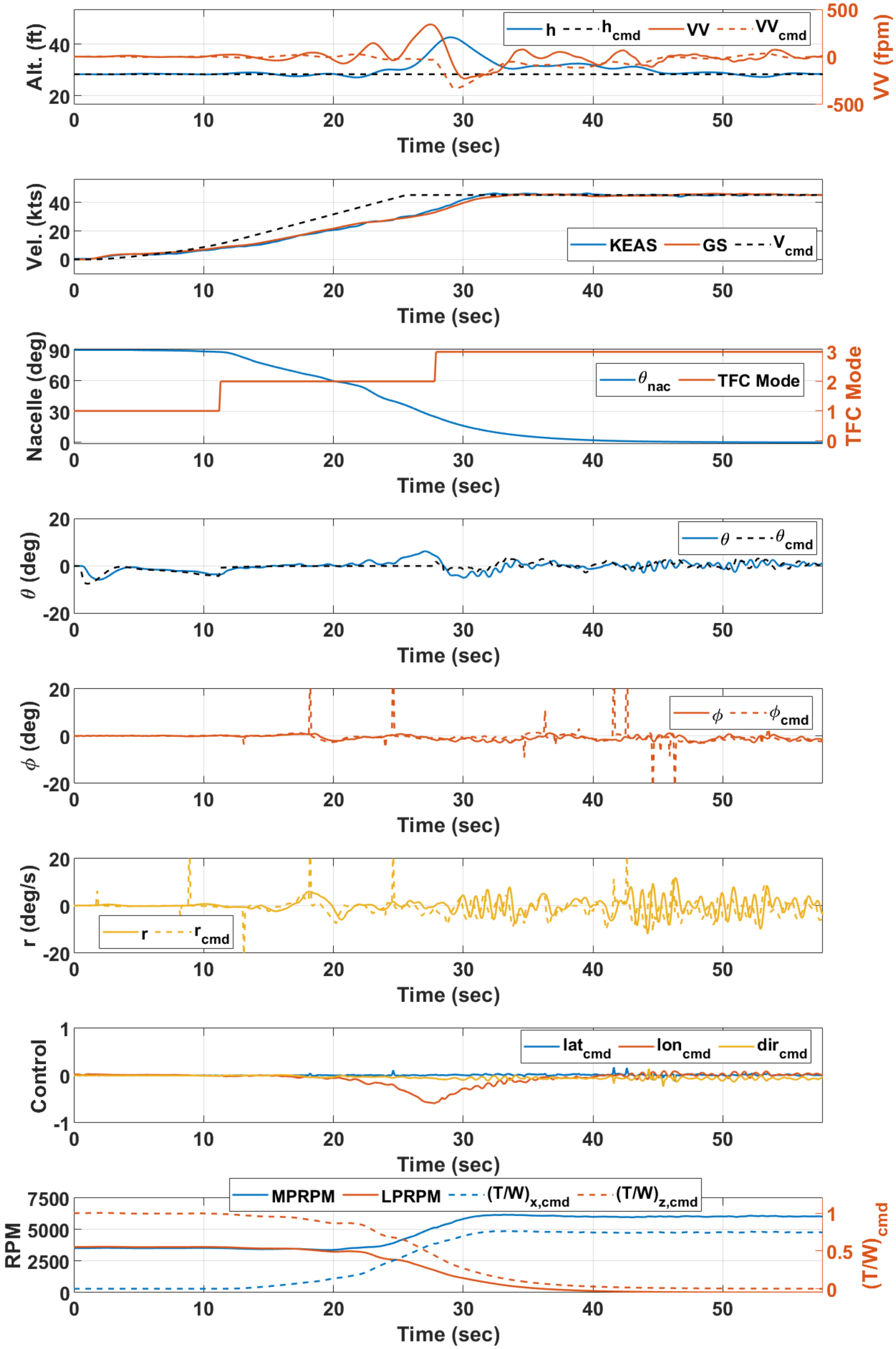


Figure 17: SEARCH VTOL outbound transition from 0 kts to 45 kts.

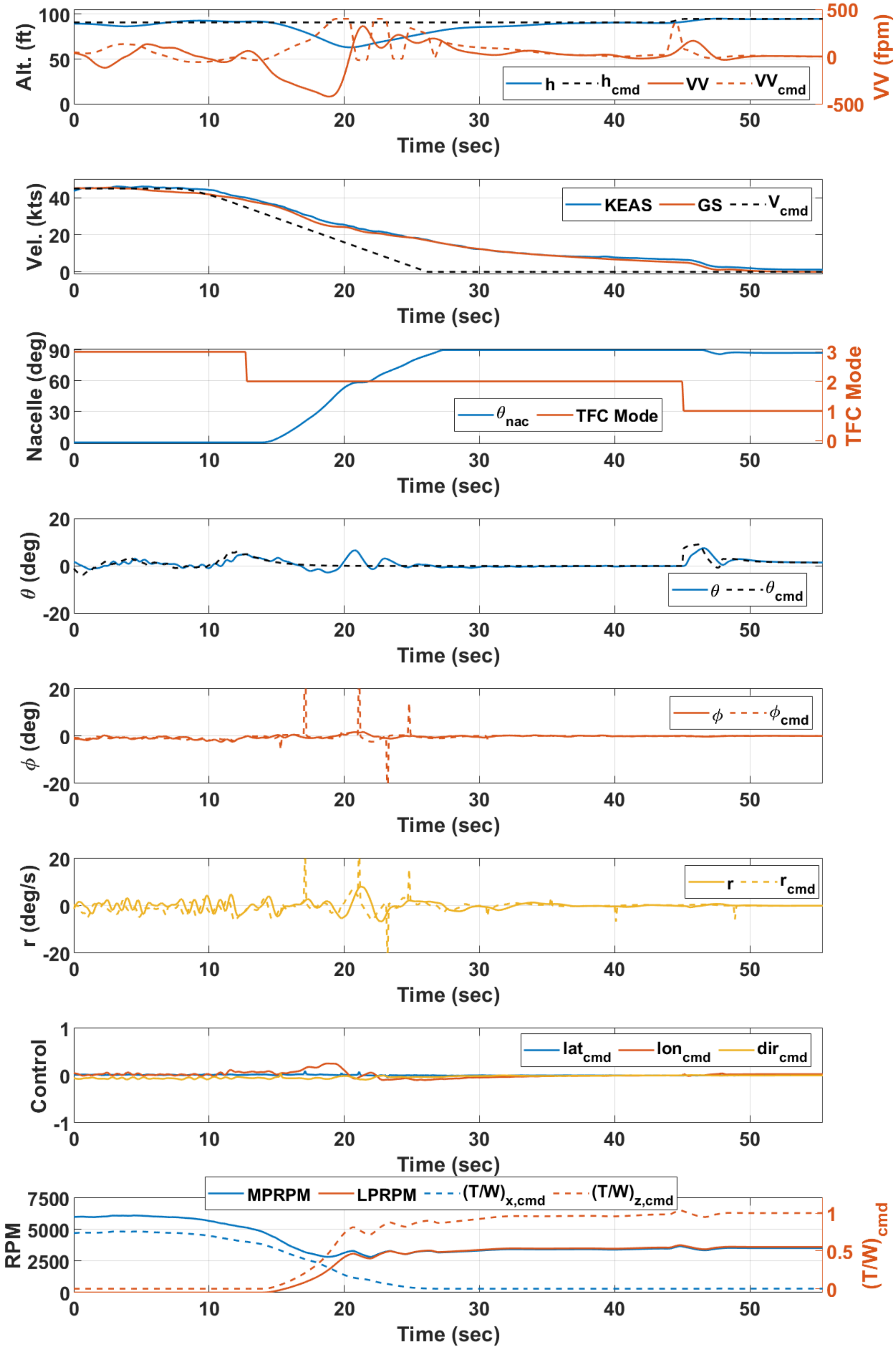


Figure 18: SEARCH VTOL inbound transition from 45 kts to 0 kts.

6 Hardware Integration for Flight Testing

The flight control system was implemented onto a Pixhawk® Cube Orange+ flight controller for SEARCH flight testing. For the IMPACT vehicle, the controller was implemented on a Pixhawk® Cube Blue. This process involves the creation of a *Hardware Model*, which is uploaded to the Pixhawk® using the MathWorks® UAV Toolbox and its Support Package for PX4 Autopilots [25]. This toolbox has been used in several applications by the authors [1, 2, 18–20, 23]. The flight testing process presented in this work uses the MATLAB®/Simulink® R2024a release.

Appendix A presents several minor modifications to the PX4 source code to achieve the flight test objectives described in this report. A custom telemetry stream was developed to send signals internal to the Simulink® flight control logic to a custom ground control station (GCS) in real time. The process for enabling the telemetry stream is presented in Appendix A.1 along with a complete list of telemetry states sent to the GCS from the vehicle in Table A1. Additionally, custom parameters were built into the firmware to allow live changes to flight control system parameters and gains. A complete list of tunable parameter groupings for the SEARCH vehicle is shown in Table 4. Integration of the custom parameters into the FCS and the necessary firmware modifications are described in Appendix A.2. Finally, the FCS model developed in Simulink® required additional flash memory to successfully deploy the algorithm to the Pixhawk® flight controller. The process for reducing the flash memory space occupied by PX4 is presented in Appendix A.3 following the guidance in [42].

Table 4: SEARCH vehicle tunable parameters grouped by flight control system models

Grouping	Quantity	Description
EMF - Lateral	10	EMF lateral axis parameters/gains
EMF - Longitudinal	13	EMF longitudinal axis parameters/gains
EMF - Directional	13	EMF directional axis parameters/gains
TFC	24	TFC parameters/gains
LATCS	14	LATCS parameters/gains
IPPM	8	IPPM parameters
FCS	5	General FCS parameters (e.g., flaps, mass)
PCM Lateral	9	PCM lateral axis parameters
PCM Longitudinal	10	PCM longitudinal axis parameters
PCM Directional	9	PCM directional axis parameters
PCM Acceleration	10	PCM acceleration axis parameters
PCM Autopilot	10	PCM autopilot (hold) parameters
PCM Position Hold	10	PCM position hold parameters/gains
Servo Mixing	6	Servo mixing and sign convention parameters
Sensors	9	Sensor parameters (e.g., LiDAR, Airspeed)
Geofence	4	Geofence coordinates and parameters
Safety	5	Flight fail-safe and safety parameters
Total	169	

The SIMPL platform is designed to facilitate easy conversion from simulation to hardware-ready Simulink® models. The SEARCH *Hardware Model* is shown in Fig. 19, featuring an identical flight control system subsystem housed within the orange *FCS* block. The hardware model houses a set of core subsystems developed based on prior work by Comer [18–20] that are detailed below:

- **Sensors:** This subsystem is responsible for reading data streams from the Pixhawk® via

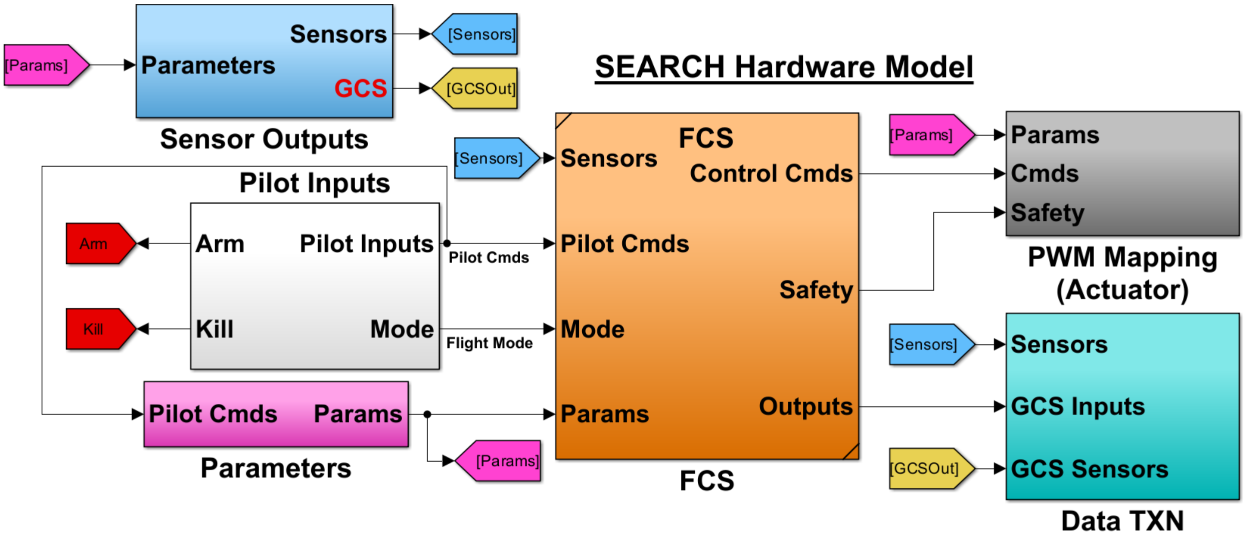


Figure 19: SEARCH hardware-ready Simulink® model to be uploaded to the Pixhawk® controller.

uORB message topics and performing any necessary filtering or sensor reconstruction required to provide the FCS with the sensor states.

- **Pilot Inputs:** Pilot commands from the transmitter are captured here and mapped as required by the FCS. Incoming channels from the transmitter are then used as required to develop logical switches (e.g., Arm, Terminate, Mode, Flaps, etc.) and normalized pilot commands p_{lat}, p_{ver}, p_{dir}, and p_{acc}.

- **Flight Control System:** The FCS houses the entire flight control system architecture (see Fig. 2). The FCS is a referenced subsystem, ensuring commonality between simulation and hardware Simulink® models.

- **PWM Mapping:** This module converts raw control effector commands (control surface deflections and RPM commands) into pulse width modulation (PWM) signals $\in [1000, 2000]$ to be sent to PWM-driven hardware items (servos, ESCs). A linear mapping between commanded position and PWM value was used in this work.

- **Data Transmission:** This subsystem is responsible for communication to the GCS. For the SEARCH vehicle, this module sent the desired internal FCS states back to the GCS via the telemetry stream.

7 SEARCH Fixed-Wing Aircraft Simulation and Flight Testing

The SEARCH vehicle, shown in Fig. 20, is a subscale foam general aviation aircraft. The vehicle features seven control effectors, including conventional control surfaces: a set of outboard ailerons, inboard flaps, an elevator, and a rudder. A single tractor propeller is mounted on the nose of the vehicle. Table 5 lists the control effectors of the SEARCH vehicle along with the respective limits.

SEARCH vehicle properties are given in Table 6. The SEARCH moment of inertia tensor elements were assumed to be the same as those determined for a similar aircraft flown in previous work [12, 13]. The inertia tensor elements were estimated using the compound pendulum method [43], as described further in Ref. [44].

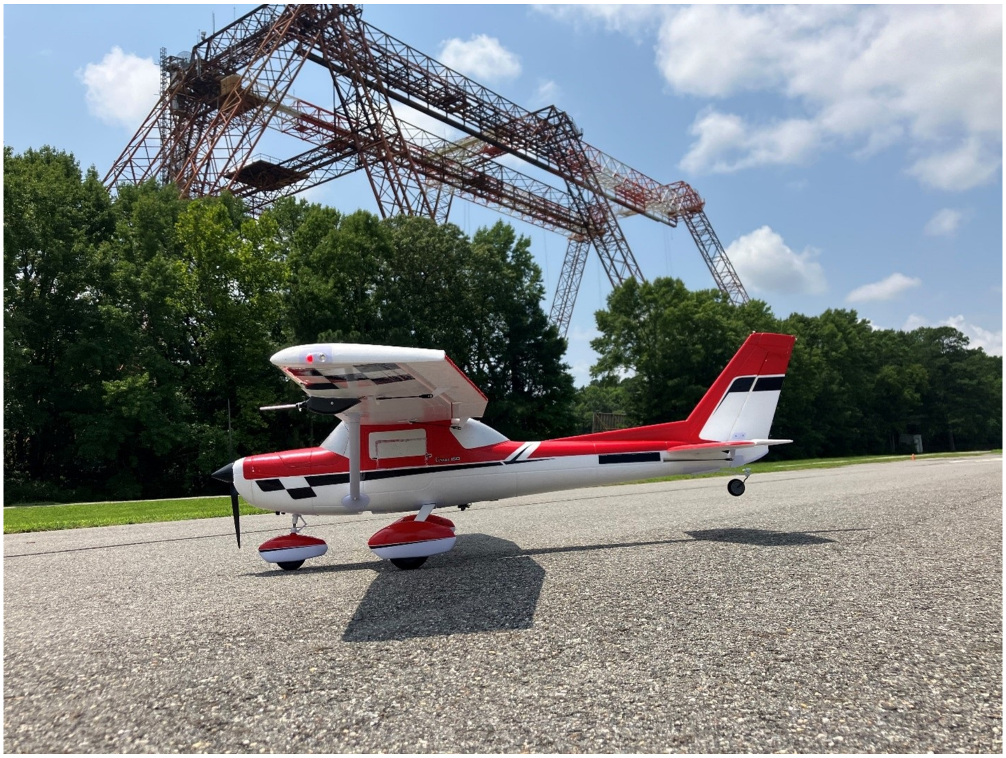


Figure 20: SEARCH vehicle at takeoff site on the CERTAIN range.

Table 5: SEARCH vehicle control effectors.

Effector	Description	Position Limits
$\delta_{a,L}$	Left aileron	$[-30°, +30°]$
$\delta_{f,L}$	Left flap	$[0°, +30°]$
$\delta_{f,R}$	Right flap	$[0°, +30°]$
$\delta_{a,R}$	Right aileron	$[-30°, +30°]$
δ_e	Elevator	$[-30°, +30°]$
δ_r	Rudder	$[-30°, +30°]$
N	Propeller RPM	$[0, 9000]$ RPM

Table 6: SEARCH vehicle configuration properties.

Parameter	Name	Value
Weight	W	10.2 lb
Mass Moment of Inertia	I_{xx}	0.403 slug-ft^2
Mass Moment of Inertia	I_{yy}	0.317 slug-ft^2
Mass Moment of Inertia	I_{zz}	0.591 slug-ft^2
Mass Product of Inertia	I_{xz}	0.049 slug-ft^2
Wing Area	S	7.32 ft^2
Wingspan	b	6.97 ft
Aspect Ratio	AR	6.64
Wing Loading	W/S	1.39 lb/ft^2
Motor Power	P	1350 W
Propeller Diameter	d	15.0 in

7.1 Control Allocation

The control deflection nomenclature scheme employed on the SEARCH vehicle follows conventional fixed-wing aircraft control. Positive deflections are considered to be trailing edge down and trailing edge right for control surfaces. The control effectors are deflected based on the incoming control group commands RPM_{cmd}, lat_{cmd}, lon_{cmd}, and dir_{cmd}. Each on-axis command is a normalized command spanning $[-1, +1]$. The control surfaces are mapped as follows:

$$\delta_{a,L} = -\delta_{a,R} = \pm \left(lat_{cmd}\right) \delta_{a,max} \ , \qquad \delta_{a,max} = 30^\circ \tag{74}$$

$$\delta_e = \pm \left(lon_{cmd}\right) \delta_{e,max} \ , \qquad \delta_{e,max} = 30^\circ \tag{75}$$

$$\delta_r = \pm \left(dir_{cmd}\right) \delta_{r,max} \ , \qquad \delta_{r,max} = 30^\circ \tag{76}$$

SEARCH also features a set of inboard flaps. These flaps can operate in both manual and autonomous modes. In manual mode, the flap setting is directly commanded such that $\delta_{f,L} = \delta_{f,R} = \delta_{f,cmd}$. This provides direct control of the flaps from the pilot. Autonomous flaps aim to reduce high angle of attack operating conditions by slowly offsetting maintained pitch attitudes as follows:

$$\dot{\delta}_{f,cmd} = K_f \ \theta \ , \qquad \delta_{f,cmd} \in [0^\circ, 20^\circ] \tag{77}$$

This provides a controlled integration of the flap setting, also providing the benefits of a low-pass filter to the pitch attitude through the selection of the aggressiveness gain K_f.

7.2 Flight Modes

The FCS on the SEARCH vehicle can operate in three modes, selected by the pilot. *Mode* 1 represents open-loop control, mapping pilot inputs directly to the four commands for the *Control Allocation* subsystem (i.e., “stick-to-surface”). *Mode* 2 provides inner-loop attitude stabilization, commanding RPM, bank angle, pitch attitude, and yaw rate directly with the EMF controllers active. *Mode* 3 includes the outer-loop holds and middle-loop longitudinal and lateral controllers. This loop closure is visualized further in Fig. 21, which shows the inceptor-to-effector control system architecture with color-coded output signals based on the current control mode.

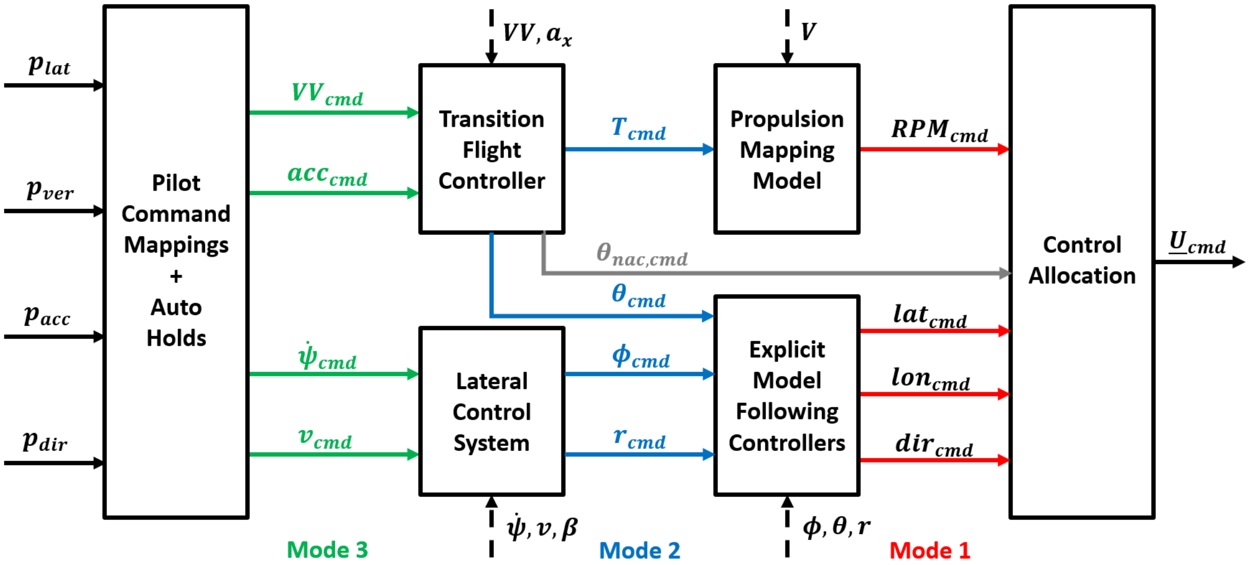


Figure 21: Block diagram of the generalized control architecture with color-coded modes.

7.3 Simulation Model Development

The SEARCH vehicle simulation model was developed using the SIMPL platform and used to simulate and test the proposed FCS architecture described in Sec. 4. The top-level diagram of the SIMPL model for the SEARCH vehicle is shown in Fig. 22, showcasing the numerous subsystems described in Sec. 3. The simulation model was used for pre-flight testing and piloted simulation studies.

The FCS parameters were optimized using a genetic algorithm-based optimization algorithm, designed to optimize the FCS at different levels of loop closure (synonymous with flight modes shown in Fig. 21). This optimization approach is discussed further in works by Comer [22–24] to optimize and flight test several VTOL vehicles, enabling a direct path from simulation to flight test. The FCS gains and parameters used in the flight tests came directly from the simulation parameters.

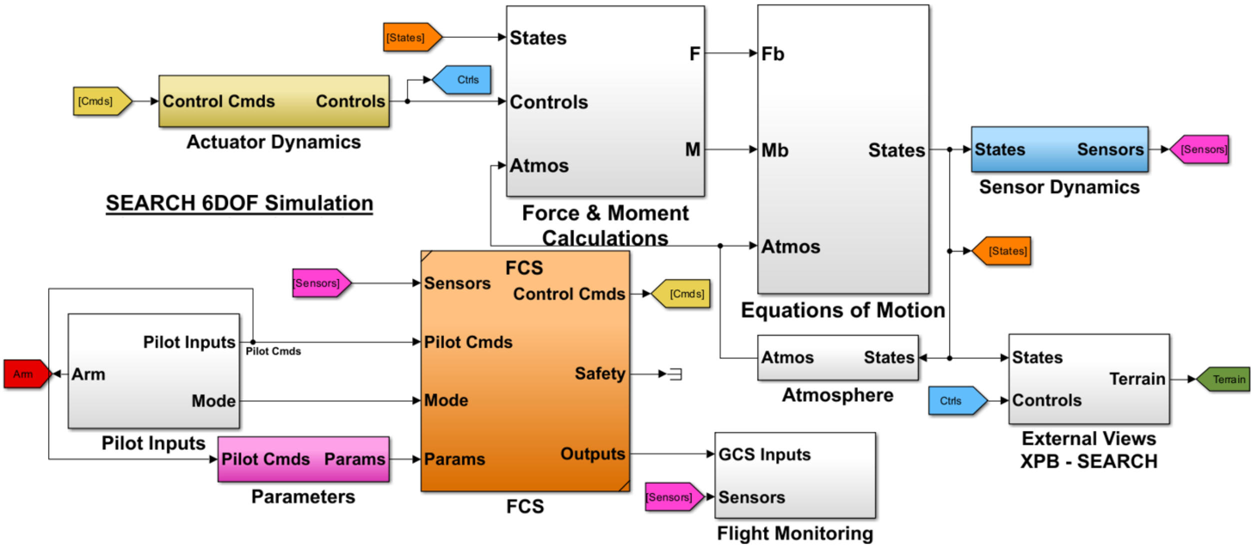


Figure 22: SEARCH SIMPL Simulink®-based simulation model.

The SEARCH vehicle visual model was developed using OpenVSP [41] based on the vehicle geometry. This vehicle was then exported into Blender [45] to be animated following the method described in Appendix A to create an X-Plane visualization [26]. The SEARCH OpenVSP model and SEARCH X-Plane model are shown in Fig. 23.

Prior to flight testing, piloted simulations were performed to assess the vehicle's handling qualities and familiarize the pilot with the command architecture between flight modes. Simulations were performed on a standard desktop setup with monitors showing the Simulink® ground control station panel and X-Plane external visualization.

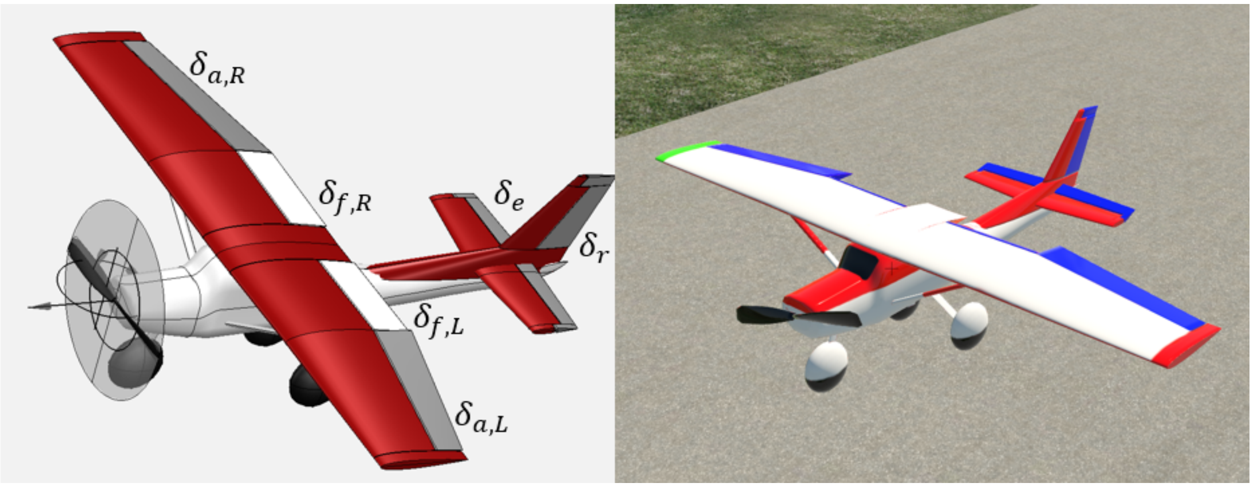


Figure 23: SEARCH OpenVSP and X-Plane external visualization models.

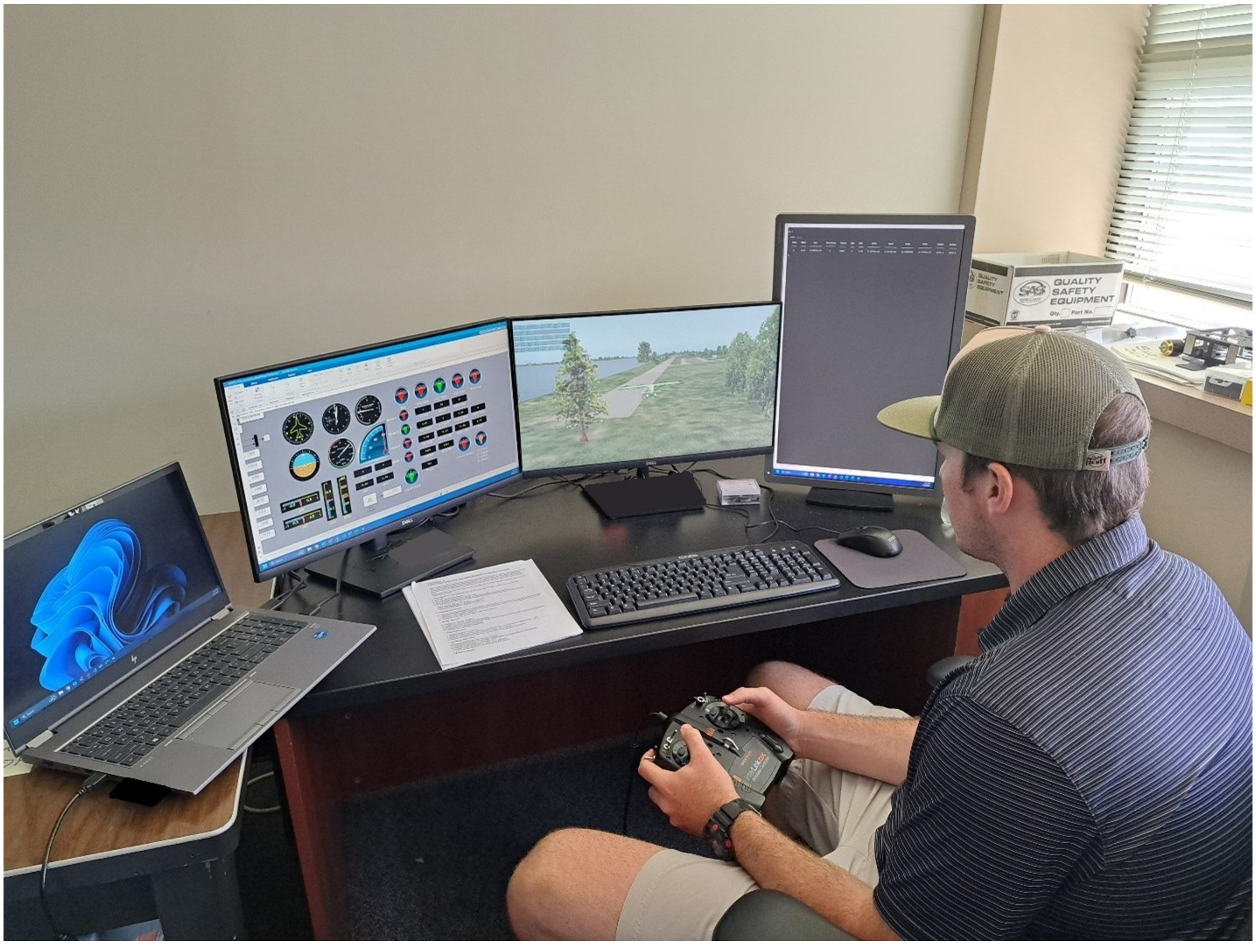


Figure 24: Simulation session with a NASA LaRC UAS pilot conducted prior to flight testing.

7.4 Flight Testing

Flight testing for the SEARCH vehicle was performed at the NASA LaRC City Environment Range Testing for Autonomous Integrated Navigation (CERTAIN) range, shown with the bounding geofence used for SEARCH in Fig. 25. The CERTAIN range features a 1000 ft paved runway and a large space for UAV flight testing on-site at NASA LaRC.

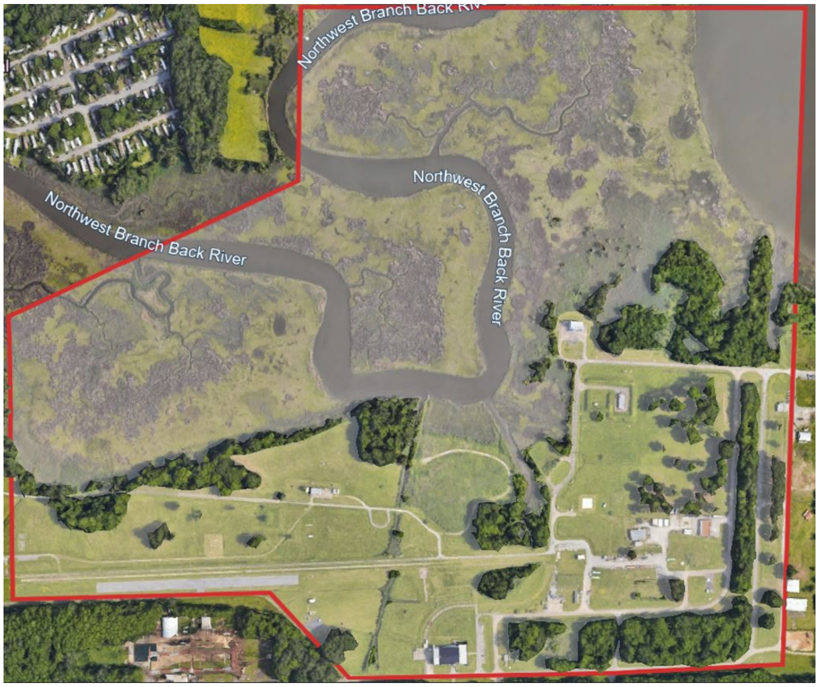

Northwest Branch Back Riv
Northwest Branch Back River
Northwest Branch Back River

Figure 25: CERTAIN flight test range at NASA LaRC with the geofence shown as a red polygon.

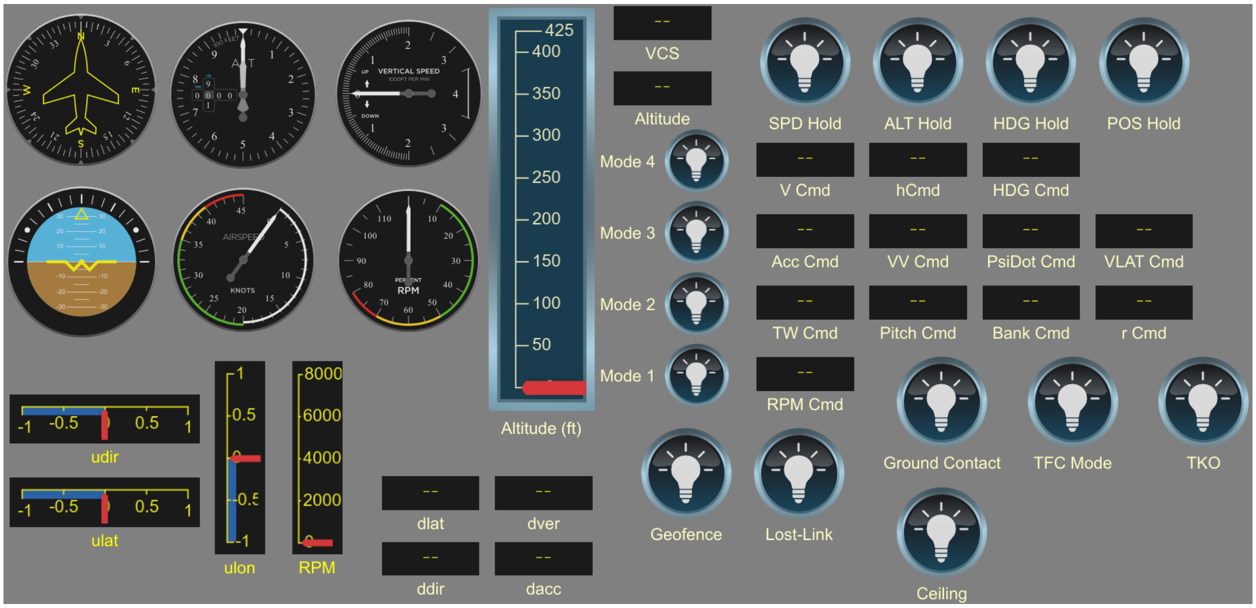


Figure 26: Custom SEARCH GCS used for flight testing.

The SEARCH vehicle was piloted by a NASA LaRC UAS pilot over a series of flights to evaluate the custom flight control system. The vehicle was flown in all modes: (i) open-loop control, (ii) inner-loop control, and (iii) middle- and outer-loop control. Results are shown for Mode 2 (inner-loop) and Mode 3 (middle- and outer-loop control) in the subsequent sections. The recorded flight data were saved at an average rate of 10 Hz from a live telemetry stream from the vehicle to the GCS. In addition to the display available in QGroundControl [46], the GCS operator monitored the custom Simulink® GCS display shown in Fig. 26. Further details regarding data acquisition are discussed in Sec. 6.

7.4.1 Explicit Model Following (Mode 2) Maneuvers

A series of maneuvers were performed in Mode 2 control, including a series of single-axis multistep inputs and sustained flight at varied airspeeds. In this mode, stick inputs map directly to bank angle, pitch angle, and yaw rate commands. The maximum allowable bank angle command $\phi_{cmd,max}$ was set to $\pm 45^\circ$, the maximum allowable pitch attitude command $\theta_{cmd,max}$ was set to $\pm 30^\circ$, and the maximum allowable yaw rate command $r_{cmd,max}$ was set to $\pm 30^\circ$/sec. The pilot commands thrust directly, requiring the pilot to modulate thrust as necessary. Vertical velocity data was recorded and has been displayed.

The first data trace (Fig. 27) shows a period of 65 seconds of flight in Mode 2 with a target speed of approximately 30 kts. The velocity trace shows ground speed (GS), airspeed (KEAS), and control system speed (VCS). The control system speed represents the speed the FCS uses to determine the aircraft velocity. The control system speed blends from ground speed to airspeed from $12 - 16$ kts to ensure accurate airspeed measurements. Throughout the flight test day, a steady wind from the West was present. This was also captured in the data trace through the alternating variation between GS and KEAS, changing as the vehicle turns into and out-of the wind.

Mode 2 provides attitude stabilization via the inner-loop EMF controllers. The EMF controllers perform well throughout the flight, meeting the desired commands with favorable responsiveness while rejecting the disturbances successfully. The control activity throughout the flight was minimal, highlighting the success in the selection of the control system gains and parameters. The roll and pitch axes both feature feed-forward elements of EMF control, providing a quicker response time than the yaw axis, acting only on PI-feedback through the command model.

Figure 28 shows a 4 minute window of flight performed in Mode 2 with a target speed of approximately 30 kts. Once again, the consistent wind can be seen in the velocity traces. The inner-loop EMF controllers exhibited desirable performance, meeting attitude and rate commands as requested by the pilot while showing low control activity. Sporadic multistep inputs were performed by the pilot and can be seen in the occasional spike in attitude/rate commands. These multistep inputs were mostly washed out by the command model (see Sec. 4.5) on account of their quick input time. Future testing would benefit from increased input durations, allowing for better observation of the response character.

Figure-eight maneuvers were performed with full control inputs to assess turning capabilities in Mode 2. The results of these tests are shown in Fig. 29, which depicts around 140 seconds of flight with sustained 45° turns. The SEARCH vehicle tracked the full-input turns well, meeting the bank angle and yaw rate commands in around 2 seconds, matching the desired command model. The directional command dir_{cmd} shows some increased activity on account of the commanded turn rate r_{cmd}, but shows desirable tracking performance. As the turns become more frequent, the vehicle does begin to lose some speed. The reduction in speed served as a direct validation of the scheduled control system gains and parameters by illustrating aircraft stability from $20 - 40$ kts.

Overall, the inner-loop EMF controllers performed well based on the simulation parameters and gain schedule, further validating the fidelity of the aerodynamic model and control system optimization methodology.

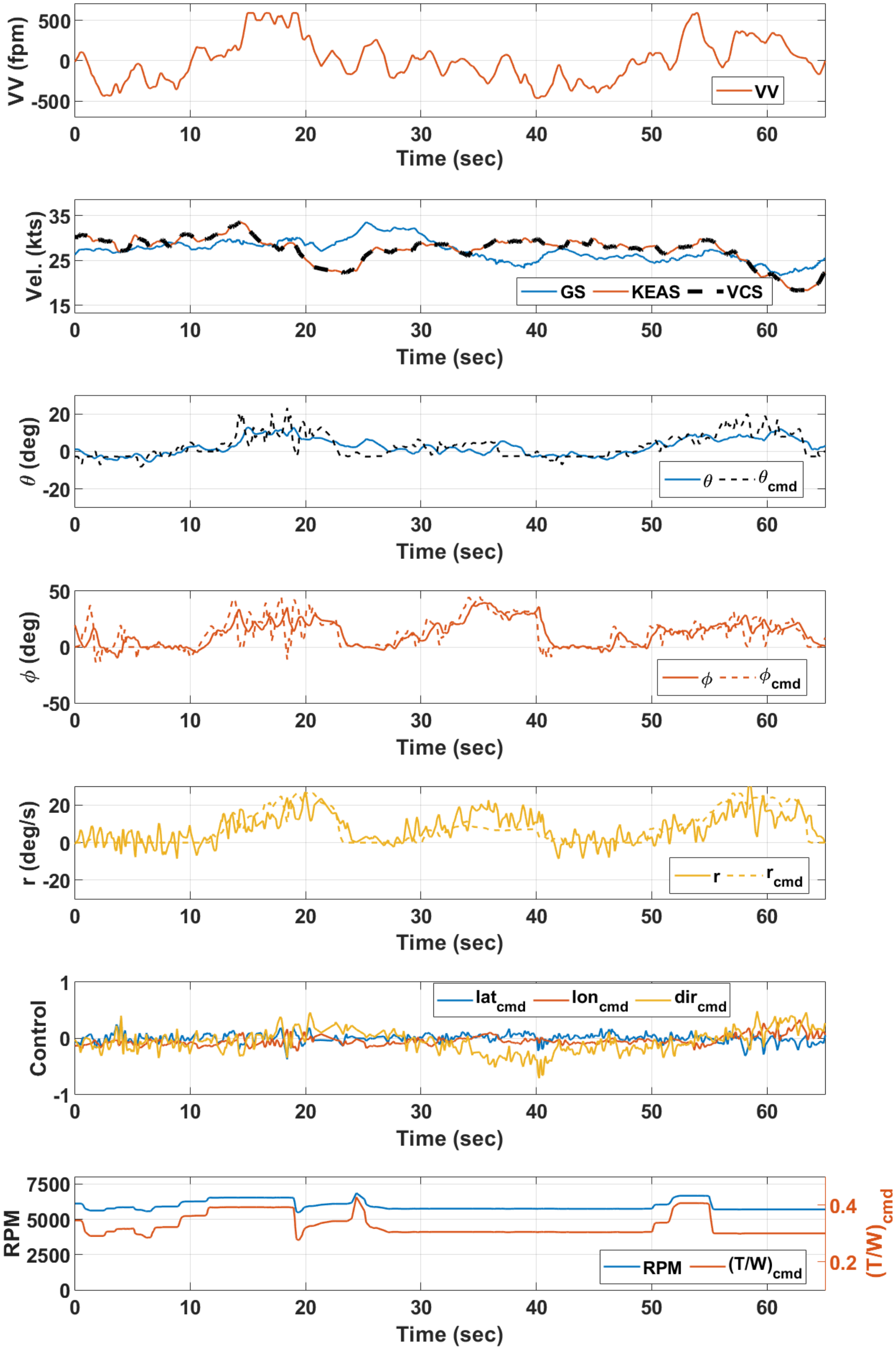


Figure 27: SEARCH Mode 2 flight data—maneuvers at 30 knots.

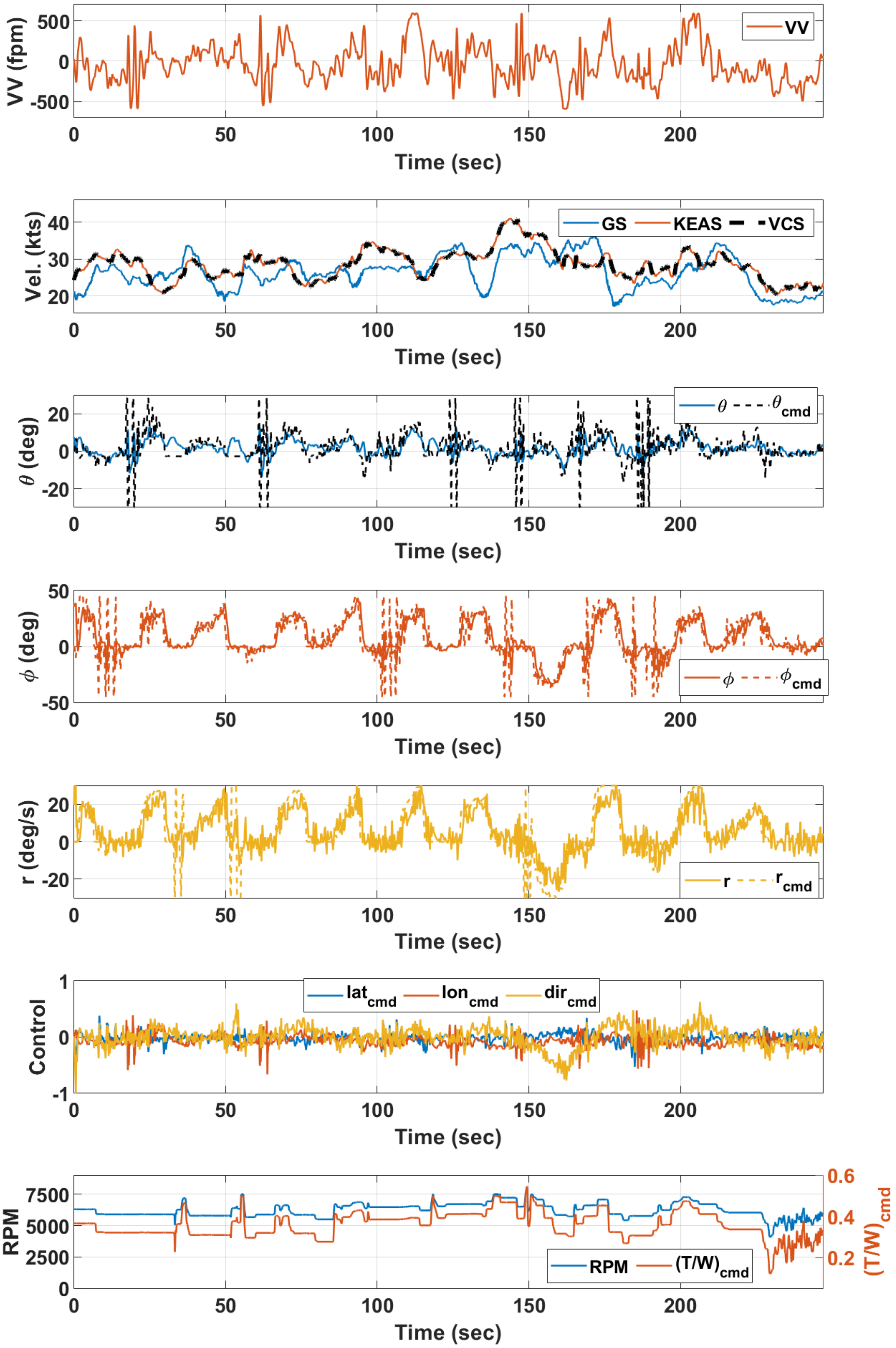


Figure 28: SEARCH Mode 2 flight data—maneuvers at 30 knots.

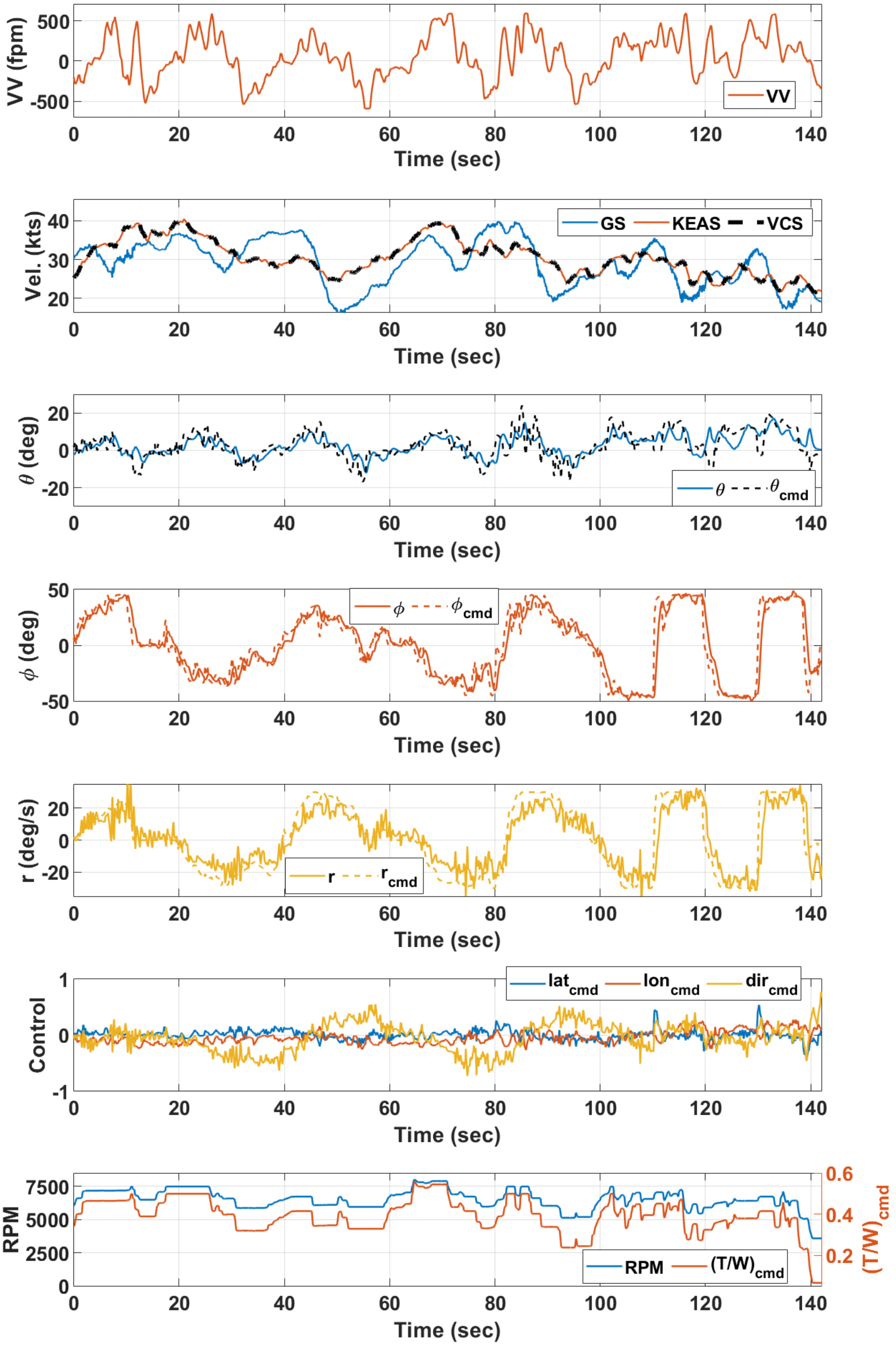


Figure 29: SEARCH Mode 2 flight data—figure-eight maneuvers.

7.4.2 Transition Flight Controller (Mode 3) Maneuvers

Following successful Mode 2 flight, Mode 3 flight was performed in which the entire FCS architecture was enabled, including outer-loop holds. The results shown were gathered over a single flight with a flight time of just over 13 minutes. The wind conditions during this flight were slightly more severe than the Mode 2 flights, resulting in a larger disagreement between ground speed and airspeed during some portions of the flight. The stick command mappings (see Sec. 4.1) in Mode 3 now map to turn rate command $\dot{\psi}_{cmd}$, vertical velocity command VV_{cmd}, lateral velocity command (which results in a sideslip command) $VLAT_{cmd}$ (β_{cmd}), and velocity rate command $\dot{V}_{cmd}$. The vehicle was first stabilized at a desired altitude before the vehicle was switched into Mode 3 flight, so takeoff and landing portions are not present.

Figure 30 begins moments after the enabling of Mode 3. In previous flights, pilot trim inputs on the pitch axis were not removed, resulting in a slight velocity rate command being present. The trim offset (unknown to the pilot at the time and to be resolved for future flight testing) slowly returned the commanded velocity to the upper limit after the commands were removed. This can be seen from $t = 0$ sec to $t = 17$ sec as a slight positive velocity command rate, resulting in the velocity command rising from 28 kts to 40 kts (maximum allowable command) over a span of 25 seconds. The vehicle responds to this velocity command change, tracking the desired velocity while maintaining the altitude hold of $h_{cmd} = 173$ ft from $t = 0$ sec until the pilot commands a climb at $t = 54$ sec. The velocity is met during multiple turns, during which the vehicle turns from a headwind to a tailwind, as indicated by the disagreement of ground speed and airspeed. The altitude command is maintained during the turn maneuvers, highlighting the effectiveness of the TFC to compensate for turns. The TFC controller simultaneously tracks the vertical velocity and velocity commands through the modified TECS control scheme successfully, providing disturbance rejection in the presence of the wind conditions.

Figure 31 depicts an approximately 160 second segment of the flight with sustained vertical velocity inputs while maintaining 40 kts. The TFC met the desired vertical velocity commands smoothly, resulting in a period of very stable flight even in the presence of sustained winds, observed in the velocity trends.

Another 90 second segment of flight is shown in Fig. 32. In this segment of the flight, the pilot countered the trim by commanding a deceleration. The TFC tracked the commanded velocity with favorable performance. A sustained climb rate was also commanded which was simultaneously tracked by the vehicle. At around $t = 33$ sec, a turn rate multistep input was performed, resulting in a bank angle command ϕ_{cmd} and yaw rate command r_{cmd} that was tracked by the EMF controllers. Resulting control activity perturbations can be seen during this multistep input, marking the peak control activity throughout the entire segment. As mentioned in the Mode 2 flights, the command model diminished these multistep inputs slightly on account of the multistep duration. A vertical velocity multistep command was performed at $t = 70$ sec, but the duration of the multistep input was too fast for the TFC controller to adequately track.

In summary, the resulting plots show successful tracking of the TFC and outer-loop controllers in response to all pilot commands, even in the presence of wind and turbulence. The inner-loop EMF controllers, as shown in all flight data, performed well with low control activity throughout all flights, showing stable flight at different speeds within the speed envelope. Future flights could maintain longer multistep commands if desired by the flight engineer to fall within the bandwidth of the control system.

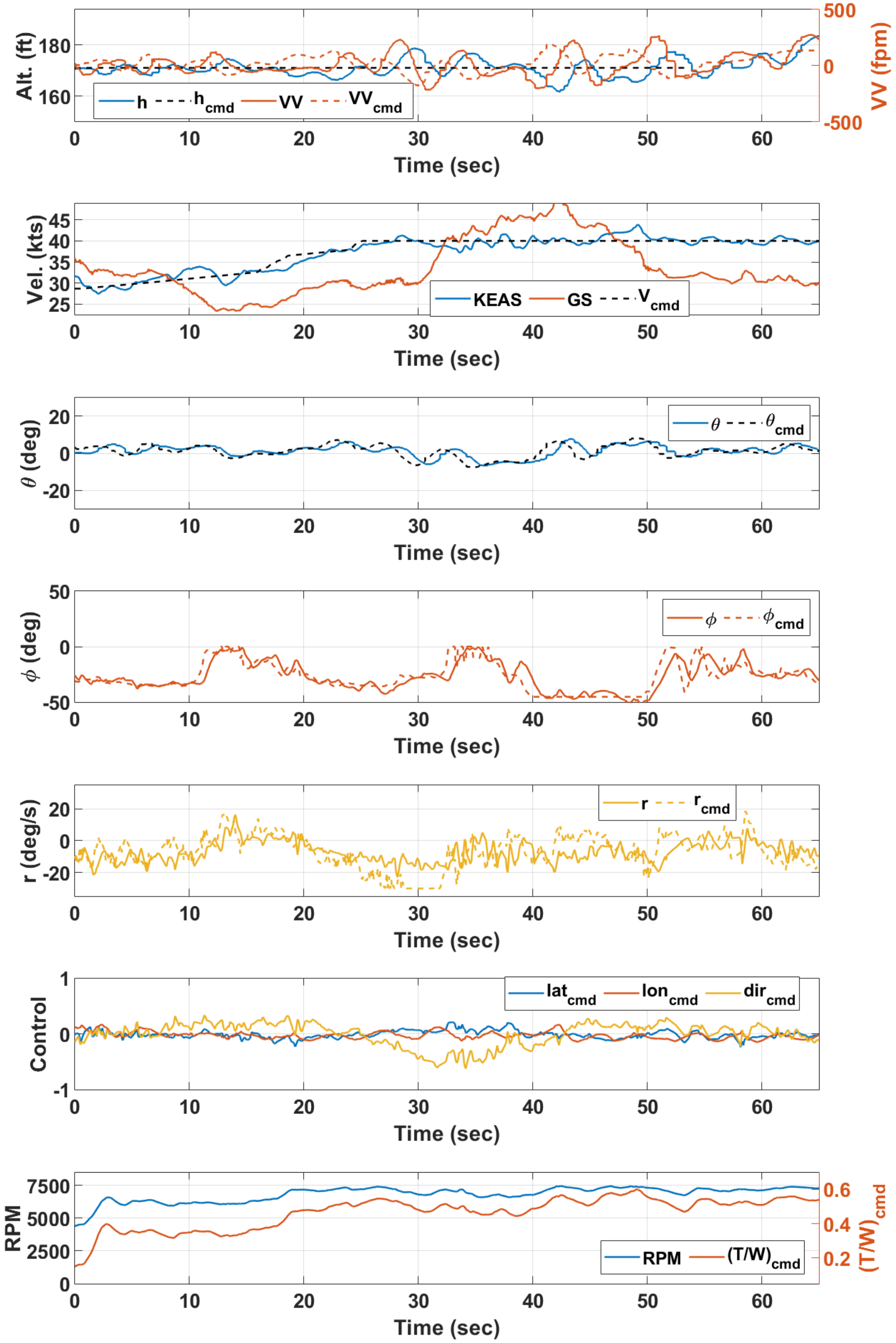


Figure 30: SEARCH Mode 3 flight data—altitude and velocity hold.

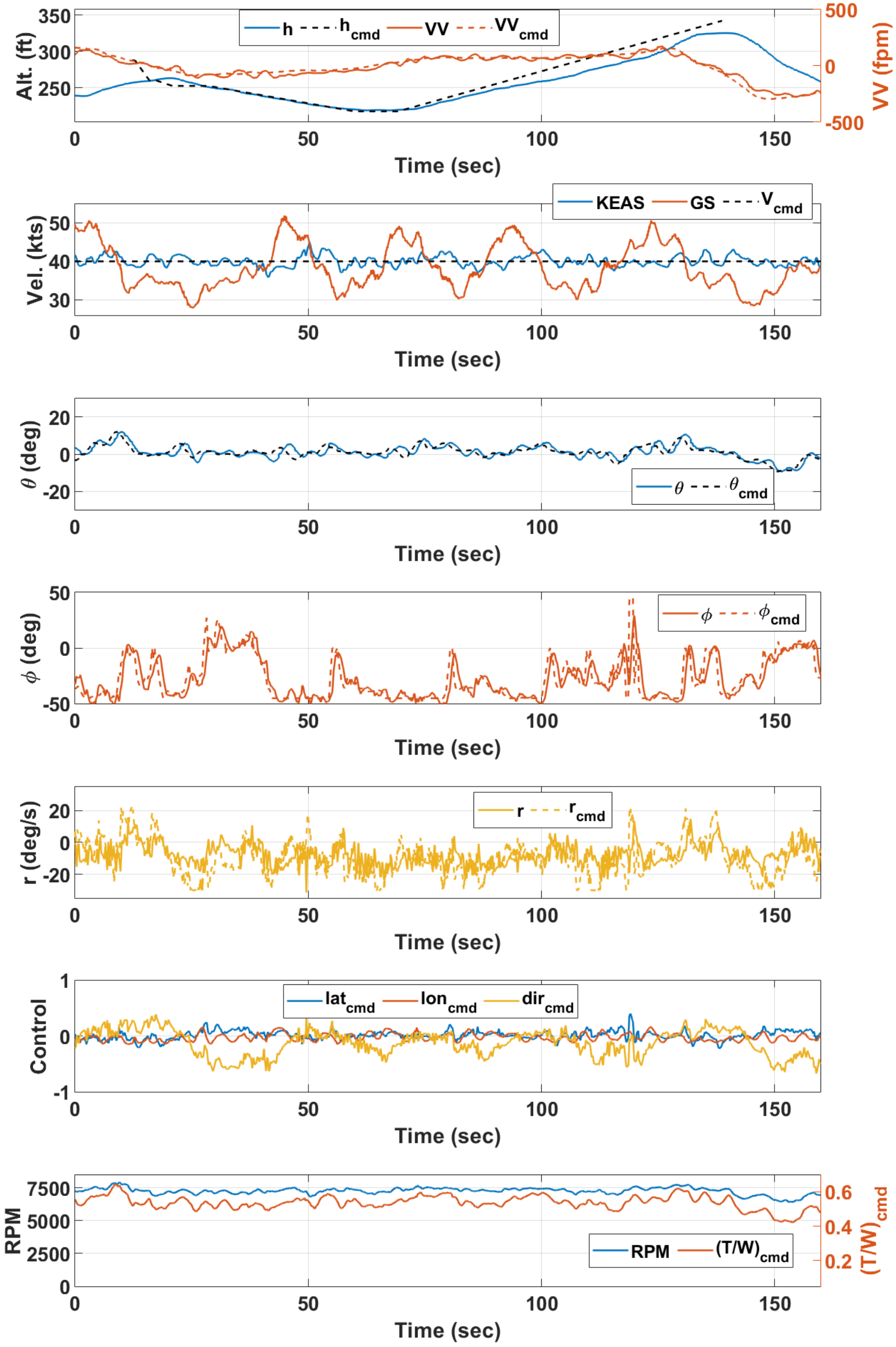


Figure 31: SEARCH Mode 3 flight data—descent and ascent.

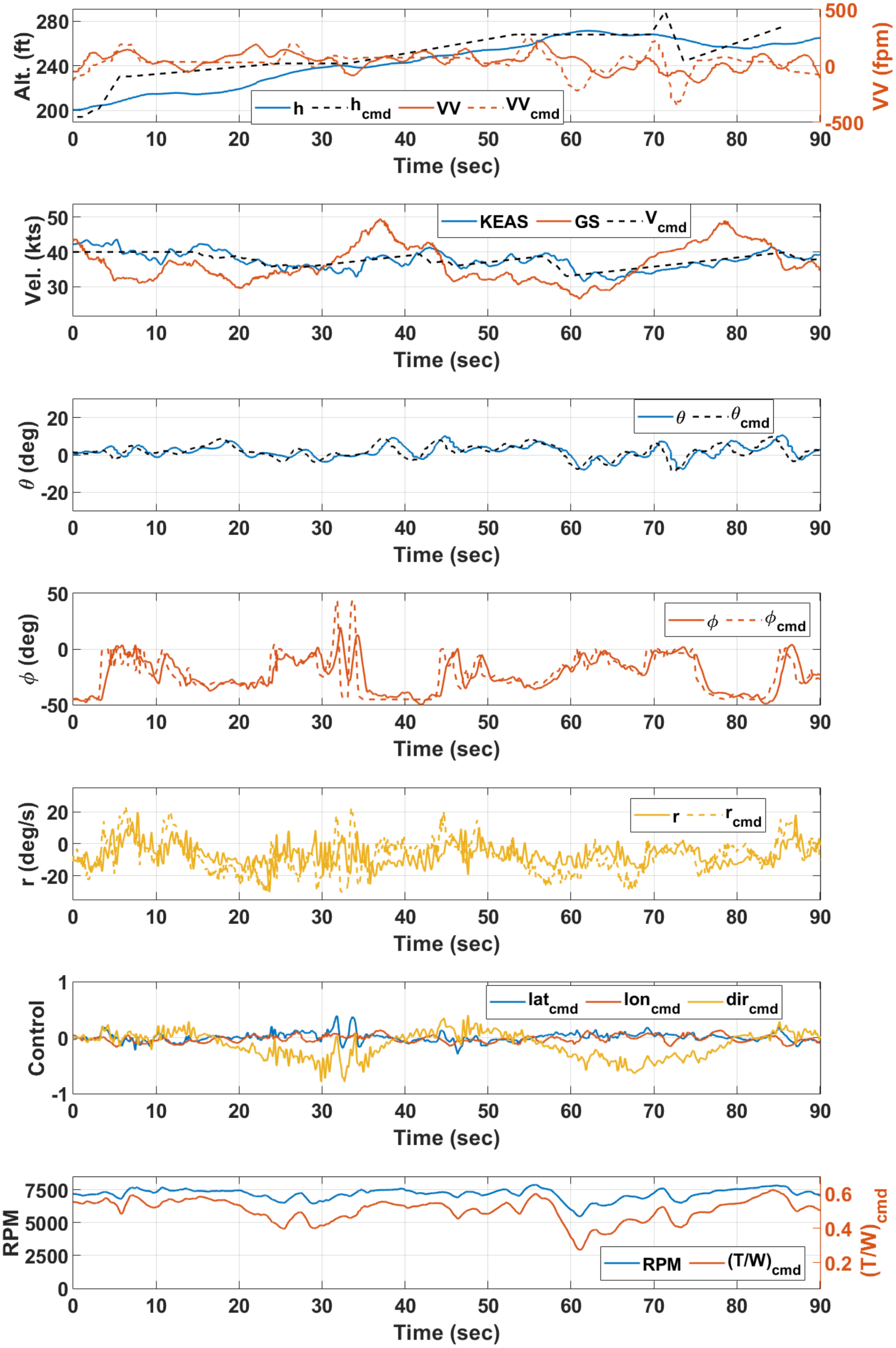


Figure 32: SEARCH Mode 3 flight data—velocity perturbations.

8 IMPACT Multirotor Flight Testing

The IMPACT vehicle, shown in Fig. 33, was developed as a versatile, cost-effective, expendable testbed for validating novel control laws and flight control architectures. The IMPACT vehicle is a multirotor configuration designed to offer a balance between performance and affordability, enabling extensive testing without the high costs associated with more complex UAV platforms. As discussed in Asper et al. [2], the IMPACT vehicle bridges the gap between simulation and rapid, low-risk, real-world flight testing. This platform has supported ongoing research at NASA LaRC, providing a reliable and adaptable testbed for evaluating hardware, flight control algorithms, and integration strategies. Key properties of the IMPACT vehicle are given in Table 7.

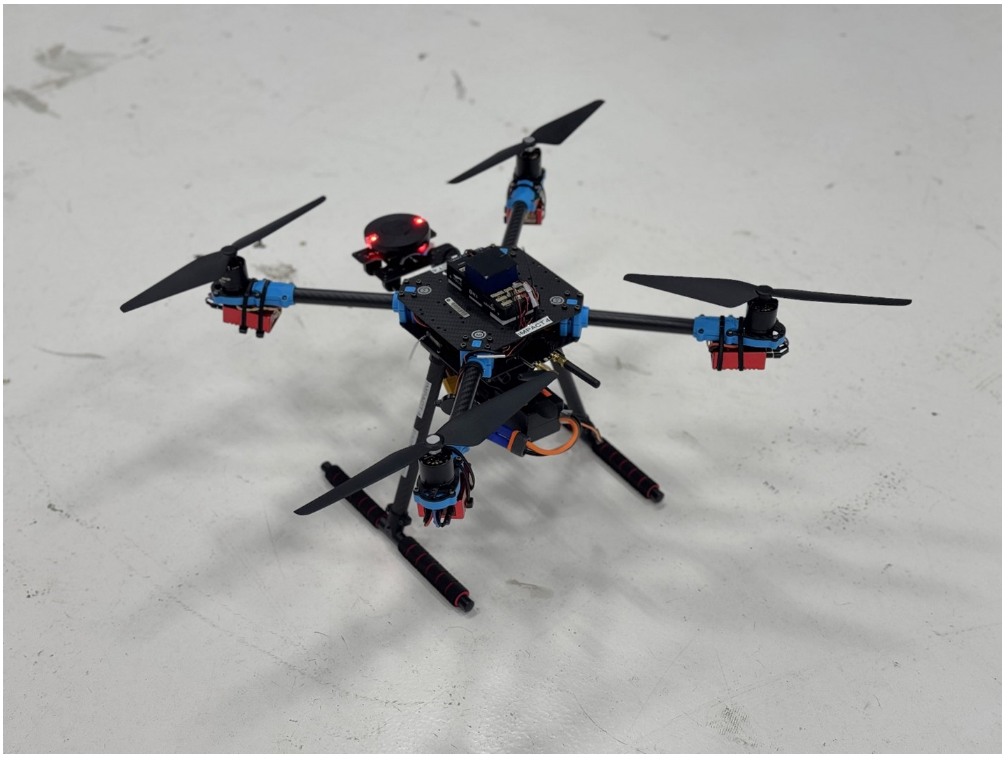


Figure 33: IMPACT vehicle pictured at the ALIFT facility.

Table 7: IMPACT vehicle properties.

Parameter	Name	Value
Weight	W	4.32 lb
Motor Arm Length	L	0.820 ft
Mass Moment of Inertia	I_{xx}	0.0216 slug-ft^2
Mass Moment of Inertia	I_{yy}	0.0233 slug-ft^2
Mass Moment of Inertia	I_{zz}	0.0353 slug-ft^2
Propeller Diameter	d	10 in

8.1 Flight Modes

The FCS tested on IMPACT can operate in three modes, determined by the pilot. Mode 1 represents closed-loop control, where the pilot commands roll angle ϕ, pitch angle θ, yaw rate r, and overall thrust. Mode 2 provides similar control but incorporates the altitude control portion of the flight control system instead of direct thrust pass-through. Mode 3 is the full control system (PCM, TFC, and LATCS) for the hover portion of flight with the inner-loop stabilization maintained. Mode 3 of the FCS was tuned heuristically via tethered flight tests.

8.2 Flight Testing

A series of tests were conducted at the Autonomy Lab for Intelligent Flight Technology (ALIFT) facility at NASA LaRC, leveraging both the indoor and outdoor facilities available for UAV flight testing. The indoor facility, shown in Fig. 34, was initially used for tethered flight tests while verifying smooth, in-flight switching between the three previously described flight modes. Addition of a LiDAR distance sensor allowed the vehicle to track altitude commands indoors, where GPS estimates can become less reliable. The LiDAR distance sensor also provided increased altitude accuracy in indoor and outdoor flight test activities. Finally, the ground contact flag and subsequent thrust-to-weight commands when on versus off the ground were validated to enable smooth takeoff and landing operations (further discussed in Sec. 4.2.4) that (i) mitigate integrator windup, (ii) reduce bouncing on the ground, and (iii) keep the vehicle out of ground effect for prolonged periods of flight. The outdoor netted facility, shown in Fig. 35, allows for testing in a manner in which the vehicle is constrained only by the outer limits of the net, enabling testing without requiring authorization for fully unconstrained flight testing in the National Airspace System.

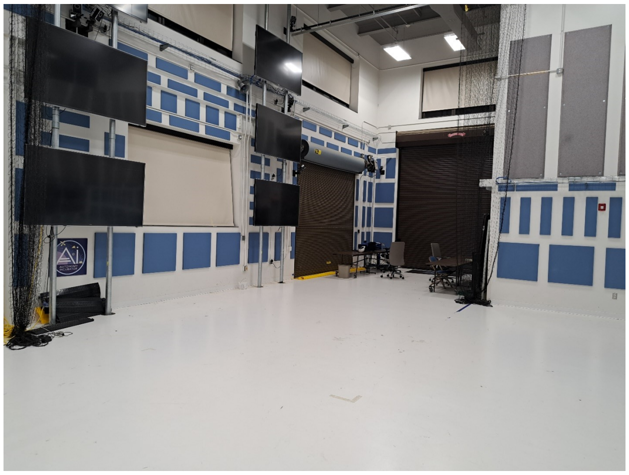


Figure 34: ALIFT indoor flight area.

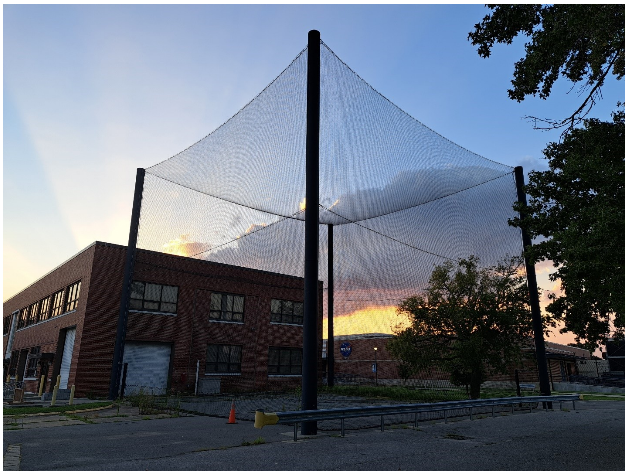


Figure 35: ALIFT outdoor flight area.

8.2.1 Altitude Hold Maneuvers

The altitude hold mode, initially developed for use during research activities, enables the vertical axis of the TFC controller in vertical flight operation and enables the altitude hold controller. Roll, pitch, and yaw rate commands are still controlled directly from the inner-loop control system. The pilot now commands vertical velocity VV_{cmd} instead of thrust directly, enabling altitude hold after centering the thrust stick.

To validate the altitude controller, representative multistep commands in the roll, pitch, and yaw axes were input by the pilot while maintaining altitude hold. The inner-loop attitude controllers previously developed for IMPACT [2] were not the focus of this study, so inner-loop command tracking performance will not be commented on. In general, the inner-loop controllers responded in an acceptable way to keep the vehicle stable and controllable. To assess performance, the maneuvers showcase the altitude h and vertical velocity VV_{cmd} of the vehicle (and commands), the forward velocity $VFWD$ and x-axis acceleration a_x, the pitch attitude θ and command θ_{cmd}, the lateral velocity $VLAT$ and command $VLAT_{cmd}$, the bank angle ϕ and command ϕ_{cmd}, the yaw rate r and command r_{cmd}, and the individual propeller rotational speeds N_1, N_2, N_3, N_4.

The altitude hold controller was tested in an outdoor environment in the presence of winds, coming from the right side of the vehicle, to verify performance in turbulent conditions. This can also be seen in the periodic positive roll commands by the pilot to counter active leftward drift due to wind. The results from this flight are shown in Fig. 36, which show a 105 second portion of flight in altitude hold mode. The altitude controller maintained altitude with close tracking, and responded to vertical velocity commands smoothly.

Figure 37 shows the IMPACT vehicle holding an altitude of 9 ft while undergoing piloted yaw, roll, and pitch multistep maneuvers, respectively. This figure is a focused look at the results from Fig. 36, showing the results from $t = 80$ sec to $t = 105$ sec. The altitude was held to within 1 ft of the commands, showing desirable performance characteristics throughout the multistep maneuvers and validating the altitude hold controller in a representative use-case scenario.

The addition of the altitude hold controller provides a simple, yet effective method for maintaining altitude during research maneuvers. This implementation also showcases the ability of the TFC controller to be re-purposed to provide single-axis control as required.

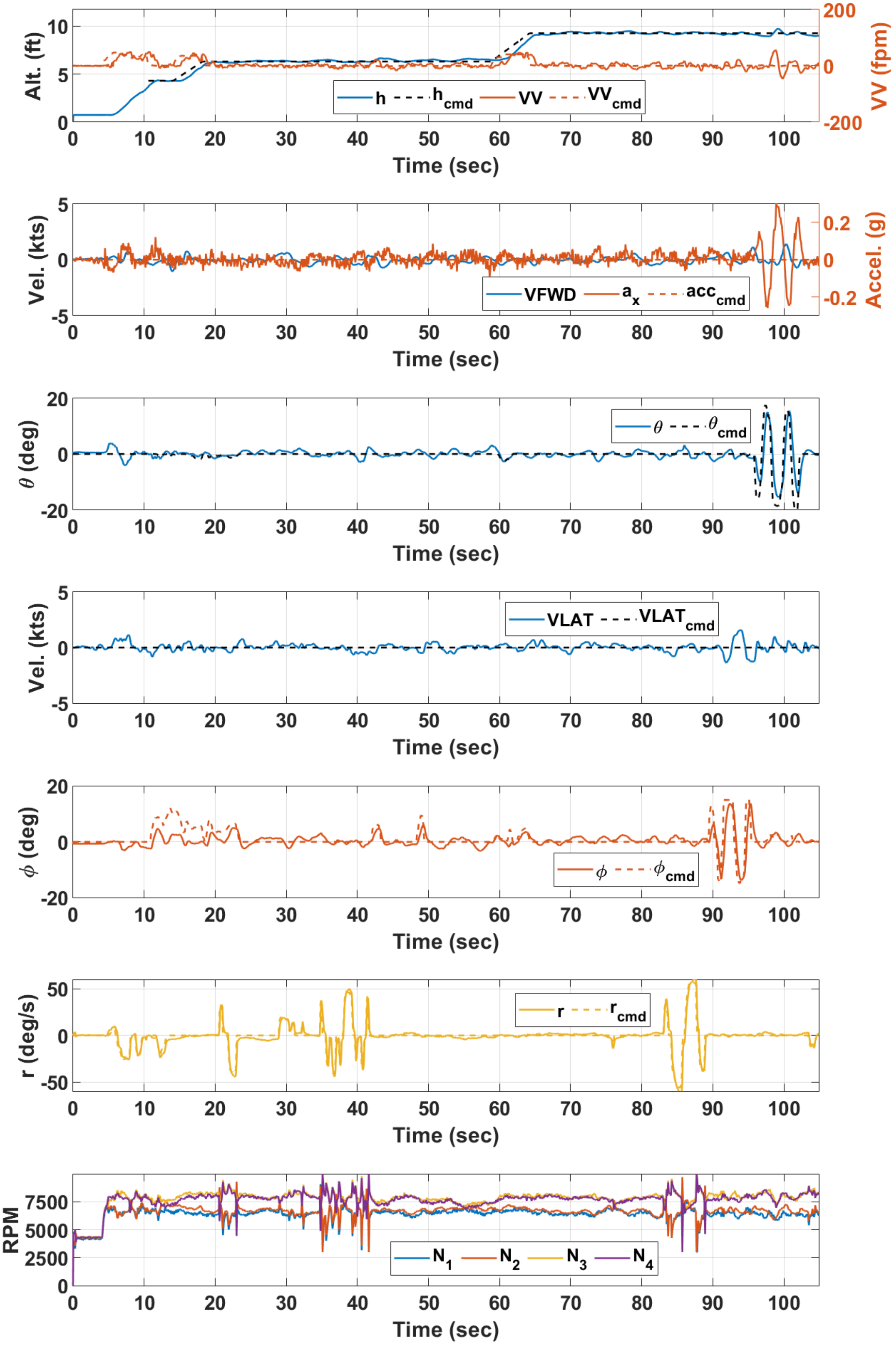


Figure 36: IMPACT altitude mode (Mode 2) flight with inner-loop multistep maneuvers.

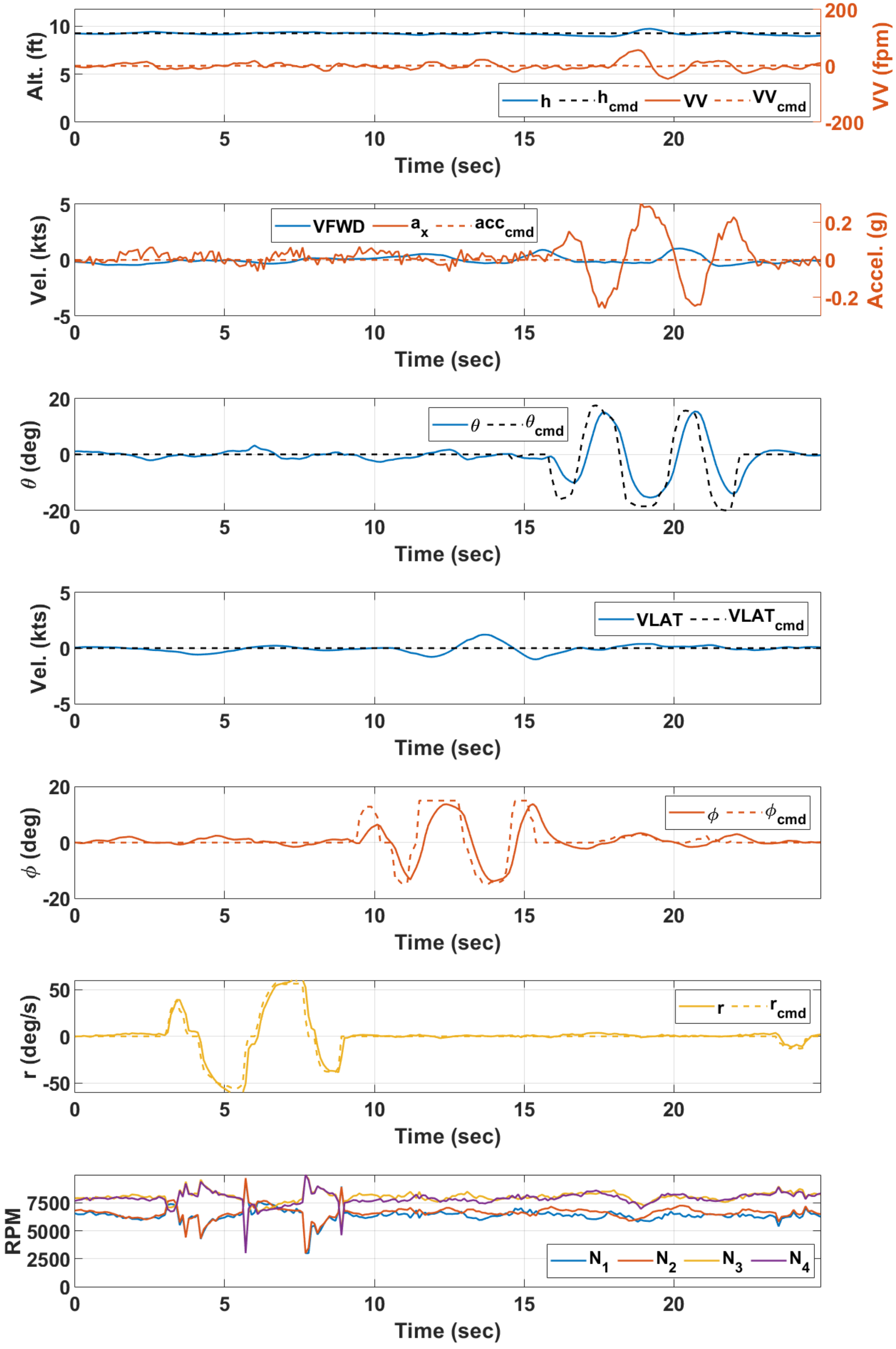


Figure 37: IMPACT altitude mode (Mode 2) flight with inner-loop multistep maneuvers.

8.2.2 Full FCS Maneuvers

The middle- and outer-loop portions of the control system architecture depicted in Fig. 2 were incorporated on the IMPACT in the final flight mode. This included the left three subsystems: PCM, TFC, and LATCS. The control commands in this flight mode matched those used in the SEARCH VTOL vehicle, and this addition made use of the existing inverse propulsion and inner-loop controllers. The IMPACT vehicle was used as a testbed for validating the hovering flight operating conditions of the FCS architecture where the SEARCH fixed-wing vehicle could not operate in. The additional controllers were validated through a series of flight tests, both indoors and outdoors. Piloted commands were performed to verify vehicle responsiveness and controllability. The wind speeds picked up further in the second round of flight tests, with the wind direction still being from the right side of the vehicle.

Figure 38 shows a 57 second hover flight, beginning with an altitude hold at 5 ft. The pilot counters to the wind conditions through periodic positive lateral velocity commands, resulting in positive bank angle commands. Throughout the maneuver, the vehicle maintained a slightly positive bank angle to counter the winds. Although it is not the focus of this study, additional integral control on the roll axis inner-loop controller would have likely improved the performance further. The altitude hold is disengaged at $t = 18$ sec as a result of the vehicle momentarily dropping below the "ground effect" threshold of $h = 5$ ft. The pilot them climbs at $t = 25$ sec, after which the altitude hold re-engages at around $h = 9$ ft and is attained shortly after. The pitch axis is now controlled from acceleration commands acc_{cmd}, and the feed-forward gain K_A^{ff} (see Eq. (43)). This relationship equates a 0.1 g command to a 6° pitch attitude command. The heading hold controller is also active, providing minor adjustments to maintain a specified heading. At around $t = 40$ sec, the pilot commands a slight descent rate of around 40 fpm. This command is difficult to spot in Fig. 38, but is tracked to produce a smooth descent. In the first few seconds and last few seconds of the flight data, the takeoff logic described in Sec. 4.2.4 shows the spooling up and spooling down of (T/W) based on the ground contact.

A set of multistep commands were performed on the control system to show responsiveness to vertical velocity commands, lateral velocity commands, and acceleration commands as shown in Fig. 39. The maneuver begins with a set of vertical velocity step commands at around $t = 25$ sec, showing desirable vertical velocity tracking throughout the maneuver. The maneuver finishes at around $t = 90$ sec, engaging the altitude hold. A lateral velocity multistep input was performed from $t = 105$ sec to $t = 120$ sec, which generates bank angle commands that are met by the inner-loop controllers. A final set of acceleration step commands were performed from $t = 125$ sec to $t = 150$ sec. The feed-forward gain K_A^{ff} increases the responsiveness considerably, and the acceleration a_x follows the multistep commands in an acceptable manner.

The PCM, TFC, and LATCS portions of the proposed vehicle were successfully implemented on the IMPACT vehicle through a series of outdoor tests and flight maneuvers. These components were successfully integrated seamlessly with the existing inner-loop controllers and provided smooth control of the vehicle. The takeoff and landing operation logic provided improved takeoff performance and reduced bounding during landings, cutting thrust quickly on ground contact.

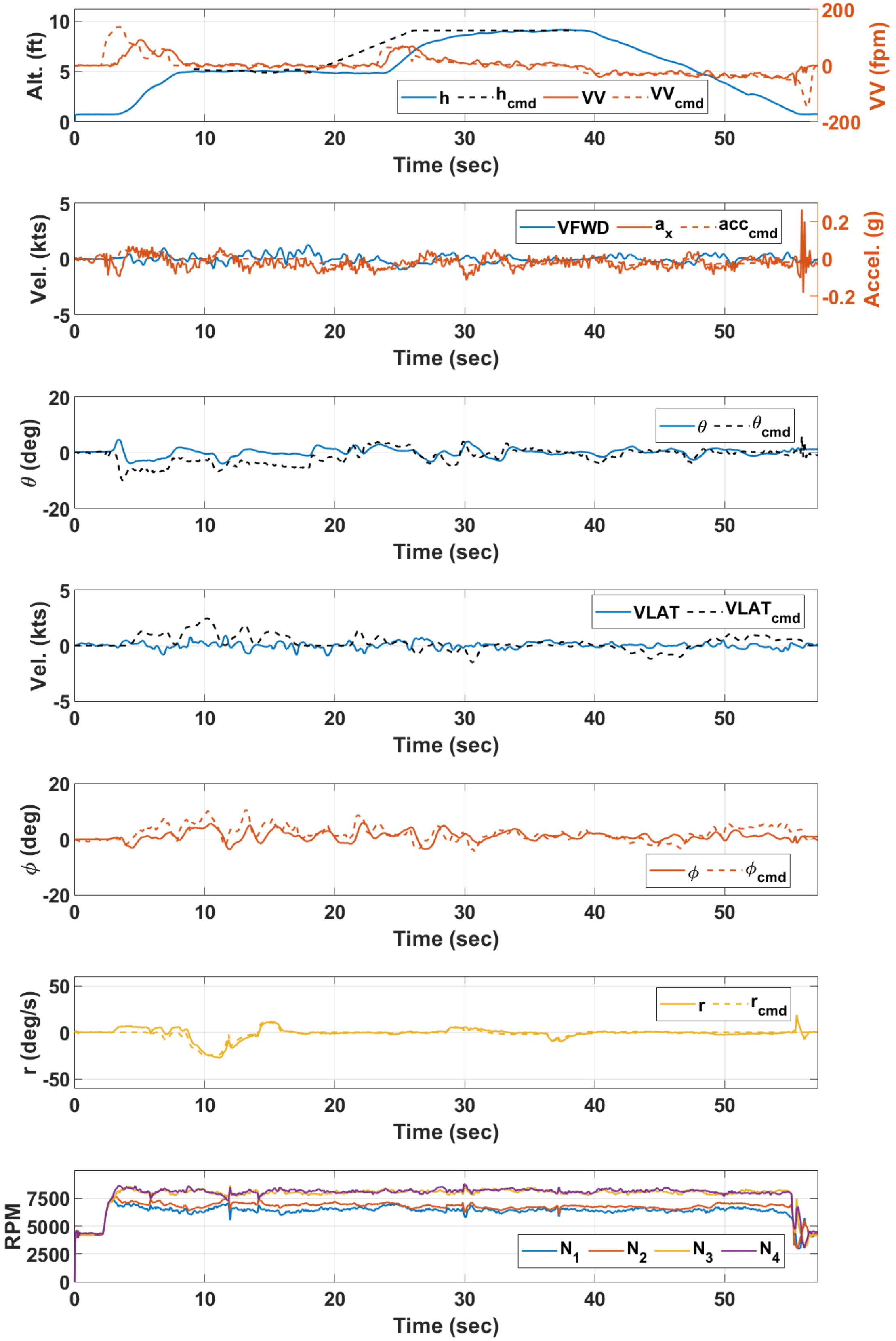


Figure 38: IMPACT outdoor flight test with crosswind conditions in Mode 3.

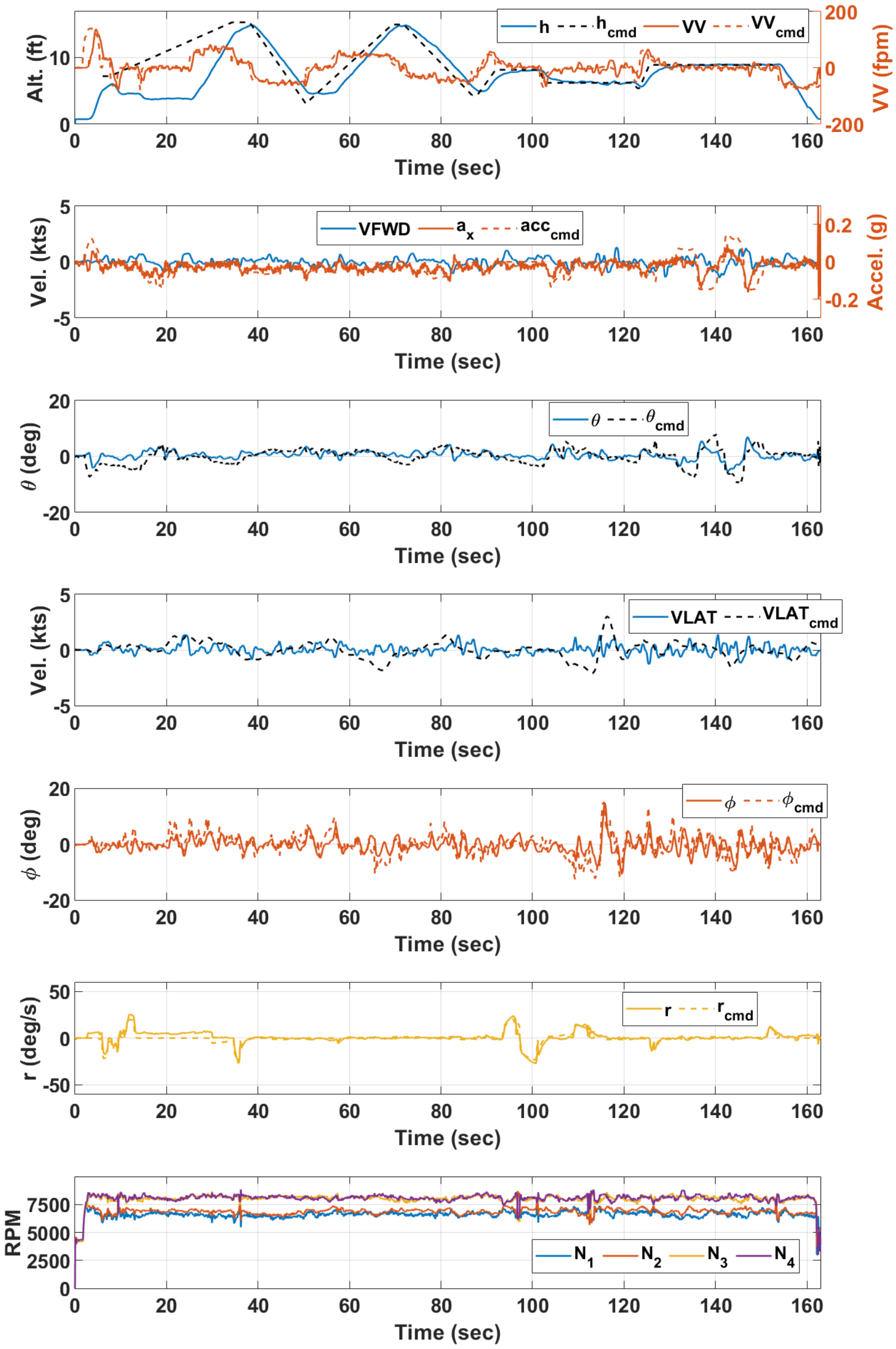


Figure 39: IMPACT indoor flight with multistep commands in Mode 3.

9 Conclusions

The research presented in this work outlines the design, simulation, and flight testing of a novel flight control system architecture tailored for vertical takeoff and landing vehicles. The development of the SIMulation PLatform (SIMPL) enabled rapid control system development and provided a robust framework for the validation of the proposed control system on different vehicle configurations, including the SEARCH VTOL and the SEARCH fixed-wing aircraft. The integration of concepts from the Total Energy Control System (TECS) into the flight control system, particularly for the transition and forward flight phases, proved effective in managing energy states and ensuring stability across flight regimes.

Flight tests conducted at NASA LaRC validated the simulation results, demonstrating the feasibility and effectiveness of the control system architecture. The test data showed that the Explicit Model Following (EMF) inner-loop controllers successfully stabilized and controlled the SEARCH vehicle in various flight conditions, while also providing seamless transition between open-loop, inner-loop, and outer-loop control laws. The Transition Flight Controller (TFC) successfully controlled the SEARCH and IMPACT vehicles in forward flight and hover flight conditions, respectively, strengthening simulation results and VTOL-driven design of the control algorithm. Additionally, the integration with a Pixhawk® flight computer using the MathWorks® UAV Toolbox facilitated a streamlined process from simulation to flight testing, highlighting the practicality of the control system for real-world applications.

Acknowledgments

The authors would like to acknowledge the flight test support provided by Brayden Chamberlain, Mark Motter, and Skyler Hudson, as well as additional research support provided by Mario Smith. The SEARCH vehicle used in this report was formed in collaboration with Mark Motter and Eugene Morelli. Personal discussions with Ben Davis helped to formulate the general methods to create the X-Plane visualization. Personal discussions with Yevhenii Kovryzhenko helped to develop real-time telemetry communication methods.

References

1. Asper, G. D., and Simmons, B. M., "Rapid Flight Control Law Deployment and Testing Framework for Subscale VTOL Aircraft," NASA TM–20220011570, Sep. 2022.

2. Asper, G. D., Simmons, B. M., Axten, R. M., Ackerman, K. A., and Corrigan, P. E., "Inexpensive Multirotor Platform for Advanced Controls Testing (IMPACT): Development, Integration, and Experimentation," NASA TM–20240000223, Mar. 2024.

3. Lambregts, A., "Integrated System Design for Flight and Propulsion Control Using Total Energy Principles," *AIAA Aircraft Design, Systems and Technology Meeting*, AIAA-83-2561, Fort Worth, TX, USA, Oct 17-19, 1983. https://doi.org/10.2514/6.1983-2561.

4. Lambregts, A., "Functional Integration of Vertical Flight Path and Speed Control Using Energy Principles," *First Annual NASA Aircraft Controls Workshop*, NASA Langley Research Center, Hampton, VA, USA, Oct 25-27, 1983.

5. Lambregts, A., "Vertical Flight Path and Speed Control Autopilot Design using Total Energy Principles," *AIAA Guidance and Control Conference*, AIAA-1983-2239, Gatlinburg, TN, USA, 15-17 August, 1983. https://doi.org/10.2514/6.1983-2239.

6. Lambregts, A., “Operational Aspects of the Integrated Vertical Flight path and Speed Control System,” *SAE Transactions*, Vol. 92, No. 4, 1983, pp. 29–41. https://doi.org/10.4271/831420.

7. Bruce, R., “NASA B737 Flight Test Results of the Total Energy Control System,” NASA CR-178285, Boeing Commercial Airplane Company, Seattle, WA, January, 1987.

8. Comer, A., and Chakraborty, I., “Total Energy-Based Control Architecture Design and Optimization for Lift-Plus-Cruise Aircraft,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 47, No. 7, 2024, pp. 1414–1436. https://doi.org/10.2514/1.G007605.

9. Comer, A., and Chakraborty, I., “Full Envelope Flight Control System Design and Optimization for a Tilt-Wing Aircraft,” *Journal of the American Helicopter Society*, Vol. 69, No. 3, 2024, pp. 1–18. https://doi.org/10.4050/JAHS.69.032003.

10. Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., John Wiley & Sons, Hoboken, New Jersey, 2015.

11. Etkin, B., and Reid, L. D., *Dynamics of Flight: Stability and Control*, 3rd ed., John Wiley & Sons, Inc, New York, NY, 1996.

12. Simmons, B. M., Gresham, J. L., and Woolsey, C. A., “Aero-Propulsive Modeling for Propeller Aircraft Using Flight Data,” *Journal of Aircraft*, Vol. 60, No. 1, 2023, pp. 81–96. https://doi.org/10.2514/1.C036773.

13. Simmons, B. M., Gresham, J. L., and Woolsey, C. A., “Nonlinear Dynamic Modeling for Aircraft with Unknown Mass Properties Using Flight Data,” *Journal of Aircraft*, Vol. 60, No. 3, 2023, pp. 968–980. https://doi.org/10.2514/1.C037259.

14. Brandt, J. B., Deters, R. W., Ananda, G. K., Dantsker, O. D., and Selig, M. S., *UIUC Propeller Database*, Vol. 3, University of Illinois at Urbana-Champaign, 2020. https://m-selig.ae.illinois.edu/props/propDB.html, Accessed 15 July 2024.

15. Dantsker, O. D., Caccamo, M., Deters, R. W., and Selig, M. S., “Performance Testing of Aero-Naut CAM Folding Propellers,” *AIAA AVIATION 2020 Forum*, AIAA-2020-2762, Jun. 2020. https://doi.org/10.2514/6.2020-2762.

16. Simmons, B. M., and Hatke, D. B., “Investigation of High Incidence Angle Propeller Aerodynamics for Subscale eVTOL Aircraft,” NASA TM–20210014010, May 2021.

17. Tischler, M., Berger, T., Ivler, C., Mansur, M., Cheung, K., and Soong, J., *Practical Methods for Aircraft and Rotorcraft Flight Control Design - An Optimization-Based Approach*, 1st ed., American Institute of Aeronautics and Astronautics, 2017.

18. Comer, A., Bhandari, R., Putra, S., and Chakraborty, I., “Design, Control Law Development, and Flight Testing of a Subscale Lift-Plus-Cruise Aircraft,” *AIAA SCITECH 2024 Forum*, AIAA-2024-2644, Orlando, FL, 2024. https://doi.org/10.2514/6.2024-2644.

19. Comer, A., Chakraborty, I., Yavhenii, K., Taheri, E., Bhandari, R., Kunwar, B., and Putra, S., “Flight Testing of Explicit Model-Following Trajectory Control System for Lift-Plus-Cruise and Tilt-Wing Configurations,” *VFS 80 Forum*, Vertical Flight Society, Montreal, QC, Canada, 2024. https://doi.org/10.4050/F-0080-2024-1306.

20. Comer, A., Chakraborty, I., Putra, S., Bhandari, R., Kunwar, B., and Davis, B., “Design, Optimization, and Flight Testing of Trajectory Control System for Lift-Plus-Cruise VTOL Aircraft,” *AIAA AVIATION 2024 Forum*, AIAA-2024-4563, Las Vegas, NV, 2024. https://doi.org/10.2514/6.2024-4563.

21. “PX4 Autopilot User Guide,” https://docs.px4.io/main/en/, Accessed 27 October 2024.

22. Comer, A., and Chakraborty, I., "Explicit Model Following Trajectory Control System for Multiple Vertical Takeoff and Landing Configurations," *34th ICAS Congress*, ICAS-2024-1166, Florence, Italy, 2024.

23. Comer, A., "A Generalized Flight Control Architecture for Transitioning Flight Vehicles with Flight Test Validation," Ph.D. thesis, Auburn University, Auburn, AL, USA, Dec. 2024. https://doi.org/10415/9555.

24. Comer, A. M., and Chakraborty, I., "Optimizing Explicit Model-Following Trajectory Control Laws for a Vectored Thrust Conguration," *Journal of the American Helicopter Society*, Vol. 70, No. 1, 2025, pp. 1–20. https://doi.org/10.4050/JAHS.70.012007.

25. "MathWorks Documentation," https://www.mathworks.com/help/, Accessed 06 August 2024.

26. "X-Plane," https://www.x-plane.com/, Accessed 28 September 2024.

27. Thomas, P., "X-Plane Blockset," https://www.mathworks.com/matlabcentral/fileexchange/76028-x-plane-blockset, Accessed 09 October 2024.

28. "FlightGear Flight Simulator," https://www.flightgear.org/, Accessed 25 January 2025.

29. Lambregts, A., "Generalized Automatic and Augmented Manual Flight Control," https://www.studocu.com/row/document/university-of-nairobi/electrical-engineering/automatic-and-augmented-manual-fc/47292576, May 2006. Berlin Technical University Colloquium, Accessed: July 27, 2020.

30. Lambregts, A., "TECS Generalized Airplane Control System Design - An Update," *Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation and Control*, FrAT3.1, Delft University of Technology, Delft, The Netherlands, April 10-12, 2013. https://doi.org/10.1007/978-3-642-38253-6_30.

31. Chakraborty, I., and Comer, A., "Optimizing Explicit Model-Following Trajectory Control Laws for a Vectored Thrust Configuration," *VFS 80 Forum*, Vertical Flight Society, Montreal, QC, Canada, 2024. https://doi.org/10.4050/F-0080-2024-1277.

32. Niedermeier, D., and Lambregts, A., "Design of an Intuitive Flight Control System," *CEAS 2009 European Air and Space Conference*, Manchester, U.K., Oct 26-29, 2009.

33. Ganguli, S., and Balas, G., "A TECS Alternative using Robust Multivariable Control," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2001-4022, Montreal, Canada, 6-9 August, 2001. https://doi.org/10.2514/6.2001-4022.

34. Comer, A., and Chakraborty, I., "Flight Control System Architecture for Urban Air Mobility Simplified Vehicle Operations," *AIAA SCITECH 2023 Forum*, AIAA-2023-0399, National Harbor, MD, 2023. https://doi.org/10.2514/6.2023-0399.

35. Saetti, U., and Horn, J. F., "Use of Harmonic Decomposition Models in Rotorcraft Flight Control Design with Alleviation of Vibratory Loads," *European Rotorcraft Forum 2017*, Milan, Italy, 2017.

36. Saetti, U., Horn, J., Lakhani, S., Lagoa, C., and Berger, T., "Design of Dynamic Inversion and Explicit Model Following Flight Control Laws for Quadrotor UAS," *Journal of the American Helicopter Society*, Vol. 65, No. 3, 2020, pp. 1–16. https://doi.org/10.4050/JAHS.65.032006.

37. Berger, T., Tischler, M., Hagerott, S., Gangsaas, D., and Saeed, N., "Longitudinal Control Law Design and Handling Qualities Optimization for a Business Jet Flight Control System," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, 2012. https://doi.org/10.2514/6.2012-4503.

38. Berger, T., Tischler, M., Hagerott, S., Gangsaas, D., and Saeed, N., "Lateral/Directional Control Law Design and Handling Qualities Optimization for a Business Jet Flight Control System," *AIAA Atmospheric Flight Mechanics Conference*, Boston, MA, 2013. https://doi.org/10.2514/6.2013-4506.

39. Standard, M., “Flying qualities of piloted vehicles,” Tech. rep., MIL-STD-1797A, 1990.

40. Hodgkinson, J., “A History of Low Order Equivalent Systems for Aircraft Handling Qualities Analysis and Design,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA-2003-5466, Austin, TX, 2003. https://doi.org/10.2514/6.2003-5466.

41. “OpenVSP,” https://openvsp.org/, Accessed 28 September 2024.

42. “Troubleshooting PX4 Firmware Build Failure Due to Flash Memory Overflow on the Hardware,” https://www.mathworks.com/help/uav/px4/ug/troubleshooting-flash-overflow-px4.html, Accessed 11 December 2024.

43. Miller, M. P., “An Accurate Method of Measuring the Moments of Inertia of Airplanes,” NACA TN-351, Oct. 1930.

44. Gresham, J. L., “Aerodynamic Modeling in Nonlinear Regions, including Stall Spins, for Fixed-Wing Unmanned Aircraft from Experimental Flight Data,” Ph.D. Dissertation, Virginia Tech, Mar. 2022.

45. “Blender,” https://www.blender.org/, Accessed 28 September 2024.

46. “QGroundControl,” http://qgroundcontrol.com/, Accessed 31 July 2024.

47. “RFDesign Tools,” https://files.rfdesign.com.au/tools/, Accessed 14 August 2024.

Appendix A

Hardware Setup

A.1 Custom Telemetry Stream

A.1.1 PX4 Code Modifications

The custom telemetry stream setup requires modifications to the MAVLink main file in PX4. The following steps outline the necessary changes:

1. Navigate to the `<px4FirmwareRoot>\src\modules\mavlink\mavlink_main.cpp` file.

2. Under the “`case MAVLINK_MODE_NORMAL`” line, make the following changes (shown in red):

   ```
   configure_stream_local("ATTITUDE", 12.0f);
   configure_stream_local("ATTITUDE_QUATERNION", 5.0f);
   ```

   Then, directly after the nearby “`configure_stream_local("COLLISION", unlimited_rate);`” line, add:

   ```
   configure_stream_local("DEBUG_FLOAT_ARRAY", 10.0f);
   ```

3. Under the “`#if !defined(CONSTRAINED_FLASH)`” statement following

   ```
   case MAVLINK_MODE_NORMAL
   case MAVLINK_MODE_ONBOARD
   case MAVLINK_MODE_ONBOARD_LOW_BANDWIDTH
   case MAVLINK_MODE_EXTVISIONMIN
   ```

   delete the following line:

   ```
   configure_stream_local("DEBUG_FLOAT_ARRAY", 1.0f);
   ```

   or

   ```
   configure_stream_local("DEBUG_FLOAT_ARRAY", 10.0f);
   ```

4. Under `case MAVLINK_MODE_CONFIG`, make the following addition to enable the DEBUG_FLOAT_ARRAY while connected over USB:

   ```
   configure_stream_local("DEBUG_FLOAT_ARRAY", 50.0f);
   ```

   Additionally, under the `CONSTRAINED_FLASH` condition for the `MAVLINK_MODE_CONFIG` case, delete the following line:

   ```
   configure_stream_local("DEBUG_FLOAT_ARRAY", 50.0f);
   ```

A.1.2 Sending and Receiving Signals Through Simulink®

To send internal Simulink® signals over the telemetry connection, the user must write to the 58 element `data` variable in the "DebugArray" uORB topic. Writing to this uORB topic through Simulink®, as shown in Fig. A1, sends the signal to the `data` variable in the "DEBUG_FLOAT_ARRAY" MAVLink message. This MAVLink message can then be accessed from a custom GCS built in Simulink®.

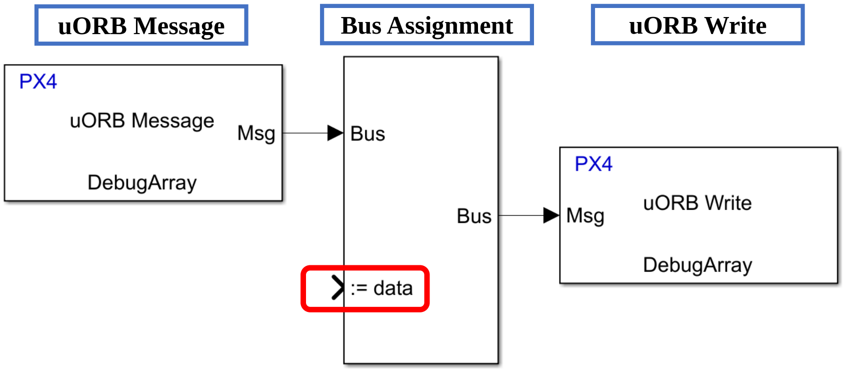
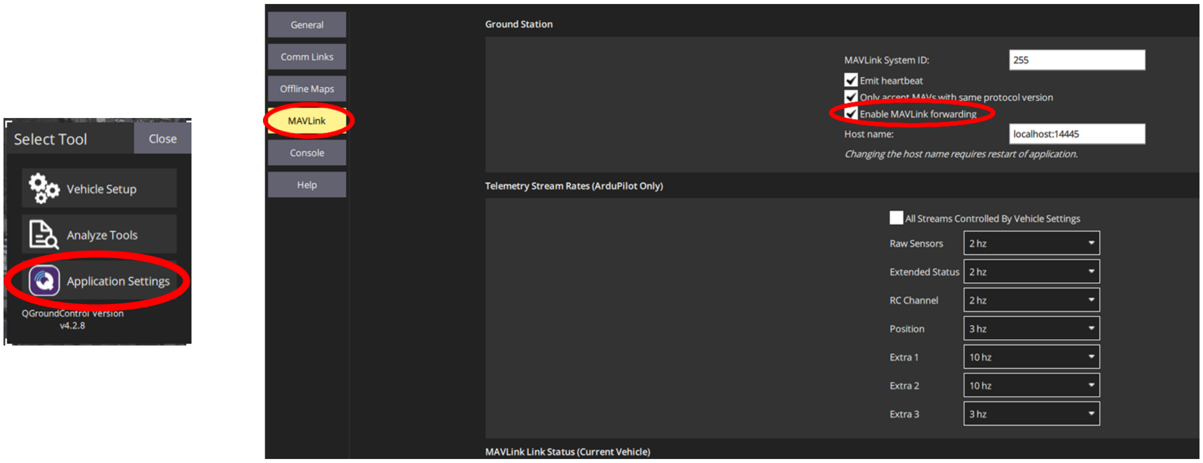


Figure A1: Simulink® logic for sending custom telemetry signals from the FCS.

To receive the signals sent over the telemetry connection, the user must first open QGC. Then, navigate to "Application Settings," and then "MAVLink." Ensure that MAVLink forwarding is enabled and note the IP address (host name). Next, there must be a successful connection between the telemetry radio connected to the GCS computer and the telemetry radio onboard the vehicle.



Ground Station

MAVLink System ID: 255

☑ Emit heartbeat

☑ Only accept MAVs with same protocol version

☑ Enable MAVLink forwarding

Host name: localhost:14445

Changing the host name requires restart of application.

Telemetry Stream Rates (ArduPilot Only)

□ All Streams Controlled By Vehicle Settings

Raw Sensors	2 hz
Extended Status	2 hz
RC Channel	2 hz
Position	3 hz
Extra 1	10 hz
Extra 2	10 hz
Extra 3	3 hz

MAVLink Link Status (Current Vehicle)

With QGC open, create a blank diagram in Simulink®. Using a "UDP Receive" block and a "MAVLink Deserializer" block, replicate the logic shown in Fig. A2. In the "UDP Receive" block, set the "Local IP port" to 14445 to match the IP address given by QGC. In the block parameters for the "MAVLink Deserializer" block, select the "DEBUG_FLOAT_ARRAY" MAVLink topic. Using a bus selector, select the `data` variable to access the vector of 58 signals sent from the vehicle.

The telemetry radios used for both the GCS and Pixhawk® were configured with a baud rate of 230400 using the RFD Tools program [47]. In PX4, the telemetry stream over the TELEM1 port was configured with the same baud rate using the `SER_TEL1_BAUD` PX4 parameter. The best streaming rate found was 11000 Bites/second and was set using the `MAV_0_RATE` PX4 parameter.

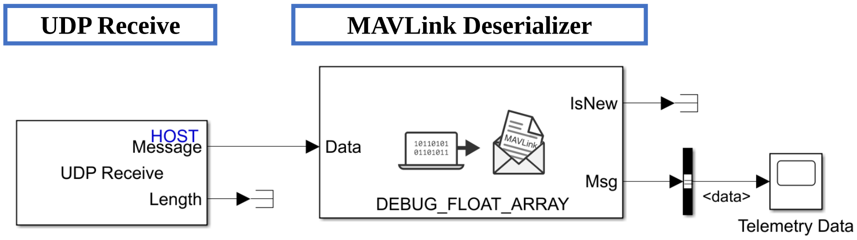


Figure A2: Simulink® logic for receiving custom telemetry signals in a custom ground control station diagram.

A.1.3 Custom Telemetry Signals

A complete list of the telemetry signals sent to the ground control station for the SEARCH vehicle is shown in Table A1. Telemetry signals are sent at a target frequency of 15 Hz, but the actual rates can fluctuate based on connection strength. On average, recorded rates were between 10-12 Hz during SEARCH flight tests.

Table A1: List of SEARCH vehicle telemetry signals sent to ground control station.

Grouping	No.	Variable	Description
Flight Time	1	t	Total flight time measured since vehicle arming
PCM	2	VV_{cmd}	Vertical velocity command
	3	acc_{cmd}	Acceleration command
	4	$\dot{\psi}_{cmd}$	Turn rate command
	5	v_{cmd}	Lateral velocity command
	6	p_{lat}	Lateral stick input
	7	p_{lon}	Longitudinal stick input
	8	p_{dir}	Directional stick input
	9	p_{acc}	Acceleration stick input
	10	h_{cmd}	Altitude hold anchor
	11	ψ_{cmd}	Heading hold anchor
	12	V_{cmd}	Velocity (hold) command
	13	x_{cmd}	Longitudinal position hold anchor
	14	y_{cmd}	Lateral position hold anchor
	15	ALT	Altitude hold engagement flag
	16	HDG	Heading hold engagement flag
	17	SPD	Velocity hold engagement flag
	18	TKO	Automatic takeoff engagement flag
	19	POS	Position hold engagement flag
TFC	20	$(T/W)_{lp,cmd}$	Lift propulsor thrust-to-weight command
	21	$(T/W)_{mp,cmd}$	Main propulsor thrust-to-weight command
	22	$\theta_{nac,cmd}$	Nacelle angle command
	23	$(T/W)_{cmd}$	Total thrust-to-weight command
	24	ζ_{nac}	Nacelle blending term
	25	$TFC\ Mode$	TFC mode indicator
EMF	26	θ_{cmd}	Pitch attitude command
	27	ϕ_{cmd}	Bank angle command
	28	r_{cmd}	Yaw rate command
	29	lat_{cmd}	Normalized lateral command
	30	lon_{cmd}	Normalized longitudinal command
	31	dir_{cmd}	Normalized directional command
IPPM	32	$LPRPM_{cmd}$	Lift propulsor RPM command
	33	$MPRPM_{cmd}$	Main propulsor RPM command
Safety	5	GEO	Geofence breach flag
	34	n_{geo}	Number of geofence coordinates enabled
	35	$Lost-Link$	Transmitter lost-link flag
	36	$CEILING$	Ceiling (altitude limit) breach flag
Sensors	37-57	$--$	Sensed vehicle states (omitted for brevity)
Mode	58	$--$	Flight mode switch setting

A.2 Custom Parameters

To add custom parameters in the PX4 firmware:

1. Navigate to the `<px4FirmwareRoot>\src\modules\battery_status\module.yaml` file.

2. Add desired parameters beneath the existing parameters (do not remove the existing battery parameters). Refer to the example parameter code below that shows the definition of a parameter group `IMPACT` under which the example `IMPACT_KP_SCALE` and `IMPACT_KI_SCALE` parameters are defined.

Example Parameter Code

```
group: IMPACT
  definitions:
    IMPACT_KP_SCALE:
      description:
        short: Kp Scale Factor
        long: |
          Scale factor for Kp gains
      type: float
      decimal: 3
      default: 3.5
      min: 0
      max: 500

    IMPACT_KI_SCALE:
      description:
        short: Ki Scale Factor
        long: |
          Scale factor for Ki gains
      type: float
      decimal: 3
      default: 3
      min: 0
      max: 500
```

3. After the custom parameter code is saved to the `module.yaml` file, proceed through the Hardware Setup Screens in the MathWorks® UAV Toolbox to rebuild the PX4 firmware. This is accomplished in MATLAB® by navigating to Home → Add-Ons → Manage Add-Ons and clicking on the gear icon next to "UAV Toolbox Support Package for PX4 Autopilots."

Once these steps are completed, the parameters are able to be read using the UAV Toolbox Support Package for PX4 Autopilots "Read Parameter" block (e.g., as shown in Fig. A3 for the `IMPACT_KP_SCALE` parameter). The parameters will build and appear in QGC if called by the Read Parameter block in the Simulink® diagram deployed to the vehicle.

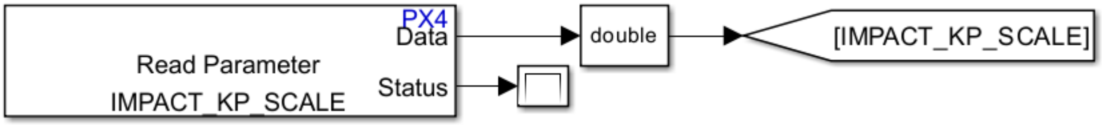


Figure A3: IMPACT read PX4 parameter example.

A.3 Flash Memory

The flash memory on the Pixhawk® flight computer is limited, and complex Simulink® diagrams can require more flash memory than the space available when using the unmodified PX4 code. To reduce flash memory usage from the default PX4 modules, certain PX4 modules that are not required can be disabled (note that certain modules are only able to be removed when using a custom Simulink® controller that replaces the PX4 controller, as is the case in this report). The following modifications were made to disable certain unnecessary modules for the fixed-wing build used in this work:

1. Navigate to the `<px4FirmwareRoot>\boards\cubepilot\cubeorange \fixedwing.px4board.original` file.

2. Modify the following lines to disable unnecessary modules:

```
CONFIG_DRIVERS_CAMERA_CAPTURE=n
CONFIG_DRIVERS_CAMERA_TRIGGER=n
CONFIG_DRIVERS_DSHOT=n
CONFIG_MODULES_CAMERA_FEEDBACK=n
CONFIG_MODULES_FW_ATT_CONTROL=n
CONFIG_MODULES_FW_AUTOTUNE_ATTITUDE_CONTROL=n
CONFIG_MODULES_FW_POS_CONTROL=n
CONFIG_MODULES_MC_AUTOTUNE_ATTITUDE_CONTROL=n
CONFIG_MODULES_MC_HOVER_THRUST_ESTIMATOR=n
CONFIG_MODULES_VTOL_ATT_CONTROL=n
```

These changes increase the amount of available flash memory by preventing the unused PX4 modules from being built onto the Pixhawk®.

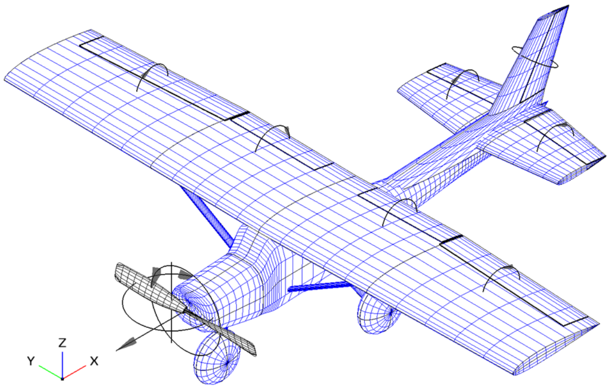
Appendix B

Flight Simulation Visualization Tutorial

This appendix describes an approach to create a piloted desktop flight simulation using X-Plane 11 [26] and Simulink® [25]. The aircraft geometry is created using OpenVSP [41] and processed using Blender [45] to convert the geometry into a format compatible with X-Plane. Flight simulations are then executed with the flight physics model and control system running in Simulink®, while the aircraft movement is visualized in X-Plane 11. An X-Plane Simulink® Blockset [27] is used for communication between Simulink® and X-Plane.

B.1 Creating Aircraft Geometry Using OpenVSP

1. Create an OpenVSP model of the aircraft outer mold line without independent control surfaces, or with control surfaces only defined as wing sub-surfaces. For example, the initial SEARCH OpenVSP model with control surfaces defined as wing sub-surfaces is shown below.



2. Create control surfaces as separate components. The recommended steps to create a control surface on a wing are described below. The following video also explains how to define control surfaces in OpenVSP: https://www.youtube.com/watch?v=zOoXH5xoZXo (from 9:39 to 14:37).

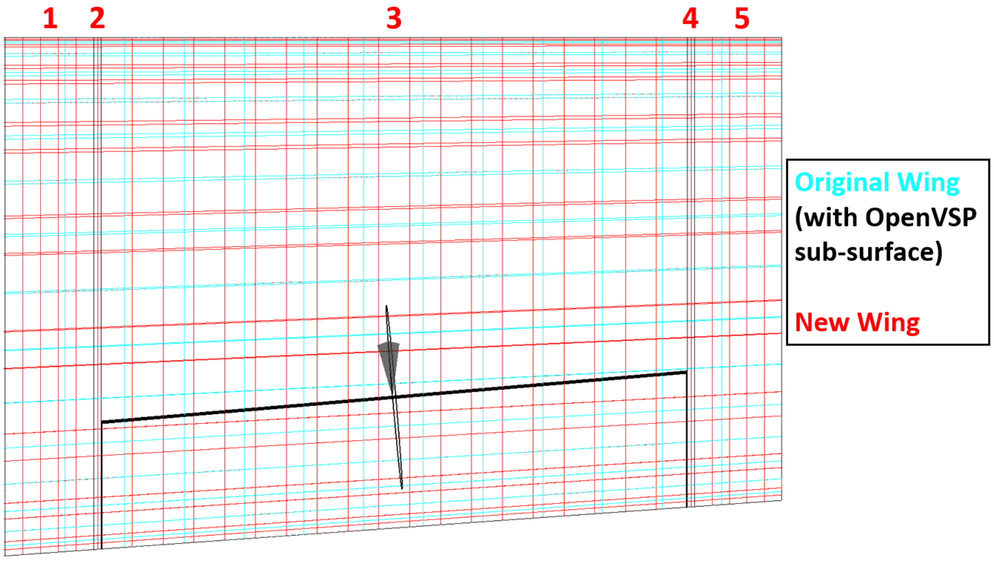
   (a) Make a copy of the original wing. Use the original wing as a reference and the new wing to create cutouts for control surfaces.

   (b) Split the new wing into additional sections, as required by the control surface geometry. For example, a single control surface on the interior of a single-section wing requires five wing sections.

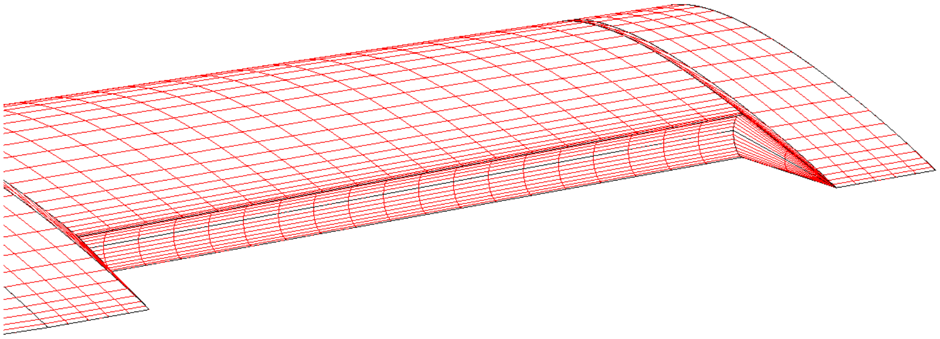
   (c) Adjust the span length of each wing section to specify the spanwise start and end of the control surface. There should be a small wing section located at the spanwise start and end of the control surface. Additional tips:

   - Adjust each section span length for the new wing under the “Sect” tab sequentially from the most inboard section to the most outboard section using the original wing as a reference.

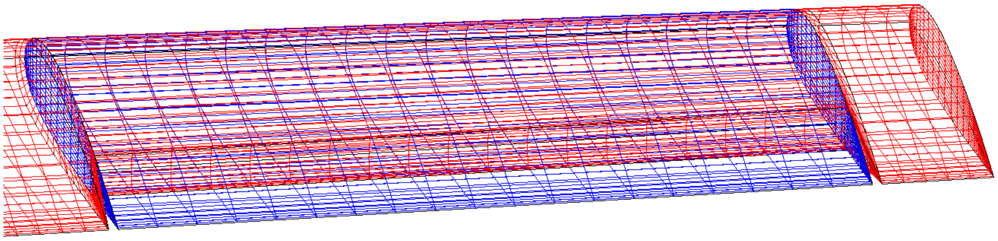
- Modify the section span while holding the “Root C” (Root Chord), and “Sec SW” (Secondary Sweep of Wing Section) parameters constant.
- For each sequential section used for defining a control surface along a single wing section, copy the “Sec SW” parameter from the previous section.



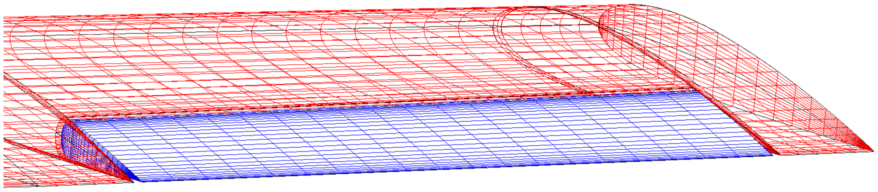
(d) Under the “Modify” tab, trim the trailing edge for the airfoil sections at the start and end of the control surface (under “Trailing Edge” set “Trim” to “X” and adjust the relative “X” length to form the gap for the control surface). Next, make the trailing edge cap for the airfoil sections at the start and end of the control surface rounded to allow for control surface rotation (under “Trailing Edge” set “Cap” to “ROUND” and adjust the “Length” to be a negative value.)



(e) Create copy of the new wing with control surface gaps to use as the control surface. Under the “Sect” Tab, “Cut” the wing sections around the control surface cutout (retain the section where the control surface is defined). Remove the trimmed trailing edge under the “Modify” tab. Adjust the control surface wing section location under the “Xform” tab to place the control surface wing section in its appropriate location relative to the wing.

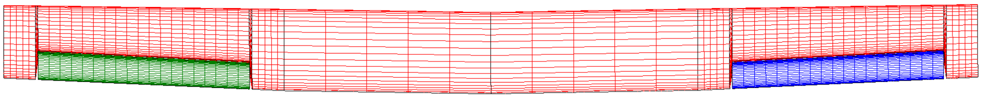


(f) Under the "Modify" tab, trim the leading edge of the control surface airfoil sections (under "Leading Edge" set "Trim" to "X" and adjust the relative "X" length to form the control surface). Next, make the leading edge cap rounded (under "Leading Edge" set "Cap" to "ROUND" and adjust the "Length" to be a positive value.)

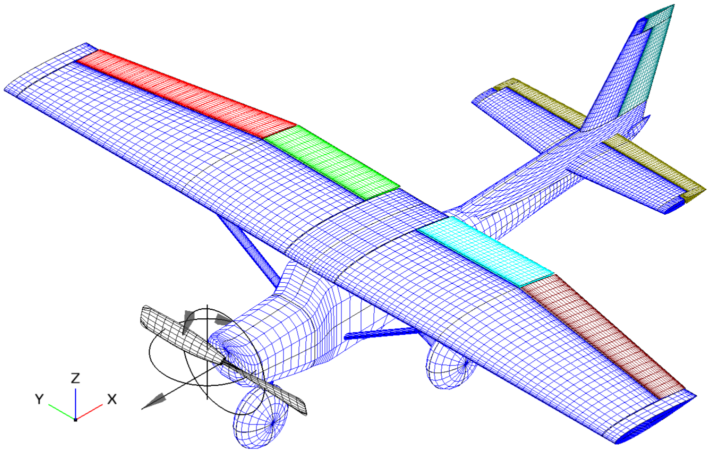


(g) The steps to create symmetric control surface as independent components are as follows:

   i. Disable planar symmetry for the control surface component.
   ii. Create a copy of the control surface component.
   iii. Rotate the replicate control surface by 180 deg around the axis of symmetry (for example, for a conventional horizontal wing, under the "XForm" tab, set "XRot" to be "180" deg, if the original "XRot" value is "0" deg).
   iv. Translate the control surface location around the axis of symmetry (for a conventional horizontal wing, under the "XForm" tab, change the "Yloc" sign to be opposite of the original control surface).
   v. Under the "Airfoil" tab, check "invert airfoil" for each airfoil section.
   vi. The incidence, twist, and/or dihedral parameters may require a sign change depending on how the wing is designed.



The final SEARCH OpenVSP model with individual control surfaces is shown below.



3. Export each component from OpenVSP to a .stl file using the default settings. All non-moving components (fuselage, wing, etc.) can be exported together. Each control effector (control surfaces, propellers, etc.) should be exported separately so that they can be independently animated.

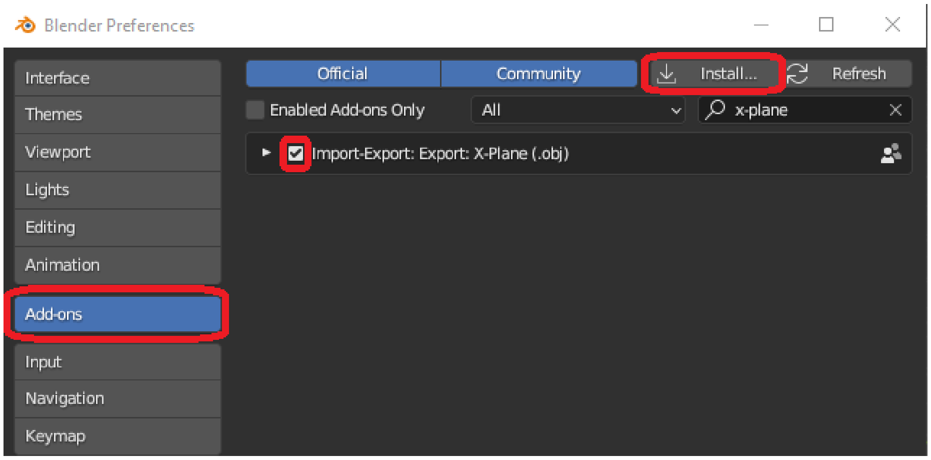
B.2 Blender Setup

1. Download the "XPlane2Blender" Blender add-on (https://developer.x-plane.com/tools/blender/), which is available for download here: https://github.com/X-Plane/XPlane2Blender/releases. XPlane2Blender v4.2.0-rc-3 was used for this tutorial using the file called "io_xplane2blender_4_2_0-rc_3-117_20230225221007-v12.zip" (i.e., the file that is not labeled "source code").

2. Download and install Blender (https://www.blender.org/). Blender Version 3.6.9 was used for this tutorial.

3. Enable the XPlane2Blender add-on using the following steps (this video shows the process using a previous version of the software: https://www.youtube.com/watch?v=VwFVEcGI7Ws).

4. Within Blender, navigate to Edit → Preferences → Add-ons → Install. Select the XPlane2Blender .zip file downloaded above and enable the X-Plane add-on.



5. Restart Blender.

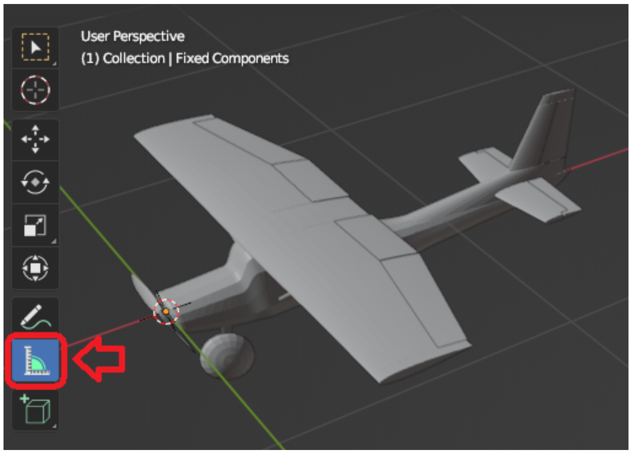
B.3 Painting an Aircraft Model in Blender

1. Open a new session of Blender.

2. Delete default objects by typing “a” to select all, then “x” to delete. Save the new session.

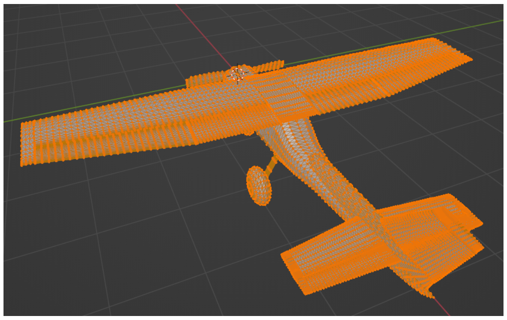
3. Load the .stl files exported from OpenVSP by navigating to File → Import → Stl (.stl). Select all of the components used to define the aircraft geometry and specify the “Scale” to convert the units into meters (for example, if the OpenVSP model units used to create the .stl files were in feet, then the scale should be set to a value of 0.3048.

4. Check that all .stl files have been properly imported. The units can be verify by using the “Measure” tool.



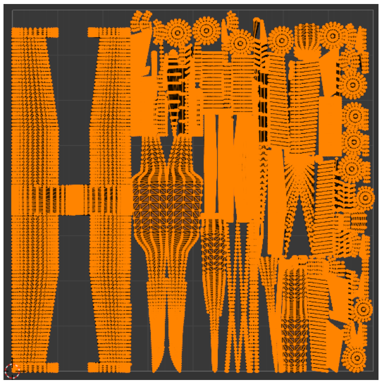
5. Under the “Layout” workspace, select all objects by typing “a”. Navigate to Object → Parent → Clear and Keep Transformation (this may be optional but is beneficial in some cases).

6. Click on “UV Editing” workspace. Then type “a” → “Tab” so that all components are selected. (The action of typing “a” → “Tab” may need to be repeated multiple times to select all components.) Ensure that the vehicle is fully covered with orange dots.



7. With all vehicle components selected, click Mesh → Normals → Recalculate Outside

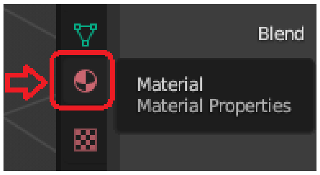
8. With all vehicle components selected, click "UV" → "Smart UV Project" → OK (using default settings). Note: these settings should be adjusted if the textured paint step below is not yielding satisfactory results. An image similar to the one below should appear on the left pane.



9. Create a blank 2048×2048 pixel canvas image file. This can be accomplished using the Windows Paint program, among other methods. Open Paint, navigate to File → Properties, set both the width and height to be 2048 pixels, and save as a ".png" file. The file used for this tutorial is named "SEARCH_Blender_Canvas.png".

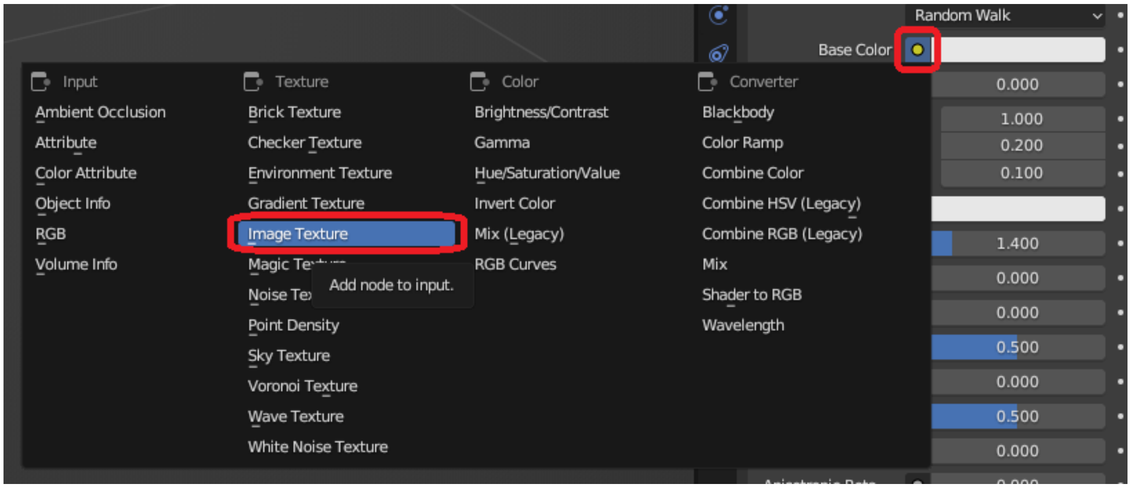
10. Click on the "Texture Paint" workspace. Load the blank canvas into Blender by clicking "Open" on the left pane and selecting the image file.

11. Navigate to the "Layout" workspace, select a single (arbitrary) component, and click on the "Material" icon (circled below).



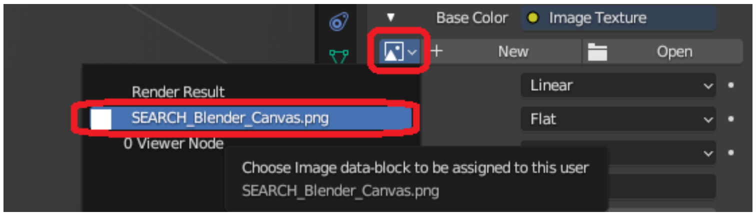
12. Click "New" to create a new material.

13. Under the "Surface" drop-down menu, click on the dot next to "Base Color" and select "Image Texture"



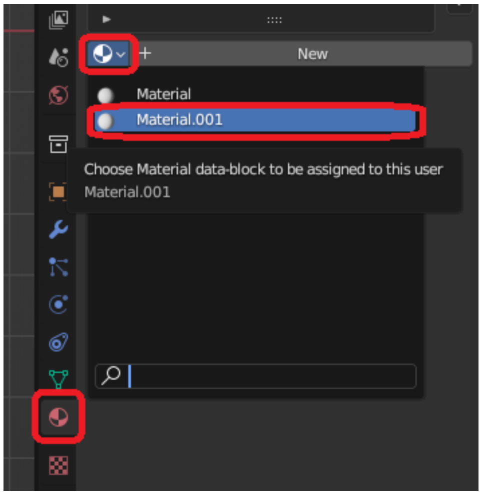
Add node to input.

14. On the line below “Base Color,” select the image icon and canvas, as shown below.



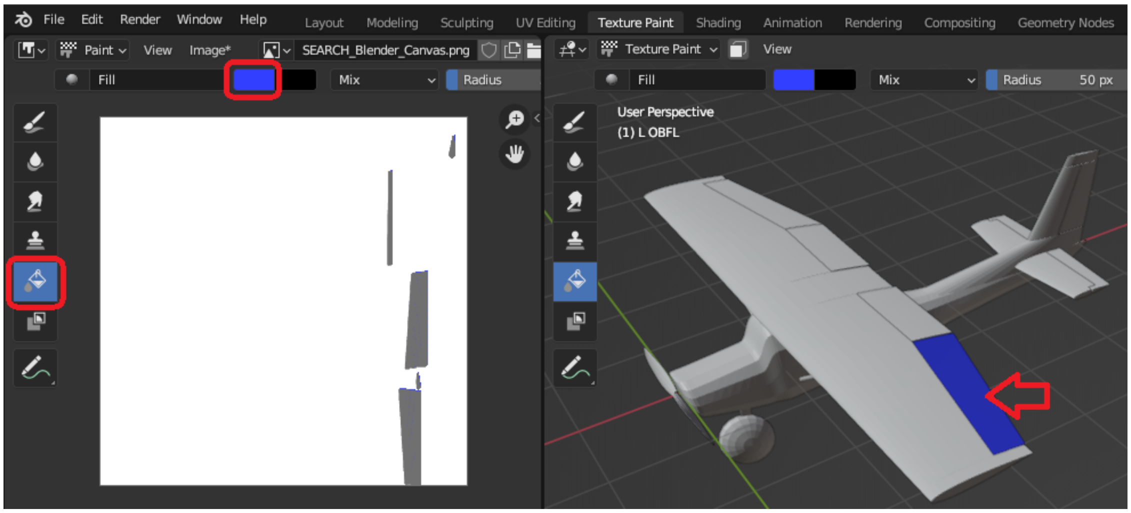
Choose Image data-block to be assigned to this user
SEARCH_Blender_Canvas.png

15. Individually select each additional component, click on the “Material” icon, and select the same material created above (e.g., “Material.001”).



Choose Material data-block to be assigned to this user
Material.001

16. To fill a component with a solid color, select the component to paint in the “Layout” workspace, navigate to the “Texture Paint” workspace, click on “Fill,” select the desired color, and click on the component in the right pane. Repeat for each component. Additional more complex painting options can also be employed to add additional detail.

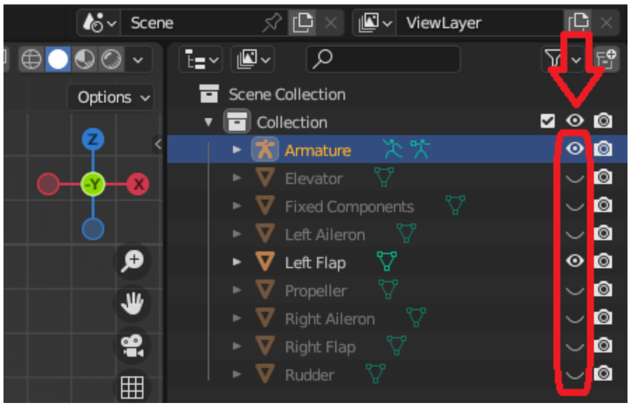


17. Once the image is completed, save the image by navigating to the "Texture Paint" workspace, selecting "Image" at the top of the right pane, and "save as" a new file name.

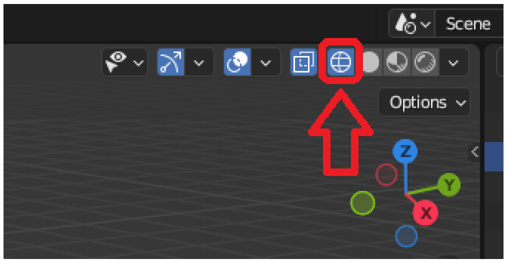
B.4 Animating Control Effectors in Blender

1. Navigate to the "Layout" workspace.

2. Hide everything except the component currently being animated and its armature using the "eye" icon on the right side of the screen (type "Shift+H" to hide all components; alternatively, type "alt+H" to make all components visible).

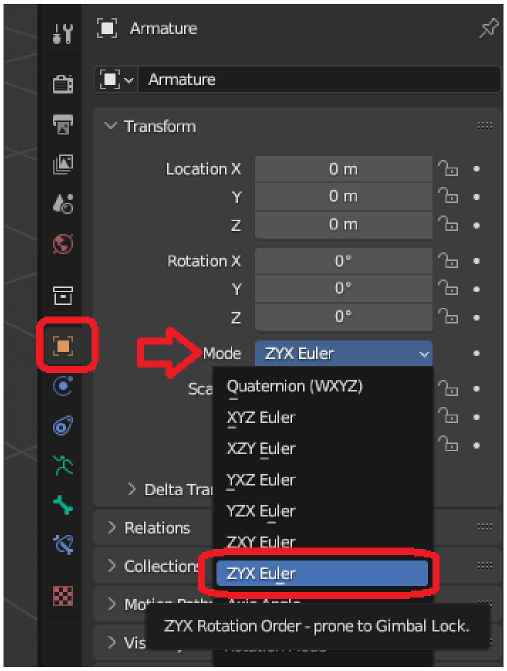


3. Turn on the wireframe by clicking on the Viewport Shading: Wireframe icon. This is helpful to align the armature with the hinge axis.



4. To add an Armature component, type "Shift+a" then select "Armature."

5. Select the armature, click on the "object" icon, and change to "ZYX Euler" mode. This is the rotation sequence used by X-Plane and is important to specify to avoid incorrect rotations in X-Plane. This step must be done for each new armature.



Armature

Armature

Transform

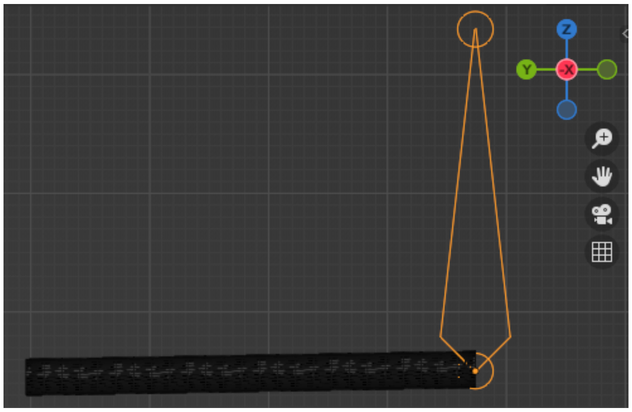
Location X 0 m
Y 0 m
Z 0 m

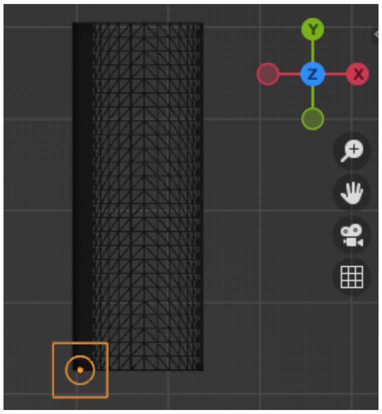
Rotation X 0°
Y 0°
Z 0°

Mode ZYX Euler

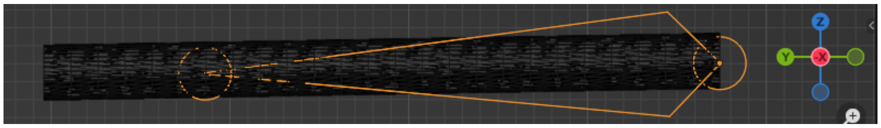
ZYX Rotation Order - prone to Gimbal Lock.

6. Select the armature then type "g" then to move it. Place the armature on the hinge line of the control surface. It is helpful to use preset rotations to place the armature accurately.

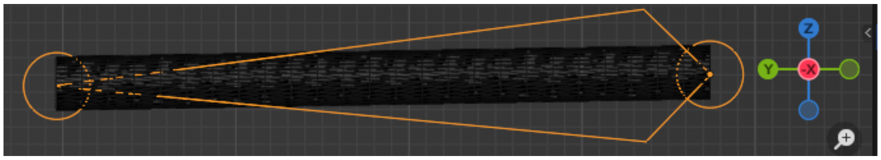




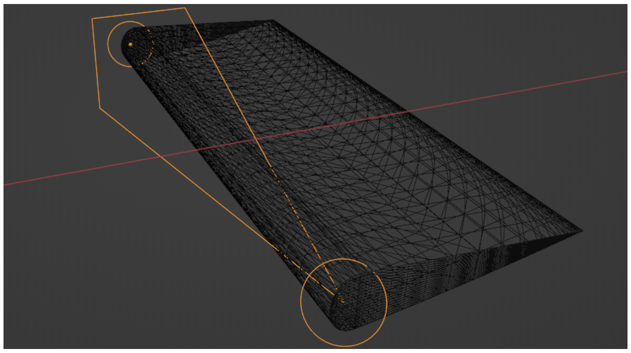
7. Type “r” to rotate about the current view to move the other end of the armature onto the hinge line of the control surface. Rotate the armature either by moving the mouse to the desired point. Alternatively, the rotation angle value can be specified in the “Object” tab. Again, it is helpful to use preset rotations to place the armature accurately.



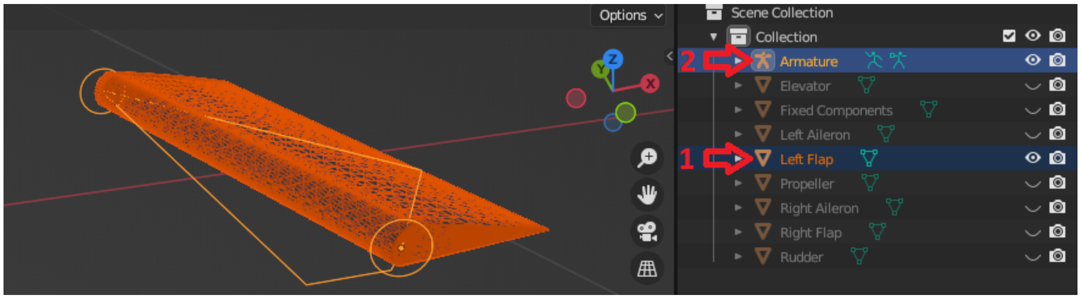
8. Type “s” to scale the armature to reach the other end of the control surface.



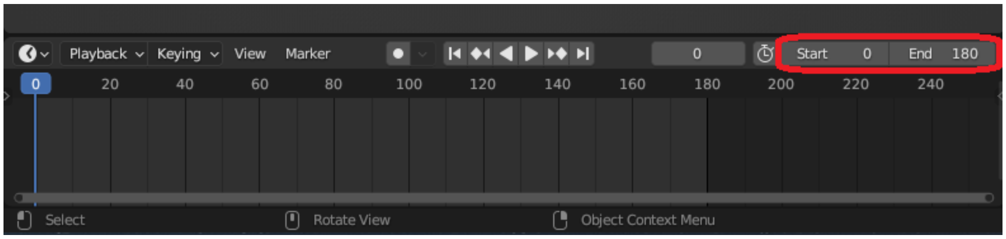
9. Check in each view to confirm that the armature is correctly placed along the hinge line.



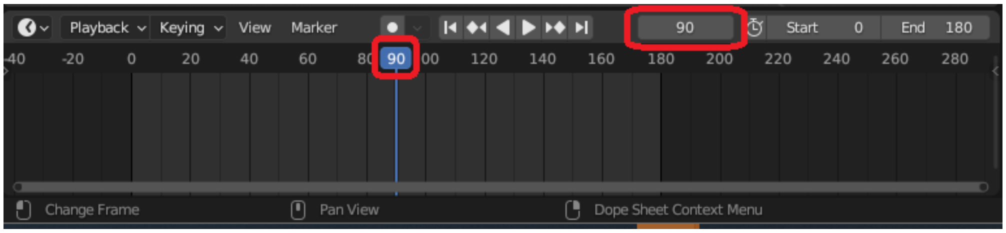
10. Select the control surface to be animated. Hold the “ctrl” key and click on the corresponding armature so that both are selected. Right click, then select “Parent” → “Object.” This step links the objects together.



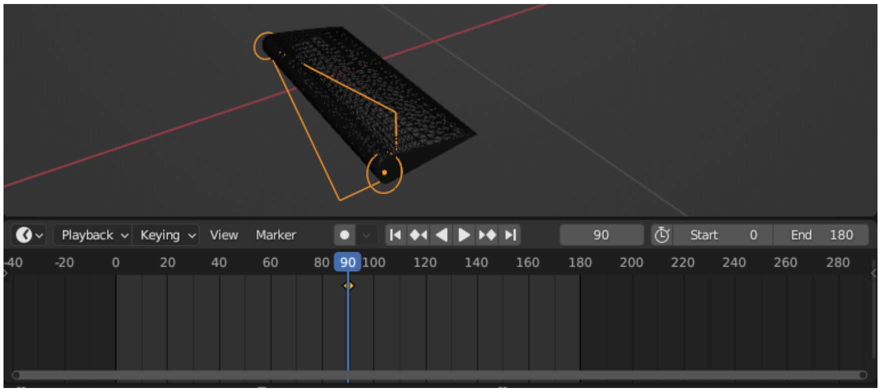
11. In the Animation Bar at the bottom of the screen, set the “Start” to 0 and “End” to 180.



12. Set 90 as the middle point by sliding the blue bar or typing the value into the "Current Frame" field on the Animation Bar.

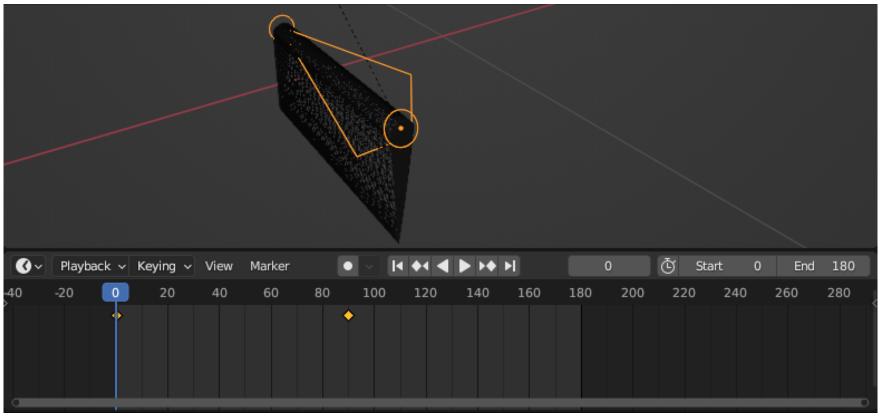


13. Select the Armature, then type "r" → "z" → "z" → "0" → "Enter" to set the neutral position. Then type "i" and select "Rotation." The armature "Transform" tab Rotation X, Y, Z pane will turn yellow and indicates that the key frame has been properly set.

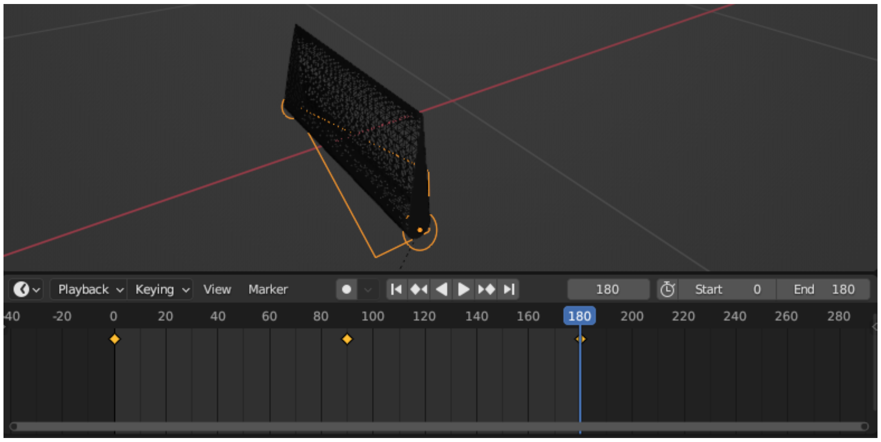


14. Set the start and end points in the Animation Bar using the following steps:

    (a) Move the blue bar to 0. Select the armature. Type "r" → "z" → "z" → "90" → "Enter" → "i" → "rotation".



(b) Move the blue bar to 180. Select the armature. Type "r" → "z" → "z" → "-90" → "Enter" → "i" → "rotation". Note that yellow diamonds appear on the animation pane for each point.



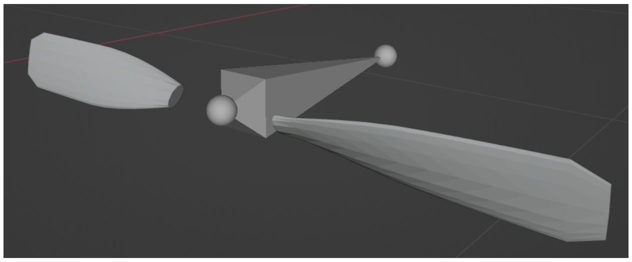
(c) To create a control surface that travels past ±90 deg, create and intermediate point following the above steps to ensure correct interpolation. (Note that the specified rotation values are relative to the previous rotation point.)

- Note that the sign convention is set in a later step. It is recommended to follow above the steps for simplicity (regardless of the sign convention).

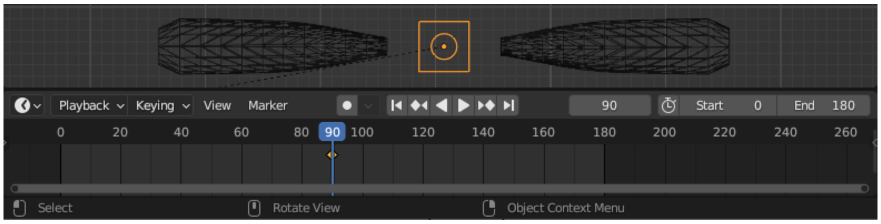
15. To rename the armature, select the armature, Click "Edit" → "Rename Active Item" or type "F2." Rename the armature to include the name of its corresponding control surfaces for bookkeeping purposes.

16. For multiple control surfaces, a previous armature can be copied-and-pasted (ctrl+c, crtl+v) to save time. After the new armature is copied-and-pasted, move the armature to its new location for its corresponding control surface. Then, right click and select "Parent" → "Object."

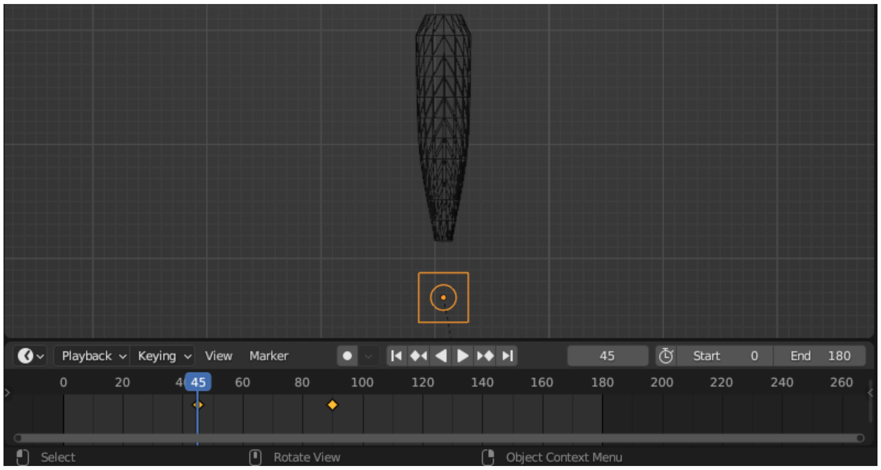
17. For a propeller, follow similar steps to above for a new armature (add a new armature, change to "ZYX Euler" mode, translate, rotate, and link to propeller). The following substeps demonstrate how to rotate the propeller 360 deg.



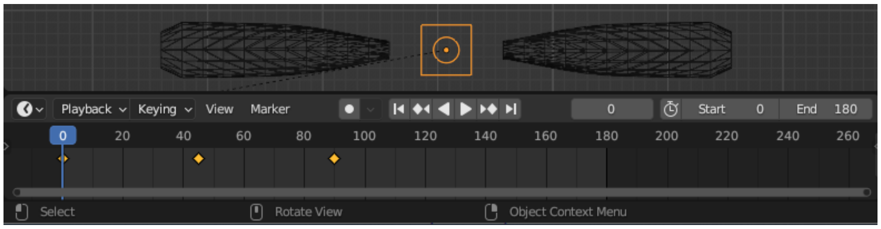
(a) Set the blue bar to 90. Select the armature. Type "r" → "z" → "z" → "0" → "Enter" → "i" → "Rotation" to set the neutral position.



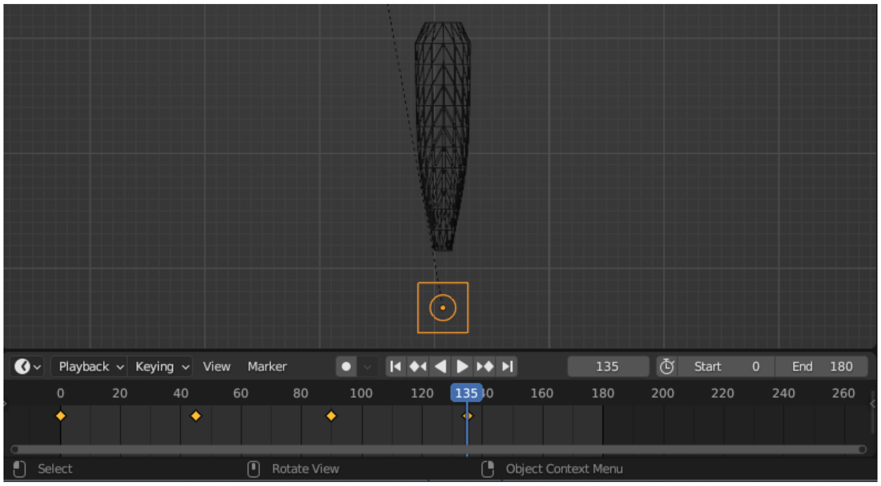
(b) Move the blue bar to 45. Select the armature. Type "r" → "z" → "z" → "90" → "Enter" → "i" → "Rotation" to set the 90 deg angle.



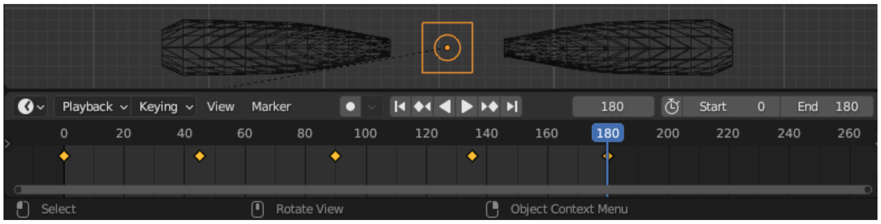
(c) Move the blue bar to 0. Select the armature. Type "r" → "z" → "z" → "90" → "Enter" → "i" → "Rotation" to set the 180 deg angle. (Note that the rotation is relative to the previous point.)



(d) Move the blue bar to 135. Select the armature. Type "r" → "z" → "z" → "-90" → "Enter" → "i" → "Rotation" to set the -90 deg angle.

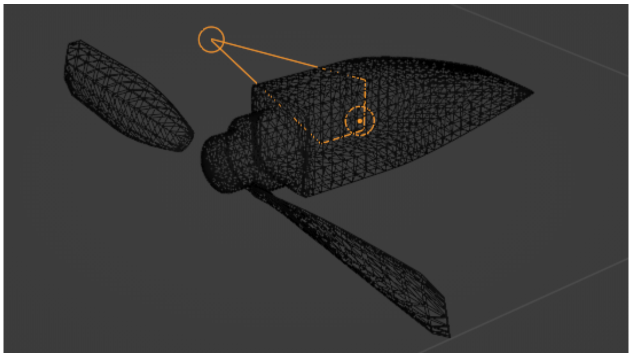


(e) Move the blue bar to 180. Select the armature. Type "r" → "z" → "z" → "-90" → "Enter" → "i" → "Rotation" to set the -180 deg angle.



18. VTOL vehicles commonly require a component to be assigned multiple rotations (for example, a tilting propeller). The following sub-steps demonstrate how to add a second rotation of a tilting propeller that rotates with a nacelle from 0 to 180 deg. (These steps assume that the propeller armature was already created using the above steps.)

    (a) Create a new armature, change to "ZYX Euler" mode, and place at the propeller/nacelle rotation point.



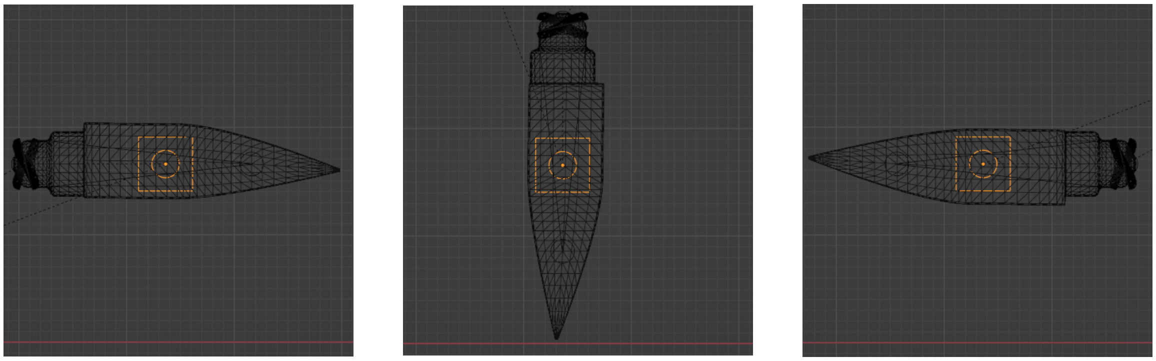
    (b) Hold "shift" or "ctrl" and select the previous propeller armature and nacelle, then the new armature. Right click, select "Parent" → "Object." Note that the propeller armature, nacelle, and new armature must all be visible in the workspace for this step to work correctly.

    (c) Set rotation points following these steps

   i. Move the blue bar to 0. Select the armature. Type “r” → “z” → “z” → “0” → “Enter” → “i” → “Rotation” to set nominal point.

   ii. Move the blue bar to 90. Select the armature. Type “r” → “z” → “z” → “90” → “Enter” → “i” → “Rotation” to set the 90 deg point.

   iii. Move the blue bar to 180. Select the armature. Type “r” → “z” → “z” → “90” → “Enter” → “i” → “Rotation” to set the 180 deg point.

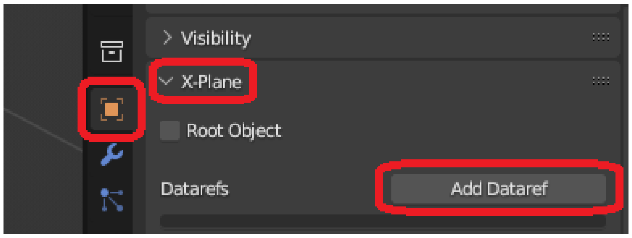


B.5 Assigning an X-Plane Dataref to Each Armature in Blender

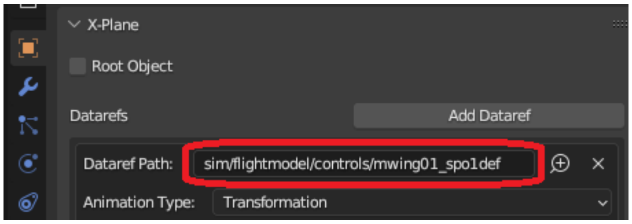
1. Each armature must be assigned to an X-Plane “Dataref” to allow for control surface movement in X-Plane. A list of 38 available X-plane Datarefs are given in the table below. Select and document the Dataref for each control effector. A full list of X-plane Datarefs can be found at: https://www.siminnovations.com/xplane/dataref/.

sim/flightmodel/controls/mwing01_elv1def	sim/flightmodel/controls/mwing03_rud2def
sim/flightmodel/controls/mwing01_elv2def	sim/flightmodel/controls/mwing03_spo1def
sim/flightmodel/controls/mwing01_rud1def	sim/flightmodel/controls/mwing03_spo2def
sim/flightmodel/controls/mwing01_rud2def	sim/flightmodel/controls/mwing04_ail1def
sim/flightmodel/controls/mwing01_spo1def	sim/flightmodel/controls/mwing04_ail2def
sim/flightmodel/controls/mwing01_spo2def	sim/flightmodel/controls/mwing04_elv1def
sim/flightmodel/controls/mwing02_ail1def	sim/flightmodel/controls/mwing04_elv2def
sim/flightmodel/controls/mwing02_ail2def	sim/flightmodel/controls/mwing04_rud1def
sim/flightmodel/controls/mwing02_elv1def	sim/flightmodel/controls/mwing04_rud2def
sim/flightmodel/controls/mwing02_elv2def	sim/flightmodel/controls/mwing04_spo1def
sim/flightmodel/controls/mwing02_rud1def	sim/flightmodel/controls/mwing04_spo2def
sim/flightmodel/controls/mwing02_rud2def	sim/flightmodel/controls/mwing05_ail1def
sim/flightmodel/controls/mwing02_spo1def	sim/flightmodel/controls/mwing05_ail2def
sim/flightmodel/controls/mwing02_spo2def	sim/flightmodel/controls/mwing05_elv1def
sim/flightmodel/controls/mwing03_ail1def	sim/flightmodel/controls/mwing05_elv2def
sim/flightmodel/controls/mwing03_ail2def	sim/flightmodel/controls/mwing05_rud1def
sim/flightmodel/controls/mwing03_elv1def	sim/flightmodel/controls/mwing05_rud2def
sim/flightmodel/controls/mwing03_elv2def	sim/flightmodel/controls/mwing05_spo1def
sim/flightmodel/controls/mwing03_rud1def	sim/flightmodel/controls/mwing05_spo2def

2. Select an armature in the “Layout” workspace, navigate to the “Object” tab, open the “X-Plane” menu, and click “Add Dataref.”



3. Paste the name of the assigned Dataref into “Dataref Path” to assign the animation name in X-Plane.



Dataref Path: sim/flightmodel/controls/mwing01_spo1def

Animation Type: Transformation

4. Set the deflection values interpreted by X-Plane (with the appropriate sign convection) by adding X-Plane Dataref keyframes for each keyframe set above. The Blender and X-Plane Dataref keyframes for each control effector shown above are as follows.

Control Surface:

Blender Keyframe Value	X-Plane Dataref Keyframe Value
0	90
90	0
180	-90

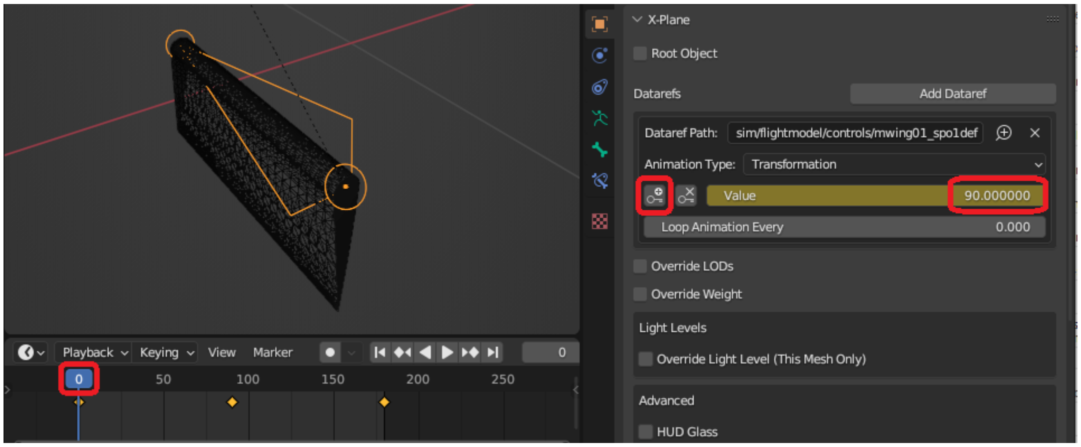
Propeller:

Blender Keyframe Value	X-Plane Dataref Keyframe Value
0	0
45	90
90	180
135	270
180	360

Tilting Nacelle:

Blender Keyframe Value	X-Plane Dataref Keyframe Value
0	0
90	90
180	180

The steps are shown visually below for the control surface. Note that the X-Plane Dataref keyframe value box will turn gold when the value is correctly entered/assigned.



X-Plane

□ Root Object

Datarefs — Add Dataref

Dataref Path: sim/flightmodel/controls/mwing01_spo1def

Animation Type: Transformation

Value 90.000000

Loop Animation Every 0.000

□ Override LODs

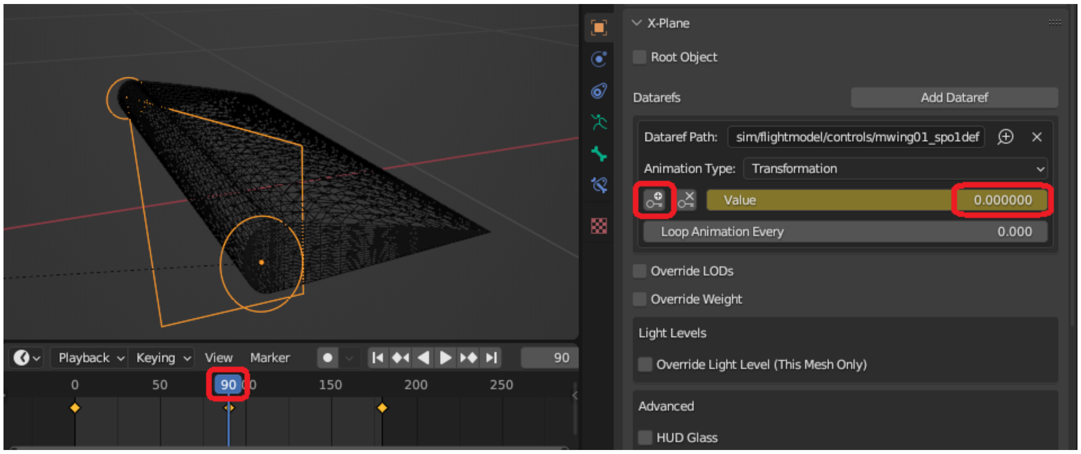
□ Override Weight

Light Levels

□ Override Light Level (This Mesh Only)

Advanced

□ HUD Glass



X-Plane

□ Root Object

Datarefs — Add Dataref

Dataref Path: sim/flightmodel/controls/mwing01_spo1def

Animation Type: Transformation

Value 0.000000

Loop Animation Every 0.000

□ Override LODs

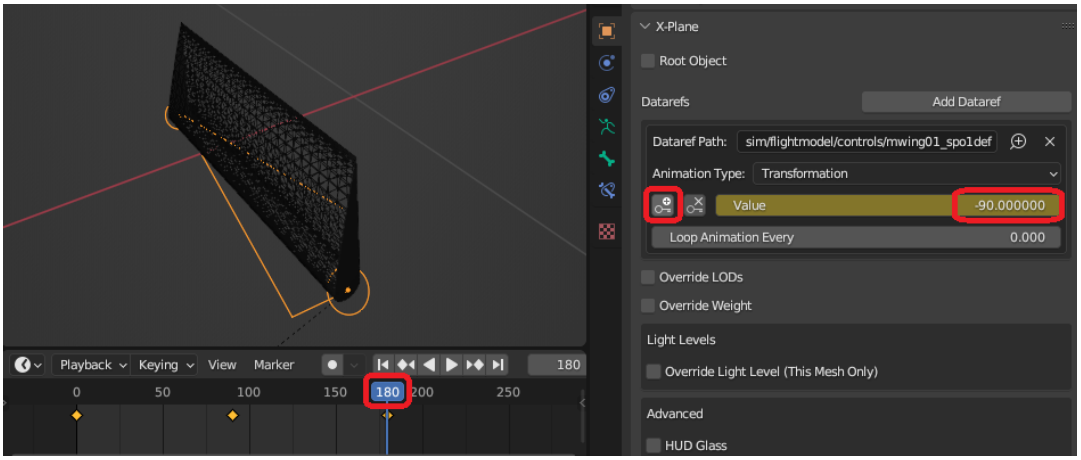
□ Override Weight

Light Levels

□ Override Light Level (This Mesh Only)

Advanced

□ HUD Glass



X-Plane

□ Root Object

Datarefs — Add Dataref

Dataref Path: sim/flightmodel/controls/mwing01_spo1def

Animation Type: Transformation

Value -90.000000

Loop Animation Every 0.000

□ Override LODs

□ Override Weight

Light Levels

□ Override Light Level (This Mesh Only)

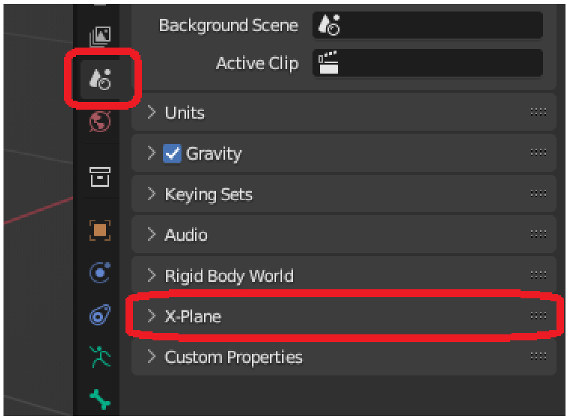
Advanced

□ HUD Glass

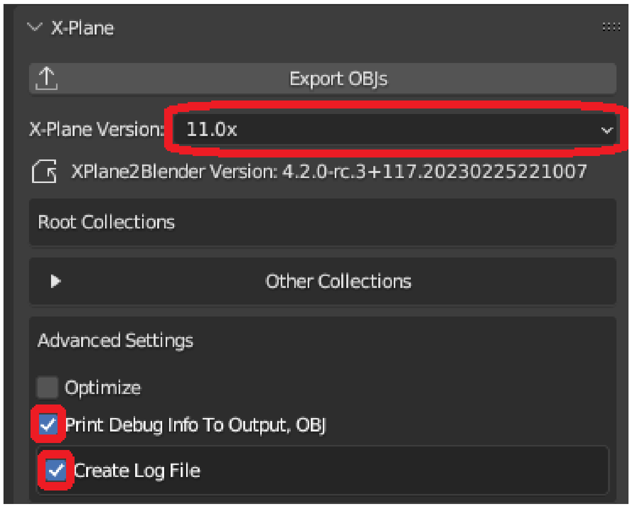
B.6 Exporting from Blender into Format Accepted by X-Plane

1. Ensure all aircraft components are visible (type “alt+H” to make all components visible).

2. Select the “Scene” tab, then open the “X-Plane” tab. Select the following options.



(a) Select X-Plane version "11.0x," Check "Print Debug Info To Output, OBJ," and check "Create Log File."



X-Plane Version: 11.0x

XPlane2Blender Version: 4.2.0-rc.3+117.20230225221007

☐ Optimize

☑ Print Debug Info To Output, OBJ

☑ Create Log File

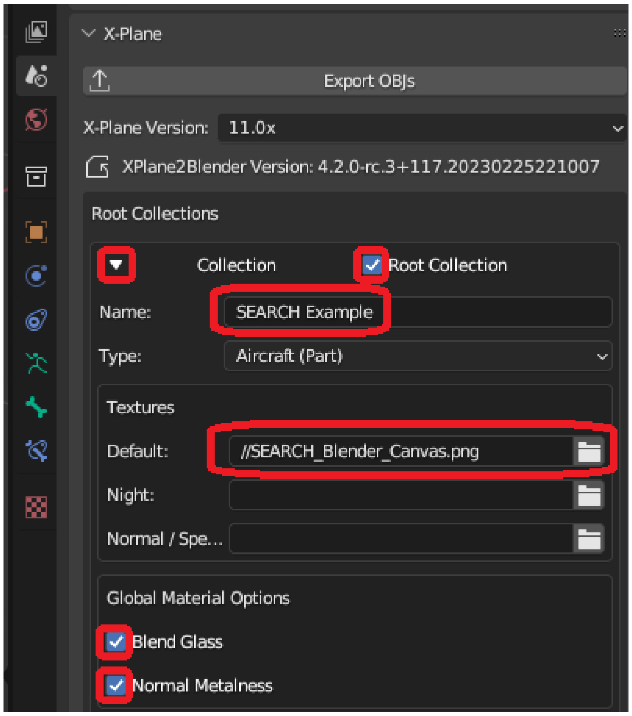
(b) Under the "Collection" tab:

i. Check "Root Collection."

ii. Enter an aircraft name, which will serve as the output file name.

iii. Under "Textures"→"Default", select the image texture file created above.

iv. Check "Blend Glass" and "Normal Metalness."



X-Plane

Export OBJs

X-Plane Version: 11.0x

XPlane2Blender Version: 4.2.0-rc.3+117.20230225221007

Root Collections

Collection ☑ Root Collection

Name: SEARCH Example

Type: Aircraft (Part)

Textures

Default: //SEARCH_Blender_Canvas.png

Night:

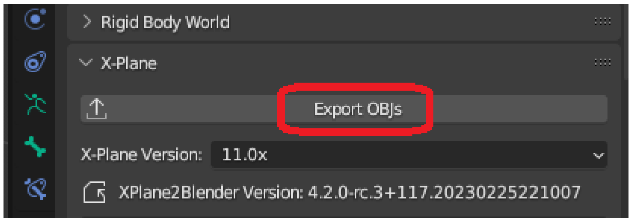
Normal / Spe...

Global Material Options

☑ Blend Glass

☑ Normal Metalness

3. Click "Export OBJs" to create a new ".obj" file, which will be located in the same folder as the blender file.



Rigid Body World

X-Plane

Export OBJs

X-Plane Version: 11.0x

XPlane2Blender Version: 4.2.0-rc.3+117.20230225221007

B.7 Creating a new file in Plane Maker

1. Create a starting directory to modify. For example, this can be taken from ".../X-Plane 11/Aircraft/Laminar Research/Cessna 172P" model. Copy the "Cessna 172P" directory to a new directory name under "Laminar Research" or "Extra Aircraft" to avoid damaging the original aircraft. The remaining steps use this Cessna 172P example. Using an existing aircraft simplifies the setup process.

2. Copy the ".obj" file and ".png" texture file created above using Blender into the new X-Plane aircraft directory.

3. Run the "Plane Maker.exe" executable file located in the "X-Plane 11" installation folder.

4. Open the "Cessna_172P.acf" file from the new directory.

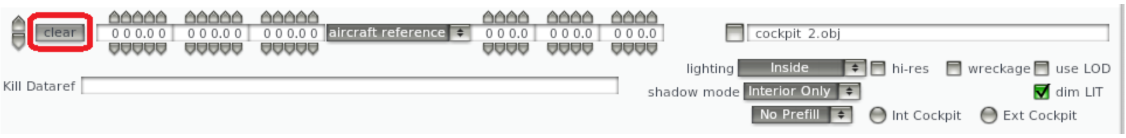
5. To enable control effector movement, under the "Special" tab, click "Show With Still / Moving Controls." Having the control effectors moving helps to verify that the model was correctly imported.

6. To change the view around the aircraft, use the following commands:

- "+" zooms in ("Shift +" zooms in faster)
- "-" zooms out ("Shift -" zooms out faster)
- "a," "w," "s," and "d" to rotate the aircraft

7. Under the "Standard" tab, select "Misc Objects."

8. Clear all objects except for "wing.obj."



clear 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | aircraft reference | 0 0 0.0 | 0 0 0.0 | 0 0 0.0

□ cockpit 2.obj

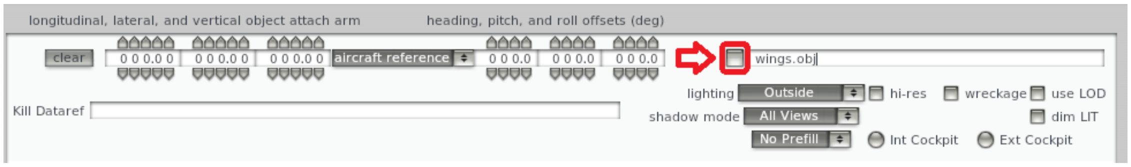
lighting: Inside □ hi-res □ wreckage □ use LOD

Kill Dataref

shadow mode: Interior Only ☑ dim LIT

No Prefill ○ Int Cockpit ○ Ext Cockpit

9. Replace "wing.obj" with the .obj file that was exported from blender above.



longitudinal, lateral, and vertical object attach arm

heading, pitch, and roll offsets (deg)

clear 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | aircraft reference | 0 0 0.0 | 0 0 0.0 | 0 0 0.0

□ wings.obj

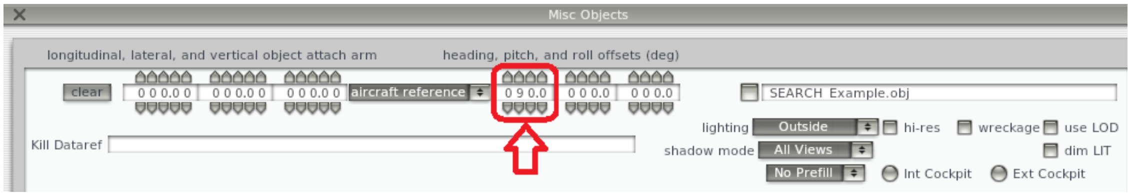
lighting: Outside □ hi-res □ wreckage □ use LOD

Kill Dataref

shadow mode: All Views □ dim LIT

No Prefill ○ Int Cockpit ○ Ext Cockpit

10. Change the heading offset to 90 deg.



Misc Objects

longitudinal, lateral, and vertical object attach arm

heading, pitch, and roll offsets (deg)

clear 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | aircraft reference | 0 9 0.0 | 0 0 0.0 | 0 0 0.0

□ SEARCH Example.obj

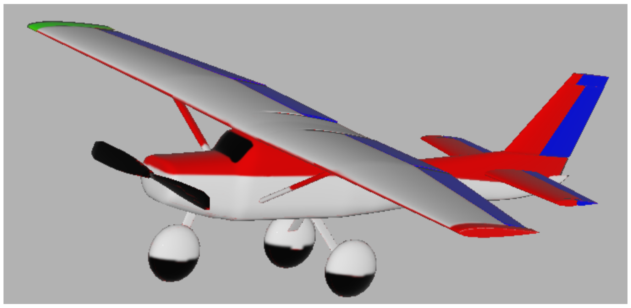
lighting: Outside □ hi-res □ wreckage □ use LOD

Kill Dataref

shadow mode: All Views □ dim LIT

No Prefill ○ Int Cockpit ○ Ext Cockpit

11. Ensure that the animation looks correct.



12. Edit aircraft name information by navigating to "Standard"→"Author."

13. Under "Standard"→"Engine Specs"→"Engine 2," set the "#engn" parameter to "0" to remove the Cessna 172P propeller component.

14. Adjust the aircraft center of gravity by navigating to "Standard"→"Weight & Balence"→"Weight & Bal"→"Center of Gravity."

15. Save the plane as a new file ".acf" file and exit Plane Maker.

16. Delete all files and folders in the new aircraft directory except for:

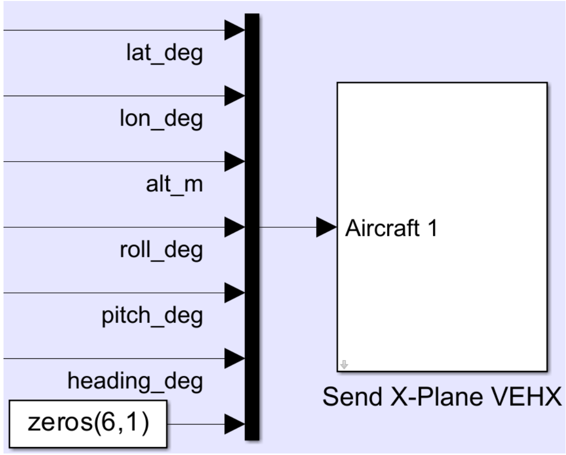
    - The new ".acf" file created in the previous step
    - The ".obj" and ".png" files created above using Blender
    - The "airfoils/NACA 2412-root.afl" and "airfoils/NACA 2412-tip.afl" airfoil files.

17. Create "[.acf file name]_icon11.png" and "[.acf file name]_icon11_thumb.png" image files to appear with the aircraft in X-Plane.

B.8 Setup a Simulink® Simulation to Run with X-Plane Visualizations

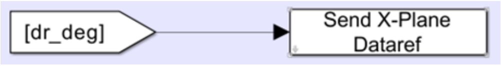
1. Download the X-Plane Blockset (XPB) [27] from https://www.mathworks.com/matlabcentral/fileexchange/76028-x-plane-blockset.

2. To send position and attitude data to X-plane, use the "Send X-Plabe VEHX" block. The first six inputs are: (1) latitude in deg, (2) longitude in deg, (3) altitude in meters, (4) roll angle in deg, (5) pitch angle in deg, and (6) heading angle in deg. The last 6 inputs are stick inputs that are not needed when the aircraft dynamics are modeled in Simulink®.

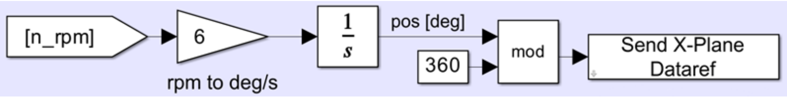


3. To send control effector position information to visualize deflection, use the "Send X-Plane Dataref" block. The "Dataref path" specified in the block mask is then set to the Dataref assigned to each control effector in Blender.

    - For a control surface, the input is simply the deflection angle from the Simulink® model.



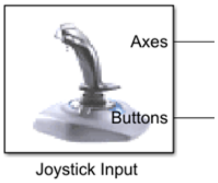
    - For a propeller, the rotational speed needs to be converted to propeller position in accordance with the rotation defined in Blender.



4. In Simulink®, under "Modeling"→"Model Settings"→"Solver," set the solver to be "Fixed-step" and set the "Fixed-step size" to the desired time step.

5. Adjust the Simulink® simulation pace to match real time (for example, using the "Simulation Pacing" capability in Simulink®).

6. A USB joystick or transmitter can be used to execute piloted simulations using the "Joystick Input" block available in the Simulink® 3D Animation toolbox [25].



B.9 Display X-Plane Visualization with Aircraft Dynamics Modeled in Simulink®

1. Open X-Plane.

2. Click "New Flight."

3. Select the custom aircraft created above and click "Start Flight."

4. Navigate to "View"→"Change (External)" to change the view. Here, "Circle" (Shift+4) is selected. The view can be rotated using the arrow keys. The "+" and "−" keys can be used to zoom in and zoom out.

5. In Simulink®, click "Run."

6. Adjust X-Plane graphics settings and/or simulation time step to achieve real time performance.