

AA241X Problem Set 2

Apoorva Sharma, Justin Schneider, Miao Zhang, Victoria M Dax, Zhe Zhang and Zhengyu Huang

May 23, 2017



Contents

I Experimental determination of trim control inputs	3
I.1 Experimental results	3
I.2 Choosing control surface deflections and throttle setting	5
I.2.1 Effect of control surfaces	5
I.2.2 Settings	5
II Flight dynamics and performance estimates	6
II.1 Basic Data	8
II.2 Aerodynamic performance	9
II.3 Stability and Control	10
II.3.1 Stability derivative estimation	11
II.3.2 Stability Analysis	13
II.4 Propulsive model	19
II.4.1 Motor parameters	19
II.4.2 Motor/Propeller Matching	22
II.5 Power consumption model	26
II.5.1 level flight	26
II.5.2 Climb	27
III Mission Plan and Strategy: Simulations	29
III.1 Simulator Dynamics	30
III.2 Simulator Results: Simple Waypoint Following	31
III.3 Path Planning	33

IV Design, Construct and Fly Your Own Aircraft	34
IV.1 Dynamic Scaling	34
IV.2 Aircraft Design	36
IV.3 Aircraft Manufacturing	41
IV.4 Flight Testing	45
IV.5 Design Changes	47
V Closing control loops	48
V.1 Longitudinal Autopilot	49
V.2 Lateral Autopilot	50
V.3 Bixler3 Flight Tests	51
VI Goals and Plan of Action	54

I Experimental determination of trim control inputs

I.1 Experimental results

This section will show the result of trim control inputs based on the flight test of Bixler3 on 10th May. With wind speed 2.68224, Bixler3 can reach maximum speed of about 30m/s, when the throttle input keeps 0.85. Its stall speed for level flight is

$$V_{stall} = \sqrt{\frac{2W}{\rho Cl_{max}S}} = 6.5m/s$$

From 20 minutes' flight, we choose 6 flight paths in Fig 1, which has constant heading.

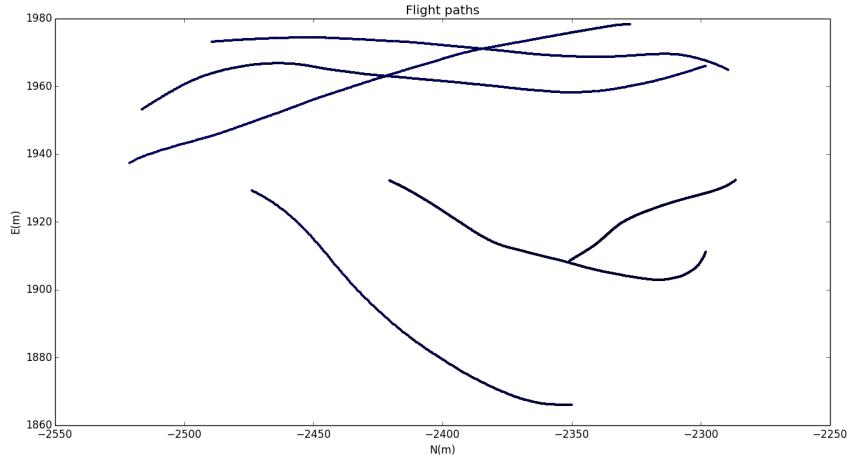


Figure 1: Bixler3 Model Trim flight paths

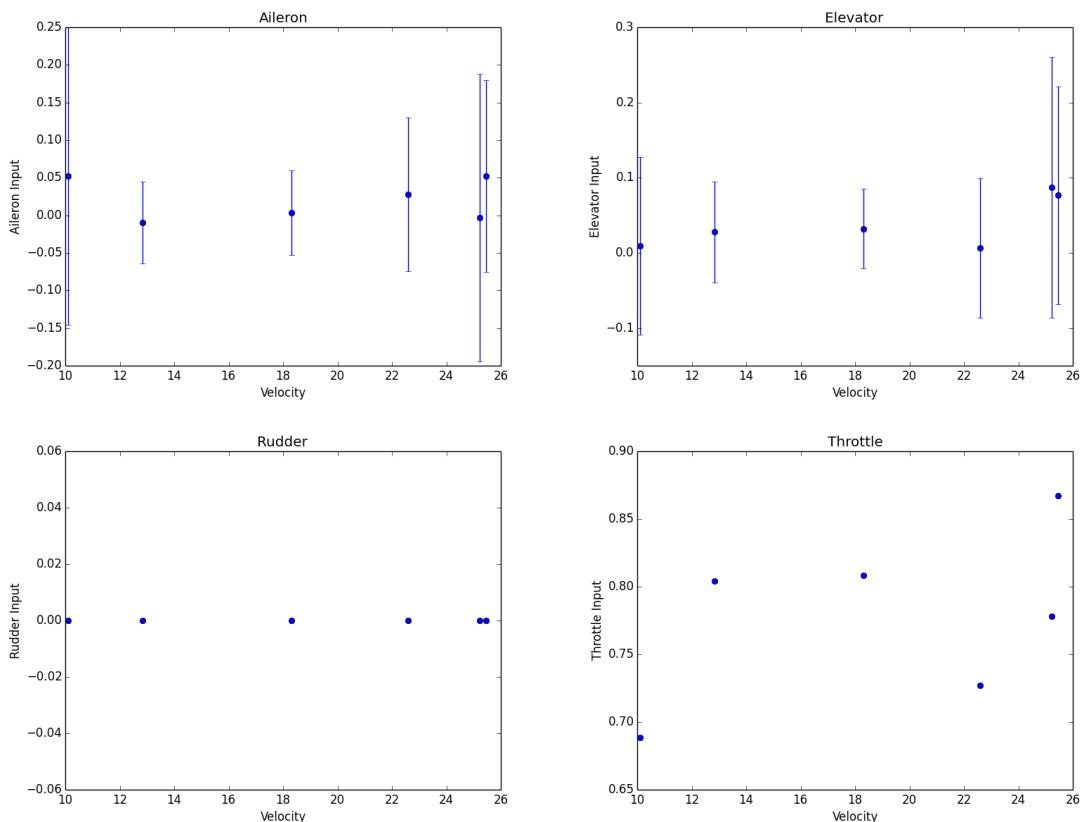


Figure 2: Bixler3 Model Trim control inputs

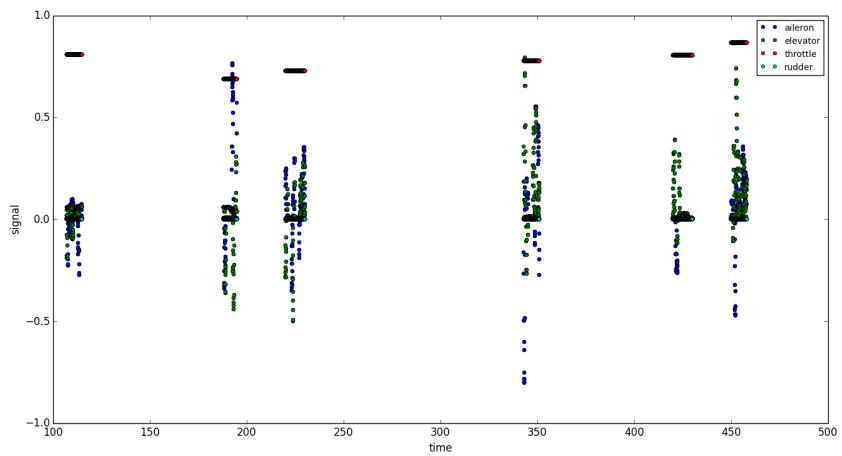


Figure 3: Bixler3 Model Trim flight inputs

I.2 Choosing control surface deflections and throttle setting

I.2.1 Effect of control surfaces

Ailerons

Ailerons are primarily used for roll control in order to perform a banking turn. When left aileron goes up and right aileron goes down, the lift on left wing is reduced while the lift on right wing is increased, the airplane will roll to the left. Similarly, when left aileron goes down and right aileron goes up, the airplane will roll to the right. However, rolling moment is not the only effect of aileron deflections, there will also be adverse yaw due to higher induced drag on the wing with increased lift.

Elevator

Elevator is primarily used for pitch control. When the elevator goes up, the tail lift goes down hence tail moves downward, and nose will pitch up, resulting in increased angle of attack, lift and induced drag. Similarly, when the elevator goes down, the nose will pitch down, leading to reduced angle of attack, lift and induced drag.

Rudder

Rudder is primarily used for yaw control. Left deflection of rudder causes the tail to move rightward, and nose yaw to the left. Right deflection of rudder causes the tail to move leftward, and nose yaw to the right. Moreover, if we continuously apply rudder deflection in the same direction, the lift on the wing opposite to yaw direction will increase due to higher airspeed while the lift on the other will drop due to lower airspeed. This will result in the airplane banking in the direction of yaw.

I.2.2 Settings

We need to decide the settings for flight at constant altitude and constant speed. The most important equations for this part are the lift/weight, thrust/drag relations.

$$\begin{cases} L = W = \frac{1}{2}\rho V^2 S C_L \\ T = D = \frac{1}{2}\rho V^2 S C_D = \frac{1}{2}\rho V^2 S (C_{D_0L} + K C_L^2) \end{cases}$$

Since the settings depend heavily on the weather, primarily on wind speed and direction, it is better to discuss in two cases.

No wind

Without wind, there is no need to use roll or yaw control. After the aircraft climbs to the desired altitude, the density is fixed assuming constant temperature and humidity at that altitude. Then we can set the throttle so that the flight speed matches the desired speed. After that, to stay level flight, the C_L is to be adjusted through elevator deflections. However, the adjustment in the C_L leads to changes in C_D , hence a slight change in the throttle is needed to balance the new drag.

Windy

On a windy day (which is very possible), the settings will be more complicated. For simplicity, we assume the wind direction and speed are not continually changing.

If the airplane is flying parallel to the wind direction, then this is essentially the same as the no wind condition, and all we need to do is to subtract/add the wind speed from/to the desired speed.

If the wind induces a net pitching moment, then we need to adjust the elevator to counteract the pitching up/down of the nose.

If the wind induces a net yawing moment, then we need to adjust the rudder to counteract that moment. Moreover, since we are assuming constant wind, the application of rudder deflection should be continuous, leading to banking in the yaw direction. We therefore need some aileron deflections to cancel that banking effect.

II Flight dynamics and performance estimates

This section will introduce our airplane Albatross, our goal is to match the dynamical scaling of actual Mars airplane. The basic parameters for Mars airplane are

$$\begin{aligned} \rho &= 0.015 \text{kg/m}^3 & m &= 2.3 \text{kg} & S &= 2.94 \text{m}^2 & c &= 0.7 \text{m} \\ AR &= 6 & \mu_s &= \frac{2m}{\rho Sc} = 149.01 & C_l &= \frac{2mg}{\rho u^2 S} = 0.9499 \end{aligned}$$

The temperature on Mars varies from -153°C at the poles to 20°C at noon at the equator. By using a common value of -55°C , and approximate the atmosphere as 100% CO_2 instead of 95% CO_2 (to simplify calculation), we can get the dynamic viscosity of $\mu = 1.1055271 \times 10^{-5} \text{kg/(m - s)}$. Dividing it by the density, we can get the kinematic viscosity of $\nu = 8.8513 \times 10^{-4} \text{m}^2/\text{s}$.

For viscosity in other conditions, we can use this calculator (<http://www.lmnoeng.com/Flow/GasViscosity.php>)

For reference, below is a list of Reynolds numbers on Mars according to given chord length and flight velocity:

$$\begin{aligned} u = 15m/s & \quad c = 0.15m \quad Re = \frac{\rho cu}{\mu} = 2542 \\ u = 20m/s & \quad c = 0.15m \quad Re = \frac{\rho cu}{\mu} = 3389 \\ u = 22m/s & \quad c = 0.7m \quad Re = \frac{\rho cu}{\mu} = 1.74 \times 10^4 \end{aligned}$$

The gas constant for CO_2 is $R_{CO_2} = 188.9J/(kg - K)$, the ratio of specific heats for CO_2 is $\gamma_{CO_2} = 1.28$. An estimate of speed of sound on Mars using is: $a = \sqrt{\gamma RT} = 229.667m/s$

To match the Mars airplane parameters, the chord length c , velocity u and mass m should satisfy the relations

$$\frac{m}{c^3} = 658.20586 \quad \frac{m}{u^2 c^2} = 0.3564957 \quad \frac{c}{u^2} = 0.000542$$

For reference, below is a list of different chord length and mass corresponding to different velocity

$$\begin{aligned} u = 15m/s & \quad c = 0.1219m \quad Re = \frac{\rho cu}{\mu} = 1.23 \times 10^5 \\ u = 20m/s & \quad c = 0.216646m \quad Re = \frac{\rho cu}{\mu} = 2.9 \times 10^5 \end{aligned}$$

The main wing of Bixler3 is NACA2412, and we choose FX 63-137 Smooth as the main wing of Albatross to achieve high lift. We use XFLR5 to build aerodynamic model for both the Bixler3 and our Albatross. Fig 4 is Bixler3 model and Fig 5 is Albatross model

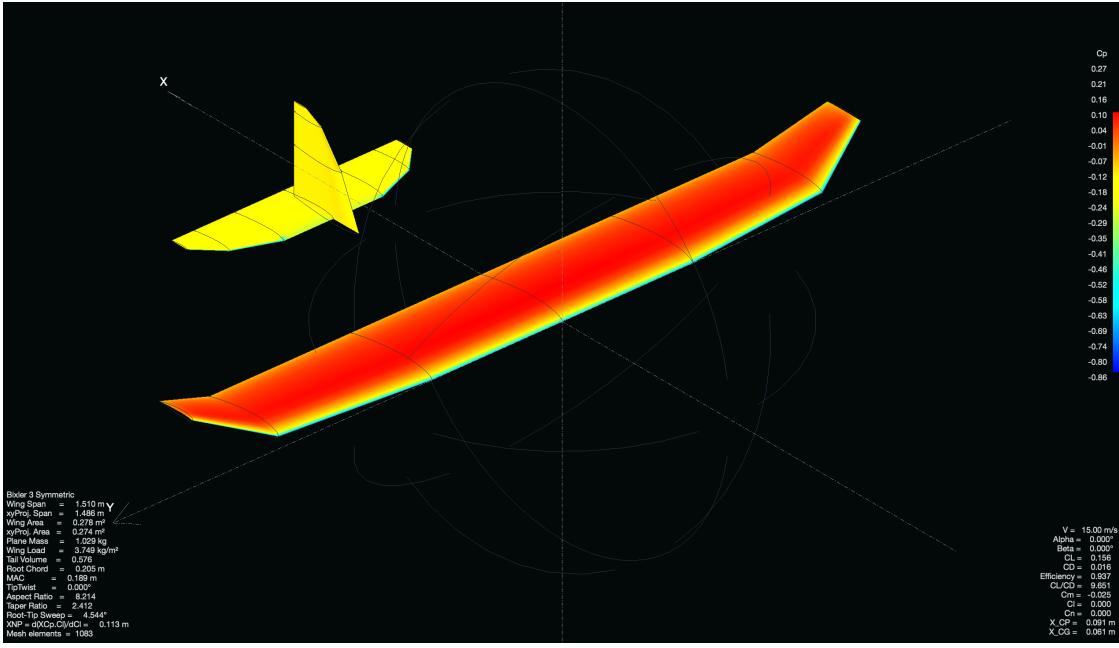


Figure 4: Bixler3 Model

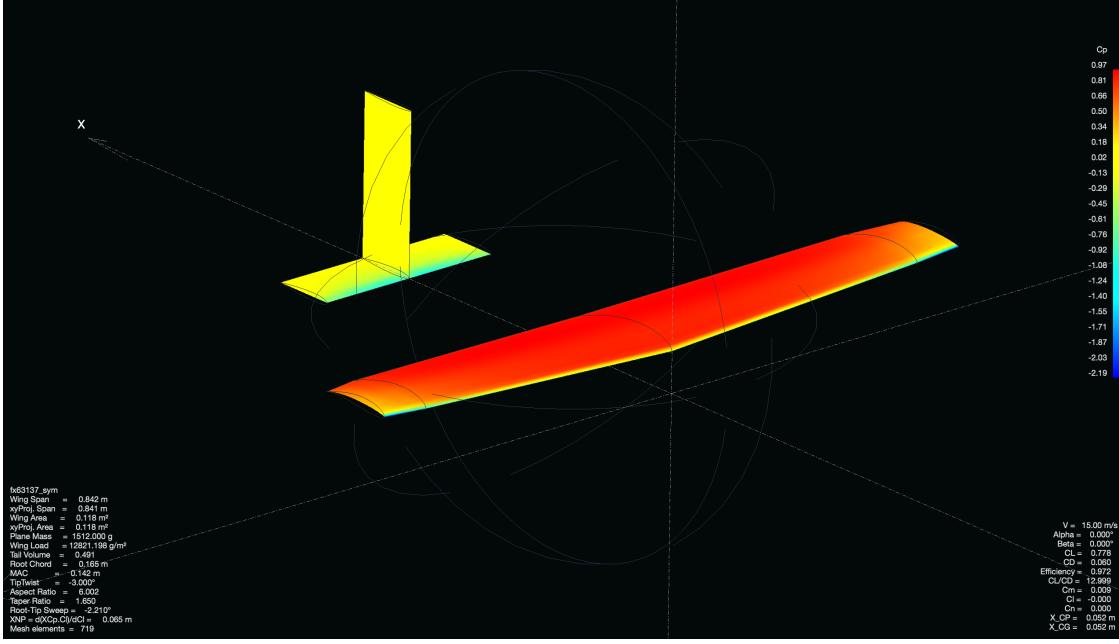


Figure 5: Albatross Model

II.1 Basic Data

Table 1: Inertias in the body axis

	Bixler3($g \cdot m^2$)	Albatross ($g \cdot m^2$)
I _{xx}	51.37	12.4
I _{yy}	53.91	24.87
I _{zz}	106.29	35.29
I _{xz}	2.55	-1.282

Table 2: Weight estimates of Bixler3 and Albatross

	Bixler3(g)	Albatross(g)
Main Wing	334	214
Vertical	40	35
Horizontal	80	58
Fuselage	229	192
Battery	174	174
GPS & Pixhawk	150	150
left Servo	11	11
right Servo	11	11
Total	1029	845

II.2 Aerodynamic performance

Fig 6 shows the lift coefficients vs. angle of attack at a velocity of 15m/s for Bixler3 and Albatross. Albatross has $C_l = 0.8$ even at zero angle of attack and can achieve $C_{lmax} = 1.6$ at about 10°

Figure 6: Bixler3 Model vs Albatross at 15m/s

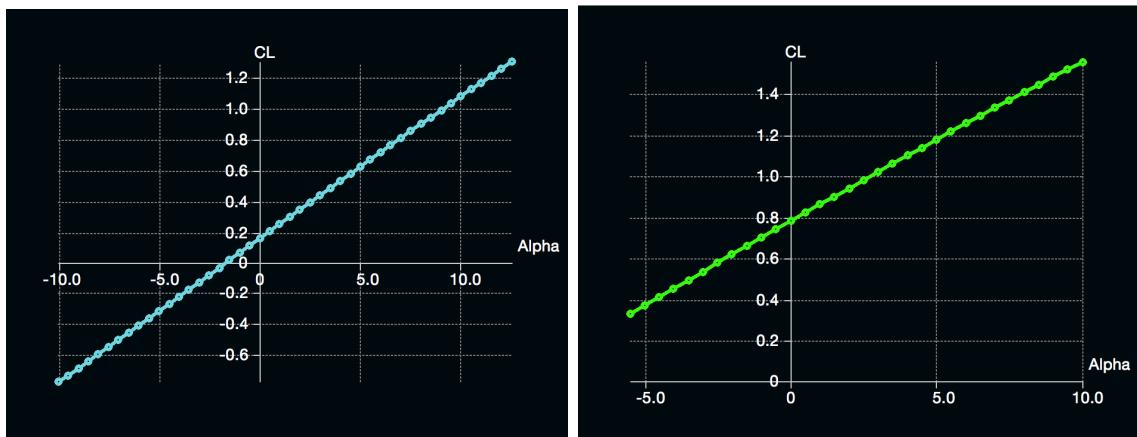


Fig 7 shows the lift coefficients vs. drag coefficients at a velocity of 15m/s for Bixler3 and Albatross. Albatross uses high lift wing, it also has larger drag coefficients.

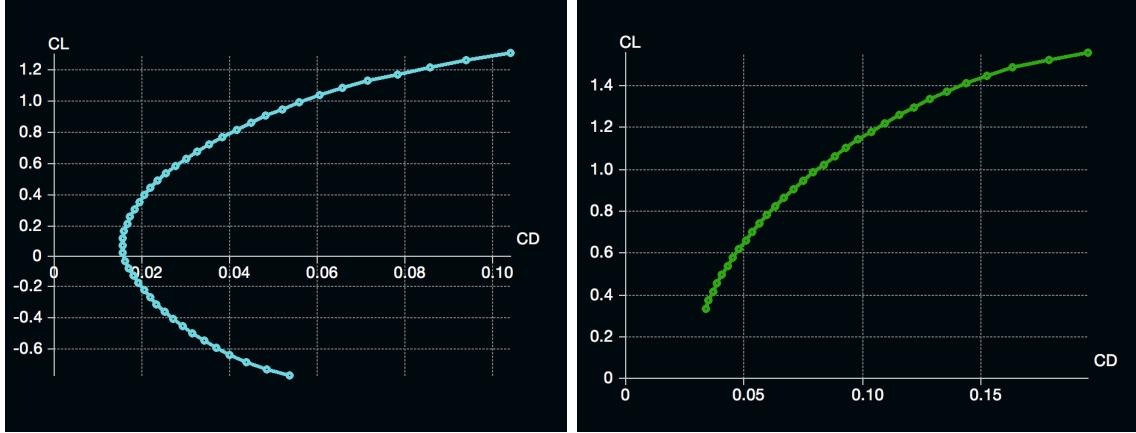


Figure 7: Bixler3 Model vs Albatross at 15m/s

Fig 8 shows the lift drag ratio vs. airspeed for level flight. Albatross has stall speed of 11m/s and Bixler3 has stall speed of 6m/s. Maximum lift drag ratio for Bixler3 is about 18 at 10m/s and for Albatross is about 13.5 at 17.5m/s.

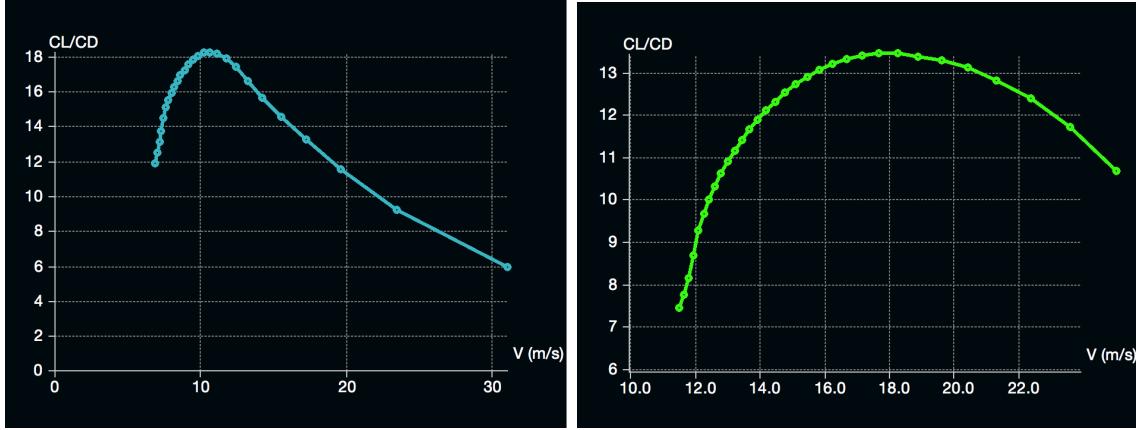


Figure 8: Bixler3 Model vs Albatross at constant lift

II.3 Stability and Control

In order to be able to effectively compare them: We will refer to flight testings at similar (almost equal) speeds and angles of attack (in future referred to as AOA or α) in stable level flight condition.

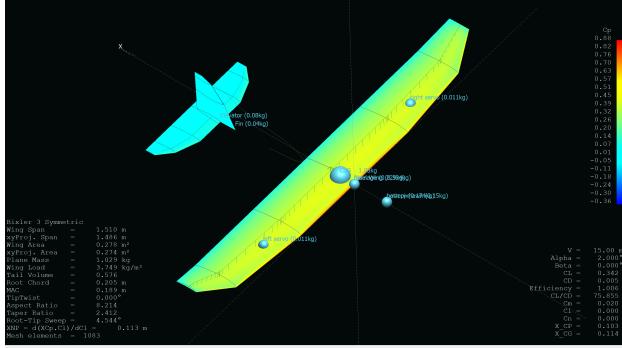


Figure 9: Bixler3 Analysis Results at AOA= 2°

We build a model of the Bixler 3 in XFLR5, taking into account the exotic weight distribution, i.e. placement of the battery and the other software equipment. The main wing is modeled with a NACA 2412 with a max thickness of 22mm which refers to 12% thickness to chord ratio. The tail is modeled with at NACA 0012.

Note that we would achieve the desired Cl of 0.8 at an AOA of $\alpha = 7^\circ$

For the stability and control parameters we are going to determine them experimentally and with the analytical simulation via XFLR5. For further comparison we might want to use formally listed values as well if our results seem tremendously inaccurate.

II.3.1 Stability derivative estimation

Both Bixler3 and “Albatross” are modeled in XFLR5 and the stability derivatives are estimated in the software and displayed in the tables below.

Table 3: Longitudinal Stability Derivatives

	Bixler 3	Albatross
C_{xu}	-0.016953	-0.070928
$C_{x\alpha}$	0.086425	0.43552
C_{zu}	-0.0017619	-0.00050627
C_{za}	-5.2892	-4.6105
C_{zq}	-9.6172	-5.2513
C_{mu}	$5.3238 \cdot 10^{-6}$	9.514910^{-6}
C_{ma}	-1.4596	-0.74857
C_{mq}	-15.033	-9.9449

Table 4: Lateral Stability Parameters

	Bixler3	Albatross
$C_{y\beta}$	-0.28032	-0.51378
C_{yp}	-0.1467	-0.11347
C_{yr}	0.19605	0.57569
C_{lb}	-0.066003	-0.056539
C_{lp}	-0.50158	-0.4459
C_{lr}	0.027572	0.24909
C_{nb}	0.082813	0.24473
C_{np}	-0.0056912	-0.10094
C_{nr}	-0.063759	-0.27193

XFLR5 does not output the control derivatives without the control surfaces modeled, thus the control derivatives were estimated from the flight test data around one trimmed condition $\theta_0 = 0$, $U_0(\text{Bixler3}) = 11.47[\text{m/s}]$. The control derivative was only estimated for the Bixler3 since no flight data were available for “Albatross”. Fig. 10 plotted the flight data used for lateral control derivative estimation.

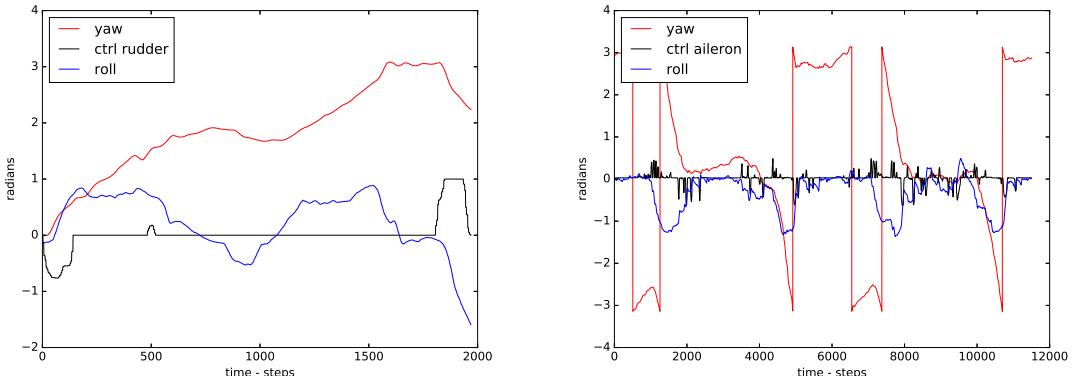


Figure 10: Lateral control flight data

As shown in Fig.10, the control sequences used and the variations of yaw and roll command and the corresponding yaw and roll rate are recorded. The corresponding derivatives were then numerically computed, all shown in the table below.

Table 5: Control derivatives for Bixler3

$C_{x\delta_e}$	0.088	$C_{y\delta_a}$	-0.004288
$C_{x\delta_T}$	0.48	$C_{y\delta_r}$	0.02451
$C_{z\delta_e}$	-0.420	$C_{l\delta_a}$	0.04666
$C_{z\delta_T}$	-0.172	$C_{l\delta_r}$	0.13942
$C_{m\delta_e}$	0.086	$C_{n\delta_a}$	-0.057383
$C_{m\delta_T}$	-0.463	$C_{n\delta_r}$	-0.0121

II.3.2 Stability Analysis

Stability analysis was done on both Bixler3 and designed aircraft “Albatross”. Using the stability derivatives derived above, the equation of motion for both longitudinal and lateral motion at trimmed flight condition $\theta_0 = 0$, $U_0(\text{Albatross}) = 15.95[\text{m/s}]$, $U_0(\text{Bixler3}) = 11.47[\text{m/s}]$ can be expressed as following:

$$\begin{bmatrix} \frac{mU_0}{Sq_\infty} & -\frac{c}{2U_0}C_{x\dot{\alpha}} & 0 & 0 \\ 0 & \frac{mU_0}{Sq_\infty} - \frac{c}{2U_0}C_{z\dot{\alpha}} & 0 & 0 \\ 0 & -\frac{c}{2U_0}C_{m\dot{\alpha}} & \frac{I_{yy}}{Sq_\infty c} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} C_{xu} & C_{x\alpha} & \frac{c}{2U_0}C_{xq} & C_w \cos(\theta_0) \\ C_{zu} & C_{z\alpha} & \frac{mU_0}{Sq_\infty} - \frac{c}{2U_0}C_{zq} & C_w \sin(\theta_0) \\ C_{mu} & C_{m\alpha} & \frac{c}{2U_0}C_{mq} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix}$$

$$+ \begin{bmatrix} C_{x\delta_e} & C_{x\delta_T} \\ C_{z\delta_e} & C_{z\delta_T} \\ C_{m\delta_e} & C_{m\delta_T} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix}$$

$$\begin{bmatrix} \frac{mU_0}{Sq_\infty} & 0 & 0 & 0 & 0 \\ 0 & \frac{I_{xx}}{Sq_\infty b} & -\frac{I_{xz}}{Sq_\infty b} & 0 & 0 \\ 0 & -\frac{I_{xz}}{Sq_\infty b} & \frac{I_{zz}}{Sq_\infty b} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} C_{y\beta} & \frac{b}{2U_0}C_{yp} & \frac{b}{2U_0}C_{yr} - \frac{mU_0}{Sq_\infty} & C_{y\phi} & C_{y\psi} \\ C_{l\beta} & \frac{b}{2U_0}C_{lp} & \frac{b}{2U_0}C_{lr} & 0 & 0 \\ C_{n\beta} & \frac{b}{2U_0}C_{np} & \frac{b}{2U_0}C_{nr} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix}$$

$$+ \begin{bmatrix} C_{y\delta_a} & C_{y\delta_r} \\ C_{l\delta_a} & C_{l\delta_r} \\ C_{n\delta_a} & C_{n\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

$C_{x\dot{\alpha}}$, $C_{z\dot{\alpha}}$ for both planes were estimated to be 0, and $C_{m\dot{\alpha}}$ was estimated as -3 as a

typical value for conventional aircraft configuration. $C_{y\phi} = \frac{mg}{Sg_\infty} \cos(\theta_0)$, $C_{y\psi} = \frac{mg}{Sg_\infty} \sin(\theta_0)$. The equations were then rearranged in to linearized form $\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u}$. The control derivative (coefficients in matrix \mathbf{B}) of the Bixler was estimated from flight test as shown in previous section. Without flight test experiment, the control derivatives for the “Albatross” were undetermined. Therefore, for “Albatross” the stability was analyzed without taking control into account. For longitudinal stability analysis, the simulated response for important longitudinal states are monitored given a disturbance. With the knowledge of control effect of the Bixler3 from flight data, the disturbance used for simulation is an impulse in elevator deflection, while the disturbance used for “Albatross” is an initial $\alpha = 2^\circ$. The behavior of short period and phugoid modes are noted on the plots.

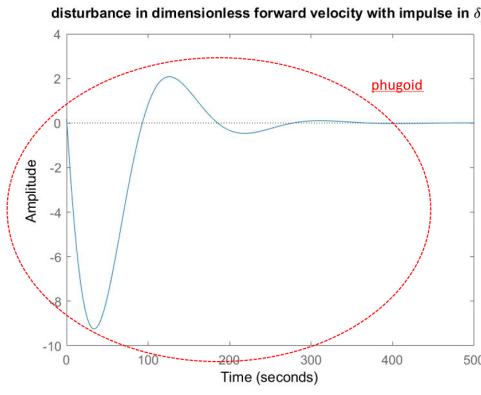


Figure 11: dimensionless forward velocity response for Bixler3

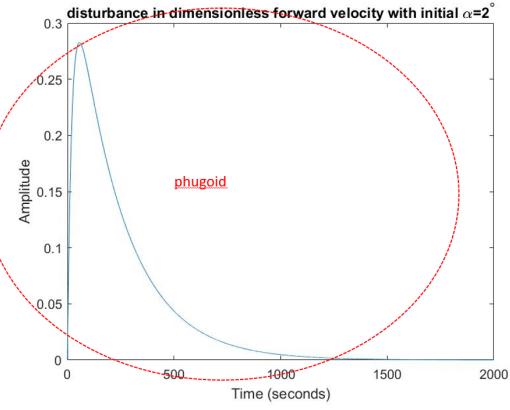


Figure 12: dimensionless forward velocity response for Albatross

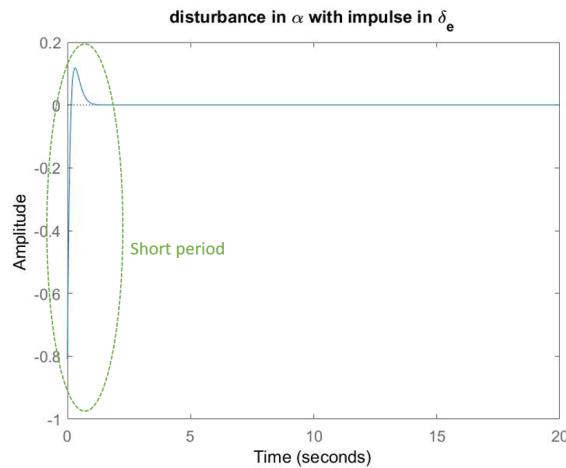


Figure 13: α response for Bixler3

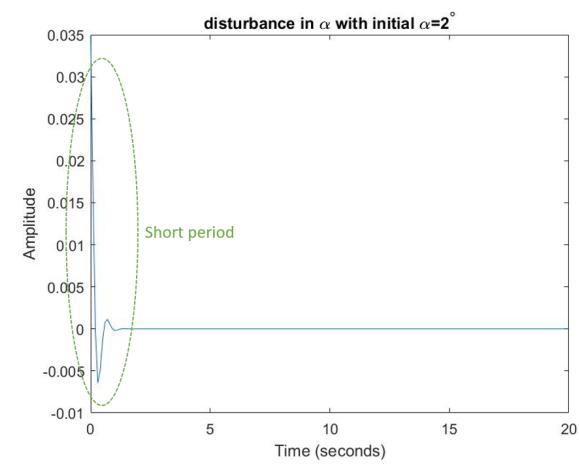


Figure 14: α response for Albatross

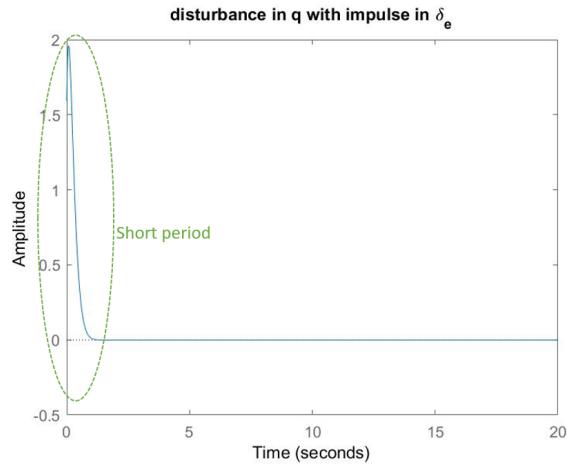


Figure 15: q response for Bixler3

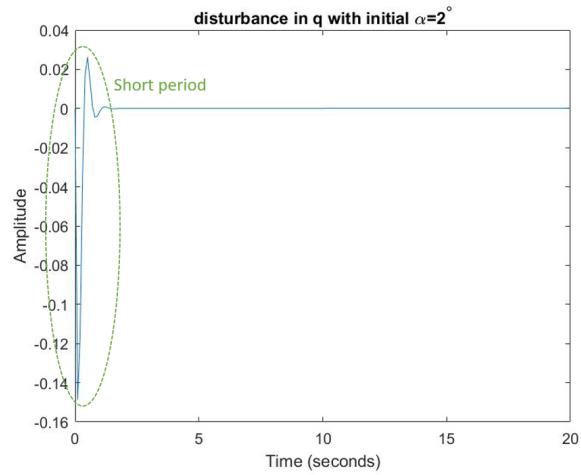


Figure 16: q response for Albatross

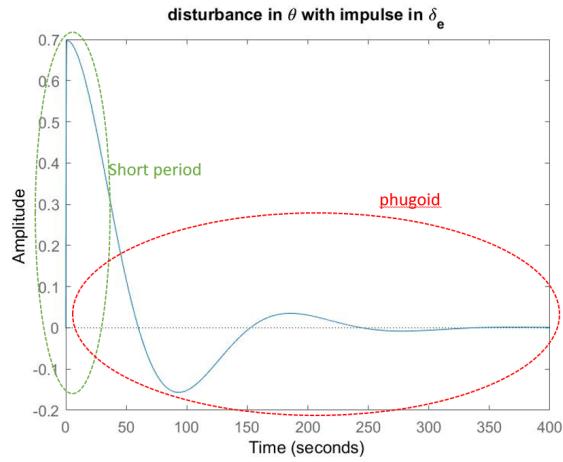


Figure 17: θ response for Bixler3

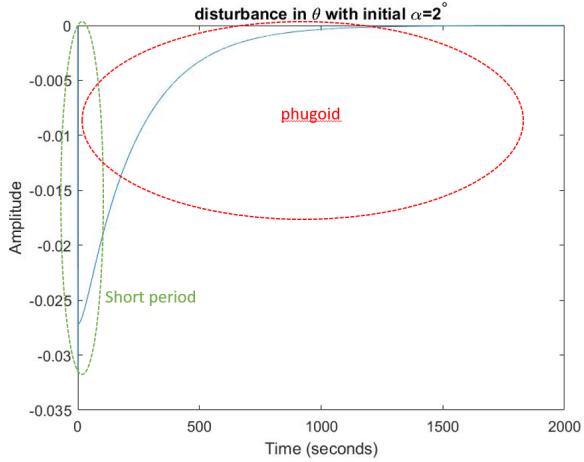


Figure 18: θ response for Albatross

As shown in the figures above, the response for all important longitudinal states are stable, and short period and phugoid behaviors were properly reflected in corresponding states. Table 13 shows a more quantitative description of the longitudinal dynamic modes for the two aircrafts.

Table 6: Longitudinal dynamic modes parameters

		Bixler3	Albatross
$\omega_{S.P.}$	short period mode natural frequency	$6.65[\frac{rad}{\tau}]$	$10.1[\frac{rad}{\tau}]$
$\tau_{S.P.}$	short period mode time constant	0.16[s]	0.205[s]
$\zeta_{S.P.}$	short period mode damping ratio	0.939	0.483
$\omega_{phugoid}$	phugoid mode natural frequency	$3.77 \times 10^{-2}[\frac{rad}{\tau}]$	$5.05 \times 10^{-2}[\frac{rad}{\tau}]$
$\tau_{phugoid}$	phugoid mode time constant	61.9[s]	123.4[s]
$\zeta_{phugoid}$	phugoid mode damping ratio	0.429	0.5

For lateral stability analysis, the disturbance introduced to both aircraft is an initial sideslip $\beta = 2^\circ$. The response for important lateral states are shown in the following figures.

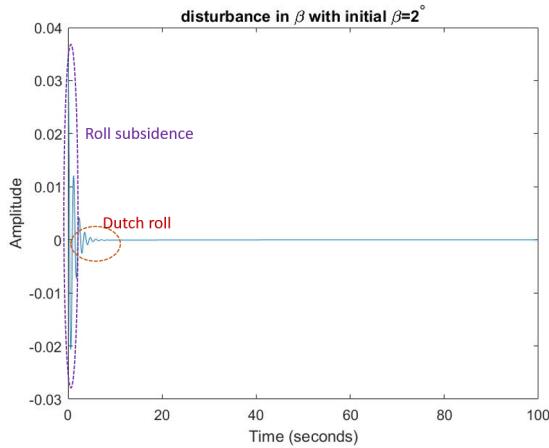


Figure 19: β response for Bixler3

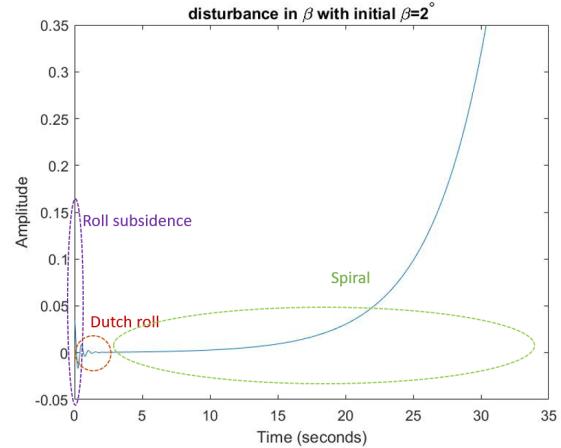


Figure 20: β response for Albatross

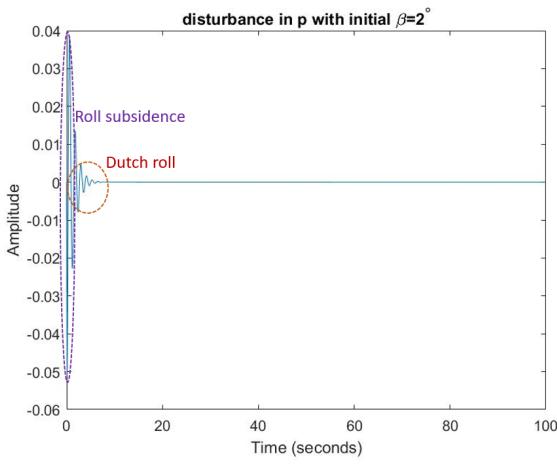


Figure 21: p response for Bixler3

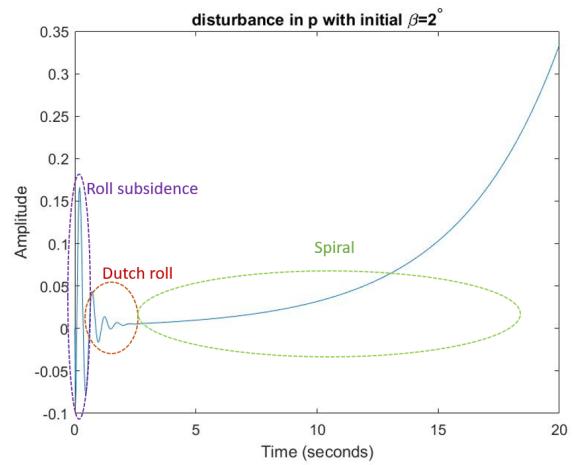


Figure 22: p response for Albatross

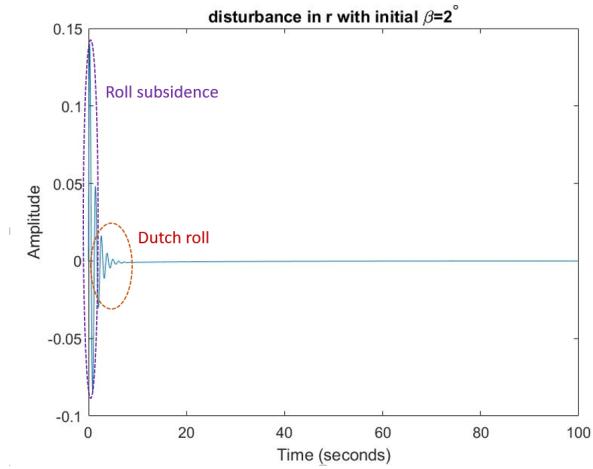


Figure 23: r response for Bixler3

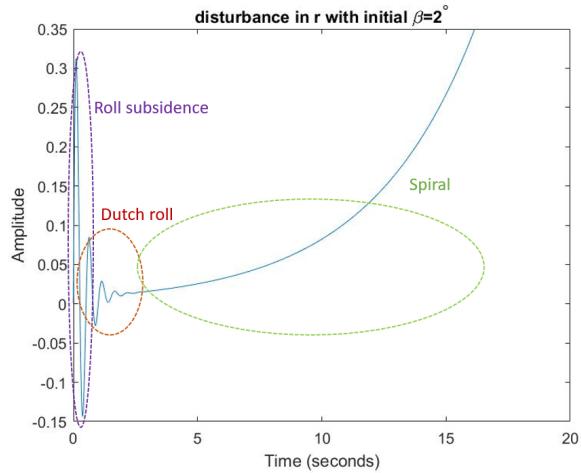


Figure 24: r response for Albatross

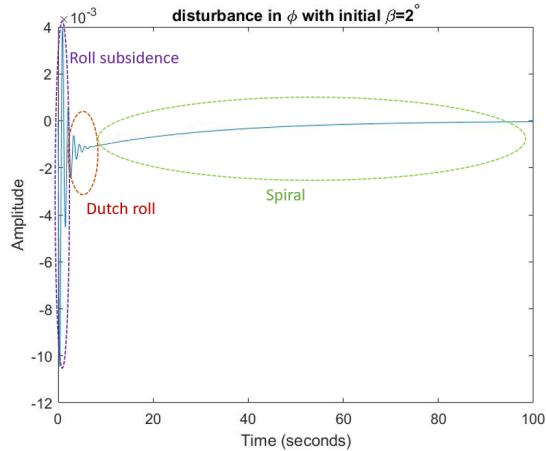


Figure 25: ϕ response for Bixler3

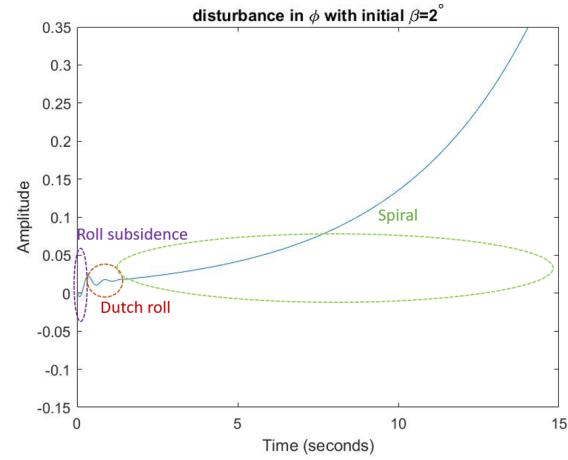


Figure 26: ϕ response for Albatross

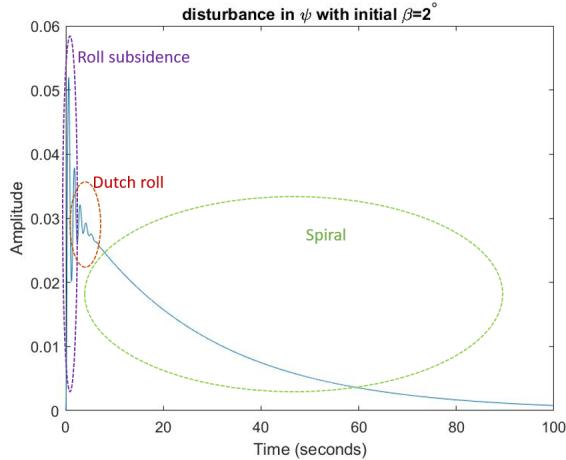


Figure 27: ψ response for Bixler3

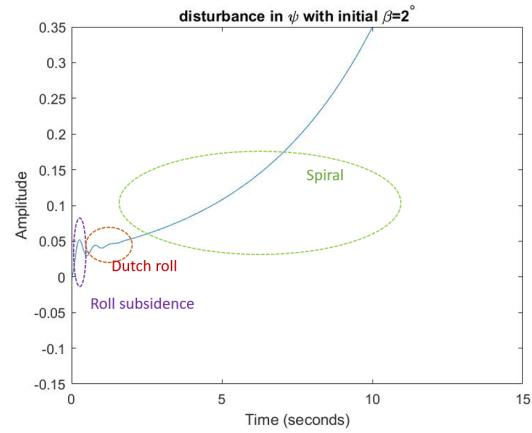


Figure 28: ψ response for Albatross

The major difference of the “Albatross” comparing to Bixler3 is that the “Albatross” has an unstable spiral mode. Based on the plots above, the fastest divergence displays in yaw angle ψ , which would diverge 5° after letting go of the control for 5 seconds, which might require a lot of control attention. The specific parameters of the lateral dynamic modes are shown in Table 7.

Table 7: Lateral dynamic modes parameters

	Bixler3	Albatross
$\omega_{D.R.}$	Dutch roll mode natural frequency	$5.4[\frac{rad}{\tau}]$
$\tau_{D.R.}$	Dutch roll mode time constant	1.12[s]
$\zeta_{D.R.}$	Dutch roll mode damping ratio	0.165
ω_{spiral}	spiral mode natural frequency	$3.72 \times 10^{-2}[\frac{rad}{\tau}]$
τ_{spiral}	spiral mode time constant	26.9[s]
ζ_{spiral}	spiral mode damping ratio	1
ω_{roll}	roll mode natural frequency	$23.5[\frac{rad}{\tau}]$
τ_{roll}	roll mode time constant	$4.26 \times 10^{-2}[s]$
ζ_{roll}	roll mode damping ratio	1

As shown in Table 7, “Albatross” has higher Dutch roll frequency, but higher Dutch roll damping ratio. “Albatross” has faster spiral mode, and unstable comparing to the Bixler. The roll subsidence mode is very comparable for the two airplanes. Comparing the time constant of the spiral mode of “Albatross” with typical aircraft with unstable spiral mode, it is about 8 times shorter, which means “Albatross” requires shorter control response time, which may need some extra caution. The simulation was also ran when the speed is $10[m/s]$,

this is slower than cruising speed and might happen during take-off and landing. The natural frequencies and damping ratio at this speed is also calculated for both planes, as shown in Table 8.

Table 8: ω_n and ζ at a different speed

Bixler3			Albatross		
$V = 10[m/s]$	$\omega_{D.R.}$	3.78[rad/s]	$V = 10[m/s]$	$\omega_{D.R.}$	16.98[rad/s]
	$\zeta_{D.R.}$	0.184		$\zeta_{D.R.}$	0.325
	$\omega_{phugoid}$	0.058 [rad/s]		$\omega_{phugoid}$	0.056[rad/s]
$V = 11.47[m/s]$	$\zeta_{phugoid}$	0.09	$V = 15.95[m/s]$	$\zeta_{phugoid}$	0.002
	$\omega_{D.R.}$	4.82[rad/s]		$\omega_{D.R.}$	33.97[rad/s]
	$\zeta_{D.R.}$	0.165		$\zeta_{D.R.}$	0.218
	$\omega_{phugoid}$	$6.1 \times 10^{-4}[\text{rad/s}]$		$\omega_{phugoid}$	$4.09 \times 10^{-4}[\text{rad/s}]$
	$\zeta_{phugoid}$	0.429		$\zeta_{phugoid}$	0.5

The modes are still reasonably stable at a lower speed for both aircrafts, however phugoid mode became faster and lightly damped, which suggests that we might need to pay attention to the phugoid mode during take off and landing.

II.4 Propulsive model

II.4.1 Motor parameters

The motor for Albatross is Turnigy D2836/9 950KV Brushless Outrunner Motor, the no load current is $i_0 = 1A$ and internal resistance is $R = 0.070\Omega$. and the RPM is 950Kv. We can compute motor speed constant as

$$Kv = \frac{950}{1Volt - i_0R} = 1021.50 \frac{rpm}{Volt}$$

The motor for Bixler3 is 2620-1400kv Brushless Outrunner, the no load current is $i_0 = 1A$ and internal resistance is $R = 0.070\Omega$. and the RPM is 1400Kv. We can compute motor speed constant as

$$Kv = \frac{1400}{1Volt - i_0R} = 1505.38 \frac{Rpm}{Volt}$$

The torque power, efficiency vs. rotation rate are computed as

$$Q_m(\Omega, v) = [(v - \frac{\Omega}{K_v})\frac{1}{R} - i_0]\frac{1}{K_v}$$

$$P_{shaft}(\Omega, v) = [(v - \frac{\Omega}{K_v})\frac{\Omega}{R} - i_0]\frac{1}{K_v}$$

$$\eta_m(\Omega, v) = [1 - \frac{i_0 R}{v - \Omega/K_v}]\frac{\Omega}{v K_v}$$

and showed in Fig 29 30 31

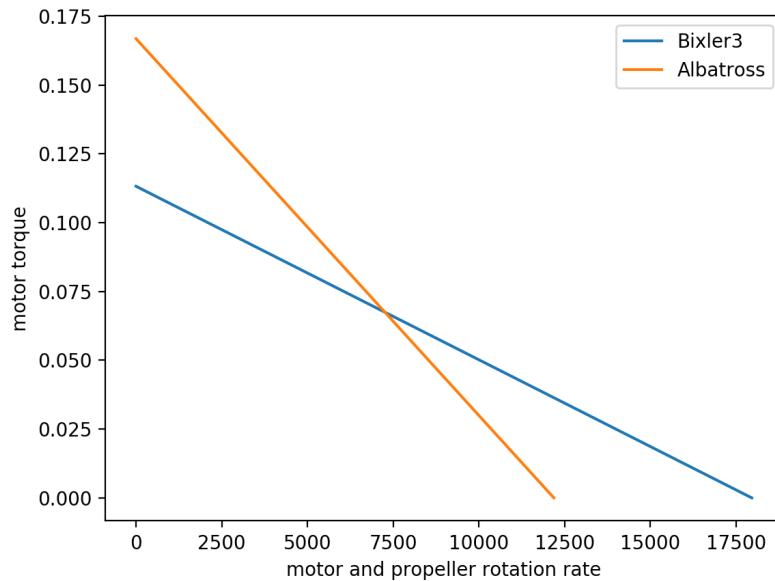


Figure 29: Torque vs. rotation rate at V=12Volt

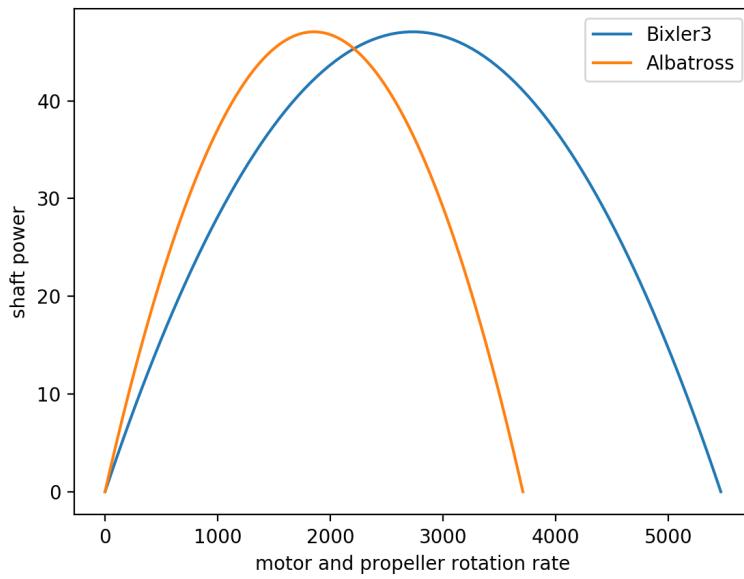


Figure 30: Power vs. rotation rate at V=12Volt

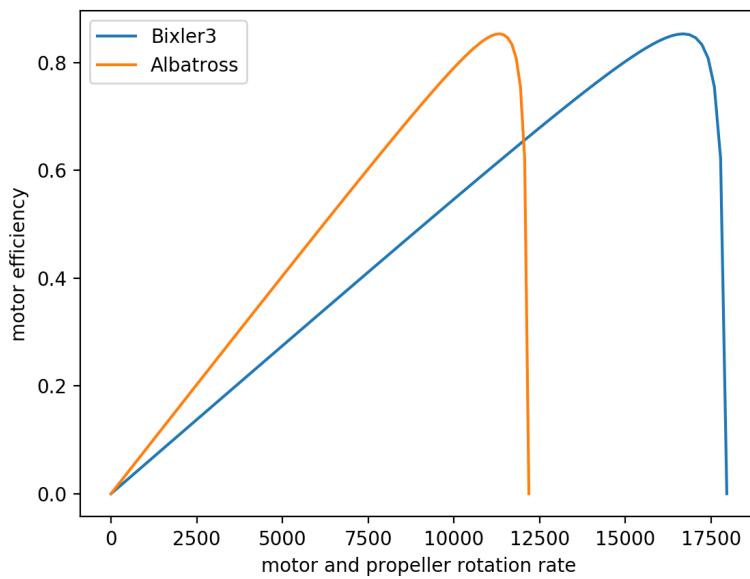


Figure 31: Efficiency vs. rotation rate at V=12Volt

The motor for Albatross has lower rotation rate compared with that of Bixler3. The motor of Albatross is most efficient at rotation rate about 10000 rpm, and the motor of

Bixler3's motor reaches maximum efficiency at 16000 rpm.

II.4.2 Motor/Propeller Matching

After carrying out some calculations, we can get the following plots:

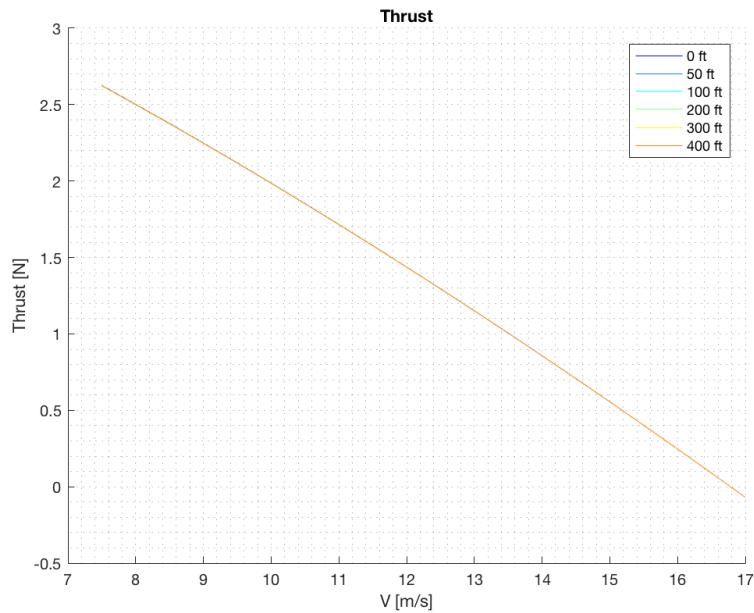


Figure 32: Thrust vs. Velocity

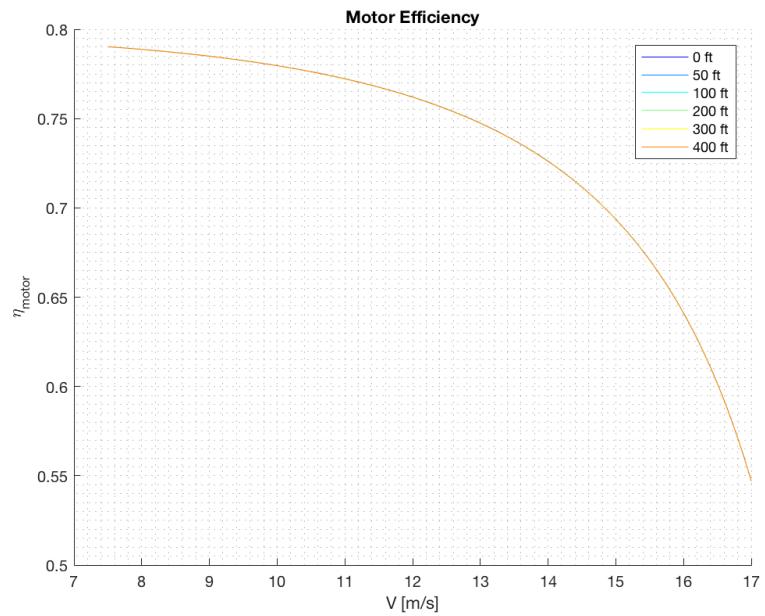


Figure 33: Motor Efficiency vs. Velocity

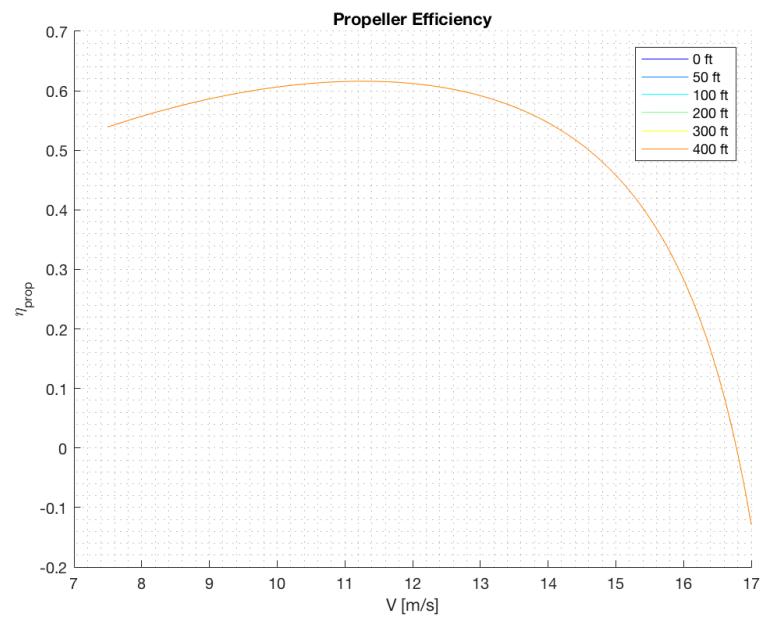


Figure 34: Propeller Efficiency vs. Velocity

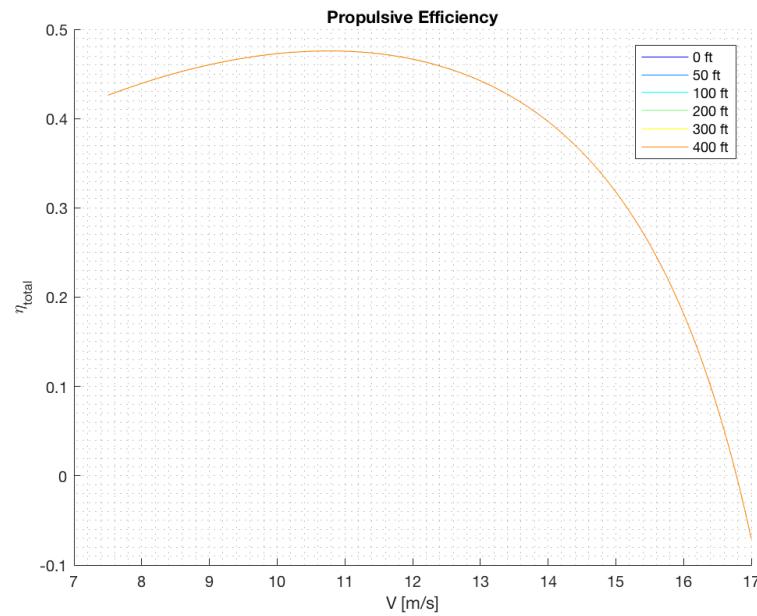


Figure 35: Propulsive Efficiency vs. Velocity

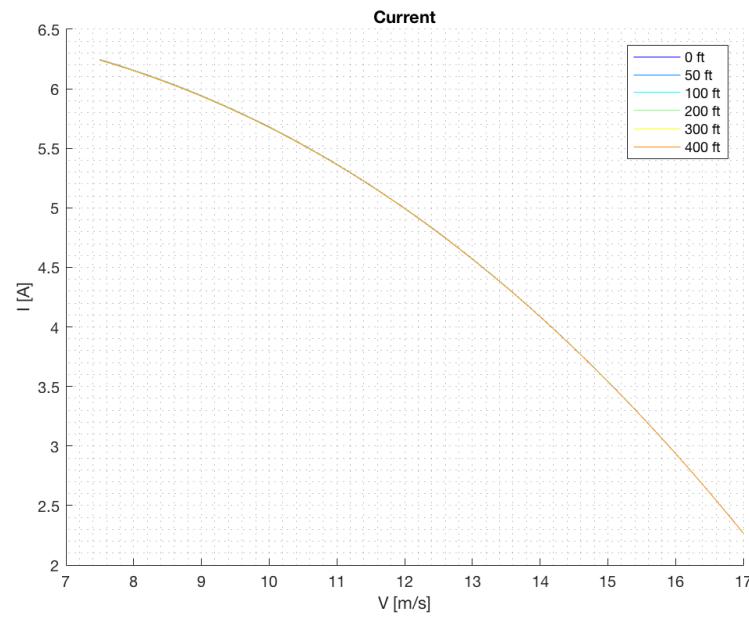


Figure 36: Current vs. Velocity

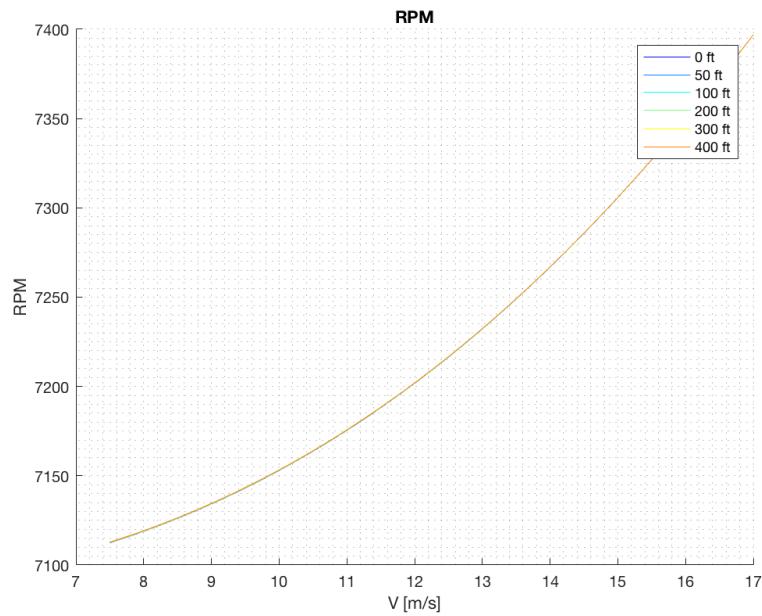


Figure 37: RPM vs. Velocity

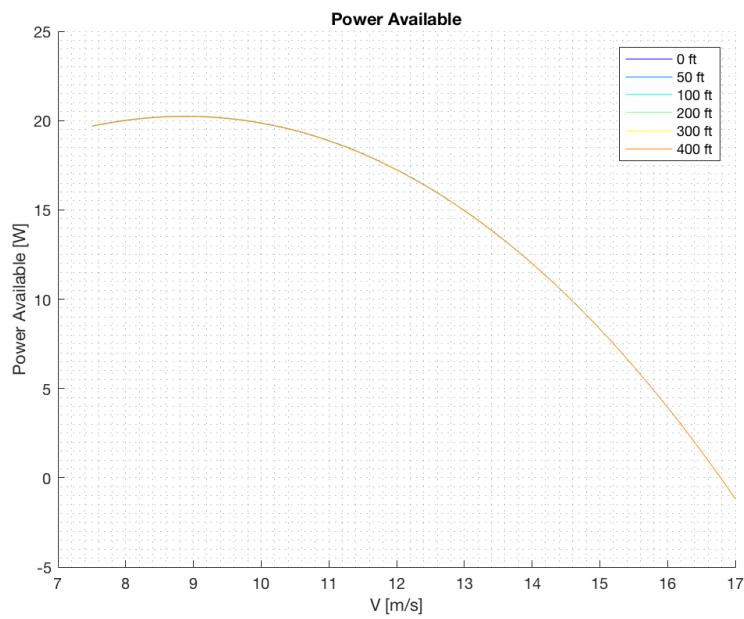


Figure 38: Power Available vs. Velocity

II.5 Power consumption model

II.5.1 level flight

For steady level flight, the thrust is approximately equal to the drag D, the power required is

$$P_{\text{required}} = DV = \frac{1}{2}\rho C_{D_0} V^3 S + \frac{W^2}{\frac{1}{2}\rho VS} \left(\frac{1}{\pi e R} \right)$$

The velocity for minimum power is:

$$V_{\text{min power}} = \left[\frac{4}{3} \left(\frac{W}{S} \right)^2 \frac{1}{\rho^2} \frac{1}{C_{D_0}} \left(\frac{1}{\pi e R} \right) \right]^{\frac{1}{4}}$$

To keep level flight, the drag coefficient goes as:

$$C_D = C_{D_{0L}} + KC_L^2$$

Using least-square method, we can estimate the coefficients using the XFLR5 simulation data:

$$C_{D_{0L}} = 0.0055, \quad K = 0.0737.$$

The velocity vs. drag coefficient is in Fig 42

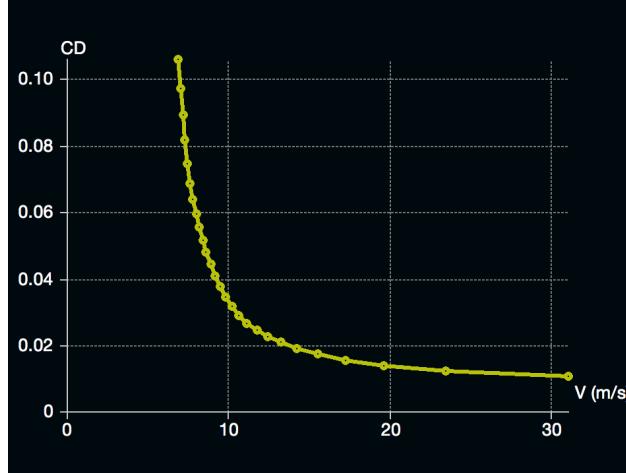


Figure 39: Bixler3 level flight drag coefficient vs. velocity

The power consumption plot is

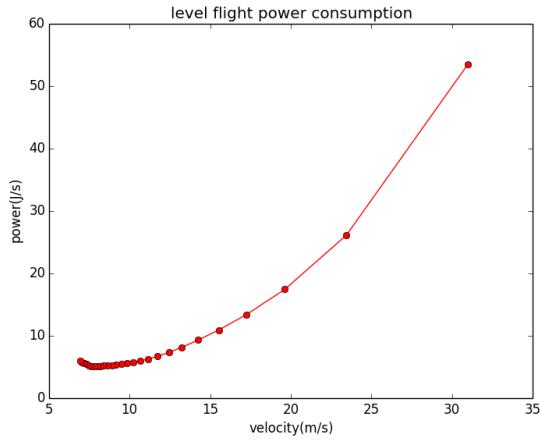


Figure 40: Bixler3 level flight required power

The power consumption for level flight is about 12 W at speed 16m/s, which is reasonable for RC planes.

II.5.2 Climb

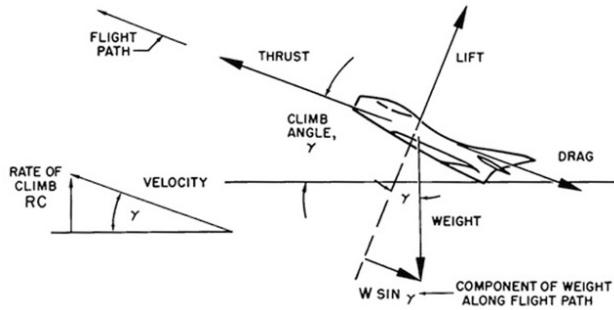


Figure 41: Climbing Aircraft

Given altitude h and velocity V of the aircraft, the energy is given by:

$$\varepsilon = \frac{1}{2}mV^2 + mgh,$$

from which the specific energy is:

$$h_e = \frac{\varepsilon}{mg} = \frac{V^2}{2g} + h,$$

For a climbing aircraft, the following relation holds:

$$m\dot{V} = T - D - mg \sin \gamma,$$

and the rate of climb is given by

$$\dot{h} = V \sin \gamma.$$

The rate of change of specific energy is:

$$\dot{h}_e = \frac{V}{g} \dot{V} + \dot{h} = \frac{V(T - D)}{mg}.$$

The ability of the aircraft to climb or accelerate, depends on the specific excess power, P_s . It has to be equal to the rate of change of specific energy:

$$P_s = P_{\text{available}} - P_{\text{required}} = \dot{h}_e = \frac{V(T - D)}{mg}.$$

The minimum time to climb is given by:

$$t_2 - t_1 = \int_{h_{e1}}^{h_{e2}} \frac{dh_e}{P_s},$$

where subscripts ‘1’ and ‘2’ refer to initial and final times respectively.

Let γ be the climb angle, the power required is

$$P_{\text{required}} = (D + \cos \gamma mg)V = \frac{1}{2}\rho C_d V^3 S + \cos \gamma mgV$$

We assume the drag coefficient velocity relation is the same as Fig 42

The power consumption plot is

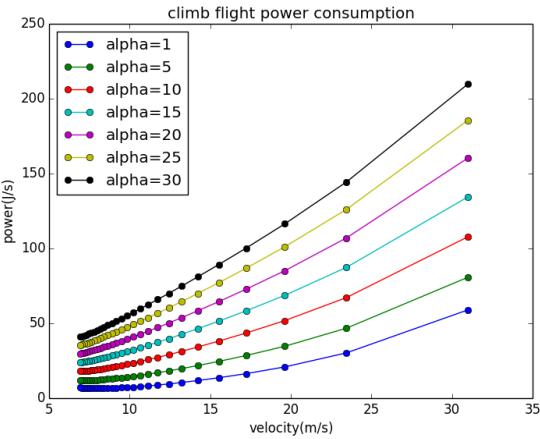


Figure 42: Bixler3 level flight power consumption

The velocity for maximum rate of climb is given by solving the following differential equation using appropriate mathematical models for thrust and drag:

$$\frac{T - D}{mg} + \frac{V}{mg} \left(\frac{\partial T}{\partial V} - \frac{\partial D}{\partial V} \right) = 0$$

If the thrust is independent of the speed, then $\frac{\partial T}{\partial V} = 0$, and the velocity for maximum rate of climb is:

$$V^2 = \frac{T}{6A} + \frac{1}{2} \sqrt{\left(\frac{T}{3A} \right)^2 + \frac{4B}{3A}},$$

where

$$A = \frac{1}{2} \rho S C_{D_{0L}}, \quad B = \frac{K(mg)^2}{1/2 \rho S}$$

The C_L for max rate of climb is:

$$C_L = \frac{1}{2K} \left[-\frac{T}{mg} + \sqrt{\left(\frac{T}{mg} \right)^2 + 12C_{D_{0L}}K} \right]$$

III Mission Plan and Strategy: Simulations

We developed a flight simulator in order to test various mission strategies and the different levels of controllers discussed in the previous problem set. The simulator performs simple forward Euler integration on a model of controlled aircraft dynamics, interfacing with a controller written in python. In the following section, we detail the basic dynamics model used, and subsequently demonstrate the performance of a simple waypoint following controller on

this simulator.

III.1 Simulator Dynamics

The full nonlinear dynamics of an aircraft are hard to model accurately, so instead for the simulator we assume a simpler model of the aircraft, based on the model discussed in Beard and McLain [2]. We assume that the aircraft's position in north/east/height coordinates is given by (p_n, p_e, h) . We let γ and χ denote the aircraft's flight path angle and course angle in the inertial frame respectively. ϕ denotes the aircraft's roll angle. V_a denotes the aircraft's wind speed. We describe the state of the aircraft in the model with the a priori mentioned variables:

$$\mathbf{x} = [p_n, p_e, h, \gamma, \chi, \phi, V_a]^T$$

Given these definitions, we can model the system as:

$$\begin{aligned}\dot{p}_n &= V_a \cos \psi \cos \gamma_a + w_n \\ \dot{p}_e &= V_a \sin \psi \cos \gamma_a + w_e \\ \dot{h} &= V_a \sin \gamma_a - w_d \\ \dot{\chi} &= \frac{g}{V_g} \tan \phi \cos(\chi - \psi)\end{aligned}$$

where ψ is the heading angle, γ_a is the flight path angle in the body frame ($\theta - \alpha$), and (w_n, w_e, w_d) is the wind velocity vector given in the North-East-Down coordinate frame. We assume wind to be a known value. ψ, γ_a and V_g can be computed from the state variables using the equations below, and hence are not considered part of the state.

$$\begin{aligned}0 &= V_g^2 - 2V_g \begin{pmatrix} \cos \chi \cos \gamma \\ \sin \chi \cos \gamma \\ -\sin \gamma \end{pmatrix}^T \begin{pmatrix} w_n \\ w_e \\ w_d \end{pmatrix} + V_w^2 - V_a^2 \\ \gamma_a &= \sin^{-1} \left(\frac{V_g \sin \gamma + w_d}{V_a} \right) \\ \psi &= \chi - \sin^{-1} \left(\frac{1}{V_a \cos \gamma_a} (w_e \cos \chi - w_n \sin \chi) \right)\end{aligned}$$

This models the aircraft's motion through space, assuming coordinated turning. For the roll, flight path angle, and airspeed dynamics, we make the simplifying assumption that under good low level control, these are essentially independent first-order systems. Thus, we

have have the equations below.

$$\begin{aligned}\dot{\gamma} &= b_\gamma(\gamma^c - \gamma) \\ \dot{\phi} &= b_\phi(\phi^c - \phi) \\ \dot{V}_a &= b_{V_a}(V_a^c - V_a)\end{aligned}$$

Where the commanded values γ^c, ϕ^c and V_a^c are the control inputs that the path following controller commands.

The constants $b_\phi, b_\gamma, b_{V_a}$ can be determined experimentally from flight data of the mission aircraft with low level control applied. We have not yet performed these tests.

III.2 Simulator Results: Simple Waypoint Following

With the above dynamics written as a simulator method in Python, we wrote and tested a simple waypoint following controller on this framework. The waypoint following controller consists of two main parts, a path following controller which controls the aircraft to follow a straight line path, and a state machine that changes the target path once the plane reaches a waypoint.

The path following controller is based on the model in Beard and McLain [2]. We assume the paths are at constant altitude and constant airspeed, and decouple the altitude and speed tracking from the x/y path tracking. Altitude is controlled by commanding the flight path angle, using the proportional control law $\gamma^c = k_{\text{alt}}(h^c - h)$. Speed is controlled by commanding the desired airspeed to a constant value. The lateral path following is done by commanding a course angle that transitions smoothly from some χ_∞ to χ_q as the perpendicular error between the plane's position and the line to follow decreases. χ_∞ is a fixed angle of approach directed toward the path. χ_q is the angle of the line segment to follow in the inertial frame. The equation that gives this behavior is shown below:

$$\chi^c = \chi_q - \chi_c \tan^{-1}(k_{\text{path}} e_y)$$

where k_{path} is a tunable control gain and e_y is the perpendicular distance between the plane's position and the path, projected onto the lateral plane. Beard and McLain [2] show using Lyapunov analysis that this system stabilizes e_y around zero, which is the desired behavior. To achieve this desired course angle, we command a roll angle proportional to the error in the course angle: $\phi^c = k_{\text{course}}(\chi^c - \chi)$.

The path that the above control laws should track is set by the Path Manager. The Path Manager parses the mission, here defined to be a sequence of north/east waypoints, and then

gives the path follower appropriate paths to follow. The current system keeps a counter to keep track of the current target waypoint. The desired path to follow is chosen to be the line connecting the previous waypoint with the current waypoint. The counter is incremented when the plane reaches a position within some threshold radius from the current target. One drawback of this technique is that if the plane fails to reach the target on its initial flight toward it, it will not turn around and adjust its course, as the controllers are tracking the line connecting the two waypoints, instead of controlling the position of the plane toward the target waypoint directly. More sophisticated logic in the Path Manager can solve this problem.

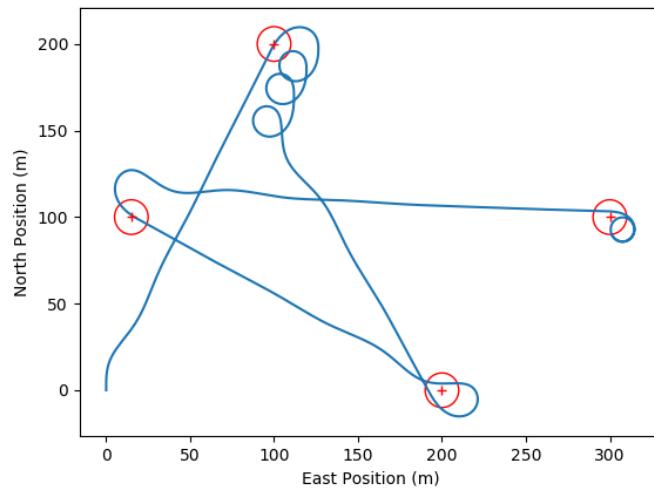


Figure 43: Simulated mission trajectory. Waypoints are shown in red, trajectory is in blue. Note that the simulation models some instability in path following due to control saturations.

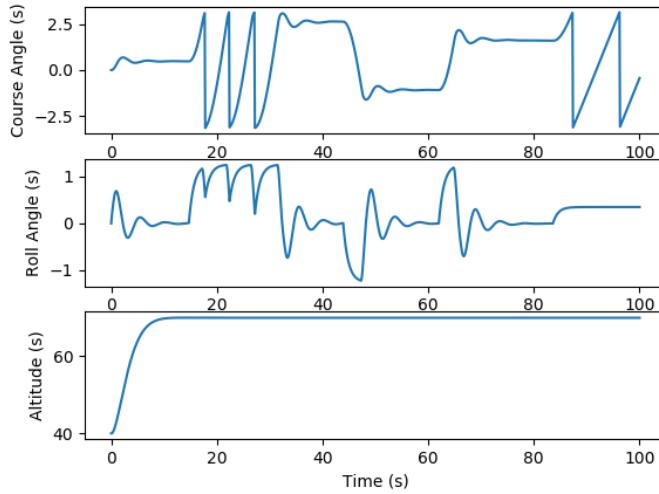


Figure 44: Relevant aircraft states during the simulated mission.

Figures 43 and 44 show the performance of the full system in simulation. We see that simulated plane navigates through the waypoints successfully, and takes a route that is realistic given the nature of aircraft dynamics. We can see that due to poorly tuned gains and control saturations, the plane ends up taking many more turns than necessary to turn from the first waypoint to the second.

Note that this simulator does not reproduce aerodynamic failure modes of the plane such as stalling, or structural failures due to too high loading. We will consider adding calculations of required C_L and load factor during the flight simulation to check that the trajectory doesn't require the plane to operate beyond its physical limits.

III.3 Path Planning

The path planning algorithm work sin two separate steps. The first one is based on a dynamic modeling of the planes behavior, outlined in section III.1. It is used to find an optimal path from the planes current position through all the waypoints optimizing for minimal path length, assuming that at an almost constant speed of the plane minimizing the path length is equal to minimizing the time required to accomplish the mission. Since the dynamic model is taking wind into account that assumption seems to increasingly accurate.

This algorithm involves more computational effort, hence it should be run once after take off as soon as the mission has been published.

In order to accommodate that possibility that new waypoints are added to the mission during flight. The second part of the path planning algorithm is updating the mission flight path by adding these additional waypoints in the currently executed flight path. Since the plane can not stop for re-computation, the time required for the algorithm to run short be really short. Hence a simple DP method, i.e. dynamic programming method, based on a cost system that involves only the distances between the new waypoints and the path.

The code structure is working in the following way:

```
#input: mission - list of waypoints, x - current state
#output: path - list of ordered

#initialize:
path=[x]
cost=0.

#loop over all waypoints
for i in range(len(mission)):
    for all path options:
        cost(i,j)=cost of adding waypoint i at position j in
        the path.
    pos_i=argmin(cost)
    path.include(pos_i, mission[i])

return path
```

IV Design, Construct and Fly Your Own Aircraft

IV.1 Dynamic Scaling

The mission of our aircraft is to simulate an autonomous aircraft sampling mission maximizing the number of interesting sites flown through during a limited communications window. Since the test mission will be flown on Earth, the plane must be designed to represent the dynamics of flight in a Martian atmosphere. To simulate these conditions, the aircraft was designed to match several dimensionless parameters that scale to Mars. The important aerodynamic parameters are Reynolds number, Mach number, lift coefficient, shape, and dimensionless mass.

The dimensionless parameters to match are based on the NASA Mars ARES project, which had a similar mission. The aircraft characteristics of the ARES are the following:

Table 9: NASA ARES Aircraft Characteristics

	ARES
Mass (kg)	2.3
Span (m)	4.2
Aspect Ratio	6.0
Cruise Velocity ($\frac{m}{s}$)	22.0
Atmospheric Density ($\frac{kg}{m^3}$)	0.15
Speed of Sound ($\frac{m}{s}$)	225
Gravitational Acceleration ($\frac{m}{s^2}$)	3.7

These values can be used to extrapolate the wing area and chord. Other important values are density, gravitational acceleration, and speed of sound on Mars. These values, particularly the density and speed of sound, can vary drastically.

The Reynolds number for any aircraft in general on Mars is going to be very low. The Reynolds number of the ARES aircraft was calculated to be about 60,000, which is almost impossible for us to simulate on Earth without flying at extremely high altitudes. Fortunately, the boundary layer can be tripped on the wings to avoid low Reynolds number uncertainties. This results in both our aircraft and the Mars design effectively flying at higher Reynolds numbers where the boundary layers are turbulent. Therefore, our plane can achieve a similar Reynolds number to a Mars aircraft without having to fly at a Reynolds number of 60,000.

The Mach number of the ARES aircraft on Mars was calculated to be about 0.1. This is small enough to neglect compressibility effects, so our plane can have a different Mach number as the ARES as long as we fly below about 0.3. Doing a rough calculation of our predicted aircraft, we expect to fly at about Mach 0.05 which is well below the compressibility region.

The lift coefficient is one of the parameters that we will try to match exactly. Using the equation for lift coefficient $C_L = \frac{L}{\frac{1}{2}\rho V^2 S_w}$, our lift coefficient will be 0.80 at cruise condition.

To match the shape of the ARES, we will match the given aspect ratio of 6.0. This coupled with a matching lift coefficient will give our aircraft many of the same aerodynamic quantities.

Finally, the non-dimensional mass of the ARES will be matched. The non-dimensional mass will affect many aspects of the aircraft handling, such as damping and stability modes. The equation for non-dimensional mass is $\mu_s = \frac{m}{\rho S_w c}$. The calculated non-dimensional mass for the ARES that we will match is 74.5.

IV.2 Aircraft Design

Before we started the design of our Martian aircraft, we brainstormed the most important characteristics of the plane we wanted to design for. Because performing well in our mission depends on how many spots we can reach in an allotted amount of time, we decided a high cruising velocity was critical. However, due to the lift coefficient and non-dimensional mass matching, the mass of our plane increased dis-proportionally with an increase in cruise velocity.

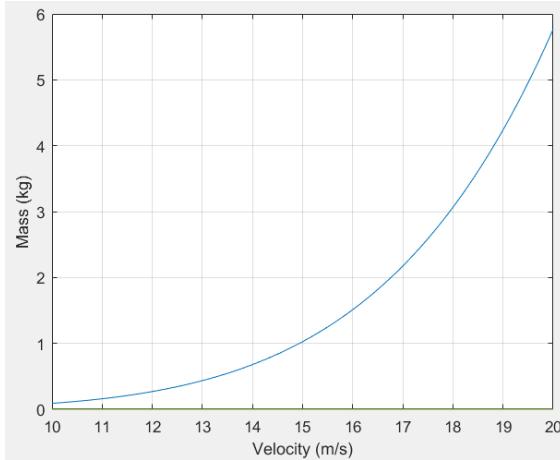


Figure 45: Aircraft Mass for Fixed Velocity

Setting the cruise velocity of the plane also fixed what our chord, span, and wing area would be. We ended up choosing a cruising velocity of 16 m/s which resulted in a rather hefty 1.51 kg plane. Preliminary weight estimates based on fuselage weight, wing area, and material choices placed the overall mass of our plane to naturally settle around 900 g, so we would have to ballast our plane with an additional 0.6 kg to meet the dynamic scaling requirements. A summary of our aircraft specifications is shown below.

Table 10: Albatross Aircraft Parameters

	Albatross
Mass (kg)	1.512
Span (m)	0.842
Aspect Ratio	6.0
Cruise Velocity ($\frac{m}{s}$)	16.0
Wing Area (m^2)	0.118
Chord (m)	0.14

Another design point we placed a heavy value on was our maximum lift coefficient. Because our cruising velocity was so high, we would need a lift coefficient of 1.2 just to reach a 1.5 g turn. Our maximum roll angle will be determined directly by our maximum lift coefficient. The max lift coefficient also determines our level flight stall velocity. Even at high maximum Cls around 1.6, the stall velocity of our plane is still estimated to be greater than $11 \frac{m}{s}$, so a high lift coefficient will be critical during the landing and takeoff portions of our flight.

We determined that the drag of the plane is not as critical as the previously mentioned characteristics because our plane is not power limited. When designing the plane, the propulsion model had not been developed yet, but it was assumed that our plane could reach the desired 16 m/s velocity easily with any reasonable drag coefficient. Therefore, a lower drag was preferable in any otherwise comparable design, but it was not a driving factor in determining the design of the wings.

Having identified our important aircraft characteristics, our next step was to determine an airfoil shape for our main wing. Our search concentrated on high-lift low Reynolds number airfoils, but we also learned to be careful that some of those airfoils only reach a high performance in a theoretical manner and would be impractical on a real aircraft. The chosen airfoil candidates and their 2D XFLR5 performances are shown below:

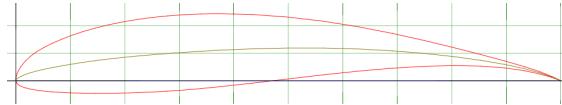


Figure 46: FX 63-137 Sm Airfoil

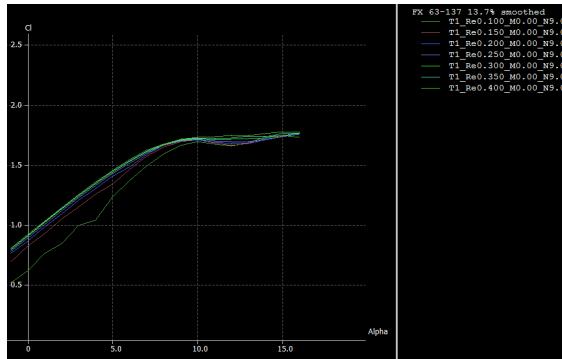


Figure 47: FX 63-137 Sm Cl vs Alpha

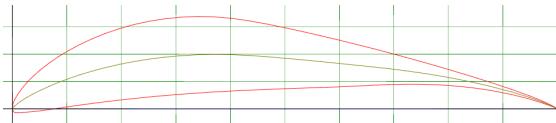


Figure 48: FX 74-CL5-140 MOD Airfoil

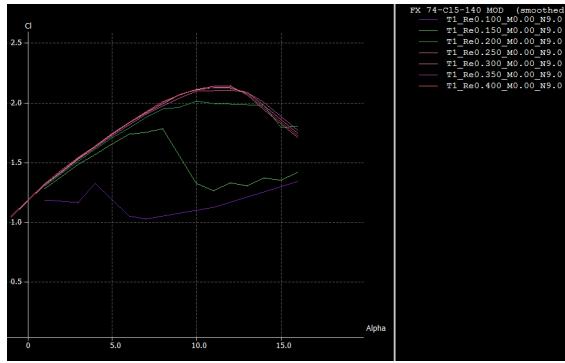


Figure 49: FX74-CL5-140 MOD Cl vs Alpha

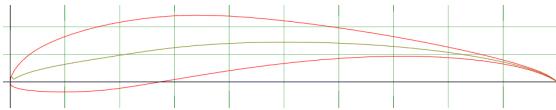


Figure 50: S1210 Airfoil

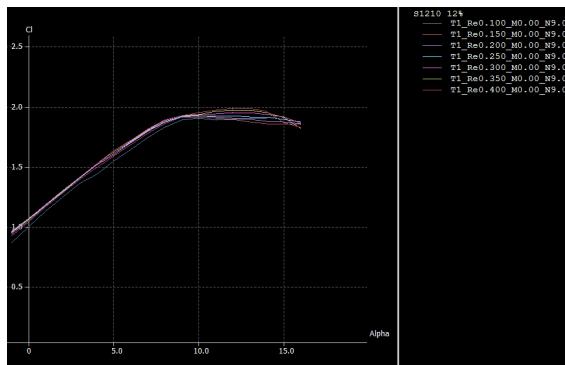


Figure 51: S1210 Cl vs Alpha

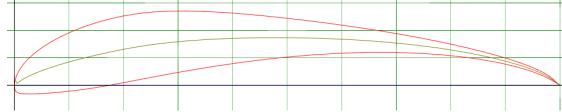


Figure 52: S1223 Airfoil

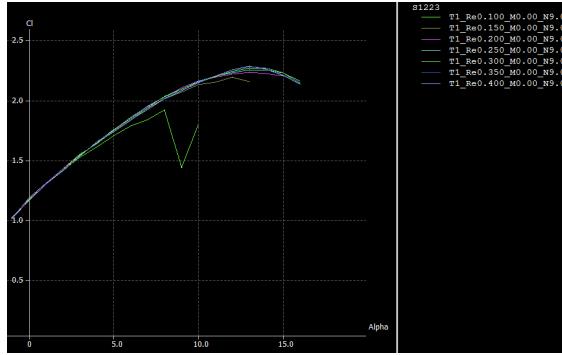


Figure 53: S1223 Cl vs Alpha

At a first glance, the S1223 airfoil would be the ideal choice due to its high lift coefficient. However, after discussions with the TAs and Professor Kroo, it was decided to choose an airfoil with more thickness and less camber for structural stability and stable flight characteristics. Our final airfoil chosen was the FX 63-137 Sm.

A full 3D wing analysis was then analyzed and iterated upon in XFLR5, attempting to achieve a high maximum lift coefficient while meeting the wing geometry requirements and hopefully resulting in low drag coefficients. Our final wing design resulted in a maximum lift coefficient of 1.67 and a drag coefficient of 0.61. A tail was then designed to try to achieve stability in the short, phugoid, roll, dutch roll, and spiral modes, but this is where we ran into some issues. The dutch roll and spiral stability modes were unstable in addition to an excessively high negative pitching moment. After attempting and to fix these issues with just the tail and ultimately failing, the wing was modified with more dihedral and less aggressive wing twist. We were able to reduce our pitching moment and stabilize the dutch roll mode, however the spiral mode proved difficult to stabilize. After consulting literature online, it was found that an unstable spiral mode is not uncommon in aircraft, so we went forward with our design. Our final wing shape, geometry, and stability derivatives are shown below:

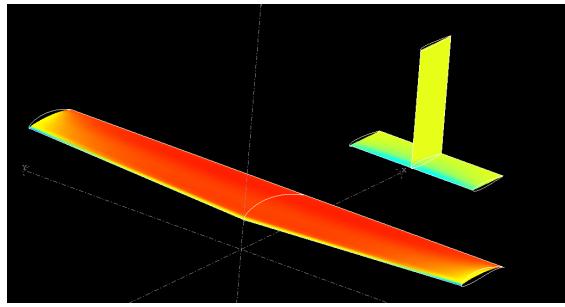


Figure 54: Plane Design in XFLR5

Table 11: Albatross Wing Configuration

	Albatross
Wing Area (m^2)	0.118
Dihedral (deg)	3.0
Wing Twist (deg)	-1.0
Root Chord (m)	0.165
Tip Chord (m)	0.115
Mean Aerodynamic Chord (m)	0.141
Half-Span (m)	0.421
Root-Tip Sweep (deg)	-1.701
Aspect Ratio	6.01
Distance from Wing LE to Tail LE (m)	0.45
Horizontal Tail Span (m)	0.24
Horizontal Tail Chord (m)	0.08
Vertical Tail Span (m)	0.2
Vertical Tail Chord (m)	0.08
Tail Volume	0.494

Table 12: Albatross at Cruise Condition

	Albatross
Cruise Velocity ($\frac{m}{s}$)	16.0
Angle of Attack (deg)	0.0
Lift Coefficient	0.804
Drag Coefficient	0.061
Moment Coefficient	0.008

Table 13: Albatross at Maximum Lift Condition

	Albatross
Velocity ($\frac{m}{s}$)	16.0
Angle of Attack (deg)	10.5
Lift Coefficient	1.604
Drag Coefficient	0.211
Moment Coefficient	-0.175

Stability analyses were carried out during the design process and greatly influenced the design of the aircraft, but this has already been discussed in Section 2.

IV.3 Aircraft Manufacturing

Having no prior experience with balsa wood manufacturing, we decided to make our wing out foam with a carbon spar. The foam cutter was able to effectively cut our taper, twist, and dihedral, but we ran into the issue of having a discontinuous spar due to the dihedral. We epoxied a carbon spar down the quarter chord for both halves of our wings, but failed to add a spar connecting the two halves. This would come back to haunt us later.

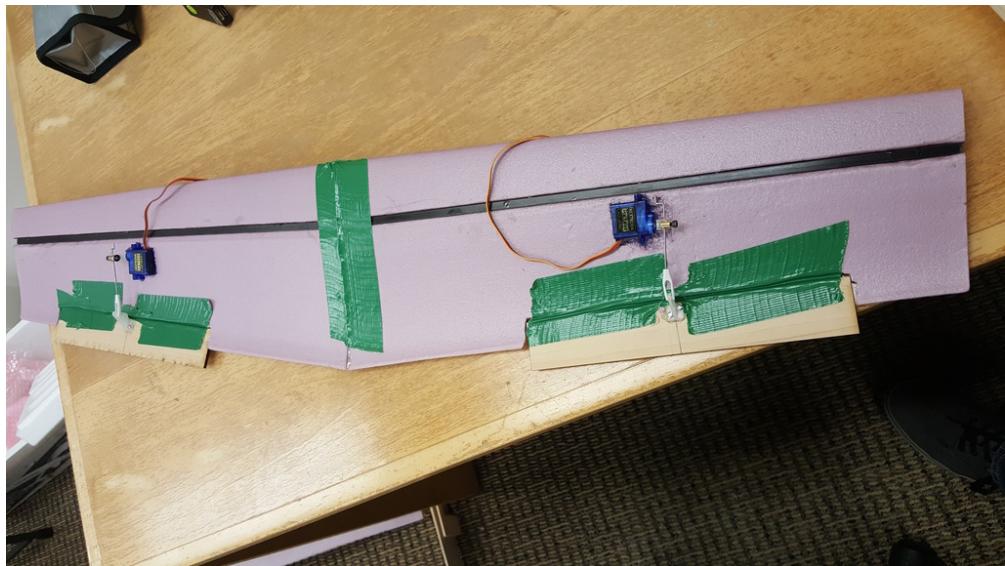


Figure 55: Wing Structure

As mentioned previously, we found it likely when fixing our cruise velocity that we would have to add mass to the aircraft to match the non-dimensional parameters of the NASA ARES aircraft. Therefore, when determining the design and materials for our plane, we were indifferent to extra weight. In this spirit, we decided to manufacture our tail out of wooden flat plates instead of foam airfoils, hoping it would prove to be sturdier and more robust while still providing enough force for the necessary pitching and yawing moments.

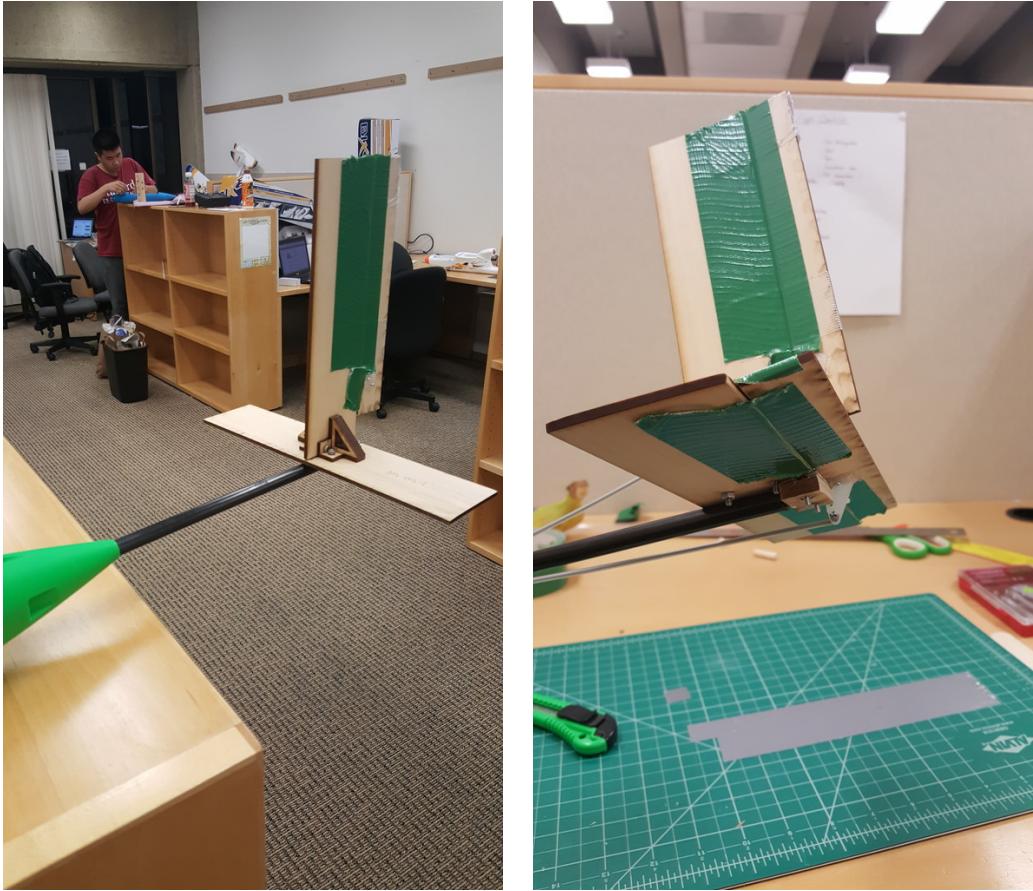


Figure 56: Tail Assembly and Mounting

To mount the tail, an ‘anchor’ piece was bonded into the end of our carbon tube boom to provide a tail mount. Two bolts connecting the horizontal tail to the anchor secured the tail to the boom. This would prevent the tail from spinning around the tube. Duct tape was used to secure the tail boom to the fuselage and prevent turning.

To fasten the main wing to the fuselage, a wooden wing box was made with the laser cutter. This consisted of three wing profile cutouts close to the wing fuselage that encapsulate the wing and hold it in place. The wing was glued inside the box, and the box then Velcro’d to the fuselage and held down with rubber bands to securely fasten it.

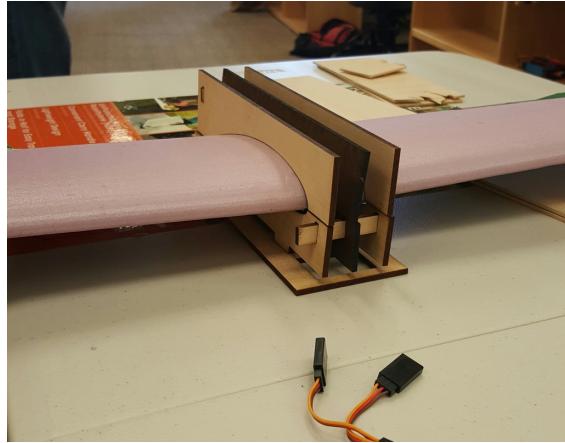


Figure 57: Wing Mounting

Our control surfaces were also made of wooden plates because we were unsure if the thin trailing edge of the foam would be stiff and sturdy enough to handle actuator inputs with deforming or breaking. The ailerons were sized to be a quarter of the span and a quarter of the chord. The elevator was sized to be 30 percent of the horizontal tail chord and across the entire span. However due to manufacturing design errors, our produced elevator covered only 90 percent of the span. The rudder covered the spanned the whole vertical tail except for a small section cut off at the bottom to avoid interference with the elevator. The servos for the ailerons were partially embedded inside the wing, while the elevator servo was placed inside the fuselage with a push rod reaching to the tail. We did not connect the rudder for the first flight because we typically do not use the rudder during manual flight.



Figure 58: Albatross Aircraft

IV.4 Flight Testing

Flight 1: Our first flight was very short-lived. We decided not to do any glide tests before flying to avoid the risk of damaging any parts of the plane, but therefore our plane was not properly trimmed. For our first flight, the plane pitched straight up immediately after being thrown, rolled over backwards, and crashed behind us.



Figure 59: Flight 1: Failure due to improper trimming.

Fortunately, there was very minimal damage, nothing a couple pieces of duct tape couldnt fix. We attributed this crash to a CG too far aft so we moved the battery forward in the fuselage and changed the elevator trim to add more pitch down moment.

Flight 2: The second flight achieved successful takeoff, however the aileron controls were too strong and caused the plane to veer heavily side to side before running into the ground.



Figure 60: Flight 2: Failure due to high aileron control sensitivity.

Again, we were fortunate that only the wing box had damage and everything else was intact. We taped up the wing box, added exponential to the aileron controls on the transmitter, and decided to try again.

Flight 3: The third flight attempt had a smooth takeoff and the rolling was much more controlled. The only problem occurred about 15 seconds into the flight when the plane pulled up from a diving maneuver and appeared to break the wings at the root. From a distance, it looked like our plane went from a 3 degree dihedral to a 30 degree dihedral.



Figure 61: Flight 3: Failure due to wing collapse during pull-up maneuver

We are not sure if this would have happened or not if our wing hadn't already been damaged in an earlier flight, but the next iteration of our plane will place a large focus on improving the wing structure. This was particularly troubling because our plane wasn't loaded to its full gross mass. We were flying at about 0.9 kg whereas our plane was designed to fly at 1.5 kg.

IV.5 Design Changes

Weight In general we are concerned about the mass we have initially planned to fly at. Currently we would fly at 16 m/s with 1.5 kg. If we lower our velocity by 0.5 m/s to 15.5 m/s, we would decrease our mass by 0.25 kg. The new span width would be of 0.79m and the chord length was recomputed as being 0.132 m.

The lower bound of our planes weight is at 900g. This would correspond to a cruise velocity of 14.7 m/s. The chord would then be 0.1184 m and the span 0.71 m.

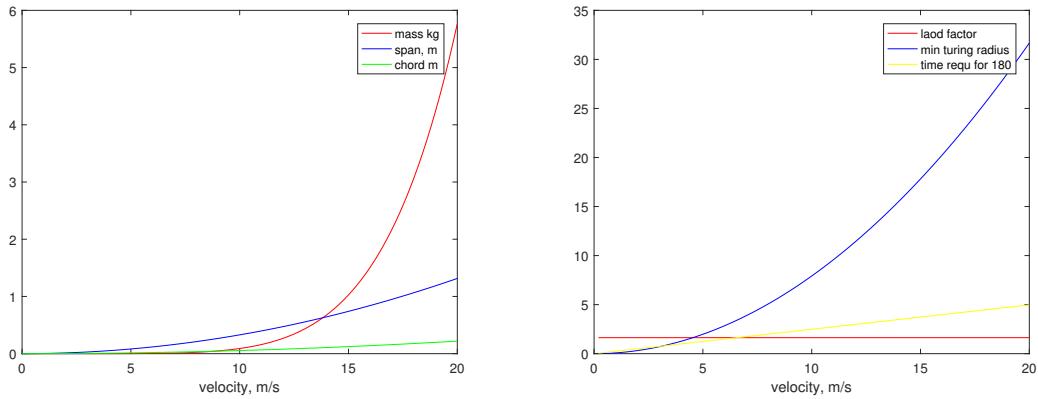


Figure 62: Variation of design parameters with cruise velocity

The figures above depicts the relation of mass, chord and span to cruise velocity. As a reference the effective values for our initial design are summarized in Table 10.

Unstable Spiral Mode As discussed in the stability analysis, refer to section II.3.2., the spiral mode of our initial design is unstable. We might want to consider tackling this issue in the design for our second iteration and construction.

The spiral mode usually is only lightly damped, or may even be unstable. Dihedral effect is an important stabilizing influence, while weathercock stability is destabilizing, for this mode. A too large elevator can also cause a destabilization of this mode.

However, for the Dutch Roll mode, dihedral effect is generally destabilizing, hence we want to be careful about that effect. This mode is responsible for a coordinated yawing, rolling, side-slipping motion. Also the phugoid mode is affected by these changes.

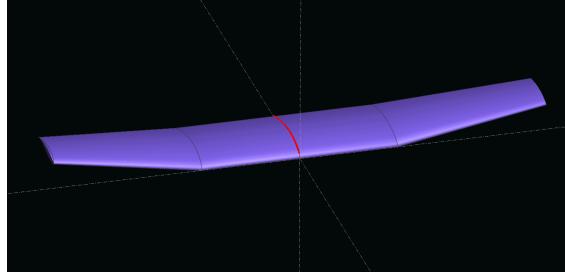


Figure 63: Design: Iteration 2

For the second iteration of our wing, a design we consider manufacturing is the following one. Layed out with a cruise speed of 14.8 m/s and a weight of 905g, this design is easy to manufacture (compared to other ideas we had), but the spiral mode is still slightly unstable. However the stability analysis resulted in a time constant of $\tau_s = 9.2$ sec for this mode. Which corresponds to a frequency of 0.11 Hz. Compared to the first design iteration, the frequency is half as high which makes it easier to control.

The maximum C_l for this design is a 1.7 which will facilitate the landing maneuvers a lot. The required AOA at which the wing should be mounted is approx. +2 degrees. Please note that in this design and adapted, i.e. lighter, weight distribution was used for the simulations and that the tail was scaled down as well.

Construction For one, we want a carbon spar that goes through the entire mid section to enhance stability at the spots it broke last time. Our current plan is to have a carbon spar going along the mid-section and two separate, shorter ones connecting the two side parts.

The tail assembly has quite a few design errors that can be easily fixed by the next iteration. Finally, the wing box was thrown together last minute but more careful design and shaping could make it both more compact and more effective at mounting the wing.

V Closing control loops

In order to implement autonomous flying on the Bixler 3 and on our airplane subsequently we investigated stability and control derivatives as well as the corresponding dimensionless parameters. The simulations with XFRL5 were successfully providing stability parameters for the Bixler 3 (please refer to Section II.3) and our plane design (refer to Section II.1). The control derivatives at cruising condition were estimated from executing test-flights (refer to Table 5).

From literature, we could see that PID controller were widely used in industry. Knowing that the performance of these controllers are more sensitive to noise and "suboptimal" gains, we chose to use a successive loop closure technique to achieve longitudinal and lateral control.

For low-speed flight the inertial cross-coupling can be neglected. A Matlab simulation using the pre-computed stability coefficients provided reference values and starting points. They were adapted and tuned during test flights on May 10th and 11th.

We choose the gain in the speed control loop as an example for the gain design process. The aircraft dynamics is modeled using equations of motion expressed in Section II.3.2, and transfer function from control input δ_T (throttle) to output u (speed) is then derived. The gain was chosen based on desired pole locations in the root locus diagram, shown in

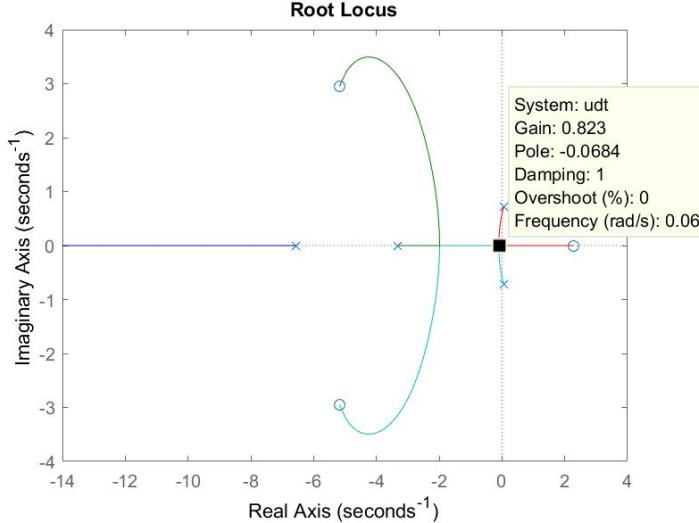


Figure 64: Root locus for $\frac{u(s)}{\delta_T(s)}$

As shown in the root locus diagram, in order to increase damping and stability, the gain K_u was chosen as 0.821. The gains to start the experiment with for other control loops were picked using a similar process.

V.1 Longitudinal Autopilot

We are implementing the speed-hold and altitude hold in two separate loops. The speed hold is, most intuitively, controlled by a feedback loop acting on the throttle servo. The controller is a simple gain of $K_u = 0.821$, tracking the difference of u to $u_{desired}$.

After closing the speed control loop, the altitude control loop is then written. First the commanded pitch angle θ_c is computed from the commanded altitude h_c through gain K_h .

$\theta_c = K_h(h_c - h)$. Then the pitch servo output δ_e was issued, by multiplying the gain K_θ with the error in pitch $\theta_c - \theta$, as follows: $\delta_e = K_\theta(\theta_c - \theta)$. The longitudinal control block diagram is shown in Fig.65.

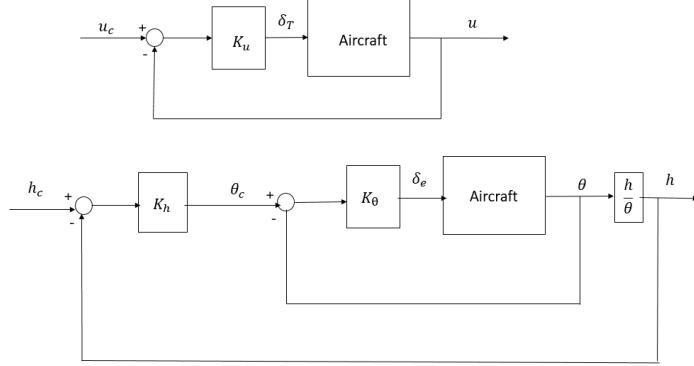


Figure 65: Longitudinal control loop

V.2 Lateral Autopilot

First in lateral control we want the turns to be coordinated, i.e., no sideslip ($\beta_c = 0$). The side-slip β is calculated by $\frac{v}{V_{total}}$ where v is the velocity in y-direction in body-axis (out the right wing) and V_{total} is the total speed. Here the measured v comes from the IMU and the ground speed was used as V_{total} , which might not be accurate when wind is not present. In the code a small tolerance 10^{-5} was added to V_{total} in the denominator to avoid singularity when $V_{total} = 0$. The rudder servo output is chosen to drive the error in sideslip to zero, using the proportional control law: $\delta_r = K_\beta(\beta_c - \beta)$.

Secondly, with the coordinated turn loop closed, the aileron control loop was written to maintain desired heading. First, desired roll angle is calculated from error in yaw, $\phi_c = K_\psi(\psi_c - \psi)$. Then, the aileron servo output is calculated from the error in roll, $\delta_a = K_\phi(\phi_c - \phi)$. The lateral control block diagram is shown in Fig.66.

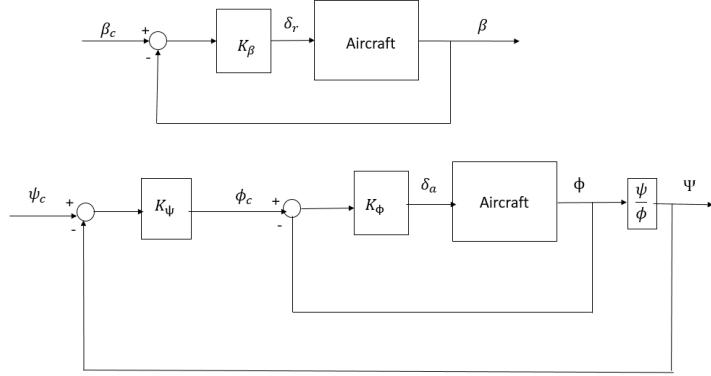


Figure 66: Lateral control loop

The gains determined experimentally is shown in Table 14.

Table 14: Proportional controller gains

K_u	0.821
K_β	-0.01
K_ψ	3.0
K_h	-70.0
K_θ	-0.032
K_ϕ	3.0

As a result, shown in Section V.3, we can see the control loop for the second task of keeping a straight flight path for at least 100m.

In the many of these control laws, particularly those commanding servo outputs or euler angles, we apply saturation to the command output. This serves to keep the system in reasonable modes of operation (for example, having a maximum climb angle, even if error in altitude is very high).

Furthermore, in implementing these control laws, we take care to wrap any computed angle errors to within $[-\pi, \pi]$, to ensure that the system corrects in the direction that requires the least effort.

V.3 Bixler3 Flight Tests

In this section, we will show Bixler3's flying result in autonomous mode. When we switch from RC mode to autonomous mode, our control law will make Bixler3 to maintain heading and fly at about 70 meters about see-level. During the flight test on May 11th, we switched

between R/C mode and autonomous mode several times, the altitude , flight paths and control inputs under autonomous mode are shown in Fig 67

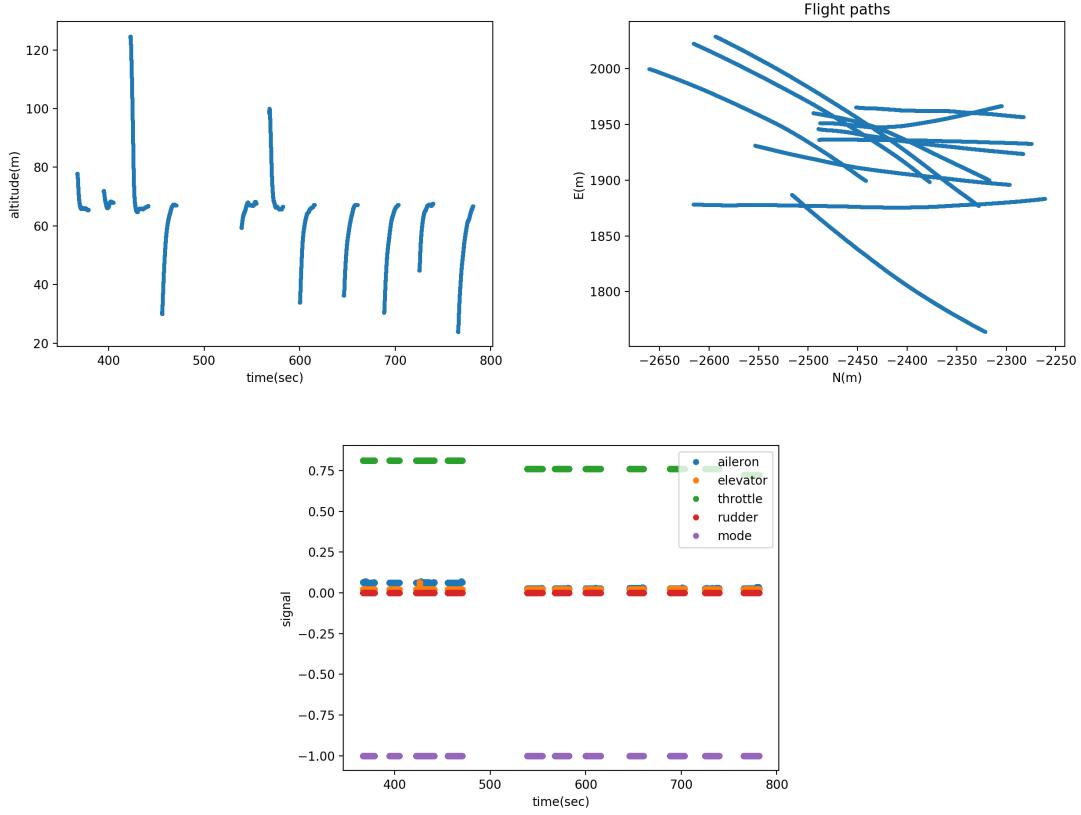


Figure 67: Autonomous mode flight data

From Figs 67, we can see switching to autonomous mode, Bixler3 maintains the heading and goes back to prior specified level(70 meters) rapidly, and keeps the altitude. Figs 68 show the zoomed straight level flight, although it is not exact 70 meters. It looks similar to the pictures in Pset2, the actual path is not exactly the intend path, but well approximates the intend path. The sensor inaccuracy and wind might cause the this less than %5 error, which is acceptable.

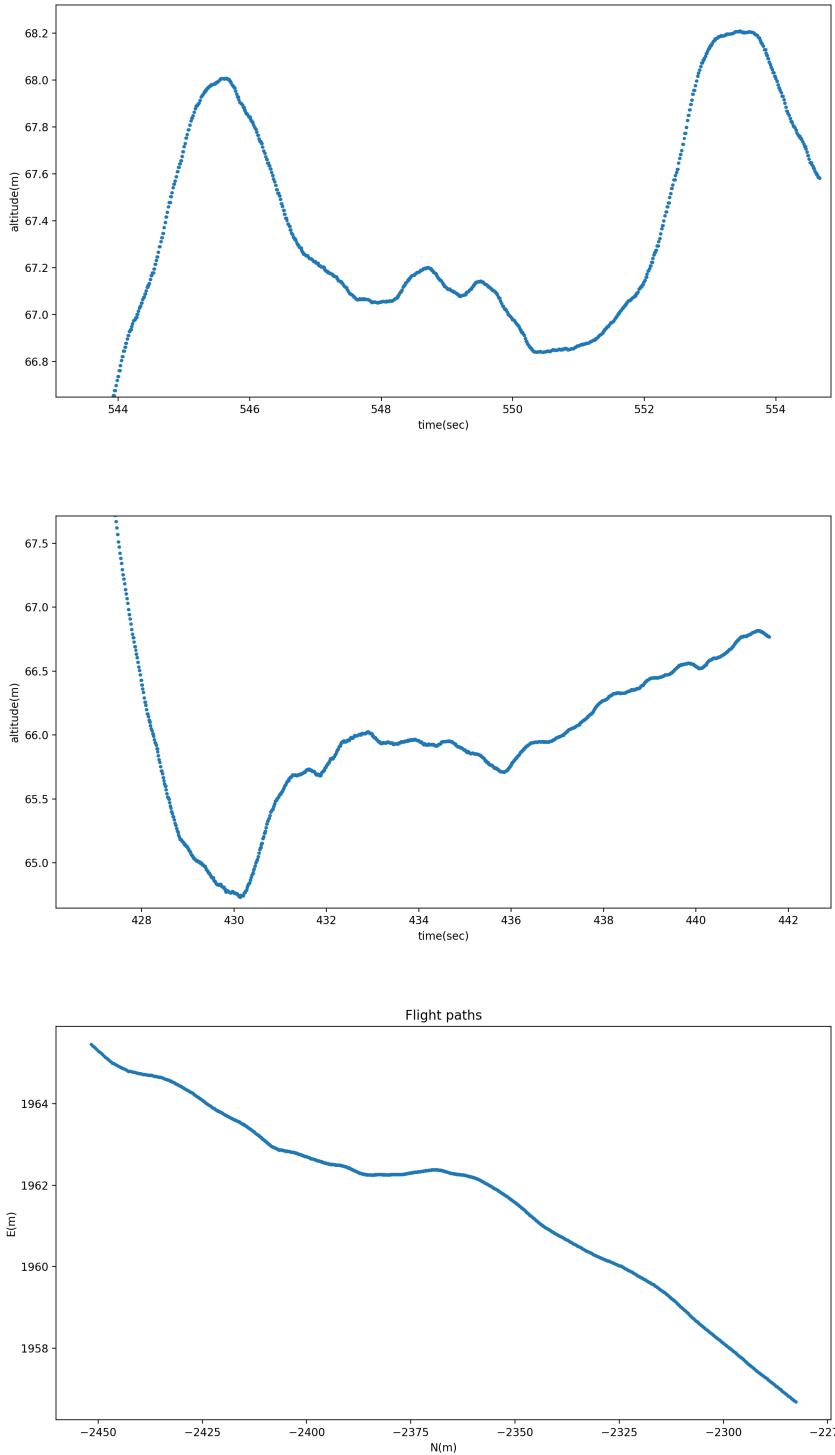


Figure 68: Zoomed autonomous mode flight data

VI Goals and Plan of Action

Up to now, we have designed and built a first prototype of our dynamically scaled Mars aerial vehicle, and have tested it in flight. From these initial flights, we have obtained feedback on the quality of our design, and looking forward, we will use these lessons to improve our design for the next prototype. A second iteration of the aircraft will be designed and built in the coming days and we will aim to gather flight data by the beginning of next week.

In addition to designing an improved aircraft, we will need to apply the same techniques we used to control the Bixler's flight to create and tune an autopilot for the new prototype. This will involve both better system identification and tuning of gains in the successive loop closure autopilot system.

Finally, we will need to design and implement higher level decision making modules that can more robustly navigate the mission waypoints and also tackle the problem of choosing the optimal order of the waypoints.

We also plan to improve the fidelity of the simulator and add C_L and load factor calculations so that we can ensure the mission planner and waypoint tracking controllers are physically feasible for our aircraft.

Appendix

Table 15: Writing Contribution

Section	Name
I.1	Zhengyu Huang
I.2	Zhe Zhang
II.1	Zhengyu Huang
II.2	Zhe Zhang
II.3.1	Miao Zhang; Victoria M Dax
II.3.2	Miao Zhang
II.4	Zhengyu Huang
II.5	Zhe Zhang
III	Apoorva Sharma
III.3	Victoria M. Dax
IV	Justin Schneider
IV.5	Victoria M Dax
V	Victoria M Dax; Miao Zhang
V.3	Zhengyu Huang
VI	Apoorva Sharma

References

- [1] John H. Blakelock, *Automatic Control of Aircraft and Missile* A Wiley-Interscience publication: John Wiley & Sons, 1991.
- [2] Randal W. Beard, Timothy W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.