

Modelling Simulation and Control of Fixed-wing UAV: CyberSwan

Jon Bernhard Høstmark

Master of Science in Engineering Cybernetics
Submission date: June 2007
Supervisor: Amund Skavhaug, ITK

Problem Description

Candidate: Jon Bernhard Høstmark

Title: Modelling, simulation and control of fixed-wing UAV: CyberSwan.

The main focus of this assignment is modelling, simulation, implementation of flight control system (FCS) and testing on fixedwing aircraft.

The work is based on the earlier work on construction and design of electrical-powered fixedwing aircraft by Jon Bernhard Høstmark, fall 2006.

Task: The main task of this assignment is to come up with a working UAV prototype. Due to the wide scope and timelimit of the project simple PID controllers will be used for stabilization. Then later work can be based on the working prototype. To achieve this goal, the candidate has to look into necessary literature to ensure the final AFCS is up to the task. In addition to this, necessary test equipment for measuring physical properties will have to be developed. Based on this data, mathematical models will be derived and simulated. Implementation of a flight control system (FCS), based on simulations will then be made. Final flight tests will be conducted spring 2007. This will give feedback on the performance of the total system. Comparison of sampled data from testflights and simulation is scheduled for spring 2007.

The assignment will consist of:

Modelling of aircraft

Measuring HW and implementing results into model

Design of control strategy based on simulations

Implementation of FCS on hardware

Flight tests

Task given: 15.01.2007

Deadline: 12.06.2007

Department of Engineering Cybernetics, NTNU

Supervisor: Amund Skavhaug

Trondheim, June 12Th 2007

Amund Skavhaug

Supervisor

Assignment given: 15. January 2007

Supervisor: Amund Skavhaug, ITK

Modelling, simulation and Control of Fixed-wing UAV: CyberSwan

Norwegian University of Science and Technology, NTNU
Department of Engineering Cybernetics

Jon Bernhard Høstmark
Trondheim, June 2007

Task

Candidate: Jon Bernhard Høstmark

Title: Modelling, simulation and control of fixed-wing UAV: CyberSwan.

The main focus of this assignment is modelling, simulation, implementation of flight control system (FCS) and testing on fixedwing aircraft.

The work is based on the earlier work on construction and design of electrical-powered fixedwing aircraft by Jon Bernhard Høstmark, fall 2006.

Task: The main task of this assignment is to come up with a working UAV prototype. Due to the wide scope and timelimit of the project simple PID controllers will be used for stabilization. Then later work can be based on the working prototype. To achieve this goal, the candidate has to look into necessary literature to ensure the final AFCS is up to the task. In addition to this, necessary test equipment for measuring physical properties will have to be developed. Based on this data, mathematical models will be derived and simulated. Implementation of a flight control system(FCS), based on simulations will then be made. Final flight tests will be conducted spring 2007. This will give feedback on the performance of the total system. Comparison of sampled data from testflights and simulation is scheduled for spring 2007.

The assignment will consist of:

1. Modelling of aircraft
2. Measuring HW and implementing results into model
3. Design of control strategy based on simulations
4. Implementation of FCS on hardware
5. Flight tests

Task given: 15.01.2007

Deadline: 12.06.2007

Department of Engineering Cybernetics, NTNU

Supervisor: Amund Skavhaug

Trondheim, June 12Th 2007

Amund Skavhaug

Supervisor

Preface

The goal of this project has been to develop a working prototype UAV system. Making a intuitive system to ease further work in later projects, has demanded extra effort. Therefore making the system outside the report has been timeconsuming. To get a total understanding of what this means one should take the time to take a look at the videos, code, simulations and computing files attached on the DVD supplied. Making hardware for the system and testing has also been necessary. Of practical reasons only pictures of this hardware are included, this should thought not confine the effort put into this development. Obtaining the last essential bits and pieces to make a almost working system work, has been a huge task. On the way help has been indispensable, therefore I would like to thank: Amund Skavhaug for support and help on the computersystem. Per Åge Krogstad and Luca Oggiano with help with wind tunnel testing. Øyvind Bjørnson Langen and Bård Olav Høstmark for assisting with tunneltesting and other equipment testing. Helge Nørstrud need a thank for helping with literature in Aircraft Dynamics. If it had not been for the people at the workshop at Department of Engineering Cybernetics and Department of Energy and Process Engineering, EPT this project had not been successfull. Last but not least a HUGE thank to my girlfriend Kjersti Witzøe Brøste for support and understanding all the way.

Summary

This report treats modelling, simulation and control of a fixed-wing aircraft, including implementation of a Aircraft Flight Control System (AFCS). The design and construction of a suitable airframe [12] by Jon Bernhard Høstmark is continued in this work. This system was designed to be suitable for surveillance purposes, using electrical propulsion and being low cost. Preferable characteristics considering stability and control to ease control, implementation and tuning of controllers were built into the airframe. The work done here confirms that the goal in [12] was met, and compleating the autonomous system using feedback regulation.

The work finished this spring were divided in to three reports.

- Design and implementation of sensor and computer system for fixed-wing UAV, by Edgar Bjørnntvedt
- Modelling, simulation and control of fixed-wing UAV, Jon Bernhard Høstmark
- Ground Station and hardware peripherals for fixed-wing UAV, Mikael K. Eriksen

Each task was assigned to one person. This report includes the work done with respect to modeling, simulation, control and testflights. The main focus was building a working prototype. In addition to the technical work, the author of this report has ensured that the project stayed on track. The conclusion in the end of the report discuss this further. June 2007 the three subsystems was integrated for flight testing. Video supplied with this work show stable autonomous flight confirming a working AFCS. Having a working system is thought to be of great value for further work. The working system can also be used for verifying the analytical models obtained thought this work. Adding more sensors to the computer system enable further and tighter controllers, such as height control and waypoint navigation. With these features the UAV system should be ready for surveillance purposes in the commercial market.

Contents

1	Introduction	1
1.1	Background	1
1.2	Existing Components	2
1.2.1	Airframe	2
1.2.2	Radio system	2
1.2.3	Computer system	2
1.2.4	Communication system	2
1.3	Disposition of this report	2
1.3.1	Part One: Mathematical model	2
1.3.2	Part Two: Physical properties	2
1.3.3	Part Three: Controller and Navigation	3
1.3.4	Part Four: Implementation of FCS	3
1.3.5	Part Five: Flights	3
1.4	DVD	3
2	Equations of motion	7
2.1	Reference Frames	7
2.1.1	Transformations between BODY and NED	8
2.2	Kinematics	8
2.3	Dynamics	10
2.4	Aerodynamic axes	11
2.5	Airflow Angles	11
2.6	Stability axis	12
2.7	Equations of motion for Steady Maneuvering Flight	12
2.7.1	Steady, Straight Flight	13
2.7.2	Steady, Turns	13
2.7.3	Steady, Pitching Flight	14
2.7.4	Steady, Rolling (Spinning) Flight	14
3	Aerodynamics of general flight	15
3.1	Aerodynamic Forces and Moments	15
3.1.1	Aerodynamic Coefficients	16
3.1.2	Precautions when scaling aircrafts	16
3.1.3	Control derivatives	16
3.1.4	The Dynamic Coefficients	17
3.1.5	Total Aerodynamic Coefficients	18
3.2	Stability and control equations	18
3.2.1	The Stability Derivatives	19
3.2.2	Control surface Stability derivatives	20

4 Linearized equations of motion	23
4.1 Lateralization of kinematic equations	23
4.2 Aerodynamic linearized equations	25
4.3 Combining the kinetic and aerodynamic models	28
4.3.1 Linearized Longitudinal dynamics	28
4.3.2 Linearized Lateral dynamics	29
5 Control Background	31
5.1 Simple decoupled control	31
5.1.1 Empiric model	32
5.1.2 Analytical models	32
5.2 Observer, and Kalman filter	32
5.2.1 Basics of a Kalmanfilter	32
5.2.2 Limitations of the Kalman filter	32
5.3 Guidance	33
6 Coefficients and values	35
6.1 Aerodynamic coefficients, forces and moments	35
6.1.1 Lift	35
6.1.2 Control Forces	37
6.1.3 Aerodynamic Moments	37
6.1.4 Drag	37
6.1.5 Thrust	38
6.2 Mass and Moment	39
6.3 Inertia	39
7 Software	41
7.1 Matlab and SIMULINK	41
7.1.1 Identification Toolbox	41
7.1.2 AeroSim	43
7.1.3 Linearized Stability Derivative Model	43
7.2 RealTime Workshop	44
7.3 PCM and Fail Safe	44
7.4 Additional Software	45
8 Flights	47
8.1 Main flight conditions	47
8.2 First flight (Manual)	47
8.3 Second flight (Data collection)	48
8.4 Third flight (with feedback)	49
8.5 Fourth flight (Heading control)	50
9 Electronics Development	51
9.1 Motor tester	51
9.2 The test rig	51
9.3 Battery tests	54
9.4 Servo tester	55
9.4.1 Resolution	58
9.4.2 Dynamics	58
9.4.3 Torque	60

9.4.4	Stall Current	60
10	Wind tunnel	61
10.1	Setup and measurements	61
10.2	Test selection	61
10.3	Results	63
11	Simulation	65
11.1	Analytic Models	65
11.2	Empiric Models	66
11.2.1	Rudder Yaw	67
11.2.2	Verifying models	67
12	Results	71
12.1	Wind tunnel	71
12.2	Analytical models	71
12.3	Empiric model	71
12.4	Tests	72
12.4.1	Motor	72
12.4.2	Servos	72
12.4.3	Battery	72
12.5	Test flights	72
12.5.1	First	72
12.5.2	Second	72
12.5.3	Third	73
12.5.4	Fourth	73
13	Discussion	75
13.1	Mathematical Models	75
13.1.1	Analytic Models	75
13.1.2	Empiric Models	75
13.2	Measurements	75
13.2.1	Air Frame	76
13.2.2	Motor	76
13.2.3	Battery	76
13.2.4	Servo	77
13.3	Developement of Air Frame	77
13.4	Simulation	77
13.5	Additonal Work	78
13.6	Implementation	78
13.7	Test Flight One:	78
13.8	Test Flight Two:	78
13.9	Test Flight Three:	79
13.10	Test Flight Four:	79
13.11	Goal Fulfillment	79
14	Conclusion	81
14.1	Further Work	81
A	Symbols	87

B Matlab-code	89
B.1 Nonlinear model	89
B.2 Linear model	95
B.3 Empiric model	98
B.4 Motor analysis	103
C SIMULINK diagrams	105
C.1 Nonlinear model	105
C.2 Second order model	106

Chapter 1

Introduction

1.1 Background

Using theory and knowhow from the last five years in a challenging but rewarding project has been the main motivation for this project. Combining interest and theory in to a practical problem ensured a steep learning curve, demanding using earlier experience and expanding ones horizon. Fall 2006 a airframe with low drag, high lifting capabilities suitable for surveillance were developed[12]. When doing this properties easing regulation of the system was vital, excluding the need for advanced controller. This meant among others, minimizing nonlinearity(stall characteristics), attempt keeping poles in the left half plane, reducing coupling and dimensioning controlsurfaces for proper control. The development and testing continued here also serves as a verification on the work done fall 2006.

Similar projects such as the AeroSonde UAV and other existing military UAVs are available on the market when writhing. But only a handful of these use electrical propulsion and even fewer use of the shelf RC components lowering costs. World wide the market and demand for low cost surveillance UAVs are fast expanding giving rise to constant development. At NTNU the CyberEagle project[8] and Modeling, simulation and design of a Flight Control System for an UAV [2] has been done, prior to this work.

A full UAV system is a huge task taken the available time into consideration. This necessitate constantly progressing work and focusing on solutions working in real life. Using sufficient solutions without lingering with perfecting each subsystem has been central in this work.

This report is one of three reports working on designing a complete UAV system spring 2007. The three reports are divided into three subtasks:

- Designing sensor and computer system for the UAV, Edgar Bjørntvedt [3]
- Setting up equations, making measurements, building models for simulation and designing controller for the UAV, Jon Bernhad Høstmark
- Ground Station and hardware peripherals for the fixed-wing UAV, Mikael K. Eriksen [9]

The reports were individual but the practical part was a joined effort, tested spring 2007. Below a overview of the total system is presented, figure 1.1.

1.2 Existing Components

1.2.1 Airframe

The airframe was developed in the work [12]. Some modifications has been made to fit hardware, without altering aerodynamics. Figure 1.1 show the airframe configuration.

1.2.2 Radio system

Manual control and testing was done with traditional RC-modelling equipment. In figure 1.2 the components who make up the radio system are shown.

1.2.3 Computer system

The computer system is a continuation of the work [4]. The total computer system used in the combined work is described in the report [3]. Figure 1.1 show the hart of this system, the PC104 and I/O card.

1.2.4 Communication system

The communication system was intended for tracking and route planning purposes. It also includes a camera. The report [9] describes this subsystem in details. Figure 1.1 show the components on board the airframe.

1.3 Disposition of this report

Five sections make out this report.

1.3.1 Part One: Mathematical model

Part one is concerned with deriving and building the mathematical framework, and simplifying this to make it suitable to implement on a computer system with limited resources. Mathematical representation of reference systems and forces acting on the airframe is covered in the beginning of the first part. These mathematical models are then simplified to make them practical in use. This includes liberalization of equations, disregarding all but the most dominating factors and looking at special flight conditions. Chapters 1, 2 3, 4 and 5 make up the first part.

1.3.2 Part Two: Physical properties

The second part treats making measurements and equipment to measure the physical properties of the UAV system. This includes wind tunnel testing of complete aircraft and looking at controlsurfaces, together with battery, servo and motor testing. This data were analyzed and combined with the first section of the report, giving the basis of the SIMULINK and MATLAB models.

The second part threats obtaining numerical values for use in the equations. To do this the airframe has undergone wind tunnel testing. The data has also been compared to simulation results. Aerodynamic coefficients for the mathematical model was derived from the moment and force measurements in the wind tunnel. This task has by far been the most timeconsuming one, because there were hundreds of coefficients to concider. Obtaining

performance of the Outrunner motor, batteries and servos is also threatened in part two. Chapters 6, 9 and 10 and section 7.1.1 make up the second part.

1.3.3 Part Three: Controller and Navigation

In the third part off the report the equations from the first part and values from the second part are combined in SIMULINK and Matlab. Based on these models and simulations controllers was designed. Chapter 7, 11 threats these subjects.

1.3.4 Part Four: Implementation of FCS

The fourth part mainly consist of implementing the controller derived in part three into the PC104 and Atmel AVR ATMega128 computersystem. This is described further in [4]. In addition to this, programming the transmitter, motorcontroller and FailSafe system was needed. Actuator, rudders, motor and battery system was also made ready and tested for flight.

1.3.5 Part Five: Flights

In the last part all three reports was combined and worked together when testing the total system. Real flighttest with live video transfer during autonomous flight were conducted. Evaluation of the testflights and results make up part of part five. See chapter 8 and 12. In addition to these chapters the video material on the DVD make up this part.

1.4 DVD

The DVD attached this report includes the following catalogues.

- documents: PDF documents on control theory
- Excel: Spreadsheets with aerodynamic, mass and motor data
- Matlab_and_SIMULINK: Matlab code and data source files
- Measurement_data: Sampled data form tests
- Pictures: Pictures from the development
- Programs: Programs used under development
- Report: The master thesis report
- Video: Videos of motor testing and test flights

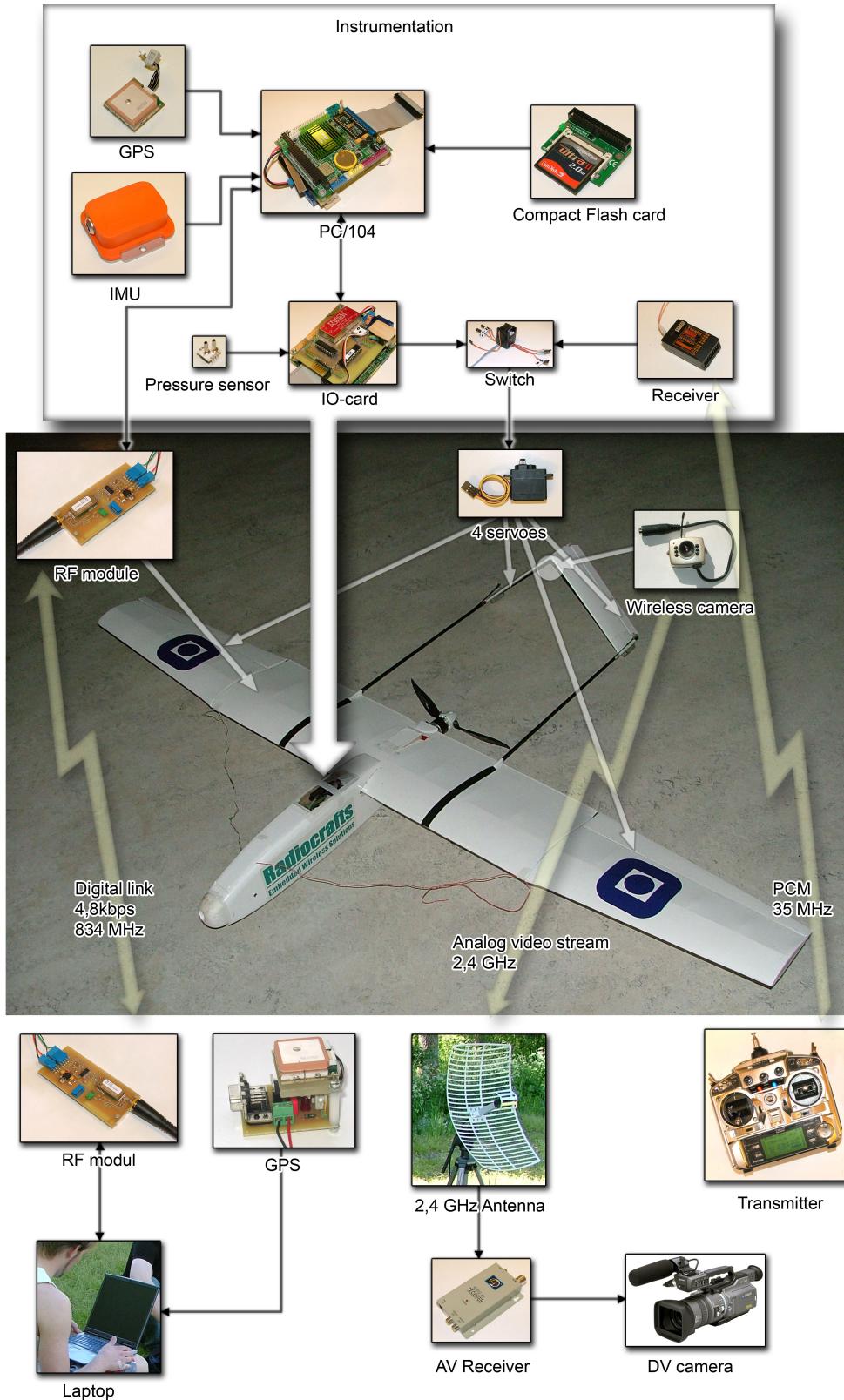


Figure 1.1: Overview of the complete UAV system

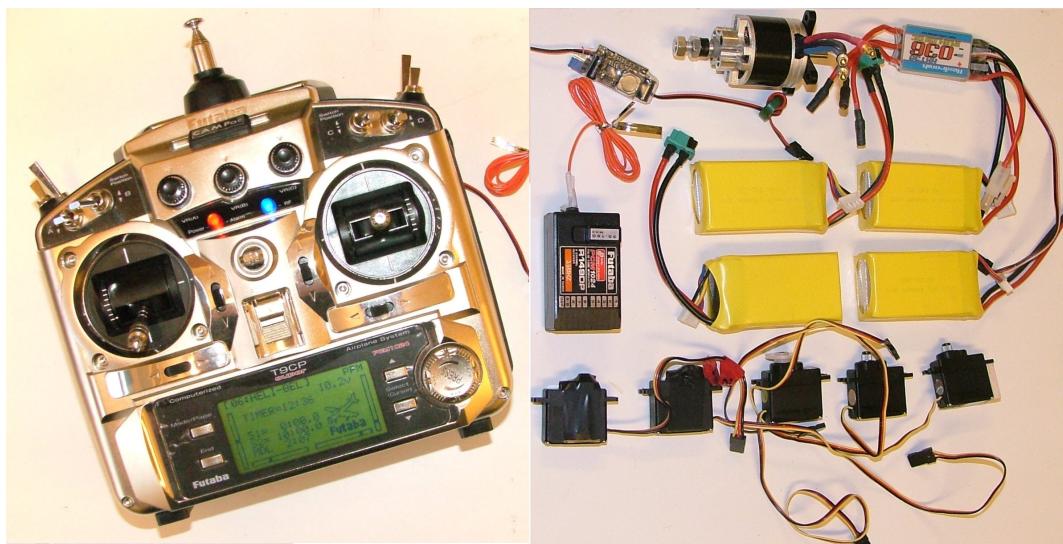


Figure 1.2: The Futaba9C transmitter and the different components of the radiosystem in the airframe.

Chapter 2

Equations of motion

This chapter is concerned with mathematical description of the physics, and the forces action on a airframe. Conversion between different referenceframes will also be covered. The equations and syntax is kept in accordance with [14] and [5]. Some of the material presented in this chapter is also covered in [2] and [12].

First the advantages with the UAV airframe used are explained. Rigid and rugged airframe ensures almost no elastic effects, therefore these are neglected. Electrical propulsion ensures no loss of mass during flight as with gas powered flight. All properties dependent on mass will therefore be constant, this is preferable since the inertia is constant and the controllaw is simplified.

The equations for a general body with six degrees of freedom can be divided in two, kinematics and kinetics. Kinematics is concerned with the motion when subjected to a general force. Kinetics describes the forces involved to produce the motion.

The equations in this chapter are concerned with the linear and angular momentum's. Effects of gravity is also described. Aerodynamic forces is mentioned but examined in detail in the next chapter.

First reference frames need to be defined. In general the measurements used to stabilize and control an aircraft is not given in the body coordinate system as for the motion. Rotation matrices gives an strait forward method of converting between coordinate systems.

2.1 Reference Frames

Whenever working with vessels moving relative to the earth at least two reference frames are present. Commonly the North East Down(NED) frame are chosen for the Earth and BODY for the vessel. The coordinates in each frame is $x_n y_n z_n$ and $x_b y_b z_b$.

The NED frame is not generally inertial because of the rotation and velocity of the earth. For many tasks the NED frame can be simplified and regarded as inertial without making too big an error. Further simplifying it by looking at the earth as a 2D flat surface, and adding the z_n as height coordinate makes an sufficient representation of the NED frame in many cases, as for this. The same simplifications can not be done to the body-frame as this frame is accelerated and rotated with respect to the NED frame. Coordinate and velocity transformation matrices is therefore needed.

Other reference frames of concern when working with aircrafts are, i.e. wind axis systems, see section 2.5. For further explanation see [5]. The stability axes and wind axes systems are also essential, see section 2.6.

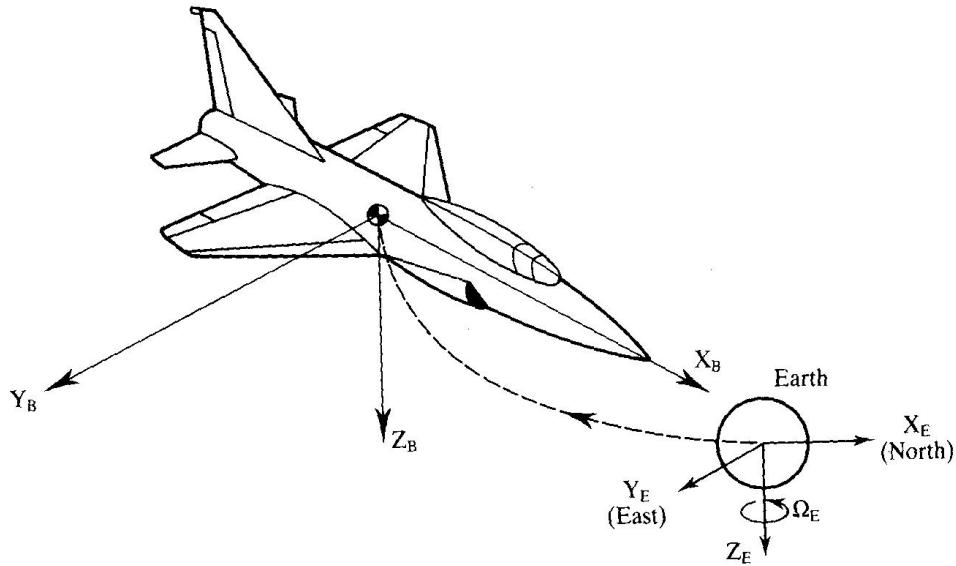


Figure 2.1: EARTH definitions (McLean)

2.1.1 Transformations between BODY and NED

To be able to relate the rotation and motion in the Body frame to the NED frame a rotation matrix is used:

$$R_b^n = \begin{bmatrix} c\Psi c\Theta & -s\Psi c\Phi + c\Psi s\Theta s\Phi & s\Psi s\Phi + c\Psi c\Phi s\Theta \\ s\Psi c\Theta & c\Psi c\Phi + s\Phi s\Theta s\Psi & -c\Psi s\Phi + s\Theta s\Psi c\Phi \\ -s\Theta & c\Theta s\Phi & c\Theta c\Phi \end{bmatrix} \quad (2.1)$$

The velocity vector v_o^b in the Body frame is decomposed in the NED-frame as:

$$\dot{p}^n = R_b^n(\Theta)v_o^b \quad (2.2)$$

Using the Euler angles Θ : roll (ϕ), pitch (θ) and yaw (ψ).

Angular Velocity Transformation

The body-fixed angular velocity vector $\omega_{nb}^b = [P \ Q \ R]^T$ and the Euler rate vector $\dot{\Theta} = [\dot{\Phi} \ \dot{\Theta} \ \dot{\Psi}]$ are related through a transformation matrix $T_{\bar{\Theta}}(\Theta)$ according to:

$$\dot{\Theta} = T_{\bar{\Theta}}(\Theta)\omega_{nb}^b \quad (2.3)$$

where

$$T_{\bar{\Theta}}(\Theta) = \begin{bmatrix} 1 & s\Phi t\Theta & c\Phi t\Theta \\ 0 & c\Phi & -s\Phi \\ 0 & s\Phi/c\Theta & c\Phi/c\Theta \end{bmatrix} \quad (2.4)$$

2.2 Kinematics

The kinematics is concerned with the motion of a body without regarding the forces that cause the motion. The geometrical aspects of motion is based upon Newton's equation of motion

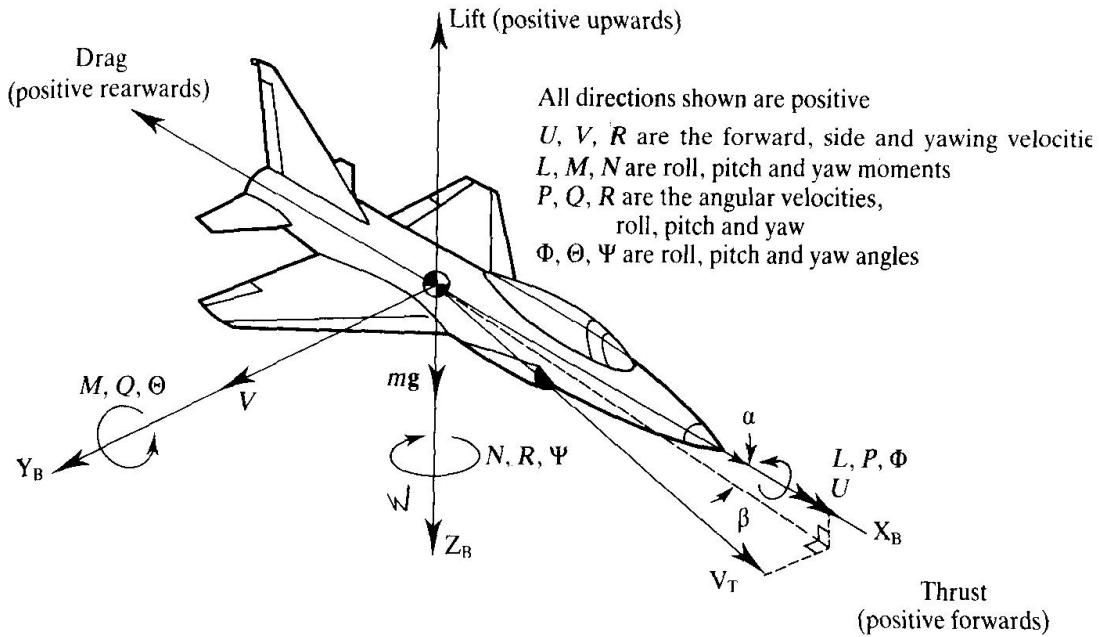


Figure 2.2: BODY definitions (McLean)

$$\sum_{abs} F = ma \quad (2.5)$$

$$\sum_{abs} M = I\dot{\omega} \quad (2.6)$$

The velocity vector ν in the BODY frame is defined as, where the angular velocity is expressed with respect to the NED frame:

$$\nu = \begin{bmatrix} v_o^b \\ \omega_{nb}^b \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \text{forward velocity} \\ \text{sideway velocity} \\ \text{vertical velocity} \\ \text{roll rate} \\ \text{pitch rate} \\ \text{yaw rate} \end{bmatrix} \quad (2.7)$$

The position and orientation vector of the airplane with respect to the NED coordinate system is defined as:

$$\eta = \begin{bmatrix} p^n \\ \Theta \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ z_n, -h \\ \Phi \\ \Theta \\ \Psi \end{bmatrix} = \begin{bmatrix} \text{North} \\ \text{East} \\ \text{Down} \\ \text{roll} \\ \text{pitch} \\ \text{yaw} \end{bmatrix} \quad (2.8)$$

Where p^n is the Euclidean space of dimension 3 and S^3 defines 3 angles on the interval $[0, 2\pi]$

6 DOF Rigid- Body forces and moments in the BODY frame are defined as:

$$\tau_{RB} = \begin{bmatrix} f_o^b \\ m_o^b \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} forward\ force \\ sideways\ force \\ vertical\ force \\ roll\ moment \\ pitch\ moment \\ yaw\ moment \end{bmatrix} \quad (2.9)$$

These forces and moments are the sum of aerodynamic(with disturbance), gravitational and actuator forces and moments.

6 DOF Kinematic Equations

The 6 DOF kinematic equations for an airframe can now be expressed in vector form:

$$\dot{\eta} = J(\eta)\nu \Leftrightarrow \begin{bmatrix} \dot{p}^n \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} R_b^n(\bar{\Theta}) & 0_{3x3} \\ 0_{3x3} & T_{\bar{\Theta}}(\bar{\Theta}) \end{bmatrix} \begin{bmatrix} \eta \\ \omega_{nb}^b \end{bmatrix} \quad (2.10)$$

2.3 Dynamics

The 6 DOF nonlinear dynamic equations for a general BODY expressed in this system takes the form:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau + g_o + w \quad (2.11)$$

Where

M -system inertia matrix(including added mass)

$C(\nu)$ -Coriolis-centripetal matrix (including added mass)

$D(\nu)$ -damping matrix

$g(\eta)$ -vector of gravitational/buoyancy forces and moments

τ -vector of control inputs

g_o -vector used for pre trimming (ballast control)

w -vector of environmental disturbances (wind, waves and currents)

This is made on the assumption that the origin is at the centre of gravity

Neglecting the added mass, buoyancy and ballast this equation simplifies to:

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + D(\nu)\nu + g(\eta) = \tau_{RB} + w \quad (2.12)$$

Where

$$M_{RB} = \begin{bmatrix} mI_{3x3} & 0_{3x3} \\ 0_{3x3} & I_o \end{bmatrix}, C_{RB} = \begin{bmatrix} mS(\omega_{nb}^b) & 0_{3x3} \\ 0_{3x3} & S(I_{CG}\omega_{nb}^b) \end{bmatrix} \quad (2.13)$$

The mapping $S(\bullet)$ is the skew symmetric matrix form of the cross product.

The inertia matrix I_o can be obtained by solving the triple integrals $I_{**} = \int_V (**)\rho_m dV$, where $** = x^2 + y^2, x^2 + z^2, z^2 + y^2, xy, zy$ and xz . Though the moments $I_{xy} = I_{yz} = 0$, due to symmetry of the body. This is xz plane symmetry, resulting in:

$$I_{CG} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \quad (2.14)$$

For further on this subject see, [14], [5] and [7].

The forces and moments acting on the body can now be written:

$$\tau_{RB} = -g(\eta) + \tau \quad (2.15)$$

τ is the contribution from the rudders and thrust forces and moments. The gravity force is decomposed into the BODY system by means of:

$$g(\eta) = -(R_b^n)^T \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} mgsin\Theta \\ -mgcos\Theta sin\Phi \\ -mgcos\Theta cos\Phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

The airplane model may now be written as:

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu + g(\eta) = \tau + w \quad (2.17)$$

On component form this yields:

$$m(\dot{U} + QW - RV + g \sin \Theta) = X \quad (2.18)$$

$$m(\dot{V} + UR - PW - g \cos \Theta \sin \Phi) = Y \quad (2.19)$$

$$m(\dot{W} + PV - QU - g \cos \Theta \cos \Phi) = Z \quad (2.20)$$

$$I_x\dot{P} - I_{xz}(\dot{R} + PQ) + (I_z - I_y)QR = L \quad (2.21)$$

$$I_y\dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR = M \quad (2.22)$$

$$I_z\dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR = N \quad (2.23)$$

These equations are generally used in the remaining report, but will be converted to stability derivatives, see chapter 5.

2.4 Aerodynamic axes

The airplanes velocity vector V_T through the air is generally not aligned with the airplanes body axis. This is illustrated in figure 2.3. Therefore airflow angles need defining, giving rise to Wind axis:

The negative of V_T is as a rule defined as *the relative wind*.

2.5 Airflow Angles

The flight direction with respect to the air is expressed by the two angles β *sideslip angle* and α *angle of attack*, SE figure 2.2. The angles are defined:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} V_T \cos \alpha \cos \beta \\ V_T \sin \beta \\ V_T \sin \alpha \cos \beta \end{bmatrix} \quad (2.24)$$

Leading to the inverse relationship:

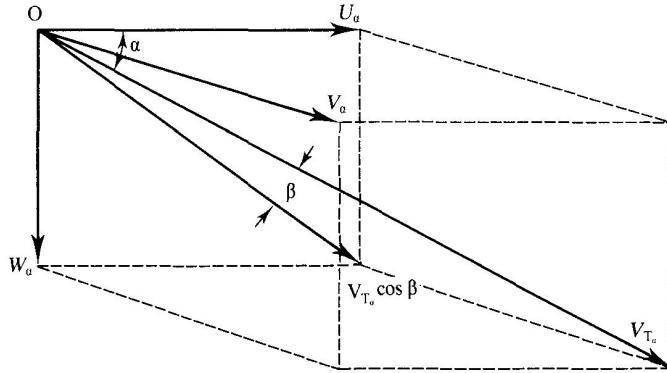


Figure 2.3: The direction of aerodynamic forces,[5]

$$\begin{bmatrix} V_T \\ \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{U^2 + V^2 + W^2} \\ \sin^{-1}(V/V_T) \\ \tan^{-1}(W/U) \end{bmatrix} \quad (2.25)$$

For an airplane, aerodynamic forces and moments arise in the wind axes system. The corresponding effect of these forces and moments are mapped into in the BODY axes by the rotation matrix from BODY to WIND axes. This is defined:

$$R_b^w = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \sin(\beta) & \sin(\alpha)\cos(\beta) \\ -\cos(\alpha)\sin(\beta) & \cos(\beta) & -\sin(\alpha)\sin(\beta) \\ -\sin(\alpha) & 0 & \cos(\beta) \end{bmatrix} \quad (2.26)$$

The aerodynamic forces are more or less linear in the wind axes system given some limitations i.e α and β . This rotation matrix enables these forces to be rotated into the BODY-frame.

2.6 Stability axis

When symmetric flight is assumed, V_0 equals zero. Orienting a axis system such that W_0 is zero gives both α_0 and β_0 equal to zero. This results in the X_B axis, in steady state, pointing into the relative wind, giving

$U_0 = V_T$ Such an orientation results in a stability axis system which, initially, is inclined to the horizon at some path angle, γ_0 , since:

$\Theta_0 \triangleq \gamma_0 + \alpha_0$ and α_0 is zero. This initial alignment does not affect the body-fixed characteristics of the axis system. However, the alignment of the stability axis system with respect to the body axis system changes as a function of the trim condition. This is illustrated in 2.4.

2.7 Equations of motion for Steady Maneuvering Flight

In “standard” flight airplanes usually perform a limited set of maneuvers, these are described in the next section. Combining these flight conditions a complete flight including takeoff, climbing, waypoint navigation, descending and landing can be constructed.

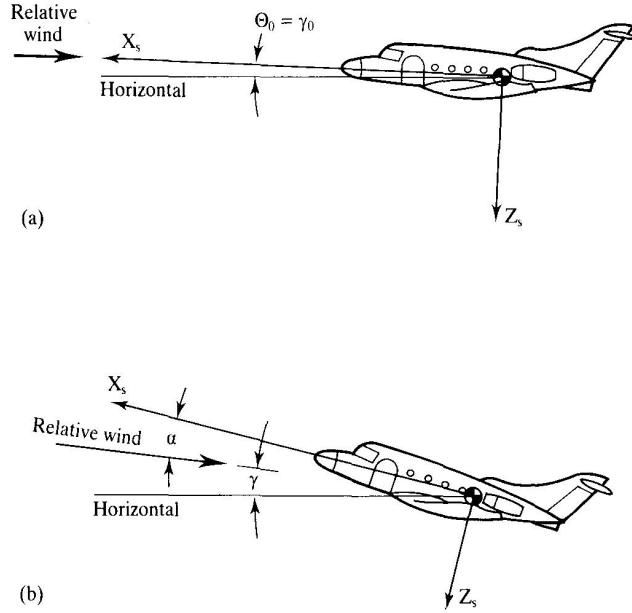


Figure 2.4: Direction of stability axes with respect to the relative wind. (a)Steady flight
(b)Perturbed flight.[5]

2.7.1 Steady, Straight Flight

This is the simplest case of steady flight. Steady, straight flight, implies all time derivatives, angular velocities P, Q, R, and time derivatives of attitude are zero. This gives:

$$X_0 = mg \sin \Theta \quad (2.27)$$

$$Y_0 = -mg \cos \Theta \sin \Phi \quad (2.28)$$

$$Z_0 = -mg \cos \Theta \cos \Phi \quad (2.29)$$

$$L_0 = M_0 = N_0 = 0 \quad (2.30)$$

$$(2.31)$$

These equations can be applied to a steady sideslip manoeuvre, however for symmetric flight the bank angle being zero leads to:

$$X_0 = mg \sin \Theta \quad (2.32)$$

$$Y_0 = 0 \quad (2.33)$$

$$Z_0 = -mg \cos \Theta \quad (2.34)$$

$$(2.35)$$

All moments are zero.

2.7.2 Steady, Turns

For Steady turning flight the same rules apply as for Steady, strait flight, except the rate of turn $\dot{\Psi}$ is constant. Generally steady, turning maneuvers are carried out for small pitching angles, or shallow climbing/diving turns. Hence the relationships below holds.

$$P = -\dot{\Psi} \sin \Theta \simeq -\dot{\Psi} \Theta \quad (2.36)$$

$$Q = -\dot{\Psi} \cos \Theta \sin \Phi \simeq \dot{\Psi} \sin \Phi \quad (2.37)$$

$$R = -\dot{\Psi} \cos \Theta \cos \Phi \simeq \dot{\Psi} \cos \Phi \quad (2.38)$$

$$(2.39)$$

For steady, coordinated, shallow turn, $\dot{\Psi}$ and the velocities V and W are small. Coordinated turns lead to force $Y = 0$, by definition so the equations become:

$$X = mg\Theta \quad (2.40)$$

$$Y = 0 \quad (2.41)$$

$$Z = -m(\dot{\Psi}U \sin \Phi + g \cos \Phi) \quad (2.42)$$

$$\dot{\Psi} = \frac{g}{U_0} \tan \Phi \quad (2.43)$$

$$(2.44)$$

All moments are zero.

2.7.3 Steady, Pitching Flight

Symmetric flight along a curved flight path, with constant pitching velocity, leads to a quasi-steady flight condition. The states V, P, R, Φ and Ψ are zero, but U and W can vary. This gives:

$$X = m(\dot{U} + QW) + mg \sin \Theta \quad (2.45)$$

$$Z = m(\dot{W} + QU) - mg \cos \Theta \quad (2.46)$$

$$L = M = N = Y = 0 \quad (2.47)$$

$$(2.48)$$

For reasonable values of $\dot{\Theta}$, the linear accelerations \dot{u} and \dot{w} are small; then 2.45 become:

$$X_0 = m(Q_0W_0 + g \sin \Theta_0) \quad (2.49)$$

$$Z_0 = -m(Q_0U_0 + g \cos \Theta_0) \quad (2.50)$$

$$Q_0 = \frac{g}{U_0} \left(-\frac{Z_0}{mg} - \cos \Theta_0 \right) \quad (2.51)$$

$$(2.52)$$

2.7.4 Steady, Rolling (Spinning) Flight

This fight condition cannot be simplified without improperly describing the physics involved. No simplified equations are presented, as special methods of threatmet are required. Further Steady, spinning flight are not needed to perform a standard flight form A to B. This is a dogfight or aerobatic manoeuvre.

Chapter 3

Aerodynamics of general flight

Basic aerodynamic forces and moments are presented in this chapter. A detailed description of the aerodynamic properties of an airplane will not be covered in this report as this was covered in [12]. Please refer to [1], [15] and [18] for complete description of aerodynamics.

3.1 Aerodynamic Forces and Moments

Forces and moments are usually written as products of a nondimensional coefficients and dimensionless quantities. The aerodynamic coefficients can be found theoretically, or experimentally found by i.e wind tunnel testing as with this report. The dimensionless coefficients are useful because scaling is made easier. But as will be seen in section 3.1.2 this is not as easy as it sounds. The dimensional forces are found by multiplying the coefficients with the reference area $S [m^2]$ and the *dynamic pressure* $q [N/m^2]$ (from Bernoulli's theorem for streamlined flow):

$$q = \frac{1}{2} \rho V_T^2 \quad (3.1)$$

Where ρ is the air density. The aerodynamic forces thus may be written:

$$\begin{bmatrix} -X \\ Y \\ -Z \\ L \\ M \\ N \end{bmatrix}_{WIND} = \begin{bmatrix} Drag \\ Side Force \\ Lift \\ Rolling Moment \\ Pitching Moment \\ Yawing Moment \end{bmatrix}_{WIND} = \begin{bmatrix} qSC_D \\ qSC_Y \\ qSC_L \\ qSbC_l \\ qS\bar{c}C_M \\ qSbC_N \end{bmatrix} \quad (3.2)$$

where the *nondimensional aerodynamic coefficients* are:

- C_D : Drag coefficient
- C_Y : Side Force coefficient
- C_L : Lift coefficient
- C_l : Rolling Moment coefficient
- C_M : Pitching Moment coefficient
- C_N : Yawing Moment coefficient

The Drag and Lift forces are defined positive in the negative X and Z direction. The Moments are also a function of the mean chord, \bar{c} , and the span, b.

These forces and moments are linearized and simplified by the use of Stability derivatives in chapter 4.

3.1.1 Aerodynamic Coefficients

Several factors heavily impact on the aerodynamic coefficients, these are:

- Aircraft shape and proportion: section 6.1.1
- Airflow angles : section 2.5
- Angular rates : section 3.1.4
- Control settings : section 3.1.3
- Mach and Reynolds numbers 3.1.2

The first four of these factors are easy to understand, and does not imply any scaling problems. The next section briefly explains the last property and its impact on smaller airframes.

3.1.2 Precautions when scaling aircrafts

Aerodynamic forces are highly dependent on the Reynolds number, see equation 3.3. Traditional aerodynamic models are based on laminar theory. Generally model airplanes operate at low Reynold numbers compared with traditional airplanes. Viscous and turbulent effects are therefore more dominating for smaller aircrafts. To compensate for this the airfoils chosen in [12] was special laminar lowspeed foils, still precaution should be taken when using traditional aerodynamic theory. This subject is treated thorough in [1].

$$Re = \frac{\text{Density}}{\text{Viscosity}} \times \text{Velocity} \times \text{Length} \quad (3.3)$$

$$= \frac{\rho}{\mu} VL, \quad (3.4)$$

It is essential to know the Reynolds number when working with aerodynamic coefficients, when using windtunnel data the turbulencefactor of the tunnel should also be known, as this affects flow.

L is the characteristic length of the object, for airfoils the cord, alternatively the MAC is used. At lower Reynolds numbers(50.000-80.000) a transition from laminar to turbulent flow often happens, therefore the length of the airfoils of the UAV have a MAC matched to the operating speed. Effects of compressibility is not an issue for the kind of flight considered for the UAV. The compressibility is determined by the MACH number. Usually for speeds below 0.3 Mach no compressibility is considered, model airplanes are well into this subsonic area.

3.1.3 Control derivatives

The total nondimensional aerodynamic coefficients are functions of several components. Generally one can write the coefficients, known as nondimensional derivatives, as:

$$C_X = \sum_i C_{X_{state_i}} state_i + \sum_i C_{X_{\delta_i}} \delta_i \quad (3.5)$$

Where *state* are the aerodynamic states and δ are the *control derivatives*. These are associated with control surfaces provided by the rudder, ailerons, flaps and elevator. Most aircrafts have the following 5 control surfaces, these are illustrated in figure 3.1:

- Ailerons: C_{δ_a}
- Flaps: C_{δ_f}
- Elevator: C_{δ_e}
- Rudder: C_{δ_r}
- Thrust: C_{δ_T}

In addition canards and three surface planes have:

- Canard: C_{δ_c}

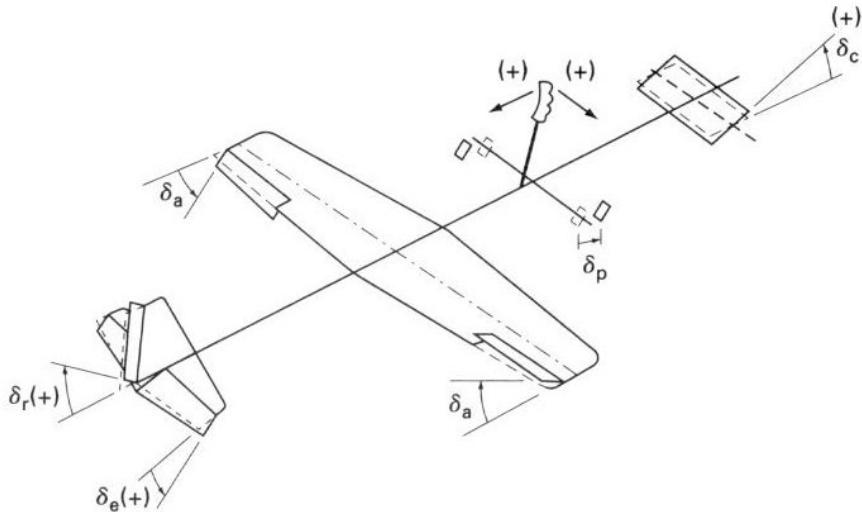


Figure 3.1: The rudder(δ_r), elevator(δ_e), ailerons(δ_a) and canard(δ_c), [2]

Figure 3.1 show control surfaces and their angle definitions, of typical airplanes. Based on using the right-hand rule, the elevator is defined to have a positive angle about y_b , the rudder a positive angle about z_b , and the ailerons is described as the sum $\delta_a = \frac{1}{2}(\delta_{a_L} + \delta_{a_R})$ defined positive about x_b .

3.1.4 The Dynamic Coefficients

Steady flight produces steady aerodynamic forces and moments, but maneuvering flight produces additional forces and moments arising from rotational and unsteady motion. This is due to a disturbed airflow pattern coming from the linear and rotational accelerations. These are essential for damping. The most important dynamic coefficients are:

- $C_{L_{\dot{q}}}$
- $C_{Y_{\dot{p}}}$
- $C_{l_{\dot{p}}}, C_{l_{\dot{r}}}$

- $C_{M_{\dot{q}}}, C_{M_{\dot{\alpha}}}$
- $C_{N_{\dot{p}}}, C_{N_{\dot{r}}}$

The dynamic coefficients are derived from non dimensional rates and accelerations.

3.1.5 Total Aerodynamic Coefficients

The total definition of aerodynamic coefficients, including static and dynamic effects adds up to:

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\dot{q}}} \frac{q\bar{c}}{2V_T} + C_{L_{\dot{\alpha}}} \frac{\dot{\alpha}\bar{c}}{2V_T} \dot{\alpha} + C_{L_{\delta_e}} \delta_e \quad (3.6)$$

$$C_D = C_{D_0} + \epsilon C_L^2 + C_{D_\delta} \delta \quad (3.7)$$

$$C_Y = C_{Y_\beta} \beta + C_{Y_{\dot{p}}} \frac{pb}{2V_T} + C_{L_{\delta_r}} \delta_r \quad (3.8)$$

$$C_l = C_{l_\beta} \beta + C_{l_{\dot{p}}} \frac{pb}{2V_T} + C_{l_{\dot{r}}} \frac{rb}{2V_T} + C_{l_{\delta_r}} \delta_r + C_{l_{\delta_a}} \delta_a \quad (3.9)$$

$$C_M = C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\dot{\alpha}}} \frac{\dot{\alpha}\bar{c}}{2V_T} + C_{M_{\dot{q}}} \frac{q\bar{c}}{2V_T} + C_{M_{\delta_e}} \delta_e \quad (3.10)$$

$$C_N = C_{N_\beta} \beta + C_{N_{\dot{p}}} \frac{pb}{2V_T} + C_{N_{\dot{r}}} \frac{rb}{2V_T} + C_{N_{\delta_r}} \delta_r + C_{N_{\delta_a}} \delta_a \quad (3.11)$$

With the resulting total description of the aerodynamic forces:

$$\begin{bmatrix} Drag \\ Side Force \\ Lift \\ Rolling Mom. \\ Pitching Mom. \\ Yawing Mom. \end{bmatrix} = \frac{1}{2} \rho S V_T^2 \begin{bmatrix} C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\dot{q}}} \frac{q\bar{c}}{2V_T} + C_{L_{\dot{\alpha}}} \frac{\dot{\alpha}\bar{c}}{2V_T} \dot{\alpha} + C_{L_{\delta_e}} \delta_e \\ C_{D_0} + \epsilon C_L^2 + C_{D_\delta} \delta \\ C_{Y_\beta} \beta + C_{Y_{\dot{p}}} \frac{pb}{2V_T} + C_{L_{\delta_r}} \delta_r \\ b(C_{l_\beta} \beta + C_{l_{\dot{p}}} \frac{pb}{2V_T} + C_{l_{\dot{r}}} \frac{rb}{2V_T} + C_{l_{\delta_r}} \delta_r + C_{l_{\delta_a}} \delta_a) \\ \bar{c}(C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\dot{\alpha}}} \frac{\dot{\alpha}\bar{c}}{2V_T} + C_{M_{\dot{q}}} \frac{q\bar{c}}{2V_T} + C_{M_{\delta_e}} \delta_e) \\ b(C_{N_\beta} \beta + C_{N_{\dot{p}}} \frac{pb}{2V_T} + C_{N_{\dot{r}}} \frac{rb}{2V_T} + C_{N_{\delta_r}} \delta_r + C_{N_{\delta_a}} \delta_a) \end{bmatrix} \quad (3.12)$$

3.2 Stability and control equations

Looking at the linearized equations of motion of an aircraft, in chapter 4, it is obvious that determining all pieces of the equation is a time-consuming task. To simplify the equations, it is convenient to make the following substitutions:

$$\begin{aligned} X_x &= \frac{1}{m} \frac{\partial X}{\partial \dot{x}} \\ Z_x &= \frac{1}{m} \frac{\partial Z}{\partial \dot{x}} \\ M_x &= \frac{1}{I_{yy}} \frac{\partial M}{\partial \dot{x}} \end{aligned} \quad (3.13)$$

The variables such as X_x , Z_x and M_x are called stability derivatives. As will be shown in chapter 4 this makes it possible to describe the total aircraft system in compact form with two 4×4 -matrix's. First the stability derivatives are described here.

3.2.1 The Stability Derivatives

[5] and [17] gives an good understanding of stability derivatives. Because of the great number of derivatives it is common to assume that the plane is operating about a trimmed condition. Further to assume that the disturbances from steady flight is small enough, so that the sine and cosines of the disturbance angles can be neglected. Then that the steady state trim conditions are $P_0 = R_0 = V_0 = \Theta_0 = 0$. That the longitudinal forces and moments due to lateral perturbations about these trim conditions are negligible. Last that the flow is quasisteady, omitting all derivatives related to change of velocities. These four assumptions greatly reduces the number of stability derivatives.

Under is a table of most the of the important standard and primed stability derivatives, which is the ones used in the simulations presented in this report. To eliminate the cross-product inertia terms, primed stability derivatives are used , see [5] and [17]. Second order effects are ignored. Only w and v derivatives are presented below.

$$\alpha = \frac{w}{U_0} \quad (3.14)$$

$$\beta = \frac{v}{U_0} \quad (3.15)$$

The w and v -derivatives can easily be obtained from the α and β derivatives, see [5]. This holds for small angles:

Longitudinal Motion Related Derivatives:

$$\begin{aligned}
X_u &= \frac{\rho S U}{m} (-C_D - C_{D_u}) & X_w &= \frac{\rho S U}{2m} (C_L - C_{D_\alpha}) \\
Z_u &= \frac{\rho S U}{2m} (-C_{D_\delta}) & Z_w &= \frac{\rho S U}{2m} (-C_\alpha - C_D) \\
M_u &= \frac{\rho S U c}{I_y} (C_M + C_{M_u}) & M_w &= \frac{\rho S U c}{2I_y} (C_{M_\alpha}) \\
M_{\dot{w}} &= \frac{\rho S c^2}{4I_y} (C_{M_\alpha}) & M_q &= \frac{\rho S U c^2}{4I_y} (C_{M_q}) \\
\tilde{M}_u &= (M_U + M_{\dot{w}} Z_u) & \tilde{M}_w &= (M_w + M_{\dot{w}} Z_w) \\
\tilde{M}_q &= (M_q + U_0 M_{\dot{w}}) & \tilde{M}_\Theta &= (-g M_w \sin \gamma_0) \\
\tilde{M}_{\delta_E} &= (M_{\delta_E} + M_{\dot{w}} Z_{\delta_E})
\end{aligned} \quad (3.16)$$

Lateral Motion Related Derivatives:

$$\begin{aligned}
Y_v &= \frac{\rho S U}{2m} C_{y_\beta} & L_\beta &= \frac{\rho S U^2 b}{2I_x} (C_{l_\beta}) \\
L_p &= \frac{\rho S U b^2}{4I_x} (C_{l_p}) & L_r &= \frac{\rho S U b^2}{4I_x} (C_{l_r}) \\
N_\beta &= \frac{\rho S U^2 b}{4I_z} (C_{n_\beta}) & N_p &= \frac{\rho S U b^2}{4I_z} (C_{N_p}) \\
N_r &= \frac{\rho S U b^2}{4I_z} (C_{N_r}) & L'_\beta &= (L_\beta + I_B N_\beta) \\
L'_p &= (L_p + I_B N_p) & L'_r &= (L_r + I_B N_r) \\
N'_\beta &= (N_\beta + I_A L_\beta) & N'_p &= (N_p + I_A L_p) \\
N'_r &= (N_r + I_A L_r)
\end{aligned} \tag{3.17}$$

In witch

$$I_A \triangleq I_{xz}/I_{xx} \tag{3.18}$$

$$I_B \triangleq I_{xz}/I_{zz} \tag{3.19}$$

3.2.2 Control surface Stability derivatives

Generally for an airplane one is looking at three moments, three velocities and three rotations. For this airplane three main control surfaces and throttle are considered. This gives rise to numbersome equations. Here just the most significant equations are presented. For further information on the subject see [5]. From the wind tunnel tests in chapter 10, data for the control surfaces were found. The control surface stability derivatives are easily calculated using the corresponding coefficient. The coefficients take the general form $(Force \ or \ Moment)/qS*$, * being b or \bar{u} depending on the coefficient. Substituting this into the equations above the value of all stability derivatives were found. Below the dominating control related equations are presented.

Longitudinal:

$$M_{\delta_E} = \frac{\rho S U^2 c}{2I_y} (C_{M_{\delta_E}}) \text{ Pitch moment} \tag{3.20}$$

(“elevator effectiveness”)

$$X_{\delta_E} = \frac{\rho S U^2}{2m} (-C_{D_{\delta_E}}) \text{ Drag force} \tag{3.22}$$

$$Z_{\delta_E} = \frac{\rho S U^2}{2m} (-C_{L_{\delta_E}}) \text{ Elevator lift} \tag{3.23}$$

$$X_T = T \cos(eet - \alpha_0) \text{ Thrustforce} \tag{3.24}$$

$$\tilde{M}_{\delta_E} = (M_{\delta_E} + M \dot{w} Z_{\delta_E}) \tag{3.25}$$

$$\tilde{M}_{\delta_E} = (M_{\delta_E} + M \dot{w} Z_{\delta_E}) \tag{3.26}$$

Lateral:

$$L_\delta = \frac{\rho U^2 S b}{2 I_x} C_{l_\delta} \text{ roll moment, } (C_{l_{\delta_A}} = \text{"aileron effectiveness"}) \quad (3.27)$$

$$N_\delta = \frac{\rho U^2 S b}{2 I_z} C_{n_\delta} \text{ yaw moment, } (C_{n_{\delta_R}} = \text{"rudder effectiveness"}) \quad (3.28)$$

(3.29)

Primed lateral control derivatives

$$L'_{\delta_A} = (L_{\delta_A} + I_B N_{\delta_A}) \quad (3.30)$$

$$L'_{\delta_R} = (L_{\delta_R} + I_B N_{\delta_R}) \quad (3.31)$$

$$N'_{\delta_A} = (L_{\delta_A} + I_A N_{\delta_A}) \quad (3.32)$$

$$N'_{\delta_R} = (L_{\delta_R} + I_A N_{\delta_R}) \quad (3.33)$$

(3.34)

The motion and control related stability derivatives presented here are only a small portion of all the stability derivatives of a 6-DOF airframe, but sufficient to make a god model of the system. Using feedback these modeling errors will not be of great importance, especially considering the airframe used.

Chapter 4

Linearized equations of motion

In this chapter the general nonlinear equations of motion from chapter 2 are linearized. These are linearized about trimmed flight conditions. For the flight envelope of this aircraft, with constraints on the maximum angles and angular velocities this will be adequate for *steady flight*. The chapter is based on [5], [2]. The states are expanded about the trimmed condition as:

Linearized State = Nominal value + perturbed state

This gives the following definitions:

$$\tau = \tau_0 + \Delta\tau = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta L \\ \Delta M \\ \Delta N \end{bmatrix} \quad (4.1)$$

$$\nu = \nu_0 + \Delta\nu = \begin{bmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \\ Q_0 \\ R_0 \end{bmatrix} + \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad (4.2)$$

And

$$\begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix} = \begin{bmatrix} \Phi_0 \\ \Theta_0 \\ \Psi_0 \end{bmatrix} + \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (4.3)$$

giving the new states $u, v, w, p, q, r, \theta, \phi$ and ψ with the addition of the perturbed wind axes states $\alpha_0 + \alpha$ and β (assume $\beta_0 = 0$). This is the deviations from the equilibrium condition or trim.

4.1 Lateralization of kinematic equations

The airplane's state's nominal values are: (presume $\dot{\nu}_0 = 0$)

$$C_{RB}(\nu_0)\nu_0 + g(\eta_0) = \tau_0 \quad (4.4)$$

On component form this is:

$$m(Q_0 W_0 - R_0 V_0 + g \sin \Theta_0) = X_0 \quad (4.5)$$

$$m(U_0 R_0 - P_0 W_0 - g \cos \Theta_0 \sin \Phi_0) = Y_0 \quad (4.6)$$

$$m(P_0 V_0 - Q_0 U_0 - g \cos \Theta_0 \cos \Phi_0) = Z_0 \quad (4.7)$$

$$(I_z - I_y)Q_0 R_0 - P_0 Q_0 I_{xz} = L_0 \quad (4.8)$$

$$(P_0^2 - R_0^2)I_{xz} + (I_x - I_z)P_0 R_0 = M_0 \quad (4.9)$$

$$(I_y - I_x)P_0 Q_0 - Q_0 R_0 I_{xz} = N_0 \quad (4.10)$$

The perturbed equations are usually found by a first order Taylor expansion. Linearizing all the states in the equation set, (2.18)-(2.23) yields:

$$m(\dot{u} + Q_0 w - W_0 q - R_0 v - V_0 r + (g \cos \Theta_0)\theta) = \Delta X \quad (4.11)$$

$$m(\dot{v} + U_0 r + R_0 u - P_0 w - W_0 p - (g \cos \Theta_0 \cos \Phi_0)\phi + (g \sin \Theta_0 \sin \Phi_0)\theta) = \Delta Y \quad (4.12)$$

$$m(\dot{w} + V_0 p + P_0 v - U_0 q - Q_0 u + (g \cos \Theta_0 \sin \Phi_0)\phi + (g \sin \Theta_0 \cos \Phi_0)\theta) = \Delta Z \quad (4.13)$$

$$I_x \dot{p} - I_{xz} \dot{r} + (I_z - I_y)(Q_0 r + R_0 q) - 2I_{xz}(R_0 r + P_0 p) = \Delta L \quad (4.14)$$

$$I_y \dot{q} + (I_x - I_z)(P_0 r + R_0 p) - I_{xz}(P_0 q + Q_0 p) = \Delta M \quad (4.15)$$

$$I_z \dot{r} - I_{xz} \dot{p} + (I_y - I_x)(P_0 q + Q_0 p) + I_{xz}(Q_0 r + R_0 q) = \Delta N \quad (4.16)$$

Assuming that the initial velocity V_0 , the angle Φ_0 , and the angular rates are zero (straight flight with wings level) yields:

$$m(\dot{u} - W_0 q + (g \cos \Theta_0)\theta) = \Delta X \quad (4.17)$$

$$m(\dot{v} + U_0 r - W_0 p - (g \cos \Theta_0)\phi) = \Delta Y \quad (4.18)$$

$$m(\dot{w} - U_0 q + (g \sin \Theta_0)\theta) = \Delta Z \quad (4.19)$$

$$I_x \dot{p} - I_{xz} \dot{r} = \Delta L \quad (4.20)$$

$$I_y \dot{q} = \Delta M \quad (4.21)$$

$$I_z \dot{r} - I_{xz} \dot{p} = \Delta N \quad (4.22)$$

Note that equations 4.17, 4.19, and 4.21 contains only terms involving forces, moments, and motion in the x-z plane: constituting the *longitudinal states*. Equations 4.18, 4.20, and 4.22 equally constitutes the *lateral states*. These two sets of equations are now entirely independent and may be solved as two 3 DOF systems. The assumptions made here correspond to the flight case *steady flight*, see below. The most important flight conditions for steady maneuvering flight are presented in section 2.7.

On compact vector form the equations are stated as:

$$M_{RB} \dot{\nu} + N_{RB} \nu + G\eta = \Delta\tau \quad (4.23)$$

Where

$$\begin{aligned}
M_{RB} &= \begin{bmatrix} m & 0 & 0 & \\ 0 & m & 0 & 0_{3 \times 3} \\ 0 & 0 & m & \\ & & & I_x & 0 & -I_{xz} \\ & & & 0_{3 \times 3} & 0 & I_y & 0 \\ & & & & -I_{xz} & 0 & I_z \end{bmatrix} \\
N_{RB} &= \begin{bmatrix} 0 & 0 & 0 \\ 0_{3 \times 3} & 0 & mU_0 \\ 0 & -mU_0 & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & \end{bmatrix} \\
G &= \begin{bmatrix} 0 & mg \cos \Theta_0 & 0 \\ 0_{3 \times 3} & -mg \cos \Theta_0 & 0 \\ 0 & mg \sin \Theta_0 & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & \end{bmatrix}
\end{aligned}$$

The kinematics is linearized as:

$$T_{\bar{\Theta}}(\bar{\Theta}) = \begin{bmatrix} 1 & 0 & \tan \Theta_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/\Theta_0 \end{bmatrix} \quad (4.24)$$

$$\begin{bmatrix} V_T \\ \beta \\ \alpha \end{bmatrix} \approx \begin{bmatrix} U_0 \\ \frac{v}{U_0} \\ \frac{w}{U_0} \end{bmatrix} \quad (4.25)$$

Resulting in

$$\begin{aligned}
\dot{\phi} &= p + \tan \Theta_0 q \\
\dot{\theta} &= q \\
\dot{\psi} &= \frac{r}{\cos \Theta_0}
\end{aligned} \quad (4.26)$$

From (4.26) we see that the NED longitudinal angles also are separated from the lateral angular rates in BODY and that the NED lateral angles are separated from longitudinal angular rates in BODY.

4.2 Aerodynamic linearized equations

It is assumed that any aerodynamic force or moment behaves linearly with any perturbation quantity, as shown below.

$$\Delta X = (\sum_i \frac{\partial X}{\partial state_i} state_i + \sum_i \frac{\partial X}{\partial \delta_i} \delta_i) = \sum_i X_{state_i} \Delta state_i + \sum_i X_{\delta_i} \Delta \delta_i \quad (4.27)$$

Sticking to general aviation literature notation, we get:

$$\frac{1}{m} \Delta X = \frac{1}{m} \left(\sum_i \frac{\partial X}{\partial state_i} state_i + \sum_i \frac{\partial X}{\partial \delta_i} \delta_i \right) = \sum_i X_{state_i} \Delta state_i + \sum_i X_{\delta_i} \Delta \delta_i \quad (4.28)$$

Where the derivatives are evaluated about a trimmed condition.

The forces and moments from chapter 4 can now be expressed by the use of stability derivatives as:

$$\Delta X = X_u \dot{u} + \dots + X_u u + X_v v + \dots + X_\delta \delta \quad (4.29)$$

$$\Delta Y = Y_u \dot{u} + \dots + Y_u u + Y_v v + \dots + Y_\delta \delta \quad (4.30)$$

$$\Delta Z = Z_u \dot{u} + \dots + Z_u u + Z_v v + \dots + Z_\delta \delta \quad (4.31)$$

$$\Delta L = L_u \dot{u} + \dots + L_u u + L_v v + \dots + L_\delta \delta \quad (4.32)$$

$$\Delta M = M_u \dot{u} + \dots + M_u u + M_v v + \dots + M_\delta \delta \quad (4.33)$$

$$\Delta N = N_u \dot{u} + \dots + N_u u + N_v v + \dots + N_\delta \delta \quad (4.34)$$

$$(4.35)$$

The perturbed aerodynamic state thus may be written in vector form as:

$$\Delta \tau = -M_A \dot{\nu} - N_A \nu + Bu \quad (4.36)$$

Where $-M_A$ is the aerodynamic added mass, $-N_A$ is aerodynamic damping and B is the actuator configuration matrix. The matrices under are given in BODY:

$$-M_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ L_{\dot{u}} & L_{\dot{v}} & L_{\dot{w}} & L_{\dot{p}} & L_{\dot{q}} & L_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (4.37)$$

$$-N_A = \begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ L_u & L_v & L_w & L_p & L_q & L_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix} \quad (4.38)$$

$$B = \begin{bmatrix} X_{\delta_e} & X_{\delta_a} & X_{\delta_r} & X_{\delta_f} & X_{\delta_t} \\ Y_{\delta_e} & Y_{\delta_a} & Y_{\delta_r} & Y_{\delta_f} & Y_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_a} & Z_{\delta_r} & Z_{\delta_f} & Z_{\delta_t} \\ L_{\delta_e} & L_{\delta_a} & L_{\delta_r} & L_{\delta_f} & L_{\delta_t} \\ M_{\delta_e} & M_{\delta_a} & M_{\delta_r} & M_{\delta_f} & M_{\delta_t} \\ N_{\delta_e} & N_{\delta_a} & N_{\delta_r} & N_{\delta_f} & N_{\delta_t} \end{bmatrix} \quad (4.39)$$

Many of the above derivatives are negligible, especially the longitudinal derivatives sensitivity to lateral states and vice versa. Table 4.2 list the most common aerodynamic

Aerodynamic Stability and Control Derivatives						
Independent Variable	Dependent variable					
	X	Y	Z	L	M	N
u	+	-	+	-	-	-
v	-	+	-	+	-	+
α	+	-	+	-	+	-
$\dot{\alpha}$	-	-	+	-	+	-
p	-	-	-	+	-	+
q	-	-	+	-	+	-
r	-	+	-	+	-	+
δ_e	-	-	+	-	+	-
δ_r	-	+	-	+	-	+
δ_a	-	-	-	+	-	-
ϕ	-	+	-	-	-	-
θ	+	-	-	-	-	-
ψ	-	-	-	-	-	-
- is normally zero or small. + value can be significant.						

Table 4.1: Table of most important stability derivatives for airplanes.

stability and control derivatives, considered for conventional aircrafts. This is taken from [19] and [5]:

Conforming to this yields:

$$-M_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.40)$$

$$-N_A = \begin{bmatrix} X_u & 0 & X_w & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & Y_r \\ Z_u & 0 & Z_w & 0 & Z_q & 0 \\ 0 & L_v & 0 & L_p & 0 & L_r \\ 0 & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & N_p & 0 & N_r \end{bmatrix} \quad (4.41)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & X_{\delta_f} & X_{\delta_t} \\ 0 & 0 & Y_{\delta_r} & 0 & 0 \\ Z_{\delta_e} & 0 & 0 & Z_{\delta_f} & Z_{\delta_t} \\ 0 & L_{\delta_a} & L_{\delta_r} & 0 & 0 \\ M_{\delta_e} & 0 & 0 & M_{\delta_f} & M_{\delta_t} \\ 0 & 0 & N_{\delta_r} & 0 & 0 \end{bmatrix} \quad (4.42)$$

$$u = \begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \\ \delta_f \\ \delta_t \end{bmatrix} \quad (4.43)$$

4.3 Combining the kinetic and aerodynamic models

By combining equations (4.23) and (4.36) we get:

$$\begin{aligned} M_{RB}\dot{\nu} + N_{RB}\nu + G\eta &= -M_A\dot{\nu} - N_A\nu + Bu \\ (M_{RB} + M_A)\dot{\nu} + (N_{RB} + N_A)\nu + G\eta &= Bu \\ &\Downarrow \\ M\dot{\nu} + N\nu + G\eta &= Bu \end{aligned} \quad (4.44)$$

And if we write the linearized kinematics as:

$$\dot{\eta} = J_\nu\nu + J_\eta\eta \quad (4.45)$$

We can write the equations on conventional state space form as:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\nu} \end{bmatrix} = \begin{bmatrix} J_\eta & J_\nu \\ -M^{-1}G & -M^{-1}N \end{bmatrix} \begin{bmatrix} \eta \\ \nu \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u \quad (4.46)$$

These equations are now suitable for analysis in order to design the flight control system. If the actuator dynamics are important, this could be implemented by augmenting the states with the actual actuator control setting δ , driven by a low pass filtered commanded input δ_c . Refer to [13] for methods on implementing this. By designing a FCS based on the linearized states one must ensure that the perturbed states does not deviate too far from the trimmed condition. Since the perturbed states are small and kept small, the first order approximation is good. The transition to new Trim conditions, being that of increased angle of attack , speed etc, is done in an "orderly fashion" keeping the contribution from the nonlinearities small.

4.3.1 Linearized Longitudinal dynamics

If we partition the 6DOF equation set into a longitudinal 3DOF set we get (assuming $I_{xz}=0$):

$$m\dot{u} = X_uu + X_ww - g \cos \Theta_0 \theta \quad (4.47)$$

$$m\dot{w} = Z_uu + Z_ww + Z_{\dot{w}}\dot{w} + (Z_q + mU_0)q - mg \sin \Theta_0 \theta + Z_{\delta_e}\delta_e \quad (4.48)$$

$$I_y\dot{q} = M_wu + M_{\dot{w}}\dot{w} + M_qq + M_{\delta_e}\delta_e \quad (4.49)$$

$$\dot{\theta} = q \quad (4.50)$$

Or in wind axes:

$$m\dot{u} = X_u u + X_\alpha \alpha - mg \cos \Theta_0 \theta \quad (4.51)$$

$$mU_0\dot{\alpha} = Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}} \dot{\alpha} + (Z_q + mU_0)q - mg \sin \Theta_0 \theta + Z_{\delta_e} \delta_e \quad (4.52)$$

$$I_y \dot{q} = M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q q + M_{\delta_e} \delta_e \quad (4.53)$$

$$\dot{\theta} = q \quad (4.54)$$

The longitudinal equations may also be augmented by a height state:

$$\dot{h} = U_0 \theta - w = U_0(\theta - \alpha) = U_0 \gamma, \quad (4.55)$$

where γ is the *flight path angle*, the direction of the velocity vector compared to NED.

4.3.2 Linearized Lateral dynamics

The 6DOF equation set can also be partitioned into a lateral 3DOF set:

$$m\dot{v} = Y_v v + Y_r r + Y_p p + Y_{\delta_r} \delta_r + mg \cos \Theta_0 \phi - mU_0 r \quad (4.56)$$

$$I_x \dot{p} = L_v v + L_r r + L_p p + L_{\delta_r} \delta_r + L_{\delta_a} \delta_a \quad (4.57)$$

$$I_z \dot{r} = N_v v + N_r r + N_p p + N_{\delta_r} \delta_r \quad (4.58)$$

$$\dot{\phi} = p \quad (4.59)$$

$$\dot{\psi} = r \quad (4.60)$$

Or in wind axes:

$$m\dot{v} = Y_\beta \beta + Y_r r + Y_p p + Y_{\delta_r} \delta_r + mg \cos \Theta_0 \phi - mU_0 r \quad (4.61)$$

$$I_x \dot{p} = L_\beta \beta + L_r r + L_p p + L_{\delta_r} \delta_r + L_{\delta_a} \delta_a \quad (4.62)$$

$$I_z \dot{r} = N_\beta \beta + N_r r + N_p p + N_{\delta_r} \delta_r \quad (4.63)$$

$$\dot{\phi} = p \quad (4.64)$$

$$\dot{\psi} = r \quad (4.65)$$

It can be shown that the equations from chapter 2 and 3, can be simplified to the two decoupled 4 by 4 systems below. This is done by using Stability derivatives:

$$A_{longitudinal} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & M_\Theta \\ X_u & X_w & 0 & -g \cos \gamma_0 \\ Z_u & Z_w & U_0 & -g \sin \gamma_0 \end{bmatrix} \quad (4.66)$$

$$B_{longitudinal} = \begin{bmatrix} 0 \\ \tilde{M}_{\delta_E} \\ X_{\delta_E} \\ Z_{\delta_E} \end{bmatrix} \quad (4.67)$$

$$x \triangleq \begin{bmatrix} \Theta \\ q \\ u \\ w \end{bmatrix} \quad (4.68)$$

$$(4.69)$$

and

$$A_{lateral} = \begin{bmatrix} Y_v & 0 & -1 & g/U_0 \\ L'_{\beta} & L'_{p} & L'_{r} & 0 \\ N'_{\beta} & N'_{p} & N'_{r} & 0 \\ 0 & 1 & \tan \gamma_0 & 0 \end{bmatrix} \quad (4.70)$$

$$B_{lateral} = \begin{bmatrix} 0 & Y_{\delta_R}^* \\ L'_{\delta_A} & L'_{\delta_R} \\ N'_{\delta_A} & N'_{\delta_R} \\ 0 & 0 \end{bmatrix} \quad (4.71)$$

$$x_{lateral} \triangleq \begin{bmatrix} \beta \\ p \\ r \\ \Phi \end{bmatrix} \quad (4.72)$$

The first being longitudinal and the second lateral motion. To avoid confusion notice the definition below:

$$Z_\alpha \triangleq \frac{\partial Z}{\partial \alpha} = Z_w U_0. \quad (4.73)$$

In the remainder of this report Z_w will generally be used, because no direct α measurement exists. It must be emphasized that in straight and level flight γ_0 is zero. Consequently, for this flight condition the terms $\sin \gamma_0$, $\tan \gamma_0$ equals zero, and $\cos \gamma_0$ and $\sec \gamma_0$ equals unity. For this particular assignment no flaps are used so all elements dealing with this can be omitted.

Chapter 5

Control Background

When designing a FCS, 6DOF, coupling and nonlinearity adds to the difficulty and complexity of designing robust controllers. Controllers often correspond to one flight condition or trim condition. Using a setpoint the approach often used is to linearized about a set of conditions. This simplifies the equations and dynamics making the design task more manageable. Linear system theory can also be applied, making the task easier than with nonlinear theory. The drawback of this approach is that global stability can not be guaranteed. In spite of this airplanes often operate within a bounded set of conditions, performing just a limited set of maneuvers. This makes the linearized approach more suitable than first assumed. For this UAV small angles and low angular rates further supports the use of an simplified model. Therefore a "simple" control law can be designed with sufficient performance. To further reduce the complexity of the system and make it more intuitive the system was divided into two subsystems, see [17] for more on this subject. Dividing the system can be done because none or little coupling exist between the lateral and longitudinal motion. As seen from the equations thrust and elevator are the main control inputs for the longitudinal system and aileron and rudder the main control inputs for the lateral system.

For the initial tests three uncoupled controllers were used for one set of conditions. The three controllers were pitch, roll and yaw controllers. By keeping the angular rates small the error not regarding coupling effects in roll/yaw was limited. Keeping in mind the limitations of the system non aggressive controllers were used to minimize the angular rates.

5.1 Simple decoupled control

Much effort has gone into the analytical linear and nonlinear models presented in Chapter 2 and 6. A Matlab/SIMULINK simulation environment with controllers were compleated. Because of uncertainty about the validity of some of the aerodynamic coefficients, a limited effort was done to tune the controllers. To ensure fulfilling the goal of having a working prototype by summer 2007, an empiric model using the basics of the setup form the analytical model was made. MATLAB Identification toolbox was used to find the system model from logged data form the second testflight, see 8. Details on model estimation can be found in [11] and [10]. This was done because the computer system was finished very late. It was believed this would be a faster solution than finding and correcting the errors in the aerodynamic coefficients. It was also believed that a working(and tested) simulation model would help finalizing analytic models in later work, even if it was "a

black box” model.

5.1.1 Empiric model

This model and how it was obtained are described in detail in Chapter 7. Stabilization and guidance was intended solved with cascade controllers. The inner loop stabilizes orientation and speed, position is controlled with the outer loop making sure all waypoints are reached. This gives an easy to understand and easy to tune setup. As seen in Chapter 8 the inner loop proved successful. The empiric model consists of two lateral and one longitudinal model each with two control inputs. For stabilizing PD controllers using Ziegler-Nichols method in simulation were used. Position control were not compleated, because no position measurements were available from the computer system [3]. Values and simulations are presented in 11, test results is presented in 8 and 12

5.1.2 Analytical models

The simplified models are based on the theory from [5], and Chapter 3 and 4. These models are described in detail in Chapters 2, 4, 6 and 10. Tuning of controllers were done using poleplacement, then simulating the analytical models with feedback in SIMULINK. Because of limited time, some work trying out alternative controllers after verifying the models persists.

5.2 Observer, and Kalman filter

The IMU and GPS used in the system includes built-in Kalman filters, this reduces the need for filters on the measurements themselves. A observer was implemented in the linear model. The observer was tuned using pole placement. Lack of time and numerical values for the stochastic parameters for the measurements, resulted in no implementation of a Kalman filter. Though Kalman filter for the model should be considered in further work on the UAV. The benefits and drawbacks of the Kalman filter is explained below.

5.2.1 Basics of a Kalmanfilter

In order to get optimal estimates of the real states of an airplane all measurements must be considered. This is necessary to minimize the effect of errors and noise in the measurements. The linear Kalman filter introduces an elegant and optimal solution to this. This enables a solution that can take advantage of the accuracy of the GPS and update speed of the IMU. Minimizing the effect of drift in the IMU and low update rate of the GPS.

5.2.2 Limitations of the Kalman filter

The Kalman filter is based on predicting the future stated of the system. High number of states and a time invariant system adds to the computing complexity of the Kalman filter. Limited computing resources therefore sets limitations for the complexity of the Kalman filter. Only a discrete Kalman filter is practically feasible to implement on the given hardware.

The stability of the Kalman filter is in practice limited to the system being observable. This is the eigenvalues of $\Phi - KD$ lies inside the unit circle.

The Kalman filter is optimal with regard to bias and minimum variance, but this is heavily dependent on stochastic parameters of the process and measurement noise. Obtaining good values of these parameters can prove to be hard.

For nonlinear systems the Kalman filter is not optimal. This is of importance since the dynamics of an aircraft is generally nonlinear, though for the air frame used in this report, nonlinearity are less dominant than usual. The Kalman filter can still give good state estimates provided a linearized system model is used. This also reduces the generality of the filter in use. Two different methods for linearization are common. One linearize about some nominal trajectory in state space, without dependence on measurement data. This is the basic linearized Kalman filter. By continuously updating the linearization with the state estimates, the Extended Kalman filter (EKF) is obtained.

5.3 Guidance

The guidance part of the system generates set points for the low level stabilizing controllers. A simple heading controller was simulated and used in the final tests. Further work will show if crosstracking or line of sight controller performs better. As no GPS measurements were supported by the computer system at the time of the final flight, the waypoints used in the live testes reduced to heading commands. Simulations and results of the flights can be found in chapter 11 and 12.

Chapter 6

Coefficients and values

In this Chapter the work done for determining aerodynamic coefficients, mass, moments and inertia is presented. These values are critical for the mathematical model, simulation and controller design. Values found spring 2007 are valid for the setup used, modifications may change this. Therefore programs automatic calculating new constants and coefficients, when changing the setup, has been made. This is done to ease development in the future. The Matlab scripts, SIMULINK diagrams and aero_engine_mass_inertia_parameters.xls all contribute to this feature.

6.1 Aerodynamic coefficients, forces and moments

Finding good estimates of an complete air frame analytically is hard and time consuming due to 3D flow, scale effects and nonlaminar flows. Looking at each surface's contribution in 6DOF analytically would take too much time in this master thesis. Further this level of accuracy was not necessary for this application. Therefore wind tunnel testing was performed, giving fast and sufficient data. In this thesis the wind tunnel at Institute for Energy and Process Engineering, was used to find the 3 moments and 3 forces acting on the air frame. From these values the most critical aerodynamic coefficients has been found using the equations from Chapter 2. The goal of the wind tunnel testing was to obtain values of moment and forces, at different flight conditions. Due to practical circumstances only measurements of static moment and forces were executed. Damping and second derivatives was not tested in the wind tunnel. These derivatives were calculated using XFOIL simulation.

The wind tunnel measurements were used to find the most dominating dynamical first order coefficients. Secondorder derivatives, contributing to longitudinal and lateral damping, were found using XFOIL. To back up the coefficients calculated from the simulations and measurements, comparison to other aircrafts was done. Thought precautions were taken because of different Reynold numbers(Rn).

Before the tests could start an estimate of the forces and moments were needed, to set the ranges on the testing equipment. Some of these estimates as Lift, Drag, etc. was done in [12]. The lacking estimates was calculated using XFOIL and basic airfoil theory form Chapter 2, and [1], [15] and [18]

6.1.1 Lift

Airplanes are designed according to different criteria; speed, lift, stability, comfort, being some among many many design criteria. Different criteria again leads to different char-

acteristics of the *static* and *dynamic stability* of an airplane. The aircraft geometry is described by certain standards, the wing configuration being the key element. This is because wings provide most of the lift needed to sustain the aircraft's weight in flight. There are several parameters that characterize wing geometry. The span, b , is the distance from one wingtip to the other. The chord c , is defined as the distance from the leading edge to the trailing edge of the wings. The aspect ratio A , is given as:

$$A = \frac{b^2}{S} \quad (6.1)$$

Where S is the planform area of the wing. Mean chord \bar{c} is defined as

$$\bar{c} = \frac{S}{b} \quad (6.2)$$

These definitions are used when faced with nondimensional aerodynamic coefficients as described in section 3.1.

Probably the most important feature of an airplane is the lift. Using 6.3 and the items of 6.1.1 the values below was obtained.

$$L = q * C_L * S, \text{ or } C_L = \frac{L}{q * S} \quad (6.3)$$

Where S is the wing area: $MAC * b$, MAC is the mean aerodynamic chord, and b is the wing span. q is the dynamic pressure: $q = \frac{1}{2} * \rho * V^2$, where V is the airspeed and ρ the density of air.

The items below list assumptions made for estimating lift.

- $C_{L_{max}}$: ~ 1.0 This was found from XFOIL simulation with Reynold number's in the range 80.000-200.000 in [12]
- airspeed: The speedrange has been estimated from testflights to be in the range $8 - 30 \frac{m}{s}$
- ρ : $1.229 \frac{kg}{m^3}$ (from www.nasa.gov)
- MAC : $0.216m$ This was calculated in [12].
- b : $1.700m$ Same as above [12].

$$\begin{aligned} L_{slow} &= \frac{1}{2} * \rho * U^2 * S = \frac{1}{2} * 1.229 * 9^2 * 1.0 * 0.216 * 1.700 = 18.27N \\ L_{fast} &= \frac{1}{2} * \rho * U^2 * S = \frac{1}{2} * 1.229 * 30^2 * 1.0 * 0.216 * 1.700 = 203.07N \end{aligned} \quad (6.4)$$

This shows that the landing speed with mass $2.0[kg]$ will be slightly above $9[m/s]$ and that 10G turns can be performed @ $30[m/s]$, as lift is concerned. This corresponds well to the tunnel testing and test flights performed. See the file `aero_engine_mass_inertia_parameters.xls` for details.

6.1.2 Control Forces

The same method deriving lift is valid for the other aerodynamic surfaces. Besides the main wing the tail, canard and wingtips produces positive or negative lift locally by changing the camber or α of the airfoil. Resulting in forces and moments acting on the aircraft. XFOIL was used to find the values of C_l in the table 6.1.2. These values were compared to the values found in tunneltesting and used in the nonlinear model. In the linear model no direct relationship between forces and moments and the numerical values in the model exists. This is because stability derivatives are used, including mass and inertia terms. To compare the values of the model with the simulated and measured values, mass and inertia was included. Table 6.1.2 presents the some of the data obtained form XFOIL simulation.

Profile	Reynolds number	α	deflection[°]	C_l	Force(lift)	Drag
Eppler 197	200.000	0.0	0.0	0.279	5.89N	0.03N
Eppler 197	200.000	7.0	0.0	1.063	22.45N	0.30N
Eppler 197	200.000	0.0	-20.0	-0.621	13.12N	0.45N
Eppler 197	200.000	7.0	-20.0	0.371	7.83N	0.49N
Eppler 197	200.000	0.0	20.0	1.370	21.91N	0.53N
Eppler 197	200.000	7.0	20.0	1.487	31.42N	0.93N
Eppler 168(7)	200.000	0.0	20.0	0.968	1.713N	0.05N
Eppler 168(7)	200.000	0.0	0.0	0.000	0.00N	0.01N
Eppler 168(7)	150.000	0.0	20.0	0.955	1.690N	0.01N
Eppler 168(7)	150.000	0.0	0.0	0.000	0.00N	0.00N

Table 6.1: Coefficients for different airfoils form XFOIL simulations.

The values in the table above are found by running simulations for Rn of 150-200.000 for EPPLER 197 and EPPLER 168. 20[°] rudder deflection was used.

6.1.3 Aerodynamic Moments

The torque form the propeller, fin, horizontal tail and control surfaces all produce moments. The motors contribution is neglectable, but the tails contribution is essential in the static stability of the aircraft. Values for different α and β angles are presented in the aero_engine_mass_inertia_parameters.xls file. The moments caused by controlsurface deflections are also presented here. These moments are the basis for the stability derivatives making up the linear model.

6.1.4 Drag

Finding an good estimate of the drag was difficult because many factors are involved. Especially the contribution from turbulent flow around fittings on the air frame is hard to find. One great contribution to drag is induced drag as a result of the lift. This can easily be found using laminar theory. An estimate of the induced drag was found by finding the L/D (Lift/Drag) ratio for the Eppler197 profile in XFOIL. Values for Rn in the range 150.000-200.000 was tested and α of 0-2° giving CL -values form 0.27-0.59 and L/D from 20-40. In Stable level flight the lift is equal to the weight $2.0kg * 9.81 \frac{m}{s^2} = 19.62N$. The drag contribution from the induced drag alone is therefore less than 1.0N, worst case. The total drag of the air frame should be less than twice the drag of the wing, looking at similar aircraft. This gives a total drag of less than 2.0N, worst case. The measured value

@ cruise speed and an α of 6° was 0.66N . This indicates that the power consumption should be less than 25W , see chapter 9. Using this power consumption gives a flight time with $4*1000\text{mAh}$ batteries, of almost 2 hours worst-case. The maximum obtainable battery capacity with current technology of $8*2100\text{mAh}$ corresponds to almost 10 hours @ $17\frac{\text{m}}{\text{s}}$.

It is also worth noticing that the L/D -ratio increases as the wings α is increased, for reasonable values. This means that the induced drag will increase less than proportional with the weight of the air frame. Taking this into account an maximum flight time of over 10 hours should be obtainable.

6.1.5 Thrust

Static thrust was measured using testing bench. Because thrust is dependent on the airspeed, an estimate of the thrust in a moving airstream a propeller calculator was used. With the calculator propeller pitch, diameter, airspeed and rpm. can be adjusted. Outputs are thrust and power required.

The propeller calculator used was **ExtendedPropSelector.exe**, this is a free program downloadable from internet(also supplied on the DVD). Earlier testing [12] has proved this program to give 20% too optimistic values. This accuracy is sufficient for initial estimates. The table below gives calculated estimates and measured values of thrust, with different powerinput and airspeed for an $12*8$ inch 2 blad propeller. For more on testing of motor and propeller, see chapter 9. In addition to this, the motor propeller combination was tested in the wind tunnel, with the air frame. The graph in figure 6.1 show the values found.

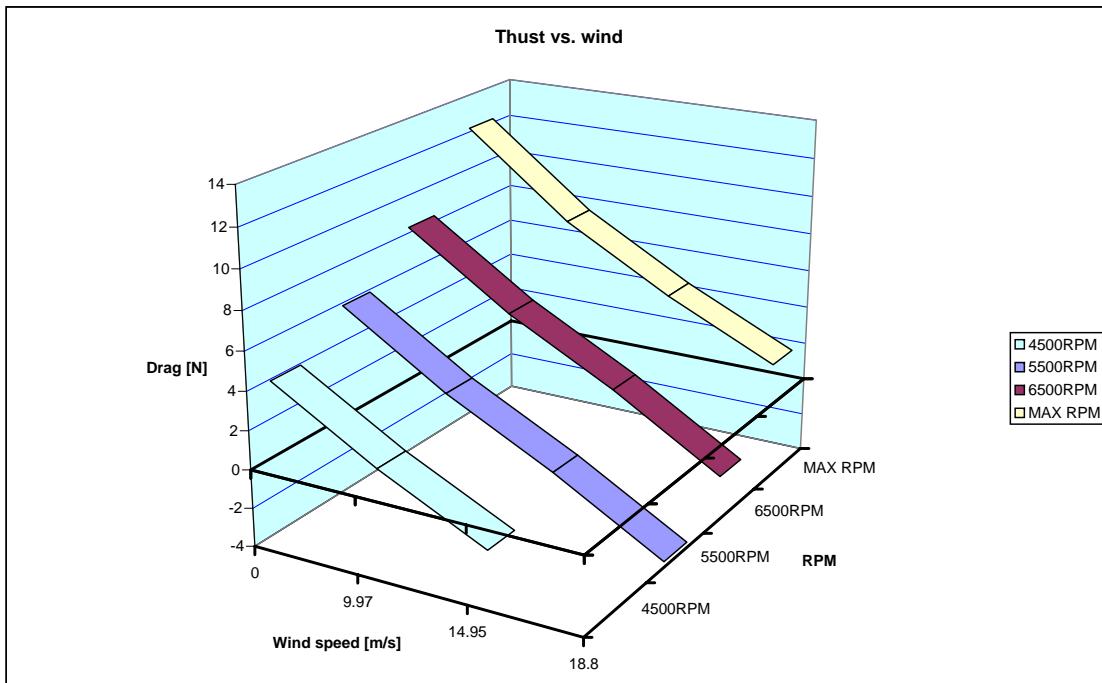


Figure 6.1: Thrust vs. drag at different airspeed

The measured value shows that the corresponding calculated value is 20-25% too optimistic. The reason for this can be many, one being that the propeller used is not ideal.

Airspeed/Power	50W	100W	200W	350W	measured 228W
$1 \frac{m}{s}$	5.0N	8.0N	12.7N	18.4N	10.8N
$10 \frac{m}{s}$	3.6N	6.4N	10.9N	16.5N	-
$20 \frac{m}{s}$	1.6N	3.8N	7.8N	12.9N	-
$30 \frac{m}{s}$	-0.4N	1.3N	4.6N	9.0N	-

Table 6.2: Power and thrust data for different airspeed found using ExtendedPropSelector.exe

Comparing the values for the calculated and measured maximum thrust at zero wind speed, the calculated values at different wind speeds was corrected.

6.2 Mass and Moment

Determining mass and moment for all three axis was straight forward. The mass was found by weighing. Y-axis symmetry gives no moment about the X-axis, that is CG and CL coincide along the X-axis. Balancing the plane about CG gives the CG's X location. The moment about the Y-axis was found by multiplying the arm CG-CL by the weight of the air frame. This CG-CL margin gives rise to the trim about the Y-axis, and ensures pitch stability. The calculations in the next section also determines the CG location.

6.3 Inertia

The inertia deals with the mass distribution in the air frame. No general function for the mass distribution of the air frame exists. All components mass and center of mass(XYZ-location) was put into a spreadsheet. From this data, mass, moment and inertia was found using the relation $I_a = m_a * d_a$ (where d_a is the distance from CG of the total system). The spreadsheet also adds to the flexibility of the system, as new hardware can be added by just entering the mass and components XYZ location. The new inertia and mass matrix is automatically computed in the spreadsheet. For most airplanes only I_{xx} , I_{yy} , I_{zz} and I_{xz} is of importance, because of XZ-symmetry gives $I_{xy} = I_{yz} = 0$. This is also valid for this airplane.

Chapter 7

Software

A variety of programs have been used to develop and test the different parts of the system. An overview of these programs are presented here.

7.1 Matlab and SIMULINK

The backbone of effectively designing controllers for the UAV is a good simulation environment. Therefore MATLAB and SIMULINK was chosen for data analysis and simulation. Using powerfull known software with automatic Real Time Workshop (RTW) code generation reduces time spent on implementing controllers. Additionally this eases the work with implementing observers, models, filters, controller etc. on the PC104 computer system.

Windtunnel testing and analytical analysis were the basis for obtaining aerodynamic coefficients and stability derivatives. This data should be sufficient to build the simulation environment. However lack of corelating data or similar projects has made this difficult, because small measurement errors can lead to huge errors in the simulations. Literature found on the subject has generally been concerned with lager aircrafts, although the coefficients are dimensionless MAC and Re numbers differ. Therefore coefficients can not be directly compared, with the numerical values found in this master thesis. Finding expertise on this subject at NTNU has proved to be difficult. Therefore it was decided to log input and output from real life test and base model approximations on these. This gave data and models to verify the coefficients and models obtained analytically. In total two different approaches and four models has been developed. This has taken most of the time available, in effect the correlation of the four models needs more work. Altought the analytical models are not playing an important part in the working system at the moment, they are important in that they set the basis for understanding the behavior of the UAV, thus giving the needed foundation to make the empirical models work.

7.1.1 Identification Toolbox

The identification toolbox in MATLAB is a powerful tool for obtaining models from input and output data. The graphical interface adds to the efficiency and understanding when finding models. The main window is illustrated in figure 7.1.

The empiric model is based on actual input and output data, but gives little concrete understanding of the dynamics of the aircraft. Despite this it has proved a powerfull

and effective tool in developing a working prototype. The drawback of this model is that compensation for nonlinearities, trim and alternations in the setup, is not as easy to find. This is because the model don't have a direct relationship with the analytical models.

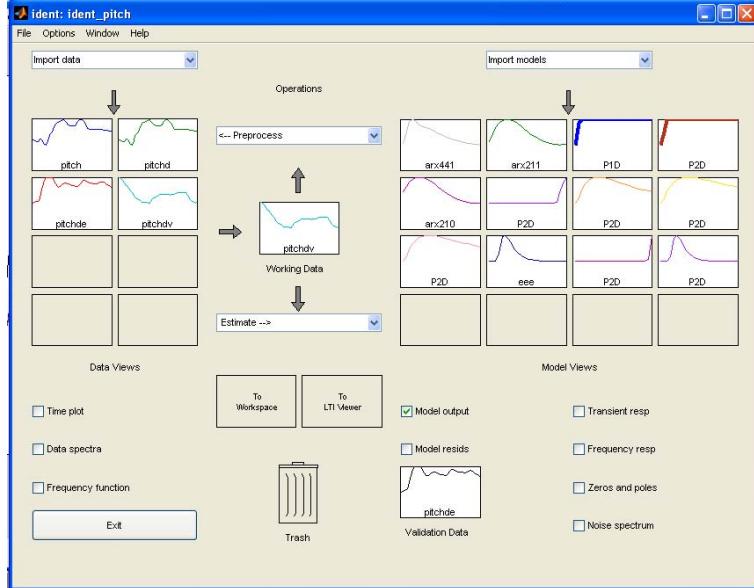


Figure 7.1: Main window of the identification toolbox

The models obtained from this analysis were subjected to the total range of data, tuned, and checked for validity. When satisfactory accordance between the estimated and real model was derived, the models were implemented in SIMULINK. Figure 7.2 show the real data compared to the estimated model. Tuning of controllers was then performed using the Ziegler Nicold method.

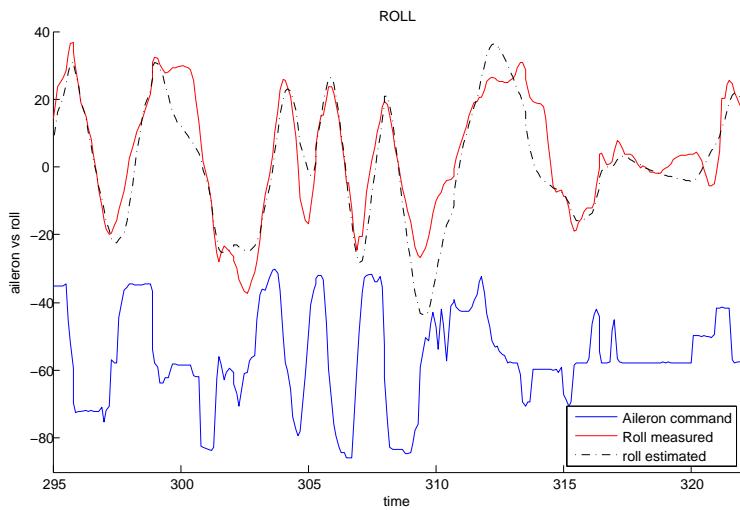


Figure 7.2: Real values, model estimate and input

7.1.2 AeroSim

This model is based on the AeroSim toolbox developed at M.I.T. It is based on standard blocks in SIMULINK, and therefore supports Real Time Workshop C/C++ code generation. The toolbox is not standard in MATLAB and is therefore supplied on the DVD. The AeroSim model is huge and not suitable for presentation without running in MATLAB/SIMULINK. To explore this model use the “look under mask” feature model in SIMULINK. Additional simulation blocks have been developed to support the electrical propulsion system. The AeroSim simulations are based on the aerodynamic dimensionless coefficients found from the test data. The aerodynamic coefficients, mass and inertia terms were entered into AeroSonde_CyberSwan_template. Running this template generates the necessary files for the SIMULINK simulations. Figure 7.3 shows the main window of the simulation environment used.

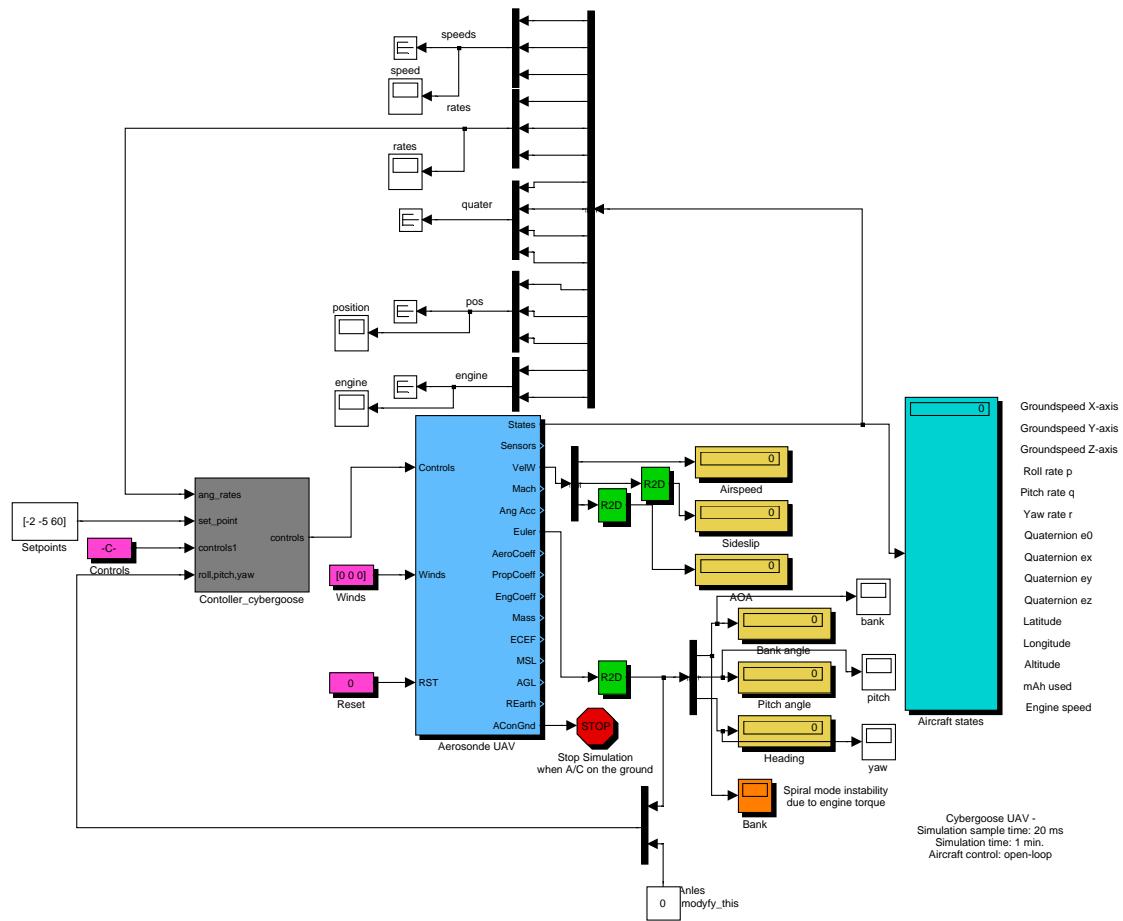


Figure 7.3: Main window of AeroSim environment

7.1.3 Linearized Stability Derivative Model

This model is based on two decoupled linearized models from [5], one lateral and one longitudinal. The model uses the stability derivatives computed from testing, found in the Excel file *aero_engine_mass_inertia_parameters.xls* file on the DVD. As mentioned above the stability derivatives have not been verified and therefore this model should be used

with care. The linearized model is valid for small perturbations($\pm 10 - 15^\circ$) from the equilibrium point. The linear model implemented in SIMULINK is illustrated in figure 7.4.

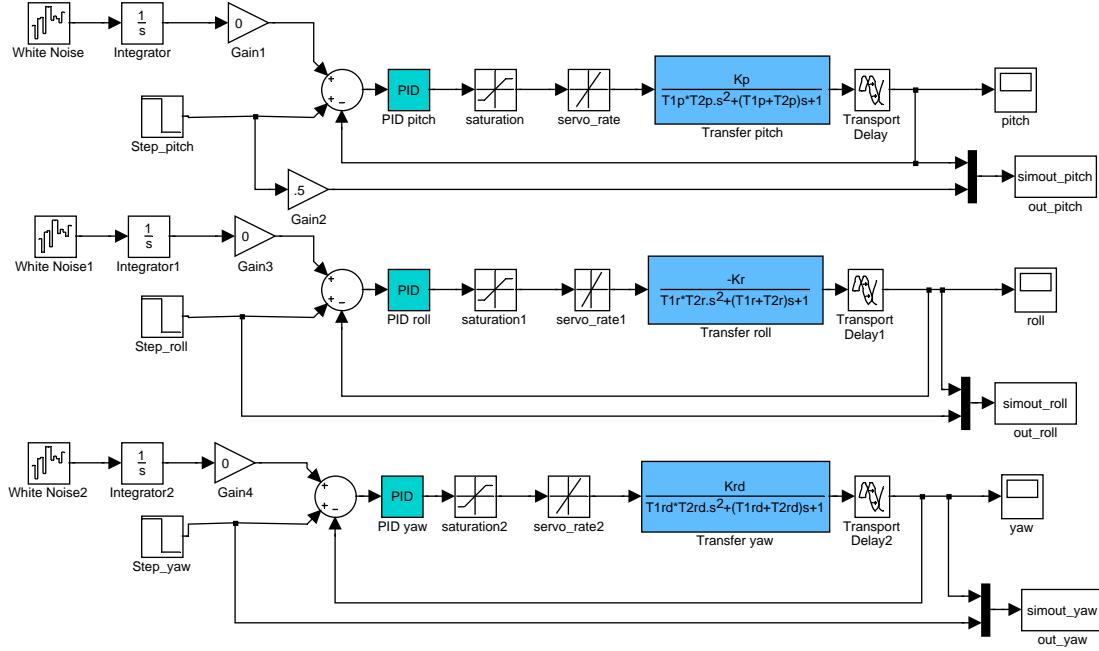


Figure 7.4: SIMULINK diagram of the linear model

7.2 RealTime Workshop

Real-Time Workshop generates and executes stand-alone C code for developing and testing algorithms modeled in SIMULINK. The resulting code enables rapid prototyping, and hardware-in-the-loop testing. Using this powerful tool in combination with Matlab and SIMULINK saves time developing software for the target. See [3] for a detailed description on how to set up RTW with SIMULINK and generate code.

7.3 PCM and Fail Safe

A 2[kg] airplane traveling at speed up to 30[m/s] is not at toy. Safety has been important throughout the work, the Futaba PCM1024 system significantly contributed to this during the flight tests. The R149DP PCM receiver used has a feature called Fail Safe or F/S. This allows the user to program the receiver prior to flying. In case radio signals are lost the receiver set the servos to the predefined position. For stable aircrafts this allows rudder commands ensuring safe decent. If no F/S system is present the outcome can be much worse. Below are the setup of the 9C transmitter mixes and rates including F/S. The main contribution to noise in an electric aircraft is the switching of the Electronic Speed Controller (ESC) and motor. F/S setting the throttle to 0%, helps restore radio contact, in case radio contact is lost. In addition slight positive elevator and rudder to one side keeps the plane in a controlled circular decent. The F/S feature was set to put the plane in manual mode if something happened while flying autonomously.

The transmitter and receiver was programmed as shown below:

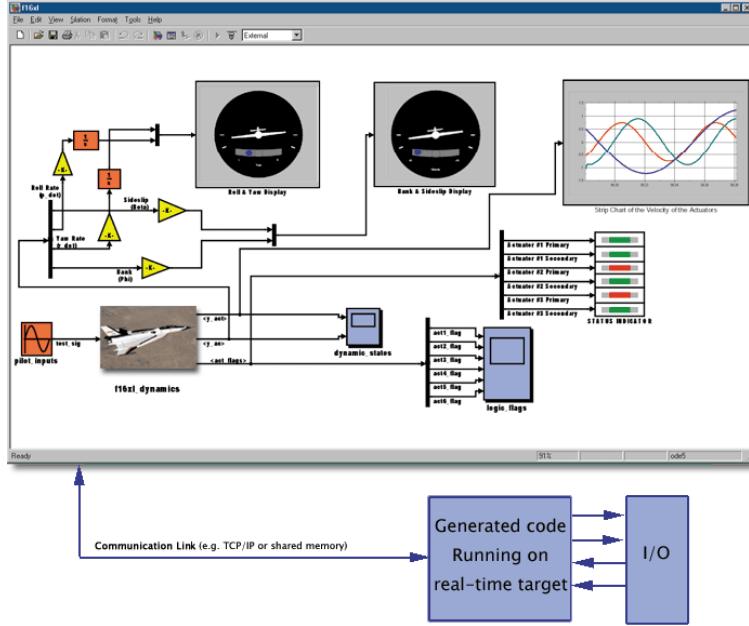


Figure 7.5: RTW, source <http://www.mathworks.com>

- Mix1: Aileron Differential, channel one and seven mixed proportional to each other because two aileronservos are used.
- Mix2: V-Tail Mix, elevator and rudder mixer turning both servos for elevator and rudder commands. This mix works the same way for an A-tail.
- EndPoint: Endpoint was set up so all servos had same travel, $\pm 15^\circ$.
- SubTrim: Setting all servos initially to zero.

The figure 7.6 and 7.7 show the mixes used.

7.4 Additional Software

In addition to the software described above XFOIL was used for foil simulations.

- ExtededPropSelector.exe was used to calibrate the motor propeller data found from test rig experiments.

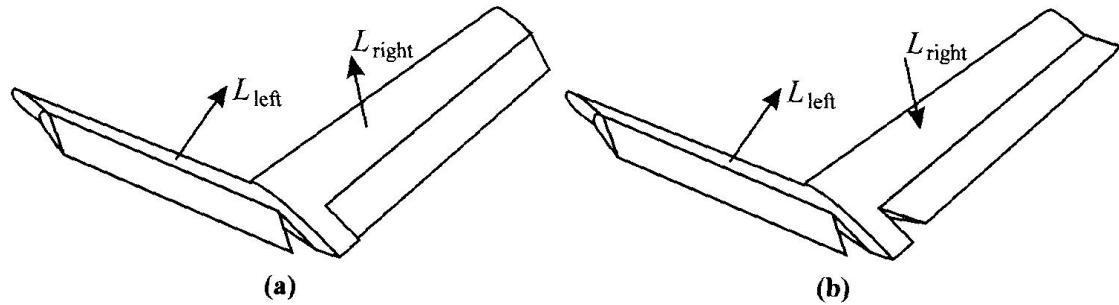


Figure 7.6: a) show V-tail mix response for positive elevator and b) port rudder, source [5].

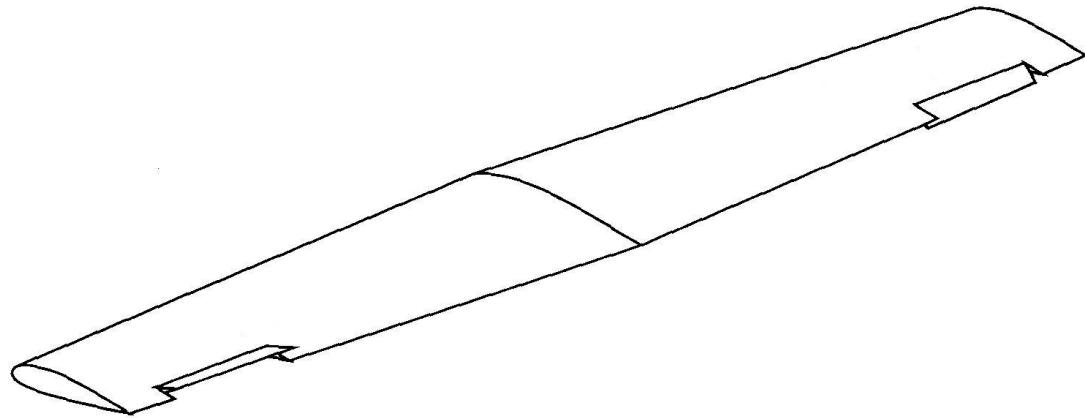


Figure 7.7: The aileron differential makes it easy to use two servos on one channel, the picture shows response from port stick, source [5].

- HexTronik ESC Config Setup.exe was used to program timing, rotation direction, and brake setup of the HXT036 motor controller. This program was also used to obtain real time RPM data for the motor testing, see chapter 9 and the rig test video on the DVD.
- The Feiago Power Analyzer comes with a RS232 interface and the program PowerView. This allows continuous logging of voltage, current and power. Graphs of these properties can be saved or the data can be saved to .txt-file. The PowerAnalyzer was used for testing the motors and propellers, see chapter 9, graphs using PowerView are included in the same chapter, under the battery testing section.

Chapter 8

Flights

This project's main goal was to implement a FCS on an actual aircraft, and test the system. Going from theory to a working system usually involves a lot of testing and trouble shooting, this assignment has not been any different. To reach this goal some simplifications have been made. Some origin from limitations in the sensor system, but the main limiting factor has been workload and the great span of tasks solved. The control and guidance system has for this reason been kept as simple and intuitive as possible. When working with the theory a complete nonlinear model was described. This model was then linearized and some states was approximated (i.e. $\alpha = w/U_0$). Further challenge was encountered because not all aerodynamic properties were feasible to find or measure experimentally. There has also been a challenge that most literature is concerned with bigger aircrafts, remembering the difficulties with scaling in Chapter 2.

Last but not least this project has involved three subsystems and three students working with these. For the total system to work all students had to solve their tasks within the defined deadlines. As work progressed not all reached this goal limiting the progress of others.

The initial test was conducted with a simplified model, because of this. Although being sufficient to solve the stabilizing, heading and surveillance task.

8.1 Main flight conditions

Using the special cases Steady Level Flight from Chapter 2 a flight path was followed. In [5] a guideline of $\pm 15^\circ$, for deviations from the setpoint are given. This ensures the errors made with respect to the linear model are sufficiently small to sustain stable flight. A study of the different dynamics of the nonlinear and linear model for a comparable aircraft was conducted in the report [2]. This confirmed the guidelines above.

In addition to analytical computation, windtunnel testing, simulation and four flight tests has been conducted. The first two tests were done with manual control. The second flight was conducted to collect data for the identification analysis. The third flight was the first flight with feedback, in this flight the goal was to have the UAV follow an specific orientation and speed. In the final fourth flight the plane followed a preassigned heading.

8.2 First flight (Manual)

Date:10.02.2007 Location:Jonsvannet Description:Manual control Weight:1845gram
Weight:1350grams

The first powered flight was conducted in February 2007. In total two flights and landings

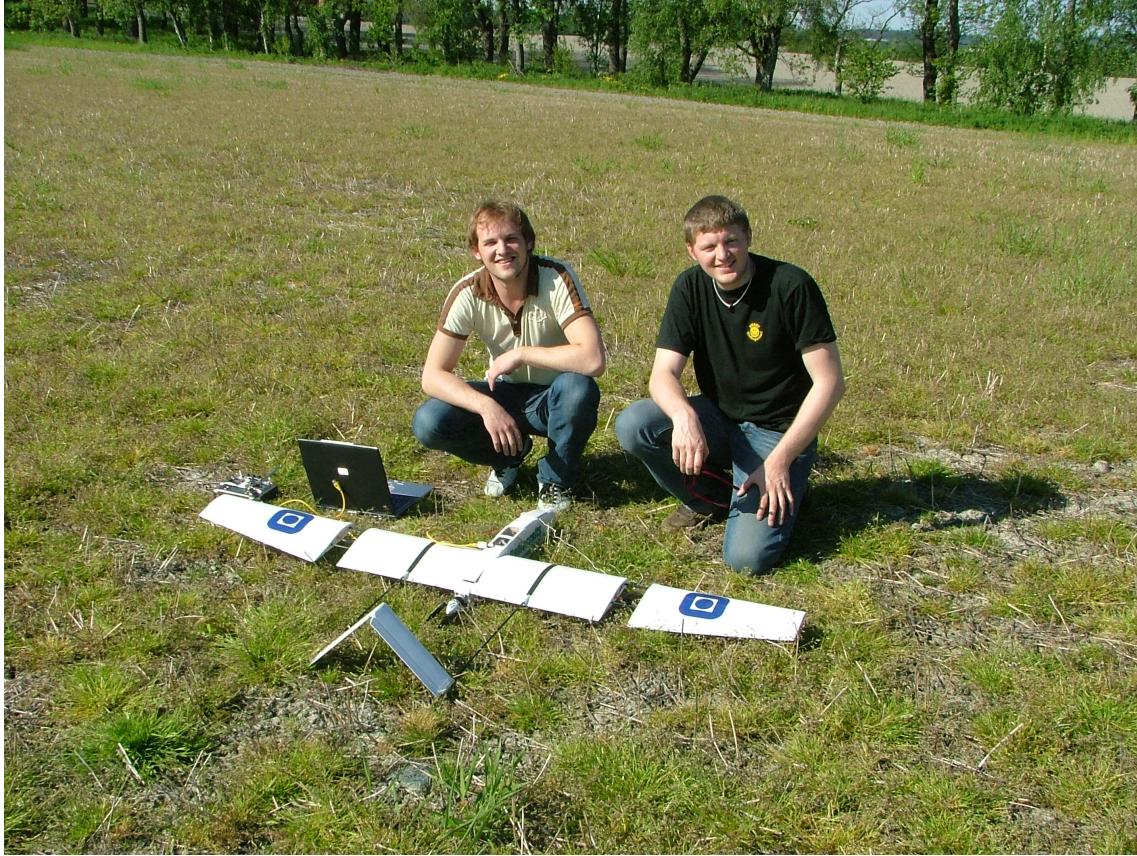


Figure 8.1: UAV and crew, testflight2

was done this day. Video of the second flight and landing is supplied on the DVD attached. The UAV was hand launched by a second person, the same person did the filming, therefore no video of the launch exist. As the video show the PR.25 have plenty of power to climb effectively, Looking at the video one also notice that the UAV is quite fast compared to other model airplanes, this is because of the low drag, and linear, smooth control characteristics. Some planes gets “nervous” when flown fast, the UAV was a joy to fly. When trying to land, the landing was overshot several times before successfully landing. This was because of very low landing speed and very low drag, with no airbrake it is necessary to get a low and slow final with the UAV. On one of the landings the UAV touched down once before before landing, timing the video and measuring the jump suggest a landing speed of 9[m/s]. This is within the specification from the design criteria in [12]. Judging from this testflight the design makes a UAV which flies very well at all speeds. The rudders were smooth and did not get nervous at high speeds, thought plenty of control authority was present at low speeds.

8.3 Second flight (Data collection)

Date:04.06.2007 Location:Fossegrenda Trondheim, Description:Manual control, with datacollection

Weight:1754grams

The second flight was conducted in June 2007. The objective of this flight was to collect

input data from the receiver, and IMU data from the physical system. Performing oscillating maneuvers in with aileron, rudder, elevator and engine/thrust. Figure 8.2 show the Euler angles collected. In addition to this accelerations, angular rate and receiver/servo commands were logged. This data was later analyzed using Matlab Identification toolbox, first and second order systems were constructed. The models obtained form this data are not analytical, but can be used to verify the analytical models. This has not been done as the data collection came too late in the project, because the computer system was not working before this. More on the empiric models is presented in Chapter 11. Video of the flight is supplied on the DVD attached. Figure 8.1 show the crew on test flight2.

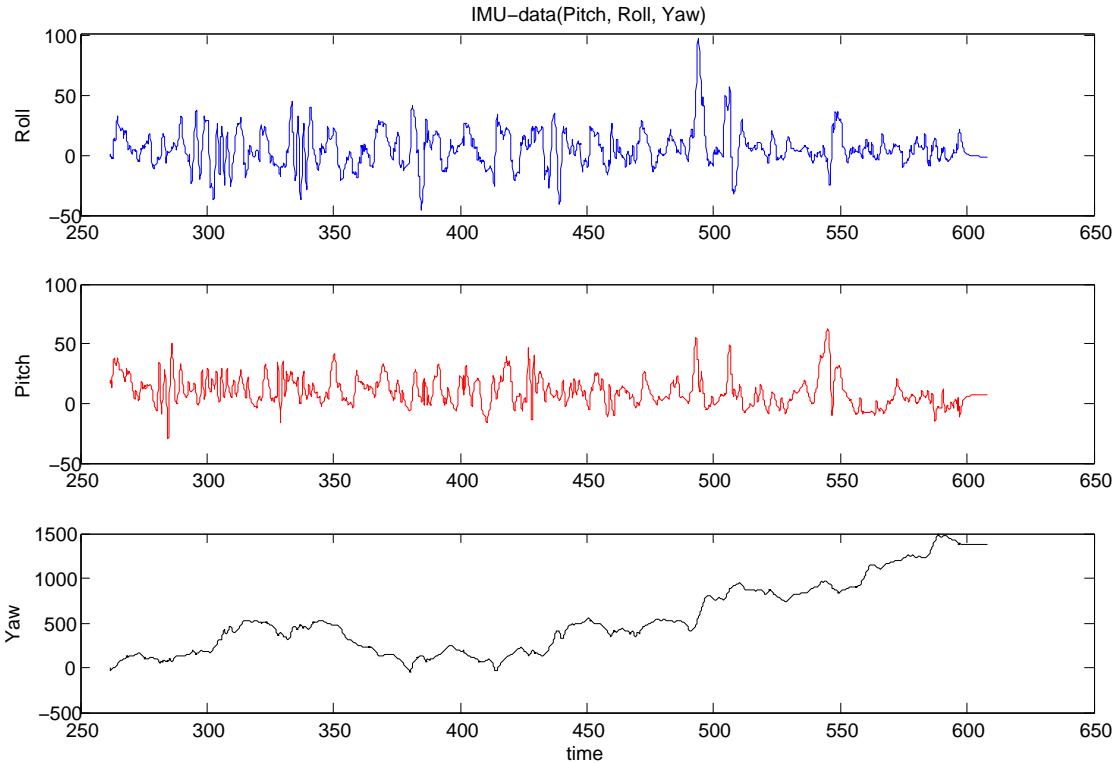


Figure 8.2: IMU Euler angles

8.4 Third flight (with feedback)

Date:08.06.2007 Location:Fossegrenda Trondheim, Description:Manual control on launch and landing. autonomous stabilizing control

Weight:1846grams

Based on the empirical model a control law was constructed, see Chapter 5 and 11. The test flight was done in June 2007. Video of the flight is supplied on the DVD attached. This video describes the flights and adjustments in detail. The video shows that steady flights were performed with great success. Manual control allowed the system to be tested outside the equilibrium. Switching to autonomous flight with bank and/or pitch enabled observation of the controllers performance to be evaluated. Bank of $\pm 45^\circ$ and pitch of $\pm 30^\circ$ were tested with success. The throttle was under manual control during the flight to test the robustness of the controller with respect to speed, and throttle input. Data from the IMU and controller were logged through the flights, this data is presented in



Figure 8.3: UAV and crew, testflight3-4

Chapter 12. Fine tuning of the control law and verification of the analytical models can be done with this data, in a follow up project.

8.5 Fourth flight (Heading control)

Date:08.06.2007 Location:Fossegrenda Trondheim, Description:Manual control on launch and landing, autonomous stabilizing heading control

To complete the AFCS heading control was added to the stabilized system from the third test. The flights showed that the UAV could follow a heading reference. As with the third flight throttle was under manual control to test the robustness of the controller. The controllers performed satisfactorily in spite of wind gusts and yaw speed. These flights completed the goal of this project. The UAV system is capable of autonomous flight through feedback. Position control has not been tested as the computer system did not provide position measurements from GPS. Video of the fourth flight is supplied on the DVD. Figure 8.3 show the UAV and crew from the third and fourth testflights.

Chapter 9

Electronics Development

This Chapter presents the work done in developing testing equipment and the electronics included in the FCS.

9.1 Motor tester

The key idea for this test equipment was measuring the performance of motor and propeller combinations. This was necessary because the manufacturer doesn't provide this information with any accuracy. The collected data was analyzed in Matlab. In this way plots of torque, power, and efficiency could be evaluated for optimal performance and endurance. In addition to the PR.25 engine half a dozen other motors were tested for further development and engine choice. These and key specifications are listed in table 9.1. All the motor data collected is stored in motortester_new.xls, then exported to mat-file for analysis in Matlab. The figure 9.2 shows a 4 degree polynomial approximation of all motor properties, of the Dualsky PR.25. The testing has given concrete guidelines for an optimal engine choice. The data also provides information for the propeller(load) choice.

Motor	Max power	Max thrust	Weight	KV
HXT24	80[W]	3.73[N]	24[g]	1300[RPM/V]
Dualsky XM200	97[W]	3.94[N]	31[g]	1500[RPM/V]
Dualsky XM300B	120[W]	5.90[N]	50[g]	1200[RPM/V]
HXT28-30Cal	170[W]	8.24[N]	69[g]	1050[RPM/V]
Dualsky XM400C	206[W]	9.74[N]	65[g]	1200[RPM/V]
HXT25D28	215[W]	6.67[N]	70[g]	1050[RPM/V]
KD28-22	303[W]	12.75[N]	85[g]	1440[RPM/V]
Dualsky PR.25	320[W]	14.60[N]	120[g]	950[RPM/V]
HXT 42-50	618[W]	28.40[N]	239[g]	700[RPM/V]

Table 9.1: Different motors tested, with key data parameters

9.2 The test rig

The rig works by measuring thrust and torque with two modified IKEA weights. Using the loadcell and calibrating circuit from the off the shelf weights, time was saved. The mechanical device was designed and sent to the ITK-workshop for construction, only leaving the assembly to be done. A Feiago Power Analyzer was used to measure voltage

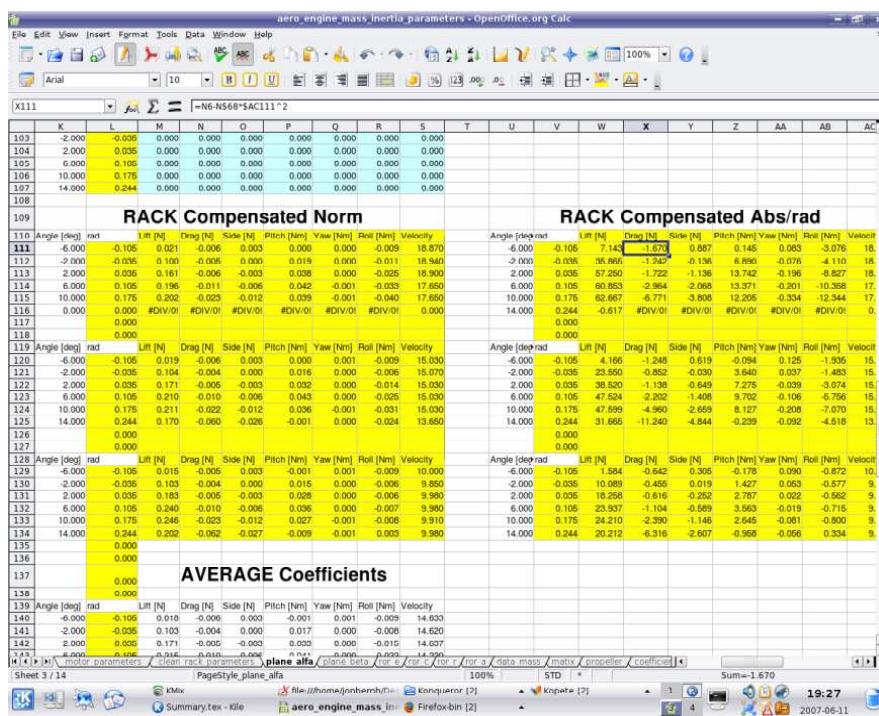


Figure 9.1: Spreadsheet with all tested motor parameters

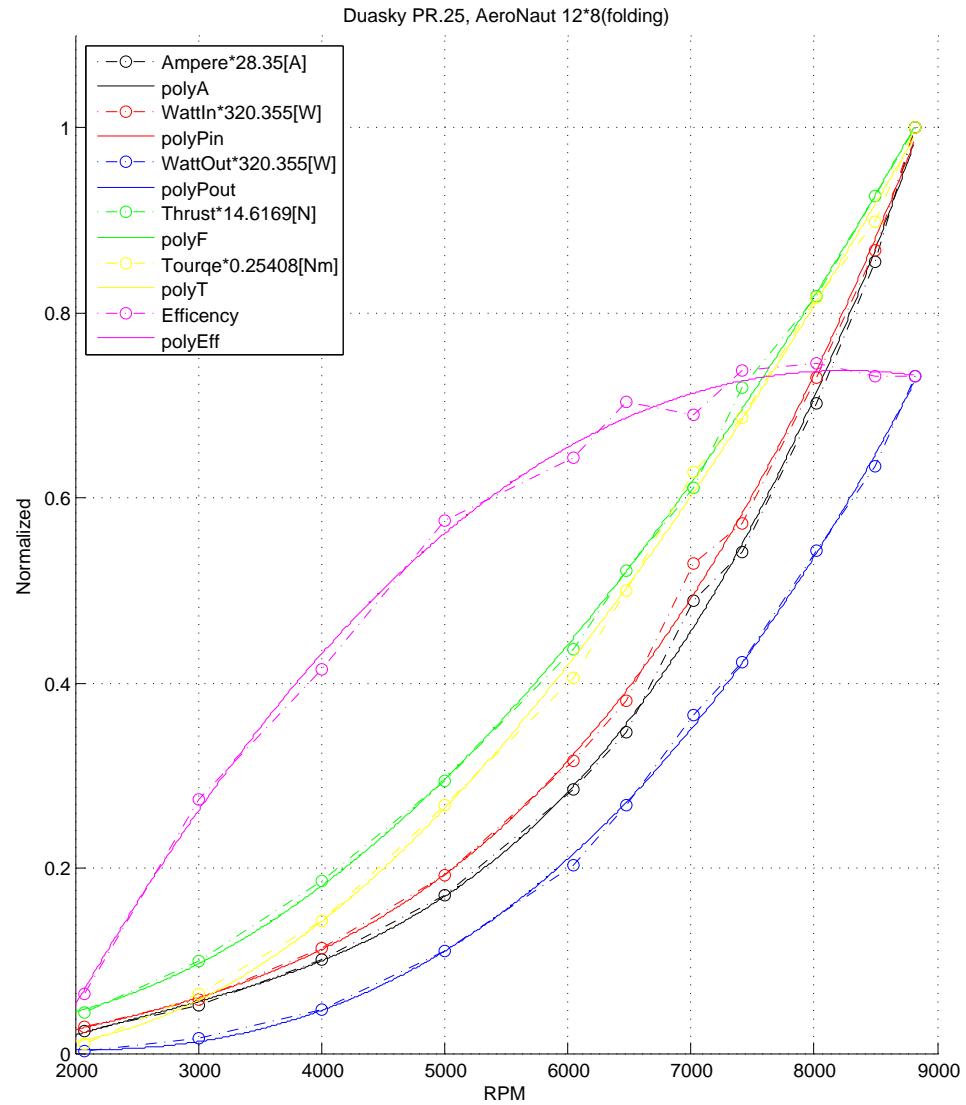


Figure 9.2: Engine properties, measurements and approximation

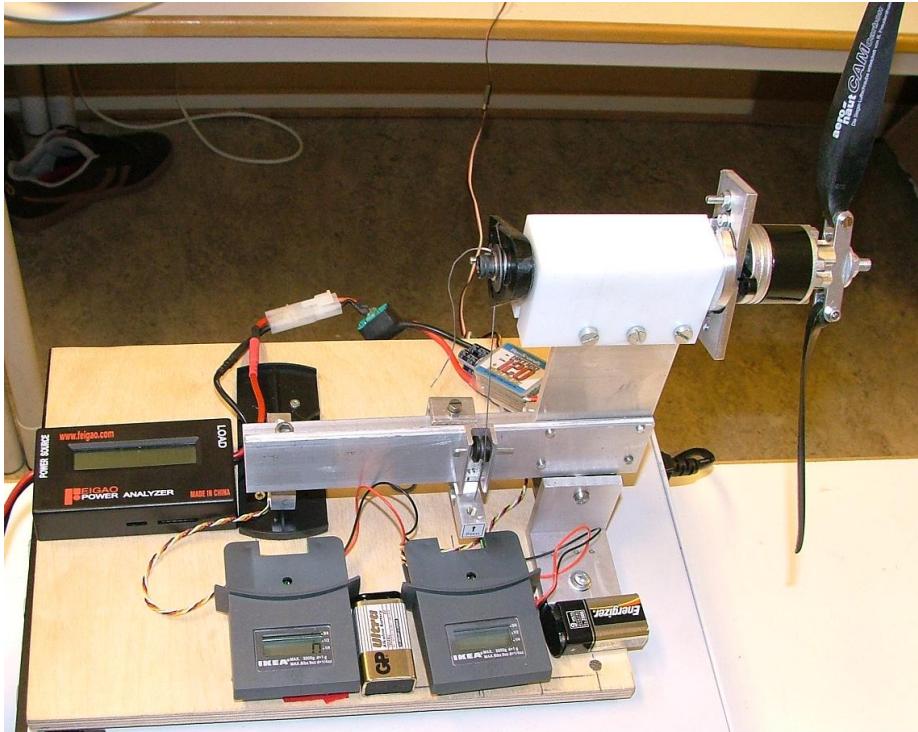


Figure 9.3: Test rig

and ampere, thus giving the input power. A HexTronik USB dongle with additional software, see 7, was used to measure RPM. As power source the desired Li-Po-batteries was used, see section 9.3. See figure 9.3 for rig setup.

Using the equations 9.1-3, efficiency was calculated. The rig proved to be a vital tool for finding the optimal motor, battery and propeller combination for duration flight.

$$P_{in}[W] = Voltage[V] * Current[A] \quad (9.1)$$

$$P_{out}[W] = torque[Nm] * \omega[\frac{1}{rad}] \quad (9.2)$$

$$\eta = P_{in}/P_{out} \quad (9.3)$$

9.3 Battery tests

To verify the performance of the Li-Po-batteries intended for the project, they were tested. There are three properties of special interest: Max discharge current as a function of voltage, maximum continuous discharge and total capacity. The two first properties are usually given relative to the mAh-capacity of the battery, called C-rating. One C equals the capacity of the battery, for a 600mAh cell $1 * C = 1 * 600[mAh]$.

To find the desired properties two tests were performed.

1. Short (30-40seconds) discharge @ 3Volts/cell (The safe voltage limit without dam-

aging the cells) using Feigao Power analyzer

2. Duration (until empty) discharge @ Ampere=8-10C, = 100W (When running the batteries harder than this, excessive heat is generated) using Feigao Power analyzer
3. Duration discharge was also performed using Shulze charger (This make is known for its precision), the values found are presented in table 9.3

For the chosen HXT1000 battery short discharge gave values of 20C, this was 2/3 of the claimed C-rating. Duration discharge also showed that the battery capacity and continuous discharge C-rating was less than claimed. The test showed capacity of 907mAh maximum, while the distributor claims 1000mAh. The table 9.3 show the values found from testing. Test of other batterypacks (Thunder Power) showed better accordance between claimed and measured C-rating and capacity. Graphs of these tests are included on the DVD attached to this report.

Late in the work the distributor of the batterypacks informed that there had been problems with the production of the HXT D9 series batteries. This may explain why the test results showed lower specifications than claimed.

Make	Capacity calmed	Cells	Capacity found	Percentage	Production	Cycles
HXT	360	3	311	86.4	6L	10-15
HXT	450	2	431	95.8	6L	5-10
HXT	620	2	542	87.4	7A	10-15
HXT	620	3	558	90.0	6L	15-20
HXT	1000	3	907	90.7	7A	3
HXT	1000	3	640	64.0	6K	10-15
HXT	1000	3	785	78.5	6FHXT	10-15
ThunderPower	1320	3	1250	94.7	?	15-20
HXT	1700	3	1392	81.9	7CGR	4
HXT	2200	4	1918	87.2	7A	2-3
HXT	3250	3	2815	86.6	7CGR	3-4
HXT	3250	4	2856	87.9	7CGR	2-3
HXT	4100	4	3643	88.8	7CGR	2-3

Table 9.2: Table of batteries tested with data obtained

Caution: The voltage of Li-Po cells should not be discharged under 2,75V, experience shows that discharging @3V causes the battery to heat up quickly. Loading the cells harder than this lowers the voltage, saturating the power delivered. Li-Po-cells has been known to catch fire if not handled with care.

Figure 9.6 show the size of four HXT1000 batteries compared with three ThunderPower Pro Lite 1320 batteries. These two piles equals in calmed capacity, but the TP batteries are just 63% of the weight of the HXT's and as the picture shows the TP's size is just 3/4 of the HXT. Despite this the HXT's are sufficient for testing and a little over $\frac{1}{2}$ the price of TP, this is why they were chosen.

9.4 Servo tester

The servo tester was developed for making a model of the servo dynamics and finding the worst case current draw. Initially the circuit in figure 9.7 was drawn and intended to

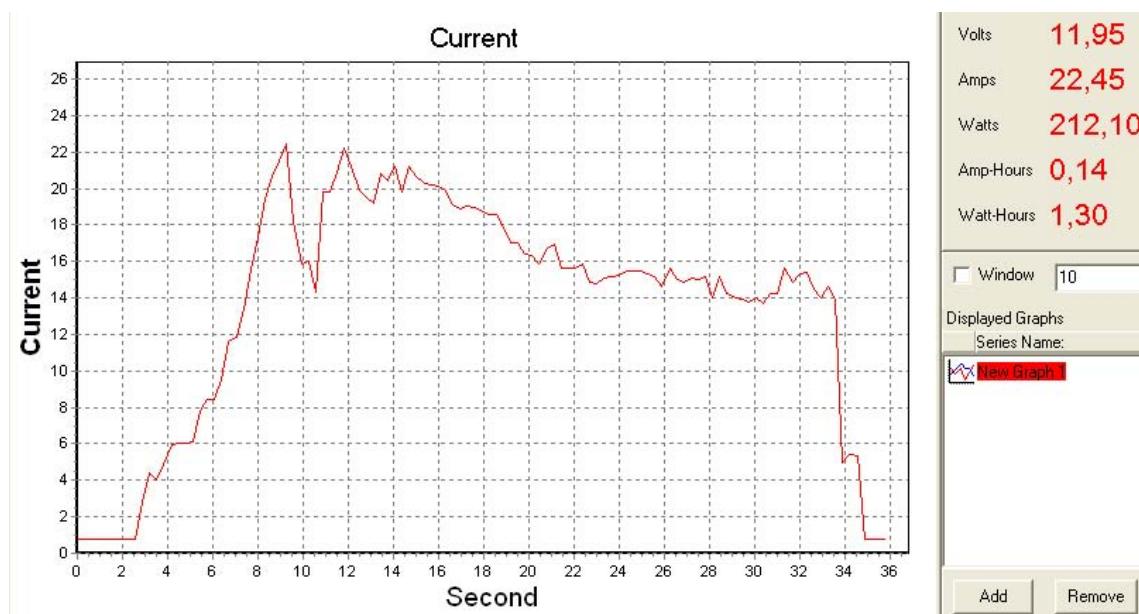


Figure 9.4: Short discharge test @9V

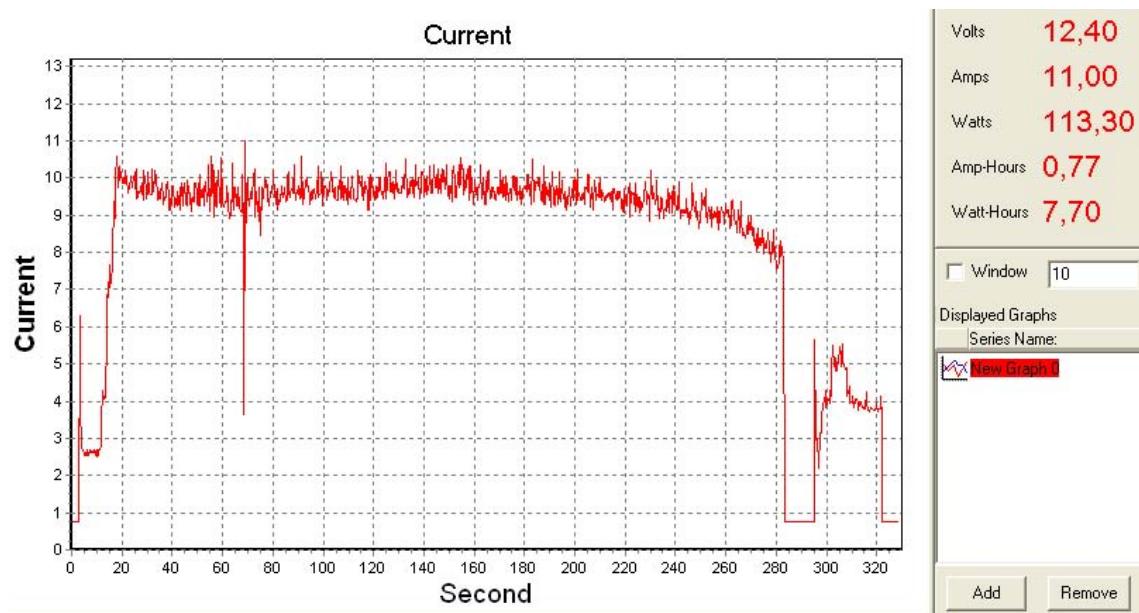


Figure 9.5: Duration discharge test @100W



Figure 9.6: ThunderPower vs. HXT batteries

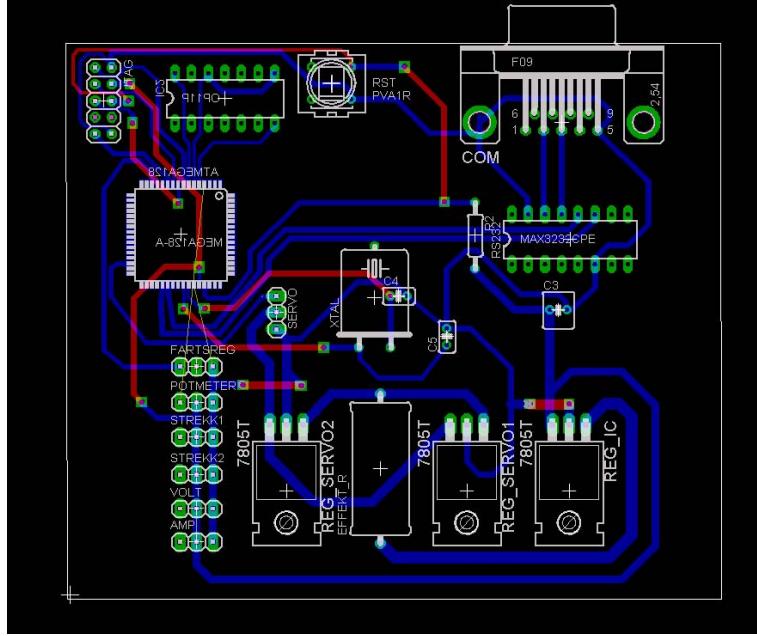


Figure 9.7: Initial servo tester board

be used, but because of lack of time the simpler version in figure 9.8 was used. Using a oscilloscope and a switch channel from a Futaba 9C, transmitter dynamics were found. For measuring the stall current an external power supply with ampere meter was used. Eagle Layout Editor was used for drawing the initial circuit, and laying out PCB board routing.

9.4.1 Resolution

The angular resolution of an RC servo is typically $\frac{1}{5}[\circ]$. This was confirmed during testing fall 2006 in the work [4]. The testing showed small variations from brand to brand, with worst case angular resolution of $\frac{1}{2}[\circ]$. For a servo arm of typically 15mm the “command” error from angular resolution is 0.12mm. This is less than the mechanical resolution of the system.

9.4.2 Dynamics

Figure 9.9 shows the step response of a HS-81 servo @ 5V. The tests showed that the dynamics of the Hi-tec HS81 servos can be simplified to $\frac{dPosition}{dt} = constant$. The biggest error being a slight speed reduction when the servo approaches the reference command. The constant was found to be 530°/s @ 4.8V and 695°/s @ 6V. Looking at the graph no significant overshoot or other dynamics was discovered. The current consumption had a big spike just in the beginning of the step response as would be expected as the acceleration command here equals ∞ . This is normal and judging from experience, does not create problems.

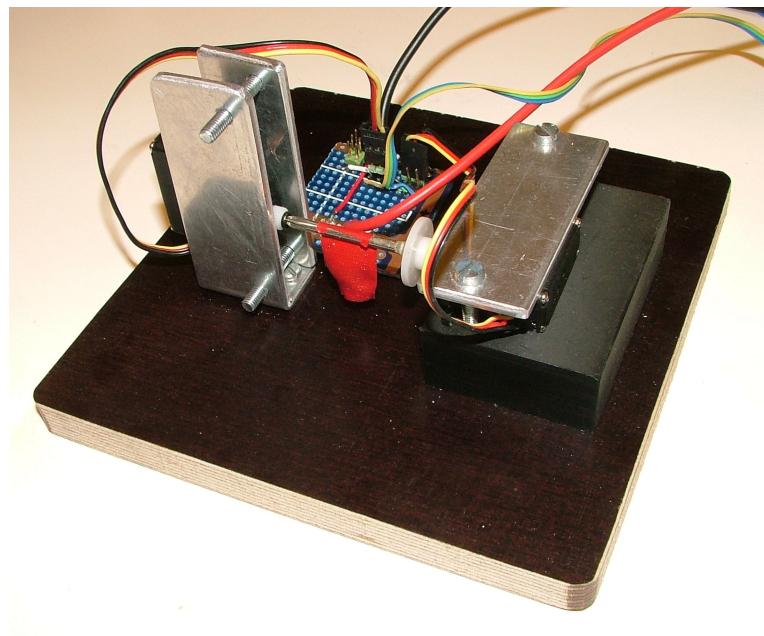


Figure 9.8: Servotester circuit used

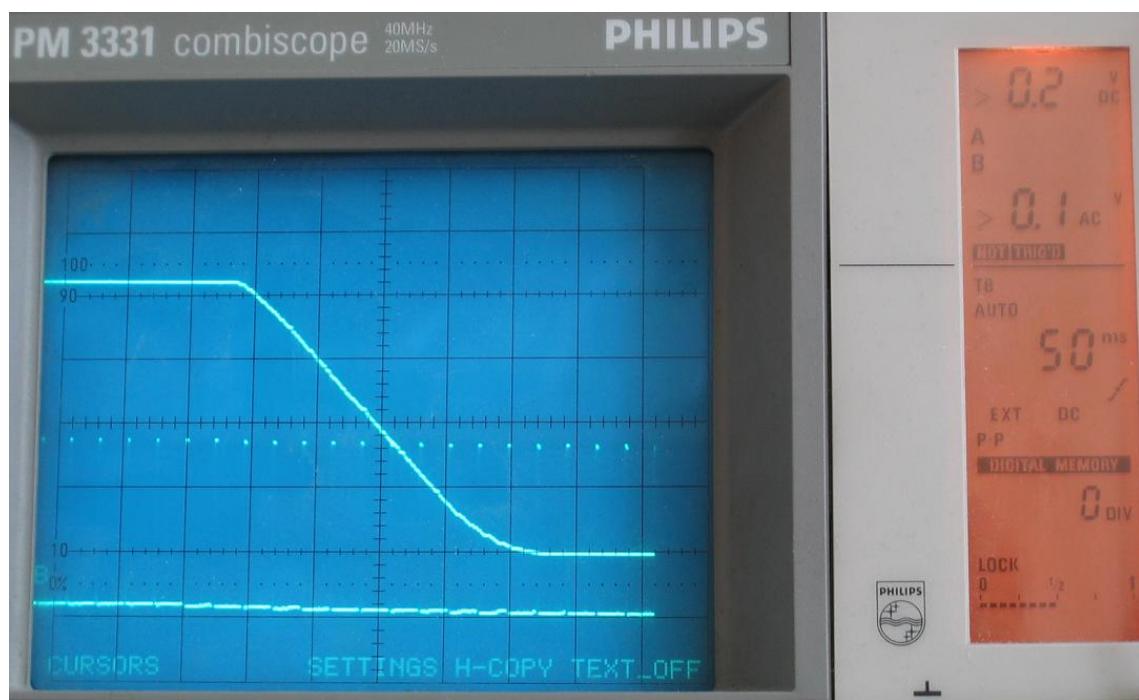


Figure 9.9: Step response of Hi-tec HS-81 servos

9.4.3 Torque

The servo torque specified on the manufacturers website (<http://www.hitecrcd.com/>) is 3kg/cm @ 6.0V. This is more than adequate for this application, the HS81 servo compares to a "standard servo" usually used for much bigger rudders.

9.4.4 Stall Current

This is of importance in the case of "radio glitches" where all the servos may try to oscillate, because of erroneous servo commands. This can cause maximum current drain from all servos. Therefore the current supply circuit was dimensioned to cope with this.

In addition the PCM1024 receiver almost eliminates the risk of glitches. This receiver includes an micro controller with Cyclic Redundancy Check (CRC) to verify the radio signals, this lowers the risk of radio "glitches". If radio transmission is lost the receiver also includes a Fail Safe feature turning all servos to a predefined position. Using this, the motor command was set to zero in Fail Safe mode, as this is the biggest source of radio disturbance.

Chapter 10

Wind tunnel

This Chapter presents the wind tunnel testing done February 2007. The values sampled in the wind tunnel are the data analyzed in Chapter 3, thus giving the aerodynamic coefficients used in the linear and nonlinear system models.

10.1 Setup and measurements

The windtunnel testing was done at Department of Energy and Process Engineering at NTNU. The air frame was fitted to three metal rods bolted to a weighing cell, see figure 10.1. To find the aerodynamic properties of the air frame alone, the properties of the tree metal rods was subtracted from the measurements. This can give errors from the real values because the flow around the muting brackets alone slightly differs from when the plane is attached. But the error made is so small it does not compromise the validity of the data. Runs of different alpha, beta, engine settings and rudder settings, at different airspeed was performed. For each setting an average of 5 measurements was used for aerodynamic analysis. The data was imported, and processed in a spreadsheet converting the three moments and three forces to aerodynamic coefficients. These values were then evaluated together with mass and inertia to give the stability derivatives used in the Matlab/SIMULINK model. Problems with the LabView software used, resulted measurements of raw forces and moments only. The program was later fixed giving the aerodynamic coefficients directly.

10.2 Test selection

To keep within the time limit of the wind tunnel and project scope, only a limited selection of the most critical test settings was tested. Lift and pitching moment for different α 's and speed were considered of great importance. Side slip, the effect of β was also important. The last property to be evaluated, was the force and moments of rudder, ailerons, elevator and throttle. For each test setting drag was obtained. The flightconfigurations tested are listed below.

- Steady Level flight at different speeds(10,15,18[m/s]) and alpha (-6,-2, 2, 6, 10, 14 [°])
- Steady Level flight at different speeds(10,15,18[m/s]) and beta (12,-8, -4, 0, 4, 8, 12 [°])

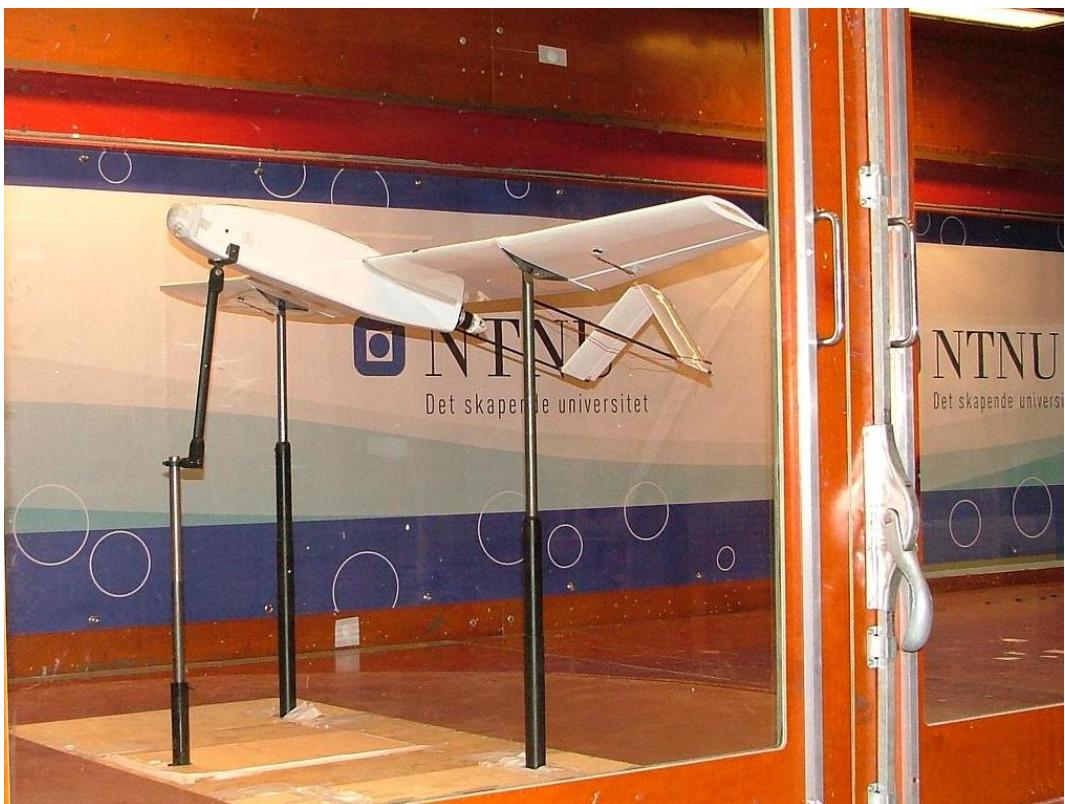


Figure 10.1: Setup in wind tunnel

- Steady Level flight at different speeds(10,15,18[m/s]) with different aileron($9[^\circ]$), rudder($14[^\circ]$) and elevator($15[^\circ]$) commands
- Steady Level flight at different speeds(10,15,18[m/s]) with different throttle setting (4000-8100 RPM)

10.3 Results

The windtunnel tests were the first feedback on the design and construction work done in [12]. When settling for the design in fall 2006 margins in size were added to garanti lifting capabilities etc. The tunnelresults showed a stallspeed of approx. $9[m/s]$ for a plane with weight of $2[kg]$. This is more or less just the same as what was calculated in the design phase. Further, a low drag configuration was important in the design phase. Results from the tunneltesting suggest a L/D-ratio of the total aircraft of 30, at $\alpha = 4[^\circ]$. At low Reynoldsnumbers this is considered as a very good result. Actually some airfoils alone have a lower L/D-ratio. Manual flight testing in February 2007 gave the same impression, a very low drag air frame. Especially when trying to land, resulting in overshooting the landing strip several times. Video of these flights are supplied on the DVD attached to this report. The obtained L/D-ratio suggest thrust of only $0.66[N]$ was sufficient to sustain level flight. For climbing purposes and safety in initial testing the motor(Dualsky PR.25) was used. This is to big to be ideal for duration flights. Optimizing the motor choice for duration flight i.e. using a HXT24 outrunner motor, a input power of only $30[W]$ is sufficient in Stabile Level flight. A flight time of just below 2 hours can be archived, using $4*1320[mAh]=5280[mAh]$ (ThunderPower 3S 1320mAh packs) This is less than half of the designed amount of batteries. At $11.1[V]$ this is almost $60[W/h]$. Thus the design goal of 2 hours flighttime has been met with good margin.

A compleate presentation of the numerical results of the tunneltesting are not suitable for presentation here, figure 10.2 gives and idea of the amount of work done. All the data from the tunnel testing are gathered in aero_engine_mass_inertia_parameters.xls file. Aerodynamic coefficients and stabilityderivatives are calculated in this file. Much effort has be put into deriving the results in this file. The file was made dynamical, that is future changes of the air frame i.e location and weight of hardware, can be put into this file. All coefficients are then automatically updated.

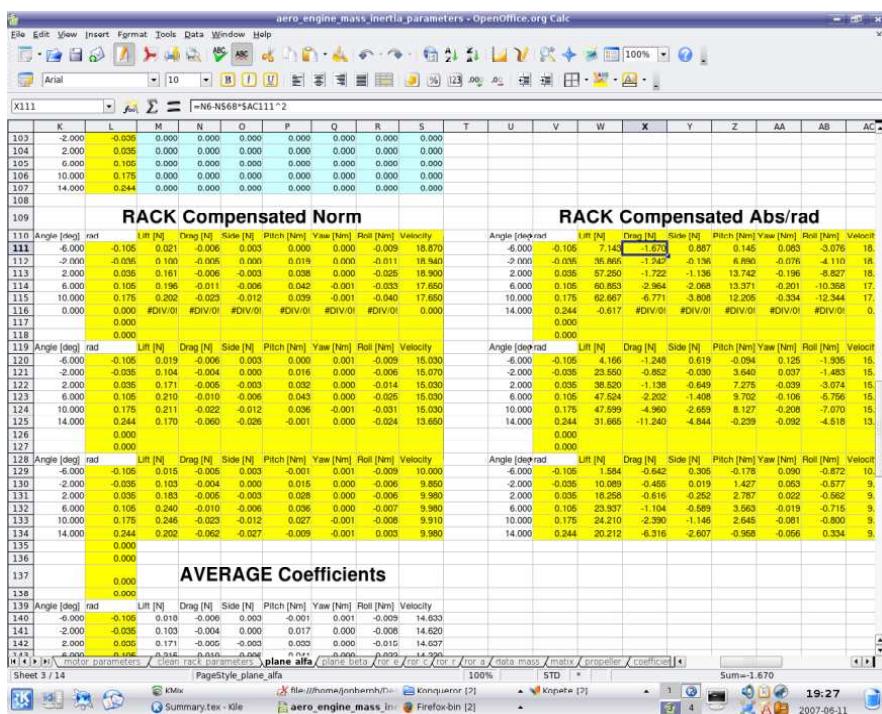


Figure 10.2: Spreadsheets data

Chapter 11

Simulation

The simulation of the two analytic and two empiric models are described in this Chapter. The data in the analytic models was calculated from the data collected by windtunnel tests, Chapter 10. The empiric models are based on model description from the data logged in testflight two, Chapter 8. Figure 11.1 show the second order model in SIMULINK.

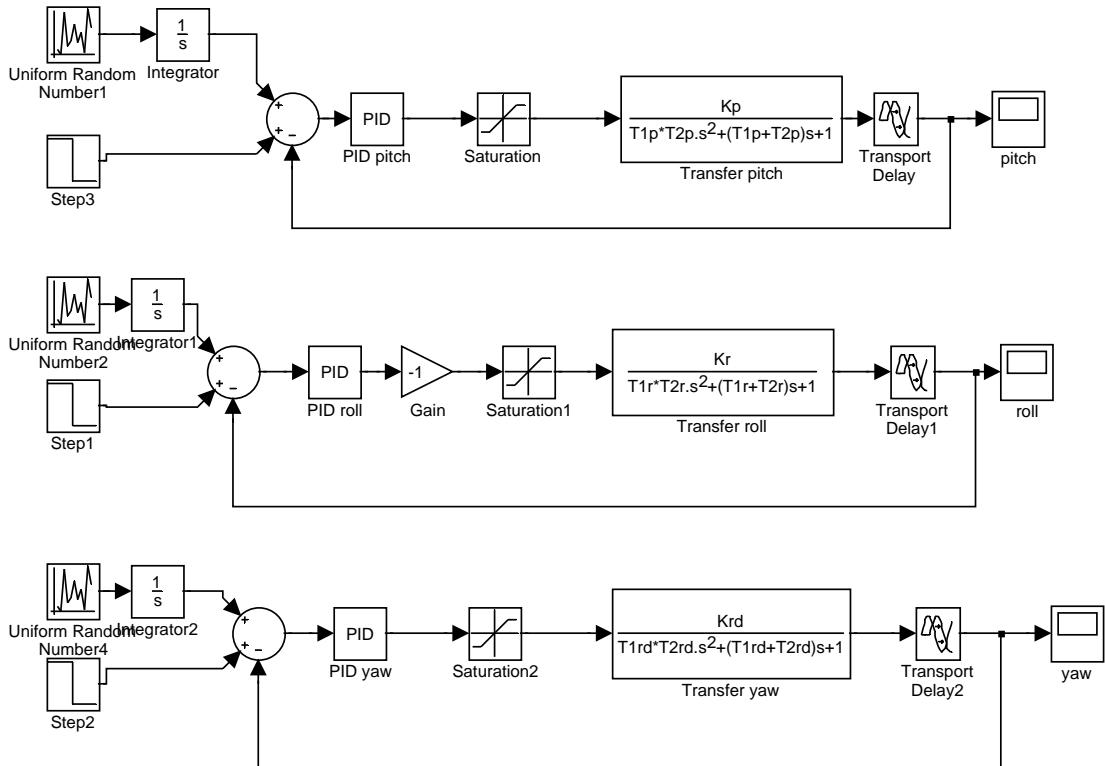


Figure 11.1: Second order model

11.1 Analytic Models

The first analytical model is based the AeroSim toolbox in Matlab. Data from XFOIL simulations, wind tunnel testing, and mass and inertia calculations make up the parameters needed for the model. The library was expanded to enable electrical propulsion, with no loss of mass, as the library only included fuel based propulsion systems.

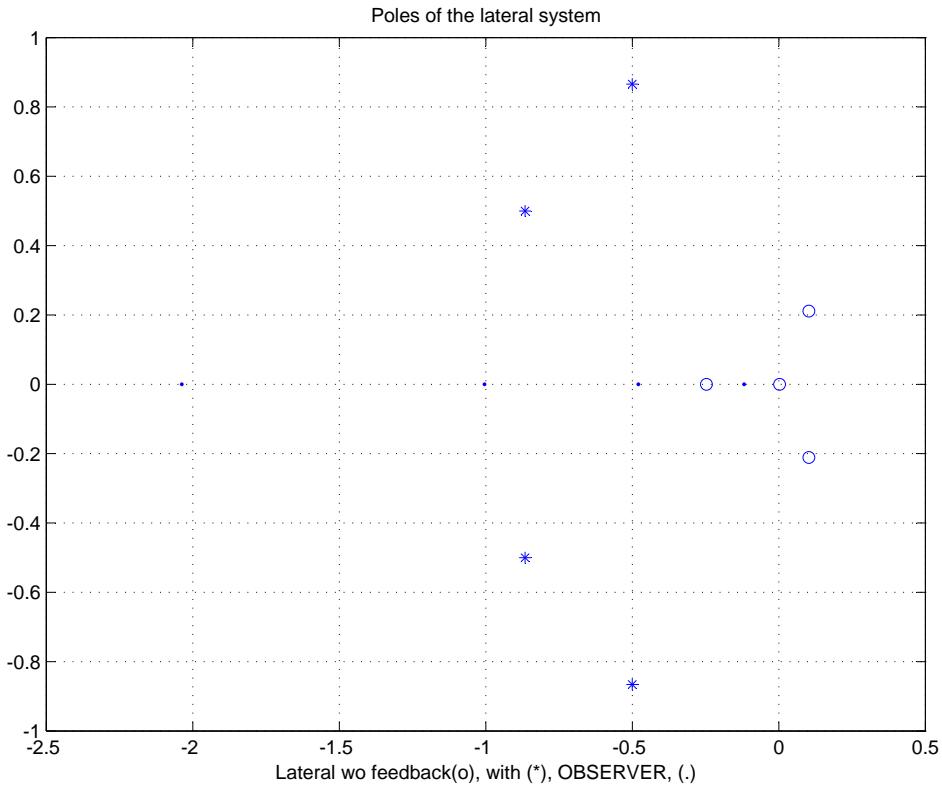


Figure 11.2: Poles of the linearized lateral system

In addition to the nonlinear model a linear model was made. This was based on common theory from i.e [14] and [5], using stability derivatives. In this model the longitudinal and lateral dynamics were separated into two 4 by 4 systems, each with two control inputs. The same data as for the nonlinear model are used to find the stability derivatives, this is done in `aero_engine_data_parameters.xls`. The longitudinal and lateral system from Chapter 2 were implemented in Matlab/SIMULINK. When comparing the performance of the UAV in the first testflight, with the poles of the systems (some in the right half plane), it became obvious some of the stability derivatives was erroneous, see figure 11.2. The derivatives were compared to the values of the AeroSonde parameters supplied with the AeroSim toolbox and other similar aircrafts from [17] and [5]. Because of the scaling issues discussed in Chapter 3 a alternative approach using empiric models for analysis and controller tuning was decided. As the analytical models both were based on the same data. Using a working empiric model, the flaws in the analytic model can be found contributing to a more general analytical model. The late arrival of measurement data, left this task for the continuation of this project.

11.2 Empiric Models

Based on the data from testflight two, two models were derived. First a firstorder model was calculated based on the natural frequency form the collected data, and tuning the process gain to best fit the model to the data. Simulations did not provide proper response as the P-controller gain could increased unreasonable without making the model

unstable. The physical model can be viewed as a mass-damper-spring system. A P-controlled first order system with just one pole in the l.h.p can not become unstable, clearly violating the physical system. Using Matlab Identification toolbox, a secondorder systems were derived, see figure 11.3. The model was tested on the entire datarange, as figure 11.3 show the approximation looks adequate. The inputdata included noise and glitches, while the outputdata included disturbances from wind. Nevertheless the modeapproximation looks good. The model derived from Identification toolbox is included in the file data_secondorder2.m. The parameters were slightly adjusted to fit the timedelay and gain of the real process. Figure 11.4 show the pitch step response from SIMULINK with with noise.

11.2.1 Rudder Yaw

The rudder yaw model was derived in the same way as the pitch and roll models, but as the yaw data was π -periodic and A-tailconfiguration mix were used, additional computation was needed. First a program was made to make the yaw-measurement continuous, see 8.2. Subtracting the ch2 from the ch4 data to find the rudder command was not possible, as the ch4 was beyond recognition because of noise. Examining the rate of turn was done to find time ranges of rudder action. Then the ch2 and yaw were used as input and output in the Identification toolbox. The secondorder model obtained were then verified on the entire datarange. Figure 11.5 show the stepresponse of the yaw model with a 10[*degree*] change in reference.

11.2.2 Verifying models

The data obtained form autonomous flight was plotted to verify the models. The pitch model was in good accordance with the real response of the physical system, while to little gain was found in the roll model. Verifying the yaw model was hard provided the data form the fourth test flight. This is discussed further in 13. Figure 11.6 show the actual response of the system compared to the model response.

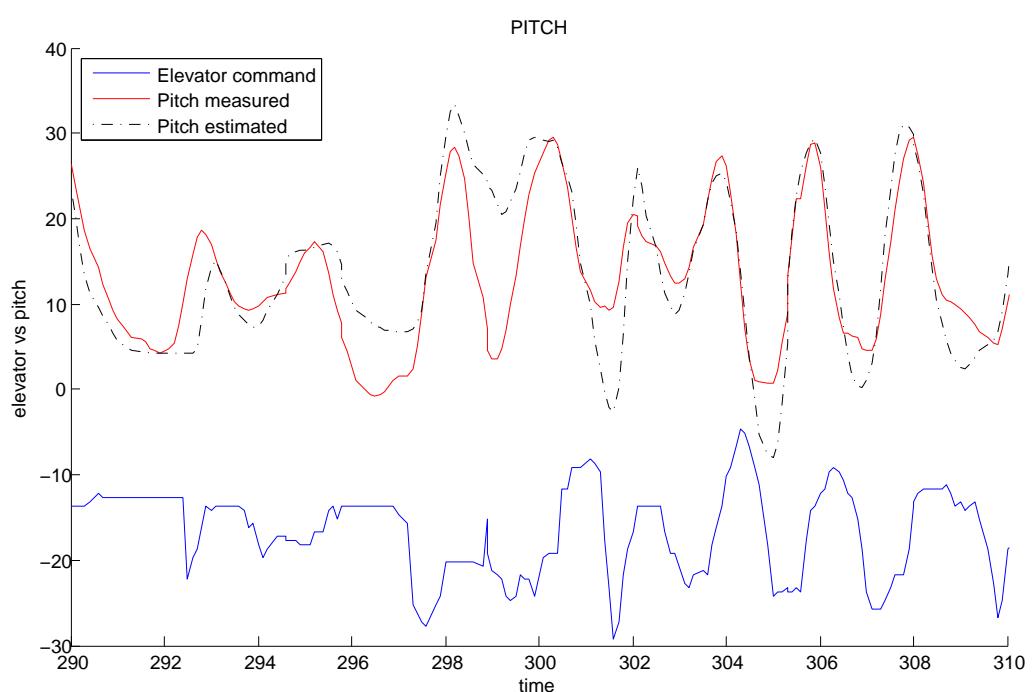


Figure 11.3: Estimated pitch model run on measurement data form second test flight

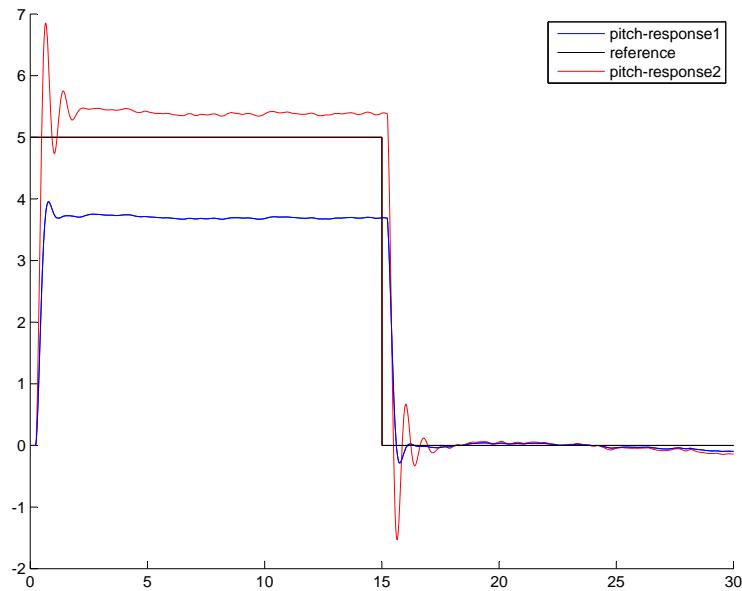


Figure 11.4: Pitch step response with PD-controller tuned using Ziegler Nichols-method, then response with slightly reduced P-controller gain is shown.

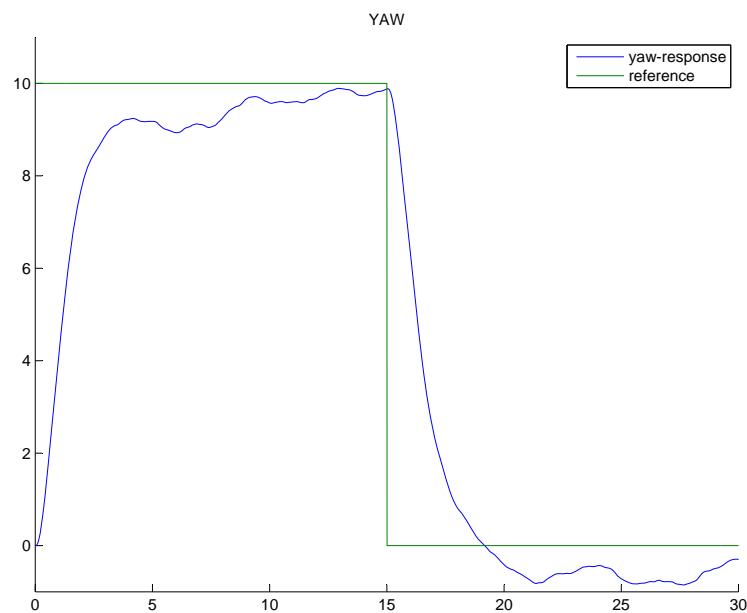


Figure 11.5: Step response of yaw model with noise.

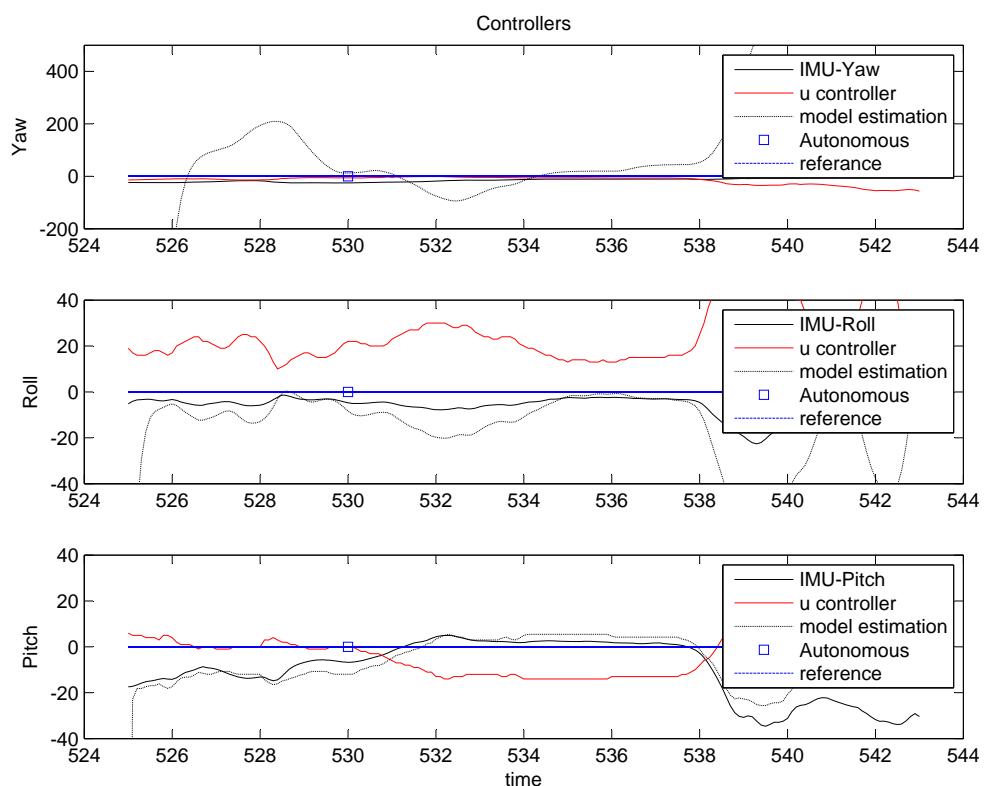


Figure 11.6: Real model vs. estimated model, data from autonomous flight.

Chapter 12

Results

12.1 Wind tunnel

Testing in the windtunnel has confirmed the desired properties from the simulations and dimension analysis in [12]. L/D numbers of 33-36 were achieved for weight of $1.850 - 3.8[\text{kg}]$ within the speedrange $10 - 15[\text{m/s}]$, see sheet plane_alpha. This data show that higher cruise speeds are preferable to achieve maximum range with full battery capacity on board. In addition to these key elements, data for different β , control surface and engine settings were found. These are presented in aero_engine_mass_inertia_parameters.xls.

12.2 Analytical models

Two analytical models were derived. The nonlinear model is based on adaption of the AeroSim library, to allow electrical propulsion. This model is based on the coefficients found from the windtunnel data and the mass/inertia parameters calculated in aero_engine_mass_inertia_parameters.xls.

A linear model was developed and simulated based on the theory form from Chapter 2, 3 and 4. The model consisted of two decoupled systems, one longitudinal and one lateral. Using stability derivatives derived form the data in section 12.1 and findig the poles of the systems in MATLAB, uncertainty of the validity of the model rose. This was due to that some of the poles were well into the r.h.p. This did not correlate with the experience from the first testflight. Time did not allow this model to be compared with the empiric model.

12.3 Empiric model

In addition to he analytical models two empiric models were built based on input and output data from the second testflight. The input data sampled from the receiver/servos and output data measured by the IMU were analyzed with the Identification toolbox in MATLAB. A first order and second order model was found. Both were simulated in SIMULINK. As the physical system represent a mass-damper-spring system, it was believed that the secondorder system was the best representation. Using Ziegler-Nichols method from[16] PD controllers for roll, pitch and yaw were found. The performance of these controllers are presented in the testflights section 12.5.1 and onwards.

12.4 Tests

Four testflights were performed, each with different goals. In addition different hardware tests was conducted. Below the goal of each test and the results are presented.

12.4.1 Motor

The goal for the motor testing was to find the physical properties of different motors. Functions for thrust, RPM and efficiency with respect to current, voltage and propeller were found. Max efficiency was found at $\pm 80\%$ of maximum thrust for all motors. The efficiency also relied on the propeller choice. The data collected were gathered in motortester_new.xls this file also include data for the different propellers tested. For the motor/propeller configuration used on the air frame, maximum thrust of 14.6[N] and max efficiency of almost 75% were found. The motorgraph for the Dualsky PR.25 was wide giving good efficiency over wide RPM range, see 9.2.

12.4.2 Servos

Test equipment to find the actuator dynamics were made, see Chapter 9. From these tests the slewrate of the servos were found to be $530[^{\circ}/s]$ @ 4.8V and $695[^{\circ}/s]$ @ 6V, the last value being, used in simulations and real life. The timedelay of the servos was found to vary from $40 - 80[m/s]$. Stallcurrent was found to ensure not exceeding the UBEC capacity of $3000[mA]$. Each servo had a stall current of approximately $600[mA]$, giving a maximum current draw of $2400[mA]$ for all four servos. The manufacturer claim torque of $3.0[kg/cm]$, this value was not controlled, but the servos were tested for adequate torque in the windtunnel @ $20[m/s]$.

12.4.3 Battery

The batteries were tested for C-rating(max and duration) and capacity. The HXT batteries were found to be from 80% to 90% of claimed capacity. While 50% of the claimed C-rating was found. For the ThunderPower batteries better accordance with claimed ratings were found. C-rating of 12-20 and 97% capacity for a worn battery.

12.5 Test flights

12.5.1 First

The first testflight was conducted in February 2007. The goal of the test was to evaluate the flight characteristics of the UAV. Manual flight was easy as the plane was just on the stable side of neutral. This gave responsive but stable flight characteristics. Further the lowdrag configuration was confirmed as height flightspeed with low throttle setting was achieved. Low landingspeed of $8 - 10[m/s]$ was found with a weight of $1.350[kg]$.

12.5.2 Second

The goal of the second testflight was to collect data for the empiric model analysis. Oscillating maneuvers with ailerons, rudder, elevator and throttle provided the data later analyzed, see 11. Video of the testflight is supplied on the DVD.

12.5.3 Third

Using the models found using Identification toolbox in MATLAB and second test flight data, pitch and roll controllers were simulated and tested in SIMULINK. Then both controllers were run on the real system. The pitch controller performed very well, with fast response and no significant overshoot. The aileron controller started oscillating when the parameters derived from simulation was used. Reducing the gain to half of the initial gain eliminated the oscillations. The pitch and roll controllers were tested at different speeds and bank/pitch settings, always stabilizing the UAV to steady level flight. The first two flights in the video *test_flights3-4.wmv* on the DVD show the performance of the controllers.

12.5.4 Fourth

The goal of the final test was to test the pitch, roll and yaw controllers together, achieving steady level heading flight. The tests were performed using the controllers found in the third testflight, adding the yaw controller. The ruddervator and malfunctioning saturation limit on the yaw controller made the plane pitch, when autonomous flight was engaged far from the reference 180(South). This happened as the pitch and yaw controlsurfaces were the same, thus for extreme deflections disabling the other controller, more on this in 13. When keeping closer to true S the controllers stabilized to strait level heading flight in autonomous mode. The second part of *test_flights3-4.wmv*, show autonomous flight with all controllers engaged.

Chapter 13

Discussion

13.1 Mathematical Models

13.1.1 Analytic Models

An aircraft is a complex system, both regarding complexity and size of the equations. After setting up a complete nonlinear mathematical model of the system, model reduction and simplification was essential to finish on time. The final linear model found is much simpler than the initial nonlinear system. In [2] a similar model was compared with a nonlinear model. Both models showed the same response when operating close to the equilibrium. Using a linear model is also preferable since this simplifies the task of designing a good Kalman filter.

The final linear model was not tested in the airframe. This was due to concern with right half plane poles from simulations. Later work using the empiric model to verify the analytical, will provide a more general model. Having a analytical model is a great advantage as modifications are easily accounted for in the stability derivatives. For the empiric models discussed below this is not as simple.

13.1.2 Empiric Models

The first order model found performed well when tested on the input and output data form the second testflight. However, this was not sufficient, as a fist order system can not become unstable when Proportional control is used. This is clearly not true for the physical system. Matlab Identification toolbox was used to find a second order model, describing the "mass-damper-spring" physical system. Even with wind noise contaminating IMU-data and glitches and noise on the sampled data form the receiver, a good model in pitch, roll and yaw was found. These models were then tested on the entire datarage from the second testflight with good results. Tuning of the model parameters was needed to get the correct time delay and gain. Then the models were implemented in SIMULINK, Ziegler-Nichols method was used to find basic PD-control parameters. The real life performance of these controllers are presented in 13.7

13.2 Measurements

Testing of the subsystems were performed to obtain numerical values.

13.2.1 Air Frame

Measurements of the air frame included wind tunnel testing, and weighing and finding x,y,z coordinates of all components. Weight and location of components was essential when finding the inertia, weight and CG of the system. The forces and moments found with tunnel testing was in accordance with the values found from simulations in [12]. The aerodynamic data was used as the basis for the aerodynamic coefficients calculated. Combined with the weight data, stability derivatives were found. Simulation of the linear model showed right half plane poles not in agreement with the flight characteristics suggesting poles on the imaginary axis or in the left half plane. Especially the data from the β tests, did not agree with data from similar aircrafts. One source of error was identified to be the rods holding the air frame in the tunnel. When changing the β angle, the rod configuration became asymmetric. The forces and moments from the rods were subtracted from the aircraft data, but rod data for different β values was not logged. No valid rod data for different β exist making it difficult to correct the air frame β -data. Computations and processing of the measured data is discussed in section 13.3. Theory on stability derivatives can be found in [14], [5], [17] and [6], but none of these gave a description on how to compute these from measured data. To find competence on the subject it was turned to Department of Energy and Process Engineering, EPT with little success. It was then decided to find an empiric model. This model was meant to verify the analytical models, but time to verify the analytical model was not present at the time of completing the empiric models. It is believed that this is of great importance for later work, as verification of the linear model also will verify the nonlinear model. This is because both models are based on the same data. Further modifications are easily accounted for when using an analytic model. Using the developed software, automatically computing coefficients, modifications can be made to the air frame, without need of complete model rebuild.

13.2.2 Motor

The propulsion system is a complex system including dynamics from battery, ESC, motor and propeller. Although analytical model can be found for each subsystem it was decided to construct a test rig finding the input output dynamics of the total motor subsystem. Using “of the shelf” Chinese RC components, with little, faulty or lacking specifications, further supports this decision. Measurements found was compared with analytical results from models found on the internet, i.e <http://www.peakeff.com>. Publishing the data on unitedhobbies.com/community also gave good feedback on the measurements, giving them some credibility. It is worth noticing that the ESC thought using feedback when controlling the brushless motor, does not include a feedback system holding a commanded RPM or power output. It works as the accelerator of a car, revving the motor depending on the load. Lack of current or RPM sensor in the sensorsystem [3] limited the performance of the speed feedback loop. When this was written although intended, the code for the speed sensor in [3] was implemented. For the flight test a constant throttle setting was therefore used. This was not ideal, but a speed feedback loop was not essential for the system to work.

13.2.3 Battery

For the batteries three properties were found, these were capacity, continuous and peak C-rating. All properties found to be below spec for the HXT1000 batteries. Publishing this on a RC forum confirmed that others had the same problem. Late in the project the distributor of the HXT1000, claimed there had been problems with the production at

the factory. Replacement of the batteries was guarantied by the factory. For the more expensive ThunderPower 1320 batteries tested, the batteries performed as specified. As the ThunderPower are much lighter, packing almost twice the amperehours per kg, they are worth the extra money. The complete battery tests are found in 9.

13.2.4 Servo

The actuator dynamics can influence the stability of a system if this is not accounted for. To find the dynamics of the servos, a servo tester was built. The sevotester finally used had a simpler design than initially intended, but proved to be a powerful tool. Five different servos were tested, HS-81 was the fastest and most accurate. None of the servos had any perceptible dynamics, except slewrate limit. The slewrate found was included in the simulations without affecting the simulation result. This was because the servos had higher bandwidth than the air frame. See Chapter 9 for plots and details on the servo tests.

13.3 Developement of Air Frame

From the work in [12] most of the air frame hardware was finished. The main development in this assignment consisted in combining the theory and measurements found from Matlab and SIMULINK analysis. Setting up simulation models from equations was straight forward. Modification of the propulsion model in the AeroSim library was necessary as this did not support electrical propulsion. Electrical propulsion means no loss of mass during flight, which was added as a feature. For the linear model Stability derivatives was found combining the aerodynamic and weight data discussed in section 13.2. The dynamics found during the servo testing was also added to the model. A figure of the model is presented in 11.

Hardware modifications to the air frame was necessary, mainly on the power system and signal routing to the computer system.

Beside the development mentioned above, hardware for the motor rig, servotester, and windtunnel was designed and built.

13.4 Simulation

The framework of the Simulation model with additional scripts for analysis of the UAV were completed in this work. Lack of comparable data coefficients and stability derivatives, has left some work on verifying the analytical models. The second order empirical model found using Matlab Identification toolbox, was postponed because the sensor and computer system in [3] was not ready for flight. Necessary data for this analysis came just days before finishing the work, leaving too little time to verify the analytical models. This showed that pitch and yaw controllers harmonized well with the IMU data, while the roll model had too little gain. This may explain why the P-parameter for the roll controller was too big, resulting in oscillations on the third test flight. Bisect of the original value solved the oscillations, in the third test flight. Time delay in the computer system may also be part of this problem.

13.5 Additonal Work

The goal of this assignment was to develop a stable UAV system. To reach this goal the work was divided into 3 subprojects. This report was concerned with simulation, control and navigation. The supplementary task of being the driving factor, making sure the total project stayed on track was also essential. This was done by motivating and planning in advance for all participants, keeping the goal of the whole system in focus.

A time schedule was constructed for the 3 projects. The first testflight was due two weeks after Easter. Six weeks later one of the subprojects were still not ready, jeopardizing the total project. A decision had to be made either continuing on this report, or helping out on the subproject lacking behind. The latter was chosen, to be able to get the system running before end of the semester. Week 21 and beginning of week 22 was devoted, affecting the work in this report. But without the computer system this work could not be tested. The extra work done consisted of ordering and providing missing hardware. Finding hardware and software solutions and innstalling the hardware in the air frame. Then making necessary cables, and testing hardware and software, freeing time for the software development of the project lacking behind.

13.6 Implementation

Implementation of the system consisted of connecting the sensor and computer system to the actuator and propulsion hardware. The Real Time Workshop interface intended from [3] was not finished when test flights were carried out. IMU sensorsignals and actuators were therefore connected through a C-script. As both angle and angular rate measurements were available a PD-controller was implemented. Hardware in loop testing was successfully compleated, before the test flights.

13.7 Test Flight One:

The goal for the first flight was to verify the properties of the air frame form [12]. Judging from the "pilot workload", a easy to control stable system was constructed. Landing speed and gliding properties proved the low drag configuration with plenty of lift. From these properties it was concluded that the system was suitable for autonomous flight.

13.8 Test Flight Two:

This flight provided the data for the empiric model. The flight was performed manually, while logging actuator commands and UAV response. Series of maneuvers with step and sinusoidal elevator, aileron, rudder and throttle commands were performed. Two ThunderPower 1320 batteries were used for propulsion. After 6 minutes of circular flight the batteries were charged $364 + 370[mAh] = 734[mAh]$, at a speed of $20[m/s]$ a distance of $7.2[km]$ was covered. With 8 ThunderPower 1320 batteries a flight time of 1hours 40 minutes, and range of over $120[km]$ can be achieved. The computer system used $267[mAh]$ in 10 minutes, 2 ThunderPower 1320 batteries would be sufficient for a flighttime of almost 1 hours 40 minutes. With 10 batteries the weight of the UAV is below the limit of the maximun design weight of $2.8[kg]$. This meets the design goal set in [12].

13.9 Test Flight Three:

Using the controllers found from the empiric model, autonomous flight was tested in the third test flight. Pitch and roll controllers were manually switched on from the transmitter. The first test showed good pitch control, but oscillations in roll. Reducing the roll controller gain to half of the initial gain solved this problem. Running the roll model on the data obtained from the last flight showed too little gain in the roll model. In combination with time delay in the computer system, this was thought to be the source of the roll oscillations. Video of the third test flight are supplied on the report DVD.

13.10 Test Flight Four:

The goal of the final test was to test the pitch, roll and yaw controllers together, achieving steady level heading flight. Initially the yaw controller affected the pitch, when autonomous flight was engaged far from the reference ($pitch = 0, roll = 0, yaw = 180[^{\circ}]$). This happened because the ruddervator tail configuration both control yaw and pitch. The limit for rudder command was not working adequate, saturating the ruddervator and thus, disabling the pitch controller. Keeping within $\pm 20[^{\circ}]$ from each reference the system stabilized about 0, 0, 180. The second part of test_flights3-4.wmv show autonomous flight with all controllers engaged. The trim and saturation problems limiting the performance of the system should be a quick fix adding a few lines in the C-script, but lack of time has left this for following work.

13.11 Goal Fulfillment

The main goal for this project was to construct a working UAV prototype by summer 2007. During the development it was ensured further work could be based on this work. This meant working systematically, documenting all tasks, minimizing the effort needed to continue the development. The third and fourth flight test confirms that the first part of the main goal was met. The second part is hard to verify now, but it is believed this also was met. Some work persists in completing and refining the system. For the FCS part of the system, this mainly consists of verifying the nonlinear and linear model, and finetuning the controllers. Fulfilling this the system should prove to be a good platform for testing hardware and controllers, from coming projects. Using more advanced control theory and models, i.e Lyapunov theory, MPC and μ -analysis, alternative controllers could be tested and compared with existing controllers.

Chapter 14

Conclusion

In this master thesis a Flight Control System (FCS) has been designed and tested on the UAV-prototype developed fall 2006, [12]. A major task has been to determine the physical values and parameters of the air frame. This includes wind tunnel testing, analytical calculation and bench testing hardware. Using the obtained data, modeling and simulation of a complete aircraft system was done. The models are based on a combination of general aircraft-, aerodynamic- and control-theory. Three simulation models were developed and analyzed in Matlab/SIMULINK. One linear model based on the theory from [5], one based on the AeroSim environment developed by M.I.T, and one based on model identification using Identification toolbox in Matlab. Initially finding an analytical simulation model was intended. Then using this for evaluating the openloop characteristics of the airplane, and designing controllers. The analytical model proved hard and time-consuming to obtain, therefore an alternative approach was followed. Using Matlab Identification toolbox and input output data from manual flight testing, an empiric model was constructed. Hardware and software problems, postponed the computer system development [3], resulting in late arrival of required data. This limited the amount of time left to calibrate the analytical models. To ensure the goal of having a working FCS, the empiric model was used directly to find longitudinal and lateral controllers. These controller were tested in flight test three, proving the controllers sufficient for stable flight. In flight test four heading control was added to the pitch and roll FCS, compleating the goal of stable autonomous flight.

The FCS derived thought this thesis enables stable heading flight. A tight timeframe and budget has left the need of further testing and development to ensure the robustness of the system, this is discussed below.

Looking back this project has been demanding and hectic, but also motivating and rewarding. Solving a practical problem has given a deep understanding of the challenges faced in real life cybernetics. It is believed this has contributed to expanding ones horizon.

14.1 Further Work

Future development on the FCS could involve finishing the simulation environment using the data from the empiric model. Using the simulation environment μ -analysis and/or multivariate MPC controll can be used to further enhance the robustness and performance of the FCS system. Developement of a robust sensor and computer system prior to further flight test should be considered. Including IMU, GPS, and windspeed sensor in this system will enable waypoint navigation, with robustness to wind. Further including a height

sensor, i.e using ultra sound, autonomous takeoff and landing will be feasible. For this feature an extra non powered control routine will need to be designed.

Looking from a commercial view point in an expanding UAV market, reduction of the cost of the sensor system is beneficial. This could be done by enhancing the performance of the Observer and or Kalmanfilter, excluding sensors.

Regardless of the continuation of this thesis, UAVs are taking part in our world now, and will be present in the future.

Weblinks

<http://www.u-dynamics.com/aerosim/default.htm>
(Unmanned Dynamics, AeroSim)

<http://www.futaba-rc.com/radios/futj85.html>
(Radiosender og mottaker)

<https://www.unitedhobbies.com>
(Forhandler av motorer, regulatorer og batterier)

<http://www.hitecrcd.com/>
(Servoer)

<http://www.hextronik.com/>
(Produsent av batterier og regulatorer)

<http://www.dualsky.com>
(Produsent av motorer)

<http://uav.wff.nasa.gov/>
(Overview over existing UAVs)

<http://aircraft-world.com/>
(RC store)

<http://www.nasg.com/afdb/list-airfoil-e.phtml>
(airfoil-database)

<http://nastytoesaviation.com/>
(Foam parts)

Bibliography

- [1] Abbot and Doenhoff. *Theory of wing sections*. Dover Publications Inc, New York, 1959.
- [2] Lasse Bjermland. *Modeling, simulation and design of a Flight Control System for an UAV*. Institutt for Tekniskybernetikk, 2005.
- [3] Edgar Bjoerntvedt. *Design and construction of sensor and computersystem for fixed-wing UAV: CyberSwan*. Institutt for teknisk kybernetikk, NTNU, 2007.
- [4] Edgar Bjrtvedt. *Instrumentering av AUAV*. Institutt for Teknisk Kybernetikk, 2006.
- [5] McLean D. *Automatic Flight Control Systems*. Prentice Hall International, 1st edition, 1990.
- [6] Dipl.-Ing. X. Hafer Dr.Ing. habil. W. Just. *Seitenstabilitat und seitenssteuerung*. Verlag Flugtechnik Stuttgart/Ernst von Olnhausen, 1957.
- [7] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation of Automatic Control*. Marine Cybernetics, Trondheim Norway, 2003.
- [8] Ole-Johan Ellingsen. *Utvikling av styresystem for modell-helikopter*. Institutt for teknisk kybernetikk, NTNU, 2002.
- [9] Mikael K. Eriksen. *Ground Station and hardware peripherials for fixed-wing UAV: CyberSwan*. Institutt for teknisk kybernetikk, NTNU, 2003.
- [10] Rolf Henriksen. *Adaptive systemer*. Institutt for teknisk kybernetikk, NTNU, 1994.
- [11] Rolf Henriksen. *STOKASTISKE SYSTEMER, analyse, estimering og regulerig*. Institutt for teknisk kybernetikk, NTNU, 1998.
- [12] Jon Bernhard Hstmark. *Design og konstruksjon av ubemannet fly for visuell overvii $\ddot{\circ}$ ning*. Institutt for Teknisk Kybernetikk, 2006.
- [13] Fossen T. I. Matematiske modeller for styring av fly og satelitter. 1998.
- [14] Fossen T. I. *Marine Control Systems*. Tapir Trykkeri, 1st edition, 2002.
- [15] "JAR". *Principles of Flight 2nd edition*. Nodian AS, 2005.
- [16] Bjarne A Foss Jens G. Balchen, Trond Andresen. *Reguleringssteknikk*. Institutt for teknisk kybernetikk, NTNU, 2003.
- [17] Ashkenas McRuer and Graham. *Aircraft Dynamics and Automatic Control*. Princeton University Press, 1th edition, 1990.

- [18] Warren F. Phillips. *Mechanics of Flight*. Wiley, 2004.
- [19] McCormick B. W. *Aerodynamics, Aeronautics, and Flight Mechanics*. John Wiley and Sons, INC., 2d edition, 1995.

Appendix A

Symbols

3DOF Three Degrees Of Freedom

6DOF Six Degrees Of Freedom

α -angle of attack

AFCS: Automatic Flight Control System AoA: angle of attack

AR -aspect ratio = b^2/S

Bank: Roll angle β -sideslip angle

BEC: Battery Eliminator Circuit

b -wingspan

C_{Di} -inducted drag coefficient = D_i/qS

C_{Dp} -parasite drag coefficient = D_p/qS

C_D -total drag coefficient

C_L -lift coefficient

D_i -inducted drag

D -total drag

$D_{i,p}$ -parasite drag

ESC: electronic speed controller

FCS: Flight Control System fuselage: aircrafts body

F -force

h -height

HTA: horisontal-tail area

L/D lift drag relation

l -length

L -lift

m -mass

MAC -mean aerodynamic chord

p -pressure

PCM: Pulse Code Modulation

q -dynamic pressure = $\rho V^2/2$

r -radius

ρ -density

Rn : Reynolds number

S -aera

TMA tail-moment arm

V_∞ freestream velocity

v speed along x_b

w -speed along y_b

W -weight

WA -wing aerea

wash out geometric twist of wing

XFOIL: Program for simulating airfoils

Appendix B

Matlab-code

B.1 Nonlinear model

PARAMETER FILE:

```
%From Cybergooe file
%Initial Longitudinal parameters
x_init_Long=[10 -1 0 1]';
R_Long=[12 -1 0 -1];
P_Long=[-.866+.5i -.5+.866i -.5-.866i -.866-.5i]';
%Initial Lateral parameters
x_init_Lat=[1 0.1 0 0]';
R_Lat=[0 0 0 0];
P_Lat=[-.866+.5i -.5+.866i -.5-.866i -.866-.5i]';

%Physical constants
g=9.81;                                %From CG
U0=15;                                     %forward speed %From CG

%Servo paramters
T_servo=1/25;                            %From CG
limit_SERVO=1;                            %From CG
slew_SERVO=10;                            %From CG

%Motion
Xu=5.4618;
%- .09;                                    %From CG
Xw=.000;                                   %From CG

Zu=0%.17;                                 %From CG%Normaly small?
Zw=-2.7272;                               %From CG%Form UAV -

Mu=4.0587;
%4.0587;                                  %From CG%Form UAV
Mw=61.5718;
%-13.3104;                                %From CG%Form UAV -
```

```

Yv=-0.0368; %From CG%From UAV -
Lr=0.0131
%0.5; %From CG%Missing and signigicant -

%Control
XdE=0
%-0.3; %From CG
ZdE=-2.5485
%-5; %From CG

Xengine=10;

%%% Aerodynamic parameter bounds %%%
% Airspeed bounds
VaBnd = [8 30]; % m/s OK
% Sideslip angle bounds
BetaBnd = [-0.5 0.5]; % rad NOK
% Angle of attack bounds
AlphaBnd = [-0.1 0.3]; % rad NOK

%%% Aerodynamic reference parameters %%%
% Mean aerodynamic chord Ae
MAC = .215 %OK %0.189941; % m
% Wind span
b = 1.7 %OK %2.8956; % m
% Wing area
S = 0.3675 %OK %0.55; % m^2

% ALL aerodynamics derivatives are per radian:
%%% Lift coefficient %%%
% Zero-alpha lift
CL0 = -2.61; %OK
% alpha derivative
CLa = 0.8900; %OK
% Lift control (flap) derivative
CLdf = 0.00; %OK
% Pitch control (elevator) derivative
CLde = 0.0246 %OK div.degrees?
%0.13;
% alpha-dot derivative
CLalphadot = 0%
1.9724;
% Pitch rate derivative
CLq = 0
%7.9543;
% Mach number derivative
CLM = 0;

```

```

%%% Drag coefficient %%%
% Lift at minimum drag
CLmind = 0.4412
%0.23;
% Minimum drag
CDmin =-0.0159
% 0.0434;
% Lift control (flap) derivative
CDdf =0;
%0.1467;
% Pitch control (elevator) derivative
CDde =0;
%0.0135;
% Roll control (aileron) derivative
CDda = 0;
%0.0302;
% Yaw control (rudder) derivative
CDdr =0;
% 0.0303;
% Mach number derivative
CDM = 0;
% Oswald's coefficient
osw = 0.75;

%%% Side force coefficient %%%
% Sideslip derivative
CYbeta =-0.0029
% -0.83;
% Roll control derivative
CYda=0.0020           %SIGN
%-0.075;
Yda=CYda                  %Yda=0;      From CG
% Yaw control derivative
CYdr = -0.0092           %SIGN
%0.1914;
% Roll rate derivative
CYp = 0;
% Yaw rate derivative
CYr = 0;
Ydr=CYr%Ydr=-0.0001;      From CG

%%% Pitch moment coefficient %%%
% Zero-alpha pitch
Cm0 = 0;                 %Due to trim
%0.135;

```

```

% alpha derivative
Cma = 0;                                %Due to trim
%-2.7397;
% Lift control derivative
Cmdf = 0.0467;
% Pitch control derivative
Cmde = 0.1194
%-0.9918;
MdE=Cmde%MdE=-2.5;                      %From CG
% alpha_dot derivative
Cmalphadot = 0;
%-10.3796;
% Pitch rate derivative
Cmq = 7.4316E-04;
%-38.2067;
Mq=Cmq%Mq=0%-.55                         %From CG%Form UAV
% Mach number derivative
CmM = 0;

%%% Roll moment coefficient %%%
% Sideslip derivative
Clbeta = -0.0208;
%-0.13;
Lbeta=Clbeta
%Lbeta=-7.6829;                            %From CG%From UAV -
% Roll control derivative
Clda = -0.0484                             %OK
%-0.1695;
Lda=Clda% Lda=-1.6784;                     From CG
% Yaw control derivative
Cldr = 0.0074                             %OK
%.0024;
% Roll rate derivative
Clp = -0.0019                             %OK
%-0.5051;
Lp=Clp
%Lp=-3.5;                                  %From CG%Missing and significant - moves the
%real part of the complex poles
% Yaw rate derivative
Clr = 0.0000;
%.4994;%0.2519;
Ldr=Clr%Ldr=0.4994;                        From CG

%%% Yaw moment coefficient %%%
% Sideslip derivative
Cnbeta = 0.0042;
%.0726;
Nbta=Cnbeta%Nbta=0.7577;                  %From CG%From UAV
% Roll control derivative

```

```

Cnda = -0.0020;
%0.0108;
Nda=Cnda%Nda=-0.0142; %From CG
% Yaw control derivative
Cndr = 0.0074;
%0.1587%-0.0693; NOK
Ndr=Cndr%Ndr=0.1587; %From CG
% Roll rate derivative
Cnp = 0.0000;
%-0.069;
Np=Cnp%Np=-0.1; %From CG%Missing and signigicant -
% Yaw rate derivative
Cnr = -0.0002;
%-0.0946; %From Ae
Nr=Cnr%Nr=-1.0; %From CG%Missing and signigicant -
moves the real part of the compex poles
{

```

INIT FILE:

```

%%% AIRCRAFT CONFIGURATION SCRIPT %%%
%%% TEMPLATE %%%

% Copyright 2002 Unmanned Dynamics, LLC
% Revision: 1.0 Date: 08/07/2002

%%% IMPORTANT %%%
% Airframe origin (reference point) can be arbitrarily chosen
% Body axes convention is as follows:
% x - forward towards the nose
% y - spanwise, towards the right wing tip
% z - vertical, pointing down
% All data should be specified in metric units, unless otherwise noted

% Clear workspace
clear all;

%%% Begin editing here %%%

% Insert the name of the MAT-file that will be generated (without .mat extension)
cfgmatfile = 'cybergoose';

%Load all aerodynamical coeffecients
run('Aerosonde_CyberSwan_parameters');

%%% SECTION 2 %%%
%%% PROPELLER %%%
%Propulsion force application point [x y z]
% (location of propeller hub with respect to the origin

```

```

rHub = [0 0 0]; % m
% Advance ratio vector
% (arbitrary size, but sizes for J, CT, and CP must match)
J = [1 1];
% Coefficient of thrust look-up table CT = CT(J)
CT = [1 1];
% Coefficient of power look-up table CP = CP(J)
CP = [1 1];
% Propeller radius
Rprop = 0.254; % m
% Propeller moment of inertia
Jprop = 0; % kg*m^2

%%% SECTION 3 %%%
%%% ENGINE %%%
% Engine rpm vector
% (arbitrary size)
RPM = [0 12000]; % rot/min
% Manifold pressure vector
% (arbitrary size)
MAP = [0 0 0 0]; % kPa
% Sea-level fuel flow look-up table fflow = fflow(RPM, MAP)
% (Number of rows must match size of RPM vector,
% number of columns must match size of MAP vector)
FuelFlow = [
    0 0 0 0
    0 0 0 0
];
% Sea-level power look-up table P = P(RPM, MAP)
% (Number of rows must match size of RPM vector,
% number of columns must match size of MAP vector)
Power = [
    0 0 0 0
    0 0 0 0
];
% W
% Sea-level pressure and temperature at which the data above is given
pSL = 100000; % Pa
TSL = 300; % deg K
% Engine shaft moment of inertia
% (generally can be neglected)
Jeng = 0;
%neglisjerbar

%%% SECTION 4 %%%
%%% INERTIA %%%
% Empty aircraft mass (zero-fuel)
mempty = 1.884; % kg
% Gross aircraft mass (full fuel tank)

```

```

mgross = 1.884; % kg
% Empty CG location [x y z]
% (with respect to the origin)
CGempty = [0 0 0]; % m
% Gross CG location [x y z]
% (with respect to the origin)
CGgross = [0 0 0]; % m
% Empty moments of inertia [Jx Jy Jz Jxz]
Jempty = [0.109 0.225 0.333 0.0]; % kg*m^2
% Gross moments of inertia [Jx Jy Jz Jxz]
Jgross = [0.109 0.225 0.333 0.0]; % kg*m^2

%%% SECTION 5 %%%
%%% OTHER SIMULATION PARAMETERS %%%
% WMM-2000 date [day month year]
dmy = [27 03 2007];

%%% FINISHED ALL SECTIONS %%%
%%% Do not edit below this line %%%

% Save workspace variables to MAT file
save(cfgmatfile);

% Output a message to the screen
fprintf(strcat('\n Aircraft configuration saved as:\t', strcat(cfgmatfile), '.mat'));
fprintf('\n');

{

```

B.2 Linear model

INIT FILE

```

%Stability and Dynamics of Cybergoose

*****%
%This script finds the eigenvalues of the linerized Longditudinal and
%Laterals UAV-system, this is it finds the poles of the system.
%The model used is:

*****%
%All parameters are loaded %
*****%
clear all;

run('Aerosonde_Cybergoose_parameters');

%Model

```

```

*****Longitudinal*****
%x=[u w q theta]'

A_Long=[Xu Xw 0 -g;
        Zu Zw U0 0;
        Mu Mw Mq 0;
        0 0 1 0];

B_Long=[XdE ZdE MdE 0]';
% B_Long=[XdE      Xengine];           Mulitivariable controller
%          ZdE      0;
%          MdE      0;
%          0        0];

%C_Long=[0 0 0 1]
C_Long=[1 0 0 0;
         0 1 0 0;
         0 0 1 0;
         0 0 0 1];

l_Long=[1.2 0 0 0;
        0 .1 0 0;
        0 0 .5 0;
        0 0 0 1]; %[0 0 0 0]';

% *****
% * Computing the gain matirx K_Long by poleplacemet
% *****

K_Long = PLACE(A_Long,B_Long,P_Long)

%K_Long=[-.075 0.02 2.75 0];           %Old values
%K_Long=[0.0 -0.5 -2.0 1.5];          %OK experimental values
%K_Long=[10.0 0.5 20.0 1.5];          %for testing

A_Long_reg=A_Long-B_Long*K_Long*C_Long; %System with feedback

A_Long_lC=A_Long-l_Long*C_Long;          %OBSERVER

lambda_Long=eig(A_Long);
lambda_Long_reg=eig(A_Long_reg);
lambda_Long_lC=eig(A_Long_lC);

%clf;
figure(1)
plot(real(lambda_Long),imag(lambda_Long),'o');
% xlabel('Longitudinal without feedback, o');
hold on
plot(real(lambda_Long_reg),imag(lambda_Long_reg),'*');
% xlabel('Longitudinal with feedback, *');

```

```

title('Poles of the longditudinal system')
plot(real(lambda_Long_1C),imag(lambda_Long_1C),'.');
xlabel('Longditudinal wo feedback(o), with (*), OBSERVER, (.)');
grid on
hold off

%*****Lateral*****
%x=[v p r phi]

A_Lat=[ Yv      0    -1   g/U0;
        Lbeta   Lp   Lr   0;
        Nbeta   Np   Nr   0;
        0       1    0    0];

B_Lat=[Yda      Ydr;
        Lda      Ldr;
        Nda      Ndr;
        0       0];

C_Lat=[1 0 0 0;
        0 1 0 0;
        0 0 1 0;
        0 0 0 1];

l_Lat=[2.0 0 0 0; %should be n*1 for SISO system
        0 0.5 0 0;
        0 0 1.0 0;
        0 0 0 0.1];
% *****
% * Computing the gain matirx K_Lat by poleplacemet
% *****
K_Lat = PLACE(A_Lat,B_Lat,P_Lat)

A_Lat_reg=A_Lat-B_Lat*K_Lat*C_Lat;
A_Lat_1C=A_Lat-l_Lat*C_Lat;

lambda_Lat=eig(A_Lat);
lambda_Lat_reg=eig(A_Lat_reg);
lambda_Lat_1C=eig(A_Lat_1C);

figure(2)
plot(real(lambda_Lat),imag(lambda_Lat),'o');
% xlabel('Longditudinal without feedback, o');
hold on
title('Poles of the lateral system')
plot(real(lambda_Lat_reg),imag(lambda_Lat_reg),'*');
% xlabel('Longditudinal with feedback, *');
plot(real(lambda_Lat_1C),imag(lambda_Lat_1C),'.');
xlabel('Lateral wo feedback(o), with (*), OBSERVER, (.))';

```

```

grid on
hold off

% *****
% *      KONTROLLERS      *
% *****
% Heightcontrol
K_theta=0;
K_q=0;
K_c=0;

%Lateral
K_theta=0;
K_q=0;
K_c=0;

% *****
% *      ENGINE      *
% *****
% Engine controller parameters
K_engine=[0.5 0.0 0.0 0.0];
RPM=[12 0 0 0]';                                %Just for testing
{

```

B.3 Empiric model

SECOND ORDER MODEL

```

% help IDENT ident
% IDENT to start identification toolbox
% IDDEMO for demo
%
% %type ident
% %import ch1 as u1 data
% import pitch as y1 data
% Operations-> remove means
% Time plot
% Operations-> select range to split into estimaiton and validation data
% Drag pitche to Working data
% Estimate-> parametric model 2 0 1
% Estimate-> proscess model K=074361 T1=1.8218 T2=1.7421 Td=2.2969

%*****
%          PITCH      second order      *
%*****
```

```
%PADEAPPROX
%[NUM,DEN]=PADE(timedelay,order)

timedelay_p=.22;
exp_p=exp(-timedelay_p*tf([1 0],1))

corr_pitch=-10;
Kp=1.24361;
T1p=0.18218;
T2p=0.17421;
Tdp=2.2969;
Tt=0.1; %Sampling time
SYS_pitch = tf(Kp,[T1p*T2p (T1p+T2p) 1]);
%Adding timedelay
SYSC_pitch =SYS_pitch*exp_p
model_pitch=LSIM(SYSC_pitch,[ch1],[0:0.1:0.1*(length(ch1)-1)]);
model_pitch=model_pitch-corr_pitch;
%figure(10);
%bode(SYS_pitch);
%find maximum gain
%[Ap,Bp,Cp,Dp]=tf2ss(Kp,[1 Tp]);
%P_pitch=0; %Pole at origin
%Kpp = PLACE(Ap,Bp,P_pitch)
%Setting PID_pitch parameters
Kpk_p=1.9; %simulation gives a unstable system when Kpk_p
w180_p=11/(25-15);%s %foud from plot;
Tk_p=w180_p;% (2*pi)/(w180_p);
%Using Zeiger Nicholds values for PID with Ti=0
Kp_p=0.5*Kpk_p;
Ti_p=0.01;
Td_p=0.12*Tk_p;

%PLOTT Elevator vs. Pitch
fig_min_p=290;
fig_max_p=310;
% fig_min_p=200;
% fig_max_p=500;
figure(1)
subplot(3,1,1)
plot(time,-ch1,'blue');
AXIS([fig_min_p fig_max_p -100 100]);
ylabel('elevator')
subplot(3,1,2)
plot(time,pitch,'red');
ylabel('pitch')
AXIS([fig_min_p fig_max_p -50 50]);
subplot(1,1,1) % This should be 3,1,3
hold on
```

```

plot(time,(-0.5*ch1-15),'blue');
plot(time,pitch,'red');
%plot(time,0,'green');
plot(time,model_pitch,'-.black');
ylabel('elevator vs pitch')
xlabel('time');
title('PITCH');
AXIS([fig_min_p fig_max_p -30 40]);
legend(['Elevator command'], 'Pitch measured', 'Pitch estimated', 'Location', 'NorthWest')

%*****
%          ROLL      second order      *
%*****
%From identification toolbox
% Process model with transfer function
%
% G(s) = ----- * exp(-Td*s)
%           (1+Tp1*s)(1+Tp2*s)
%
% with    K = -0.86384+-0.22486
%        Tp1 = 1.5484+-0.51272
%        Tp2 = 0.152+-0.10915
%        Td = 0+-0.099414
%
% Estimated using PEM from data set rollde
% Loss function 96.7722 and FPE 99.2695
% Created:      06-Jun-2007 13:06:10
% Last modified: 06-Jun-2007 13:06:14

%PADEAPPROX
%[NUM,DEN]=PADE(timedelay,order)
timedelay_r=.02;
exp_r=exp(-timedelay_r*tf([1 0],1));

corr_roll=-5;
Kr=-1.16384;      %+-0.22486
T1r=1.5484;       %+-0.51272
T2r=0.152;         %+-0.10915
Tdr=0;             %+-0.099414
Tt=0.1; %Sampling time
SYS_roll = tf(Kr,[T1r*T2r (T1r+T2r) 1]);
%Adding timedelay
SYSC_roll =SYS_roll*exp_r
model_roll=LSIM(SYSC_roll,[ch5],[0:0.1:0.1*(length(ch5)-1)]);
model_roll=model_roll-corr_roll;
%figure(4);
%bode(SYSC_pitch);
%find maximum gain
%[Ap,Bp,Cp,Dp]=tf2ss(Kp,[1 Tp]);

```

```

%P_pitch=0; %Pole at origin
%Kpp = PLACE(Ap,Bp,P_pitch)
%Setting PID_roll parameters
Kpk_r=8; %simulation gives a unstable system when Kpk_p
w180_r=12/(20-5);%s %foud from plot;
Tk_r=w180_r%;(2*pi)/(w180_p);
%Using Zeiger Nicholds values for PID with Ti=0
Kp_r=0.5*Kpk_r;
Ti_r=0.01;
Td_r=0.12*Tk_r;

%PLOTT Aileron vs. Roll
fig_min_r=295;
fig_max_r=322;
% fig_min_r=200;
% fig_max_r=500;
figure(2)
subplot(3,1,1)
plot(time,-ch5,'blue');
AXIS([fig_min_r fig_max_r -100 100]);
ylabel('aileron')
subplot(3,1,2)
plot(time,roll,'red');
ylabel('roll')
AXIS([fig_min_r fig_max_r -50 50]);
subplot(1,1,1) % This should be 3,1,3
hold on
plot(time,(-0.3*ch5-55),'blue');
plot(time,roll,'red');
%plot(time,0,'green');
plot(time,model_roll,'-.black');
ylabel('aileron vs roll')
xlabel('time');
title('ROLL');
AXIS([fig_min_r fig_max_r -90 40]);
legend(['Aileron command'], 'Roll measured', 'roll estimated', 'Location', 'SouthEast')

% %*****%
% %          YAW      second order      *
% %*****%
% %From identification toolbox
% %          K
% % G(s) = ----- * exp(-Td*s)
% %           (1+Tp1*s)(1+Tp2*s)
% %
% % with    K = -21.895+-3.6384
% %         Tp1 = 2.362+-73.568
% %         Tp2 = 2.3999+-75.626

```

```

% %      Td = 1.6801+-0.091171
% %
% % Estimated using PEM from data set mydatadve
% % Loss function 92.0738 and FPE 99.6675
% % Created:      06-Jun-2007 14:52:43
% % Last modified: 06-Jun-2007 14:52:48

%PADEAPPROX
%[NUM,DEN]=PADE(timedelay,order)
timedelay_rd=1.9;
exp_rd=exp(-timedelay_rd*tf([1 0],1));

corr_rudder=-520;
Krd=25;      %+-0.22486
T1rd=2.3;    %+-0.51272
T2rd=2.4;    %+-0.10915
Tdrd=0;       %+-0.099414
Tt=0.1; %Sampling time
SYS_rudder = tf(Krd,[T1rd*T2rd (T1rd+T2rd) 1]);
%Adding timedelay
SYSC_rudder =SYS_rudder*exp_rd
model_yaw=LSIM(SYSC_rudder,[ch1],[0:0.1:0.1*(length(ch1)-1)]);
model_yaw=model_yaw+corr_rudder;
%figure(4);
%bode(SYSC_pitch);
%find maximum gain
%[Ap,Bp,Cp,Dp]=tf2ss(Kp,[1 Tp]);
%P_pitch=0; %Pole at origin
%Kpp = PLACE(Ap,Bp,P_pitch)
%Setting PID_pitch parameters
Kpk_rd=1.1;           %simulation gives a unstable system when Kpk_p
w180_rd=7/(30-10);%s   %foud from plot;
Tk_rd=(2*pi)/(w180_p)*0.5;
%Using Zeiger Nicholds values for PID with Ti=0
Kp_rd=0.5*Kpk_rd;
Ti_rd=0.01;
Td_rd=0.12*Tk_rd;

%PLOTT Rudder vs. Yaw
fig_min_rd=467;
fig_max_rd=474;
% fig_min_rd=200;
% fig_max_rd=500;
figure(3)
subplot(3,1,1)
plot(time,-ch1,'blue');
AXIS([fig_min_rd fig_max_rd -100 100]);
ylabel('rudder')

```

```

subplot(3,1,2)
plot(time,yaw,'red');
ylabel('rudder')
AXIS([fig_min_rd fig_max_rd -50 50]);
subplot(1,1,1) % This should be 3,1,3
hold on
plot(time,(-0.5*ch5+330),'blue');
plot(time,yaw,'red');
plot(time,0,'green');
plot(time,model_yaw,'-.black');
ylabel('rudder vs yaw')
xlabel('time');
title('YAW');
AXIS([fig_min_rd fig_max_rd 320 470]);
legend(['Rudder command'] , 'Yaw measured' , 'Yaw estimated' , 'Location' , 'NorthWest')

%Write all values to MATLAB
Kp_p=0.5*Kp_p           %To reduce over shoot
Td_p=0.5*Td_p
Kp_r=0.7*Kp_r
Td_r
Kp_rd=0.7*Kp_rd
Td_rd
{

```

B.4 Motor analysis

PLOT GENERATOR

```

close all
clear all
%motorkurver

%laster data ENDRE HER FOR å KJELLE FORSJELLIGE MOTORER
run KD2822
poly=3;

%regner om til SI
w=2*pi*rpm./60;
T=9.81*(radi/1000000)*tm;

%regner ut effekt og virkningsgrad
pin=a.*v;
pout=T.*w;
n=pout./pin;
F=f*9.81/1000;

%finner maksimalverider
rpmMax=max(rpm);
aMax=max(a);

```

```

vMax=max(v);
fMax=max(f);
tMax=max(tm);
TMax=max(T);
pinMax=max(pin);
poutMax=max(pout);
FMax=max(F);

%lager polynomer
[polya,x]=polymake(a/aMax,rpm,rpmMax,poly);
[polypin,x]=polymake(pin/pinMax,rpm,rpmMax,poly);
[polypout,x]=polymake(pout/pinMax,rpm,rpmMax,poly);
[polyf,x]=polymake(f/fMax,rpm,rpmMax,poly);
[polyT,x]=polymake(T/TMax,rpm,rpmMax,poly);
[polyn,x]=polymake(n,rpm,rpmMax,poly);

%plotter
hold on
xlabel('RPM')
ylabel('Normalized')
ylim([0 1.1])
grid on
title(motordata)

plot(rpm,a/aMax,'-.oblock')
plot(x,polya,'black')

plot(rpm,pin/pinMax,'-.ored')
plot(x,polypin,'red')

plot(rpm,pout/pinMax,'-.obblue')
plot(x,polypout,'blue')

plot(rpm,f/fMax,'-.ogreen')
plot(x,polyf,'green')

plot(rpm,T/TMax,'-.oyellow')
plot(x,polyT,'yellow')

plot(rpm,n,'-.omagenta')
plot(x,polyn,'magenta')
legend(['Ampere*' num2str(aMax) '[A]'], 'polyA', ['WattIn*' num2str(pinMax) '[W]', 'polyPin', ['WattOut*' num2str(pinMax) '[W]', 'polyPout', ['Thrust*' num2str(FMax) '[N]', 'polyF', ['Tourqe*' num2str(TMax) '[Nm]', 'polyT', 'Efficiency', 'polyEff', 'Location', 'NW'])

```

Appendix C

SIMULINK diagrams

C.1 Nonlinear model

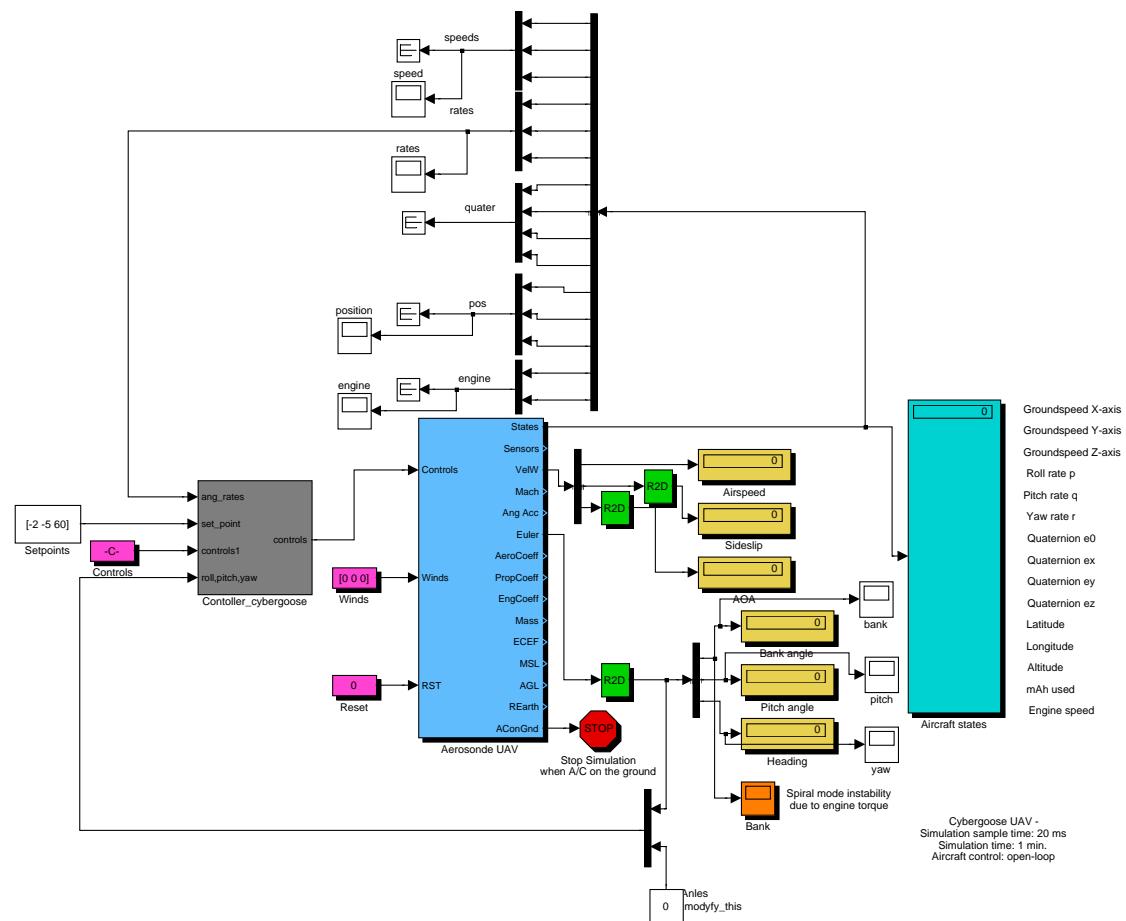


Figure C.1: Nonlinear simulink model

C.2 Second order model

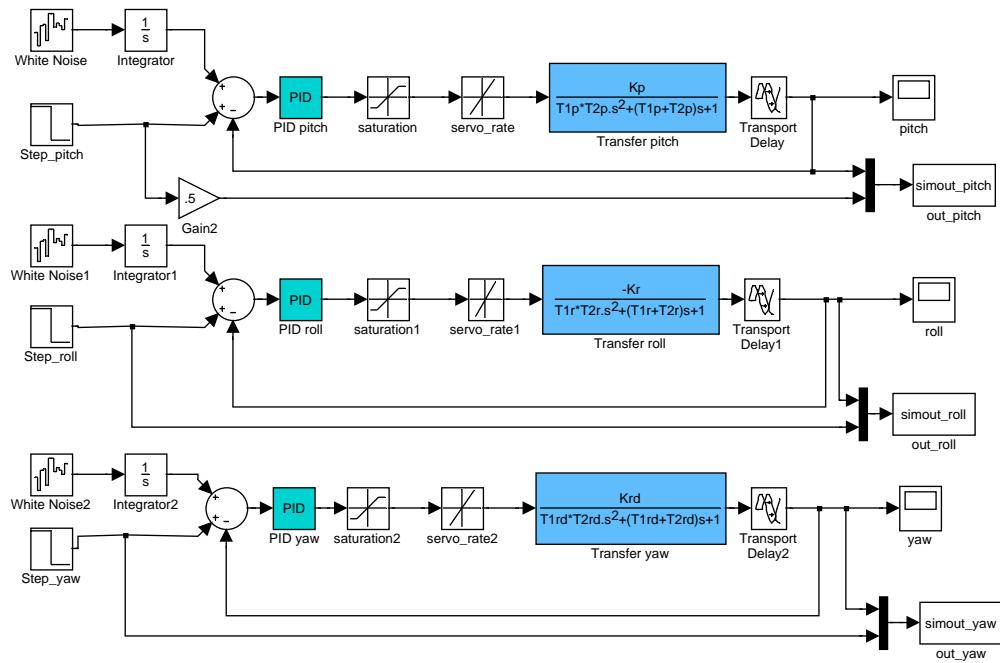


Figure C.2: Secondorder simulink model