

# **Robust control system development for VTOL-to-fixed wing flight transition with the EcoSoar UAV**

A thesis in Automatic Control

**Robert Hedman**

November 6, 2019



A thesis presented for the degree of  
Masters of Science / Civil Engineer

Department of Computer Science, Electrical and Space Engineering

Luleå University of Technology

*Supervisors:* K. ATTA, K. KOCHERSBERGER

*Examinator:* G. NIKOLAKOPOULOS

## Abstract

A non linear quaternion based attitude controller of type P2 controller, together with a sensitivity normalizing function for the control surfaces has been simulated and implemented on a flying fixed wing with non vectored engines. In simulations the controller worked well in all flight modes, hovering, transition and flying, and also rejected a simple wind disturbance in all modes. The first implementation on hardware did not work due to programming errors causing crashes with unrepairable damages. The second aircraft was built out of a piece of plywood to further simplify the testing and tolerate more crashes. A non-existing airfoil, flat plate, has no effects due to camber and is therefor easier to both simulate and tune. The controller worked acceptable in reality, but does need further tuning. Due to time constraints the weighing of airflow inside and outside the propeller wash could not be fully determined resulting in different gain in the different flight modes, but initial estimation of the parameters were enough to achieve robust, stable hovering transitioning and flying even in quite strong wind (more than 5 m/s). The controller was never implemented on a EcoSoar due to time constraints, but the proof of concept flying piece of plywood proved the controller feasible for future embedding in a modified EcoSoar.

## ***Acknowledgements***

Thank teachers that helped, Mr. K, Avory, Wife, Brianna, etc.

[Lets see if they require rent before adding below]

I also want to extend a very warm thank you to Montford Special Needs College in Malawi for allowing me to stay there together with my wife during this thesis. The staff also provided all the support needed to become integrated with the local culture and helping out with practical issues such as getting a car, finding the hospital and providing a safe circle of friends.

# Contents

<b>1</b>	<b>List of variables</b>	<b>5</b>
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	EcoSoar hardware . . . . .	6
<b>3</b>	<b>Problem Forumulation</b>	<b>6</b>
<b>4</b>	<b>Theory</b>	<b>6</b>
4.1	State Space . . . . .	6
4.2	State estimation/observers . . . . .	7
4.3	Reference Frames . . . . .	7
4.4	Quaternions . . . . .	7
4.5	Basic aerodynamics . . . . .	8
4.5.1	Propeller lift . . . . .	8
<b>5</b>	<b>Mathematical Model of EcoSoar</b>	<b>8</b>
5.1	Assumptions . . . . .	8
5.2	Aircraft force diagram . . . . .	9
5.2.1	Aerodynamic Lift force . . . . .	9
5.2.2	Aerodynamic drag . . . . .	10
5.2.3	Lift and drag in body frame . . . . .	11
5.3	Moment due to force . . . . .	11
5.3.1	Pitching moment . . . . .	12
5.4	Equations of motion . . . . .	12
5.4.1	Attitude . . . . .	12
5.4.2	Gravity . . . . .	12
5.4.3	Forces and moments due to accelerated body reference frame . . . . .	12
5.4.4	elevons . . . . .	13
5.5	Winglets, forces and moments due to sideslip . . . . .	13
5.5.1	Restoring moment . . . . .	14
5.5.2	Non linear Forces and Moments . . . . .	14
5.5.3	Rolling moment due to sideslip . . . . .	14
5.6	Propeller Thrust . . . . .	14
5.7	Restoring moment due to roll rate . . . . .	15
5.8	Restoring moment due to yaw rate . . . . .	16
5.9	Rolling moment due to yaw rate . . . . .	16
5.10	Restoring moment due to pitch rate . . . . .	16
5.11	Actuator dynamics . . . . .	17
5.11.1	Elevon dynamics . . . . .	17
5.11.2	Motor dynamics . . . . .	17
5.12	Full equations of motion . . . . .	17
<b>6</b>	<b>EcoSoar replacement</b>	<b>18</b>
6.1	Parameter adjustments . . . . .	18
6.2	Building . . . . .	18
6.3	Parameter estimation/declarations . . . . .	19
<b>7</b>	<b>Robust controller development and Implementation</b>	<b>19</b>
7.1	Feedback Linearization . . . . .	19
7.2	Simulink . . . . .	20
7.3	Pixhawk firmware . . . . .	20
7.4	Initial testing on Bixler . . . . .	20
7.5	P2-controller . . . . .	21
7.6	Torque translator . . . . .	21
7.6.1	Finding $k_2$ and $k_4$ . . . . .	22

7.6.2	Finding $k_1$ and $k_3$ : Single Engine Board . . . . .	22
7.7	Full controller diagram . . . . .	22
<b>8</b>	<b>Results</b>	<b>23</b>
8.1	Performance in Simulations . . . . .	23
8.2	Performance of the flying plywood board in reality . . . . .	25
<b>9</b>	<b>Conclusion</b>	<b>25</b>

# 1 List of variables

$L_L$	Lift from left wing
$L_R$	Lift from right wing
$D_L$	Drag from left wing
$D_R$	Drag from right wing
$F_L$	Lift due to aileron deflection, left side, outside propeller wake
$F_{LT}$	Lift due to aileron deflection, left side, inside propeller wake
$F_R$	Lift due to aileron deflection, right side, outside propeller wake
$F_{RT}$	Lift due to aileron deflection, right side, inside propeller wake
$D_{\alpha L}$	Drag due to aileron deflection, left side, outside propeller wake
$D_{\alpha L,T}$	Drag due to aileron deflection, left side, inside propeller wake
$D_{\alpha R}$	Drag due to aileron deflection, right side, outside propeller wake
$D_{\alpha R,T}$	Drag due to aileron deflection, right side, inside propeller wake
$L_{WL}$	Lift from left winglet
$L_{WR}$	Lift from right winglet
$D_{WL}$	Drag from left winglet
$D_{WR}$	Drag from right winglet
$S_a$	Surface area of one aileron, parts outside propeller wake
$S_{aw}$	Surface area of one aileron, area inside propeller wake
c.g.	center of gravity (point)
$T_L$	Thrust from left propeller
$T_R$	Thrust from right propeller
$S_w$	Area of one wing, area outside of propeller wake
$S_{ww}$	Area of one wing, area inside propeller wake
$S_W$	Area of one winglet
$x_f$	distance in $x$ from c.g. to aileron force
$x_T$	distance in $x$ from c.g. to thrust force
$x_{ac}$	distance in $x$ from c.g. to wing aerodynamic center
$x_W$	distance in $x$ from c.g. to winglet force
$y_T$	distance in $y$ from c.g. to thrust force
$y_f$	distance in $y$ from c.g. to aileron force
$y_{ac}$	distance in $y$ from c.g. to wing aerodynamic center
$y_w$	width of propeller wake
$b$	wingspan
$z_w$	distance in $z$ from c.g. to winglet force

## 2 Background

Virginia Tech, a University in Virginia, USA, has developed a fixed wing drone called the EcoSoar; a flying wing. It is an aircraft designed for fabrication, operation and maintenance in low- resource environments. The total cost of one EcoSoar is only around 350 USD when costs for all tools and materials have been accounted for and spread over ten aircraft.

EcoSoar has been successfully flown in Malawi at the test corridor established by UNICEF in Kasungu, where a simulated payload of dried blood spot samples (used for HIV testing) was delivered 19 km from a remote health clinic to the Kasungu airport. The aircraft is constructed of poster-board and 3D printed parts, making it easy to source in Malawi and low cost to repair if damaged.

Many flight operations occurring in Africa are converging on hybrid aircraft designs where the aircraft functions as a quadcopter in takeoff and landing but is capable of higher speed flight in a fixed wing configuration.

EcoSoar can currently be launched with a big bungee-cord or with a custom built launcher. It thus needs a field to both take off and land in since it can only land on its belly similar to that of any normal aircraft. Modifying the EcoSoar to have two engines, one on the front of each wing, instead of the current design of having one at the back, might allow it to hover, take off and land vertically eliminating the need for a launching system as well as the requirement of a big field for take-off and landing. It also eliminates the risk of high speed crashes at launch due to a badly calibrated aircraft.

Adding VTOL capacity to this drone would greatly increase its range of possible missions and thus help humanitarian efforts centered around EcoSoar in Malawi.

To achieve VTOL not only is hardware modifications necessary but also a new control scheme is necessary.

### 2.1 EcoSoar hardware

The EcoSoar frame is mostly built out of 3D printed parts for the body and spars. The outer layer of the wings are made out of poster board and packing tape. Small micro servos actuate on the Elevons for control. The on board electronics include a flight computer (Pixhawk 1), any compatible receiver, an ESC (Electronic Speed Controller) for the brushless motor powering the propeller, a telemetry module with antenna and a gps module.

## 3 Problem Forumulation

This thesis will attempt to solve the problem of mathematically modeling a flying fixed wing, develop a VTOL capable model based controller that can handle both hovering, transitioning into flying and flying, and finally implement the controller on the Pixhawk controller onboard the aircraft.

## 4 Theory

To develop this controller some basic concepts are required.

### 4.1 State Space

Many systems can be written as a set of differential equation.

Physical systems may sometimes be mathematically modeled by state space representation, given that it can be written on the same form as equation 1 below.

$$\dot{\bar{x}} = f(\bar{x}) + g(\bar{u}, \bar{x}) \quad (1)$$

The states,  $\bar{x}$  are related by a set of first order differential equations and some input.

Systems can have one input and one state/output, called Single Input Single Output (SISO), or have either multiple input or outputs or both; Multiple Input Multiple Output (MIMO) systems.

A system with several states, i.e.  $\bar{x}$  being a vector, and several inputs, i.e.  $\bar{u}$  being a vector is thus represented by a set of functions  $\bar{f}$  and  $\bar{g}$ .

From basic fundamental physical laws, e.g. newtons laws, a set of differential equation for many physical systems can be derived. Once the state space equations are known simulating the system is only a matter of choosing a differential equation solver. Simulink has built in solver for these equations over time making it easy to simulate the system once the equations are known.

## 4.2 State estimation/observers

A full state vector may not always be exactly measurable so estimating it is of importance. The Pixhawk PX4 firmware already includes an extended Kalman filter for estimating the position and attitude from an angular rate sensor, accelerometer, barometer and magnetometer. For this project the built in Kalman filter will be used to estimate the state with which the control system will act upon.

## 4.3 Reference Frames

When working with rigid bodies in 3d space a reference system is needed. In most of airplane dynamics for small scale aircraft the standard is a Cartesian coordinate system with the x-axis aligned with North, y-axis aligned with East and the z-axis pointing down. The body frame is similar, x-axis pointing forwards, y-axis along right wing and z-axis through the belly of the aircraft, i.e. down when flying normally. This correspond to the well used Roll-Pitch-Yaw system.[3]

## 4.4 Quaternions

Euler angles are intuitive but suffer from singularities. Since normal aircraft normally do not spend time pointing straight up these singularities do not matter too much, but a VTOL capable tail-sitter will, so using quaternions is beneficial.

A quaternion,  $\mathbf{q}$  is a four dimensional hyper complex number.[4]

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \quad (2)$$

where  $i^2 = j^2 = k^2 = ijk = -1$ .[7] The imaginary part can be referenced to as the vector part and  $q_0$  the scalar part. The conjugate of a quaternion is defined as

$$\mathbf{q}^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k} \quad (3)$$

The quaternion can also be represented as a vector:

$$\mathbf{q} = [q_0 \quad q_1 \quad q_2 \quad q_3] \quad (4)$$

The norm of a quaternion is defined by

$$\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (5)$$

If the norm of a quaternion is equal to 1 it is a unit quaternion. Unit quaternions are very computationally efficient at describing rotations in 3D. They also do not suffer from any singularities like the common Roll-Pitch-Yaw / Euler angles system. Unit quaternions can always be written as

$$\mathbf{q} = \begin{bmatrix} \cos\theta \\ \mathbf{u}\sin\theta \end{bmatrix} \quad (6)$$

where  $\mathbf{u}$  is the vector  $[q_1 \quad q_2 \quad q_3]$  describing the axis of which to rotate around and  $\theta$  is proportional to the amount of rotation about that axis. Multiplying unit quaternions yield a unit quaternion describing the joint rotation of the two initial quaternions. The conjugate of a quaternion gives the opposite rotation direction around the same vector. This can correspond to going from a world frame to body frame, or vice versa. Multiplying a reference quaternion describing a set orientation for an object in 3D with the conjugate of its actual quaternion therefore gives the error in orientation, attitude.

A more in depth description of quaternions and their mathematics please see [7].

## 4.5 Basic aerodynamics

An object moving through a medium displaces mass in the medium it travels through. A wing moving through the air will displace air particles. If the total movement of the particles generate a net momentum a force will be generated on the object displacing them, in this case the wing. The air will thus exert pressure on the wing and it is useful to consider the pressure distribution. By integrating the pressure distribution over the entire surface of the wing one can find the center of pressure.[8] If the center of pressure and center of gravity are not aligned a torque will also be generated. The equivalent force and moment can be placed at the center of pressure to simplify calculations. The aerodynamic force can be split into two parts, one in line with the incoming airflow and one perpendicular to it. These are usually denoted Lift and Drag forces. The lift force can further be broken down into several parts, but aerodynamics is a complicated field, and for the purposes of this thesis we only consider the following: Lift due to camber (the non symmetry of the wing in the airstream), lift due to angle of attack (the angle which the incoming air makes with the wing) and lift due to control surface deflection.

### 4.5.1 Propeller lift

The force from propellers will be dealt with in greater detail further into this thesis. But as an introduction they can be thought of as wings moving through the air, only in a rotational matter instead of straight on. As they move through the air lift is generated in a similar fashion to normal wings, but since the propeller is mounted at a right angle to the wings the lift force propels the aircraft forward. The air a propeller displaces is called a wake and will be treated as a separate airflow from the aircraft moving through the air.

## 5 Mathematical Model of EcoSoar

To develop a control system first a simulator needs to be developed, and so the aircraft has to be mathematically modeled. To not make the simulator too time consuming to develop nor to complex a few assumptions will be made:

### 5.1 Assumptions

- There is no external wind and the freestream pressure does not change near the aircraft; and so  $v_\infty$  is replaced with  $v$  which is the aircraft's velocity through the air.
- All forces operate on points in the  $x - y$  plane, i.e. no offset in the  $z$ -direction (with the notable exception of the winglet forces). Forces may still have a  $z$ -component though.
- The propeller wake is unaffected by aircraft velocity, i.e. only determined by propeller speed.
- Airflow outside of wake is only determined by aircraft velocity.
- The aerodynamic centre (a.c.) is static and does not move with changes in speed and pressure. The moment around it, however, will vary and thus account for the effects of the a.c. movement.
- Airflow speed in wake is proportional to propeller rotational speed,  $v_w = b_w \omega_i$ , where  $b_w$  is a constant.
- The principal moments axis align with the symmetry plane; only diagonal elements in the inertia matrix aligned with the body frame coordinate axis.
- Roll moment due to sideslip angle, negligible due to lack of tail with vertical offset from c.g.
- Yawing moment due to roll rate is negligible due to the lack of a tail.
- Forces and moments due to acceleration are not explicitly calculated; they are already included in dynamic model since it is based on forces.

## 5.2 Aircraft force diagram

The aircraft in question is the EcoSoar; a flying wing. It has two control surfaces, one on each wing called elevons. In front of each wing a motor and propeller is mounted providing thrust. The motors also supply an airflow over the control surfaces when the aircraft is not moving through the air.

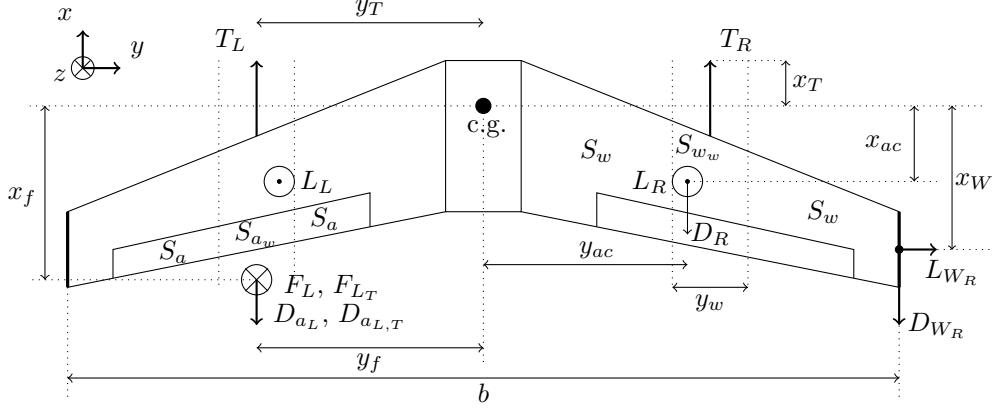


Figure 1: Drawing of the EcoSoar, its propellers with wake and its control surfaces in body frame.

The elevon forces,  $F_L$ ,  $F_{L_T}$ ,  $D_{aL}$  and  $D_{aL,T}$ , have only been marked on the left wing, but exist symmetrically on the right wing with index  $R$  instead of  $L$ . The same is true for the drag force,  $D_R$  and the winglet forces,  $L_{W_R}$  and  $D_{W_R}$ , existing symmetrically on the left side.  $F_L$  is the force due to elevon deflection over surface area  $S_a$ , and  $F_{L_T}$  is the force due to elevon deflection over  $S_{aw}$ ; the surface in the wake of the propeller. Similarly for the drag,  $D_{aL}$ ,  $D_{aL,T}$ , forces. We also split the wing area into two sections:  $S_w$  for the wing area outside the wake, and  $S_{ww}$  for the wing area in the wake. The thrust forces,  $T_i$ , are assumed to act at the arrow end of the vector due to the offset in  $x$  created by the motor and drive shaft. The lift forces,  $L_L$  and  $L_R$ , are functions of airflow over wing outside of wake and in the wake, which in turn are functions of aircraft velocity, propeller speed, angle of attack, air density, etc.

### 5.2.1 Aerodynamic Lift force

The lift force of a wing depends on several factors. Mainly: geometrical shape, dynamic pressure and angle of attack.[8] The bigger the wing the more lift. The faster air flows the more lift. The further away from hitting the wing straight on, usually more lift.

Important note: Lift is defined as the **Aerodynamic force component perpendicular to the incoming airflow**. This means that, for example, the lift force labeled in figure 16 as  $L_L$  is not the aerodynamic lift but rather the lift force in body frame. The same argument applies to the drag forces.

Aerodynamic lift is usually modeled as[3][8]:

$$L = QC_L S \quad (7)$$

where  $S$  is a reference surface area of the wing and  $Q$  is the dynamic pressure:

$$Q = \frac{1}{2}\rho v^2 \quad (8)$$

$\rho$  is the density of air and  $v$  the velocity of the air hitting the wing.  $C_L$  is the coefficient of lift, a function of the angle of attack. This coefficient is not linear. Empirical studies[5][6] have found functions similar to the one in figure 2.

The angle of attack,  $\alpha$ , is formally the angle at which the air is hitting the wing relative the body frame x-z axis. If air is aligned with the  $x$ -axis  $\alpha$  is zero, and if aligned with the  $z$ -axis  $\alpha$

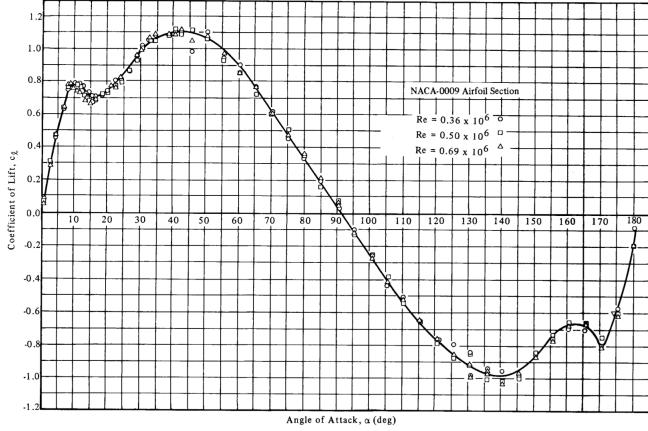


Figure 2: Full Range Section Lift Coefficients for the NACA-0009 Airfoil at Reynolds Numbers of  $0.36 \times 10^6$ ,  $0.50 \times 10^6$ , and  $0.69 \times 10^6$

Figure 2: Coefficient of lift for a symmetrical wing vs angle of attack for the entire 180 degree rotation of the wing. The function is periodic.

is 90. (The airflow will obviously be in the negative axis direction since the airplane is flying in the positive direction and hitting the air). Thus, if we assume no wind the angle of attack is only dependent on the aircraft velocity  $\bar{v}$ . Normally this is modeled as:

$$\alpha = \tan^{-1}\left(\frac{v_z}{v_x}\right) \quad (9)$$

but in our case the velocity may have any direction in the  $x$ - $z$  plane, and is not limited to the right half plane. The polar coordinates for vector  $\bar{v}$  in the figure below, figure 3, are:

$$\begin{aligned} v_x &= r \cos(\alpha) \\ v_y &= r \sin(\alpha) \\ r &= \sqrt{v_x^2 + v_z^2} \end{aligned} \quad (10)$$

where  $\alpha$  is in the interval  $[-\pi, \pi]$ .

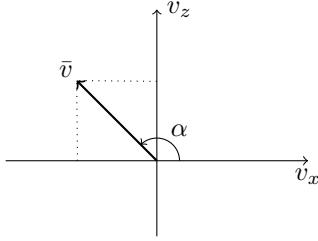


Figure 3: Velocity vector  $\bar{v}$  in polar coordinates.

### 5.2.2 Aerodynamic drag

Similarly to aerodynamic lift the wing will also have an aerodynamic drag force, but parallel to the incoming flow and not perpendicular as lift. The drag is also modeled like so:

$$D = Q C_D S \quad (11)$$

where the coefficient of drag,  $C_D$ , can also be experimentally obtained as in figure 4.

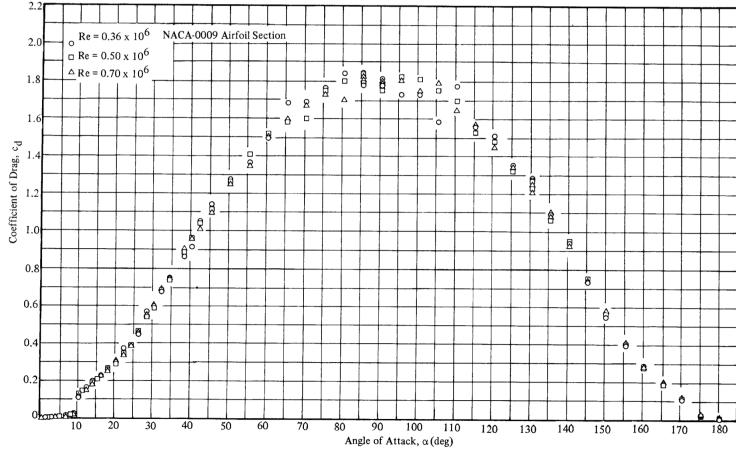


Figure 18. Full Range Section Drag Coefficients for the NACA-0009 Airfoil at Reynolds Numbers of  $0.36 \times 10^6$ ,  $0.50 \times 10^6$ , and  $0.69 \times 10^6$

Figure 4: Coefficient of drag for a symmetrical wing vs angle of attack for the entire 180 degree rotation of the wing.

### 5.2.3 Lift and drag in body frame

Since the aerodynamic lift and drag forces are relative the incoming airflow, i.e. functions of the angle of attack,  $\alpha$ , the body frame Lift and Drag have to be transformed:

$$\begin{aligned} L_i &= \cos(\alpha)L + \sin(\alpha)D \\ D_i &= -\sin(\alpha)L + \cos(\alpha)D \end{aligned} \quad (12)$$

Note that the above forces are in the direction labeled in figure 16 and not necessarily in the positive axis they act in. From equation 10 we see that

$$\begin{aligned} v_x &= r \cos(\alpha) \Rightarrow \cos(\alpha) = \frac{v_x}{\sqrt{v_x^2 + v_z^2}} \\ v_z &= r \sin(\alpha) \Rightarrow \sin(\alpha) = \frac{v_z}{\sqrt{v_x^2 + v_z^2}} \end{aligned} \quad (13)$$

If we substitute in expressions for aerodynamic lift and drag we obtain:

$$\begin{aligned} L_i &= \frac{v_x}{\sqrt{v_x^2 + v_z^2}} QC_L S + \frac{v_z}{\sqrt{v_x^2 + v_z^2}} QC_D S \\ D_i &= -\frac{v_z}{\sqrt{v_x^2 + v_z^2}} QC_L S + \frac{v_x}{\sqrt{v_x^2 + v_z^2}} QC_D S \end{aligned} \quad (14)$$

which simplify into:

$$\begin{aligned} L_i &= S \frac{1}{2} \rho v^2 \left( \frac{C_L v_x + C_D v_z}{\sqrt{v_x^2 + v_z^2}} \right) \\ D_i &= S \frac{1}{2} \rho v^2 \left( \frac{-C_L v_z + C_D v_x}{\sqrt{v_x^2 + v_z^2}} \right) \end{aligned} \quad (15)$$

where  $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ .

### 5.3 Moment due to force

Any force not acting at the center of gravity will produce a moment around c.g. The moment of a force is:

$$\bar{M} = \bar{r} \times \bar{F} \quad (16)$$

With our assumptions the above equation can be written as:

$$\bar{M} = (r_x, r_y, 0) \times (F_x, F_y, F_z) = \begin{bmatrix} F_z r_y \\ -F_z r_x \\ F_y r_x - F_x r_y \end{bmatrix} \quad (17)$$

### 5.3.1 Pitching moment

Lift due to camber on a wing acts at 50% of the cord line. Lift due to angle of attack acts at roughly 25% of the cord line.[8] This makes the force acting point, center of pressure, move along the cord line when angle of attack and aircraft velocity changes. A common simplification is to choose the aerodynamic center, a.c., as the point at which Lift acts. It can be shown that placing the a.c. at the 25% cord position makes the moment generated vary little as angle of attack changes, yielding simpler equations. The pitching moment due to lift can now be expressed as:

$$\tau_{y, lift} = \sum_{j \in [w, w_w]} C_{m,y,l} Q S_j l + \sum_{j \in [w, w_w]} \cancel{C_{m,y,t,\alpha} Q S_j l \alpha}^0 \quad (18)$$

where  $l$  is the characteristic length of the wing, usually taken as the mean cord for pitching moments and the wingspan for rolling and yawing moments.

## 5.4 Equations of motion

The equations of motion can be split into three parts: one part due to gravity, one part due to the rigid body frame being an accelerated reference frame, and one part due to aerodynamic forces. These forces depend on the aircraft's attitude and will be expressed in terms of it.

### 5.4.1 Attitude

The attitude is represented by a quaternion:

$$\bar{q} = [q_0, q_1, q_2, q_3]^T \quad (19)$$

Any vector,  $\bar{r}_w$ , in world frame can be represented in the body frame by the conversion:

$$\bar{r}_b = \bar{q} \bar{r}_w \bar{q}^* \quad (20)$$

### 5.4.2 Gravity

In the case of gravity,  $\begin{bmatrix} 0 \\ mg \end{bmatrix}_w$ , the force,  $m\bar{g}$ , becomes

$$m\dot{\bar{v}}_b = \begin{bmatrix} 2(\bar{q}_1\bar{q}_3 + \bar{q}_0\bar{q}_2) \\ 2(\bar{q}_2\bar{q}_3 - \bar{q}_0\bar{q}_1) \\ (\bar{q}_0^2 - \bar{q}_1^2 - \bar{q}_2^2 + \bar{q}_3^2) \end{bmatrix} mg \quad (21)$$

Gravity acts in the center of gravity and creates no moment. Since gravity is in world frame and we want to express it in body frame we need to conjugate the quaternion.

### 5.4.3 Forces and moments due to accelerated body reference frame

In the case of a rigid body with inertia matrix  $I$ , velocity  $\bar{v}$  and angular velocity  $\bar{\omega}$  relative a fixed frame we have the moment equation:

$$I\dot{\bar{\omega}} = \bar{\tau} - \bar{\omega} \times I\bar{\omega} \quad (22)$$

where  $\tau$  is the external torque acting on the body.

Similarly, if we include the coriolis force we obtain the forces in our body frame:

$$\bar{F} = \bar{F}_e + 2m\bar{\omega} \times \bar{v} \quad (23)$$

where  $\bar{F}_e$  is the external forces acting on the body. No other inertial forces are of interest; the linear acceleration is irrelevant since we are not flying inside a linearly accelerating frame, and the other two depend on rotation around a non principal axis.

In our case, with assumptions made, the above equations simplify to:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = - \begin{bmatrix} 2(v_z\omega_y - v_y\omega_z) \\ 2(v_x\omega_z - v_z\omega_x) \\ 2(v_y\omega_x - v_x\omega_y) \\ \frac{1}{I_x}(I_z - I_y)\omega_y\omega_z \\ \frac{1}{I_y}(I_x - I_z)\omega_x\omega_z \\ \frac{1}{I_z}(I_y - I_x)\omega_x\omega_y \end{bmatrix} + \text{external forces and moments} \quad (24)$$

#### 5.4.4 elevons

Elevon forces are modeled similarly to lift, as well, but with an extra term,  $\delta_i$ , for the deflection degree in radians:[3]

$$F_i = C_{L_\delta} \sum_{j \in [a, a_w]} Q_{j,i} S_j \delta_i \quad (25)$$

and similarly the drag created by the elevons:

$$D_{\alpha_i} = C_{D_\delta} \sum_{j \in [a, a_w]} Q_{j,i} S_j \delta_i \quad (26)$$

The elevon lift, and drag, coefficient is assumed to be static over all angle of attacks since they will be in the wake of the wing, and the lift and drag are linear in the region of angles the ailerons can deflect.

### 5.5 Winglets, forces and moments due to sideslip

The aircraft considered here has no tail nor real body. The only surfaces providing relevant forces due to sideslip are the winglets.

The winglets each have an area  $S_W$  and are flat plate wings which will generate aerodynamic lift and drag forces due to  $\beta$  providing an angle of attack in their reference frame.

$\beta$  can be, similarly to  $\alpha$ , be expressed in polar coordinates:

$$\begin{aligned} v_x &= r \cos(\beta) \\ v_y &= r \sin(\beta) \\ r &= \sqrt{v_x^2 + v_y^2} \end{aligned} \quad (27)$$

The aerodynamic forces,  $L_\beta$  and  $D_\beta$ , from the winglets convert into body forces, directed as in figure 16, similarly to lift and drag from the main wings:

$$\begin{aligned} L_{W_i} &= \cos(\beta)L_\beta + \sin(\beta)D_\beta \\ D_{W_i} &= -\sin(\beta)L_\beta + \cos(\beta)D_\beta \end{aligned} \quad (28)$$

where

$$\begin{aligned} L_\beta &= QC_{L_\beta} S_W \\ D_\beta &= QC_{D_\beta} S_W \end{aligned} \quad (29)$$

and  $C_{k_\beta}$  are the lift and drag coefficients for each winglet, here assumed to be flat plates.  $v_z$  can be assumed to be negligible in the following equations since it will have no measurable effect on the winglets. If we assume  $v_z$  to be negligible then the equations simplify into

$$\begin{aligned} L_{W_i} &= \frac{\rho v S_W}{2} (C_{L_\beta} v_x + v_y C_{D_\beta}) \\ D_{W_i} &= \frac{\rho v S_W}{2} (-v_y C_{L_\beta} + v_x C_{D_\beta}) \end{aligned} \quad (30)$$

If the sideslip angle,  $\beta$ , is small (which implies  $v_y$  small) the lift/drag coefficients can be approximated as

$$\begin{aligned} C_{L_\beta} &\approx \frac{dC_{L_\beta}}{d\beta} \Big|_{\beta=0} \beta \\ C_{D_\beta} &\approx 0 \end{aligned} \quad (31)$$

which results in:

$$\begin{aligned} L_{W_i} &\approx \frac{\rho v S_W}{2} \frac{dC_{L_\beta}}{d\beta} \Big|_{\beta=0} \beta v_x \\ D_{W_i} &\approx 0 \end{aligned} \quad (32)$$

This gives us, in body frame, one restoring force from each winglet roughly proportional to the forward velocity squared of the aircraft and the sideslip angle.. The forces have the same magnitude and direction on both sides of the aircraft.

### 5.5.1 Restoring moment

$D_{W_i}$  are symmetrical on both sides and only create an acceleration in  $y$ . They do not generate any moment.  $L_{W_i}$ , however, act asymmetrically around c.g. and do generate a moment/torque (note that the force  $L_{W_R}$  generates a negative torque around  $z$  in our coordinate system):

$$M_z = L_{W_L} x_W + L_{W_R} x_R = -\rho v S_W \frac{dC_{L_\beta}}{d\beta} \Big|_{\beta=0} \beta v_x x_W \quad (33)$$

### 5.5.2 Non linear Forces and Moments

If we do not make the assumption that  $\beta$  is small we instead get the equations:

$$\begin{aligned} M_z &= -x_W \rho v S_W (C_{L_\beta}(\beta) v_x + v_y C_{D_\beta}(\beta)) \\ D_W &= \rho v S_W (-v_y C_{L_\beta}(\beta) + v_x C_{D_\beta}(\beta)) \end{aligned} \quad (34)$$

where  $D_W = D_{W_L} + D_{W_R}$ , and the coefficients of lift and drag are as shown in the figures above, but functions of  $\beta$  instead of  $\alpha$  as  $\beta$  becomes the effective angle of attack in the winglets reference frame.

### 5.5.3 Rolling moment due to sideslip

The center of the winglet is slightly offset in the  $z$ -direction from the c.g. by distance  $z_W$ . Given the forces  $L_{W_i}$  the rolling moment due to sideslip,  $M_x$ , is simply

$$M_x = z_W (L_{W_L} + L_{W_R}) \quad (35)$$

## 5.6 Propeller Thrust

Thrust will be modeled as

$$T_i = K_T \omega_i^2 \quad (36)$$

where  $K_T$  is the thrust coefficient of the propeller and  $\omega_i$  the rotational speed of the propeller.

Behind the propellers an area of airflow is generated, called a wake. Given a propeller diameter of  $d_P$ , pushing air through at a rate of  $v_w$  the volume of air being moved by the propeller during time  $t$  is:

$$V_{air} = \int_0^t \frac{d}{2} \pi^2 v_w dt. \quad (37)$$

The air has density  $\rho$  converting the volume into mass:

$$m_{air} = \rho m_{air} \quad (38)$$

If we assume the air to be stationary in front of the propeller and the aircraft standing still the change in momentum in the air is:

$$\Delta p = m_{air} v_{air} = \rho \frac{d\pi^2}{2} v_w^2 t \quad (39)$$

Force due to change in momentum is:

$$Ft = \Delta p \Rightarrow F = \frac{\Delta p}{t} \quad (40)$$

giving us:

$$F = \rho \frac{d\pi^2}{2} v_w^2. \quad (41)$$

By combining all parameters into one coefficient  $K_v$ , the following equation is obtained:

$$F = K_v v_w^2. \quad (42)$$

If we instead look at the lift generated by the propeller blades we can find the thrust as a function of propeller rotation speed. Assuming the aircraft has no speed and there is no wind we can model the lift from one propeller blade as:

$$L_i = \int_0^{\frac{d}{2}} \frac{\rho}{2} (\omega_i r)^2 C_L(pitch) dr \quad (43)$$

If we also assume a constant pitch along the diameter of the propeller blade the above integral simplifies into:

$$L_i = \frac{\rho d^3}{48} C_L(pitch) \omega_i^2 \quad (44)$$

and thus with  $n$  blades we find the total lift, i.e. thrust:

$$T_i = L = \sum_{i \in [1, n]} L_i = n L_i \quad (45)$$

By comparing the equation for thrust due to rotation speed and thrust due to airflow we conclude that:

$$K_v v_w^2 = n \frac{\rho d^3}{48} C_L(pitch) \omega_i^2 \Rightarrow \\ v_w = b_w \omega_i \quad (46)$$

and so the assumption that airflow in the wake is proportional to the propeller angular speed holds.

## 5.7 Restoring moment due to roll rate

If we assume that  $v_y$  is negligible then the Lift and Drag forces in body fram simplify into:

$$L_i = S \frac{1}{2} \rho v^2 \left( \frac{C_L v_x + C_D v_z}{\sqrt{v_x^2 + v_z^2}} \right) = S \frac{1}{2} \rho \sqrt{v_x^2 + v_z^2} (C_L v_x + C_D v_z) \\ D_i = S \frac{1}{2} \rho v^2 \left( \frac{-C_L v_z + C_D v_x}{\sqrt{v_x^2 + v_z^2}} \right) = S \frac{1}{2} \rho \sqrt{v_x^2 + v_z^2} (-C_L v_z + C_D v_x) \quad (47)$$

If the plane is experiencing rolling, a rotation about the  $x$ -axis with a rotational speed of  $\omega_x$  then a restoring moment will be created. This is because the air is resisting the wings motion through the air.

The rotational speed will yield an increased local  $v_z$ ,  $v_{z,l}$ , along the wing:

$$v_{z,l} = v_z + \omega_x y \quad (48)$$

where  $y$  is the distance away from the body along the wing. The change in velocity changes the angle of attack locally. The local angle of attack is denoted  $\alpha_l$ .

Lets consider the full moment as the integral of the full moment across both wings. This way the symmetrical lift forces will cancel and only the rolling moment will remain.

$$\begin{aligned} M_x &= - \int_{-b/2}^{b/2} y dL_i(y) dy \\ &= - \frac{\rho}{2} \int_{-b/2}^{b/2} yc(y) \sqrt{v_x^2 + v_{z,l}^2} (C_L v_x + C_D v_{z,l}) dy \end{aligned} \quad (49)$$

If we make the crude assumption that  $C_L = a * \sin(2 * \alpha)$ , and the less crude assumption that  $C_D = -b * \cos(\alpha) + b$ , it is still impossible to find the function  $M_x(\omega_x)$  for the entire state space. A term:

$$\int_{-b/2}^{b/2} y \sqrt{(v_1 + \omega y)^2 + v_2^2} dy \quad (50)$$

will remain, and one has to assume  $\alpha \in [-\pi/2, \pi/2]$ ,  $v_2 \neq 0$  and/or many of the variables always positive, to find the function without the integral.

A simulator can solve the integral numerically but in order to find smooth analytical expressions for the controller approximations have to be made.

It turns out that the four dimensional function for restoring moment due to roll as a function of roll rate  $\omega_x$ ,  $v_x$  and  $v_z$  is actually quite linear with respect to  $\omega_x$ . The remaining function can be closely approximated by a two dimensional polynomial in  $v_x$  and  $v_z$  allowing for a feedback linearized model to be implemented with very little error. This can also help with computational efficiency should an estimator need to run onboard the aircraft.

## 5.8 Restoring moment due to yaw rate

Similarly the restoring moment due to yaw rate can be obtained from the following integral:

$$M_z = \int_{-b/2}^{b/2} y \left( \frac{1}{2} \rho \sqrt{v_{x,l}^2 + v_z^2} (-C_L v_z + C_D v_{x,l}) \right) c(y) dy \quad (51)$$

where  $v_{x,l} = v_x - \omega_z y$ .

## 5.9 Rolling moment due to yaw rate

Due to the change in forward velocity a change in lift is also produced, inducing a rolling moment obtained similarly:

$$M_x = \int_{-b/2}^{b/2} y \left( \frac{1}{2} \rho \sqrt{v_{x,l}^2 + v_z^2} (C_L v_{x,l} + C_D v_z) \right) c(y) dy \quad (52)$$

where  $v_{x,l} = v_x - \omega_z y$ .

## 5.10 Restoring moment due to pitch rate

Due to the absence of a tail this moment will be much smaller than on aircraft with tail. Most of the wing surface area is behind the center of gravity and due to the wing sweep more surface will be at a longer distance from c.g. in  $x$  direction. This means that we can approximate the entire wing to be behind the c.g. This aircraft also has a relatively small inertia around the  $y$  axis; the aircraft will quickly turn into the wind.

Since we have modeled the lift in such a way that the pitching moment does not vary with angle of attack we can assume here that it is only proportional to the aircraft velocity and  $\omega_y$ , similarly as in section 5.7 but with constants instead of coefficients of lift and drag, giving us the following expression directly:

$$M_y = -\omega_y \rho S \sqrt{v_x^2 + v_z^2} C_{D,\omega_y} \frac{x_{\omega_y}^2}{24} = -\omega_y \rho S C_{\omega_y} \sqrt{v_x^2 + v_z^2} \quad (53)$$

where  $C_{\omega_y}$  is some constant that needs to be approximated experimentally.

## 5.11 Actuator dynamics

The actuators onboard are not direct term but contain some dynamics. In this thesis they will be modelled as first order systems.

### 5.11.1 Elevon dynamics

Given an input signal  $u_{\delta_i}$  for elevon  $i$  we model the deflection,  $\delta_i$ , like so:

$$\dot{\delta}_i = K_{\delta_i}(-\delta_i + u_{\delta_i}) \quad (54)$$

### 5.11.2 Motor dynamics

Given an input signal  $u_{\omega_i}$  we model the motor dynamics as so:

$$\dot{\omega}_i = K_{\omega_i}(-\omega_i + u_{\omega_i}) \quad (55)$$

The parameters  $K_{\omega_i}$  and  $K_{\delta_i}$  are chosen to roughly match the response times listen in the Servo and ESC manufacturers data sheets.

## 5.12 Full equations of motion

We now include all terms to find the full state space equations:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{\delta}_L \\ \dot{\delta}_R \\ \dot{\omega}_L \\ \dot{\omega}_R \end{bmatrix} = \begin{bmatrix} 2g(q_1 q_3 + q_0 q_2) - \sum_i (D_i + D_{i,T} + D_{W_i} + D_{\alpha_i} + D_{\alpha_{i,T}}) + T_L + T_R \\ 2g(q_2 q_3 + q_0 q_1) + L_{W_L} + L_{W_R} \\ g(q_0^2 - q_1^2 - q_2^2 + q_3^2) + \sum_i (L_i + F_i + F_{i,T}) \\ f_1/I_x \\ f_2/I_y \\ f_3/I_z \\ \frac{1}{2}(q_1 \omega_x + q_2 \omega_y + q_3 \omega_z) \\ \frac{1}{2}(q_2 \omega_z - q_3 \omega_y - q_0 \omega_x) \\ \frac{1}{2}(q_3 \omega_x - q_0 \omega_y - q_1 \omega_z) \\ \frac{1}{2}(q_1 \omega_y - q_2 \omega_x - q_0 \omega_z) \\ K_{\delta_L}(-\delta_L + u_{\delta_L}) \\ K_{\delta_R}(-\delta_R + u_{\delta_R}) \\ K_{\omega_L}(-\omega_L + u_{\omega_L}) \\ K_{\omega_R}(-\omega_R + u_{\omega_R}) \end{bmatrix} - \begin{bmatrix} 2(v_z \omega_y - v_y \omega_z) \\ 2(v_x \omega_z - v_z \omega_x) \\ 2(v_y \omega_x - v_x \omega_y) \\ \frac{1}{I_x}(I_z - I_y)\omega_y \omega_z \\ \frac{1}{I_y}(I_x - I_z)\omega_x \omega_z \\ \frac{1}{I_z}(I_y - I_x)\omega_x \omega_y \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (56)$$

where  $f_1$  is given by:

$$\begin{aligned} f_1 = & y_f C_{L_\delta} \sum_{j \in [a, a_W]} S_j (Q_{j,L} \delta_L - Q_{j,R} \delta_R) \text{ :: difference in elevon deflection} \\ & - \frac{\rho}{2} \int_{-b/2}^{b/2} y c(y) \sqrt{v_x^2 + v_{z,l}^2} (C_L(\alpha_l) v_x + C_D(\alpha_l) v_{z,l}) dy \text{ :: restorint moment due to roll} \quad (57) \\ & + \int_{-b/2}^{b/2} y \left( \frac{\rho}{2} \sqrt{v_{x,l}^2 + v_z^2} (C_L v_{x,l} + C_D v_z) \right) c(y) dy \text{ :: roll due to yaw rate} \end{aligned}$$

,  $f_2$  by:

$$\begin{aligned} f_2 = & x_f C_{L_\delta} \left( \sum_{j \in [a, a_W]} S_j (Q_{j,L} \delta_L + Q_{j,R} \delta_R) \right) \text{ :: joint elevon deflection} \\ & + \sum_{j \in [w, w_w]} C_{m,y,l} Q_j S_j l \text{ :: pitch moment} \quad (58) \\ & - \omega_y \rho 2 S_w C_{\omega_y} \sqrt{v_x^2 + v_z^2} \text{ :: restoring moment due to pitch rate} \end{aligned}$$

and  $f_3$  by:

$$\begin{aligned}
f_3 = & y_T (T_L - T_R) :: \text{diff in T} \\
& - x_W \rho v S_W (C_{L_\beta}(\beta) v_x + v_y C_{D_\beta}(\beta)) :: \text{moment due to sideslip} \\
& + \int_{-b/2}^{b/2} y \left( \frac{\rho}{2} \sqrt{v_{x,l}^2 + v_z^2} (-C_L v_z + C_D v_{x,l}) \right) c(y) dy :: \text{restoring moment due to yaw rate}
\end{aligned} \tag{59}$$

$Q_{j,i}$  is given by  $\frac{\rho}{2} v_k^2$  where  $v_k$  is the speed in the relevant area, which can be either the aircraft velocity if the surface in question is outside the propeller wake, or the propeller wake air speed given by the left or right propeller speed if inside the propeller wake.

Variables like  $D_i$  and  $D_{i,T}$  differ by both which area they use and which speed the airflow has since they are separated by if they are on the right or left side, inside or outside the propeller wake, and thus the correct expressions for the airspeed have to be used. The same argument goes for expressions which depend on  $\alpha$ .

## 6 EcoSoar replacement

Due to logistical issues in Malawi, time limitations and limitations in the Pixhawk library provided by Mathworks, the full dual engine Eco-Soar could not be built. As a proof of concept a flying board was built to see if the controller would be feasible to implement on an EcoSoar in the future.

### 6.1 Parameter adjustments

Since a flying board has no camber all effects due to camber has been removed in the simulator. Also all physical measurements are replaced with those of the flying board. All other critical physical phenomena has been accounted for: The flying board has similar winglets, it has a similar c.g. vs center of pressure, two control surfaces partially covered by the propeller wakes and electronics in the middle to separate the two wings slightly. Measurements are roughly similar, the flying board is slightly longer and has a much shorter wingspan, but the physics should be comparable. The simulator was adjusted to match the flying board.

### 6.2 Building

A piece of plywood was bought at the local market area. Engine mounts were 3D-printed. Elevon areas were cut out of the board and poster board was cut to fit in the slots as elevons. The hinges were made out of tape. Winglets were cut out in poster board to roughly same size as on the EcoSoar, and hot glued onto the top of the board to provide the physics of sideslip stability. A 5.5Ah 3S-LiPo was zip tied to the board in the very front. Two Electronic Speed controllers of type KDEXF-UAS35 were hot glued on the sides of the battery in the very front as well. The ESCs were selected specifically to have a very fast response and lots of power to spare as to not become a limiting factor. The engines are the same as on the EcoSoar: 1300KV 420W three phase brushless motors. Two small 2kg/cm servos were mounted close to the elevons as actuators, linked with musical wire. A telemetry module and receiver were mounted far away from other electronics to minimize interference. The Pixhawk was glued behind the battery as close to the center of gravity as reasonable. Gluing provides a sturdier mount to the board for more accurate reading than using double sided tape. Due to the many crashes of the EPO wing which resulted in broken propellers two carbon rods were mounted perpendicular to the board, one in front on each side. When the board lands the propellers will not hit the ground as easily. The rods are held in place by tight fitting rubber bands on each side of the board so to provide some dampening effect should the landing be very hard.

After several crashes a few extra modifications were made which can be seen in Figure 6. The carbon rods were replaced with vertical pieces of plywood large enough to act as propeller guards but with a small enough area as to not make the aircraft statically unstable in yaw. The actual area of the front and rear vertical surfaces are equal, but the moment arm to the c.g. is much greater for the rear surfaces making the aircraft still be statically stable in yaw. After one of the

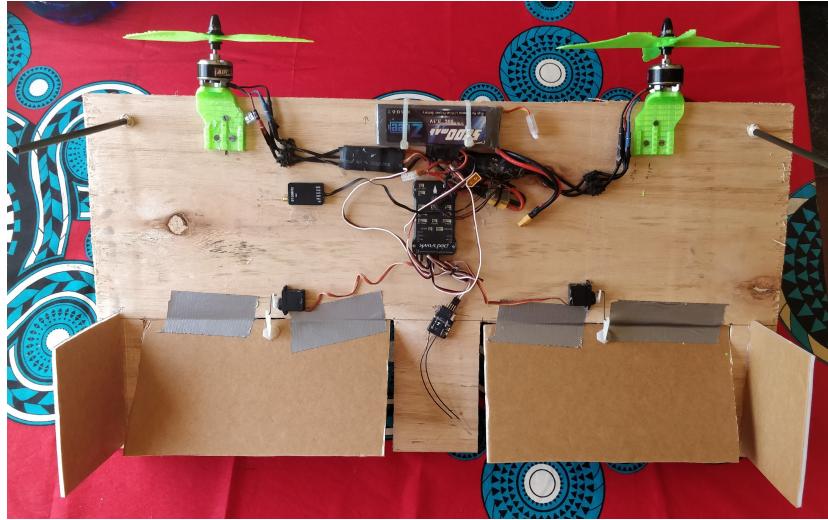


Figure 5: A picture of the flying board, made out of plywood. The propellers are just placeholders for the photo.

batteries was crushed in a particularly hard crash, turning the battery into ashes, a battery guard was made and mounted. The engine mounts were slightly reinforced and made straight as well. The plywood areas under the winglets were broken off in crashes, glued back with carbon rod reinforcements.

### 6.3 Parameter estimation/declarations

Most parameters listed can be directly measured on the flying board, such as wing area, weight, center of gravity, etc. Inertias have been estimated in a CAD program, and other parameters were estimated from measurements/data sheet on similar aircraft. The simulator is, anyway, just a feasibility check and not a proper tuning ground for the controller, finding bugs and rough settings is good enough.

## 7 Robust controller development and Implementation

The controller implemented is based on a non-linear Proportional squared ( $P^2$ ) controller by Emil Fresk[4]. The controller was first implemented on a simulator of the EcoSoar made in Simulink. A few other controllers were first attempted, among them a Feedback Linearization controller.

### 7.1 Feedback Linearization

Due to the complex nonlinear state space equations solving the system symbolically for feedback simulations was infeasible. Matlab would compute for days without result and without time estimate. Filling in certain parameters with numerical values made a semi-symbolic solution possible, which behaved nicely in simulations. However, the decoupling matrix had a condition number in the order of  $10^5$  making the system very susceptible to any noise or uncertainties. The equations for calculating the necessary actuation also contained very large numbers (order of  $10^{80}$ ) divided by other large numbers meaning that any discrepancies would crash the airplane.

Solving the inverse numerically on the fly was also considered but had similar issues as well as being computationally heavy for a small computer as the Pixhawk 1. The numerical approach also left no guarantees on existence of solution, because theoretically some of the integrals do have singularities and so the decoupling matrix dependent on them can theoretically be non-invertible in certain (unlikely) situations.

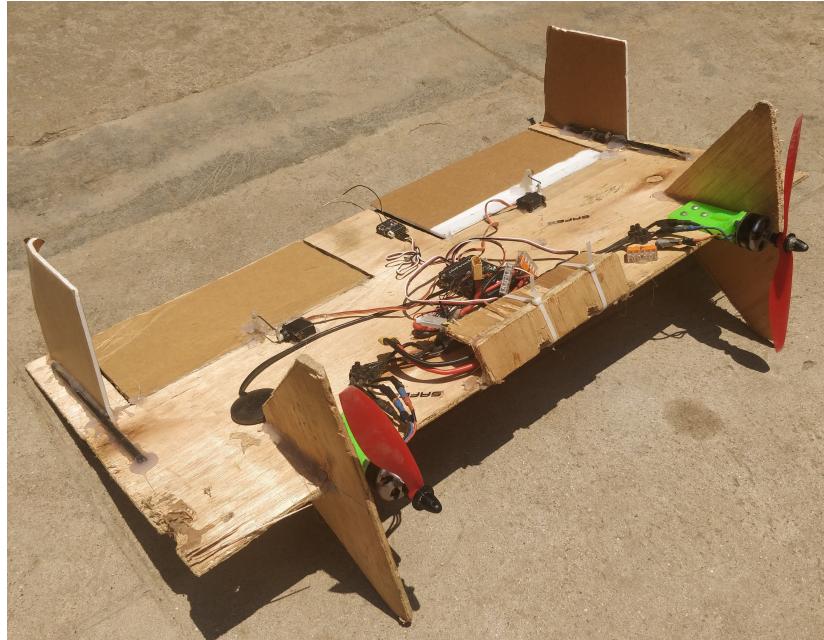


Figure 6: A picture of the final version of the flying board. The propeller guards have been replaced with plywood, a battery guard has been added, the rear winglet holding areas are reinforced with carbon fibre rods and the gps module is added on the right side.

After spending too much time trying to work around these issues the plan was abandoned.

## 7.2 Simulink

The full non linear state space equations derived above in section 5.12 were put in a matlab function block surrounded by state integrators to generate the full state. Parameters were either measured or estimated to complete the numerical simulator. Controllers could then be tested and evaluated quickly. An instant 3D render of the simulation was also added through Simulinks VR support blocks.

TODO: INSERT PICTURE OF SIMULATOR

## 7.3 Pixhawk firmware

Mathworks has support for the embedded coder to directly upload apps into the PX4 NuttX Real Time OS that can run on the Pixhawk 1. The firmware has to be flashed to PX4 on the Pixhawk, files modified on the SD card and a wide range of supporting softwares must be installed on the host computer. Mathworks has great documentation on the subject.[10]

## 7.4 Initial testing on Bixler

To evaluate the firmware/simulink controller deployment into a Pixhawk a Bixler was used. The Pixhawk was put into the Bixler to replace the receiver inside, necessary hardware added and then a simple FlyByWire controller was implemented and uploaded into the Pixhawk from Simulink. Results were very good, the Bixler flew as expected and no issues with the software on-board. This was done as a sanity check for the complete software to hardware deployment strategy.

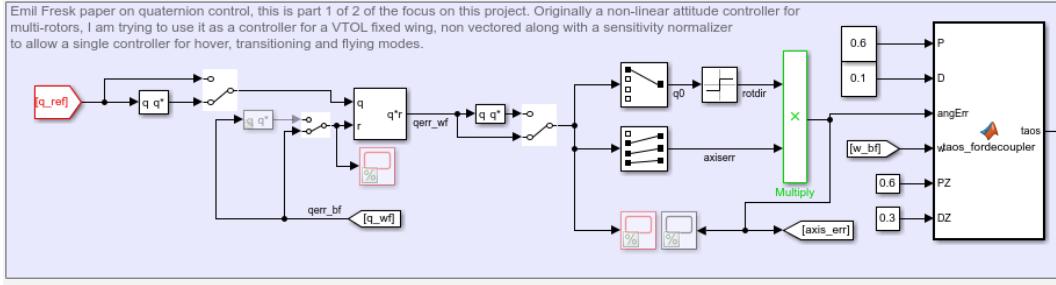


Figure 7: The central P2 controller. The manual switches are for debugging, and the very right side block separates the P-D ratios for the propellers (z-axis) from the rest of the system as they require separate tuning.

## 7.5 P2-controller

The Pixhawk was then removed from the Bixler and placed on the flying board to test a simple version of the full controller. The P2 controller [4] was implemented along with separate tuning for the three axis but a hardcoded, non adjustable, reference attitude of nose up, belly to the north, no logs and no other features. The controller can be seen in Figure 7

The implemented controller can thus be thought of as a set of three P-D controllers, one for each axis. The board would hover, but no logs could initially be produced due to a bug in the embedded coder in Simulink. The sample frequency was set to 120 Hz since the full standard 250 Hz was too much load on the Pixhawk resulting in boot loops after the OS being overrun. This also allowed the PWM frequency to match the main control loop as anything above 120 Hz was too fast for the servos.

## 7.6 Torque translator

Neither the EcoSoar nor the flying board have propeller wakes that cover the entire elevons. Therefor it is of interest to normalize the actuation signals to the elevons given the current state so that they produce the requested torque from the P2 controller in any given state. By assuming that the airflow inside the wake and outside the wake do not affect each other and that the dynamic pressure is the dominating effect a simple function can be found that takes the current rotational speeds of the propellers, the current velocity of the aircraft and requested torque around  $x$  and  $y$  axis and gives the deflection angle of the control surfaces.

We assume that damping due to rotational speeds do not exist, winglets have no  $z$ -offset, and no net pitch moment is created by the lift from the wings. The equations for torque around each axis then become:

$$\begin{aligned}\tau_x &= y_f(F_L + F_{LT} - F_{RT} - F_R) \\ \tau_y &= x_f(F_{LT} + F_L + F_{RT} + F_R) \\ \tau_z &= y_T(T_L - T_R)\end{aligned}\tag{60}$$

which indicates that the elevons force per deflection is proportional to the airflow speed over them squared. The P2 controller outputs desired angular acceleration,  $\dot{\omega}_i$ , not torque. Necessary torque,  $\tau_i$ , for axis  $i$  is given by

$$\tau_i = \dot{\omega}_i I_i\tag{61}$$

Solving for differential thrust is possible. Taking the inertia around each axis into account the necessary deflection for each elevon,  $u_{\delta_i}$  can be solved for given the current state:

$$\begin{aligned}u_{\delta_L} &= \frac{-0.5\tau_y}{k_1 sgn(v_x)v^2 + k_2\omega_L^2} - \frac{0.5\tau_x}{k_3 sgn(v_x)v^2 + k_4\omega_L^2} \\ u_{\delta_R} &= \frac{-0.5\tau_y}{k_1 sgn(v_x)v^2 + k_2\omega_R^2} + \frac{0.5\tau_x}{k_3 sgn(v_x)v^2 + k_4\omega_R^2}\end{aligned}\tag{62}$$

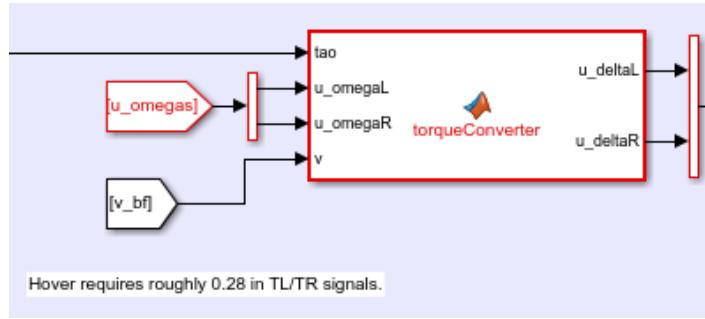


Figure 8: Controller implementation of equation 62.

where  $k_{[1-4]}$  are numerical constants given by aircraft parameters.

The torque translator is implemented in the Simulink controller as a matlab function, which can be seen in Figure 8.

### 7.6.1 Finding $k_2$ and $k_4$

Bypassing the torque translator and running the controller directly to the control surfaces a suitable step response can be tuned in while hovering. While hovering still in the air  $v_x$  can be assumed to be negligible, and thus once the wanted response has been found one can solve for  $k_2$  and  $k_4$ ; the rotational speeds are known signals.

### 7.6.2 Finding $k_1$ and $k_3$ : Single Engine Board

By flying normally with a standard Fly By Wire system and PD controller for pitch and roll a suitable response can be tuned in, granted that the board only has one single engine which propeller wash does not interfere with the control surfaces. This means that one can assume the contribution from the terms  $k_2$  and  $k_4$  are included in to be zero. Since velocity and the ratio torque/actuation is known the remaining parameters can be found.

For this very task an 80% scale model of the flying plywood board was built, but with a single engine in the front instead of two.

This test only gives a rough estimate, however, since the propeller wash does cover the elevons partially. A rear mounted propeller would have been better but there was no way to keep the c.g. in front of the center of pressure without a complete redesign of the aircraft had the engine been mounted in the rear. Therefore the parameter estimations found through flying this board only serve to find rough estimates good enough for flying the dual engine one without crashing instantly.

## 7.7 Full controller diagram

The full controller has an adjustable reference quaternion generated by user inputs on the transmitter, a manual emergency motor and servo shutdown, an altitude holder, SD signal logging blocks, a step response generator, low battery voltage protection cutoff and LED control for the onboard led to indicate arming state and GPS accuracy.

The horizontal layout of the Simulink blocks in the full controller make an overview picture difficult. The full schematics can instead be found in the git repo.[11]

In Figure 10 the quaternion and angular rates from the Extended Kalman Filter for attitude estimation in the firmware is read. In Figure 11 the reference quaternion is generated. The reference quaternion, estimated quaternion and angular rates are passed onto the P2 controller shown in Figure 7. The reference quaternion can be modified by the step function generator shown in Figure 12. The emergency shutoff function is shown in Figure 13.

The battery voltage reading is low pass filtered due to large noise in the signal triggering short engine cutouts in flight. The lowest allowed cell voltage is on purpose set quite low due to voltage

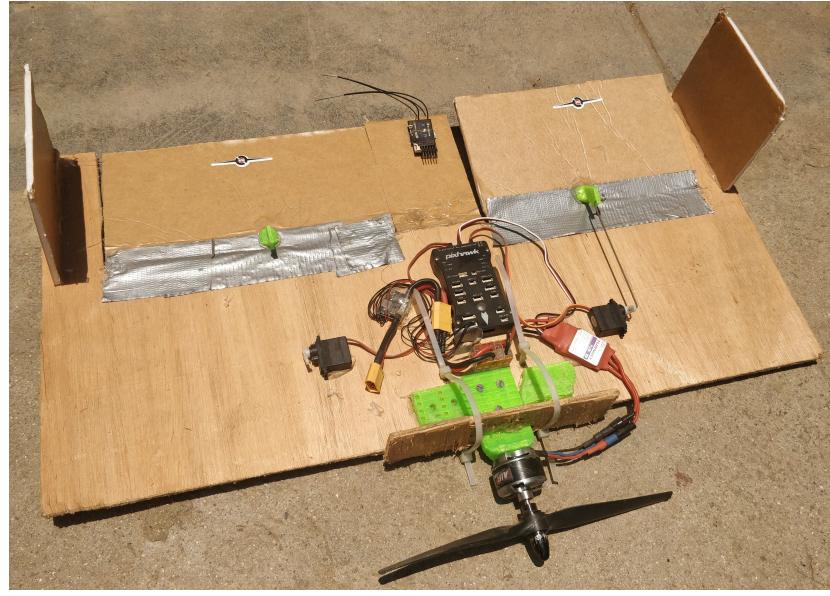


Figure 9: A picture of the 80% scale model, single engine version, flying board used for parameter estimation.

being much lower under load. A 3 Volt per cell cutoff results in the resting voltage stabilizing around 3.5 without load which is more than a safe discharge voltage.

## 8 Results

- Talk about the logging issues, hard faults and time slowdown here or in conclusion.

The plywood board can hover and transition into flight. Further tuning would be preferable but not possible for the scope of this thesis due to time and weather constraints.

### 8.1 Performance in Simulations

Since all parameters are well known/set in the simulator and assumptions are the same for the simulator and controller the system works very well in simulation.

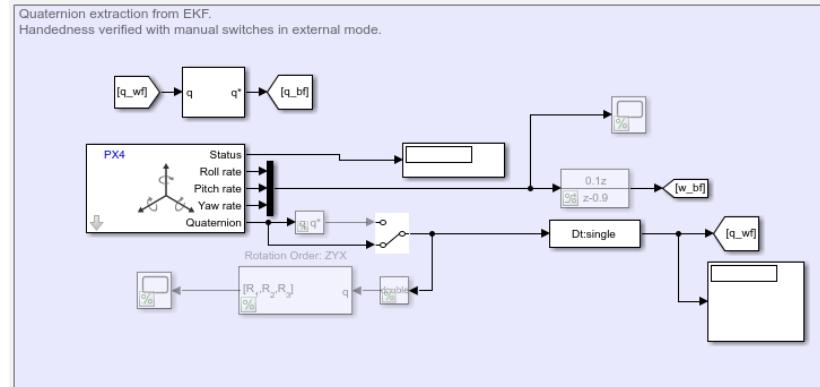


Figure 10: The quaternion estimation retrieval from the Pixhawk firmware.

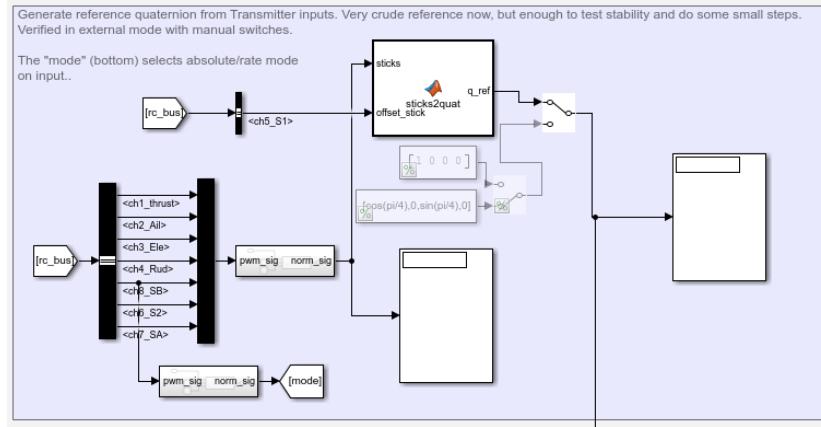


Figure 11: The quaternion reference generation, modified with RC inputs.

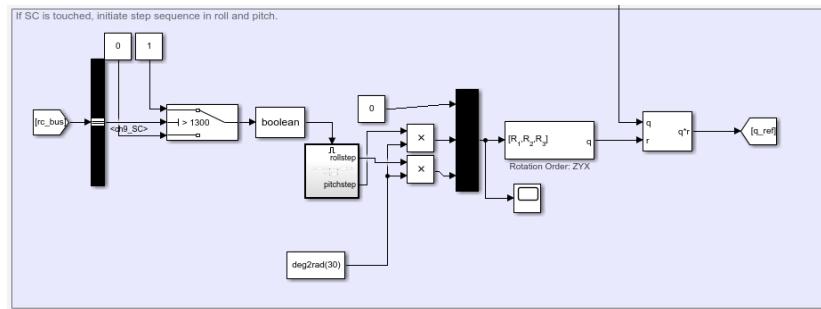


Figure 12: The attitude step generation routine.

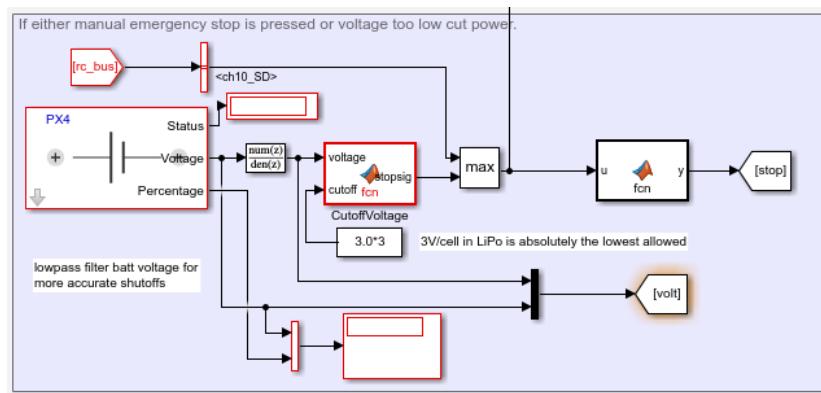


Figure 13: Emergency shutoff routine which overrides all outputs should it be necessary.

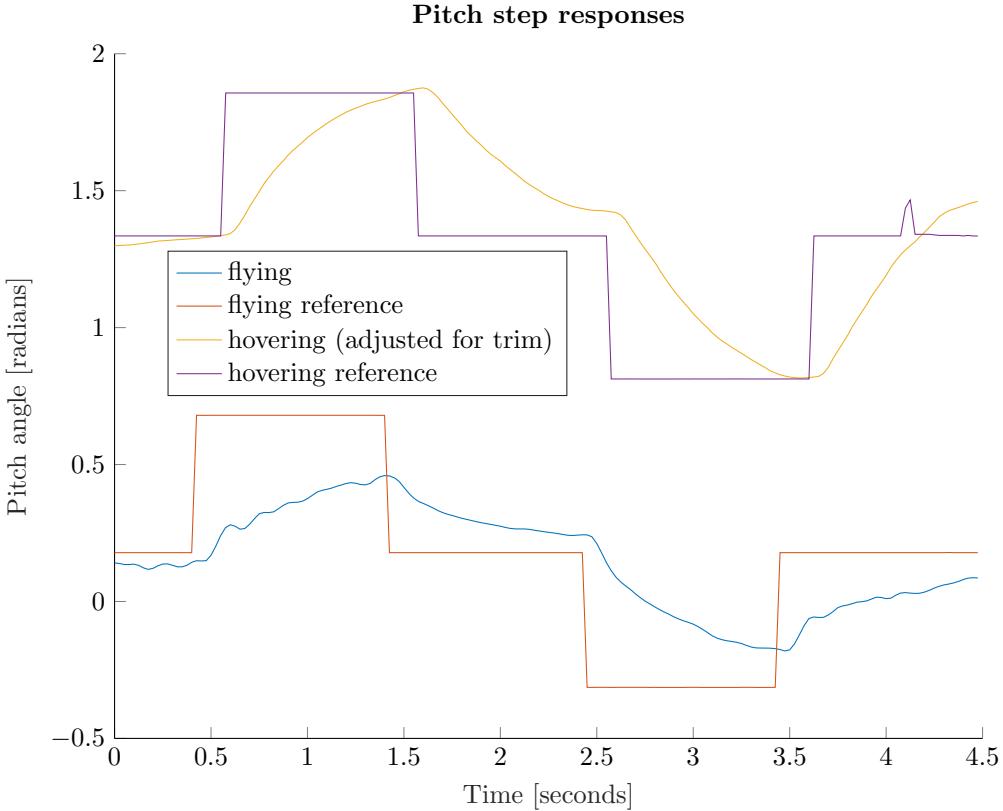


Figure 14: Step responses in pitch for the flying Plywood board while both hovering and flying. The data has been adjusted to compensate for mechanical trim offset.

TODO: Maybe add graphs of simulations? The simulations worked so well with almost any settings maybe they aren't of interest?

## 8.2 Performance of the flying plywood board in reality

The board hovers and flies controllably. To achieve very smooth flight more tuning is necessary but the fact that it flies at all without the need of any switching controller indicates that the concept of normalizing the sensitivity of the control surfaces is very much valid.

Below, in figure X, step responses in pitch and roll while both hovering and flying are displayed. The responses are similar but of different gain.

## 9 Conclusion

Some of the results presented are adjusted for mechanical trim. This system, contrary to the multirotor discussed in [4], does need an integrator since it is not embedded in the system by nature. A perfectly mechanically trimmed version of the flying board would not need the integrator but the current setup does not allow for fine tuning. A slow integrator would also be able to adjust for varying gain due to unforeseen nonlinear effects.

The step responses in flying vs hovering mode have different gain which is most likely due to the  $k_i$  parameters in the torque translator not being perfectly tuned. As stated the single engine board only served to find rough estimates to allow for controlled flight, which it did, but not perfect parameter estimations. The gain when flying was too low, indicating that the expected

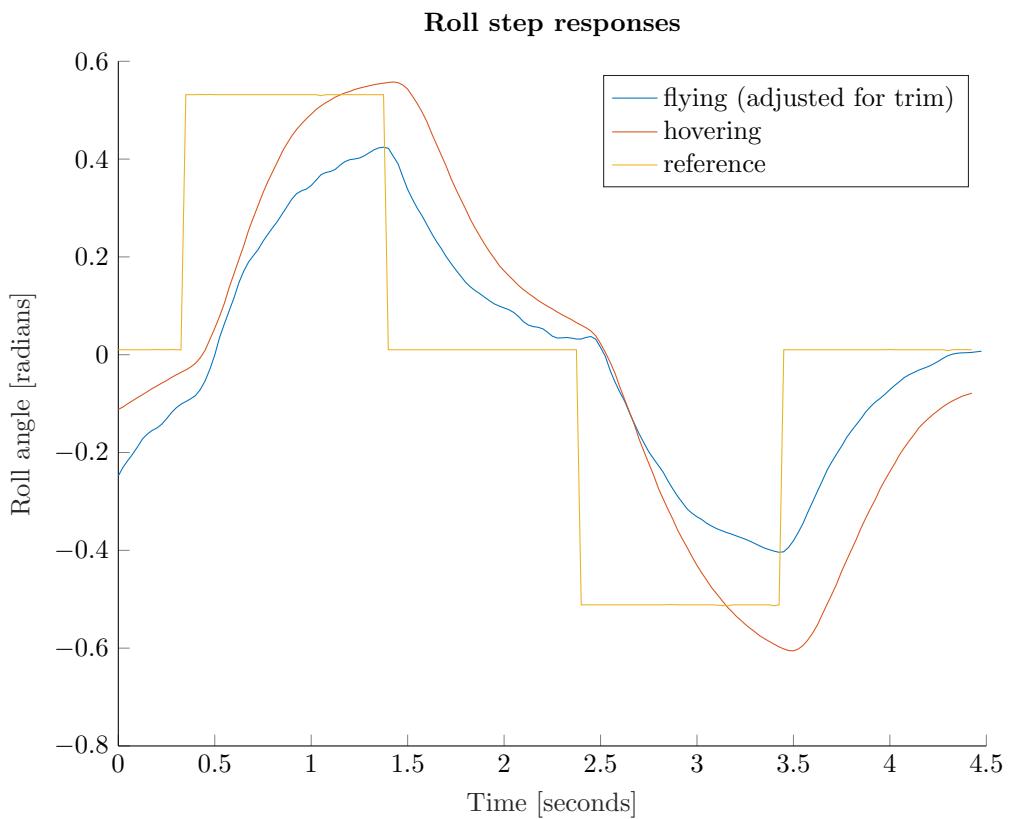


Figure 15: Step responses in roll for the flying Plywood board while both hovering and flying. The control surfaces were not mechanically perfectly trimmed so the data above has been offset to compensate.

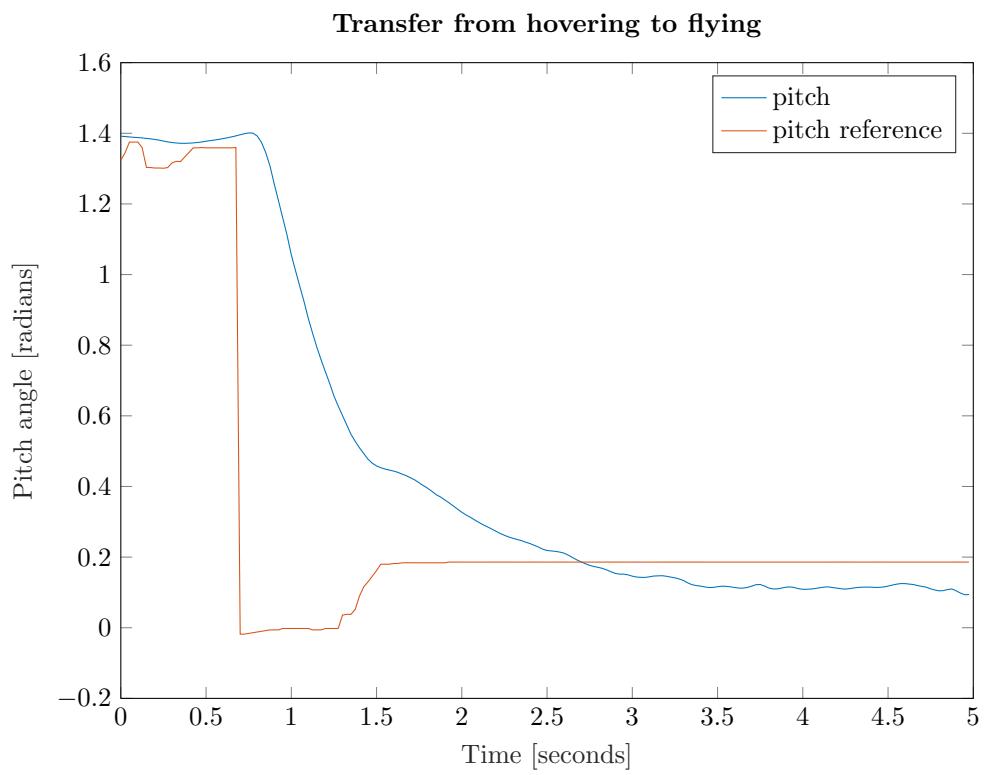


Figure 16: Step response in pitch when transitioning from hovering to flying. The end reference value had to be adjusted to keep the altitude above ground.

sensitivity of the elevon was higher, i.e.  $k_1$  and  $k_3$  are too big. Physically this means that the external airflow has a smaller effect than anticipated which probably is due to the propellers wake covering a much bigger proportion of the elevon than expected. Either a few more test flights in calm wind to tune those parameters in, or some form of online L1 parameter estimation/correction would probably solve this issue. When moving this control system to the very similar EcoSoar an online parameter estimation algorithm would probably be preferable to avoid the difficult tuning process which comes with the risk of crashing; rebuilding an EcoSoar is not only difficult and time consuming but more expensive than buying some more plywood at the local market.

In the step responses while flying some small oscillations can be seen. This seems to correspond to oscillation frequency that can be seen in the elevon servos when the system has mechanical play. The linkage from servo to elevon have play that results in maybe a 5 degree uncertainty and it is possible for the servos to end up in resonance around this play. When flying the dynamic pressure is higher on the elevons and maybe that increased pressure is enough to initiate the oscillations during transitioning. Either a higher precision linkage or a higher quality servo might remove these oscillations.

The vast number of logged/recoded crashed throughout this thesis (in excess of 40) would have been alleviated with a vicon/motion capture system. Many of the initial crashes were simple quaternion bugs that would have been found by just moving and rotating the aircraft in an external known coordinate system. This prolonged the development time significantly and the scarce resources, along with limited information about the surrounding supply chains, made progress even more slow and uncertain.

Another issue that made development very difficult is the very limited hardware and processing power of the Pixhawk 1. Although this system in its now working state can be run on the Pixhawk 1, development should have been done on something more powerful to allow for all the debugging, logging and testing unoptimized algorithms. The current algorithm would also probably be more suited to be implemented in c++ code directly in the firmware. The actual algorithm would not take many rows of code nor much resources. What causes the slowdowns is probably the unoptimized simulink generated c++ code that now needs to handle everything from receiving RC input to generating PWM signals. In the firmware these tasks have been more optimized and coded for the specific hardware in a more dedicated way.

Throughout this project two Simulink bugs have been found, reported and fixes were received from Mathworks. The first one caused several, hard to debug, mid flight reboots resulting in the aircraft falling out of the sky. This was extra difficult to understand because Mathworks has implemented the SD logging feature such that upon reboot it overwrites all log files up until the reboot, leaving the developer with no data to look at other than a hex dump with the stack pointer location from the firmware. Mathworks support was great at helping fix this issue by providing custom SD logging code.

The second issue was that the SD logging block in Simulink would buffer log data in ram and not write directly to the SD card even if that option was set. This means a maximum of every other data point could be logged, or data would be lost in case of anything but a clean soft shutdown of the system. Mathworks supplied custom code to fix this as well; the Mathworks support from the developers has been excellent.

Next big issue with the Simulink - Pixhawk interaction was a silent slowdown. Even when the Simulink app was set to, and reported, a specific refresh rate it would not agree with reality. Many weeks were spent tuning and calibrating the flying board for just hovering without much success. The board would barely hover, and sometimes without wind hover acceptably but any disturbances would easily ground it involuntarily. After too much time a step response signal was implemented and the steps made to be one second long. The first time it was tested the steps were 6 seconds long in reality, but the log files indicated correct sampling frequency and duration for the steps. Turns out that getting a dual rotor tail sitter to hover is very difficult at an effective 20Hz sampling frequency. Disabling all logging and timing the step responses externally with a stopwatch verified that the issue was still with the logging. The Mathworks supplied hard fault fix meant that the log files would be opened, written to and closed on every iteration of the control system. Opening and closing files is a very expensive task so the system was redesigned to log everything in one big file to be separated externally after the flight. That reduced the time dilation significantly but not enough. A reduced logging frequency got the time error within 1% and had

to suffice. This meant that the controller would run at the intended 120 Hz but only every third datapoint would be logged. Once this was fixed the flying board hovered nicely corrected itself much more smoothly.

All these time delays meant that very little time was left to the final testing and evaluation, but the data acquired and presented here do seem to indicate that what was set out to test is possible: A non switching controller can handle the entire flight envelope of a VTOL capable flying fixed wing when the control surface forces are normalized. To further improve this some wind tunnel tests and maybe airflow sensors on the control surfaces would help further understand how to tune this system, maybe even remove the tuning steps completely.

- SOMEWHERE explain exactly how to go from the [1,1,0.7] without torque translator to finding K parameters both ways. Maybe show substitute stuff?
- SOMEWHERE present the intended operating frequencies / sampling time
- Maybe add link to matlab/simulink code as reference?

## References

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [On the electrodynamics of moving bodies]. Annalen der Physik, 322(10):891–921, 1905.
- [3] Robert C Nelson *Flight Stability and Automatic Control [Second Edition]* ISBN: 0-07-066110-3
- [4] Emil Fresk and George Nikolakopoulos *Full Quaternion Based Attitude Control for a Quadrotor* 2013 European Control Conference (ECC) July 17-19, 2013, Zürich, Switzerland. 978-3-033-03962-9/©2013 EUCA
- [5] Robert E. Sheldahl, Paul C. Klimes *Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines* Sandia National Laboratories - energy report SAND80-2114 Unlimited Release UC-60
- [6] Roberto Naldi, Lorenzo Marconi *Optimal transition maneuvers for a class of V/STOL aircraft* Center for Research on Complex Automated Systems (CASY) Giuseppe Evangelisti, DEIS - Department of Electronic, Computer Science and Systems, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy 0005-1098 2011 Elsevier Ltd. doi:10.1016/j.automatica.2011.01.027
- [7] Joan Solà *Quaternion kinematics for the error-state KF* February 2, 2016
- [8] John Anderson *Fundamentals of Aerodynamics* Sixth Edition McGraw-Hill Education ISBN 978-1-259-12991-9
- [9] Ryan G. Beall June 2017 *ENGINEERING OF FAST AND ROBUST ADAPTIVE CONTROL FOR FIXED-WING UNMANNED AIRCRAFT* NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA
- [10] <https://se.mathworks.com/hardware-support/px4-autopilots.html>
- [11] [https://github.com/Troblot/MastersThesis\\_matlab](https://github.com/Troblot/MastersThesis_matlab)