

J.A.H. Kuijpers

Determining the lateral-directional handling characteristics of a Skysurfer X8



Determining the lateral-directional handling characteristics of a Skysurfer X8

By

J.A.H. Kuijpers

In partial fulfilment of the requirements for the degree of

Master of Science
In Aerospace Engineering

At the Delft University of Technology,
To be defended publicly on October 7th 2021 at 9:00 am.

Supervisor:	Dr. ir. R. Vos	TU Delft
Duration:	2019 -2021	
Thesis committee:	Dr. ir. R. Vos	TU Delft
	Ir. C. de Wagter	TU Delft
	Dr. ir. G. la Rocca	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Summary

With standard aircraft configurations reaching their limits in efficiency, new aircraft configurations are being theorized and analyzed. For analysis of the new configurations there are multiple options, from computer simulations and wind tunnel tests to actual flight tests. To limit the cost as well as the risks associated with building and test flying a full-scale prototype, remotely piloted scaled flight tests can be conducted. These scaled flight tests have an advantage over wind tunnels in cost. Furthermore using free flight models, flight tests can be conducted in a similar environment the full scale aircraft will operate in. One of the aspects that can be analyzed through remotely piloted scaled flight tests is the dynamic behavior of the aircraft. This thesis shows a method to determine the lateral-directional stability and controllability of a remotely piloted scaled flight model.

The aircraft chosen to perform the flight tests with is the X-UAV Skysurfer X8. The aircraft is equipped with a Pixhawk 4 flight controller running a custom version of the PX4 software stack. The Pixhawk 4 records measurements of many different sensors as well as the control inputs given to the servos.

From the acquired data a linearized state space model for lateral-directional stability and controllability is created through the optimization of a cost function depending on the difference between the estimated model and the measurements.

From the optimized state space system the lateral-directional handling characteristics are determined by looking at the eigenvalues. The resulting eigenvalues show that the aircraft as configured during the flight test has an unstable spiral mode with a time to double amplitude of 5.18 seconds. The Dutch roll mode experienced by the aircraft has a frequency of 0.93 Hertz. This mode is damped with a time to half amplitude of 0.303 seconds. The final mode experienced by the aircraft is the roll subsidence mode which is highly damped with a time to half amplitude of just 0.0073 seconds.

As a separate method for determining the lateral-directional dynamics of the aircraft an aerodynamic model is analyzed with a vortex lattice method. Despite not analyzing the fuselage, pitot tube mount, wing fuselage interactions or the effect of the propeller, the model still showed the unstable spiral mode, damped Dutch roll mode and highly damped roll subsidence mode.

To validate the results of the flight test, a different set of inputs are given on the rudder and ailerons which are then analyzed using the previously obtained model.

Once the results are deemed validated, the results for the lateral-directional dynamics are analyzed according to military specification MIL-HDBK-1797. This analysis shows the aircraft has level 3 behavior with respect to the spiral mode, meaning constant attention is required from the pilot if no control augmentation system is used, but it is still controllable. The roll subsidence and Dutch roll mode are of level 1 which is considered satisfactory behavior. The roll performance of the aircraft is of level 2, meaning it is acceptable behavior.

Acknowledgements

I would like to thank my family for their support and encouragement during my entire studies. Also, I would like to thank my friends of the United Shuttles Smashing Right badminton club for all times that we played some games, taking my mind off the stressful moments within the thesis.

Special thanks go out to my project teammate Stefan, the time brainstorming, building, testing, crashing, rebuilding, trying again and never giving up has been motivating throughout the project. Also Alexandru who was always there to make jokes and have some fun times. The trips that Stefan, Alexandru and I have made to different airports and the beach for many different flight tests have been amazing.

I want to thank Johan, Victor, Fred, Johan and Frank for providing us with the space, tools, materials and expertise required to build the aircraft. Also, I would like to thank Erik for his advice and help with everything RC-model related. Furthermore, I would like to thank Michiel and Otto for their visits at the end of the day making us laugh. Also, thanks go out to Kimberly for always ordering the parts required for the project.

Finally, I would like to thank Roelof, for giving the freedom to try and make the project fly, but also the support and guidance to finish the project.

Jan-Willem Kuijpers
Delft, September 2021

Nomenclature

Alphabetical variables

Variable	Description	Units
a	Acceleration	$[m/s^2]$
$A[k]$	Attitude transformation matrix	-
b	Bias of measurements	-
b	Wingspan	$[m]$
\bar{b}_a	Vector containing 3 accelerations in body reference frame	-
b_h	Barometric height	$[m]$
\bar{b}_ω	Vector containing 3 angular rates in body reference frame	-
C_L	Aircraft lift coefficient	-
C_n^b	Projection of the body reference frame to the east north up earth fixed reference frame	-
C_{Y,l,n_x}	Normalized coefficients Y_x, l_x, n_x	-
F	Force	$[N]$
f	Cost function	-
g	Earth gravitational acceleration	$[m/s^2]$
I	Current	$[A]$
I_{XX}	Moment of inertia around the X axis	$[kg \cdot m^2]$
I_{XZ}	Product of inertia, inertia around the Z axis when body is rotated on the X axis	$[kg \cdot m^2]$
I_{YY}	Moment of inertia around the Y axis	$[kg \cdot m^2]$
I_{ZZ}	Moment of inertia around the Z axis	$[kg \cdot m^2]$
L_x	Moment around X axis due to unit of x , where x can be $v, \beta, \dot{\beta}, p$ or r	$\frac{[N \cdot m]}{[unitx]}$
m	Mass	$[kg]$
n	Observation noise	-
N_T	Number of measurement points	-
N_x	Moment around Z axis due to unit of x , where x can be $v, \beta, \dot{\beta}, p$ or r	$\frac{[N \cdot m]}{[unitx]}$
P	Prediction covariance matrix	-
p	Roll rate	$[rad/s]$
\bar{p}_n	position vector in east north up earth fixed reference frame	-
p_∞	Steady state roll rate	$[rad/s]$
q	Quaternion	-
q	Pitch rate	$[rad/s]$
\bar{q}	Dynamic pressure	$[Pa]$
\bar{q}_n	quaternions vector from body reference frame to east north up earth fixed reference frame	-
R	Measured covariance matrix	-
R	Resistance	$[\Omega]$
r	Yaw rate	$[rad/s]$
r	Normalized innovation	-
r_n	Averaged normalized innovation	-

S	Wing surface area	$[m^2]$
S	Mahalanobis distance matrix	-
t	Time	$[s]$
$T_{\frac{1}{2}}$	Time to half amplitude	$[s]$
T_R	Roll rate time constant	$[s]$
T_s	Spiral mode time constant	$[s]$
U	Voltage	$[V]$
u	Control input	-
$U[k]$	Gravity corrected acceleration in east north up earth fixed reference frame	-
V	Aircraft velocity	$[m/s]$
V_{ge}	Aircraft GNNS groundspeed in east direction	$[m/s]$
V_{gn}	Aircraft GNNS groundspeed in north direction	$[m/s]$
V_{we_e}	Estimated wind speed relative to ground in east direction	$[m/s]$
V_{we_n}	Estimated wind speed relative to ground in north direction	$[m/s]$
v	Velocity in Y direction of body frame	$[m/s]$
\bar{v}_n	Velocity vector in east north up earth fixed reference frame	-
W	Aircraft weight	$[N]$
X	Body axis pointing from the center of gravity to the nose	-
\bar{x}	State vector	-
\bar{x}^E	Extended state vector	-
\hat{x}_p	Predicted x position of aircraft	$[m]$
x_m	Measured x position of aircraft	$[m]$
Y	Body axis pointing from the center of gravity to the right of the aircraft	-
\hat{y}_p	Predicted y position of aircraft	$[m]$
y_m	Measured y position of aircraft	$[m]$
Y_x	Force in Y direction due a unit of x , where x can be $v, \beta, \dot{\beta}, p$ or r	$\left[\frac{N}{unitx} \right]$
Z	Body axis pointing from the center of gravity down to the ground in the case of level flight	-
z_m	Measured z position of aircraft	$[m]$

Greek variables

Variable	Description	Units
β	Sideslip angle	$[rad \text{ or } ^\circ]$
Γ	Vorticity per unit length	-
δ_a	Deflection of aileron, actual or PWM input to aileron servo, where neutral position is set to 0	$[rad]$ in verification or $[100us]$ anywhere else
δ_r	Deflection of rudder, actual or PWM input to rudder servo, where neutral position is set to 0	$[rad]$ in verification or $[100us]$ anywhere else
ζ	Damping ratio	-
θ	Pitch angle	$[rad \text{ or } ^\circ]$

λ	Eigenvalue	-
ρ	Air density	$[kg/m^3]$
Φ	Velocity potential	-
ϕ	Roll angle	$[rad \text{ or } ^\circ]$
ψ	Yaw angle	$[^\circ]$
ω	Frequency	$[rad/s]$
ω	Angular rate	$[rad/s]$

List of Figures

Figure 2.1 Main components in a flight controller [7].....	3
Figure 2.2 1/7 scaled model of the XB-35 airplane in NACA-Langley 19 foot pressure tunnel [8].....	8
Figure 2.3 Dynamically and aeroelastically scaled model of a multijet cargo airplane on a two cable mount system [9].....	9
Figure 2.4 X-48C remotely piloted liftoff at Edwards air force station for its first test flight [10].....	9
Figure 3.1 Top-level overview of method used in this thesis.....	15
Figure 3.2 Recommended configuration for Skysurfer X8.....	16
Figure 3.3 Aircraft used in flight test campaign [13].....	17
Figure 3.4 (Electrical) System diagram of the X-UAV Skysurfer X8 in flight condition.....	19
Figure 3.5 Data flow between different components on the aircraft an at the ground.....	20
Figure 3.6 Sensor calibration screen within QGroundControl showing compass calibration in progress.....	26
Figure 3.7 Screenshot of QGroundControl during one of the test flights used to test the aircraft functionality.....	27
Figure 3.8 Pitot tube mounting location and tube routing. The approximate free stream velocity direction during 18 m/s cruise is shown in red.....	28
Figure 3.9 Top view of Skysurfer X8 showing body axis X and Y , earth north N , free stream velocity vector V , positive yaw angle Ψ and positive sideslip angle β	29
Figure 3.10 Back view of Skysurfer X8 showing body axis Y and Z , earth down axis D and positive roll angle ϕ	29
Figure 3.11 Overview of method to determine the lateral-directional stability characteristics from flight tests.....	37
Figure 3.12 Single vortex ring element at a surface element [21].....	41
Figure 4.1 Control deflection input to the state space system of the Cessna Ce500 'Citation'.....	46
Figure 4.2 Outputs of the state space system of the Cessna Ce500 'Citation'.....	47
Figure 4.3 Resulting outputs of the estimated state space model compared to the true (original) state space model using Equation (62) as the starting point of the optimization.....	48
Figure 4.4 Output data of the model compared to the noisy output data used for the estimation of the model.....	50
Figure 4.5 Resulting outputs of the estimated state space model from the noisy data compared to the true (original) state space model using Equation (62) as the starting point of the optimization.....	51
Figure 5.1 Indicated Airspeed during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	53
Figure 5.2 Controller inputs during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	53
Figure 5.3 Servo input signals during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	54
Figure 5.4 Aircraft yaw angle during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	54
Figure 5.5 Aircraft barometric altitude during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	54

Figure 5.6 Windspeed north and east during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	55
Figure 5.7 Vibration metrics for the entire flight.....	55
Figure 5.8 Acceleration power spectral density during entire flight	55
Figure 5.9 Percentage of maximum unloaded servo speed during test flight portion containing the aileron sweep on the left and rudder sweep on the right.....	56
Figure 5.10 State space model created from the rudder sweep estimation of the different states compared to the measured states in the rudder sweep flight test.....	57
Figure 5.11 State space model created from the aileron sweep estimation of the different states compared to the measured states in the aileron sweep flight test	58
Figure 5.12 State space model created from the combined flight test estimation of the different states compared to the measured states in both aileron and rudder sweep flight tests. The top four figures show the aileron sweep. The lower 4 figures show the rudder sweep....	61
Figure 5.13 Rendering of the test aircraft including all components in Catia V5	62
Figure 5.14 Inertia of the test aircraft as measured by Catia V5 in the Catia V5 reference frame	63
Figure 5.15 Model of the Skysurfer X8 without pitot tube mount used for the stability analysis.....	64
Figure 5.16 Roll angle as function of time from the estimated state space system from the rudder sweep using maximum aileron and rudder inputs to achieve optimal roll rate	67
Figure 6.1 Different states compared to the measured states in the rudder step flight test with the estimation using the state space model optimized using the rudder sweep data set	69
Figure 6.2 PWM signals sent to servos in the rudder step flight test.....	70
Figure 6.3 Percentage of maximum unloaded servo speed in the rudder step flight test	70
Figure A.1 Front view of the redundant power supply PCB	79
Figure A.2 Back view of the redundant power supply PCB.....	79
Figure A.3 Picture of one of the prototype boards.....	79

List of Tables

Table 2.1 Levels and qualitative suitability of flying qualities [12]	10
Table 2.2 Flight phase categories as described in the MIL-HDBK-1797 [12].....	10
Table 2.3 Aircraft classes as described in the MIL-HDBK-1797 [12]	10
Table 2.4 Recommended maximum roll mode time constant in seconds from MIL-HDBK-1797 [12].....	11
Table 2.5 Recommended minimum time to double amplitude during spiral mode in seconds from MIL-HDBK-1797 [12].....	12
Table 2.6 Roll performance of class I and II aircraft in seconds from MIL-HDBK-1797 [12]	13
Table 2.7 Roll performance of class III aircraft, time to achieve 30 degrees bank angle change in seconds from MIL-HDBK-1797 [12]	13
Table 2.8 Roll performance for class IV aircraft in seconds from MIL-HDBK-1797 [12].....	13
Table 2.9 Recommended minimum Dutch roll frequency and damping coefficient from MIL-HDBK-1797 [12]	14
Table 2.10 Increase in minimal $\zeta d\omega d$ in the case that $\omega d^2 \phi \beta d$ is higher than 20 [rad/sec ²] from MIL-HDBK-1797 [12].....	14
Table 3.1 Dimensions of Skysurfer X8.....	16
Table 3.2 Required components for Skysurfer X8	17
Table 3.3 Components used for performing the test flight	18
Table 3.4 Sensors onboard the Pixhawk 4 flight controller.....	20
Table 3.5 Important characteristics of the magnetometer.....	21
Table 3.6 Important characteristics of the barometer	21
Table 3.7 Important characteristics of the accelerometers and gyroscopes	22
Table 3.8 Important characteristics of the GNSS receiver	22
Table 3.9 Channel allocation in Skysurfer mixer file	23
Table 3.10 PID settings of the PX4 autopilot in stabilized flight mode	23
Table 3.11 Controller channel setup	25
Table 3.12 Characteristics of the MS 4525DO pressure sensor	28
Table 3.13 Start of flight test campaign checklist.....	33
Table 3.14 Before flight checklist.....	34
Table 3.15 I'M SAFE checklist.....	34
Table 3.16 Initial setup flight test briefing.....	35
Table 3.17 Experimental flight test briefing for the roll and yaw axes	36
Table 3.18 Overview of letter variables in Figure 3.11	38
Table 4.1 Asymmetric stability and control derivatives of the Cessna Ce500 'Citation' [19].	45
Table 4.2 Eigenvalue comparison between the original and estimated system rounded to two significant digits.....	48
Table 4.3 Damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model and the original model up to two significant digits..	49
Table 4.4 Eigenvalue comparison between the original and the estimated system from noisy data up to two significant digits	52
Table 4.5 Damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from noisy data and the original model up to two significant digits.....	52
Table 5.1 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the rudder sweep flight test.....	57

Table 5.2 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the aileron sweep flight test	59
Table 5.3 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the combined flight test	60
Table 5.4 Inertial characteristics of the test aircraft in the aerodynamic reference frame	63
Table 5.5 Airfoils used in the XFLR5 model of the Skysurfer X8 test aircraft.....	63
Table 5.6 Lateral stability derivatives of the Skysurfer X8 as found by the XFLR5 analysis	64
Table 5.7 Eigenvalues found in the XFLR5 stability model with their resulting damping, natural frequency and time to half amplitude	64
Table A.1 Components used on redundant variable power supply module, where D are diodes, C capacitors, R resistors, H inductors, U varying chips and M connectors	78
Table A.2 Address selection resistor value for PAC1932T-I/JQ from Mauser	78
Table C.3 Weight and approximate location of center of gravity used for the Catia V5 model with respect to the origin which is at the tip of the nose as measured by Juffermans [13]	85

Contents

Summary	V
Acknowledgements	VII
Nomenclature	IX
List of Figures	XIII
List of Tables	XV
Contents	XVII
1 Introduction	1
2 Background	3
2.1 Extended Kalman filter used in PX4	3
2.2 Previous use of Scaled flight testing	7
2.3 Military specifications for lateral-directional aircraft dynamics	9
3 Methodology	15
3.1 Aircraft used for flight tests	15
3.1.1 Skysurfer X8 overview in test flight condition	15
3.1.2 Flight controller	20
3.1.3 Control servos	24
3.1.4 Receiver, transmitter and telemetry system	25
3.1.5 Airspeed measurement system	27
3.2 State space system for aircraft lateral-directional dynamics	29
3.3 Flight test plans	33
3.4 Method used to determine the lateral-directional stability characteristics of the aircraft from flight tests	37
3.5 Method used to determine the lateral-directional stability characteristics of the aircraft from a vortex lattice method	41
4 Verification	45
4.1 Using exact output data of a model to estimate the model	45
4.2 Using noisy output data from a model to estimate the model	50
5 Results	53
5.1 Flight test data	53
5.2 Resulting state space models	56
5.2.1 Rudder sweep only	56
5.2.2 Aileron sweep only	57
5.2.3 Combination of rudder and aileron sweep	59
5.3 Model of aircraft dynamics using vortex lattice method	62
5.4 Comparison between flight test and vortex lattice method and military standards ..	65
6 Validation	69
7 Conclusions and Recommendations	71
7.1 Conclusions	71

7.2	Recommendations.....	73
8	Bibliography	75
Appendix A	Redundant power supply module.....	77
Appendix B	Mixer and airframe PX4.....	81
Appendix C	Mass and location of test aircraft parts.....	85
Appendix D	Python code	87
Appendix E	Matlab code	101

1 Introduction

The earth is experiencing global warming due to excessive CO₂ emissions. There is a concern of the impact of aircraft on climate change. Using data from the UK, 6% of total CO₂ emissions are related to aviation. Of this around 90% is caused by international flights [1]. From this it can be seen that the development of a more efficient international / intercontinental aircraft will have the greatest benefit to the environment.

The current tube and wings design has been optimized for a long time and seems to be reaching its maximum efficiency. A number of new aircraft configurations have been theorized that could improve the efficiency of aircraft drastically like the Flying V [2] and the blended wing body aircraft [3]. In order to progress from a theoretical model and simulations to an actual aircraft the models that are created have to be validated. One of the methods that has been used for validating new concepts is scaled flight testing. Whilst wind tunnel tests have their advantages, the use of a free flying model allows to compliment the wind tunnel data with data that is recorded in conditions that are analogous to where the final aircraft will be operating [4]. This thesis shows a method to analyze the lateral-directional stability and controllability characteristics of a radio-controlled aircraft, which is one of the aspects that can be validated using scaled flight testing.

The advantage of (scaled) flight testing is the reduced cost compared to wind tunnel tests, whilst limiting safety risks by limiting the size and speed of the aircraft. By eliminating the pilot(s) from the aircraft the risks to the occupant(s) related to a fatal crash are eliminated. This means that experiments where dangerous conditions can be achieved, from either structural load constraints or from uncertain outcomes can be performed at low risk. These tests can include post-stall characteristics and flying with given system failures [5]. The accuracy of the data recorded in a free flight, depending on the ability to produce precision maneuvers, can be better than those achieved in wind tunnel tests [6]. In order to optimize the safety of the flight tests and limit risks to the aircraft tested, a redundant power supply board is designed which is shown in Appendix A. This is not used in the aircraft used for the test flights, but can be used in larger and more expensive models that operate with multiple batteries.

The research questions that are answered in this thesis are shown below. The focus in the thesis is put on the Skysurfer X8, however, it should be noted that the method used in this thesis can be used for determining the lateral-directional dynamics of any remotely controlled aircraft configuration.

- 1 What are the lateral-directional controllability and stability characteristics of the Skysurfer X8?
 - 1.1 How can a linearized state space model for the lateral-directional stability characteristics of the aircraft be determined from flight tests?
 - 1.2 How does the linearized state space model of the aircraft determined from the flight test compare to the measurements taken during the test flight?
 - 1.3 How do the lateral-directional stability characteristics compare between the state space model obtained from flight tests and the state space model obtained from a vortex lattice method?
- 2 Are the lateral-directional handling characteristics of the Skysurfer X8 satisfactory?
 - 2.1 What are the lateral-directional handling characteristics according to the determined linearized state space model(s)?
 - 2.2 How do the handling characteristics of the aircraft compare to the military specifications MIL-HDBK-1797 (damping ratio and frequency of spiral, Dutch roll and roll subsidence mode as well as the maximum roll performance)?

This thesis is built up as follows. First some background information on the extended Kalman filter, scaled flight testing and military specifications for the lateral-directional aircraft dynamics is provided. Next an overview of the methods used to achieve the final answers is provided which includes the aircraft itself including its systems, a derivation of the state space system for lateral-directional stability and controllability, the flight test plans and an overview of the analysis programs used. Later a verification of the method used is performed. After this the results are shown from the flight test as well as the vortex lattice method simulation. Once the results are discussed, the model is validated using a different set of maneuvers. Lastly some conclusion and recommendations are provided.

2 Background

In the background the extended Kalman Filter, previous use of scaled flight testing and military specifications for the handling characteristics are described. The extended Kalman filter is indirectly used within the thesis. The PX4 software that is run on the flight controller uses an extended Kalman filter to combine all sensor data that are obtained and provide the states. These states are saved onboard and used for the thesis. The previous use of scaled flight testing shows some examples in which smaller versions of an aircraft are used to determine aerodynamic characteristics. This thesis focusses on a toy remotely piloted aircraft, but the method can be used for any type of remotely piloted aircraft including scaled models of for example the Flying V. The military specifications are used in aircraft design since the research that was performed to obtain the specifications has looked at what makes an aircraft easy to fly or barely flyable for a pilot. In order to look at the characteristics found in the flight test from a pilot perspective, it makes sense to compare the results to these military specifications.

2.1 Extended Kalman filter used in PX4

The flight control and navigation system onboard RC aircraft receives commands from the ground like transmitter inputs from the pilot and planned navigation missions, obtains sensor data from accelerometers, gyroscopes, magnetometers, barometer, GNSS and possibly a pitot tube. The flight control system uses these inputs to determine the signals to the engine speed controllers and servos in order to keep the aircraft stable and perform the maneuvers that are commanded. The flight controller uses sensor fusion to determine the current attitude and speed of the aircraft. A proportional integral derivative controller is used to determine the required controls. Lastly a mixer is used to convert the required controls to signals that are sent to the servos and engine speed controllers. A schematic of the components of the flight controller is provided in Figure 2.1.

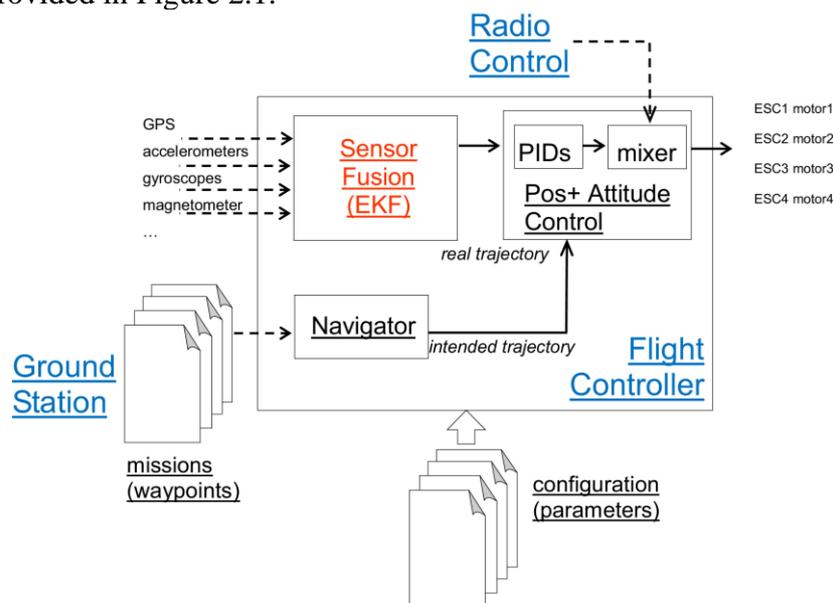


Figure 2.1 Main components in a flight controller [7]

For the RC aircraft to determine its attitude and location robustly an approach using fusion of complimentary sensors is used. In this way an RC aircraft would be able to continue

operations when signals from GNSS are jammed or spoofed or any system of the RC aircraft provides noisy or incorrect measurements. In the control system of the Skysurfer X8 an Extended Kalman Filter (EKF) is used.

Before any sensor data is used to update the prediction of the aircraft state, first a step is performed to check if the received sensor data is correct. In the tracking filter when an update is performed, first the residual, also sometimes called innovation, is calculated. An example with position x and y is provided where sub-index p stands for the predicted state and sub-index m for the measured state. The innovation r is calculated as an average over a certain amount of time with N_T measurements as can be seen in Equation (1) below. The variable k is used to denote the different instances for the estimation. The innovation can be thought of as the sum of the average of the squared errors of all different parameters that are estimated [7].

$$r = \frac{1}{N_T} \sum_k (x_p(k) - x_m(k))^2 + (y_p(k) - y_m(k))^2 \quad (1)$$

Next the averaged normalized innovation r_n is calculated as shown in Equation (2) below. This calculation uses a covariance matrix for taking into account the uncertainties in the prediction and observation. This matrix is called the Mahalanobis distance matrix $S(k)$ [7].

$$r_n = \frac{1}{N_T} \sum_k [x_p(k) - x_m(k) \quad y_p(k) - y_m(k)] S(k)^{-1} \begin{bmatrix} x_p(k) - x_m(k) \\ x_p(k) - y_m(k) \end{bmatrix} \quad (2)$$

The Mahalanobis distance matrix is the sum of covariances in predicted and measured vector with the assumption that the errors are independent. The predicted covariance depends on the assumed uncertainty of the model and the measurement covariance depends on the sensor used. In this P_p is the covariance matrix for the predicted vector, R is the covariance matrix of the measurement vector and σ denotes the covariance between the variables in the subscript. The Mahalanobis distance matrix is shown in Equation (3) below [7].

$$S(k) = P_p(k) + R(k) = \begin{bmatrix} \sigma_{x_p}^2(k) & \sigma_{x_p y_p}(k) \\ \sigma_{x_p y_p}(k) & \sigma_{y_p}^2(k) \end{bmatrix} + \begin{bmatrix} \sigma_{x_m}^2(k) & \sigma_{x_m y_m}(k) \\ \sigma_{x_m y_m}(k) & \sigma_{y_m}^2(k) \end{bmatrix} \quad (3)$$

To verify that the assumed uncertainty of the model used is correct the consistency of the residual r_n is checked. For a correct model the r_n should follow a normalized Chi-squared χ^2 distribution with the degrees of freedom equal to the number of measurements minus the dimension of the Mahalanobis distance matrix. This is described Equation (4) below.

$$r_n \sim \frac{1}{N_T} \chi_{(2N_T - \dim(S))}^2 \quad (4)$$

If the full covariance matrix is not available, only the variances on the diagonal are considered which results in Equation (5) below [7].

$$r_n = \frac{1}{N_T} \sum_k \left(\frac{(x_p(k) - x_m(k))^2}{\sigma_{x_p}^2 + \sigma_{x_m}^2} + \frac{(y_p(k) - y_m(k))^2}{\sigma_{y_p}^2 + \sigma_{y_m}^2} \right) \quad (5)$$

When it is discovered that a certain data source produces inconsistent data, it might be rebooted or switched off. This is usually done only when the normalized innovation is more than 5 standard deviations.

Once the sensor data have been verified they can be used in the navigation filter. Below the centralized extended Kalman filter is described. It can help the inertial navigation system by loosely coupling the GNSS position and velocity measurements. In this way the GNSS measurements, that are usually done at a relatively low rate like 1 Hz, can help update the bias of different sensors that operate at a much higher rate like 50Hz to make the aircraft able to perform preplanned maneuvers or cope with any external disturbances like turbulence or gusts. The required output of the GNSS/INS filter should contain the attitude, position, velocity and biases for the measurements. A representative dynamic model that integrates local sensed inputs, absolute position and sensor bias dynamics with an uncertainty model for the predictions is shown below. It uses the east, north, down frame with an origin at the arming point at the start of the flight.

A state vector $\bar{x}(k)$ with 16 dimensions is used containing 3 dimensions of position $\bar{p}_n(k)$, 3 dimensions of velocity $\bar{v}_n(k)$, 4 dimensions of attitude(quaternions) $\bar{q}_n(k)$, 3 dimensions for bias corrections of the acceleration in the body frame $\bar{b}_a(k)$ and 3 dimensions for bias corrections of the angular rate in the body frame $\bar{b}_\omega(k)$ as shown in Equation (6) below. The superscript t denotes a transpose of the vector [7].

$$\bar{x}(k) = [\bar{p}_n^t(k) \quad \bar{v}_n^t(k) \quad \bar{q}_n^t(k) \quad \bar{b}_a^t(k) \quad \bar{b}_\omega^t(k)]^t \quad (6)$$

Different states can easily be added to the state vector to integrate different measurements and improve the overall accuracy. An example is given in Equation (7) below, where the barometric height $b_h(k)$ has been added to the original state vector from Equation (6) to obtain an extended state vector \bar{x}^E [7].

$$\bar{x}^E(k) = [\bar{x}(k)^t \quad b_h(k)]^t \quad (7)$$

Within the extended Kalman filter the GNSS position and barometric height are used as observations. The inertial measurement unit is used as a control input.

The GNSS position measurement \bar{p}_{GNSS} is considered to be the actual position plus an observation noise n_{GNNS} in all directions as is shown in Equation (8) below. Not only the x and y position, but also the height z is given by the GNNS signal. The position variables without a subscript denote the exact position at the time instance [7].

$$\bar{p}_{GNSS}(k) = \begin{bmatrix} x_m(k) \\ y_m(k) \\ z_m(k) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} n_{GNNS_x}(k) \\ n_{GNNS_y}(k) \\ n_{GNNS_z}(k) \end{bmatrix} \quad (8)$$

The barometric height Z_{baro} is the actual height plus an observation noise n_{baro_z} and a bias of the barometer b_{baro_z} as shown in Equation (9) below [7].

$$Z_{baro}(k) = [z_m(k)] = z(k) + b_{baro_z}(k) + n_{baro_z}(k) \quad (9)$$

The ideal 3-dimensional acceleration $\bar{a}_u(k)$ and 3-dimensional angular rate $\bar{\omega}_u(k)$ in the body fixed reference frame are used in the as control input \bar{u} in the extended Kalman filter. This control input is shown in Equation (10) below.

$$\bar{u}(k) = [\bar{a}_u^t(k) \quad \bar{\omega}_u^t(k)]^t \quad (10)$$

These accelerations and angular rates are measured by the inertial measurement unit resulting in $\bar{a}_m[k]$ and $\bar{\omega}_m[k]$. The measurements have to be corrected with the estimated biases for the accelerations and angular rates resulting in a bias corrected control input \hat{u} with the subscript b denoting a bias corrected value. The bias corrected control input is shown in Equation (11) below.

$$\hat{u}(k) = [\bar{a}_b^t(k) \quad \bar{\omega}_b^t(k)]^t \quad (11)$$

The bias corrected acceleration is shown in Equation (12) below. In this \bar{b}_a is the acceleration bias vector of the accelerometer.

$$\bar{a}_b(k) = \bar{a}_m(k) - \bar{b}_a(k) \quad (12)$$

To obtain the bias corrected angular rates, the Coriolis effect caused by the earth rotation $\bar{\omega}_{En}$ transformed to the body reference frame using the transformation matrix C_n^b as well as the angular rates bias $\bar{b}_\omega(k)$ is subtracted to create the bias corrected control input as shown in Equation (13) below. In this p is the roll rate, q the pitch rate and r the yaw rate.

$$\bar{\omega}_b(k) \equiv [p(k) \quad q(k) \quad r(k)]^t = \bar{\omega}_m(k) - C_n^b(k)^{-1} \bar{\omega}_{En} - \bar{b}_\omega(k) \quad (13)$$

To obtain the Coriolis effect in the body reference frame, a projection of the earth rotation from the east-north-up, subscript n , to the body reference frame, superscript b , $C_n^b(k)^{-1}$ is made. This transformation uses the quaternion matrix shown in Equation (14) below. In this q_0 , q_1 , q_2 and q_3 are the four quaternions [7].

$$C_n^b(k) = \begin{bmatrix} 1 - 2\{q_2(k)^2 + q_3(k)^2\} & 2\{q_1(k)q_2(k) - q_0(k)q_3(k)\} & 2\{q_1(k)q_3(k) + q_0(k)q_2(k)\} \\ 2\{q_1(k)q_2(k) + q_0(k)q_3(k)\} & 1 - 2\{q_1(k)^2 + q_3(k)^2\} & 2\{q_2(k)q_3(k) - q_0(k)q_1(k)\} \\ 2\{q_1(k)q_3(k) - q_0(k)q_2(k)\} & 2\{q_2(k)q_3(k) + q_0(k)q_1(k)\} & 1 - 2\{q_1(k)^2 + q_2(k)^2\} \end{bmatrix} \quad (14)$$

The extended Kalman filter uses the state vector and a dynamic stochastic model to define the timewise evolution of the state vector as shown in Equation (15) below. In this $\bar{x}(t)$ is the state vector, $\dot{\bar{x}}(t)$ the time derivative of the state vector, $\bar{u}(t)$ is the deterministic control input as determined by the inertial measurement unit and $\bar{v}(t)$ is an unobservable additional noise model to take deviations from the predictions into account.

$$\dot{\bar{x}}(t) = \frac{d\bar{x}(t)}{dt} = \bar{y}(t, \bar{x}(t), \bar{u}(t), \bar{v}(t)) \quad (15)$$

The prediction is created by performing an integration of the nonlinear differential equations. The numerical variant of this prediction is shown in Equation (16) below. In this $A(k)$ is the attitude transition matrix shown in Equation (17). $U(k)$ corrects the velocity computed from the control input shown in Equation (20). The entire prediction is made in the earth fixed east-north-up reference frame.

$$\begin{bmatrix} \bar{p}_n[k+1] \\ \bar{v}_n[k+1] \\ \bar{q}_n[k+1] \\ \bar{b}_a[k+1] \\ \bar{b}_\omega[k+1] \end{bmatrix} = \begin{bmatrix} I_3 & \Delta t I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 & 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{4 \times 3} & 0_{4 \times 3} & A(k) & 0_{4 \times 3} & 0_{4 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} & 0_{3 \times 3} & I_3 \end{bmatrix} \begin{bmatrix} \bar{p}_n(k) \\ \bar{v}_n(k) \\ \bar{q}_n(k) \\ \bar{b}_a(k) \\ \bar{b}_\omega(k) \end{bmatrix} + \Delta t \begin{bmatrix} 0_{3 \times 1} \\ U(k) \\ 0_{4 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix} \quad (16)$$

The attitude transition matrix $A(k)$ is shown in Equation (17) below. This attitude transition matrix is dependent on $s(k)$ which is a variable dependent on $\bar{\omega}_b(k)$, the corrected angular velocity in the body reference frame as shown in Equation (18) and a matrix $\Omega(k)$ to transform the pitch, roll and yaw rates to quaternions as shown in Equation (19).

$$A(k) = \left(I_4 \cos(s(k)) - \frac{1}{2} \Delta t \Omega[k] \frac{\sin(s(k))}{s(k)} \right) \quad (17)$$

In the attitude transition matrix $s(k)$ is calculated using Equation (18) shown below.

$$s(k) = \frac{1}{2} \Delta t \sqrt{(p(k))^2 + (q(k))^2 + (r(k))^2} \quad (18)$$

The last element required to calculate the attitude transition matrix is $\Omega(k)$. This matrix shows the derivative of the quaternions as a function of the pitch, yaw and roll rates as shown in Equation (19) below.

$$\Omega(k) = \begin{bmatrix} 0 & p(k) & q(k) & r(k) \\ -p(k) & 0 & -r(k) & q(k) \\ -q(k) & r(k) & 0 & -p(k) \\ -r(k) & -q(k) & p(k) & 0 \end{bmatrix} \quad (19)$$

$U(k)$ corrects the velocity computed from the control input, which is the acceleration vector in the inertial reference frame to the east north up reference frame with the gravitational effect included as shown in Equation (20) below. In this \bar{g} is the gravitational acceleration vector [7].

$$U(k) = C_n^b(k) \bar{a}_b(k) + \bar{g} \quad (20)$$

2.2 Previous use of Scaled flight testing

A lot of early aircraft development was confidential. So also, the research of a XB-35 model in 1946. It was added to the public NASA documents in 2013 after being made unclassified. The report is on the effects of adding a leading-edge tip slot in the pursuit of solving the poor stick fixed longitudinal stability at low speeds that had been attributed to the highly swept back and tapered wing. Furthermore, it looks into the three-dimensional flow to determine if it is possible to solve the elevon up-floating tendencies at high angles of attack resulting in stick free instability. Due to the limited size of wind tunnels available, a 1/7 scale semispan model with a reflection plate was used to maximize the size of the model and therefor obtain the information at relatively high Reynolds numbers and at low Mach numbers. The only corrections made are for the wind tunnel effects. There is no mention with regard to aeroelastically scaling the model. The test was performed at Reynolds number 7.5 million and Mach number 0.12. The Reynolds number of the full-scale aircraft at stall is not given,

but it can be assumed that this is higher. Looking at the report it seems that the aircraft was geometrically scaled. Later all forces were put to non-dimensional numbers to make them applicable to the full-scale aircraft. For wind tunnel tests dynamic scaling does not achieve anything, but the effect of the Reynolds number was not addressed directly by aerodynamic scaling, but effort was made to test at the highest possible Reynolds number to make the wind tunnel test as representative as possible for the full scale aircraft [8].

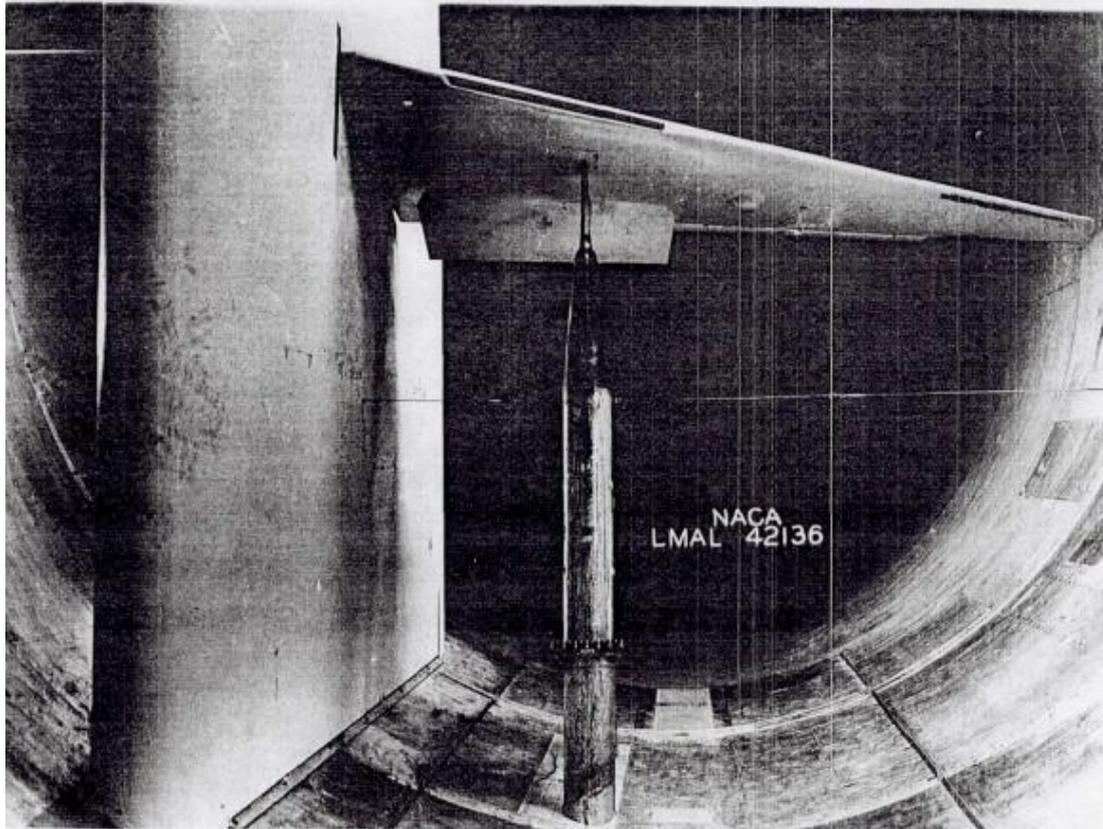


Figure 2.2 1/7 scaled model of the XB-35 airplane in NACA-Langley 19 foot pressure tunnel [8]

An investigation into the aileron effectiveness, reversal boundary and roll rates below the reversal boundary was performed to demonstrate the technique to determine these parameters from an aeroelastically and dynamically scaled model. The test was performed after a low speed flutter model showed a sufficient operating margin, but the analytical studies based on later aerodynamic data predicted a lower roll reversal boundary. The manufacturer was also interested in the roll rates obtainable close to roll reversal so this was also tested. The test with the aeroelastically and dynamically scaled model was compared to the results obtained in flight and it showed that the aileron effectiveness and damping in roll as well as the roll rates at speeds somewhat below the roll reversal boundary were successfully determined [9].

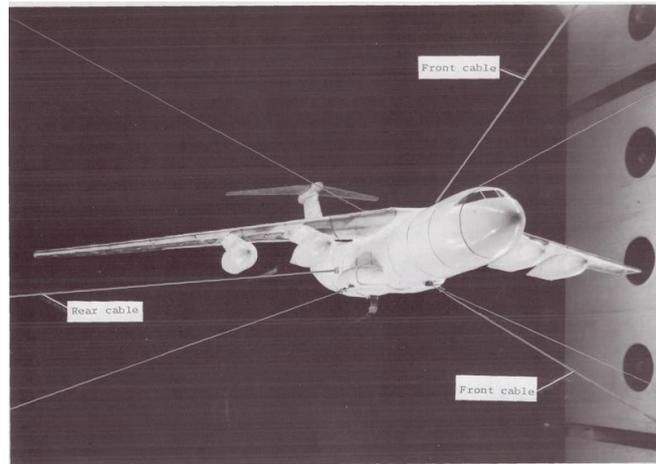


Figure 2.3 Dynamically and aeroelastically scaled model of a MultiJet cargo airplane on a two cable mount system [9]

Recently free flight models have been used to determine the performance of novel configurations. One of which is the X-48 series. The latest X-48C was used to evaluate low speed stability and control characteristics [10]. There is more information on the X-48B which has performed several flight tests in order to validate the stability and control of the full-scale aircraft across a large portion of the low speed regime as well as recovery from idle thrust stalls with slats extended and retracted. Parameter identification techniques have been used to determine the static and dynamic stability and control derivatives at a range of flight conditions using control doublets and frequency sweeps of individual control surfaces [11]. By using these models, relatively cheap tests can be performed to determine the feasibility of new designs, but it must be noted that scaled flight test results in the feasibility study of new aircraft configurations must be looked at carefully to determine what the effects of the lower Reynolds number might mean for the full-scale aircraft.



Figure 2.4 X-48C remotely piloted liftoff at Edwards air force station for its first test flight [10]

2.3 Military specifications for lateral-directional aircraft dynamics

Within MIL-HDBK-1797 aircraft behavior is described in 3 levels as shown in Table 2.1. The flight phases as used in the description of Table 2.1 are shown in Table 2.2. The aircraft classes are shown in Table 2.3. For each class an addition of a C or L can be made which stand for Carrier and Land based aircraft respectively. Within the handbook a lot of different supporting data are shown from different researchers with respect to the pilot perception that lead to the different requirements [12]. The only requirements that are shown are those that can be obtained from a linearized model of the lateral-directional dynamics of an aircraft,

namely the roll subsidence, spiral and Dutch roll. Also the roll performance requirements are shown as this can also be determined from the model.

Table 2.1 Levels and qualitative suitability of flying qualities [12]

Satisfactory Level 1	Flying qualities clearly adequate for the mission Flight Phase. Desired performance is achievable with no more than minimal pilot compensation.
Acceptable Level 2	Flying qualities adequate to accomplish the mission Flight Phase, but some increase in pilot workload or degradation in mission effectiveness, or both, exists.
Controllable Level 3	Flying qualities such that the aircraft can be controlled in the context of the mission Flight Phase, even though pilot workload is excessive or mission effectiveness is inadequate, or both. The pilot can transition from Category A Flight Phase tasks to Category B or C Flight Phases, and Category B and C Flight Phase tasks can be completed.

Table 2.2 Flight phase categories as described in the MIL-HDBK-1797 [12]

Category A	These nonterminal Flight Phases that require maneuvering, precision tracking, or precise flight-path control. Included in this Category are: a. Air-to-air combat (CO) b. Ground attack (GA) c. Weapon delivery/launch (WD) d. Aerial recovery (AR) e. Reconnaissance (RC) f. In-flight refueling (receiver) (RR) g. Terrain following (RF) h. Antisubmarine search (AS) i. Close formation flying (FF) j. Low-altitude parachute extraction (LAPES) delivery
Category B	Those nonterminal Flight Phases that are normally accomplished using gradual maneuvers and without precision tracking, although accurate flight-path control may be required. Included in this Category are: a. Climb (CL) b. Cruise (CR) c. Loiter (LO) d. In-flight refueling (tanker) (RT) e. Descent (D) f. Emergency descent (ED) g. Emergency deceleration (DE) h. Aerial delivery (AD)
Category C	Terminal Flight Phases are normally accomplished using gradual maneuvers and usually require accurate flight-path control. Included in this Category are: a. Takeoff (TO) b. Catapult takeoff (CT) c. Approach (PA) d. Waveoff/go-around (WO) e. Landing (L)

Table 2.3 Aircraft classes as described in the MIL-HDBK-1797 [12]

Class I	Small light aircraft such as: Light utility, Primary trainer, Light observation
Class II	Medium weight, low-to-medium maneuverability aircraft such as: Heavy utility/search and rescue, Light or medium transport/cargo/tanker, Early warning/electronic countermeasures/airborne command, control, or communications relay, Antisubmarine, Assault transport, Reconnaissance, Tactical bomber, Heavy attack, Trainer for Class II
Class III	Large, heavy, low-to-medium maneuverability aircraft such as: Heavy transport/cargo/tanker, Heavy bomber, Patrol/early warning/electronic countermeasures/airborne command, control, or communications relay, Trainer for Class III
Class IV	High-maneuverability aircraft such as: Fighter-interceptor, Attack

The first stability aspect described in the handbook MIL-HDBK-1797 is the roll mode, also referred to as roll subsidence. This mode describes the delay of the aileron input on the roll rate achieved. Pilots get the perception of having higher roll precision in the case of a lower roll mode time constant. The recommended maximum values for the roll mode time constant are shown in Table 2.4. For a normal aircraft the effect of a step input on the ailerons causes a behavior that is described by Equation (21) below where T_R is the roll rate time constant, p_∞ the roll rate after an infinite time and t is the time [12].

$$p(t) = p_\infty \left(1 - e^{-\frac{t}{T_R}} \right) \quad (21)$$

Table 2.4 Recommended maximum roll mode time constant in seconds from MIL-HDBK-1797 [12]

Flight phase category	Class	Level 1 [s]	Level 2 [s]	Level 3 [s]
A	I, IV	1.0	1.4	10
	II, III	1.4	3.0	
B	All	1.4	3.0	
C	I, II-C, IV	1.0	1.4	
	II-L, III	1.4	3.0	

With respect to the spiral stability the MIL-HDBK-1797 recommends a time to double amplitude as shown in Table 2.5. It is said that during operations where the pilot is fully attended, the spiral mode does not pose a big problem, however during low-gain tasks, the spiral mode can be a nuisance or even dangerous in the case the time to double is too low. This mainly is due to the fact that in the case the time to double for the spiral mode is too low, the attention of the pilot has to be constantly with flying the aircraft causing a high pilot workload and making it intolerable for longer durations. The spiral mode time constant T_s can be approximated by Equation (22) below. In this g is the gravitational acceleration, V the airspeed and C_{L_1} the aircraft lift coefficient in a 1g, or straight and level flight condition which is $\frac{W}{qS}$ or weight over the dynamic pressure multiplied by the wing planform area. I_{XX} is the mass moment of inertia around the X axis. I_{ZZ} is the mass moment of inertia around the Z axis. I_{XZ} is the cross mass moment of inertia around the X,Z axis. C_{l_β} is the normalized coefficient that shows the moment around the X axis caused by the sideslip angle β . C_{l_p} is the normalized coefficient that shows the moment around the X axis caused by the roll rate p . C_{l_r} is the normalized coefficient that shows the moment around the X axis caused by the yaw rate r . C_{n_β} is the normalized coefficient that shows the moment around the Z axis caused by the sideslip angle β . C_{n_p} is the normalized coefficient that shows the moment around the Z axis caused by the roll rate p . C_{n_r} is the normalized coefficient that shows the moment around the Z axis caused by the yaw rate r [12].

$$\frac{1}{T_s} \approx \frac{g}{V} \frac{\left(\begin{array}{cc} C_{n_r} + \frac{C_{l_r} I_{XZ}}{I_{XX}} & C_{l_\beta} + \frac{C_{n_\beta} I_{XZ}}{I_{ZZ}} \\ \frac{C_{l_r} + \frac{C_{n_r} I_{XZ}}{I_{ZZ}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}} - \frac{\frac{C_{n_r} + \frac{C_{l_r} I_{XZ}}{I_{XX}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}} \cdot \frac{C_{l_\beta} + \frac{C_{n_\beta} I_{XZ}}{I_{ZZ}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}}}{C_{n_\beta} + \frac{C_{l_\beta} I_{XZ}}{I_{XX}}} \end{array} \right)}{\frac{C_{l_p} + \frac{C_{n_p} I_{XZ}}{I_{ZZ}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}} - \left(\frac{C_{n_p} + \frac{C_{l_p} I_{XZ}}{I_{XX}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}} - 2 \frac{I_{ZZ}}{mb^2} C_{L_1} \right) \frac{\frac{C_{l_\beta} + \frac{C_{n_\beta} I_{XZ}}{I_{ZZ}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}}}{C_{n_\beta} + \frac{C_{l_\beta} I_{XZ}}{I_{XX}}} \frac{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}}{1 - \frac{I_{XZ}^2}{I_{XX} I_{ZZ}}}} \quad (22)$$

Table 2.5 Recommended minimum time to double amplitude during spiral mode in seconds from MIL-HDBK-1797 [12]

Flight phase category	Level 1 [s]	Level 2 [s]	Level 3 [s]
A and C	12	8	4
B	20	8	4

The MIL-HDBK-1797 prescribes roll control power to aircraft in the different phases of flight. All times are measured from the time that the control force is first applied. The input should be as abrupt as possible. Because a true step input is never given, the starting time of the maneuver may also be taken at the time where half of the control input is given assuming the control input is given as abrupt as possible. The pitch control should be kept fixed. The yaw control can be used to reduce sideslip that retards the roll rate except for class IV aircraft in level 1 or category C flight phases in level 1 or 2. In all other cases the yaw control should be free. For some flights at high angles of attack the rudder pedals are used to roll due to the large adverse yaw due to the roll control effectors. In these cases the aileron rudder interconnect can be used to achieve the roll performance required. For the takeoff flight phase the time can be increased by the rolling moment of inertia at takeoff divided by the largest rolling moment of inertia possible at maximum landing weight conditions. All times are shown in Table 2.6, Table 2.7 and Table 2.8. For class IV aircraft there exist further requirements for air to air combat and ground attack flight phases, however these are not shown here [12].

Table 2.6 Roll performance of class I and II aircraft in seconds from MIL-HDBK-1797 [12]

Class	Level	Category A [s]		Category B [s]		Category C [s]	
		60 deg	45 deg	60 deg	45 deg	30 deg	25 deg
I	1	1.3		1.7			1.3
I	2	1.7		2.5			1.8
I	3	2.6		3.4			2.6
II-L	1		1.4		1.9	1.8	
II-L	2		1.9		2.8	2.5	
II-L	3		2.8		3.8	3.6	
II-C	1		1.4		1.9		1.0
II-C	2		1.9		2.8		1.5
II-C	3		2.8		3.8		2.0

Table 2.7 Roll performance of class III aircraft, time to achieve 30 degrees bank angle change in seconds from MIL-HDBK-1797 [12]

Level	Minimum speed	Maximum speed	Category A [s]	Category B [s]	Category C [s]
1	Vmin	1.8 Vmin	1.8	2.3	2.5
1	1.8 Vmin	0.7 Vmax	1.5	2.0	2.5
1	0.7 Vmax	Vmax	2.0	2.3	2.5
2	Vmin	1.8 Vmin	2.4	3.9	4.0
2	1.8 Vmin	0.7 Vmax	2.0	3.3	4.0
2	0.7 Vmax	Vmax	2.5	3.9	4.0
3	Vmin	Vmax	3.0	5.0	6.0

Table 2.8 Roll performance for class IV aircraft in seconds from MIL-HDBK-1797 [12]

Level	Minimum speed	Maximum speed	Category A [s]			Category B [s]	Category C [s]
			30 deg	50 deg	90 deg		
			30 deg	50 deg	90 deg	90 deg	30 deg
1	Vmin	Vmin+20kts	1.1			2.0	1.1
1	Vmin+20kts	1.4 Vmin	1.1			1.7	1.1
1	1.4 Vmin	0.7 Vmax			1.3	1.7	1.1
1	0.7 Vmax	Vmax		1.1		1.7	1.1
2	Vmin	Vmin+20kts	1.6			2.8	1.3
2	Vmin+20kts	1.4 Vmin	1.5			2.5	1.3
2	1.4 Vmin	0.7 Vmax			1.7	2.5	1.3
2	0.7 Vmax	Vmax		1.3		2.5	1.3
3	Vmin	Vmin+20kts	2.6			3.7	2.0
3	Vmin+20kts	1.4 Vmin	2.0			3.4	2.0
3	1.4 Vmin	0.7 Vmax			2.6	3.4	2.0
3	0.7 Vmax	Vmax		2.6		3.4	2.0

MIL-HDBK-1797 prescribes the dynamic lateral-directional response characteristics, also known as the Dutch roll characteristics, of the aircraft. There is a minimum frequency as well as a minimum damping as well as a minimum frequency multiplied by damping prescribed as shown in Table 2.9. It is important to check this characteristic using the highest yawing moment of inertia to receive the lowest frequency. The A-10 design called for even higher requirements on the Dutch roll damping coefficient to achieve the best air to ground accuracy. Only for class III aircraft there is a maximum damping coefficient required of 0.7 from the combined frequency multiplied by damping coefficient. If this is the case, the damping coefficient should be 0.7 or higher and the minimal frequency has to be met. Finally for aircraft that have high Dutch roll frequencies, the combined damping multiplied by frequency should be increased according to Table 2.10 to account for possible turbulence effects on the aircraft. Stability augmentation systems can be used to achieve high damping ratios for the Dutch roll. The system specifically to increase the Dutch roll damping coefficient is the yaw damper [12].

Table 2.9 Recommended minimum Dutch roll frequency and damping coefficient from MIL-HDBK-1797 [12]

Level	Flight phase category	Class	Minimal damping ζ_d	Minimal $\zeta_d \omega_d$ [rad/sec]	Minimal frequency ω_d [rad/sec]
1	A (CO, GA, RR, TF, RC, FF, AS)	All	0.4	0.4	1.0
	A	I, IV	0.19	0.35	1.0
		II, III	0.19	0.35	0.4
	B	All	0.08	0.15	0.4
	C	I, II-C, IV	0.08	0.15	1.0
II-L, III		0.08	0.10	0.4	
2	All	All	0.02	0.05	0.4
3	All	All	0	-	0.4

Table 2.10 Increase in minimal $\zeta_d \omega_d$ in the case that $\omega_d^2 \left| \frac{\phi}{\beta} \right|_d$ is higher than 20 [rad/sec²] from MIL-HDBK-1797 [12]

Level 1	$\Delta \zeta_d \omega_d = 0.014 \left(\omega_d^2 \left \frac{\phi}{\beta} \right _d - 20 \right)$
Level 2	$\Delta \zeta_d \omega_d = 0.009 \left(\omega_d^2 \left \frac{\phi}{\beta} \right _d - 20 \right)$
Level 3	$\Delta \zeta_d \omega_d = 0.005 \left(\omega_d^2 \left \frac{\phi}{\beta} \right _d - 20 \right)$

Given the size of aircraft used for the flight tests in this thesis, the comparisons will be made to the Class I aircraft. Because the system is created for a cruise condition, Flight phase B will be used to analyze the aircraft performance. In the case of scaled flight testing, one should estimate the full-scale characteristics and compare them to the class that suits the description of the full-scale aircraft best.

3 Methodology

For the completion of this thesis, certain steps had to be taken to come to the results. First the aircraft used for the flight tests had to be designed and built. For the design part, the focus is on the data acquisition. During the design of the aircraft a state space system for lateral-directional dynamics of the aircraft is derived. Using this model flight tests can be constructed to find the actual lateral directional characteristics of the aircraft. These flight tests have limitations from regulations and aircraft/pilot performance. Once the flight tests have been performed the data can be used to obtain the lateral directional dynamic model of the aircraft. The programs used in the analysis of the data as well as the program used to obtain an independent model using a vortex lattice method are the last steps in answering the research questions. A top-level overview of the method is shown in Figure 3.1.

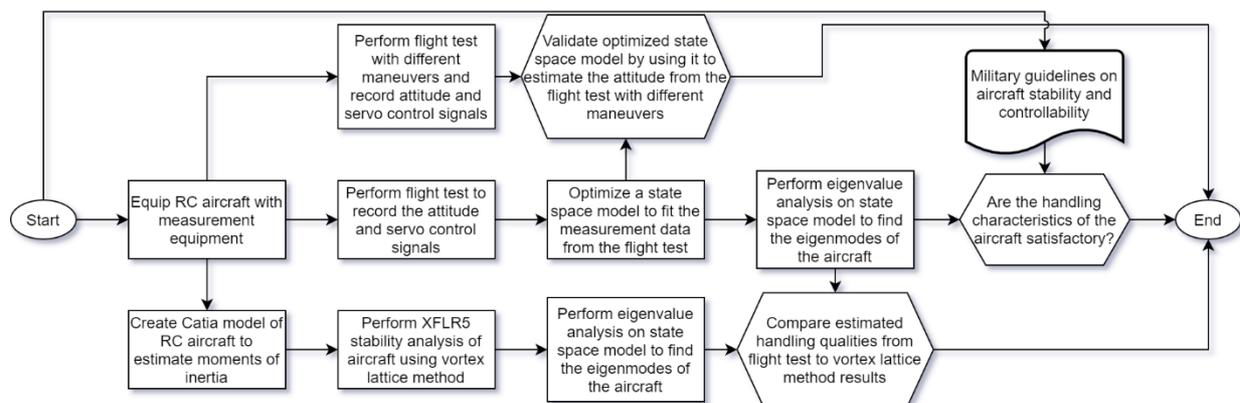


Figure 3.1 Top-level overview of method used in this thesis

3.1 Aircraft used for flight tests

The aircraft used for the flight tests is the Skysurfer X8. The following parts describe the aircraft used, an electrical system overview of the total system as well as a more detailed description of the subsystems that are of the most importance for the determination of the lateral-directional dynamics, which are the flight controller, servos, receiver, transmitter, telemetry and the airspeed sensor.

3.1.1 Skysurfer X8 overview in test flight condition

In this part first an overview of the Skysurfer X8 aircraft is provided, after this an overview is provided of all components that were used during the test flight in a table as well as the electrical system overview.

The Skysurfer X8 comes as a build kit consisting of the following:

- Two fuselage halves
- Right and left wing that include a carbon rod that is pre-mounted
- Vertical tail
- Horizontal tail
- Canopy insert including clear plastic cover
- Carbon reinforcement tube for connecting the wings
- Two long and two short pushrods, 4 servo horns including mounting hardware
- Wooden reinforcements for wing-mount, motor-mount and servo-mount
- Decals
- Installation manual

The model is made from Expanded PolyOlefin (EPO). The dimensions of the aircraft are shown in Table 3.1 below. Figure 3.2 and Figure 3.3 show the manufacturer recommended configuration of the Skysurfer X8 and the configuration that is used in the test flight campaign respectively.

Table 3.1 Dimensions of Skysurfer X8

Wingspan	141 cm
Wing chord at fuselage connection	20 cm
Aircraft Length	92 cm
Horizontal tail span	48 cm
Horizontal tail root chord	10.5 cm
Vertical tail height	17 cm
Vertical tail root chord	12 cm



Figure 3.2 Recommended configuration for Skysurfer X8¹

¹ <http://www.valuehobby.com/sky-surfer-1400-kit.html>



Figure 3.3 Aircraft used in flight test campaign [13]

Table 3.2 shows the components that are required for the Skysurfer X8. Table 3.3 shows the components that were used for the test flight. The major difference is the use of two motors and propellers on the configuration that is flown for the test flights. Not only will this allow for future tests with a one engine out condition, the use of a clockwise and counterclockwise rotating engine during the flight eliminates resultant torque from the engines, allowing to test the lateral-directional dynamics from a trimmed condition that is not influenced by the motor power.

Table 3.2 Required components for Skysurfer X8¹

Motor	E400 (2830) kv2200 motor with 5" or 6" propeller
Engine Speed Controller (ESC)	30A
Receiver and Transmitter	Minimum 4 channel with one Y harness
Servos	9g Servo times 4 with two servo extensions
Battery	11.1V 2200mAh 3S LiPo battery

Table 3.3 Components used for performing the test flight

Component	Variant
Onboard the aircraft	
Airframe	X-UAV Skysurfer X8 1400mm Wingspan FPV Aircraft RC Airplane KIT ²
Flight controller	Holybro Pixhawk 4 (Plastic case) including the UBLOX NEO-M8N GPS module ³
Air speed sensor	MS 4525DO pressure sensor and full metal pitot tube ⁴
2 Engines	2x Multistar Viking 1808-2600kv-CCW motor ⁵ with three blade X4040300-G propeller ⁶ (one clockwise and one counterclockwise spinning)
2 Engine Speed Controllers	DYS XS 30A (3-6S) BLHeli-S ESC ⁷
Battery	XTRON 3000mAh 11.1V 30C 3S1P LiPo battery ⁸
Battery Elimination Circuit	Castle Creations CC BEC High performance 2-6S 10A switching regulator ⁹
Power module	Mauch 100A 2-6S Hall effect sensor HS-100-LV ¹⁰
Receiver	Jeti Duplex REX12 EPC 2.4GHz ¹¹
Telemetry	RFD868+ including two quarter wave monopole 2.1dBi antennas ¹²
4 Control Servos	MG90S 13.4g coreless motor metal gear servo. ¹³
Used at the ground during flight operations	
Transmitter	Jeti DS-14 mode 2 transmitter JDEX-TSD14-M2 ¹⁴
Telemetry	RFD868+ including two half wave dipole 3dBi antennas and FTDI USB cable to connect to laptop running QGroundControl ¹²

Figure 3.4 shows the (electrical) system overview of the vehicle used in test conditions. One of the aspects shown is that on one of the engines at the connection between the motor and the ESC two of the three wires have been switched around. This change causes the engine to turn the other way around. All links with the Pixhawk are shown in purple. All control signals are shown in orange. In the case the link has an arrow, the arrow points in the direction where the power flows.

² https://www.banggood.com/X-UAV-Sky-Surfer-X8-1400mm-Wingspan-FPV-Aircraft-RC-Airplane-KIT-p-1064615.html?cur_warehouse=CZ

³ https://shop.holybro.com/pixhawk-4_p1089.html

⁴ <https://www.flyingtech.co.uk/electronics/digital-air-speed-sensor-pitot-tube-pixhawk>

⁵ https://hobbyking.com/nl_nl/multistar-viking-brushless-outrunner-drone-racing-motor-1808-2600kv-ccw.html

⁶ <http://ftec-shop.nl/shop/FMPro?-db=Ftec%20Producten.fp3&-lay=CGI&-format=HWdetail.htm&-RecID=37434&-token=&-find>

⁷ https://nl.banggood.com/DYS-XS-30A-3-6s-Lipo-BLheli-S-ESC-Support-Oneshot125-Oneshot42-Multishot-for-High-KV-Motor-for-RC-Drone-p-1060355.html?akmClientCountry=NL&cur_warehouse=CN

⁸ <https://www.stefansliposshop.de/en/batteries/sls-xtron/sls-xtron-30c/sls-xtron-3000mah-3s1p-11-1v-30c-60c::1032.html>

⁹ <https://www.trxxs-winkel.nl/castle-voeding-p-10778.html>

¹⁰ <https://www.onedrone.com/store/mauch-standard-line-100a-2-6s-sensor-hs-100-lv.html>

¹¹ <http://www.jetimodel.com/en/katalog/New-Products/@produkt/Duplex-REX12-EPC/>

¹² <http://store.rfdesign.com.au/rfd868-modem-bundle/>

¹³ https://www.banggood.com/nl/MG90S-Metal-Gear-RC-Micro-Servo-13_4g-for-ZOHD-Volantex-Airplane-RC-Helicopter-Car-Boat-Model-p-74870.html?utm_source=googleshopping&utm_medium=cpc_organic&gmcCountry=NL&utm_content=minha&utm_campaign=minha-nl-nl-

[pe¤cy=EUR&cur_warehouse=CN&createTmp=1&utm_source=googleshopping&utm_medium=cpc_bgcs&utm_content=sandra&utm_campaign=sandra-ssc-nl-css-all-0220&ad_id=499640888574&gclid=Cj0KCQjw2tCGBhCLARIsABJGmZ5SG5ImRwZaZri5Ttb2XoCI0B2r6hBGwcyzEq89gKmTkoyKByAj8YIaAruZEALw_wcB](https://www.banggood.com/nl/MG90S-Metal-Gear-RC-Micro-Servo-13_4g-for-ZOHD-Volantex-Airplane-RC-Helicopter-Car-Boat-Model-p-74870.html?utm_source=googleshopping&utm_medium=cpc_bgcs&utm_content=sandra&utm_campaign=sandra-ssc-nl-css-all-0220&ad_id=499640888574&gclid=Cj0KCQjw2tCGBhCLARIsABJGmZ5SG5ImRwZaZri5Ttb2XoCI0B2r6hBGwcyzEq89gKmTkoyKByAj8YIaAruZEALw_wcB)

¹⁴ <http://www.jetimodel.com/en/katalog/Discontinued-products/@produkt/DS-14/>

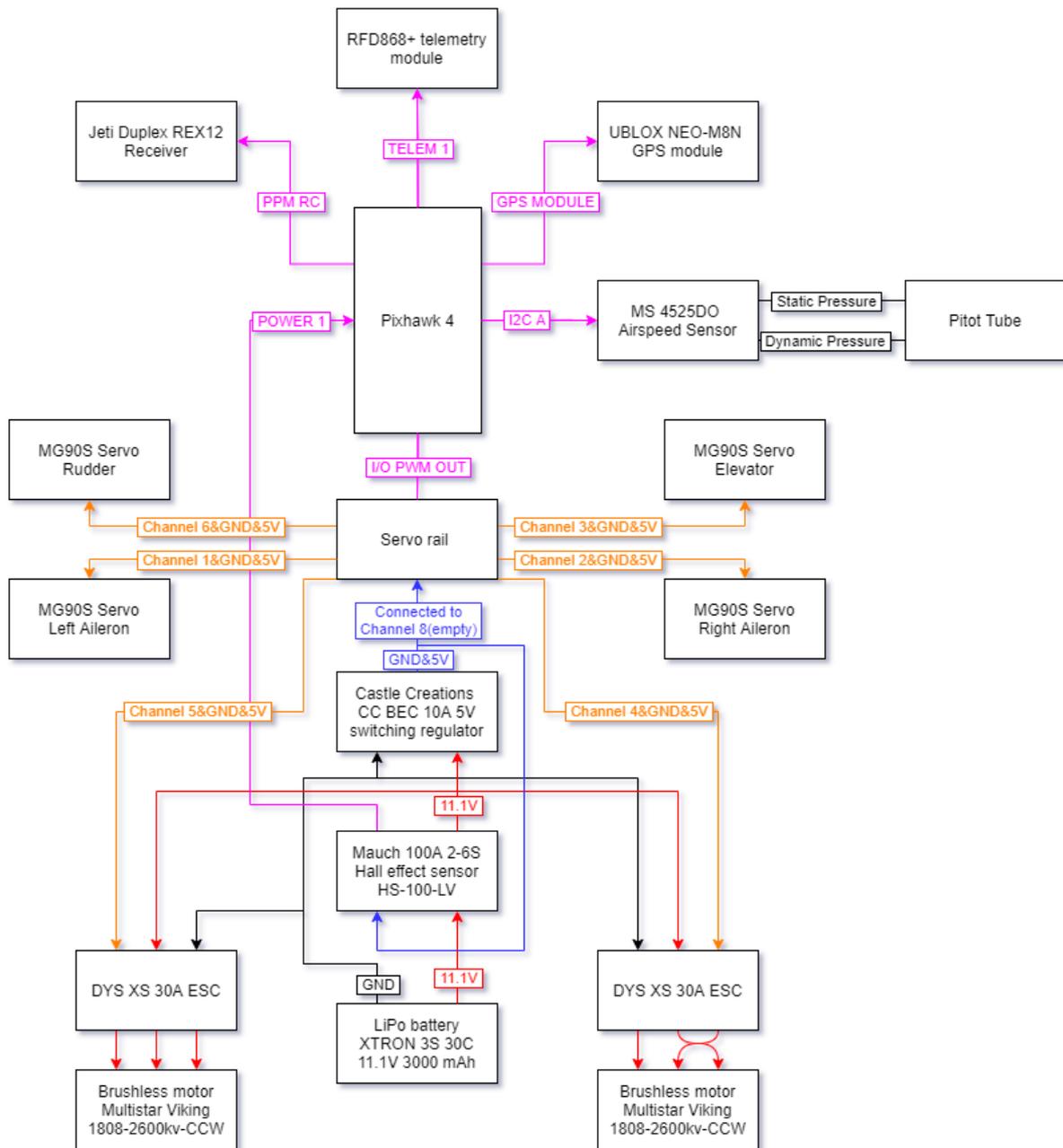


Figure 3.4 (Electrical) System diagram of the X-UAV Skysurfer X8 in flight condition

An overview of the dataflow to and from the aircraft is shown in Figure 3.5. During the flight the observer is assisting the pilot in obtaining the correct airspeed, altitude and position. In the case something goes wrong with the pilot the observer is able to change the aircraft flight mode to return-to-home, limiting risk to the aircraft and people on the ground.

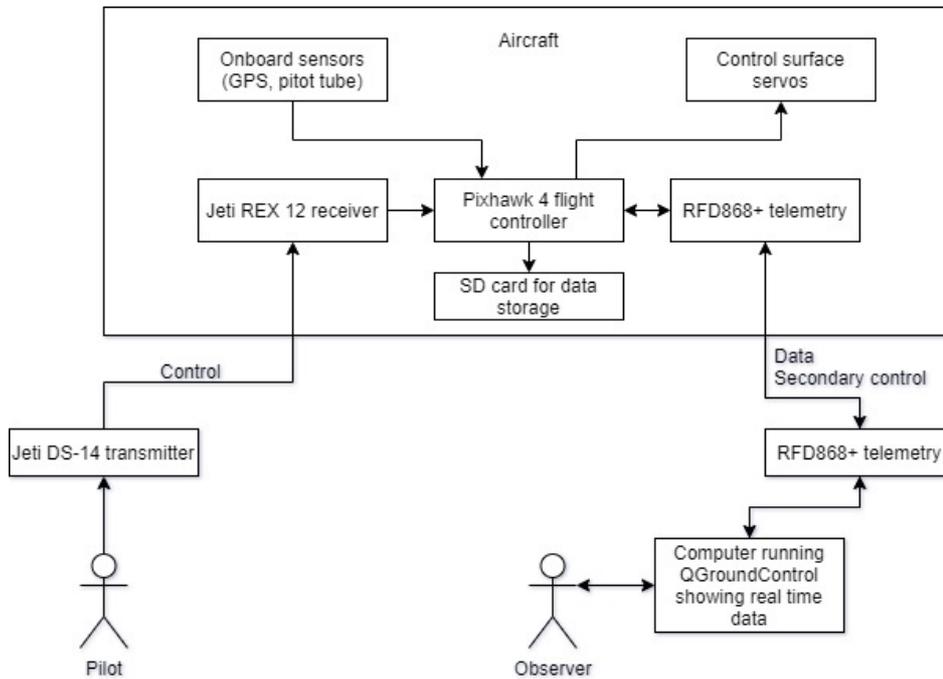


Figure 3.5 Data flow between different components on the aircraft and at the ground

3.1.2 Flight controller

The flight controller used in the flight test vehicle is the Holybro Pixhawk 4 (Plastic case) including the UBLOX NEO-M8N GPS module. First the sensor hardware will be discussed, later also the software that is run on the flight controller is shown.

The Pixhawk 4 and GPS module come with several onboard sensors, which are listed in Table 3.4.

Table 3.4 Sensors onboard the Pixhawk 4 flight controller¹⁵

Sensor Type	Sensor name
Onboard Pixhawk 4	
Accelerometer and gyroscope	ICM-20689 ¹⁶
Accelerometer and gyroscope	BMI055 ¹⁷
Magnetometer	IST8310 ¹⁸
Barometer	MS5611 ¹⁹
Onboard UBLOX NEO-M8N GPS module	
Magnetometer	IST8310 ¹⁸
GNSS receiver	UBLOX NEO-M8N ²⁰

The most important statistics of the different sensors are shown in Table 3.5, Table 3.6, Table 3.7 and Table 3.8 below.

¹⁵ https://github.com/PX4/px4_user_guide/raw/master/assets/flight_controller/pixhawk4/pixhawk4_technical_data_sheet.pdf

¹⁶ <https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/03/DS-000143-ICM-20689-TYP-v1.1.pdf>

¹⁷ https://datasheet.lcsc.com/szlcsc/Bosch-Sensortec-BMI055_C189620.pdf

¹⁸ https://groups.google.com/group/drones-discuss/attach/2028573861245/TE-DS-8310_8310%20datasheet.pdf?part=0.2

¹⁹ <https://www.digchip.com/datasheets/parts/datasheet/1352/MS5611-pdf.php>

²⁰ https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf

Table 3.5 Important characteristics of the magnetometer

Magnetometer type	IST8310 ¹⁸
Manufacturer	Isentek Motion Sensor Provider
Dynamic range (at 25°C) / resolution	X,Y axis +/-1600uT, Z axis +/-2500uT / 0.3uT
Zero Gauss RMS offset (at 25°C)	+/-0.3uT
Zero Gauss RMS offset temperature drift	0.024uT/°C
Sensitivity temperature drift	+/-0.016%/°C
Maximum frequency	200Hz

Given the earth magnetic field strength in the Netherlands of approximately 50uT [14], the magnetometer has an adequate resolution to measure the earth magnetic field as shown in Table 3.5. The higher dynamic range is of importance given the possible interference of powerlines near the sensor. The maximum frequency allows the flight controller to use the magnetometer information for relatively high rate attitude determination adjustments.

Table 3.6 Important characteristics of the barometer

Barometer type	MS5611 ¹⁹
Manufacturer	TE connectivity
Barometer sensor range / resolution	10mbar to 1200mbar / 0.065mbar (256 oversample ratio) to 0.012mbar (4096 oversample ratio)
Operational pressure range	450mbar to 1100mbar
Barometer accuracy at 25°C, 750mbar	+/-1.5mbar
Barometer total error 0-50°C 450-1100mbar	+/-2.0mbar
Barometer long term stability	+/-1mbar/year
Temperature sensor range / resolution	-40°C to 85°C / <0.01°C
Temperature sensor accuracy at 25°C	+/-0.8°C
Temperature sensor accuracy from -20°C to 85°C	+/-2.0°C
Response time	0.5ms (256 oversample ratio) to 8.22ms (4096 oversample ratio)

Given the maximum total error expected in the measured pressure by the barometer as shown in Table 3.6, the altitude can be reasonably accurately determined, to within around 14m at worst. The operational pressure range allows for flights up to around 6.3km above sea level, significantly higher than the 120 meters above ground level (at the beach) that regulations allow.

The two accelerometer/gyroscope packages used in the Pixhawk 4, as shown in Table 3.7, show some interesting aspects. First of all, the Bosh sensor has a lower resolution on the accelerometer compared to the TDK sensor. Although the cross-axis sensitivity of the Bosch sensor is lower, the zero-g output error is higher. The gyroscope resolution is the same, but the zero tolerance and cross axis sensitivity show that the Bosch sensor is more accurate. In all, it can be argued the Bosch is a better gyroscope and the TDK is a better accelerometer, however using both sensors, the Kalman filter is able to find an even better approximation for both. Also, the update frequency can be set very high, allowing for accurate attitude monitoring. The software uses both sensors at the same accelerometer and gyroscope range of +/-8g and +/-500°/s respectively.

Table 3.7 Important characteristics of the accelerometers and gyroscopes

Package name	ICM-20689 ¹⁶	BMI055 ¹⁷
Manufacturer	TDK InvenSense	Bosch
Accelerometer data	16 Bit	12 Bit
Accelerometer lowest measurement range / corresponding resolution	+/-2g / 0.061mg	+/-2g / 0.98mg
Accelerometer highest measurement range / corresponding resolution	+/-16g / 0.488mg	+/-16g / 7.81mg
Accelerometer zero-g output	+/-20mg	+/-70mg
Accelerometer cross axis sensitivity	+/-2%	+/-1%
Gyroscope data	16 Bit	16 Bit
Gyroscope lowest measurement range / corresponding resolution	+/-250°/s / 27.5°/h	+/-125°/s / 13.7°/h
Gyroscope highest measurement range / corresponding resolution	+/-2000°/s / 219.5°/h	+/-2000°/s / 219.5°/h
Gyroscope cross-axis sensitivity	+/-2%	+/-1%
Gyroscope zero tolerance	+/-5°/s	+/-1°/s
Temperature sensor (on die) range / corresponding resolution	-40°C to 85°C / 0.003°C	8 Bit centered around 23°C / 0.5°C
Update frequency	3.9Hz to 8000Hz	100Hz to 2000Hz

Table 3.8 Important characteristics of the GNSS receiver

Package name	NEO-M8N ²⁰
Manufacturer	U-blox
GNSS reception (3 simultaneously)	GPS, Galileo and BeiDou or GLONASS
Receiver type	72 channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C
Operational limitations	< 4g dynamics, max velocity 500m/s, max altitude 50km
Velocity accuracy / heading accuracy (50% at 30m/s)	0.05m/s / 0.3°
Horizontal position accuracy	2.5m (GPS) to 4 m (GLONASS)
Maximum navigation update rate	5Hz (GPS & GLONASS combined) or 10Hz (any GNSS separate)
Tracking & navigation sensitivity	-167dBm (GPS&GLONASS combined) to -159dBm (Galileo)

As can be seen in Table 3.8, the GNSS receiver used onboard the aircraft is able to receive any of the most common GNSS signals. During the test flight the amount of satellite signals that are received by the GNSS receiver is around 16, allowing for a high degree of precision both vertically as well as horizontally. The operational limitations are high enough not to cause any problems during the test flights.

The software that is run on the Pixhawk 4 is the PX4 beta version 1.9.0-beta1, which was released January 26th 2019²¹. An additional airframe and mixer file are created for the test aircraft which allows the Pixhawk 4 to be switched between other aircraft and be put back easily in the correct mode for the test aircraft. This airframe and mixer file setup is shown in Appendix B. The mixer file defines the channels as shown in Table 3.9.

Table 3.9 Channel allocation in Skysurfer mixer file

PWM channel number	Function
1	Aileron left
2	Aileron right
3	Elevator
4	Motor one
5	Motor two
6	Rudder

The PX4 software that is run on the Pixhawk 4 flight controller has the ability to function as many different types of controller, which it calls its flight modes. The most basic variant is the manual control mode. In this mode the Pixhawk 4 receives the control inputs from the pilot and transforms them directly into the signals it sends to the servos. The flight mode in which the flight tests are performed is the stabilized mode. In this mode the controller functions as a PID controller in which the roll and pitch inputs of the pilot are used as setpoint for the roll angle and pitch angle. The rudder control is used as the setpoint of the yaw rate. The throttle is still directly linked through to the engines. The PID settings used in the stabilized flight mode are shown in Table 3.10. The second mode that is selectable during the test flights is the altitude hold mode. In this mode the pitch controller is switched to an altitude controller in which the elevator input of the pilot gives the rate of ascend or descend. In the altitude hold mode the throttle input is used as an airspeed setpoint. The last flight control mode that can be selected during the flight is the return-to-home flight mode. This is also the flight mode that is entered in the case that the aircraft flies outside the pre-set geo fence or the link to the controller is lost. This mode causes the aircraft to fly to a pre-set altitude and airspeed towards the home position at which it will start to circle. In this mode the aircraft flies itself and the stick inputs have no effect on the aircraft. Switching to a different flight mode allows the pilot to regain control.

Although the test flight is performed in the stabilized mode, the controller itself does not have to be taken into account for the analysis of the system. This is because of the fact that not the input towards to flight controller, but the output of the flight controller towards the servos is used in the analysis. This analysis would be equivalent to flying in the manual control mode and providing as inputs the outputs that are currently given by the PID controller.

Table 3.10 PID settings of the PX4 autopilot in stabilized flight mode

	Roll angle	Pitch angle	Yaw rate
Feed forward gain	0.5	0.5	0.3
Differential gain	0.05	0.08	0.05
Integrator gain	0.01	0.02	0.01
Integrator gain limit	0.2	0.4	0.2

²¹ <https://github.com/PX4/PX4-Autopilot/releases/tag/v1.9.0-beta1>

The PX4 software uses quaternions for the attitude of the aircraft to avoid computational problems that arise from working with Euler angles. One of the main problem is that a small change in angle might lead to a larger jump in the actual number like from 180 degrees to 0 degrees in the case of the yaw and roll angle in the case of a looping. Also it prevents gimbal lock. In the case the pitch angle goes to 90 degrees, the roll and yaw both cause the same rotation, making it impossible for the computer to figure them out. Furthermore in quaternions all positions are uniquely defined, whereas with Euler angles, this is not the case. Lastly quaternions offer better computational performance compared to Euler angles. All problems are solved using the quaternions where small angle changes also lead to small differences in the quaternions. To return from the recorded quaternions to the attitude of the aircraft the Euler angle sequence (1,2,3) is used [15]. Equation (23) below shows how the roll angle ϕ is calculated from the quaternions. q_0 , q_1 , q_2 and q_3 are the quaternions as recorded in the flight data under data `_vehicle_attitude_0`.

$$\phi = \tan^{-1} \left(\frac{2 \cdot (q_3 \cdot q_2 + q_0 \cdot q_1)}{q_3^2 - q_2^2 - q_1^2 + q_0^2} \right) = \tan^{-1} \left(\frac{2 \cdot (q_3 \cdot q_2 + q_0 \cdot q_1)}{1 - 2 \cdot (q_2^2 + q_1^2)} \right) \quad (23)$$

Equation (24) below shows how the pitch angle θ is calculated from quaternions.

$$\theta = \sin^{-1}(2 \cdot (q_2 \cdot q_0 - q_3 \cdot q_1)) \quad (24)$$

Equation (25) below shows how the heading ψ is calculated from quaternions.

$$\psi = \tan^{-1} \left(\frac{2 \cdot (q_3 \cdot q_0 + q_2 \cdot q_1)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right) = \tan^{-1} \left(\frac{2 \cdot (q_3 \cdot q_0 + q_2 \cdot q_1)}{2 \cdot (q_0^2 + q_1^2) - 1} \right) \quad (25)$$

The PX4 software does not provide the sideslip angle β as an output. In order to obtain the sideslip angle, the angle between the velocity vector relative to the freestream velocity vector and the body heading angle ψ is determined using Equation (26) below. In this V_{g_e} is the GPS groundspeed in the east direction, V_{we_e} the estimated wind speed in the east direction, V_{g_n} is the GPS groundspeed in the north direction, V_{we_n} the estimated wind speed in the north direction.

$$\beta = \left(\psi - \text{atan} \left(\frac{V_{g_e} - V_{we_e}}{V_{g_n} - V_{we_n}} \right) \right) \quad (26)$$

3.1.3 Control servos

The control servos that are used are the MG90S²². The MG90S is an analog metal gear servo. The use of metal gears allows for better durability. The operating voltage for the servo is between 4.8V and 6.0V. The use of a higher voltage increases the stall torque and operational speed of the servo, but reduces lifetime. In the test aircraft the voltage used is 5.0V. At this voltage the stall torque of the servo is approximately 1.8kgf·cm. The operating speed at 5.0V is approximately 600°/s without a load applied. This last number is validated using a servo tester. The operating speed with resistance is not tested and not known. The total operating range is 180° which corresponds to a PWM signal from 1000ns to 2000ns. The PWM period is 50Hz and can not be changed, which means that the servo can receive instructions for the required position at 50Hz. The weight of a single servo including its cable is 13.4g.

²² <https://datasheetspdf.com/pdf-file/1106582/ETC/MG90S/1>

3.1.4 Receiver, transmitter and telemetry system

Onboard the aircraft a Jeti Duplex REX 12¹¹ receiver is placed which is linked to a Jeti DS-14 controller. The Jeti Duplex REX 12 is powered by and sends signals to the Pixhawk 4 flight computer via its E1 port. This port is setup from the controller as a PPM positive port which allows for the transmission of all control signals over a single line. The controller provides the output signals as shown in Table 3.11. The controller is setup in a mode 2 configuration, which has throttle and rudder on the left and aileron and elevator on the right stick. In the setup of the controller the transmission frequency is set at the standard 50Hz. The maximum frequency that is supported by the Jeti DS-14 is 100Hz²³.

Table 3.11 Controller channel setup

Channel number	Function
1	Throttle
2	Rudder
3	Elevator
4	Aileron
5	Arm switch
6	Flight mode selector switch

To provide the frequency sweep as well as the doublet inputs for the flight tests a LUA script is setup to be run on the controller. This LUA script is provided by Sobron [16]. The LUA script provided allows to choose different maneuvers like a doublet (first up for x seconds than down the same amount for again x seconds), 3-2-1-1 (up for $3x$ seconds, then down the same amount for $2x$ seconds, then up the same amount for x seconds and lastly down the same amount again for x seconds), exponential decay (following $a \cdot e^{-xt}$ profile) and frequency sweep (signal is $a \cdot \sin(b \cdot dt + c)$ where c starts at 0 and at every timestep in increased by $b \cdot dt$. In this the b value is controlled by a rotary dial on the controller allowing to sweep through the frequencies). For all maneuvers, the setup requires to set the amplitude of the maneuver as well as the time of the steps in the maneuver, or in the case of the frequency sweep the minimum and maximum frequency. The signals created by the LUA script are mixed using the free mixes option on the Jeti DS-14 with the signals for the rudder and aileron to perform the flight tests required for the lateral and directional dynamics.

To allow for monitoring the flight remotely a RFD868+ telemetry module¹² is used onboard the aircraft as well as connected to a laptop. A special cable is made to connect the RFD868+ to the TELEM 1 port of the Pixhawk 4 flight controller. To create this cable the pinouts were matched up from the documentation of the RFD868+²⁴ and Pixhawk 4²⁵. The RFD868+ is setup using the RFD900 tools application²⁶ on the computer to follow Dutch telecommunication laws. The “Regeling gebruik van frequentieruimte zonder vergunning en zonder meldingsplicht 2015” [17] allows the use of the 865.000MHz to 868.400MHz frequency range for telemetry purposes at a duty cycle of less than 1% and a 25mW maximum effective radiated power.

To monitor the flight as well as do some setups for the flight QGroundControl²⁷ version 3.5.5 is used. Before each flight QGroundControl is used to perform a number of sensor calibrations which are shown in Figure 3.6. In this setup section of QGroundControl it can also be seen if the radio and flight modes have been setup or not, indicated by a figure on

²³ https://doe.hu/sites/default/files/hasznalati_utmutatok/file/dc-ds-en-2014-fw3.00.pdf

²⁴ <http://files.rfdesign.com.au/Files/documents/RFD900x%20DataSheet.pdf>

²⁵ <http://www.holybro.com/manual/Pixhawk4-Pinouts.pdf>

²⁶ <https://files.rfdesign.com.au/tools/>

²⁷ <http://qgroundcontrol.com/>

the left of the screen. The Pixhawk will only allow arming if all setup is complete and none of the figures are red. For the test flights in the flight mode setup one of the two point switches from the controller is setup as the arm switch and a three point switch of the controller is setup as the flight mode selector. During the test flights the three flight modes that are selectable through the controller are set as stabilized flight mode, altitude hold flight mode and return-to-home flight mode. In the case of loss of signal of more than 5 seconds the aircraft is switched automatically to the return-to-home function. All low battery actions are set to provide a warning to the pilot.

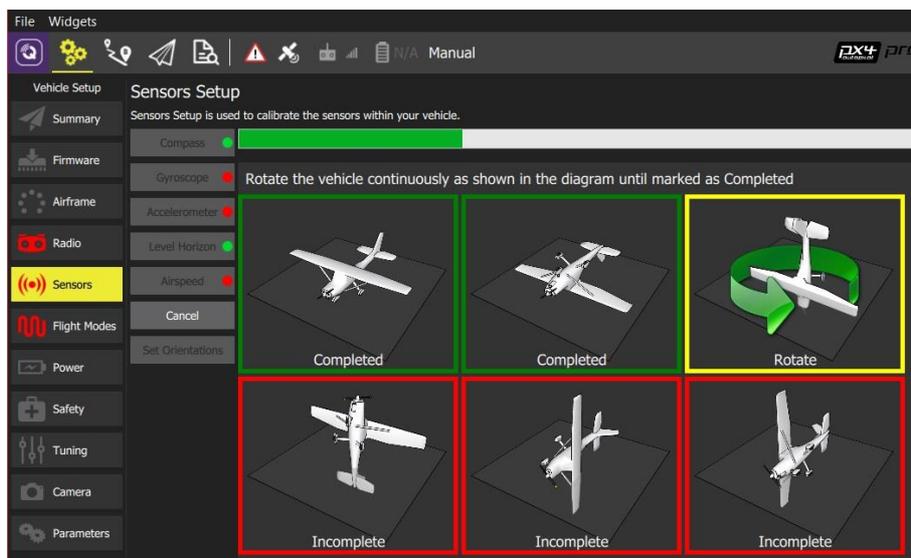


Figure 3.6 Sensor calibration screen within QGroundControl showing compass calibration in progress²⁸

During the flight QGroundControl is used to monitor the flight using the screen shown in Figure 3.7. In the top row some important information is shown for the flight like remaining battery percentage, telemetry signal strength, controller signal strength, the number of GNSS satellites are linked to the Pixhawk, the current flight mode and if the aircraft is Armed or Disarmed. Pressing on the different icons shows more information, but this is not used during the flight. On the right side there are two round diagrams with of the left an artificial horizon and on the right a compass. Below this some of the data from the aircraft is shown. The setup used during the test flights showed in large the altitude of the aircraft, the ground speed, the total flight time and the airspeed. Below this in a smaller format the distance to home, climb rate, distance to ground control station, heading to home, and pitch angle are displayed. The data is used during the test flight to make sure the aircraft operates within regulations and to setup the correct conditions to perform the different maneuvers. Before starting the maneuvers, the aircraft is trimmed and the throttle is adjusted for the correct airspeed and zero climb rate at the starting altitude of the maneuvers.

²⁸ https://docs.qgroundcontrol.com/master/en/SetupView/sensors_px4.html

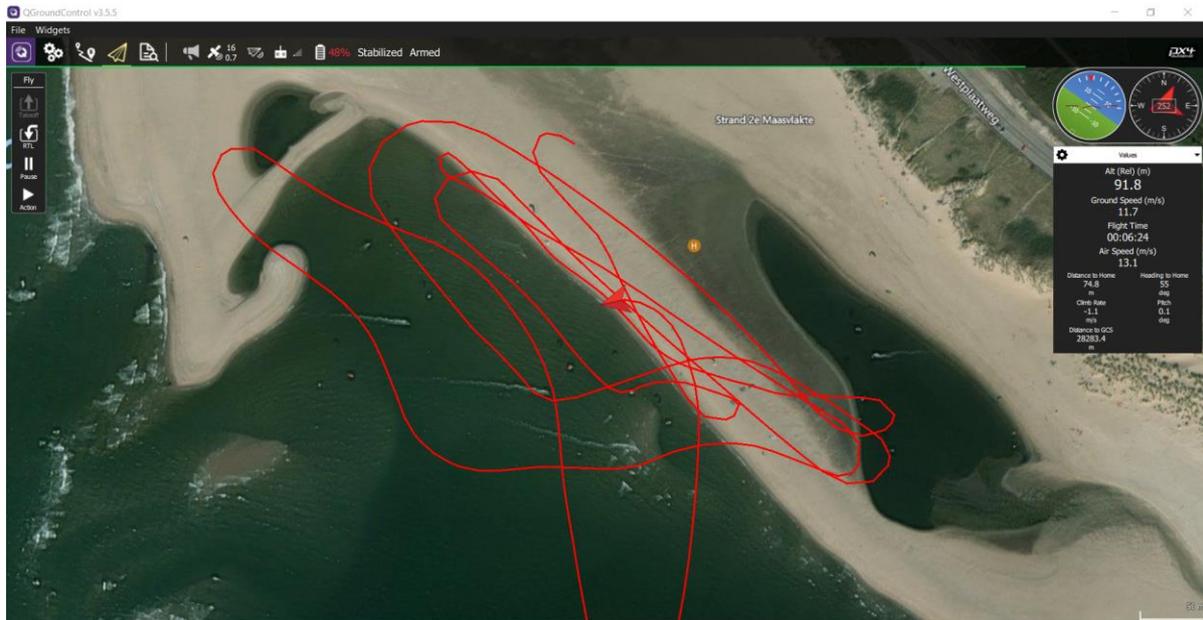


Figure 3.7 Screenshot of QGroundControl during one of the test flights used to test the aircraft functionality

3.1.5 Airspeed measurement system

The airspeed sensor used is the MS 4525DO by TE Connectivity²⁹. It is combined with an all-metal pitot tube which is mounted at the nose of the aircraft using a 3d-printed mount as shown in Figure 3.8. The pitot mount was designed in a way to make the pitot point in the approximate direction of the free-stream in an 18 m/s cruise condition. This was achieved by analyzing a picture of the aircraft flying at eye level at 18 m/s to estimate the approximate angle required to point in the direction of the free-stream. The Sensor that is used to analyze the difference between the total and static pressure of the pitot tube has a pressure range of 1psi, which allows for measuring speeds of up to around 106m/s at standard sea level conditions. Some of the important characteristics of the sensor are shown in Table 3.12. Whilst testing the sensor indoors without wind, the indicated airspeed fluctuates to a maximum of around 3m/s which indicates a fluctuation of around 6Pa. The fluctuation in indicated airspeed whilst in flight is significantly lower given the higher differential pressures. There are no data available for the accuracy of the pitot tube. The effect of the body and propellers is also not investigated. The location of the pitot tube is chosen to limit the effects of the body and propellers and limiting potential damage during landings at the beach. The propellers are mounted at over 2 propeller diameters from the centerline of the aircraft, at which the pitot tube is mounted, making their effect on the flow field minimal [18].

²⁹ https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FMS4525DO%7FB2%7Fpdf%7FEnglish%7FENG_DS_MS4525DO_B2.pdf%7FCAT-BLPS0002

Table 3.12 Characteristics of the MS 4525DO pressure sensor²⁹

Package name	MS 4525DO-DS 5 001 D
Differential pressure range	+/- 1psi \approx +/- 6900Pa
Maximum airspeed measurement, air density 1.225kg/m ³	106 m/s
Compensated temperature range	-10°C to 85°C
Maximum error at 25°C	+/-34Pa
Maximum total error	+/-138Pa
Maximum total error at 18m/s, air density 1.225kg/m ³	-8.1m/s to 5.4m/s
Differential pressure data	14 bit
Differential pressure resolution	0.84Pa
Airspeed resolution at 18m/s, air density 1.225kg/m ³	+/-0.038m/s
Temperature data	11 bit
Temperature range	-50°C to 150°C
Interface type	I ² C
Update time	0.5ms
Time from start up to ready data	8.4ms
Pixhawk update frequency (defined by PX4 software ³⁰)	100Hz



Figure 3.8 Pitot tube mounting location and tube routing. The approximate free stream velocity direction during 18 m/s cruise is shown in red

³⁰ https://github.com/PX4/PX4-Autopilot/blob/master/src/drivers/differential_pressure/ms4525/ms4525_airspeed.cpp

3.2 State space system for aircraft lateral-directional dynamics

For the analysis of the lateral-directional dynamics of the aircraft first a model has to be created that incorporates the lateral-directional of the aircraft. Once the model has been found, the model can be updated to find the optimal model that represents the dynamics of the aircraft during the test flight.

Before creating the model, the different angles and directions are defined in Figure 3.9 and Figure 3.10.

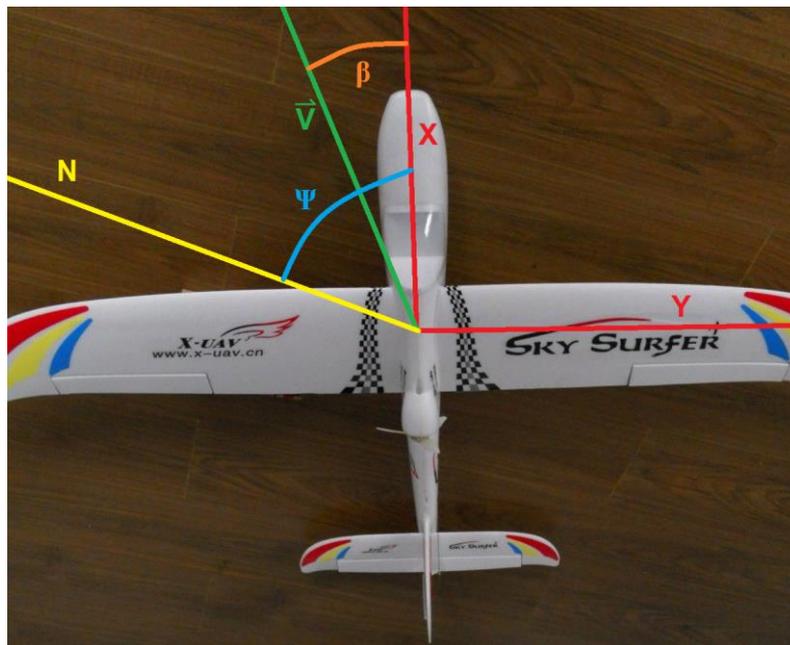


Figure 3.9 Top view of Skysurfer X8³¹ showing body axis X and Y, earth north N, free stream velocity vector \vec{V} , positive yaw angle Ψ and positive sideslip angle β

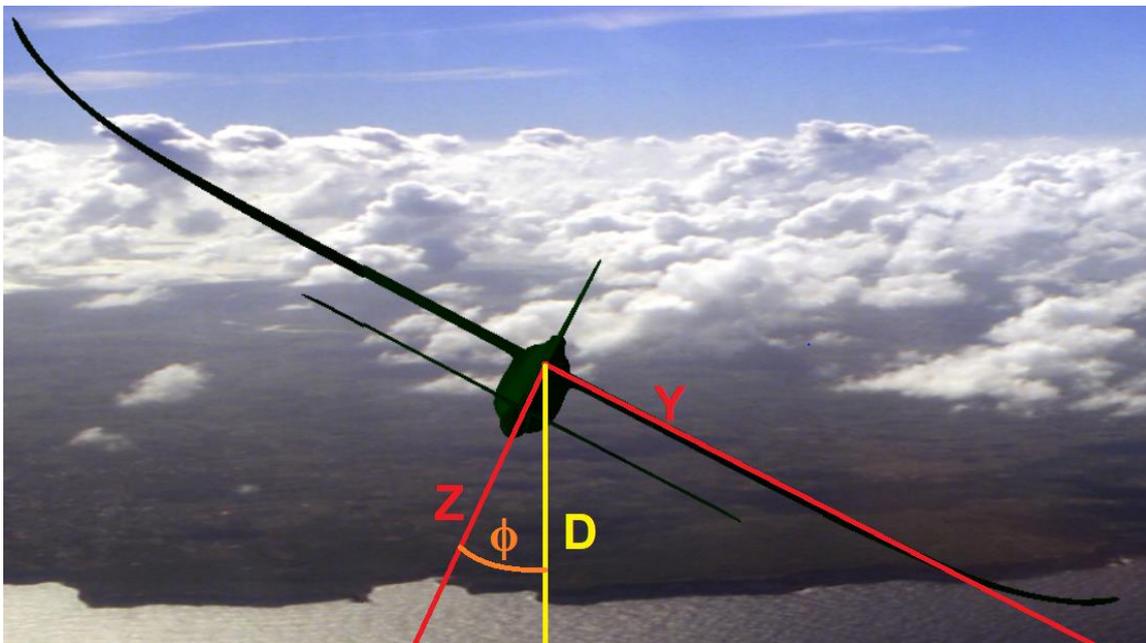


Figure 3.10 Back view of Skysurfer X8 showing body axis Y and Z, earth down axis D and positive roll angle ϕ

³¹ <https://www.ebay.com/p/1241206879>

The general equation for linear acceleration is provided in Equation (27) below. This shows a three dimensional representation of force F is equal to mass m times acceleration a .

$$m \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \sum F_x \\ \sum F_y \\ \sum F_z \end{bmatrix} \quad (27)$$

The equation for angular acceleration $\dot{p}, \dot{q}, \dot{r}$ in three dimensions as a function of the sum of the moments $\sum M$ in the three directions is provided in Equation (28) below where p is roll rate, q is pitch rate and r is the yaw rate.

$$\begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \sum M_x \\ \sum M_y \\ \sum M_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (28)$$

Given the fact only the lateral directional motion is of interest, the Y axis rotation, or pitch, is assumed to remain constant and removed from the equation which results in Equation (29) below.

$$\begin{bmatrix} I_{XX} & -I_{XZ} \\ -I_{XZ} & I_{ZZ} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \sum M_x \\ \sum M_z \end{bmatrix} - \begin{bmatrix} p \\ r \end{bmatrix} \times \begin{bmatrix} I_{XX} & -I_{XZ} \\ -I_{XZ} & I_{ZZ} \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} \quad (29)$$

Also, the accelerations in X and Z are not of interest for the lateral directional dynamics and therefor removed. The remaining acceleration in Y can be rewritten to include the effect of the change in the velocity vector of the aircraft as shown in Equation (30) below where V is the velocity of the aircraft and \dot{v} the change in sideways velocity.

$$\sum F_y = m(\dot{v} + rV) \quad (30)$$

Equation (31) below shows how the sideways velocity v is related to the sideslip angle. The assumption that the angle is small is made to find make the equation linear. The same equation holds for the change is sideslip related to the change is sideways velocity.

$$\beta = \sin^{-1} \left(\frac{v}{V} \right) \approx \frac{v}{V} \quad (31)$$

Following the expansion of the different forces and moments according to Mulder [19], Equation (32) below shows the equation of force in the Y direction. In the following four equations the fact the flight path is horizontal is used to simplify the equations. In this W is the weight of the aircraft, δ_a the aileron deflection and δ_r the rudder deflection. Y_x stands for the change in the force in Y direction due to a unit of x , which can also be written as $\frac{dF_y}{dx} = Y_x$. In this x can be any variable.

$$W\phi + Y_\beta \beta + Y_{\dot{\beta}} \dot{\beta} + Y_p p + Y_r r + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r = m(\dot{\beta}V + rV) \quad (32)$$

Equation (33) below shows the equation of the sum of the moments around the X axis. Similar to the Y_x before, here L_x is used to show the change in moment around the X axis due to a unit of x .

$$L_\beta \beta + L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r = I_{XX} \dot{p} - I_{XZ} \dot{r} \quad (33)$$

Equation (34) below shows the sum of the moments around the Z axis. Similar to the L_x before, here N_x is used to show the change in moment around the Z axis due to a unit of x .

$$N_\beta \beta + N_{\dot{\beta}} \dot{\beta} + N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r = I_{ZZ} \dot{r} - I_{XZ} \dot{p} \quad (34)$$

To this one kinematic equation for the change in roll angle $\dot{\phi}$, Equation(35) below, is added to create a system of four first order differential equations that can be used to find the lateral directional dynamics of the aircraft. A fifth equation that shows that the change in yaw angle is equal to the yaw rate could be added, but this equation would not influence the other equations at all given that the actual yaw angle does not influence any aerodynamic forces.

$$\dot{\phi} = p \quad (35)$$

Given the four equations above a matrix model is created as shown in Equation (36) below.

$$\begin{aligned} & \begin{bmatrix} mV - Y_{\dot{\beta}} & 0 & 0 & 0 \\ 0 & 0 & I_{XX} & -I_{XZ} \\ -N_{\dot{\beta}} & 0 & -I_{XZ} & I_{ZZ} \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} \\ & = \begin{bmatrix} Y_\beta & W & Y_p & Y_r & -mV \\ L_\beta & 0 & L_p & L_r \\ N_\beta & 0 & N_p & N_r \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \end{aligned} \quad (36)$$

Most commonly the equations are nondimensionalized which results in Equation (37) as shown below [19]. This strategy is not taken in this thesis due to the lack of available aerodynamic characteristics of the aircraft. No advantage would be obtained from going this route. In this b is the wingspan of the aircraft, S the wing planform area, C_L the aircraft lift coefficient, ρ the air density, $\frac{d}{dt}$ the time derivative and all C are nondimensionalized constants. For the force and moment equation the nondimensionalization is done by dividing any length unit by the wingspan b , any velocity component by the speed V and any mass component by $\rho S b$.

$$\begin{bmatrix}
C_{Y\beta} + \left(C_{Y\dot{\beta}} - 2\frac{m}{\rho S b}\right)\frac{b}{V}\frac{d}{dt} & C_L & C_{Yp} & C_{Yr} - 4\frac{m}{\rho S b} \\
0 & -\frac{1}{2}\frac{b}{V}\frac{d}{dt} & 1 & 0 \\
C_{L\beta} & 0 & C_{Lp} - \frac{4I_{XX}}{\rho S b^3}\frac{b}{V}\frac{d}{dt} & C_{Lr} + \frac{4I_{XZ}}{\rho S b^3}\frac{b}{V}\frac{d}{dt} \\
C_{N\beta} + C_{N\dot{\beta}}\frac{b}{V}\frac{d}{dt} & 0 & C_{Np} + \frac{4I_{XZ}}{\rho S b^3}\frac{b}{V}\frac{d}{dt} & C_{Nr} - \frac{4I_{ZZ}}{\rho S b^3}\frac{b}{V}\frac{d}{dt}
\end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} = \begin{bmatrix} -C_{Y\delta_a} & -C_{Y\delta_r} \\ 0 & 0 \\ -C_{L\delta_a} & -C_{L\delta_r} \\ -C_{N\delta_a} & -C_{N\delta_r} \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (37)$$

To create the state space model which can be used to find the eigenmodes of the aircraft, the differential part in front of the equal sign should have an identity matrix. This can be obtained by pre multiplying all elements with the inverse of the differential matrix as shown in Equation (38) below.

$$\begin{aligned}
& \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} \\
& = \begin{bmatrix} mV - Y_{\dot{\beta}} & 0 & 0 & 0 \\ 0 & 0 & I_{XX} & -I_{XZ} \\ -N_{\dot{\beta}} & 0 & -I_{XZ} & I_{ZZ} \\ 0 & 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Y_{\beta} & W & Y_p & Y_r - mV \\ L_{\beta} & 0 & L_p & L_r \\ N_{\beta} & 0 & N_p & N_r \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} \\
& + \begin{bmatrix} mV - Y_{\dot{\beta}} & 0 & 0 & 0 \\ 0 & 0 & I_{XX} & -I_{XZ} \\ -N_{\dot{\beta}} & 0 & -I_{XZ} & I_{ZZ} \\ 0 & 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (38)
\end{aligned}$$

The resulting state space model, shown in Equation (39) below, is simplified and shows any entry which is aircraft or flight dependent as an X . These are the values that are optimized to obtain the model of the lateral-directional dynamics of the aircraft.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} X & X & X & X \\ 0 & 0 & 1 & 0 \\ X & X & X & X \\ X & X & X & X \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} X & X \\ X & X \\ X & X \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (39)$$

3.3 Flight test plans

In order to perform the experiments, first a test plan has to be created which includes checklists to increase the safety of the different flight tests. First the checklists will be shown. After this the number of flight tests are shown.

At the beginning of each flight test campaign a number of checks are made that don't have to be repeated for each successive flight as long as there are no changes made to the environment or the aircraft. Those checks are shown in Table 3.13.

Table 3.13 Start of flight test campaign checklist

Range check: Take the aircraft 1000m from the ground station and controller and check if the reception remains. Rotate the aircraft around all axis at the 1000m distance and make sure that the reception does not drop out in any aircraft orientation.
Battery voltage correspondence check: Check if the battery voltage measured from the battery corresponds to the battery voltage displayed in the top row of the QGroundControl flight screen as shown in Figure 3.7.
Sensor calibrations: Run through all sensors shown in the QGroundControl sensor tab as shown in Figure 3.6 and perform the calibration for each sensor even though the sensor might already show a green light.
Controller setup: Calibrate controller withing QGroundControl, ensure correct channel for arming has been selected, ensure correct channel for flight mode switch is selected, ensure the correct flight modes are selected for the flight mode switch (stabilized, altitude hold, return-to-home)
Geofence setup: Check within QGroundControl if the geofence has been setup correctly, meaning the actual shape of the geofence as well as that the aircraft will switch to return-to-home mode in case of a breach of the geofence.
Engine components not overheating: Arm aircraft and perform 30 seconds full throttle test whilst holding aircraft behind at the fuselage part connecting the wings and tailplane. After 30 seconds disarm aircraft and feel if the battery, wires, ESC and motor are not too hot, meaning you can touch the component without burning your fingers, or around 45 Celsius.

Before each flight the aircraft has to be inspected according to Table 3.14. This is to ensure that the aircraft is in the correct configuration for the test flight and risks are limited.

Table 3.14 Before flight checklist

Aircraft structure	No visible cracks or missing parts
Wing mounting	Bolts secure, wings not pulling out
Propeller	Securely mounted
Battery voltage	above 12.0V
Battery	Strap secured, no movement, lead connected
Canopy	Closed and secured
Center of gravity position	Within accepted bounds
Sensor calibration screen (Figure 3.6)	All sensors green (if not recalibrate all sensors)
Controller arm switch	Disarmed
Controller flight mode switch	Stabilized flight mode selected
Safety switch	Pressed, safety off
Control surfaces	All free and rotating correctly according to control inputs
Stabilized mode	Control surfaces deflect to return aircraft to wings level and zero pitch angle when aircraft is pitched and rolled
Frequency sweep setup (if applicable)	Frequency sweep performs correctly on selected control surface
Environment	No external people under planned flight path, no people within 200 meters in the direction of takeoff, landing area free of people
Before flight checklist	Complete

Before each flight the pilot does the I'M SAFE checklist for themselves and anyone who is directly involved in the flight procedures as shown in Table 3.15.

Table 3.15 I'M SAFE checklist

I	Illness	Are you feeling any symptoms of any illness that can affect your performance during the flight operations
M	Medication	Are you using any medication that can affect your performance negatively during the flight operations
S	Stress	Are you experiencing any form of stress (psychological, financial, health, family) that can affect your performance negatively during the flight operations
A	Alcohol	Have you consumed any alcohol during the last 10 hours or have you consumed a significant amount of alcohol the last 24 hours that still affects your performance during the flight operations
F	Fatigue	Are you feeling adequately rested for the flight operations
E	Eating	Are you adequately nourished such that it does not affect your performance during the flight operations

The first flight test performed, as shown in Table 3.16, is to ensure the correct setup of the aircraft. In the case the responsiveness during the maneuvers or in the return-to-home mode are deemed unsatisfactory, the PID settings of the controller are changed. In the case that the response of the aircraft is too low to perform the flight safely, the observer is asked to switch to the manual flight mode such that the pilot can return and land the aircraft without interference of the controller. During the flight the observer will announce if the battery level falls below 30% to give the pilot enough time to perform a landing.

Table 3.16 Initial setup flight test briefing

Checklists to complete	Before flight test campaign	Before flight	I'M SAFE
<ol style="list-style-type: none"> 1) Observer holds fuselage below the main wings of the aircraft one handed 2) Arm vehicle 3) Full throttle 4) Observer throws aircraft straight to the front 5) Climb out straight to around 80m altitude 6) Whilst retaining at least 50m altitude perform a number of maneuvers to check the responsiveness and maximum pitch and roll angle of the aircraft 7) Whilst flying away from the home position at around 80m switch to return-to-home flight mode 8) Once aircraft returns home and has started circling switch back to stabilized flight mode 9) Fly at an altitude of 80 meters towards the edge of the geofence and cross it 10) Once aircraft returns home and has started circling switch flight mode to altitude hold and back to stabilized flight mode 11) Fly a missed approach to the landing area 12) Land the aircraft at the landing area making sure to reduce throttle to 0% before landing 13) Disarm vehicle 			

Once the aircraft responsiveness in the stabilized flight mode is deemed adequate, the flight tests to obtain the experimental data for the lateral-directional dynamics of the aircraft are performed. Frequency sweeps are chosen on the different axes to allow for good excitations and therefor good reconstruction of the state space model. As a validation for the model, a doublet input is given on the different axes to ensure the model captures the correct aircraft dynamics. The flight test for the roll and yaw axes is shown in Table 3.17. In the case the battery voltage drops too low, the test is aborted and the remaining maneuvers are performed in a separate test trying to ensure the same center of gravity position is used. Fortunately this did not have to be done, ensuring the environmental conditions as well as the aircraft center of gravity and weight remain the same for all tests.

Table 3.17 Experimental flight test briefing for the roll and yaw axes

Checklists to complete	Before flight test campaign (if required)	Before flight	I'M SAFE
		<ol style="list-style-type: none"> 1) Observer holds fuselage below the main wings of the aircraft one handed 2) Arm vehicle 3) Full throttle 4) Observer throws aircraft straight to the front 5) Climb out straight to around 80m altitude 6) Turn to fly with the wind and fly to around 400 meters from the home position 7) Turn 180° 8) Whilst flying to a maximum of around 400 meters away from the home position, adjust throttle setting and pitch trim settings to fly straight and level at 18m/s at an 80m altitude 9) Turn 180° 10) Fly with the wind and fly to around 400 meters from the home position 11) Turn 180° making sure to end up at around 80m altitude 12) Wait 5 seconds without any control inputs 13) Perform frequency sweep on the roll axis making sure first two oscillations at 0.4Hz are performed before slowly increasing the frequency to 5Hz at an 20% maximum control signal 14) Wait 5 seconds without any control inputs making sure not to fly more that 450m away from the home position 15) Turn 180° 16) Fly with the wind and fly to around 400 meters from the home position 17) Turn 180° making sure to end up at around 80m altitude 18) Wait 5 seconds without any control inputs 19) Perform a number of doublets with a 0.5s roll right and 0.5s roll left signal at 20% maximum control signal 20) Wait 5 seconds without any control inputs making sure not to fly more that 450m away from the home position 21) Turn 180° 22) Fly with the wind and fly to around 400 meters from the home position 23) Turn 180° making sure to end up at around 80m altitude 24) Wait 5 seconds without any control inputs 25) Perform frequency sweep on the yaw axis making sure first two oscillations at 0.4Hz are performed before slowly increasing the frequency to 5Hz at an 80% maximum control signal 26) Wait 5 seconds without any control inputs making sure not to fly more that 450m away from the home position 27) Turn 180° 28) Fly with the wind and fly to around 400 meters from the home position 29) Turn 180° making sure to end up at around 80m altitude 30) Wait 5 seconds without any control inputs 31) Perform a number of doublets with a 0.5s yaw right and 0.5s yaw left signal at 80% maximum control signal 32) Wait 5 seconds without any control inputs making sure not to fly more that 450m away from the home position 33) Land the aircraft at the landing area making sure to reduce throttle to 0% before landing 34) Disarm vehicle 	

3.4 Method used to determine the lateral-directional stability characteristics of the aircraft from flight tests

A top-level overview of the method used to determine the lateral-directional stability characteristics of the aircraft from flight tests is provided in Figure 3.11. The letter variables from Figure 3.11 are shown in Table 3.18.

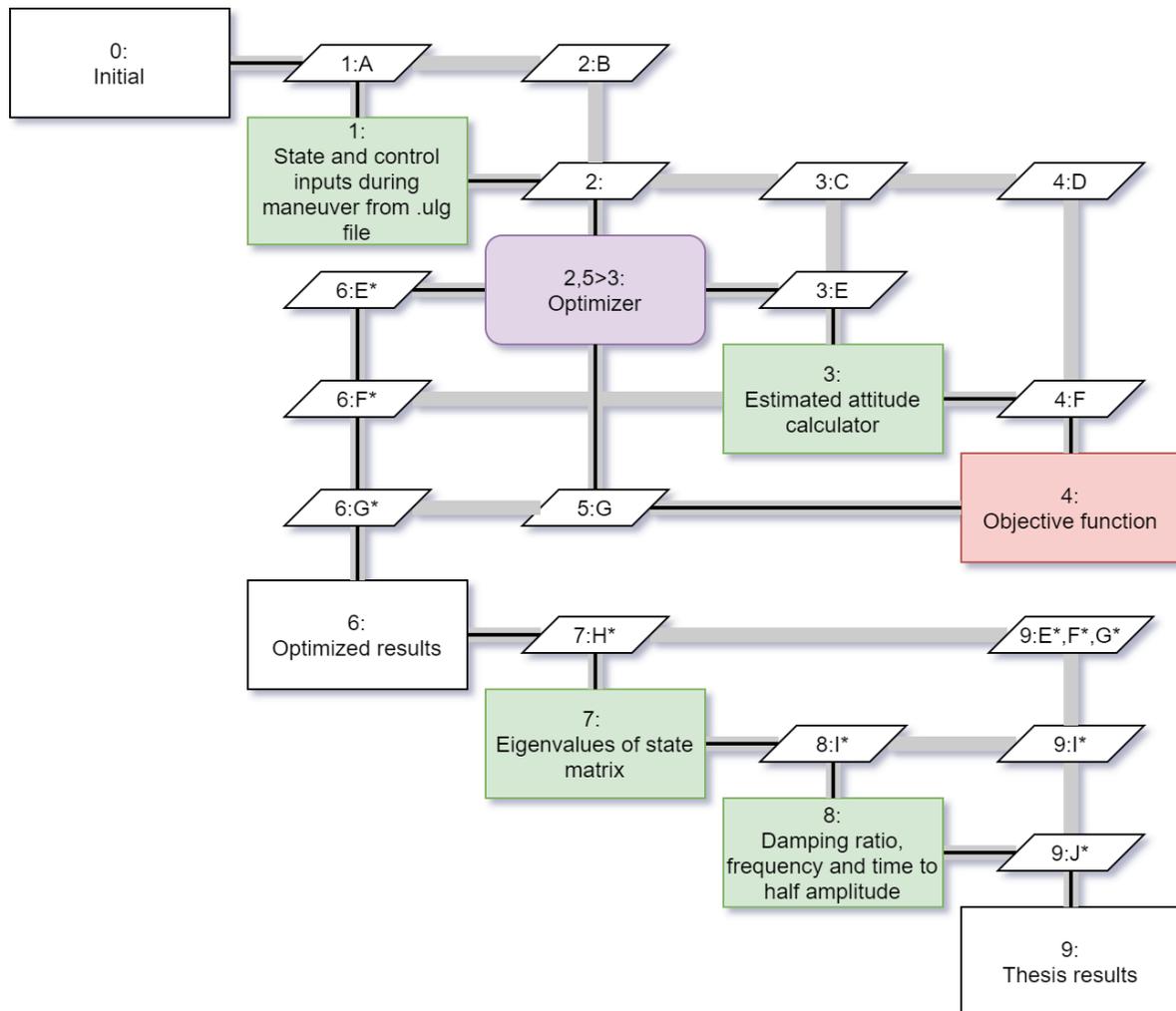


Figure 3.11 Overview of method to determine the lateral-directional stability characteristics from flight tests

Table 3.18 Overview of letter variables in Figure 3.11

A	Data .ulg file
B	Zero vector as initial conditions for optimization vector with 12 State matrix unknowns, 6 control matrix unknowns, 4 (or 8 in case of double flight test estimation) initial state estimate unknowns, β bias during measurements, ϕ bias during measurements, δ_a bias during measurements, δ_r bias during measurements
C	Measured values for PWM signal $\delta_a\{i\}$ and $\delta_r\{i\}$ sent to servos
D	Measured $\beta_m\{i\}, \phi_m\{i\}, p_m\{i\}, r_m\{i\}$
E	Optimization vector with 12 state matrix unknowns, 6 control matrix unknowns, 4 (or 8 in case of double flight test estimation) initial state estimate unknowns, β bias during measurements, ϕ bias during measurements, δ_a bias during measurements, δ_r bias during measurements
F	Estimated $\beta_e\{i\}, \phi_e\{i\}, p_e\{i\}, r_e\{i\}$
G	Cost of optimization vector, $\sum_i\{c1 \cdot (\beta_m\{i\} - \beta_e\{i\})^2 + c2 \cdot (\phi_m\{i\} - \phi_e\{i\})^2 + c3 \cdot (p_m\{i\} - p_e\{i\})^2 + c4 \cdot (r_m\{i\} - r_e\{i\})^2\}$
H	Optimized vector with 12 state matrix unknowns
I	4 eigenvalues of optimized state matrix
J	Damping ration ζ , frequency ω_n and time to half amplitude $T_{\frac{1}{2}}$ of the 4 eigenvalues

The data stored on the SD card located in the flight controller is in the form of a .ulg file. This file can not be directly read. To be able to read the data a python program is created. Within the python program first the .ulg file is transformed to a number of .csv files. Next the data from the .csv files that are of interest to the lateral-directional stability analysis are interpolated at a user defined frequency and in a user defined time range. These data are written to a text file that can be read by the main MATLAB program. The code is shown in Appendix D. Within this code the roll, pitch and yaw angle are calculated from the quaternions saved within the .ulg file according to Equation (23), Equation (24) and Equation (25). From these and other measurements the sideslip angle is calculated as shown in Equation (26).

To obtain the state space system shown in Equation (39), an optimization program is written that uses the fmincon optimizer build into MATLAB. This allows the greatest flexibility for optimizing the results.

There exists a system identification toolbox within MATLAB as well. This is also tested, however the resulting models showed fairly bad behavior. The toolbox does not allow the user to set the cost function for the optimization leading to a bad result for the test data. It is therefore chosen to create the optimization as a separate code. The code created to perform the optimization is shown in Appendix E.

The optimizer chooses a certain number of unknowns, 26 in the case of a single and 34 in the case of a double flight test optimization. Below an explanation is provided for all unknowns.

There are 12 unknowns for the state matrix in the state space system, given that the equation for the roll angle is known (integral of the roll rate).

Using the known values from the roll angle equation the control matrix has 6 unknowns.

To allow for the bad mounting position and or calibration, the sideslip angle and roll angle estimated by the system get shifted by an unknown value up or down. This results in 2 additional unknowns for the single case and 4 additional unknowns in the double flight test case.

The input control signal for rudder and aileron is also given an unknown shift up or down to adjust for the slight misalignment of the control servos and wind conditions. This results again in 2 additional unknowns for the single case and 4 additional unknowns in the double flight test case.

The last 4 unknowns for the single flight test and 8 unknowns for the double flight test case are used to setup a starting vector containing the sideslip angle, roll angle, roll rate and yaw rate at the start time of the maneuver.

Within the function that determines the cost given all the unknowns, the program creates the state space system using unknowns, inputs to the state space system from the recorded servo deflection and a starting position of the states $[\beta(0) \ \phi(0) \ p(0) \ r(0)]^t$ from unknowns. Once these things are created, they are then used to call the lsim function which creates the output of the model assuming the control signal between the input points is linear. The lsim function performs a numerical integration of the state space system following Equation (40) below. Where the time step Δt is set very low and the deflection of the aileron and rudder are interpolated linearly between two data points. The only data that is saved is the state vector at the time instances where the control inputs are provided.

$$\begin{bmatrix} \beta(i+1) \\ \phi(i+1) \\ p(i+1) \\ r(i+1) \end{bmatrix} = \begin{bmatrix} X & X & X & X \\ 0 & 0 & 1 & 0 \\ X & X & X & X \\ X & X & X & X \end{bmatrix} \cdot \begin{bmatrix} \beta(i) \\ \phi(i) \\ p(i) \\ r(i) \end{bmatrix} \cdot \Delta t + \begin{bmatrix} X & X \\ 0 & 0 \\ X & X \\ X & X \end{bmatrix} \cdot \begin{bmatrix} \delta_a(i) \\ \delta_r(i) \end{bmatrix} \cdot \Delta t + \begin{bmatrix} \beta(i) \\ \phi(i) \\ p(i) \\ r(i) \end{bmatrix} \quad (40)$$

Once this output is known, a cost function is used which multiplies a factor with the squared difference of the estimated and measured state at each point. These are all summed and provided as a cost back to the optimizer as shown in Equation (41) below, where $UD1$, $UD2$, $UD3$ and $UD4$ are user defined numbers to allow for different paths to be taken by the optimizer which sometimes result in very different local minima.

$$\begin{aligned} cost = \sum & \left(UD1 \cdot (\Delta\beta(i) - \beta_{offset})^2 + UD2 \cdot (\Delta\phi(i) - \phi_{offset})^2 \right. \\ & \left. + UD3 \cdot (\Delta p(i))^2 + UD4 \cdot (\Delta r(i))^2 \right) \end{aligned} \quad (41)$$

Given that there are no constraints given, the algorithm that is used for the optimization is a quasi-newton algorithm which uses a Broyden-Fletcher-Goldarb-Shanno update. This method depends on a quadratic model of the objective function around the iteration position as shown in Equation (42) below [20]. In this f is the cost function, m the model of the cost function around the current position, H is the approximated Hessian at the current position, t is the transpose and p is a vector that allows to move from the current position. All are given at iteration number k .

$$m_k(p) = f_k + \nabla f_k^t p + \frac{1}{2} p^t H_k p \quad (42)$$

The search direction p_k is found from Equation (43) below.

$$p_k = -H_k^{-1} \nabla f_k \quad (43)$$

The new iteration variables x_{k+1} are formed from Equation (44) below. In this α_k is the step size which is chosen such that it satisfies the Wolfe conditions and minimizes the model of the cost function using a line search [20].

$$x_{k+1} = x_k + \alpha_k p_k \quad (44)$$

The Wolfe conditions are shown in Equation (45) below. In this $0 < c_1 < c_2 < 1$ [20].

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^t p_k \wedge \nabla f(x_k + \alpha_k p_k)^t p_k \geq c_2 \nabla f_k^t p_k \quad (45)$$

To make the program efficient by not having to calculate the Hessian at all iterations, the Hessian is updated using the knowledge of the last iterations as shown in Equation (46) below. In this ρ and y a placeholder variable defined later [20].

$$H_{k+1}^{-1} = (I - \rho_k \alpha_k p_k y_k^t) H_k^{-1} (I - \rho_k y_k (\alpha_k p_k)^t) + \rho_k \alpha_k p_k (\alpha_k p_k)^t \quad (46)$$

The placeholder variable ρ is defined in Equation (47) below [20].

$$\rho_k = \frac{1}{y_k^t \alpha_k p_k} \quad (47)$$

The placeholder variable y is the average hessian multiplied by the step that is performed as shown in Equation (48) below. In this τ is just an integration constant [20].

$$y_k = \left[\int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau \right] \alpha_k p_k \quad (48)$$

Once the state space system is known, the MATLAB function damp is used to find the eigenvalues, damping, natural frequency and time to double amplitude of the state space system. The eigenvalues are calculated from Equation (49) below. In this A is the state matrix, I is the identity matrix and λ are the eigenvalues. For the eigenvalues the determinant of the state matrix minus the eigenvalue multiplied by the identity matrix equals zero. This results in four eigenvalues for the 4x4 matrix that is used for the lateral-directional dynamics.

$$|A - \lambda I| = 0 \quad (49)$$

To calculate the natural frequency of the system ω_n , Equation (50) below is used. In this \mathbb{C} is the complex part and \mathbb{R} the real part.

$$\omega_n(i) = \left(\mathbb{C}(\lambda(i))^2 + \mathbb{R}(\lambda(i))^2 \right)^{0.5} \quad (50)$$

Next the damping ratio ζ is calculated according to Equation (51) below.

$$\zeta(i) = -\cos \left(\tan^{-1} \left(\frac{\mathbb{C}(\lambda(i))}{\mathbb{R}(\lambda(i))} \right) \right) \quad (51)$$

Lastly the time to half amplitude $T_{\frac{1}{2}}$ is calculated using Equation (52).

$$T_{\frac{1}{2}}(i) = \frac{1}{\omega_n(i)\zeta(i)} \quad (52)$$

3.5 Method used to determine the lateral-directional stability characteristics of the aircraft from a vortex lattice method

To perform a vortex lattice method, an XFLR5 model of the aircraft is created. XFLR5 allows the user to perform a vortex lattice method on a given wing and tail configuration. The estimation for the stability of the aircraft is performed assuming ring vortices at all elements defined on the wings and tailplane. The advantage of the ring vortex elements is the small programming effort as well as the fact that the boundary conditions will be satisfied exactly at the wing surface, allowing it to have thickness and camber, compared to a lifting line method where this is not possible. The ring vortex element which is placed at all surfaces modeled is shown in Figure 3.12. Besides these ring vortices at all surfaces, the trailing edge of the wings are given a ring vortex with the strength to make the trailing edge vorticity zero. These rings are given an infinite length. This allows the airstream to leave the trailing edge smoothly to fulfill the Kutta condition [21].

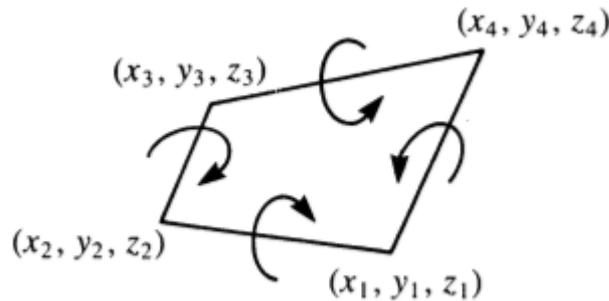


Figure 3.12 Single vortex ring element at a surface element [21]

When combining all ring vortex elements and setting up boundary conditions where the flow is required to be parallel to the surface on the center of every surface element defined, the strength of all ring vortex elements can be determined. The condition of the flow being parallel to the surface is given in Equation (53) below. In this \bar{n} is the normal direction to the surface element, Φ the velocity potential due to the different ring vortices and Φ_{∞} the free stream velocity potential [21].

$$\nabla(\Phi + \Phi_{\infty}) \cdot \bar{n} = \bar{0} \quad (53)$$

The effect of a single vortex element on a given position on the flow velocity in u , v and w direction is shown in Equation (54) below. For a single ring vortex element four of these calculations are made. These are combined in a big matrix that is solved to find the vorticity per unit length Γ of all vortex rings. In this equation $R1R2X$, $R1R2Y$, $R1R2Z$, $R1$, $R2$, $R0R1$ and $R0R2$ are placeholder variables defined later [21].

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} R1R2X \\ R1R2Y \\ R1R2Z \end{bmatrix} \cdot \frac{\Gamma \cdot \frac{R0R1}{R1} \cdot \frac{R0R2}{R2}}{4 \cdot \pi \cdot (R1R2X \cdot R1R2X + R1R2Y \cdot R1R2Y + R1R2Z \cdot R1R2Z)} \quad (54)$$

The definitions of the different placeholder variables in the equation above are defined in Equation (55) below. In these the X, Y, Z is the position where the local velocity is determined, X_1, Y_1, Z_1 is the location at which the vortex line starts and X_2, Y_2, Z_2 is the location where the vortex line ends [21].

$$\begin{bmatrix} R1R2X \\ R1R2Y \\ R1R2Z \\ R1 \\ R2 \\ ROR1 \\ ROR2 \end{bmatrix} = \begin{bmatrix} (Y - Y_1) \cdot (Z - Z_2) - (Z - Z_1) \cdot (Y - Y_2) \\ (Z - Z_1) \cdot (X - X_2) - (X - X_1) \cdot (Z - Z_2) \\ (X - X_1) \cdot (Y - Y_2) - (Y - Y_1) \cdot (X - X_2) \\ \sqrt{(X - X_1)(X - X_1) + (Y - Y_1)(Y - Y_1) + (Z - Z_1)(Z - Z_1)} \\ \sqrt{(X - X_2)(X - X_2) + (Y - Y_2)(Y - Y_2) + (Z - Z_2)(Z - Z_2)} \\ (X_2 - X_1) \cdot (X - X_1) + (Y_2 - Y_1) \cdot (Y - Y_1) + (Z_2 - Z_1) \cdot (Z - Z_1) \\ (X_2 - X_1) \cdot (X - X_2) + (Y_2 - Y_1) \cdot (Y - Y_2) + (Z_2 - Z_1) \cdot (Z - Z_2) \end{bmatrix} \quad (55)$$

The output file that is created from the XFLR5 stability analysis shows the state space system as well as the eigenvalues, damping ratio, frequency and time to half amplitude of the different modes. This is created by changing the free stream conditions to see the effect of sideslip, roll angle, yaw rate and roll rate. Using the values that are obtained from this analysis the state space system can be created using Equation (38). It must be noted that the aileron and rudder are not modeled and such only the state matrix is estimated. To calculate the aerodynamic characteristics used to calculate the state matrix also the weight, center of gravity and moments of inertia have to be known. The aerodynamic analysis of the roll angle, yaw angle, roll rate and yaw rate are performed by changing the freestream velocity such that the aircraft is rotated around the center of gravity location.

In order to input the correct weight and moments of inertia into the XFLR5 stability analysis, a model of the aircraft including its components is created in Catia V5 which allows to directly extract the moments of inertia. First the center of gravity is calculated using Equation (56) below. In this $\rho(x, y, z)$ is the density distribution as defined by the model that is put into the software, which is a function of the position in x, y, z . $[X_{cg} \ Y_{cg} \ Z_{cg}]^t$ is the center of gravity vector in x, y, z .

$$\begin{bmatrix} X_{cg} \\ Y_{cg} \\ Z_{cg} \end{bmatrix} \cdot \int_{x_{start}}^{x_{end}} \int_{y_{start}}^{y_{end}} \int_{z_{start}}^{z_{end}} \rho(x, y, z) \, dzdydx = \begin{bmatrix} \int_{x_{start}}^{x_{end}} \int_{y_{start}}^{y_{end}} \int_{z_{start}}^{z_{end}} \rho(x, y, z) \cdot x \, dzdydx \\ \int_{x_{start}}^{x_{end}} \int_{y_{start}}^{y_{end}} \int_{z_{start}}^{z_{end}} \rho(x, y, z) \cdot y \, dzdydx \\ \int_{x_{start}}^{x_{end}} \int_{y_{start}}^{y_{end}} \int_{z_{start}}^{z_{end}} \rho(x, y, z) \cdot z \, dzdydx \end{bmatrix} \quad (56)$$

Given the result of the center of gravity location, the moments of inertia around the center of gravity position can be calculated. The equation is provided for a random moment of inertia, namely I_{bc} , is given in Equation (57) below. By inputting the correct axis as b and c , as well as the corresponding b_{cg} and c_{cg} , the moment of inertia will be calculated around those axes. Examples can include I_{xx}, I_{zz}, I_{xz} , which are the once of interest for the lateral-directional stability characteristics. It must be noted that the axis system used within the Catia V5 model

is different from the axis system used in the report which requires to change some variables such that they can be used in this thesis.

$$I_{bc} = \int_{x_{start}}^{x_{end}} \int_{y_{start}}^{y_{end}} \int_{z_{start}}^{z_{end}} \rho(x, y, z) \cdot (b - b_{cg}) \cdot (c - c_{cg}) \, dzdydx \quad (57)$$

4 Verification

The verification of the optimization to find the best fit for the model is divided in two parts. The first deals with the exact output of a model to estimate the model. The second deals with the output of a model with an additional noise level to estimate the original model.

4.1 Using exact output data of a model to estimate the model

As a verification of the method used, an existing state space model using parameters of the Cessna Ce500 'Citation' [19] is used to generate the input to the program. Using these generated inputs, it is checked if the results that are obtained from the analysis are equal to those used to generate the input of the analysis.

The values used to generate the state space system from Equation (36) are shown in Table 4.1. The resulting state space system up to two significant digits is shown in Equation (58) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \frac{\dot{pb}}{2V} \\ \frac{\dot{rb}}{2V} \end{bmatrix} = \begin{bmatrix} -0.14 & 0.16 & -0.013 & -8.9 \\ 0 & 0 & 9.0 & 0 \\ -0.42 & 0 & -2.9 & 1.6 \\ 0.30 & 0 & -0.13 & -0.29 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} + \begin{bmatrix} 0 & 0.044 \\ 0 & 0 \\ -1.4 & 0.13 \\ -0.021 & -0.24 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (58)$$

Table 4.1 Asymmetric stability and control derivatives of the Cessna Ce500 'Citation' [19]

$\frac{m}{\rho S b}$	15.5	V	59.9 m/s
$\frac{I_{xx}}{mb^2}$	0.012	$C_{l\beta}$	-0.0772
$\frac{I_{zz}}{mb^2}$	0.037	C_{lp}	-0.3444
$\frac{I_{xz}}{mb^2}$	0.002	C_{lr}	0.2800
b	13.36 m	$C_{l\delta_a}$	-0.2349
C_L	1.1360	$C_{l\delta_r}$	0.0286
$C_{Y\beta}$	0	$C_{n\beta}$	0
C_{Yp}	-0.9896	$C_{n\beta}$	0.1638
C_{Yr}	-0.0870	C_{np}	-0.0108
$C_{Y\delta_a}$	0.4300	C_{nr}	-0.1930
$C_{Y\delta_r}$	0	$C_{n\delta_a}$	0.0286
	0.3037	$C_{n\delta_r}$	-0.1261

As an input both the aileron and rudder are excited but with a different formula. This is done to simulate the stabilized flight mode which causes inputs to the aileron and rudder at all times due to the corrections of the flight controller. Equation (59) below shows the input of the aileron. In this H is the Heaviside step function, t is the time and δ_a the aileron deflection.

$$\delta_a(t) = -H(t - 3) \cdot 0.005 \cdot \sin((t - 3) \cdot 2 \cdot \pi \cdot 0.01 \cdot t) \quad (59)$$

Equation (60) below shows the rudder deflection δ_r .

$$\delta_r(t) = H(t - 1) \cdot 0.025 \cdot \sin((t - 1) \cdot 2 \cdot \pi \cdot 0.01 \cdot t) \quad (60)$$

In the last two equations H is the Heaviside step function as defined in Equation (61) below.

$$H(x) = 0 \forall x \leq 0 \wedge H(x) = 1 \forall x > 0 \quad (61)$$

The resulting inputs are shown in Figure 4.1. The function is sampled at 30Hz. Running the state space system with a $\bar{0}$ original state vector, the resulting outputs are shown in Figure 4.2.

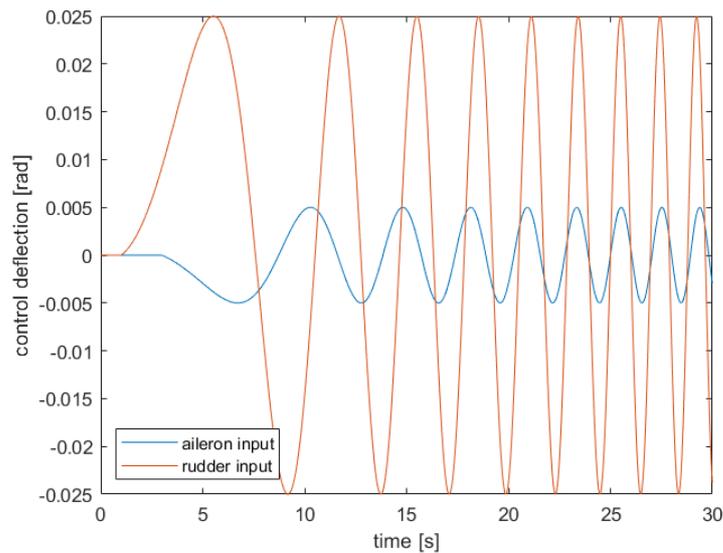


Figure 4.1 Control deflection input to the state space system of the Cessna Ce500 'Citation'

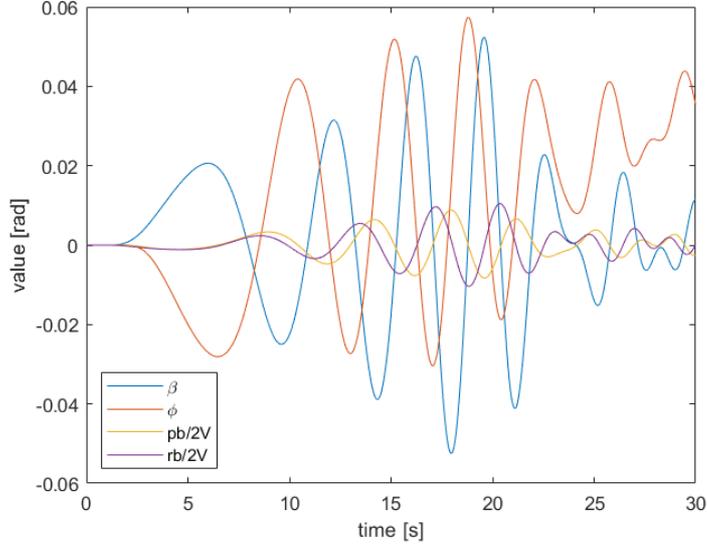


Figure 4.2 Outputs of the state space system of the Cessna Ce500 'Citation'

Given that the optimization tries to find a minimum, this minimum is not guaranteed to be the global minimum. The starting point of the optimization is defined in Equation (62) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{pb} \\ \dot{rb} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 9.0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (62)$$

The cost function used for the optimization is shown in Equation (63) below.

$$cost = \sum \left((\Delta\beta - \beta_{offset})^2 + (\Delta\phi - \phi_{offset})^2 + (\Delta p)^2 + (\Delta r)^2 \right) \quad (63)$$

The resulting output of the estimation compared to the actual outputs are shown in Figure 4.3. In this figure it can be seen that the results of the original model and the estimated, optimized model are pretty much exactly the same. Within Equation (62) the numbers marked in red show the numbers that are fixed during the estimation. The optimized state space system, up to two significant digits is shown in Equation (64) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{pb} \\ \dot{rb} \end{bmatrix} = \begin{bmatrix} -0.14 & 0.16 & 0.0014 & -8.9 \\ 0 & 0 & 9.0 & 0 \\ -0.42 & 0 & -2.1 & 1.6 \\ 0.30 & 0 & -0.14 & -0.29 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} + \begin{bmatrix} -0.0012 & 0.044 \\ 0 & 0 \\ -1.4 & 0.13 \\ -0.023 & -0.24 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (64)$$

The total cost of this answer is approximately $5.9 \cdot 10^{-9}$. The optimization options used for the analysis are no upper and lower bounds on all variables, a maximum of $1 \cdot 10^6$ iterations, a maximum of $3 \cdot 10^5$ function evaluations and a tolerance of $1 \cdot 10^{-23}$. These options are chosen to limit the maximum time of the analysis and stopping when very close to a local minimum. The final answer is found after 664 iterations with 12951 function evaluations. The first order optimality is approximately $6.5 \cdot 10^{-4}$. The changes in all variables for a next step are below the tolerance set. This combined shows that the answer found is a local minimum. We know that it is very close to the global minimum, but this can not be guaranteed for an unknown system.

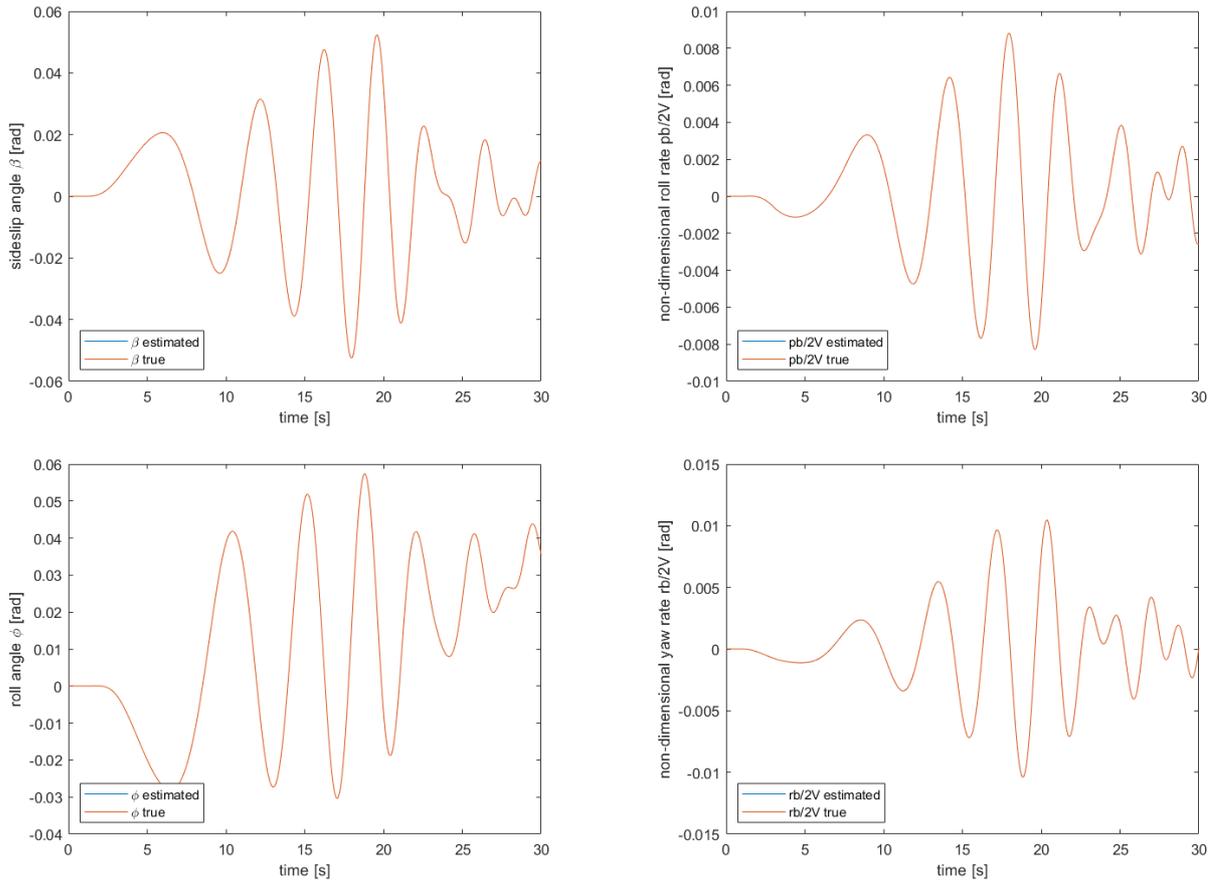


Figure 4.3 Resulting outputs of the estimated state space model compared to the true (original) state space model using Equation (62) as the starting point of the optimization

In order to compare the results in a different way, the eigenvalues of the different state matrices are calculated. The results of the eigenvalue analysis are shown in Table 4.2. It can be seen that one mode is real and highly damped, one pair is complex conjugate and moderately damped and the last one is negatively damped.

Table 4.2 Eigenvalue comparison between the original and estimated system rounded to two significant digits

Eigenvalue original system	Eigenvalue estimated system
-2.2 + 0.0i	-2.2 + 0.0i
-0.19 + 1.8i	-0.19 + 1.8i
-0.19 - 1.8i	-0.19 - 1.8i
0.076 + 0.0i	0.076 + 0.0i

Table 4.3 shows the three eigenmodes up to two significant digits. A highly damped roll subsidence mode, a moderately damped Dutch roll oscillation and an unstable spiral mode can be seen. As can be expected from choosing the least square error, the unstable mode is very accurately found, the moderately damped oscillatory mode is also very accurately found and the highly damped mode is found with a bit higher error, but these are still extremely close. This result is expected due to the damping already being high, meaning there will be less of a difference if the damping is slightly different. Within the two significant digits shown, no difference is observed.

Table 4.3 Damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model and the original model up to two significant digits

Mode	Damping ratio ζ		Frequency ω_n [rad/s]		Time to half amplitude $T_{\frac{1}{2}}$ [s]	
	Original	Estimated	Original	Estimated	Original	Estimated
Roll subsidence	1	1	2.2	2.2	0.45	0.45
Dutch roll	0.11	0.11	1.8	1.8	5.4	5.4
Spiral	-1	-1	0.076	0.076	-13	-13

4.2 Using noisy output data from a model to estimate the model

Using the same exact model used before, an additional random noise element is added to all output variables. The random noise level for the sideslip angle β is selected to be ± 0.002 . The random noise level for the roll angle ϕ is selected to be ± 0.002 . The random noise level for the non-dimensional roll rate $\frac{pb}{2V}$ is selected to be ± 0.00035 . The random noise level for the non-dimensional yaw rate $\frac{rb}{2V}$ is selected to be ± 0.0005 . The selected numbers correspond to around a $\pm 5\%$ noise with respect to the range of the data. The noisy data compared to the true data is shown in Figure 4.4.

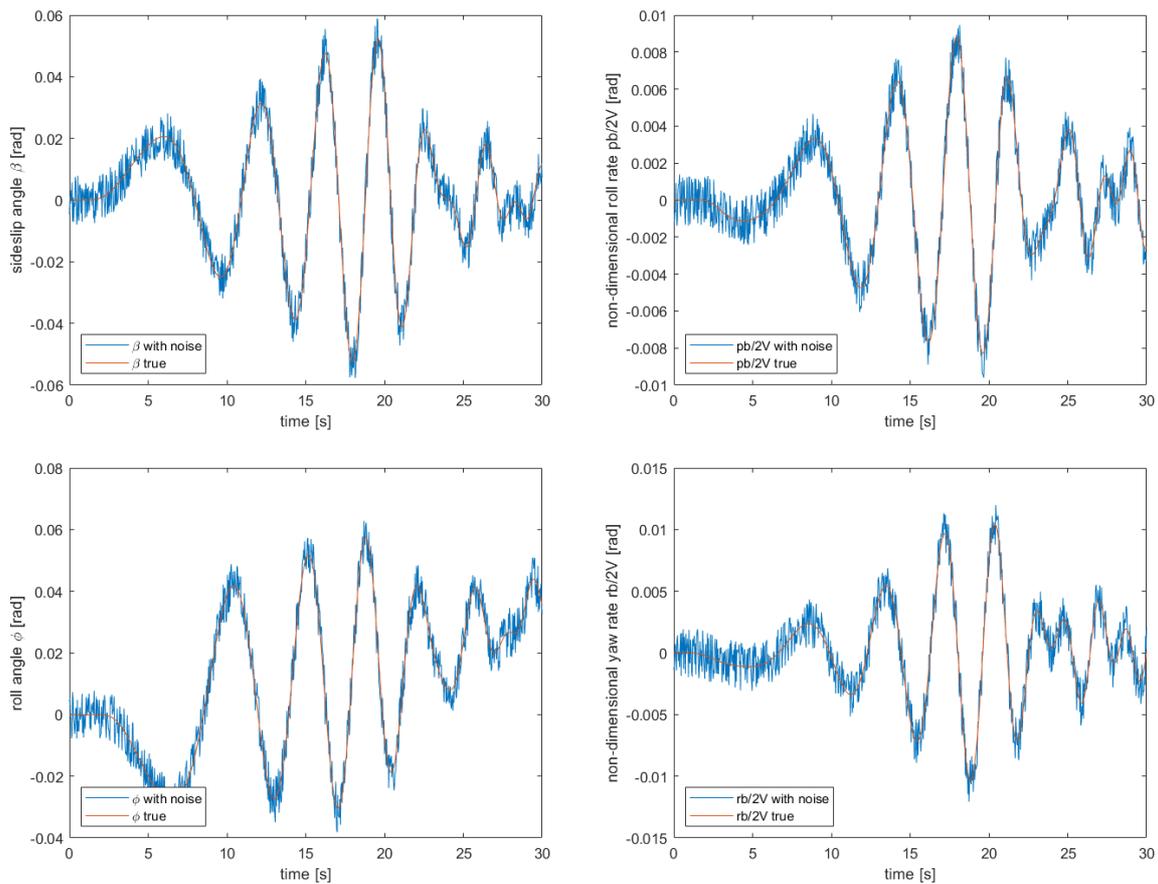


Figure 4.4 Output data of the model compared to the noisy output data used for the estimation of the model

Using the same starting point and optimization options as used for the exact model before, the resulting state space model, up to two significant digits is shown in Equation (65) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \frac{\dot{pb}}{2V} \\ \frac{\dot{rb}}{2V} \end{bmatrix} = \begin{bmatrix} -0.061 & 0.094 & 0.34 & -9.3 \\ 0 & 0 & 9.0 & 0 \\ -0.50 & 0.016 & -2.4 & 1.7 \\ 0.30 & -0.0028 & -0.11 & -0.31 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} + \begin{bmatrix} 0.75 & -0.053 \\ 0 & 0 \\ -1.8 & 0.16 \\ -0.056 & -0.25 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (65)$$

The total cost, using the cost function shown in Equation (63), of the optimized result is approximately $5.67 \cdot 10^4$. The estimation took 1986 iterations and 138,290 function evaluations. The first order optimality of the answer is approximately $2.46 \cdot 10^{-4}$. The changes in all variables for a next step are below the tolerance set. This again shows that a local minimum is achieved.

To show the behavior of the model that is estimated from the noisy data, the results are shown compared to the original model output in Figure 4.5. As can be seen in Figure 4.5, the estimated model from the noisy is still very similar to the original model.

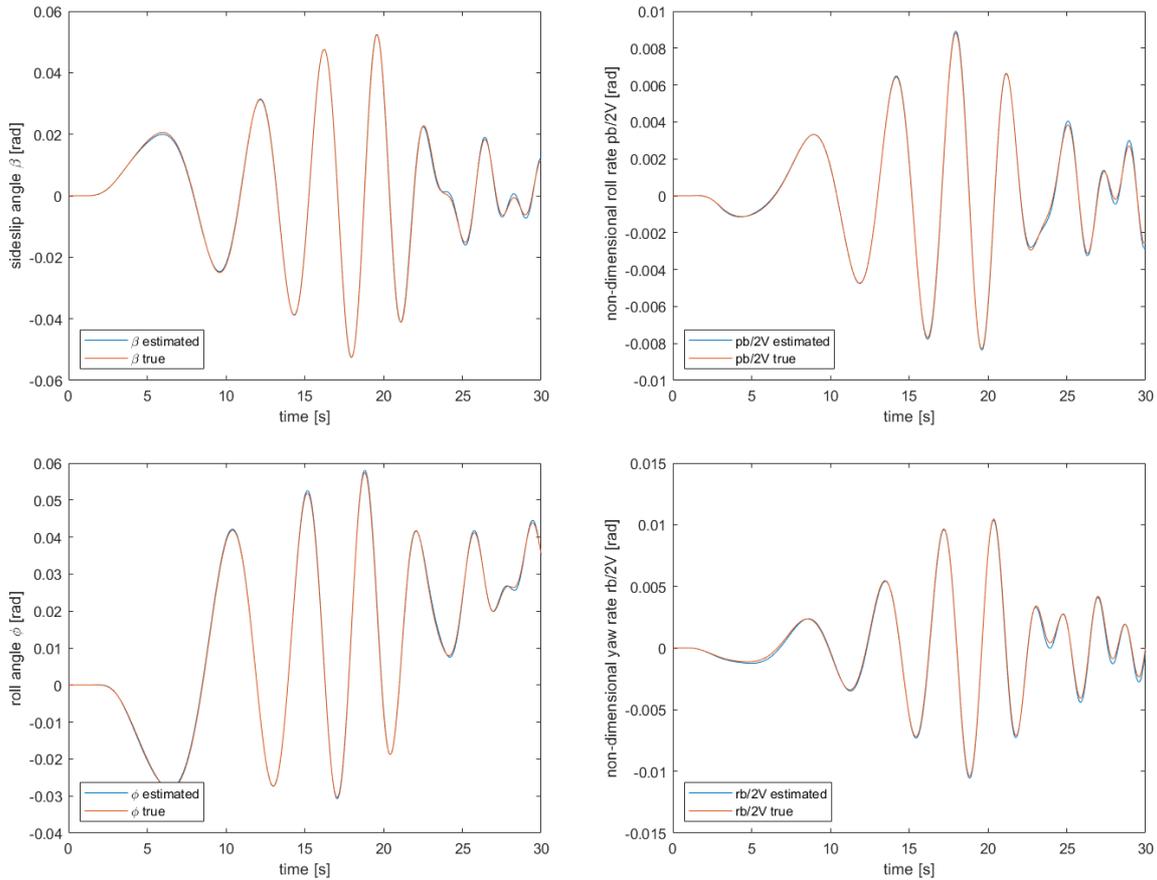


Figure 4.5 Resulting outputs of the estimated state space model from the noisy data compared to the true (original) state space model using Equation (62) as the starting point of the optimization

To have a closer look at the difference in the models, the eigenvalues are compared. Table 4.4 shows the eigenvalues of the estimated system from the noisy data compared to the original system eigenvalues. From these eigenvalues the damping, frequency and time to half amplitude are calculated and shown in Table 4.5. As can be seen, the values are just slightly different, allowing to say that the method used also works for a noisy model. As can be seen the roll subsidence mode has the highest difference. This can be expected because the roll subsidence mode exists on the smallest timescales and is therefore most affected by the random noise added. From these results it can be said that the optimization method used is verified for analyzing a state space model from data with or without noise.

Table 4.4 Eigenvalue comparison between the original and the estimated system from noisy data up to two significant digits

Eigenvalue original system	Eigenvalue estimated system from noisy data
$-2.2 + 0.0i$	$-2.5 + 0.0i$
$-0.19 + 1.8i$	$-0.19 + 1.8i$
$-0.19 - 1.8i$	$-0.19 - 1.8i$
$0.076 + 0.0i$	$0.075 + 0.0i$

Table 4.5 Damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from noisy data and the original model up to two significant digits

Mode	Damping ratio ζ		Frequency ω_n [rad/s]		Time to half amplitude $T_{\frac{1}{2}}$ [s]	
	Original	Estimated	Original	Estimated	Original	Estimated
Roll subsidence	1	1	2.2	2.5	0.45	0.41
Dutch roll	0.11	0.11	1.8	1.8	5.4	5.3
Spiral	-1	-1	0.076	0.075	-13	-13

5 Results

This chapter is divided in four main parts. The first one shows the data obtained from the flight test. The second part estimates state space systems from the experimental test flight. The third part shows the model of the aircraft in Catia V5 as well as the model of the aircraft in XFLR5 with the estimated stability of the aircraft using a ring vortex lattice method. The fourth part discusses the results and determines if the lateral-directional stability characteristics of the aircraft are sufficient according to the military guidelines.

5.1 Flight test data

The flight test data are shown in the figures below. For each figure the left-hand side shows the aileron sweep and the right-hand side shows the rudder sweep. These are smaller snapshots from the entire flight test that are used for the analysis of the state space system of the aircraft. The data for the sideslip angle, roll angle, roll rate and yaw rate are shown in section 5.2.

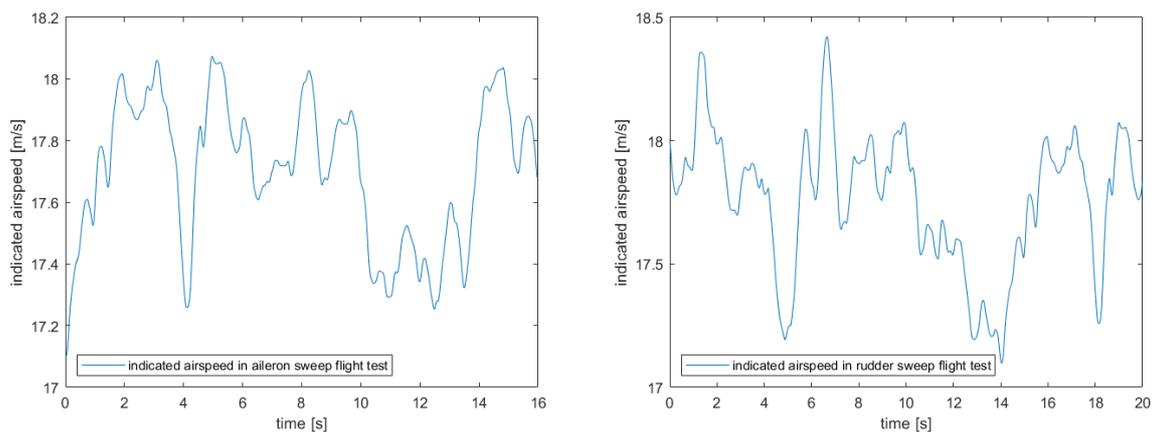


Figure 5.1 Indicated Airspeed during test flight portion containing the aileron sweep on the left and rudder sweep on the right

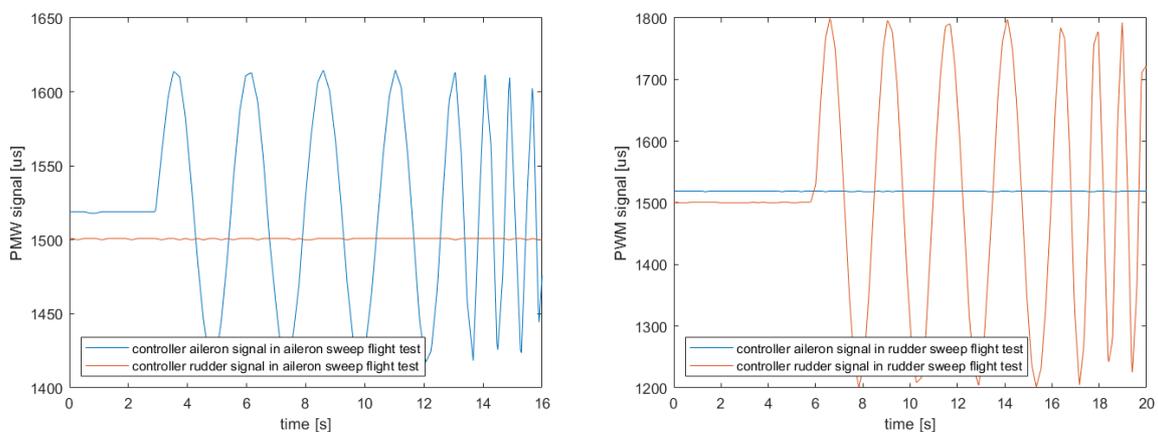


Figure 5.2 Controller inputs during test flight portion containing the aileron sweep on the left and rudder sweep on the right

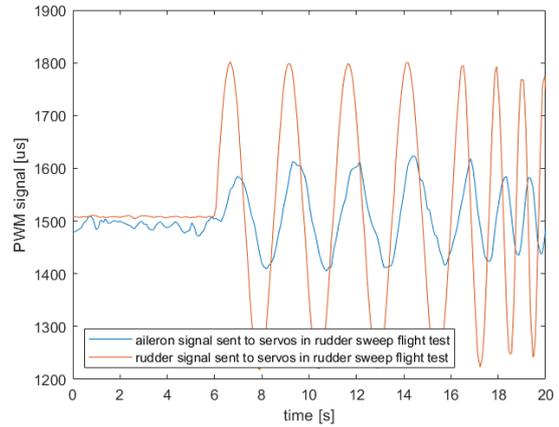
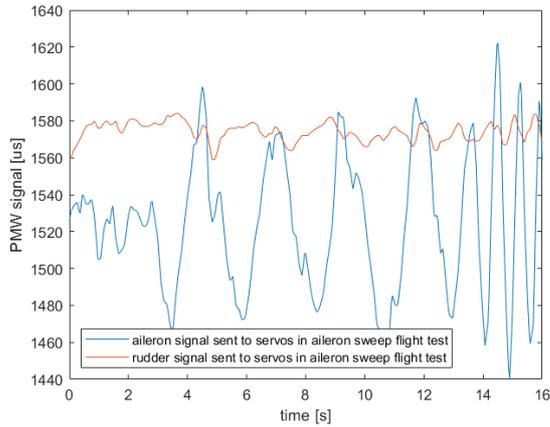


Figure 5.3 Servo input signals during test flight portion containing the aileron sweep on the left and rudder sweep on the right

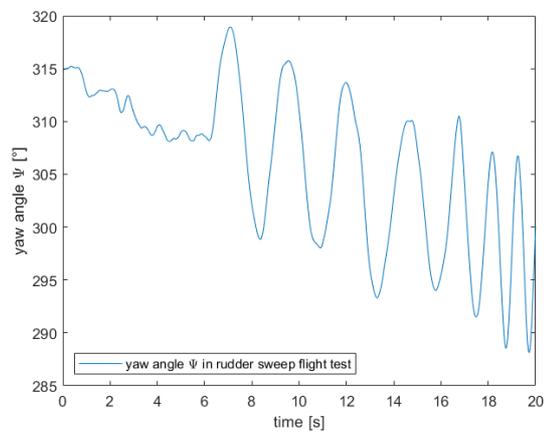
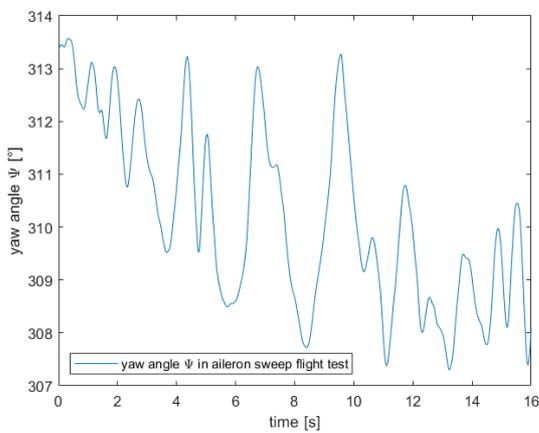


Figure 5.4 Aircraft yaw angle during test flight portion containing the aileron sweep on the left and rudder sweep on the right

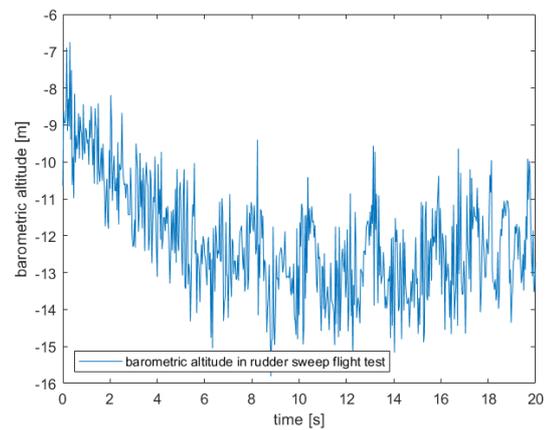
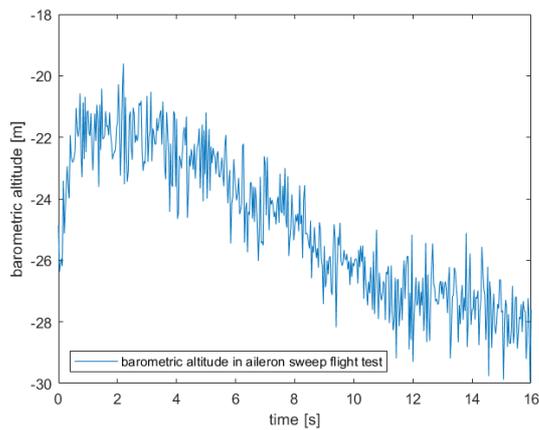


Figure 5.5 Aircraft barometric altitude during test flight portion containing the aileron sweep on the left and rudder sweep on the right

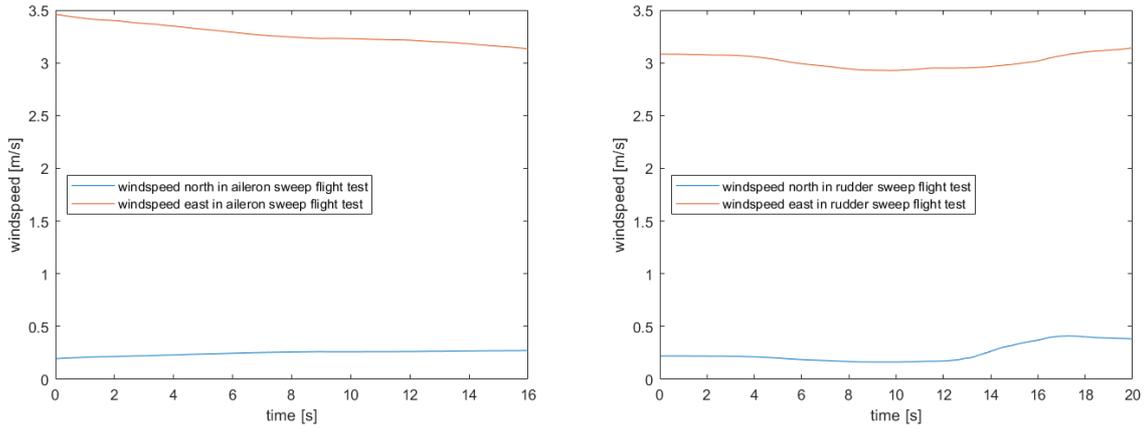


Figure 5.6 Windspeed north and east during test flight portion containing the aileron sweep on the left and rudder sweep on the right

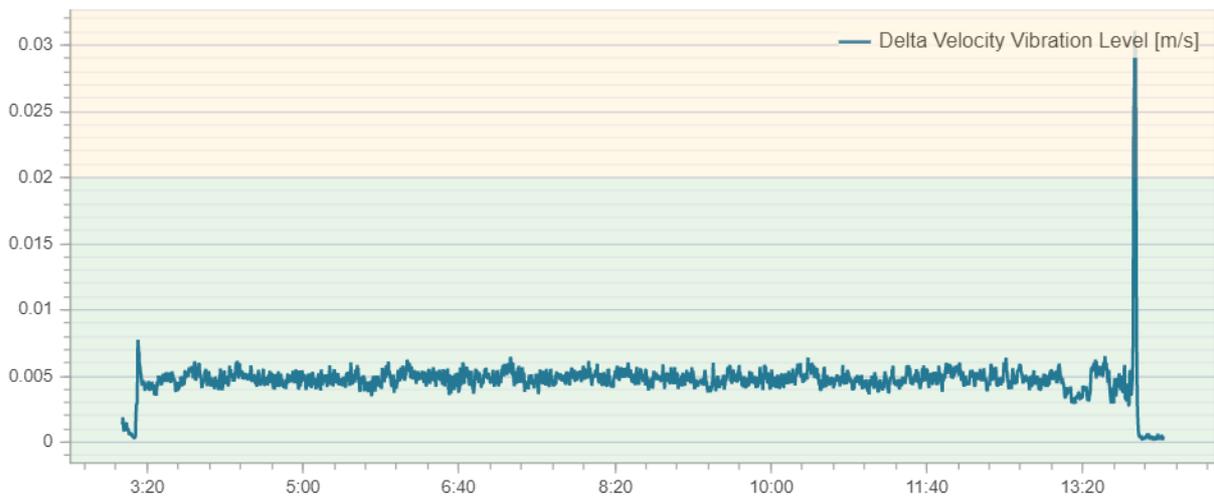


Figure 5.7 Vibration metrics for the entire flight

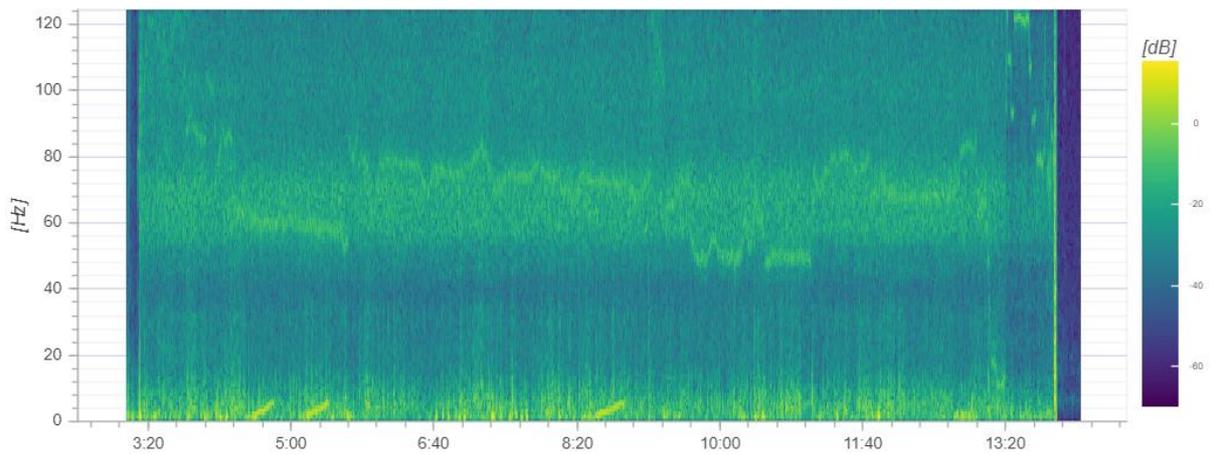


Figure 5.8 Acceleration power spectral density during entire flight

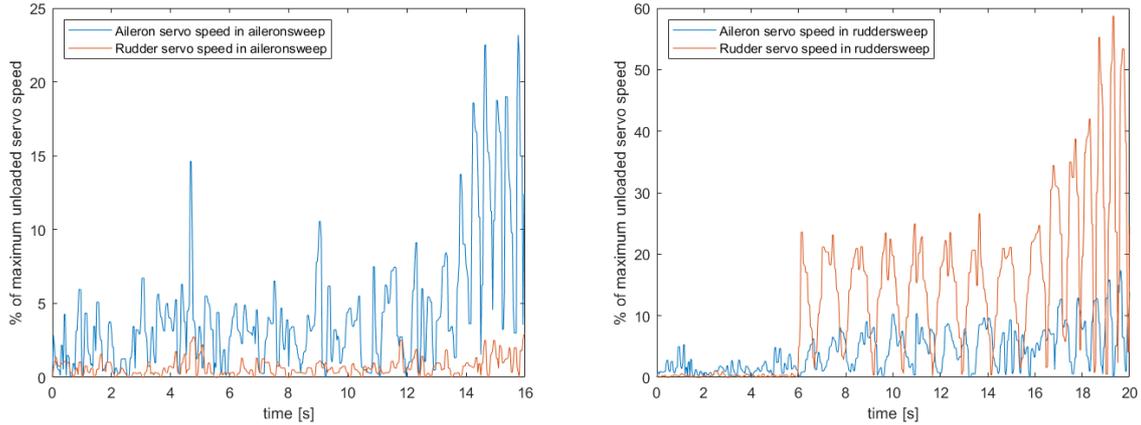


Figure 5.9 Percentage of maximum unloaded servo speed during test flight portion containing the aileron sweep on the left and rudder sweep on the right

5.2 Resulting state space models

The state space system is created for 3 different inputs. The first one uses only the rudder sweep as an input to analyze the system. The second uses only the aileron sweep as an input to analyze the system. The third uses both as inputs to analyze the system. For every state space model, a cost function is tailored to achieve a good fit of the data. The process of updating the cost function is rather simple. In the case that the state space model does not cover one of the 4 inputs correctly, the cost of that specific aspect is increased. In this subchapter only the results are provided. The main conclusions that can be made from the resulting state space models is shown later in Chapter 5.4. The codes that have been used for the analysis are shown in Appendix E.

5.2.1 Rudder sweep only

To obtain the state space model, an optimization with a limit of 1,000,000 function evaluations is run. The cost function used in the optimization is shown in Equation (66) below. The resulting total cost after optimization is approximately $4.15 \cdot 10^9$ with a total of 601 points used in the estimation.

$$\begin{aligned} cost = \sum & \left(1,000,000 \cdot (\Delta\beta - \beta_{offset})^2 + 1,000,000 \cdot (\Delta\phi - \phi_{offset})^2 \right. \\ & \left. + 100 \cdot (\Delta p)^2 + 10,000 \cdot (\Delta r)^2 \right) \end{aligned} \quad (66)$$

The state space system rounded to three significant digits that is found is shown in Equation (67) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -11.5 & -11.3 & 79.0 & -74.8 \\ 0 & 0 & 1 & 0 \\ -13.0 & 11.0 & -120 & 110 \\ -31.9 & -6.92 & 5.66 & -11.4 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 422 & 1.25 \\ 0 & 0 \\ -580 & 114 \\ 104 & 123 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (67)$$

The eigenvalue analysis of the state space model that is estimated rounded to three significant digits is shown in Table 5.1. The eigenvalue analysis shown a slightly unstable spiral mode, a moderately damped Dutch roll mode and a highly damped roll subsidence mode. Figure 5.10 shows the estimated model compared to the measured data. It can be observed that the fit is good on all parameters.

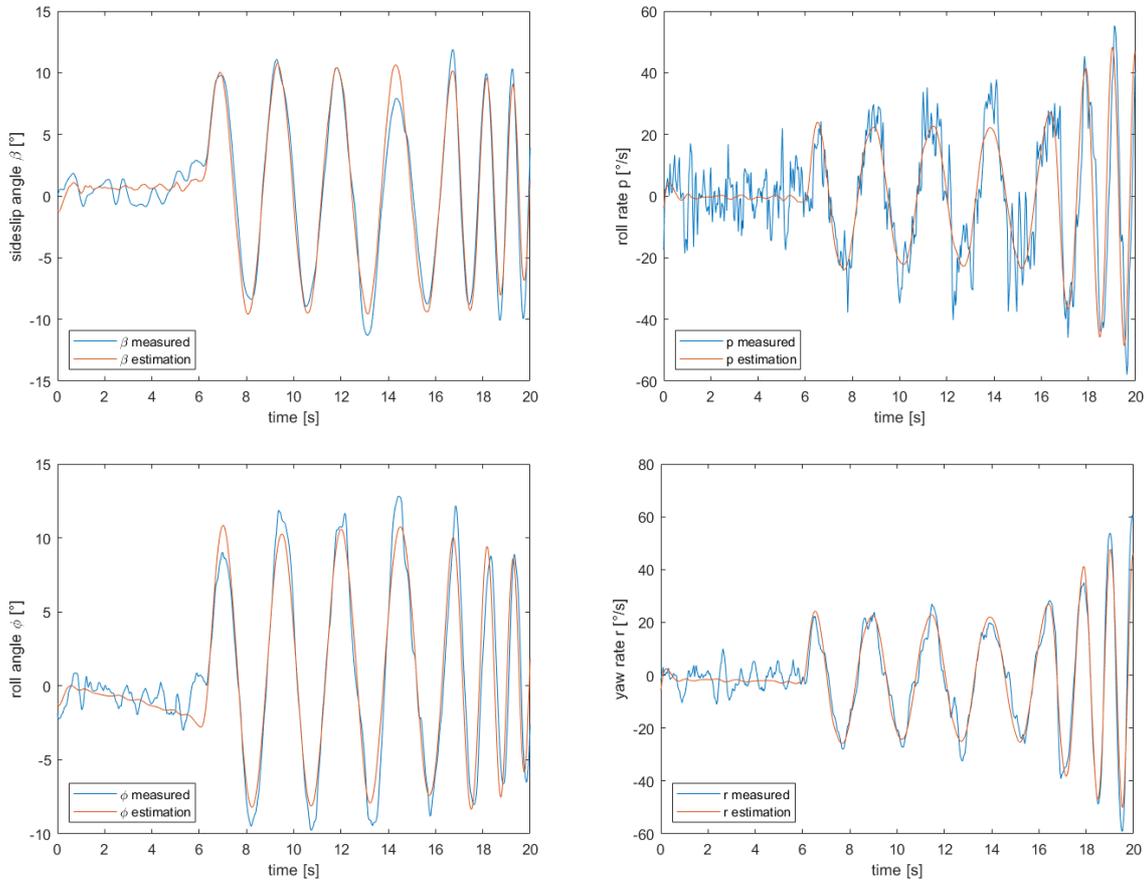


Figure 5.10 State space model created from the rudder sweep estimation of the different states compared to the measured states in the rudder sweep flight test

Table 5.1 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the rudder sweep flight test

Mode	Eigenvalue	Damping ratio ζ	Frequency ω_n [rad/s]	Time to half amplitude $T_{\frac{1}{2}}$ [s]
Roll subsidence	-137 + 0 i	1	137	0.00730
Dutch roll	-3.31 $\pm 4.84 i$	0.564	5.86	0.303
Spiral	0.193 + 0 i	-1	0.193	-5.18

5.2.2 Aileron sweep only

To obtain the state space model, an optimization with a limit of 1,000,000 function evaluations is run. The optimization is completed after 14,846 function evaluations and 491 iterations. The total cost for this model using the cost function shown in Equation (68) for the optimization is approximately $6.33 \cdot 10^9$ with a total of 481 points used in the estimation.

$$\begin{aligned}
 cost = \sum & \left(100,000 \cdot (\Delta\beta - \beta_{offset})^2 + 1,000,000 \cdot (\Delta\phi - \phi_{offset})^2 \right. \\
 & \left. + 100,000 \cdot (\Delta p)^2 + 1000 \cdot (\Delta r)^2 \right) \quad (68)
 \end{aligned}$$

The state space system that is found rounded to three significant digits is shown in Equation (69) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -4.44 & 1.21 & -0.167 & 0.404 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -2.20 & 4.96 & -2.73 & 5.58 \\ 5.77 & 10.2 & -24.3 & -1.20 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 4.64 & 12.0 \\ \mathbf{0} & \mathbf{0} \\ -34.1 & 11.0 \\ -698 & 179 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (69)$$

The eigenvalue analysis, up to three significant digits, of the state space model that is estimated is shown in Table 5.2. The eigenvalue analysis shown a slightly unstable spiral mode, a moderately damped Dutch roll mode and a highly damped roll subsidence mode. Figure 5.11 shows the estimated model compared to the measured data. It can be observed that the fit is quite good for the roll rate. The fit for the roll angle is also still relatively good. The fit for the sideslip and yaw rate are not good.

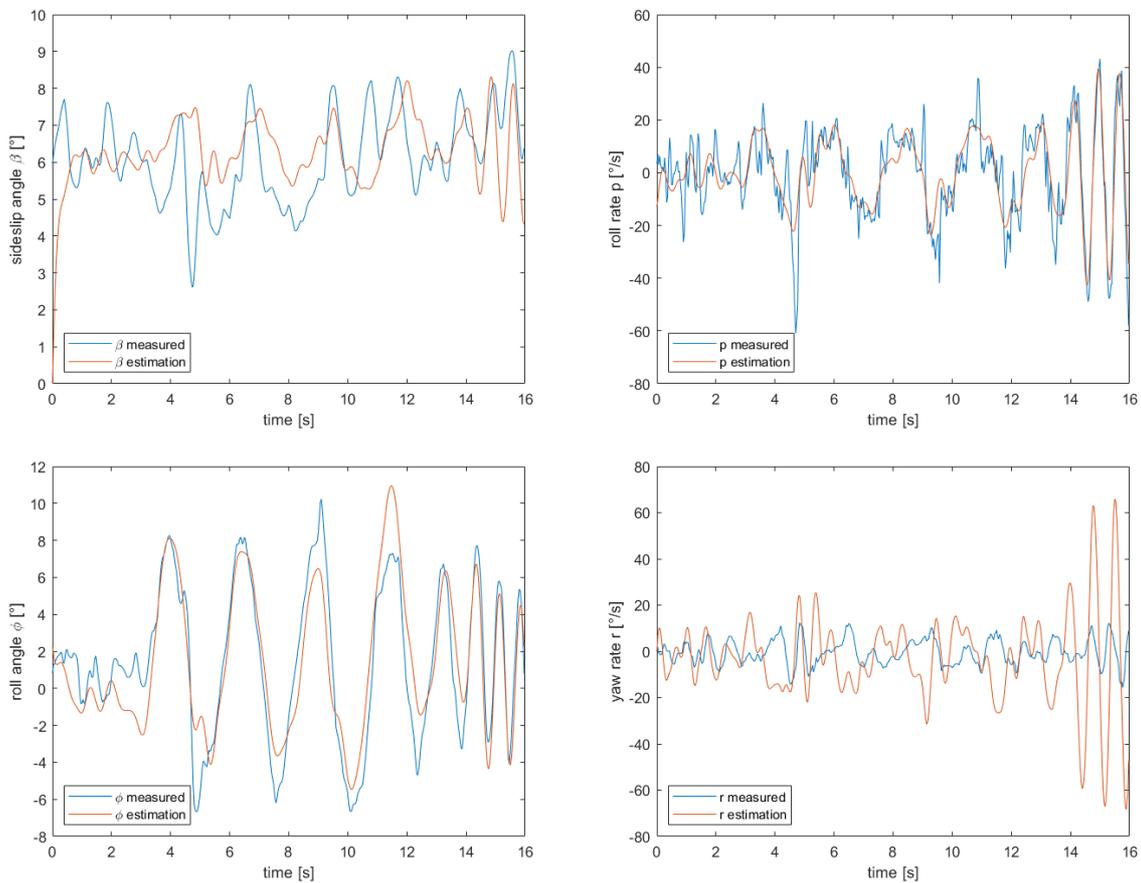


Figure 5.11 State space model created from the aileron sweep estimation of the different states compared to the measured states in the aileron sweep flight test

Table 5.2 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the aileron sweep flight test

Mode	Eigenvalue	Damping ratio ζ	Frequency ω_n [rad/s]	Time to half amplitude $T_{\frac{1}{2}}$ [s]
Roll subsidence	-4.41 + 0 i	1	4.41	0.227
Dutch roll	-2.23 $\pm 11.3 i$	0.193	11.6	0.448
Spiral	0.498 + 0 i	-1	0.498	-2.01

5.2.3 Combination of rudder and aileron sweep

To obtain the state space model for the combination of the rudder and aileron sweep, an optimization with a limit of 1,000,000 function evaluations is run. The optimization is completed after 47,112 function evaluations and 1228 iterations. The total cost for this model using the cost function for the optimization shown in Equation (70) below is approximately $6.33 \cdot 10^9$ with a total of 1082 points used in the estimation.

$$cost = \sum \left(10,000,000 \cdot (\Delta\beta - \beta_{offset})^2 + 100,000 \cdot (\Delta\phi - \phi_{offset})^2 + 1,000,000 \cdot (\Delta p)^2 + 10,000,000 \cdot (\Delta r)^2 \right) \quad (70)$$

The state space system that is found rounded to three significant digits is shown in Equation (71) below.

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -0.127 & -0.162 & -0.0775 & 1.71 \\ 0 & 0 & 1 & 0 \\ 29.0 & 2.71 & -8.70 & 8.55 \\ -20.2 & 1.46 & -2.33 & -2.06 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 1.72 & -7.11 \\ 0 & 0 \\ -327 & -28.8 \\ -46.7 & 81.0 \end{bmatrix} \cdot \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (71)$$

The eigenvalue analysis, rounded to three significant digits of the state space model that is estimated is shown in Table 5.3. The eigenvalue analysis shown a slightly unstable spiral mode, a moderately damped Dutch roll mode and a highly damped roll subsidence mode. Figure 5.12 shows the estimated model compared to the measured data. It can be observed that the fit for the yaw rate, roll rate and sideslip of the rudder sweep are quite good. The roll angle for the rudder sweep is modelled a bit worse. For the aileron sweep part the roll rate and roll angle are modelled quite alright. The sideslip angle and yaw rate are modelled quite badly.

Table 5.3 Eigenvalues, damping ratio, frequency and time to half amplitude for the three different dynamic modes of the estimated model from the combined flight test

Mode	Eigenvalue	Damping ratio ζ	Frequency ω_n [rad/s]	Time to half amplitude $T_{\frac{1}{2}}$ [s]
Roll subsidence	-8.21 + 0 i	1	8.21	0.122
Dutch roll	-1.56 $\pm 7.08 i$	0.215	7.25	0.643
Spiral	0.432 + 0 i	-1	0.431	-2.32

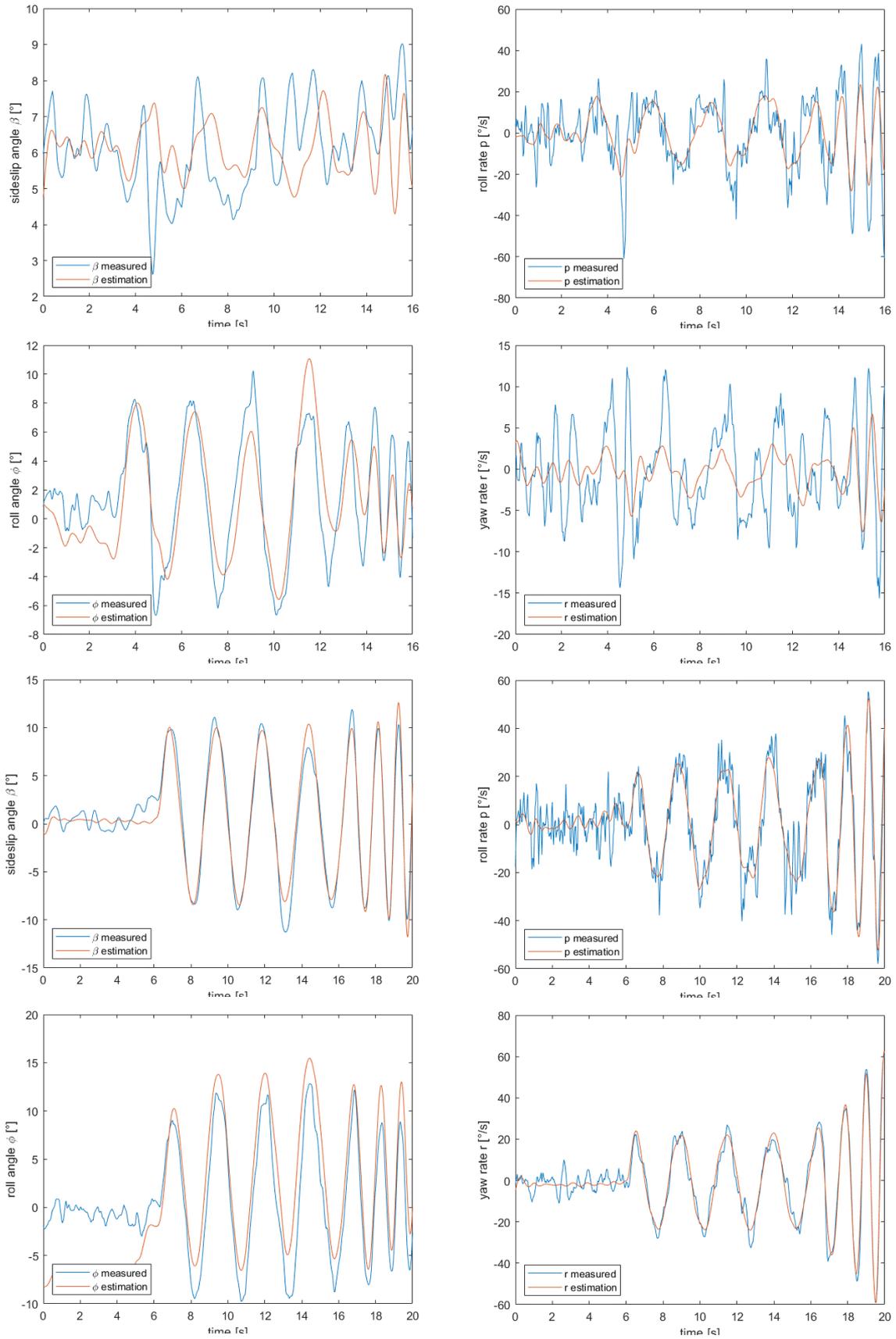


Figure 5.12 State space model created from the combined flight test estimation of the different states compared to the measured states in both aileron and rudder sweep flight tests. The top four figures show the aileron sweep. The lower 4 figures show the rudder sweep

5.3 Model of aircraft dynamics using vortex lattice method

As a separate model, the lateral-directional dynamics of the Skysurfer X8 are determined using a vortex lattice method on XFLR5. To acquire the aerodynamic characteristics and to translate them to the actual dynamics of the aircraft, the moments of inertia, mass and center of gravity have to be known. For this a Catia V5 model of the Skysurfer X8 including all of the components is created. The aircraft outer model is taken from a point cloud file³² to a solid. The adaptation of the point cloud to a solid causes the model to become slightly nonsymmetric around the X - Z plane. Once the aircraft solid is created, the interior of the fuselage is hollowed out to leave the correct wall thickness.

For each component that was added to the aircraft, like the battery, motors, Pixhawk and telemetry module, the shape was simplified to either a cylinder or a cuboid. Once the component is created the material of the component is changed as to provide the density required for the model of the component to have the correct weight. All components are then put together into a single product which allows to obtain the center of gravity location and moments of inertia of the aircraft around the center of gravity location. The weights and locations with respect to the nose of the aircraft of the center of gravity of the separate components are provided in Appendix C.

As can be observed in Figure 5.13, which shows the model of the test aircraft with its different components, the axis system used in the Catia V5 model is different from the axis system used in the aerodynamic analysis of this thesis. The Y axis in the Catia V5 model points in the positive X direction. The X axis in the Catia V5 model points in the positive Y direction. The Z axis in the Catia V5 model points in the negative Z direction.

Figure 5.14 shows the measurements of inertia in the Catia V5 model. It must be noted that the origin used for the center of gravity location is not the nose of the aircraft. Also, due to the slight non symmetry around the X - Z plane, created by the approximation of the aircraft solid, the values for I_{XY} and I_{YZ} do not equal 0, but are more than 4 times smaller than the value for I_{XZ} . Table 5.4 shows the characteristics for the moments of inertia and mass in the reference frame used in this thesis.



Figure 5.13 Rendering of the test aircraft including all components in Catia V5

³² <https://free3d.com/3d-model/hobbyking-bixler-sky-surfer-89447.html>

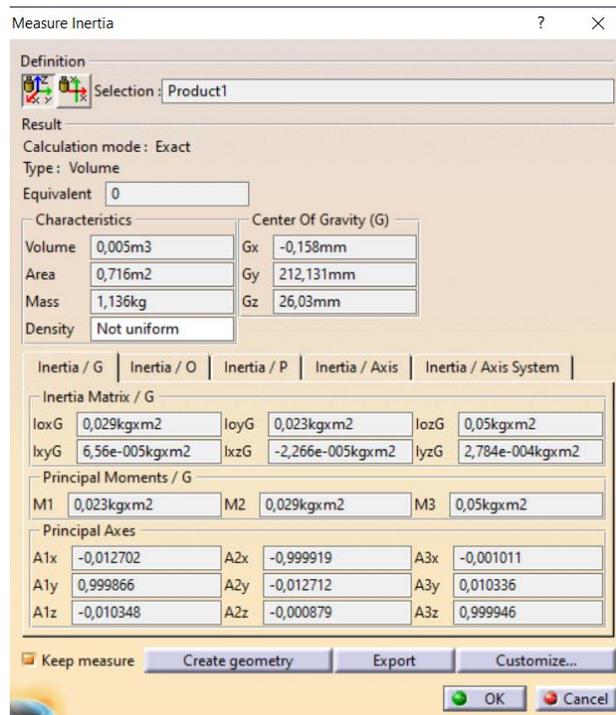


Figure 5.14 Inertia of the test aircraft as measured by Catia V5 in the Catia V5 reference frame

Table 5.4 Inertial characteristics of the test aircraft in the aerodynamic reference frame

Mass	1.136kg
I_{XX}	$0.023kg \cdot m^2$
I_{YY}	$0.029kg \cdot m^2$
I_{ZZ}	$0.05kg \cdot m^2$
I_{XZ}	$-2.784 \cdot 10^{-4}kg \cdot m^2$

Given the moments of inertia found, XFLR5 is used to obtain a model of the aircraft lateral-directional dynamics. The model of the Skysurfer X8 in XFLR5 is shown in Figure 5.15. As can be seen, the fuselage has not been modelled. The fuselage is not modelled because of the fact that this is also not done in the analysis example provided by XFLR5 themselves, where it is shown that the analysis performed by XFLR5 without the fuselage provides a reasonably accurate estimation of the Dutch roll mode as measured in flight³³. Furthermore the fuselage interactions and aerodynamic effects are not modelled accurately in XFLR5. The airfoils used on the Skysurfer X8 are shown in Table 5.5.

Table 5.5 Airfoils used in the XFLR5 model of the Skysurfer X8 test aircraft

Main wing	NACA 4410
Horizontal tail	NACA 0013
Vertical tail	NACA 0010

³³ http://www.xflr5.tech/docs/XFLR5_Mode_Measurements.pdf

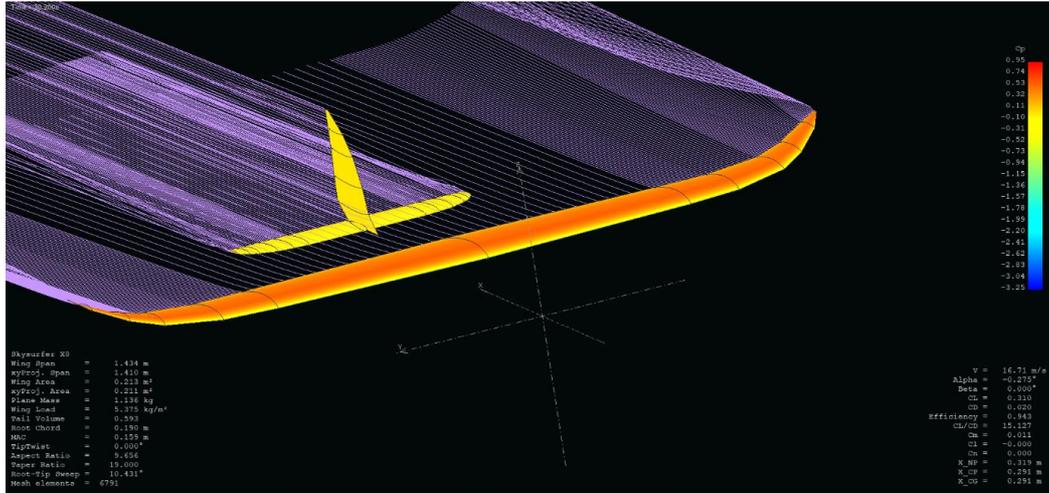


Figure 5.15 Model of the Skysurfer X8 without pitot tube mount used for the stability analysis

The state space system for lateral directional stability that is created by the XFLR5 analysis is different from the one that is used in this thesis. The XFLR5 model has the sideways velocity v , yaw rate r , roll rate p and roll angle ϕ . Furthermore the control surfaces are not modeled, which leaves out the control matrix. The stability analysis of the Skysurfer X8 in XFLR5 results in the state matrix as shown in Equation (72) below rounded to two significant digits.

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.66 & -0.31 & -16 & 9.8 \\ -14 & -51 & 10 & 0 \\ 6.4 & 0.22 & -3.6 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} \quad (72)$$

The lateral stability derivatives rounded to two significant digits are shown in Table 5.6. The eigenvalues, damping ratio, frequency and time to half amplitude of the different eigenmodes, rounded to two significant digits, are shown in Table 5.7. It can be seen that the only estimated mode that is not stable is the spiral mode, which is only slightly unstable.

Table 5.6 Lateral stability derivatives of the Skysurfer X8 as found by the XFLR5 analysis

Y_v	-0.75	N_v	0.32	C_{l_β}	-0.10
Y_p	-0.36	N_p	0.0033	C_{l_p}	-0.55
Y_r	0.41	N_r	-0.18	C_{l_r}	0.11
L_v	-0.31	C_{Y_β}	-0.35	C_{n_β}	0.10
L_p	-1.2	C_{Y_p}	-0.23	C_{n_p}	0.0015
L_r	0.23	C_{Y_r}	0.27	C_{n_r}	-0.082

Table 5.7 Eigenvalues found in the XFLR5 stability model with their resulting damping, natural frequency and time to half amplitude

Mode	Eigenvalue	Damping ratio ζ	Natural Frequency ω_n [rad/s]	Time to half amplitude $T_{\frac{1}{2}}$ [s]
Roll subsidence	-51	1.0	51	0.020
Dutch roll	$-2.0 \pm 10 i$	0.20	10	0.49
Spiral	0.027	-1.0	0.027	-38

5.4 Comparison between flight test and vortex lattice method and military standards

When creating the state space system, the assumption is made that the velocity remains constant. As can be seen in Figure 5.1, this assumption is close to reality for the flight test portions in which is measured. When looking at the controller inputs shown in Figure 5.2, the signals are nice and smooth for the frequency sweeps. The frequency sweep continues still after the point that it is cut off, however, the lines become less smooth. This is due to the recording rate of the control input at only 5Hz. The servo input signals, shown in Figure 5.3, are also recorded at only 5Hz. The transmitter sends a signal at 50Hz to the flight controller, which sends its own signal at 50Hz to the servos. The fact that only 1 in 10 signals is recorded results in non-smooth behavior. It can be seen in Figure 5.3 that the rudder sweep is still relatively smooth, making it likely that the non-recorded points also follow smoothly. The aileron sweep however shows a non-smooth behavior already at the lowest frequency of the frequency sweep. This means that the points in between are likely also non-smooth, making the assumption that is made in the creation of the state space models that the control signal in between points follows a straight line false. The result of which is that the state space models that are estimated from the aileron sweep are inherently likely to not show good behavior given that the control input is not accurate. For this reason, the following part will only talk about the rudder sweep state space model.

Figure 5.4 shows the yaw angle of the aircraft during the tests. It can be seen that the aircraft does not keep a straight line, this is not a problem given that the yaw angle itself does not affect the lateral-directional aircraft dynamics. Figure 5.5 shows the barometric altitude during the test flights. It can be seen that the height is below 0, given that the flight is performed above the beach, this indicates a high-pressure area, resulting in relatively calm weather as can be seen by the wind speed estimate shown in Figure 5.6. To allow for good functioning of the Pixhawk 4 flight controller, the vibrations should be low. As can be seen in Figure 5.7, during the flight the vibrations are very low and only during the touchdown the vibrations end up in an area which limits the accuracy of the results. Figure 5.8 shows the acceleration power spectral density during the flight. It can be observed that the accelerations are low during the flight, also the vibrations caused by the engines are way less pronounced compared to flight tests performed with a different aircraft that had its propeller mounted at the nose. Figure 5.9 shows the percentage of unloaded servo speed that is required to follow the measured servo inputs. It can be seen that the percentage is generally quite low. Only the rudder servo speed in the rudder sweep from around 17 seconds becomes relatively high, more than 25% of the maximum speed. It is decided to keep this part of the data given the relatively low expected torque on the servo given the linkage setup as well as the very long (partially sleeved) control linkage.

As discussed before the results of the aileron sweep and therefore the combined rudder and aileron sweep are less accurate due to the low recording frequency of the servo signals. As can be seen in Figure 5.11 and Figure 5.12, the state space system that is estimated is not good at predicting the actual data, especially for the aileron case. For the combined case including the aileron sweep causes the optimal result to also be less accurate for the rudder sweep.

Figure 5.10 shows results of the optimization of the rudder sweep state space model. It can be seen that all values are quite to very well estimated. From the eigenvalue analysis of the system it is found that the spiral mode is unstable, the Dutch roll is moderately damped and the roll subsidence mode is highly damped as shown in Table 5.1.

Comparing the results of the eigenvalue analysis of the rudder sweep state space model in Table 5.1 to the results obtained from the vortex lattice method in Table 5.7, it can be seen that the same behavior is predicted, namely a unstable spiral mode, a moderately damped Dutch roll mode and a highly damped roll subsidence mode. For the roll subsidence mode, the estimated damping of the state space model determined from flight tests is higher. For the Dutch roll mode, the state space model determined from flight tests estimates a lower frequency and a higher damping of the Dutch roll motion compared to the vortex lattice method. The spiral mode as estimated by the state space model determined from flight tests is more unstable than that estimated by the vortex lattice method. The vortex lattice method used has some aspects that are different from reality. First the fuselage, including the pitot mount is not modeled in the vortex lattice method. Secondly the rotating mass of and the wake created by the propellers is not modeled. Thirdly the actual weight distribution estimated is likely not exactly the same, which can cause quite significant differences in the results. Lastly the vortex lattice method assumes a rigid body, whilst the deflections caused by the lift of the aircraft are small mainly due to the carbon fiber reinforced single spar in the wing, the aileron control inputs can cause a twist of the wing around the spar resulting in different behavior. During a separate flight test with manual mode engaged, it is observed by the pilot that the spiral mode is quite unstable with a time to double in the order of a few seconds, not tens of seconds, making content adjustments necessary which gives further evidence that the spiral mode estimated from the flight test is more correct for the aircraft flown than the one estimated from the vortex lattice method.

All in all, the vortex lattice method used has shown the characteristics of the aircraft, but it is believed that the estimation of the characteristics from the flight test are more accurate. For this reason, combined with the main objective of determining the lateral-directional stability characteristics from a flight test, only the characteristics determined from the state space model determined from flight tests are compared to the military guidelines.

Analyzing the state space model created from the rudder frequency sweep, the characteristics as described in the MIL-HDBK-1797 [12] can be determined and examined with respect to a Class I aircraft in flight phase category B, given the analysis is on the cruise flight of a small aircraft.

The first aspect described is the roll subsidence mode. To determine the time constant from Equation (21), the time to half amplitude is used as shown in Equation (73) below. T_R is found to be 0.0105, which is a level 1, or satisfactory behavior.

$$0.5 \cdot p_{\infty} = p_{\infty} \cdot \left(1 - e^{-\frac{T_1}{T_R}} \right) = p_{\infty} \cdot \left(1 - e^{-\frac{0.00730}{T_R}} \right) \quad (73)$$

The second aspect that is described in the MIL-HDBK-1797 [12] is the spiral mode time to double. The estimated time to double is 5.18s, which correlates, according to Table 2.5, to a level 3, or controllable behavior. When looking at the aircraft it is seen that the wings have a slight anhedral due to the wing mounting mechanism. This is likely the reason for the worse flight behavior.

The third aspect described in the MIL-HDBK-1797 [12] is the roll performance. To obtain the actual roll performance of the aircraft, the state space model is run, starting with all 0 starting vector and the maximum control deflection PWM signal that is achieved by the aircraft, or 2000us on both the aileron and rudder to reduce the sideslip that retards the roll rate as allowed by the MIL-HDBK-1797. The result of the roll angle for the first 2 seconds is shown in Figure 5.16. It can be seen that 30 degrees is achieved in 0.744 seconds, 45 degrees in 1.35 seconds and 60 degrees in 1.90 seconds. Given that the aircraft is class I and category B due to the small aircraft in cruise flight condition, this performance correlates, according to Table 2.6, to level 2, or acceptable behavior. Given the high aspect ratio large wings, with relatively small control surfaces, having one level lower flight characteristics may be expected.

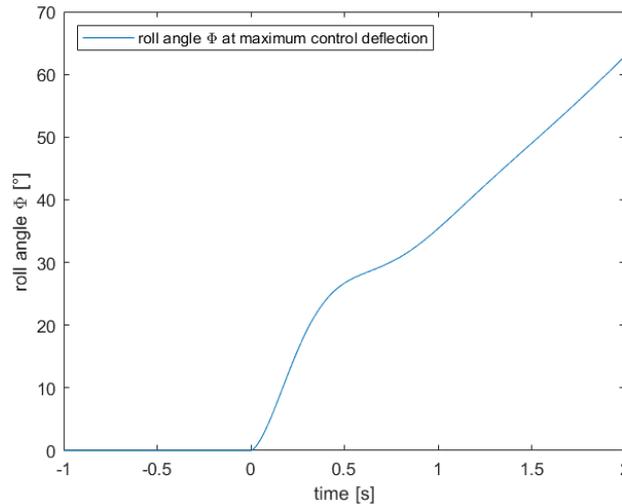


Figure 5.16 Roll angle as function of time from the estimated state space system from the rudder sweep using maximum aileron and rudder inputs to achieve optimal roll rate

The last aspect described in the MIL-HDBK-1797 [12] is the Dutch roll performance. When comparing the results of the eigenvalue analysis of the state space system created from the rudder sweep to the once required to achieve according Table 2.9, it can be concluded that the Dutch roll performance is of level 1, or satisfactory behavior.

6 Validation

To validate the model that is created, the model is used to determine the behavior of the aircraft for a different set of maneuvers, namely doublets. The result of the analysis is shown in Figure 6.1. The control signals that are sent to the servos are shown in Figure 6.2. Given the short and very sharp inputs as shown in Figure 6.2, the actual inputs will be slightly different from those actually recorded, given the low 5Hz recording rate. Given the actual 0.5s up signal and down signal, a 5Hz recording means one of either up or down signals can have 1 more measurement point compared to the other, resulting in longer control inputs than are actually provided. As can be seen in Figure 6.3, the required servo speed to match the signal of the rudder (at 5Hz) is already too high to match the results at some points. This does make the signal that is estimated using the 5Hz data more accurate than if the results at 50Hz could have been taken given that it matches the actual servo deflection more accurately, however the actual deflection of the servo will be even slower than the one modeled with the data. This may cause the slightly higher maximum roll angle yaw rate and roll rate achieved, slightly lower sideslip angle achieved. Apart from this slight mismatch, the data is still modelled fairly accurately, showing that the model that was created from the rudder sweep flight data is capable of providing an accurate result for different maneuvers. The initial conditions for this test have been determined according to the cost used for the rudder sweep analysis shown in Equation (66). The total cost is determined to be approximately $7.32 \cdot 10^9$ with a total of 811 points used in the estimation. Given the fact that the spiral mode is quite unstable, having the roll angle estimation not diverge provides further validation of the model.

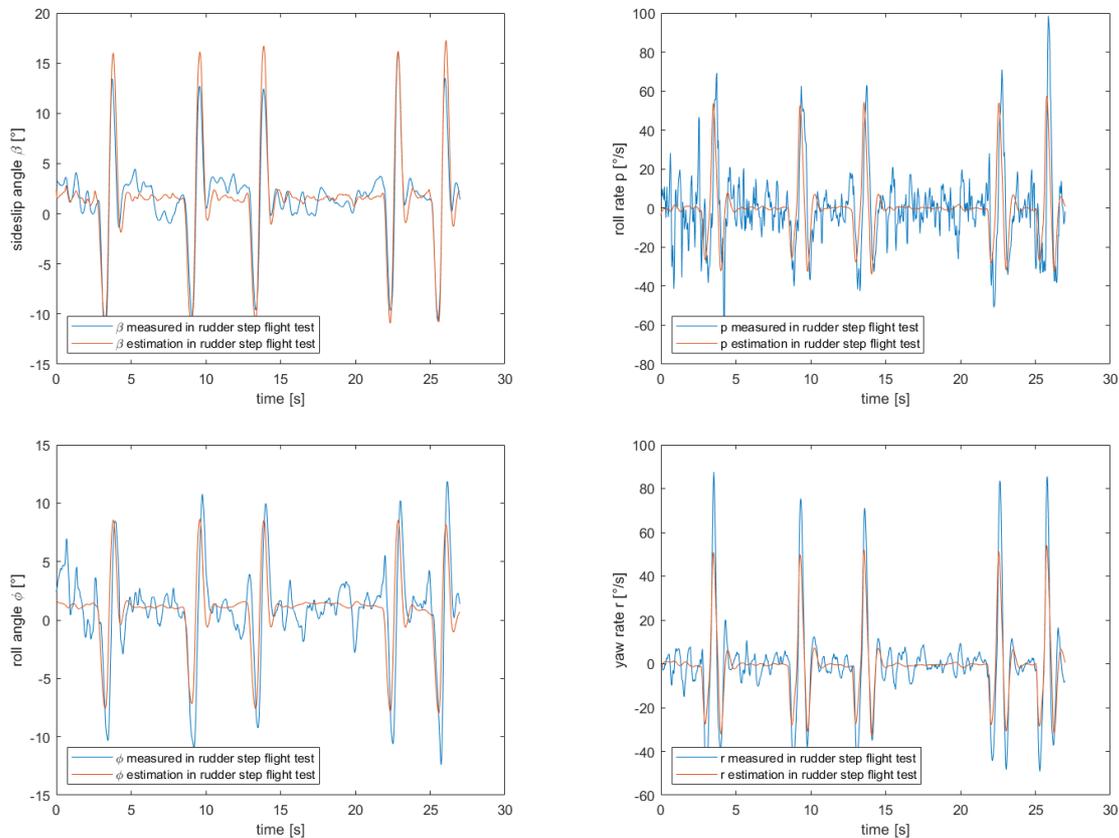


Figure 6.1 Different states compared to the measured states in the rudder step flight test with the estimation using the state space model optimized using the rudder sweep data set

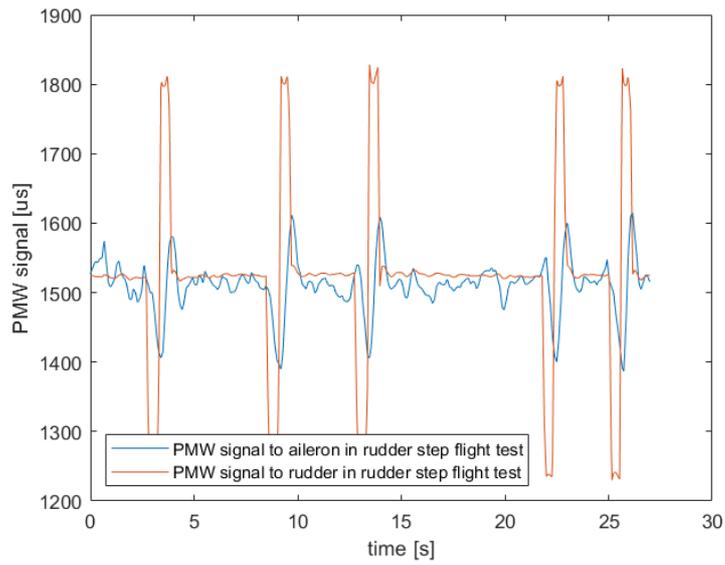


Figure 6.2 PWM signals sent to servos in the rudder step flight test

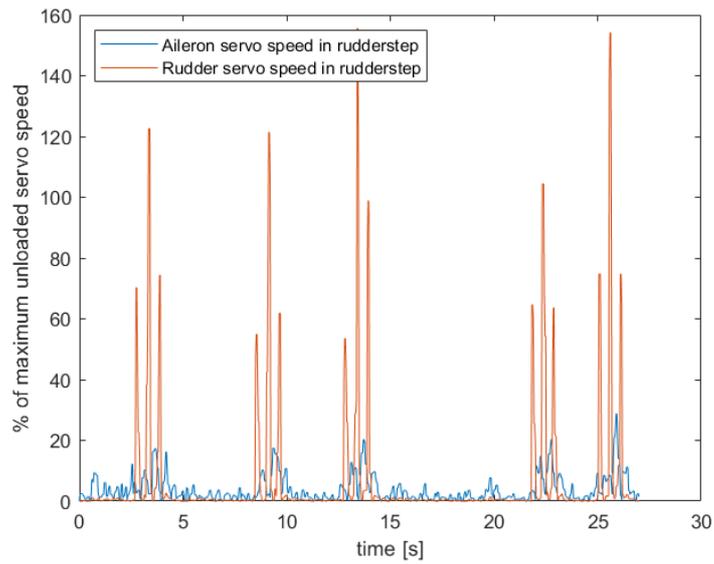


Figure 6.3 Percentage of maximum unloaded servo speed in the rudder step flight test

7 Conclusions and Recommendations

7.1 Conclusions

The thesis has shown a method that is capable of finding a state space model and estimating the lateral-directional stability and controllability characteristics of the Skysurfer X8 using a single measurement flight which requires just a limited number of maneuvers, namely a frequency sweep on the rudder and a frequency sweep on the ailerons. To validate the results doublet inputs on the ailerons and rudder are given. The flight instrumentation and control systems have been designed and built to allow for measuring the data required to create a state space model.

The first main research question asks what the lateral-directional controllability and stability characteristics are. To answer this a linearized state space model is created to show the lateral-directional stability and controllability characteristics. To create the state space model, first a method has to be created to determine the state space model from flight tests. To analyze the created model, it is compared to the measurements of the test flight and compared to a model created from a vortex lattice method.

The linearized state space model for lateral-directional stability characteristics of the aircraft can be created by putting a Pixhawk 4 flight controller including its sensors onboard the aircraft during the flight to measure and record the attitude and control inputs to the aircraft during the flight. During the flight the flight controller functions as a PID controller to allow the aircraft to remain stable. From the ground a frequency sweep on the rudder and aileron controls are given in separate maneuvers performed at the same speed and altitude. Also a number of doublets on the rudder and aileron controls are given as a validation for the model. Given the results of the test flight, a state space system model is estimated using an optimization in which the values of the state space model are unknowns. The only values that are known in the state space system are those that correspond to the compatibility equation that links the roll rate to the roll angle. The state space system that is created by the optimization is then used for the further analysis of the characteristics of the aircraft by evaluating the eigenvalues of the state matrix.

Comparing the linearized state space model of the aircraft determined from the flight test to the measurements taken during the test flight, it can be seen that the results match quite nicely to the measured sideslip angle, roll angle, roll rate and yaw rate. To give an objective description of the accuracy of the model compared to the measurements, the square of the difference between the measured sideslip and estimated sideslip at all measured points is multiplied by 1,000,000 and added to the square of the difference between the measured roll angle and estimated roll angle at all measured points multiplied by 1,000,000. To this the difference between the measured roll rate and estimated roll rate at all measured points multiplied by 100 and the difference between the measured yaw rate and estimated yaw rate at all measured points is multiplied by 10,000 are added. When combining all 601 points used in the optimization, a value of $4.15 \cdot 10^9$ is obtained. For a validation of the model, the output of the model is compared to the measurements in a flight test with different control inputs. Using the same objective description of the accuracy of the model, for 811 points used as measurement points, a value of $7.32 \cdot 10^9$ is obtained. Also when comparing the results to the true results, the estimation is quite accurate. The model that is used is the one created

from a rudder frequency sweep flight test. The model that is created from an aileron frequency sweep flight test is less accurate because the PID controller that is used creates a non-smooth input to the control surfaces, which due to the low frequency of the control signal measurement compared to the frequency at which it is sent to the servos which creates a mismatch between the actual deflection and the one measured and used to create the model. This can be resolved by increasing the frequency at which the signal towards the servos is recorded. This was not done because this specific value could not be set directly and changing it in the main code might lead to memory overload situations leading to a potential crash.

When comparing the lateral-directional stability characteristics of the state space model obtained from flight tests to the state space model obtained from a vortex lattice method, it is seen that the models both predict an unstable spiral, moderately damped Dutch roll mode and highly damped roll subsidence mode. It is determined that the vortex lattice method used has a number of aspects that are not modeled, namely a geometric modelling difference in which the fuselage, including the pitot mount, are not modeled in the vortex lattice method. Secondly the rotating mass of and the wake created by the propellers is not modeled. Thirdly the actual weight distribution estimated is likely not exactly the same, which can cause quite significant differences in the results. Lastly the vortex lattice method assumes a rigid body, whilst the deflections caused by the lift of the aircraft are small mainly due to the carbon fiber reinforced single spar in the wing, the aileron control inputs can cause a twist of the wing around the spar resulting in different behavior. Given that the model is of the flight test aircraft, the model that is created from the flight test data is deemed better. The time to half amplitude of the Dutch roll of the state space model created from the flight tests is around 62% of the same value for the state space model created from the vortex lattice method. The time to half amplitude of the roll subsidence of the state space model created from the flight tests is around 37% of the same value for the state space model created from the vortex lattice method. The time to double amplitude of the spiral of the state space model created from the flight tests is around 14% of the same value for the state space model created from the vortex lattice method. It can be seen that all modes are experienced slower in the model created from the vortex lattice method compared to the model created from the flight test data. This shows that the XFLR5 program results should be taken with a grain of salt. It has determined if modes are stable or not, but the actual values are not determined accurately.

The second main research question asks if the lateral-directional handling characteristics of the Skysurfer X8 are satisfactory. For this first the characteristics are determined and later compared to the military guidelines.

The lateral-directional handling characteristics according to the determined linearized state space model from the flight test can be subdivided in four different categories. The roll subsidence flight mode, the Dutch roll flight mode, the spiral flight mode and the roll authority. The roll subsidence mode has a time to half amplitude of 0.00730 seconds, meaning it is highly damped. The Dutch roll flight mode has a damping ratio of 0.564 and a natural frequency of 5.86 radians per second. The time to half amplitude of the Dutch roll mode is 0.303 seconds. This shows that the Dutch roll is moderately damped compared to the roll subsidence mode. The spiral flight mode has a time to half amplitude of negative 5.18 seconds, meaning that the spiral mode is slightly unstable. When looking at the roll performance of the aircraft, we find that a roll angle of 30 degrees is achieved in 0.744 seconds, 45 degrees is achieved in 1.35 seconds and 60 degrees is achieved in 1.90 seconds.

The characteristics that are shown above are compared to the military specifications MIL-HDBK-1797 [12]. Using an aircraft class I, flight phase category B to find the controllability level for the model that is created from the flight tests, it is shown that the roll subsidence and Dutch roll are satisfactory at a level 1. The roll performance is acceptable at a level 2. The spiral mode is controllable at a level 3. From a pilot perspective the aircraft is relatively easy to fly with respect to the lateral-directional dynamics. The only aspect that makes it more demanding is that the aircraft is relatively unstable in spiral requiring constant attention in the case it is flown manually. Besides this the aircraft does not possess fully optimal roll performance. None of the characteristics are worse than level 3, meaning the aircraft can be flown by a pilot manually without any control augmentation system for the characteristics that are described.

7.2 Recommendations.

The first recommendation that is made is to create a PX4 firmware file that locks the recording rates for the control signals to the servos at a higher rate of about 50Hz. By doing this the model that is estimated will more closely follow reality.

In the case of larger RC models that are tested, it is recommended to use a separate control surface angle recorder that measures the actual control surface deflection. This would make the model that is created more accurate since the model that is created using the PWM signal that is sent to the servo assumes the servo moves to the new point linearly, whilst this new point is not yet sent to the servo. This can also be adjusted by moving every timestep of the PMW signal one forward. This is not done in this thesis due to the fact that the PMW signal is only recorded at 5Hz. Furthermore the model assumes the servo can move at the speed that is set under flight conditions which is not guaranteed. Also it is assumed that the deflection of the control surface is linear with respect to the PWM signal to the servo which is not always fully correct. If this is not done, it is recommended to create a model of the servo deflection that takes into account the dynamics of the servo which would depend on the airspeed and deflection.

In the case of a larger model it is recommended to also use a measurement device that is capable of measuring the sideslip angle directly, allowing for a better model than the one used in this thesis that assumes the wind estimate of the Pixhawk is correct. Furthermore given the recording frequency of the of only 5Hz of the GPS speed and windspeed estimate the sideslip angle estimate is slightly less accurate. It is also recommended to increase the recording frequency of these parameters to around 50Hz.

It is recommended that in the case that the firmware is changed a flight test is performed with a cheaper model to make sure that the CPU usage and RAM usage of the flight controller remain low enough such that no faults will occur on a more expensive model.

Depending on the use case, it might be recommended to increase the state space system from a 4 by 4 system to a larger system to take into account more non-linear behavior. This would only be of interest in the case that the outputs are those that are of interest, not the easy reconstruction of the different modes from the eigenvalues of the system.

It is recommended to use a higher speed servo in the case the very high damping of the roll subsidence mode must be determined precisely. Given the servo speed of $600^\circ/\text{s}$ which translates to a frequency of 20.9 rad/s (assuming a full 180° rotation), this allows to estimate the Dutch roll and spiral mode accurately, especially since only a limited rotation of maximally 80% is used. Given the high damping of the roll subsidence mode, having some error there is not affecting the behavior of the aircraft significantly and is therefore not always important to estimate accurately.

For future research it is recommended to not only create a single state space model for one airspeed, but to create multiple state space models for different airspeeds. This would give more knowledge about the handling characteristics in different conditions.

When performing flight tests with a light model, any sort of turbulence or windshear will influence the aircraft significantly. In order to minimize these disturbances, it is recommended to perform the flight in the early morning at a day with low winds. In choosing the location where the flight test is performed, it is recommended to choose a location with relatively flat surroundings without tall objects nearby. This limits disturbances in the air.

It is recommended to perform a number of flight tests in manual control mode in which the Dutch roll and spiral eigenmodes are excited separately. When leaving controls free for some time after the mode is excited, the measurements can be used to determine the damping ratio, frequency and time to half amplitude. These can then be used to validate the state space model of the aircraft.

For future studies it would be interesting to investigate the accuracy of the different estimations by looking at the differential of the cost function in the direction of the different variables. From this the more certain values can be found. Also the accuracy of the Pixhawk should be investigated in a natural environment. Given its internal bias corrections, the accuracy of the Pixhawk can not be tested in something like a wind tunnel environment.

8 Bibliography

- [1] Civil Aviation Authority, "Information on aviation's environmental impact CAP 1524," Civil Aviation Authority, Gatwick Airport South, 2017.
- [2] F. Faggiano, *Aerodynamic Design Optimization of a Flying V Aircraft*, Delft: Delft University of Technology, Aerospace Engineering, 2016.
- [3] P. Zhoujie Lyu and J. R. Martins, "Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft," *Journal of Aircraft*, vol. 51, pp. 1604-1617, 2014.
- [4] J. R. Chambers, *Modeling Flight: The Role of Dynamically Scaled Free-Flight Models in Support of NASA's Aerospace Programs*, National Aeronautics and Space Administration, 2009.
- [5] D. E. Cox, K. Cunningham and T. Jordan, "Subscale Flight Testing for Aircraft Loss of Control: Accomplishments and Future Directions," in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, 2012.
- [6] M. J. Hinton and C. N. Eastlake, "The Construction and Flight Testing of a Scaled, Remotely-Piloted, Flight-Test Vehicle," Cessna Aircraft Company/Embry-Riddle Aeronautical University.
- [7] J. García, J. M. Molina and J. Trincado, "Real evaluation for designing sensor fusion in UAV platforms," *Information fusion*, vol. 63, pp. 136-152, November 2020.
- [8] J. Teplitz, G. Kayten and P. Cancro, "Tests of a 1/7-scale semispan model of the XB-35 airplane in the Langley 19-foot pressure tunnel," National Advisory Committee for Aeronautics Langley memorial aeronautical laboratory, Langley, 1946.
- [9] A. Irving, "Evaluation of a technique for determining airplane aileron effectiveness and roll rate by using an aeroelastically scaled model," Langley Research Center, Hampton, 1969.
- [10] Y. Gibbs, *NASA Armstrong fact sheet: X-48 hybrid/blended wing body*, NASA, 2017.
- [11] T. Risch, G. Cosentino, C. Regan, M. Kisska and N. Princen, "X-48B flight-test progress overview," Orlando, 2009.
- [12] Department of Defence, *Handbook flying qualities of piloted aircraft*, 1997.
- [13] S. Juffermans, "The longitudinal handling characteristics of the Skysurfer X8," Delft University of Technology, Delft, 2021.
- [14] P. Alken, E. Thébault, C. Beggan and et al. (62 more authors), "International Geomagnetic Reference Field: the thirteenth generation," *Earth, Planets and Space*, Vols. 73, Article number 49, 2021.
- [15] J. Diebel, *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*, Stanford, California: Stanford University, 2006.
- [16] A. Sobron, D. Lundström, R. Larsson, P. Krus and C. Jouannet, "Methods for efficient flight testing and modelling of remotely piloted aircraft within visual line-of-sight," in *31st Congress of the International Council of the Aeronautical Sciences*, Belo Horizonte, 2018.
- [17] H. Kamp, *Regeling gebruik van frequentieruimte zonder vergunning en zonder meldingsplicht 2015*, 's-Gravenhage: De minister van Economische zaken, 2015.

- [18] D. Hunsaker, "A Numerical Blade Element Approach to Estimating Propeller Flowfields," Brigham Young University, Provo, 2007.
- [19] J. Mulder, W. van Staveren, J. van der Vaart, E. de Weerd, C. de Visser, A. in 't Veld and E. Mooij, Lecture notes AE3202 Flight Dynamics, Delft: Faculty of Aerospace Engineering, 2013.
- [20] J. Nocedal and S. J. Wright, Numerical Optimization, Evanston, Madison: Springer, 2006.
- [21] J. Katz and A. Plotkin, Low-Speed Aerodynamics Second Edition, Cambridge: Cambridge university press, 2001.

Appendix A Redundant power supply module

To improve the reliability of the power delivery to the separate components and to limit the required wire size, a redundant variable power supply module is developed. This module is not used onboard the aircraft used for the test flights, but is incorporated in the design of the Flying-V from TU Delft. The list of components of the PCB is shown in Table A.1.

Table A.2 shows the SMBus address that the PAC1932T-I/JQ chip sends the measured voltage over the 0.1Ω 1% sense resistor R4 to, depending on the chosen resistor R3.

The module is designed to be supplied by two 2S to 6S LiPo batteries. The output voltage delivered by the module can be changed smoothly by rotating the variable resistor R5 up to a maximum of 12V or the battery voltage minus some losses due to the diodes, whichever is lower. The output voltage is created by the XTPS56637RPAT chip by switching on and off at a frequency of 500kHz where the portion of time that the switch is on is determined to maintain the voltage that is set. The output is smoothed by a number of capacitors and a hall effect inductor to provide a smooth DC voltage to the servo. In this way the servos can be supplied by the desired voltage to increase the power of the servo, but limit the heat buildup. To further aid in monitoring the servos, the current running through the servos is measured. This is done by measuring the voltage over the 0.1Ω 1% sense resistor R4 and using Equation (74) below to find the current drawn by the servo.

$$I_{supplied} = \frac{U_{sens}}{R_{sens}} \quad (74)$$

The modules have a reverse current protection build in which ensures that power can not be transferred to a failing battery that has lower voltage. Two wires that run to the positives of two different batteries can be soldered to the two pads marked with the positive sign. The servo power wires can be soldered to the Out + and Out – pads. To avoid ground loops and retain redundancy, a single negative pad is included with the intention to have a single wire going from the negative of the first battery through all negative pads to the negative of the other battery. In this way, if the wire brakes at any point, all boards will still be connected to a negative of a battery.

The design of the redundant power supply module consists of three layers. The top and bottom layer are shown in Figure A.1 and Figure A.2. The middle layer is a ground plain that allows for better heat dissipation and prevents electromagnetic interference. The design uses all surface mount components with a minimum package size of 0603 (.6mm x .3mm), to allow for relatively easy manufacturing whilst creating the smallest formfactor and weight. The total board is 56 mm by 36 mm. The board can be mounted securely using its four 2mm holes.

Table A.1 Components used on redundant variable power supply module, where D are diodes, C capacitors, R resistors, H inductors, U varying chips and M connectors

Silkscreen name	Quantity	Component name
D1	4	SBRT15U50SP5-13
C1	3	C2012X5R1H475K125AB
C2	4	CC0603JPX7R9BB104
C3	1	C0603C106M8PACTU
C4	2	GRT32EC81C476KE13L
C5	1	C1206X105J1RACTU
C6	1	CBR06C101F5GAC
R1	1	CRCW0603100KFKEAC
R2	1	CHP0603-FX-1001ELF
R3	1	0603 type resistor, see Table A.2 for the values required
R4	1	WSHM2818R1000FEA
R5	1	TC42X-2-204E
R6	1	CRGP0603F10K
H1	1	XAL7070-562MEB
U1	1	XTPS56637RPAT
U2	1	PAC1932T-I/JQ
U3	1	ZXTR2105FQ-7
M1	1	502386-0370

Table A.2 Address selection resistor value for PAC1932T-I/JQ from Mouser³⁴

Resistor 1% error	SMBus Address
0 Ω (Tie to ground, short)	0010_000 (r/w)
499 Ω	0010_001 (r/w)
806 Ω	0010_010 (r/w)
1.27 k Ω	0010_011 (r/w)
2.05 k Ω	0010_100 (r/w)
3.24 k Ω	0010_101 (r/w)
5.23 k Ω	0010_110 (r/w)
8.45 k Ω	0010_111 (r/w)
13.3 k Ω	0011_000 (r/w)
21.5 k Ω	0011_001 (r/w)
34 k Ω	0011_010 (r/w)
54.9 k Ω	0011_011 (r/w)
88.7 k Ω	0011_100 (r/w)
140 k Ω	0011_101 (r/w)
226 k Ω	0011_110 (r/w)
Tie to positive (leave open)	0011_111 (r/w)

³⁴ <https://nl.mouser.com/datasheet/2/268/PAC1931-Family-Data-Sheet-DS20005850E-1519053.pdf>

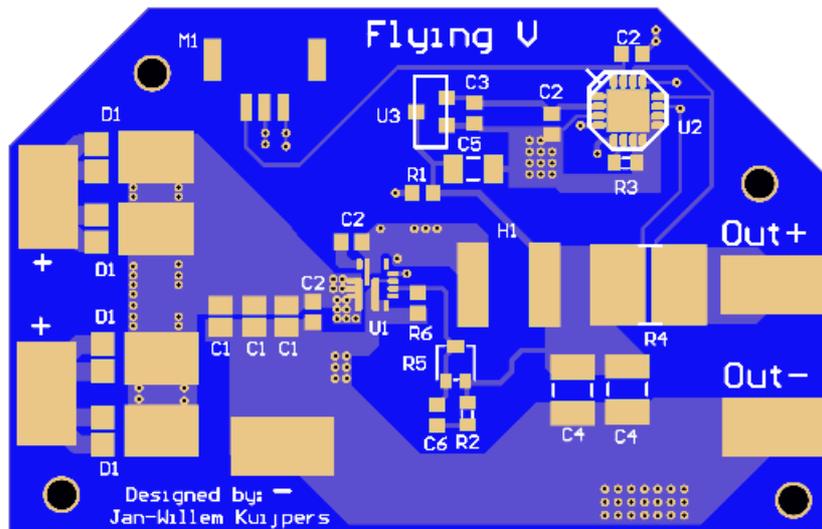


Figure A.1 Front view of the redundant power supply PCB

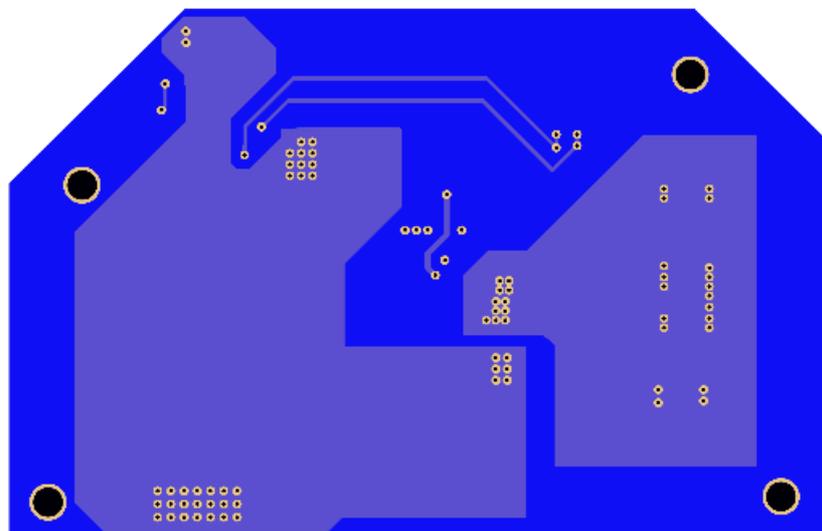


Figure A.2 Back view of the redundant power supply PCB

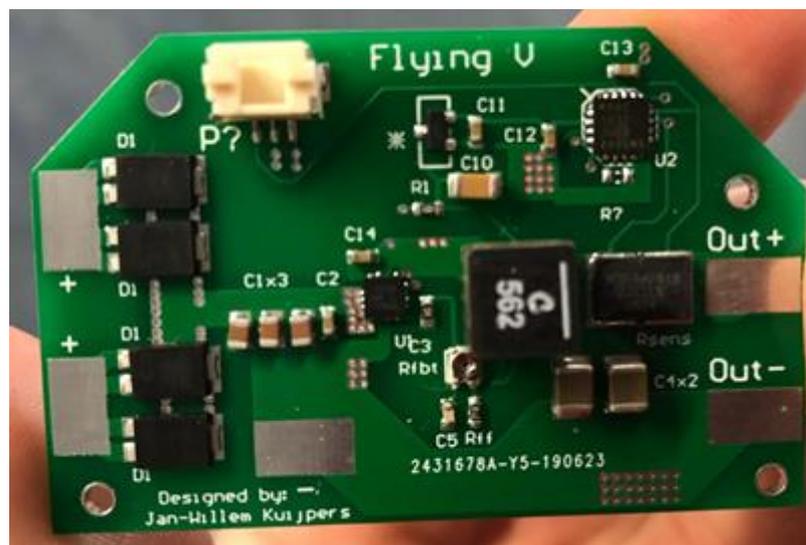


Figure A.3 Picture of one of the prototype boards

Appendix B Mixer and airframe PX4

Mixer file: The following code is equal to the one used by Juffermans [13]

Location: Firmware/ROMFS/px4fmu_common/mixers/SKYSURFER.main.mix

This file defines mixers suitable for controlling a fixed wing aircraft with aileron, rudder, elevator, throttle, gear, flaps controls. The configuration assumes the aileron servo(s) are connected to output 0, the elevator to output 1, the throttle to output 2 and the rudder to output 3.

Inputs to the mixer come from channel group 0 (vehicle attitude), channels 0 (roll), 1 (pitch), 2 (thrust), 3 (yaw), 4 (flaps), 7 (landing gear)

CH1 CH2: Aileron mixer

Two scalars total (output, roll).

This mixer assumes that the aileron servos are set up correctly mechanically; depending on the actual configuration it may be necessary to reverse the scaling factors (to reverse the servo movement) and adjust the offset, scaling and endpoints to suit.

As there is only one output, if using two servos adjustments to compensate for differences between the servos must be made mechanically. To obtain the correct motion using a Y cable, the servos can be positioned reversed from one another.

M: 1

S: 0 0 -10000 -10000 0 -10000 10000

M: 1

S: 0 0 -10000 -10000 0 -10000 10000

CH3: Elevator mixer

Two scalars total (output, roll).

This mixer assumes that the elevator servo is set up correctly mechanically; depending on the actual configuration it may be necessary to reverse the scaling factors (to reverse the servo movement) and adjust the offset, scaling and endpoints to suit.

M: 1

S: 0 1 -10000 -10000 0 -10000 10000

CH4 CH5: Motor speed mixer

Two scalars total (output, thrust).

This mixer generates a full-range output (-1 to 1) from an input in the (0 - 1) range. Inputs below zero are treated as zero.

M: 1

S: 0 3 0 20000 -10000 -10000 10000

M: 1

S: 0 3 0 20000 -10000 -10000 10000

CH6: Rudder mixer

Two scalers total (output, yaw).

This mixer assumes that the rudder servo is set up correctly mechanically; depending on the actual configuration it may be necessary to reverse the scaling factors (to reverse the servo movement) and adjust the offset, scaling and endpoints to suit.

M: 1

S: 0 2 10000 10000 0 -10000 10000

Change in mixer overview file:

Location: ROMFS/px4fmu_common/mixers/CMakeLists.txt

To make the mixer file discoverable add in

“ROMFS/px4fmu_common/mixers/CMakeLists.txt” Between “px4_add_romfs_files(“ and the final closing bracket the following in a new line: SKYSURFER.main.mix

Airframe file: : The following code is equal to the one used by Juffermans [13]

Location: Firmware/ROMFS/px4fmu_common/init.d/airframes/2103_skysurfer

```
#!/bin/sh
#
# @name Skysurfer
#
# @type Standard Plane
# @class Plane
#
# @output MAIN1 aileron
# @output MAIN2 elevator2x
# @output MAIN3 throttle2x
# @output MAIN4 rudder
# @output MAIN5 flaps
# @output MAIN6 gear
#
# @output AUX1 feed-through of RC AUX1 channel
# @output AUX2 feed-through of RC AUX2 channel
# @output AUX3 feed-through of RC AUX3 channel
#
#
sh /etc/init.d/rc.fw_defaults
if [ $AUTOCONF = yes ]
then
param set PWM_AUX_RATE 50
param set PWM_RATE 50
fi
set MIXER SKYSURFER
# Rate must be set by group (see pwm info).
# Throttle is in the same group as servos.
```

Change in Airframe overview file:

Location: ROMFS/px4fmu_common/init.d/airframes/CMakeLists.txt

To make the airframe file show up in the standard plane options of the Pixhawk add in “ROMFS/px4fmu_common/init.d/airframes/CMakeLists.txt” In the correct position under the # [2000,2999] Standard planes” a new line with the following: 2103_skysurfer
If the number 2103 is already used change this number in the airframe file name and use that number in the CMakeLists.txt as well, be sure to use a number between 2000 and 2999 such that the model can be found under the standard planes.

An explanation on how to compile the firmware is provided on the PX4 website³⁵.

³⁵ https://docs.px4.io/master/en/dev_setup/building_px4.html

Appendix C Mass and location of test aircraft parts

Table C.3 Weight and approximate location of center of gravity used for the Catia V5 model with respect to the origin which is at the tip of the nose as measured by Juffermans [13]

Object	Mass [g]	X _{CG} location [mm]	Y _{CG} location [mm]	Z _{CG} location [mm]
Foam part of aircraft	440	-414	0	62
Left aileron servo	15	-331	-347	63
Right aileron servo	15	-331	347	63
Rudder servo	15	-331	-27	11
Elevator servo	15	-331	27	11
Left engine mount	6	-258	-155	63
Right engine mount	6	-258	155	63
Left engine (including propeller)	26	-225	-155	63
Right engine (including propeller)	26	-225	155	63
Left ESC	10	-276	-155	47
Right ESC	10	-276	155	47
Telemetry mount	5	-554	7	47
Telemetry module	34	-547	3	60
Battery	262	-110	0	-12
BEC	15	-252	0	-29
GPS module	33	-180	0	42
Power module	25	-254	-10	-25
Receiver	23	-375	0	11
Pitot tube (including mount and tubes)	20	-3	0	41
Pixhawk 4	33	-210	0	0
Total	1034	-291	0	38

Appendix D Python code

There are 2 parts of the python code. The first one transforms the .ulg data file from the Pixhawk to a number of .csv files. It is advised to rename the .ulg file to something manageable that shows the tests. The second part of the code retrieves the important characteristics for the analysis of the dynamics of the aircraft and creates text files for them. It requires the user to set the range in which the data is collected as well as the interpolated frequency at which the data will be stored to the file.

```
"""
Code created by Jan-Willem Kuijpers for TU Delft thesis
code runs on Python3
requires pyulog, numpy, matplotlib, cvs (also os but this is standard)
if these sidepackages have not been installed execute pip install ... in comma
nd prompt for all sidepackages
"""

from pyulog import ulog2csv
import csv
import matplotlib.pyplot as plt
import os
import numpy as np
from numpy import cos, sin, tan
directoryused="C:\python"
os.chdir(directoryused)

#####
# change name of ulg file and directory to where ulg file is located, in this
# also the output will be provided
# adjust timestamprange in range1 and frequency in frequency
#####

ulgname="aileronrudder" #don't add .ulg to the end
range1=[0,10*10**8]
frequency=300 #Hz
#####

infile=ulgname+".ulg"
ulog2csv.convert_ulog2csv(infile,[],directoryused,',')

os.chdir(directoryused)
```

```

def createlists(name,lists='everything'):
    exit1=[]
    with open(name) as csvfile:
        interlist = csv.reader(csvfile, delimiter=',', quotechar='|')
        for row in interlist:
            exit1.append(row)
    if lists=='everything':
        lists=range(len(exit1))
    #print (exit1[0])
    exit1.pop(0)
    list1=np.array(exit1)
    output=[]
    for i in range (len(lists)):
        time1=list1[:,lists[i]]
        time1= list(time1.astype(float))
        output.append(time1)
    return output

[time2,actuator0]=createlists('aileronrudder_actuator_outputs_0.csv',[0,3])
for i in range(7,14):
    [time2,input0]=createlists('aileronrudder_input_rc_0.csv',[0,i])
    #plt.plot(time2,input0)
#plt.show()

def readdata(filename):
    f=open(filename,"r")
    data=f.readlines()
    f.close()
    datanew=[]
    for line in data:
        adtolist = line.split(",")
        adtolist[-1] = adtolist[-1].split("\n")[0]
        if map(str.isdigit,adtolist[0][0])==[True]:
            for i in range (len(adtolist)):
                adtolist[i]=float(adtolist[i])
            datanew.append(adtolist)
    return datanew

def createfile(listoflists,ulname):
    a=len(listoflists)
    b=ulname+'.txt'

    f=open(b,"w")
    for i in range(len(listoflists[0])):
        for j in range(len(listoflists)-1):

```

```

        f.write(str(listoflists[j][i]))
        f.write(",")
    f.write(str(listoflists[-1][i]))
    f.write("\n")

```

```

f.close()
return

```

```

def createinput(timerange, name, frequency):
    timestep=int(1000000./frequency)
    x=name+'_estimator_status_0.csv'
    y=name+'_actuator_outputs_0.csv'
    z=name+'_airspeed_0.csv'
    a=name+'_vehicle_air_data_0.csv'
    b=name+'_vehicle_attitude_0.csv'
    c=name+'_vehicle_gps_position_0.csv'
    d=name+'_wind_estimate_0.csv'
    e=name+'_input_rc_0.csv'
    x1=readdata(x)
    y1=readdata(y)
    z1=readdata(z)
    a1=readdata(a)
    b1=readdata(b)
    c1=readdata(c)
    d1=readdata(d)
    e1=readdata(e)
    time=[]
    ty1=[]
    ailerony1=[]
    ruddery1=[]
    tz1=[]
    vairindz1=[]
    vairtruez1=[]
    vairtempz1=[]
    ta1=[]
    rhoa1=[]
    baroalta1=[]
    baropressa1=[]
    tb1=[]
    yawspeed=[]
    rollspeed=[]
    tc1=[]
    vnorth=[]
    veast=[]
    td1=[]
    vnorthwind=[]
    veastwind=[]
    te1=[]

```

```

ruddersignal=[]
aileronssignal=[]
throttlesignal=[]
elevatorsignal=[]
for i in range (1,len(y1)):
    ty1.append(y1[i][0])
    ailerony1.append(y1[i][2])
    ruddery1.append(y1[i][7])
for i in range (1,len(z1)):
    tz1.append(z1[i][0])
    vairindz1.append(z1[i][1])
    vairtruez1.append(z1[i][2])
    vairtempz1.append(z1[i][3])
for i in range (1,len(a1)):
    ta1.append(a1[i][0])
    rhoa1.append(a1[i][4])
    baroalta1.append(a1[i][1])
    baropressa1.append(a1[i][3])
for i in range (1,len(b1)):
    tb1.append(b1[i][0])
    yawsspeed.append(b1[i][3])
    rollspeed.append(b1[i][1])
for i in range (1,len(c1)):
    tc1.append(c1[i][0])
    vnorth.append(c1[i][15])
    veast.append(c1[i][16])
for i in range (1,len(d1)):
    td1.append(d1[i][0])
    vnorthwind.append(d1[i][1])
    veastwind.append(d1[i][2])
for i in range (1,len(e1)):
    te1.append(e1[i][0])
    ruddersignal.append(e1[i][9])
    aileronssignal.append(e1[i][10])
    elevatorsignal.append(e1[i][8])
    throttlesignal.append(e1[i][7])
for i in range(int(timerange[0]),int(timerange[1]),timestep):
    time.append((i-timerange[0])/1000000.0)
"""plt.subplot(2,2,1)
plt.plot(te1,ruddersignal)
plt.subplot(2,2,2)
plt.plot(te1,aileronssignal)
plt.subplot(2,2,3)
plt.plot(te1,elevatorsignal)
plt.subplot(2,2,4)
plt.plot(te1,throttlesignal)
plt.show()
"""

```

```

tangles=[]
q0=[]
q1=[]
q2=[]
q3=[]
Vn=[]
Ve=[]
Vd=[]
Alt=[]
yawspedd=[]
for i in range (1,len(b1)):
    tangles.append(b1[i][0])
    q0.append(b1[i][4])
    q1.append(b1[i][5])
    q2.append(b1[i][6])
    q3.append(b1[i][7])
    yawspedd.append(b1[i][3])

tangles=np.asarray(tangles)
q0=np.asarray(q0)
q1=np.asarray(q1)
q2=np.asarray(q2)
q3=np.asarray(q3)
#plt.show()

#plt.plot(tangles,yawspedd)
# plt.show()

roll=np.arctan2(2*(q3*q2+q0*q1),1-2*(q1*q1+q2*q2))
pitch=np.arcsin(2*(q2*q0-q3*q1))
yaw=np.arctan2(2*(q3*q0+q2*q1),-1+2*(q0*q0+q1*q1))

pitch=180/np.pi*pitch
roll=180/np.pi*roll
yaw=180/np.pi*yaw

previous=yaw[0]
for i in range(len(yaw)):
    next=yaw[i]
    if previous-next>180:
        yaw[i]=yaw[i]+360
    if next-previous>180:
        yaw[i]=yaw[i]-360
    previous=yaw[i]
tanglesb=tangles[:]
rollb=roll[:]

```

```

yawb=yaw[:]

tangles=[]
q0=[]
q1=[]
q2=[]
q3=[]
Vn=[]
Ve=[]
Vd=[]
Alt=[]
yawspeedd=[]
for i in range (1,len(x1)):
    tangles.append(x1[i][0])
    q0.append(x1[i][1])
    q1.append(x1[i][2])
    q2.append(x1[i][3])
    q3.append(x1[i][4])
    yawspeedd.append(x1[i][5])

tangles=np.asarray(tangles)
q0=np.asarray(q0)
q1=np.asarray(q1)
q2=np.asarray(q2)
q3=np.asarray(q3)
#plt.show()

#plt.plot(tangles,yawspeedd)
#plt.show()

roll=np.arctan2(2*(q3*q2+q0*q1),1-2*(q1*q1+q2*q2))
pitch=np.arcsin(2*(q2*q0-q3*q1))
yaw=np.arctan2(2*(q3*q0+q2*q1),-1+2*(q0*q0+q1*q1))

pitch=180/np.pi*pitch
roll=180/np.pi*roll
yaw=180/np.pi*yaw

previous=yaw[0]
for i in range(len(yaw)):
    next=yaw[i]
    if previous-next>180:
        yaw[i]=yaw[i]+360
    if next-previous>180:
        yaw[i]=yaw[i]-360
    previous=yaw[i]

```

```

#plt.subplot(1,2,1)
#plt.plot(tangles/1000000.-timerange[0]/1000000.,roll)
#plt.xlim(0,timerange[1]/1000000.-timerange[0]/1000000.)
#plt.ylim(-15,15)
#plt.xlabel('time [s]')
#plt.ylabel('roll angle [ $\phi$ ]')
#plt.title("Raw roll data")
#plt.subplot(1,2,2)
#plt.plot(tanglesb/1000000.-timerange[0]/1000000.,rollb)
#plt.xlim(0,timerange[1]/1000000.-timerange[0]/1000000.)
#plt.xlabel('time [s]')
#plt.ylabel('roll angle [ $\phi$ ]')
#plt.ylim(-15,15)
#plt.title("EKF filtered roll data")
#plt.show()
extratime=time[-1]-time[0]
aileronfinal=[]
rudderfinal=[]
j=1
timevaluesfinal=[]
iii=0
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    timevaluesfinal.append(timevalue)
    while ty1[j]<timevalue:
        j=j+1
    if iii==0:
        jstart=j
        iii=1
    jend=j
    front=ty1[j-1]
    back=ty1[j]
    deltat=back-front
    extra=timevalue-front
    aileronfinal.append(aileron1[j-
1]+float(extra)/float(deltat)*(aileron1[j]-aileron1[j-1]))
    rudderfinal.append(ruddery1[j-
1]+float(extra)/float(deltat)*(ruddery1[j]-ruddery1[j-1]))

    print ('freq pmw output '),
    print((j-jstart)/extratime)
    iii=0
    rollfinal=[]
    vehicleheadingfinal=[]

j=1
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]

```

```

while tanglesb[j]<timevalue:
    j=j+1
front=tanglesb[j-1]
if iii==0:
    jstart=j
    iii=1
back=tanglesb[j]
deltat=back-front
extra=timevalue-front
rollfinal.append(rollb[j-1]+float(extra)/float(deltat)*(rollb[j]-
rollb[j-1]))
vehicleheadingfinal.append(yawb[j-
1]+float(extra)/float(deltat)*(yawb[j]-yawb[j-1]))
print ('freq roll angle '),
print((j-jstart)/extratime)
iii=0
yawspeedfinal=[]
rollspeedfinal=[]
j=1
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    while tb1[j]<timevalue:
        j=j+1
        front=tb1[j-1]
        back=tb1[j]
        if iii==0:
            jstart=j
            iii=1
        deltat=back-front
        extra=timevalue-front
        rollspeedfinal.append((rollspeed[j-
1]+float(extra)/float(deltat)*(rollspeed[j]-rollspeed[j-1]))*180./np.pi)
        yawspeedfinal.append((yawspeed[j-
1]+float(extra)/float(deltat)*(yawspeed[j]-yawspeed[j-1]))*180/np.pi)
        print ('freq roll speed '),
        print((j-jstart)/extratime)
        iii=0

for i in range (1,len(a1)):
    ta1.append(a1[i][0])
    rhoa1.append(a1[i][4])
    baroalta1.append(a1[i][1])
    baropressa1.append(a1[i][3])

j=1
vairindfinal=[]
vairtruefinal=[]
airtempfinal=[]

```

```

for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    while tz1[j]<timevalue:
        j=j+1
    front=tz1[j-1]
    back=tz1[j]
    deltat=back-front
    if iii==0:
        jstart=j
        iii=1
    extra=timevalue-front
    vairindfinal.append(vairindz1[j-
1]+float(extra)/float(deltat)*(vairindz1[j]-vairindz1[j-1]))
    vairtruefinal.append(vairtruez1[j-
1]+float(extra)/float(deltat)*(vairtruez1[j]-vairtruez1[j-1]))
    airtempfinal.append(vairtempz1[j-
1]+float(extra)/float(deltat)*(vairtempz1[j]-vairtempz1[j-1]))

densityfinal=[]
print ('freq airspeed '),
print((j-jstart)/extratime)
iii=0
baroaltitudefinal=[]
baropressurefinal=[]
j=1
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    while ta1[j]<timevalue:
        j=j+1
    front=ta1[j-1]
    back=ta1[j]
    if iii==0:
        jstart=j
        iii=1
    deltat=back-front
    extra=timevalue-front
    densityfinal.append(rhoa1[j-1]+float(extra)/float(deltat)*(rhoa1[j]-
rhoa1[j-1]))
    baroaltitudefinal.append(baroalta1[j-
1]+float(extra)/float(deltat)*(baroalta1[j]-baroalta1[j-1]))
    baropressurefinal.append(baropressa1[j-
1]+float(extra)/float(deltat)*(baropressa1[j]-baropressa1[j-1]))

aileroninputsignalfinal=[]
rudderinputsignalfinal=[]
print ('freq barometer '),
print((j-jstart)/extratime)
iii=0

```

```

j=1
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    #print timevalue
    while te1[j]<timevalue:
        j=j+1
    front=te1[j-1]
    if iii==0:
        jstart=j
        iii=1
    back=te1[j]
    deltat=back-front
    extra=timevalue-front
    aileroninputsignalfinal.append(aileroninputsignal[j-
1]+float(extra)/float(deltat)*(aileroninputsignal[j]-aileroninputsignal[j-1]))
    rudderinputsignalfinal.append(rudderinputsignal[j-
1]+float(extra)/float(deltat)*(rudderinputsignal[j]-rudderinputsignal[j-1]))

    print ('freq pmw input '),
    print((j-jstart)/extratime)
    iii=0
    for i in range (1,len(c1)):
        tc1.append(c1[i][0])
        vnorth.append(c1[i][15])
        veast.append(c1[i][16])

    for i in range (1,len(d1)):
        td1.append(d1[i][0])
        vnorthwind.append(d1[i][1])
        veastwind.append(d1[i][2])

gpsnorthspeedfinal=[]
gpseastspeedfinal=[]
j=1
for i in range(len(time)):
    timevalue=time[i]*1000000.0+timerange[0]
    while tc1[j]<timevalue:
        j=j+1
    front=tc1[j-1]
    back=tc1[j]
    if iii==0:
        jstart=j
        iii=1
    deltat=back-front
    extra=timevalue-front
    gpsnorthspeedfinal.append(vnorth[j-
1]+float(extra)/float(deltat)*(vnorth[j]-vnorth[j-1]))

```

```

        gpseastsspeedfinal.append(veast[j-
1]+float(extra)/float(deltat)*(veast[j]-veast[j-1]))
        windnorthspeedfinal=[]
        windeastsspeedfinal=[]
        print ('freq gps speed '),
        print((j-jstart)/extratime)
        iii=0
        j=1
        for i in range(len(time)):
            timevalue=time[i]*1000000.0+timerange[0]
            while tc1[j]<timevalue:
                j=j+1
            front=tc1[j-1]
            back=tc1[j]
            deltat=back-front
            if iii==0:
                jstart=j
                iii=1
            extra=timevalue-front
            windnorthspeedfinal.append(vnorthwind[j-
1]+float(extra)/float(deltat)*(vnorthwind[j]-vnorthwind[j-1]))
            windeastsspeedfinal.append(veastwind[j-
1]+float(extra)/float(deltat)*(veastwind[j]-veastwind[j-1]))
            vtotnegfinal=[]
            relativewindheadingfinal=[]
            sideslipfinal=[]
            print ('freq windspeed '),
            print((j-jstart)/extratime)
            iii=0
            for i in range(len(time)):
                vnn=gpsnorthspeedfinal[i]-windnorthspeedfinal[i]
                ven=gpseastsspeedfinal[i]-windeastsspeedfinal[i]
                vtotn=(vnn*vnn+ven*ven)**0.5
                vtotnegfinal.append(vtotn)
                relativetowindheading=np.arctan2(vnn,ven)*180/np.pi
                relativewindheadingfinal.append(relativetowindheading)
                x=(vehicleheadingfinal[i]+relativetowindheading-90-360)
                if x<-100:
                    x=x+360
                if x<-50:
                    x=x+90
                if x<-50:
                    x=x+90
                if x<-50:
                    x=x+90
                if x<-50:
                    x=x+90
                if x>50:

```

```

        x=x-90
    if x>50:
        x=x-90
    if x>50:
        x=x-90

    sideslipfinal.append(x)

plt.subplot(2,2,1)
plt.plot(time,relativewindheadingfinal)
plt.subplot(2,2,2)
plt.plot(time,vehicleheadingfinal)
plt.subplot(2,2,3)
plt.plot(time,sideslipfinal)
plt.subplot(2,2,4)
plt.plot(time,sideslipfinal)
plt.show()

"""
plt.title('Aileron steps')
plt.subplot(3,2,1)
plt.title('Roll angle')
plt.xlabel('time [s]')
plt.ylabel('roll angle [deg]')
plt.plot(time,rollfinal)
plt.subplot(3,2,2)
plt.title('Sideslip angle')
plt.ylabel('sideslip angle [deg]')
plt.xlabel('time [s]')
plt.plot(time,sideslipfinal)
plt.subplot(3,2,3)
plt.title('Aileron commanded pwm signal')
plt.ylabel('pwm signal [us]')
plt.xlabel('time [s]')
plt.plot(time,aileronfinal)
plt.subplot(3,2,4)
plt.title('Rudder commanded pwm signal')
plt.ylabel('pwm signal [us]')
plt.xlabel('time [s]')
plt.plot(time,rudderfinal)
plt.subplot(3,2,5)
plt.title('Controller roll signal')
plt.ylabel('pwm signal [us]')
plt.xlabel('time [s]')
plt.plot(time,aileroninputsignalfinal)
plt.subplot(3,2,6)
plt.title('Controller yaw signal')

```

```
plt.ylabel('pwm signal [us]')
plt.xlabel('time [s]')
plt.plot(time,rudderinputsignalfinal)
plt.show()
"""
```

```
return [time, rollfinal, vehicleheadingfinal, sideslipfinal, aileronfinal,
rudderfinal, aileroninputsignalfinal, rudderinputsignalfinal, rollspeedfinal,
yawspeedfinal, vairindfinal, vairtruefinal, vtotnegfinal, airtempfinal, densi
tyfinal, baroaltitudefinal, baropressurefinal]
```

```
ruddersweep=[1.55*10**8,1.99*10**8]
ruddersteps=[3.35*10**8,3.62*10**8]
aileronssweep=[6.3*10**8,6.68*10**8]
aileronsteps=[4.80*10**8,4.865*10**8]
ulgname="aileronrudder"
a=createinput(ruddersweep,ulgname,30)
createfile(a,'ruddersweep')
b=createinput(ruddersteps,ulgname,30)
createfile(b,'rudderstep')
c=createinput(aileronssweep,ulgname,30)
createfile(c,'aileronssweep')
d=createinput(aileronsteps,ulgname,30)
createfile(d,'aileronstep')

print('finished')
```

```

Anew=[x(1),x(2),x(3),x(4);0,0,1,0;x(5),x(6),x(7),x(8);x(9),x(10),x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
[~,xestimate1] = lsim(Anew,Bnew,C,D,u1,t1,x01);
%showing damping and frequency
damp(Anew)
%display estimated state space system matrices
disp(Anew)
disp(Bnew)
%display eigenvalues of the estimated and real system
disp(eig(Anew))
x01 = [x(21);x(22);x(23);x(24)]; %initial condition of system states
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];

y1=figure;
x1=[xadapted1(:,2),xestimate1(:,2)+x(25)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['roll angle \phi [' char(176) ']]')
legend('\phi measured', '\phi estimation','location','southwest')
y2=figure;
x1=[xadapted1(:,4),xestimate1(:,4)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['yaw rate r [' char(176) '/s]]')
legend('r measured', 'r estimation','location','southwest')
y3=figure;
x1=[xadapted1(:,1),xestimate1(:,1)+x(26)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['sideslip angle \beta [' char(176) ']]')
legend('\beta measured', '\beta estimation','location','southwest')
y4=figure;
x1=[xadapted1(:,3),xestimate1(:,3)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['roll rate p [' char(176) '/s]]')
legend('p measured', 'p estimation','location','southwest')

```

find100rms function used in main program

```

function [totalerror] = find100rms(x,xtrue1,u1,t1,cost)
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
for i = 1:1:length(u1)
    u1(i,1)=(u1(i,1)+x(19)-1500)/100;
    u1(i,2)=(u1(i,2)+x(20)-1500)/100;
end

```

```

x01 = [x(21);x(22);x(23);x(24)]; %initial condition of system states

```

```

Anew=[x(1),x(2),x(3),x(4);0,0,1,0;x(5),x(6),x(7),x(8);x(9),x(10),x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];

```



```

for i = 1:1:length(u1)
    u1(i,1)=(u1(i,1)+x(19)-1500)/100;
    u1(i,2)=(u1(i,2)+x(20)-1500)/100;
end
for i = 1:1:length(u2)
    u2(i,1)=(u2(i,1)+x(27)-1500)/100;
    u2(i,2)=(u2(i,2)+x(28)-1500)/100;
end

Anew=[x(1),x(2),x(3),x(4);0,0,1,0;x(5),x(6),x(7),x(8);x(9),x(10),x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
[~,xestimate1] = lsim(Anew,Bnew,C,D,u1,t1,x01);
[~,xestimate2] = lsim(Anew,Bnew,C,D,u2,t2,x02);

%showing damping and frequency
damp(Anew)
%display estimated state space system matrices
disp(Anew)
disp(Bnew)
%display eigenvalues of the estimated and real system
disp(eig(Anew))
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];

y1=figure;
x1=[xadapted1(:,2),xestimate1(:,2)+x(25)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['roll angle \phi [' char(176) ']]')
legend('\phi measured', '\phi estimation','location','southwest')
y2=figure;
x1=[xadapted1(:,4),xestimate1(:,4)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['yaw rate r [' char(176) '/s]]')
legend('r measured', 'r estimation','location','southwest')
y3=figure;
x1=[xadapted1(:,1),xestimate1(:,1)+x(26)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['sideslip angle \beta [' char(176) ']]')
legend('\beta measured', '\beta estimation','location','southwest')
y4=figure;
x1=[xadapted1(:,3),xestimate1(:,3)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['roll rate p [' char(176) '/s]]')
legend('p measured', 'p estimation','location','southwest')

y5=figure;
x2=[xadapted2(:,2),xestimate2(:,2)+x(34)];
plot(t2,x2)
xlabel('time [s]')
ylabel(['roll angle \phi [' char(176) ']]')
legend('\phi measured', '\phi estimation','location','southwest')

```

```

y6=figure;
x2=[xadapted2(:,4),xestimate2(:,4)];
plot(t2,x2)
xlabel('time [s]')
ylabel(['yaw rate r [' char(176) '/s]'])
legend('r measured', 'r estimation','location','southwest')
y7=figure;
x2=[xadapted2(:,1),xestimate2(:,1)+x(33)];
plot(t2,x2)
xlabel('time [s]')
ylabel(['sideslip angle \beta [' char(176) ' ]'])
legend('\beta measured', '\beta estimation','location','southwest')
y8=figure;
x2=[xadapted2(:,3),xestimate2(:,3)];
plot(t2,x2)
xlabel('time [s]')
ylabel(['roll rate p [' char(176) '/s]'])
legend('p measured', 'p estimation','location','southwest')

```

Find100rms2 function used for combined optimization

```

function [totalerror] = find100rms2(x,xtrue1,u1,t1,xtrue2,u2,t2,cost)
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
for i = 1:1:length(u1)
    u1(i,1)=(u1(i,1)+x(19)-1500)/100;
    u1(i,2)=(u1(i,2)+x(20)-1500)/100;
end

for i = 1:1:length(u2)
    u2(i,1)=(u2(i,1)+x(27)-1500)/100;
    u2(i,2)=(u2(i,2)+x(28)-1500)/100;
end

x01 = [x(21);x(22);x(23);x(24)]; %initial condition of system states
x02 = [x(29);x(30);x(31);x(32)]; %initial condition of system states

Anew=[x(1),x(2),x(3),x(4);0,0,1,0;x(5),x(6),x(7),x(8);x(9),x(10),x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
[~,xestimate1] = lsim(Anew,Bnew,C,D,u1,t1,x01);
[~,xestimate2] = lsim(Anew,Bnew,C,D,u2,t2,x02);

totalerror=0;
for i=1:1:length(xtrue1)
totalerror=totalerror+cost(1)*(xtrue1(i,1)-xestimate1(i,1)-
x(26))^2+cost(2)*(xtrue1(i,2)-xestimate1(i,2)-x(25))^2+cost(3)*(xtrue1(i,3)-
xestimate1(i,3))^2+cost(4)*(xtrue1(i,4)-xestimate1(i,4))^2;
end
for i=1:1:length(xtrue2)
totalerror=totalerror+cost(1)*(xtrue2(i,1)-xestimate2(i,1)-
x(33))^2+cost(2)*(xtrue2(i,2)-xestimate2(i,2)-x(34))^2+cost(3)*(xtrue2(i,3)-
xestimate2(i,3))^2+cost(4)*(xtrue2(i,4)-xestimate2(i,4))^2;
end

end

```

Verification function with input state variables and output 100 * total difference squared in answer for exact solution used with the exact solution

```

function [totalerror] = owncodetosolve(x)
t = 0:1/30:30;
cybd=0;
cnbd=0;
mub=15.5;
b=13.36;
v=59.9;
S=24.2;
cl=1.1360;
cyb=-0.9896;
cyp=-0.0870;
cyr=0.4300;
clb=-0.0772;
clp=-0.3444;
clr=0.2800;
kxx=0.012;
kxz=0.002;
kzz=0.037;
cnb=0.1638;
cyda=0;
cydr=0.3037;
clda=-0.2349;
cldr=0.0286;
cnda=0.0286;
cndr=-0.1261;
cnp=-0.0108;
cnr=-0.1930;
Afront=[-(cybd-2*mub)*b/v,0,0,0;0,0.5*b/v,0,0;0,0,4*mub*kxx*b/v,-
4*mub*kxz*b/v;-cnbd*b/v,0,-4*mub*kxz*b/v,4*mub*kzz*b/v];
Aback=[cyb,cl,cyp,cyr-4*mub;0,0,1,0;clb,0,clp,clr;cnb,0,cnp,cnr];
A=inv(Afront)*Aback;
Bback=[cyda,cydr;0,0;clda,cldr;cnda,cndr];
B=inv(Afront)*Bback;
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
u=[];
for i = 1:length(t)%slowly increasing sin starting at 2 seconds
    if t(i)<=1
        newu=0.0;
        newu2=0;
    end
    if t(i)>1

        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
end

```



```

        newu=0.0;
        newu2=0;
    end
    if t(i)>1
        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
    u=[u;newu2,newu];
end
%obtaining results from estimated and actual state space system
x0 = [0;0;0;0]; %initial condition of system states
[~,xestimate] = lsim(Aestimated,Bestimated,C,D,u,t,x0);
[~,xtruest] = lsim(Atruest,Btruest,C,D,u,t,x0);
%plotting
y1=figure;
x1=[xestimate(:,1),xtruest(:,1)];
plot(t,x1)
xlabel('time [s]')
ylabel('sideslip angle \beta [rad]')
legend('\beta estimated', '\beta true','location','southwest')
y2=figure;
x1=[xestimate(:,2),xtruest(:,2)];
plot(t,x1)
xlabel('time [s]')
ylabel('roll angle \phi [rad]')
legend('\phi estimated', '\phi true','location','southwest')
y3=figure;
x1=[xestimate(:,3),xtruest(:,3)];
plot(t,x1)
xlabel('time [s]')
ylabel('non-dimensional roll rate pb/2V [rad]')
legend('pb/2V estimated', 'pb/2V true','location','southwest')
y4=figure;
x1=[xestimate(:,4),xtruest(:,4)];
plot(t,x1)
xlabel('time [s]')
ylabel('non-dimensional yaw rate rb/2V [rad]')
legend('rb/2V estimated', 'rb/2V true','location','southwest')

```

Verification with solution with noisy model

```

X0=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Atruest=[-0.143125941665057,0.164299787521731,-0.0125828182344987,-
8.90487492756423;0,0,8.96706586826347,0;-0.415614299260716,0,-
2.09785952288970,1.63925064094685;0.297675031169333,0,-0.134505988023952,-
0.288602779777688];
Btruest=[0,0.043924159745026;0,0;-1.419034886605089,0.132468018508438;-
0.020807011343881,-0.239297065692662];
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];

```

```

u=[];
for i = 1:1:length(t)%slowly increasing sin starting at 2 seconds
    if t(i)<=1
        newu=0.0;
        newu2=0;
    end
    if t(i)>1
        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
    u=[u;newu2,newu];
end
%obtaining results from estimated and actual state space system
x0 = [0;0;0;0]; %initial condition of system states
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
[~,xadapted] = lsim(Atruest,Btruest,C,D,u,t,x0);
[~,xtruest] = lsim(Atruest,Btruest,C,D,u,t,x0);
for i =1:1:length(xadapted)
    xadapted(i,1)=xadapted(i,1)+(0.004*rand(1,1)-0.002)*4;
    xadapted(i,2)=xadapted(i,2)+(0.004*rand(1,1)-0.002)*4;
    xadapted(i,3)=xadapted(i,3)+(0.0007*rand(1,1)-0.00035)*4;
    xadapted(i,4)=xadapted(i,4)+(0.001*rand(1,1)-0.0005)*4;
end
y1=figure;
x1=[xadapted(:,1),xtruest(:,1)];
plot(t,x1)
xlabel('time [s]')
ylabel('sideslip angle \beta [rad]')
legend('\beta with noise', '\beta true','location','southwest')
y2=figure;
x1=[xadapted(:,2),xtruest(:,2)];
plot(t,x1)
xlabel('time [s]')
ylabel('roll angle \phi [rad]')
legend('\phi with noise', '\phi true','location','southwest')
y3=figure;
x1=[xadapted(:,3),xtruest(:,3)];
plot(t,x1)
xlabel('time [s]')
ylabel('non-dimensional roll rate pb/2V [rad]')
legend('pb/2V with noise', 'pb/2V true','location','southwest')
y4=figure;
x1=[xadapted(:,4),xtruest(:,4)];
plot(t,x1)
xlabel('time [s]')
ylabel('non-dimensional yaw rate rb/2V [rad]')
legend('rb/2V with noise', 'rb/2V true','location','southwest')

fun=@(x)owncodetosolvenotexact(x,xadapted);

```

```

options = optimoptions('fmincon',"Display",'iter-
detailed','MaxIter',1000000,'TolX',0.000000000000000000000001);
options.MaxFunctionEvaluations = 3.000000e+05;
% performing optimization
x=fmincon(fun,x0,[],[],[],[],[],[],[],options);
% getting 100 * total square difference
owncodetosolve(x,xadapted)
Aestimated=[x(1),x(2),x(3),x(4);0,0,8.96706586826347,0;x(5),x(6),x(7),x(8);x(9
),x(10),x(11),x(12)];
Bestimated=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
Atruest=[-0.143125941665057,0.164299787521731,-0.0125828182344987,-
8.90487492756423;0,0,8.96706586826347,0;-0.415614299260716,0,-
2.09785952288970,1.63925064094685;0.297675031169333,0,-0.134505988023952,-
0.288602779777688];
Btruest=[0,0.043924159745026;0,0;-1.419034886605089,0.132468018508438;-
0.020807011343881,-0.239297065692662];
%showing damping and frequency
damp(Aestimated)
damp(Atruest)
%display estimated state space system matrices
disp(Aestimated)
disp(Bestimated)
%display eigenvalues of the estimated and real system
disp(eig(Aestimated))
disp(eig(Atruest))
t = 0:1/30:30;
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
%setting up the input to the state space system
u=[];
for i = 1:length(t)%slowly increasing sin starting at 2 seconds
    if t(i)<=1
        newu=0.0;
        newu2=0;
    end
    if t(i)>1
        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
    u=[u;newu2,newu];
end
%obtaining results from estimated and actual state space system
x0 = [0;0;0;0]; %initial condition of system states
[~,xestimate] = lsim(Aestimated,Bestimated,C,D,u,t,x0);
[~,xtruest] = lsim(Atruest,Btruest,C,D,u,t,x0);
%plotting
y5=figure;
x1=[xestimate(:,1),xtruest(:,1)];
plot(t,x1)
xlabel('time [s]')
ylabel('sideslip angle \beta [rad]')

```

```

legend('\beta estimated', '\beta true','location','southwest')
y6=figure;
x1=[xestimate(:,2),xtruest(:,2)];
plot(t,x1)
xlabel('time [s]')
ylabel('roll angle \phi [rad]')
legend('\phi estimated', '\phi true','location','southwest')
y7=figure;
x1=[xestimate(:,3),xtruest(:,3)];
plot(t,x1)
xlabel('time [s]')
ylabel(' non-dimensional roll rate pb/2V [rad]')
legend('pb/2V estimated', 'pb/2V true','location','southwest')
y8=figure;
x1=[xestimate(:,4),xtruest(:,4)];
plot(t,x1)
xlabel('time [s]')
ylabel(' non-dimensional yaw rate pb/2V [rad]')
legend('rb/2V estimated', 'rb/2V true','location','southwest')

```

owncodetosolve function used in the verification with a noisy model

```

function [totalerror] = owncodetosolve(x,xtrue)
t = 0:1/30:30;
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];

u=[];
for i = 1:length(t)%slowly increasing sin starting at 2 seconds
    if t(i)<=1
        newu=0.0;
        newu2=0;
    end
    if t(i)>1
        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
    u=[u;newu2,newu];
end

x0 = [0;0;0;0]; %initial condition of system states
Anew=[x(1),x(2),x(3),x(4);0,0,8.96706586826347,0;x(5),x(6),x(7),x(8);x(9),x(10),x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
[~,xestimate] = lsim(Anew,Bnew,C,D,u,t,x0);
totalerror=0;

```

```

for i=1:1:length(xtrue)
totalerror=totalerror+100*(xtrue(i,1)-xestimate(i,1))^2+100*(xtrue(i,2)-
xestimate(i,2))^2+100*(xtrue(i,3)-xestimate(i,3))^2+100*(xtrue(i,4)-
xestimate(i,4))^2;
end

end

```

owncodetosolvenotexact function used in verification of a noisy model

```

function [totalerror] = owncodetosolvenotexact(x,xtrue)
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
t = 0:1/30:30;
u=[];
for i = 1:1:length(t)%slowly increasing sin starting at 2 seconds
    if t(i)<=1
        newu=0.0;
        newu2=0;
    end
    if t(i)>1

        if t(i)<=3
            newu=0.0;
        end
        if t(i)>3
            newu2=-0.005*sin((t(i)-3)*2*pi*0.01*t(i));
        end
        newu=0.025*sin((t(i)-1)*2*pi*0.01*t(i));
    end
    u=[u;newu2,newu];
end

x0 = [0;0;0;0]; %initial condition of system states
Anew=[x(1),x(2),x(3),x(4);0,0,8.96706586826347,0;x(5),x(6),x(7),x(8);x(9),x(10),
x(11),x(12)];
Bnew=[x(13),x(14);0,0;x(15),x(16);x(17),x(18)];
[~,xestimate] = lsim(Anew,Bnew,C,D,u,t,x0);
totalerror=0;
for i=1:1:length(xtrue)
totalerror=totalerror+100*(xtrue(i,1)-xestimate(i,1))^2+100*(xtrue(i,2)-
xestimate(i,2))^2+100*(xtrue(i,3)-xestimate(i,3))^2+100*(xtrue(i,4)-
xestimate(i,4))^2;
end

end

```

Main Code used for validation purposes

```

aileronssweep = readtable('aileronssweep.txt');

```



```

xlabel('time [s]')
ylabel(['yaw rate r [' char(176) '/s']'])
legend('r measured in rudder step flight test', 'r estimation in rudder step
flight test','location','southwest')
y3=figure;
x1=[xadapted1(:,1),xestimate1(:,1)+x(7)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['sideslip angle \beta [' char(176) '']'])
legend('\beta measured in rudder step flight test', '\beta estimation in
rudder step flight test','location','southwest')
y4=figure;
x1=[xadapted1(:,3),xestimate1(:,3)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['roll rate p [' char(176) '/s']'])
legend('p measured in rudder step flight test', 'p estimation in rudder step
flight test','location','southwest')
y5=figure;
x1=[u1(:,1),u1(:,2)];
plot(t1,x1)
xlabel('time [s]')
ylabel(['PMW signal [us]'])
legend('PMW signal to aileron in rudder step flight test', 'PMW signal to
rudder in rudder step flight test','location','southwest')

```

Function find100rms3 Code used for validation purposes

```

function [totalerror] = find100rms3(x,xtrue1,u1,t1,cost)
C=[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
D=[0,0;0,0;0,0;0,0];
for i = 1:1:length(u1)
    u1(i,1)=(u1(i,1)+x(5)-1500)/100;
    u1(i,2)=(u1(i,2)+x(6)-1500)/100;
end

x01 = [x(1);x(2);x(3);x(4)]; %initial condition of system states

Anew=[-11.5397, -11.2794,79.0499, -74.0499;0,0,1,0; -13.0178,10.9553, -
120.3375,109.5296; -31.8915, -6.9247,5.6598, -11.4465];
Bnew=[422.2687,1.2454;0,0; -580.1320,114.1896;103.5699,122.9555];
[~,xestimate1] = lsim(Anew,Bnew,C,D,u1,t1,x01);

totalerror=0;
for i=1:1:length(xtrue1)
totalerror=totalerror+cost(1)*(xtrue1(i,1)-xestimate1(i,1)-
x(26))^2+cost(2)*(xtrue1(i,2)-xestimate1(i,2)-x(25))^2+cost(3)*(xtrue1(i,3)-
xestimate1(i,3))^2+cost(4)*(xtrue1(i,4)-xestimate1(i,4))^2;
end
end

```