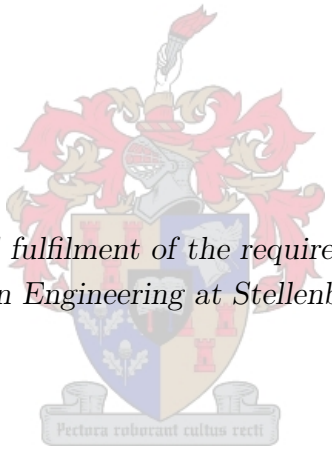
Autonomous Flight Control System for an Airship

by

Gerrit Christiaan Avenant



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering at Stellenbosch University

Supervisor: Prof. Willem Hermanus Steyn

Department of Electrical and Electronic Engineering

March 2010

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2010

Abstract

In recent years, the use of airships has become popular for observation purposes since they provide a cost effective alternative to other aircraft. For this project a lateral and longitudinal flight control system are required for waypoint navigation flight of an 8m long, non-rigid airship. The airship's actuators include a rudder, elevator and a propulsion system which can be vectored longitudinally.

Two airship models are evaluated for this project. A chosen model is linearised and a modal analysis is done. The modal analysis is compared to a previous modal study done on the YEZ-2A airship and is found to compare well. Each airship mode is discussed and the linear behaviour is compared to the behaviour of the non-linear model.

A fuzzy logic controller design approach was undertaken for the design of speed, heading and height controllers. These non-linear controllers were designed for the non-linear model, due to the following reasons:

- Fuzzy logic controllers show tolerance to model inaccuracies.
- Complexity of design is simple.
- Controllers can be adjusted intuitively.
- Fuzzy logic controllers can be combined with conventional control techniques.

Simulation results showed adequate lateral and longitudinal performance, even when subjected to light wind conditions and disturbances.

The inertial measuring unit implemented in a previous project is used and additional hardware is designed and implemented for the control of the airship's actuators. Several improvements are made to the groundstation software to allow for activation of different controllers as well as for setting up the desired flight plan.

The controller performance is tested through flight tests and shows adequate performance as well as controller potential. Although further work is still required for improving the controllers' performance, this thesis acts as a platform for future research.

Opsomming

In die afgelope paar jaar het die gebruik van die lugskepe gewild geword vir waarnemings doeleindes aangesien dit 'n koste effektiewe alternatief vir ander lugvaartuie bied. In hierdie projek word 'n laterale en longitudinale beheerstelsel benodig vir merker navigasie vlugte met 'n 8m lang, nie-rigiede lugskip. Die lugskip se aktueerders sluit in 'n rigtingroer, hoogteroer asook 'n aandrywing stelsel wat oorlangs gestuur kan word.

Vir hierdie projek is twee lugskip modelle geïmplementeer. Die gekose model is gelineariseer en 'n modale analise is gedoen. Die modale analise is met 'n vorige modale studie vir die YEZ-2A lugskip vergelyk en wys soortegelyke linieêre gedrag. Die lugskip modusse is bespreek en die linieêre gedrag word met die gedrag van die nie-linieêre model vergelyk.

Daar is op 'n fuzzy logiese beheerder ontwerp besluit vir die ontwerp van spoed, rigting en hoogte beheerders. Hierdie nie-linieêre beheerders is ontwerp vir die nie-linieêre model a.g.v. die volgende redes:

- Fuzzy logiese beheerders toon toleransie vir modellering of meetfoute.
- Kompleksiteit van die ontwerp is eenvoudig.
- Beheerders kan intuïtief aangepas word.
- Fuzzy logiese beheerders kan met konvensionele beheertegnieke gekombineer word.

Simulasie resultate toon voldoende werkverrigtinge, selfs in die teenwoordigheid van ligte wind sowel as ander versteurings.

Die inersiële metings eenheid, wat geïmplementeer is in 'n vorige projek, is gebruik en addisionele hardeware vir die beheer van die lugskip is aktueerders is ontwerp en geïmplementeer. Talle verbeterings is aangebring aan die grondstasie sagteware vir die aktiveer van die beheerders sowel as die uitleg van die gekose vlugplan.

Die beheerders se werksverrigtinge is getoets gedurende vlugtoetse en toon voldoende beheer vermoë sowel as beheerder potensiaal. Alhoewel verdere werk steeds nodig is vir die verbetering van die beheerders, dien hierdie tesis as 'n platform vir toekomstige navorsing.

Acknowledgements

I would like to thank and acknowledge the following people who have contributed to make this project possible:

- Prof WH Steyn for his guidance, suggestions, problem solving and hands on approach
- My parents for their love and support
- My girlfriend Karla for her love and support and handling all my stress
- Danie for being the airship project partner in crime
- Friends inside and outside the lab including: AM, Chris, all the Ruans, Alwyn, Iwan, Lionel
- Quintis, Johan, Andre and the rest of the SED staff for the technical support they provided.

"Ruiter van die windjie wil ek bly, vryer as die voëltjies rondom my"
Bles Bridges

Contents

List of Figures

Nomenclature

Symbols

α_F, α_e	Geometric and effective angles of attack
ϵ	Axial distance from the nose of the airship
ϵ_v	Axial position at which the flow ceases to be potential
η	Crosss Drag Flow Efficiency factor accounting for finite length of airship
η_n	Efficiency factor for added mass of fins
η_d	Efficiency factor of flap
$\delta_T, \delta_{TA}, \delta r, \delta e$	Thrust, Thrust Vector Angle, Rudder and Elevator change with respect to the initial condition
α	Angle of attack
β	Sideslip angle
γ	Angle between airships centerline and velocity vector
ϕ, θ,ψ	Roll angle, Pitch angle, Yaw angle
ρ_a	Air density
τ	Theoretical Flap Effectiveness Factor
$\boldsymbol{\tau}$	Indicates 6x1 vector containing forces and moments, in and about the $x-$, $y-$ and $z-$axis
ζ	Damping Coefficient
A, B	State and input matrices of the linear model
b	Fin semi span
b_e	Semi Fin Span
c, c_f	The chord of the fin and flap respectively
D	Maximum diameter of the airship
$\hat{\mathbf{g}}$	Unit vector in direction of gravity
g	Acceleration due to gravity
I	Identity Matrix
J	Inertia Matrix
L	The length of the airship
m	Total mass of the airship
p	Rotation speed around the X-axis in the Body Axis

q	Rotation speed around the Y-axis in the Body Axis
r	Rotation speed around the Z-axis in the Body Axis
$\mathbf{r}_g$	Position of the C.G. relative to the C.V
$\mathbf{r}_t$	Position of the propulsion
$\mathbf{M}$	6x6 Mass matrix
$\mathbf{M}_A$	6x6 Added Mass Matrix
$\mathbf{M}_{AH}$	6x6 Added Mass Matrix due to the hull
$\mathbf{M}_{AF}$	6x6 Added Mass Matrix due to the fins
R	Hull Cross Sectional Radius
S_f	Flap area
S_{fe}	Exposed fin area
S_H	Surface area of the hull
u	Velocity along the X-axis in the Body Axis
v	Velocity along the Y-axis in the Body Axis
w	Velocity along the Z-axis in the Body Axis
V	Volume of the airship's hull
$\boldsymbol{\upsilon}$	Velocity vector, $[u,v,w]^T$
$\boldsymbol{\omega}$	Angular rates vector, $[p,q,r]^T$
C_{DC}	Cross flow drag coefficient
C_D	Drag Coefficient
$C_{p\alpha}$	Pressure coefficient of the fin
$C_{L\alpha}$, $C_{l\alpha}$	3D and 2D lift curve slopes of the fins

Abbreviations

1D, 2D, 3D	One Dimensional, Two Dimensional, Three Dimensional
C.B.	Centre of Bouyancy
C.V.	Centre of Volume
C.G.	Centre of Gravity
COA	Centre of area method
DCM	Direct Cosine Matrix
FIFO	First In First Out
GPS	Global Positioning System
IMU	Inertial Measuring Unit
NED	North East Down
OBC	Onboard Computer
PWM	Puse Width Modulation
R/C	Radio Controlled
RF	Radion Frequency
RTOS	Real Time Operating System

Chapter 1

Introduction

1.1 Historic Background

Airships were man's first realisation and implementation of controlled flight, with the first airship design proposed by Jean-Baptiste Meusnier in 1784 [3]. The invention of the steam powered engine in the 19th century sparked the beginning of the airship industry, as it was capable of providing the power needed for sustained flight.

The airship industry achieved great success in the beginning of the 20th century, especially the German Luftschiff Zeppelin company. Airships became the aircraft of choice for long distance commercial flights and were also used during the First World War.

During the late 1930's, airships became obsolete, due to key advances made with regard to the development of the aeroplane. Some airship accidents, such as the infamous Hindenburg disaster in 1937, dented its image in the public eye. Therefore airships were abandoned as a way of travel in the 1940's.

Since the 1980's, airships slowly started to gain popularity again. This is due to the fact that airships require less energy then conventional aircraft to stay afloat, since they use aerostatic forces instead of aerodynamic forces for vertical lift. This makes it suitable for a wide variety of modern day applications, including advertising, surveillance and environmental monitoring.

1.2 Motivation

The airship hold certain advantages that are not met by airplanes or helicopters, as listed in [4]. Some of the main advantages are:

- Low fuel consumption - fuel is only required for maneuvering, not to float in the air. The airship uses its aerostatic (buoyancy) force instead of aerodynamic forces to stay afloat.
- Long endurance flights are therefore also possible.
- Hovering capabilities as the airship does not need to move to stay in the air.

- Vertical Take Off and Landing (VTOL) - making it possible to use in areas without long runways.
- Low vibration due to smaller motors required, and therefore sensor noise is reduced.
- An electric system also opens up the possibility for the use of solar energy in the future.

These advantages have created many modern applications for airships, especially for earth monitoring purposes.

Airships provide a low cost alternative for aerial advertising, such as at sporting events. They can also be used for security purposes, as high risk areas can easily be monitored from above. Environmental monitoring is another useful application, as airships has the ability to stay afloat over sensitive areas without causing an interference in the environment through movement and noise. The possible use of airships for planetary exploration is also currently investigated by NASA's Jet Propulsion Lab [5].

High altitude platform systems (HAPS) provide a viable future for airships. Several projects have been launched to investigate the possibility of using airships as a substitute for low-orbit satellites and communication platforms. This aspect is discussed in the project done by Danie Fourie [6].

The use of electrical airships is fast becoming a possibility and since no additional fuel is required, the weight is reduced substantially. This technology also simplifies the airships dynamics, as there is no change in weight due to fuel consumption. This option only started to become practical recently since great advances has been made in battery technology such as Lithium polymer (LiPo) batteries which are becoming smaller and more powerful.

A number of recent airship projects are discussed in Chapter 2.

1.3 Airship Used

Figure 1.1: The FZ800E airship in flight.

The airship/blimp that was used throughout this project is the non-rigid FZ-800E, which is also illustrated in Fig. 1.1. Table 1.1 lists the basic characteristics of this airship.

Length	8 m
Maximum Diameter	1.9 m
Fineness Ratio (Length/diameter)	4.2
Volume	14.2 m^3
Total Lift Capability (Helium)	17.2 kg

Table 1.1: FZ-800E specifications.

The airship was filled with helium, as helium is a lighter than air gas and therefore ideal for providing the necessary buoyancy force. Although hydrogen is a more cost effective solution and also provides more lift, as it poses an increased safety risk and requires greater handling care. More of this airship's properties can be found in Addendum A.

1.4 Project outcomes

The initial outcomes targeted of this project were:

1. Model a simplified aerodynamic model.
2. Identify changeable parameters of the model.
3. Simulation of the airship model.
4. Develop a direction, position and speed controller for autonomous flight.
5. Flight tests to verify the model and the controller.

These outcomes are addressed in this thesis and the outline for this thesis is as follows:

- Chapter 2 presents a brief overview of recent airship projects around the world.
- Chapter 3 discusses the implementation of two different airship models.
- Chapter 4 discusses the linearisation of the model, which is used to gain additional insight on certain aspects of the model. The results of the modal analysis is also shown, as well as comparisons to the non-linear model.
- Chapter 5 discusses the design of the airship's controllers. Simulations results are also shown.
- Chapter 6 discusses the hardware and software environment on which the controllers have been implemented.
- Chapter 7 discusses the practical results for the controllers, which was obtained by flight tests.
- Chapter 8 concludes this thesis and provides an overview of the work done, as well as recommendations and improvements for the future.

Chapter 2

Literature Study

This chapter summarises a number of recent airship projects and discusses the modelling and control aspects of each project.

2.1 The Airship "Lotte"

The "Lotte" airship is a project undertaken by the University of Stuttgart.

Figure 2.1: The Airship 'Lotte'. [7]

The airship will be used as a platform for multi-disciplinary investigation into several fields, from aerodynamics mentioned in [7], to solar powered propulsion systems. The airship used in this study has a length of 15-m, a volume of 107.42-m^3 and a maximum payload of 12-kg.

2.1.1 Model

The modeling of the flight mechanical model included system identification techniques [8] to obtain the flight dynamical characteristics of the airship. These methods are assumed to be adequate, since this model corresponds well to actual flight data.

2.1.2 Control

Instrumentation for inflight measurements and autonomous control is discussed in [9] as a way to validate the models and implement the controllers. Accurate orientation and position information is obtained by measuring the following attributes:

1. Position through use of GPS.
2. Horizontal attitude through use of an electronic compass.
3. The vertical attitude, longitudinal accelerations and angular rates through use of an inertial measuring unit (IMU).
4. Air temperature and 3D-velocity through use of an Ultra Sonic Anemometer.
5. Helium temperature as well as pressure.

Several controllers were designed and the focus was placed on the design of a complex robust controller to compensate for the inaccuracies in the model [10].

2.2 YEZ-2A

This doctorate [11] was undertaken at the Cranfield Institute of Technology and was performed on the YEZ-2A airship, which has a length of 129.5-m and a volume of 107.42-m^3. The aerodynamic model is determined from wind-tunnel tests. The aerodynamic coefficients are stored in a database and are accessed in a simulation model by using lookup tables. The simulation was implemented in Advanced Continuous Simulation Language (ACSL) and the model linearised using a ACSL function.

The modal and stability characteristics are determined from this model and are also analysed. This airship has a fineness ratio (length/diameter) of 4.2, which makes the aerodynamic database useful for airships with a similar fineness ratio and shape, such as the AURORA project as discussed in Section 2.3. The stability and control characteristics of the airship is also discussed in [12], which gives a good overview of the airships behaviour in simulation.

2.3 Project AURORA

Through the AURORA project the Information Technology Institute of Campinas (Brazil) aims to create an autonomous unmanned airship for environmental monitoring purposes.

The non-rigid airship used is 9-m long and has a 2.25-m diameter. This results in a total volume of 30-m^3, which means this airship is capable of a payload of 10-kg.

Figure 2.2: The AURORA I AS800. [13]

2.3.1 Model

This airships model was obtained by adapting the YEZ-2A's wind tunnel data from [11]. Although the airships are very different in size, they have the same shape and share a fineness ratio close to 4.2.

2.3.2 Control

The controller's initial design was based on a simple four layer design [4]. On the bottom layer is the basic actuators that control the airship. The secondary level is for the actuator controllers,controlling the actuators according to the decisions made by the tertiary level. This level decides on the actuators to be activated and which commands are to be given. A fourth level for autonomous control is still being developed and will compute the flight plans for the specific mission.

The lateral/directional controller implemented in [14], consists of two controllers. The first is a proportional integral path trajectory controller, which takes the position error between the desired and actual path and provides a desired heading angle to a secondary heading controller which consists of a proportional and derivative controller. The controller was validated using simulation and proved to be sufficient for waypoint tracking. The gains were obtained through trial and error or through H-inifinity optimization approach. The controllers were implemented successfully, with the H-infinity controller working especially well during wind disturbances.

The special topic of hover control was also covered in [15], since airships are underactuated during hover as the rudder and elevator provide little control when moving at low speeds. This article provides a simple longitudinal and lateral controller using an image sensor to determine the offsets from the original position. Further work on hover control for the AURORA project is presented in [16] where a robust controller is discussed to solve the lateral underactuation problem. Control signal saturation limits are investigated and several control techniques are used to stabilise the airship, including aligning itself against the wind and minimising the lateral force for better actuation.

A further investigation into the performance of three nonlinear controllers is covered in [17].

Although the linear quadratic regulator (LQR) controller discussed showed good results with regards to path following even during wind disturbances, linear control has limitations as the controller is only valid for small disturbances about its designed equilibrium. Therefore gain scheduling, dynamic inversion, as well as backstepping control were investigated for path tracking performance, robustness, computational efficiency and controller parameter tuning.

Wind disturbance was modelled by the Dryden spectrum which consists of a turbulent gust component added to a constant mean value.

2.3.3 Other

Another spinoff from the AURORA project is an web-based Java airship simulator. Unfortunately the simulator's website was down at the time of writing, but the simulator is discussed in [18]. This simulator was intended as a pilot training tool - although the airship would be flown autonomously, a human pilot is needed to take manual control if the onboard computer should fail.

The concept and components of an air-ground-cooperative ensemble are discussed in [19], which also gives an a overview of the AURORA project.

2.4 The LAAS-CNRS Autonomous Blimp Project

This project by the LAAS-CNRS[20], focussed on autonomous control and terrain mapping, with the long term outcome of creating aerial robots which can provide mission related information to robotic ground units. The electric, non rigid airship (Airspeed AS-500) is 8-m long with a maximum diameter of 1.9-m, thus giving a volume close to 15-m^3 and thus also a fineness ratio of 4.25. This airship has a maximum payload of 3.5-kg.

Figure 2.3: The Airship "Karma", AS-500. [20]

2.4.1 Model

While Newton's laws were used for the dynamic modelling, the identification of the aerodynamic model is discussed in the French report [21] and is derived from Munk's notes on airship hull aerodynamics [22], as well as [23] which discusses the evaluation of inertia coefficients.

2.4.2 Control

The flight control system consists of three layers. The first layer is the motion planner which calculates the required trajectory and reference path. The secondary layer, the flight controller, computes the control signal inputs required by the actuators to perform the required manoeuvres. The final layer is the state monitor, which estimates the current state of the airship from the predicted state as well as measured data.

The control strategy is also discussed in [20]. As there is little coupling between the lateral and longitudinal dynamics, the model is split and separate lateral and longitudinal controllers were designed. Four different flight phases are defined - Takeoff, longitudinal navigation, lateral navigation and landing, with the necessary transition phases between these phases. The navigation and control strategy for each of these phases is discussed with the focus on lateral navigation, as well as a velocity controller which regulates the velocity for the specific navigation phase.

2.4.3 Other

Terrain mapping is also discussed in [20]. A map building technique using a set of images, together with the required algorithms like stereovision, interest points recognition and visual motion estimation, are discussed.The results show that it is possible to accurately position an airship using stereovision exclusively, as well as the possibility to build high resolution environment maps.

2.5 NASA/JPL Titan Aerobot Project

A dense atmosphere has been discovered on Titan, one of Saturn moons, NASA and the Jet Propulsion Laboratory at the University of California has been investigating the possibility of using an airship for planetary surface exploration [24]. The airship is based on the Airspeed airship AS-800B, which has a length of 11-m and a maximum diameter of 2.5-m, giving it a fineness ratio of 4.4. The total volume is 34-m^3 and the airship can carry a payload of 12-kg.

2.5.1 Model

A new nonlinear airship model intended for control system design is presented in [26]. This model combines the results of previous models and adds further features. It is also parametrically defined for easy design implementation. This model was programmed using the ADAMS modelling software and is further discussed in [27]. The information for this model could not be obtained and therefore not incorporated in this study.

Figure 2.4: The JPL Aerobot during testflight. [25]

2.5.2 Control

The controller is based on a structured layered approach. The airships trajectory is divided in three modes: loiter, hover and cruise. Each of these modes use a combination of four controllers: ascent, descent, turn and altitude controllers. These controllers set a desired attitude and thrust required which are commanded to the actuators. The pitch, yaw and altitude controllers were implemented as proportional integral controllers in the initial version.

In [25] a number of controllers from different airship projects are mentioned and two new controllers are added to the list: a sequential-loop-closure(SLC) controller based and a hybrid classical/linear-quadratic-regulator (LQR). The LQR controller outperformed the SLC controller for full 3D flight manoeuvres, although the SLC controller performed slightly better for pure lateral movement.

Since the airship will be flown autonomously on a different planet it should be able to create the appropriate trajectory for the specific mission. Trajectory generation and path planning is discussed in [5]. Trajectory planning in 2D and 3D is done using linear interpolation methods and cubic splines, which is further developed into piecewise Bezier curved trajectories. A method for creating a trajectory for the visual mapping of an area is also explained.

2.6 McGill University

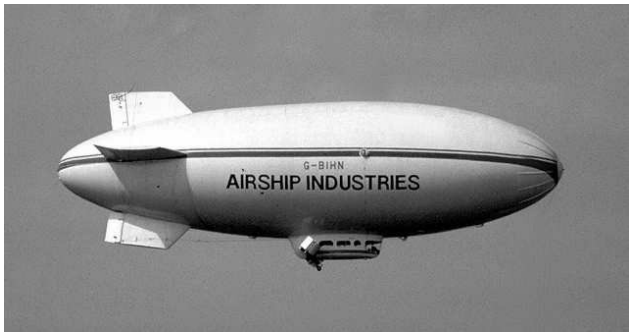


Figure 2.5: Skyship 500 [1]

2.6.1 Model

A PhD thesis [1], done at the McGill University in Canada, presents two derived models for an airship. The first model is based on a rigid body assumption with the aerodynamic effects on the model is determined computationally. This model is verified by comparing simulation results to existing flight data and wind tunnel results.

The second model incorporates aeroelastic effects of the airship, including the effects of bending deformations and structural elasticity on the aerodynamics. This elastic model implemented for the Skyship 500, illustrated in Figure 2.5 and simulation results were used to obtain the influence of aerolastic effects on the aerodynamics.

Chapter 3

Airship Model

3.1 Introduction

Two models were implemented in this study: the rigid body model presented by Yuwen Li in [1], as well as the model present by S.B.V. Gomes in [11]. This chapter discusses the implementation of these models which differ mainly on the definition of the aerodynamic forces. These models were implemented in a Matlab/Simulink environment in the form of S-functions.

3.2 Axis and Notation

3.2.1 Inertial Reference Axis

An inertial axis system is needed as a point of reference for the airship's position and orientation. In order to use a fixed-earth inertial axis system, it is assumed that the earth is flat and non-rotating. The popular North-East-Down(NED) axis system is chosen for this purpose. The x-axis is positive towards the north, while the y-axis is positive towards the east. The z-axis is perpendicular to the xy-plane and is positive towards the centre of the earth.

3.2.2 Airship Body Axis

To describe the airship's position and its orientation with regards to the inertial frame, an additional axis needs to be established on the airships body.

A right-hand body axis system is defined at the center of volume (C.V.) of the airship, with the x-axis increasing along the centerline of the hull, the y-axis increasing in the starboard direction and the z-axis increasing downward, perpendicular to the xy-plane as illustrated in Figure 3.1. The linear body velocities ($u,\ v,\ w$) and the angular body velocities ($p,\ q,\ r$) are defined according to the right-hand rule, is also shown on Figure 3.1.

Defining the origin at the C.V. differs from the usual convention which defines the center of gravity (C.G.) of a rigid body. The origin is chosen at the C.V. due to several reasons. Firstly, to simplify the calculations for the additional mass experienced by airships as explained in 3.5.2. This

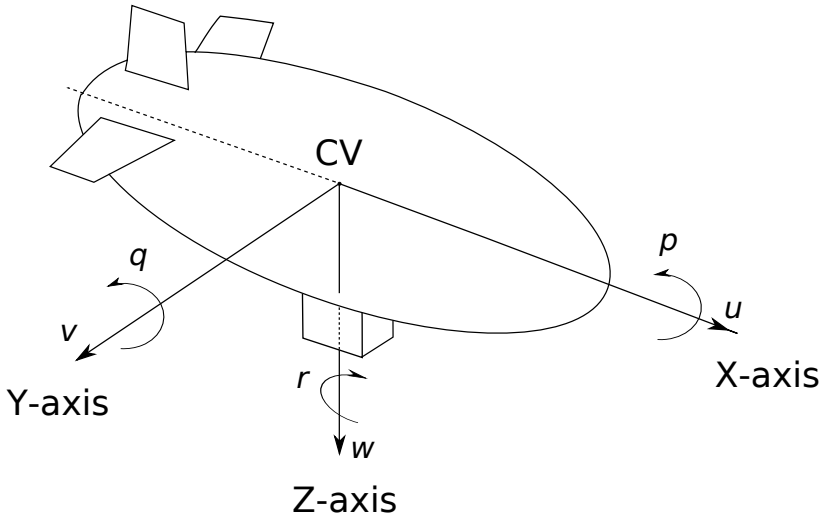


Figure 3.1: The body-axis of the airship

is also the convention for airships which uses ballonets to store the helium, due to possible shift of the C.G. during flight, since the weight of the gas inside changes due to change in pressure with height.

The following velocity vector angles which are used throughout the modeling are shown in Figure 3.2 and are the angle of attack angle,

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{3.1}$$

and the sideslip angle

$$\beta = \tan^{-1}\left(\frac{v}{u}\right) \tag{3.2}$$

and total angle between the velocity vector and the x-axis,

$$\gamma = \tan^{-1}\left(\frac{\sqrt{v^2 + w^2}}{u}\right) \tag{3.3}$$

3.2.3 Changing from Body Axis to Inertial Axis System

A rotation matrix is needed to change between the body axis and the inertial axis system . The Euler 3-2-1 rotation system was, where the rotations are described by a yaw (ψ) rotation, followed by pitch(θ) rotation and then a roll (ϕ) rotation.

These rotations are illustrated in Figure 3.3 and can be mathematically described by Eqs. 3.4, 3.5 and 3.6 respectively:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{3.4}$$

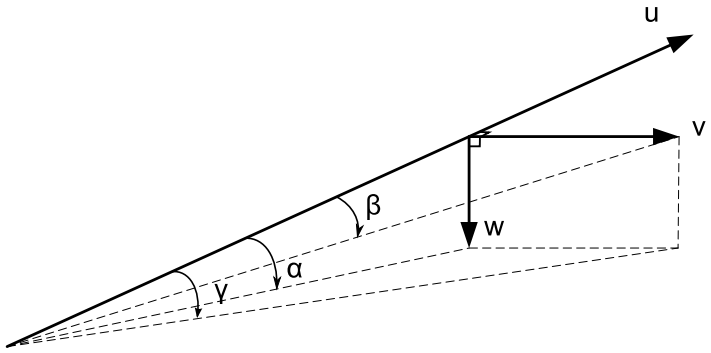


Figure 3.2: Additional velocity vector angles defined

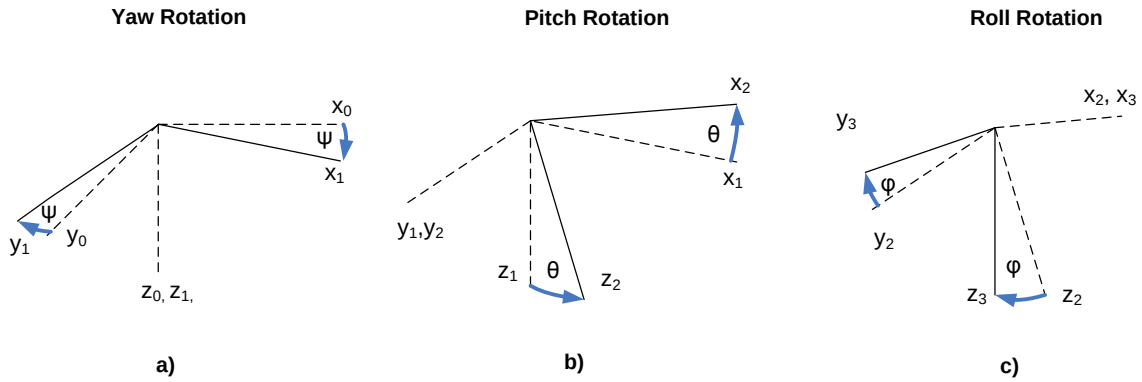


Figure 3.3: Euler angle 3-2-1 rotations

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{3.5}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{3.6}$$

The transformation matrix used to transform from inertial frame to body frame, also called the direction cosine matrix (DCM), can be defined as the multiplication of the rotation matrices in Eqs. 3.4,3.5 and 3.6:

$$\mathbf{T}_{DCM} = \mathbf{T}_\phi \mathbf{T}_\theta \mathbf{T}_\psi \tag{3.7}$$

To transform from the body frame to the inertial frame, the inverse of the DCM can be used.

3.3 Control Surfaces

The control surfaces and actuators are shown in Figure 3.4. The rudder and elevator control surfaces are configured in a '+' configuration, which reduces the coupling between elevator and

rudder control. Thrust can be vectored longitudinally around the shaft connected to the rotors. The control surfaces are as follows:

δ_E	Elevator Control Surface. Positive deflection causes the elevator to point upwards.
δ_R	Rudder Control Surface. Positive deflection turns the rudder towards the port.
δ_T	Thrust Control Surface. Positive increase is a positive increase in thrust. Negative or reverse thrust is not possible with the current propulsion system.
δ_A	Thrust Angle Control Surface. Positive deflection causes a negative rotation about the rotors shaft.

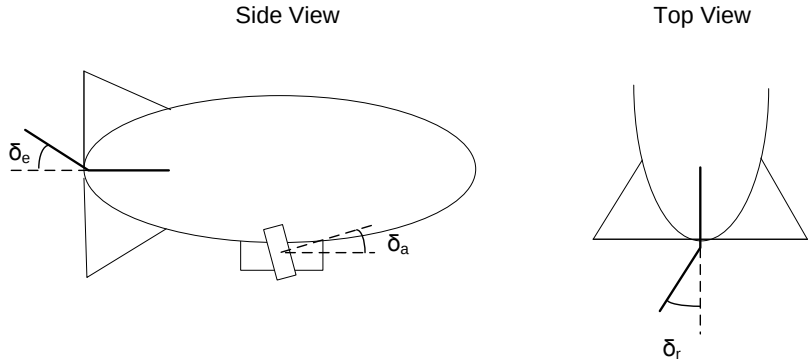


Figure 3.4: The different control actuators on the airship.

3.4 Equations of Motion

Using Newtons second law, the six equations of motion can be summarised as the summation of following vectors:

$$\boldsymbol{M}_{6\times6}\dot{\boldsymbol{\upsilon}} = \boldsymbol{\tau}_I + \boldsymbol{\tau}_G + \boldsymbol{\tau}_{AS} + \boldsymbol{\tau}_A + \boldsymbol{\tau}_P, \tag{3.8}$$

where

$\mathbf{M}_{6\times6}$	The mass matrix as discussed in Section 3.5 as well as the added mass due to aerodynamic forces.
$\boldsymbol{\upsilon}$	The 6×1 vector of linear (u, v, w) and angular velocities (p, q, r) as shown in Figure 3.1.
$\boldsymbol{\tau}_I$	The 6×1 vector containing the inertial forces and moments (due the Coriolis and Centrifugal forces as discussed in Section 3.6.
$\boldsymbol{\tau}_G$	The 6×1 vector containing the forces and moments due to gravity as discussed in Section 3.7.
$\boldsymbol{\tau}_{AS}$	The 6×1 vector containing the forces and moments due to aerostatic (buoyant force) as discussed in Section 3.7.
$\boldsymbol{\tau}_P$	The 6×1 vector containing the forces and moments due to the propulsion of the airship as discussed in Section 3.8
$\boldsymbol{\tau}_A$	The 6×1 vector containing the forces and moments due to the aerodynamics of the airship. The aerodynamic forces according to the Yuwen Li model are discussed in Section 3.9 and the S.B.V. Gomes model discussed in Section 3.10.

The following assumptions were made to simplify the model of the airship as far as possible without losing to much information.

- The airship's physical mass stays constant during flight.
- To approximate the airship's hull as a rigid body, all aero-elastic effects on the hull will be ignored.
- The airship is symmetric about the oxz-plane.
- The centre of gravity, as well as the centre of buoyancy lies in the oxz plane.

3.5 Mass Matrix

The 6 × 6 mass matrix is composed of the standard mass matrix as well as the additional added mass matrix. Therefore,

$$\boldsymbol{M}_{6\times6} = \boldsymbol{M}_{standard} + \boldsymbol{M}_{added}. \tag{3.9}$$

3.5.1 Standard Mass Matrix

The standard mass matrix is given by

$$\mathbf{M}_{standard} = \begin{bmatrix} m\mathbf{I} & -m\mathbf{r}_g^{\times} \\ m\mathbf{r}_g^{\times} & \mathbf{J} \end{bmatrix} \tag{3.10}$$

where $\mathbf{m}$ is the total physical mass of the airship and $\mathbf{J}$ the inertia matrix. The $\mathbf{r}_g^{\times}$ matrix is the position vector (r_x, r_y, r_z) of the C.G. in its skew-symmetric matrix form and is defined as

$$\mathbf{r}_g^{\times} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{3.11}$$

Multiplication with a skew-symmetric matrix corresponds to the cross product with the corresponding vector.

3.5.2 Added Mass

Added mass is a result of the airship accelerating through the air. This aerodynamic property can be described as the units of force (Newton) the airship experiences per units acceleration($m.s^{-2}$) it undergoes and therefore has the units of mass (kilograms), as shown by

$$\frac{N}{m.s^{-2}} = \frac{kg.m.s^{-2}}{m.s^{-2}} = kg. \tag{3.12}$$

Added mass is composed of two 6×6 matrices $\boldsymbol{M}_{AH}$ and $\boldsymbol{M}_{\boldsymbol{AF}}$, which refer to the added mass contributions of the hull and of the fins, thus

$$\boldsymbol{M}_{added} = \boldsymbol{M}_{AH} + \boldsymbol{M}_{AF} \tag{3.13}$$

To calculate the added mass of the hull, it is approximated as a ellipsoid of revolution. Since the origin of the body axis was chosen at the C.V., the added mass calculation is simplified so that only the diagonal elements of $\boldsymbol{M}_{\boldsymbol{AH}}$ needs to be calculated. The matrix is:

$$\boldsymbol{M}_{\boldsymbol{AH}} = \begin{bmatrix} k_{axial}V\rho_a & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{transverse}V\rho_a & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{transverse}V\rho_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{rotation}I' & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{rotation}I' \end{bmatrix} \tag{3.14}$$

where k_{axial}, $k_{transverse}$ and $k_{rotation}$ are the added mass factors; V and ρ_a are the volume and density of the displaced air; and I' the moment of inertia of the displaced gas around the x-axis.

The added mass factors are illustrated in Figure 3.5 and are a function of the fineness ratio (Length/Diameter). The analytical method for calculating the added mass factors can be found in [28], but reading the coefficients of the graph in Figure 3.5 is sufficient.

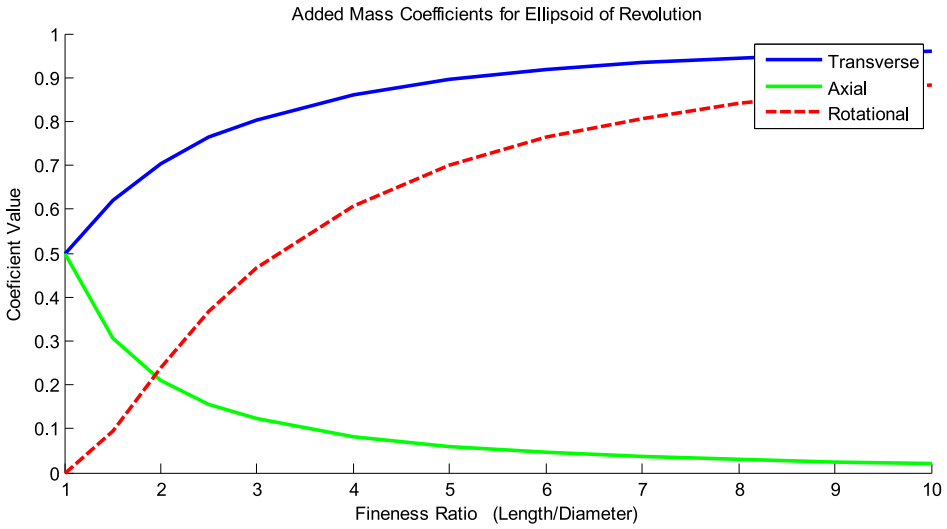


Figure 3.5: Added Mass Factors

The k-factors are estimated as

$$k_{axial} = 0.09 \tag{3.15}$$

$$k_{transverse} = 0.89 \tag{3.16}$$

$$k_{rotational} = 0.62 \tag{3.17}$$

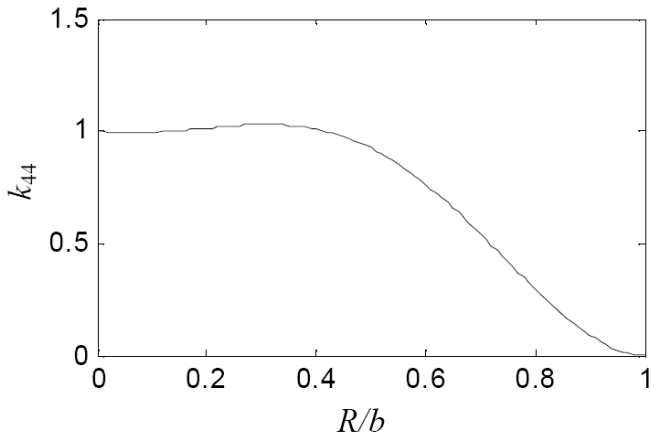


Figure 3.6: Effect of Body Radius on Fin Factor k_{44} (from [1])

The added mass due to the fins can be calculated by first calculating the coefficients:

$$\bar{m}_{f22} = \bar{m}_{f33} = \rho_a \pi (b - \frac{R^2}{b^2})^2 \tag{3.18}$$

$$\bar{m}_{f44} = \frac{2}{\pi} k_{44} \rho_a b^4 \tag{3.19}$$

where R is the radius from the center line to the fin and b is the fin semi-span, as illustrated in Figure 3.7. The body fin factor k_{44} is a function of R/b [29] and is therefore estimated at

$$k_{44} = 1 \tag{3.20}$$

for this case. These coefficients are then integrated over the fin area in Figure A.1, where x_{fe} and x_{fs} are the start and end positions of the fins on the x-axis:

$$m_{f22} = m_{f33} = \eta_f \int_{x_{fs}}^{x_{fe}} \bar{m}_{f22} dx \tag{3.21}$$

$$m_{f44} = \eta_f \int_{x_{fs}}^{x_{fe}} \bar{m}_{f44} dx \tag{3.22}$$

$$-m_{26} = m_{35} = -\eta_f \int_{x_{fs}}^{x_{fe}} \bar{m}_{f22} dx \tag{3.23}$$

$$m_{f55} = m_{f66} = \eta_f \int_{x_{fs}}^{x_{fe}} \bar{m}_{f22} x^2 dx \tag{3.24}$$

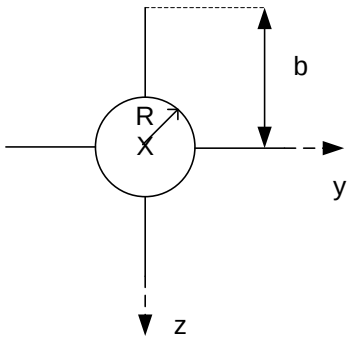


Figure 3.7: FZ-800E Fin Cross Section

These coefficients then make up the added mass contribution of the fins as shown in

$$\boldsymbol{M}_{AF}^{\times} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{f22} & 0 & 0 & 0 & m_{26} \\ 0 & 0 & m_{f33} & 0 & m_{35} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{f55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{f66} \end{bmatrix} \tag{3.25}$$

An efficiency factor η_f is also estimated to account for 3-D effects on the fins and is a function of the fins aspect ratio as illustrated in Figure 3.9. The aspect ratio can be calculated by

$$AR = \frac{(2b_e)^2}{S_{FE}} \tag{3.26}$$

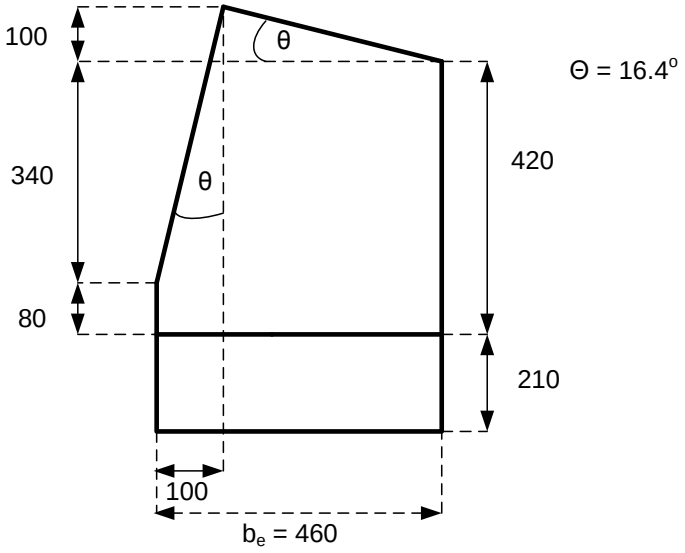


Figure 3.8: FZ-800E Fin Area

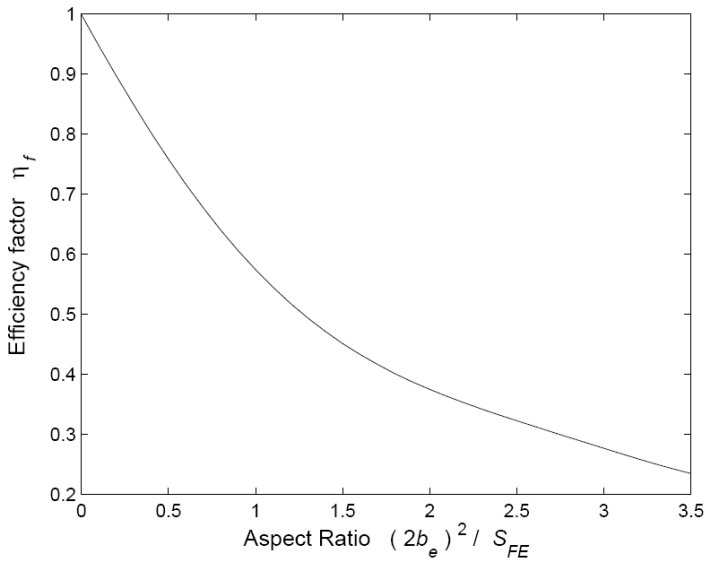


Figure 3.9: Fin efficiency factors

where b_e is the corresponding semi fin span and S_{FE} the fin area. This leads to a efficiency factor of

$$\eta_f = 0.45 \tag{3.27}$$

3.6 Inertial Forces

The Coriolis and Centrifugal forces are virtual forces that are experienced in a rotating axis system with reference to an inertial reference frame.

The inertial forces that arise from working with the body-axis, can be calculated using,

$$\tau_I = -\begin{bmatrix} M_{(1:3,1:3)}\boldsymbol{\omega}^\times & \boldsymbol{\omega}^\times M_{(1:3,4:6)} \\ M_{(4:6,1:3)}\boldsymbol{\omega}^\times & \boldsymbol{\omega}^\times M_{(4:6,4:6)} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} \tag{3.28}$$

where $\boldsymbol{\omega}^\times$ corresponds to the skew symmetric matrix of the rotation rates $\boldsymbol{\omega}$.

3.7 Buoyancy and Gravity

The gravitational force is:

$$\boldsymbol{F}_G = mg\hat{\boldsymbol{g}} \tag{3.29}$$

where m is the total mass, g the constant $9.81m/s^2$ gravitational acceleration and $\hat{\boldsymbol{g}}$ the vector towards the earth. Due to the fact that the origin of the body axis is not at the C.G., additional moments due to gravity is experienced:

$$\boldsymbol{\tau_G} = \begin{bmatrix} \boldsymbol{F}_G \\ \boldsymbol{r}_g^\times \boldsymbol{F}_G \end{bmatrix} \tag{3.30}$$

The buoyancy force depends on the density of gas used to fill the airship. Helium (He) provides a lift of $10.359N/m^3$ while Hydrogen (H) a lift of $11.183N/m^3$. By filling with hydrogen instead of helium, an airship with a volume of $14m^3$ can lift roughly $1.2kg$ more.

The aerostatic force is:

$$\boldsymbol{F}_{AS} = -\rho_a g V \hat{\boldsymbol{g}} \tag{3.31}$$

where ρ_a is density of air, V the volume of the airship, g the gravitational acceleration with the $\hat{\boldsymbol{g}}$ the vector towards the centre of the earth.

There are no moments due to aerostatic lift, because the buoyancy force works in on the C.V. Neutral buoyancy is assumed to simplify the model. This implies that the buoyancy force is equal the the gravitational force,

$$\boldsymbol{F}_{AS} = -\boldsymbol{F}_G \tag{3.32}$$

3.8 Propulsion

The propulsion forces and moments vector is:

$$\boldsymbol{\tau}_p = \begin{bmatrix} \boldsymbol{F}_T \\ \boldsymbol{r}_T^\times \boldsymbol{F}_T \end{bmatrix} \tag{3.33}$$

where $\boldsymbol{r}_T^x$ is the skew symmetric form of the vector to the position of the propulsion system. $\mathbf{F}_T$

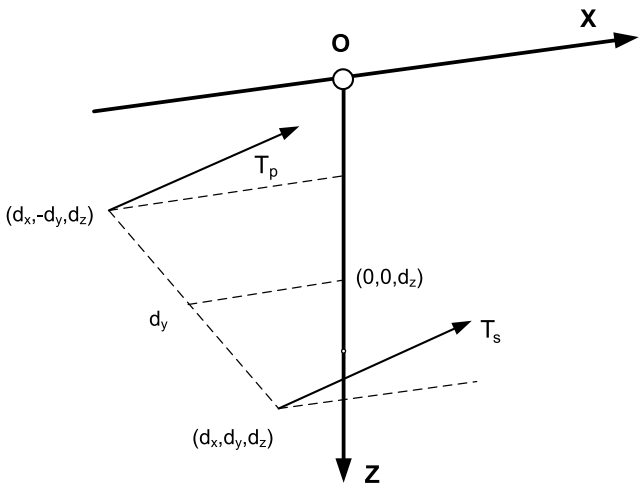


Figure 3.10: Propulsion system position and vector angle geometry

is the vector of thrust forces, given by

$$\boldsymbol{F}_T = \begin{bmatrix} F_T \cos \delta_{TA} \\ 0 \\ -F_T \sin \delta_{TA} \end{bmatrix} \tag{3.34}$$

where F_T is the magnitude of the total thrust in Newton and consists of the port and starboard thrust, therefore

$$F_T = T_s + T_p \tag{3.35}$$

The propulsion system geometry is illustrated in Figure 3.10, while the results for the thrust test are illustrated in Figure 3.11.

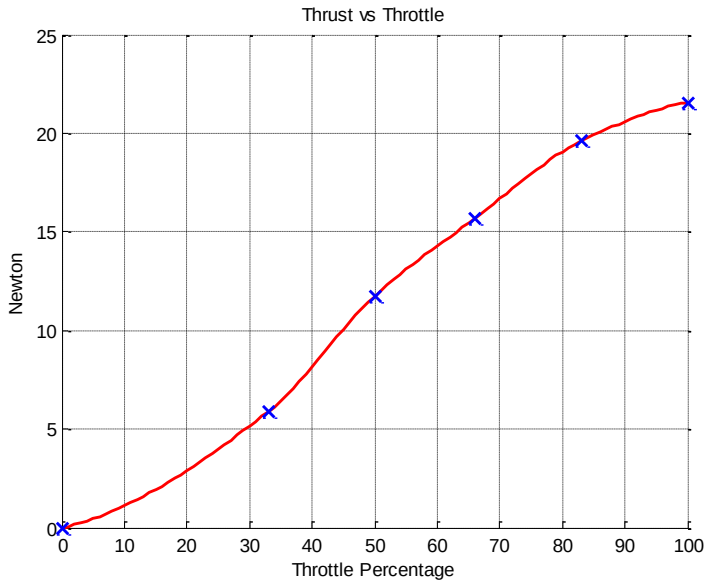


Figure 3.11: Thrust measured vs commanded throttle

3.9 Aerodynamic Forces - Yuwen Li

In this section the aerodynamic forces, as presented in [1], are discussed. The aerodynamic forces are a summation of the aerodynamic forces on the hull of the airship, as well as on the fins.

3.9.1 Munk Moment

The Munk moment tends to destabilize the airships in pitch and yaw. It is best described by [30]:

"Any shape other than a sphere generates a moment when inclined in an inviscid flow. The Munk moment arises because of the asymmetric location of the stagnation points, where the pressure is highest on the front of the body (decelerating flow) and lowest on the back (accelerating flow). Due to this fact, the Munk moment is always destabilizing, in the sense that it acts to turn the vehicle perpendicular to the flow."

The Munk moment can be added to the moment equations and is mathematically described by

$$\boldsymbol{M}_{Munk} = -\boldsymbol{\upsilon}^{\times}(M_{11} \times \boldsymbol{\upsilon}) \tag{3.36}$$

3.9.2 Viscous Forces on the Hull

The viscous forces can be attributed to the viscosity of the hull and are normal to the centerline(x-axis) of the airship. The magnitude of the force can be calculated as

$$F_{VN} = -q_o \sin 2\gamma.(k_{transverse} - k_{axial}) \int_{\epsilon_v}^{L} \frac{dS}{d\epsilon} d\epsilon + q_o \eta_{dc} C_{dc} \sin^2 \gamma \int_{\epsilon_v}^{L} 2R d\epsilon \tag{3.37}$$

where

C_{dc} is the cross-flow drag coefficient of an infinte cylinder,

η_{dc} is an efficiency factor accounting for the finite length of the airship

q_o is the dynamic pressure,

γ is the angle between the x-axis and velocity vector,

k_{axial}, $k_{transverse}$ are the added mass factors discussed in Section 3.5.2,

S is the local cross-sectional area

R is the local cross-sectional radius

ϵ is the distance from the nose of the airship and

L is the length.

The coefficient C_{dc} and the efficiency factor η_{dc}, which is a function of the fineness ratio, can be found in [31] and are estimated as

$$C_{dc} = 1.2 \tag{3.38}$$

$$\eta_{dc} = 0.62 \tag{3.39}$$

The point where the flow ceases to be potential is at position ϵ_v, where

$$\epsilon_v = 0.378L + 0.527\epsilon_1 \tag{3.40}$$

in which ϵ_1 is the position measured from the nose of the airship where the surface area decreases the most, thus where $\frac{\partial S}{\partial \epsilon}$ is the most negative. This position was estimated as

$$\epsilon_1 = 6.5[m] \tag{3.41}$$

and therefore

$$\epsilon_v = 6.45[m] \tag{3.42}$$

By substituting in all the values for this implementation and integrating over the specified area, this equation simplifies to:

$$F_{VN} = 0.89q_o \sin 2\gamma + 0.792q_o \sin^2 \gamma \tag{3.43}$$

This force is then split up in the lateral as well as vertical components so that

$$\boldsymbol{F_{VN}} = F_{VN} \left[0, \frac{-v_v}{\sqrt{v_V^2 + w_V^2}}, \frac{-w_v}{\sqrt{v_V^2 + w_V^2}} \right]^T \tag{3.44}$$

.

The equation for the magnitude of the moments due to the viscous effect is

$$M_{VN} = -q_o \sin 2\gamma.(k_{transverse} - k_{axial}) \int_{\epsilon_v}^{L} \frac{dS}{d\epsilon}(\epsilon_m - \epsilon)d\epsilon + q_o \eta_{dc} C_{dc} \sin^2 \gamma \int_{\epsilon_v}^{L} 2R(\epsilon_m - \epsilon)d\epsilon \tag{3.45}$$

where ϵ_m is the distance from the nose of the airship to the C.V., and is

$$\epsilon_m = 3.6[m] \tag{3.46}$$

With all the required values substituted in, this equation simplifies to

$$M_{VN} = -3.28q_o \sin 2\gamma - 2.89q_o \sin^2 \gamma \tag{3.47}$$

This is used to calculate the pitching and yawing moments as follows:

$$\boldsymbol{M_V} = M_{VN} \left[0, \frac{w_v}{\sqrt{v_V^2 + w_V^2}}, \frac{-v_v}{\sqrt{v_V^2 + w_V^2}} \right]^T \tag{3.48}$$

These forces and moments have a small influence on the behaviour of a small airship, but will have a much greater effect on very large airships.

3.9.3 Normal Forces on the Fins

The forces on the fin are a summation of the force distribution over fin area, where the force on a specific point on the fin is given by

$$dF_{NF} = q_o \Delta C_{p\alpha}(x, s) \alpha_e dx ds \tag{3.49}$$

The pressure coefficient was assumed constant as the wing is rectangular, and can be obtained from [32] and [33] and is

$$\Delta C_{p\alpha} = 0.05 unit/^o \tag{3.50}$$

The effective angle of attack is defined as,

$$\alpha_e = \frac{C_{L\alpha}}{C_l \alpha}(1 + \frac{R^2}{s^2})\alpha_F \tag{3.51}$$

where $(1 + \frac{R^2}{s^2})$ accounts for the effect of the hull and $\frac{C_{L\alpha}}{C_l \alpha}$ represents 3D-effects. The latter is obtained from [34] as a function of the aspect ratio,

$$\frac{C_{L\alpha}}{C_l \alpha} = \frac{AR}{AR + 2} \tag{3.52}$$

and therefore

$$\frac{C_{L\alpha}}{C_l \alpha} = 0.43 \tag{3.53}$$

The geometric angle of attack of the fin, α_F, can be calculated by

$$\alpha_{Fx} = \tan^{-1} \frac{v_{nx}(1/4c)}{u(1/4c)} \tag{3.54}$$

where v_{nx} is the normal air velocity on the fin. This velocity can be calculated by estimating the local velocity on each fin, as show in Figure 3.12. The local velocities on the four fins are:

$$v_{n1} = 0.72p + 3.75r - v_w \tag{3.55}$$

$$v_{n2} = -0.72p + 3.75r - v_w \tag{3.56}$$

$$v_{n3} = -0.72p - 3.75q - w_w \tag{3.57}$$

$$v_{n4} = 0.72p - 3.75q - w_w \tag{3.58}$$

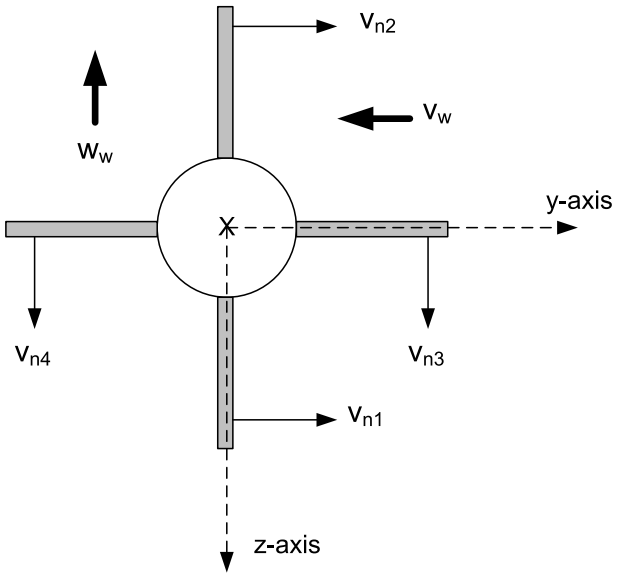


Figure 3.12: The local velocities on the fins

Using these local velocity equations, as well Equation 3.55, an angle of attack can be calculated for all of the fins, which results in α_{F1}, α_{F2}, α_{F3} and α_{F4}. If the geometric angle of attack is larger than the stall angle (α_{stall}), then the value of α_{stall} used in the equation. The stall angle was estimated as 26.3^o by using the following approximation in [35],

$$\alpha_{stall} = 15(1 + \frac{1.1}{AR}) \tag{3.59}$$

Therefore the normal forces on a fin can be given by

$$F_{NF} = q_o \alpha_{Fx} F_{nfc} \tag{3.60}$$

where F_{nfc} is

$$F_{nfc} = \Delta C_{p\alpha} \frac{C_{L\alpha}}{C_l \alpha} \int_{s_s}^{s_f} (1 + \frac{R^2}{s^2}) dx ds \tag{3.61}$$

The total force in the y-axis and z-axis is given by

$$\boldsymbol{F_{NF}} = q_o F_{nfc} \begin{bmatrix} 0 \\ (\alpha_{F1} + \alpha_{F2}) \\ (\alpha_{F3} + \alpha_{F4}) \end{bmatrix} \tag{3.62}$$

where q_o is the dynamic pressure and is $q_o = 0.5 \rho_{air} U^2$ where U is the airspeed of the airship. The moment due to these fin forces can be calculated by using the moment arm to each corresponding fin. This results in the following moments:

$$\boldsymbol{M}_{NF} = q_o \begin{bmatrix} r_{fx}F_{nfc}(\alpha_{F3} - \alpha_{F1} - \alpha_{F4} + \alpha_{F2}) \\ r_{fz}F_{NF(3)} \\ -r_{fz}F_{NF(2)} \end{bmatrix} \tag{3.63}$$

where r_{fx} is the position of the fins from the x-axis, and the r_{fz} is the position of the fins from the z-axis.

3.9.4 Force acting on the hull due to the fins

Figure 3.13 shows the forces which act perpendicular to the centerline of the hull, as a result of fin induced downwash. This plot relates to flight at an airspeed of 3m/s and an angle of attack of 20^o. It can be seen from the figure that these forces are very small and can therefore be considered negligible.

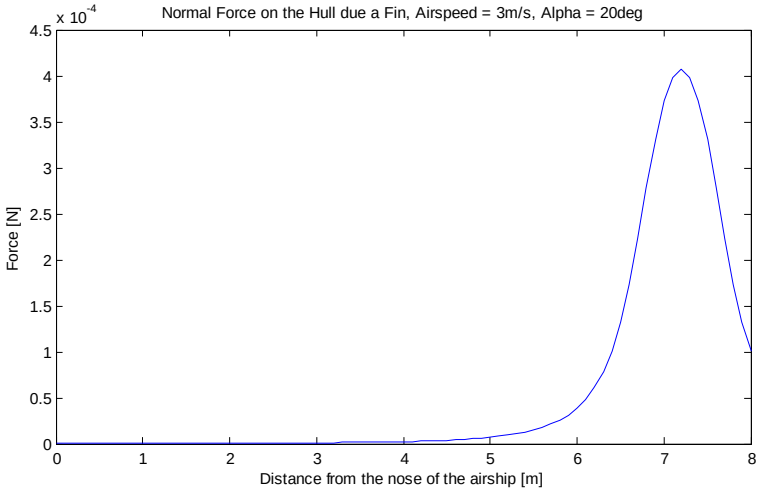


Figure 3.13: Normal force acting on hull due to the fins

For completeness sake, the equations for calculating these forces are stated below, but are further discussed in Section 2.3.4 in [1]. The force per unit length on the hull is defined as:

$$f_{H(F)} = \pi\rho R^2 V \frac{dw_{dF}}{dx} \tag{3.64}$$

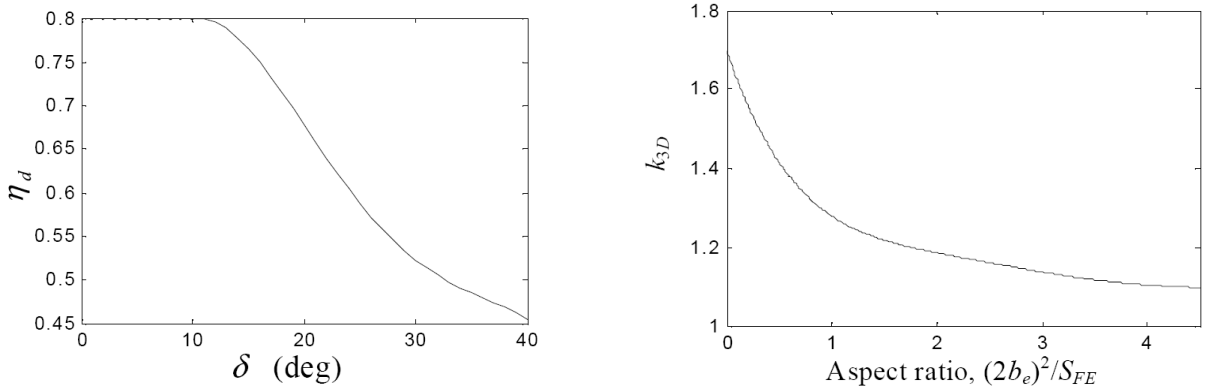
The local fin-induced downwash is given by

$$w_{df}(x) = V\frac{C_{L\alpha}}{C_l\alpha}\alpha_v \int_{x_s}^{x_f} \int_R^b [8\pi d_f(d_f - x + x_f)]^{-1} \Delta C_{p\alpha}(x_f, s)(1 + \frac{R^2}{s^2}) ds dx_f \tag{3.65}$$

where

$$d_f = \sqrt{(x - x_f)^2 + s^2} \tag{3.66}$$

3.9.5 Forces from control surface deflection



(a) Flap Efficiency

(b) 3D Effectiveness

Figure 3.14: Flap efficiency and 3D effectiveness factors

The lift generated by the deflection of the control surface is given by

$$\Delta C_L = C_{L\alpha}\tau\eta_d k_{3D}\delta \tag{3.67}$$

where k_{3D} accounts for the 3D-effect and is a function of the aspect ratio, η_d accounts for the fins efficiency, and is a function of the deflection angle. Both these coefficients are illustrated in Figure 3.14. The effectiveness factor τ can be calculated by

$$\tau = 1 - \frac{\theta_f - \sin(\theta_f)}{\pi} = 0.82 \tag{3.68}$$

$$\theta_f = \cos^{-1}(\frac{2c_f}{c} - 1) = \frac{\pi}{2} \tag{3.69}$$

where the flap chord c_f and airfoil chord c is measured

$$c_f = 0.21 \tag{3.70}$$

$$c = 0.42 \tag{3.71}$$

The drag due to surface deflection is given by

$$\Delta C_D = 1.7(\frac{c_f}{c})^{1.38}(\frac{S_F}{S_{FA}})\sin^2\delta = 0.0127\sin^2\delta \tag{3.72}$$

where S_F is the flap area and S_{FA} the fin area. These areas are:

$$S_F = 0.097m^2 \tag{3.73}$$

$$S_{FA} = 0.483m^2 \tag{3.74}$$

$$\tag{3.75}$$

The influence of the deflection on pitching moment due to deflection is very small and is given by

$$\Delta C_{m_{\frac{1}{4}}} = \Delta C_L \frac{2\sin\theta_f - \sin 2\theta_F}{8(\pi - \theta_F + \sin\theta_F)} = -0.097\Delta C_L \tag{3.76}$$

The forces due to rudder deflection are given by

$$\boldsymbol{F_{rud}} = q_o S_F \begin{bmatrix} -\Delta C_D \\ \Delta C_L \\ 0 \end{bmatrix} \tag{3.77}$$

and the forces due to elevator deflection are given by

$$\boldsymbol{F_{elv}} = q_o S_F \begin{bmatrix} -\Delta C_D \\ 0 \\ \Delta C_L \end{bmatrix} \tag{3.78}$$

The moments due to the rudder and elevator forces are also added to the simulation, and can be calculated by multiplying the corresponding force with the moment arm r_{fz}, which is the distance between the fins and the C.V.

3.9.6 Axial drag

The axial drag is modeled as

$$\boldsymbol{F_{Drag}} = \left[-q_o C_D S_H / \cos^2\alpha, 0, 0\right]^T \tag{3.79}$$

The drag coefficient C_D is obtained from the windtunnel data for the Gomes model [11] and can be described as a function of the angle of attack, α. The reference area of the hull can be estimated by using the volume of the airship, therefore

$$S_H = V^{\frac{2}{3}} \tag{3.80}$$

The axial drag due to the fins is negligibly small compared to that of the hull because of the very small reference area of the fins. The fins drag coefficient can be modeled as that of a small rectangular plate, which is much smaller than that of the hull. Drag due to fin deflection is already included in Section 3.9.5.

The lift and side forces are not specified directly in the Yuwen Li model and it assumed to be contained in the other aerodynamic forces.

3.9.7 Forces and moments during a turn

The forces on the airship during a turn are briefly mentioned, as explained in Section 2.6 in [1]. These forces and moments are shown in Figure 3.15. The following forces and moments assist the turn:

1. Force and moment due to rudder deflection ($F_{\delta r(y)}$,$M_{\delta r(z)}$).
2. The Munk moment.

The following forces and moments counteracts the turn:

1. Forces and Moments due to the viscosity of the hull ($F_{V(y)}$,$M_{V(z)}$).
2. Normal Forces and moments due to the fins ($F_{NF(y)}$,$M_{NF(z)}$).

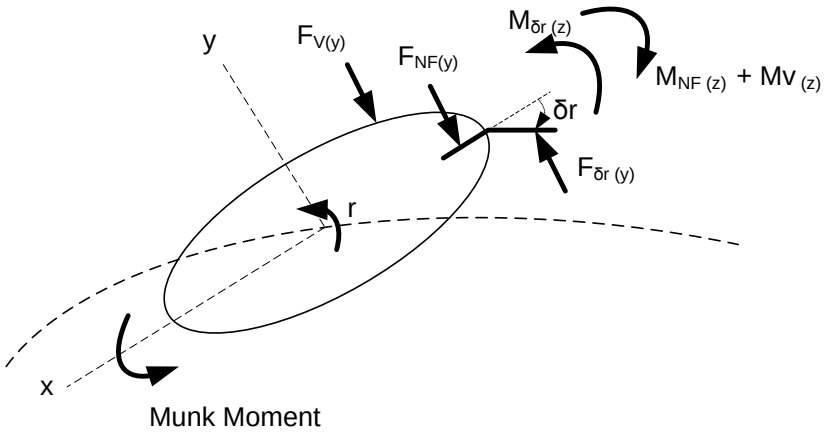


Figure 3.15: Turning forces and moments during a steady turn. Adapted from [1]

3.10 Aerodynamic Forces - S.B.V Gomes

The aerodynamic forces are estimated by consulting lookup tables of the aerodynamic coefficients as determined by wind tunnel tests on a scale model of the YEZ-2A. This should provide a fairly accurate representation, since the airship used in thus study has a similar fineness ratio.

The aerodynamic forces and moments are

$$T_A = q_o \begin{bmatrix} C_D \times V^{\frac{2}{3}} \\ C_Y \times V^{\frac{2}{3}} \\ C_L \times V^{\frac{2}{3}} \\ C_l \times V \\ C_m \times V \\ C_n \times V \end{bmatrix} \tag{3.81}$$

where $C_d, C_y, C_L, C_l, C_m, C_n$ are the aerodynamic coefficients which can be obtained from the wind tunnel data, and V is the volume of the airship.

Each coefficient is determined by searching for the value in the tables which corresponds to the airships current angle of attack (α), sideslip angle (β), rudder (δ_r) and elevator deflection (δ_e). Firstly, a coefficient's base value is looked up, based solely on the current angle of attack and slideslip angle. Thereafter the effect of the rudder's deflection as well as the elevator's deflection at a specific angle of attack and sideslip angle is included. The composition of the drag coefficient as an example is:

$$C_d(\alpha, \beta, \delta_r, \delta_e) = C_{d0}(\alpha, \beta, 0, 0) + \Delta C_{dr}(\alpha, \beta, \delta_r, 0) + \Delta C_{de}(\alpha, \beta, 0, \delta_e) \tag{3.82}$$

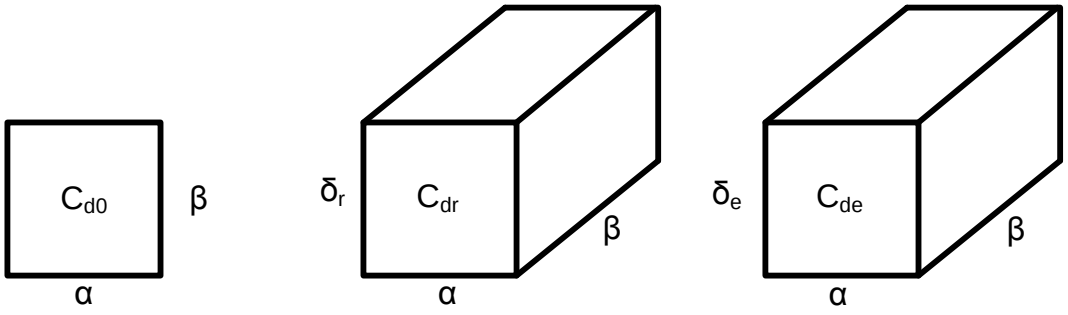


Figure 3.16: Obtaining aerodynamic coefficients from the tables as prescribed in [11]

Therefore, this method uses six 2D-matrix searches, as well as twelve 3D-matrix searches to look-up all the coefficients. By simply noting that the angle of attack and sideslip angle stays constant, the correct coefficient should be at the same coordinate in each 3D matrix. By simply reorganising the matrices as shown in Figure 3.17, a more efficient method of obtaining the coefficients can be implemented.

For this method, 3D matrices are created to contain the rudder or elevator contributions for all the aerodynamic coefficients based on the angle of attack and the control surface deflection angle. There are several 3D matrices for the rudder and elevator contributions, each matrix set specified at a different sideslip angle.

The aerodynamic coefficient values due to the rudder or elvator, can then be calculated by linear interpolation between the two corresponding sideslip matrices. This method reduces the total number of searches to five 2D-matrix lookups.

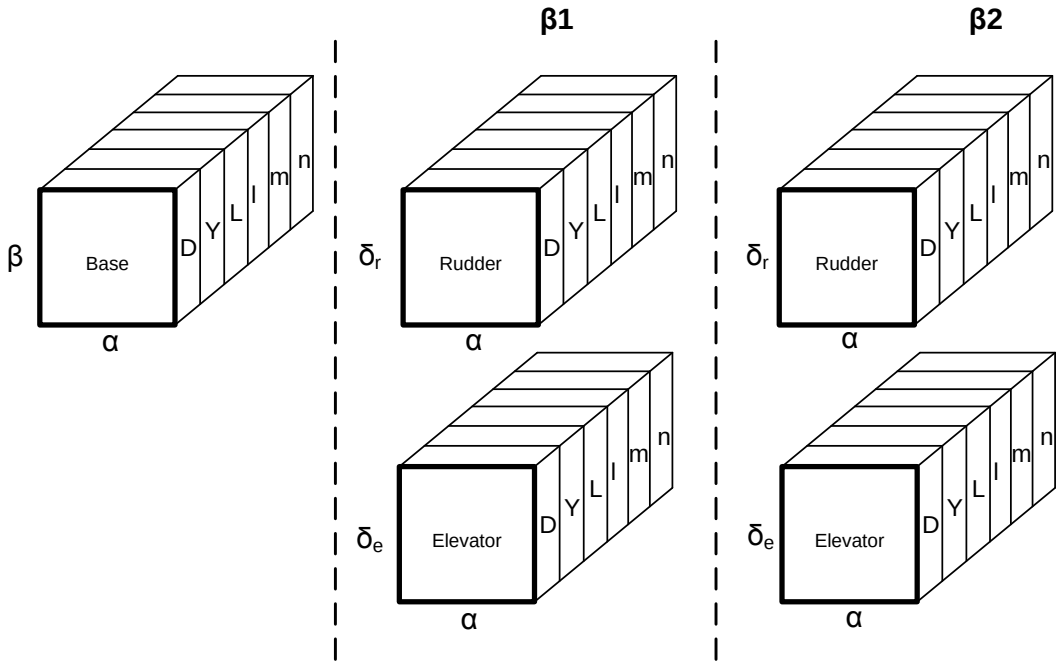


Figure 3.17: Obtaining aerodynamic coefficients from the tables

The following damping moments are added to the model as well:

$$L_p = q_o b^2 S_H l_p \tag{3.83}$$
$$M_q = q_o b^2 S_H m_q \tag{3.84}$$
$$M_w = q_o b S_H m_w \tag{3.85}$$
$$N_r = q_o b^2 S_H n_r \tag{3.86}$$
$$N_v = q_o b S_H n_v \tag{3.87}$$

where b the airship length and the aerodynamic coefficients are

$$l_p = -0.012 \tag{3.88}$$
$$m_w = 0.110 \tag{3.89}$$
$$m_q = -0.256 \tag{3.90}$$
$$n_v = -0.258 \tag{3.91}$$
$$n_r = -0.180 \tag{3.92}$$

3.11 The effect of wind

In [36], it is stated that wind does not effect the airship's dynamic model equations. The wind's effect can rather be modeled in terms of a relative air velocity under a constant wind. For more complex wind simulations, the Dryden turbulence model can also be used to simulate wind turbulence effects at higher altitudes. This model is discussed in [37], as well as [38].

The wind, specified in NED coordinates $[u_w, v_w, w_w]_{NED}$ needs to be converted to the body

axis.

$$\begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}_{body} = T_{DCM} \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}_{NED} \tag{3.93}$$

The airship is very susceptible to wind. In theory, the airship should theoretically tend to weathercock into the wind direction. However, this behaviour was not always experienced, since the airship tends to turn perpendicular. This could possibly be attributed to the Munk moment as described in Section 3.9. The airship is also assumed to be ellipsoidal, when in fact the hull has a larger surface area in front than at the back. This could also tend to counter the weathercock effect of the fins. Lateral control becomes increasingly difficult, especially with only the rudder as a lateral control actuator.

3.12 Conclusion

The Yuwen Li model is easier to adjust intuitively, since most of the forces can be adjusted to compensate for differences between the model and the actual behaviour. A study of the influence of certain parameters on the airship's behaviour for the Yuwen Li model is presented in the next chapter as well as adjustments that were made to this model.

Chapter 4

Stability and Modal Analysis

4.1 Introduction

The characteristics and modes of YEZ-2A airship which were investigated in [11] and [12] provide valuable insight into the behaviour of airships. In this chapter the model as presented by [1] is linearised and the airships behaviour is compared to that of the YEZ-2A airship. The influence of the different forces on the modes of the airship are investigated to gain a clearer understanding of the airship's behaviour and adjustments are made to adjust the lateral behaviour of the model.

4.2 Linearising the Equations of Motions

The non-linear model consists of several equations of motions F which consist of the state vector X and the input vector U, therefore,

$$\dot{\boldsymbol{X}} = F(\boldsymbol{X}, \boldsymbol{U}) \tag{4.1}$$

Equations for the roll and pitching rates can be added to the non-linear set of equations, therefore adding

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{4.2}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{4.3}$$

The full state vector consists of 8 states which describe the current linear and angular velocities, as well as the attitude of the airship:

$$\boldsymbol{X} = [u_o, v_o, w_o, p_o, q_o, r_o, \phi, \theta]^T \tag{4.4}$$

Both methods require an equilibrium state to be defined as the work point around which small perturbations will occur. For straight and level flight, the equilibrium state can be around a constant

axial velocity u_o, which would give the state vector

$$\boldsymbol{X}_{eq} = [u_o, 0, 0, 0, 0, 0, 0, 0]^T \tag{4.5}$$

To calculate the thrust required for the constant velocity, the axial forces equation can be used. The sum of the thrust and drag force must be zero, as no acceleration should be present, therefore

$$F_T - 0.5\rho C_d u_o^2 S_H = 0 \tag{4.6}$$

As the angle of attack should be zero for straight and level flight, the weight of the airship can be moved slightly forward, to create a pitching moment to oppose the pitching due to the thrust, as mentioned in [11].

Two different linearising techniques were used to obtain a linear representation from the non-linear model. In [1] the non-linear equations are first linearised around the equilibrium state $\boldsymbol{X_{eq}}$, by calculating the Jacobian of the non-linear equations with regards to the states:

$$\boldsymbol{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}|_{\boldsymbol{X}eq} \tag{4.7}$$

and with regard to the input parameters:

$$\boldsymbol{B} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}|_{\boldsymbol{X}eq} \tag{4.8}$$

This can be used to create a linearised model for small perturbations around $\boldsymbol{X_{eq}}$,

$$\Delta \dot{\boldsymbol{X}} = \boldsymbol{A}\Delta\boldsymbol{X} + \boldsymbol{B}\Delta\boldsymbol{U} \tag{4.9}$$

Numerical linearisation [39] was applied to the Gomes model. This was done as it is impossible to linearise it mathematically, since most of the aerodynamic data is contained in lookup tables. Numerical linearisation was also applied to the Yuwen Li model and the results compare well with the mathematical linearisation.

The linearised model is obtained by perturbing each state and measuring the increase in each acceleration component in the A-matrix. If a perturbation Δx is added to the states, the change in the non-linear equations is given by

$$\frac{\partial \boldsymbol{F}}{\partial x} = \frac{\boldsymbol{F}_{x+\Delta x} - \boldsymbol{F}_{x-\Delta x}}{2\Delta x} \tag{4.10}$$

A higher order approximation for this partial derivative is derived in [39] and is defined as

$$\frac{\partial \boldsymbol{F}}{\partial x} = \frac{8(\boldsymbol{F}_{x+\Delta x} - \boldsymbol{F}_{x-\Delta x}) - (\boldsymbol{F}_{x+2\Delta x} - \boldsymbol{F}_{x-2\Delta x})}{12\Delta x} \tag{4.11}$$

An algorithm for obtaining the linearised model numerically is also discussed in [39]. Firstly, an equilibrium state is chosen around which the model will be linearised, as well as the initial perturbation size for each of the states. Then a partial derivative is calculated for each non-linear

equation with regards to each state's perturbation using Equation 4.11. If the partial derivative does not converge, the perturbation is halved and the process repeated.

If convergence is not achieved within n iterations, the following changes can be made to the algorithm:

1. Number of iterations can be increased.
2. Initial perturbation size can be chosen to be smaller.
3. The tolerance set for convergence can be increased.

The algorithm can be summed up as follows:

Result: Linearised A-matrix
Equilibrium position for states;
Calculate suitable perturbation size for each input;
for *f non-linear equations* **do**
  for *s states* **do**
    for *n iterations* **do**
      Calculate Partial Derivative;
      if *converged* **then**
        break;
      else
        Halve perturbation size;
      end
    end
  end
end

Algorithm 1: Numerical Linearisation

4.3 Longitudinal Flight

4.3.1 Linearised State Matrices

The longitudinal dynamics can be described by the forward and vertical velocities (u and w),the pitch rate (q) and the pitch angle(θ). Therefore the state vector is

$$\boldsymbol{X}_{long} = \begin{bmatrix} u & w & q & \theta \end{bmatrix}^T \tag{4.12}$$

The longitudinal motion can be can controlled through the use of the propellers for thrust, thrust vectoring as well as the elevator.

The linearised state equations are then be given by

$$\Delta \dot{\boldsymbol{X}}_{\boldsymbol{long}} = \boldsymbol{A}_{long}\Delta \boldsymbol{X}_{long} + \boldsymbol{B}_{long}\Delta \boldsymbol{U} \tag{4.13}$$

As discussed in [33] , the simplified A-matrix is:

$$\boldsymbol{A}_{long} = \begin{bmatrix} x_u & 0 & 0 & 0 \\ 0 & z_w & z_q & 0 \\ 0 & m_w & m_q & m_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.14}$$

whereas the input B matrix consists of terms which account for the effect of the thrust force, deflection angle and elevator as shown in:

$$\boldsymbol{B}_{long} = \begin{bmatrix} x_{\delta t} & x_{\delta e} \\ z_{\delta t} & z_{\delta e} \\ m_{\delta t} & m_{\delta e} \\ 0 & 0 \end{bmatrix} \tag{4.15}$$

A deflection angle of zero degrees was chosen for the linearisation. The eigenvalues of the A-matrix are the poles of the longitudinal dynamics. The eigenvalues results in four stable poles - two on real axis, as well as two oscillatory poles. The longitudinal eigenvalues and eigenvectors for steady level flight at hovering ($u_o = 0.01m/s$) are:

	Poles		
	-0.0002	-0.0005	-0.0012±0.8276i
u	$1\angle 180$	$0\angle 180$	$0.3048\angle \mp 90.35$
w	0	$1.0000\angle 180$	$0.0353\angle \mp 95.84$
q	0	0	$0.8230\angle \mp 89.65$
θ	$0.0000\angle 180$	$0.0012\angle 180$	$0.9955\angle \mp 179.74$

For hovering, the eigenvalues and eigenvectors show similar behaviour to that of the YEZ-2A airship [11]. The first eigenvalue can be described as the "surge" mode, since it describes the behaviour of the axial velocity (u). The second eigenvalue has one dominant eigenvector value - that of vertical velocity (w). A longitudinal disturbance will cause a vertical displacement and therefore this mode can be called the "heave" mode. The pair of complex eigenvalues describes an oscillatory motion. The phase shift between the eigenvector values for pitch rate (q) and pitch angle (θ) is close to 90^o, and indicates that this mode describes a pendulum type oscillation for the pitch angle.

The longitudinal eigenvalues and eigenvectors for steady level flight at $u_o = 1m/s$ are:

	Poles		
	-0.0153	-0.0516	-0.1183±0.7986i
u	1	$0.0033\angle 180^o$	$0.2556\angle \pm 110.54^o$
w	$0.003\angle 180^o$	$1\angle 180^o$	$0.3261\angle \mp 160.34^o$
q	$0.0000\angle 180^o$	0.0063	$0.6883\angle \mp 70.54^o$
θ	0.0015	$0.1217\angle 180^o$	$0.8525\angle \mp 168.9^o$

At moderate speeds, the eigenvalues and eigenvectors exhibit similar behaviour to those in [11]. The first eigenvalue still describes the "surge" mode, but has a much faster settling time. The second eigenvalue still has a large eigenvector value corresponding to the vertical velocity (w), but the value corresponding to that of the pitch angle (θ) becomes more prominent than at hovering. The pair of complex eigenvalues still describe the pendulum type oscillation for the pitch angle, but the oscillation about the vertical velocity becomes more prominent at this flight speed.

The longitudinal eigenvalues and eigenvectors for steady level flight at $u_o = 4m/s$:

	Poles		
	-0.0613	-0.9064	-0.123±0.3650i
u	$1\angle 180^o$	$0.1886\angle 180$	$0.0711\angle \mp 27.1^o$
w	0.0035	$1\angle 180$	$0.7568\angle \pm 65.8^o$
q	0.0003	0.4748	$0.1844\angle \mp 143.86^o$
θ	$0.005\angle 180^o$	$0.523\angle 180$	$0.4788\angle \pm 35.24^o$

The eigenvalues and eigenvectors exhibit similar behaviour to those in [11] at a high flight speed. The first eigenvalue still describes the "surge" mode, but still with a much faster settling time. The second eigenvalue still describes the "pitch subsidence". Although the effect of the vertical velocity is still dominant, the effect of the pitch angle has become even more prominent. The oscillatory eigenvalues still refer to the pendulum type oscillation about the pitch angle. The oscillation about the heave velocity becomes more prominent as the flight speed increases.

The longitudinal eigenvalue and eigenvectors corresponds well to the behaviour and modes discussed in [11], which also describes ways of demonstrating the modes. Simple approximations from the A-matrix can also be used to approximate the poles and are discussed in [33] and [12]. The longitudinal modes will be discussed in the following subsections.

4.3.2 Surge Mode

The surge mode describes the lag in the response of the axial speed and can be approximated by

$$s_{surge} = x_u \tag{4.16}$$

which yields a stable real pole. Since this pole is very close to the origin, this mode has a very long time constant which can be calculated by

$$T_{surge} = \frac{-1}{s_{surge}} \tag{4.17}$$

As indicated Figure 4.1, the value of x_u decrease as the forward speed increase. The time constant T_{surge} also decreases which means the airship's axial speed tends to settle much quicker at higher speeds.

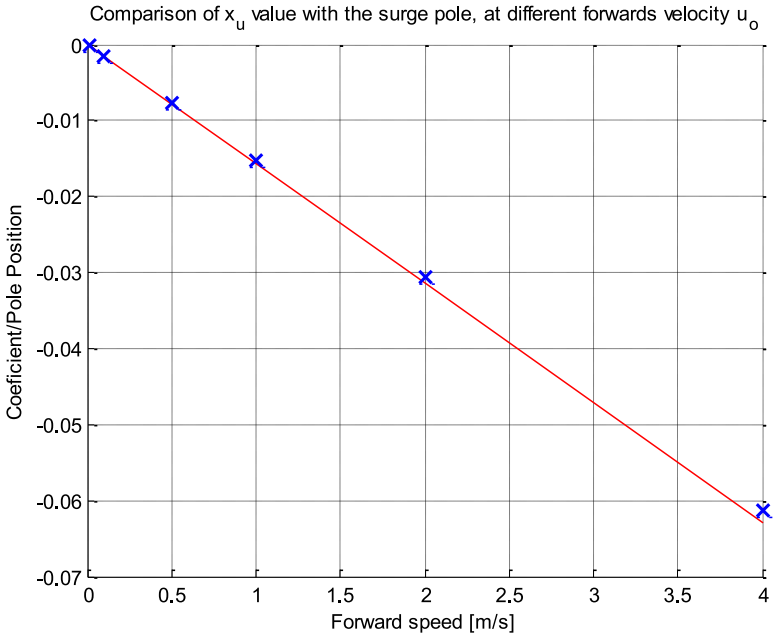


Figure 4.1: Comparison of x_u with the linearised poles

Effect of changing parameters

The drag force has the largest contribution to this mode. The drag force coefficient can be calculated by equating the drag and thrust forces. This would yield

$$C_D = \frac{2F_T}{\rho V^2 S} \tag{4.18}$$

and by substituting an appropriate thrust force and speed reached. Doing this for a thrust force of $12N$ and an axial velocity of $4m/s$, a drag coefficient in the order of $C_D = 0.2$ is calculated. This is much higher than the drag coefficient of $C_D = 0.04$ originally modeled. The higher drag coefficient decreases the surge time constant from $T_{surge} = 63.7s$ to $T_{surge} = 13s$ at 1m/s flight. This indicates that a larger drag coefficient, cause the airships forward speed to settle much quicker. Altering this coefficient has a negligible effect on the other modes.

The large increase in drag can attributed to the following reasons:

- The aerodynamic influence of the gondola - although the assumption can be made that most of the drag are due to aerodynamics of the hull, the gondola will still have a large influence which is unaccounted for.
- The non-smooth surface of the hull - the airship has a lot of non-smooth lassos for attaching the fins and for the mooring of the airship.

The drag also has a direct effect on the deceleration of the airship. This fact can be used for additional drag tests by measuring the deceleration of the airship.

Comparison to non-linear model

This mode can be demonstrated by applying a thrust increase during straight and level flight [11]. The time constant can be verified by measuring the time before the airship axial velocity reaches 63% of its final value, as illustrated in Figure 4.2. For the drag coefficient of $C_D = 0.2$ at 1-m/s flight, the calculated time constant amounts to 13 seconds, whereas the non-linear model results returns 12.1 seconds. This illustrates that this mode describes the non-linear behaviour very well.

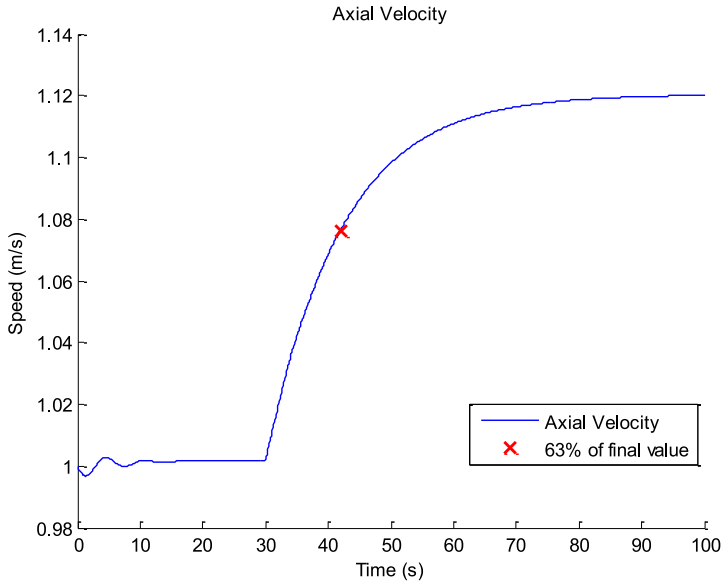


Figure 4.2: Illustration of the surge mode

4.3.3 Heave, Pitch Subsidence Mode

This mode starts out as the heave mode at hovering or low speeds which then converts to a pitch subsidence at higher speeds.

Approximation

The approximation for the heave mode pole is defined as

$$s_{heave} = z_w \tag{4.19}$$

and the approximation for the pitch subsidence mode pole at higher speeds is

$$s_{pitchsub} = m_q \tag{4.20}$$

In Figure 4.3 it can be seen that the pole follows the value of z_w and then diverges to the value of m_q and even beyond m_q indicating that there might be additional stability derivatives

influencing this mode at higher speeds. The time constants for these modes are $T_h = -1/s_{heave}$ and $T_{pitchsub} = -1/s_{pitchsub}$ respectively.

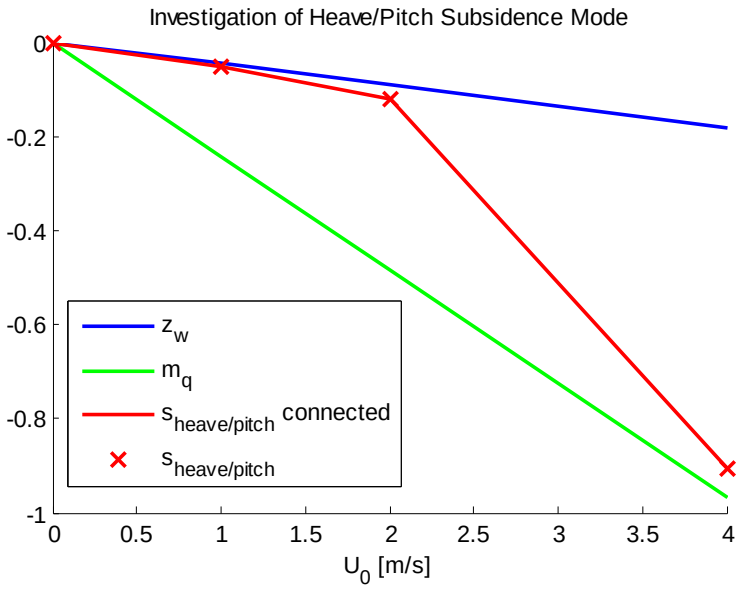


Figure 4.3: Comparison of the Pitch Subsidence/Heave Modes with the corresponding stability derivatives

Effect of changing parameters on the Pitch Subsidence

The influence of changing parameters on the heave/pitch subsidence pole was investigated at a flight speed of $u_o = 1m/s$. By increasing the normal fin forces by 20%, the pole changes from $s_{pitchsub} = -0.0516$ to $s_{pitchsub} = -0.0612$. This means that the time constant $T_{pitchsub}$ decreases as the normal forces on the fins increase. This implies that the higher damping provided by these fins will cause the heave/pitch subsidence mode to settle faster.

Comparison to non-linear model

The heave mode is quite dominant at low speeds. Therefore it is investigated at $u_o = 0.01m/s$ and $u_o = 0.1m/s$ with a constant elevator deflection of 20^o to perturb the trim hover state.

The response of the vertical velocity for different speeds is shown in Figure 4.4. At a flight speed of 0.01m/s, the time constant estimated as $\tau_{heave} = 2000$ seconds. The non-linear model yields a similiar time constant of close to 2000 seconds.

The pitch subsidence mode was illustrated by flying the airship at a constant reference velocity of 4m/s and then applying an elevator deflection of 10 upwards for 20 seconds. The pitch subsidence time constant is calculated as $T_{pitchsub} = 1.1$ seconds. The response of the pitch angle is shown in Figure 4.5 and 63% of the final steady state value is reached in approximately 2.2 seconds.

As the eigenvectors indicate, the vertical velocity's behaviour is still dominant at this flight speed. The time to reach 63% of the steady state value is approximately 1.2 seconds, which is closer

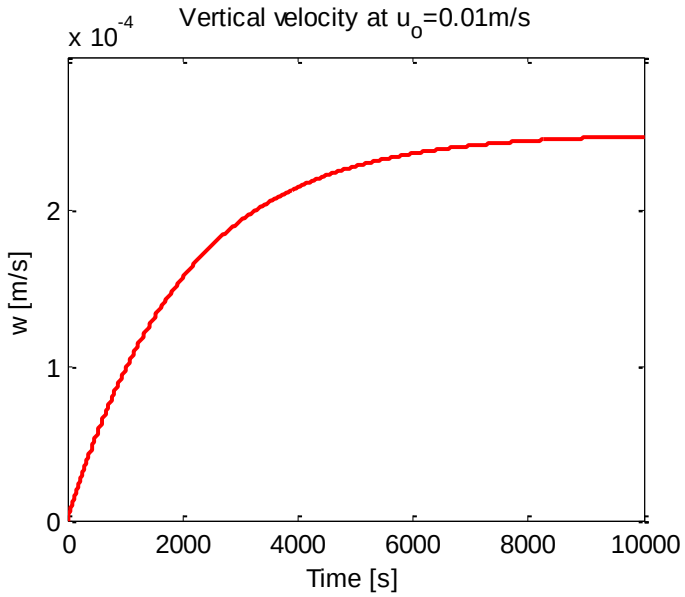


Figure 4.4: The heave mode

to the prediction. Figure 4.5 also shows slight oscillatory behaviour about the pitch angle, which will be discussed in more detail in the next section.

4.3.4 Pitch Oscillation Mode

Approximation

The airship experiences a stable pitch oscillation mode, also known as the longitudinal pendulum. This mode can be approximated at low speeds by the characteristic equation

$$s^2 + m_q s + m_\theta \tag{4.21}$$

and at higher speeds by

$$s^2 + z_w s + \frac{m_\theta z_w}{m_q} \tag{4.22}$$

The roots of these equations give 2 complex poles, indicating an oscillatory motion. The period of these oscillations can be calculated by using the natural frequency w_n in

$$T_{pendulum} = \frac{2\pi}{w_n} \tag{4.23}$$

The following table compares the approximations with the poles from the **A** matrix.

u_o	Low Speed Approx	High Speed Approx	Poles	w_n, rad/s	ζ	$T_{pendulum}$
0.01	-0.0012±0.8267j	-0.0002±0.3592j	-0.0012±0.8267j	0.8267	0.0015	7.6s
1	-0.121±0.8178j	-0.0228±0.35855j	-0.1183±0.7986j	0.8073	0.1465	7.78s
4	-0.4840±0.6702j	-0.0914±0.3474j	-0.123±0.3650j	0.3852	0.3193	16.3s

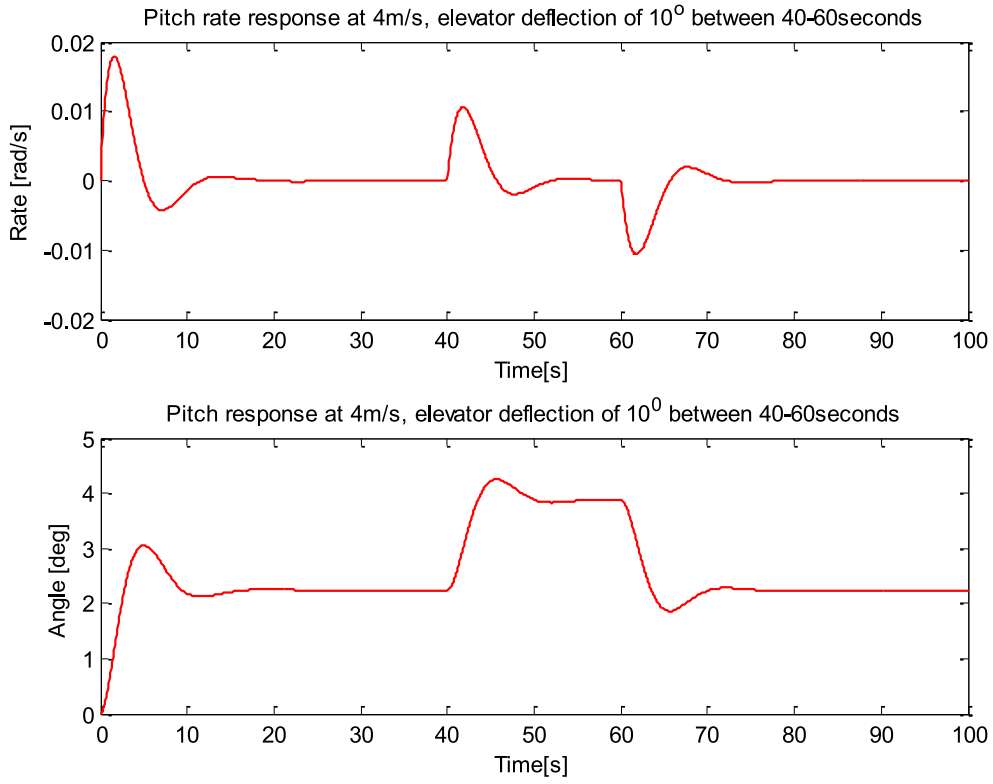


Figure 4.5: Pitch rate and pitch angle response at $u_o = 4m/s$

It can be seen that the poles follow the low speed approximation at lower speeds and then starts to deviate towards the higher speed approximation. As the speed increase, the damping of this mode also increase, while the natural frequency of the oscillation decrease. This corresponds to the behaviour discussed in [11].

Effect of changing parameters

The normal forces on the fins as described in Section 3.9.3, act as a damping force since it provides resistance against the movement of the fins. When this force is decreased, the oscillation becomes more prominent and can even become unstable. When the fin forces' effect is decreased by 20% at a flight speed of $u_o = 1$m/s, the damping of the pendulum mode decreases to $\zeta = 0.1184$ and when increased by 20%, the damping increases to $\zeta = 0.180$.

The effect of the inertia around the y-axis also has an effect on the pitch pendulum mode. A 20% increase in inertia would result in a slight decrease in the damping to $\zeta = 0.138$ as well as a decrease the natural frequency to $w_n = 0.7648$ rad/s.

Comparison to non-linear model

This mode is illustrated by performing an elevator doublet of 15^o at a 20 second interval, with a reference forward speed of 0.01m/s and is repeated at 4m/s. Figure 4.6 shows the pitch response at $u_o = 0.01m/s$. It can be seen that oscillation has a period of 7.6 seconds and almost no damping,

which compares well to the oscillatory behaviour predicted by the complex poles at the same flight speed.

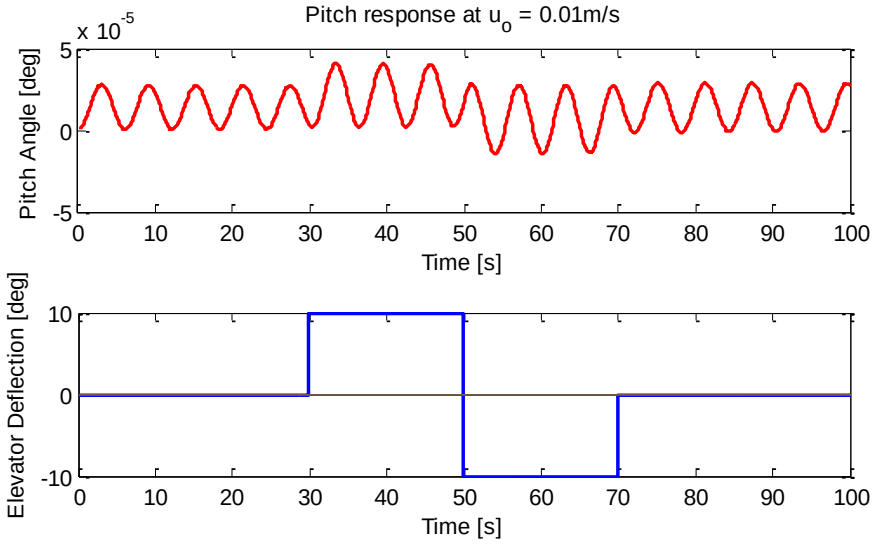


Figure 4.6: Illustration of the longitudinal pendulum mode at hover flight speed of $u_o = 0.01m/s$

Figure 4.7 shows the pitch response at $u_o = 4m/s$. It is clear that that pendulum motion's frequency decreases and that the damping increases at higher flight speeds, although the non-linear model show some slight deviations. This could possibliy attributed to the oscillation about the heave becoming more prominent at this flight speed as indicated by the longitudinal eigenvectors.

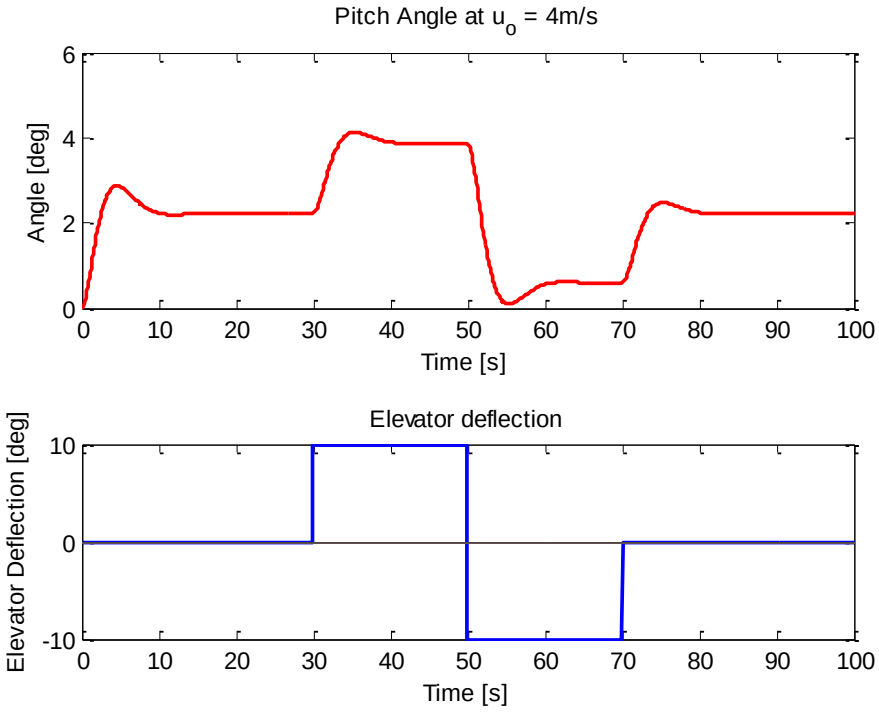


Figure 4.7: Illustration of the longitudinal pendulum mode at hover flight speed of $u_o = 4m/s$

4.4 Lateral Flight

4.4.1 Linearised State Matrices

The lateral dynamics can be described by the side velocity (v), the roll and yaw rates (p and r respectively) and the roll angle(ϕ). Therefore state vector is

$$\boldsymbol{X} = \begin{bmatrix} v & p & r & \phi \end{bmatrix}^T \tag{4.24}$$

The lateral motion can be controlled by use of the rudder or differential thrust. Since differential thrust was not implemented, only the effects of the rudder on the lateral dynamics were investigated.

The linearised state equations are then be given by

$$\Delta \dot{\boldsymbol{X}}_{\boldsymbol{lat}} = \boldsymbol{A}_{lat} \Delta \boldsymbol{X}_{lat} + \boldsymbol{B}_{lat} \Delta \boldsymbol{U}_{\boldsymbol{lat}} \tag{4.25}$$

As discussed in [33], the simplified A-matrix for lateral dynamics is:

$$\boldsymbol{A}_{lat} = \begin{bmatrix} y_v & 0 & y_r & y_\phi \\ l_v & l_p & l_r & l_\phi \\ n_v & 0 & n_r & n_\phi \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.26}$$

whereas the B-matrix for the rudder as the only lateral input, is:

$$\boldsymbol{B}_{lat} = \begin{bmatrix} y_{\delta_r} \\ 0 \\ n_{\delta_r} \\ 0 \end{bmatrix} \tag{4.27}$$

The eigenvalues of the A-matrix are the poles of the lateral dynamics. As discussed in [11] and [33] the eigenvalues results in four stable poles - two on real axis, as well as two oscillatory poles. The lateral eigenvalues and eigenvectors for steady level flight at hovering ($u_o = 0.01 m/s$):

	Poles		
	-0.0001	-0.0032	-0.0007±2.3363
v	1	1	$0.2106\angle\pm 89.98^o$
p	0	$0.0000\angle 180^o$	$0.9999\angle\pm 90^o$
r	$0.1860\angle 180^o$	0.6489	$0.0081\angle\mp 90.27^o$
ϕ	$0.0002\angle 180^o$	0.0003	$0.4280\angle\mp 0.02^o$

At the hovering flight speed, the eigenvalues and eigenvector conditions corresponds well to the results of [11]. The first eigenvalue describes the "sideslip subsidence" mode, and shows that this mode has a large influence on the side velocity (v). The second eigenvalue also describes the changes about the side velocity, but the effect on the yaw rate (r) is much more prominent than

in the "sideslip subsidence" mode. Therefore this mode is called the "yaw subsidence" mode. The pair of complex eigenvalues describe an oscillatory motion. The phase shift between the eigenvector values for roll rate (p) and roll angle (ϕ) is close to 90^o. This shows that this mode describes a pendulum type oscillation for the roll angle.

The lateral eigenvalues and eigenvectors for steady level flight at $u_o = 1m/s$ are:

	Poles		
	-0.0124	-0.3197	-0.0737±2.3359
v	$1\angle 180^o$	1	$0.2091\angle \pm 178^o$
p	$0\angle 180^o$	$0.0104\angle 180^o$	$0.9949\angle \pm 179.7^o$
r	0.1828	0.6430	$0.0089\angle \mp 23.44^o$
ϕ	0.02	0.0326	$0.4257\angle \pm$-88^o

At moderate speeds, the eigenvalues and eigenvectors show similar behaviour as those in [11]. The first eigenvalue, corresponding to the "sideslip subsidence mode", still describes the behaviour of the side velocity. The second eigenvalue shows dominant behaviour for the side velocity and yaw rate. The complex eigenvalues still describe the oscillatory pendulum roll mode, as the eigenvector values for the roll rate and roll angle are out of phase by close to 90^o.

The lateral eigenvalues and eigenvectors for steady level flight at $u_o = 4m/s$ are:

	Poles		
	-0.0492	-1.2750	-0.2967±2.3290
v	$1\angle 180^o$	$1\angle 180^o$	$0.2004\angle \mp 8.53^o$
p	$0\angle 180^o$	0.1467	$0.9806\angle \mp 1.14^o$
r	0.1827	$0.6547\angle 180^o$	$0.0161\angle \pm 126.62^o$
ϕ	0.08	$0.1151\ \angle 180^o$	$0.4177\angle \mp$-98.4^o

At higher speeds, the eigenvalues and eigenvectors show similar behaviour to those in [11]. The first eigenvalue, corresponding to the "sideslip subsidence" mode, still describes the behaviour of the side velocity. The second eigenvalue still shows dominant behaviour for the side velocity and yaw rate, although the slight increase in effect of the roll angle could indicate the airship banking during the turn. The complex eigenvalues still describe the oscillatory pendulum roll mode.

These results show that the lateral modes show the same behaviour as those in [11]. The modes are discussed in more detail in the following sections.

4.4.2 Sideslip Subsidence

Approximation

The sideslip subsidence describes the exponential subsidence in lateral velocity and is usually the smaller of the two real lateral poles. The sideslip subsidence pole can be approximated by

$$s_{sideslip} = (y_v + l_v y_\phi / l_\phi) \tag{4.28}$$

The approximation's results do not compare well with the poles, as shown in the following table:

u_0	Approximation	Pole
0.01	-0.0049	-0.0012
1	-0.00489	-0.0124
4	-0.1954	-0.0492

The fact that the approximation and actual pole deviate at a constant ratio could indicate that other stability derivatives are influencing this mode for this model. The time constant can be calculated as $T_{sideslip} = -1/s_{sideslip}$.

Effect of changing parameters

The normal fin forces also plays an important roll in the sideslip subsidence mode. The pole changes from $s_{sideslip} = -0.0124$ to $s_{sideslip} = -0.0497$ with an increase of 20% in the normal fin force. The time constant of this mode decreases considerably, which also confirms that higher normal fin forces improve the settling time of the sideslip angle.

Comparison with non-linear model

This mode can be demonstrated at a flight speed of $u_o = 1m/s$ by applying a constant rudder deflection of 10^0. The pole predicts a time constant of $T_{sideslip} = 80.6$ seconds. As shown in Figure 4.8, the sideslip angle reaches approximately 63% of its steady state value in 94.6 seconds. A possible reason for the increase in time could be due to the fact that the axial velocity decreases slightly as the airship starts to turn. Except for this deviation, this pole accurately describes the behaviour of the airship.

4.4.3 Yaw Subsidence

Approximation

The yaw subsidence pole can be approximated by

$$s_{yaw} = n_r + y_v \tag{4.29}$$

where the yaw damping derivative n_r provides the largest contribution as shown in Figure 4.9. The approximation is accurate at low speeds, but slowly diverges from the actual pole, indicating that there might be additional derivatives which have an influence on this pole in the Yuwen Li model, especially at higher speeds. The yaw subsidence increases with speed, which enables better control at higher speeds since the yaw time constant is much faster and the pole moves further from the neutrally stable position at the origin.

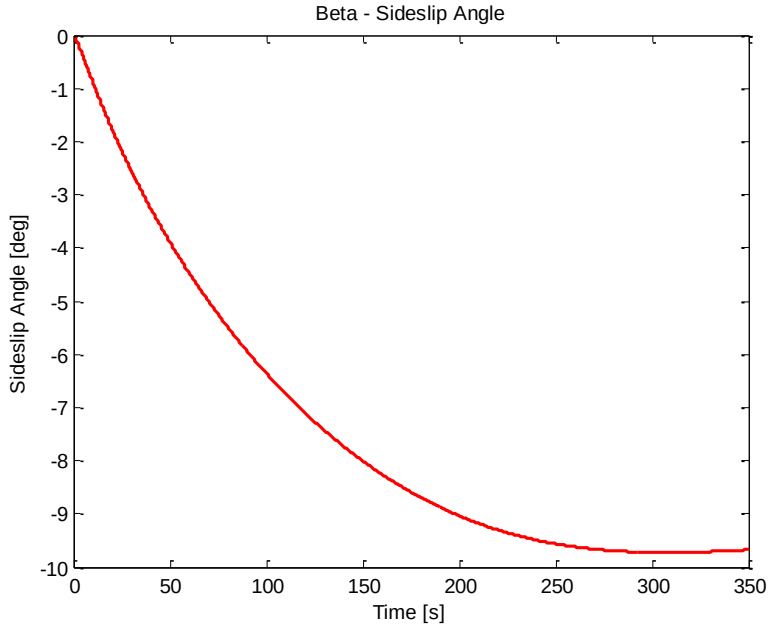


Figure 4.8: Sideslip angle at a flight speed of 1m/s

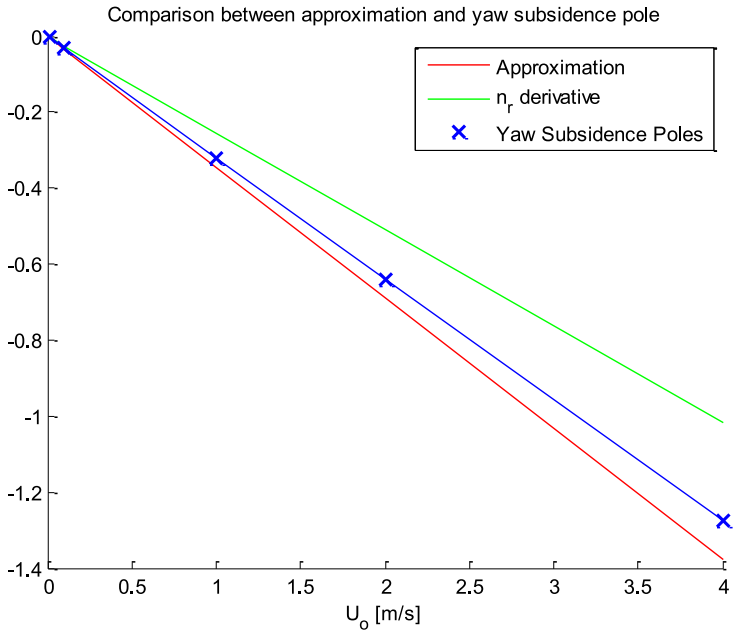


Figure 4.9: Comparison of the yaw subisidence pole with the approximation

Effect of parameters

The effects of certain parameters which would effect the yaw subsidence are investigated. The effect of increasing the inertia around the z-axis by 20% had a small influence on this pole, moving it from $s_{yaw} = -0.3197$ to $s_{yaw} = -0.2963$ at 1m/s. This is understandable, since a larger inertia would mean the airship would take longer to settle at its final value and therefore have a larger time constant.

The fin forces also play an important role in the yaw subsidence mode. An increase in the fin forces would provide better damping of this mode and therefore increase the controllability. By increasing the normal fin forces by 20%, the pole changes from $s_{yaw} = -0.3197$ to $s_{yaw} = -0.3410$, which means a smaller time constant for the yaw subsidence mode. This mode settles much faster, since the fins provide more resistance to the yawing motion of the airship.

Comparison with non-linear model

This mode can be be demonstrated at the flight condition by applying a constant rudder deflection of 20^o. At a flight speed $u_o = 1m/s$, the time constant is $T_{yaw} = 3.13$seconds. This pole does not seem to describe the non-linear behaviour as the response is much slower with a time constant of $T_{yaw} = 50$seconds, as shown in Figure 4.10. The non-linear response is much closer to that of the the pole describing the sideslip angle as shown in Figure 4.10.

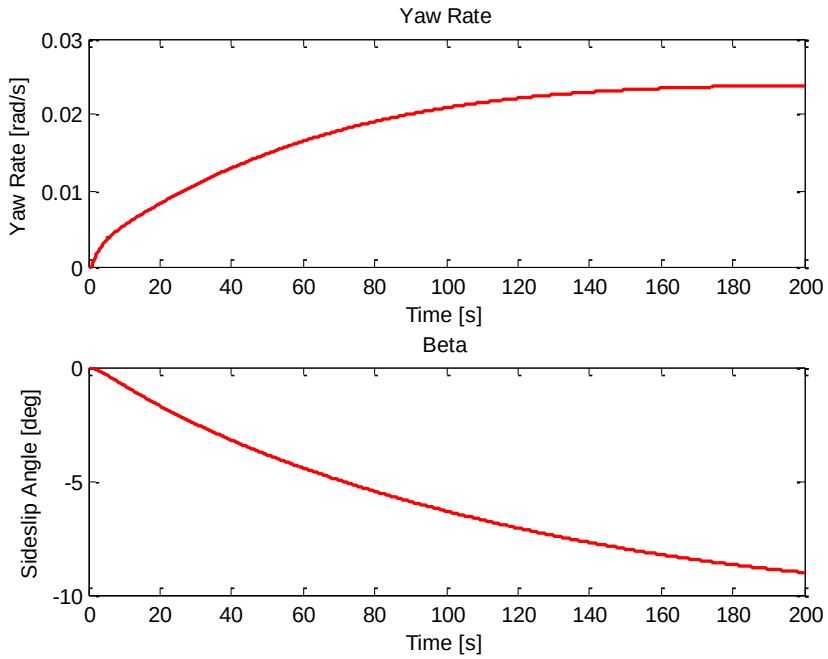


Figure 4.10: Yaw rate response

It is interesting to note that when the normal fin forces effect is increased by 30% the pole shifts to $s_{yaw} = 0.3520$. This means that the time constant is $T_{yaw} = 2.8$ seconds. Although the time constant does not change much, it corresponds much better to the behaviour of the non-linear model, which now yields a time constant of 4.2 seconds. This is shown in Figure 4.11. The corresponding eigenvector also indicates that this pole should describe the yaw rate much better than before, as shown in the following table:

	-0.3520
v	1
p	$0\angle 180^o$
r	0.81843
ϕ	0.0465

4.4.4 Roll Oscillation

Approximation

The roll oscillation approximation is described as the pair of complex roots [12] of

$$s^2 - (l_p + yv - \frac{l_v y_\phi}{l\phi})s - l_\phi \tag{4.30}$$

The approximation and the actual poles of the linearised **A**-matrix are shown in the following table:

u_o	Approximation	Poles	w_n, rad/s	ζ	$T_{pendulum}$
0.01	-0.0011±2.3363j	-0.0007±2.3363j	2.33	0.00003	2.69s
1	-0.1078±2.334j	-0.0737±2.3359j	2.33	0.03	2.69s
4	-0.431±2.2962j	-0.2967±2.329j	2.36	0.126	2.67s

The approximated oscillatory poles are better damped than the actual poles at higher speeds, but still have the same natural frequency. Figure 4.12 also illustrates that this mode is better damped at higher velocities with the damping improving from $\zeta = 0.014$ to $\zeta = 0.67$.

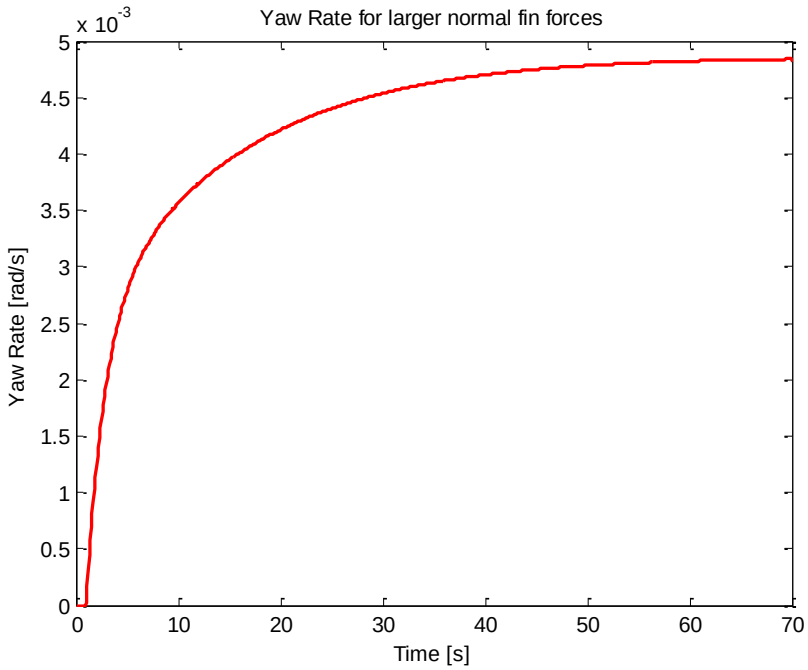
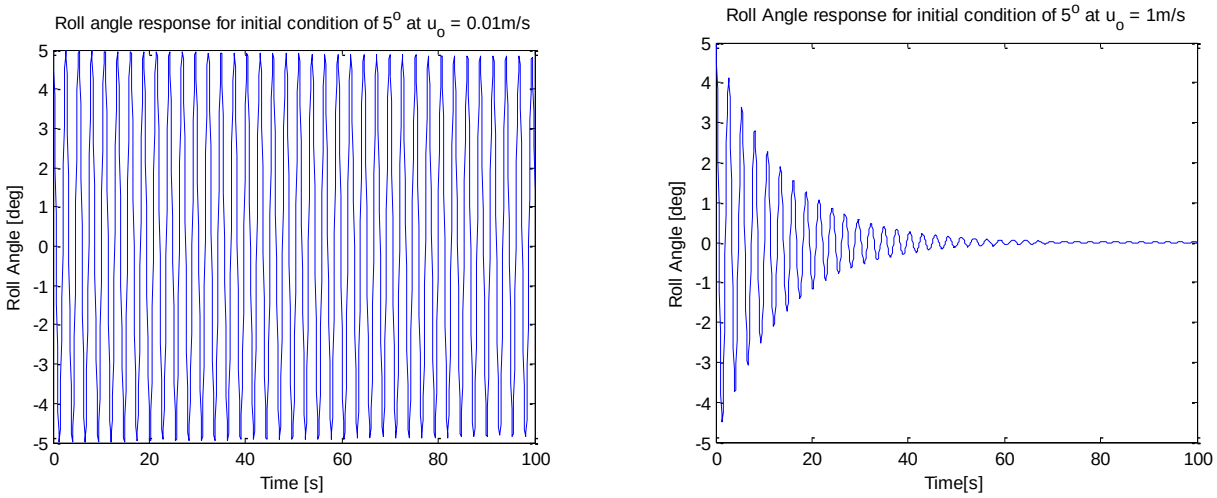


Figure 4.11: Yaw rate reponse with larger fin forces

Comparison with non-linear model

Since no aileron function was implemented for this model, it was decided to test the mode's characteristics by giving the airship an initial roll angle of 5^o. As shown in Figure 4.12 the poles correspond well to the non-linear model, illustrating higher damping at a higher speed and that the natural frequency does not change much at different flight speeds.



(a) 0.01m/s (b) 1m/s

Figure 4.12: Roll Pendulum mode at different flight speeds

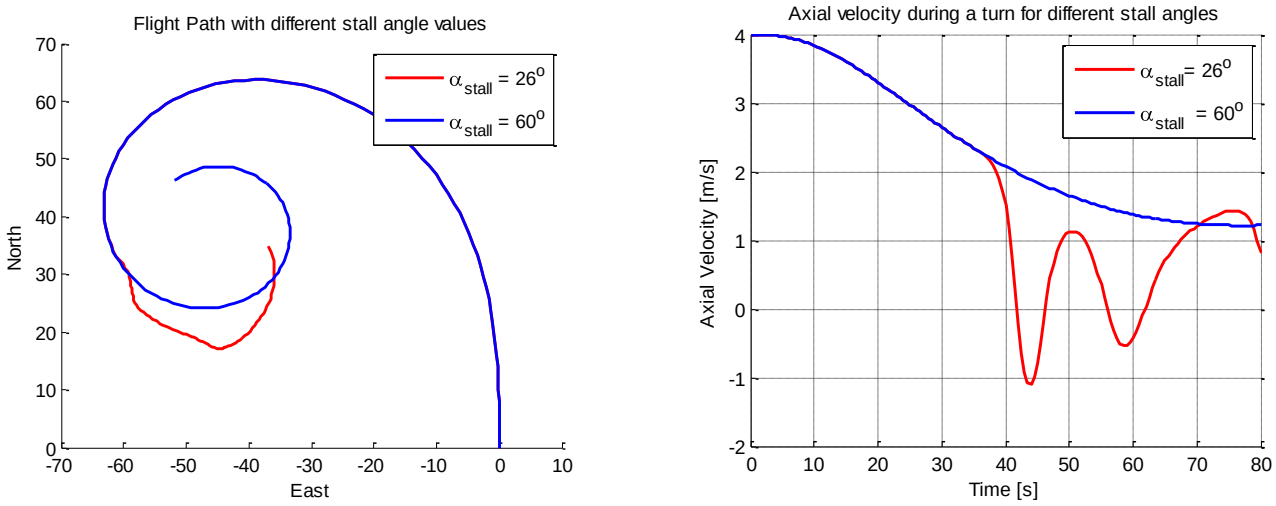
4.4.5 Effect of the stall angle

The effect of the stall angle is not visible in the linearised dynamics because the perturbations around the trim point is very small. The stall angle defined for the Yuwen Li model in Section 3.9.3 plays an important part during a turning maneuver, since it limits the maximum size of the normal forces on the fins that counteracts against the turn. When the normal force starts to counteract the turn, the airship experiences a decrease in axial velocity and an increase in side velocity causing the airship to turn much sharper. When the airship has lost most of its axial velocity, the effect of the rudder decreases and the airships axial velocity starts increasing as it moves forward. The effect of different stall angles at an initial axial velocity of 4m/s and rudder deflection of 15^o are illustrated in Figure 4.13

Although it is difficult determining the precise value of the stall angle, it can be adjusted intuitively until the airship's lateral response delivers the same behavior at similar flight conditions than that of the test flights.

4.5 Lateral Behaviour Adjustment

In this section, the actions taken to provide a better correlation between the model and the actual behaviour are described.



(a) North East Diagram

(b) Axial velocity

Figure 4.13: Comparison of the effect of different stall angles

For a rudder deflection of 16^0 at average flight speed of 3m/s:

Parameter	Model	Typically Measured
Maximum Sideslip Velocity	$0.45m/s$	$2m/s$
Yaw Rate Achieved	$0.1rad/s$	$0.18rad/s$
Typical Yaw Acceleration	$0.005rad/s^2$	$0.035\ rad/s^2$

This Table shows that the model's lateral behaviour is much slower that that of the actual measurements. The normal forces on the fins, as discussed in Section 3.9.3, are the main opposing force during a turn. The size of this force has a direct influence on the yaw rate settling time and the maximum yaw rate , as discussed Sections 4.4.2 and 4.4.3. The normal forces on all the fins were therefore increased by 30% to improve the settling time of the yaw rate.

The force and moment due to rudder deflection initiates the airships turn and creates the initial yaw rate. The effect of these forces were adjusted using the summation of the lateral forces and moments discussed in Section 3.9.7.

Therefore using the lateral force and moment equations

$$\tau_I(2) + \tau_V(2) + \tau_{NF}(2) + \tau_R(2) = 0 \tag{4.31}$$

$$\tau_I(6) + \tau_V(6) + \tau_{NF}(6) + \tau_R(6) = 0 \tag{4.32}$$

correction factors for the forces and moments due to rudder deflection can be calculated. This is done by solving Equations 4.31 and 4.32 for a specific flight condition. The flight parameters that should be specified are the axial velocity u, side velocity v, rudder deflection δ_r and yaw rate r.

After the correction factors are calculated, the stall angle should be adjusted. The original stall angle is to small, hence the sharp turns at relatively small deflection angles and therefore the stall angle needs to be increased to approximately 60^o to cause stalling at larger angles.

4.6 Conclusion

In this chapter the Yuwen Li model is linearised and the effect of certain parameters of the model on the linear modes and poles are analysed. The Yuwen Li models modes are investigated in the same manner as that of Gomes YEZ-2A airship in [11] and compares well to expected modes. All the poles were stable for all the speeds they were analysed .

The lateral model was adjusted to compensate for differences between the model and actual airship lateral behaviour, to provide a reasonable model for controller testing. These differences indicate modelling errors, especially on the normal fin forces as well as the contributions of the fin deflection forces.

Due to large differences between the lateral airship model compared to the measured behaviour, a less model intensive controller design method is followed and discussed in the next chapter.

Chapter 5

Control system design

5.1 Introduction

This chapter discusses the design of the automatic flight control system. A controller that mimics a human pilot's actions and reasoning should be adequate as the first iteration of controller design, as the airship is slow moving and does not require quick responses for stable flight or to stay afloat. A fuzzy logic controller design approach was undertaken due to the following reasons:

- Fuzzy logic controllers show tolerance to model inaccuracies as well as measurement noise.
- The complexity of the design is simple.
- The controllers can be adjusted intuitively from the controller results. This opens the door for future adaptive fuzzy logic controllers.
- Fuzzy logic controllers can be combined with conventional control techniques.

The fuzzy logic controllers are non-linear controllers designed for the non-linear model. A framework for the control of an unmanned aerial vehicle (UAV) using fuzzy logic is discussed in [40] as well as [41]. The simulation results of these controllers show good controller performance as well as the potential of fuzzy logic controllers.

5.2 Introduction to Fuzzy Logic

An overview of fuzzy logic systems can be found in [42] and a good example of designing a fuzzy logic controller is shown in [43]. As shown in Figure 5.1, fuzzy logic controllers are rule based controllers which borrows a lot from standard set-theory and consists of several stages.

The inputs to the controllers usually consists of measurement or state values, whereas outputs can be specified as actuator commands.

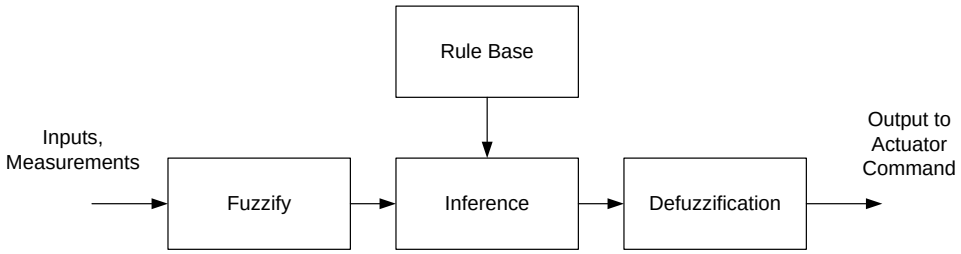


Figure 5.1: Overview of a fuzzy logic control system

5.2.1 Fuzzifying

In the fuzzyifying stage input measurement values are assigned membership values to the fuzzy sets. These fuzzy sets can usually be described by linguistic terms. For example, the error measurement can be SMALL or LARGE, as illustrated in Figure 5.2. The sets usually overlap since certain inputs are "fuzzy" and can be classified into two or more sets.

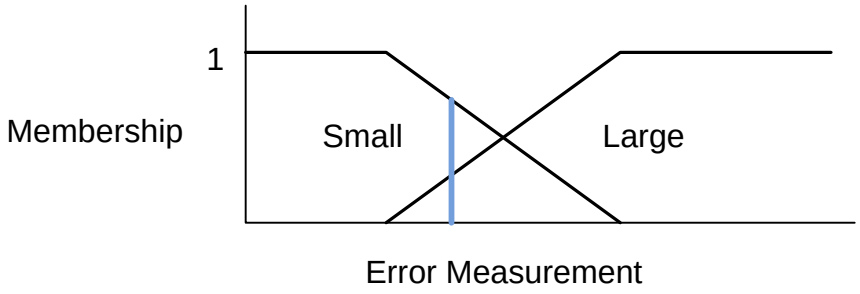


Figure 5.2: Example of fuzzifying a temperature measurement

Membership functions determine the degree to which a value belongs to a certain set. For example, the error can be 70% SMALL, 30% LARGE as illustrated in Figure 5.2. Different membership functions can be chosen to assign the input variables membership to the fuzzy sets, as shown in Figure 5.3.

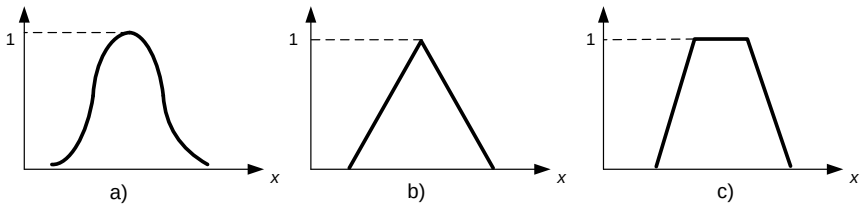


Figure 5.3: Examples of normalised membership functions: a) Gaussian, b) Triangular, c) Trapezoidal

As discussed in [44] the input membership functions also have an effect on the robustness of the controllers, but the choice of input membership functions is mainly determined by the problem at hand. A good initial choice for membership functions is described in [45]:

- Symmetrical triangles of equal width should be used as membership functions with the left most and right usually being shouldered ramps
- The membership function's widths should be chosen in such a way that they preferably overlap by 50%, so that a input variable is a member of at least 2 sets. This will ensure that the inputs may relate to several rules, and will result in a much smoother output.
- All the values in the input range should be assigned to fuzzy variables - this ensures that there are no values which may lead to an undefined control output.

A triangular membership functions was chosen to simplify the design and reduce the computational complexity.

Scaling gains can be used as a tuning parameter for the membership functions. By applying a simple gain to the input or the output variables, as with normal controllers, the performance of the controller can be changed. The effect of these gains is the same as changing the position and width of the original membership functions as discussed in [43] - for example applying a gain G_i to an input variable, is the same as multiplying the intervals of the membership functions with $1/G_i$.

5.2.2 Rule Base

The actions of the controller are specified in this section of the controller. The output of the controller is mapped to the fuzzy input by simple rules. These rules can usually be described by *if-then* conditional statements and can be represented in several ways as shown in Figure 5.4. The rules in this figure gives a LARGE output if the error is LARGE and a SMALL output if the error is SMALL.

if *input* **is** *SMALL* **then** *output* **is** *SMALL*

if *input* **is** *LARGE* **then** *output* **is** *LARGE*

Input	Output
SMALL	SMALL
LARGE	LARGE

Figure 5.4: Examples of Rule Representation: a) Described in *if-then* logic, b) Described in a decision table

The fuzzy logic controllers in this project are kept to simple "two-inputs one-output" systems, so that the rule base can be represented by a 2D table which is easier to visualize during the design phase. The initial rule base is designed from experience of the airships behaviour, controller knowledge as well as from the desired controller actions during certain flight conditions.

5.2.3 Inference Engine

The inference engine determines which rules in the rule base's conditions are met (which rules fired) and to what extent (their firing strength). The firing strength or degree of fulfillment of a rule can

be calculated by simply calculating the minimum of the membership values of the inputs for the specific rule.

The resulting output set of each rule that has fired is combined into one output set, using the union operation and the conclusions of all the rules are thus accumulated, as shown in Figure 5.5. The firing strength is used to scale the contribution from each rule to the resulting output.

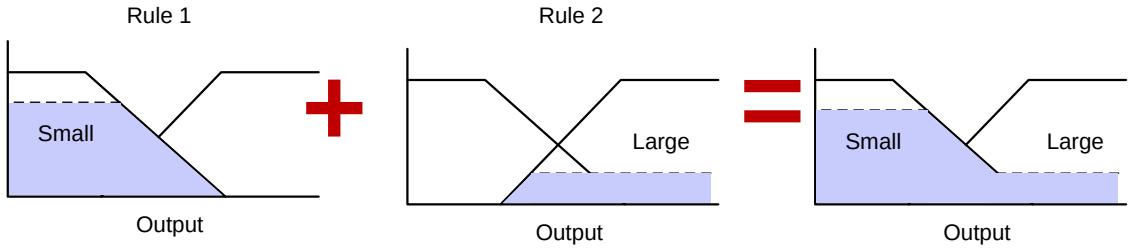


Figure 5.5: Combining the output sets of all the rules

5.2.4 Defuzzifying

In the defuzzifying stage a control signal (u) is calculated from the resulting fuzzy output set. There exist several methods for calculating a crisp value from the output set. The method used in this project is the popular centre of the area (COA) method,

$$u = \frac{\sum \mu(x_i)x_i}{\mu(x_i)} \tag{5.1}$$

which is also shown in Figure 5.6.

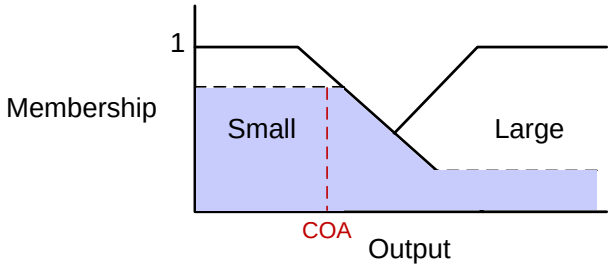


Figure 5.6: Calculating the centre of the output set.

This method is effective as it is relatively simple to calculate and takes into account all the rules that have fired, acting as a weighted average operation.

5.3 Fuzzy Logic Design Process

The fuzzy logic controller design process can be described in the following steps:

1. Decide on the input and output variables - e.g. which measurements are required to decide on a given control output.

2. Decide on the linguistic terms and thus the number of membership functions, to describe the input and output variable states, e.q. for a rudder output deflection - [Large Negative, Small Negative, Zero, Small Positive, Large Positive]. A uneven number of sets is recommended especially if the the range has a center value of zero.

3. Decide on membership function to assign membership to a specific conditions - membership functions are chosen as overlapping triangular functions in this project. The size of the function should be large enough to include measurement noise, as well as uncertainty around a specific operating point.

4. Decide on a rule base - how should the input conditions be matched to specific output ranges, etc. The rules can be decide on from personal experience, or the required performance at certain flight conditions. For ease of design, a simple 2D rule base in the table format is recommended, since it is easier to spot input states for which there is no output defined. The transition between rules should be as smooth as possible to avoid sudden changes in the output command. By plotting a 3D input-output map of the rulebase, it also is easy to see where irregularities occur in the specified rules.

5. Decide on a defuzzification method - a way of calculating the resulting output. The COA method was chosen for all the controllers in this project, as it provides a weight average answer for all the rules that has fired.

The following aspects of a fuzzy logic controller can be used as tuning parameters to change the behaviour of the controller:

1. The membership functions - by changing the membership functions used, the relation of a specific measurement to a fuzzy set can be changed.

2. The width and position of the membership functions - by changing the ranges of the fuzzy sets, a more vague/fuzzy description can be given to noisy measurements, where as accurate measurements can be described with a smaller and more precise range.

3. Scaling gains. Sensitivity to the specific input can be decreased or increased by applying a scaling gain on the input membership functions. By increasing the gain on the output membership functions can increase the control sensitivity.

4. Rule base can be changed to adjust the behaviour of the controller

Different structures for the fuzzy logic controller can also be tried. In [41], the PI-type fuzzy logic controller shown Figure 5.7 is discussed. The main reason for considering this controller structure, is it is easier to adapt the current output with the required change instead of aiming for an exact output. Unfortunately this controller can be very aggressive, especially if the incremental change

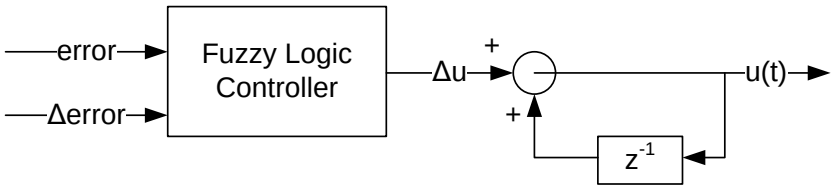


Figure 5.7: Fuzzy Logic PI-type Controller

is large, while if the incremental actuation change is too small, the controller may suffer from large overshoot.

The implementation of the controllers in C-code is discussed in Addendum D. In the following sections, the design and simulation of the longitudinal and lateral controllers are discussed. The rule base in each case consists of seven possible outcomes unless stated otherwise: [NL NM NS Z PS PM PL]. These tokens are abbreviations for [*"Negative Large", "Negative Medium", "Negative Small", "Zero", "Positive Small", "Positive Medium", "Positive Large"*]

5.4 Fuzzy Logic Heading Controller

5.4.1 Block Diagram

The fuzzy logic heading controller is shown in Figure 5.8. It requires the heading error and the yaw rate as an input and outputs a rudder deflection angle command.

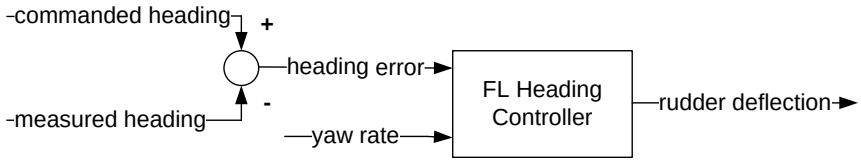


Figure 5.8: Heading Controller Block Diagram

5.4.2 Implementation

The heading error can be calculated by

$$\psi_{error} = \psi_{commanded} - \psi_{actual} \tag{5.2}$$

The input membership functions for the heading error measurement is shown in Figure 5.9. The heading error membership function's centers are initially chosen at the coordinates [-40 -20 -10 0 10 20 40] degrees - where anything above 40^o error is considered a large error, 20^o is considered a medium error and 10^o error is considered a small error.

The input membership functions for the yaw rate measurement is shown in Figure 5.10. The maximum yaw rate achieved during test flights suggests that a yaw rate of 0.2 rad/s would serve

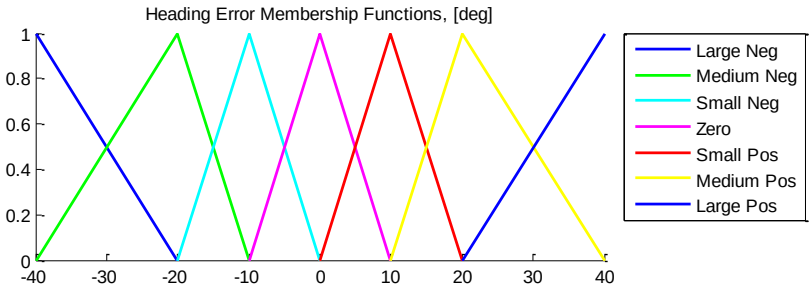


Figure 5.9: Fuzzy heading controllers heading error input

well as a "high" measurement value. Therefore, the yaw rate membership functions centers are chosen at [-0.2 -0.1 0 0.1 0.2] rad/s, to provide variables for large and smaller turning rates in both directions.

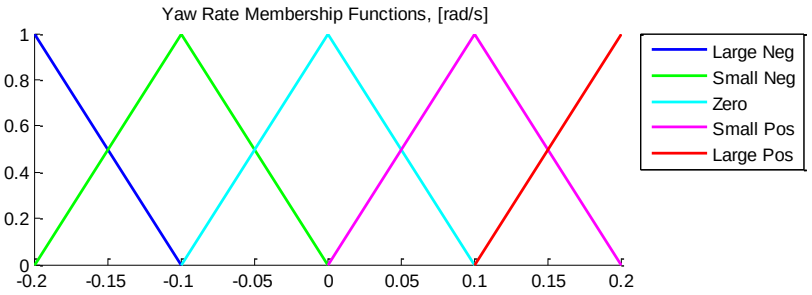


Figure 5.10: Fuzzy heading controllers yaw rate input

The output membership functions are shown in Figure 5.11. The centers for each of the functions were chosen as the rudder outputs [-25, -15, 5, 0, 5, 15, 25], to represent the rudder deflection in degrees for zero, small, medium and large angles in both positive and negative directions.

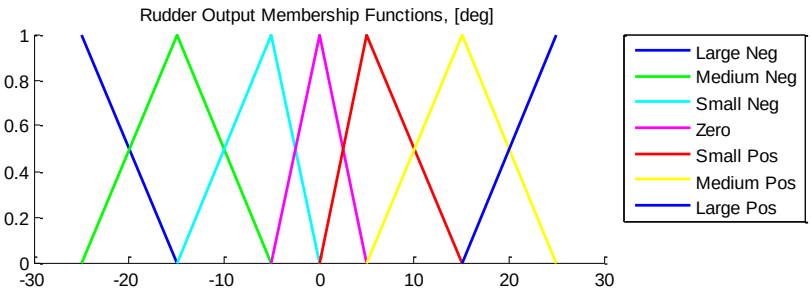


Figure 5.11: Fuzzy heading controllers output membership functions

The rule base was designed with several aspects in mind:

- When the heading error is small, the rudder deflection angle should be small. [red]
- When the heading error is large, the rudder deflection angle should be large. [blue]
- When the heading is deviating from the commanded heading angle, small rudder deflection angles should be used to correct its path. [green]

- When the heading error is very small and the airship is approaching the commanded heading very quickly, the rudder angle should deflect in the opposite direction to decrease possible overshoot. [magenta]

Taking all these aspects in consideration, the rule base can be summarized in the following 2D matrix:

Heading Error	-0.2	-0.1	0	0.1	0.2
-40	PM	PM	PL	PL	PL
-20	PS	PM	PM	PL	PL
-10	NS	Z	PS	PM	PM
0	NS	NS	Z	PS	PS
10	NM	NM	NS	Z	PS
20	NL	NL	NM	NM	NS
40	NL	NL	NL	NM	NM

Yaw Rate

This rule base delivers the symmetrical 3D input-output map shown in Figure 5.12.

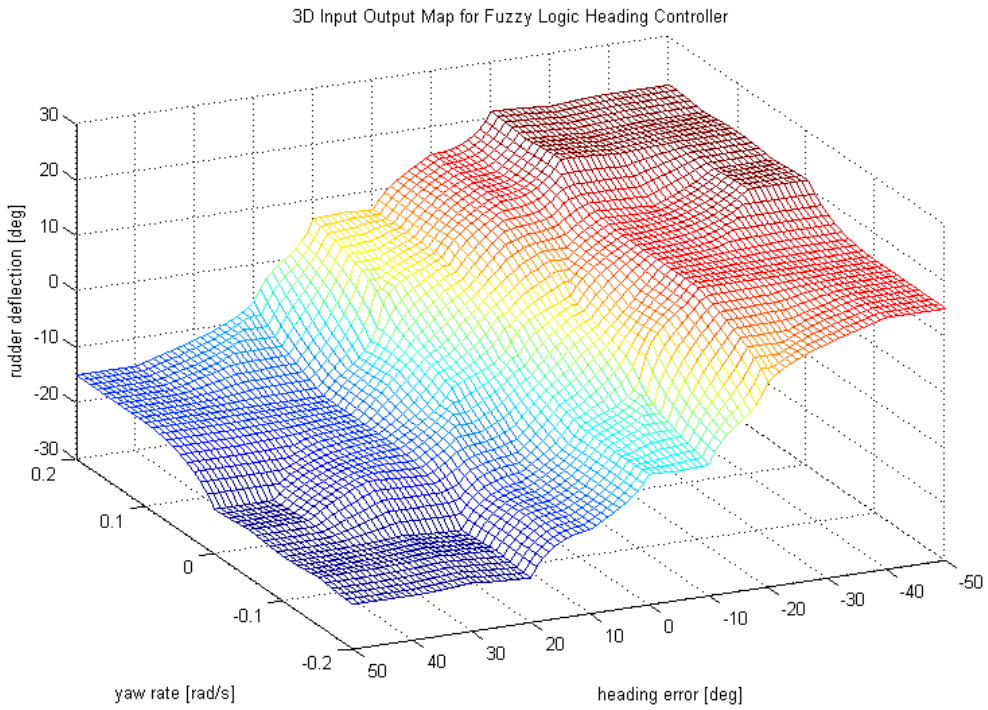


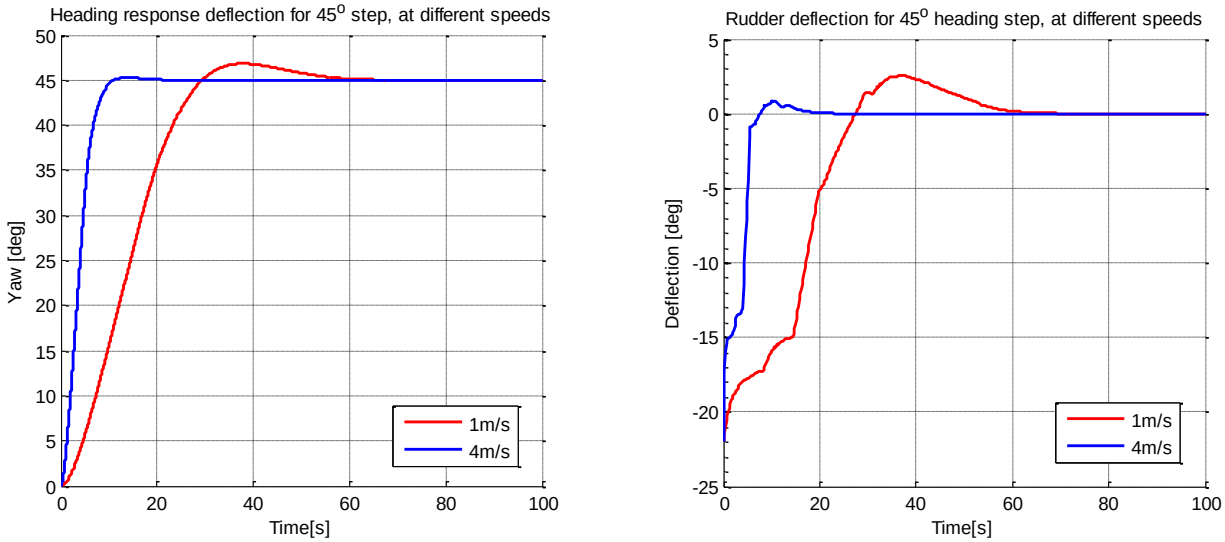
Figure 5.12: 3D Control Surface

5.4.3 Simulation Results

The simulated step response and the rudder deflection commands, are illustrated in Figure 5.13 for 1m/s and 4m/s forward flight. The response characteristics for a 45^o step are given in the following table:

Speed	2% Settling time	Overshoot %
1	49s	3.7
4	9.4s	0.7

It is clear that at the higher speed, the damping of the controller is much better therefore less overshoot is experienced. The heading angle also settles much faster at the reference command.



(a) Heading Response

(b) Actuator Commands

Figure 5.13: Heading controller step response for 45^o step command at different speeds.

The effect of scaling gains can be illustrated on the heading controller. The sensitivity to one of the input measurements can be increased by applying a gain to the specific membership functions. The control effort of the controllers can be adjusted by using a scaling gain - increasing the gain would result in a more aggressive controller using a lot more control effort, whereas reducing the gain will result in a less aggressive controller. This is illustrated in Figure 5.14 for a scaling gain of 0.5 on the output. The overshoot is reduced to 0.4%, while the 2% settling time increased to 15.2s.

5.5 Height Control

5.5.1 Block Diagram

Thrust vectoring is used to control the height due to the inefficiency of the elevator, especially at low speeds. This is illustrated for the YEZ-2A airship in [12] where comparisons are made between the thrust vectoring and the effect of the elevator at different speeds.

The height controller consists of two sections and the block diagram is shown in Figure 5.15. The first section consists of a fuzzy logic controller which uses the height error to decide upon an initial estimate for the vector angle of the thrust. The second part uses a fuzzy logic incremental controller which slowly adjusts the initial estimate based on the height error as well as the change in error.

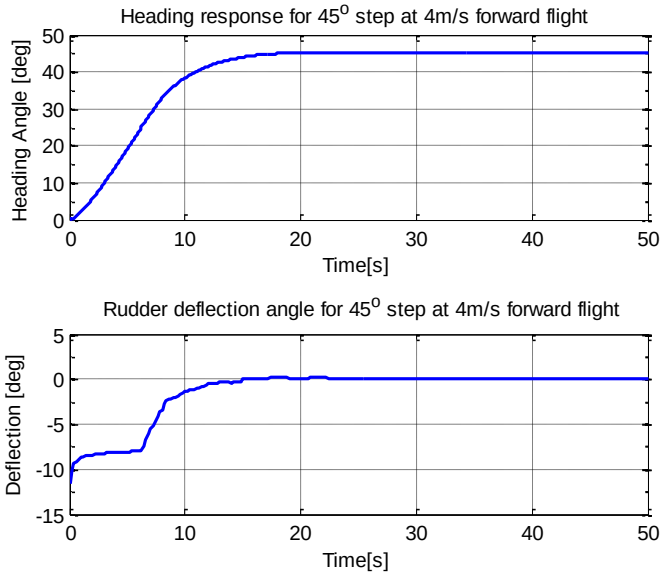


Figure 5.14: The effect of a 0.5 output scaling gain on the controller characteristics

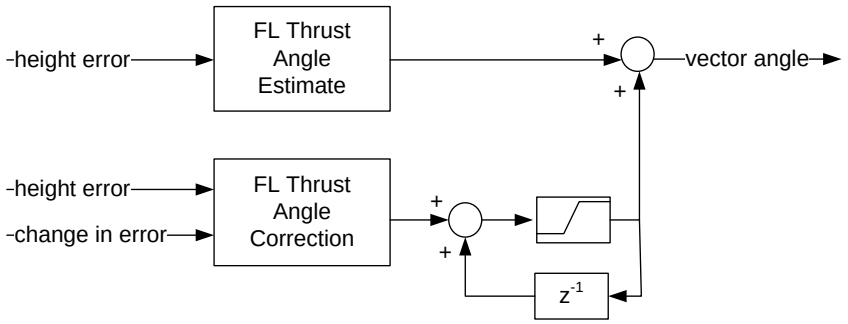


Figure 5.15: Height Controller Block Diagram

This compensates for steady state errors which may arise as well as prevent possible overshoot as far as possible.

5.5.2 Implementation

The height error can be calculated as the difference between the commanded height and the measured height, thus

$$h_{error} = h_{commanded} - h_{measured} \tag{5.3}$$

The height error input as well as vector angle output membership functions are shown in Figure 5.16. The centers for the membership functions are chosen at [-10, -5, -1, 0, 1, 5, 10] to represent zero, small, medium and large height errors in both directions.

The output membership function's centers are chosen at [-50, -20,-10, 0, 10, 20, 50]. These angle output values are chosen low to ensure forward flight while climbing. The inputs are directly mapped to the outputs. Thus a larger positive height error will result in a positive vector angle.

The height corrector controllers input membership functions for the height error and the change

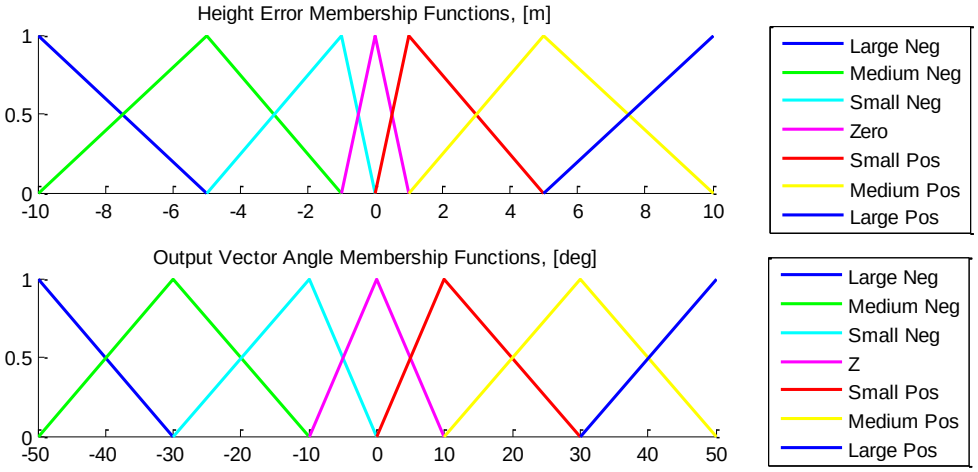


Figure 5.16: Membership functions for the vector angle estimate controller

in height error, as well as the output membership function for the incremental change in the angle are shown in Figure 5.17.

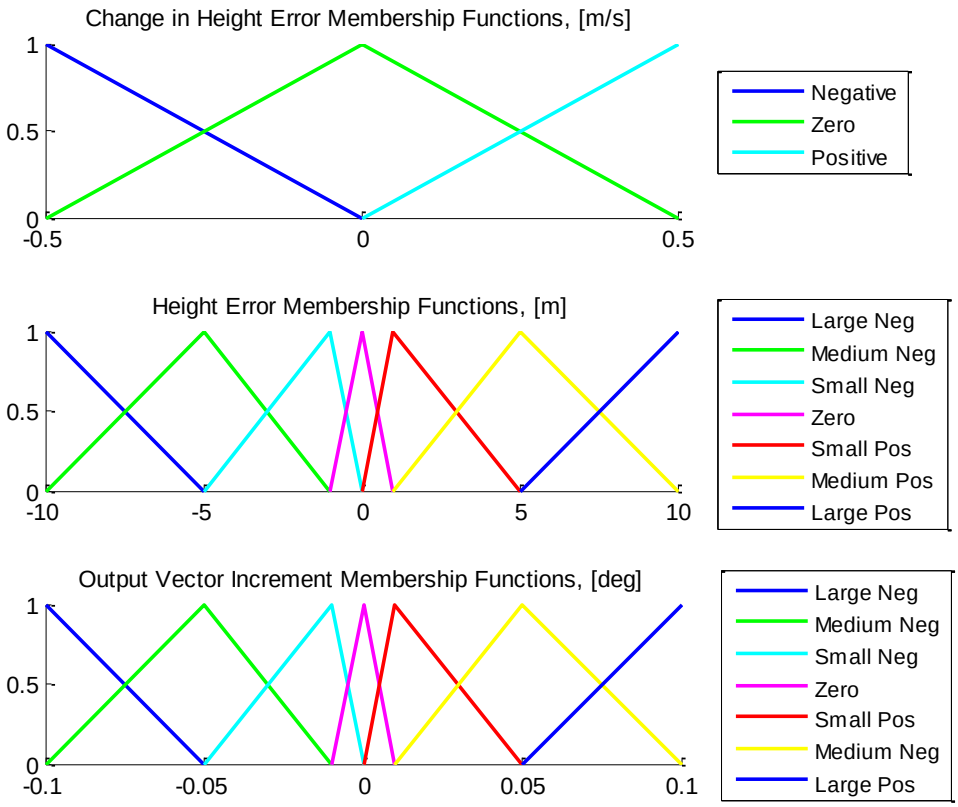


Figure 5.17: Membership functions for the vector angle correction

The centers for height error input functions are chosen at [-10, -5, -1, 0, 1, 5, 10] as before. The

centers for the membership functions for the change in height error are chosen at [-0.5, 0, 0.5]. These values are considered sufficient for an indication of change in both directions and are estimated from in-flight measurements and from simulation. The purpose of using this additional input parameter is to ensure that the vector angle is decreased before the commanded height is reached, so as to decrease overshoot.

The incremental outputs membership functions are chosen at [-0.1, -0.05, -0.01, 0, 0.01, 0.05, 0.1]. An easy way of visualising these increments is by converting it to a "change per second" value. This translates into additional change of 0.05^o per second for small increments, and 0.5^o per second for larger increments.

The rule base was designed with several aspects in mind:

- When the height error is small, the vector angle should be adjusted slowly. Therefore a small change should be made to the current vector angle. [red]
- When the height error is large, the thrust vector should be adjusted as quickly as possible. [blue]
- When the height error becomes very small and the airship is approaching the commanded very quickly, the vector angle should be adjusted slightly in the opposite direction. [green]

Taking all these aspects into consideration, the rule base can be summarized in the following 2D matrix:

Height Error			
-10	NS	NL	NL
-5	PL	NM	NL
-1	PM	NS	NM
0	Z	Z	Z
1	NM	PS	PM
5	NL	PM	PL
10	PS	PL	PL
	-0.5	0	0.5

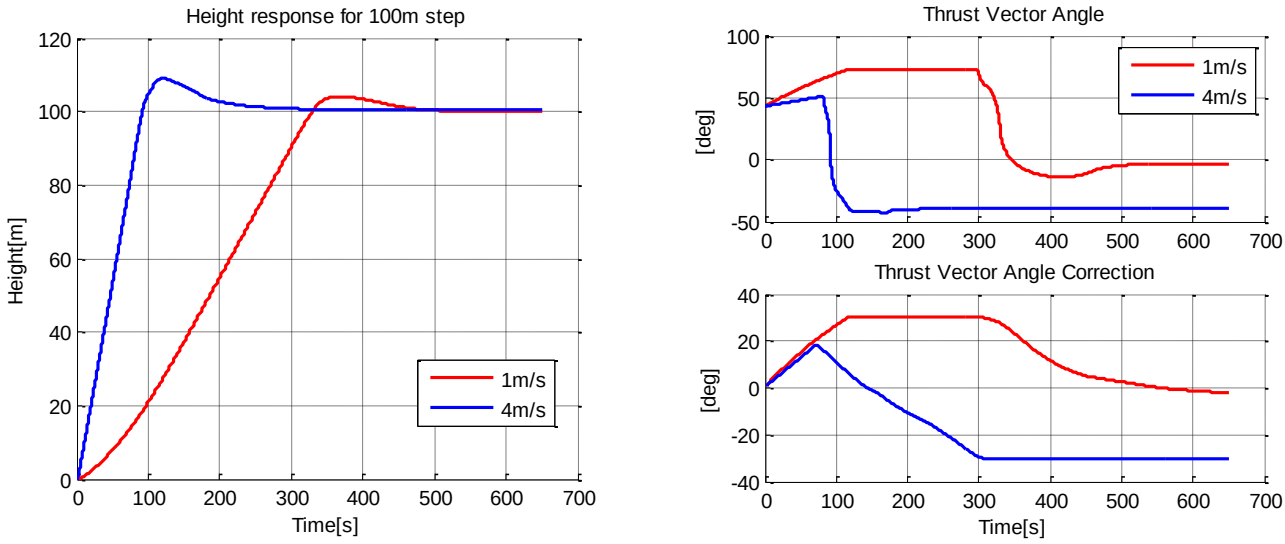
Change in Height Error

5.5.3 Simulation Results

The simulation results for a 100m step at 1m/s and 4m/s forward flight are shown in Figure 5.18.

The controller shows adequate performance at both speeds. The faster climb rate and increased overshoot at the faster speed can be attributed to the additional lift experienced due to increase in angle of attack. The following table illustrates the effect of the angle corrector on the settling time, overshoot as well as the steady state error:

	With Corrector			Without Corrector		
Speed m/s	2% Settling	Overshoot %	SS Error	2 % Settling	Overshoot %	SS Error
1	435s	4.5%	0m	558s	0	0.1m
4	220s	9%	0.5m	N/A	0	9.3m



(a) Height Response

(b) Actuator Commands

Figure 5.18: 100m Height Step Command at different 1m/s and 4m/s forward flight

This illustrates that the height corrector controller improves the steady state error at high speeds and decreases the settling time at lower speeds. The 2% settling time is not applicable at the higher speed without the corrector since a steady state error prevents the height to settle within 2%. Although the overshoot for a 100m step is rated as a percentage, it should be noted that for an airship climbing at the maximum rate, the overshoot would be close to 10m for any height command.

Small steady state errors may still arise at high speeds since the vector angle may not provide sufficient actuation to counter the additional lift. Therefore, the saturation limit for the corrector is also an important design factor. If the limit is set too low, the steady state error may not be removed completely, whereas if the limit is set too high, the overshoot may increase as it takes longer to adjust the correction angle. Initially a saturation limit of $\pm 30^o$ is set to provide a balance between actuation range and overshoot.

5.6 Speed Controller

5.6.1 Block Diagram

The block diagram for the speed controller is shown in Figure 5.19. It consists of 2 parts - the initial thrust assumption and then followed by a thrust corrector which corrects the thrust based on the velocity error.

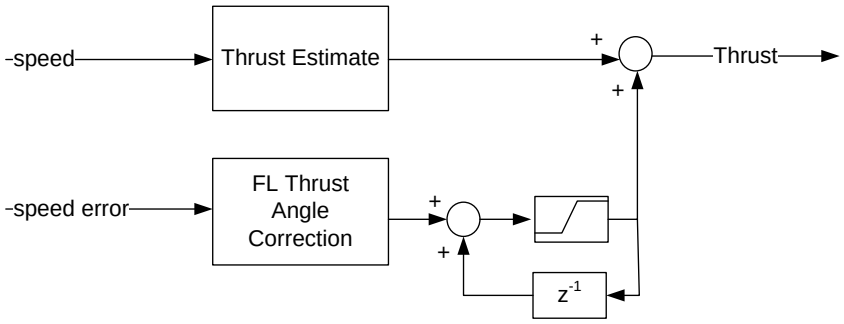


Figure 5.19: Block diagram of velocity controller

5.6.2 Implementation

The initial thrust assumption is made by calculating the thrust required to overcome the estimated drag at the required speed. Therefore

$$T_{estimate} = 0.5\rho C_D u^2_{commanded} S \tag{5.4}$$

This alone will not ensure that the required speed is reached as this estimate could be inaccurate. External disturbances such as wind would also influence the performance. Therefore a fuzzy logic thrust corrector was implemented. As the error increases and decreases the thrust is constantly adjusted to compensate for this error command.

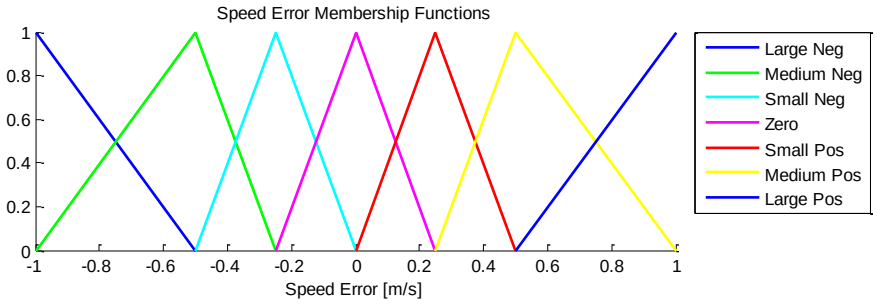


Figure 5.20: Input Velocity Error Membership Functions

The speed error can be calculated as the difference between the commanded speed and the measured speed, so that

$$u_{error} = u_{commanded} - u_{measured} \tag{5.5}$$

The speed error membership functions and the output is shown in Figure 5.20. The input membership functions centre coordinates is chosen initially at [-1, -0.5, -0.25, 0, 0.25, 0.5, 1] as to represent small, medium and large speed errors.

The output membership functions center coordinates are shown in Figure 5.21 and are chosen at [-0.4, -0.2, -0.1, 0, 0.1, 0.2, 0.4] for the incremental change every 0.2 seconds.

The rule base simply maps the speed error directly to specific increment values. The current saturation on the incremental total is set at ±30% to provide a reasonable balance between actuation

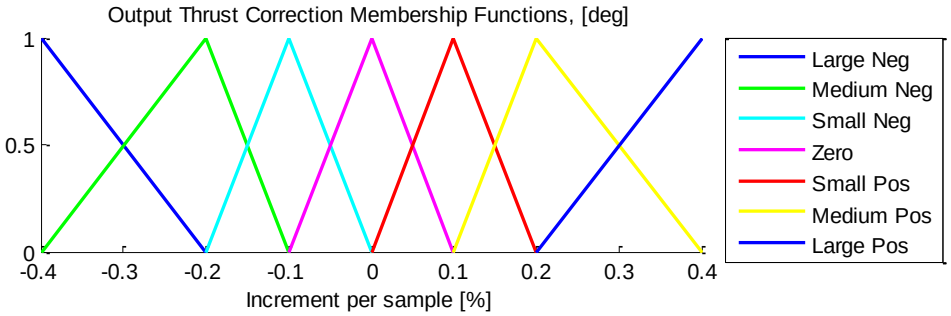


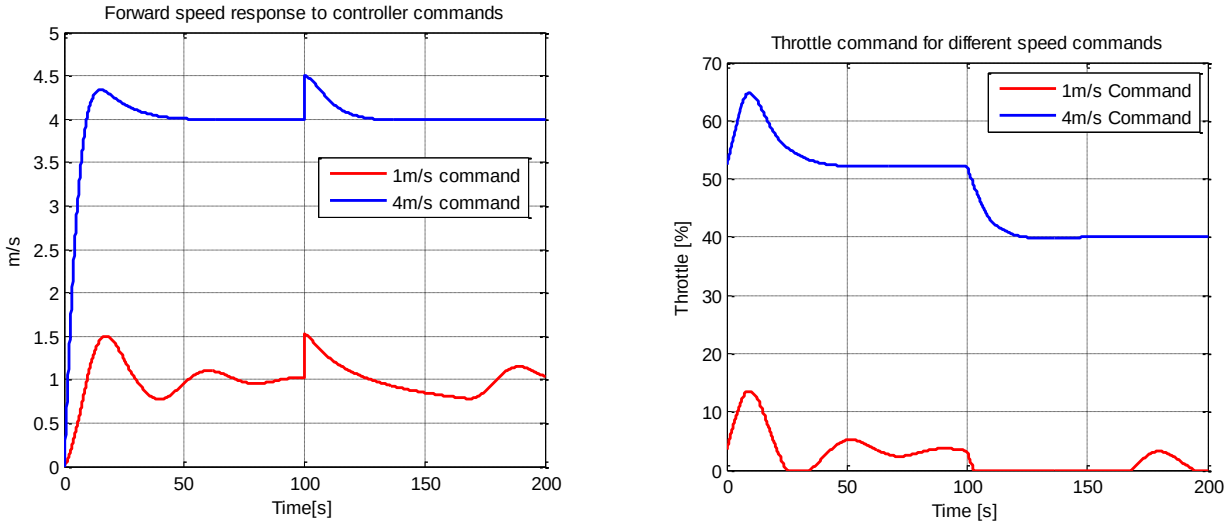
Figure 5.21: Output Throttle Increment Membership Functions

range and overshoot.

5.6.3 Simulation Results

The controller response for 1m/s and 4m/s commands is shown in Figure 5.22. For 1m/s command, the response shows large overshoot and a long settling time compared to that of the 4m/s command's response. This is also shown in the following table:

Speed	5% Settling Time	Overshoot %
1	67.5s	49%
4	23.5s	8.4%



(a) Speed Response

(b) Throttle Command

Figure 5.22: The speed and throttle command response of 1m/s and 4m/s speed commands

Both show reasonable controller command as well as sufficient handling of a sudden 0.5m/s step disturbance at 100 seconds. Steady state errors may arise when the saturation of the fuzzy controllers output is reached, and cannot provide additional actuation.

Several ways of improving the thrust controller include:

- Changing the saturation limit on the corrector as required.
- Changing the thrust estimation function to provide a better initial estimate.
- Changing the gain of the thrust corrector according to the required velocity.

5.7 Lateral Guidance

The lateral guidance strategy is constantly focused on minimizing the error between the vehicle and the trajectory, also called the cross track error. The cross track error is calculated as the distance between the airship's position perpendicular to the path between two waypoints. The cross track error can be calculated by first calculating the airships relative position to the first waypoint,

$$N_{rel} = N_{current} - N_1 \tag{5.6}$$

$$E_{rel} = E_{current} - E_1 \tag{5.7}$$

By rotating the axis system by the angle of the current trajectory heading between the two waypoints, the relative coordinates change to

$$\begin{bmatrix} N_{new} \\ E_{new} \end{bmatrix} = \begin{bmatrix} \cos\psi_{traj} & \sin\psi_{traj} \\ -\sin\psi_{traj} & \cos\psi_{traj} \end{bmatrix} \begin{bmatrix} N_{rel} \\ E_{rel} \end{bmatrix} \tag{5.8}$$

The cross track error y_{track} is given by the east position E_{new}, and the distance to next waypoint projected onto the trajectory line, x_{track}, is given by N_{new}. The airship's heading is set to the required trajectory heading plus an additional heading,

$$\psi_{commanded} = \psi_{traj} + \psi_{add} \tag{5.9}$$

This ensures the airship is directed to a position on the trajectory between the current position and the next waypoint, as illustrated in Figure 5.23. This additional heading can be calculated by

$$\psi_{add} = atan\left(\frac{y_{track}}{k.x_{track}}\right) \tag{5.10}$$

The parameter k is chosen as 0.5, therefore ensuring the commanded heading constantly directs the airship to a point halfway between its current position and the next waypoint. A higher value of k will cause the airship to move more gradually to the next waypoint, whereas a lower value of k will cause it to turn much sharper and cause more overshoot.

The commanded heading increases drastically if the cross track error is large and the airship closes in on the next waypoint. Waypoints are considered reached when the airship moves within a 10m radius in any direction of the waypoint.

A 0.2-second sampling was set for the controllers. This value was chosen to ensure that the controller will update the actuators at least once every 1-m of flight when moving at a speed of

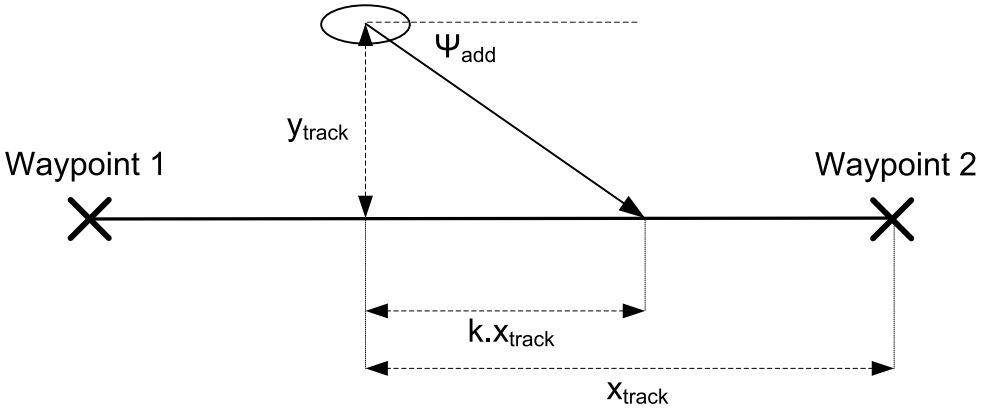


Figure 5.23: Guidance of the airship between two waypoints.

5-m/s. This value proved to be sufficient for adequate controller performance.

5.8 Three Dimensional Waypoint Flight

Figure 5.24 illustrates the performance of the heading controller combined with the guidance strategy under different wind speeds for waypoints placed 100m apart. A constant speed of 3m/s was commanded to ensure that the airship is able to overcome the simulated northen wind as shown. With a wind of 2m/s the airship misses the third waypoint, but is successfully guided back towards this waypoint against the wind.

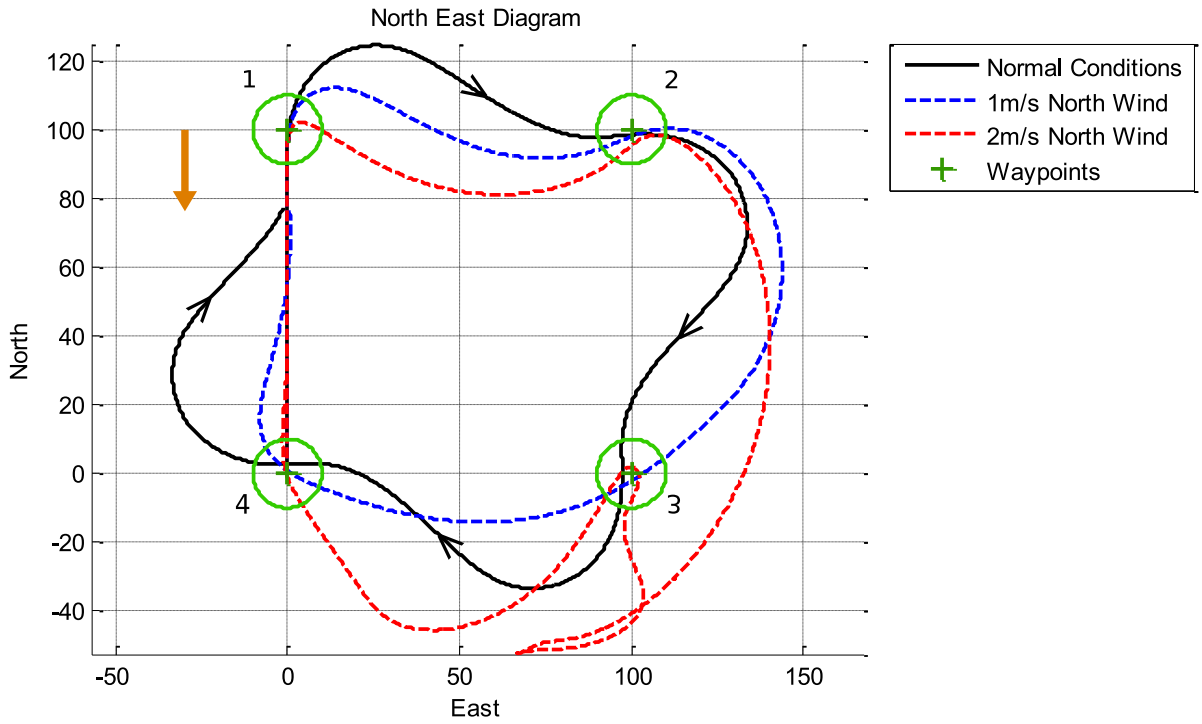


Figure 5.24: Trajectory of the airship during lateral waypoint flight at a commanded velocity of 3m/s

Figure 5.25 illustrates 3D flight. The second waypoint is placed 250m above the first waypoint to demonstrate vertical flight. The controllers successfully guide the airship in an upward spiral trajectory until the waypoint is reached. Slight overshoot of close to 10m is also visible, but the airship corrects the height while circling the waypoint. The reason for the spiral is due to the fact that the controllers are constantly trying to correct the lateral position until the airship is within 10-m of the waypoint and has reached the correct height.

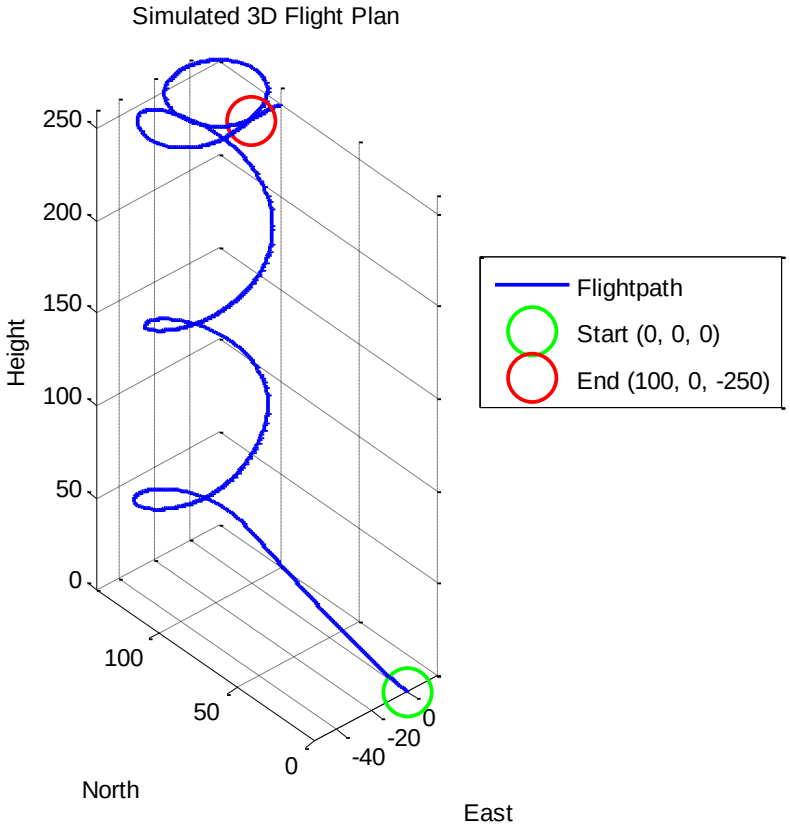


Figure 5.25: 3D Flight at a speed command of 3m/s

The OpenGL simulator developed in the ESL[1] is used to give a 3D visual output for the simulations. A scaled version of a Flight Gear[2] model of the Zeppelin NT is used to represent the airship. The simulator in action is shown in Figure 5.26.

[1]Electronics Systems Laboratory, Stellenbosch University. Credit to Reghardt Busch and Rudi Gaum.
[2]Flight Gear, Opensource Flight Simulator, www.flightgear.org

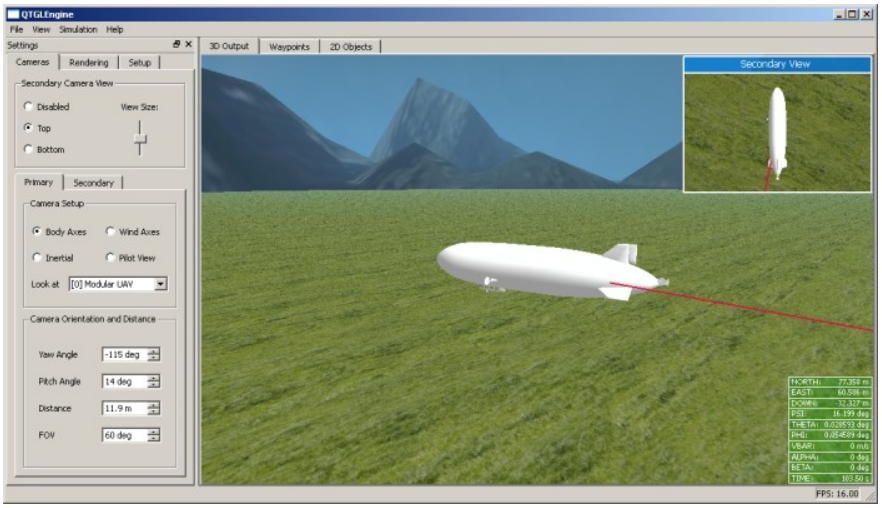


Figure 5.26: Screenshot of the OpenGL Simulator with an Airship model

5.9 Robustness and Stability

Fuzzy logic controllers are usually robust with regards to measurement noise, since the input measurements are classified into fuzzy variables. As previously mentioned, the membership functions have a large influence on the robustness of the controllers, as shown by the study done in [44]. The robustness of the fuzzy logic controllers are improved by the fuzzy integrator controllers which slowly correct the output based on the error.

Since the fuzzy logic controllers are non-linear controllers which are designed for non-linear models, traditional stability analysis methods such as the use of the s-plane or the Nyquist criteria cannot be performed. Several non-linear stability analysis methods which have been adapted for fuzzy logic control exist, as discussed in [46]. Due to time constraints, the stability of the designed controllers were not evaluated using these methods. Current stability is greatly improved by the following aspects of the original design:

1. The rules are chosen to settle at a constant output value. This can be seen from the rule base for the heading controller shown in Figure 5.27.

2. The integrator type fuzzy logic controllers such as the height and thrust corrector are slow and makes small changes.

3. The integrator fuzzy logic controllers' outputs are saturated to lessen overshoot and prevent integrator "windup".

5.10 Conclusion

The design and simulation results of the fuzzy logic controllers are discussed in this chapter. Simulation results prove sufficient controller performance for 3D flights and also suggests that the controllers should perform reasonably well during light wind conditions.

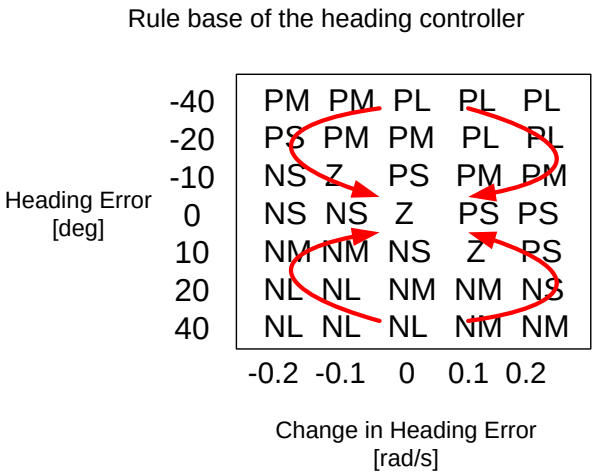


Figure 5.27: Indication of rules for the heading controller settling at the zero position

Although the initial implementation of these controllers are simple, the controllers provide a viable platform for future improvement, especially with the combination of adaptive control where the membership functions and rules can be adjusted dynamically.

Chapter 6

Hardware and Software Design

6.1 Introduction

The hardware profile is illustrated in Figure 6.1. The inertial measurement unit(IMU), Extended Kalman filters as well as the initial Groundstation implementation was designed and implemented by Johan Bijker in [47]. An additional servo control board was designed to control the airship's actuators, while several updates were made to the initial groundstation implementation to enable the controllers and the autopilot, to allow interfacing directly to the actuators and also for the design of the desired flight plan.

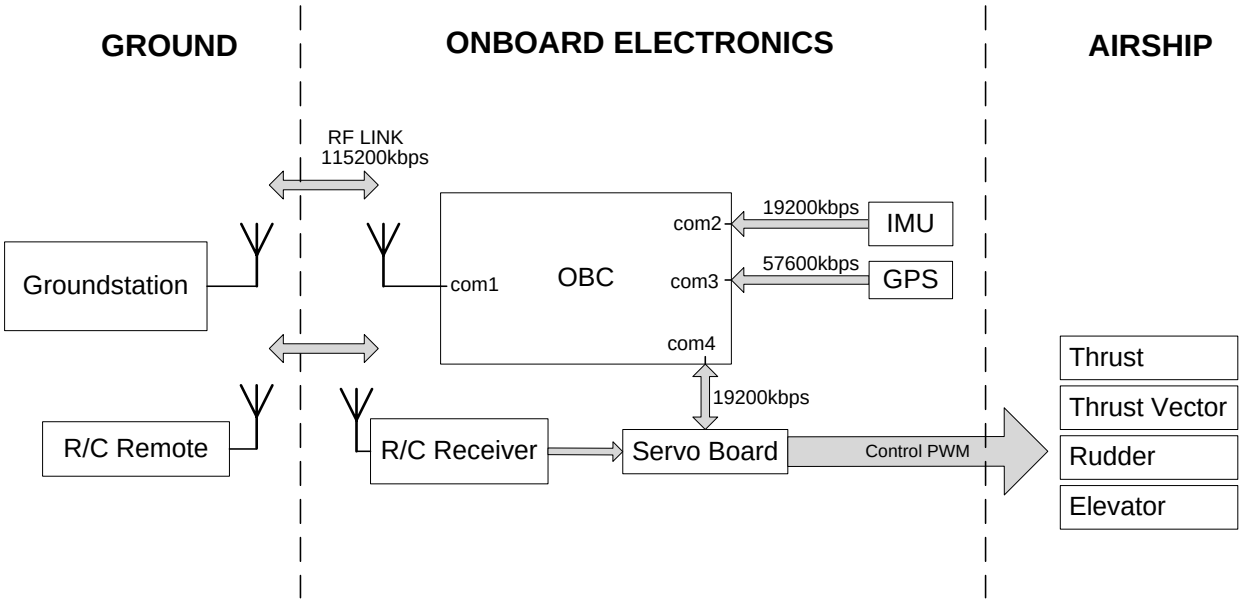
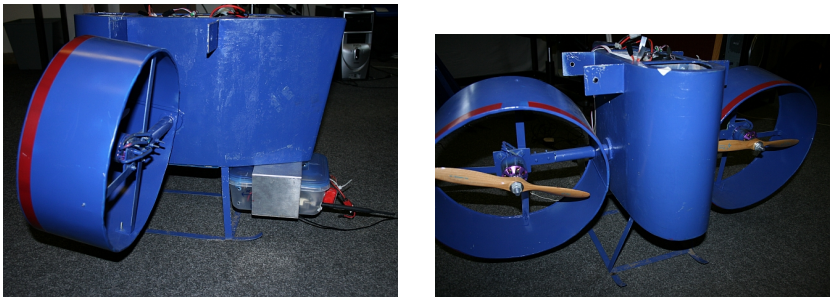


Figure 6.1: System diagram for the airship.

The FZ-800E gondola is pictured in Figure 6.2. No changes were made to this gondola, apart from adding a mount for the electronics, which can be seen in the side view.



(a) Side View (b) Front View

Figure 6.2: The FZ-800E's Gondola, Side and front view

6.2 Onboard Computer and Sensors

The onboard computer (OBC) is the Ampro 420 CoreModule PC/104 Computer. It features a 133MHz 486-processor, 64MB SDRAM and 512MB CompactFlash card for program storage. The OBC also includes four RS-232 serial ports which are used for input and output to the RF link, IMU, GPS as well the servo board (COM Ports 1 to 4 respectively).

The following sensors were used in determining the position, attitude and velocity of the airship.

- 3 Axis Accelerometers
- 3 Axis Magnetometers
- 3 Axis Rate sensors
- GPS Receiver

These sensors, their respective calibration and the implemented Extended Kalman filters are discussed in [47].

6.3 Servo Control Board

The functional architecture for the servoboard is shown in Figure 6.3. The servoboard is considered the most crucial part of the hardware profile, since it is responsible for controlling all the actuators during manual and automatic flight. The schematics for the hardware design is attached in Addendum B.

6.3.1 Power Supply

Two separate power supplies were implemented on the servoboard. All the electronic components is supplied 5V from a linear regulator, while the servos are supplied from a 5.5V, 10A switchmode supply. Provision has been made for a backup battery supply, using the configuration in Figure B.2. The battery voltage is constantly monitored and sent back to the groundstation.

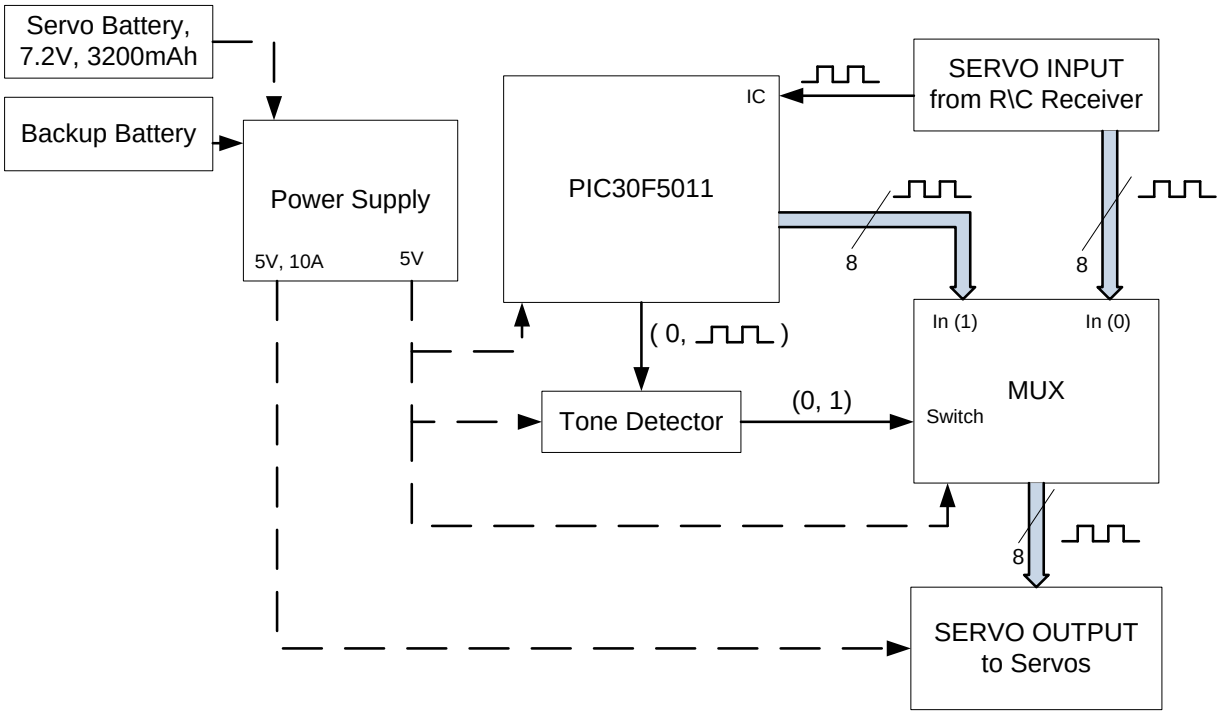


Figure 6.3: Diagram of Servoboard Architecture

6.3.2 Servo Control

The servos can be controlled with pulse-width modulated (PWM) signals directly from the R/C receiver, or by those generated by the Microchip dsPIC30F5011 microcontroller.

To control the servos from the microcontroller, the microcontroller should first send a PWM signal to the tone detector. The tonedetector used, is the LM567C integrated circuit, with the circuit diagram shown in Figure B.3. The tonedetector is currently setup to respond to a 100kHz signal. Upon receiving a valid signal, the tone detector switches the multiplexer to direct the microcontroller's PWM signals to the servos instead of the R/C receiver's commands.

This is done for extra hardware redundancy - should the dsPIC fail, it is assumed that it will not be able to generate a PWM signal to the tone detector, thus switching the output to that of the R/C receiver.

One of the R/C receivers channels was mapped to an "Input Capture" pin on the microcontroller. This channel is used to switch between the microcontroller and the R/C receiver's PWM signals, directly from the R/C remote. This is useful as it can be used as a backup to regain control of the airship should the OBC or software fail.

The instruction to switch between the two sets of signals can also be sent from the groundstation, which will instruct the servoboard via the OBC.

6.4 Software Development

6.4.1 Data flow

The data flow between the groundstation, OBC, sensors and the IMU can be divided into 3 parts

1. Data flow due to the sensors and Kalman filter estimations as implemented by Johan Bijker.
2. Data flow for control from the onboard computer.
3. Data flow due to the controllers.

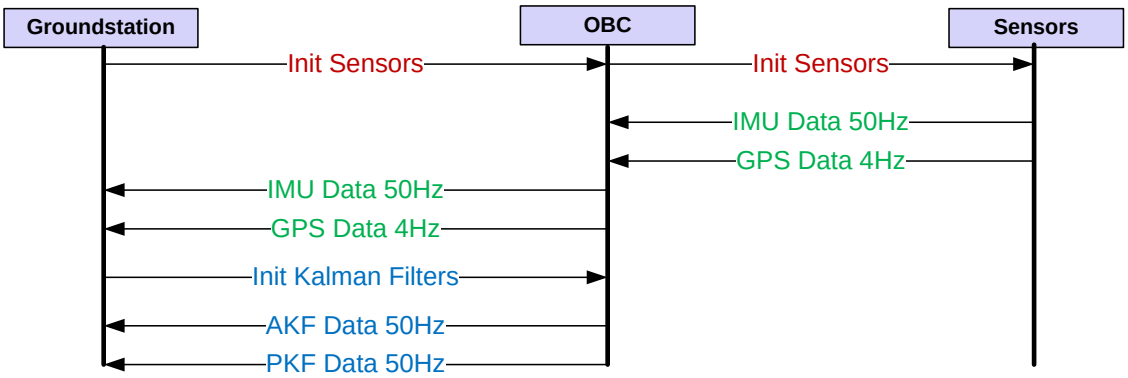


Figure 6.4: Sensor data packets flow diagram

Data flow due to the sensors and estimations is illustrated in Figure 6.5. The sensors can be initialised and calibrated from the groundstation. The attitude Kalman filters (AKF) and position Kalman filters (PKF) can be intialised from the groundstation. The raw sensor data, as well as the estimations from the Kalman filters are send back to the groundstation.

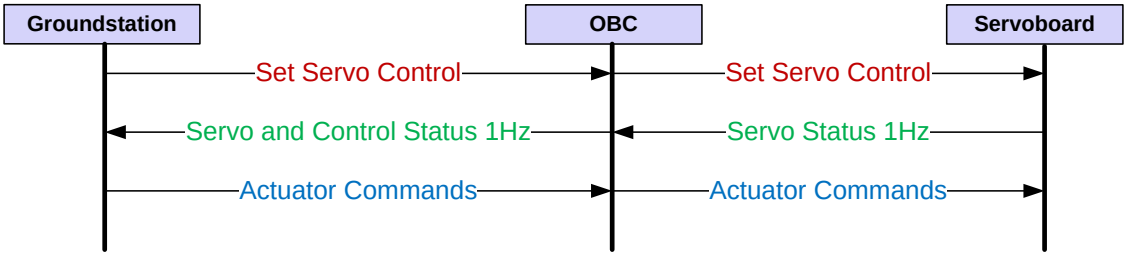


Figure 6.5: Servo data packets flow diagram

Data flow due to the control from the computer is illustrated in Figure 6.5. The commands enabling the servoboard to be controlled from the computer can be send from the groundstation. The status of the servoboard, which includes the transmission statistics, the state of the computer control, is sent by the OBC to the groundstation. When computer control is activated and the first servo status data packet reaches the groundstation, the groundstation is able to send actuator commands.

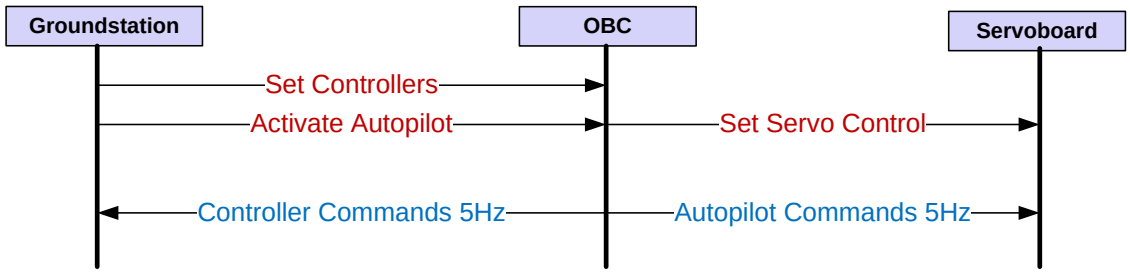


Figure 6.6: Autopilot/Controllers data packets flow diagram

Data flow due to the control actions is illustrated in Figure 6.6. Before the autopilot is activated, the controllers first have to be set. When the controllers are activated, the OBC sends commands to the actuators through the servoboard and also sends controller specific data to the groundstation for logging.

Data packets sent between the groundstation, OBC and servoboard contains an 8-bit Fletcher algorithm checksum. This checksum adds an additional 2 bytes to each datapacket and can be calculated by simply summing all the bytes of the data package together, using the following algorithm:

Result: 8-Bit Fletcher Checksum
chksum1 = 0;
chksum2 = 0;
for *i* ***in*** *n-bytes data* **do**
    chksum1 += data[i];
    chksum2 += chksum1;
end

The telemetry from the flight is sent to the groundstation at a speed of 115200 baud. As the highest frequency data is transmitted is 50Hz, the maximum number of bytes that can be sent each 0.02 seconds amounts to 230-bytes. Although most of the data is sent at a much lower frequency, the worst case scenario is shown in Table 6.1 as a percentage of the 230-byte limit.

Packet	Frequency (Hz)	Size (bytes)	% of worst case
IMU	50	30	8.7%
AKF	50	33	9.6%
AKF Variances	50	33	9.6%
PKF	50	29	8.5%
PKF Variances	50	29	8.5%
GPS	4	50	14.6%
Flight Data	5	42	12.2%
Longitudinal Controller	5	12	3.5%
Lateral Controller	5	23	6.7%
Velocity Controller	5	16	4.7%
Servoboard Status	1	18	5.2%
OBC Status	1	28	8.2%
TOTAL		343	100%

Table 6.1: Data Packets Sizes

Although this result is not ideal, since the worst case scenario data packet is 49% more then the transmittable packet, actual performance can be improved upon using the following features:

1. Disabling certain data packets from being sent to the groundstation.
2. Implementing large transmit and receive FIFO buffers on the groundstation as well as the OBC software. This will give the OBC time to process the data packets that arrive at the same time.

Most of the controller data are recorded for controller verification purposes. Several future improvements can be made and are discussed in Chapter 8.

6.4.2 Groundstation Software

The groundstation, pictured in Figure 6.7 is responsible for the following functions:

- Logging of all the data received from the OBC - including sensor outputs, Kalman Filter estimations, as well as the controller commands.
- Displaying all the data received from the OBC.
- Controlling the actuators directly.
- Arming of the different automatic controllers as required.
- Setting up the required flight plan in NED coordinates and uploading it to the OBC.
- Upload of sensor calibration data to the OBC.

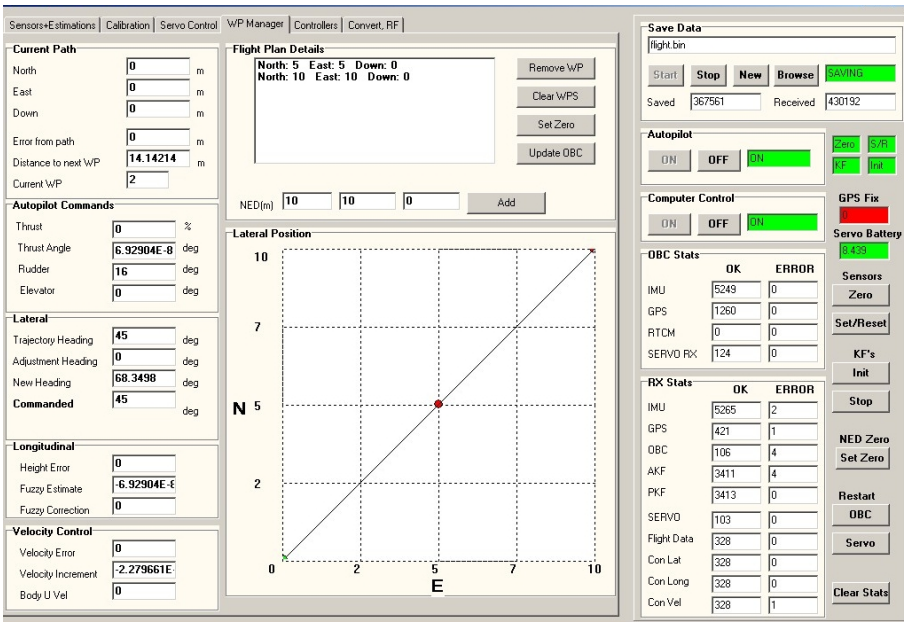


Figure 6.7: Groundstation screenshot

A more detailed discussion of the groundstation functions and graphical user interface can be found in Addendum C.

6.4.3 OBC Software

The initial implementation of the OBC software was done by Johan Bijker. Further development and improvements were carried out to perform the additional tasks required for controlling the servos, handling waypoints and the automatic control.

A real time operating system (RTOS), AMX86 [48], runs on the OBC. The software is implemented in C and integrated with the AMX libraries and a single executable is created when compiled. This file is executed when MS-DOS boots up and then takes control of the hardware. Using an event driven RTOS is advantageous when it comes to timing sensitive tasks. When two tasks occur at the same time, preference is given to the task with the higher priority ensuring that a critical task gets preference above a non-critical task. A more detailed discussion of the AMX implementation can be found in [47].

6.4.4 Servoboard Software

The servoboard has a simple communication interface, since its primary function is to control the servos. It receives the servo commands from the OBC and converts them to the appropriate PWM signal. To ensure a valid pulse width, the PWM signal is saturated below 1-ms and above 2-ms.

The servo board also sends a status update which contains the following information:

1. Transmission statistics - the number of commands received correctly as well as the number of transmission errors
2. The "listening" status of the servoboard - e.g. if it listens to the commands from the computer.

The servoboard is also responsible for switching between the PWM signals from the R/C receiver and the PWM from the dsPIC as commanded by the OBC. The control of the actuators can be switched from manual to automatic and back at any time during the flight, using the R/C remote.

6.5 Conclusion

The hardware implementation proved to be sufficient for the implementation of the controllers. Several improvements can be made and are discussed in Section 8.2.3.

Chapter 7

Flight Test Results

7.1 Introduction

Flight tests were done at Coetzenburg Sport Stadium in Stellenbosch. Flight testing was limited due to wind conditions and time constraints. The airship was always kept at a very low altitude, in line of sight and had a safety line attached should it drift away due to wind. The airship's weight was slightly heavier then neutral buoyancy, to ensure the airship tends to slowly come down when not moving forward.

The airship was transported using the trailer shown in Figure 7.1. This trailer is used to house and transport the airship after the hull has been filled with helium. This was need to lower the cost due to cost due to helium loss, when deflating the airship after flight tests. The trailer is also able to transport 2 helium cylinders, should the airship need to be refilled on-site.

(a) Side View (b) Back View

Figure 7.1: Trailer used to transport the airship.

The measurements shown in the following sections are from the Extended Kalman filter implemented in [47].

7.2 Heading Controller

The heading controller was tested by supplying the controller with a reference angle and applying a constant thrust. Flight test results are shown Figure 7.2 in which a heading step command close to 85^o is given.

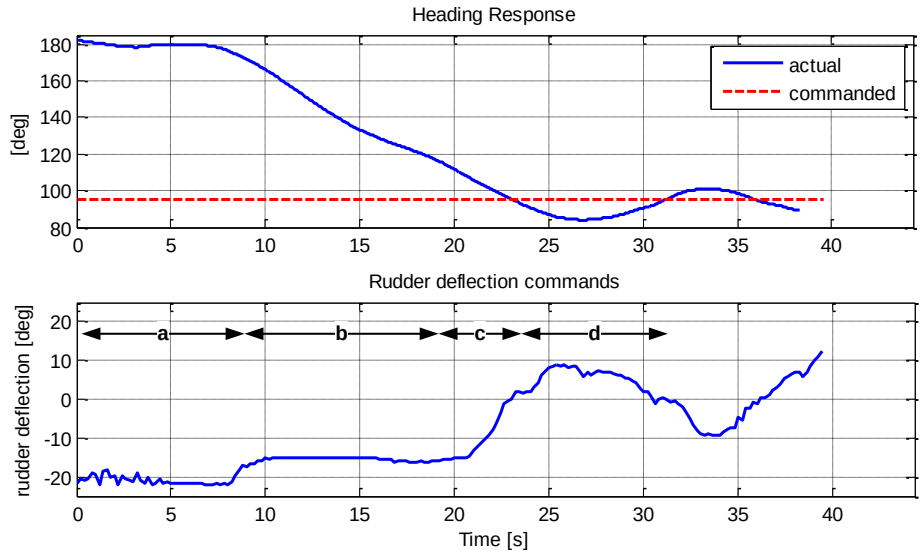


Figure 7.2: Heading controller response at an average flight speed of 3m/s

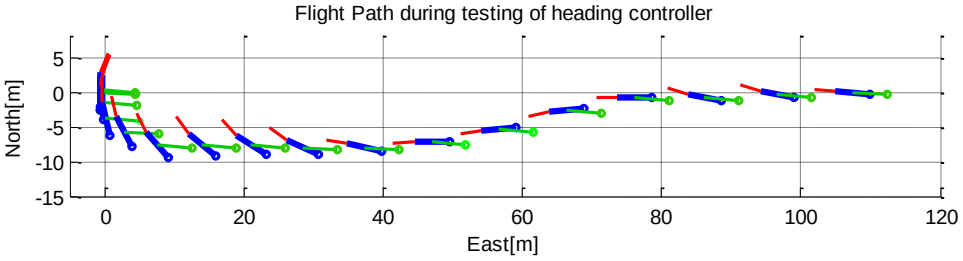


Figure 7.3: Flight path during heading controller test

This flight experienced an overshoot of 12%, which is reasonable considering that there was a light wind experienced. The evidence of wind is clearly visible in the flight path of the airship as shown in Figure 7.3. The blue arrow indicates the airship's position and orientation, the red section is the rudder direction and the green line indicates the commanded heading. The commanded rudder deflection can be divided into 4 different stages:

Stage a: Larger rudder deflections as no rotation about yaw axis is experienced initially.

Stage b: As the heading error is decreasing at a steady rate, the rudder angle is kept at a constant deflection angle.

Stage c: As the heading error is smaller and the heading is approaching the commanded heading, the rudder angle is decreased to compensate for possible overshoot.

Stage d: The controller has overshot the commanded heading, therefore counter steer.

By noticing the different stages during the control of the heading angle, the heading controller can be improved upon by changing the membership functions and the rule base. The following changes are possible:

Stage a: A faster initial yaw rate can be achieved by increasing the deflection angle specified as a *large* deflection.

Stage b: The yaw rate for this stage can be increased by increasing the deflection angle specified as a *medium* deflection.

Stage c: The heading error angle where the rudder deflection size should decrease, should be larger, to ensure that the overshoot is reduced. This can be done by specifying another variable to classify heading errors between *small* and *medium* heading errors.

Stage d: The overall overshoot should be reduced by improving stage c, thereby reducing the need for this stage.

Figure 7.4 shows the results for another heading controller test, this time for a step command close to 50^o. This response experiences 4% overshoot, and similiar controller behaviour is experienced.

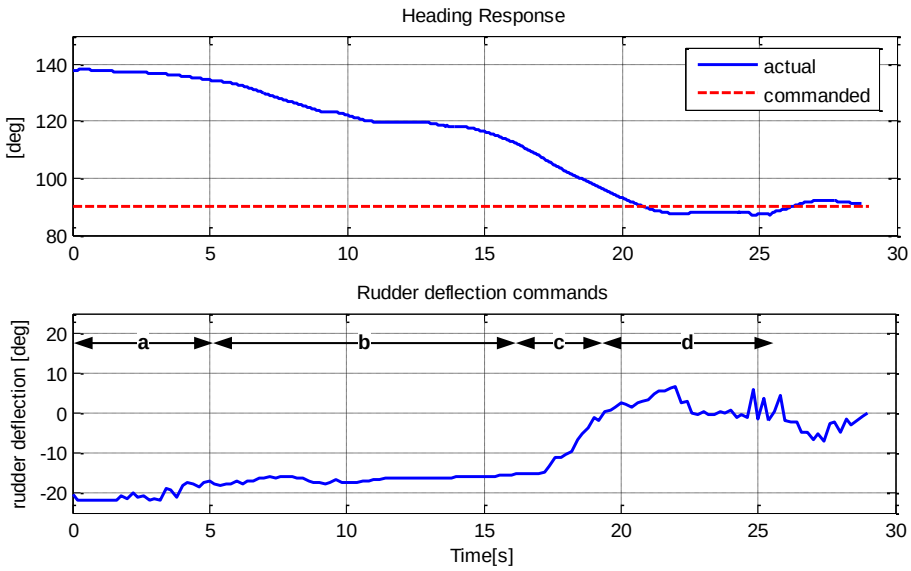


Figure 7.4: Heading controller response at an average flight speed of 4m/s

Although more accurate tests can be done by flying at a constant speed and then applying a heading step command, these tests show the potential of the fuzzy logic heading controllers. These tests were also done in slight windy conditions, where wind gusts could also have an influence on controller performance.

7.3 Height Controller

The performance of this controller greatly depends on the height measurement. In Figure 7.5 the flight response for a 10m height command is illustrated while the flight data is shown 7.6. It can clearly be seen how the vector angle command decreases as the height error becomes smaller.

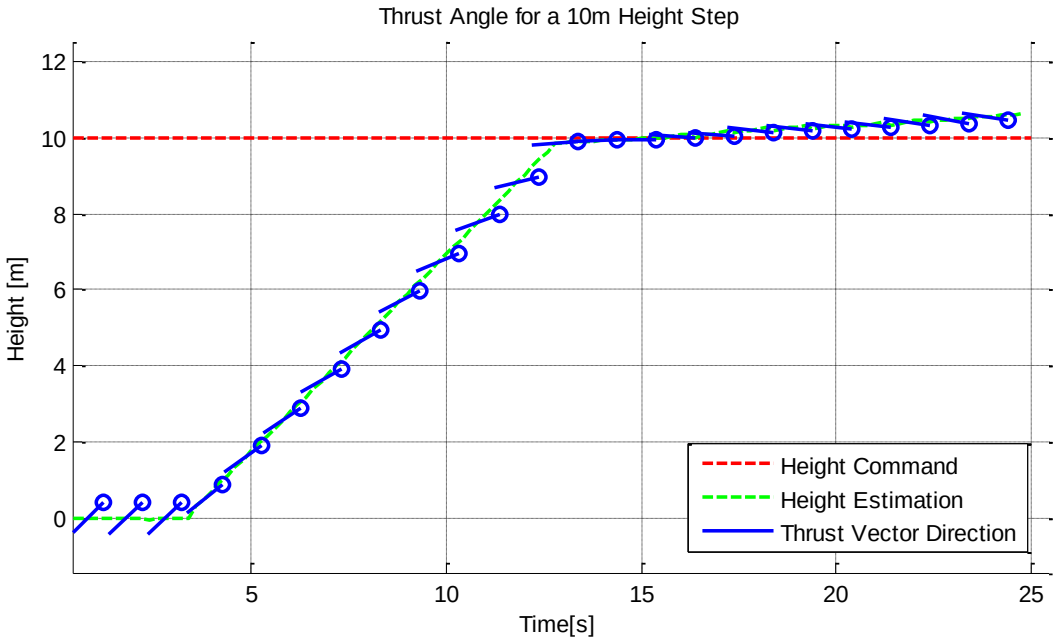


Figure 7.5: Height Controller Response

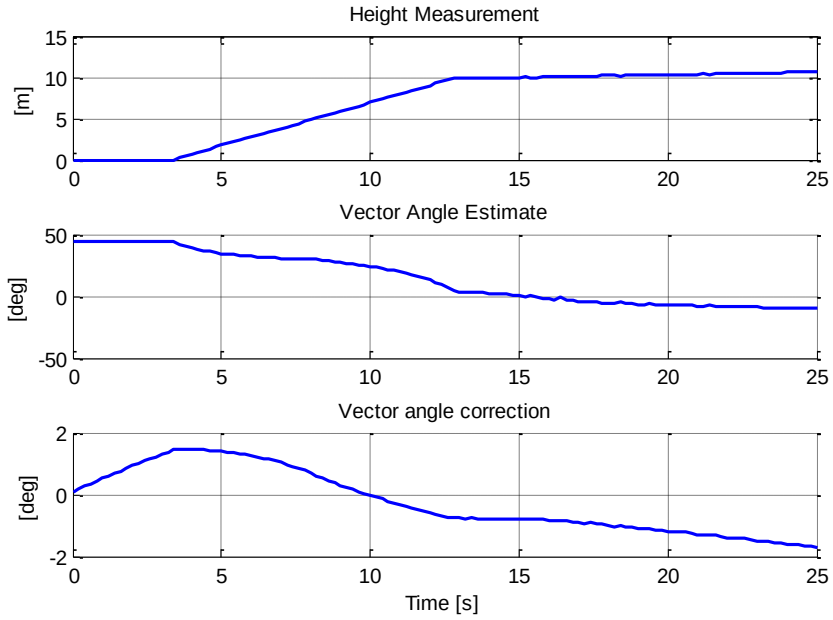


Figure 7.6: Height Controller Flight Data

The effect of the vector angle correction controller is not that evident during this flight test. Although further testing is required to validate the performance of the correction controller, the following can be seen from the flight data:

- The angle increases slowly when the measured height is lower than the commanded height.
- The angle decreases slowly as the height error decreases.
- The correction is kept constant when the commanded height is reached.
- The correction angle is slowly lowered to combat the increasing steady state error.

Problems were experienced with the height estimation accuracy, due to the fact that GPS height accuracy has a large effect on small height step commands. If the airship is slightly nose heavy, it is not able to create a large enough angle of attack to climb automatically when at flight. The performance of the height controller would improve considerably if the height controller is only switched on after a certain forward speed is attained. This aspect should be build into the guidance strategy and mission planner.

The following improvements can be considered for the height controller:

- By using the elevator the pitch angle can be increased to increase the lift due to angle of attack.
- The height controller is only switched on after a certain forward speed is attained.
- Additional height measurement sensors for determining height such as pressure sensors.

7.4 Speed Controller

Figure 7.7 illustrates the performance of the speed controller for a commanded speed of 3.5m/s in forward flight.

The thrust correction during this flight test can be divided into 4 parts:

Stage a: Build up thrust slowly as the current speed is lower than the commanded speed.

Stage b: Do not increase thrust as the commanded speed is reached.

Stage c: Decrease thrust slowly as slight overshoot is experienced.

Stage d: Decrease thrust at a faster rate as larger overshoot is experienced.

Although controller performance is compromised due to external disturbances, the controller tries to compensate by adjusting the thrust accordingly. The initial thrust estimate proved to be reasonable as the thrust correction controller seems to work effectively around this point.

The main improvement that can be made to this controller is that additional measurements can still be included, e.g. axial acceleration measurement. This controller could also include a learning capability, where the results of previous estimates and the corrections upon them are stored, to improve the initial estimate.

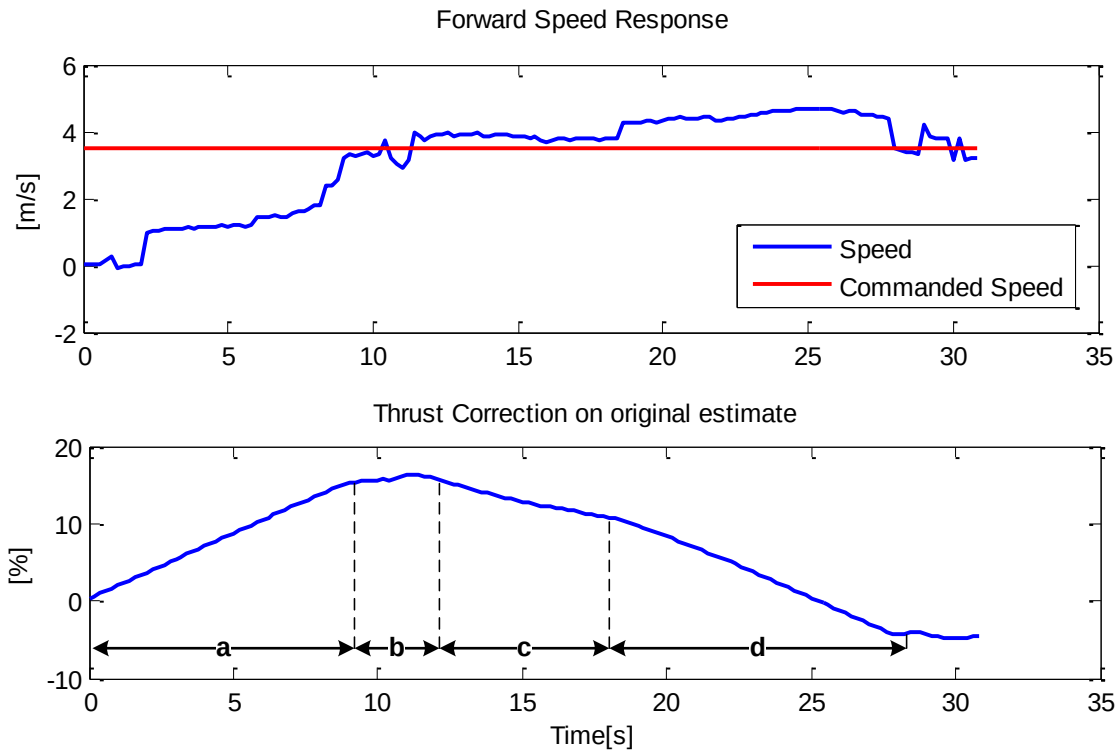


Figure 7.7: Speed Controller Response

7.5 Lateral Waypoint Flight

Waypoint flight was tested by specifying several waypoints. One waypoint was chosen close and another waypoint chosen slightly further away to illustrate the changes in the guidance strategy once the waypoint is reached. With these tests, the guidance strategy, heading controller as well as the speed controller are tested simultaneously.

Figure 7.8 shows the flight path for for a commanded flight speed of 4m/s. The blue arrow indicates the airship's position and orientation, the red section is the rudder direction and the green line indicates the commanded heading. The waypoints as well as the rendezvous radius (the target radius to reach), are also shown. The airship experiences a wind gust coming from the west, pushing it off course. Figure 7.9 shows the estimated distance to the next waypoint, the path error which increases due to the wind gust, as well as the current waypoint's number.

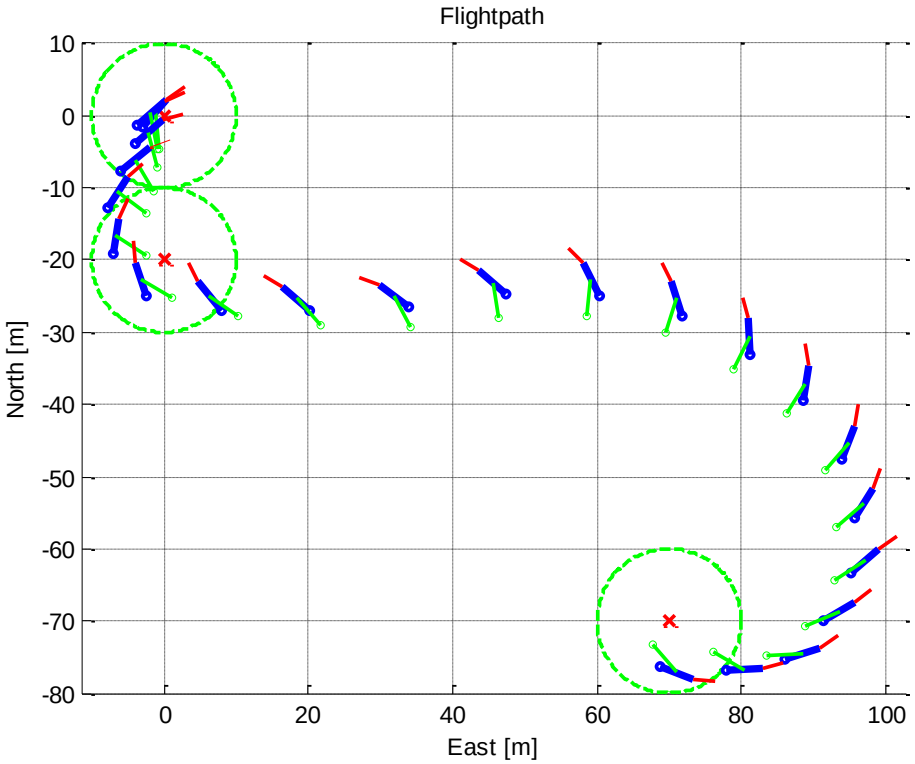


Figure 7.8: Flight Path at 4m/s commanded speed

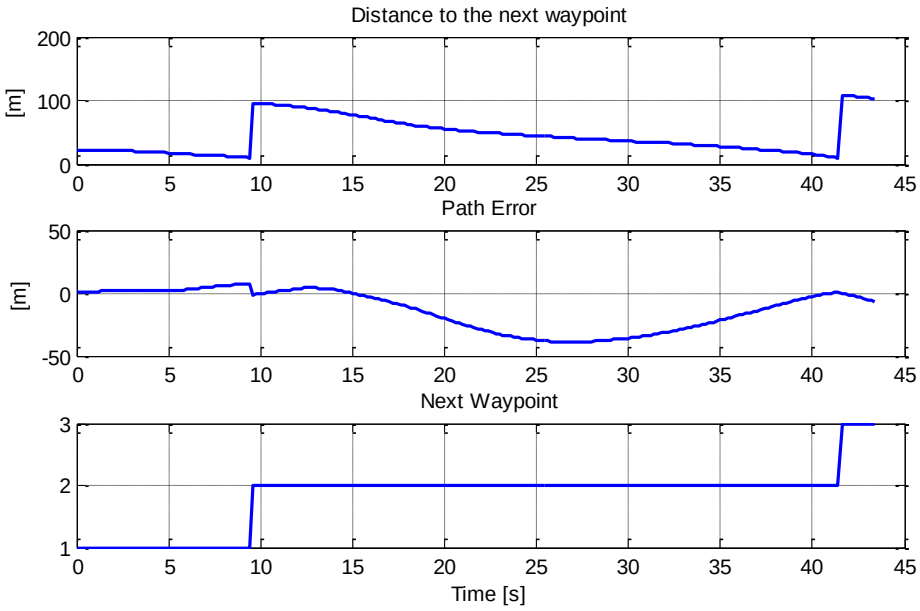


Figure 7.9: Flight data during flight test

This flight test was also recreated in simulation by specifying the same controller commands. Simulation show similiar behaviour under light wind conditions.

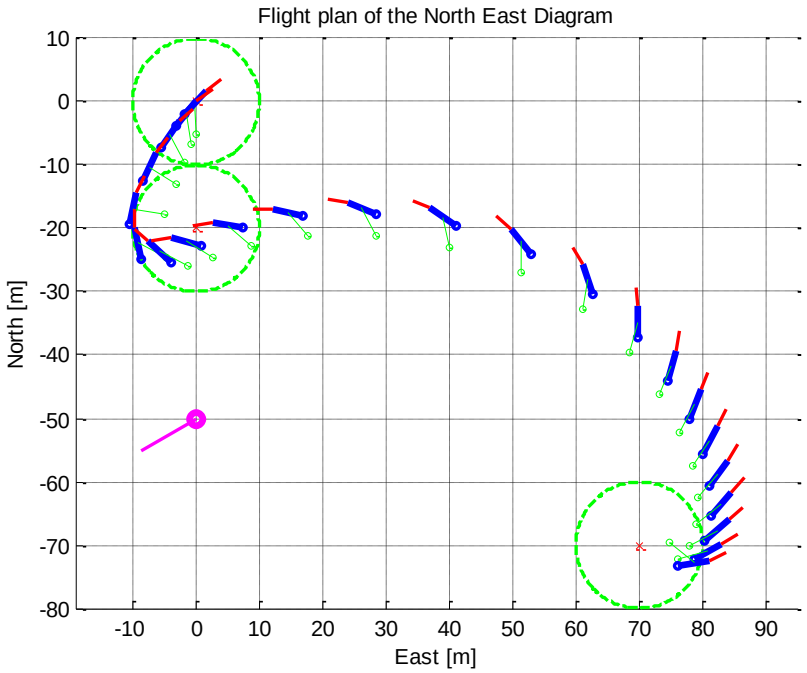


Figure 7.10: Simulated Flight Path at 4m/s commanded speed under light wind conditions

Figure 7.11 shows the flight path for the same waypoints, but at a commanded flight speed of 3m/s. During this flight, a wind gust moves the airship from the required trajectory when it is close to the second waypoint. The airship is laterally underactuated and unable to respond quickly enough to the disturbance, but the guidance strategy still tries to guide it back to the missed waypoint.

7.6 Conclusion

The ideal was to test the controllers in windless conditions, but unfortunately light winds were experienced during these tests, which affected the performance of these controllers. The controllers still showed reasonable performance, but further tuning is required for better performance. Full autonomous 3D flight was not tested due to the underperformance of the height controller.

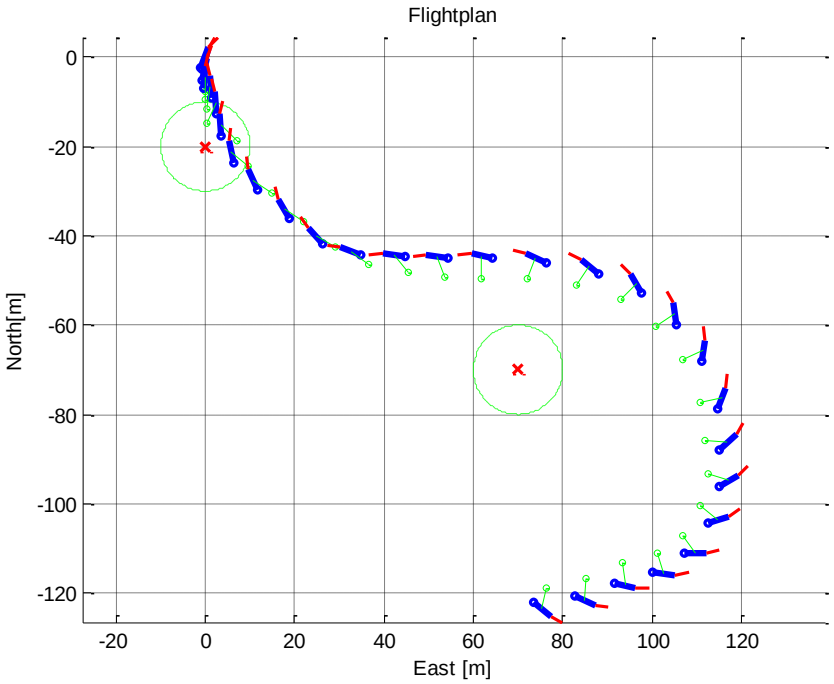


Figure 7.11: Flight Path at 3m/s commanded speed

Chapter 8

Summary and Recommendations

8.1 Summary

In this thesis the design and implementation of a flight control system for an airship is discussed. This section summarises the work done.

Two separate airship models were implemented in a Matlab/Simulink environment and these models differ mainly on the specification of the aerodynamic forces. The model derived by S.B.V. Gomes in [11] uses wind tunnel data which was stored in lookup tables. The Gomes model does not model the control surfaces separately and it is difficult to adjust the aerodynamic response intuitively using only the windtunnel data.

It was decided to continue with the dynamical model as derived by Yuwen Li [1], as it is easier to break up into separate forces. Although the model deviates from the actual flight results, the model's lateral behaviour can be adjusted to yield similar lateral behaviour as obtained from practical results.

Two linearisation methods were used to linearise the airship model - analytical and numerical linearisation. Numerical linearisation was implemented should the need arise to linearise the Gomes model, since it is difficult to linearise due to the aerodynamic data being stored in lookup tables.

The Yuwen Li model was linearised using both linearisation methods which delivered similar results. A modal analysis was performed on the Yuwen Li model. The model's linear behaviour compares well with the linear behaviour of the modal characteristics of the YEZ-2A airship [11]. To form a clearer understanding of the model, the effects of certain forces on the open-loop poles were investigated. The normal forces on the fins proved to be one of the most influential parts of the model, since it has a large influence on the yawing rate of the airship, as well as the damping of the oscillatory pitch mode.

Although the linearisation results compared well with the non-linear model's behaviour, it was not used for the design of the controllers. These results can be useful for future work on the airship model as well as the design of conventional controllers.

The hardware consists of the original hardware implementation done by Johan Bijker [47], while an additional servoboard was designed to control the airship's actuators. The overall hardware

performance was sufficient for the implementation of the autopilot, but several recommendations regarding improvements are made in Sections 8.2.2 and 8.2.3. Several improvements were made to the original groundstation software, for the purpose of controlling the airship's actuators directly from the groundstation laptop, as well as the arming and disarming of the controllers. The desired flight plan can also be set on the groundstation and the airship's flight can also be monitored on a 2D map of the waypoints.

Simple fuzzy logic controllers were designed to control the speed, heading as well as the height of the airship. The controllers proved to be sufficient in simulation - the heading controller, combined with the guidance strategy, should be able to handle wind disturbances sufficiently and steer the airship between waypoints.

Flight tests show the airship to be laterally underactuated and very susceptible to wind. Although the controllers show adequate performance, all of the controllers require further tuning and testing for improved results.

8.2 Recommendations and Improvements

8.2.1 Airship Model

Linear longitudinal and lateral models can be implemented by employing the system identification techniques derived in [8]. Using these techniques the model can be determined from flight tests.

Although difficult to adapt intuitively, the Gomes model wind tunnel database is still a valuable tool for the development of an airship model using wind tunnel data, and adjustments can be made by incorporating data from an accurate computational fluid dynamics (CFD) simulation of the fins and gondola of airship.

8.2.2 Gondola, Propulsion and Fins

The drag due to the gondola maybe reduced if a more streamlined gondola is designed and fitted as close as possible to the airship's hull. Direct access to batteries and electronics as well as a simpler way of attaching the gondola to the airship would also improve setup speed during flight tests.

A different approach to the design of the gondola could improve the airship's yawing, as the motors for the current gondola are too closely positioned to the z-axis to provide a large enough yawing moment should differential thrust be implemented.

The following changes can be made to the fins:

- Flap sizes could be increased to provide a larger surface area for more lateral actuation.
- Seperate control of all four fins. By controlling the fins separately, aileron type control can be used to damp rolling oscillations.

A mooring mast, similar to that used by the AURORA project as seen in Figure 2.2, would also be helpful during flight tests as the airship can be kept completely still for the initialisation of the sensors.

8.2.3 Hardware

Additional height sensors should be added to the current sensor configuration to ensure accurate height measurements, which will also improve the performance of the height controller.

Several improvements can be made to the original servoboard design. All of the R/C receiver's outputs can be mapped to the rest of the Input Capture pins so that actual instructions from the R/C receiver can also be monitored.

Problems were experienced with the servoboard's control of the elevator during automatic control. This can be attributed to the microcontroller providing 5 Volt logic instead of the 3.3 Volt from the R/C receiver. A voltage drop caused during thrust vectoring causes the power supply of the servo to drop below the voltage of the received PWM signal. To prevent this, the output PWM signals from the PIC should be converted from 5 Volt to 3.3 Volt before being sent to the multiplexers. This will ensure that line losses do not cause the logic signals of the servos to have a higher voltage than the supplied power voltage.

No problems were experienced during manual flight when the servos PWM signals come directly from the R/C receiver.

The RF link used was sufficient for the flight tests done, and an insignificant number of data packets were lost due to transmission errors. Loss of transmission was also experienced when the airship was close to the ground at a distance of a few 100-m. A faster RF link should increase the transmission performance of the current setup. Another way of solving this problem is by storing most of the telemetry onboard the OBC, while only a sample of the data can be sent down to the groundstation for monitoring purposes. A RF module with a much larger range would also be required, as the current setup allows a few 100-m, and requires line of sight between the groundstation and the airship.

A CAN-bus implementation can also be considered, which would simplify the wiring and communication between the sensors, OBC, RF module and the servoboard.

8.2.4 Software

The current OBC development cycle takes very long due to a number of reasons.

- Most of the testing of the code can only be done directly on the OBC.
- The OBC software can currently only be uploaded over a serial cable at a speed of 9600 baud. Faster ways of copying the software to the OBC would require an OBC with extended interfacing protocols such as network connectivity or easily accessible removable storage.

Since the OBC software is programmed using a very old C compiler (Turbo C++) due to the RTOS prerequisites, problems arose when the software was originally coded and tested using the GNU GCC compiler and then ported to the Turbo C++ compiler. Therefore a hardware in the loop setup is recommended as it is easier to test the controller for all the flight conditions with the current hardware setup.

8.2.5 Control System

The control strategy can be improved by using the speed controller during flight to increase and decrease the speed at certain parts of the mission. This could be used to let the airship hang still at a certain position. The height controller should also only be switched on once a certain forward speed is reached.

Path planning is also another interesting research topic. Calculation of time optimal trajectories in which the shortest lateral path is determined for an underactuated airship is discussed in [49]. Wind conditions can also be taken into account when calculating the time optimal flight path as discussed for an UAV in [50]. Trajectory calculation for area coverage will also be useful for future research, as airships' modern day functions consist primarily of earth monitoring.

An investigation into the effect of different membership functions could prove useful in improving the controllers. Further refinements can be made to the number of membership functions used to describe the measurement values, thereby improving the accuracy although the rule base will be more complex.

Different fuzzy logic control structures should also be investigated - by combining more measurements into the decision making process, the performance of the controllers can be improved.

Adaptive fuzzy control should be investigated. Membership functions, as well as the scaling gains can be adjusted during flight to improve the controller performance. Further work would also call for a non-linear stability analysis to be used to evaluate and improve the stability of the fuzzy logic controllers, as discussed in [46].

The following future projects can be considered:

- Hovering Control.
- Controller design through conventional control design techniques.
- Nonlinear controllers as those presented in [37].
- Area mapping controllers, as those implemented by [20].
- Stabilizing and damping controllers to suppress any rolling/pitching oscillations
- Adaptive fuzzy logic control. By adjusting the membership function's positions and widths and the scaling gains, the fuzzy logic controllers performance can be improved upon.
- Artificial neural network control. This controller learns from the skill of the human operator, as discussed for a $12 - m$ long airship in [51].

8.2.6 Flight Tests

In Figure 8.1, the wind probability for Cape Town Airport Area is shown. Flight testing in becomes difficult over the summer months in the Cape Town area, due to strong high probability of strong winds. The ideal months for flight testing would be March, April, August and September. Although

Cape Town Airport (CAPTOWN)

Stats based on observations taken between 11/2000 - 9/2009 daily from 7am to 7pm local time.

Month of year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	SUM
	01	02	03	04	05	06	07	08	09	10	11	12	1-12
Dominant Wind Dir.													
Wind probability > = 4 Beaufort (%)	65	62	52	38	33	29	36	42	50	55	63	71	**49**

Figure 8.1: Wind probability for Cape Town Airport [2]

the winter months have the lowest probability for strong winds, rain fall probability for these months is much higher.

A compact portable weather station would also be ideal for flight tests, since the wind speed and direction can be measured and the effect there of on the controllers can be validated. Onboard wind sensors can also be used to provide additional information for the controllers.

References

[1] Li, Y.: *Dynamics Modeling and Simulation of Flexible Airships*. Ph.D. thesis, McGill University, 2007.

[2] Windfinder: Wind and weather statistics. Website, www.windfinder.com/windstats/, November 2009.

[3] Wikipedia: Airship. Website, en.wikipedia.org/wiki/Airship, October 2009.

[4] Elfes, A., Bueno, S.S., Bergerman, M. and Ramos, J.: Project AURORA: Development of an autonomous unmanned remote monitoring robotic airship. *Journal of the Brazilian Computer Society*, 1998.

[5] Sharma, S., Kulczycki, E. and Elfes, A.: Trajectory generation and path planning for autonomous aerobots. In: *EEE International Conference on Robotics and Automation*. 2007.

[6] Fourie, D.: *Dynamic Modelling and Control System of a tethered aerostat for remote sensing applications*. Master's thesis, University of Stellenbosch, 2009.

[7] Lutz, T., Funk, P., Jakobi, A. and Wagner, S.: Summary of aerodynamic studies on the Lotte airship. In: *4th International Airship Convention and Exhibition*. 2002.

[8] Kornienko, A.: *System Identification Approach for Determining Flight Dynamical Characteristics of an Airship from Flight Data*. Ph.D. thesis, University of Stuttgart, 2006.

[9] Kungl, P., Schlenker, M., Wimmer, D.-A. and Kröplin, B.H.: Instrumentation of remote controlled airship "Lotte" for in-flight measurements. *Aerospace Science and Technology*, 2004.

[10] Wimmer, D., Bildstein, M., Well, K.H., Schlenker, M., Kungl, P. and Kröplin, B.-H.: Research airship "Lotte" development and operation of controllers for autonomous flight phases. In: *International Conference on Intelligent Robots and Systems*. 2002.

[11] Gomes, S.: *An investigation of the flight dynamics of airships with application to the YEZ-2A*. Ph.D. thesis, Cranfield Institute of Technology, 1990.

[12] Goineau, F. and Cook, M.: The stability and control characteristics of the neutrally bouyant non-rigid airship. Tech. Rep., Cranfield University, 1999.

[13] Ramos, J., de Paiva, E.C., Azinheira, J.R., Bueno, S., Maeta, S.M., Mirisola, L.G.B., Bergerman, M. and Faria, B.G.: Autonomous flight experiment with a robotic unmanned airship. In: *IEEE International Conference on Robotics & Automation*. 2001.

[14] Ramos, J.J.G., de Paiva, E.C., Azinheira, J.R., Bueno, S.S., Bergerman, M., Ferreira, P.A.V. and Carvalho, J.R.H.: Lateral/directional control for an autonomous unmanned airship. *Aircraft Engineering and Aerospace Technology*, vol. 73, pp. 453–458, 2001.

[15] Azinheira, J.R., de Paivab, E.C., Ramosb, J.J.G. and Buenob, S.S.: Hovering control of an autonomous unmanned airship. In: *Controlo'2000: 4th Portuguese Conference on Automatic Control*. 2000.

[16] Azinheira, J.R., Moutinho, A. and de Paiva, E.C.: Airship hover stabilization using a backstepping control approach. *Journal of Guidacne, Control and Dynamics*, vol. 29, 2006.

[17] Moutinho, A.B.: *Modeling and nonlinear control for airship autonomous flight*. Ph.D. thesis, Instituto Superior Tecnico, Technical University of Lisbon., 2007.

[18] Ramos, J.J.G., Maeta, S.M., Bergerman, M., Bueno, S.S., Mirisola, L.G.B. and Bruciapagliab, A.: Development of a VRML/Java unmanned airship simulating environment. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.

[19] Elfes, A., Bergerman, N., Carvalho, J., de Paiva abd J.J.G. Ramaos, E.C. and Bueno, S.: Air-ground robotic ensembles for cooperative applications : Concepts and preliminary results. *2nd International Conference on Field and service Robotics*, pp. pp 75–80, 1999.

[20] Hygounenc, E., Jung, I.-K., Soueres, P. and Lacroix, S.: The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *Experimental Robotics VIII*, 2003.

[21] Bonnet, A.: Identification des coefficients arodynamiques du dirigeable as500 du laas. Tech. Rep., Etude Hydro-Arodynamique, 2003.

[22] Munk, M.M.: Notes on aerodynamic forces 3: the aerodynamic forces on airships. Technical report 106. Tech. Rep., Naca Technical Notes, 1922.

[23] Bryson, A.: Evaluation of the inertia coefficients of the cross section of slender body. *Journal of the Aeronautical Sciences*, vol. 21, pp. 424–427, 1954.

[24] Elfes, A., Montgomery, J.F., Hall, J.L., Joshi, S.S., Payne, J. and Bergh, C.F.: Autonomous flight control for a Titan exploration aerobot. In: *ISAIRAS*. 2005.

[25] Kulczycki, E.A., Joshi, S.S. and Hess, R.A.: Towards controller design for autonomous airships using SLC and LQR methods. *AIAA Guidance Navigation and Control Conference and Exhibit*, 2006.

[26] Payne, J. and Joshi., S.S.: 6 Degree-of-freedom non-linear robotic airship model for autonomous control. Tech. Rep., University of California, Davis, 2004.

[27] Payne, J.: *6-DOF Dynamics Airship Model in ADAMS*. Master's thesis, University of California, Davis, 2005.

[28] Lamb, H.: The inertia-coefficients of an ellipsoid moving in a fluid. Reports and memoranda, no. 623, NACA, 1918.

[29] Nielsen, J.: *Missile Aerodynamics.* Mcgraw-Hill, 1960.

[30] Traiantafyllou, M. and Hover, F.: Maneuvering and control of marine vehicles. Tech. Rep., Department of Ocean Engineering, 2002.

[31] Hopkins, E.J.: A semiemperical method for calculating the pitching moment of bodies of revolution at low mach numbers. Tech. Rep., NACA, 1951.

[32] Freeman, H.B.: Pressure-distribution measurements on the hull and fins of a 1/40 scale model of the U.S. airship "Akron". Tech. Rep., NACA, 1931.

[33] Khoury, G. and Gillet, D. (eds.): *Airship Technology.* Cambridge, 1999.

[34] McCormick, B.: *Aerodynamics, Aeronautics and Flight Mechanics.* 2nd edn. John Wiley and Sons, 1995.

[35] LaBudde, E.: Extending the burrowman method for large angles of attack, 1999. R&D Project, NARCON.

[36] Azinheira, J., de Paiva, E.C. and Bueno, S.S.: Influence of wind speed on airship dynamics [erratum]. *Journal of Guidance, Control and Dynamics*, vol. 31, 2008.

[37] Mouintho, A.: *Modeling and nonlinear control for airship autonomous flight.* Ph.D. thesis, Technical University of Lisbon, 2007.

[38] McLean, D.: *Automatic Flight Control Systems.* Prentice Hall, 1990.

[39] Stevens, B. and Lewis, F.: *Aircraft Control and Simulation.* Wiley, 2003.

[40] Doitsidis, L., Valavanis, K., Tsourveloudis, N. and Kontitsis, M.: A framework for fuzzy logic based UAV navigation and control. *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 4041–4046, 2004.

[41] Kurnaz, S., Cetin, O. and Okyay Kaynak, O.: Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. *Intelligent Robot Systems*, 2009.

[42] Jantzen, J.: Design of fuzzy controllers. Tech. Rep., Technical University of Denmark, 1998.

[43] Gopal, M.: *Digital Control and State Variable methods.* McGraw Hill, 2003.

[44] Jantzen, J.: *A robustness study of fuzzy control rules.* Technical University of Denmark.

[45] Hill, ., Horstkotte, G. and Teichrow, J.: *Fuzzy-C Development System User's Manual.* Togai Infralogic, Inc, 2nd edn, 1989.

[46] Galluzo, M., Cosenza, B. and Aronadio, G.: Stability analysis of type-2 fuzzy logic controllers. Available at: www.aidic.it/icheap9/webpapers/237Galluzzo.pdf

[47] Bijker, J.: *Development of an Attitude Heading Reference System for an Airship.* Master's thesis, Stellenbosch University, 2006.

[48] KADAK: AMX: AMX 86 user guide. March 2005.

[49] Hima, S. and Bestaoui, Y.: Time-optimal paths for lateral navigation of an autonomous underactuated airship. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.

[50] McNeely, R.L. and Iyer, R.V.: Tour planning for an unmanned air vehicle under wind conditions. *Journal of Guidance, Control and Dynamics*, vol. Vol. 30, No. 5, 2007.

[51] Rao, J., Gong, Z., Luo, J., Jiang, Z., Xie, S. and Liu, W.: Robotic airship mission path-following control based on ann and human operator's skill. *Transactions of the Institute of Measurement and Control*, vol. 29, 1, pp. pp. 5–15, 2007.

[52] Explanation of navigation tables. Website, National Imagery and Mapping Agency (US), 2002. Available at: www.pollux.nss.nima.mil/NAV_PUBS/APN/Tables/TblExpl.pdf

Appendix A

Airship Specifications

A.1 General Information

Volume	$14.2m3$
Maximum Diameter	$1.9m$
Length	$8m$
Fineness Ration	4.2
Hull mass	$5.7kg$
Gondola mass	$5.2kg$
Fin mass	$1.4kg$
Rudder Deflection Range	-27 to 16°
Elevator Deflection Range	-16 to 27 °
Thrust Vector Range	-90 to 180 °
Thrust	0 to 22 N

A.2 Calculation of Moments of Inertias

The moment of inertia calculation for each part of the airship can be calculated individually and then added together. The coefficient used in the equations for the modelling of the inertia of the airship are:

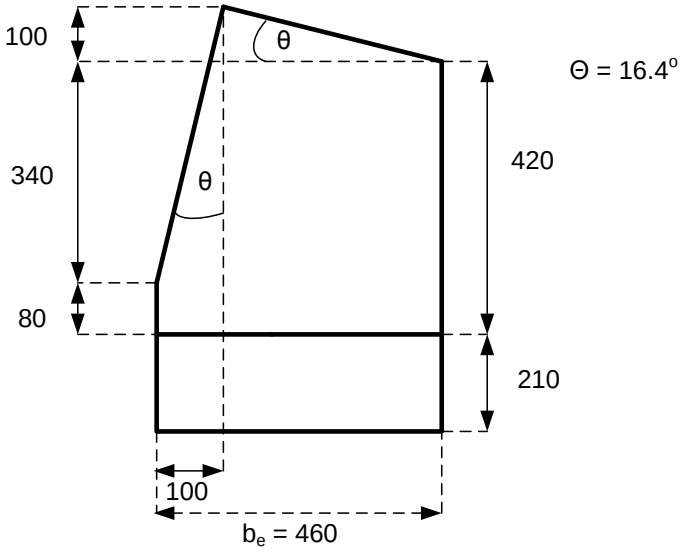


Figure A.1: FZ-800E Fin Area

Coefficient	Value	Description
m_{hull}	5.7	Mass of the hull [kg]
m_{fin}	1.4	Mass of the fins [kg]
m_{gon}	5.4	Mass of the gondola [kg]
r_a	4	Radius of the semi-major axis of the hull[m]
r_b	0.95	Radius of the semi-minor axes of the hull[m]
r_g	0.5	Rough estimate for gondola cube approximation lengths[m]
r_{gx}, r_{gy}, r_{gz}	1.15, 1.25, 0.5	Radius from the gondola to the x,y,z rotation axes respectively[m]
r_{fx}, r_{fy}	0.36,3.75	Distance from x-axis and y-axis to the fins[m]
l_w, l_l	0.46, 0.63	Width and Length of the fins[m]

A.2.1 Helium

The inertia of the gas inside the envelope can be calculated by using the moment of inertia for a solid sphere with the specified volume V and density ρ.

$$J_{xx} = \frac{1}{5}\rho V(2r_b^2) \tag{A.1}$$

$$J_{yy} = \frac{1}{5}\rho V(r_a^2 + r_b^2) \tag{A.2}$$

$$J_{zz} = \frac{1}{5}\rho V(r_a^2 + r_b^2) \tag{A.3}$$

A.2.2 Hull

The hull was modeled as a ellipsoidal shell, which yields:

$$J_{xx} = \frac{1}{3} m_{hull}(2r_b^2) \tag{A.4}$$

$$J_{yy} = \frac{1}{3} m_{hull}(r_a^2 + r_b^2) \tag{A.5}$$

$$J_{zz} = \frac{1}{3} m_{hull}(r_a^2 + r_b^2) \tag{A.6}$$

A.2.3 Fins

The fins were modeled as rectangular plates.

The J_{xx} equation,

$$J_{xx} = 4(\frac{1}{12} m_{fin} l_w^2 + m_{fin} r_{fx}^2) \tag{A.7}$$

and consists of the fins being modeled as a rectangular plate with the axis passing through the plane of the fin. This is then shifted using the parallel axis theorem.

The equations for J_{yy} and J_{zz} consists of 2 fins modeled as rectangular plates with the axis passing through the plane of the fin, as well as 2 fins modeled as rectangular plates with the axis of ration through the center of the plate. The terms due to the shift using the parallel axis theorem is also added, therefore

$$J_{yy} = 2(\frac{1}{12} m_{fin} l_l^2 + m_{fin} r_{fy}^2 + \frac{1}{12} m_{fin}(l_w^2 + l_l^2) + m_{fin}(r_{fy}^2 + r_{fx}^2)) \tag{A.8}$$

$$J_{zz} = Jyy \tag{A.9}$$

A.2.4 Gondola

A rough estimate was made by approximating the gondola as a cube, since the gondola's inertia is more complex do the motors, and added electronics. This inertia is also shifted using the parallel axis theorem.

$$J_{xx} = \frac{1}{6} m_{gon} r_g^2 + m_{gon} r_{gx}^2 \tag{A.10}$$

$$J_{yy} = \frac{1}{6} m_{gon} r_g^2 + m_{gon} r_{gy}^2 \tag{A.11}$$

$$J_{zz} = \frac{1}{6} m_{gon} r_g^2 + m_{gon} r_{gz}^2 \tag{A.12}$$

Appendix B

Servoboard Design

B.1 Component List

Description	Designator	Quantity	Value
Microchip Microcontroller	dsPIC1	1	30F5011
10A, 5V switching regulator	U3	1	
3.3V regulator, 200ma	U6	1	
Dual EIA-232 Driver/Receiver	U5	1	MAX232D
Header for 8 servos	SERVOOUT1	1	
Header for 9 servos	SERVOIN1	1	
Header for ICD2 programmer, 6-pin	J5	1	
Header, 2-Pin	J1	1	
Header, 3-Pin	J6	1	
Low-Power Operational Amplifiers	U4	1	
Schottky Diode	D1, D2	2	10BQ015
Single bit level translater, dual bus	U7	1	
Terminal Block	J2, J4	2	
TI Multiplexer dual 2:1	MUX1, MUX2, MUX3, MUX4	4	TS5A23159
Tone Detector	1	1	LM567C
Typical GaAs LED	DS1, DS2, DS3	3	
Resistor	R7	1	1k
Capacitor	C20	1	1uF
Resistor	R1, R2, R3	3	2.2K
Capacitor	C36	1	2.2uF
Capacitor	C19	1	5.6nF
Crystal Oscillator	Y1	1	7.3728MHz
Resistor	R8	1	10K
Capacitor	C21, C22	2	10nF
Capacitor	C17	1	22nF

Electrolytic Capacitor	C3	1	22uF
Capacitor	C2	1	22uF
Capacitor	C23, C24	2	27pF
Resistor	R9, R10, R11	3	33
Capacitor	C25, C26	2	33pF
Capacitor	C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C27, C28, C29, C30, C31, C32, C33, C34, C35	21	100nF
Capacitor	C16	1	220uF
Resistor	R4, R5	2	270
Capacitor	C18	1	330uF
Resistor	R6	1	470

B.2 Schematics

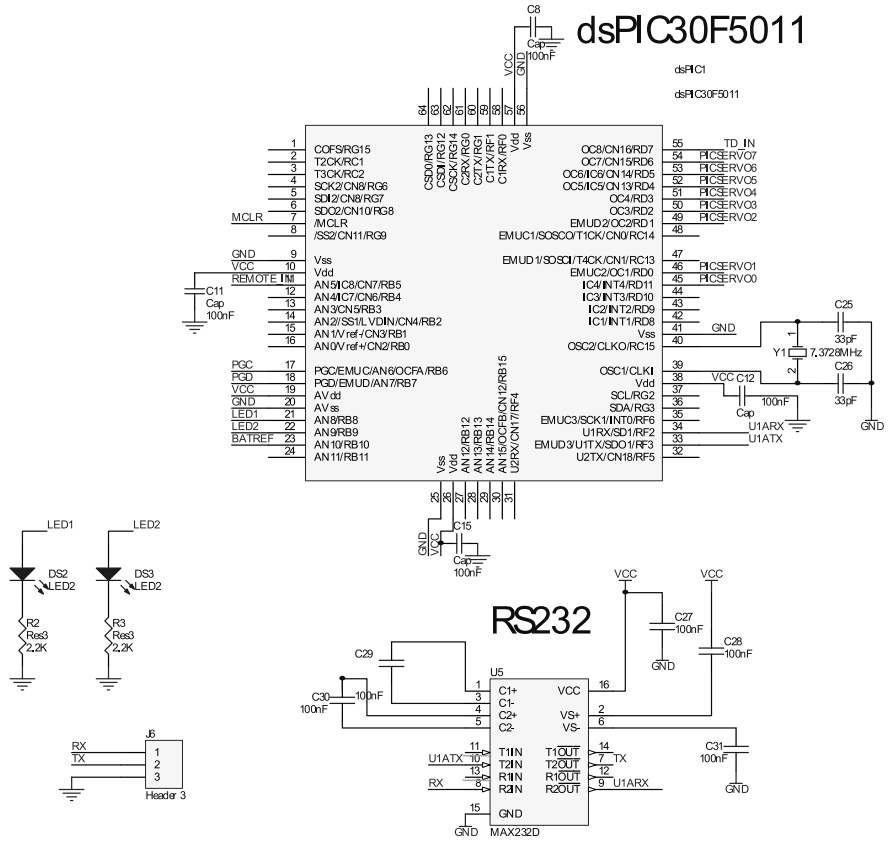


Figure B.1: dsPIC30f5011 and RS232

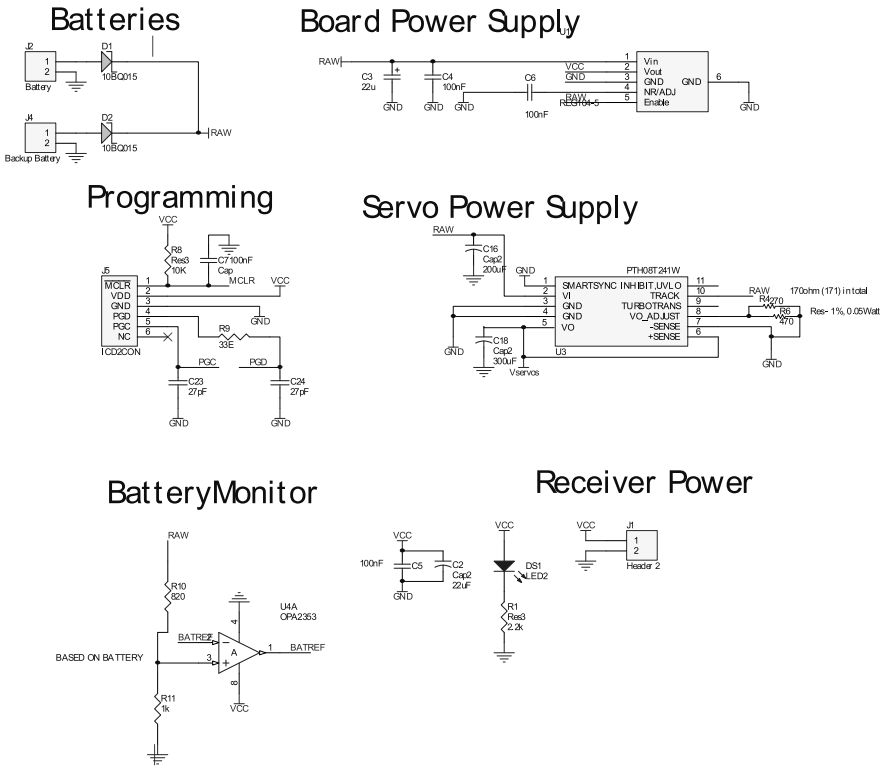


Figure B.2: Power Supply of the Servoboard and Servos

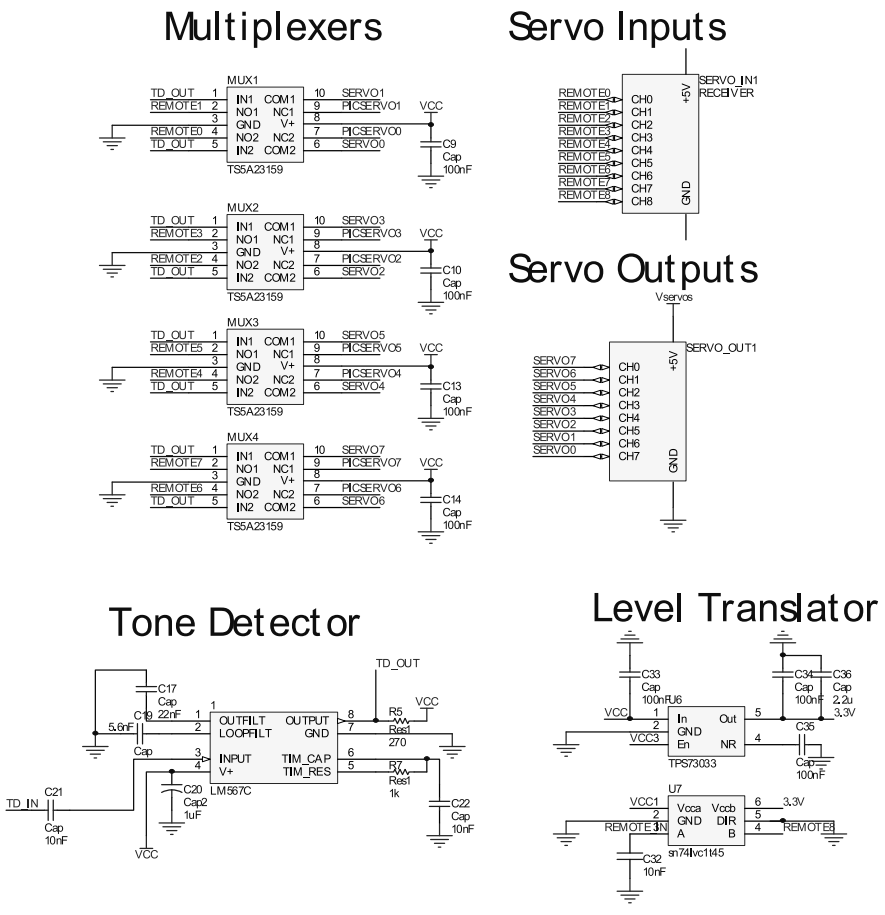


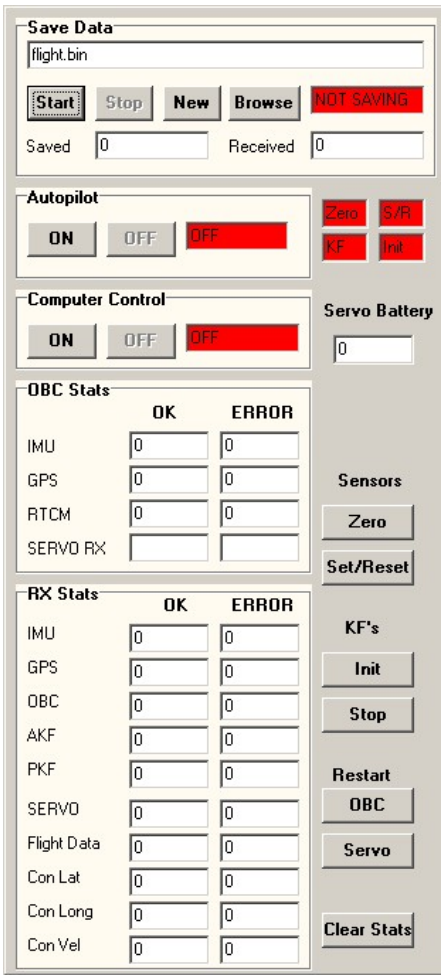
Figure B.3: Multiplexers, tone detector, and output to servos

Appendix C

Groundstation

C.1 Introduction

The groundstation software was designed so that the most important features are visible at all time as shown in Figure C.1. These include the buttons necessary for the control of the data logging, autopilot and computer control, as well as the control of most import hardware and software functions.



The main window is separated in several tabbed windows and are as follows:

C.2 Sensors and Esimations

Consider Figure C.1. This tab shows the actual readings from all the sensors: the gyrometers, accelerometers, magnetometers as well as the GPS.

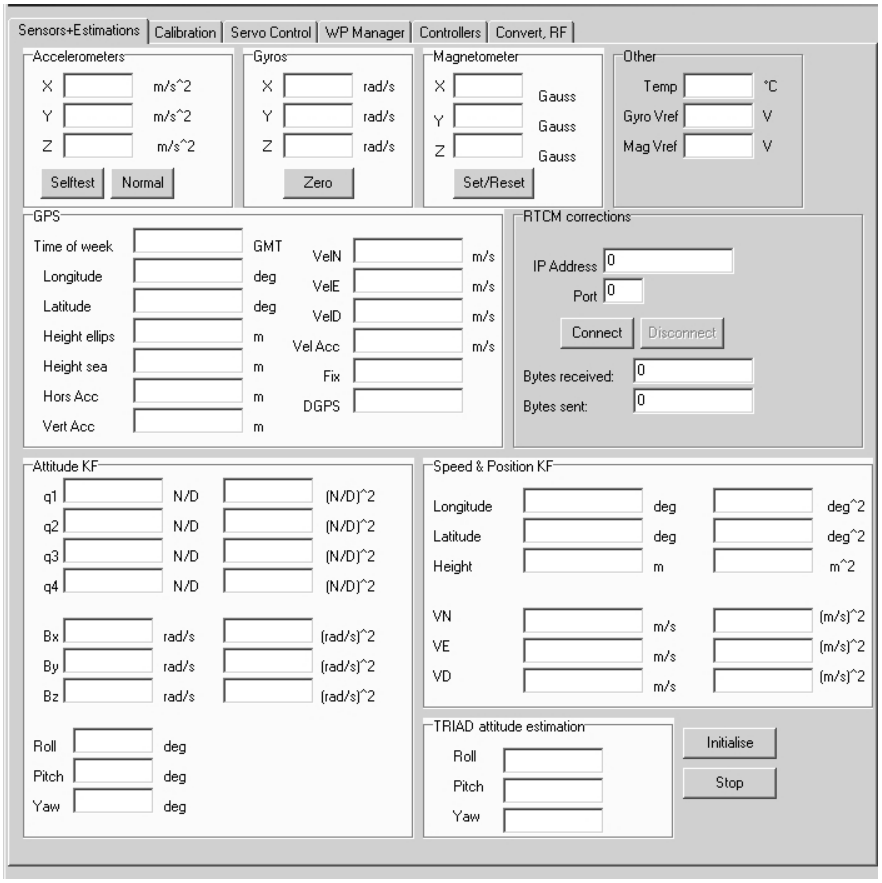


Figure C.1: Groundstation: Sensors

C.3 Calibration

Consider Figure C.2. Contains the calibration data required by the sensors on the IMU. If necessary, this calibration data can be changed on the ground and uploaded to the airship.

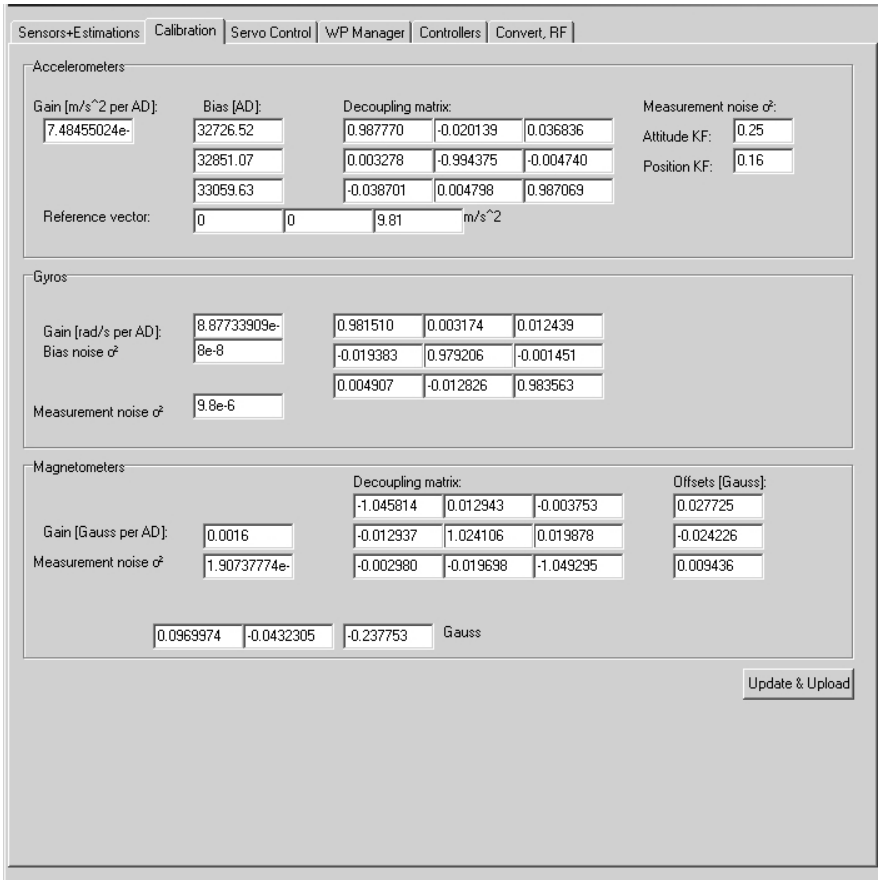


Figure C.2: Groundstation: Calibration

C.4 Servo Control

Consider Figure C.3. When computer control is active, the actuators can be controlled directly from the groundstation, for example, the rudder can be controlled by specifying a specific deflection angle. A step command can also be issued, and is useful for system identification processes. The deflection ranges and the corresponding servo can be changed for each airship actuator.

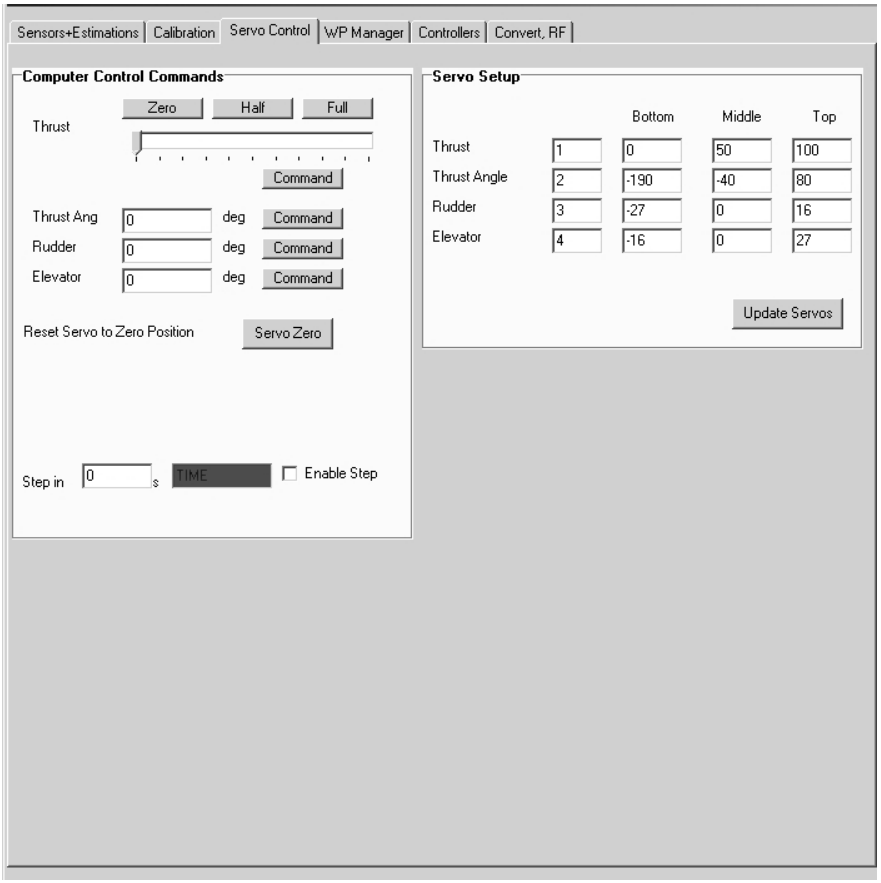


Figure C.3: Grounstation: Servo Control

C.5 Waypoint Manager

Consider Figure C.4. The flight path of the airship can be determined on this screen. The waypoints are specified in NED notation and stored using a linked list. Waypoints can be added and removed to and from the current flight path, and the airships progress can be followed as it transgresses between the waypoints.

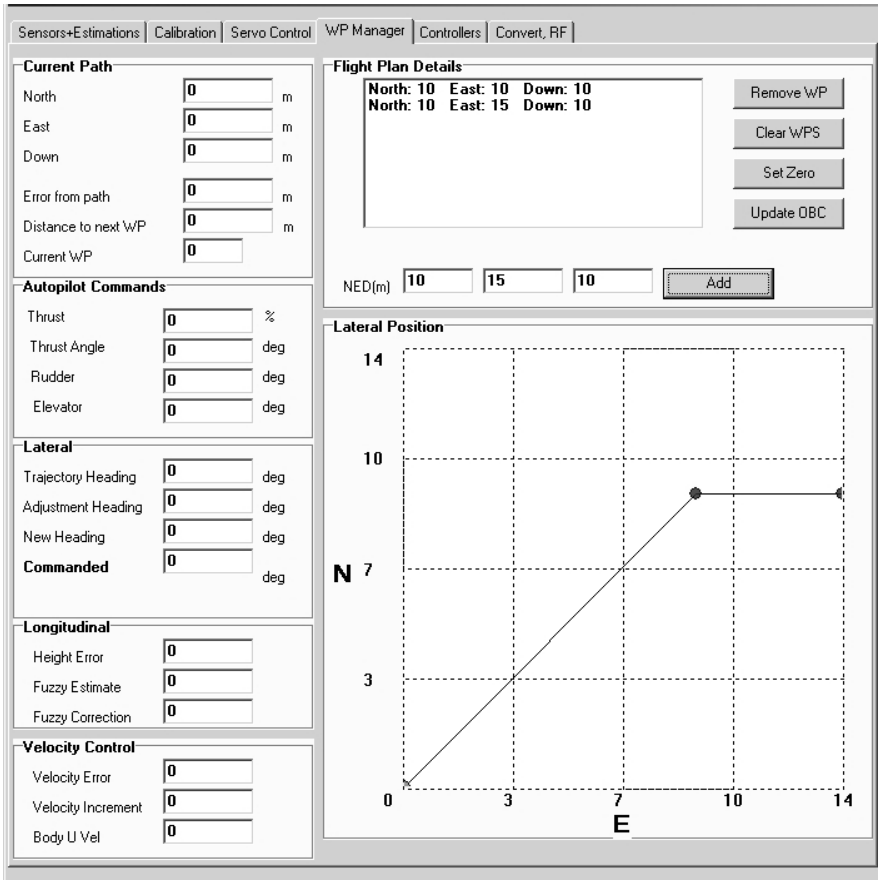


Figure C.4: Groundstation: Waypoint Manager

C.6 Controller Setup

Consider Figure C.5. The controllers that are used is set on this screen. Simple step commands can also be given to individual controllers.

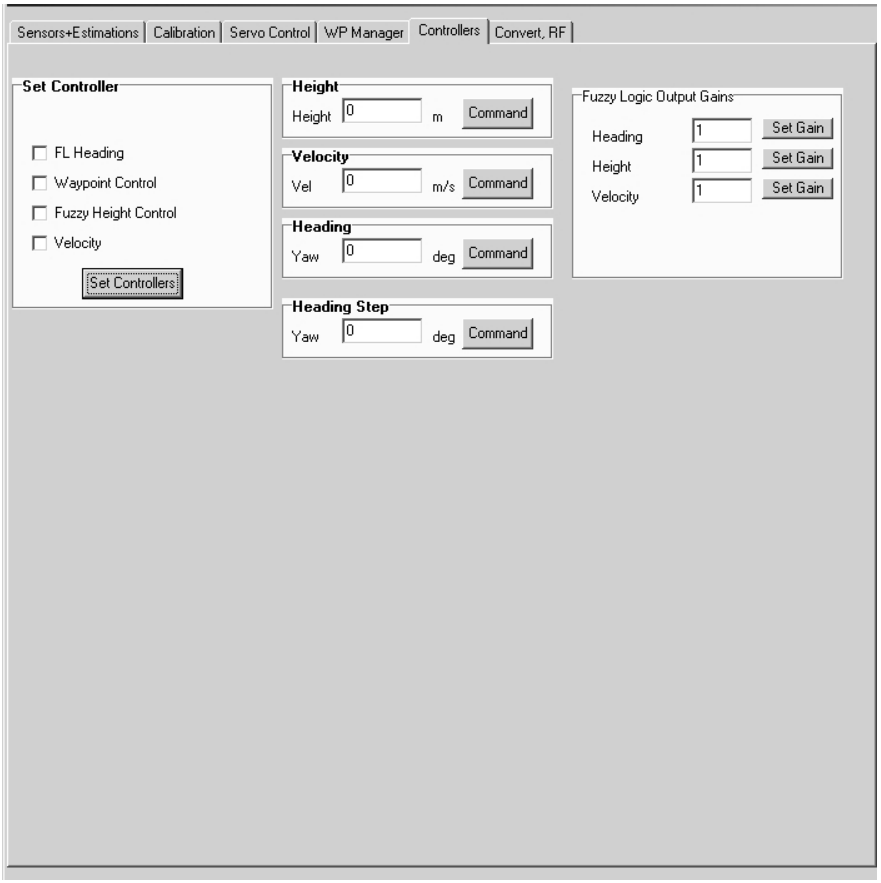


Figure C.5: Groundstation: Controller Setup

C.7 Convert

Consider Figure C.6. The stored telemetry from the airship can be converted to several comma separated value files which can easily be used for further processing.

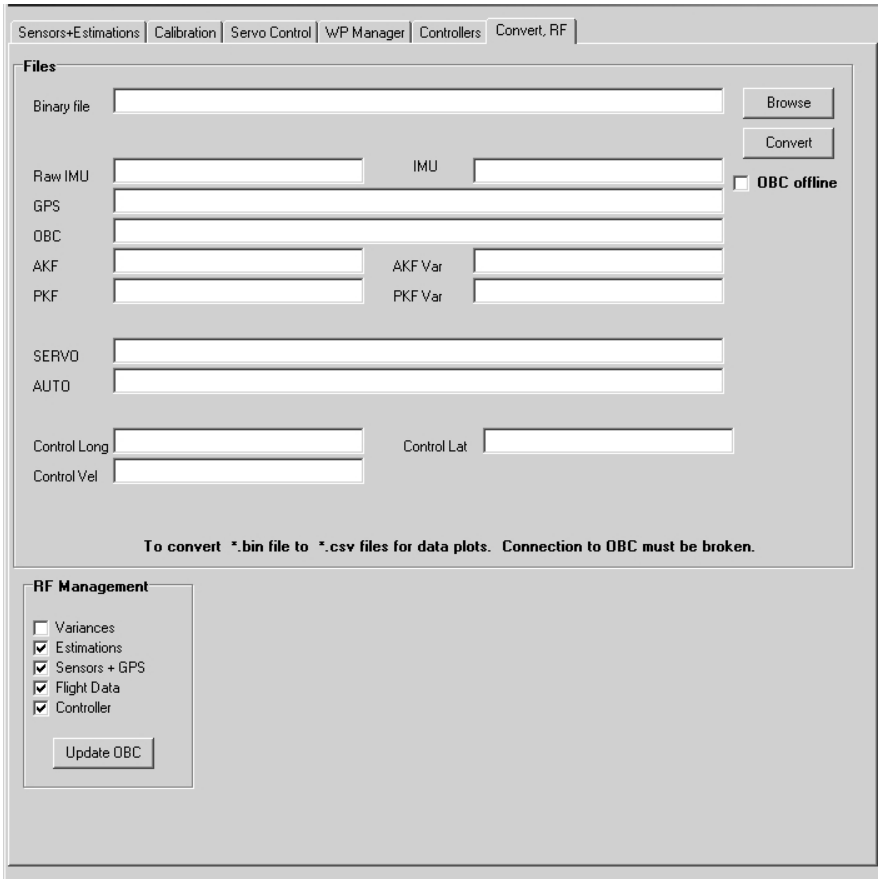


Figure C.6: Groundstation: Convert

Appendix D

Fuzzy Logic implementation in C

D.1 Introduction

This implementation was done with using the framework found in Section 12.4 in [43]

D.2 Fuzzifying

Fuzzifying the input measurements were made simple by working with triangular membership functions. Each measurement/input is fuzzified using the following steps:

1. First locate the fuzzy variables to which the input would belong. This is done by searching through the fuzzy variables to determine in which ranges the input fall.
2. Determine the degree of membership by using the appropriate membership functions. As triangle membership functions were used, the membership can simply be determined by using the equation for a straight line.
3. Store the membership in an array the size of the number of fuzzy variables. The index should correspond to that of the fuzzy variable.

D.3 Rule Base and Inference method

The rulebase was implemented in the 2D-array form. The total number of rules is simply the number of fuzzy variables for one input times the number of fuzzy variables for the other output. The indexes of the fuzzyified inputs correspond to the indexes of rules that fired. The value of each element corresponds with the index of the specific output membership function.

The inference algorithm is as follows:

1. Create an array for the whole output area, and initialize to zero. The size is determined as the range between the first value of the first fuzzy output variable and the last value of the last fuzzy output variable.

2. For each rule that has fired:

   (a) Calculate the firing strength as the minimum of the 2 corresponding fuzzy values.

   (b) Lookup the corresponding output fuzzy variable.

   (c) For all the elements that are members of the output fuzzy variable

   - If the membership value is smaller than the firing strength, set the corresponding elements in the output array to the membership value.
   - If the membership value is larger than the firing strength, set the corresponding elements in the output array to the value of the firing strength.
   - The value in the output array is only changed when the new valueis not larger than the current value.

D.4 Defuzzifying

The defuzzyifying method used is the centre of area method discussed in Section 5.2.4. It was implemented in its discrete form since the output area created by the inference method is represented in an array data structure.

The position of each element is its index added to the offset provided by the first output membership function value.

Appendix E

Additional Simplifications

E.1 GPS to NED coordinates

The controller was designed in a NED coordinate system, instead of the longitude and latitude manner of the GPS Data. A simplified as well as more computationally efficient way of converting from GPS coordinates to NED coordinates is by calculating the change in meters per degree longitude and latitude [52].

$$\frac{meters}{degree} latitude = l^o_{lat} = 111132.92 - 559.82\cos(2lat) + 1.175\cos(4lat) - 0.0023\cos(6lat) \quad \text{(E.1)}$$

$$\frac{meters}{degree} longitude = l^o_{lat} = 111412.84\cos(lat) - 93.5\cos(3lat) + 0.118\cos(5lat) \quad \text{(E.2)}$$

$$\text{(E.3)}$$

For Stellenbosch ($33^o55'12''$S, $18^o51'36''$E) these values approximate to:

$$l^o_{lat} = 110.92km \quad \text{(E.4)}$$

$$l^o_{long} = 92.47km \quad \text{(E.5)}$$

$$\text{(E.6)}$$

The north (N) and east (E) position relative to a specified zero coordinate(lat_0, $long_0$) can be calculated from

$$N = (lat - lat_0).l^o_{lat} \quad \text{(E.7)}$$

$$E = (long - long_0).l^o_{long} \quad \text{(E.8)}$$