Development and Optimization of Flight Dynamics, Control Laws and Avionics System for a UAV with a Multi-scale Optimized Blended Wing Body Configuration

Ze Feng (Ted) Gan[1], Jiajun (Kevin) Feng[2], Fidel Khouli[3], Mostafa S.A. ElSayed[4], and Fred Nitzsche[5]*
Carleton University, Ottawa, Ontario, K1S 5B6, Canada

Abstract

The research presented in this paper was developed by the multiscale-optimized Blended Wing Body (BWB) Unmanned Aerial Vehicle (UAV) team, an undergraduate capstone project at the Carleton University Department of Mechanical and Aerospace Engineering. The development of the flight dynamics model of the BWB-UAV for stability and control is presented. The stability of the longitudinal and the lateral directional flight modes is examined as a function of flight and design parameters. The design of pole placement and LQR controllers is discussed, and their performance is compared. The PixHawk 2.1 Cube is selected as the flight control unit and the integration of the developed flight dynamics and control laws as a new airframe in the PX4 firmware is described. Parameter optimization and tuning of the flight dynamics model and the control laws using a hardware-in-the-loop simulation setup is presented and discussed. Lessons learned from flight tests, which test the developed firmware using off-the-shelf platforms before migrating to the presented BWB-UAV, and the collected test data are discussed.

1.0 Introduction

Blended-wing-body (BWB) aircraft configurations were first sought primarily for military stealth applications (e.g. Northrop Grumman B-2 spirit, Lockheed F-117 Nighthawk), given their small radar and acoustic signature. However, present interest in BWB aircraft by NASA, Boeing, and ACARE (Advisory Council for Aeronautics Research in Europe) involve civilian applications [1, 2]. This is partially because BWB aircraft offer increased payload capacity for passenger and/or cargo transport, without increasing ground footprint relative to a conventional aircraft, thus making them compatible with existing infrastructure. Furthermore, the increased lifting surface area yet reduced total wetted area increases the lift-to-drag ratio. The streamlined geometry also reduces both interference and pressure/form drag, thus reducing fuel consumption [3, 4, 5].

BWB aircraft designs often incorporate numerous control surfaces, engines (enabling distributed propulsion), and even tails for increased maneuverability [2, 4, 5]. When tails are present, they have smaller surface area and moment arm (about the aircraft centre of gravity) relative to conventional aircraft, reducing the tails' effectiveness in maneuverability. However, this acrobatic maneuverability is not required for cargo transport missions, justifying the design simplicity of two control surfaces (termed elevons because they serve as both elevators and ailerons), one propulsion power plant, and no tail. This configuration poses control challenges, especially due to relying on roll-yaw coupling for yaw control [2, 4, 5]. Although such 2-elevon, tailless, single engine aircraft configurations are common among hobbyist UAVs (e.g. Versa Wing, Skywalker X8, Telink Toro 900), they are less-frequently studied in the academic literature [6, 7, 8], thus motivating the present study.

[1] BEng Aerospace Engineering, Carleton University, Ottawa, ON, Canada
[2] BEng Aerospace Engineering, Carleton University, Ottawa, ON, Canada
[3] Assistant Professor, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, Canada
* Corresponding author: fidel.khouli@carleton.ca
[4] Assistant Professor, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, Canada
[5] Professor, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, Canada

2.0 Aircraft Model

During the first two years (2017-2019) of the BWB UAV undergraduate capstone project at the Carleton University Department of Mechanical and Aerospace Engineering, the Peregrine-1 UAV was designed based off the commercial off-the-shelf Skywalker X8 UAV.

The Skywalker X8 is used for flight tests to verify the aircraft model and controllers; accordingly, it is analyzed in this report, as the Peregrine-1 will be manufactured during the 2019-2020 academic year. The Peregrine-1 has a nearly identical outer mold line (OML) except for a removed front bulge; therefore, its aerodynamic properties are very similar to the Skywalker X8.

State space modeling was chosen to model the aircraft dynamics to permit the use of simple conventional linear controllers and decouple longitudinal and lateral uncontrolled dynamics. However, the controlled dynamics are not decoupled, since the elevon inputs are common to both the longitudinal and lateral input vectors. Reconciling differing demanded elevon angles from the longitudinal and lateral controllers is a priority for future work.

Given the simple cargo transport mission, the governing equations were linearized about steady level cruise, and gain scheduling was deemed unnecessary.

The symbolic longitudinal and lateral state equations were derived to be [9, 10]:

$$\dot{x} = \begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{V}_T \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{\bar{q}S}{mV_{T_e}}\left(C_{D_e} + C_{L_\alpha}\right) & 1 - \frac{\bar{q}Sc}{mV_{T_e}^2}C_{L_\alpha}(h - h_0) & -\frac{2\bar{q}S}{mV_{T_e}}C_{L_e} & 0 \\ \frac{\bar{q}Sc}{I_{yy}}C_{m_\alpha} & \frac{-2C_{L_\alpha}\bar{q}Sc}{I_{yy}}(h-0.5)^2 & 0 & 0 \\ \frac{\bar{q}S}{m}\left(C_{L_e} - C_{D_\alpha}\right) & 0 & -\frac{3\bar{q}S}{mV_{T_e}}C_{D_e} & -g \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ V_T \\ \theta \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{\bar{q}S}{2mV_{T_e}}C_{L_{\delta_e}} & \frac{\bar{q}S}{2mV_{T_e}}C_{L_{\delta_e}} & 0 \\ \frac{\bar{q}Sc}{2I_{yy}}C_{m_{\delta_e}} & \frac{\bar{q}Sc}{2I_{yy}}C_{m_{\delta_e}} & 0 \\ 0 & 0 & \frac{1}{m}T_{\delta_t} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_L} \\ \delta_{e_R} \\ \delta_t \end{bmatrix}$$

$$\dot{y} = \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{\bar{q}S}{mV_{T_e}}\left(C_{y_\beta}\right) & \frac{\bar{q}bS}{2mV_{T_e}}\left(C_{y_p}\right) & \frac{\bar{q}bS}{2mV_{T_e}}\left(C_{y_r}\right) - V_{T_e} & g\cos(\theta_0) \\ \frac{\bar{q}bS}{V_{T_e}I_x'}\left(C_{l_\beta}\right) + \frac{\bar{q}bSI_{xz}'}{V_{T_e}}\left(C_{n_\beta}\right) & \frac{\bar{q}b^2S}{2V_{T_e}I_x'}\left(C_{l_p}\right) + \frac{\bar{q}b^2SI_{xz}'}{2V_{T_e}}\left(C_{n_p}\right) & \frac{\bar{q}b^2S}{2V_{T_e}I_x'}\left(C_{l_r}\right) + \frac{\bar{q}b^2SI_{xz}'}{2V_{T_e}}\left(C_{n_r}\right) & 0 \\ \frac{\bar{q}bSI_{xz}'}{V_{T_e}}\left(C_{l_\beta}\right) + \frac{\bar{q}bS}{V_{T_e}I_z'}\left(C_{n_\beta}\right) & \frac{\bar{q}b^2SI_{xz}'}{2V_{T_e}}\left(C_{l_p}\right) + \frac{\bar{q}b^2S}{2V_{T_e}I_z'}\left(C_{n_p}\right) & \frac{\bar{q}b^2SI_{xz}'}{2V_{T_e}}\left(C_{l_r}\right) + \frac{\bar{q}b^2S}{2V_{T_e}I_z'}\left(C_{n_r}\right) & 0 \\ 0 & 1 & \tan(\theta_0) & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{\bar{q}S}{2m^2}C_{y_{\delta_a}} & -\frac{\bar{q}S}{2m^2}C_{y_{\delta_a}} \\ \frac{\bar{q}bS}{2I_xI_x'}\left(C_{l_{\delta_a}}\right) + \frac{\bar{q}bSI_{xz}'}{2I_z}\left(C_{n_{\delta_a}}\right) & -\frac{\bar{q}bS}{2I_xI_x'}\left(C_{l_{\delta_a}}\right) - \frac{\bar{q}bSI_{xz}'}{2I_z}\left(C_{n_{\delta_a}}\right) \\ \frac{\bar{q}bSI_{xz}'}{2I_x}\left(C_{l_{\delta_a}}\right) + \frac{\bar{q}bS}{2I_zI_z'}\left(C_{n_{\delta_a}}\right) & -\frac{\bar{q}bSI_{xz}'}{2I_x}\left(C_{l_{\delta_a}}\right) - \frac{\bar{q}bS}{2I_zI_z'}\left(C_{n_{\delta_a}}\right) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_L} \\ \delta_{e_R} \end{bmatrix}$$

The output equations are redundant, as the output vector is simply the state vector x. The longitudinal state vector consists of angle of attack α, pitch rate q, true airspeed perturbation V_T, and pitch angle θ. The lateral state vector consists of sideslip velocity perturbation v, roll rate p, yaw rate r, and roll angle φ. The inputs are left and right aileron deflections (positive down) δ_{e_L} and δ_{e_R}, and throttle setting (from 0 to 1) δ_t. All state variables are perturbations from their equilibrium values; accordingly, the body-centred stability axes serve as the reference coordinate frame.

Major assumptions in the state space model include neglecting propeller gyroscopic and unsteady aerodynamic effects. Compressibility effects due to Mach number and elevation were neglected, and the aircraft was assumed to be rigid.

The numerical values of each parameter for the Skywalker X8 are listed in Table 1.

Table 1: State Space Parameter Values.

Parameter	Symbol	Value	Units	Notes/Sources
True airspeed at steady level cruise	V_{T_e}	15	m/s	Flight tests
Dynamic pressure at steady level cruise	$\bar{q}$	$\frac{1}{2}\rho V_{T_e}^2$	Pa	Assume sea level air density = 1.225 kg/m^3, [9]
Wing span	b	2.1	m	Measured
Wing mean aerodynamic chord (MAC)	c	0.357	m	Measured
Wing planform reference area	S	0.75	m^2	CATIA. For a BWB, this is taken as the entire aircraft's projected area.
Total aircraft mass	m	3.5	kg	Measured on scale; constant for a battery-powered aircraft
Centre of gravity (CG) position aft of MAC leading edge (LE), divided by c [10]	h	0.2	none	CATIA
Value of h for which C_{L_q} is zero	h_0	0.75	none	[10]
Aircraft moment of inertia about the roll stability axis	I_{xx}	1.2	kg· m^2	Assume the battery mass is dominant
Aircraft moment of inertia about the pitch stability axis	I_{yy}	0.17	kg· m^2	Assume the battery mass is dominant
Aircraft moment of inertia about the yaw stability axis	I_{zz}	0.9	kg· m^2	Assume the battery mass is dominant
Aircraft product of inertia about the aircraft symmetry plane	I_{xz}	0.9	kg· m^2	Assume the battery mass is dominant
Nameless inertia term	$I_x{}'$	$\frac{I_x I_z - (I_{xz})^2}{I_z}$	kg· m^2	[10]
Nameless inertia term	$I_z{}'$	$\frac{I_x I_z - (I_{xz})^2}{I_x}$	kg· m^2	[10]
Nameless inertia term	$I_{xz}{}'$	$\frac{I_{xz}}{I_x I_z - (I_{xz})^2}$	kg^{-1}· m^{-2}	[10]
Lift coefficient at equilibrium cruise	C_{L_e}	0.26	none	Aircraft weight
Drag coefficient at equilibrium cruise	C_{D_e}	0.012	none	Thrust from flight test
Change in lift coefficient with respect to angle of attack	C_{L_α}	4.0	rad^{-1}	RANS (Reynolds-averaged Navier-Stokes) simulation using SST (shear stress transport) turbulence model in ANSYS CFX

Change in pitching moment coefficient with respect to angle of attack	C_{m_α}	-0.2	rad^{-1}	RANS in ANSYS CFX
Change in drag coefficient with respect to angle of attack	C_{D_α}	0	rad^{-1}	RANS in ANSYS CFX
Change in lift coefficient with respect to elevon deflection (both elevons deflected down)	$C_{L_{\delta_e}}$	0.7	rad^{-1}	RANS in ANSYS CFX
Change in pitching moment coefficient with respect to elevon deflection	$C_{m_{\delta_e}}$	-0.4	rad^{-1}	RANS in ANSYS CFX
Change in thrust with respect to throttle setting	T_{δ_t}	40	N	Flight test and motor test; requires further testing, but not used for any further calculations
Gravitational acceleration	g	9.806	m/s^2	SI standard
Equilibrium pitch angle	θ_0	0	rad	Negligible
Change in sideforce coefficient with respect to sideslip angle	C_{y_β}	-0.195	rad^{-1}	Inviscid panel methods [7]
Change in sideforce coefficient with respect to roll rate	C_{y_p}	-0.117	s/rad	Inviscid panel methods [7]
Change in sideforce coefficient with respect to yaw rate	C_{y_r}	0	s/rad	Inviscid panel methods [7]
Change in rolling moment coefficient with respect to sideslip angle	C_{l_β}	-0.0765	rad^{-1}	Inviscid panel methods [7]
Change in yawing moment coefficient with respect to sideslip angle	C_{n_β}	0.04	rad^{-1}	Inviscid panel methods [7]
Change in rolling moment coefficient with respect to roll rate	C_{l_p}	-0.40	s/rad	Inviscid panel methods [7]
Change in yawing moment coefficient with respect to roll rate	C_{n_p}	-0.025	s/rad	Inviscid panel methods [7]
Change in rolling moment coefficient with respect to yaw rate	C_{l_r}	0.025	s/rad	Inviscid panel methods [7]
Change in yawing moment coefficient with respect to yaw rate	C_{n_r}	-0.1252	s/rad	Inviscid panel methods [7]
Negative change in sideforce coefficient with respect to aileron deflection (right elevon up, left elevon down)	$C_{y_{\delta_a}}$	0	rad^{-1}	Factor of -1 by convention [7, 10]
Change in rolling moment coefficient with aileron deflection	$C_{l_{\delta_a}}$	0.3	rad^{-1}	[7]
Change in yawing moment coefficient with aileron deflection	$C_{n_{\delta_a}}$	0.0076	rad^{-1}	[7]

Continue - Table 1

3.0 Uncontrolled Dynamics

3.1 Aircraft Modes

The rigid aircraft modes were determined from the eigenvalues of the longitudinal and lateral state matrices (A), and are summarized in Table 2. Eigenvalues for a two-elevon BWB UAV similar to the Skywalker X8 are shown in Table 3 [6].

Table 2: Aircraft Modes of Skywalker X8.

Mode	Eigenvalue	Damping Ratio	Natural Frequency [rad/s]
Short Period	-6.56 ± 6.81j	0.694	9.46
Long period/Phugoid	-0.0267 ± 0.569j	0.0469	0.570
Dutch roll	-0.411 ± 2.93j	0.139	2.96
Spiral	-0.231	1	N/A
Roll Subsidence	-36.1212	1	N/A

Table 3: Aircraft Modes of Min's BWB UAV [6].

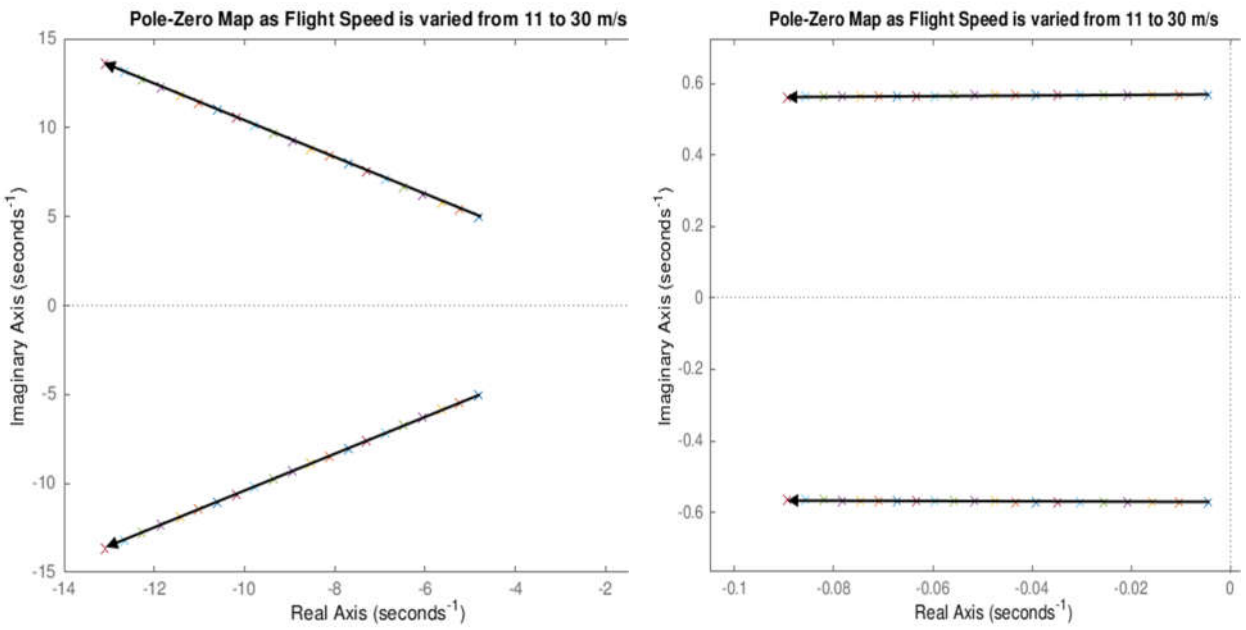
Mode	Eigenvalue	Damping Ratio	Natural Frequency [rad/s]
Short Period	-6.93 ± 12.50j	0.485	14.3
Long period/Phugoid	-8.67 ± 1.04j	0.993	8.73
Dutch roll	1.14 ± 0.7j	-0.848	1.34
Spiral	-3.64	1	N/A
Roll Subsidence	-9.20	1	N/A

Min's results have an unstable Dutch roll, and the Skywalker X8 has a barely-stable Dutch roll; these results are expected given that each UAV has no tail. Min does not have an order of magnitude difference between the natural frequencies of the short and long period modes, as with spiral and roll subsidence modes; the relative differences of the Skywalker X8 more closely agree with the literature in this respect [9, 10]. The Phugoid mode of Min is much more stable than that of the Skywalker X8, as indicated by the higher damping ratio.

3.2 Parameter Sensitivity ("Root Locus") Studies

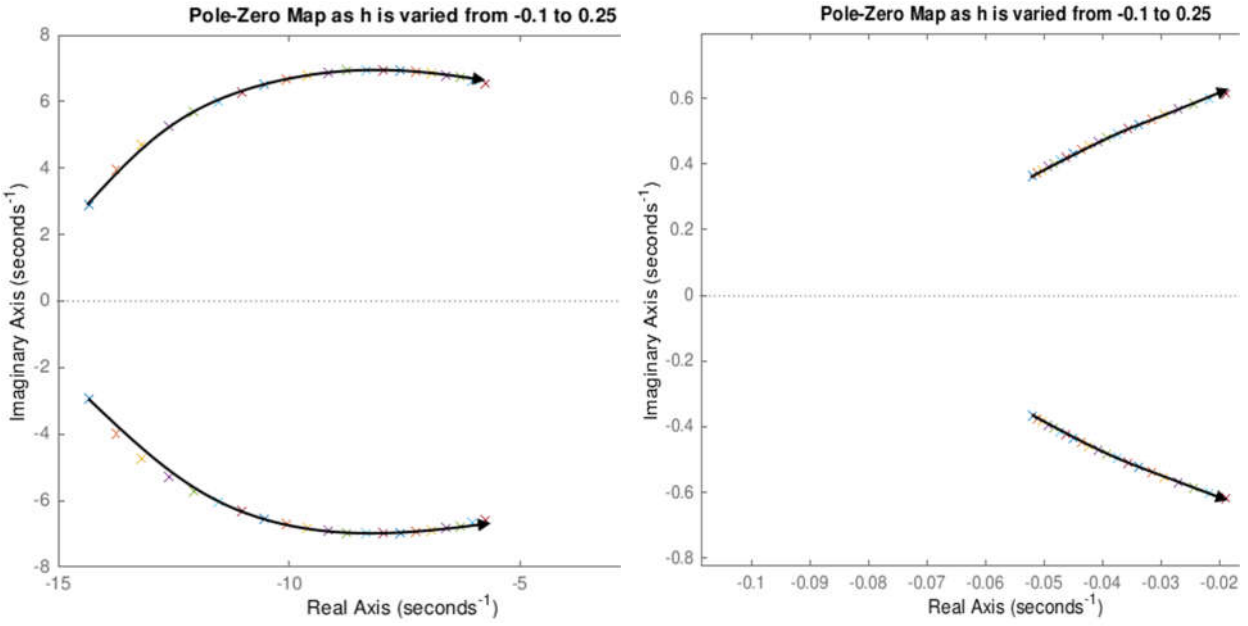
The sensitivity of the aircraft mode poles/eigenvalues to variation in parameter values is not only useful for designing for stability without the use of a controller, but for designing robust controllers: that is, controllers that are insensitive to inaccuracies in the aircraft dynamic model [9]. The varied parameters were selected to be the parameters that affect many entries of the state matrices. These varied longitudinal parameters were V_{T_e}, m, h, and I_{yy}. The varied lateral parameters were V_{T_e}, m, I_{xx}, I_{zz}, and I_{xz}. b)

Figure 1 to Figure 8 show how the longitudinal and lateral eigenvalues change when one parameter is varied within a physically-sensible range; therefore, they can be considered analogous to the root locus.



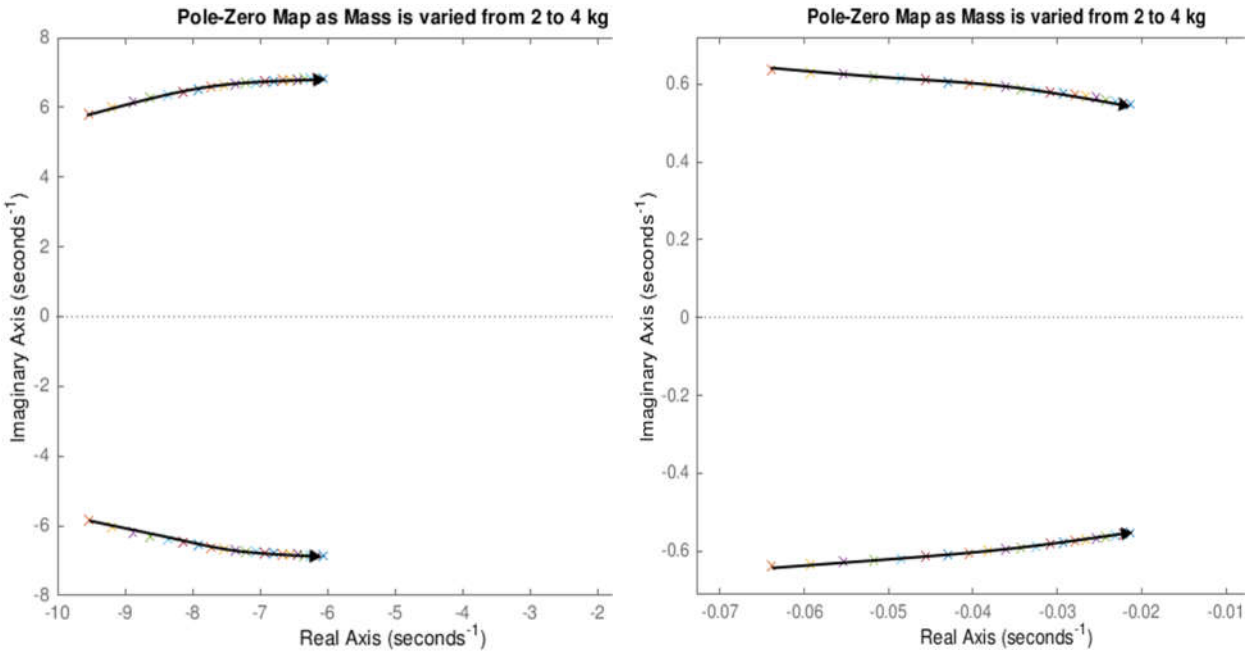
a) b)

Figure 1: Longitudinal "Root Locus" as Flight Speed Changes: a) Short Period Mode; b) Phugoid Mode.



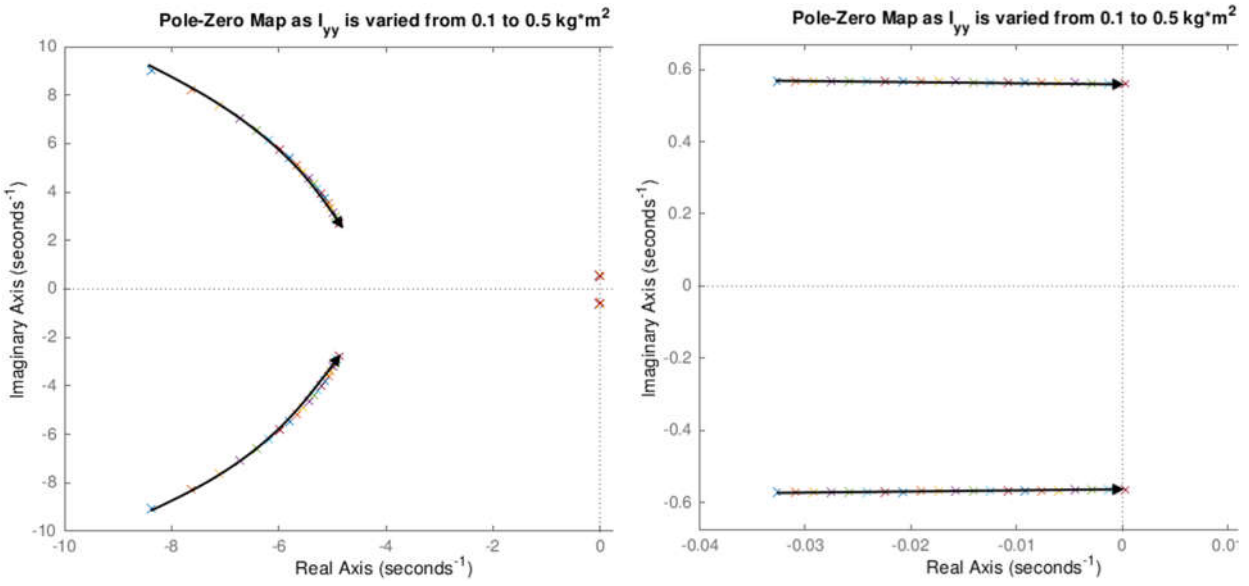
a) b)

Figure 2: Longitudinal "Root Locus" as Centre of Gravity Position Changes: a) Short Period Mode; b) Phugoid Mode.



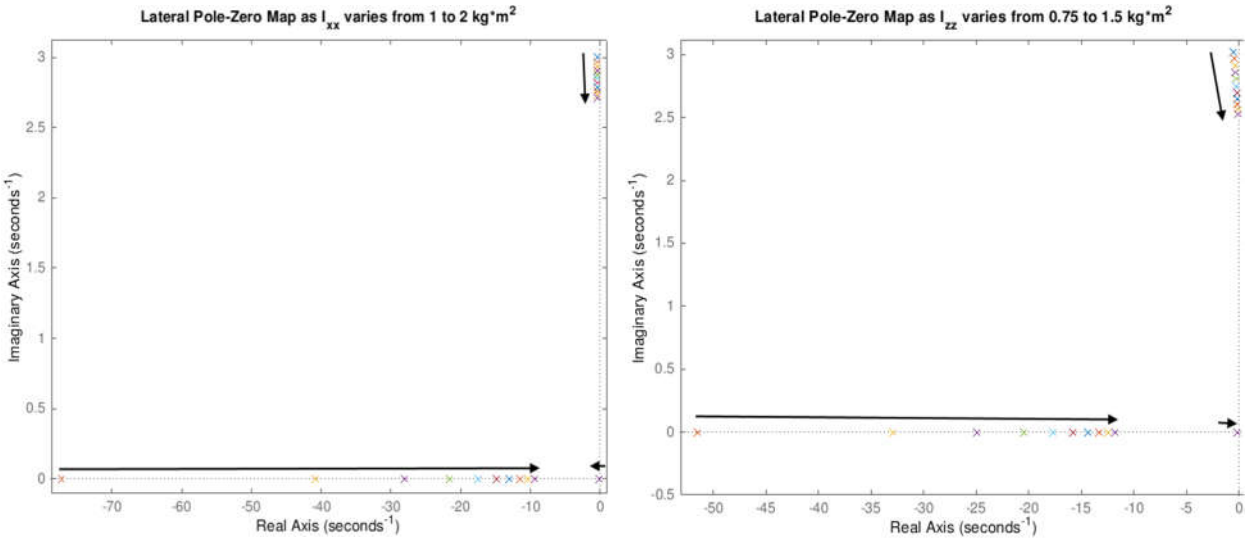
a) b)

Figure 3: Longitudinal "Root Locus" as Aircraft Mass Changes: a) Short Period Mode; b) Phugoid Mode.



a) b)

Figure 4: Longitudinal "Root Locus" as Pitch Axis Moment of Inertia Changes: a) Short Period Mode; b) Phugoid Mode.



a) b)

Figure 5: Lateral Mode "Root Locus" as Moment of Inertia Changes: a) Roll Axis; b) Yaw Axis.

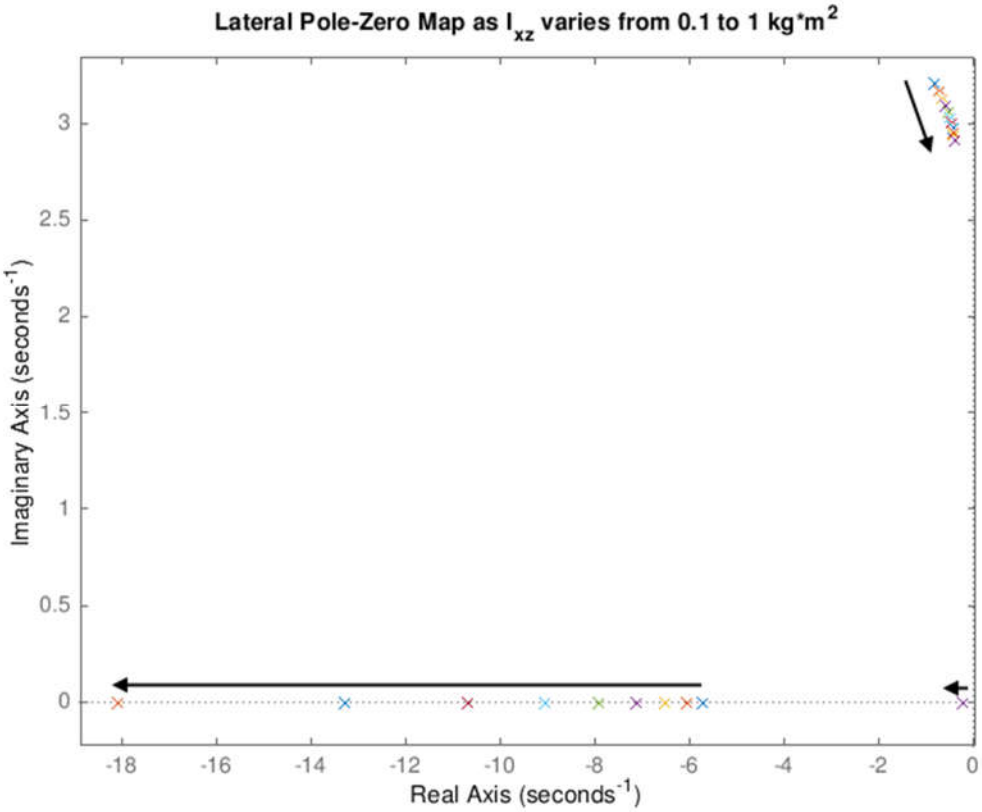


Figure 6: Lateral Mode "Root Locus" as Product of Inertia Changes.

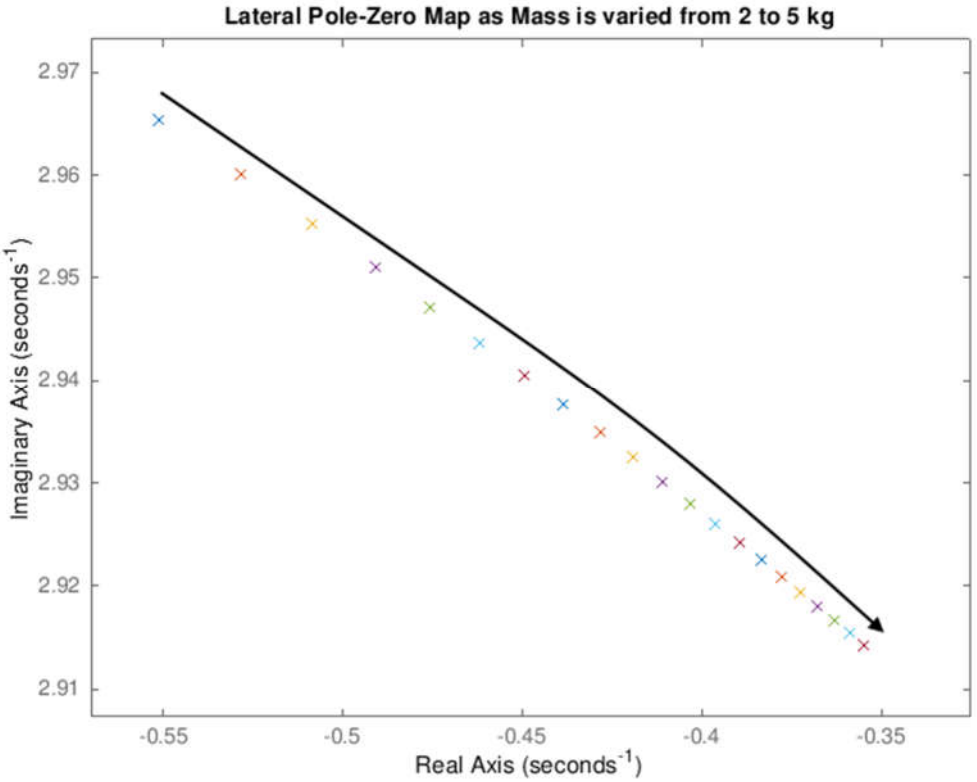


Figure 7: Dutch Roll Mode "Root Locus" as Aircraft Mass Changes (other modes insensitive).

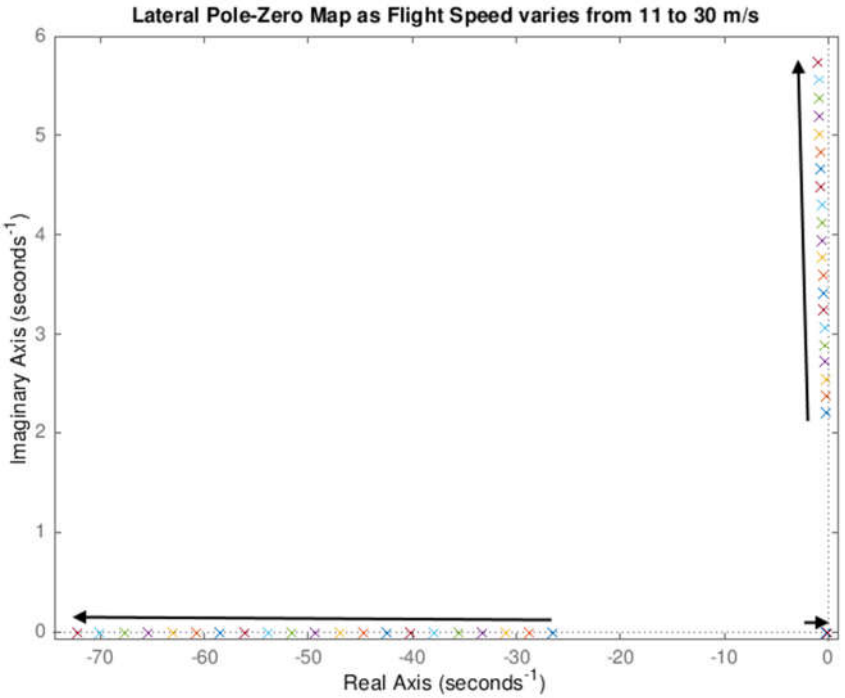


Figure 8: Lateral Mode "Root Locus" as Flight Speed Changes.

The spiral mode was found to be relatively insensitive to all varied lateral parameters, even when considering percent changes in the eigenvalue (as its magnitude is always small). In contrast, the roll subsidence mode is sensitive and is often the lateral pole that becomes unstable first.

These root locus figures can be used in design to offset changes in one parameter. For example, increasing mass or any inertia component can be offset by increasing the flight speed; this is because the air's inertia dominates over the aircraft's inertia. As expected, flight speed and mass/inertia were the most sensitive parameters.
Although one parameter was varied at a time, some varied parameters depend on each other. For example, it is difficult to change the centre of gravity position without also changing the inertia.

3.3 Uncontrolled Response

The unit impulse response is the time-domain analogue to the transfer function and is thus an insightful system property. It physically represents a pilot moving the stick quickly from and to its starting position over a short, but finite time period.
Therefore, the pitch angle response to a 1° symmetric elevon input was analyzed. Two elevon inputs were analyzed: an impulse and a doublet. The former is analytically equivalent to the Dirac Delta function, but this is physically unrealistic as it involves an impulse of infinite magnitude but infinitesimal time duration. Therefore, the impulse was modelled by a Heaviside step up, followed by a Heaviside step down after 0.1 seconds, a time duration selected based on physical UAV pilot finger movement and reaction time. Similarly, the doublet was defined by a Heaviside step up followed by a Heaviside step down with twice the magnitude after 0.1 seconds, followed by another Heaviside step back up to zero, after 0.1 additional seconds. The elevon deflection was chosen to be 1° because the system is linear, such that scaling the input scales the output by the same magnitude. The pitch angle response for step and doublet 1° elevon down input are shown in Figure 9.

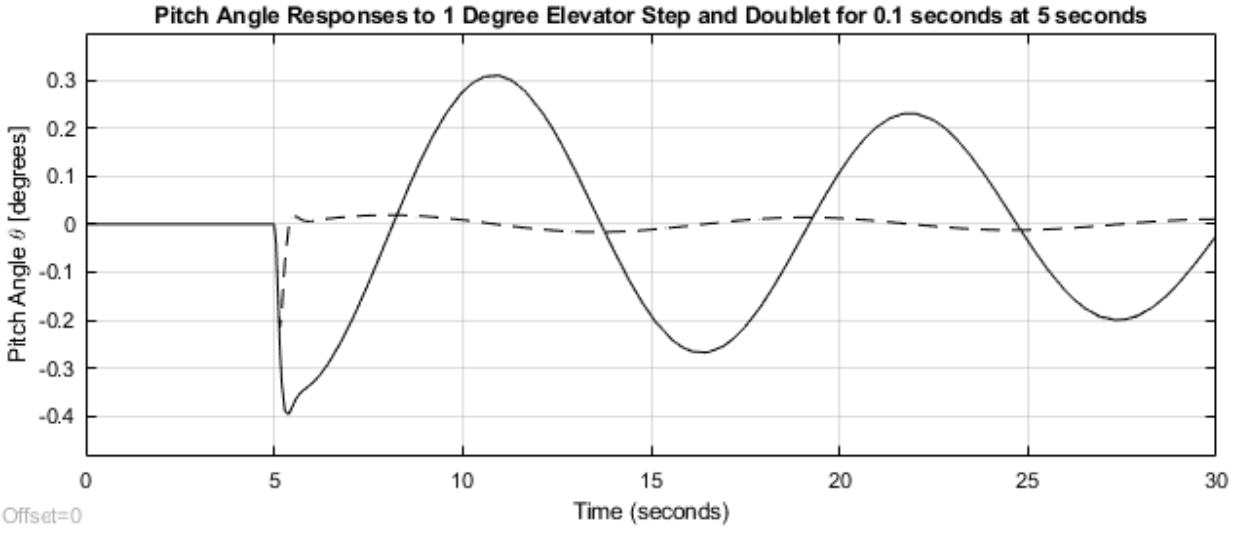


Figure 9: Uncontrolled Elevon Unit Step (solid line) and Doublet (dashed line) Responses.

These results were found to be insensitive to shorter but physically realistic step durations between 0.05 and 0.2 seconds. The maximum amplitude, period of oscillation, and 98% settling time for the responses are summarized in Table 4 below.

Table 4: Summary of Uncontrolled Response. [11]

Input	Maximum Pitch Amplitude [°]	Period [s]	Settling Time [s]
1° elevon step	0.38	11	162
1° elevon doublet	0.22	10	93

Although the settling time appears large, recall that these are the uncontrolled dynamic responses. The results of these responses will be compared to hardware in the loop (HITL) simulation responses to the same inputs, to validate the state space model against the HITL X-Plane simulation aircraft model. This is necessary because the commercial software X-Plane used for HITL simulations do not make use of a state space (i.e. linearized aerodynamic coefficients) to determine the aerodynamics of the aircraft, but instead solve inviscid CFD panel methods in real time [12].

4.0 Controlled Dynamics

4.1 State Feedback Controller

Although classical control techniques such as the PID (proportional-integral-derivative) controller are widely-used (even in complex systems such aircraft), successive loop closure of MIMO (multi-input, multi-output) systems suffer from each designed feedback loop "undoing" the progress of previously-designed loops. Thus, tuning PID controllers is an iterative process that relies on testing. This is evident in the standard open source ArduPlane controller used on the PixHawk commercial autopilot board, which relies on tuning a small range of gain values, as well as rate limiters to prevent an unbounded response (see Section 4.4) [13].

Therefore, it is desirable to design controllers for the Skywalker X8 and Peregrine-1 that involve simultaneous gain calculations using matrix equations, rather than successive scalar gains being repeatedly guessed and checked. This is one of the philosophies behind optimal control. Perhaps the simplest optimal controller is the state feedback controller, which involves a simple feedback loop with a gain matrix as the feedback transfer function, as seen in Figure 10.

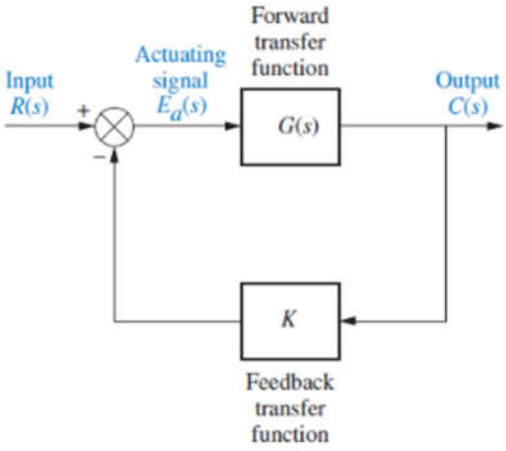


Figure 10: State Feedback Optimal Controller Structure (adapted from [14]).

State feedback can be implemented because all state variables are directly measured by the UAV's various sensors [9]. If this were not the case, output feedback would be used instead, or the states

would be estimated by use of a state estimator, such as an extended Kalman filter (EKF) used by the PixHawk's ArduPlane firmware [13].

Although the structure of the controller has been chosen, the gain matrix K can be computed using a variety of methods. One design philosophy is pole placement, which involves calculating K to obtain desired aircraft mode eigenvalues and eigenvectors; this is the subject of Section 4.2. Another design philosophy involves minimizing a chosen "cost" function or maximizing a performance index based on chosen criteria; this is the subject of Section 4.3.

4.2 Pole Placement

In addition to obtaining desired aircraft mode eigenvalues (oscillation frequency and damping), desired eigenvectors control the relative magnitude and phase of state variable oscillations for each mode. It is desirable for the short period mode to only exchange energy between α and q, and for the Phugoid mode to exchange energy between V_T and θ. These oscillations should ideally be 90° apart in phase [9]. These desired eigenvectors in Continue – Figurea are close to the uncontrolled Phugoid mode eigenvectors in Continue – Figureb, but the pitch rate varies more than what is desired. Since eigenvectors are relative, they were (as expected) unaffected by relative magnitude scaling or phase shift.

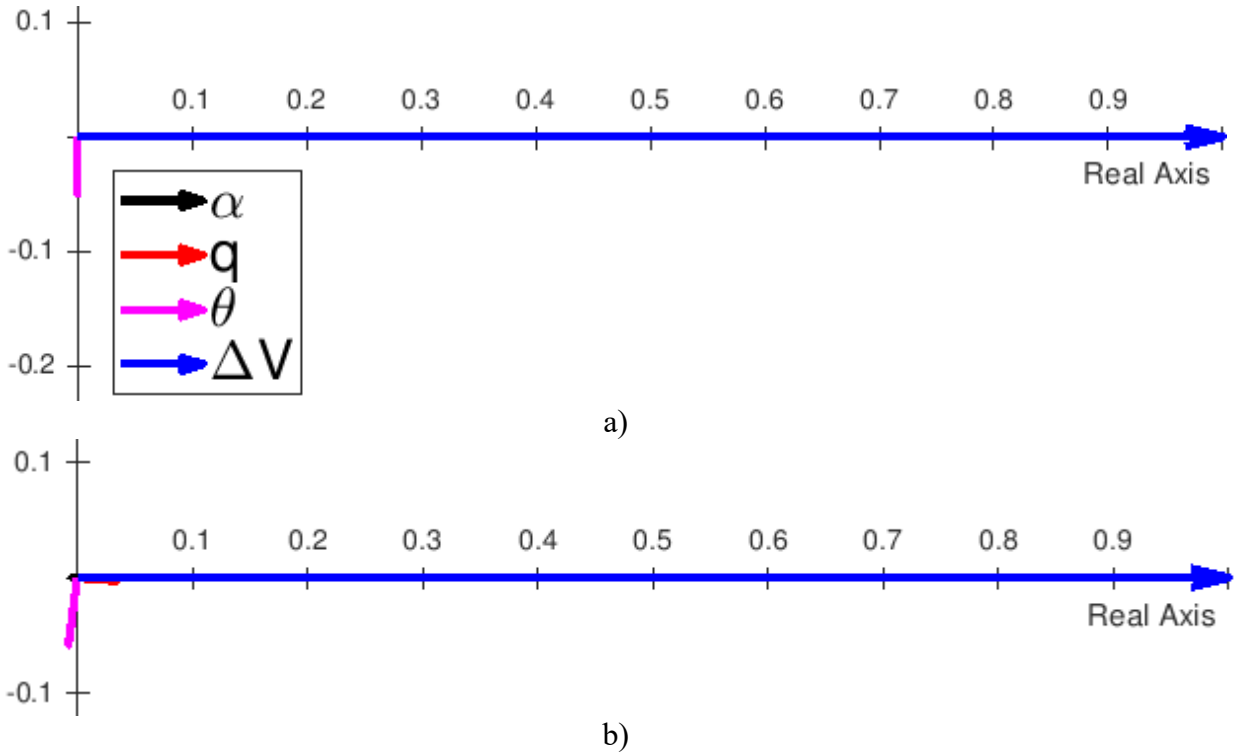
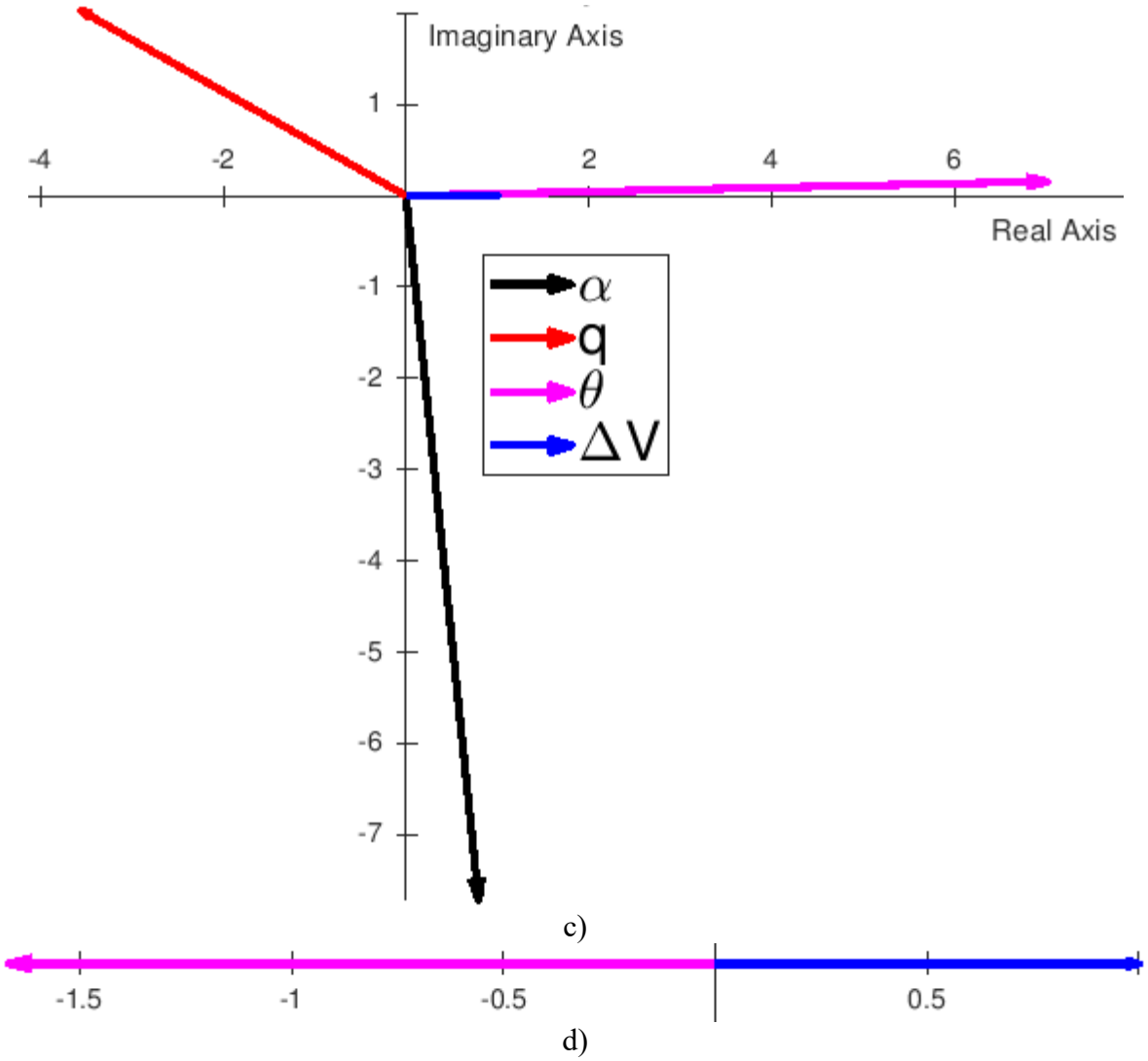


Figure 11: Phugoid Mode Eigenvector Phasor Diagrams: a) Demanded Eigenvectors; b) Uncontrolled Eigenvectors; c) Eigenvectors of Kautsky's Algorithm; d) Eigenvectors of Stevens' Algorithm.



Continue – Figure 11: Phugoid Mode Eigenvector Phasor Diagrams: a) Demanded Eigenvectors; b) Uncontrolled Eigenvectors; c) Eigenvectors of Kautsky's Algorithm; d) Eigenvectors of Stevens' Algorithm.

Kautsky's algorithm places poles to be robust (minimize sensitivity to variations in model parameters) [15] [16]. Although this method successfully assigns the eigenvalues to the demanded pole locations, the resultant Phugoid mode eigenvectors shown in Continue – Figurec are far from desirable: α and q have relatively large magnitudes, and V_T and θ are in phase.

Stevens' algorithm allows for specification of both eigenvalues and eigenvectors; the latter were selected to be those of Continue – Figurea. However, not all demanded eigenvalues were not achieved, as seen on Figure 12: purely real Phugoid poles resulted despite complex poles being demanded, as seen in Continue – Figured. Since the eigenvalues are purely real numbers, so are the eigenvector components seen in Continue – Figured; this is not desirable as a 90° phase difference cannot be achieved without imaginary components.

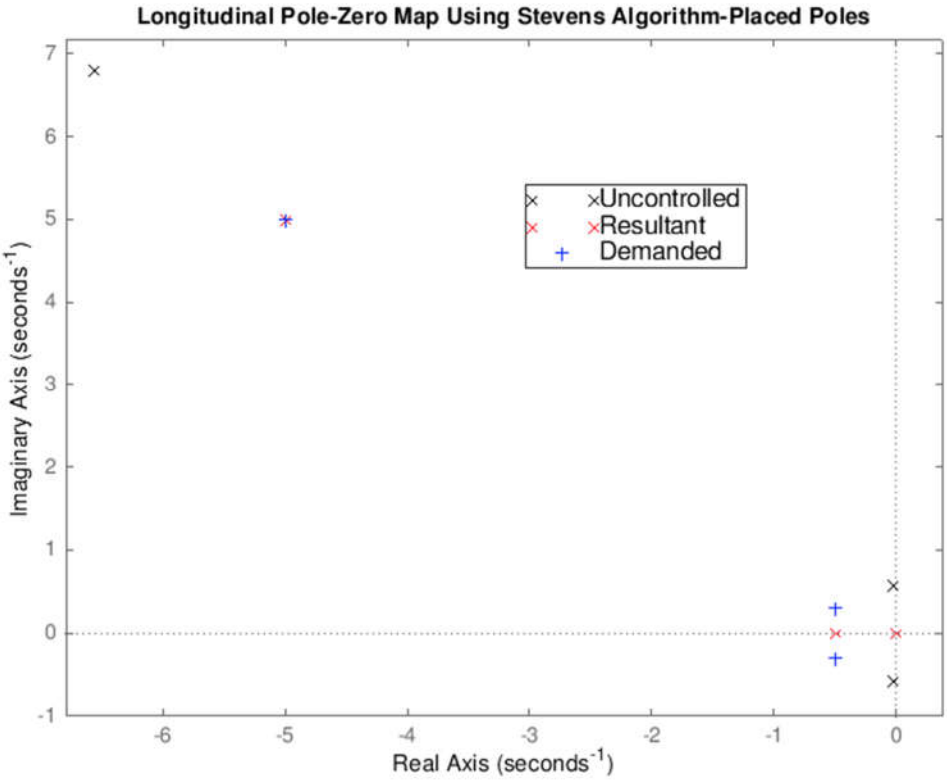


Figure 12: Closed Loop Eigenvalues from Stevens' Pole Placement Algorithm.

The resulting eigenvalues and eigenvectors were also unchanged by interchanging what component led the other by 90°, and the relative magnitude of eigenvectors components (including using arbitrarily small values instead of zeros); phase governed the results.

4.3 Linear Quadratic Regulator (LQR)

Designing a linear quadratic regulator (LQR) involves selecting the symmetric positive semi-definite matrices Q and R in the performance index cost function J to be minimized [9]:

$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)\, dt$$

where x and u are the state and input vectors, superscript T denotes the transpose, and the integration is with respect to time.

Weighting matrices Q and R were selected as identity matrices to minimize all state variable values (i.e. deviation from equilibrium) and inputs [9]. The longitudinal and lateral state spaces were determined to be observable by computing the observability Gramian matrix [17]. This is important to ensure that all states can be weighted.

The gain matrix K is then computed by solving the algebraic Riccati equation, and the resulting closed loop poles/eigenvalues are shown in Figure 13.

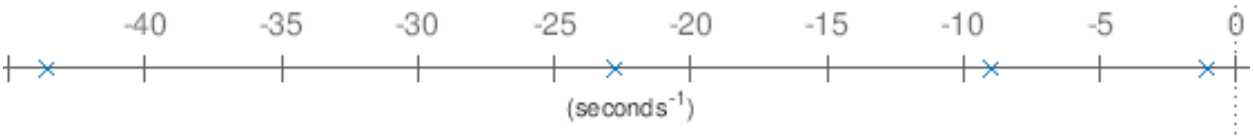


Figure 13: Closed Loop Poles of Linear Quadratic Regulator.

All closed loops poles are high magnitude real numbers; this is expected, as fast exponential decay represents small state "energy" (small departure from equilibrium integrated over time).

4.4 ArduPlane Pitch PID-like Controller

The default ArduPlane PID-like pitch controller (see Figure 14) [13] implemented on the PixHawk was constructed in Simulink for testing with the longitudinal state space. The lateral state space was not used to determine the bank angle (instead, it was set to zero), as the mixer used by the PixHawk is unknown. A mixer is required for a 2-elevon aircraft to reconcile the differences between the demanded elevator and aileron angles.

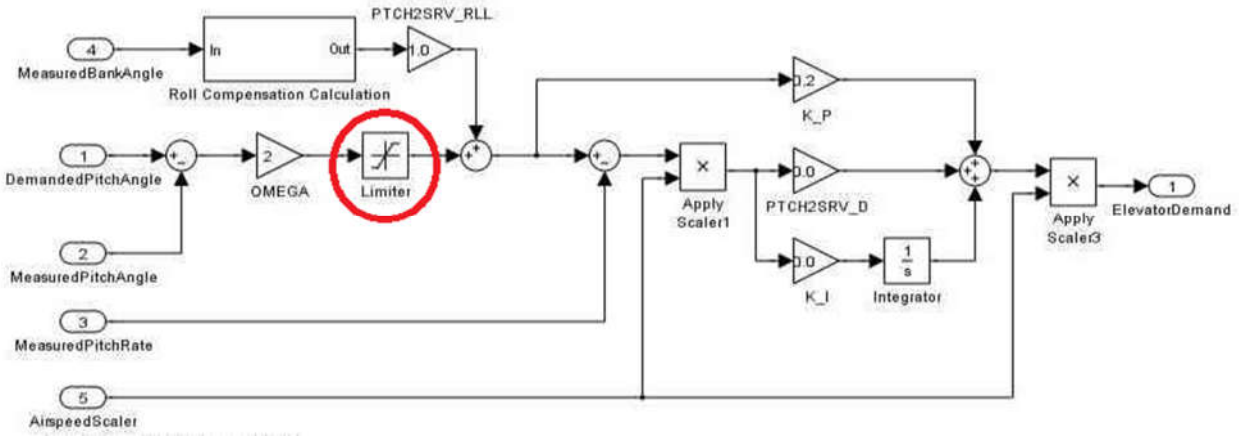


Figure 14: Default ArduPlane PID-like Pitch Controller.

Only the proportional gain ("RLL2SRV_P") was selected to have a non-zero value (0.7) to be conservative (less likely to be unstable), as recommended by ArudPlane documentation for the first test flight, in which the aircraft will be tuned using the autotune feature [18].

For zero integral and derivative gains, a bounded (non-infinite) pitch angle response was only achieved when the pitch rate limiter was set to its smallest allowable value of 1° per second. In this simulated response in MATLAB/SIMULINK, the oscillation of 4° amplitude and period of 18 seconds was essentially undamped simple harmonic motion, for a 0.1 second impulse input of 20° symmetric elevon deflection. Larger pitch rate limits cause the pitch angle response to diverge to infinity.

The input magnitude is important because the rate limiter makes the response non-linear, and possibly introduces disturbances. This may explain why a non-zero integral gain destabilizes the closed loop transfer function, even though a canonical PI (proportional-integral) controller is stable for a linear system without disturbances [14]. However, the ArduPlane PID pitch controller is not a true canonical PID controller found in the literature [14]. Furthermore, there is an error in the roll compensation calculation provided in the documentation [13], which indicates division by the

sine and tangent of the bank angle, even though this causes division by zero in the simplest limiting equilibrium case of zero bank angle. Therefore, the documented controller cannot be fully trusted and should be obtained from the source code instead [19]. However, this default ArduPlane controller is only important for preliminary flight tests before designed controllers are implemented onto the PixHawk's firmware.

Future work should conduct fundamental studies of rate limiters, which appear to be crucial to stabilizing the system, but also introduces nonlinearities and disturbances that could explain the observed instabilities. The ArduPlane pitch controller should be compared to canonical PI and PID controllers [14] in Simulink to obtain the closed loop transfer functions with the system.

5.0 Avionics

The PixHawk 2.1 Cube [11] is the master autonomous flight control hardware chosen for this project. The Cube flight controller is a flexible autopilot system. The controller is designed to use a domain-specific carrier board in order to reduce the wiring and improve reliability. It also offers its own complete flight control stack and project, which allows the flexibility to choose different flight control firmware such as Ardupilot Mega Plane (APM) and the PX4.

The APM [18, 20, 13] is a free open source autopilot firmware that supports different aerial platform configurations. Similarly, the PX4 [21] is another open source autopilot that can control many types of autonomous vehicles. It was originally designed to run on the PixHawk hardware but now runs on other hardware and operating systems. Both types of firmware were utilized in this project. Also, the APM can run on different control hardware and to import features from other firmware such as the PX4. Furthermore, the PX4 has a flexible system that allows it to support any vehicle type, called frame, through a single code base. This added flexibility allows for the development of custom frames for new types of autonomous vehicles.

5.1 Hardware-In-The-Loop

A HITL simulation environment was developed for testing and validating the developed control laws and to perform system identification. In the HITL mode the firmware is run on the PixHawk 2.1 Cube flight controller hardware and a simulation environment encompassing the airplane, the sensors and the atmosphere models communicates with the controller to mimic a real flight scenario. The HITL mode can be run using the PX4 or the Ardupilot firmware [21, 22].

X-Plane [12] was chosen as the flight simulator to run in the HITL simulation mode. It uses blade element theory to calculate the aerodynamic forces on the main surfaces, including propeller blades if present, of an aircraft with any geometric shape. X-Plane includes a program called "PLANE MAKER" that enables the user to create new aircraft configurations that can be flown in the HITL simulation environment. In contrast to other available commercial flight simulators, X-Plane allows for real-time data to be easily exchanged with the connected hardware. Additionally, it incorporates capabilities that are suitable for future UAV simulation and HITL such as markers, navigation, changing weather conditions and air traffic communication. Version 10 of X-Plane was used in this investigation.

The aircraft built in PLANE MAKER was immersed in the X-Plane simulation environment to provide a realistic model for HITL simulations. An aircraft model based on the Skywalker X8 [23] was created using PLANE MAKER with the correct geometry outlines and inertial parameters. The created Skywalker X8 plane simulation is shown in Figure 15.

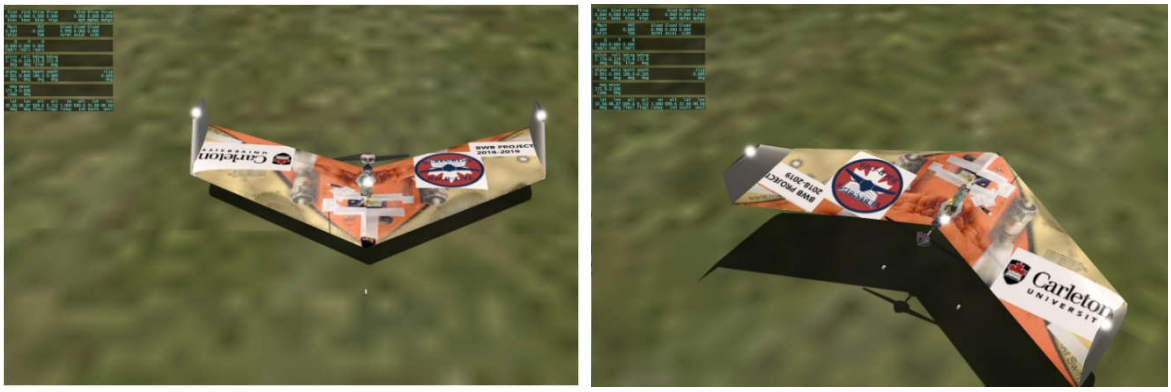


Figure 15:X-Plane Skywalker X8 Model.

Mission Planner [24] and QGroundControl [25] were used as ground control station to receive and display telemetry data, such as aircraft attitude and flight conditions, in real-time of the Skywalker X8 in X-Plane. Waypoint Missions were planned using the HITL simulation to assess the aircraft dynamic and static responses. The complete HITL setup is shown in Figure 16 below.

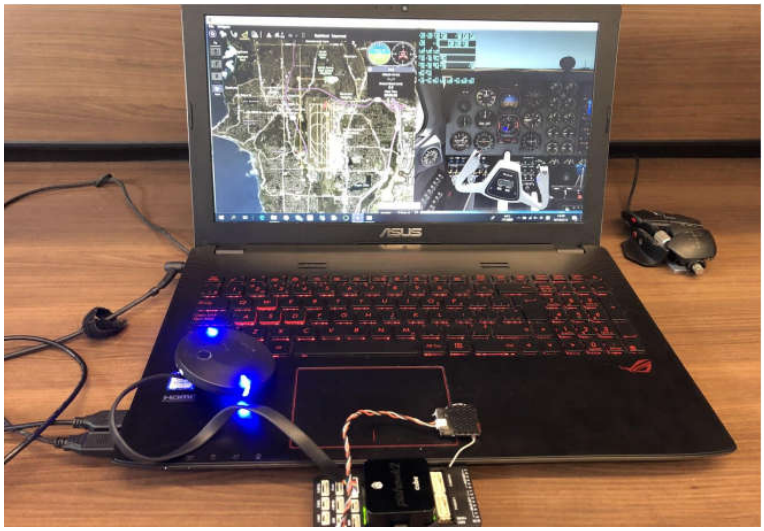


Figure 16: Overall setup of the Hardware-In-The-Loop.

6.0 Flight Tests

Flight tests were conducted using two commercial off-the-shelf UAVs: the Versa Wing and Skywalker X8. The Versa Wing served as a testbed for avionics and pilot training. The Skywalker X8 was used to test the developed dynamic model, as well as the default ArduPlane controller of Figure 14.

6.1 Versa Wing Flight Tests

The flight test configuration is shown in Figure 17. Flight tests were conducted using manual pilot RC (radio control) to train pilots to fly tailless UAVs.

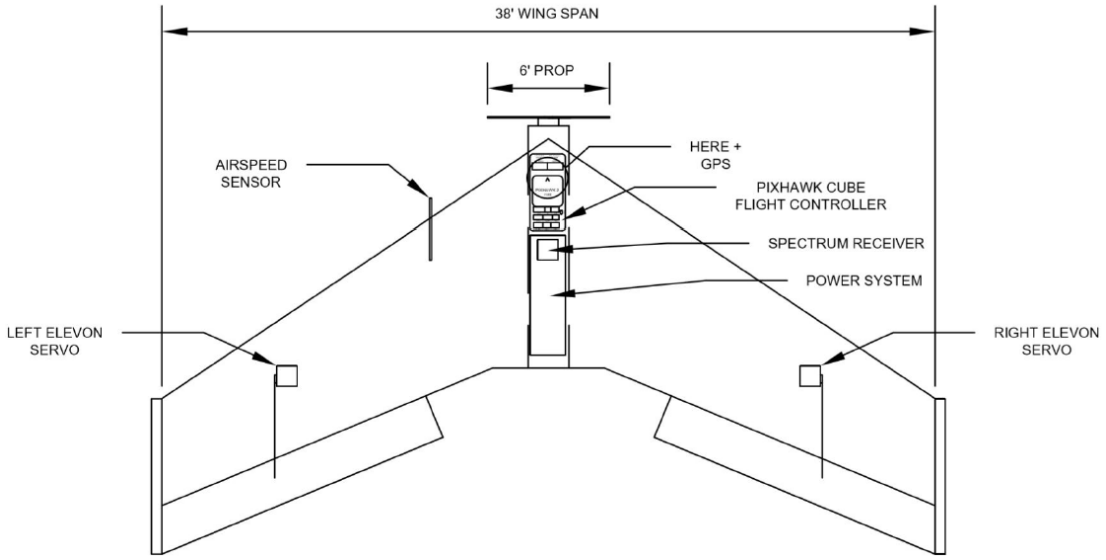


Figure 17: Versa Wing Configuration.

The first flight test crashed due to pilot inexperience with tailless UAVs causing the aircraft to stall. This motivated the use of the fly-by-wire A (FBWA) semi-autonomous flight mode to limit the maximum pitch and roll angles; this FBWA flight mode was tested during the Skywalker X8 flight tests (see Section 6.2).

All avionics components were found to function properly, except for the compass. Although this does not cause the UAV to crash due to the Kalman filter rejecting the compass' erroneous sensor data in favour of GPS data for navigation, redundancy is desirable for safety. It was theorized that the compass failure was caused by close proximity to the battery's magnetic field. Placing the compass further from the battery is not possible because the Versa Wing is designed for visual flight rules (VFR) flight and thus is not designed to accommodate a compass.

Although the 0.2 kg mass of the PixHawk 2.1 Cube may seem negligible, it makes up a significant proportion of the UAV's total weight. It was evident that the Versa Wing is not designed to accommodate additional components given that adding the PixHawk made takeoff and maneuvering difficult, requiring headwind to takeoff. For these reasons, autonomous flight was only tested on the Skywalker X8.

6.2 Skywalker X8 Flight Tests

The flight test configuration is shown in Figure 18. The parameters used for the default ArduPlane PID-like controller of Figure 14 are shown in Table 5. These parameters were obtained from ArduPlane documentation [20] and hardware-in-the-loop simulation (HITL) results, which closely agreed with one another. No rate limiter was used.

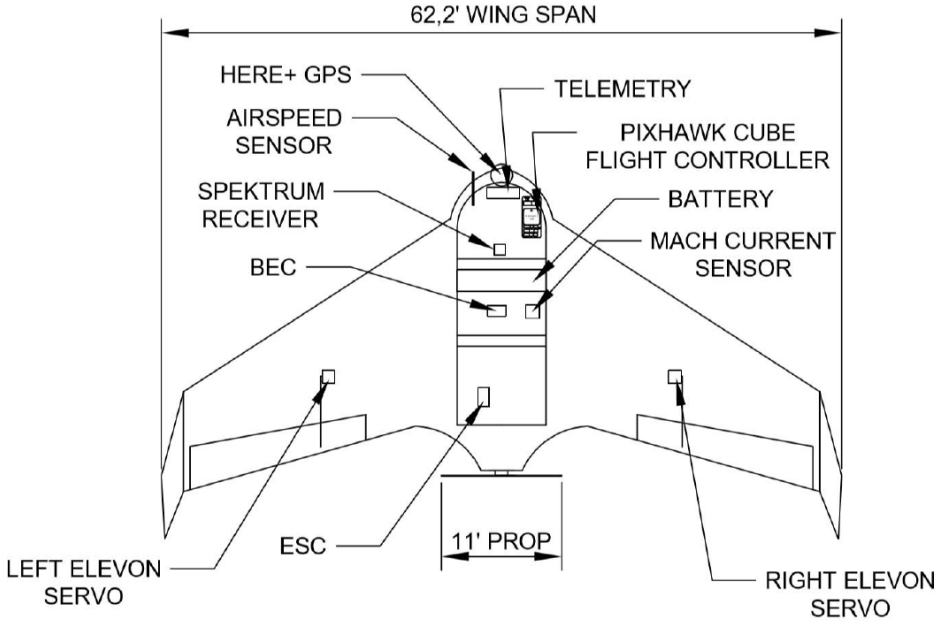


Figure 18: Skywalker X8 Configuration.

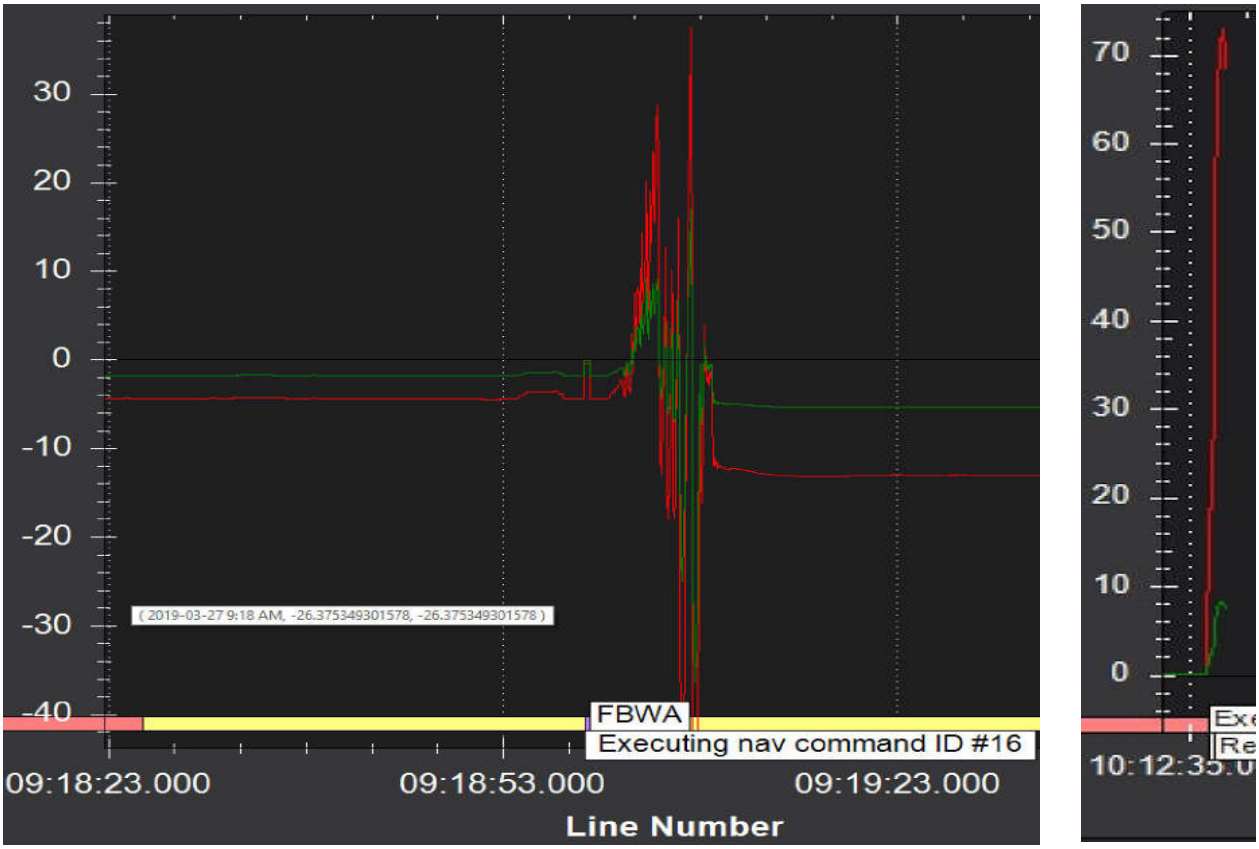
Table 5: Controller Parameters used for Skywalker X8 Test Flights.

Parameter	Value
RLL2SRV_P	0.7
PTCH2SRV_P	0.4
RLL2SRV_D	0
PTCH2SRV_D	0
RLL2SRV_I	0
PTCH2SRV_I	0
NAVL1_PERIOD	19
PTCH2SRV_TCONST	0.4

The first Skywalker X8 flight test used the fly-by-wire A (FBWA) flight mode during takeoff to limit the pitch and roll angles, due to stall encountered during the Versa Wing Test flight (see Section 6.1). The takeoff took place on fairly-smooth snow, with the Skywalker X8's landing gear wheels replaced with skis. Before the rotation phase of takeoff, the nose landing gear passed over an uneven patch of snow and "jumped" upwards, but the pilot remained in control of the aircraft and began rotating the aircraft and climbing. However, during climb, the controller automatically (without pilot input) commanded the aircraft to suddenly pitch down sharply, causing the aircraft to crash and the nose landing gear to fracture upon impact. This pitch down command from the controller was expected to be a result of the PixHawk becoming detached from the fuselage after hitting the uneven patch of snow. This is likely because the FBWA flight mode relies position data from the PixHawk, and the PixHawk was found to be detached from the fuselage after the final crash.

In addition, the flight logs showed an overcurrent when 100% throttle was employed. For this reason, the next and final flight test took off using 80% throttle in manual RC mode, to avoid the issue encountered during the first flight test with the FBWA mode. However, the pilot switched from manual mode to autonomous flight mode (to carry out the mission set by waypoints) too early

during climb, causing the aircraft to strike a soccer goal post. From a human factors perspective, the pilot remarked that switching flight modes was unintuitive as a first response when the plane is about to crash in autonomous flight mode, as the rational immediate reaction is to use the deactivated RC sticks. Furthermore, the pilot remarked that one shortcoming of hardware-in-the-loop simulation is that it is always in first person view with the pilot behind the aircraft, thus not serving as useful preparation for flights in which the UAV approaches the pilot head-on.
As expected, the gains were conservative, as they are only intended for an initial flight to tune using the Autotune flight mode [13]. This is evident as the resultant pitch rate is much smaller in magnitude than the demanded pitch rate, as seen in Figure 19.



a) b)

Figure 19: Desired Pitch Rate (red) vs Actual Pitch Rate (green) in centidegrees/second for a) First Flight Test; b) Final Flight Test.

7.0 Conclusions

In summary, a state space model (symbolic and numeric) was created, then used to calculate the rigid aircraft mode eigenvalues and eigenvectors, and their sensitivity to varied parameters. A state feedback pole placement controller and a linear quadratic regulator (LQR) were designed in MATLAB Simulink. The uncontrolled response and elevon step response with the ArduPlane default pitch controller were simulated in MATLAB Simulink. Hardware-in-the-loop (HITL) simulation in X-Plane was successful in obtaining initial gains for test flights, although these gains were conservative and similar to the developer-recommended values. Flight tests successfully

verified the operation of the avionics, but verifying the gains for the default ArduPlane pitch controller requires further testing.

8.0 Future Work

The state space model could be further refined to include unsteady aerodynamics and propeller effects [26] [27]. Unsteady aerodynamics primarily affects the tail due to the lag of downwash reaching the tail at low speeds. However, the slowly-convecting (due to slow flight speed) shed wingtip vortices also induce velocity on the wing by the Biot-Savart Law [28]. The unsteady aerodynamic models of the Kussner and Wagner functions are recommended as low speed alternatives to Theodoresen's function [29] [30].

Refined values for lateral aerodynamic derivatives can be obtained via computational fluid dynamics (CFD) and/or wind tunnel testing. Refined inertia tensor values can be obtained from CAD software such as CATIA.

Since there are only 2 elevons, longitudinal and lateral control is coupled. Therefore, the aircraft model could be improved to couple longitudinal and lateral motions, enabling the inclusion of gyroscopic effects, which have inherent longitudinal-lateral coupling [10]. Only rigid aircraft modes were analyzed; the aircraft model should be refined to include structural flexibility considerations.

The unintuitive destabilizing effects of the default ArduPlane PixHawk controller should be investigated, as outlined in Section 4.4. The reason for the overcurrent observed during the Skywalker X8 flight tests when 100% throttle was commanded must also be investigated; damage to the motor or electronic speed controller (ESC) are suspected.

Future HITL or Software-In-The-Loop (SITL) simulations and flight tests should use the designed controllers (pole placement, LQR), in addition to future Lyapunov-based and H-infinity controllers to be designed. Presently, HITL simulations using X-Plane are difficult to compare with the Simulink response (even with validation by comparing responses), since X-Plane does not use a state space, but blade element theory. Furthermore, the ArduPlane firmware on the PixHawk uses state estimators (an extended Kalman filter) instead of a state space model. Therefore, to achieve more valid comparisons between Simulink, HITL, and flight tests, it is recommended that future HITL simulation be set up through MATLAB Simulink FlightGear and SimScape, as these simulations would directly use the state space model [31]. The flight test data can be used for system identification to validate and refine the state space model.

Acknowledgments

The authors would like to acknowledge Bombardier Aerospace and the Carleton University Engineering Student Equipment Fund (CUSEF) for their funding and mentorship.

References

[1] National Aeronautics and Space Administration, "NASA - The X-48B Blended Wing Body," National Aeronautics and Space Administration, 2013. [Online]. Available: https://www.nasa.gov/topics/aeronautics/features/bwb_main.html. [Accessed 24 May 2019].

[2] M. Kozek and A. Schirrer, *Modeling and Control for a Blended Wing Body Aircraft*, New York: Springer, 2014.

[3] R. H. Liebeck, M. A. Page and B. K. Rawdon, "Blended-wing-body subsonic commercial transport," in *36th AIAA Aerospace Sciences Meeting & Exhibit*, 1998.

[4] M. Saucez, "The Handling Qualities Resolution of a Flying Wing Configuration," Ph.D Thesis, University of Toulouse, Toulouse, France, 2013.

[5] Y. Deniuel, "Preliminary Design of Control Surfaces and Laws for Unconventional Aircraft Configurations," Ph.D Thesis, University of Toulouse, Toulouse, France, 2016.

[6] B.-M. Min, S.-S. Shin, H.-C. Shim and M.-J. Tahk, "Modeling and Autopilot Design of Blended Wing-Body UAV," *International Journal of Aeronautical and Space Sciences,* vol. 9, no. 1, pp. 121-128, 2008.

[7] K. Gryte, "High Angle of Attack Landing of an Unmanned Aerial Vehicle," M.Sc Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2015.

[8] R. M. Farhadi, V. Kortunov, A. Molchanov and T. Solianyk, "Estimation of the Lateral Aerodynamic Coefficients for Skywalker X8 Flying Wing from Real Flight-Test Data," *Acta Polytechnica,* vol. 58, no. 2, pp. 77-91, 2018.

[9] B. L. Stevens, F. L. Lewis and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Wiley, 2015.

[10] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*, 3rd ed., Wiley, 1996.

[11] Project, PixHawk Hardware, 2019. [Online]. Available: http://www.proficnc.com/content/13-PixHawk2.

[12] X-Plane, "X-Plane SDK Documentation," X-Plane, 2019. [Online]. Available: https://developer.x-plane.com/sdk/. [Accessed 30 March 2019].

[13] ArduPilot Dev Team, "Roll, Pitch and Yaw Controller Tuning," ArduPilot, 2019. [Online]. Available: http://ardupilot.org/plane/docs/roll-pitch-controller-tuning.html. [Accessed 28 March 2019].

[14] N. S. Nise, *Control Systems Engineering*, 7th ed., John Wiley & Sons, 2015.

[15] J. Kautsky, N. N. K and P. Van Dooren, "Robust Pole Assignment in Linear State Feedback," *International Journal of Control,* vol. 41, pp. 1129-1155, 1985.

[16] MathWorks Inc., "Pole Placement Design - MathWorks," MathWorks Inc., 2019. [Online]. Available: https://www.mathworks.com/help/control/ref/place.html. [Accessed 31 March 2019].

[17] T. Kailath, *Linear Systems*, Englewood Cliffs, New Jersey: Prentice-Hall, 1980.

[18] ArduPilot Dev Team, "Complete Parameter List," ArduPilot , 2019. [Online]. Available: http://ardupilot.org/plane/docs/parameters.html#ptch2srv-parameters. [Accessed 31 March 2019].

[19] MathWorks Pilot Engineering Group, "PixHawk Pilot Support Package (PSP) User Guide Version 3.04," 2018.

[20] ArduPilot Dev Team, "Configuration Values for Common Airframes," ArduPilot, 2019. [Online]. Available: http://ardupilot.org/plane/docs/configuration-files-for-common-airframes.html#skywalker-x8. [Accessed 27 May 2019].

[21] Dronecode Project Inc.,, "PX4 Autopilot," 2019, [Online]. Available: https://px4.io/.

[22] ArduPilot Dev Team, "HITL Simulators," 2019. [Online]. Available: http://ardupilot.org/dev/docs/hitl-simulators.html.

[23] Skywalker Technology Co.,Ltd, 2015. [Online]. Available: http://skywalkermodel.com/en/.

[24] ArduPilot Dev Team, "Mission Planner Home," 2019. [Online]. Available: http://ardupilot.org/planner/.

[25] Drone Control, "QGROUNDCONTROL," 2019. [Online]. Available: http://qgroundcontrol.com/.

[26] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*, Princeton University Press, 2012.

[27] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, 3rd ed., American Institute of Aeronautics and Astronautics, 2012.

[28] A. M. Kuethe and C. Y. Chow, *Foundations of Aerodynamics*, 5th ed., Wiley, 1997.

[29] T. Cebeci, M. Platzer, H. Chen, K.-C. Chang and J. P. Shao, *Analysis of low-speed unsteady airfoil flows*, Springer Berlin Heidelberg, 2005.

[30] R. L. Bisplinghoff, H. Ashley and R. L. Halfman, *Aeroelasticity*, 1st ed., Mineola, New York: Dover Publications, 1955.

[31] MathWorks Inc., "Hardware-in-the-Loop Simulation with Simulink Real-Time," MathWorks Inc., 2019. [Online]. Available: https://www.mathworks.com/help/physmod/simscape/hardware-in-the-loop-simulation-with-simulink-real-time.html. [Accessed 28 March 2019].