Design, Integration, and Experimental Validation of a Lifting-Body Quadrotor

Jason S. K. Zhou

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Aeronautics & Astronautics

University of Washington

2026

Supervisory Committee:

Behçet Açıkmeşe, Chair

Mehran Mesbahi

Benjamin Chung

Program Authorized to Offer Degree:

William E. Boeing Department of Aeronautics & Astronautics

University of Washington

Abstract

Design, Integration, and Experimental Validation
of a Lifting-Body Quadrotor

Jason S. K. Zhou

Chair of the Supervisory Committee:
Professor Behçet Açıkmeşe
William E. Boeing Department of Aeronautics & Astronautics

This thesis describes the design, integration, and experimental validation of a lifting-body tailsitting quadrotor intended to retain vertical takeoff and landing capability while generating meaningful aerodynamic lift in forward flight. The work is divided into four areas. The first covers a PX4 Autopilot firmware port onto a custom flight controller with software-defined hardware, validated through manual and autonomous flight on a conventional quadrotor platform. The second presents the aerodynamic design of the tailsitter airframe through parametric modeling, vortex-lattice simulation, and wind tunnel testing. The third addresses mechanical design, covering structural reinforcement, manufacturability, and electronics packaging for an airframe manufactured entirely by FDM. The fourth presents hover flight results and a longitudinal transition simulation. The results establish a baseline for future work on multirotor vehicles that combine hover capability with efficient high-speed flight.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

Variables and Coefficients

α: Angle of attack, degrees

β: Sideslip angle, degrees

δ: Control surface deflection angle, degrees

ρ: Air density

$\bar{c}$: Mean aerodynamic chord

C_D: Drag coefficient

C_{D_0}: Zero-lift drag coefficient

C_L: Lift coefficient

C_{L_0}: Lift coefficient at zero angle of attack

C_{L_α}: Lift-curve slope, per degree

C_{l_β}: Rolling moment coefficient derivative with respect to sideslip angle, per degree

C_m: Pitching moment coefficient

C_{m_α}: Pitching moment coefficient slope with respect to angle of attack, per degree

C_{n_β}: Yawing moment coefficient derivative with respect to sideslip angle, per degree

C_{Y_β}: Side force coefficient derivative with respect to sideslip angle, per degree

D: Aerodynamic drag force

e: Oswald efficiency factor

L: Aerodynamic lift force

S: Reference wing area

T: Thrust

V: Freestream velocity

W: Vehicle weight

x_{AC}: Aerodynamic center location, mm from nose

x_{cg}: Center of gravity location, mm from nose

x_{np}: Neutral point location, mm from nose

x_{ref}: Moment reference center location, mm from nose

Acronyms and Abbreviations

AC: Aerodynamic Center

AOA: Angle of Attack

CG: Center of Gravity

COTS: Commercial Off-The-Shelf

DDS: Data Distribution Service

DOF: Degree of Freedom

ESC: Electronic Speed Controller

FAA: Federal Aviation Administration

FDM: Fused Deposition Modeling

FPGA: Field-Programmable Gate Array

GPIO: General-Purpose Input/Output

GPS: Global Positioning System

HSE: High-Speed External

HSI: High-Speed Internal

I2C: Inter-Integrated Circuit

L/D: Lift-to-Drag Ratio

LFS: Low Force Stereolithography

LIPO: Lithium Polymer

MAC: Media Access Control (networking); Mean Aerodynamic Chord (aerodynamics)

MAVLINK: Micro Air Vehicle Link

MCU: Microcontroller Unit

NACA: National Advisory Committee for Aeronautics

NSH: NuttX Shell

PCB: Printed Circuit Board

PDB: Power Distribution Board

PHY: Physical Layer Transceiver

PID: Proportional-Integral-Derivative

PLA: Polylactic Acid

RMII: Reduced Media-Independent Interface

RTOS: Real-Time Operating System

RTV: Room Temperature Vulcanizing

SM: Static Margin

SPI: Serial Peripheral Interface

UART: Universal Asynchronous Receiver-Transmitter

UORB: Micro Object Request Broker

USB: Universal Serial Bus

UXRCE-DDS: Micro XRCE Data Distribution Service

VLM: Vortex-Lattice Method

VTOL: Vertical Takeoff and Landing

ACKNOWLEDGMENTS

A memorable journey is not possible without extraordinary people, and my experience during this degree is a testament to that. While I cannot thoroughly thank all who've made this adventure possible just with words on a page, I will do with this space what I can.

First, to my advisor, Professor Behçet Açıkmeşe, thank you for being the best advisor I could have imagined. Thank you for giving me the freedom to explore my projects (and side projects) while encouraging me to be more rigorous and formal in my approach. Thank you for providing me with opportunities and opening doors that I would have never had access to otherwise. All the exciting and memorable explorations over the course of this degree would not have been possible without your firm belief and support.

I would like to thank Professor Mehran Mesbahi for serving on my committee and for providing valuable feedback and perspective on this work. I would also like to thank Professor Eli Livne for his valuable teaching of configurational aerodynamics, which was crucial for the projects in this thesis.

To Skye Mceowen, thank you for making this entire adventure possible by believing in me and giving me the opportunity to pursue research here. Thank you for your invaluable mentorship, guidance, advocacy, friendship, and support since day one. I will always be inspired by the example you set, and be grateful for all you've done for me.

To Ben Chung, thank you for your tireless mentorship and guidance, for the irreplaceable role you played in the development of this work, and for serving on my committee. Without your dedication, these projects would not have been possible. Between the firmware bugs, FastRTPS mysteries, and mechanical design mishaps, the evenings and all-nighters spent working together will always be a special memory for me. I look forward to our future

collaborations, and I hope this work does justice to all your time and contributions.

To the incredibly brilliant and talented demo team: Chris Hayner, Aman Tiwary, Natalia Pavlasek, and Samuel Buckner; thank you for all the good times in the lab, in the field, and at the most enjoyable dinners. While I hope to never experience another sprint week quite like the ones we have had, there is no group of people I would rather have been sleep-deprived and delirious with. Professionally as colleagues and personally as friends, I am grateful to have met all of you, and I am excited to see the amazing things you will all accomplish.

To Govind Chari, thank you for your advice and perspective throughout the course of this degree. Your personal drive and tenacity have been uniquely inspiring and motivating to witness, and our happy hour conversations have contributed greatly to my creativity, clarity, and curiosity. I look forward to having many more.

To my friends and colleagues at ACL: Fabio Spada, Justin Ganiban, Chris Sota, Carlos Morales, Avi Mittal, Taewan Kim, Samet Uzun, Abhi Kamath, Purnanand Elango, Dayou Luo, Oliver Sheridan, and Jake Gonzales; it has been a privilege to work alongside you. Thank you for the enjoyable conversations and interactions in my time here, and I wish you all success in your future endeavors.

This work relied on many physical components and hours of experimentation, which was made possible by the efficiency and professionalism of the staff in the AA department at UW. To Jordan Allen and Noelle Hardman, thank you for handling my many purchase orders in the last two years. To Colin Baxter and Sean Krewson, thank you for your help in the machine shop and wind tunnel, which were essential in realizing key parts of this work. To Kenneth Wiersma, thank you for your initial advice and help with aerodynamic design and simulation, which greatly contributed to the starting point of the project.

I am thankful for all the wonderful housemates and people I've had the pleasure of meeting during this degree. To Mony Thach, Khoa Nguyen, and Tapan Khanal, thank you for helping me settle into Seattle, for an exciting and memorable year living together, and for

never failing to remind me how old I am. To Ava Jeanne Gutheil, thank you for your many encouragements and your delicious baked goods. To Archer Garey and the entire Husky MMA community, thank you for the opportunity to teach and train Muay Thai, and to have an outlet for the stresses of graduate school.

I am incredibly fortunate to have amazing friends whose care I can feel despite long distances. To Dawson Horvath, Victoria Schuster, Ivana Susac, and Yasmine Lau, thank you for cheering me on and for offering your earnest advice and support whenever I have needed it, during this degree as well as all the times before. I'm excited to catch up with all of you soon.

To my Sunshine, Hanna Palmer, thank you for your love, care and support during the tough days and in the deepest trenches of this journey. Thank you for teaching me all about meeps and nematodes and keystone species, and for listening to me go on about something something stars and gravity. Thank you for gifting me so many silly days, and I cannot wait for all of our adventures to come.

Most importantly, to my loving parents, thank you for all you have done and all the sacrifices you have made to give me this and every opportunity I have ever had. Your love, care, and support are the foundation of everything I have and ever will accomplish. Please take this work as a testament to all your efforts in raising and educating me, for this would not have been possible without the values you have worked so hard to instill in me. I hope this makes you proud.

DEDICATION

to my dear parents,

Chapter 1

INTRODUCTION

1.1 Motivation

Conventional quadrotors rely entirely on rotor thrust for lift and propulsion. In forward flight, the vehicle must pitch nose-down to produce a horizontal thrust component, reducing the vertical force available to support the vehicle weight. The central body and exposed arms present a large bluff frontal area, and the resulting parasitic drag scales with the square of airspeed. Together, these factors impose a low speed ceiling and poor cruise efficiency.

Several approaches exist to combine vertical takeoff with efficient forward flight. Tilt-rotor systems mechanically redirect rotor thrust for wing-borne cruise but add complexity and are typically optimized for forward flight rather than hover agility. Configurations with separate hover and cruise actuators carry mass that is inactive in each flight regime. Within the quadrotor design space, faired platforms such as the Red Bull Drone 1 and the Peregreen V2 have reduced drag through aerodynamic shells [35, 4], but these airframes generate no meaningful lift, and the rotors remain the sole source of vertical force.

Tailsitter configurations offer a different compromise. The vehicle takes off vertically and transitions to forward flight by rotating its body, allowing fixed aerodynamic surfaces to generate lift during cruise. All actuators are used in both flight regimes. The challenge for small, low-cost platforms is that the aerodynamic geometry and internal electronics layout are tightly coupled, and design changes in one domain propagate to the others. This thesis addresses that integration problem through the design, build, and staged validation of a custom hybrid quadrotor.

1.2 Project Overview

The work presented in this thesis resulted in two platforms. The first is Firelight, a conventional agile quadrotor used to validate the custom flight control stack under demanding flight conditions, including autonomous execution of externally generated reference trajectories. The second is Raven, a tailsitting quadrotor with a lifting-body fuselage, four wing-mounted pusher rotors, and actuated forward canards.

The flight control electronics use a custom board called Sparkie, developed in the Autonomous Controls Laboratory (ACL) at the University of Washington. Sparkie integrates an STM32H745 MCU, an iCE40HX FPGA for configurable signal routing, and a Raspberry Pi CM4 companion computer. A full PX4 Autopilot port was implemented, including RTOS configuration, peripheral integration, and system validation.

The Raven airframe was designed from the aerodynamic shell inward. The external geometry was defined first through parametric modeling in OpenVSP and evaluated using VLM, and the internal structure and electronics were subsequently packaged within that shell. The airframe was manufactured entirely using FDM with PLA filament on a desktop 3D printer. Two airframe configurations were analyzed: the wind tunnel configuration, which was fabricated and tested in the University of Washington 3×3 ft wind tunnel, and the flight configuration, which was modified for electronics integration and flown. Aerodynamic characteristics were evaluated using vortex-lattice simulation and wind tunnel testing, and a static stability and control authority analysis was performed.

The Raven vehicle was flight tested in the VTOL regime. The flight demonstrated that the custom avionics, FDM airframe, and PX4 integration function as a flyable system. Forward cruise flight was not attempted due to incomplete canard integration, insufficient transition controller tuning, and the absence of validated aerodynamic data across the angle-of-attack range required for transition. A longitudinal simulation with simplified physics was developed to evaluate the feasibility of the transition maneuver.

1.3 Thesis Organization

Chapter 2 details the PX4 Autopilot port onto the Sparkie flight controller, including FPGA firmware, companion computer infrastructure, and flight validation on the Firelight platform. Chapter 3 presents the aerodynamic design of the Raven airframe and its evaluation through vortex-lattice simulation, wind tunnel testing, and stability analysis. Chapter 4 describes the mechanical design, covering structural reinforcement, manufacturability, and electronics integration. Chapter 5 presents the hover flight results and the longitudinal transition simulation. Chapter 6 discusses contributions, limitations and areas of future work.

Chapter 2

FLIGHT FIRMWARE

2.1 Introduction

This chapter describes the firmware architecture and PX4 porting process for the custom Sparkie flight controller board. Sparkie is the most recent custom flight controller developed at the Autonomous Control Laboratory at the University of Washington. The board is built around the STM32H745IIK6 MCU [38] and integrates an iCE40HX FPGA for hardware routing and peripheral abstraction [36], as well as a Raspberry Pi CM4 companion computer [21]. This architecture separates real-time flight control tasks on the MCU from higher-level tasks such as networking, messaging, and trajectory generation on the companion computer.

An MCU plus companion computer architecture is present in some COTS flight controllers, such as the CM4-based Holybro Pixhawk variant [16]. However, the use of the STM32H745 and the inclusion of the iCE40HX FPGA increase Sparkie's computational capability and flexibility but introduce substantial challenges in hardware bring-up and PX4 porting.

A new board definition was implemented within the PX4 file structure [42]. The RTOS configuration was validated, peripheral buses and devices were integrated, and the system was tested to ensure deterministic execution during flight. The firmware development extended beyond parameter configuration or driver modification and required full-stack integration, from FPGA firmware and MCU bootloader configuration to PX4 estimation and control modules.

2.2 System Architecture

The Sparkie architecture is partitioned into three firmware domains. The STM32H745IIK6 MCU executes the real-time flight stack as a custom PX4 Autopilot port [38, 42]. The iCE40HX FPGA implements configurable signal routing and peripheral interfaces [36]. The Raspberry Pi CM4 runs embedded Linux and manages networking, MAVLink routing, motion capture input, and external communication [21, 41, 29]. Figure 2.1 shows the three domains and the connections between them.

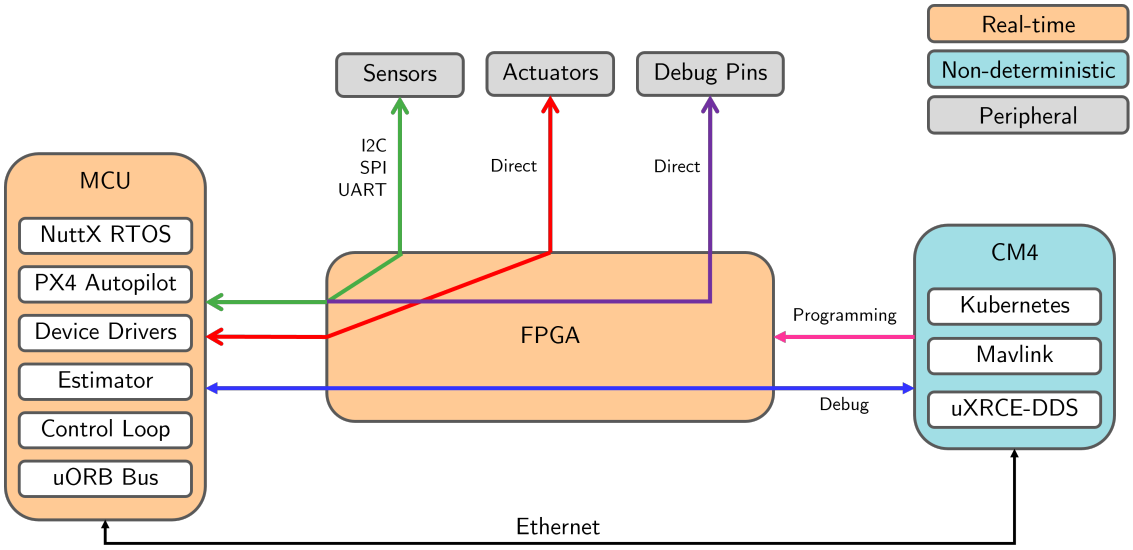

Figure 2.1: System architecture of the Sparkie flight controller

2.3 Microcontroller

The STM32H745 contains two computational cores: a Cortex-M7 and a Cortex-M4. Only the M7 core was used in this implementation. Enabling the M4 would have introduced inter-core communication, synchronization, and shared-memory management requirements that would have complicated bring-up for a custom board with no established procedures.

Initial bring-up used the MCU's HSI clock at 64 MHz. This bypassed hardware issues with

the external clock source during early debugging. The limitations of the internal oscillator, particularly its drift and reduced precision, became apparent as development progressed. The MCU was then reconfigured to use a 25 MHz HSE oscillator, which provided greater stability and enabled reliable bring-up of the Ethernet and USB buses. Peripheral device clocks were derived from the HSE through the MCU clock tree.

2.4 FPGA

The iCE40HX FPGA serves as a configurable hardware abstraction layer between the MCU and the other peripherals. Rather than hard-wiring all electrical paths on the PCB, the FPGA provides programmable routing of serial buses, interrupts, resets, debug signals, and auxiliary interfaces. Signal routing and interface behavior can be modified through Verilog source and pin constraint definitions without physically altering the PCB. During bring-up and development, this flexibility accelerated debugging and validation.

2.4.1 Serial Protocol Routing

Three serial protocols are used for onboard communication: UART, SPI, and I^2C , listed here in order of increasing implementation difficulty through the FPGA.

UART and SPI use actively driven, push-pull signaling with fixed direction per signal line [45]. For these protocols, the FPGA maps pins directly to the device input and output buffers on either side. The timing is edge-based and the electrical behavior is straightforward, so the FPGA routes the correct signal paths without additional protocol handling. These signals could also be routed to exposed debug pins and analyzed with a logic analyzer or oscilloscope.

Routing I^2C through the FPGA is more involved because both the data and clock lines require bidirectional open-drain behavior [27]. The FPGA pins must actively drive the line low and release it to high impedance so that an external pull-up resistor defines the high logic level. A wire-level I^2C repeater was implemented in Verilog to permit the clock signal to pass through while conditionally bridging the data line in one direction at a time. The

bidirectional and partially passive nature of the bus made it impractical to break I^2C signals out to debug pins as the additional path length and capacitance can degrade signal integrity. The FPGA also addressed cases where the required pull-up resistors were not present in hardware. The internal pull-ups on the FPGA pins provided sufficient pull-up resistance to ensure valid logic levels on the I^2C data and clock lines.

2.4.2 Debug Routing and Console Access

The FPGA pins can be routed to the GPIO pins of the CM4, enabling direct observation of the NSH output from the MCU using serial monitoring tools on the CM4 [8]. By routing the MCU console through the FPGA to the CM4, boot logs and debug output could be captured without requiring additional debug hardware. This allowed PX4 porting and debugging to occur using the Sparkie board itself, accelerating incremental development.

2.4.3 Programming the FPGA

The iCE40HX was configured at each power cycle from a binary image. The intended programming routine uses an SPI connection with the HSE oscillator as the clock source. However, much of the initial implementation occurred while the board operated from the HSI clock, and an alternative programming method was required.

Following the programming logic in the iCE40HX documentation [37], a bit-bang routine was compiled and stored on the CM4. This routine used the CM4 GPIO pins to emulate a clock source and SPI interface, then wrote the binary file directly onto the FPGA fabric.

2.5 Companion Computer

The Raspberry Pi CM4 [21] serves as a companion computer responsible for networking, telemetry routing, motion capture integration, and communication with external systems. The CM4 runs embedded Linux and connects to the MCU via Ethernet. This architecture keeps flight-critical real-time computation on the MCU and delegates inherently non-deterministic communication and networking tasks to the CM4.

The CM4 functions as the gateway between the flight controller and the external development environment. All external telemetry, ground control interaction, and trajectory commands pass through the CM4. Each CM4 operates as part of a Kubernetes cluster [2], which allows it to pull software from GitHub and to program and configure the other onboard components. Because the CM4 can connect to wireless networks and operate independently through a full operating system, it is the first major configuration step in board bring-up.

2.5.1 Ethernet Communication

The MCU contains an integrated Ethernet MAC but requires an external PHY for standard Ethernet signaling [38]. On Sparkie, a Microchip KSZ8081RNA provides the 10/100BASE-T physical layer and connects the MCU MAC to the CM4 network interface via the RMII protocol [18]. The PHY registers were configured through PX4 to establish link speed, duplex mode, and auto-negotiation.

Since Ethernet routing is implemented directly on the PCB rather than through the FPGA, the link could not be probed or routed to debug pins. Initial failures were challenging to diagnose because the error occurred at the physical configuration layer and was effectively silent; the MAC layer appeared functional even though no link had been established. Once configured correctly through PX4 and NuttX parameters, the Ethernet connection was reliable and consistent. The resulting link operated at 100 Mbps full-duplex, compared to 1 Mbps or less for the other serial protocols in the system. This bandwidth supported high-rate motion capture data and telemetry streaming without imposing excessive processing overhead on the MCU. The CM4's Linux networking stack managed routing and multiple concurrent connections independently of the MCU firmware.

2.5.2 uORB and uXRCE-DDS Communication Bridge

Within PX4, inter-module communication uses the uORB publish-subscribe messaging bus [43]. Sensor drivers, estimators, controllers, and other flight subsystems publish and subscribe to uORB topics for asynchronous message passing.

To allow the companion computer to interact with PX4, the firmware was configured to use the uXRCE-DDS client [44]. This component exposes selected uORB topics through a DDS-compatible interface over the Ethernet link. The uXRCE-DDS client starts on the MCU as part of the PX4 startup procedure. The CM4 runs the corresponding DDS agent, which bridges messages into the Linux networking environment where other devices on the network can access them.

Configuring this interface required defining which uORB topics would be exported and ensuring correct network parameters. During early integration, an issue arose when opening multiple NSH consoles. One console was maintained on a UART port through the CM4 for debug access, and a second became accessible through QGroundControl. Opening the second console caused the uXRCE-DDS client to lose connection because NuttX reinitialized the network stack whenever a new NSH console opened. A modification to NuttX's network management resolved this by allowing a configurable number of shells to open before network reinitialization.

2.5.3 Systems Integration and External Interfaces

The CM4 serves as the primary interface between the embedded flight controller and external systems.

A Kubernetes cluster integrates the CM4 into a distributed compute infrastructure [2]. When new firmware or onboard software is committed to the project repository, any wirelessly connected CM4 can retrieve the updated code and deploy it onboard. This resolved a recurring issue on older hardware, where the installed firmware version was often unknown and updates required manual transfer. Over-the-air updating improved configuration traceability and accelerated the development workflow.

The CM4 handles MAVLink telemetry and command communication [41]. MAVLink data from PX4 on the MCU is transmitted over the Ethernet link to the CM4, which distributes it to ground-control software on an external laptop. MAVLink routing was implemented through Linux tools and scripts rather than within the real-time firmware environment.

For indoor flight experiments, the OptiTrack motion capture system provided high-precision position measurements through the NatNet SDK [29]. The CM4's Linux networking stack acquired this data and passed it to the MCU over the Ethernet link.

Trajectory commands from external development systems also pass through the CM4. Research software transmits reference trajectories to the CM4, which forwards them to the MCU. This allows experimental trajectory-generation software to be developed independently and tested on the vehicle through a network connection, separate from the real-time control loop on the MCU.

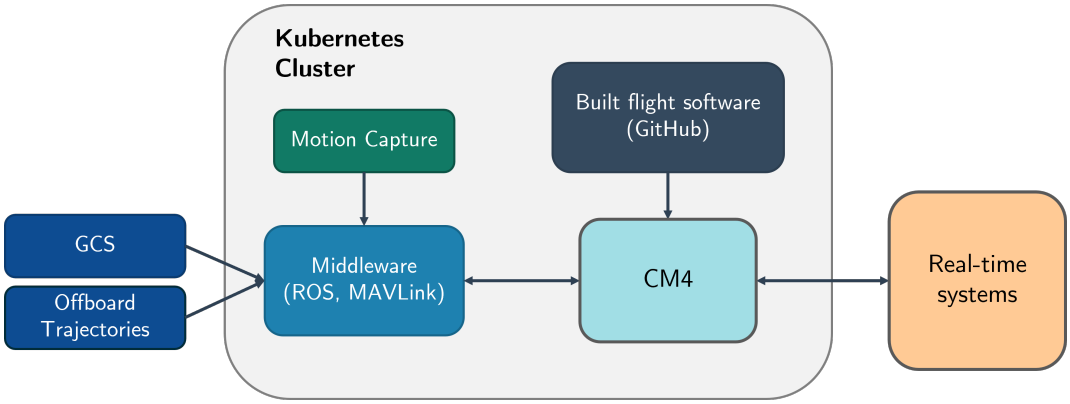

Figure 2.2: Infrastructure and communication architecture

2.6 Flight Validation

The first completed version of the firmware was validated on the Firelight platform shown in Figure 2.3. Firelight is a conventional quadrotor with a thrust-to-weight ratio of 14:1 at a flight weight of 500 g, making it highly agile and requiring the firmware to be efficient and correct for reliable flight.

The platform was test-flown manually both indoors using motion capture and outdoors using its onboard sensor suite and GPS. Once manual tuning produced stable flight, various trajectory-generation frameworks commanded the drone autonomously [13, 14]. An embedded controller using the Julia programming language [7] was also implemented and tested.

These tests verified the firmware's ability to accept trajectory and control inputs from a range of external software stacks and perform adequately during demanding trajectories.

Figure 2.3: Firelight quadrotor platform with the Sparkie flight controller

Chapter 3

AERODYNAMIC DESIGN AND ANALYSIS

3.1 Introduction

A conventional quadrotor places four rotors in an X planform on arms extending from a central fuselage, as illustrated by the Firelight platform in Figure 2.3. The central body houses flight electronics, battery, and payload. Motors are mounted at the arm tips at approximately equal radii from the center of mass, producing relatively symmetric inertial properties and simplifying the control allocation problem. This geometry is the dominant choice in both research and commercial platforms because it offers low design and manufacturing complexity, and because common multirotor flight-control stacks, including PX4, ArduPilot, and Betaflight, provide mature support for this arrangement.

However, this baseline configuration is optimized for hovering efficiency and mechanical simplicity, not for forward flight. Two characteristics limit its high-speed and long-range capability. The first is the thrust penalty inherent to forward flight: because the rotors are the sole source of both lift and thrust, the vehicle must pitch nose-down to produce a forward component, reducing the vertical thrust available to support its weight. The second is high parasitic drag: the central body and exposed arms present a large frontal area of bluff, unshaped surfaces. Together, these factors reduce forward-flight efficiency and limit practical cruise performance, even on racing-oriented platforms.

Prior high-speed quadrotor projects addressed the drag problem through aerodynamic fairing. The Red Bull Drone 1 enclosed a quadrotor within an aerodynamic canopy and exceeded 350 km/h while tracking a Formula 1 car [35]. The Peregreen 2 reached a top speed 480 km/h using a streamlined and ducted fuselage [4]. These vehicles demonstrated that shaped airframes can dramatically reduce drag and increase top speed.

More recently, several projects indicate that lift generation is considered along with drag reduction. The FastBoy FPV drone achieved 557.64 km/h by generating lift through an asymmetric planform with a streamlined fuselage [12]. The Peregreen V4 achieved 657.59 km/h using a shaped fuselage that generates lift while reducing drag [5].

The vehicle presented in this chapter draws from these designs and from non-quadrotor aircraft, combining cruise-oriented aerodynamic shaping with the VTOL and low-speed maneuverability of a quadrotor propulsion system. This chapter presents one iteration of a multi-cycle design process (discussed further in Chapter 6). An airframe was parametrically designed in OpenVSP, analyzed using vortex-lattice simulation, fabricated, and wind tunnel tested. It was subsequently revised for electronics integration and re-analyzed. Two configurations are documented: the wind tunnel configuration and the flight configuration.

3.2 Design Requirements and Constraints

The Raven vehicle has two distinct flight regimes: VTOL (including hover and low-speed maneuvering) and forward cruising flight. The vehicle must perform vertical takeoff and landing, stationary hover, and transition to cruise in which aerodynamic lift substantially offloads the rotors. The initial design is not optimized by any single metric; the goal was to meet preliminary aerodynamic requirements and form a baseline for subsequent refinement.

A cruising speed was selected to enable both wind tunnel verification and outdoor flight. The governing constraints were the University of Washington 3×3 wind tunnel speed limit (135 mph) [46] and the FAA Part 107 ground speed limit (100 mph) [11]. The cruise speed was set at Mach 0.118 (90.54 mph, 40.47 m/s), within both limits and high enough for conventional airfoil profiles to operate in a well-characterized regime.

The initial geometry prioritized minimizing frontal cross-sectional area to reduce parasitic drag, with electronics packaging treated as secondary. The fuselage cross-section was constrained by the battery dimensions (Tattu R-Line 5.0, 1550 mAh 6S, 37×52 mm). The 7-inch propeller diameter constrained the nacelle spacing and wingspan. The diameter of the T-Motor F100 brushless DC motor (33.4mm) constrained the minimum nacelle diameter.

3.3 Configuration Rationale

The starting point is a tailsitting quadrotor with four rotor-mounted lifting surfaces and a central fuselage. The decisions in this section are informed configuration choices for a first-pass design rather than uniquely optimal solutions. The Piaggio P.180 Avanti served as a reference because it combines a lift-generating fuselage, forward lifting surfaces, and aft-mounted pusher propulsion [32, 9].

3.3.1 Rotor Configuration

For a tailsitter, an X configuration produces an X-wing planform with all four surfaces canted at 45° in cruise, creating aerodynamic interference that is difficult to model at this design stage. A plus (+) configuration was selected instead: in cruise, two horizontal wings generate lift while two vertical wings serve as stabilizers. This also simplifies control mapping, with differential thrust between upper/lower rotors controlling pitch and left/right rotors controlling yaw.

The propellers are in a pusher configuration, preventing blockage behind the propeller disks. As Raven has no control surfaces on the wings, the benefits of blown wing surfaces are limited, and the absence of them simplifies aerodynamic analysis and simulation.

3.3.2 Component Design

Lifting-body fuselage. Piaggio reports that the Avanti's fuselage contributes approximately 20% of total lift [32]. Raven's fuselage takes inspiration from this and is symmetric only about the vertical plane, with greater curvature on the upper surface. In forward flight, this fuselage shape can create faster flow over the curved upper surface, producing a pressure difference and generating a net upward force.

Canards. Forward-mounted actuated canards provide additional lift, pitch, and yaw control authority ahead of the CG. They shape the pitching-moment characteristics and, when

designed to stall before the main wing, provide a nose-down tendency that helps prevent full-wing departure.

Boattail. The aft fuselage is tapered to reduce base drag. For axisymmetric conical boattails, minimum drag occurs at 14° [17]. This value was used as the design point.

Nacelles. Wing-tip nacelles house the motor mounts with a parabolic nosecone profile selected for low subsonic drag [25].

3.3.3 Airfoil Selection

All airfoils are from the NACA four-digit series, chosen for simple parameterization, predictable behavior, and available low-speed data.

Lift wings: NACA 4412. 4% camber at 40% chord, 12% thickness. The camber produces a zero-lift angle of approximately $-4°$, meaning the wing generates lift at zero vehicle AoA. At the operating Reynolds number, the NACA 4412 has a gradual stall onset with a maximum C_L of approximately 1.2 to 1.3 [1]. The 12% thickness provides volume for structure and printed walls.

Vertical stabilizers: NACA 0012. Symmetric profile producing zero side force at zero sideslip and zero pitching moment at zero AoA [1], both desirable for a passive stabilizer. Thickness matches the lift wings.

Canards: NACA 0015. Symmetric, 15% thick. Zero lift at zero AoA simplifies trim analysis. The NACA 0015 has a lower maximum C_L [1] than the NACA 4412 main wing, which favors canard-first stall. When the canard stalls before the wing, it produces a nose-down pitch break that reduces the α and protects the main wing from full stall [34]. Validated as a canard airfoil in ARL projectile studies [10].

3.4 Wind Tunnel Configuration

3.4.1 Parametric Design

The geometry was developed in OpenVSP [23]. The model consists of a central fuselage, two horizontal lift wings (NACA 4412), two vertical stabilizers (NACA 0012), two horizontal canards (NACA 0015), two vertical canards (NACA 0015), and four nacelles. The fuselage upper surface follows a cambered profile with the lower surface kept flat. The nose fuses smoothly from a point into the body, and the aft tapers into a 14° boattail. Propellers are not modeled. The moment reference center was placed at the CG assuming homogeneous solids. Table 3.1 lists the parameters and Figure 3.1 shows the model.

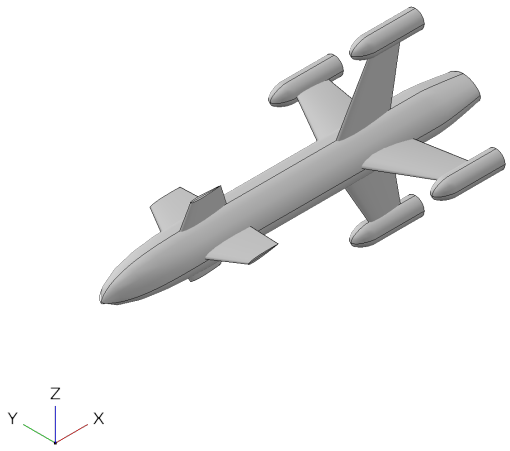

Figure 3.1: OpenVSP model of the wind tunnel configuration

3.4.2 VSPAERO Analysis

VSPAERO provides a VLM solver integrated into OpenVSP [22]. VLM models lifting surfaces as thin vortex sheets under inviscid, incompressible, irrotational flow assumptions and does not capture boundary layer effects, flow separation, or skin friction drag [26].

Two consequences are relevant to this geometry. First, VLM cannot model stall or

Table 3.1: Wind tunnel configuration geometric parameters

Component	Parameter	Value
Fuselage	Length	672 mm
	Max width	84 mm
	Max height	56 mm
	Boattail angle	14°
Rear wings (NACA 4412 and NACA 0012)	Root chord	112 mm
	Tip chord	56 mm
	Total wingspan	280 mm
	Reference area	23520 mm^2
	Aspect ratio	3.33
	Incidence angle	0°
	Location (in x)	420 mm
Canards (NACA 0015)	Span	91 mm
	Reference area	15288 mm^2
	Incidence angle	0°
	Location (in x)	130 mm
Reference quantities	$\bar{c}$	87.11 mm
	Moment ref. center (x, y, z)	(388.0, 0.0, -1.9) mm

separation drag, so results at high α are unreliable. At the operating Reynolds number (2.4×10^5), skin friction drag dominates over induced drag at moderate α, meaning VLM substantially underpredicts total drag. Second, VLM models the fuselage as a thin-surface approximation of a thick body, which does not capture its pressure-based lift contribution. The fuselage design relies on physical reasoning and reference aircraft, and it is validated through wind tunnel testing.

Tables 3.2 and 3.3 list the freestream conditions and solver settings. Settings were chosen per OpenVSP developer guidance [31] and a study on VSPAERO wing simulation [22]. Higher iteration counts caused numerical instability and large outliers during simulation.

Table 3.2: VSPAERO freestream conditions

Parameter	Value
Freestream velocity	40.47 m/s
Air density	1.225 kg/m^3
Mach / Reynolds number	0.118 / 2.4×10^5
S / $\bar{c}$	23520 mm^2 / 87.1 mm
Moment ref. center (x, y, z)	(388.0, 0.0, -1.9) mm
AoA range / increment	$-5°$ to $30°$ / $1°$
β	$0°$

Table 3.3: VSPAERO solver settings

Parameter	Value
Solver mode	VLM
Wake nodes	64
Wake iterations	7

Aerodynamic lift and drag are computed from the VLM-predicted coefficients as:

$$L = \tfrac{1}{2}\rho V^2 S\, C_L, \qquad D = \tfrac{1}{2}\rho V^2 S\, C_D \tag{3.1}$$

For a tailsitter in forward flight, the rotors are oriented along the body axis rather than vertically. At α, horizontal equilibrium requires $T = D/\cos\alpha$, and vertical equilibrium requires:

$$L + D\tan\alpha = W \tag{3.2}$$

This calculation uses vehicle mass 1.8 kg ($W = 17.66$ N), $\rho = 1.225$ kg/m^3, $V = 40.47$ m/s, and $S = 0.02352$ m^2. Vehicle mass is based on the flight configuration discussed in section 3.5. The thrust-to-weight ratio of 5:1 based on measured thrust data. The α that satisfies Equation 3.2 is referred to as the force trim angle of attack throughout this chapter. This calculation is detailed in Section 3.5.2.

Figure 3.2 presents the VLM predictions. The predicted C_{L_α} is 0.073, minimum C_D is 0.041, and maximum L/D is 7.9 at $\alpha = 6°$. The force trim α is 7.7°.

3.4.3 Wind Tunnel Validation

A full-scale PLA model was tested in the University of Washington 3×3 foot open-loop wind tunnel [46]. Forces were measured using an aft sting balance (Figure 3.3). Table 3.4 summarizes test conditions. Measured forces were gravity-compensated using static tare runs, rotated to wind axes, and the pitching moment was transferred from the sting to the model CG. Coefficients use the same reference quantities as the VSPAERO analysis. Blockage correction was not required as model frontal area ≈3.5% of test section [15].

Figure 3.4 overlays VLM predictions and tunnel measurements. Figure 3.5 provides a focused comparison of the C_{m_α} slope. Table 3.5 summarizes key parameters. All slope fits are computed over $\alpha = 0°$ to 10°.

Both sources confirm positive lift at zero α and peak L/D near $\alpha = 6°$. VLM underpredicts drag by approximately 43% (C_{D_0} of 0.041 vs. 0.072), as expected because VLM computes only induced drag and does not capture skin friction or pressure drag from the

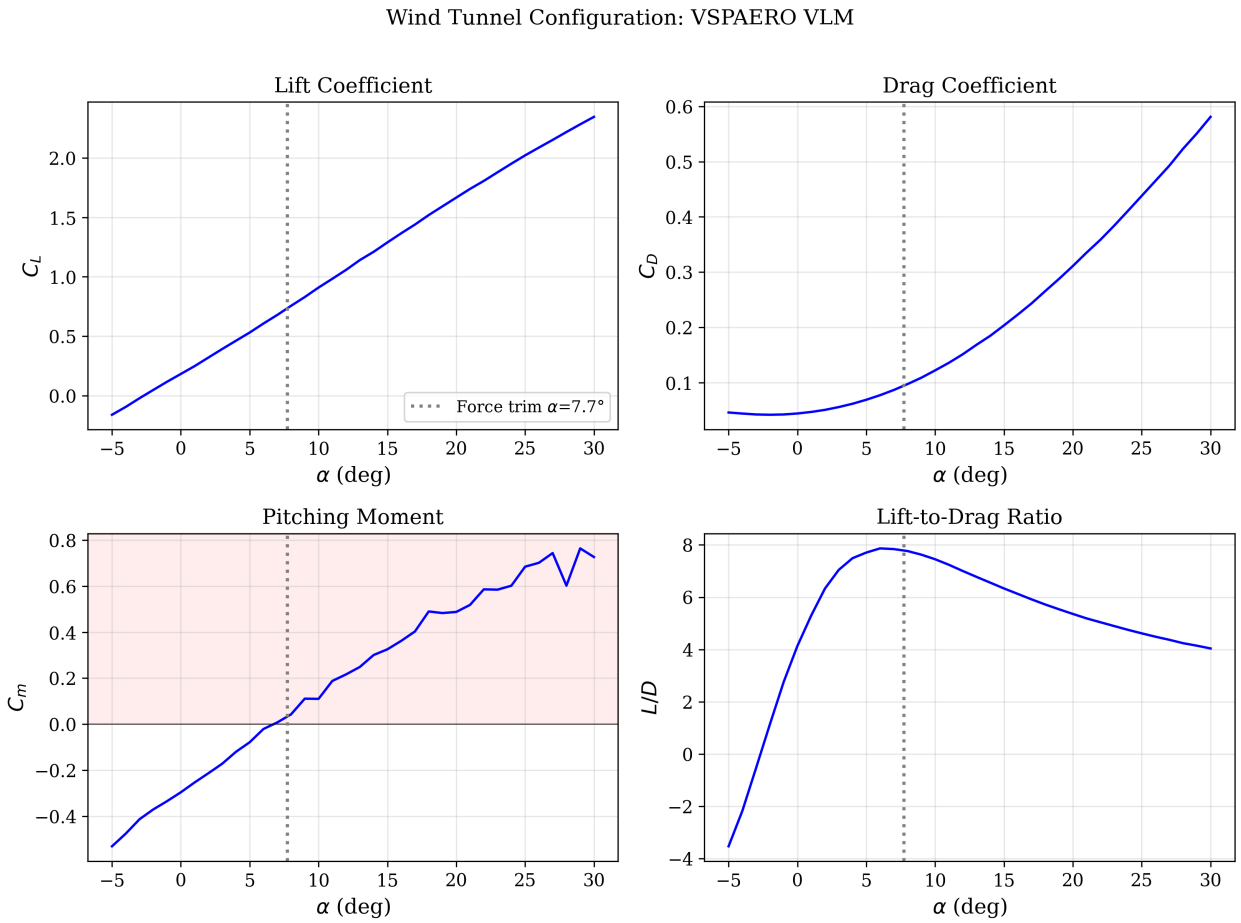

Figure 3.2: VSPAERO VLM predictions for the wind tunnel configuration. The force trim α is indicated.

Figure 3.3: Wind tunnel configuration model mounted on the sting balance

Table 3.4: Wind tunnel test conditions

Parameter	Value
Facility	UW 3×3 ft wind tunnel
Freestream velocity	40.6 m/s (90.8 mph)
Dynamic pressure	21.1 psf
Reynolds number	2.4×10^5
AoA range	$-5°$ to $30°$
β	$0°$
Balance / Model scale	Aft sting / 1:1

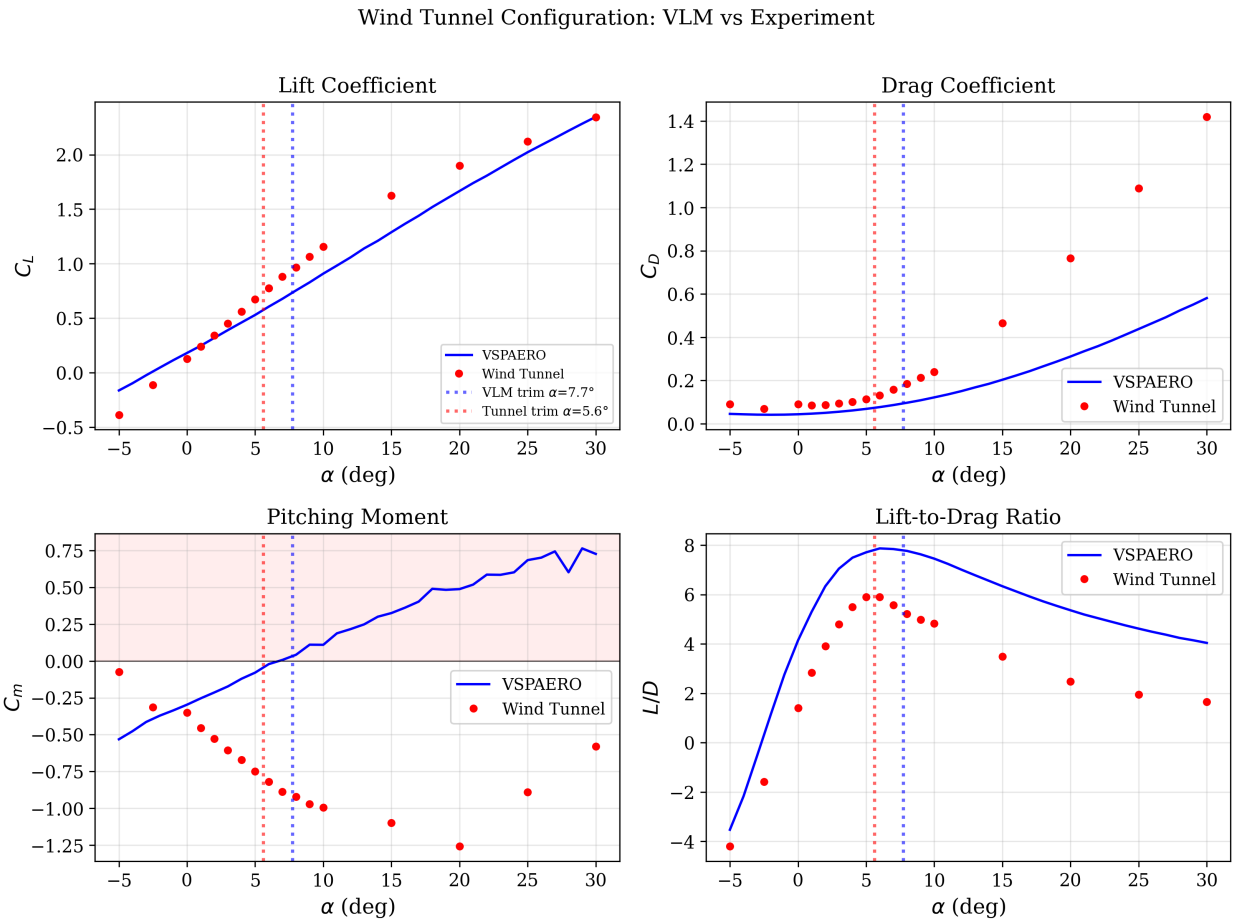

Figure 3.4: VSPAERO VLM predictions and wind tunnel measurements for the wind tunnel configuration. Force trim α from VLM (blue) and tunnel (red) indicated.

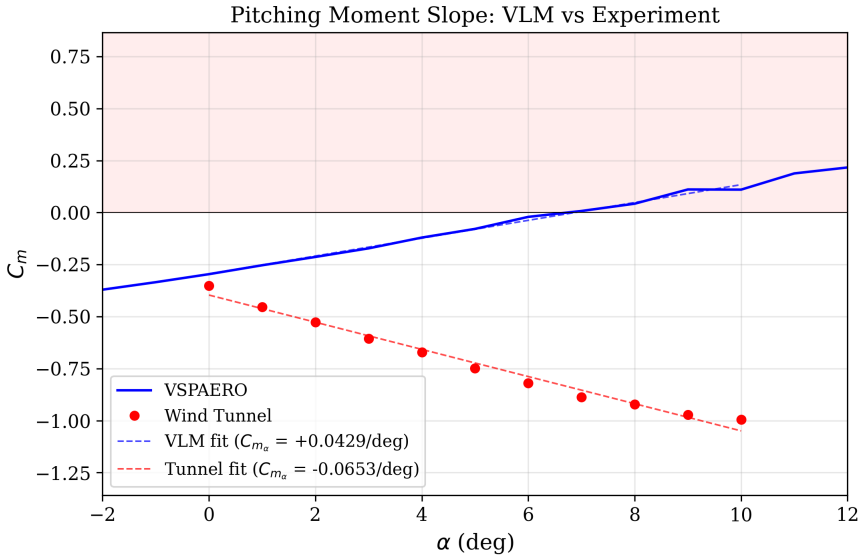

Figure 3.5: C_{m_α} comparison ($\alpha = 0$–$10°$). VLM predicts a positive slope (unstable); the tunnel measures a negative slope (stable). Shaded region indicates nose-up (destabilizing) moment.

Table 3.5: Comparison of VSPAERO predictions and wind tunnel measurements

Parameter	VSPAERO	Wind Tunnel	Difference
C_{L_α}	0.073	0.103	−30%
α at $C_L = 0$	−2.7°	−1.3°	–
C_{D_0} (polar fit)	0.041	0.072	−43%
e	0.97	0.80	–
$(L/D)_{\max}$	7.87	5.90	+33%
α at $(L/D)_{\max}$	6°	6°	0
C_{m_α}	+0.043	−0.065	sign reversal
C_m at $\alpha = 0°$	−0.296	−0.351	+16%
Force trim α	7.7°	5.6°	+2.1°

printed model surface. At this Reynolds number and moderate α, skin friction is the dominant drag component, making this underprediction expected. The 30% underprediction of C_{L_α} is attributed to VLM's inability to model the thick-body lift contribution of the fuselage, as the higher experimental slope is consistent with meaningful fuselage lift generation.

The most significant disagreement is C_{m_α} (Figure 3.5). VLM predicts $C_{m_\alpha} = +0.043$ (unstable) while the tunnel measures $C_{m_\alpha} = -0.065$ (stable). The likely cause is the fuselage lift acting at a location that produces a stabilizing moment increment with α, shifting the neutral point aft. The C_m values at $\alpha = 0°$ agree within 16%, indicating the discrepancy is in the slope rather than the moment level.

Figure 3.6 shows the AC computed from both sources. The AC is the point about which the pitching moment is independent of lift, computed as:

$$x_{AC} = x_{ref} - \frac{dC_m}{dC_L}\,\bar{c} \tag{3.3}$$

For the VLM data, VSPAERO computes this internally from its influence coefficient matrix. For the wind tunnel data, dC_m/dC_L was evaluated by central finite differences between adjacent α data points. VLM places the AC ahead of the CG (unstable) while the tunnel data places it behind the CG (stable), consistent with the sign reversal in C_{m_α}.

The force trim α differs by 2.1° between VLM (7.7°) and tunnel (5.6°), a direct consequence of the higher experimental C_{L_α}. VLM is therefore not a reliable predictor of absolute stability or absolute trim conditions for this geometry, but it can be used for comparative analysis and canard control authority assessment.

3.5 Flight Configuration

Electronics integration constraints (Chapter 4) required three modifications: the vertical canard fins were removed as there was insufficient space for canard actuators and passive fins would only add drag, the fuselage was lengthened from 672 mm to 722 mm, and the nose-to-body transition was made more abrupt for increased internal volume. Table 3.6 summarizes the changes and Figure 3.7 shows the revised model. The moment reference

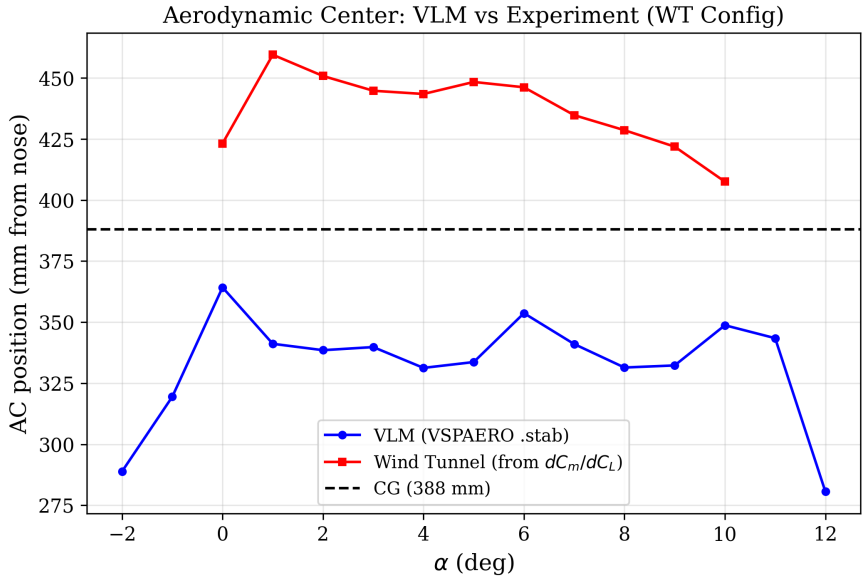

Figure 3.6: AC from VLM (.stab file) and from wind tunnel data (Equation 3.3), with CG indicated. VLM predicts AC ahead of CG; tunnel data shows AC behind CG.

center was measured on the assembled vehicle at $x = 430$ mm.

The flight configuration was not wind tunnel tested. VLM results are therefore used for comparative analysis and control authority assessment, not for absolute stability determination.

3.5.1 VSPAERO Analysis

The flight configuration was analyzed with the same VSPAERO setup, geometry updated per Table 3.6, and moment reference center at 430.0 mm. Figure 3.8 compares the two configurations, and Table 3.7 presents parameter changes.

Lift and drag characteristics are nearly unchanged. The small drag reduction is consistent with removing the vertical canard fins. The increase in C_{m_α} from +0.043 to +0.061 and the less negative C_m at $\alpha = 0°$ both reflect the aft CG shift.

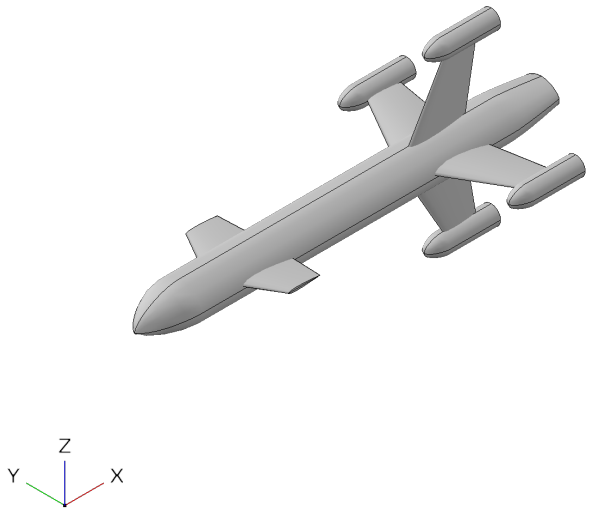

Figure 3.7: OpenVSP model of the flight configuration

Table 3.6: Geometric differences between configurations

Parameter	Wind Tunnel	Flight
Fuselage length	672 mm	722 mm
Vertical canard fins	Present	Removed
Nose transition (in x)	168 mm	80 mm
Moment ref. center (in x)	388 mm	430 mm
Rear wings (in x)	420 mm	470 mm
All other dimensions	Unchanged	

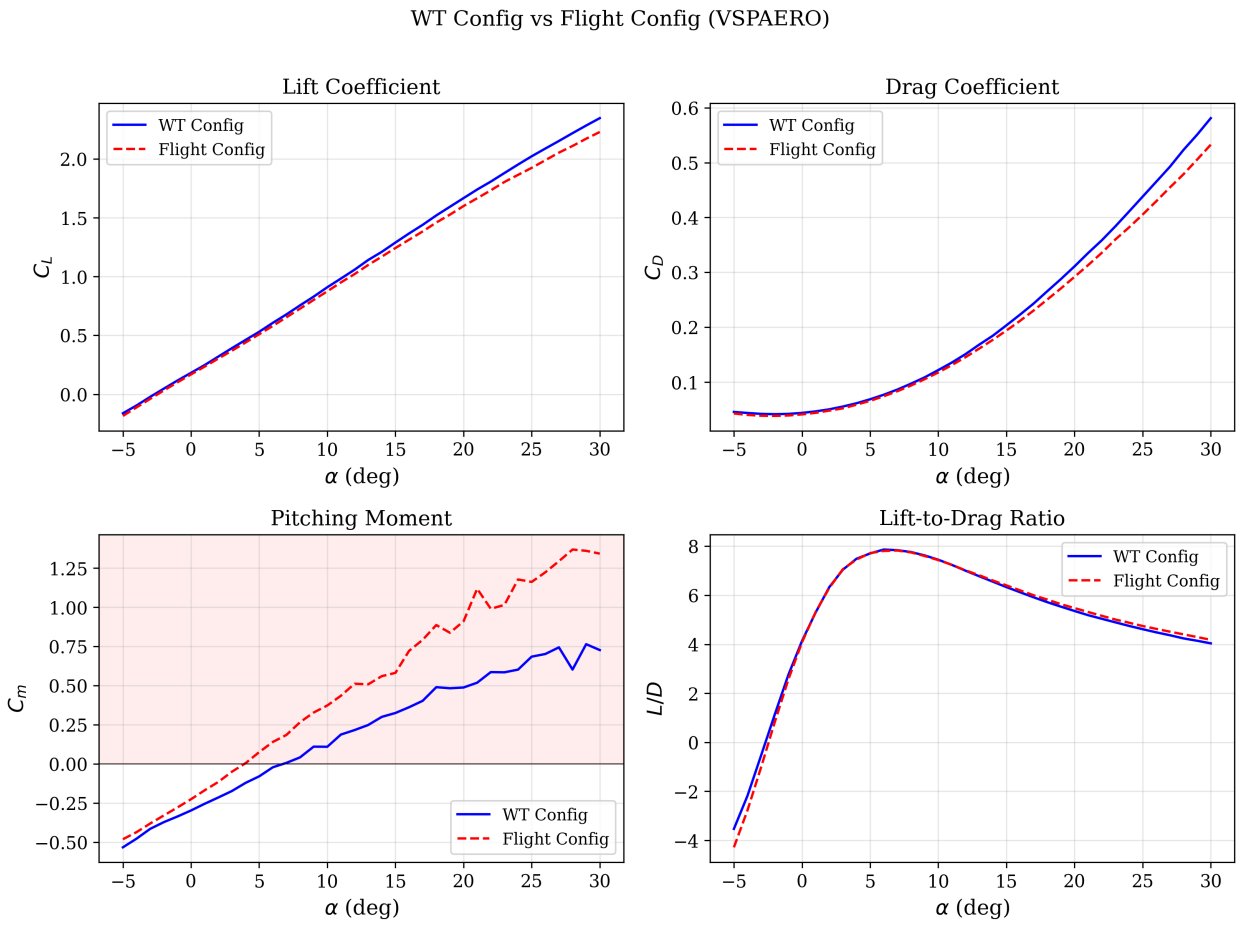

Figure 3.8: VSPAERO comparison of wind tunnel and flight configurations

Table 3.7: VSPAERO parameters: wind tunnel vs flight configuration

Parameter	Wind Tunnel	Flight	Δ
C_{L_α}	0.073	0.071	-0.002
α at $C_L = 0$	$-2.7°$	$-2.5°$	$+0.2°$
C_{D_0} (polar fit)	0.041	0.039	-0.002
e	0.97	0.92	-0.05
$(L/D)_{\max}$	7.87	7.83	-0.04
C_{m_α}	$+0.043$	$+0.061$	$+0.018$
C_m at $\alpha = 0°$	-0.296	-0.224	$+0.072$

3.5.2 Vertical Force Balance

The force trim condition (Equation 3.2) is evaluated using the VLM-predicted C_L and C_D curves for the flight configuration. The VLM-predicted force trim α is 8.1°. At this condition, the aerodynamic lift is 17.34 N, the vertical thrust component contributes 0.32 N, and the required thrust of 2.26 N is 2.6% of the maximum available thrust. The L/D at force trim is 7.7.

Figure 3.9 shows the vertical force balance as a function of α. The aerodynamic lift dominates the vertical force budget; the rotor thrust contribution is small at the moderate α near force trim.

Even at α well below force trim, the aerodynamic surfaces carry a large fraction of W with L/D between 6.3 and 7.8 for $\alpha = 2°$ to 6°. The thrust required at all operating points remains below 6% of maximum available thrust.

3.6 Stability and Control Analysis

This section evaluates static stability and control authority of the flight configuration. Dynamic stability analysis is omitted: the vehicle operates under closed-loop control which

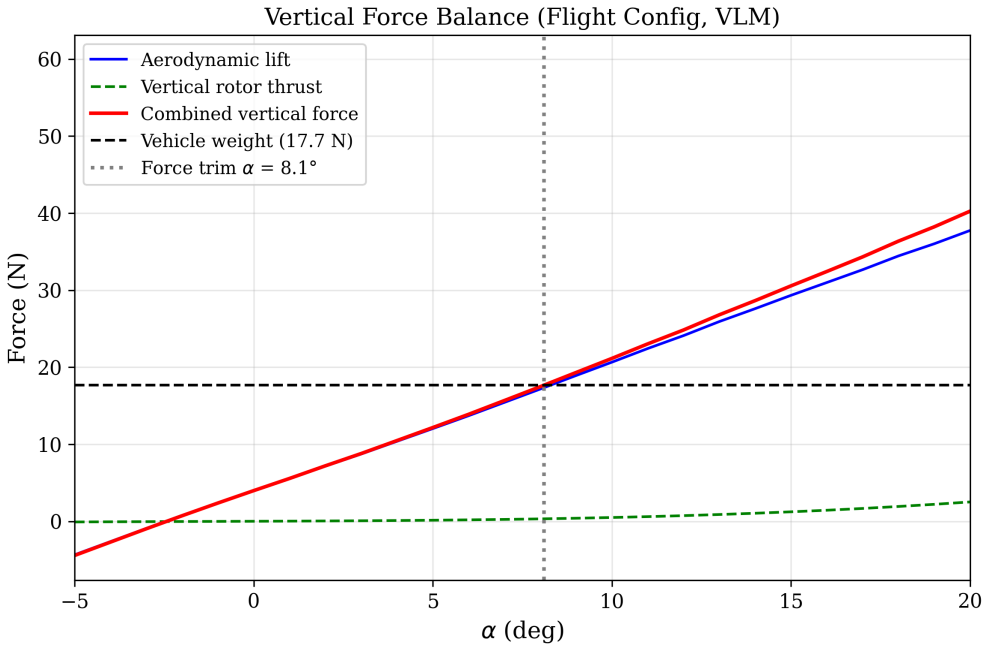

Figure 3.9: Vertical force balance for the flight configuration: aerodynamic lift, vertical thrust component, and their sum against W. Force trim α indicated.

Table 3.8: Force balance at representative α (flight configuration, VLM)

α (deg)	L (N)	Vertical Rotor Force (N)	T_{req} (N)	$T/T_{\max}$	L/D
0	3.98	0.00	0.97	1.1%	4.1
2	7.16	0.04	1.14	1.3%	6.3
4	10.38	0.10	1.39	1.6%	7.5
6	13.70	0.18	1.76	2.0%	7.8
8.1	17.34	0.32	2.26	2.6%	7.7
10	20.65	0.49	2.82	3.2%	7.4
15	29.33	1.23	4.74	5.4%	6.4

alters the plant dynamics, and the rotational damping derivatives required are not reliably produced by VLM for this geometry. Closed-loop approaches are discussed in Chapter 6.

3.6.1 Longitudinal Stability

Static pitch stability requires $C_{m_\alpha} < 0$. The SM and neutral point can be respectively found with:

$$SM = \frac{x_{np} - x_{cg}}{\bar{c}} \tag{3.4}$$

$$x_{np} = x_{ref} - \frac{C_{m_\alpha}}{C_{L_\alpha}}\bar{c} \tag{3.5}$$

Table 3.9 summarizes the VLM-predicted parameters. VLM predicts static pitch instability, with the CG 75.2 mm aft of the neutral point. As established in Section 3.4.3, VLM is biased toward predicting instability for this geometry; the wind tunnel configuration was also predicted unstable by VLM but measured stable in the tunnel. The flight configuration may be less unstable than predicted, but this cannot be confirmed without further testing.

Table 3.9: Longitudinal stability parameters (VLM, flight configuration)

Parameter	Value
C_{L_α}	0.071
C_{m_α}	+0.061
Neutral point	354.8 mm from nose
CG location	430.0 mm from nose
SM	$-86.4\%\ \bar{c}$
$C_m = 0$ crossing	$\alpha = 3.9°$

Figure 3.10 shows the pitching moment curve. The C_m shape has operational relevance for a tailsitter: negative C_m below $\alpha \approx 3.9°$ opposes pitch-up perturbations from cruise, while positive C_m above this crossing assists cruise-to-VTOL reversion (pitching up against

gravity). While this was not a deliberate design target, it can be operationally compatible with a tailsitter cruise-to-VTOL transition strategy.

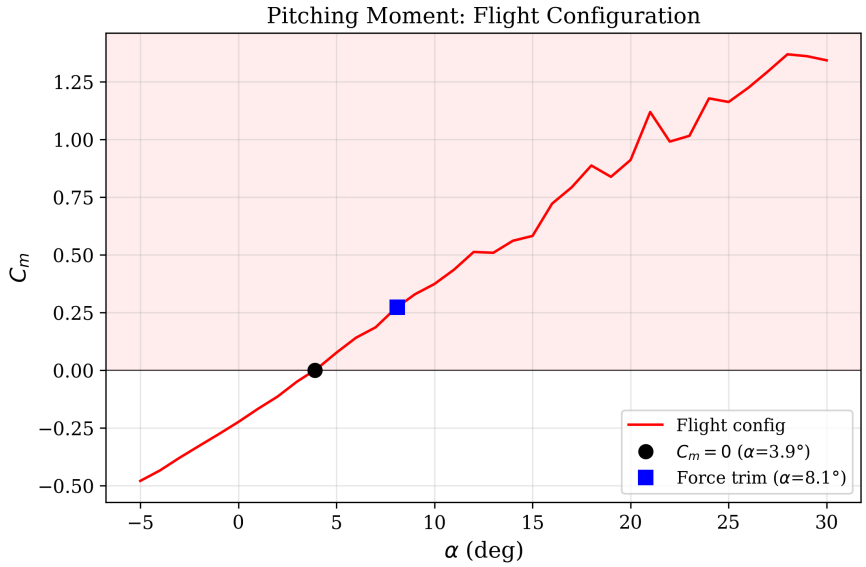

Figure 3.10: C_m vs α for the flight configuration, with $C_m = 0$ crossing and force trim α marked

Figure 3.11 shows the VSPAERO-computed AC (Equation 3.3) for both configurations over the linear range ($\alpha = -2°$ to $12°$). The AC is computed internally by VSPAERO as the point about which C_m is locally insensitive to α, so it is shown only where this computation is well-conditioned.

3.6.2 Airframe Variation Effects on Stability

Two airframe variations were simulated in VLM to isolate their effects on longitudinal stability. As these results carry the same thick-body fuselage limitations discussed in Section 3.4.3, they are useful for studying relative effects rather than absolute values.

The first variation moved the CG to 300.0 mm, retaining canards. VLM predicts $C_{m_\alpha} = -0.049$ and SM $= +68.7\%\ \bar{c}$. The second variation removed canards entirely at the actual

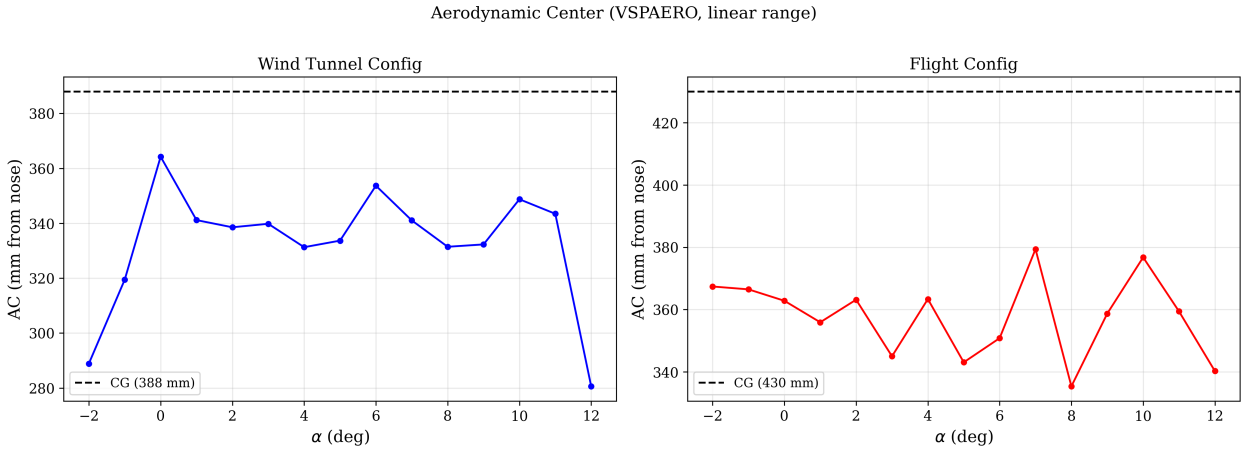

Figure 3.11: AC for both configurations (VLM, $\alpha = -2°$ to $12°$), with CG locations indicated

CG. VLM predicts $C_{m_\alpha} = -0.012$ and SM $= +23.4\%\ \bar{c}$. Figure 3.12 and Table 3.10 present the results.

Table 3.10: Airframe variation stability results (VLM)

Configuration	C_{m_α}	SM (% $\bar{c}$)	Stable?
Flight (baseline)	$+0.061$	-86.4	No
Forward CG (300 mm)	-0.049	$+68.7$	Yes
Canardless	-0.012	$+23.4$	Yes

The canards provide beneficial control authority for the tailsitter configuration, but their forward placement shifts the neutral point forward, driving the VLM-predicted instability. The forward CG study confirms stability is recoverable with appropriate mass placement. The canardless study confirms the wing and fuselage alone produce a stable configuration.

3.6.3 Lateral and Directional Stability

A sideslip sweep at $\alpha = 8.1°$ (force trim) with $\beta = -2°$ to $+2°$ yields the derivatives in Table 3.11. Figure 3.13 shows the variation. VLM predicts directional instability ($C_{n_\beta} < 0$)

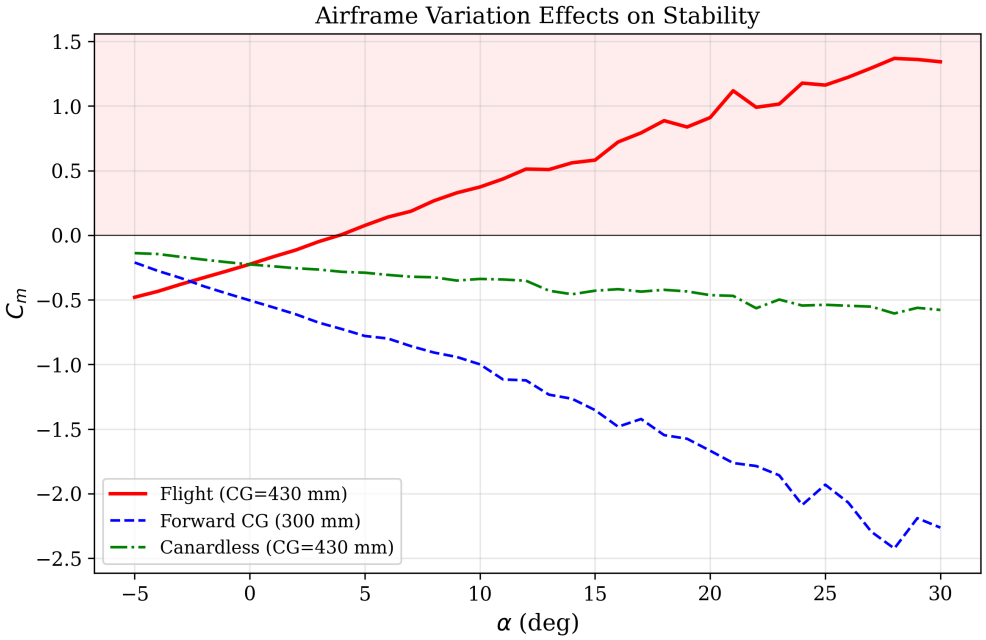

Figure 3.12: C_m vs α for baseline, forward CG, and canardless configurations (VLM)

and no restoring dihedral effect ($C_{l_\beta} > 0$). With the vertical canard fins removed, the NACA 0012 vertical stabilizers alone are insufficient. Directional stability therefore depends on active yaw control based on the simulation. However, as VLM fails to accurately model forces on the fuselage, this result may not be representative of the actual vehicle's lateral and directional stability.

Table 3.11: Lateral/directional derivatives (VLM, $\alpha = 8.1°$)

Parameter	Value	Requirement
C_{Y_β}	-0.054	–
C_{n_β}	-0.014	> 0 (not met)
C_{l_β}	$+0.005$	< 0 (not met)

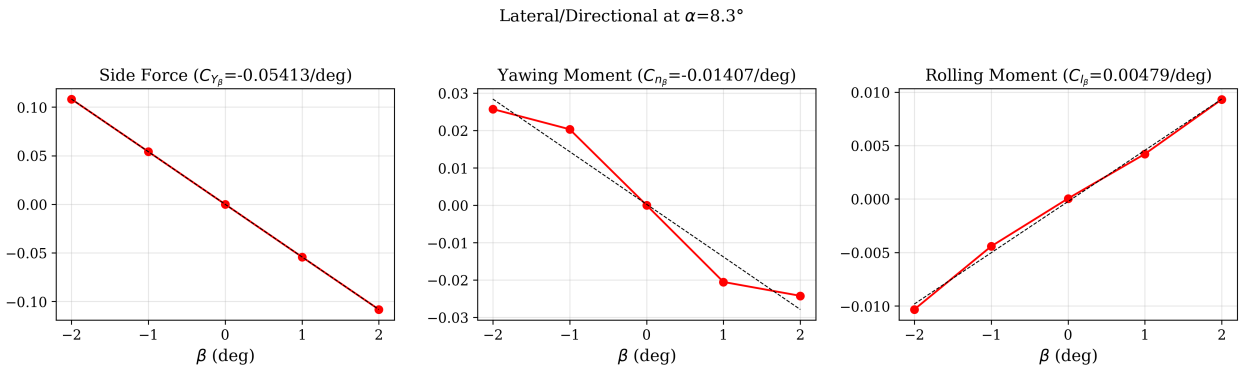

Figure 3.13: Lateral/directional coefficient variation with β at $\alpha = 8.1°$ (VLM)

3.6.4 Canard Control Authority

VSPAERO runs at canard deflections of $\pm 5°$ and $\pm 10°$ assess pitch control authority. Figure 3.14 shows the effect on C_m and C_L across α. Figure 3.15 shows how C_L, C_D, and C_m vary with δ at several representative α values. Table 3.12 gives values at the force trim $\alpha = 8.1°$.

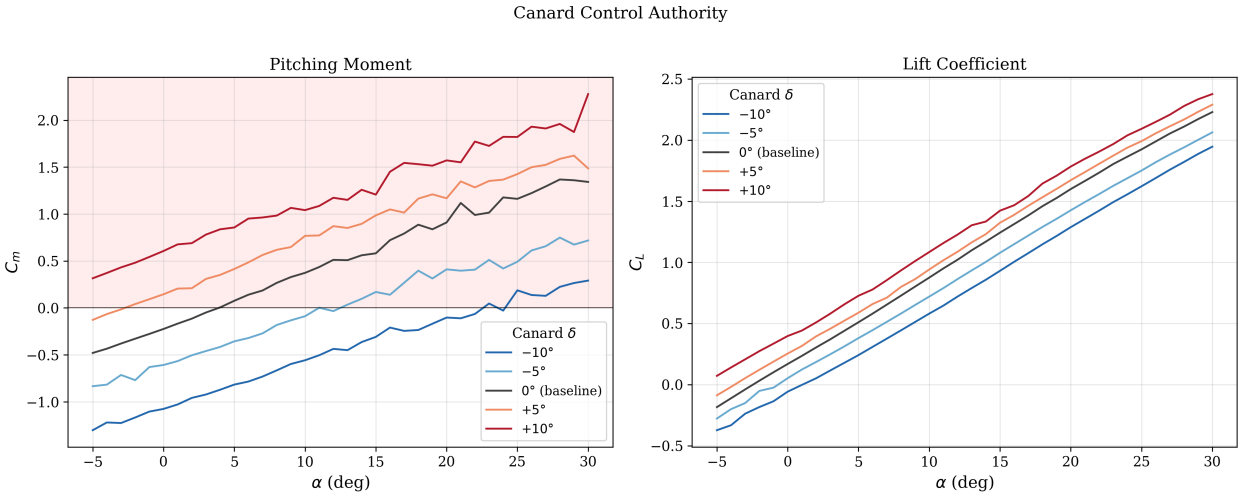

Figure 3.14: C_m (left) and C_L (right) vs α at five canard deflections (VLM)

At force trim, $\delta = -5°$ shifts C_m from $+0.273$ to -0.177, providing a nose-down restoring

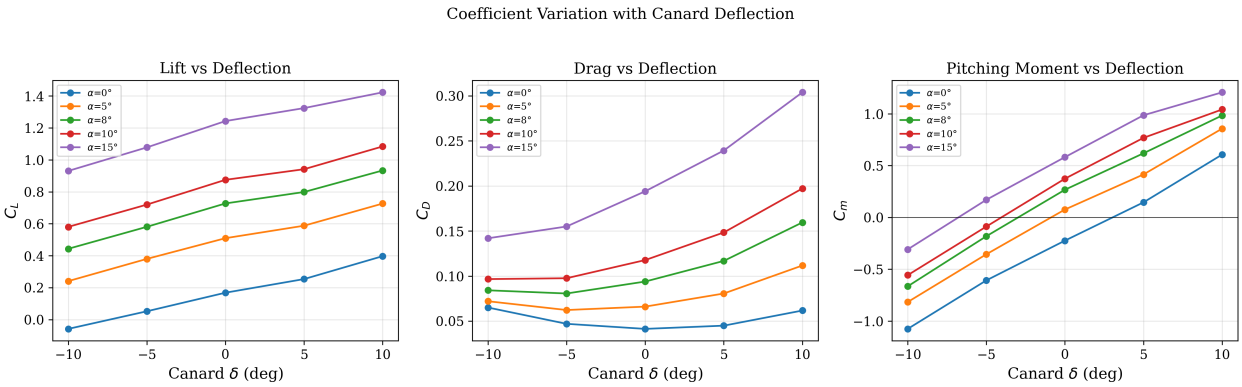

Figure 3.15: C_L, C_D, and C_m vs δ at representative α (VLM)

Table 3.12: C_m and C_L at force trim $\alpha = 8.1°$ vs δ (VLM)

Canard δ	C_m	C_L
$-10°$	-0.659	0.450
$-5°$	-0.177	0.588
$0°$	$+0.273$	0.735
$+5°$	$+0.622$	0.805
$+10°$	$+0.992$	0.941

moment. The control effectiveness $\partial C_m/\partial\delta \approx 0.084$ is approximately linear across the $\pm 10^\circ$ range. The trimmable α range spans -2.7° ($\delta = +5^\circ$) to 22.6° ($\delta = -10^\circ$), covering the entire operating envelope. Figure 3.16 summarizes the stability picture.

Table 3.13: Moment trim α ($C_m = 0$) vs δ

Canard δ	Trim α
-10°	22.6°
-5°	11.0°
0°	3.9°
$+5^\circ$	-2.7°

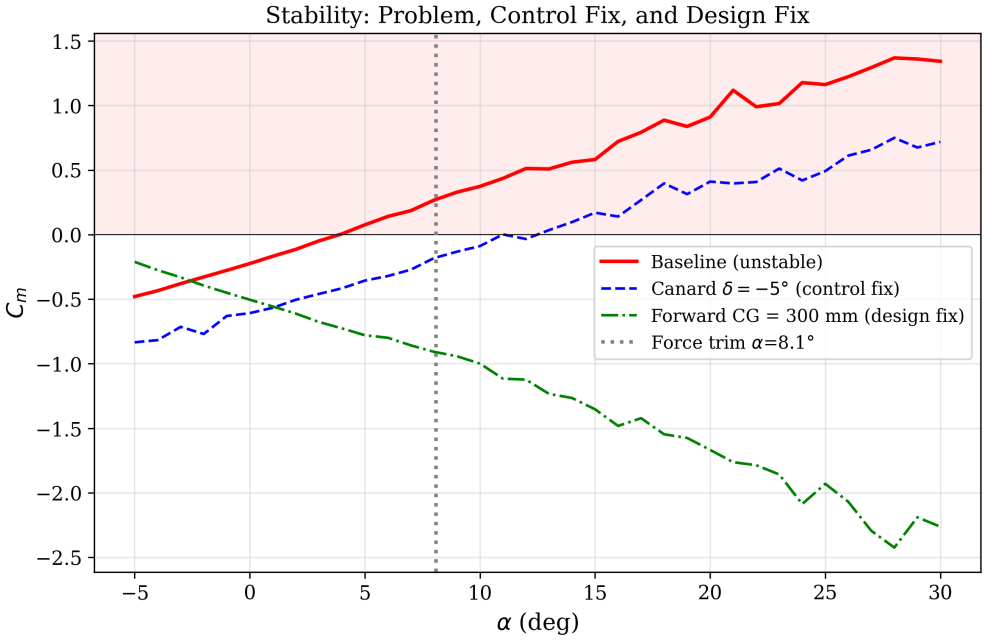

Figure 3.16: Baseline flight configuration (unstable), $\delta = -5°$ canard (control fix), and forward CG at 300 mm (design fix). Force trim α indicated.

3.7 Summary of Results

VLM comparison against wind tunnel data showed qualitative agreement in lift behavior but quantitative discrepancies: 30% underprediction of C_{L_α}, 43% underprediction of drag, and a sign reversal in C_{m_α}. These are attributed to VLM's thin-surface assumption, which cannot model fuselage thick-body lift or skin friction drag.

The flight configuration modifications had minimal effect on VLM-predicted lift and drag. Airframe variation studies showed the VLM-predicted pitch instability is driven by canard placement forward of the CG; both forward CG placement and canard removal produce stable predictions. Canard authority analysis showed $\delta = -5°$ produces negative C_m at force trim, with a trimmable range of $-2.7°$ to $22.6°$.

At cruise speed, force trim occurs at $\alpha = 8.1°$ (VLM) with L/D = 7.7, requiring only 2.6% of available thrust. The flight configuration is the geometry equipped with electronics

and flight tested in Chapters 4 and 5.

Table 3.14: Summary of aerodynamic and stability parameters

Parameter	WT Config	Flight Config
C_{L_α} (VLM)	0.073	0.071
C_{L_α} (experiment)	0.103	–
C_{D_0} (VLM polar fit)	0.041	0.039
e (VLM)	0.97	0.92
$(L/D)_{\max}$ (VLM)	7.87	7.83
C_{m_α} (VLM)	+0.043	+0.061
C_{m_α} (experiment)	−0.065	–
SM (VLM)	–	−86.4% $\bar{c}$
C_{n_β} (VLM)	–	−0.014
C_{l_β} (VLM)	–	+0.005
Force trim α (VLM)	7.7°	8.1°
Force trim α (experiment)	5.6°	–

Chapter 4
MECHANICAL DESIGN

4.1 Introduction

The mechanical design of the Raven airframe introduced challenges not typically encountered in conventional quadrotor systems. In most multirotor platforms, the airframe is developed from the electronics outward. The internal layout is defined first, and the external geometry is built around it. Aerodynamic refinement is then performed within those constraints. For Raven, the process was reversed. The design began with the parametrically defined outer mold line from Chapter 3, and the internal structure was designed to fit within that shell. The design needed to preserve the aerodynamic geometry while packaging flight electronics, maintaining structural integrity under aerodynamic and inertial loading, and remaining practical to manufacture and assemble. Figure 4.1 shows an exploded view of the complete airframe.

4.2 Manufacturing Considerations

The entire airframe was manufactured using a Bambu Lab P1S FDM printer with PLA filament. FDM builds parts layer by layer, with overhangs beyond approximately 45 degrees generally requiring support structures. The build volume of the P1S imposed a hard limit on individual part size and influenced how the vehicle was divided into printable components.

Within the ACL workspace, machining, injection molding, and other subtractive or tooling-based methods were not available. Two printers were accessible: a Formlabs Form 3+, which uses LFS, and the Bambu Lab P1S. LFS produces higher dimensional accuracy and smoother surfaces, but requires longer print times, a more involved post-processing workflow (washing, UV curing), uses resins that are generally more brittle and more expensive per

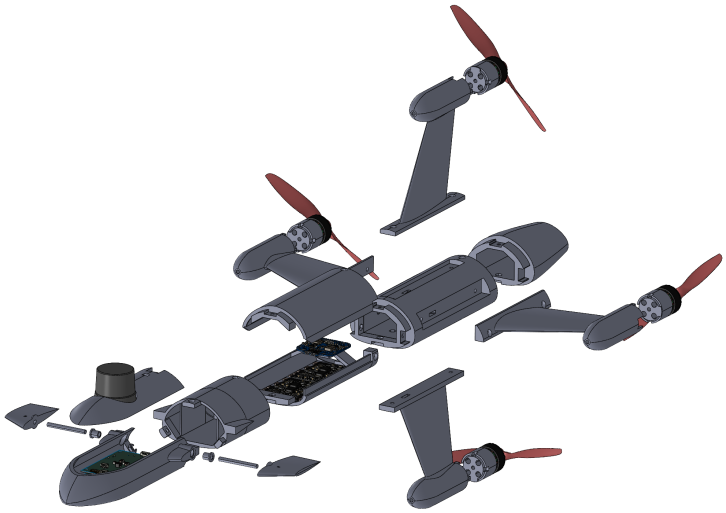

Figure 4.1: Exploded view of the Raven airframe showing all structural components

unit volume, and has a smaller build volume. The P1S was selected for its faster iteration cycle, lower cost, and larger build volume.

Two filament materials were considered: PLA and PA6-CF20, a carbon-fiber-reinforced nylon. PA6-CF20 offers higher specific strength and heat resistance but requires longer print times, tighter process control, rougher surface finishes, higher extrusion temperatures, stricter humidity management, and costs more per kilogram. PLA was selected for this iteration. For a more mature design with fewer expected revisions, PA6-CF20 would be a reasonable upgrade.

FDM introduces anisotropic mechanical behavior: loads normal to the layer planes are resisted primarily by interlayer adhesion rather than by continuous deposited material. This creates a tradeoff between aerodynamic surface quality and structural strength. Printing a part in an orientation that produces smooth external surfaces may place aerodynamic loads across the weaker interlayer bonds. Orienting for structural strength instead often degrades surface quality through layer stepping [3]. These effects influenced the mechanical design.

4.3 Component Design

4.3.1 Wings and Rear Fuselage

Other high-speed quadrotor designs typically split the wing chordwise into two halves bonded or fastened to a carbon fiber X-frame [4]. This simplifies assembly but introduces adhesive failure modes and produces surface discontinuities at the wing-frame junction. Printing orientation also presents tradeoffs: printing from split upwards towards the trailing edge produces stair-stepping or material rolling, while printing with the trailing edge facing the build plate places support interfaces where they can damage edge geometry.

To prioritize aerodynamic surface quality at the wing root, the rear wings were printed root-to-tip with motor wiring routed through an internal channel. Due to the fuselage outer surface curves aggressively near the wing root, separating the wings at their geometric intersection was not practical for printing or for maintaining surface continuity. The solution is shown in Figure 4.2, where a tab structure on each wing nests into a matching recess in the fuselage.

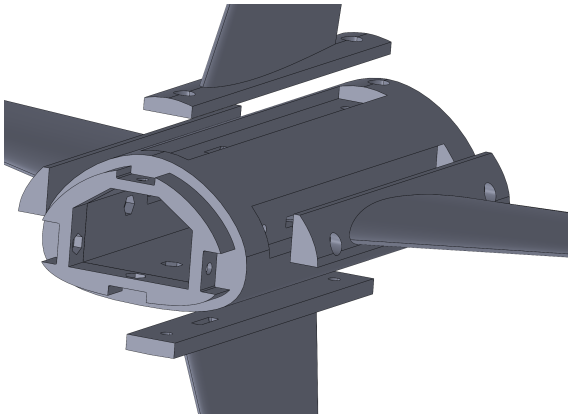

Figure 4.2: Wing-fuselage tab interface exploded view

The tab provides a flat printing base, ensures the wing is located properly during assembly, and allows M4 socket head cap screws to clamp the wing to the fuselage. This design also makes the wing attachment modular, allowing wings with different airfoils or planforms to be

substituted without modifying the fuselage provided that their root width does not exceed the tab width. The assembled junction is shown in Figure 4.3.

Figure 4.3: Assembled wing-fuselage junction

4.3.2 Nacelles and Motor Mounts

The nacelles are streamlined pods placed near the wing tips to maximize exposed wing area while providing propeller clearance. The junction between nacelle and wing is geometrically thin, and the FDM layer lines are oriented unfavorably with respect to the loads on the nacelle. In the initial geometry taken directly from the OpenVSP model, this junction failed under a qualitative hand-applied bending load even with increased wall thickness and infill.

Additional material was added by uniformly scaling the local wing cross-section and lofting the scaled profile into the wing-nacelle intersection. This approximately preserves local airfoil character while increasing the structural cross-section. The reinforced junction passed subsequent load testing.

As the vehicle does not use a conventional quadrotor X-frame, the motors mount directly to the aft face of each nacelle. The nacelle had to remain hollow for wire routing while providing a structurally adequate motor interface. The aft end of each nacelle was therefore separated into a removable motor seat.

A button-head screw entering from the nacelle tip provides axial clamping, while anti-

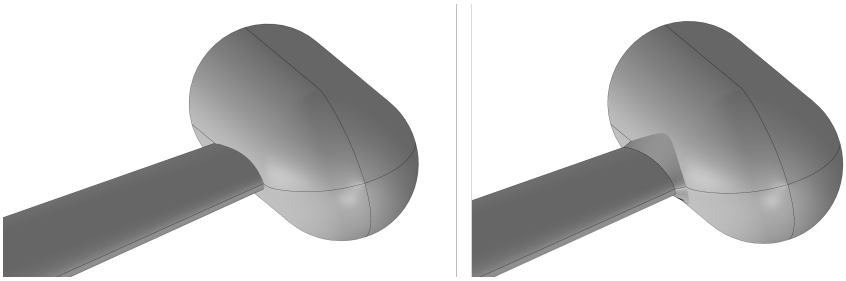

Figure 4.4: Nacelle-wing junction reinforcement. Left: original geometry from the OpenVSP model. Right: reinforced geometry with lofted cross-section fillet.

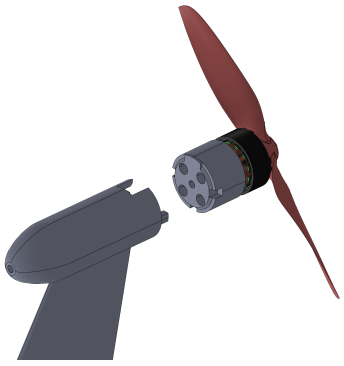

Figure 4.5: Nacelle motor mount assembly exploded view

rotation tabs extending from the nacelle body into recesses in the motor seat react motor torque. A later increase in motor size required thicker wires, and the original wire-routing holes in the motor seat began to interfere with the motor mounting holes. The design was revised so that wires enter through a slot cut into the aft face of the nacelle body, partially exposing them. This was a concession relative to the original fully enclosed routing but resolved the packaging conflict without a nacelle redesign.

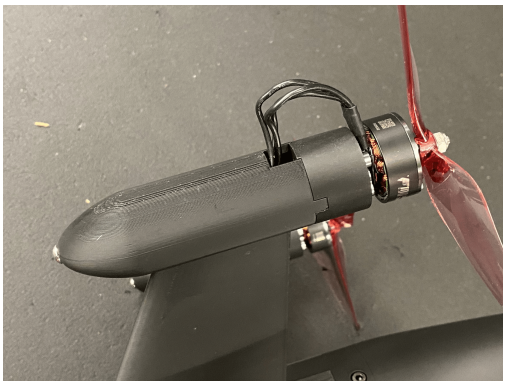

Figure 4.6: Assembled nacelle with motor mount

4.3.3 ESC and Power Distribution Integration

The aerodynamic design assumed that a custom ESC could be packaged inside each nacelle. Due to project time constraints, COTS power electronics were used instead. The Flycolor X-Cross HV3 80A ESC and the MATEK FCHUB 12S VS PDB were selected for voltage and current compatibility with the motor and power architecture.

Integrating these components introduced a packaging problem. The aft fuselage volume was reserved for the wing attachment structure and motor wiring. The ESCs needed to be close to the motors to limit wire length and resistive losses, and their heatsinks needed external airflow exposure for thermal management. These requirements placed the ESCs in the fuselage section immediately forward of the rear fuselage, with heatsinks oriented outward.

This problem illustrates the cost of designing from the aerodynamics inward. The fuselage cross-section had already been set by the aerodynamic design and by the LiPo battery volume. There was no way to package the ESCs and PDB without either increasing the fuselage cross-section or accepting thermal and assembly compromises. The concession was to elongate the fuselage; the aerodynamic consequences are discussed in Chapter 3. The ESC mounting slots were staggered to accommodate asymmetric filtering capacitors and distribute motor wires between the left and right sides without crossing. Figure 4.7 shows the designed layout. The ESCs were bonded into the fuselage using insulating, heat-resistant RTV silicone, which

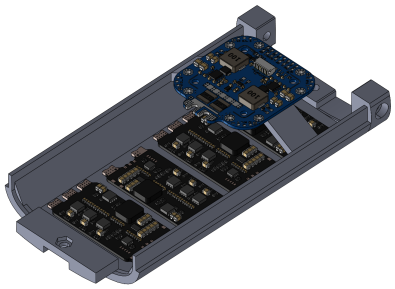

Figure 4.7: ESC and PDB integration bay without cap

provided retention and vibration damping while keeping heatsink faces exposed to airflow. The PDB was mounted above the ESCs to shorten signal and power wiring. Figure 4.8 shows the installation before closure.

The assembly process was designed so that components could be installed and wired before the fuselage was enclosed. The fuselage section was split along a horizontal parting line, and a removable cap was installed after wiring was complete. This maintained access for assembly, soldering, and maintenance.

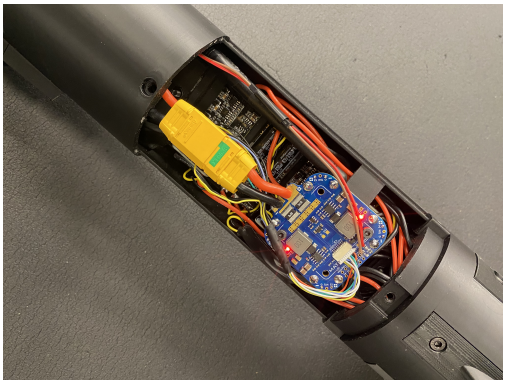

Figure 4.8: ESC and PDB installation in the mid-fuselage bay prior to cap installation

4.3.4 Fore Fuselage, Active Canards, and Avionics Integration

The fore fuselage houses the canard actuation system, LiPo battery, and flight avionics. A rectangular pocket was cut into the fuselage interior to accommodate the battery. Because this created the thinnest shell region on the airframe, vertical support members were added at the pocket boundaries. The battery section was otherwise kept open to allow the power and signal harness to pass between the aft power electronics and forward avionics sections.

Each horizontal canard is actuated independently by a servo motor aligned with the canard rotation axis, avoiding the need for linkage mechanisms. A custom 3D-printed servo horn connects the servo output spline to a D-profile shaft. The shaft passes through a self-lubricating bushing pressed into the fuselage wall and inserts into the canard root with an interference fit. Set screws prevent axial displacement; the D-profile prevents relative rotation at both the horn and canard interfaces. A HobbyPark 5.5 N·m rated servo was selected as its torque rating exceeds the aerodynamic hinge moment at design conditions by a large margin.

The avionics section used the same removable-cap approach as the power electronics section. The Sparkie flight controller and its carrier board were placed forward of the canard servos. The GPS module was mounted on top of the removable cap, with structural material separating it from the avionics boards below. This separation was found during integration

testing to be necessary for reliable GPS performance due to electromagnetic interference from the avionics. Because GPS was flight-critical, the aerodynamic penalty of the external mount was accepted. An exploded view of the assembly is shown in Figure 4.9.

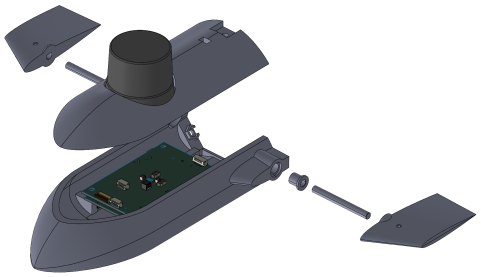

Figure 4.9: Avionics bay with Sparkie flight controller, canard mechanism, and GPS module

4.3.5 Complete Vehicle Assembly

The fully assembled vehicle is shown in Figure 4.10.

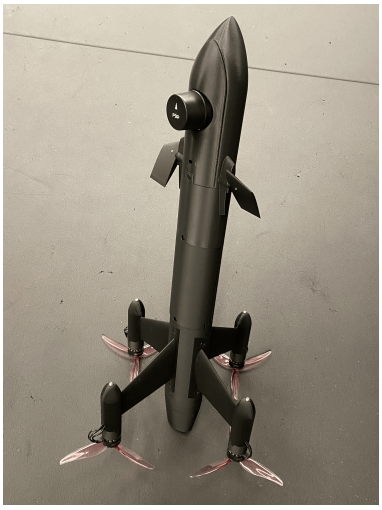

Figure 4.10: Fully assembled Raven vehicle

Chapter 5

FLIGHT EXPERIMENTS

5.1 Real-World Experiment

The Raven vehicle was tested in outdoor untethered flight only in the VTOL regime. Forward cruise flight was not attempted due to incomplete electromechanical integration of the canard control surfaces, insufficient controller tuning for the transition maneuver, and the absence of validated aerodynamic data for the flight configuration across the α range required for transition, as discussed in Chapter 3.

The flight was conducted in PX4 position mode using the cascaded PID control architecture described in Chapter 2. In this mode, stick inputs command velocity setpoints; the position controller generates attitude setpoints, which the attitude and rate controllers track. All gains were tuned by hand with conservative values that prioritized stability over responsiveness. The flight lasted approximately 59 seconds, reached a maximum altitude of 3 m, and covered a maximum horizontal distance of approximately 10 m from the takeoff point. The vehicle was manually commanded through several planar maneuvers and yaw rotations.

Figure 5.1 presents the flight telemetry. The position trace shows that the vehicle maintained altitude within approximately 3 m while drifting laterally in response to stick commands. The roll and pitch traces show attitude amplitudes of $\pm 32^\circ$ in roll and $\pm 41^\circ$ in pitch. These are large for a hover flight and reflect the difficulty of stabilizing this airframe: the CG sits approximately 15 cm above the rotor plane, creating an inverted pendulum effect that the attitude controller must continuously correct. The actual attitude follows the commanded setpoints but with noticeable oscillation, consistent with the conservative gain selection.

The motor command plot shows persistent high-frequency activity across all four motors,

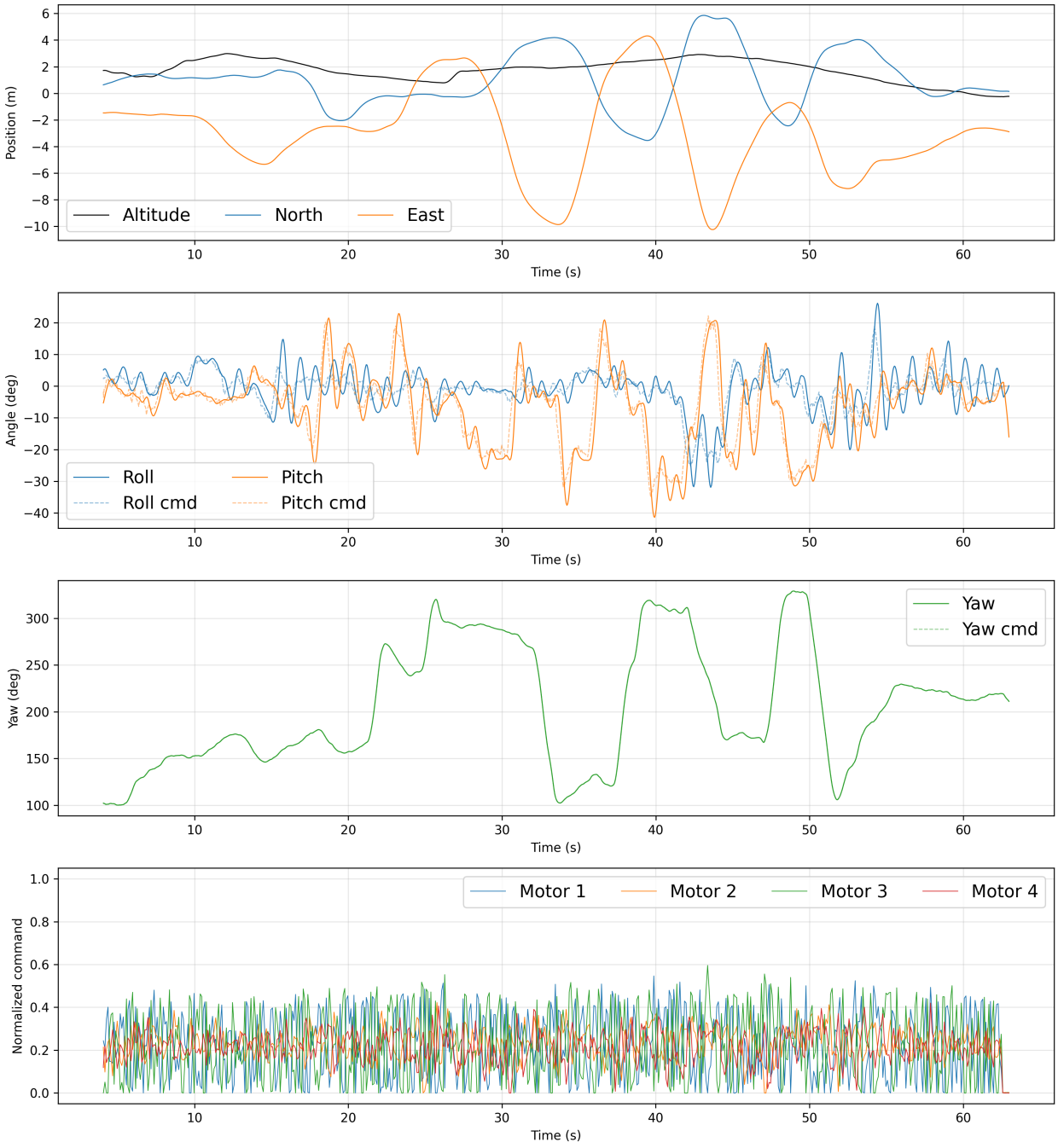

Figure 5.1: VTOL hover flight telemetry. From top to bottom: position versus time, roll and pitch attitude with commanded setpoints, yaw with commanded setpoint (unwrapped), and normalized motor commands.

with normalized commands ranging from 0 to 0.60 and a mean of 0.22. The vehicle operated well below motor saturation, indicating adequate thrust margin. However, motors 1 and 3 exhibit roughly twice the variance of motors 2 and 4, suggesting that one diagonal motor pair carried the majority of the attitude correction workload. This asymmetry suggests either a CG offset or a difference in the roll and pitch inertia axes.

The flight demonstrates that the custom avionics stack from Chapter 2, the FDM airframe from Chapter 4, and the PX4 autopilot integration function together as a flyable system. The vehicle maintained controlled hover throughout the flight without any loss-of-control events or system failures. The oscillatory behavior and attitude deviations indicate that further gain tuning is necessary before more aggressive flight testing.

5.2 Simulated Experiment

5.2.1 Setup

A 3-DOF longitudinal simulation was developed as a preliminary proof of concept for the transition and detransition maneuvers [6]. The vehicle is modeled as a rigid body in the x-z plane (two translational and one rotational DOF) with no rotor or actuator dynamics, and with passive canards. Aerodynamic lift and drag are computed from linear fits to the flight configuration VSPAERO results presented in Section 3.5. The model does not include stall, and C_L is linearly extrapolated beyond the VLM simulated α range.

Aerodynamic pitching moment is omitted, as Section 3.4.3 showed that the VLM-predicted C_m for the initial configuration disagreed with wind tunnel measurements in both magnitude and sign. Since the flight configuration was not wind tunnel tested, its VLM-derived C_m is likely incorrect, and including it would introduce error without improving model accuracy. Pitch control is provided entirely through differential thrust.

A state-machine controller sequences the vehicle through seven phases: vertical ascent, initial pitch-over, gravity turn, cruise acceleration to 100 m/s, cruise deceleration, pitch-up detransition, and vertical descent. Within each phase, a velocity PID commands total thrust

along the body axis based on the speed error, and an attitude PID distributes that thrust between fore and aft motor groups to track the phase-specific pitch setpoint. During cruise, an additional PID holds vertical speed near zero by adjusting the pitch setpoint, which maintains level flight and results in the vehicle trimming at the α required to balance lift and weight at the current airspeed. All gains were tuned by hand. The transition speed is set to 37 m/s, corresponding approximately to the minimum airspeed at which the aerodynamic surfaces can fully support W at moderate α ($\alpha \approx 10°$).

5.2.2 Results

Figure 5.2 presents the simulated maneuver. The vehicle completes the full transition, cruise, and detransition cycle successfully, taking approximately 62 seconds from takeoff to the beginning of vertical descent and covering roughly 3.3 km of horizontal distance.

During cruise ($t \approx 8.6$ to 13.5 s), the vehicle reached 100 m/s with a mean α of $-0.4°$. The negative α is expected: at 100 m/s, the zero-α lift ($C_{L_0} = 0.169$) exceeds W, so the vehicle trims slightly nose-down to reduce lift. At the design cruise speed of 40.47 m/s, zero-α lift is well below W, and the vehicle would trim at a positive α.

The α reaches approximately $-15°$ during the gravity turn and $+15°$ during the detransition. Both values are slightly outside the VLM simulation range, though the linear model does not capture the early stall behavior that could occur at these angles.

This simulation demonstrates the geometric and kinematic feasibility of the transition maneuver concept, but it is not a validated prediction of flight performance. A higher-fidelity simulation with validated aerodynamic data across the full α range, measured inertia properties, and a closed-loop transition controller would be required before attempting the maneuver in flight.

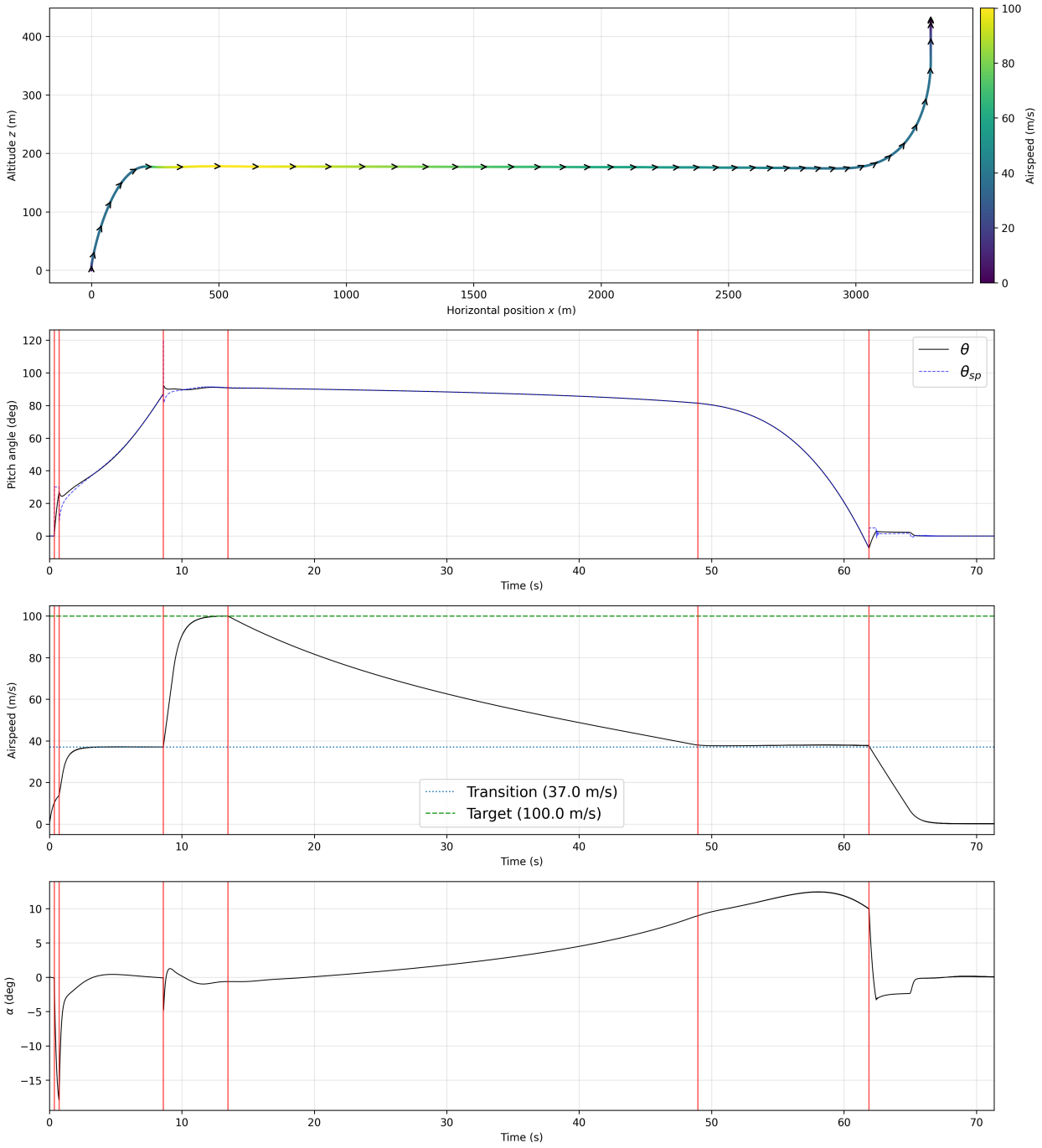

Figure 5.2: Simulated transition and detransition maneuver. From top to bottom: trajectory in the x-z plane colored by airspeed with body orientation, pitch angle and pitch setpoint versus time, airspeed versus time, and α versus time. Red vertical lines indicate state transitions.

Chapter 6

CONCLUDING REMARKS

6.1 Summary of Contributions

This thesis developed two quadrotor platforms and the supporting systems required to fly them. Firelight validated the PX4 port onto the custom Sparkie flight controller through manual and autonomous flight, including execution of externally generated reference trajectories from multiple software stacks. Raven, a tailsitting quadrotor with a lifting-body fuselage, was designed from the aerodynamic shell inward, fabricated entirely by FDM, and flown in the VTOL regime.

The firmware implementation produced a working PX4 port for a flight controller with no prior software support. The aerodynamic exploration produced a parametric airframe, VLM analysis of two configurations, and wind tunnel validation that quantified the limitations of VLM for this geometry. The mechanical design effort completed physical integration and resolved the packaging conflicts inherent to the aerodynamics-inward design approach. The VTOL flight test confirmed that the integrated system is flyable, and a longitudinal simulation demonstrated the geometric feasibility of the transition maneuver. While the system is not ready to explore its full flight envelope, it establishes a baseline with documented design choices and identified gaps.

6.2 Future Research

6.2.1 Aerodynamic Validation and Full-Envelope Control

Wind tunnel testing of the flight configuration is the highest-priority task because it gates transition controller development and cruise flight attempts. With validated aerodynamic data, a closed-loop transition controller with airspeed-scheduled pitch control can be devel-

oped. As tethered testing is impractical to explore cruise and full-envelope flight, mounting a powered vehicle in the wind tunnel integrated with an airspeed sensor to observe responses over a range of airspeeds and vehicle attitudes would provide a lower-risk path to controller validation.

The full-envelope problem is also a candidate for learning-based [30] or measurement-based control methods [33, 40]. These approaches could reduce the need for scheduled gains or discrete control modes, and remove reliance on an accurate aerodynamic model.

6.2.2 Firmware and Manufacturing

Several of the firmware solutions described in Chapter 2 are hard-coded for specific hardware configurations, and future work could generalize them for reuse. PX4 modules could be reworked or rewritten to improve computational efficiency and runtime performance. Additional sensors and peripherals could be integrated, and the networking and distributed-systems layer could be improved, moving the firmware closer to production-level robustness.

FDM manufacturing with PLA constrained the Raven airframe design, resulting in inefficient internal fuselage volume usage. This led to removed features, aerodynamic compromises, and electronics assemblies that are difficult to service. Alternative manufacturing methods and materials could improve aerodynamic quality, structural strength, and ease of assembly. Alternative designs and internal component arrangements can improve stability and control through CG variation.

6.2.3 Firelight Controller Optimization

Firelight was tuned within the PX4 control architecture with only gains changed. More advanced controllers, such as the 6-DOF controller described in [39] through frequency-domain characterization could maximize bandwidth while maintaining disturbance rejection. Firelight's validated firmware and high thrust-to-weight ratio make it a capable platform to explore robust and optimal controllers.

6.3 Integrated Co-Design

This thesis developed each Raven subsystem sequentially, and the interactions between them were addressed reactively. The fuselage was elongated to fit ESCs, vertical canards were removed for packaging, and the CG ended up where the hardware placed it. A coordinated approach that considers airframe geometry, structural layout, mass distribution, and control allocation simultaneously is a research problem because the design space is high-dimensional, the objectives compete, and the coupled physics and system dynamics span aerodynamics, structures, mechanics, and control.

The Raven baseline and its OpenVSP parametric model provide the platform and geometry for an integrated co-design study. Such a study could quantify the tradeoffs between forward-flight efficiency, hover controllability, transition robustness, structural mass, and acoustic signature, and identify optimal configurations for specific mission requirements.

BIBLIOGRAPHY

[1] Ira H. Abbott and Albert E. Von Doenhoff. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Dover Publications, New York, 1959.

[2] The Kubernetes Authors. Overview, 2026.

[3] Colin Baxter and Miguel Salguero. Design and fabrication of models for test in the 3x3 wind tunnel. Technical guide, William E. Boeing Department of Aeronautics & Astronautics, University of Washington, n.d.

[4] Mike Bell. Making the world's fastest drone, 2024.

[5] Mike Bell. 658 km/h (408 mph)! when everything just goes right, 2026.

[6] Ben Chung. Ravensim, 2026.

[7] Benjamin Chung, Rajeev S. Voleti, and Jason Zhou. *Julia Programming for Real-Time Flight Control on Embedded Platforms*.

[8] Apache NuttX Contributors. Nuttshell (nsh), 2026.

[9] Roberto de'Pompeis, Paolo Cinquetti, and Sergio Martini. Development and certification flight test on the piaggio p.180 avanti aircraft: A general overview. In *SAE Technical Paper Series*. SAE International, 1991.

[10] James DeSpirito et al. Canard enhancement with gurney flaps. Technical Report ARL-TR-8547, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, 2018.

[11] Federal Aviation Administration. Small unmanned aircraft systems (UAS) regulations (Part 107), 10 2020.

[12] Samuele Gobbi. Fastest drone fpv 557km/h 347mph guinness world record, 2024.

[13] Christopher R. Hayner, Samuel C. Buckner, Daniel Broyles, Evelyn Madewell, Karen Leung, and Behçet Açikmeşe. Halo: Hazard-aware landing optimization for autonomous systems. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3261–3267, 2023.

[14] Christopher R. Hayner, John M. Carson III, Behçet Açıkmeşe, and Karen Leung. Continuous-time line-of-sight constrained trajectory planning for 6-degree of freedom systems. *IEEE Robotics and Automation Letters*, pages 1–8, 2025.

[15] Ruiyang He, Haiying Sun, Xiaoxia Gao, and Hongxing Yang. Wind tunnel tests for wind turbines: A state-of-the-art review. *Renewable and Sustainable Energy Reviews*, 166:112675, 2022.

[16] Holybro. Pixhawk rpi cm4 baseboard, 2026.

[17] Tran Hung, Khiem Pham, Cong Dao, Trang Nguyen, Anh Dinh Le, and Van Do. Aerodynamic drag of axisymmetric models with different boattail angles under subsonic and supersonic flow conditions. *Journal of Mechanical Science and Technology*, 37:1–13, 12 2023.

[18] Microchip Technology Inc. *KSZ8081RNA/RND 10BASE-T/100BASE-TX PHY with RMII Support*. Microchip Technology Inc., 2026. DS00002199F.

[19] Joshua T Bryson, Ilmars Celmins, and Frank E Fresconi . Aerodynamic model for canard control actuation on a subsonic, gun-launched munition, gun-launched munition, 2018. DTIC report, accessed March 19, 2026.

[20] Raspberry Pi Ltd. Compute module hardware, 2026.

[21] Raspberry Pi Ltd. *Raspberry Pi Compute Module 4 Datasheet*. Raspberry Pi Ltd, 2026.

[22] Floris Mariën. Software testing: Vspaero. Master's thesis, HAW Hamburg, 2021.

[23] Robert A. McDonald and James R. Gloudemans. Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design. In *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, 2022.

[24] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, 2015.

[25] Ashley Van Milligan. Drag of nose cones, August 2013.

[26] NASA. Vortex-lattice utilization. Technical Report NASA SP-405, NASA-Langley, Washington, 1976.

[27] NXP Semiconductors. *UM10204: I2C-bus specification and user manual*. NXP Semiconductors, rev. 7.0 edition, October 2021. Document Number: UM10204.

[28] Alex E. Ockfen and Konstantin I. Matveev. Aerodynamic characteristics of naca 4412 airfoil section with flap in extreme ground effect. *International Journal of Naval Architecture and Ocean Engineering*, 1(1):1–12, 2009.

[29] OptiTrack. Natnet sdk, 2026.

[30] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022.

[31] Philipp P., Rob McDonald, and Brandon Litherland. Vspaero convergence, August 2018.

[32] Piaggio Aero Industries. P.180 avanti ii – performance, n.d.

[33] Stefan A Raab, Jiannan Zhang, Pranav Bhardwaj, and Florian Holzapfel. Proposal of a unified control strategy for vertical take-off and landing transition aircraft configurations. In *2018 Applied Aerodynamics Conference*, page 3478, 2018.

[34] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. American Institute of Aeronautics and Astronautics, Reston, VA, 5th edition, 2012.

[35] Red Bull. World's fastest camera drone vs f1 car (ft. max verstappen), 2024.

[36] Lattice Semiconductor. *iCE40 LP/HX Family Data Sheet*. Lattice Semiconductor, 2026. DS1040.

[37] Lattice Semiconductor. *iCE40 LP/HX/LM Family Handbook*. Lattice Semiconductor, 2026.

[38] STMicroelectronics. *STM32H745xI/G datasheet*. STMicroelectronics, 2026. DS12923 Rev 3.

[39] Michael Szmuk. *Successive convexification & high performance feedback control for agile flight*. PhD thesis, University of Washington, 2019.

[40] Ezra Tal and Sertac Karaman. Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Transactions on Control Systems Technology*, 29(3):1203–1218, 2020.

[41] MAVLink Development Team. Mavlink developer guide, 2026.

[42] PX4 Development Team. System startup, 2026.

[43] PX4 Development Team. uorb messaging, 2026.

[44] PX4 Development Team. uxrce-dds (px4-ros 2/dds bridge), 2026.

[45] Jonathan W. Valvano. *Embedded Microcomputer Systems: Real Time Interfacing*. Cengage Learning, Stamford, CT, 3rd edition, 2011.

[46] William E. Boeing Department of Aeronautics & Astronautics. 3x3 low-speed wind tunnel, 2026.