CONTROL DESIGN AND VALIDATION FOR AN UNMANNED, FINLESS AIRSHIP

Dipl.-Ing. Torsten Liesk

Doctor of Philosophy

Department of Mechanical Engineering

McGill University

Montreal,Quebec

2012-04-13

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of doctor of philosophy

©Torsten Liesk, 2012

ACKNOWLEDGEMENTS

The author would like to thank the following people and institutions without whom this research would not have been possible.

First of all, I would like to thank the German Academic Exchange Service DAAD for financing the first year of my research and my supervisor, Prof. Meyer Nahon, for financing the remaining years.

I would also like to thank both my supervisors, Prof. Meyer Nahon and Prof. Benoit Boulet for their support and constructive criticism for this work. I am equally thankful for the help of the Master's students with whom I worked on this project, Prashant Peddiraju and Hashim Mazhar. This also extends to the numerous summer students who helped to improve the testing hardware over the years: Gaby Waldhart, Henna Warma, Sylvain Joassard, Maja Sevaldson, Francis de Broulx, and Ryan Caverly.

Additionally, I want to acknowledge the technical support of Quanser Inc., in particular of Dr. Ernest Earon and Cameron Fulford. The design of the airship, as described in the present work, is wholly Quanser's.

Last but not least, I would of course like to thank my family, especially my wife Giuliana and my daughter Lara, for their support and for making sure that I maintain a healthy work-life balance.

ABSTRACT

There is a resurgent interest in the use of airships for long-endurance and heavylift operations, as they provide an energy-efficient means of transportation for missions in which speed is not critical. Autonomous unmanned airships that are capable of flying a predefined path without human interaction can be used for a variety of missions, from wildlife monitoring to civil security tasks. This work investigates the control aspects of a novel, finless airship design with the goal of autonomous operation.

In a first step, the dynamics of a finless airship are analysed. The conservative forces acting on the airship hull are derived via a Newton-Euler approach and compared with the literature. The non-conservative forces applicable to the vehicle studied here are taken from the literature to complete the equations of motion. A simulation of the airship dynamics including a detailed sensor noise and actuation model is built as a basis for the controller design.

Two different control strategies, linear H_{∞} control and non-linear Backstepping control, are investigated for the stabilization of the attitude and velocity of a finless airship. After achieving satisfactory results in the simulation, the controller based on the nonlinear Backstepping technique is also used in flight tests, exhibiting good control performance.

To enable trajectory tracking, a separate high-level controller is developed that

allows trajectory tracking and hover with a single control law. The high-level controller uses the nonlinear low-level controller for airship stabilization. The performance of this controller is first verified in simulation. Subsequent flight testing produced similar results as the simulation showing that the controller suite is robust to effects not modelled in the simulation.

The last aspect of this work is the development of a wind estimation algorithm. This algorithm uses the observed airship motion to calculate an estimate of the wind speed without the need for additional sensors, such as airspeed sensors. Having wind information available allows improvement of controller performance, as the airship dynamical response to control inputs can be predicted more precisely.

ABRÉGÉ

Depuis quelques années, il y a un renouvellement de l'intérêt pour l'utilisation des dirigeables pour des missions de longue durée ou pour les opérations avec des poids lourds. Les dirigeables offrent un moyen de transport efficace pour des missions pour lesquelles la vitesse n'est pas essentielle. Des dirigeables non pilotés et autonomes qui sont capables de suivre un trajet prédéfini sans aucune intéraction humaine peuvent être utilisés pour des missions diverses, comme par exemple l'observation des animaux sauvages ou des tâches concernant la sécurité civile. Cette thèse recherche les aspects de contrôle pour un nouveaux concept de dirigeable sans stabilisateurs avec le but d'une opération autonome.

Comme première étape, les dynamiques d'un dirigeable sans stabilisateurs sont analysées. Les forces conservatrices qui agissent sur l'enveloppe du dirigeable sont dérivées en utilisant la méthode de Newton-Euler et elles sont comparées avec la littérature. Les forces non-conservatrices applicables au véhicule utilisé dans cette thèse sont pris de la littérature pour compléter les équations dynamiques. Une simulation de la dynamique du dirigeable est développée comme base pour le développement des contrôleurs. Cette simulation inclut des modèles detaillés pour le bruit des capteurs et les dynamiques des actuateurs.

Deux stratégies de contrôle différentes sont investigées pour la stabilisation de l'assiette et la vitesse du dirigeable: la technique de contrôle linéaire H_{∞} et la technique de contrôle non-linéaire backstepping. Après avoir obtenu des résultats satisfaisants dans les simulations, le contrôleur utilisant la technique backstepping est aussi vérifié en essais en vol, démontrant une bonne performance.

Pour être capable de suivre un trajectoire, un contrôleur supérieur qui est capable de suivre un trajectoire et de maintenir une position fixe avec une seule loi de contrôle est développé. Le contrôleur supérieur se sert du contrôleur backstepping pour la stabilisation du dirigeable. D'abord, la performance de ce contrôleur est vérifiée en simulation. Ensuite, les essais en vol donnent des résultats similaires à la simulation, démontrant que les contrôleurs sont robustes pour les effets non-simulés.

Le dernier aspect de cette thèse est le développement d'un algorithme pour l'estimation du vent. Cet algorithme utilise le mouvement observé du dirigeable pour déterminer un estimé du vent sans avoir besoin de capteurs additionnels. Connaître le vent permettra d'améliorer la performance du contrôleur parce que la réponse dynamique du dirigeable aux changements de l'assiette peut être mieux prévue.

TABLE OF CONTENTS

ACK	NOWI	LEDGEMENTS	ii
ABS	TRAC	Τ	iii
ABR	ÉGÉ		v
LIST	OF T	ABLES	х
LIST	OF F	IGURES	xi
NON	1ENCI	ATURE	xx
1	Introd	uction	1
	1.1 1.2 1.3	MotivationLiterature review1.2.1Airship dynamic modelling1.2.2Airship control using linear control algorithms1.2.3Airship control using nonlinear control algorithms1.2.4Path tracking control1.2.5Wind estimation algorithmsDissertation contribution and content overview	$ \begin{array}{c} 1 \\ 6 \\ 8 \\ 10 \\ 14 \\ 17 \\ 18 \\ \end{array} $
2	Airshi 2.1 2.2 2.3 2.4	p dynamics simulationReference frames and kinematic couplingDerivation of the kinematic and added mass forces2.2.1Translational equations of motion2.2.2Rotational equations of motion2.2.3Effect of the Munk moment2.2.4Conservation of energySimulation setupInfluence of the airship heaviness on the attitude control authority	 22 25 29 30 34 37 41 48

3	Linear	low-level controller design	51
	3.1 3.2	Linearization of equations of motionLinear controller design $3.2.1 H_{\infty}$ controller synthesis $3.2.2 \text{Thrust allocation algorithm}$	52 58 59 65
	3.3	Closed-loop simulations	68 70 74 79
4	Nonlir	near low-level controller design	81
	4.1	Design of an integral backstepping/Lyapunov low-level controller .4.1.1 Integral backstepping controller design	81 83
	4.2 4.3	sentation	88 89 91 93 103 105
		4.3.2 Flight test results	106
5	High-l	evel controller design	117
	5.1	High-level control scheme	117
	$5.2 \\ 5.3$	Simulation based performance evaluation	120 126 137
6	Dynar	nics based wind estimation	144
	$6.1 \\ 6.2 \\ 6.3 \\ 6.4$	Wind knowledge impact on low-level controller performanceWind estimation algorithm	144 149 154 162

7	Concl	usion $\ldots \ldots 170$
	7.17.27.3	Future work1737.1.1Low-level control using H_{∞} control1737.1.2Nonlinear low-level controller design1737.1.3Path tracking controller1747.1.4Wind and parameter estimation175Transfer to other vehicles175Vehicle modification recommendations176
А	Freque	ently used mathematical relations
	A.1	Coordinate transformations
В	Extern	nal forces and moments acting on the airship
	B.1 B.2 B.3	Gravity and buoyancy forces
С	Contro	ol error weighting function $\boldsymbol{W}_{e}(s)$
D	Sensor	dynamics model
	D.1 D.2	IMU noise modelling
Е	Jacob	ian matrices of the linearized equations of motion
	E.1 E.2 E.3	Entries of the state Jacobian matrix201Entries of the wind disturbance Jacobian matrix207Subderivatives212
Refe	rences	

LIST OF TABLES

able	I	page
2–1	Airship properties and initial conditions used in the simulations	34
3–1	State values for the equilibrium points used in H_{∞} controller design $~$.	55
3–2	Minimum γ values achieved during H_{∞} controller synthesis $\ldots \ldots$	64
3–3	Nominal airship parameters and parameters used in simulation	69
4–1	Controller and thrust allocation parameters used in the simulations	94
4-2	Controller and thrust allocation parameters after tuning for optimal flight test performance.	111
5 - 1	Times t at which the individual waypoints are sequenced	127
5 - 2	High-level controller parameters for the simulations presented	127
5 - 3	Root mean square position error for the simulations with low wind	131
5–4	Root mean square position error for the simulations with high wind. $% \left({{{\bf{n}}_{{\rm{s}}}}_{{\rm{s}}}} \right)$.	132
6–1	Propulsion forces required to achieve equilibrium for different wind conditions.	150
6–2	Root mean square position error for the trajectory tracking with wind estimation.	162
B–1	Airship properties as determined in [1]	182
B-2	Thruster location as given in [1]. \ldots \ldots \ldots \ldots \ldots \ldots	186
D-1	Average values during stationary IMU test	190

Table

LIST OF FIGURES

Figure		page
1-1	The Zeppelin NT	2
1-2	The Cargolifter CL 160	2
1–3	The Boeing Skyhook	5
1-4	The ALTAV Quanser MkII	5
1 - 5	The Draganflyer X4 quadrocopter	5
2-1	Reference frames used for the equations of motion $\ldots \ldots \ldots \ldots$	23
2-2	Airship pitch evolution during the 3 Simulations with different wind conditions.	36
2-3	Airship roll evolution during the 3 Simulations with different wind conditions.	36
2-4	Evolution of the airship velocity northwards during the 3 Simulations with different wind conditions	36
2-5	Evolution of the airship velocity downwards during the 3 Simulations with different wind conditions	36
2-6	Airship attitude evolution during the tumbling motion simulation	38
2-7	Evolution of the airship kinetic energy T_{ref} and T_{new} during the tumbling motion simulation.	38
2-8	Architecture of the airship dynamics simulation	42
2-9	Airship attitude evolution during open loop simulation	44
2-10	Body-frame velocity evolution during open loop simulation	44

2–11	Euler angle evolution during the simulation with PID closed-loop control.	4'
2-12	Velocity evolution during the simulation with PID closed-loop control.	4'
2–13	Lateral velocity evolution during the simulation with PID closed-oop control.	42
2-14	Thrust generated for stationary level flight	49
2–15	Limitation of the attitude control authority for different airship heavinesses.	5
3–1	Overall control architecture.	5
3-2	State Jacobian matrix \boldsymbol{J}_x Eigenvalues	5'
3–3	Comparison of the attitude evolution in the linear and the nonlinear model	52
3–4	Closed loop system architecture with the H_{∞} attitude controller (Control architecture 1)	59
3–5	Closed loop system architecture with the H_{∞} angular rates controller (Control architecture 2)	6
3–6	Lower fractional transformation used for H_{∞} controller design	6
3–7	Constraints of the forces $F_{x,i}$ and $F_{z,i}$ for the <i>i</i> -th thruster	68
3–8	Airship reference path during the simulations.	70
3–9	Reference height profile during the simulations.	70
3–10	Roll and pitch evolution during simulation with ideal conditions using control architecture 1.	7:
3–11	Yaw evolution during simulation with ideal conditions using control architecture 1	7
3–12	Roll and pitch evolution during simulation with ideal conditions using control architecture 2.	7

3–13 Yaw evolution during simulation with ideal conditions using control architecture 2	72
3–14 Forward and upward velocity evolution during simulation with ideal conditions using control architecture 1	73
3–15 Forward and upward velocity evolution during simulation with ideal conditions using control architecture 2	73
3–16 Control commands during simulation with ideal conditions using control architecture 1.	74
3–17 Control commands during simulation with ideal conditions using control architecture 2.	74
3–18 Wind speed at the airship position during the simulation with external disturbances.	75
3–19 Roll and pitch evolution during simulation with external disturbances using control architecture 1	76
3–20 Yaw evolution during simulation with external disturbances using control architecture 1.	76
3–21 Roll and pitch evolution during simulation with external disturbances using control architecture 2.	76
3–22 Yaw evolution during simulation with external disturbances using control architecture 2.	76
3–23 Forward and upward velocity evolution during simulation with exter- nal disturbances using control architecture 1.	77
3–24 Forward and upward velocity evolution during simulation with exter- nal disturbances using control architecture 2	77
3–25 Control commands during simulation with external disturbances using control architecture 1.	78
3–26 Control commands during simulation with external disturbances using control architecture 2.	78

4–1	Block diagram of the proposed combined Lyapunov/Backstepping Controller	82
4–2	Determination of the correct representation for the desired attitude quaternion \boldsymbol{q}_d .	89
4–3	Roll and pitch evolution during simulation with zero wind and PID control.	95
4-4	Yaw evolution during simulation with zero wind and PID control. $\ .$.	95
4–5	Roll and pitch evolution during simulation with zero wind and nonlinear control	95
4–6	Yaw evolution during simulation with zero wind and nonlinear control.	95
4–7	Forward and upward velocity evolution during simulation with zero wind and PID control	97
4–8	Forward and upward velocity evolution during simulation with zero wind and nonlinear control.	97
4–9	Control commands during simulation with zero wind and PID control.	98
4–10	Control commands during simulation with zero wind and nonlinear control.	98
4–11	Wind conditions during the simulation with an average wind of $2m/s$.	99
4-12	Roll and pitch evolution during simulation with an average wind speed of 0.5m/s	.00
4–13	Yaw evolution during simulation with an average wind speed of 0.5m/s.1	.00
4–14	Roll and pitch evolution during simulation with an average wind speed of $2m/s$.00
4–15	Yaw evolution during simulation with with an average wind speed of $2m/s$.00
4–16	Forward and upward velocity evolution during simulation with an average wind speed of 0.5m/s	.01

4–17	Forward and upward velocity evolution during simulation with an average wind speed of 2m/s	101
4–18	Control commands during simulation with an average wind speed of 0.5m/s.	102
4–19	Control commands during simulation with an average wind speed of 2m/s	102
4-20	Overview of the area used for flight testing	104
4–21	Roll and pitch evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1	107
4–22	Yaw evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.	107
4–23	Forward and vertical velocity evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.	109
4-24	Control commands during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.	109
4–25	Airship path during a flight test with the simple height control (4.32) and the controller gains given in table 4–1	110
4–26	Airship height during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.	110
4–27	Roll and pitch evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2	112
4–28	Yaw evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2.	112
4–29	Forward and upward velocity evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2.	113
4–30	Control commands during a flight test with attitude compensated height control and the controller gains given in table 4–2	113

4–31	Airship path during a flight test with attitude compensated height control and the controller gains given in table 4–2.	114
4–32	Airship height during a flight test with attitude compensated height control and the controller gains given in table 4–2.	114
4–33	Inertial wind speeds measured at the flight test site	116
5–1	Illustration of the virtual spring-damper system used for trajectory tracking	118
5–2	Projection of the commanded velocity vector \boldsymbol{v}_c onto the body axes	122
5 - 3	Roll and yaw for wind compensation.	123
5–4	Steady state condition for the path control	124
5 - 5	Reference vehicle path during the high-level controller simulation	126
5-6	Path tracking performance under low wind conditions	129
5 - 7	Airship height evolution under low wind conditions	130
5–8	Position error evolution during simulation of test case 5	132
5 - 9	Position error evolution during simulation of test case 6	132
5-10	Roll and pitch angles evolution during simulation of test case 5	133
5-11	Yaw angle evolution during simulation of test case 5	133
5-12	Velocity evolution during simulation of test case 5	135
5 - 13	Lateral velocity evolution during simulation of test case 5. \ldots .	135
5-14	Forward airspeed evolution during simulation of test case 5	136
5-15	Wind speed measured at the ground station during the first flight test.	138
5-16	Airship path during square tracking flight test	139
5 - 17	Position evolution during hover flight test.	139
5-18	Position error during square tracking and hover flight test	140

5-19	Airship height during square tracking and hover flight test 140
5–20	Roll and pitch angles evolution during the high-level controller flight test
5–21	Yaw angle evolution during the high-level controller flight test 141
5-22	Airship path during second flight test
5–23	Airship height during second flight test
5-24	Wind speed at the ground station during the second flight test 143
5-25	Yaw angle evolution during the second flight test
6–1	Roll and pitch evolution during the simulation with perfect wind knowledge
6-2	Yaw evolution during the simulation with perfect wind knowledge 145
6–3	Forward and vertical velocity evolution during the simulation with perfect wind knowledge
6–4	Control commands during the simulation with perfect wind knowledge. 146
6–5	Roll and pitch evolution during the simulation with average wind knowledge
6–6	Yaw evolution during the simulation with average wind knowledge 147
6–7	Forward and vertical velocity evolution during the simulation with average wind knowledge
6–8	Control commands during the simulation with average wind knowledge.148
6–9	Roll and pitch evolution during the wind estimation simulation with constant wind
6–10	Yaw evolution during the wind estimation simulation with constant wind
6–11	Forward and vertical velocity evolution during the wind estimation simulation with constant wind

6–12 Estimated wind speed during the wind estimation simulation with constant wind.	156
6–13 Estimated wind speed during the wind estimation simulation with turbulent wind	158
6–14 Estimated wind speed during the simulation of the 2m/s test case from Section 4.2.	158
6–15 Roll and pitch evolution during the simulation of the 2m/s test case from Section 4.2.	160
6–16 Forward and vertical velocity evolution during the simulation of the 2m/s test case from Section 4.2	160
6–17 Yaw evolution during the simulation of the 2m/s test case from Section 4.2	160
6–18 Control commands during the simulation of the 2m/s test case from Section 4.2.	160
6–19 Estimated wind speed during the trajectory tracking flight	161
6–20 Estimated wind speed with a bias of $\pm 1.5^{\circ}$ on the thruster tilt angles.	164
6–21 Estimated wind speed with a bias of $\pm 3^\circ$ on the thruster tilt angles	164
6–22 Estimated wind speed with a bias of 1° on the IMU roll angle	165
6–23 Estimated wind speed with a bias of 2° on the IMU roll angle	165
6–24 Estimated wind speed with a bias of 2° on the IMU pitch angle	166
6–25 Estimated wind speed with a bias of 2° on the IMU yaw angle	166
6–26 Estimated wind speed with a bias of 1° on the IMU Euler angles and the thruster tilt angles	167
6–27 Estimated wind speed during the first high level controller flight test.	168
6–28 Estimated wind speed during the second high level controller flight test.	168
6–29 Design study for improved thruster and IMU alignment	169

D-1	Frequency spectrum of the roll rate p measurement noise	192
D-2	Frequency spectrum of the roll angle ϕ measurement noise	192
D-3	Modeled and measured zero-mean noise on the pitch angle measure- ment over a period of 600 seconds.	194
D-4	Modeled and measured zero-mean noise on the yaw angle measurement over a period of 600 seconds.	194
D–5	Stationary velocity measurement in standalone GPS configuration	195
D-6	Stationary velocity measurement in differential GPS configuration	195

NOMENCLATURE

\boldsymbol{A}	State matrix for a linear state-space system.
A	Airship frontal area.
a_1	Nominator coefficient of thruster dynamics transfer function.
$oldsymbol{a}_c \in \mathbb{R}^2$	Controller internal desired translational accelerations.
$oldsymbol{A}_J \in \mathbb{R}^{3 imes 3}$	Added inertia matrix.
$oldsymbol{A}_m \in \mathbb{R}^{3 imes 3}$	Added mass matrix.
A_p	Airship planform area.
B	Input matrix for a linear state-space system.
b_1	Denominator coefficient of thruster dynamics transfer function.
$oldsymbol{B}_w$	Wind disturbance matrix for a linear state-space system.
C	Output matrix for a linear state-space system.
$oldsymbol{c} \in \mathbb{R}^3$	Position of the vehicle CG in body frame coordinates.
$oldsymbol{C}^{ imes} \in \mathbb{R}^{3 imes 3}$	Skew-symmetric matrix representing $c \times$.
C_A	Axial drag coefficient.
C_{D_n}	Crossflow drag coefficient.
c_h	Gain of the airship height control
c_{ki}	Nonlinear controller parameter ($i = 1, 2; k = a$: Attitude controller,
	k = v: Velocity controller.
c_{ϕ}	Roll angle gain for the high-level controller.

c_ψ	Yaw rate gain for the high-level controller.
D	Feedthrough matrix for a linear state-space system.
d	Parameter vector in generic parameter estimation algorithm.
D	Viscous axial drag force.
E	Error cost function used in generic parameter estimation algorithm.
$ar{m{f}}_{b,g} \in \mathbb{R}^6$	Generalized buoyancy and gravity forces.
$oldsymbol{f}_{b,g} \in \mathbb{R}^3$	Buoyancy and gravity forces.
$oldsymbol{f}_e \in \mathbb{R}^3$	Sum of viscous, buoyancy, gravity and propulsion forces.
F_i	Thrust generated by the i -th thruster.
$ar{m{f}}_{k,a} \in \mathbb{R}^6$	Generalized kinematic and added mass forces. (See (3.4) and (3.5) .)
$oldsymbol{f}_p \in \mathbb{R}^3$	Propulsion forces.
$oldsymbol{f}_{p,c} \in \mathbb{R}^3$	Commanded total thruster force
$oldsymbol{f}_v \in \mathbb{R}^3$	Viscous aerodynamic forces.
G	Generic continuous time or discrete transfer function.
g	Earth acceleration.
$oldsymbol{h} \in \mathbb{R}^3$	Rotational momentum of the airship.
h	Sample time
$oldsymbol{J} \in \mathbb{R}^{3 imes 3}$	Airship inertia matrix about centre of buoyancy.
$oldsymbol{J}_a \in \mathbb{R}^{3 imes 3}$	Apparent inertia matrix.
$oldsymbol{J}_D \in \mathbb{R}^{3 imes 3}$	Inertia matrix of the displaced air.
$oldsymbol{J}_g \in \mathbb{R}^{3 imes 3}$	Airship inertia matrix about centre of gravity.
$ar{oldsymbol{J}}_u \in \mathbb{R}^{9 imes 5}$	Input Jacobian matrix
$ar{oldsymbol{J}}_w \in \mathbb{R}^{9 imes 6}$	Disturbance Jacobian matrix

 $ar{m{J}}_x \in \mathbb{R}^{9 imes 9}$ State Jacobian matrix

- K(s) Linear controller transfer function.
- k_p, k_d, k_i Controller gains for PID control.
- *M* Viscous aerodynamic moment.
- *m* Vehicle mass (including the helium).

 $\boldsymbol{M}_a \in \mathbb{R}^{3 \times 3}$ Apparent mass matrix.

- $\bar{M}_a \in \mathbb{R}^{6 \times 6}$ Generalized apparent mass matrix. See (2.23).
- m_D Displaced mass.

 $\boldsymbol{M}_{Da} \in \mathbb{R}^{3 \times 3}$ Apparent mass of the displaced air.

 $\bar{M}_{Da} \in \mathbb{R}^{6 \times 6}$ Generalized apparent displaced mass matrix. See (2.31).

N Viscous aerodynamic force perpendicular to airship x axis.

 $\boldsymbol{n}_e \in \mathbb{R}^3$ Sum of viscous, buoyancy, gravity and propulsion moments.

 $n_g \in \mathbb{R}^3$ Gravity moments.

 $\boldsymbol{n}_p \in \mathbb{R}^3$ Propulsion moments.

- $\boldsymbol{n}_{p,c} \in \mathbb{R}^3$ Commanded total thruster moment
- $\boldsymbol{n}_v \in \mathbb{R}^3$ Viscous aerodynamic moments.

 $p \in \mathbb{R}^3$ Translational momentum of the airship.

 $\boldsymbol{q} \in \mathbb{R}^4$ Airship attitude quaternion $[\begin{array}{cccc} q_0 & q_1 & q_2 & q_3 \end{array}]^T$. See (2.1)

 $Q_{\omega} \in \mathbb{R}^{4 \times 3}$ Coupling matrix from angular rates to quaternion derivative. See (2.2)

Q Cost function of thruster allocation optimization problem.

$$q_0$$
 Dynamic pressure. See (B.5).

 $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ Direction cosine matrix of the transformation from inertial to body frame. See (A.3)

- $r_b \in \mathbb{R}^3$ Position of the airship centre of buoyancy in the inertial NED frame.
- $r_g \in \mathbb{R}^3$ Position of the airship centre of gravity in the inertial NED frame.
- \boldsymbol{r}_r Position of the trajectory tracking reference vehicle.

 $r_{Ti,k}$ Location of *i*th thruster in *k* direction (k = x, y, z).

- $\boldsymbol{r}_w \in \mathbb{R}^3$ Location of the wind observer relative to the inertial observer.
- $\pmb{T} \in \mathbb{R}^{3 \times 3}$ Transformation matrix between Euler angle rates and body angular rates.
- T_{new} Updated kinetic energy of the system. See (2.36).
- T_{ref} Kinetic energy of the system as defined in [2]. See (2.30).
- T_{zw} Transfer matrix from the inputs to the outputs of a lower fractional transformation.
- $u_{d,h}$ Desired velocity in the horizontal plane in direction of the airship heading ψ .
- $\boldsymbol{v} \in \mathbb{R}^3$ Inertial velocity vector $\begin{bmatrix} u & v & w \end{bmatrix}^T$ of airship centre of buoyancy.
- $ar{m{v}} \in \mathbb{R}^6$ Generalized vehicle velocity $[\begin{array}{cc} m{v}^T & m{\omega}^T \end{array}]^T$.

$$\boldsymbol{v}_{ac} \in \mathbb{R}^3$$
 Airspeed at the aerodynamic centre $\begin{bmatrix} u_{ac} & v_{ac} & w_{ac} \end{bmatrix}^T$.

- \boldsymbol{v}_c Commanded velocity vector for trajectory tracking.
- v_{err} Difference between commanded and actual velocity in body y direction.
- $\boldsymbol{v}_q \in \mathbb{R}^3$ Inertial velocity of the airship centre of gravity.
- V_k Control Lyapunov function (k=1,2,v).
- \boldsymbol{v}_r Velocity of the trajectory tracking reference vehicle.
- V Volume of the airship hull.
- $\boldsymbol{v}_w \in \mathbb{R}^3$ Wind speed at location of the airship.

$ar{m{v}}_{m{w}}\in\mathbb{R}^{6}$	Generalized wind speed [$\boldsymbol{v}_w^T \ \boldsymbol{\omega}_w^T$] ^T .
w	Inputs to a lower fractional transformation.
$oldsymbol{W}_0$	Output disturbance weighing matrix for H_{∞} controller synthesis.
$oldsymbol{W}_e$	Control error weighing matrix for H_{∞} controller synthesis
$oldsymbol{W}_{u}$	Control effort weighing matrix for H_{∞} controller synthesis.
$oldsymbol{W}_w$	Wind disturbance weighing matrix for H_{∞} controller synthesis.
$oldsymbol{x} \in \mathbb{R}^{13}$	Vehicle state vector [$\boldsymbol{r}_b^T \ \boldsymbol{q}^T \ \boldsymbol{v}^T \ \boldsymbol{\omega}^T$] ^T
x_{ac}	Distance of the aerodynamic centre from the origin of the body frame.
$oldsymbol{x}_k$	kth state vector of a strict feedback system.
$oldsymbol{x}_l \in \mathbb{R}^9$	Reduced vehicle state vector used for linear controller design.
z	Outputs from a lower fractional transformation.
z	Discrete time unit delay operator.
$oldsymbol{z}_k$	Virtual state vector $(k=1,2,v)$.
α	Angle of Attack for viscous force computation. See (B.6).
γ	Upper bound on H_{∞} norm of closed loop system.
η	Crossflow efficiency factor.
θ	Airship pitch angle.
μ_i	Tilt angle of the <i>i</i> -th thruster.
$oldsymbol{\xi}_c \in \mathbb{R}^3$	Controller internal desired rotational accelerations.
ρ	Density of the surrounding fluid.
Σ	IMU noise covariance matrix.
σ_1,σ_2	High-level controller design parameters.
$ au_p$	Commanded total thruster moment

ϕ	Airship roll angle.
ψ	Airship yaw angle.
$oldsymbol{\omega} \in \mathbb{R}^3$	Airship inertial rotational velocity vector $\begin{bmatrix} p & q & r \end{bmatrix}^T$.
$oldsymbol{\omega}_w \in \mathbb{R}^3$	Rotational velocity of the wind at the airship position.

Subscripts

С	Controller internal commanded value
d	Desired value
h	Component in the airship heading frame
hor	Component in the horizontal plane
i	Thruster number. $i = 1, 2, 3, 4$.
Ι	Vector is expressed in inertial reference frame

ABBREVIATIONS

ALTAV	Almost Lighter Than Air Vehicle
CG	Centre of Gravity
CV	Centre of Volume
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LFT	Linear Fractional Transformation
LPV	Linear Parameter Varying
LQR	Linear Quadratic Regulator
MAV	Miniature Aerial Vehicle
PID	Proportional, Integral, Differential
SISO	Single In Single Out
SLC	Sequential Loop Closure

UAV Unmanned Aerial Vehicle

CHAPTER 1 Introduction

1.1 Motivation

During the last two decades, there has been a resurgence in airship research. Airships are well suited for long-duration missions during which high speeds are not critical. Like helicopters, airships are capable of hovering but they require significantly less energy to stay airborne. Most of the airship's lift is generated statically by the lifting gas inside the airship hull, reducing energy requirements to that needed to perform attitude and velocity control and to overcome environmental disturbances.

There are numerous projects around the world using modern large-scale airships for advertising and passenger sight-seeing flights. The "Zeppelin NT" shown in Figure 1–1 is built in Friedrichshafen, Germany. It is a semi rigid airship that entered service in 2001. The hull shape of a semi rigid airship is maintained in part by structural elements and in part by the pressure of the lifting gas. The Zeppelin NT is one of the few current airships certified for commercial passenger transport in Europe, Japan and the US.

Most other large airships currently in operation are so-called blimps. The airship hull of a blimp is held in shape purely by the pressure of the lifting gas. Structural elements are only required to support external elements such as fins, gondola, motors etc. Blimps are currently mainly used for advertising purposes, such as the "Goodyear GZ-20", the "WDL 1A", or the "Virgin Airship A60". The "Skyship



Figure 1–1: The Zeppelin NT [3] Figure 1–2: The Cargolifter CL 160 [4]

600" is the largest blimp currently in operation. Like the Zeppelin NT, it is used primarily for passenger transportation.

Cargo transportation is another key area of interest for airships, as airships can transport a payload over large distances and deliver it directly to its destination. This is useful especially for bulky loads that would normally require a diversity of modes of transportation, e.g. ship, truck, and helicopter to reach the destination. A frequently discussed application for large cargo transport airships is the supply of Canada's remote northern regions, for which the conference "Airships to the Arctic" is held biannually.

There are currently no large cargo airships flying, but there are numerous proposed designs. The "Cargolifter CL160" project, shown in Figure 1–2, was aimed at designing an airship capable of transporting a payload of up to 160t to remote locations of the world. While technically promising, the project ended in 2005 due to financial difficulties. Currently under development is the "Boeing Skyhook" which is designed to carry 40t payload as shown in Figure 1–3. The third key area of interest is unmanned and possibly autonomous airships. These can be used for a wide variety of purposes. Besides research missions, e.g., for wildlife monitoring, aerial photography, or atmospheric measurements, civil safety and security missions are a major area of application. Unmanned airships can be used for traffic monitoring, search and rescue missions and also law enforcement tasks such as surveillance and search for suspects or criminal activity. Their low noise level allows airships to perform these missions without disturbing the environment.

Due to the low mass of the typical payload, unmanned, autonomous airships are generally much smaller than airships for cargo or passenger transportation, making them more susceptible to atmospheric disturbances. As they float in the surrounding air, they tend to follow every movement of the air, such as wind gusts or thermals, unless measures are taken to counter these effects.

The classical way to control the airship motion and attitude is to use tail fins with movable control surfaces. These provide aerodynamic stability, naturally pointing the airship nose into the oncoming airflow. The downside of tail fins is their ineffectiveness at low airspeeds which makes them useless for hovering in low wind conditions. Fins are only effective within a limited range of angles of attack α and sideslip angles β . Turns have to be flown with a limited slip angle leading to sluggish airship response and limited maneuverability. Also, tail fins prove problematic for airship mooring. In the presence of wind gusts, the fins can generate a strong moment to rotate the airship around the mooring mast like a wind vane.

An alternative approach to airship motion control is the use of differential thrust. This method has the advantage of being equally effective for all airspeeds and largely independent of α and β . This allows for good maneuverability both at low speed hover and for attitude changes at higher speeds. If the thruster configuration of the airship allows full attitude control at zero wind hover, it also allows for attitude control during forward flight. The tail fins can then be omitted, improving ground handling characteristic and reducing the airship empty mass. Furthermore, while airships with fins are known for their sluggish response, the finless design significantly increases the maneuverability of the vehicle [5].

Airships that feature such a finless design include the original concept for the "Boeing Skyhook" shown in Figure 1–3, the "High Altitude Airship" from Galileo Systems, and the "Quanser MkII" developed by Quanser Inc. in Toronto, Ontario, shown in Figure 1–4. In addition to the advantages described above, the finless design also leads to inherent aerodynamic instability of the vehicle. Efficient stabilizing control algorithms are required for the operation of finless airships, which makes detailed study of the control of finless airships interesting.

In this thesis, the control of finless airships is studied and the developed control algorithms are validated in flight tests with the Quanser MkII. The control task includes simple stabilization and velocity control tasks as well as trajectory tracking and autonomous hover. As the airship motion depends strongly on the wind speed, possibilities to estimate the current wind conditions are also investigated.

The Quanser MkII is an "Almost Lighter Than Air Vehicle" (ALTAV), in which the majority but not all of its weight is counteracted by static lift. The remaining lift is provided by four tiltable thrusters that can produce forward, backward and upward thrust as shown in Figure 1–4. A detailed description of this vehicle can be found in [5].



Figure 1–3: The Boeing Skyhook [6]



Figure 1–5: The Draganflyer X4 quadrocopter [7]

Figure 1–4: The ALTAV Quanser MkII

Due to its small size, the Quanser MkII allows for flight testing without a large infrastructure to support flight operations. It has therefore been chosen as the vehicle to be used in this work. However, the control algorithms developed here are not limited to use in this vehicle. They can be adapted to other finless airships as well as any other aerial vehicle with attitude control actuators that are independent of α and β . Quadrocopters, such as the one shown in Figure 1–5, are an example of such vehicles.

1.2 Literature review

The relevant literature for this thesis covers three principle areas: airship dynamics, airship control, and wind estimation. The topic of airship control can be further subdivided into low-level control, which stabilizes the airship attitude and velocity, and high-level control which computes desired attitude and velocity values to track a given trajectory. References that deal with both tasks at the same time will be dealt with in the low-level control sections. The airship control references can be further split into references that use linear or nonlinear control techniques.

1.2.1 Airship dynamic modelling

The dynamics of airships have been well researched and a detailed overview of the available literature is given in [8]. Significant insight into the dynamics of airships has been provided by [9], [10], and [11]. The author used potential flow theory to derive the aerodynamic forces on airship hulls. His works give the first analytical description of the destabilizing pitch and yaw moments acting on elongated airship hulls. In the honour of the author these moments are called Munk moment. The references also confirm and explain data found previously in wind tunnel tests [12].

The Munk moment is a direct result of the added mass effect encountered by airships and other vehicles that float in the surrounding fluid. Due to the large volume of the vehicle, the surrounding air has to be accelerated with the vehicle increasing the apparent inertia of the vehicle. This effect has been investigated in detail in [13], providing methods to compute the added mass parameters for a given hull shape.

The aerodynamic forces due to the viscosity of the fluid also have significant influence on airship dynamics. The authors of [14] studied these in detail providing insight on the computation of the cross-flow drag for the entire range of angles of attack. The axial drag for airship hulls is investigated in [15] as a function of the fineness ratio (the ratio between maximum hull diameter and hull length).

Besides the aerodynamic forces and the added mass effect, the airship dynamics are influenced by the gravity force acting at the centre of gravity, the buoyancy created by the lifting gas acting at or close to the centre of volume, as well as actuation forces. Also, since the motion equations are usually expressed in a bodyfixed frame [16], the respective virtual kinetic forces need to be taken into account. References [16] and [17] both provide a good description of these forces, giving a complete picture of the airship dynamics in the absence of external disturbances.

The effect of wind gusts is investigated in detail in [18], [2], and [19]. Due to the added mass effect, wind gusts have a significant impact on the airship motion. Part of this effect cannot be represented purely by the resulting change in airspeed and the associated aerodynamic forces. Hence, the author of [18] introduces force terms related to the wind speed rate of change and the spatial wind speed gradient. In [2] and [19], instead of the spatial gradient, a rotational wind speed as well as the wind speed rate of change are introduced. The two representations are interchangeable.

The author of [1] assembled a complete model specific to the ALTAV Quanser MkII. The equations of motion include the wind gust terms derived in [19], and the associated added mass terms have been determined according to [20]. The computation of viscous forces due to the translational motion is based on [14] and [21]. A model of the thrusters used on the Quanser MkII is also developed based on stationary and transitory thrust measurements performed in stagnant air. Aspects of the vehicle motion that are not dealt with in [1] are rotational viscous terms as well as thruster gyroscopic effects.

1.2.2 Airship control using linear control algorithms

There are numerous references studying the control of airships with fins and many of these use linear control algorithms.

An approach to control all attitude parameters as well as the velocity is shown in [22] for a 15m long airship with control fins in classical cross configuration. In this approach, independent SISO-controllers are used to track the pilot commands such as desired velocity or desired pitch using the fins for attitude control and the thrust for velocity control. The authors present simulation as well as flight test results demonstrating the effectiveness of their control algorithm.

The authors of [23], [24], and [25] study the control of the airship AURORA, a 9m long airship with fins for attitude control and two tiltable thrusters that can provide differential thrust. In [23] they present three PID SISO controllers for velocity, altitude and heading or track control using the control surface and total thrust as control inputs. Controller performance is validated in a simulation environment. In [25], the parameters of the PID controllers have been optimized by minimization of the closed-loop H_2 and H_{∞} norms. In order to perform trajectory tracking, an outer-loop heading controller is added in [24] to a slightly modified PID attitude and

velocity controller. The performance of the outer loop controller is investigated in simulation. The authors also investigate the performance with an additional PD roll controller to dampen roll oscillations.

In [26], the authors discuss the control of a similar sized airship as AURORA using LQR and SLC control algorithms. The first control architecture investigated consists of an inner-loop LQR controller to control the desired angular and translational velocities and a proportional control outer-loop to track the desired trajectory. The SLC control architecture provides a control law to perform trajectory tracking and attitude control in a single step. The performance of the two controllers is compared in a simulation environment performing a variety of maneuvers such as tracking random waypoints and altitude steps.

The authors of [27] and [28] investigate the autonomous trajectory tracking control of the AURORA airship using PID and H_{∞} control algorithms. The PID control architecture consists of a classical inner-loop/outer-loop structure with the inner loop controlling the airship heading and the outer-loop commanding the required heading to perform trajectory tracking. The H_{∞} control algorithm performs the trajectory tracking in a single loop. In [24], the performance of the two controllers is compared in simulations without and with wind, making this the first reference discussed here that explicitly considers wind during the simulation. The PID controller was also validated in flight tests, and the results of these tests are presented in [27] as well as in [29] and [30]. The same research group uses a LQR controller in combination with a vision system to track optical features as described in [31]. The control system performance is verified using a simulation environment without wind disturbance.

The control of a finless airship using PID controllers is shown in [5] and [1]. In [5], the results of outdoor flight tests, including trajectory tracking trials, are briefly discussed, whereas [1] focusses on attitude stabilization. As both references deal with the same vehicle as used in this thesis, this controller is the reference for the control studies herein.

The control of elongated finless airships using more advanced linear control theories has not yet been investigated in the literature.

1.2.3 Airship control using nonlinear control algorithms

Due to the high level of nonlinearity in the airship equations of motion, nonlinear control algorithms provide a promising method to stabilize and control the airship motion.

A popular nonlinear control method is input-output linearization, also called feedback linearization. In this technique, the knowledge of the plant dynamics is fed back into the system in a specific way to create linear behaviour between the input to the plant and its output. This now linear plant is then stabilized using classical linear algorithms.

Input-output linearization is used in [32], [33], and [34] to control the attitude of airships with fins. In [32], the authors design a feedback linearization controller to control the Euler angles of an airship which is not specified in detail. They validate
the controller performance in simulations that include an uncertainty in the airship parameters.

The authors of [33] and [34] both study feedback linearization with respect to a 50m long airship equipped with tail fins in an inverse Y configuration. In [33], a feedback linearization controller is designed to control the pitch and yaw angles as well as the velocity, and its performance is analyzed in simulation. The authors of [34] introduce a neural network in the feedback linearization to achieve adaptability of the controller. Simulations for two different test cases, one at sea level and one at an altitude of 3000m are shown to evaluate controller performance.

In [35], feedback linearization is used in a single control loop to perform trajectory tracking of a large, high-altitude airship with fins in an inverse-Y-configuration. The controller includes estimation of the inertia parameters. Its performance is verified by tracking a spiral trajectory in a simulation without wind.

Feedback linearization is also used in [30] and [36] to follow a predefined path at a given altitude and speed with the airship AURORA. The controller performance is verified in a simulation that includes a wind disturbance. Additionally, in [30], a backstepping controller is designed to perform airship hover in the presence of sufficient wind to keep the tail fins effective.

Backstepping is a Lyapunov based control technique that uses a sum of Lyapunov functions to prove stability for strict feedback systems. This will be explained in more detail in Section 4. Besides [30], this technique is also used in various references to control airships with fins as discussed in the following. The backstepping control approach taken in [30] has been extended in [37], [38], and [39] to include input saturations. References [37] and [38] focus on the the hover task, while [39] adds trajectory following to the task and shows simulation results for a complete airship mission including vertical take-off and landing, path tracking and hover. The simulation includes wind as external disturbance. As the path tracking controller requires wind knowledge, a wind estimator using the measured airspeed is also presented. The hover case in all references includes sufficient wind to ensure effectiveness of the control fins.

Backstepping control is also used for trajectory tracking of a 6m long blimp with tail fins in cross configuration in [40] and [41]. Controller performance is demonstrated using simulations of the hover case in [41] and of trimmed trajectories in [40], both without wind disturbances.

The authors of [42] employ backstepping control for the trajectory tracking of a small-scale blimp with fins in an X configuration and thrusters allowing differential thrust. They demonstrate control performance in a simulation without and with wind disturbance for the tracking of a helical trajectory.

A slightly different approach to backstepping control has been taken by the authors of [43]. They design 6 individual SISO controllers based on a backstepping technique to control each of the airship position coordinates and each Euler angle with a separate controller. Additionally, the authors investigate aeroelastic effects on the airship hull and adapt their controller design to account for these. Simulation results are shown with and without aeroelastic effects considered. A mix of nonlinear control strategies, especially Lyapunov and Backstepping control is used for the control of an 8m long airship with fins in an X configuration as described in [44], [45], [46], and [47]. The control strategy of this airship consists of different controllers for takeoff, lateral flight, longitudinal flight, and landing. The transition between the controllers occurs depending on the flight phase and the deviations from the desired targets. In [44], the authors prepare the ground work by describing the equations of motion for the blimp in question. The remaining three references develop the control architecture and provide simulation results with increasingly complicated trajectories. Only in [47] are wind disturbances considered.

Lyapunov based control is used in [48] for the trajectory tracking task as well as hover. A parameter estimator updates the controller parameters in flight to better match the properties of the vehicle. The authors show a simulation of the tracking of a S-shaped trajectory with a one-time wind disturbance lasting 10 seconds.

The authors of [49] and [50] use nonlinear control with Lyapunov stability proofs to control the planar motion of a small indoor blimp with fins in the presence of uncertainty. In [49], the authors assume an uncertainty on some of the airship parameters and a constant wind. The controller performance is validated in indoor experiments. In [50], the wind is assumed to have a known direction but unknown force and the controller is adapted to include this uncertainty. The performance is validated in a simulation.

Another nonlinear control method is sliding mode control, which is also a Lyapunov based control technique, in which the controller drives the plant to remain on a defined sliding surface of the plant's states. A separate control law is used to drive the plant close to the sliding surface. Once the error is small enough, the sliding mode law takes over.

The authors of [51] use sliding mode control to perform the trajectory tracking task in 3D space for an unmanned blimp. The vehicle motion is split into lateral and longitudinal motion and, for each, a sliding mode controller is designed. Controller performance is verified in simulation with the blimp flying a rectangular search pattern over a target area.

In [52], an adaptive sliding mode controller is designed to control the lateral motion of a high altitude airship with tail fins in cross configuration and a pusher propeller. An outer loop using fuzzy logic is added to follow a predefined trajectory. Simulation results assuming a constant wind speed are presented.

Nonlinear control of an elongated finless airship that features more than 6 actuation degrees of freedom has not yet been studied.

1.2.4 Path tracking control

Some of the references discussed above deal both with the path tracking and the airship stabilization task. Some literature also deals with the trajectory tracking task without application to a vehicle or providing details on the stabilization task. These algorithms can prove useful as a high-level controller that provides commanded values to a separate low-level controller. As real-time execution is required for flight testing, only algorithms requiring little computational effort will be discussed here. Furthermore, it shall be feasible to modify the desired trajectory in flight. Therefore, algorithms requiring offline preprocessing have also been omitted here. A popular approach to the path tracking task is pursuit-based control. In this control method, the vehicle tracks a target point that is at a fixed or variable distance ahead of the vehicle on the reference path. It is used in [53] to control a Quadrotor vehicle along a predefined path using a PI-controller. The authors present simulation as well as experimental results with wind as external disturbance. In [54], the pursuit scheme is used for waypoint navigation of a fixed-wing jet aircraft without providing details on the employed low-level control. The authors also include wind disturbances in the presented simulation results. A pursuit-based control with variable distance for the target point is investigated in [55]. The scheme is applied to a fixed-wind aircraft under the assumption of a constant aircraft airspeed and constant wind speed. Controller performance is validated in a simulation environment for a variety of different wind conditions. The authors of [56] and [57] extend the pursuit scheme for improved handling of curved trajectories. They present simulation as well as experimental results for a fixed wing aircraft.

A second principal approach is based on vector fields. For this technique, a three-dimensional velocity vector field is computed that asymptotically drives the vehicle towards and along the desired path from any point in space. This approach is described in [58] for a generic UAV and in [59] for Miniature Air Vehicles. The latter reference includes experimental results for a MAV with 1.2m wingspan.

The technique described in [60] uses a reference point on the desired trajectory closest to the actual vehicle position. The desired trajectory is limited to smooth paths actually feasible by the vehicle. The authors then design a controller based on polytopic LPV systems to drive the distance between the reference point and the vehicle to zero. A look-ahead term takes the future evolution of the desired path into account. The tracking performance is shown using the dynamic simulation of a model helicopter in a disturbance-free environment.

The concept described in [61] employs a reference vehicle that travels on the desired path at a predefined speed. A virtual spring-damper system connects the reference vehicle with the actual vehicle, determining the desired motion to asymptotically track the reference vehicle motion. This ensures that the overall progress along the trajectory is done at a predefined velocity, namely the velocity of the reference vehicle.

Trajectory following for airships is discussed in [62]. The authors describe the following along a specific path within a certain distance for an airship with fins. If the maximum permissible distance is exceeded, a new trajectory is computed to bring the airship back to the point at which the path error became excessive, ensuring the airship arrives there with an orientation suitable to continue tracking of the original path. The methodology is useful, if it is important to cover every part of a given trajectory with a certain precision. Controller performance is shown in a simulation environment with a sinusoidal external disturbance.

The authors of [63] investigate the automatic station-keeping of a ball-shaped finless airship. The airship is equipped with three thrusters allowing control of yaw and roll motion as well as forward and upward velocity. The effectiveness of the control is verified in simulations with different constant wind conditions. Trajectory tracking for vehicles that have a preferred direction of travel, but can at the same time travel in all directions independent of orientation, such as elongated finless airships, has not yet been investigated in detail.

1.2.5 Wind estimation algorithms

Determining the current wind conditions at the location of an aircraft based on on-board measurements is a frequently used method in aeronautics. The large majority of the algorithms in use today require some kind of sensor to measure the airspeed, e.g. a pitot tube. Using the difference between the inertial velocity of the vehicle and the measured airspeed, the current wind speed can be computed. The wind estimator presented in [39] is a good example for such an algorithm.

For vehicles that can travel along all of the body axes simultaneously, measuring the airspeed proves very difficult. Pitot tubes require the air flow to be aligned with the direction of the tube for valid measurements. Moreover in small airships, the airspeed may be so low that it is difficult to find a pressure sensor with sufficient sensitivity. For less directional flow speed sensors, such as ultrasonic or hot-wire anemometers, it will still be difficult to find an installation location at which the flow will be undisturbed by the vehicle body for all possible directions of travel.

Algorithms that can estimate the current wind conditions purely based on the motion of the vehicle (e.g., based on GPS measurements) would allow one to find the wind speed and subsequently the airspeed, for cases in which airspeed sensing is not feasible. Very few references have investigated this topic. The authors of [64] develop a filter to estimate the wind speed based on the motion of a fixed-wing aircraft. They provide a proof of convergence and simulation results for a steady-state, coordinated flight.

In [65] and [66], the horizontal component of the wind speed is estimated using a small unmanned aircraft with a delta wing. The wind estimate update is based on the series of the last n measurements. The number n of measurements considered in the estimation is chosen high enough to ensure convergence of the solution. An optimization problem is used to eliminate measurement noise. This approach is not limited to steady-state flight. However, as with the previous reference, the vehicle studied can only travel forward relative to the surrounding air.

A wind estimation algorithm that also works during transient maneuvers and for vehicles travelling in all directions is presented in [67] for a VTOL rotorcraft. Two estimations algorithms are presented and compared, one based on neural networks, the other based on a Kalman-Bucy filter.

Wind estimation algorithms considering the particularities of the dynamics of lighter-than-air vehicles have not yet been studied. One very important particularity of finless airships is that they can encounter any range of angles of attack and angles of sideslip and do not naturally align themselves with the oncoming airflow.

1.3 Dissertation contribution and content overview

In this work, linear and nonlinear control algorithms have been studied to stabilize finless airships like the Quanser MkII. The linear control algorithm investigated is based on H_{∞} control, the nonlinear algorithm uses backstepping and Lyapunov control techniques. In addition to the stabilizing low-level controller, a high-level controller has been developed and evaluated. It is capable to autonomously track a predefined trajectory as well as hover at a fixed location using the same control law. As the airship motion is strongly dependent on the wind conditions, the determination of the ambient wind conditions without additional sensors, and the use of this knowledge in the controller are investigated too.

The key contribution of this work is to investigate in detail different options and requirements to control elongated finless airships with the aim of autonomous path tracking. Due to their inherent aerodynamic instability, the control of finless airships is more challenging than that of airships with stabilizing or actuated tail fins. The individual achievements of this thesis are:

- The derivation of the wind-related terms in the equations of motion via a Newton-Euler approach. This rounds out the work done in [2] and [19] who abandoned the Newton-Euler approach for a Lagrange approach. Furthermore, the importance of some of the wind-related terms with respect to the effect of gusts is identified. These effects cannot be modelled with the equations of motion derived in other references such as [17], which are widely used in airship control references.
- The design of a linear H_{∞} controller to track desired attitude and velocity values with an unmanned, finless airship. This control technique provides a high level of robustness to uncertainties and appears therefore a suitable candidate for the control of a highly nonlinear finless airship.
- The design of a nonlinear Backstepping controller to track desired attitude and velocity values with an unmanned, finless airship. The performance of

this controller is compared in a simulation environment to the PID controller described in [1] and [5] as well as the H_{∞} controller mentioned above.

- Extensive flight testing to validate the performance of the nonlinear Backstepping controller in an outdoor environment showing a performance very similar to the simulation test cases.
- Development of a high-level controller to allow the airship to travel along a predefined trajectory in the presence of realistic external disturbances. The controller is designed to take advantage of the fact that the vehicle under investigation can travel in all directions while ensuring that the vehicle assumes its preferred direction of travel when possible. The controller performance is also validated in flight tests.
- A wind-estimation technique that allows estimation of the wind speed based on the airship's dynamic response to thrust inputs.

This work is divided as follows. In Chapter 2, the equations of motion for finless airships in a wind-field are derived using a Newton-Euler approach. The equation parameters are adapted to the Quanser MkII and assembled into a simulation. The simulation includes airship dynamics, thruster dynamics and a sensor noise model and constitutes the basis for the controller design in subsequent chapters. An analysis of the airship response to wind gusts is used to determine the dominant wind-related terms.

The design of low-level airship controllers is discussed in Chapters 3 and 4. The low-level airship controllers stabilize the airship attitude and velocity and allow tracking of desired roll, pitch and yaw values as well as desired velocities in forward and upward direction. In Chapter 3, the equations of motion are linearized at an arbitrary operating point and a low-level controller based on H_{∞} control techniques is designed. The performance of this controller is investigated in a simulation environment. In Chapter 4, another low-level controller is designed using Backstepping techniques and Lyapunov control. The performance of this controller is first investigated in a simulation environment and subsequently also during outdoor flight tests.

Using the second low-level controller for airship stabilization, a high-level controller is developed in Chapter 5. The controller allows autonomous flight by providing trajectory tracking capability between waypoints, as well as the ability to hover at a given position. The performance of this controller is first evaluated in the simulation environment. It is then tested in outdoor flight tests and the flight test results are compared with the simulation results.

As the wind has been found to dominate the airship motion, an algorithm to estimate the current wind conditions based on the airship motion data is developed in Chapter 6. It is shown that using wind data in the low-level control algorithm promises significant performance improvements. The performance of the estimation algorithm is investigated using the airship simulation. The wind estimation results during the flight tests conducted are also discussed.

Chapter 7 provides the overall conclusions from the work performed as well as an outlook on future work on this topic.

CHAPTER 2 Airship dynamics simulation

A reliable model of the plant dynamics is essential for successful controller development. Hence, as the first step in the controller design process, a high fidelity simulation of the vehicle dynamics is assembled. The simulation includes a detailed model of the vehicle dynamics, including added mass effects, wind influence, and viscous effects, and a model of the actuator dynamics as well as a wind-gust model and sensor dynamics modelling.

2.1 Reference frames and kinematic coupling

The motion of an airship is often described employing a body-fixed reference frame of which the origin coincides with the airship hull centre of volume. While this choice of reference frame leads to an appearance of the first moments of inertia in the motion equations, it significantly simplifies the formulation of the added mass and buoyancy forces that are all acting on the centre of volume.

The orientation of the body-fixed frame is such that the x axis points forward along the longitudinal axis of the airship. The z axis points down in the symmetry plane of the vehicle. The y axis points to the right to complete a right-hand triad.

The relation between the inertial north-east-down (NED) frame, subscript Iand the body frame are shown in Figure 2–1 together with a wind observer frame, subscript w, that will be used later on.

Using this convention for the body frame, the vehicle state vector \boldsymbol{x} consists of



Figure 2–1: Reference frames used for the equations of motion

- the location of the body frame origin expressed in the inertial frame $r_b = [r_{b,n} \ r_{b,e} \ r_{b,d}]^T$,
- the orientation of the body-fixed frame, either described by the Euler angles roll φ, pitch θ, and yaw ψ according to the standard convention used in aviation (see [68]) or by a quaternion q,
- the translational velocity of the vehicle $\boldsymbol{v},$
- and the rotational velocity of the vehicle $\boldsymbol{\omega}$.

The velocity vectors \boldsymbol{v} and $\boldsymbol{\omega}$ can be formulated in any reference frame. The default here will be the body frame, which will be assumed for vectors without subscript. Vectors formulated in the inertial frame will feature the subscript I.

For the vehicle attitude, the quaternion representation has been chosen, in order to avoid the singularity problem of the Euler Angle representation. In accordance with the convention used in [68], [69], and [70], the quaternion will be defined as

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\gamma}{2} \\ \hat{\boldsymbol{k}} \sin \frac{\gamma}{2} \end{bmatrix}.$$
 (2.1)

with \hat{k} being the unit vector that represents the direction of the rotation and γ representing the total angle.

The kinematic coupling between the attitude quaternion and the angular rates is given in [68] and can be rearranged to

$$\begin{bmatrix} \dot{q}_{0} \\ \dot{q}_{1} \\ \dot{q}_{2} \\ \dot{q}_{3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_{0} \\ q_{1} \\ q_{2} \\ q_{3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_{1} & -q_{2} & -q_{3} \\ q_{0} & -q_{3} & q_{2} \\ q_{3} & q_{0} & -q_{1} \\ -q_{2} & q_{1} & q_{0} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \frac{1}{2} \mathbf{Q}_{\omega} \boldsymbol{\omega}$$

$$(2.2)$$

with $\boldsymbol{Q}_{\omega}^{T}\boldsymbol{Q}_{\omega}=\boldsymbol{I}_{3}.$

The kinematics of the airship position are given by

$$\dot{\boldsymbol{r}}_b = \boldsymbol{R}^T \boldsymbol{v} \tag{2.3}$$

with \boldsymbol{R} being the direction cosine matrix that transforms vectors from the inertial North-East-Down to the body-fixed reference frame. More details on this transformation are available in Appendix A.1.

The dynamics of the vehicle states v and ω depend on a variety of external influences such as kinematics, added mass effect, aerodynamics, as well as gravity and propulsion as shown in [16].

Added mass is an effect resulting from the fact that acceleration of the vehicle also requires the acceleration of the surrounding fluid. Due to the large size and small mass of the vehicle, the inertia of the accelerated fluid is comparable to that of the vehicle. The amount of fluid to be accelerated depends on the vehicle shape and the direction of acceleration. This effect is taken into account by increasing the inertia of the airship by the added mass terms. A detailed investigation of this effect is available in [13].

2.2 Derivation of the kinematic and added mass forces

As a first step, the conservative terms acting on the dynamics of v and ω will be derived. The conservative forces acting on the vehicle are the kinematic forces due to the body fixed reference frame, gravity forces and potential flow forces, which are identical to the added mass forces. For simplicity, gravity forces will be neglected during the derivation and added separately later on.

The respective dynamics, including the forces due to the change in wind speed $\dot{\boldsymbol{v}}_w$, have been derived using a Lagrange approach in [18] and [19]. The authors of [19] initially attempted to derive the equations of motion using a Newton-Euler approach in [2] but did not succeed.

To ensure that the equations of motion used for the controller design are correct, the Newton-Euler approach will be applied here with the goal of confirming the findings of [19] and [18]. This will also be used to investigate in detail some particularities about the definition of the kinetic energy used in [19] and [18] for the Lagrange approach.

In order to conduct a Newton-Euler derivation the translational and the angular momentum of the system need to be determined. The control volume border is chosen such that it includes the airship and the added mass. When analyzing this system, the loss of momentum of the air displaced by the airship needs to also be considered.

This leads to three components that define the translational momentum p of the system. The first component is the momentum of the vehicle itself, whose centre of gravity moves with the velocity v_g . Having a mass of m, this gives the momentum mv_g for the vehicle itself. The second component considers the momentum of the added mass A_m and the air to be displaced by the airship m_D . In the absence of the airship this air parcel $M_{Da} = A_m + m_D I_3$ moves with the wind speed v_w . Due to the presence of the airship, the velocity of this air parcel has changed to be identical with the velocity of the airship's centre of volume v. The contribution of this component to the overall translational momentum of the system is hence $M_{Da}(v - v_w)$. Additionally to these two components, the loss of the momentum of the air displaced by the airship m_D has to be considered. Before the vehicle is inserted into the control volume, this air is travelling with speed v, due to the previous acceleration of M_{Da} . The associated momentum $m_D v$ is lost in the system. This gives for the translational momentum of the system the equation

$$\boldsymbol{p} = m\boldsymbol{v}_g + \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_w) - m_D \boldsymbol{v}, \qquad (2.4)$$

which is consistent with the corresponding equation in [2].

The derivation of the angular momentum is more complex, as two separate aspects have to be considered: the angular momentum due to the actual rotation of the airship $\boldsymbol{\omega}$, as well as the angular momentum due to the translational velocity relative to the inertial observer. The latter has not been considered in [2], which is why the authors had to publish the erratum [19]. In the erratum, the Newton-Euler approach was abandoned for a Lagrange approach.

The first three components of the angular momentum are analogous to the translation momentum (2.4) and describe the angular momentum due to the rotation of the airship. Replacing the translational velocities and inertias in (2.4) with the appropriate angular terms gives

$$\boldsymbol{h}_{rot} = \boldsymbol{J}_g \boldsymbol{\omega} + \boldsymbol{J}_{Da} (\boldsymbol{\omega} - \boldsymbol{\omega}_w) - \boldsymbol{J}_D \boldsymbol{\omega}$$
(2.5)

for the angular momentum due to vehicle rotation with ω_w representing the wind rotational velocity.

The angular momentum due to the translational motion at a distance r from the inertial observer is again composed of three components. The first component is the momentum due to the motion of the vehicle itself. Relative to the inertial observer, the vehicle's centre of gravity moves with velocity v_g and is located at r_g . Considering the vehicle mass m, this gives the momentum $r_g \times mv_g$. The second component describes the change of momentum of the added mass, including the air to be displaced by the airship. In absence of the airship, this air parcel M_{Da} is at a location r_w relative to the inertial observer and travels with velocity v_w . In presence of the airship, the location of this air parcel has changed to be coincident with the airship centre of buoyancy, located at r_b . Its velocity simultaneously changed from v_w to v. The impact of this component on the angular momentum is therefore given by $(r_b - r_w) \times M_{Da}(v - v_w)$.

The last component describes the loss of momentum from the displaced air m_D . After acceleration of the air parcel M_{Da} the air to be displaced by the airship m_D is located at \mathbf{r}_b relative to the inertial observer travelling with velocity \mathbf{v} . The corresponding momentum $\mathbf{r}_b \times m_D \mathbf{v}$ is therefore lost, when the airship is introduced in the control volume.

This gives for the angular momentum due to the translational motion of the airship the equation

$$\boldsymbol{h}_{trans} = \boldsymbol{r}_g \times m\boldsymbol{v}_g + (\boldsymbol{r}_b - \boldsymbol{r}_w) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_w) - \boldsymbol{r}_b \times m_D \boldsymbol{v}$$
(2.6)

The total angular momentum of the system is given by the sum of (2.5) and (2.6) yielding

$$\boldsymbol{h} = \boldsymbol{J}_{g}\boldsymbol{\omega} + \boldsymbol{J}_{Da}(\boldsymbol{\omega} - \boldsymbol{\omega}_{w}) - \boldsymbol{J}_{D}\boldsymbol{\omega} + \boldsymbol{r}_{g} \times m\boldsymbol{v}_{g} + (\boldsymbol{r}_{b} - \boldsymbol{r}_{w}) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_{w}) - \boldsymbol{r}_{b} \times m_{D}\boldsymbol{v}.$$
(2.7)

2.2.1 Translational equations of motion

The velocity of the centre of gravity given in terms of velocity of the body fixed frame is

$$\boldsymbol{v}_g = \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{c}. \tag{2.8}$$

Putting this relation into (2.4) and transforming to the inertial frame gives

$$p_{I} = \mathbf{R}^{T}(m\mathbf{I}_{3} + \mathbf{A}_{m})\mathbf{R}\mathbf{v}_{I} - \mathbf{R}^{T}m\mathbf{C}^{\times}\mathbf{R}\boldsymbol{\omega}_{I} - \mathbf{R}^{T}(m_{D}\mathbf{I}_{3} + \mathbf{A}_{m})\mathbf{R}\mathbf{v}_{w,I}$$

$$= \mathbf{R}^{T}\mathbf{M}_{a}\mathbf{R}\mathbf{v}_{I} - \mathbf{R}^{T}m\mathbf{C}^{\times}\mathbf{R}\boldsymbol{\omega}_{I} - \mathbf{R}^{T}\mathbf{M}_{Da}\mathbf{R}\mathbf{v}_{w,I}.$$
 (2.9)

where $C^{\times} \equiv c \times$.

According to Newton's second law, we can now differentiate the momentum with respect to time and equate the result to the sum of the external forces acting on the system:

$$\frac{d\boldsymbol{p}_{I}}{dt} = \dot{\boldsymbol{R}}^{T}\boldsymbol{M}_{a}\boldsymbol{R}\boldsymbol{v}_{I} + \boldsymbol{R}^{T}\boldsymbol{M}_{a}(\dot{\boldsymbol{R}}\boldsymbol{v}_{I} + \boldsymbol{R}\dot{\boldsymbol{v}}_{I}) - \dot{\boldsymbol{R}}^{T}\boldsymbol{m}\boldsymbol{C}^{\times}\boldsymbol{R}\boldsymbol{\omega}_{I} - \boldsymbol{R}^{T}\boldsymbol{m}\boldsymbol{C}^{\times}(\dot{\boldsymbol{R}}\boldsymbol{\omega}_{I} + \boldsymbol{R}\dot{\boldsymbol{\omega}}_{I})
- \dot{\boldsymbol{R}}^{T}\boldsymbol{M}_{Da}\boldsymbol{R}\boldsymbol{v}_{w,I} - \boldsymbol{R}^{T}\boldsymbol{M}_{Da}(\dot{\boldsymbol{R}}\boldsymbol{v}_{w,I} + \boldsymbol{R}\dot{\boldsymbol{v}}_{w,I})
= \boldsymbol{f}_{e,I}.$$
(2.10)

The vector of external forces \boldsymbol{f}_e represents the sum of the viscous aerodynamic forces \boldsymbol{f}_v , buoyancy and gravity forces $\boldsymbol{f}_{b,g}$, and propulsion forces \boldsymbol{f}_p .

The derivative of the Direction Cosine Matrix \dot{R} is given in [71] and [2] as

$$\dot{\boldsymbol{R}} = -\boldsymbol{\Omega}^{\times} \boldsymbol{R} \Longrightarrow \dot{\boldsymbol{R}}^{T} = \boldsymbol{R}^{T} \boldsymbol{\Omega}^{\times}.$$
(2.11)

Using (2.11) and (A.2), (2.10) can be written as

$$\boldsymbol{R}^{T}\boldsymbol{\Omega}^{\times}\boldsymbol{M}_{a}\boldsymbol{R}\boldsymbol{v}_{I} + \boldsymbol{R}^{T}\boldsymbol{M}_{a}\dot{\boldsymbol{v}} - \boldsymbol{R}^{T}\boldsymbol{\Omega}^{\times}\boldsymbol{m}\boldsymbol{C}^{\times}\boldsymbol{R}\boldsymbol{\omega}_{I} - \boldsymbol{R}^{T}\boldsymbol{m}\boldsymbol{C}^{\times}\dot{\boldsymbol{\omega}}$$

$$-\boldsymbol{R}^{T}\boldsymbol{\Omega}^{\times}\boldsymbol{M}_{Da}\boldsymbol{R}\boldsymbol{v}_{w,I} - \boldsymbol{R}^{T}\boldsymbol{M}_{Da}(-\boldsymbol{\Omega}^{\times}\boldsymbol{R}\boldsymbol{v}_{w,I} + \boldsymbol{R}\dot{\boldsymbol{v}}_{w,I}) = \boldsymbol{f}_{e,I}.$$
(2.12)

Transforming back to body coordinates, we get the equation of motion

$$\boldsymbol{M}_{a} \dot{\boldsymbol{v}} - \boldsymbol{m} \boldsymbol{C}^{\times} \dot{\boldsymbol{\omega}} = -\boldsymbol{\Omega}^{\times} \boldsymbol{M}_{a} \boldsymbol{v} + \boldsymbol{\Omega}^{\times} \boldsymbol{m} \boldsymbol{C}^{\times} \boldsymbol{\omega} + \boldsymbol{\Omega}^{\times} \boldsymbol{M}_{Da} \boldsymbol{v}_{w} - \boldsymbol{M}_{Da} \boldsymbol{\Omega}^{\times} \boldsymbol{v}_{w} + \boldsymbol{M}_{Da} \boldsymbol{R} \dot{\boldsymbol{v}}_{w,I} + \boldsymbol{f}_{e},$$
(2.13)

which is identical to equations (15) to (17) in [2].

2.2.2 Rotational equations of motion

As a first step, the velocity and position terms in the angular momentum equation (2.7) are transformed into body frame terms using (2.8) and

$$\boldsymbol{r}_g = \boldsymbol{r}_b + \boldsymbol{c}. \tag{2.14}$$

The airship inertia matrix is also transformed to reflect the inertia with respect to the body frame origin using

$$\boldsymbol{J}_g - m\boldsymbol{C}^{\times}\boldsymbol{C}^{\times} = \boldsymbol{J}. \tag{2.15}$$

This yields

$$h = J_a \omega - J_{Da} \omega_w + r_b \times m(v + \omega \times c) + c \times mv + (r_b - r_w) \times M_{Da}(v - v_w) - r_b \times m_D v.$$
(2.16)

with $\boldsymbol{J}_a = \boldsymbol{J} + \boldsymbol{A}_J$ and $\boldsymbol{J}_{Da} = \boldsymbol{J}_D + \boldsymbol{A}_J$.

Transforming the equation to inertial coordinates gives

$$h_{I} = \mathbf{R}^{T} \mathbf{J}_{a} \mathbf{R} \boldsymbol{\omega}_{I} - \mathbf{R}^{T} \mathbf{J}_{Da} \mathbf{R} \boldsymbol{\omega}_{w,I} + \mathbf{R}^{T} (\mathbf{R} \mathbf{r}_{b,I} \times m \mathbf{R} \mathbf{v}_{I}) + \mathbf{R}^{T} (\mathbf{R} \mathbf{r}_{b,I} \times m (\mathbf{R} \boldsymbol{\omega}_{I} \times \mathbf{c})) + \mathbf{R}^{T} (\mathbf{c} \times m \mathbf{R} \mathbf{v}_{I}) + \mathbf{R}^{T} ((\mathbf{R} \mathbf{r}_{b,I} - \mathbf{R} \mathbf{r}_{w,I}) \times \mathbf{M}_{Da} (\mathbf{R} \mathbf{v}_{I} - \mathbf{R} \mathbf{v}_{w,I})) - \mathbf{R}^{T} (\mathbf{R} \mathbf{r}_{b,I} \times m_{D} \mathbf{R} \mathbf{v}_{I}).$$

$$(2.17)$$

Differentiating the angular momentum with respect to time gives the sum of the external moments acting on the airship with respect to the inertial observer

$$\frac{d\boldsymbol{h}_{I}}{dt} = \boldsymbol{n}_{e,I} + \boldsymbol{r} \times \boldsymbol{f}_{I}.$$
(2.18)

The vector of external moments n_e corresponds to the sum of the gravity moment n_g , the propulsion moments n_p as well as the viscous aerodynamic moments n_v . The force f_I represents the forces acting on the system and the position vector r gives the location at which the corresponding force acts in relation to the inertial observer.

Equation (2.17) is now differentiated with respect to time, transformed back to body coordinates and rearranged considering that $\boldsymbol{v} \times m\boldsymbol{v} = \boldsymbol{0}$ for scalar mass terms. This gives

$$\Omega^{\times} \boldsymbol{J}_{a} \boldsymbol{\omega} + \boldsymbol{J}_{a} \dot{\boldsymbol{\omega}} - \Omega^{\times} \boldsymbol{J}_{Da} \boldsymbol{\omega}_{w} - \boldsymbol{J}_{Da} \dot{\boldsymbol{\omega}}_{w} + \boldsymbol{r}_{b} \times \left(\boldsymbol{M}_{a} \dot{\boldsymbol{v}} - m \boldsymbol{C}^{\times} \dot{\boldsymbol{\omega}} - \boldsymbol{M}_{Da} \dot{\boldsymbol{v}}_{w} \right) - \boldsymbol{r}_{w} \times \boldsymbol{M}_{Da} (\dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_{w}) + \boldsymbol{v} \times m (\boldsymbol{\omega} \times \boldsymbol{c}) + \Omega^{\times} (\boldsymbol{c} \times m \boldsymbol{v}) + \boldsymbol{c} \times m \dot{\boldsymbol{v}} + (\boldsymbol{v} - \boldsymbol{v}_{w}) \times \boldsymbol{A}_{m} (\boldsymbol{v} - \boldsymbol{v}_{w})$$

$$= \boldsymbol{n}_{e,I} + \boldsymbol{r} \times \boldsymbol{f}_{I}.$$

$$(2.19)$$

The translational equation of motion (2.13) gives a description of all forces acting on the airship at the location \boldsymbol{r}_b from the inertial observer; the equation therefore gives a contribution to $\boldsymbol{r} \times \boldsymbol{f}_I$ on the right hand side of (2.19). Moving the term $\boldsymbol{M}_{Da} \dot{\boldsymbol{v}}_w$ to the left hand side of (2.13) allows computation of this contribution to be $\boldsymbol{r}_b \times (\boldsymbol{M}_a \dot{\boldsymbol{v}} - \boldsymbol{m} \boldsymbol{C}^{\times} \dot{\boldsymbol{\omega}} - \boldsymbol{M}_{Da} \dot{\boldsymbol{v}}_w)$.

A second contribution to the term $\boldsymbol{r} \times \boldsymbol{f}_I$ is given by a force that will be discussed in detail later on and is given in equation (2.34) for the generalized case including forces and moments. Taking only the force-components of (2.34) gives the force $-\boldsymbol{M}_{Da}(\boldsymbol{\dot{v}}-\boldsymbol{\dot{v}}_w)$ acting on the air parcel that travels with the wind. This air parcel is therefore located at \boldsymbol{r}_w from the inertial observer giving a contribution to $\boldsymbol{r} \times \boldsymbol{f}_I$ of $-\boldsymbol{r}_w \times \boldsymbol{M}_{Da}(\boldsymbol{\dot{v}}-\boldsymbol{\dot{v}}_w)$.

This gives for the term $\boldsymbol{r} \times \boldsymbol{f}_I$ on the right hand side of (2.19) the description

$$\boldsymbol{r} \times \boldsymbol{f}_{I} = \boldsymbol{r}_{b} \times \left(\boldsymbol{M}_{a} \dot{\boldsymbol{v}} - \boldsymbol{m} \boldsymbol{C}^{\times} \dot{\boldsymbol{\omega}} - \boldsymbol{M}_{Da} \dot{\boldsymbol{v}}_{w} \right) - \boldsymbol{r}_{w} \times \boldsymbol{M}_{Da} (\dot{\boldsymbol{v}} - \dot{\boldsymbol{v}}_{w}).$$
(2.20)

The right hand side of (2.20) cancels with the corresponding terms on the left hand side of (2.19), yielding the equation of motion

$$J_{a}\dot{\boldsymbol{\omega}} + m\boldsymbol{C}^{\times}\dot{\boldsymbol{v}} = -m\boldsymbol{C}^{\times}\boldsymbol{\Omega}^{\times}\boldsymbol{v} - \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{a}\boldsymbol{\omega} + \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{Da}\boldsymbol{\omega}_{w} + \boldsymbol{J}_{Da}\dot{\boldsymbol{\omega}}_{w} - (\boldsymbol{v} - \boldsymbol{v}_{w}) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_{w}) + \boldsymbol{n}_{e}.$$
(2.21)

To achieve the formulation (2.21), the terms $\boldsymbol{c} \times$ in equation (2.19) have been replaced by \boldsymbol{C}^{\times} and the Jacobi identity

$$\boldsymbol{v} \times (\boldsymbol{\omega} \times \boldsymbol{c}) + \boldsymbol{\omega} \times (\boldsymbol{c} \times \boldsymbol{v}) = -\boldsymbol{c} \times (\boldsymbol{v} \times \boldsymbol{\omega})$$
 (2.22)

was applied.

Applying the Jacobi identity also to the terms $-\boldsymbol{\omega} \times (\boldsymbol{c} \times m\boldsymbol{v})$ and $m\boldsymbol{v} \times (\boldsymbol{c} \times \boldsymbol{\omega})$ of equation (5) in [19], transforms that equation into a form identical to equation (2.21) above.

Equations (2.13) and (2.21) can be combined into a single equation by introduction of the generalized quantities

$$\bar{\boldsymbol{M}}_{a} = \begin{bmatrix} \boldsymbol{M}_{a} & -\boldsymbol{m}\boldsymbol{C}^{\times} \\ \boldsymbol{m}\boldsymbol{C}^{\times} & \boldsymbol{J}_{a} \end{bmatrix} \text{ and } (2.23)$$

$$\bar{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}$$
(2.24)

which yields

$$\bar{\boldsymbol{M}}_{a}\dot{\bar{\boldsymbol{v}}} = \begin{bmatrix} -\boldsymbol{\Omega}^{\times}\boldsymbol{M}_{a}\boldsymbol{v} + \boldsymbol{\Omega}^{\times}\boldsymbol{m}\boldsymbol{C}^{\times}\boldsymbol{\omega} + \boldsymbol{\Omega}^{\times}\boldsymbol{M}_{Da}\boldsymbol{v}_{w} - \boldsymbol{M}_{Da}\boldsymbol{\Omega}^{\times}\boldsymbol{v}_{w} + \boldsymbol{M}_{Da}\boldsymbol{R}\dot{\boldsymbol{v}}_{w,I} & +\boldsymbol{f}_{e} \\ -\boldsymbol{m}\boldsymbol{C}^{\times}\boldsymbol{\Omega}^{\times}\boldsymbol{v} - \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{a}\boldsymbol{\omega} + \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{Da}\boldsymbol{\omega}_{w} + \boldsymbol{J}_{Da}\dot{\boldsymbol{\omega}}_{w} - (\boldsymbol{v} - \boldsymbol{v}_{w}) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_{w}) & +\boldsymbol{n}_{e} \end{bmatrix}$$

$$(2.26)$$

This form of the equations of motion corresponds well to the definition given in [16].

As discussed above, the external force and moment terms f_e and n_e are the sum of buoyancy, gravity, viscous and propulsion forces and moments. The individual terms of the sums

$$\boldsymbol{f}_{e} = \boldsymbol{f}_{b,g} + \boldsymbol{f}_{v} + \boldsymbol{f}_{p}$$
 and (2.27)

$$\boldsymbol{n}_e = \boldsymbol{n}_g + \boldsymbol{n}_v + \boldsymbol{n}_p \tag{2.28}$$

will be derived in Appendix B.

2.2.3 Effect of the Munk moment

The term

$$\boldsymbol{n}_{Munk} := -(\boldsymbol{v} - \boldsymbol{v}_w) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_w)$$
(2.29)

on the right hand side of (2.21) is called the Munk moment. It is a destabilizing moment that acts to align the airship hull perpendicular to the oncoming airflow. To demonstrate the effect of this moment, the following simulations of the motion described by (2.26) and (2.2) have been conducted.

The values chosen for the parameters present in (2.26) and (2.2) are given in Table 2–1. They correspond to a first approximation of the parameters representing the Quanser MkII. The precision of these values will be improved further down. For a simple qualitative demonstration of the motion, the precision is sufficient.

Variable	Values
Airship mass m	$5 \mathrm{kg}$
CG-Position \boldsymbol{c}	(0.15, 0, 0.3) [m]
Displaced mass m_D	$4.5 \mathrm{kg}$
Added mass matrix \boldsymbol{A}_m	diag(3,6,6) [kg]
Initial airship velocity \boldsymbol{v}_0	(2, 0, 0) [m/s]
Initial airship angular rates $\boldsymbol{\omega}_0$	(0, 0.2, 0) [rad/s]
Initial airship attitude \boldsymbol{q}_0	(1,0,0,0)

To best demonstrate the effect of the Munk moment and to show the influence of wind on the airship motion, the simulation is initialized as follows. The initial airship attitude is $\boldsymbol{q}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ and it flies with a inertial velocity of $\boldsymbol{v}_0 = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^T$ m/s. The angular rates are initialized with $\boldsymbol{\omega}_0 = \begin{bmatrix} 0 & 0.2 & 0 \end{bmatrix}^T$ rad/s, i.e. the airship rotates around the pitch axis. Each simulation was run with a different wind speed \boldsymbol{v}_w . In the first simulation there was no wind at all. The second simulation featured a constant wind speed of $\boldsymbol{v}_w = \begin{bmatrix} 0.5 & 0 & 0 \end{bmatrix}^T \text{m/s}$. In the third simulation, the wind speed is $\boldsymbol{v}_w = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^T \text{m/s}$ making the wind speed equal to the initial airship velocity.

The wind speed has been chosen to be parallel to the airship velocity, as a crosswind component would introduce an additional yawing and rolling motion, which would be counterproductive for a qualitative understanding of the added mass and wind effects. The motion of flights with a crosswind component will be investigated later.

The pitch angle evolution during all three simulations is shown in Figure 2–2, the evolution of the roll angle shown in Figure 2–3. The evolution of the airship velocity in the inertial x-direction is shown in Figure 2–4 and the velocity in the inertial z-direction in Figure 2–5. The yaw angle as well as the velocity in the inertial y-direction are zero during the entire simulation.

The pitch angle of the airship oscillates between -90° and +90° in all three simulations. Each time, the pitch angle reaches 90°, the roll angle changes by 180°, which corresponds to a motion purely around the pitch axis. As can be seen in Figure 2–2, the pitch angle changes at a constant rate for the simulation with $v_w=2m/s$, whereas the pitch rate fluctuates for the other two test cases. The rate of change of the pitch angle is lowest for pitch angles around 0 and highest at pitch angles of 90 degrees. This is an effect of the Munk moment as shown in equation (2.29), which destabilizes the airship around the pitch attitude of 0°.



Figure 2–2: Airship pitch evolution dur- Figure 2–4: Evolution of the airship veing the 3 Simulations with different wind locity northwards during the 3 Simulaconditions. tions with different wind conditions.



Figure 2–3: Airship roll evolution during Figure 2–5: Evolution of the airship vethe 3 Simulations with different wind con- locity downwards during the 3 Simuladitions. tions with different wind conditions.

The airship inertial velocity in the inertial x direction u_i is almost constant for the simulation with $v_w=2$ m/s. There is only a slight variation in speed which is due to the fact that the velocity u_i is the velocity of the body frame origin, which is not the airship centre of gravity. Due to the rotation, the speed of the body frame changes slightly, as it rotates around the centre of gravity.

In the simulations with lower wind speeds, the airship velocity is higher than the wind speed. Hence, the effect of the added mass has an additional influence on the inertial velocity. The added mass term in the body x direction is less than the term in the body y and z directions. Hence, in order to conserve the airship's momentum and kinetic energy, the inertial velocity drops, as the airship rotates away from 0° pitch and increases as the airship approaches 0° pitch.

2.2.4 Conservation of energy

In order to further verify the correctness of (2.26), it will be investigated how these equations respect the conservation of kinetic energy. The simulation setup is identical to the simulations in the previous section with identical initial vehicle conditions. In contrast to the previous simulations, the wind speed in this simulation is initialized to be perpendicular to the airship inertial velocity at $\boldsymbol{v}_{w,I} = \begin{bmatrix} 0 & 0.5 & 0 \end{bmatrix}^T \text{m/s}$. This lateral windspeed induces additional rotation of the airship around the yaw and roll axes, allowing an investigation of the conservation principle for a more complex motion than the motion in the previous section.

The simulation is run for t=100s and the kinetic energy as defined in [2]

$$T_{ref} = \frac{1}{2} \bar{\boldsymbol{v}}^T \bar{\boldsymbol{M}}_a \bar{\boldsymbol{v}} - \bar{\boldsymbol{v}}^T \bar{\boldsymbol{M}}_{Da} \bar{\boldsymbol{v}}_w + \frac{1}{2} \bar{\boldsymbol{v}}_w^T \bar{\boldsymbol{M}}_{Da} \bar{\boldsymbol{v}}_w$$
(2.30)

using the additional generalized quantities

$$\bar{\boldsymbol{M}}_{Da} = \begin{bmatrix} \boldsymbol{M}_{Da} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_{Da} \end{bmatrix}$$
 and (2.31)

$$\bar{\boldsymbol{v}}_w = \begin{bmatrix} \boldsymbol{v}_w \\ \boldsymbol{\omega}_w \end{bmatrix}$$
(2.32)

is computed. The Euler angle evolution during this simulation is shown in Figure 2–6, the evolution of T_{ref} is shown as the solid line in Figure 2–7.

In this simulation, the airship attitude changes drastically around all three axes, although the motion was initialized with a pitch rate only. The cause of this is the wind speed initially coming from the left. This leads to a Munk moment around the yaw axis initiating a yaw rate to the right. As the centre of gravity is not coincident with the centre of volume, this combined pitch and yaw motion also induces a roll motion. The airship hence enters a full tumbling motion. As the simulation does not contain any non-conservative forces such as thruster or viscous forces, the motion is not damped and the kinetic energy of the system should be conserved.



during the tumbling motion simulation.

Figure 2–6: Airship attitude evolution Figure 2–7: Evolution of the airship kinetic energy T_{ref} and T_{new} during the tumbling motion simulation.

Interestingly, the kinetic energy T_{ref} as defined in [2] does not remain constant but oscillates around values of about 18J. To understand this phenomenon, it is necessary to take a close look at the steps undertaken in [18] and [2] to formulate the kinetic energy (2.30). In order not to recite the entire process here, we will focus on the crucial step which is the acceleration of the air parcel \bar{M}_{Da} to the generalized airship velocity \bar{v} .

In this step, the air parcel \bar{M}_{Da} consisting of the air to be displaced by the airship and the added mass around the airship is accelerated from wind speed \bar{v}_w to the airship velocity \bar{v} "via a set of impulsive pressures over its surface" [18]. Hence, the force $\bar{f}_{M_{Da}}$ achieving this acceleration acts between the air parcel \bar{M}_{Da} and the surrounding air. The work done by this force hence is

$$W = \frac{1}{2} \left(\bar{\boldsymbol{v}}^T - \bar{\boldsymbol{v}}_w^T \right) \bar{\boldsymbol{M}}_{Da} \left(\bar{\boldsymbol{v}} - \bar{\boldsymbol{v}}_w \right)$$
(2.33)

and it defines the kinetic energy of the air parcel \bar{M}_{Da} relative to an observer moving with wind speed \bar{v}_w .

However, the force $\bar{f}_{M_{Da}}$ accelerating \bar{M}_{Da} has the opposite effect on the surrounding air according to Newton's third law. This effect is not discussed in either Thomasson[18] or Azinheira[2]. Effectively, this reaction force means that accelerations of the airship also change the wind speed \bar{v}_w around the airship.

In order to perform a simulation that conserves T_{ref} it would be necessary to include a dynamic model for the surrounding air. The wind speed $v_{w,i}$ would then be a state of the overall system rather than an external parameter. The magnitude of the change of windspeed depends in this case on the size of the control volume chosen. Larger control volumes will lead to smaller changes of the wind speed.

For a sufficiently large control volume, the mass of the surrounding air is so huge that the airship has a negligible effect on its velocity, and this is why the simulation conducted here assumes a constant inertial wind speed. To account for this effect, we need to consider an external force acting on the surrounding air that maintains the wind speed constant by compensating the reaction force described above. Assuming that the acceleration of the air parcel \bar{M}_{Da} occurs at constant acceleration over the period of time Δt , the force $\bar{f}_{M_{Da}}$ becomes

$$\bar{\boldsymbol{f}}_{M_{Da}} = \bar{\boldsymbol{M}}_{Da} \frac{\bar{\boldsymbol{v}} - \bar{\boldsymbol{v}}_w}{\Delta t}$$
(2.34)

In order to cancel the effect of the airship motion on the wind speed, this force needs to be applied from the outside on the air surrounding the airship. This air moves at the velocity $\bar{\boldsymbol{v}}_w$, hence the work done by the external force is

$$W_{Ext} = \left(\bar{\boldsymbol{M}}_{Da} \frac{\bar{\boldsymbol{v}} - \bar{\boldsymbol{v}}_w}{\Delta t}\right)^T \bar{\boldsymbol{v}}_w \Delta t = \bar{\boldsymbol{v}}^T \bar{\boldsymbol{M}}_{Da} \bar{\boldsymbol{v}}_w - \bar{\boldsymbol{v}}_w^T \bar{\boldsymbol{M}}_{Da} \bar{\boldsymbol{v}}_w \qquad (2.35)$$

considering that the force is only active during the period of time Δt during which the air parcel \bar{M}_{Da} is accelerated. It is important to note, that this force is considered implicitly in the simulation by setting the wind speed constant.

Adding the work of the external force (2.35) to the kinetic energy of the system (2.30) yields, after some reorganization, a different formulation of the kinetic energy

$$T_{new} = T_{ref} + W_{Ext} = \frac{1}{2} \bar{\boldsymbol{v}}^T \bar{\boldsymbol{M}}_a \bar{\boldsymbol{v}} - \bar{\boldsymbol{v}}_w^T \bar{\boldsymbol{M}}_{Da} \bar{\boldsymbol{v}}_w.$$
(2.36)

The dotted line in Figure 2–7 shows the evolution of T_{new} during the simulation described above. It remains constant with a value of approximately 15.5J, and thus constitutes a conservative quantity of the system under the assumption of a constant wind speed. Despite numerical noise, this value of T_{new} remained constant within 10^{-10} J, and this could be further reduced by reducing the tolerance of the numerical integrator.

In order to evaluate the right hand side of (2.26), the terms f_e and n_e need to be determined. As discussed above, the external forces and moments are the sums of viscous aerodynamic terms, gravity and buoyancy terms and propulsion terms. These terms have been taken from the literature and are given in Appendix B.

2.3 Simulation setup

The architecture of the airship dynamics simulation used in this thesis is shown in Figure 2–8. The model is created in a MatLab/Simulink environment using continuous time computations with a fixed step size of 2.5ms. The continuous time solver uses a Runge-Kutta integrator. The airship dynamics model incorporates all the relevant equations from Sections 2.1 and 2.2, appendices B.1 and B.2 as well as (B.13) and (B.14). The thruster model is based on the remaining equations in Appendix B.3. The sensor model reflects the sensor noise modelling described in Appendix D. It also includes the sensor sample rates of the vehicle which are 50Hz for IMU and sonar and 10Hz for the GPS readings.

The wind model provides the wind speed in an inertial frame $v_{w,I}$ as well as its derivative $\dot{v}_{w,I}$ to the airship model. It is based on a van Karman turbulence



Figure 2–8: Architecture of the airship dynamics simulation.

model that superimposes the turbulence on the average wind speed. As the flights conducted in the scope of this work are in a very limited height band, the average wind speed is considered independent from the airship height above ground.

The statistical properties of the power spectrum are chosen in accordance with the average wind speed chosen, as described in [72]. The turbulence intensities, scale lengths and spectra depend on the airship height, with the intensities getting larger towards the ground. The instantaneous values for the turbulence velocities are computed by superimposing 50 separate sine functions, each representing a different part of the turbulence spectrum. This technique gives a frozen turbulence field moving with the average wind speed.

Details on the computation of the turbulence field are given in [73]. This reference successfully used the wind model for the dynamics of a tethered aerostat flying at a height of approximately 100m. The model was later refined in [74] to take into account the higher gust intensities for flights close to the ground. However, these refinements were not used in the simulations used in this work. Instead, discrete wind gusts were superimposed separately in simulations investigating the effect of strong wind gusts.

The control loop can be closed with a freely configurable controller. On the onboard electronic of the Quanser MkII, the controllers can only be implemented in discrete time with a maximum sample rate of 50Hz. The controllers used in the simulation are modelled accordingly. To investigate the possible effect of the sensor noise it is possible to feed the actual airship states into the controller rather than the noisy measurements. The controller provides desired angles and force values for each thruster to the thruster model.

To determine the optimal sampling rate, a series of open loop simulations have been conducted. The airship is initialized at a sufficient altitude with a forward velocity of 2m/s and all Euler angles set to zero. The wind speed is set to zero and the thruster forces are chosen such that the airship is initially in equilibrium. The attitude evolution of the first 30 seconds for this simulation is shown in Figure 2–9, the velocity evolution is shown in Figure 2–10. Despite being a simulation without external disturbances and having the vehicle initially in an equilibrium, the airship quickly becomes unstable in yaw and enters a permanent rotation around the yaw axis.

Due to the thruster model derived in [1] that operates at 400Hz, the slowest feasible sample time is h = 2.5ms. The simulation is hence executed once with this

sample time and once with a 5 times higher sample rate with h = 0.5ms. The differences encountered in the simulations are in the range of numerical noise and hence negligible. A sampling time of h = 2.5ms hence promises the fastest simulation execution time without loss of fidelity.



Figure 2–9: Airship attitude evolution Figure 2–10: Body-frame velocity evoluduring open loop simulation. tion during open loop simulation.

The results of a closed-loop simulation can be used to demonstrate the effect of wind gusts on the airship. As controller, the PID controller described in [75] will be used to stabilize the airship attitude and forward and upward velocities. This controller will also be used in later chapters as a basis of comparison, and so is described here in some detail. The controller consists of 5 SISO PID controllers to control the forward speed u, the upward speed w and the three Euler angles ϕ , θ and ψ . The two speed controllers create collective force commands for all thrusters with the forward speed controller commanding the total force in x-direction

$$F_{i,x,u} = \left(k_{p,u}(u_d - u) + k_{d,u}(\dot{u}_d - \dot{u}) + k_{i,u}\int (u_d - u)\right)$$
(2.37)

and the upward speed controller commanding the total force in z-direction

$$F_{i,z,w} = \left(k_{p,w}(w_d - w) + k_{d,w}(\dot{w}_d - \dot{w}) + k_{i,w}\int (w_d - w)\right).$$
 (2.38)

The attitude is controlled by using differential thrust. The thrusters are numbered from 1 to 4, starting from the front right, going clockwise as seen in a top view. The controller has been modified from the version published in [75] to use the angular rates p, q, r for the D-term rather than the derivative of the noisy Euler angle measurement.

The controller for the roll angle ϕ creates a moment around the body x axis by commanding left-right differential thrust in the z direction

$$\begin{bmatrix} F_{1z,\phi} \\ F_{2z,\phi} \\ F_{3z,\phi} \\ F_{4z,\phi} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \left(k_{p,\phi}(\phi_d - \phi) - k_{d,\phi}p + k_{i,\phi} \int (\phi_d - \phi) \right).$$
(2.39)

The controller for the pitch angle θ creates a moment around the body y axis by commanding forward-aft differential thrust in the z direction

$$\begin{bmatrix} F_{1z,\theta} \\ F_{2z,\theta} \\ F_{3z,\theta} \\ F_{4z,\theta} \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \left(k_{p,\theta}(\theta_d - \theta) - k_{d,\theta}q + k_{i,\theta} \int (\theta_d - \theta) \right).$$
(2.40)

Heading control is achieved by commanding left-right differential thrust in the x direction

$$\begin{bmatrix} F_{1x,\psi} \\ F_{2x,\psi} \\ F_{3x,\psi} \\ F_{4x,\psi} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \left(k_{p,\psi}(\psi_d - \psi) - k_{d,\psi}r + k_{i,\psi}\int(\psi_d - \psi) \right).$$
(2.41)

The force demand for each thruster i is computed via

$$T_{i} = \sqrt{(F_{ix,u} + F_{ix,\psi})^{2} + (F_{iz,w} + F_{iz,\phi} + F_{iz,\theta})^{2}}.$$
(2.42)

A thruster tilt angle of zero represents upwards thrust (in negative z direction). With this convention, the tilt angle μ_i for each thruster is given by

$$\mu_i = \tan^{-1} \frac{F_{ix,u} + F_{ix,\psi}}{-F_{iz,w} - F_{iz,\phi} - F_{iz,\phi}}.$$
(2.43)

It is important to note that the thrusters cannot generate thrust downwards, as the range of motion of the tilt servos is limited to $\pm 90^{\circ}$. Downward thrust demands for an individual thruster are hence set to zero.

During the simulation, the desired Euler angles and the desired forward and upward velocity are all set to zero. The average wind speed is initialized with $[0 \ 0.5 \ 0]^T$. The average wind speed is hence perpendicular to the airship orientation. At t = 15s, the average wind speed is increased to $[0 \ 2.5 \ 0]^T$ over a period of 5 seconds, which corresponds to a rate of change of 0.4m/s^2 .

The evolution of the Euler angles is shown in Figure 2–11. Obviously, the controller keeps the airship stable at the desired attitude even in the presence of
the wind disturbance. The forward velocity and the vertical velocity evolution are shown in Figure 2–12. The controller also achieves to keep both values close to zero. For these parameters, there is no particular response to the wind gust introduced at t=15s.



Figure 2–11: Euler angle evolution dur- Figure 2–12: Velocity evolution during ing the simulation with PID closed-loop the simulation with PID closed-loop concontrol. trol.

As the wind blows perpendicular to the airship heading, it affects primarily the lateral velocity v. The simulation has been executed twice, once with the $\dot{v}_{w,I}$ in (2.13) reflecting the correct rate of change of the wind speed, and once with $\dot{v}_{w,I}$ set to zero. Some references dealing with airship dynamics and control omit this term, and so these results demonstrate the impact of this omission. The evolution of the lateral velocity v for the two test cases is shown in 2–13.

The lateral velocity responds significantly more slowly to the wind gust, when the term $\dot{\boldsymbol{v}}_{w,I}$ is set to zero. In this case, only the viscous cross flow drag makes the airship follow the wind speed. With the wind rate of change $\dot{\boldsymbol{v}}_{w,I}$ taken into



Figure 2–13: Lateral velocity evolution during the simulation with PID closed-oop control.

consideration, the pressure gradient over the hull is correctly represented at the moment when the gust hits the airship.

2.4 Influence of the airship heaviness on the attitude control authority

An important issue that is particular to the ALTAV Quanser MkII platform is the net weight, also called heaviness, and how it affects the control authority of the thrusters. This is a direct result of the fact that the thrusters cannot generate downwards thrust.

The airship heaviness is defined as the difference between the airship weight, including the lifting gas, and the weight of the displaced air. Depending on the amount of payload, the vehicle studied in the work has a heaviness of about 6N to 12N. That means, the vehicle mass is about 600 to 1200g higher than the mass of the displaced air. During stationary flight at constant height, the thrusters need to generate a vertical thrust equivalent to the heaviness of the airship. This condition is illustrated in Figure 2–14 with the thrust of the left and right thrusters combined into one arrow.



Figure 2–14: Thrust generated for stationary level flight.

It is trivial to note that the thrusters need to generate more thrust for an airship with greater heaviness. There are two important consequences of this. For larger thrust forces, the thrusters react more quickly to changes in the thrust commands, as was noted in [1]. Therefore, the thrusters have an improved dynamic response for greater heaviness.

An even more important effect is the effect of control saturation depicted in Figure 2–15 for pitch control inputs. Roll and pitch control inputs are generated by using vertical thruster forces. As the thrusters cannot generate downward forces, the maximum pitch moment is limited as shown in Figure 2–15. No greater moment can be generated without violating the equilibrium of vertical forces.

Large attitude control inputs can hence lead to an excessive thrust force in the vertical direction, and tend to increase the airship height.



Figure 2–15: Limitation of the attitude control authority for different airship heavinesses.

Modifications to the airship that would allow the generation of downward thrust, such as reversible motors or reversible pitch propellers, would overcome this problem, allowing a better attitude control authority without sacrificing vertical motion control.

CHAPTER 3 Linear low-level controller design

The overall controller architecture, shown in Figure 3–1, consists of a high-level controller and a low-level controller. The high-level controller performs the trajectory tracking. It generates desired attitude and velocity values based on the evolution of the desired and actual airship position. The task of the low-level controller is to stabilize the vehicle attitude and velocity at these desired target values. The low-level controller can also be used without high-level controller. In this case, the user directly defines desired attitude and velocity values.



Figure 3–1: Overall control architecture.

In the case of the QuanserMkII, the velocity control is limited to the velocity in body x and z direction, as there are no actuator forces available in body y direction. Control of the lateral velocity will be the task of the high-level controller using appropriate combinations of velocity and attitude commands. The first approach to the design of a low-level controller for an unmanned airship is the design of a low-level controller using the H_{∞} control technique. As the plant parameters are only known with limited accuracy, a control technique with a high level of robustness to parametric uncertainties such as H_{∞} appears to be a sensible choice of control algorithm.

3.1 Linearization of equations of motion

The first step in the design of a linear controller is the linearization of the equations of motion about the desired point of operation. For the linearization, the vehicle position \mathbf{r}_b is omitted from the state vector \mathbf{x} , as these states have no influence on the remaining equations of motion. Also, the quaternion attitude description is not ideal for a linearized description of the equations of motion, as the norm of the quaternion may be violated in larger deviations from the point of linearization. Using instead an Euler angle representation of the attitude avoids this problem and additionally allows to linearize the equations of motion independent of the airship yaw angle as will be shown further down. Hence, the attitude representation is replaced with an Euler angle representation, giving the reduced vehicle state vector

$$\boldsymbol{x}_{l} = \left[\begin{array}{ccccc} \phi & \theta & \psi & u & v & w & p & q & r \end{array} \right]^{T}.$$
(3.1)

The attitude dynamics equation (2.2) is then replaced by the Euler angle dynamics equation as given in [71]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{T}(\phi, \theta) \boldsymbol{\omega} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \boldsymbol{\omega}.$$
 (3.2)

Taking the relevant equations from Chapter 2 and Appendix B, particularly (2.13), (2.21), and (B.2) the complete equation of motion can be assembled to yield

$$\dot{\boldsymbol{x}}_{l} = \begin{bmatrix} \boldsymbol{T}(\phi, \theta)\boldsymbol{\omega} \\ \bar{\boldsymbol{M}}_{a}^{-1} \begin{bmatrix} \boldsymbol{f}_{k,a} + \boldsymbol{f}_{b,g} + \boldsymbol{f}_{v} & +\boldsymbol{f}_{p} \\ \boldsymbol{n}_{k,a} + \boldsymbol{n}_{g} + \boldsymbol{n}_{v} & +\boldsymbol{n}_{p} \end{bmatrix} \end{bmatrix}$$
(3.3)

with

$$\boldsymbol{f}_{k,a} = -\boldsymbol{\Omega}^{\times} \boldsymbol{M}_{a} \boldsymbol{v} + \boldsymbol{\Omega}^{\times} \boldsymbol{m} \boldsymbol{C}^{\times} \boldsymbol{\omega} + \boldsymbol{\Omega}^{\times} \boldsymbol{M}_{Da} \boldsymbol{v}_{w} - \boldsymbol{M}_{Da} \boldsymbol{\Omega}^{\times} \boldsymbol{v}_{w} + \boldsymbol{M}_{Da} \boldsymbol{R} \dot{\boldsymbol{v}}_{w,I}$$
(3.4)

$$\boldsymbol{n}_{k,a} = -m\boldsymbol{C}^{\times}\boldsymbol{\Omega}^{\times}\boldsymbol{v} - \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{a}\boldsymbol{\omega} + \boldsymbol{\Omega}^{\times}\boldsymbol{J}_{Da}\boldsymbol{\omega}_{w} + \boldsymbol{J}_{Da}\dot{\boldsymbol{\omega}}_{w} - (\boldsymbol{v} - \boldsymbol{v}_{w}) \times \boldsymbol{M}_{Da}(\boldsymbol{v} - \boldsymbol{v}_{w})$$
(3.5)

and $\boldsymbol{f}_{b,g}$, \boldsymbol{n}_g are given in (B.2) and (B.3), \boldsymbol{f}_v , \boldsymbol{n}_v are given in (B.10) and (B.11), and \boldsymbol{f}_p , \boldsymbol{n}_p are described in Appendix B.3.

Linearizing (3.3) about an equilibrium state $\boldsymbol{x}_{l,e}$ yields

$$\dot{\boldsymbol{x}}_{l} = \begin{bmatrix} \boldsymbol{I}_{3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\bar{M}}_{a}^{-1} \end{bmatrix} \begin{pmatrix} \boldsymbol{\bar{J}}_{x} \Delta \boldsymbol{x}_{l} + \boldsymbol{\bar{J}}_{u} \Delta \begin{bmatrix} \boldsymbol{f}_{p} \\ \boldsymbol{n}_{p} \end{bmatrix} + \boldsymbol{\bar{J}}_{w} \Delta \begin{bmatrix} \boldsymbol{v}_{w,I} \\ \boldsymbol{\dot{v}}_{w,I} \end{bmatrix} \end{pmatrix}$$
(3.6)

which represents the dynamics of the airframe with the control inputs \boldsymbol{f}_p and \boldsymbol{n}_p and the external disturbance inputs $\boldsymbol{v}_{w,I}$ and $\dot{\boldsymbol{v}}_{w,I}$. The thruster dynamics are treated separately.

The structure and the individual entries of the state Jacobian matrix \bar{J}_x , the input Jacobian matrix \bar{J}_u , and the disturbance Jacobian matrix \bar{J}_w are derived symbolically in Appendix E.

The total thruster forces \boldsymbol{f}_p and moments \boldsymbol{n}_p are nonlinear as shown in Appendix B.3. During controller synthesis, a simplified linear model of the thruster dynamics consisting of a first order low-pass filter will be used. Rather than modelling the dynamics of each thruster, the low-pass filter will be applied to the resulting total thruster forces and moments. This gives the following dynamics

$$\begin{bmatrix} \dot{\boldsymbol{f}}_{p} \\ \dot{\boldsymbol{n}}_{p} \end{bmatrix} = -\frac{1}{\tau_{p}} \begin{bmatrix} \boldsymbol{f}_{p} \\ \boldsymbol{n}_{p} \end{bmatrix} + \frac{1}{\tau_{p}} \begin{bmatrix} \boldsymbol{f}_{p,c} \\ \boldsymbol{n}_{p,c} \end{bmatrix}.$$
(3.7)

A value of 0.2s was chosen for τ_p which corresponds to the slowest time constant of the thruster dynamics determined in [1]. This gives a conservative estimate of the dynamics for the thrusters, as the actual actuation may be faster than what is represented by this model.

Using this thruster model gives the inputs $\mathbf{f}_{p,c}$ and $\mathbf{n}_{p,c}$ to the actuator dynamics model for controller synthesis. The H_{∞} controller will be designed to compute desired values for $\mathbf{f}_{p,c}$ and $\mathbf{n}_{p,c}$. For application to the nonlinear airship, these values will be mapped separately to the individual thruster forces F_i and angles μ_i by a dedicated thrust allocation algorithm. When computing the entries for the Jacobian matrices in (3.6), the cross-flow drag coefficient is assumed to be constant at 1.2 which is valid for crossflow-velocities up to 1m/s. As can be seen in Appendix E, the linearization is independent of the yaw angle ψ , the equilibrium points are hence valid for the entire range of yaw angles. This permits to choose equilibrium points for which the yaw angle changes over time.

The choice of equilibrium points for linearization is therefore limited by the following constraints:

- 1. Crossflow velocity less than 1 m/s.
- 2. Angular rates p, q, r must be such that the roll angle ϕ and the pitch angle θ remain constant.
- 3. The lateral forces f_y must be zero, because thruster forces cannot be applied in this direction to achieve equilibrium.

For the controller design, four equilibrium points are chosen, forward flight, vertical ascent, a mix of the two, and a steady turn with a bank angle of 10° . The roll and pitch attitudes are zero in the first three cases. The pitch attitude is also zero in the fourth case. The pitch rate q and yaw rate r for the fourth case are defined by constraints 2 and 3.

Table $3-1$:	State	values	for	the	equilibrium	points	used	in	H_{∞}	$\operatorname{controller}$	design

Eq. point	ϕ	θ	ψ	u	v	w	p	q	r
	[°]	[°]	[°]	[m/s]	[m/s]	[m/s]	[rad/s]	[rad/s]	[rad/s]
1	0	0	any	2	0	0	0	0	0
2	0	0	any	1	0	-0.25	0	0	0
3	0	0	any	0	0	-0.5	0	0	0
4	10	0	any	2	0	0	0	0.0131	0.0742

The values of each state for the equilibrium points chosen are given in Table 3–1. External disturbances are neglected for the equilibrium determination. Using the state Jacobian matrix \bar{J}_x and the input Jacobian matrix \bar{J}_u , the matrices A, B, C, and D of the state-space system representing (3.6) are

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\bar{M}}_a^{-1} \end{bmatrix} \boldsymbol{\bar{J}}_x$$
(3.8)

$$\boldsymbol{B} = \left| \begin{array}{cc} \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\bar{M}}_a^{-1} \end{array} \right| \boldsymbol{\bar{J}}_u \tag{3.9}$$

$$\boldsymbol{C} = \boldsymbol{I}_9 \tag{3.10}$$

$$\boldsymbol{D} = \boldsymbol{0}_{9\times 5}. \tag{3.11}$$

All vehicle states can be measured by either the IMU or the GPS, hence the output matrix C is the identity matrix. The feedthrough matrix is zero as none of the control inputs influence any output directly.

The Eigenvalues of the state matrix A for each of the equilibrium points are shown in Figure 3–2. Equilibrium point 4, which corresponds to the steady turn motion has unstable poles with the highest frequencies. These poles move closer to the imaginary axis for pure forward flight at the same velocity (equilibrium point 1). As the forward velocity is reduced in equilibrium points 2 and 3, the frequency of the poles reduces significantly. This shows that the vertical ascent motion is the least unstable.



Figure 3–2: State Jacobian matrix \boldsymbol{J}_x Eigenvalues.

The large variation in frequencies of the poles for the different equilibrium points necessitates the use of a control technique capable of dealing with large levels of uncertainty. If gain scheduling is required, this allows a reduction in the number of scheduling points. The H_{∞} control technique provides robustness against large levels of uncertainty and has hence been chosen as the linear control technique for this study.

The linearized model has been validated by comparing the results for small deviations from the equilibrium point between the linear and the nonlinear model. In this comparison, the linear model is confirmed valid, if the first derivatives of the angular rates and the body-frame velocities coincide at the beginning of the simulation. For the airship attitude, the first and second derivatives have to be identical as this is the integral of the angular rates.

Figure 3–3 compares the evolution of the airship attitude for the linear and the nonlinear model. This figure is based on equilibrium point 1, as defined in Table 3–1.



Figure 3–3: Comparison of the attitude evolution in the linear and the nonlinear model.

The simulation has been initialized with some small deviations from the equilibrium: the forward velocity has been reduced by 0.1m/s and the initial roll rate is 5 deg/s instead of 0. The airship attitude is increased by 1 degree from the equilibrium for each Euler angle. The curves for the linear and the nonlinear model coincide perfectly at the start of the simulation, indicating identical values for the first two derivatives and therefore validating the linear model. As the simulation progresses, the curves quickly diverge. This has no impact on the validity of the linear model, but it highlights the high degree of nonlinearity in the airship dynamics.

3.2 Linear controller design

Two different architectures are investigated for the linear controller design. In the first architecture, a single H_{∞} controller is designed to directly control the attitude angles ϕ , θ , and ψ as well as the forward velocity u and the upward velocity w. A diagram of this closed-loop system is shown in Figure 3–4. As discussed previously, the controller computes the desired thruster forces $f_{p,c}$ and moments $n_{p,c}$. These are then translated into the thrust and tilt angle values for the individual thrusters in a separate allocation algorithm.

The second architecture, shown in Figure 3–5, uses the H_{∞} controller to control the angular rate vector $\boldsymbol{\omega}$ instead of the Euler angles ϕ , θ , and ψ . An additional outer loop generates the commanded angular rates to track the desired attitude. This concept has the potential advantage that the nonlinear relationship between $\boldsymbol{\omega}$ and $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ can be explicitly considered in the outer control loop. Also, the H_{∞} controller is potentially simpler as the control task spans only over one integration step of the plant.



Figure 3–4: Closed loop system architecture with the H_{∞} attitude controller (Control architecture 1).

3.2.1 H_{∞} controller synthesis

The computation of the H_{∞} controller matrices uses a linear fractional transformation (LFT) of the closed-loop state space system as shown in Figure 3–6. The LFT allows the incorporation of wind disturbances via the additional input matrix

$$\boldsymbol{B}_{w} = \begin{bmatrix} \boldsymbol{I}_{3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\bar{M}}_{a}^{-1} \end{bmatrix} \boldsymbol{\bar{J}}_{w}.$$
 (3.12)



Figure 3–5: Closed loop system architecture with the H_{∞} angular rates controller (Control architecture 2).



Figure 3–6: Lower fractional transformation used for H_{∞} controller design.

The additional output disturbance depicted in 3–6 has no physical meaning but is required for proper conditioning of the H_{∞} problem. If this term is omitted, one of the Riccatti matrices computed during the controller synthesis becomes singular and the controller synthesis can not be completed successfully.

The outputs of the LFT, denoted by \boldsymbol{z} , are the weighted control effort of the plant actuators and the weighted error between desired and actual plant output.

 H_{∞} controller synthesis aims at designing a controller K(s) that minimizes the H_{∞} norm of the transfer function from the LFT inputs, denoted by w, to the LFT

outputs z. The controller design hence aims at maximum possible attenuation of the disturbances represented at the LFT inputs with respect to the outputs. If a H_{∞} norm below 1 is achieved the transfer function from any input to any output is less than 1 for all frequencies. The desired control performance is defined by choosing the appropriate weighting functions for the inputs and outputs of the LFT.

The disturbances acting on the airship studied in this thesis are primarily the effects of the atmospheric wind. The associated disturbance is fed into the LFT via the weighting matrix \boldsymbol{W}_w . Potential flight tests will be conducted in low wind conditions. In these conditions, wind gusts in the range of 0.5m/s can be expected with wind speed rates of change in the range of 0.1m/s. Setting the wind disturbance matrix \boldsymbol{W}_w to

$$\boldsymbol{W}_{w} = \text{diag}(0.5, 0.5, 0.5, 0.1, 0.1, 0.1) \tag{3.13}$$

represents these conditions in the H_{∞} controller synthesis.

As discussed above, the output disturbance matrix \boldsymbol{W}_0 is only required in order to properly condition the H_∞ problem. It has hence been set to a small arbitrary value of

$$\boldsymbol{W}_0 = 0.001 \boldsymbol{I}_5. \tag{3.14}$$

The outputs of the LFT represent the control effort and the control error. The control effort weighting matrix W_u is used to limit the control effort to what is actually feasible by the control actuators. The airship Quanser MkII allows for a total propulsion force in the x direction of 10N and in the z direction of 20N. The feasible propulsion moments are approximately 10Nm in roll, and 16Nm in pitch and

yaw. These limitations lead to a control effort weighting matrix of

$$\boldsymbol{W}_{u} = \operatorname{diag}\left(\frac{1}{10}, \frac{1}{20}, \frac{1}{10}, \frac{1}{16}, \frac{1}{16}\right).$$
 (3.15)

The control error weighting function $W_e(s)$ is a diagonal state space system individually penalizing the error of each controlled vehicle state. The steady-state error is penalized most strongly, with the penalty dropping by one dB per decade beyond the respective corner frequency. The steady-state gain of $W_e(s)$ for the two controller velocities u and w are 4 and 5 respectively, allowing steady-state errors of 0.25m/s and 0.2m/s. The steady-state gains for the attitude angles are 60, allowing steady-state errors of 0.0167 radians or about 1 degree. In case of controller architecture 2, the gains for the angular rates are also 60, representing steady-state errors of 1 deg/s. The corner frequency has been determined separately for each control variable depending on the vehicle inertia about the respective degree of freedom. The equation for the control error weighting function can be found in appendix C.

The remaining adaptation to correctly setup the Lower Fractional Transformation for H_{∞} controller synthesis is to modify the plant output matrix C to limit the output to the controlled vehicle states. For controller architecture 1, this gives

$$\boldsymbol{C}_{1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$
(3.16)

-

For controller architecture 2, the vehicle state has to be reduced by the Euler angles, as the control task is limited to the angular rates. That means, for computing the lower fractional transformation, the state matrix A is reduced to the bottom right six by six matrix of the matrix given in (3.8) and the input matrix **B** is reduced to the bottom 6 lines of the matrix given in (3.9). The output matrix for this architecture then becomes

$$\boldsymbol{C}_{2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$
(3.17)

Having correctly setup the Lower Fractional Transformation, the H_{∞} controller is computed by solving the optimization problem

$$\min_{K} ||\boldsymbol{T}_{zw}||_{\infty} \tag{3.18}$$

with T_{zw} being the transfer matrix from the inputs to the LFT to its outputs.

Linearization	γ value for	γ value for
point	Architecture 1	Architecture 2
1	10.5058	4.4938
2	0.4731	0.4731
3	0.2584	0.2082
4	2.6094	2.3500

Table 3–2: Minimum γ values achieved during H_{∞} controller synthesis

Currently, there is no analytical solution for (3.18). However, an iterative approximation of the minimum is feasible, giving the suboptimal H_{∞} controller.

The associated process is based on choosing an arbitrary value γ and then verifying, if a controller K exists such that $||\mathbf{T}_{zw}||_{\infty} < \gamma$. The existence of such controller can be proven mathematically for a given γ as shown in [76]. Using an iterative bisection method, an approximate value can be computed for the lowest γ for which a controller exists. The details of this procedure are provided in [76] and will be omitted here for brevity. After determination of the lowest γ , the state-space matrices for a continuous-time H_{∞} controller are computed. These have then been transformed into a discrete time state-space controller with a sample time of 0.02s using a zero-order hold discretization.

The lowest γ values achieved for the different controller architectures for each of the equilibrium points are given in Table 3–2. If a γ value below 1 is achieved, the design criteria defined by the weighting functions are met. This is the case for both controllers for equilibrium points 2 and 3. For equilibrium points 1 and 4, the achieved value for γ is significantly larger than 1, indicating a possibly inferior controller performance. With the exception of the controller for architecture 2, equilibrium point 4, the poles of all controllers derived here have negative real parts for the continuous time representations. They are hence internally stable. After transformation into discrete time, the poles lie within the unit circle for controllers based on equilibrium points 2 and 3. Both controllers derived for equilibrium point 1 have an unstable pole outside the unit circle in discrete time. The controller derived for equilibrium point 4 is stable in discrete time for architecture 1 and unstable for architecture 2.

Internally unstable controllers are rarely suitable for control tasks subject to equipment deficiencies. Delays in the control chain or temporary loss of sensor readings can upset the controller and subsequently the plant beyond the ability to recover. The internally unstable controllers are therefore discarded.

3.2.2 Thrust allocation algorithm

To close the control loop, the desired propulsion forces $\boldsymbol{f}_{p,c}$ and moments $\boldsymbol{n}_{p,c}$ computed by the H_{∞} controller have to be translated into thrust values F_i and tilt angle values μ_i for each of the four thrusters on the airship.

The relation between the individual thruster forces F_i and angles μ_i and the total thruster forces \boldsymbol{f}_p and moments \boldsymbol{n}_p is given by the nonlinear equations (B.13) and (B.14).

These equations can be linearized employing the coordinate transformation

$$F_{i,x} = F_i \sin \mu_i$$
 and $F_{i,z} = -F_i \cos \mu_i$ (3.19)

giving a linear system of five equation with 8 variables.

$$\begin{bmatrix} f_{p,x} \\ f_{p,z} \\ n_{p,x} \\ n_{p,y} \\ n_{p,z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & r_{T1,y} & 0 & r_{T2,y} & 0 & r_{T3,y} & 0 & r_{T4,y} \\ r_{T1,z} & -r_{T1,x} & r_{T2,z} & -r_{T2,x} & r_{T3,z} & -r_{T3,x} & r_{T4,z} & -r_{T4,x} \\ -r_{T1,y} & 0 & -r_{T2,y} & 0 & -r_{T3,y} & 0 & -r_{T4,y} & 0 \end{bmatrix} \begin{bmatrix} F_{1,x} \\ F_{1,z} \\ F_{2,x} \\ \vdots \\ F_{4,x} \\ F_{4,z} \end{bmatrix}.$$

$$(3.20)$$

Hence, the airship actuation has 8 degrees of freedom to produce 5 desired generalized thrust forces. For the unconstrained case, this leads to an infinite number of solutions. However, the thrusters are constrained with respect to the permissible angles μ_i , the achievable thrust forces F_i , and the rate of change of these values, as described in Appendix B.3. As the rotational speed of the thruster motors cannot be measured in the vehicle studied here, the thruster dynamics are not explicitly considered in the optimization algorithm.

The servo dynamics model is explicitly considered in the algorithm and consists of a rate limiter that limits the tilting speed to a maximum of $\dot{\mu}_{max} = \pm 287 \text{deg/s}$.

The dynamical constraints of the thrusters can be so strict, that it may not be possible to find a feasible solution for the desired forces f_p and moments n_p for every controller input. Hence, it appears sensible to design the thrust allocation problem as an optimization problem that minimizes the cost function

$$\min Q = w_n \sum_{k=x,y,z} (n_{p,k} - n_{p,k,c})^2 + w_f \sum_{k=x,z} (f_{p,k} - f_{p,k,c})^2 + w_{Ft} \sum_{i=1}^4 F_i^2.$$
(3.21)

The first two terms of this function penalize the difference between the desired and the generated thruster forces and moments. The weighing factors w_n and w_f allow separate weights to be used on deviations in the moments and forces to prioritize attitude control or velocity control. The last term penalizes solutions with large thruster forces. This ensures that the algorithm will search for a solution with minimum thrust effort. To ensure that this term does not overly deteriorate the tracking quality, the term w_{Ft} should be 2 or more orders of magnitude less than w_n and w_f . That ensures that errors in the control forces and moments are penalized significantly larger than the control effort itself.

The constraints on the minimization of Q are defined to consider the maximum permissible thrust of 11N as well as the dynamics of the tilting servos. Also, the tilt angle cannot exceed a range from -90° (backward thrust) to $+90^{\circ}$ (forward thrust). As the controller is implemented in a discrete fashion, the servo dynamics model gives for each of the thrusters the restriction

$$\Delta \mu_{i,max} = \dot{\mu}_{max} h \tag{3.22}$$

on the change of tilt angle per time step, with h being the sampling time of the controller.

The two restrictions on the upper and lower bound of the tilt angle are limited to remain within the servo range of motion of ± 90 degrees. Together with the maximum permissible thrust of 11N, these constraints give a triangular area for each thruster within which the solution of (3.21) must remain. This is illustrated in Figure 3–7.



Figure 3–7: Constraints of the forces $F_{x,i}$ and $F_{z,i}$ for the *i*-th thruster.

Using the linear description of the thruster forces and moments, the bounds of the triangles can be described using three inequalities of the type

$$c_{i,1}F_{i,x} + c_{i,2}F_{i,z} + c_{i,3} \ge 0 \tag{3.23}$$

for each thruster.

Using the linear representation (3.20) of the thruster forces and moments also puts (3.21) into quadratic form, thereby allowing the minimization of (3.21) under the constraints (3.23) to be solved using the interior-reflective Newton algorithm described in [77]. In the simulation, this is done using the MatLab function quadprog.

3.3 Closed-loop simulations

To evaluate the controller performance, closed-loop simulations have been conducted with the aim of verifying controller performance in the presence of sensor noise, delays in the actuation chain as well as parametric uncertainty.

To simulate parametric uncertainty, the nominal airship parameters are modified for the simulation. An overview of the modified parameters is given in Table 3–3. The magnitude of the parameter modification reflects the precision within which each parameter can be determined.

Parameter	Simulation value	Nominal values		
CG position [mm] c	$[82.0 \ 30 \ 166.5]^T$	$[32.0 \ 0 \ 116.5]^T$		
Mass m [kg]	6.66	6.35		
Inertia matrix \boldsymbol{J} [kgm ²]	diag(3.19, 8.01, 9.10)	diag(3.04, 7.63, 8.67)		
Volume $V [m^3]$	4.622	4.765		
Added mass in x [kg]	0.6073	0.639		
Added mass in y and z [kg]	4.926	4.692		
Added inertia around y and z [kgm ²]	3.555	3.385		

Table 3–3: Nominal airship parameters and parameters used in simulation.

The trajectory used during the simulations has been chosen such that it covers the entire operational envelope of the vehicle. The airship motion is controlled purely based on attitude and body-frame forward velocity, so wind drift is not compensated. The motion sequence consists of vertical ascent and descent, straight legs and two 180 degree turns to the right. During the turns, the desired roll angle is set to 15 degrees into the turn to direct some of the thrust from the thrusters towards the turn centre, thus facilitating the turning motion. Halfway through each turn, the desired pitch is set to 15 degrees pitch up for turn 1 and pitch down for turn 2 to test motion about all 3 axes simultaneously.

The reference trajectory is shown in Figures 3–8 and 3–9. The desired velocity in the body z direction is -0.8m/s during the initial ascent. During the descent at the end of the flight, it is 0.5m/s. For the remainder of the flight, the desired velocity in the body z direction is 0 and the desired forward velocity is 2m/s.



Figure 3–8: Airship reference path during the simulations.

Figure 3–9: Reference height profile during the simulations.

3.3.1 Controller performance under idealized conditions

In order to verify the correct controller synthesis, a first set of simulations is run with idealized conditions. For this simulation set, the wind speed is set to zero and the airship parameters used in the model correspond to the nominal parameters used for controller synthesis. Also, the thruster dynamics model is bypassed, giving perfect actuation. However, due to the optimization algorithm used in the controller it is ensured that the servo tilt speed respects the maximum limit and the maximum thrust value cannot exceed 11N for each thruster. Sensor noise is still present in the simulation constituting a small external disturbance.

The best performance is achieved for the controllers designed for equilibirum point 3 of Table 3–1, corresponding to vertical flight. The remaining controllers give unusable results. Equilibrium point 3 has the poles with the slowest frequencies. Consequently, the controller derived from this equilibrium point exhibits slower poles than the controllers synthesized using any of the other equilibrium points. It is therefore more robust to parametric uncertainty and other unmodelled effects. The airship remains stable with the controller based on equilibrium point 2 using architecture 1. However, it does not track the desired pitch or yaw angle or the vertical velocity satisfactorily. The closed-loop system using controller architecture 1 based on equilibrium points 1 and 4 is unstable.

The closed-loop system with controller architecture 2 based on equilibrium points 1, 2, and 4 remains stable in roll and pitch but enters a permanent yaw rotation. Velocity tracking is unsatisfactory, which may be a consequence of the rotation around the yaw axis.

The overall performance of the individual controllers is consistent with the γ values achieved during controller synthesis, with the lowest γ values giving the best closed-loop performance. Interestingly, the controllers designed for equilibrium point 3 also perform well at operating conditions closer to equilibrium points 1, 2, and 4.

To compare the two different controller architectures, the closed-loop performance of two controllers based on equilibrium point 3 will be analyzed in more detail. The evolution of the airship attitude during the simulation is shown in Figures 3–10 to 3–13 with the plots corresponding to control architecture 1 on the left side. The yaw tracking performance is almost ideal for both controllers. The tracking of pitch and roll is also very good for control architecture 1. Roll and pitch tracking for architecture 2 is less good, but still acceptable, with the deviations from the desired being in the range of 5 degrees or less.

A much more significant difference in control performance is apparent in the velocity tracking shown in Figures 3–14 and 3–15. The velocity tracking performance is very good for control architecture 1. For control architecture 2, the performance is



Figure 3–10: Roll and pitch evolution Figure 3–12: Roll and pitch evolution during simulation with ideal conditions using control architecture 1.



Figure 3–11: Yaw evolution during simulation with ideal conditions using control architecture 1.



during simulation with ideal conditions using control architecture 2.



Figure 3–13: Yaw evolution during simulation with ideal conditions using control architecture 2.





Figure 3–14: Forward and upward velocity evolution during simulation with ideal conditions using control architecture 1.

Figure 3–15: Forward and upward velocity evolution during simulation with ideal conditions using control architecture 2.

less than satisfactory with the vertical velocity being permanently below the desired values and the forward velocity exhibiting large short-term deviations.

An even more important performance difference are the control commands given by the controllers, as shown in Figures 3–16 and 3–17. Using control architecture 1, the control task is achieved with overall low thrust values on all 4 thrusters and an acceptable level of modulations. Using control architecture 2 results in large, noisy control commands which will not be feasible by the actuators and will also reduce battery endurance.

Based on these simulations, it appears sensible to discard the controllers synthesized using equilibrium points 1, 2, and 4 and proceed only with the controllers





Figure 3–16: Control commands during simulation with ideal conditions using control architecture 1.

Figure 3–17: Control commands during simulation with ideal conditions using control architecture 2.

designed using equilibrium point 3. Although for these controllers control architecture 2 shows worse performance than architecture 1, the impact of wind, thruster dynamics and parametric uncertainty will be investigated for both.

3.3.2 Controller performance with disturbances

In order to investigate control performance in a more realistic scenario, the previously conducted simulations are run again with the following changes. The new simulations contain a low wind with an average speed of about 0.5m/s blowing from the north-east. The thrust dynamics model is used, but the transportation delay has for now been deactivated. The airship parameters used in the simulation correspond to the perturbed parameters given in Table 3–3 rather than the nominal parameters used for controller design. With the exception of the deactived transportation delay, this corresponds to the most realistic model of the Quanser MkII available at this time.



Figure 3–18: Wind speed at the airship position during the simulation with external disturbances.

The wind speed at the position of the airship is shown in Figure 3–18. The evolution of the airship attitude under these conditions is shown in Figures 3–19 to 3–22. Besides larger disturbances due to the wind gusts, the key difference to the flights under idealized conditions is the appearance of steady-state errors. This is most prominent in the roll angle tracking. Due to the parametric uncertainty, the CG is slightly right of the symmetry plane of the aircraft creating a permanent roll moment to the right. As the control algorithm does not contain any elements to compensate for steady-state errors, such as integrators, this roll moment results in a steady-state error of about 7 degrees for the roll angle.

Pitch and yaw angle tracking are satisfactory with both controllers. The control architecture 2 gives stronger oscillations than architecture 1.

The velocity tracking performance is shown in Figures 3–23 and 3–24. Similarly to the simulations with idealized conditions, the velocity tracking performance is significantly better for control architecture 1. Except for a steady-state error of about



Figure 3–19: Roll and pitch evolution Figure 3–21: Roll and pitch evolution during simulation with external disturbances using control architecture 1.



Figure 3–20: Yaw evolution during simulation with external disturbances using control architecture 1.



during simulation with external disturbances using control architecture 2.



Figure 3–22: Yaw evolution during simulation with external disturbances using control architecture 2.





Figure 3–23: Forward and upward velocity evolution during simulation with external disturbances using control architecture 1.

Figure 3–24: Forward and upward velocity evolution during simulation with external disturbances using control architecture 2.

0.1m/s in the vertical velocity, the velocity tracking with controller architecture 1 is almost perfect. The steady state error is due to the increase in airship weight with the modified simulation parameters. This error means that the airship is slightly losing height throughout the simulation and it impacts the ground prematurely at about t = 115s. At this point the velocities are set to zero and the attitude is frozen, based on a simplified ground interaction model.

With control architecture 2, the tracking of the forward velocity is still acceptable, although it exhibits larger oscillations. The steady-state error on the vertical velocity tracking is larger than for architecture 1 and the oscillations in the vertical velocity tracking are significant.

The thrust inputs generated with each control architecture are given in Figures 3–25 and 3–26. These plots represent the actual thrust generated by the thrusters,





Figure 3–25: Control commands during simulation with external disturbances using control architecture 1.

Figure 3–26: Control commands during simulation with external disturbances using control architecture 2.

that is, they are recorded downstream of the thruster model. The thruster model consists of a first order low-pass filter, therefore the thruster model eliminates some of the noise visible for the flights with ideal conditions.

This test confirms the findings of the simulations with ideal conditions. The level of oscillations on the thrust inputs is significantly better for control architecture 1 than architecture 2. The level of oscillations for architecture 2 is beyond acceptable levels confirming that this controller is not useable for experimental flight.

The performance of the controller using architecture 1 in combination with linearization point 3 as established in these simulations would be good enough to warrant flight testing these controllers. The steady-state error may be reduced by inserting a slow integrator on the control error at the input to the H_{∞} controller. However, the current simulations do not yet feature the time delay in the thruster model, and the GPS velocity signal is not filtered leading to a noiser signal but with less delay. The influence of adding these parameters to the simulation will be discussed in the next section.

3.3.3 Performance impact of time delays

The previous simulations did not consider the time delay of 80ms for the thruster forces and 50ms for the tilt servoes as described in Appendix B.3. Adding this delay renders the closed-loop system with the controllers discussed in the previous section unstable. The same happens, when the low-pass filter on the GPS velocity measurements as described in Appendix D is activated. The control system is also unstable for increased forward velocities of 3m/s.

In an attempt to make the closed-loop system stable in the presence of the thruster delay, the controller synthesis was repeated with the value for τ_p in (3.7) increased from 0.2s to 0.4s. While this modification allows the closed-loop system to remain stable even in the presence of the time delay the control performance itself is not satisfactory. The controller exhibits large thrust input oscillations reaching amplitudes of 10N. Attitude and velocity control exhibit fast oscillations with high amplitudes of almost 1m/s for speed control and up to 10 degrees for attitude control.

Another approach to include the thruster delays in the controller design is to use the Padé approximation. This approximation uses a high order rational transfer function to simulate the effect of a time-delay. This approach was not undertaken for two reasons. Firstly, the value of the time delay for the thrusters is not precisely known and the uncertainty on the values assumed in the simulation is very high. Secondly, the nonlinear controller shown in the next chapter provided a significantly higher level of robustness than the H_{∞} has exhibited at this point. That controller was therefore deemed superior and further research and development effort focussed on that control technique.

CHAPTER 4 Nonlinear low-level controller design

The vehicle studied in this work exhibits strongly nonlinear dynamics as shown in the previous chapter. The use of nonlinear control algorithms allows the design of a control law that is capable of dealing with the complete motion envelope of the vehicle while ensuring a guarantee of stability for all operating points.

For this work, Lyapunov based control techniques have been chosen. These provide more freedom in the control law design than input-output linearization, and in contrast to sliding mode control, only one control law is required for the entire flight envelope. Backstepping is an extension of Lyapunov based control to deal with strict feedback systems. It will be used here for attitude control, whereas regular Lyapunov control will be used for velocity control. Integral terms for both the attitude and the velocity controller will ensure a convergence of the steady-state tracking error to zero despite possible errors in the airship model parameters.

4.1 Design of an integral backstepping/Lyapunov low-level controller

The nonlinear low-level controller consists of multiple separate components as shown in figure 4–1. The first component is the combined integral backstepping and integral Lyapunov controller that calculates the desired thruster forces $f_{p,c}$ and moments $n_{p,c}$ to track the desired attitude and velocity. These consist of two forces and three moments, as the thruster layout does not allow for thrust in the body y direction. The second component is the thrust allocation algorithm described in

Section 3.2.2. This algorithm transforms the total desired thruster forces $\boldsymbol{f}_{p,c}$ and moments $\boldsymbol{n}_{p,c}$ into the individual thruster forces F_i and tilt angles μ_i .



Figure 4–1: Block diagram of the proposed combined Lyapunov/Backstepping Controller.

The combined integral backstepping and integral Lyapunov controller can be split into three parts. The integral backstepping controller calculates the desired angular accelerations $\boldsymbol{\xi}_c$ to perform attitude tracking, while the integral Lyapunov controller calculates the desired translational accelerations \boldsymbol{a}_c to track the desired velocities. To account for the coupling of translational and rotational motion, the generalized apparent mass matrix $\bar{\boldsymbol{M}}_a$ will be used to transform $\boldsymbol{\xi}_c$ and \boldsymbol{a}_c into the desired net forces and moments acting on the airship. The feedforward part then takes the relevant terms from (2.26), as well as (B.2) and (B.3) to calculate the desired thruster forces $\boldsymbol{f}_{p,c}$ and moments $\boldsymbol{n}_{p,c}$.

The viscous forces \boldsymbol{f}_v and moments \boldsymbol{n}_v are neglected for the controller design, because their magnitude and direction depend on parameters that are difficult to determine precisely. Also, a magnitude analysis has shown that the longitudinal
viscous forces and all viscous moments are relatively small compared to other terms in the equations of motion. As viscous forces represent a dissipative term, this simplification has no adverse impact on stability. If the wind speed is not known, the wind speed vector $\boldsymbol{v}_{w,I}$ in the respective terms taken from (2.26) is set to zero.

4.1.1 Integral backstepping controller design

Backstepping control is based on Lyapunov theory and can be used for strict feedback systems, i.e., for systems of the form

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{f}_1(\boldsymbol{x}_1) + \boldsymbol{g}_1(\boldsymbol{x}_1)\boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 &= \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{g}_2(\boldsymbol{x}_2)\boldsymbol{x}_3 \\ &\vdots \\ \dot{\boldsymbol{x}}_n &= \boldsymbol{f}_n(\boldsymbol{x}_n) + \boldsymbol{g}_n(\boldsymbol{x}_n)\boldsymbol{u}. \end{aligned}$$
(4.1)

When employing backstepping control, each of these equations is stabilized separately starting at the first equation with the virtual control \boldsymbol{x}_2 and then tracing the way down until the *n*-th equation is stabilized using the system input \boldsymbol{u} . A detailed description of this control technique can be found in [78].

As the thruster dynamics are neglected at this stage, the attitude dynamics of the ALTAV Quanser MkII can be represented as a strict feedback system with n = 2. In this case, the state vector \boldsymbol{x}_1 in (4.1) corresponds to the attitude quaternion \boldsymbol{q} of the airship and the state vector \boldsymbol{x}_2 corresponds to the angular rate vector $\boldsymbol{\omega}$. The first equation of the strict feedback system (4.1) is given by the kinematics equation (2.2). The relevant airship equation of motion for the second equation of (4.1) is (2.26). However, due to the coupling of the translational and rotational motion described in that equation, the equation will for now be reduced to $\dot{\omega} = \boldsymbol{\xi}$ (See (4.10)). The neglected terms of (2.26) will be dealt with in section 4.1.4.

Designing the controller to compute the desired angular acceleration vector $\boldsymbol{\xi}_c$ as input \boldsymbol{u} to the strict feedback system (4.1) is a two-step process. In the first step, a Lyapunov control law is designed to make the current attitude quaternion \boldsymbol{q} track the desired attitude quaternion \boldsymbol{q}_d using the angular rate vector $\boldsymbol{\omega}$ as virtual control. This tracking problem can be converted into a stabilization problem by introducing the virtual state

$$\boldsymbol{z}_1 = \boldsymbol{q} - \boldsymbol{q}_d \tag{4.2}$$

which is driven to zero.

In the second step, the Control Lyapunov Function (CLF) from the first step will be extended to include the virtual state

$$\boldsymbol{z}_2 = \boldsymbol{\omega} - \boldsymbol{\omega}_c \tag{4.3}$$

which describes the difference between commanded and actual angular rates. Using this extended CLF, a control law will be designed to drive the virtual state z_2 to zero, while ensuring that the current attitude quaternion q will converge to the desired quaternion q_d .

In the first step, the aim is to stabilize the virtual state \boldsymbol{z}_1 at zero. Hence the function

$$V_1 = \boldsymbol{z}_1^2 \ge 0 \tag{4.4}$$

appears a suitable candidate Control Lyapunov Function. It is zero for $\boldsymbol{q} = \boldsymbol{q}_d$ and greater than zero otherwise. Here, it is noted that the dynamics of the virtual state z_1 can only be brought into the form of the first equation in (4.1), if q_d is constant, i.e. the Lyapunov control law guarantees global asymptotic stability only for constant desired attitude q_d . However, the simulations in Section 4.2 show that for sufficiently slow changes of the desired attitude, the controller achieves good tracking. Despite this limitation of the stability proof, the derivatives of q_d that occur during the control law derivation are retained, as they significantly improve attitude tracking when the desired attitude changes.

The derivative of this CLF is

$$\dot{V}_1 = 2\boldsymbol{z}_1^T \dot{\boldsymbol{z}}_1 = 2\left(\boldsymbol{q} - \boldsymbol{q}_d\right)^T \left(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d\right).$$
(4.5)

Using the kinematics equation (2.2) yields

$$\dot{V}_1 = \left(\boldsymbol{q}^T - \boldsymbol{q}_d^T \right) \left[\boldsymbol{Q}_{\omega} \boldsymbol{\omega} - 2 \dot{\boldsymbol{q}}_d \right].$$
 (4.6)

To ensure that $\dot{V} \leq 0$, it is straightforward to set

$$\boldsymbol{Q}_{\omega}\boldsymbol{\omega}_{c} = -c_{a1}\left(\boldsymbol{q}-\boldsymbol{q}_{d}\right) + 2\dot{\boldsymbol{q}}_{d} \tag{4.7}$$

$$\boldsymbol{\omega}_{c} = -c_{a1}\boldsymbol{Q}_{\omega}^{T}\left(\boldsymbol{q}-\boldsymbol{q}_{d}\right) + 2\boldsymbol{Q}_{\omega}^{T}\dot{\boldsymbol{q}}_{d}$$

$$(4.8)$$

with $c_{a1} > 0$ being a controller parameter that has been determined empirically, as described in Section 4.2.

Since $\boldsymbol{Q}_{\omega}^{T} \boldsymbol{q} = \boldsymbol{0}$, the virtual control law becomes

$$\boldsymbol{\omega}_c = \boldsymbol{Q}_{\omega}^T \left(c_{a1} \boldsymbol{q}_d + 2 \dot{\boldsymbol{q}}_d \right). \tag{4.9}$$

The second step of the backstepping controller design is now to ensure that the actual angular rates vector $\boldsymbol{\omega}$ follows the desired angular rates vector $\boldsymbol{\omega}_c$ defined by (4.9). This means that the controller has to drive the virtual state \boldsymbol{z}_2 as defined in (4.3) to zero.

Due to the coupling of rotational and translational motion shown in equation (2.26), we will use the virtual control

$$\boldsymbol{\xi}_c := \dot{\boldsymbol{\omega}}_c \tag{4.10}$$

for this step. As described above, the desired thrust inputs will be calculated based on equations (2.26), (B.2), and (B.3), as soon as the desired angular acceleration $\boldsymbol{\xi}_c$ and the desired translational accelerations \boldsymbol{a}_c are known.

To achieve this second requirement, the Control Lyapunov Function (4.4) will be extended to also drive the virtual state z_2 to zero:

$$V_2 = \boldsymbol{z}_1^2 + \frac{1}{2}\boldsymbol{z}_2^2 \ge 0 \tag{4.11}$$

Using (4.6) and (4.3), the derivative of this function is

$$\dot{V}_2 = \left(\boldsymbol{q}^T - \boldsymbol{q}_d^T\right) \left[-2\dot{\boldsymbol{q}}_d + \boldsymbol{Q}_\omega \boldsymbol{\omega}_c + \boldsymbol{Q}_\omega \boldsymbol{z}_2\right] + \boldsymbol{z}_2^T \dot{\boldsymbol{z}}_2.$$
(4.12)

Inserting equation (4.7) yields

$$\dot{V}_2 = -c_{a1} \left(\boldsymbol{q}^T - \boldsymbol{q}_d^T \right) \left(\boldsymbol{q} - \boldsymbol{q}_d \right) + \boldsymbol{z}_2^T \boldsymbol{Q}_{\omega}^T \left(\boldsymbol{q} - \boldsymbol{q}_d \right) + \boldsymbol{z}_2^T \dot{\boldsymbol{z}}_2.$$
(4.13)

Using (4.3) and (4.9), the derivative of the virtual state \boldsymbol{z} can be computed as

$$\dot{\boldsymbol{z}}_{2} = \dot{\boldsymbol{\omega}} - \boldsymbol{Q}_{\omega}^{T} \left(c_{a1} \dot{\boldsymbol{q}}_{d} + 2 \ddot{\boldsymbol{q}}_{d} \right) - \dot{\boldsymbol{Q}}_{\omega}^{T} \left(c_{a1} \boldsymbol{q}_{d} + 2 \dot{\boldsymbol{q}}_{d} \right)$$
(4.14)

This yields for the derivative of the extended CLF

$$\dot{V}_{2} = -c_{a1}||\boldsymbol{q} - \boldsymbol{q}_{d}||_{2}^{2} + \boldsymbol{z}_{2}^{T}\left(\dot{\boldsymbol{\omega}} + \boldsymbol{Q}_{\boldsymbol{\omega}}^{T}\left(\boldsymbol{q} - \boldsymbol{q}_{d} - c_{a1}\dot{\boldsymbol{q}}_{d} - 2\ddot{\boldsymbol{q}}_{d} - \dot{\boldsymbol{Q}}_{\boldsymbol{\omega}}^{T}\left(c_{a1}\boldsymbol{q}_{d} + 2\dot{\boldsymbol{q}}_{d}\right)\right)\right).$$

$$(4.15)$$

The easiest way to ensure global asymptotic stability would be to set the term multiplied by \boldsymbol{z}_2^T equal to $-c\boldsymbol{z}_2$:

$$-c_{a2}\boldsymbol{z}_{2} = \dot{\boldsymbol{\omega}}_{c} + \boldsymbol{Q}_{\omega}^{T} \left(\boldsymbol{q} - \boldsymbol{q}_{d} - c_{a1} \dot{\boldsymbol{q}}_{d} - 2 \ddot{\boldsymbol{q}}_{d} - \dot{\boldsymbol{Q}}_{\omega}^{T} \left(c_{a1} \boldsymbol{q}_{d} + 2 \dot{\boldsymbol{q}}_{d} \right) \right)$$
(4.16)

with $c_{a2} > 0$.

However, in the presence of modelling errors, such a control law would not ensure a steady-state control error of zero. It is shown in [79] that the addition of an integral term allows to eliminate the steady-state error without jeopardizing stability for sufficiently small integrator gains.

Under consideration of (4.10), this leads to the equation

$$-c_{a2}\boldsymbol{z}_{2} - \lambda_{a1} \int_{0}^{t} \boldsymbol{z}_{2} d\tau = \boldsymbol{\xi}_{c} + \boldsymbol{Q}_{\omega}^{T} \left(\boldsymbol{q} - \boldsymbol{q}_{d} - c_{a1} \dot{\boldsymbol{q}}_{d} - 2 \ddot{\boldsymbol{q}}_{d} - \dot{\boldsymbol{Q}}_{\omega}^{T} \left(c_{a1} \boldsymbol{q}_{d} + 2 \dot{\boldsymbol{q}}_{d} \right) \right)$$
(4.17)

with $c_{a2} > 0$ and $\lambda_{a1} > 0$ being controller parameters whose values will be determined as described in Section 4.2.

Substituting \boldsymbol{z}_2 by using equation (4.3) and reorganizing the equation gives an expression for the desired angular accelerations

$$\boldsymbol{\xi}_{c} = -c_{a2}\boldsymbol{\omega} + \boldsymbol{Q}_{\omega}^{T} \left[(1 + c_{a1}c_{a2}) \, \boldsymbol{q}_{d} + (c_{a1} + 2c_{a2}) \, \dot{\boldsymbol{q}}_{d} + 2\ddot{\boldsymbol{q}}_{d} \right] + \dots \\ \dots + \dot{\boldsymbol{Q}}_{\omega}^{T} \left[c_{a1}\boldsymbol{q}_{d} + 2\dot{\boldsymbol{q}}_{d} \right] - \lambda_{a1} \int_{0}^{t} (\boldsymbol{\omega} - \boldsymbol{Q}_{\omega}^{T}(c_{a1}\boldsymbol{q}_{d} + 2\dot{\boldsymbol{q}}_{d})) d\tau$$

$$(4.18)$$

4.1.2 Solution to the ambiguity of the quaternion attitude representation

The same attitude can be represented with two quaternions that have opposite signs, depending on the desired direction of rotation to reach this attitude. The controller equation (4.18) tries to drive the actual attitude quaternion \boldsymbol{q} to be identical to the desired quaternion \boldsymbol{q}_d based on equation (4.2).

It is hence crucial to define the desired quaternion such that the rotation required to get from \boldsymbol{q} to \boldsymbol{q}_d is minimal. Since quaternions describe a four dimensional unit ball in which opposite points represent the same attitude, the choice of the correct representation of \boldsymbol{q}_d must be done such that is describes the point closest to the current attitude representation on the unit ball.

To verify the distance of the current representation of q_d from the actual attitude q, the controller calculates the norm

$$N^2 = ||\boldsymbol{q} - \boldsymbol{q}_d||_2^2 \tag{4.19}$$

As can be seen in figure 4–2, the current representation of \boldsymbol{q}_d is the closest possible representation to the actual attitude \boldsymbol{q} , if $N < \sqrt{2}$. Respectively, the correct representation of the desired attitude quaternion is determined via

$$\boldsymbol{q}_{d} := \begin{cases} \boldsymbol{q}_{d} & \text{if } N^{2} \leq 2 \\ -\boldsymbol{q}_{d} & \text{if } N^{2} > 2. \end{cases}$$

$$(4.20)$$

This revised desired quaternion is used for the computation of the control law (4.18).



Figure 4–2: Determination of the correct representation for the desired attitude quaternion q_d .

4.1.3 Lyapunov based velocity controller

The velocity can be controlled directly via the acceleration of the airship. However, the thrusters are mounted such that they can generate forces in x and z direction but not in the lateral y direction. This means that the velocity in the body y direction cannot be controlled directly. The high-level controller will have to calculate the desired attitudes and speeds in x and z direction such that the airship follows the desired path despite the inability to directly exert control over the speed in ydirection.

To control the velocity in x and z direction, the virtual state

$$\boldsymbol{z}_{v} = \begin{bmatrix} u - u_{d} \\ w - w_{d} \end{bmatrix}$$
(4.21)

is introduced. This allows to introduce the Control Lyapunov Function

$$V_v = \frac{1}{2} \boldsymbol{z}_v^T \boldsymbol{z}_v. \tag{4.22}$$

The derivative with respect to time of this CLF is

$$\dot{V}_{v} = \begin{bmatrix} u - u_{d} \\ w - w_{d} \end{bmatrix}^{T} \begin{bmatrix} \dot{u} - \dot{u}_{d} \\ \dot{w} - \dot{w}_{d} \end{bmatrix}$$
(4.23)

Analogously to the attitude controller, an integral term will be included in the control law to ensure that the long-term tracking error will converge to zero. To ensure global asymptotic stability, \dot{V}_v must be less than zero. That means that the signs of the corresponding components of each of the factors of (4.23) have to be opposite. Instead of setting $\dot{u} - \dot{u}_d = -c_{v1}(u - u_d)$, the arctan function can also be used. This has the advantage that the maximum control input is limited for large velocity differences without reducing the control characteristics in the vicinity of the desired set-point. The stability criterion is maintained, because the sign of the arctan-function is equal to the sign of its argument.

This yields the following desired accelerations

$$\begin{bmatrix} a_{x,c} \\ a_{z,c} \end{bmatrix} = \begin{bmatrix} \dot{u}_d \\ \dot{w}_d \end{bmatrix} - c_{v1} \begin{bmatrix} \tan^{-1} \left(c_{v2} \left(u - u_d \right) \right) \\ \tan^{-1} \left(c_{v2} \left(w - w_d \right) \right) \end{bmatrix} - \lambda_{v1} \int_0^t \begin{bmatrix} u - u_d \\ w - w_d \end{bmatrix} d\tau \quad (4.24)$$

with $c_{v1} > 0$, $c_{v2} > 0$, and $\lambda_{v1} > 0$ being controller parameters whose values will be determined empirically, as described in Section 4.2. The parameter c_{v1} limits the maximum desired acceleration to $|a_c| < c_{v1}\pi/2$. The slope of the desired accelerations at a velocity error of 0 is $c_{v1}c_{v2}$. In order to avoid undesirably large accelerations and integrator wind-up during sharply-varying inputs to the desired velocity values, the desired inputs are rate limited to 0.4m/s^2 . The rate limiter also ensures that the derivatives of the desired velocities \dot{u}_d and \dot{w}_d as used in equation (4.24) exist and are bounded to the value of the rate limiter. The rate limiter will not prevent wind up due to the unknown effect of the wind. However, such a wind-up has never been observed in experiment or simulation.

4.1.4 Calculation of the desired thruster forces and moments

Now that the desired angular and translation accelerations are known, the corresponding thruster forces $f_{p,c}$ and moments $n_{p,c}$ can be determined in order to ensure that

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\xi}_c \quad \text{and} \quad (4.25)$$

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} a_{x,c} \\ a_{z,c} \end{bmatrix}. \quad (4.26)$$

These calculations are based on equations (2.26), (B.2), and (B.3). However, this raises one last issue that needs to be resolved first. The controller has not calculated a value for the acceleration along the y axis \dot{v} , because the thrusters cannot exert thrust in this direction. Due to the coupling of motion described in equation (2.26), it will be necessary to compute an estimate for the acceleration \dot{v} in order to correctly consider the coupling of this acceleration into the rotational motion. In other words, we need to find a value for \dot{v} for which the desired thruster forces in y-direction $f_{p,c,y}$ vanish. To estimate the forces acting in the y direction, we will assume that the viscous forces are negligible, and that the thrusters act perfectly in the x-z plane. This gives an estimate for \dot{v} of

$$\dot{v}_{E} \approx \frac{1}{m_{ay}} \Big[m_{az} wp - m_{ax} ur + m((p^{2} + r^{2})c_{y} - pqc_{x} - rqc_{z}) + (m - \rho V)g_{y} + \dots \\ \dots + (m_{Dax} - m_{Daz})(ru_{w} + pw_{w}) + \begin{bmatrix} 0 & m_{Day} & 0 \end{bmatrix} \mathbf{R} \dot{\mathbf{v}}_{w,I} \Big]$$

$$(4.27)$$

with

$$g_y = g\cos\theta\sin\phi. \tag{4.28}$$

The desired net forces and moments acting on the airship can now be calculated via

$$\begin{bmatrix} \boldsymbol{f}_{c} \\ \boldsymbol{n}_{c} \end{bmatrix} = \bar{\boldsymbol{M}}_{a} \begin{bmatrix} a_{x,c} \\ \dot{\boldsymbol{v}}_{E} \\ a_{z,c} \\ \boldsymbol{\xi}_{c} \end{bmatrix}$$
(4.29)

The translation of these into desired thruster forces and moments is done simply by solving equations (2.26)-(2.28), (B.2), and (B.3) for \boldsymbol{f}_p and \boldsymbol{n}_p . Again neglecting the viscous forces, this yields

$$\begin{bmatrix} f_{p,c,x} \\ f_{p,c,z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \boldsymbol{f}_c + \boldsymbol{\omega} \times \boldsymbol{M}_a \boldsymbol{v} - \boldsymbol{\omega} \times \boldsymbol{m} \boldsymbol{C}^{\times} \boldsymbol{\omega} - \boldsymbol{R}(\boldsymbol{m} - \rho \boldsymbol{V}) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \dots \\ \dots + \boldsymbol{M}_{Da}(\boldsymbol{\omega} \times \boldsymbol{v}_w) - \boldsymbol{\omega} \times \boldsymbol{M}_{Da} \boldsymbol{v}_w - \boldsymbol{M}_{Da} \boldsymbol{R} \dot{\boldsymbol{v}}_{w,I} \end{pmatrix}$$

$$(4.30)$$

for the desired thruster forces and

$$\boldsymbol{n}_{p,c} = \boldsymbol{n}_c + \boldsymbol{\omega} \times \boldsymbol{J}_a \boldsymbol{\omega} + m \boldsymbol{C}^{\times} (\boldsymbol{\omega} \times \boldsymbol{v}) + (\boldsymbol{v} - \boldsymbol{v}_w) \times \boldsymbol{M}_{Da} (\boldsymbol{v} - \boldsymbol{v}_w) - \boldsymbol{c} \times \boldsymbol{R} m \begin{bmatrix} 0\\0\\g \end{bmatrix}$$
(4.31)

for the desired thruster moments.

If known, the terms related to the wind speed v_w and $\dot{v}_{w,I}$ can be used to make the controller consider the wind explicitly. If either of the terms are unknown, they can be set to zero and the integral terms in the controller equations will compensate for the steady state effect of the wind. The unknown wind effect then constitutes a bounded disturbance.

Based on the total commanded thruster forces $\boldsymbol{f}_{p,c}$ and moments $\boldsymbol{n}_{p,c}$, the thrust values F_i and tilt angles μ_i for each individual thruster are determined using the quadratic optimization algorithm described in Section 3.2.2.

4.2 Controller performance evaluation in closed-loop simulation

The performance of the nonlinear low-level controller will be investigated using the same trajectory as in Section 3.3. All the simulations presented here include thruster dynamics with the time delay described in Appendix B.3 as well as the discrepancy between controller and airship parameters as described in Section 3.3. The simulations are performed for a variety of wind conditions.

The controller and the thrust allocation parameters that provide the best controller performance have been determined iteratively in a series of simulations. While higher parameter values hold the potential for improved tracking quality, the simulation showed that too high values lead to large oscillations or instability because the thruster dynamics are not explicitly considered in the controller design. Table 4–1 shows the parameters determined in the iterative process.

Parameter	c_{a1}	c_{a2}	λ_{a1}	c_{v1}	c_{v2}	λ_{v1}	w_n	w_f	w_{Ft}
Value	$1\frac{1}{s}$	$3\frac{1}{s}$	$0.9\frac{1}{s^2}$	$2.5\frac{s}{m}$	$0.15 \frac{m}{s^2}$	$0.1\frac{1}{s^2}$	$1\frac{1}{N^2m^2}$	$0.2\frac{1}{N^2}$	$0.01\frac{1}{N^2}$

Table 4–1: Controller and thrust allocation parameters used in the simulations.

The first simulation presented here is conducted with a wind speed of zero, therefore showing the controller performance in the absence of external disturbances. To compare the controller performance, this scenario is also simulated using the PID controller taken from [75] and described in Section 2.3.

The airship attitude evolution for these test cases is shown in figures 4–3 to 4–6. With both controllers, upon lift-off the airship exhibits a roll to the right by about 10 degrees and a pitch down of about 5 degrees. This is a consequence of the parameter modification shown in table 3–3. These modifications include a shift of the airship CG forward and right. As the nonlinear controller works with the nominal and not the modified parameters, this change is initially not accounted for in the control law. The integral term in the controller manages to eliminate this error over a period of about 15 seconds.

Analogously, the integral terms of the PID controller have the task to compensate for the pitch and roll moment due to the CG location. The PID controller eliminates this steady-state error in a similar timeframe as the nonlinear controller.

Besides that, the performance of the nonlinear controller is significantly better than that of the PID controller. The nonlinear controller provides almost perfect



Figure 4–3: Roll and pitch evolution during simulation with zero wind and PID control.



Figure 4–4: Yaw evolution during simulation with zero wind and PID control.



Figure 4–5: Roll and pitch evolution during simulation with zero wind and nonlinear control.



Figure 4–6: Yaw evolution during simulation with zero wind and nonlinear control.

tracking for the yaw angle. Apart from the initial roll and pitch deviation, the controller tracks the desired roll and pitch values with a maximum error of 3 degrees.

Under PID control, the roll and pitch angles exhibit a high-frequency oscillation with an amplitude of about 2 degrees. The roll angle is tracked reasonably well, the pitch angle exhibits deviations up to 5 degrees. The yaw angle exhibits a damped low-frequency oscillation at the end of each turn with an initial amplitude in the range of 20 degrees.

Comparing these results with the attitude control performance of the H_{∞} controller, shown in figures 3–19 to 3–22, both the PID and the nonlinear controller provide the clearly superior yaw control performance. Apart from the lack of steadystate error compensation, the roll and pitch control performance of the H_{∞} controller is somewhere between the performance of the PID controller and that of the nonlinear controller.

The superiority of the nonlinear controller is even more apparent in the velocity control performance depicted in figures 4–7 and 4–8. Despite the complete absence of external disturbances, the PID controller tracks the forward velocity with error peaks of approximately 1m/s. The vertical velocity is controlled better with the error peaks being mostly in the range of 0.2m/s.

The nonlinear controller tracks the desired vertical velocity with only minor deviations for most of the simulation. The little peak at the beginning of the simulation is again an effect of the modified airship parameters. The airship has a greater mass than given in the controller parameters. Hence, the integrative term in the velocity





Figure 4–7: Forward and upward velocity evolution during simulation with zero wind and PID control.

Figure 4–8: Forward and upward velocity evolution during simulation with zero wind and nonlinear control.

controller has to compensate for this change, which takes a bit of time. The desired vertical velocity is reached after about 15s and is subsequently tracked well.

The forward velocity is tracked with four small deviations of about 0.2m/s. These deviations are coincident with commanded attitude changes and are well attenuated by the controller.

Taking the velocity control performance of the H_{∞} controller shown in figures 3– 23 and 3–24 as comparison, shows that the H_{∞} controller with architecture 1 provides the superior velocity control performance. It tracks both the vertical and the forward velocity with minimal deviations from the target value. However, it needs to be kept in mind that this simulation does not feature the delay in the thruster actuation, and that the H_{∞} controller becomes unstable in the presence of that delay.



Figure 4–9: Control commands during simulation with zero wind and PID control.

Figure 4–10: Control commands during simulation with zero wind and nonlinear control.

The evolution of the thruster variables shown in figures 4–9 and 4–10 makes the difference between the PID and the nonlinear controller even clearer. The PID controller gives strong oscillating thrust commands whereas the nonlinear controller commands very little variation in thrust F_i . The evolution of the tilt angle μ_i also exhibits large oscillations under PID control. Under NL control there are also some oscillations of the tilt angle, but these are much smaller in magnitude. The oscillations are a result of the sensor noise, especially the noise on the angular rates ω .

Comparing the thruster inputs of these controllers with the inputs generated by the H_{∞} controllers, shown in figures 3–25 and 3–26, places the performance of the H_{∞} controller again somewhere between that of the PID and the nonlinear controller.

Two additional simulations are presented here to investigate the impact of wind on the controller performance. The first simulation features wind conditions similar



Figure 4–11: Wind conditions during the simulation with an average wind of 2m/s.

to the simulations presented in Section 3.3.2, that is an average wind speed of about 0.5m/s blowing from the north-east. The second simulation features a significantly stronger average wind speed of about 2m/s, in this case blowing from the northwest. For this case, the instantaneous wind speed at the airship centre of buoyancy is shown in figure 4–11.

The evolution of the airship roll and pitch angles is shown in figure 4–12 for the simulation with an average wind of 0.5m/s. Compared to the no-wind case, shown in figure 4–5, there is little difference in the controller performance between the two cases. The deviations from the desired attitude are slightly larger but are still very small. The yaw angle evolution for this case is shown in figure 4–13. The yaw angle is still tracked very well, but in contrast to the no wind case shown in figure 4–6, the actual yaw angle deviates sufficiently from the desired value that both curves become visible.

Figures 4–14 and 4–15 depict the attitude and the yaw angle evolution for the simulation with an average wind of 2m/s. In the presence of this external disturbance,



Figure 4–12: Roll and pitch evolution during simulation with an average wind speed of 0.5 m/s.



Figure 4–13: Yaw evolution during simulation with an average wind speed of ulation with with an average wind speed $0.5 \mathrm{m/s}$.



Figure 4–14: Roll and pitch evolution during simulation with an average wind speed of 2m/s.



Figure 4–15: Yaw evolution during simof 2m/s.





Figure 4–16: Forward and upward velocity evolution during simulation with an ity evolution during simulation with an average wind speed of 0.5 m/s.

Figure 4–17: Forward and upward velocaverage wind speed of 2m/s.

the tracking deteriorates, but is still very acceptable. The roll and pitch angles exhibit more small-scale oscillations and the peak deviations increase to about 5 degrees. The yaw angle tracking, which was practically perfect in the previous simulations, now shows peak deviations in the range of 20 degrees, especially during the turns. For a constant desired attitude, the yaw angle converges well on the target value.

The velocity control performance for the simulation with an average wind speed of 0.5 m/s is shown in figure 4–16. The performance is not significantly worse than for the no wind case shown in figure 4–8. Due to the $\dot{\boldsymbol{v}}_{w,I}$ term in the equations of motion that was discussed in Section 2.3, the intensity of the small-scale high frequency deviations has increased in the presence of the wind turbulence. The deviations from the target value have only increased slightly in this scenario.



Figure 4–18: Control commands during simulation with an average wind speed of 0.5m/s.

Figure 4–19: Control commands during simulation with an average wind speed of 2m/s.

For the case with 2m/s wind, the velocity evolution is shown in figure 4–17. The high-frequency deviations have now reached an amplitude of about 0.2m/s and the peak deviations have increased in amplitude to about 0.5m/s. Considering the magnitude of the turbulence shown in figure 4–11 and considering that the airship motion follows these gusts without delay, this can be considered very good control performance.

The evolution of the thruster variables for the two simulations with non-zero wind are shown in figures 4–18 and 4–19. Naturally, the control inputs exhibit larger oscillations, as the controller tries to compensate for the turbulence. However, the overall impression of relatively smooth thrust inputs is confirmed in these cases. For the test case with an average wind of 2m/s, the thruster forces exhibit a spike of almost 10N around t=90s together with a large increase in tilt angle. This coincides

with the pitch down command. Apparently, this manoeuvre brings the airship close to control saturation requiring a large yaw moment.

The overall control performance of the nonlinear controller developed in Section 4.1 is very good, even in the presence of parametric uncertainties, actuation delays, sensor noise and external disturbances. The controller appears to be robust against a large range of disturbances and is hence deemed suitable for flight testing.

4.3 Controller performance evaluation in outdoor flight tests

The controller flight tests have been conducted on McGill's Rutherford field with the dimensions shown in figure 4–20. The desired attitude and velocity values were adapted in flight in order to maintain sufficient clearance to obstacles. At the same time, the forward velocity was held at values around 1.5m/s wherever possible. Higher speeds were not tested due to the lack of space in the test area. In case of loss of GPS signal or other anomalies, the operator could interrupt the flight at any point in time, leading to a shut down of the thrusters and a smooth sinking of the airship to the ground.

The controller has been implemented on the on-board electronics of the Quanser MkII using custom C-code. It was running at 50Hz which was the fastest rate possible with the existing hardware. The GPS has an update rate of 10Hz and the IMU readings are updated at 50Hz. This implied that the controller was running asynchronously from the GPS. The optimization algorithm is a C-implementation of the quadratic programming algorithm with inequality constraints presented in [80]. The integral terms in the controller are computed using a simple Euler method.



Figure 4–20: Overview of the area used for flight testing.

The airship height during the flights is measured by a sonar, as the precision of the vertical GPS position information is not sufficient for this purpose. The sonar has a maximum detection range of 7m, limiting the range of airship heights for the flight tests to 3m to 5m.

The low-level controller controls velocity in body x and z direction, but not the height. Therefore, the desired inputs to the low-level controller will need to be adapted in order to stay within the permissible height range. It was decided that it is not realistic to expect the airship operator to perform this task in real time. Therefore, the low-level controller was augmented by a height control algorithm.

This means that the operator provides a desired height, a desired forward velocity, and the desired attitude to the airship during flight tests. The desired values are input to a laptop and transmitted to the airship via WiFi. The operator can modify the values in discrete steps via mouse and keyboard. During later flight tests, the values could also be modified with a joystick, permitting smoother changes of the desired values.

4.3.1 Height control for flight tests

In a first step, the operator commands the desired velocity in body x direction and the desired height. The height controller then computes the desired velocity in body z direction based on the proportional control law

$$w_d = c_h(h - h_d) \tag{4.32}$$

with $c_h > 0$ being the height control gain. During the flight tests presented here, c_h was set to $0.2\frac{1}{s}$.

This height control law only works for small roll and pitch angles. During the flight tests, the desired roll and pitch angles were always zero, and so it was assumed that this condition was met.

As will be seen in the first flight test presented below, attitude deviations noticeably affected the height control with deviations in pitch having the strongest impact. Due to the pitch deviation, part of the commanded speed in the body x direction is oriented upwards or downwards driving the airship away from the commanded height.

Therefore the controller was modified to compensate for the actual attitude. Instead of commanding the velocity in the body x direction, the operator commands the desired horizontal velocity in the direction of the airship yaw angle $u_{d,h}$. The height controller described above was also altered so that it commands the desired velocity in the inertial z direction rather than the body z direction giving

$$w_{d,I} = c_h (h - h_d). (4.33)$$

These two desired values $u_{d,h}$ and $w_{d,I}$ are transformed into the body frame values u_d and w_d using the following transformations.

The desired velocity in body x direction u_d is computed via

$$u_d = u_{d,h} \cos \theta - w_{d,I} \sin \theta. \tag{4.34}$$

For the desired velocity in body z direction w_d , both the roll angle ϕ and the pitch angle θ need to be considered. If the airship has rolled to one side, the airship velocity in the body y direction v has an impact on the vertical velocity in the inertial frame w_I . Furthermore, the speed in the body z direction w has a reduced contribution to w_I , as the roll angle ϕ increases. The equation

$$w_d = -v \tan \phi + \frac{u_{d,h} \sin \theta + w_{d,I} \cos \theta}{\cos \phi}$$
(4.35)

is found to ensure the correct vertical velocity in the inertial frame $w_{d,I}$ even though the speed in body y direction v cannot be influenced directly. The derivation of this equation is omitted here for brevity.

4.3.2 Flight test results

The performance of the nonlinear low-level controller has been investigated during 73 flight tests with a total airborne time of 93 minutes. Additional flights with





Figure 4–21: Roll and pitch evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.

Figure 4–22: Yaw evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.

the high-level controller also active were performed and these will be discussed in the subsequent chapter.

The first flight tests were conducted without any modification to the controller or its parameters, as described previously in this chapter. The controller provided good and stable results using the parameters determined in the simulation. This demonstrates the robustness of the controller with respect to the effects not modelled in the simulation discussed at the end of this chapter.

Figures 4–21 and 4–22 show the evolution of the airship attitude during a flight test lasting 6.5 minutes. The commanded roll and pitch values are constant throughout the flight, as shown in figure 4–21. They are non-zero in order to compensate for the IMU misalignment. Before lift-off, the airship is held level by the ground crew and the desired attitude values are synchronized with the current IMU readings. This ensures that the airship will perform the vertical ascent to the desired height with a steady attitude.

During the first 50 seconds of the flight, the desired roll and pitch values are held very well, then larger deviations with peaks of 10 degrees are recorded. The deterioration of the attitude control performance is coincident with the commanded increase in forward speed, which will be discussed further down.

The yaw control performance is shown in figure 4–22. The desired yaw angle is tracked well, but with a slight delay. The desired yaw angle during these flights is modified in discrete steps, making the derivative of the desired quaternion undefined. The terms $\dot{\boldsymbol{q}}_d$ in (4.18) are hence set to zero causing the tracking delay.

Unfortunately, due to the forward speed and the limited space at the test site, the commanded yaw angle changes very frequently so that the steady state performance is difficult to evaluate. Around t=130s and t=300s, the commanded yaw angle is held constant for about 30s and here, the yaw angle is maintained very well by the controller.

The velocity control performance is shown in figure 4–23. Changes in desired forward velocity are tracked well and precisely. However, with a constant commanded velocity, frequent deviations from this velocity are observed. The amplitude of these deviations is mostly in the range of 0.5m/s with some peaks of 1m/s.

The vertical velocity tracking is also shown in 4-23. The commanded velocity corresponds to the output of the height control law given by (4.32). Apparently, there is a visible delay in the tracking of this desired velocity. Besides the delay, the deviations from the desired value show a similar intensity as for the forward velocity.





Figure 4–23: Forward and vertical velocity evolution during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.

Figure 4–24: Control commands during a flight test with the simple height control (4.32) and the controller gains given in table 4–1.

The large peak around t=210s, indicating a sudden upwards motion, is a result of actuator saturation as can be seen in the thrust parameters shown in figure 4–24. In contrast to the simulation results, the data shown in figure 4–24 are the controller commands, as the actual tilt angle and the actual thrust cannot be measured on the QuanserMkII. At t=210s, the thrust commands alternate quickly between the minimum and the maximum value with simultaneous large tilt angle changes. This is a sign that the thrust allocation algorithm cannot find an exact solution for the desired total thruster forces and moments commanded by the controller.

With the current optimization gains, attitude control has priority over velocity control. Also, the thrusters cannot produce downward thrust. Therefore, the constrained optimum determined by the thrust allocation generally results in a larger





Figure 4–25: Airship path during a flight Figure 4–26: test with the simple height control (4.32) flight test with the simple height control and the controller gains given in table 4– 1.

Airship height during a (4.32) and the controller gains given in table 4–1.

force in the body z direction than commanded as explained in Section 2.4. This explains the sudden upward motion at t = 210s.

The path resulting from these attitude and velocity inputs is given in figure 4–25 for reference. While the controller performance is to be judged purely based on Euler angle and velocity tracking, the path shows well how the airship is kept within the restricted testing area. Each dot of the path shown represents one GPS position reading. As the GPS is sampled at a constant rate of 10Hz, the distances between the points indicate the horizontal airship velocity. Apparently, there are some larger gaps between successive points in the airship trajectory. In these cases, the GPS reception was not good enough to provide a cm-precision DGPS solution and the GPS reported spurious position readings or a position of (0,0). In these cases, the velocity measurement required by the controller also exhibits a large uncertainty.

The desired airship height is set to 4m for the entire flight. The actual height profile flown by the airship is shown in figure 4–26. The airship oscillates around the set point and the controller manages to keep the deviations within ± 1.5 m for most of the flight, but also deviations beyond 2m are observed.

Besides some issues with the height sonar, the lack of attitude compensation in the height control law (4.32) was identified as a key culprit in the lack of height control. Especially with significant forward velocities, deviations in airship pitch led to large height excursions. For most of the flight, the commanded velocity in body x direction was set to 1.5m/s. A pitch deviation of just 5 degrees leads to a velocity in inertial z direction of 0.13m/s for this forward velocity. Due to the low value of the height control gain $c_h = 0.2\frac{1}{s}$, a deviation from the desired height of more than 0.5m is required for the height controller to compensate for this velocity component.

This issue was noted earlier in Section 4.3.1 as the motivation for the modified height controller that incorporates the attitude compensation (4.33) to (4.35) as an approach to improve height control performance.

Table 4–2: Controlle	r and thrust	allocation	parameters	after tun	ing for o	ptimal f	flight
test performance.							

Parameter	c_{a1}	c_{a2}	λ_{a1}	c_{v1}	c_{v2}	λ_{v1}	w_n	w_f	w_{Ft}
Value	$0.7\frac{1}{s}$	$2.5\frac{1}{s}$	$0.9\frac{1}{s^2}$	$2.5\frac{s}{m}$	$0.3 \frac{m}{s^2}$	$0.1\frac{1}{s^2}$	$1\frac{1}{N^2m^2}$	$1\frac{1}{N^2}$	$0.01\frac{1}{N^2}$

Besides the height control law change, the low-level controller parameters were also adjusted in a series of flight tests to improve velocity tracking. The weights in the optimization function (3.21) were changed to increase velocity control priority. The weights on the error in thruster force are now equal to the weight on the





Figure 4–27: Roll and pitch evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2.

Figure 4–28: Yaw evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2.

thruster moment error. Furthermore, the velocity control gain c_{v2} was doubled. For good performance, these changes required a slight relaxation of the attitude control parameters. The final parameter set is given in table 4–2.

The results of a flight test with the attitude compensated height controller and the new controller gains are shown in figures 4–27 to 4–32. The attitude tracking is shown in figures 4–27 and 4–28. The tracking performance is very similar to the flight test presented previously with the peak deviations recorded in pitch and roll being about 10 degrees. Most deviations are significantly less. The yaw angle shows quickly attenuated overshoots after changes to the commanded yaw angle. Apparently, the modifications to the controller did not adversely affect attitude control.



Front right thruster (1 Rear right thruster (2) Rear left thruster (3) loft thrust 60 40 80 10 Time [s] 100 120 140 180 160 100 Thruster tilt angle [deg] -50 -100L 20 40 80 100 Time [s] 120 140 160 180 60

Figure 4–29: Forward and upward velocity evolution during a flight test with attitude compensated height control and the controller gains given in table 4–2.

Figure 4–30: Control commands during a flight test with attitude compensated height control and the controller gains given in table 4–2.

Figure 4–29 shows the velocity tracking performance. The deviations from the desired value have been reduced due to the controller modifications, especially for the vertical velocity.

The evolution of the thrust commands shown in figure 4–30 shows an increase in force peaks on the thrust inputs. These are transient peaks lasting one time step occurring simultaneously with changes in the desired attitude. For this test, a method to compute a value for \dot{q}_d was included in the controller to improve yaw tracking. This contained a bug that is responsible for these peaks. After the bug was eliminated, only flight tests with the high-level controller active were conducted, hence all low-level controller flight tests with attitude compensated height control contain this bug. Due to the transient nature of these peaks, the thrusters filter them out and they have little impact on control performance.





Figure 4–31: Airship path during a flight Figure 4–32: test with attitude compensated height flight test w control and the controller gains given in height contro table 4–2. given in table

Figure 4–32: Airship height during a flight test with attitude compensated height control and the controller gains given in table 4–2.

Airship path and height evolution are shown in figures 4–31 and 4–32. Apparently, the modifications applied to the controller have significantly improved the height control performance with the peak deviations from the target value now being in the range of 1m and less. At no point in time does the airship get close to the ground or exceed the sonar range.

Summing up the control performance observed during flight test, it can be said that, while being satisfactory for outdoor flight tests, the performance is not as good as expected based on the simulation. Three key factors have been identified as the cause of this:

• Thruster model: The development of a precise, reliable thruster model has proven difficult. The results presented in [1] and [81] provide a useable model, however many parameters are not covered in this model. Among the parameters not covered are aspects such as impact of the airspeed on the thrust, the influence of the battery voltage on the actual prop rpm, and individual differences between the thrusters due to manufacturing tolerances.

- Mounting of the thrusters: In the simulation, the thrusters are mounted at a precise location, with the thrust acting exactly in the xz plane and a tilt angle of μ_i=0 corresponding exactly to vertical thrust. For the flight tests, the thrusters are mounted to the flexible airship hull using velcro patches and chords. While care is taken during airship rigging to get the best possible thruster alignment, it is impossible for the parameters to be as precise as those in the simulation. Also, the flexible airship hull allows for the thruster alignment to change depending on the current thrust and tilt angle.
- Wind gusts for flights close to the ground: Figure 4–33 shows a 90s interval of the wind measured at the flight test site during one of the flight tests. Apparently the changes in wind speed at the test site are much larger than the turbulence model employed in the simulation despite a similar average wind speed as presented for the 2m/s wind simulation test case shown in figure 4–11.



Figure 4–33: Inertial wind speeds measured at the flight test site.

CHAPTER 5 High-level controller design

To achieve autonomous trajectory tracking capability, the low-level controller discussed in the previous section is augmented by a high-level controller. As previously shown in figure 3–1, the high-level controller generates the desired values for the low-level controller in order to achieve flight along a predefined trajectory.

The vehicle studied in this work is capable of flight in all direction as well as hover. Therefore, the ideal high-level controller will provide the possibility to perform both path tracking and hover, using the full capabilities of the vehicle. Having one controller suitable for all regimes of flight eliminates the need for switching between different controllers providing a smoother closed-loop behaviour.

5.1 High-level control scheme

Different options for the high-level control algorithms have been discussed in Section 1.2.4. Tracking of a reference point that is at a fixed or variable distance ahead of the vehicle on the reference path is not suitable for this application, as this concept does not allow hover. Vector field tracking can in principle cover both cases, flight along a path and hover, but it is potentially high in computational effort. While it may be possible to compute the vector fields in real time with the appropriate hardware, this option was dismissed here due to the limited computational capacity of the on-board electronics of the Quanser MkII. The virtual spring damper principle described in [61] allows tracking of a reference vehicle independent of its motion with low computational effort. This method is hence suitable for both, trajectory tracking and hover. The method is used in [61] for a vehicle with a fixed forward velocity without giving further details about the vehicle. Here, the method will be adapted for use with a finless airship, allowing its use in vehicles with a variable velocity that can travel in all directions — forward and backward, sideways as well as vertically.



Figure 5–1: Illustration of the virtual spring-damper system used for trajectory tracking.

The principle of the virtual spring damper method is illustrated in figure 5–1. The reference vehicle is at location \mathbf{r}_r , travelling with the velocity \mathbf{v}_r . The airship is located at the position \mathbf{r}_b travelling with the velocity \mathbf{v} . The controller computes a commanded velocity \mathbf{v}_c by virtually connecting the airship centre of buoyancy with the centre of the reference vehicle with a spring damper system, yielding the following
ODE for the dynamics of \boldsymbol{v}_c

$$\dot{\boldsymbol{v}}_c = \sigma_2 \left[(\boldsymbol{v}_r - \boldsymbol{v}_c) + \sigma_1 (\boldsymbol{r}_r - \boldsymbol{r}_b) \right] + \dot{\boldsymbol{v}}_r.$$
(5.1)

with $\sigma_1 > 0$ and $\sigma_2 > 0$ being controller design parameters. The specific damping coefficient of the spring-damper system $\frac{c}{m}$ is given by σ_2 , while the specific spring constant $\frac{k}{m}$ is given by $\sigma_1 \sigma_2$. The last term, the derivative of the reference vehicle velocity $\dot{\boldsymbol{v}}_r$, guarantees convergence, if the reference vehicle travels on a curved path or changes velocity.

If the airship velocity \boldsymbol{v} tracks the commanded velocity \boldsymbol{v}_c perfectly, the proof of convergence for (5.1) is achieved using the control Lyapunov function

$$V_{h} = \frac{1}{2} \begin{bmatrix} \boldsymbol{r}_{r} - \boldsymbol{r}_{b} \\ \boldsymbol{v}_{r} - \boldsymbol{v}_{c} \end{bmatrix}^{T} \begin{bmatrix} 1 & \frac{\epsilon}{\sigma_{1}\sigma_{2}} \\ \frac{\epsilon}{\sigma_{1}\sigma_{2}} & \frac{1}{\sigma_{1}\sigma_{2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_{r} - \boldsymbol{r}_{b} \\ \boldsymbol{v}_{r} - \boldsymbol{v}_{c} \end{bmatrix}$$
(5.2)

with $\epsilon > 0$ small enough to keep (5.2) positive definite.

Perfect tracking requires that

$$\dot{\boldsymbol{r}}_b = \boldsymbol{v} = \boldsymbol{v}_c. \tag{5.3}$$

With the help of (5.3), the derivative with respect to time of (5.2) can be computed to

$$\dot{V}_{h} = -\begin{bmatrix} \boldsymbol{r}_{r} - \boldsymbol{r}_{b} \\ \boldsymbol{v}_{r} - \boldsymbol{v}_{c} \end{bmatrix}^{T} \begin{bmatrix} \epsilon & \frac{\epsilon}{2\sigma_{1}} \\ \frac{\epsilon}{2\sigma_{1}} & \frac{\sigma_{2} - \epsilon}{\sigma_{1}\sigma_{2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_{r} - \boldsymbol{r}_{b} \\ \boldsymbol{v}_{r} - \boldsymbol{v}_{c} \end{bmatrix}$$
(5.4)

For sufficiently small values of ϵ , equation (5.4) is negative definite, therefore proving exponential convergence of \mathbf{r}_b to \mathbf{r}_r and \mathbf{v}_c to \mathbf{v}_r . Equation (5.4) also shows the motivation for introducing the ϵ related terms in (5.2). For $\epsilon = 0$, all terms related to the position error $\mathbf{r}_r - \mathbf{r}_b$ vanish in (5.4), and (5.4) is only negative semi-definite, proving stability but not convergence.

A formal proof of convergence for imperfect tracking, i.e. for $v \neq v_c$, is not possible without a detailed analysis of the dynamics of the difference between v and v_c . For this work, the convergence demonstrated in simulation and flight test will be considered sufficient.

The trajectories used in this thesis consist of sequences of straight lines. The reference vehicle travels along each line with a constant velocity, abruptly changing direction at each waypoint. The derivative of the reference vehicle velocity $\dot{\boldsymbol{v}}_r$ is hence either zero or not defined. Therefore, equation (5.1) expressed in inertial coordinates

$$\dot{\boldsymbol{v}}_{c,I} = \sigma_2 \left[(\boldsymbol{v}_{r,I} - \boldsymbol{v}_{c,I}) + \sigma_1 (\boldsymbol{r}_{r,I} - \boldsymbol{r}_{b,I}) \right]$$
(5.5)

will be used in the following. Extension to non-straight trajectories is possible by adding the additional term given in (5.1).

5.1.1 Computation of the desired attitude and velocity in body frame coordinates.

The inputs to the low-level controller, the desired attitude \mathbf{q}_d and the desired velocities in body x and body z direction, u_d and w_d , are computed based on the commanded velocity $\mathbf{v}_{c,I}$. With the high-level controller active, the inputs to the low-level controller \mathbf{q}_d , u_d , and w_d are controller-internal commanded values rather than the desired set point provided by the operator. Therefore, the subscript of these values should now be c rather than d. However, for consistency with the equations given in Chapter 4 the subscript d will be retained here. Two different approaches have been investigated for the computation of the desired velocities in body x and z direction. The first approach is a simple transformation of the commanded velocity $v_{c,I}$ into body frame coordinates and omitting the y component of the desired velocity in body frame coordinates. This gives

$$\begin{bmatrix} u_d \\ w_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R} \mathbf{v}_{c,I}.$$
 (5.6)

The second approach uses the attitude compensation (4.34) and (4.35) used for the height control discussed in Section 4.3.1. The goal of this approach is to trade improved height control for a deterioration in lateral control, which may be useful for flight tests which are restricted to a tight height band.

Equations (4.34) and (4.35) compute the desired velocities u_d and w_d in body xand z direction based on the desired vertical velocity $w_{d,I}$ and the desired horizontal velocity in the direction of the airship heading $u_{d,h}$.

The desired velocity in the vertical direction $w_{d,I}$ corresponds to the z component of the commanded velocity $w_{c,I}$.

The computation of $u_{d,h}$ is illustrated in figure 5–2. The axes x_h and y_h in this figures lie in the horizontal plane with x_h pointing in the direction of the airship heading and y_h being perpendicular to x_h . The desired horizontal velocity in the direction of the airship heading $u_{d,h}$ corresponds hence to the projection of the commanded velocity vector $\boldsymbol{v}_{c,I}$ onto the x_h axis.



Figure 5–2: Projection of the commanded velocity vector \boldsymbol{v}_c onto the body axes.

The projection is computed in a two step process. In the first step, the horizontal magnitude and direction of $v_{c,I}$ are computed via

$$u_{c,hor} = \sqrt{u_{c,I}^2 + v_{c,I}^2} \tag{5.7}$$

$$\psi_{c,hor} = \tan^{-1} \frac{v_{c,I}}{u_{c,I}}.$$
 (5.8)

Subsequently, the desired velocity in airship heading direction $u_{d,h}$ is computed via

$$u_{d,h} = u_{c,hor} \cos(\psi_{c,hor} - \psi). \tag{5.9}$$

Having computed values for $u_{d,h}$ and $w_{d,I}$, equations (4.34) and (4.35) can now be used to compute the desired velocities in body x and z direction u_d and w_d .

The controller computes desired roll and yaw angles in order to reduce the lateral tracking error while compensating for the effects of the unknown wind. The computation of both angles is based on the difference between the commanded and actual velocity in body y direction

$$v_{err} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \mathbf{R}(\mathbf{v}_{c,I} - \mathbf{v}_{I}).$$
(5.10)
Wind
$$\overset{v}{\longrightarrow} \overset{v}{\longrightarrow} \overset{v}{\operatorname{drift due}} \overset{F_{p}}{\longleftarrow} \overset{v}{\longrightarrow} \overset{F_{p}}{\longleftarrow} \overset{v}{\longrightarrow} \overset{F_{p}}{\longleftarrow} \overset{v}{\longleftarrow} \overset{F_{p}}{\longleftarrow} \overset{v}{\longleftarrow} \overset{v}{\longleftarrow} \overset{v}{\longleftarrow} \overset{F_{p}}{\longleftarrow} \overset{v}{\longleftarrow} \overset{v}{\longleftrightarrow} \overset$$

Desired yaw to compensate for wind

Figure 5–3: Roll and yaw for wind compensation.

The roll of the airship is used to direct some of the upwards thruster force sideways to provide a short term reduction of v_{err} , as shown on the right side of figure 5–3. If $v_{err} > 0$, the body y component of the desired velocity vector is stronger to the right than the body y component of the actual velocity. Rolling the airship to the right, i.e. commanding $\phi_c > 0$, will allow to reduce this error by directing some thrust in the desired direction of travel. A proportional control law for the roll angle providing this effect is

$$\phi_c = c_\phi v_{err} \tag{5.11}$$

with $c_{\phi} > 0$ being a controller design parameter.

The computation of the yaw angle follows a similar principle with the key difference being that the change of desired yaw angle depends on the sideways velocity error v_{err} rather than the yaw angle itself, giving the control law

$$\dot{\psi}_c = c_\psi v_{err} \tag{5.12}$$

with $c_{\psi} > 0$ being a controller design parameter. With this approach, if $v_{err} = 0$, the airship will fly with a roll angle of 0 and a constant heading pointing into the oncoming airflow. If $v_{err} \neq 0$, the airship will roll towards the desired direction and yaw in the same direction until the error is eliminated.

The equilibrium state achieved by this control law is illustrated in figure 5–4 for two different wind conditions. The illustration on the left side of figure 5–4 shows a tail wind that is less in magnitude than the commanded velocity v_c . The right side of the same figure illustrates a case with tail wind that is stronger in magnitude than the commanded velocity \boldsymbol{v}_{c} .



Wind speed less than the commanded velocity

Wind speed greater than the commanded velocity

Figure 5–4: Steady state condition for the path control.

Apparently, this control law strives to align the airship heading with the relative wind which is the difference between airship velocity v and wind speed v_w . This minimizes the Munk moment and hence reduces the control input required for maintaining this attitude. Interestingly, this can lead to a situation in which the airship is oriented opposite to its direction of travel, if the wind speed v_w is greater than the commanded velocity v_c .

To compute the desired quaternion q_d , the commanded pitch angle remains to be computed. As the velocity in both body x and body z direction can be directly controlled, the choice of the commanded pitch angle θ_c has no direct impact on the trajectory tracking performance. In this thesis, two different choices of θ_c have been investigated.

The first choice is to set

$$\theta_c = \tan^{-1} \frac{w_{c,I}}{u_{c,hor}}.$$
(5.13)

For zero wind, this will align the airship longitudinal axis with the direction of travel, hence minimizing the Munk moment and reducing the required control effort for attitude control. However, in the presence of wind, this choice of θ_c does not necessarily align the longitudinal axis with the direction of the oncoming airflow. In order to compute the correct value for θ_c for that case, knowledge of the wind conditions is required.

Since information on the wind conditions is often unavailable, a second option is to simply set θ_c to zero. In this case, height changes are achieved purely via change of the desired velocity in the body z direction w_d .

Using (A.5), the three commanded Euler angles ϕ_c , θ_c , and ψ_c are transformed into the desired quaternion \boldsymbol{q}_d provided to the low-level controller.

5.2 Simulation based performance evaluation

Analogously to the design of the nonlinear low-level controller, the performance of the high-level controller has first been investigated in simulation. For the simulation, an arbitrary sequence of waypoints has been chosen. The reference vehicle speed has been set to 1m/s. The waypoint height is different from one point to the next to test nonplanar flight plan tracking. The flight plan includes a vertical leg between waypoints 2 and 3 to verify airship hover with altitude change. Between waypoints 5 and 7, the flight path contains strong directional changes to verify the controller's ability to cope with these. After reaching the last waypoint, the controller task is to hover at this waypoint. The path travelled by the reference vehicle is visualized in figure 5–5.



Figure 5–5: Reference vehicle path during the high-level controller simulation.

The values for t at which the reference vehicle reaches the individual waypoints are given in table 5–1. As the position error between the airship and the reference

Waypoint	Time t [s]
1	0
2	96
3	106
4	159
5	261
6	312
7	383
8	495

Table 5–1: Times t at which the individual waypoints are sequenced

vehicle is small enough when the reference vehicle reaches the different waypoints, it immediately resumes travel to the next waypoint. The switching times are hence only dependent on the reference vehicle speed and are the same for all simulations.

It should be noted that the trajectory presented in figure 5–5 does not consider the physical constraints of the airship, such as minimum turning radius. In fact, the reference vehicle abruptly changes its direction of travel at each waypoint. The derivative of the reference vehicle velocity $\dot{\boldsymbol{v}}_r$ is therefore not defined at the waypoint switching. Hence, this test allows to investigate the disturbance rejection properties of the controller as a strong disturbance is injected into the system at each waypoint.

The controller parameters promising the best trade-off between stability and tracking quality have been determined iteratively in a series of simulations. The best values have been determined as those given in table 5–2. All simulations presented here use this set of parameters.

Table 5–2: High-level controller parameters for the simulations presented

$\sigma_1 \left[\frac{1}{s}\right]$	$\sigma_2 \left[\frac{1}{s}\right]$	$C_r \left[\frac{\deg \cdot \mathbf{s}}{\mathbf{m}}\right]$	$c_y \left[\frac{\text{deg}}{\text{m}}\right]$
0.2	0.4	28	16.8

The first set of simulations presented here investigates the performance of the different options for the controller architecture described above. Test cases 1 and 2 use a commanded pitch angle according to (5.13), test cases 3 and 4 work with a commanded pitch angle of zero. The computation of the desired velocities in body x and z direction, u_d and w_d is based on the attitude compensation equations (4.34) and (4.35) for test cases 1 and 3. For the other two test cases, the desired velocities are computed using equation (5.6). All four test cases are simulated with a low average wind speed of 0.5m/s blowing from the north-east.

Figure 5–6 shows the travelled paths in the horizontal plane for the four test cases. The tracking is very good for all four cases with noticeable deviations only visible at waypoints with strong changes in direction. The path that the airship travels is practically identical for test cases 1 and 2. Test cases 3 and 4 also show very similar behaviour. That implies that the choice of the commanded pitch angle θ_c has a more significant impact on the path tracking than the two different schemes to compute the desired velocities.

The evolution of the reference height and the actual airship height for each of the test cases is shown in figure 5–7. Again, there is very little difference between the different test cases. The desired height is tracked almost perfectly with the exception of the area around t = 100s. At this time, the reference vehicle climbs vertically from waypoint 2 to waypoint 3 at a speed of 1m/s. The maximum value for w_d is limited to 0.5m/s for stability reasons. Therefore, the airship is incapable of following the reference vehicle perfectly, leading to the temporary discrepancy shown in figure 5–7.



Figure 5–6: Path tracking performance under low wind conditions.



Figure 5–7: Airship height evolution under low wind conditions.

At the beginning of the simulation, another short-term deviation is visible which is a result of the simulated parametric uncertainty discussed in Chapter 3. The airship is modelled heavier in the simulation than reflected by the low-level controller parameters. Therefore, it initially loses altitude until the integral terms in the lowlevel controller have compensated for the increased airship weight.

As the performance of the different test cases appears very similar in the figures, the RMS of the difference between reference vehicle position and airship position was computed for each of the test cases. Due to the large deviations encountered at the start of the simulation and around t = 100s, the RMS position error has been computed for t > 150s only. Furthermore, the RMS position error has been split into the trajectory tracking phase from t = 150s to t = 500s and the hover phase from t = 500s to t = 600s.

Test case	Trajectory tracking			Hover			
	North [m]	East [m]	Down [m]	North [m]	East [m]	Down [m]	
1	0.7397	0.8275	0.1222	0.3588	0.3827	0.1199	
2	0.7320	0.8043	0.1799	0.3571	0.3496	0.1031	
3	0.6112	0.6341	0.0857	0.1884	0.3339	0.0638	
4	0.6063	0.6318	0.1082	0.1880	0.3281	0.0660	

Table 5–3: Root mean square position error for the simulations with low wind.

The RMS position errors are given in table 5–3. The average position error is consistently less for test cases 3 and 4, indicating that setting the commanded pitch angle θ_c to zero provides the better control performance. The test cases using (5.6) for the computation of the desired velocities show slightly lower RMS errors for the north and east coordinates than the respective tests using (4.34) and (4.35). The height control during trajectory tracking is significantly better for the cases that use (4.34) and (4.35), while for hover, the difference between the cases is negligible. This corresponds well to the expected behaviour of each option for the computation of the desired velocities.

Based on these results, it was decided to set the desired pitch angle θ_c to zero. The two different possibilities to compute the desired velocities have been analyzed under stronger wind conditions to verify the disturbance rejection properties. The wind conditions for these tests are a wind with an average velocity of about 1.5m/s blowing also from the north-east. The new test cases 5 and 6 corresponds to test cases 3 and 4, respectively, with the increased wind speed.





Figure 5–8: Position error evolution dur- Figure 5–9: Position error evolution during simulation of test case 5.

ing simulation of test case 6.

The position difference between the reference vehicle and the airship is given in figure 5–8 for test case 5 and in figure 5–9 for test case 6. The position error is dominated by the deviations encountered at each waypoint switching. These deviations are practically identical for both test cases, and the subsequent convergence onto the path is still very good despite the stronger external wind.

Due to the similarity of figures 5–8 and 5–9, the RMS error was used to established which method for the computation of the desired velocities will be used in flight tests. The RMS position errors for these test cases are given in table 5–4.

Table 5–4: Root mean square position error for the simulations with high wind.

Test case	Trajectory tracking			ase Trajectory tracking Hover			
	North [m]	East [m]	Down [m]	North [m]	East [m]	Down [m]	
5	0.7914	0.7022	0.1496	0.4055	0.3393	0.1205	
6	0.7719	0.6956	0.1857	0.4022	0.3396	0.1287	





Figure 5–10: Roll and pitch angles evolution during simulation of test case 5.

Figure 5–11: Yaw angle evolution during simulation of test case 5.

For the hover case, the differences between the two test cases are negligible. In the trajectory tracking part, test case 5 exhibits slightly larger deviations in north and east direction, but a significantly better performance for the vertical direction. As height control is very critical during flight tests, it was decided to use the architecture used in test cases 3 and 5 for flight tests. The desired pitch angle θ_c will be set to zero, and the attitude compensation equations (4.34) and (4.35) already used for the low-level controller flight tests will also be used for the high-level controller tests.

As test cases 5 and 6 appear to give a very similar performance, only test case 5 will be analyzed here in further detail. The roll and pitch angle evolution is shown in figure 5–10, the evolution of the yaw angle together with the wind direction ψ_w and the desired direction of travel $\psi_{c,hor}$ are shown in figure 5–11. The figures show that the commands provided to the low-level controller are smooth without discontinuities despite the fact that the inputs to the high-level controller contain discontinuities.

The commanded pitch angle θ_c is zero throughout the entire simulation. As can be seen in figure 5–10, this angle is maintained with deviations of less than 5 degrees throughout most of the flight. The only exception with larger deviations is the vertical climb segment that starts at t = 100s. During the vertical climb, the pitch angle deviations increase to 15 degrees, which is the motivation for the climb speed limitation of 0.5m/s discussed earlier.

The roll angle also remains within 5 degrees for most of the flight. At waypoint switches, the commanded roll angle increases up to 15 degrees during the change in direction of travel. Once the new direction is established, the tracking only requires small roll angles.

The yaw angle, shown in figure 5–11, shows that the airship heading depends more strongly on the direction that the wind is blowing from ψ_w than the desired direction of travel $\psi_{c,hor}$. Whereas the desired direction of travel $\psi_{c,hor}$ covers the entire 360 degree range, the airship heading only ranges from about 0 to 150 degrees. The direction that the wind is blowing from ψ_w is in the range of 40 to 90 degrees for most of the flight. Therefore, the airship is pointing mainly into the wind with some corrections to the left or right for the path tracking. During the hover phase, starting at t = 500s, the simulation features a change in wind direction from about 60 to 160 degrees. The airship heading follows this change in wind direction with a small delay. The controller aims to point the airship directly into the wind during hover, which corresponds to the design goal described in Section 5.1.1.

Figure 5–13 shows the lateral airspeed during the flight. At each waypoint switching, the lateral airspeed contains a spike, but it converges back to zero during



Figure 5–12: Velocity evolution during Figure 5–13: Lateral velocity evolution during simulation of test case 5.

the tracking of each segment, indicating that the controller aligns the airship with the oncoming airflow.

The evolution of the velocities in the body x and z directions is shown in figure 5–12. Height changes between the waypoints are achieved purely by the speed in the body z direction, as the commanded pitch angle is set to zero throughout the entire flight. Therefore, the average desired velocity in the body z direction is different from zero throughout most of the flight, except during hover. The velocity in the body z direction remains within 0.5 m/s throughout most of the flight except during the initial intercept and during the vertical climb. During the vertical climb, the desired velocity is set to -0.5m/s, the actual velocity temporarily exceeds that value.

The velocity in the body x direction remains within 1.5m/s for most of the flight, significantly larger peaks occur only during waypoint switches. The reference vehicle moves with a velocity of 1m/s, the desired velocity of the airship is hence



Figure 5–14: Forward airspeed evolution during simulation of test case 5.

expected to be around 1m/s for most of the flight. However, the desired velocity in body x direction is significantly less in many parts of the flight. In this case the sideways drift due to the wind provides the missing velocity, as shown in figure 5–13. During the first leg, the forward velocity of the vehicle is close to 1m/s, this leg is pointing almost straight into the wind. Between t = 261s and t = 312s, the vehicle travels backwards with a negative velocity in body x direction. During this time, the vehicle travels from waypoint 5 to 6 which exposes the airship to a strong tail wind component. As the airship is heading into the wind, it has to travel backwards to reach the waypoint. However, the airspeed in body x direction remains positive during the entire flight, as shown in figure 5–14.

The simulation results indicate a good high-level controller performance. Therefore, it was decided that the high-level controller could be tested in outdoor flight tests.

5.3 Performance evaluation in flight test

The performance of the high-level controller has been tested in 43 flights with a total airborne time of 72 minutes. The flights were conducted at the same test site as the test flights for the low-level controller described in Section 4.3.

For the flight plan used for these test, two constraints had to be considered. The range of the sonar used to measure the airship height is limited to 6 meters over grassy terrain. The waypoints hence have to be between 3 and 5 meters above ground level to provide sufficient margin in both directions. Secondly, the test area size and the Wifi connection between the airship and the ground station limit the maximum possible distance that the airship can travel during the flight test.

Hence, for flight testing the trajectory was simplified to a planar square with an edge length of 40m at a height of 4m above ground. This trajectory is sufficient to evaluate controller performance in an outdoor environment. Take-off is in the southwest corner of the square. The airship subsequently flies north, then east, south, and finally west back to the starting position. When reaching the starting position, it is commanded to hover at that location.

The controller used during the flight tests corresponds to the architecture used for the simulation test cases 3 and 5. The commanded pitch angle θ_c is kept at zero, and equations (4.34) and (4.35) are used for the computation of the desired velocities in the body x and z directions.

The ground station from which the operator supervises the flight and sends the start and stop commands is located about 40m south and 10m east of the centre of the square. A wind sensor located at the ground station in a height of 5m above the ground records the wind data during the flight.



Figure 5–15: Wind speed measured at the ground station during the first flight test.

The controller gains used in the first flight test presented here correspond to the gains given in table 5–2. The wind conditions measured at the ground station during this flight are shown in figure 5–15. The wind initially blows from the east, turning south towards the end of the test. The wind speed is initially 1m/s but increases to up to 3m/s towards the end of the test. However, it was noted during the flight tests that the conditions at the ground station do not necessarily reflect the conditions at the location of the airship. Obstacles in the vicinity of the test site generated turbulences that led to locally varying wind conditions. Therefore, the measured wind speed only provides a general idea of the prevailing wind conditions.

The path described by the airship during the square tracking is shown in figure 5–16. The path during the hover is shown in figure 5–17. The apparent jumps in position visible in figures 5–16 and 5–17 are a result of the GPS system. When the number of tuned satellites changes, the position reading changes discontinuously.





Figure 5–16: Airship path during square tracking flight test.

Figure 5–17: Position evolution during hover flight test.

Hence, the position data have been represented including the information on the number of tuned satellites.

Apparently, there is an upset at the beginning of the flight, that leads to larger deviations from the desired path and even makes the airship describe a loop. The position deviations are also shown in figure 5–18. After the upset, the vehicle converges well onto the desired path and tracks the remaining three sides of the square with a very good precision. The overshoot at each corner is again a result of the fact that the reference vehicle changes its direction abruptly at the waypoints, and the airship can only perform smooth, continuous changes. The waypoint switches occur at t = 43s, t = 83s, and t = 123s. At t = 163s, the hover segment starts. The airship is kept in hover for 2 minutes.

The reason for the initial upset is not apparent from the data logged during the flight test. However, the wind at the test site proved to be gusty. It is likely that a wind gust hit the airship causing the upset. Simulations with separate wind





Figure 5–18: Position error during square tracking and hover flight test.

Figure 5–19: Airship height during square tracking and hover flight test.

gusts superimposed on the regular turbulent wind allowed to generate similar kinds of loops.

The height evolution during this flight is given in figure 5–19. The target height during square tracking is 4 meters, and during hover, the target height is increased to 5m to verify the vertical tracking function of the high-level controller. The desired height is maintained within 1m for most of the flight. Some peak deviations in the range of 1.5 meters to 2 meters are visible, but these are quickly attenuated.

The evolution of the roll and pitch angles during this flight are shown in figure 5–20, the evolution of the yaw angle is given in figure 5–21. The commanded pitch angle θ_c is zero throughout the entire flight. The value is maintained by the low-level controller within ±10 degrees, i.e. the pitch tracking in the flight test is a bit worse than in the simulation. This phenomenon was previously observed during the low-level controller flight tests and has been discussed at the end of Chapter 4.





Figure 5–20: Roll and pitch angles evolution during the high-level controller flight test.

Figure 5–21: Yaw angle evolution during the high-level controller flight test.

The commanded roll angle during the initial upset is 20 degrees to the left, which corresponds to the maximum permissible value for the commanded roll angle ϕ_c . After a short command of 20 degrees to the right at the first waypoint switching, the roll angle is kept within ±15 degrees. The actual angle tracks the commanded angle well.

Apart from the circle flown during the initial upset, the yaw angle remains within the range of 20 degrees to 200 degrees for most of the flight. This indicates southerly winds during this flight test, which is confirmed by the wind speed measured at the ground station during the flight test, as shown in figure 5–15. The commanded yaw angle is tracked very well.

For brevity, only the results of one more flight test will be presented here. The analysis will be limited to the square tracking part of the flight, as the hover performance of the test presented previously was very satisfactory.



Figure 5–22: Airship path during second flight Figure 5–23: Airship height during test. Second flight test.

This test was conducted later the same day as the test described before. The wind conditions for this test are shown in figure 5–24. The wind speed is again between 1m/s and 3m/s, but the wind direction has changed significantly. At the beginning of the test, the wind blows from the west, but it turns continuously towards a northerly wind. Most of the flight is hence conducted during this northerly wind.

The path and the height evolution during this flight test are given in figures 5–22 and 5–23. During this test, the trajectory is tracked very well with the usual overshoot at the waypoints. The height control exhibits one peak with a deviation of more than 2 meters. For most of the rest of the flight, the height deviation is less than 1m.

The yaw angle during this flight is shown in figure 5–25. The airship heading is limited to northerly directions throughout the flight corresponding well to the wind direction measured at the ground station, as shown in figure 5–24.





Figure 5–24: Wind speed at the ground station during the second flight test.

Figure 5–25: Yaw angle evolution during the second flight test.

Part of the remaining flight tests were used to try different controller parameters. Increasing the gains beyond those given in table 5–2 frequently led to instability and the need to interrupt the flight test. Flight tests using the gains in table 5–2 or lower gains provided generally good results, unless technical issues such as thruster failures required a flight test abortion.

In conclusion, it can be said that the high-level controller designed in this chapter provides good tracking of the desired path, both in the simulation and in flight tests. Discrepancies between the flight tests and the simulation can be tracked to the same reasons already identified at the end of Chapter 4.

CHAPTER 6 Dynamics based wind estimation

It was discussed previously in Chapter 4 that the feedforward part of the controller equations (4.30) and (4.31) contains terms related to the wind speed v_w . For all simulations and flight test results presented in the preceding chapters, these terms were set to 0. In this chapter, the effect on the low-level controller performance of having an estimate for the wind speed v_w will be investigated. Subsequently, a method for estimating the wind without additional sensors will be presented.

6.1 Wind knowledge impact on low-level controller performance

To begin with, it is necessary to reach an understanding of the potential benefits of using various degrees of accuracy of windspeed knowledge in the controller. Two simulations have been conducted, in order to determine the precision that is required for the available wind information to noticeably improve controller performance. The simulated test cases are identical to the simulations with 2m/s wind presented in Section 4.2. However, instead of assuming a value of 0 for the wind speed, the actual wind speed v_w has been used for the respective terms in equations (4.30) and (4.31) for the first simulation. The second simulation uses the average wind speed instead of the actual value for v_w .

Assuming flight close to the ground over flat, level terrain, the average value of the vertical wind speed $w_{w,I}$ is zero and does not show large gusts. This component is hence omitted in this study, and only the impact of knowledge of the horizontal



Figure 6–1: Roll and pitch evolution Figure 6–2: Yaw evolution during the during the simulation with perfect wind simulation with perfect wind knowledge.

wind speed is investigated. For the first simulation presented here, the exact wind information provided to the low-level controller corresponds hence to the $u_{w,I}$ and $v_{w,I}$ components displayed in Figure 4–11, transformed into body frame coordinates. The derivative of the inertial wind speed $\dot{\boldsymbol{v}}_{w,I}$ is not considered in this study as this quantity will be very difficult to determine.

The Euler angle evolution for the simulation with exact information on the horizontal wind speed is given in Figures 6–1 and 6–2. The same test case without wind knowledge is shown in Figures 4–14 and 4–15. Knowing the horizontal components of the wind speed exactly leads to an improvement in tracking of the desired values. The effect is most noticeable in the yaw angle tracking, which showed large deviations from the desired value in the case without wind information as shown in Figure 4–15.





Figure 6–3: Forward and vertical velocity evolution during the simulation with perfect wind knowledge.

Figure 6–4: Control commands during the simulation with perfect wind knowledge.

The velocity evolution with exact wind information is given in Figure 6–3. Comparing with the results without wind speed information, shown in Figure 4–17, shows a similar performance for both cases. The availability of wind speed information has little impact on the velocity tracking.

The thrust commands for this test case are shown in Figure 6–4. Comparing with the thruster commands for the case without wind speed information, given in Figure 4–19, shows no major differences between the two cases.

Knowledge of the exact wind speed therefore allows an improvement of the yaw tracking of the low-level controller without adverse effect on the remaining motion variables. However, exact knowledge of the wind conditions is an unrealistic scenario, as any algorithm or sensor used to determine the wind speed will be subject to a certain level of delay.



Figure 6–5: Roll and pitch evolution Figure 6–6: Yaw evolution during the during the simulation with average wind simulation with average wind knowledge.

To examine what performance improvement is feasible with a wind speed estimate subject to such a delay, another simulation was conducted in which the average wind speed, rather than the actual wind speed, has been provided to the controller. This allows to investigate the effect of knowing the average wind speed. The performance of a wind estimation algorithm being able to identify the actual wind conditions with a certain delay will be somewhere between this scenario and the exact wind information case described previously.

The average wind speed provided to the controller is $\bar{\boldsymbol{v}}_{w,I} = [-1.41 \ 1.41 \ 0]^T \text{m/s}.$ The actual wind speed during the simulation is shown in Figure 4–11.

The roll and pitch values during this simulation are shown in Figure 6–5. The tracking performance is very similar to that depicted in Figure 6–1. The yaw performance for this case is shown in Figure 6–6. The tracking quality of the yaw angle for





Figure 6–7: Forward and vertical velocity evolution during the simulation with average wind knowledge.

Figure 6–8: Control commands during the simulation with average wind knowl-edge.

this test case is significantly better than without any wind knowledge but noticeably less good than in the case of exact wind information.

The velocity tracking information depicted in Figure 6–7 and the control effort shown in Figure 6–8 do not show any significant differences to the cases without any or with exact wind information.

The availability of wind speed data allows a significant improvement in the tracking of the desired yaw angle without adversely affecting the tracking of the remaining motion variables and without increasing the control effort. Better wind speed data leads to better yaw angle tracking, but information on the average wind speed is enough to provide significant yaw angle tracking improvement. The information on the wind conditions allows the controller to better predict the changes in the aerodynamic moments during turns which leads to the improved control performance. It may be possible to further improve the controller performance by retuning the controller gains for the case in which wind speed information is available. The current controller gains are optimized for good performance in the presence of an unknown wind disturbance.

As wind speed information has been shown to improve attitude tracking, the subsequent sections will investigate the possibility of estimating the wind conditions without the need for additional sensors on board the vehicle.

6.2 Wind estimation algorithm

The wind estimation algorithm is intended to estimate the horizontal wind speed based on the difference between the observed and the expected airship response to thruster inputs. Any observed difference is either due to modelling errors or due to the \boldsymbol{v}_w terms in the equation of motion (2.26). Of the external forces \boldsymbol{f}_e and moments \boldsymbol{n}_e that are part of (2.26), only the viscous forces \boldsymbol{f}_v and moments \boldsymbol{n}_v are dependent on the wind speed \boldsymbol{v}_w .

In a first step, the effect of changes in the wind speed v_w on the forces and moments in the body frame directions x, y, and z has been investigated to determine the axes along which the wind speed has a most significant impact. The equilibrium points defined in Table 3–1 are used for this process and the thruster forces required to achieve equilibrium are calculated for four different wind conditions. In the first case, there is no wind at all, while for the remaining cases, the wind speed is 2m/s with different directions.

The thruster forces required for each of these cases to achieve equilibrium are given in Table 6–1. The change of the forces for the different wind speeds is an

Equilibrium	Wind	$f_{p,x,e}$	$f_{p,y,e}$	$f_{p,z,e}$	$n_{p,x,e}$	$n_{p,y,e}$	$n_{p,z,e}$
point	conditions	[N]	[N]	[N]	[Nm]	[Nm]	[Nm]
	Zero Wind	0.17	0	-5.97	0	1.99	0
1	2m/s from the front	0.68	0	-5.97	0	1.99	0
(Forward flight)	2m/s from the right	0.17	8.95	-5.97	0	1.99	15.53
	$2m/s$ at 45° angle	0.50	4.45	-5.97	0	1.99	19.15
	Zero Wind	0.043	0	-6.11	0	2.99	0
2	2m/s from the front	0.38	0	-6.11	0	5.02	0
(Forward and	2m/s from the right	0.043	9.02	-7.10	0	2.92	7.42
upward flight)	$2m/s$ at 45° angle	0.25	4.45	-6.77	0	4.36	13.28
	Zero Wind	0	0	-6.53	0	1.95	0
3	2m/s from the front	0.17	0	-6.53	0	6.00	0
(Vertical ascent)	2m/s from the right	0	9.22	-8.28	0	1.82	-0.70
	$2m/s$ at 45° angle	0.08	4.66	-7.63	0	4.70	7.59
	Zero Wind	0.17	0	-6.06	1.15	1.97	-0.31
4	2m/s from the front	0.68	-0.6	-5.96	1.15	1.97	-0.31
(Steady turn)	2m/s from the right	-0.44	8.76	-7.61	1.15	4.66	14.98
	$2m/s$ at 45° angle	0.06	3.86	-6.74	1.15	5.26	18.36

Table 6–1: Propulsion forces required to achieve equilibrium for different wind conditions.

Regular font: No change from the respective zero wind case.

Italic: Force/moment changed from the respective zero wind case, but change cannot be used for wind estimation.

Bold: Significant change of force/moment from the respective zero wind case, which can be used for wind estimation.

indication on the level of influence that the wind speed has along each axis. The lateral thruster forces are not necessarily zero, but thrust in the body y direction is not feasible on this vehicle. This implies that for these conditions, an equilibrium is not achievable with the current airship setup.

Table 6–1 shows clearly, how the motion is affected by the wind speed. The roll moment n_x is not affected at all by the wind speed and cannot be used for wind speed estimation. The force in the body x direction is naturally influenced by the wind speed component along the body x axis. However, the change in forces is very small, essentially, because the longitudinal drag coefficient is very small. The peak difference noted is 0.5N and, for many cases shown in Table 6–1, it is significantly less than that. This effect is likely less than the overall noise level in the measurements and the effect of thruster misalignment. Therefore the force balance in the body xdirection appears unsuitable for wind estimation purposes.

This highlights the first issue with dynamics based wind estimation. For equilibrium point 1, the only difference between the case without wind and the case with wind from the front is the force balance in the body x direction. For equilibrium point 4, there are additional small differences in the forces in the y and z directions but those are of similarly small magnitude. For flight without a velocity component in the body z direction, estimation of the wind component along the body x direction is hence impossible.

Equilibrium points 2 and 3 feature a velocity along the body z direction. The resulting Munk moment leads to a change in moment around the body y direction for changes in wind speed along the body x axis. The change is quite significant, as can be seen in Table 6–1. The peak change from the zero wind case is 4Nm for equilibrium point 3. In this case, the pitch moment balance could be used for wind estimation purposes. However, the location of the CG and the heaviness of the airship also have a large influence on the pitch moment. Errors in these parameters will hence adversely affect the wind estimation performance. Both the CG and the heaviness depend strongly on the inflation of the airship hull, which changes with ambient temperature as well as exposure to sunlight. They are hence subject to fluctuations even in flight and are especially difficult to know exactly.

The same issue of parametric uncertainty applies to the force balance along the body z axis. The maximum change in force along the body z direction due to the wind speed is in the range of 1.8N. A change in heaviness of 180g due to pressure changes in the airship hull would create a similar effect. This balance is therefore not suitable for wind estimation purposes, either.

In the presence of a side wind component, the force balance along the body y axis and the moment balance about the body z axis are consistently strongly influenced by the wind speed. Therefore, these appear to be the most suitable aspects of the vehicle motion that should be used for wind estimation. Additionally, the CG position and the vehicle heaviness have no influence on these balances in level flight, making them more robust to parametric uncertainty than the force and moment balances discussed above.

This implies that any wind estimation algorithm that has to also cope with uncertainties in thruster alignment, heaviness or CG position, can only estimate the wind speed perpendicular to the airship heading. To get a complete estimate on the horizontal wind speed, regular changes in airship heading will therefore be required. This amounts to an excitation condition for the estimation of the time-varying wind speed.

Keeping in mind that the on-board electronics of the vehicle should be able to execute this estimation in real time, a parameter estimation algorithm with low computational cost has been devised. The algorithm is based on a gradient descent method which is used in [82] to estimate the aerodynamic parameters of a fixed wing aircraft.

The gradient descent method attempts to minimize an error cost function E that depends on the set of parameters d by applying the discrete update law

$$\boldsymbol{d}_{k} = \boldsymbol{d}_{k-1} - c \left. \frac{\partial E}{\partial \boldsymbol{d}} \right|_{k}$$
(6.1)

with c > 0 being the iteration gain.

In the present application, the parameter set d corresponds to the inertial wind speed components in the horizontal plane $u_{w,I}$ and $v_{w,I}$.

Based on the analysis of the effect of the wind speed on the airship motion, the error cost function E is chosen to penalize errors on the force balance along the body y direction and errors on the moment balance about the body z axis. The error cost function E is hence defined as

$$E = \frac{1}{2} \left(n_{p,z,E} - n_{p,z} \right)^2 + \frac{1}{2} f_{p,y,E}^2$$
(6.2)

with $n_{p,z}$ being the actual moment about the body z axis created by the thrusters. It is computed based on the thruster model described in Appendix B.3 which transforms the thrust and tilt commands sent to the thrusters into the actually generated thrust and the actual tilt angle of each thruster. The actual thrust force along the body yaxis $f_{p,y}$ is always zero and has therefore been omitted from (6.2).

The expected quantities $n_{p,z,E}$ and $f_{p,y,E}$ are the thruster moment about the body z axis and the thruster force along the body y axis, respectively, that correspond to the observed airship motion for the current estimate of the inertial wind speed $v_{w,I}$.

The expected quantities $n_{p,z,E}$ and $f_{p,y,E}$ are computed by solving equations (2.26) to (2.28) for the thruster forces \boldsymbol{f}_p and moments \boldsymbol{n}_p . This yields

$$\begin{bmatrix} \boldsymbol{f}_{p,E} \\ \boldsymbol{n}_{p,E} \end{bmatrix} = \bar{\boldsymbol{M}}_{a} \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} - \bar{\boldsymbol{f}}_{k,a} - \bar{\boldsymbol{f}}_{b,g} - \bar{\boldsymbol{f}}_{v}.$$
(6.3)

Taking the second component of $\boldsymbol{f}_{p,E}$ yields $f_{p,y,E}$, while the third component of $\boldsymbol{n}_{p,E}$ gives $n_{p,z,E}$.

Using (6.2), the parameter update law (6.1) becomes

$$\begin{bmatrix} u_{w,I} \\ v_{w,I} \end{bmatrix}_{k} = \begin{bmatrix} u_{w,I} \\ v_{w,I} \end{bmatrix}_{k-1} - c \begin{bmatrix} (n_{p,z,E} - n_{p,z})\frac{\partial n_{p,z,E}}{\partial u_{w,I}} + f_{p,y,E}\frac{\partial f_{p,y,E}}{\partial u_{w,I}} \\ (n_{p,z,E} - n_{p,z})\frac{\partial n_{p,z,E}}{\partial v_{w,I}} + f_{p,y,E}\frac{\partial f_{p,y,E}}{\partial v_{w,I}} \end{bmatrix}_{k}.$$
 (6.4)

In equation (6.3), only the terms $\bar{f}_{k,a}$ and \bar{f}_v depend on the wind speed v_w . Hence, (6.4) can be simplified to

$$\begin{bmatrix} u_{w,I} \\ v_{w,I} \end{bmatrix}_{k} = \begin{bmatrix} u_{w,I} \\ v_{w,I} \end{bmatrix}_{k-1} + c \begin{bmatrix} (n_{p,z,E} - n_{p,z}) \frac{\partial(n_{k,a,z} + n_{v,z})}{\partial u_{w,I}} + f_{p,y,E} \frac{\partial(f_{k,a,y} + f_{v,y})}{\partial u_{w,I}} \\ (n_{p,z,E} - n_{p,z}) \frac{\partial(n_{k,a,z} + n_{v,z})}{\partial v_{w,I}} + f_{p,y,E} \frac{\partial(f_{k,a,y} + f_{v,y})}{\partial v_{w,I}} \end{bmatrix}_{k}.$$
(6.5)

This equation represents an update law for the wind speed estimate that requires low computational effort. The partial derivatives in (6.5) can be computed using the relations given in appendices E.2 and E.3.

6.3 Estimation performance in simulation

The performance of this wind estimation algorithm has been studied in a series of simulations with the iteration parameter set to c = 0.001. All simulations presented here include the parametric uncertainty given in Table 3–3 to verify wind estimation


Figure 6–9: Roll and pitch evolution during the wind estimation simulation with constant wind.

Figure 6–10: Yaw evolution during the wind estimation simulation with constant wind.

performance if the parameters used in the performance algorithm do not exactly match the vehicle parameters. The estimated wind speed is provided to the low-level controller to verify control stability when using the wind speed estimate.

The first simulation has been conducted to verify whether the wind estimation algorithm is capable of identifying a constant north-westerly wind of 2m/s with a simple manoeuvre. In this test case, the airship is controlled by the low-level controller only. The desired roll and pitch angles are zero throughout the entire flight and the desired heading angle alternates between 0° and 90° to fulfill the excitation condition discussed in the previous section. The flight starts with a vertical ascent, and at t = 10s, the desired forward velocity is increased from 0 to 2m/s over a 10s period of time. For the remainder of the flight, the airship flies at a constant forward velocity of 2m/s and a vertical velocity of zero.





Figure 6–11: Forward and vertical velocity evolution during the wind estimation simulation with constant wind.

Figure 6–12: Estimated wind speed during the wind estimation simulation with constant wind.

The attitude evolution during this flight is shown in Figures 6–9 and 6–10. The cause for the initial attitude upset is again the simulated parametric uncertainty. Once the integrative terms have compensated for the modified CG location and the modified heaviness, the desired angles are track very well and precisely. The yaw angle tracking exhibits a certain delay, as the derivative of the desired quaternion \dot{q}_d is set to zero in these simulations.

The velocity evolution during this simulation is shown in Figure 6–11. Apart from the initial upset, the desired values are tracked very well.

The actual wind speed during this simulation is given by the dashed lines in Figure 6–12, the result of the estimation is given by the solid lines in the same figure. The wind estimation converges quickly on the actual wind speed. Within less than 20s, the estimation error is less than 0.1m/s for each component. For the remainder

of the flight, the estimation error is kept within 0.1m/s. This is a surprising result, as the wind estimation has already converged on the actual wind speed far before the first commanded change in airship heading angle. The initial attitude upset appears to have helped satisfy the excitation condition for the estimation algorithm allowing for this fast convergence.

The same simulation has been repeated with a gusty rather than constant wind. The attitude and velocity evolution in this test case is very similar to the case with constant wind and will be omitted for brevity. The wind estimation result for this case is shown in Figure 6–13. Similarly to the previous case, the wind estimation converges quickly to values close to the actual wind speed.

However, between t = 0s and t = 30s and between t = 100s and t = 130s, when the airship is heading straight north, the wind estimation is only able to detect gusts in the eastern direction. Gusts in the northern direction are effectively filtered out. Similarly, between t = 50s and t = 80s, when the airship is heading straight east, the north component of the wind is estimated well but the east component exhibits larger errors. This fits well with the discussion of the previous section in which it was concluded that the wind component along the body x axis of the vehicle cannot be estimated.

The wind estimation has been shown to work in the simulation for a simple airship trajectory. To verify the performance during a more complex manoeuvre, the test case from Section 4.2 with 2m/s wind will be taken up again. This was also the test case used in Section 6.1 to analyze the possible effect of wind information on the low-level controller performance.





Figure 6–13: Estimated wind speed during the wind estimation simulation with turbulent wind.

Figure 6–14: Estimated wind speed during the simulation of the 2m/s test case from Section 4.2.

As previously mentioned, the result of the wind estimation algorithm is provided to the low-level controller. Therefore, the simulations allow verification of two things: the performance of the wind estimation algorithm during this trajectory and the effect that this estimate has on the low-level controller performance.

The trajectory simulated in this flight consists of three straight legs in the northsouth direction, but no straight legs in the east-west direction. The estimated wind speed in given in Figure 6–14. The wind component in the east-west direction is estimated very well, including gusts. The estimation of the north-south component of the wind is less precise with the gusts being filtered and a peak error in the range of 1m/s. However, the overall estimate of this component gives the correct order of magnitude for the wind, and only towards the end of the simulation are larger discrepancies observed. As the wind estimation can provide very good estimates for the cross-wind, the estimate in the east-west direction is significantly better than that in the north-south direction.

The motion variables for this simulation are given in Figures 6-15 to 6-18. The tracking improvement due to the wind estimation is very close to the case with exact wind information shown in Figures 6-1 to 6-4.

The tracking performance for the roll angle is similar in both cases. Pitch tracking is slightly worse towards the end of the simulation when using the estimated wind, with the actual angle showing two peaks with a deviation in the range of 5 to 10 degrees. This is very likely due to the discrepancy between the estimated and the actual wind speed component in the north-south direction, leading to a wrong prediction of the Munk moment about the pitch axis.

Disregarding the first 20s until the wind estimation has converged onto the actual wind speed, the yaw angle tracking is almost perfect in both cases. Velocity control also shows similar performance levels between the two cases. The vertical velocity control actually appears to be better when using the wind speed estimate, whereas the forward velocity control is slightly better when using the exact wind speed data. The control effort is very similar for both cases.

In the scope of this thesis, the wind estimation performance will be investigated for one more test case: flight with the high-level controller active. This test case is identical to test case 5 described in Section 5.2. The results for this case are very similar to the results shown in Section 5.2, the respective figures have hence been omitted for brevity.



Figure 6–15: Roll and pitch evolution during the simulation of the 2m/s test case from Section 4.2.



Figure 6–16: Forward and vertical velocity evolution during the simulation of the 2m/s test case from Section 4.2.



Figure 6–17: Yaw evolution during the simulation of the 2m/s test case from Section 4.2.



Figure 6–18: Control commands during the simulation of the 2m/s test case from Section 4.2.



Figure 6–19: Estimated wind speed during the trajectory tracking flight.

The wind estimate computed during this flight as well as the actual wind conditions are shown in Figure 6–19. Most of the time the wind speed estimate corresponds very well to the actual wind speed. Major differences appear primarily for the east-west component of the wind speed and generally do not exceed 0.5m/s. For large parts of the flight, the yaw angle, shown in Figure 5–11, is in the range for 50° to 100°, corresponding to an approximate easterly direction. This then explains the good estimation of the wind in the north-south direction and the larger discrepancies for the wind in the east-west direction.

Similarly to the test cases presented in Section 5.2, the RMS of the position error has been computed for this test case. The RMS values are given in Table 6–2. For comparison, the RMS values for test case 5 in Section 5.2 have been included in the table.

The data provided in Table 6–2 shows that the wind estimation has very little impact on the trajectory tracking performance. During trajectory tracking, the

Low-level ctrl	Trajectory tracking			Hover		
wind data	North [m]	East [m]	Down [m]	North [m]	East [m]	Down [m]
yes	0.8000	0.7540	0.2111	0.3821	0.3385	0.1674
no (TC5)	0.7914	0.7022	0.1496	0.4055	0.3393	0.1205

Table 6–2: Root mean square position error for the trajectory tracking with wind estimation.

availability of the wind estimate actually descreases tracking quality. During hover, the tracking is improved in the north-south direction at the expense of the height tracking precision.

The wind estimation algorithm designed in the previous section has been shown to compute a good wind speed estimate in simulation conditions. It has been shown that using the wind speed estimate in the low-level controller allows an improvement in the yaw tracking of the low-level controller without adversely affecting the other motion variables. However, when using the wind estimation algorithm in combination with both the low-level and the high-level controllers, the availability of the wind information seems to have little effect.

6.4 Considerations for wind estimation during flight tests

The wind estimation has been shown to work well in the simulation in the presence of sensor noise and parametric uncertainty. However, some uncertainties that can adversely affect the wind estimation performance are not modelled in the simulation. These uncertainties include, among others, misalignment of the thrusters and the IMU.

The thrusters are mounted to the flexible airship hull using velcro patches and cords. The tripod design allows reasonable alignment of the thrusters to ensure that a tilt angle of 0° properly corresponds to purely vertical thrust when the airship is at rest. However, due to the flexibility of the airship hull, this alignment is subject to changes in flight, depending on the airship motion, the generated thrust and the current tilt angle.

The IMU is also mounted to the flexible airship hull with a velcro patch. The size of the IMU is very small with a length of 44mm and a width of 24mm. Due to the type of attachment and the small size of the unit, correct alignment of the IMU is very difficult and a precision better than about 2° in all Euler angles is not feasible.

While the flight tests presented in the previous chapters show that the controllers are robust against these types of disturbances, the force and moment balances used for the wind estimation may be more susceptible to these uncertainties. Therefore, in addition to the simulations presented in the previous section, simulations investigating the effects of thruster and IMU misalignment have been conducted.

These simulations are identical to the trajectory tracking simulation from the previous section, shown in Figure 6–19, with the difference that bias terms have been added to the Euler angle measurements and the thruster tilt angles.

Misalignment of the thruster tilt angles may influence the moment balance about the body z axis. The strongest effect would be created if both thrusters on one side have a misalignment forward, and the thrusters on the other side have a misalignment backwards, leading to an offset in the moment generated by the thrusters.

Two simulations have been conducted to investigate this effect. Using the convention for the tilt angles given in Appendix B.3, the actual tilt angle of the thrusters



Figure 6–20: Estimated wind speed with Figure 6–21: Estimated wind speed with a bias of $\pm 1.5^{\circ}$ on the thruster tilt angles. a bias of $\pm 3^{\circ}$ on the thruster tilt angles.

on the left side has been increased by 1.5° over the commanded angle for the first simulation. The actual tilt angle for the thrusters on the right side has been reduced by 1.5° respectively. For a commanded tilt angle of 0° on all four thrusters, the thrusters will hence generate a moment about the body z axis, while the wind estimation assumes this moment to be zero. For the second simulation these values have been doubled, leading to tilt angle offsets of $+3^{\circ}$ on the left side and -3° on the right side.

The results of the wind speed estimation for these two cases are shown in Figures 6–20 and 6–21. The estimated wind speed for the same test case without any bias terms is given in Figure 6–19.

The tilt angle misalignment clearly affects the wind estimation results. A tilt angle bias of 1.5° leads to a visible deterioration of the estimation results, but the estimated wind speed could still be considered usable, as the discrepancies do not





Figure 6–22: Estimated wind speed with Figure 6–23: Estimated wind speed with a bias of 1° on the IMU roll angle.

a bias of 2° on the IMU roll angle.

exceed 0.5 m/s. If all tilt angles are misaligned by 3° , the discrepancies sometimes increase to values beyond usability.

The effect of IMU misalignment has also been investigated in a series of simulations. Bias terms on the Euler angles can adversely affect the force balance along the body y axis. Especially bias terms on the roll angle are expected to have a large influence on this force balance. The attitude reading of the IMU defines the orientation of the body frame. A misalignment in roll hence represents a rotation of the airship body, including the thrusters, within the body frame about the x axis. In this case, the force generated by the thrusters also contains a component in the body frame y direction. The wind estimation however assumes the thrust in the ydirection to be zero, therefore an error is introduced into the wind estimation.

The effect of a roll misalignment is shown in Figures 6-22 and 6-23. With a roll misalignment of 1°, the performance of the wind speed estimation has already decreased significantly. The difference between estimated and actual wind speed





Figure 6–24: Estimated wind speed with Figure 6–25: Estimated wind speed with a bias of 2° on the IMU pitch angle.

a bias of 2° on the IMU yaw angle.

frequently reaches 0.5 m/s, but the results may still be useable. When the roll misalignment is increased to 2°, the discrepancies between the actual and the estimated wind speed increase beyond useable levels.

The effects of IMU misalignment in pitch and vaw are shown in Figures 6–24 and 6–25, respectively. Both misalignments lead to a larger discrepancy between actual and estimated wind speed at the start of the simulation. Once the estimated wind speed converges onto the actual wind speed around t = 300s, the changes in wind speed are tracked correctly for both cases. Overall the estimation performance is acceptable in the presence of these misalignments, with the maximum discrepancies being in the range of 0.5 m/s.

For flight testing, the wind estimation will be subject to a combination of all the misalignments investigated separately above. The best case scenario for flight testing is a misalignment of the IMU in the range of 1° for each axis and a thruster tilt misalignment also in the range of 1°. Assuming an IMU misalignment of 1° on



Figure 6–26: Estimated wind speed with a bias of 1° on the IMU Euler angles and the thruster tilt angles.

each axis and a thruster misalignment of $+1^{\circ}$ on the left side and -1° on the right side, the wind estimation algorithm provides the results shown in Figure 6–26.

Apparently, the combination of all the misalignments upsets the wind estimation algorithm sufficiently to make the results unusable. In most cases, the current airship setup will lead to significantly larger misalignments than those simulated in this last test case. The wind estimation results from the flight tests can hence be expected to deliver unusable results. The wind estimation results during the two high-level controller test flights presented in Section 5.3 are given in Figures 6–27 and 6–28. The actual wind data given in Figures 6–27 and 6–28 do not show the wind speed at the airship location, as it was the case for the simulations presented earlier. The wind speed is measured at a height of 5m in the vicinity of the ground station; the actual wind speed at the vehicle position is unknown.

While the wind estimation results from the first flight, as shown in Figure 6-27, show some correlation between the estimated and the actual wind speed, the wind





Figure 6–27: Estimated wind speed during the first high level controller flight test.

Figure 6–28: Estimated wind speed during the second high level controller flight test.

estimation during the second flight test, as shown in Figure 6–28 provides unusable results. However, the fact that the measurements shown were made far from the airship means that this comparison should be interpreted with caution. The wind conditions at the location of the airship may therefore be very different than the measured wind.

However, the discrepancies between wind speed estimate and the measured wind speed depicted in Figure 6–28 are significantly greater than what could be explained by the shift in location. For further investigation of the wind estimation performance in flight tests, it is recommended that the airship setup be modified to guarantee alignment of the IMU and the thrusters, or to develop a strategy to precisely determine the IMU misalignment before lift-off.

A design study that addresses this issue has been performed at McGill University. Mounting both forward thrusters and both rear thrusters onto two hoops



Figure 6–29: Design study for improved thruster and IMU alignment.

made of a light-weight material, such as carbon fibre, would allow elimination of thruster alignment issues due to the flexibility of the hull. Figure 6–29 illustrates this concept. Mounting the IMU onto these frames also, will ensure a precise and reproducible alignment between the IMU and the thrusters. The hoops of this design study have already been manufactured but could not yet be tested due to time constraints.

On larger scale aircraft, the orientation of the propulsion units and the inertial measurement devices is generally known with a very high precision. Therefore, the alignment issue is solved when transferring this technology to larger vehicles.

CHAPTER 7 Conclusion

The main goal of this research was to investigate the control of an unmanned finless airship with the aim of achieving autonomous trajectory tracking capability with the best possible attenuation of disturbances such as wind gusts.

The key advantages of a finless airship design are the lower structural weight and the higher manoeuvrability compared to airships with fins. In this work, the aerodynamic instability resulting from the finless design has been investigated in detail. The added mass effect encountered by lighter-than-air vehicles leads to the Munk moment which tends to rotate the airship hull so that it travels perpendicular to its axis of revolution. Travel along the elongated direction of the hull constitutes an unstable equilibrium that needs to be actively maintained by an onboard controller.

The vehicle used in this study features four tiltable thrusters that are used to control the vehicle motion. The attitude control achievable by the four thrusters is independent of the speed of the vehicle, constituting another advantage of the finless design.

Two different low-level control algorithms have been studied to control the attitude and velocity of the vehicle using individual thrust and tilt commands to each of the four thrusters. The core design criteria for the low-level controller were the elimination of the inherent instability and the ability to cope with parametric uncertainties and external disturbances. The first algorithm was based on linear H_{∞} control theory. A particular novelty of this approach is the use of H_{∞} control in combination with a thruster allocation algorithm using quadratic programming. The H_{∞} controllers could provide good attitude and velocity tracking under idealized conditions. However, when considering all the uncertainties present in the actual vehicle, the closed-loop system proved unstable.

The second low-level control algorithm employed nonlinear control techniques to better account for the highly nonlinear dynamics of the vehicle. The controller was based on Lyapunov and Backstepping techniques using feedforward terms to account for the vehicle dynamics. Similarly to the study on the H_{∞} controller, this control algorithms also features thruster allocation for optimal use of the available actuation. The controller explicitly considers the dominant aspects of the airship dynamics, but these can be easily changed to account for the dynamics of other vehicles by adapting the feedforward terms. This controller was shown in simulation to be very robust with respect to the uncertainties expected on the actual vehicle. Subsequent flight testing confirmed this robustness and demonstrated good controller performance even in the presence of natural wind gusts.

Subsequently, a high-level control algorithm was developed to augment the airship control for the trajectory tracking task. The high-level controller computes desired attitude and velocity values based on the difference between the actual and the desired airship position. The tracking of these values is then ensured by the nonlinear low-level controller. This particular controller is designed such that it takes advantage of the fact that the vehicle can travel in all directions while having a preferred direction of travel.

The high-level control algorithm was chosen such that is allows both trajectory tracking and hover to be performed with a single control law, eliminating the need to switch between two different control laws when transitioning from forward flight to hover or vice versa. Its design aims to point the airship into the oncoming airflow, minimizing drag as well as the destabilizing Munk moment, and hence reducing the control effort.

The high-level controller is found in simulation to provide good trajectory tracking in the presence of a simulated wind disturbance. Subsequent flight testing confirmed the simulation results. The controller was able to successfully track the desired trajectory and subsequently hover at the final waypoint despite the presence of natural wind gusts.

The last part of this research was the development of a wind estimation algorithm that computes an estimate of the current wind speed based on the observed airship motion without the need for additional sensors on the vehicle. The availability of wind speed data would allow to improve the controller performance, as the airship dynamics can be predicted more precisely.

A particular challenge when performing wind estimation for finless airships lies in the fact that the vehicle can encounter any range of angles of attack and angles of sideslip, which are both not measured. Therefore, the wind estimation is based on the force and moment balances about the axes most affected by the wind speed. It is found to work well in simulation, and, within limits, it is even capable of identifying short-term gusts. However, some unmodelled effects, such as misalignment of the thrusters and the attitude sensor, jeopardize the wind estimation success. Wind estimation data from flight tests give wrong results, confirming the sensitivity to these effects.

7.1 Future work

This thesis covered a range of aspects related to the autonomous trajectory tracking task for unmanned, finless airships. Therefore, each topic could only be explored to a limited depth. For further study of the individual aspects, the following research topics are suggested.

7.1.1 Low-level control using H_{∞} control

The H_{∞} controller designed in Chapter 3 was not able to deal with the thruster actuation delay present on the actual vehicle. In the design of the controller, a first order low-pass filter was used to account for this delay. A more detailed approach to representing this delay in the controller design might allow creation of an H_{∞} controller that is capable of dealing with the delay and could subsequently be deployed in flight testing.

7.1.2 Nonlinear low-level controller design

The nonlinear low-level controller designed in Chapter 4 provides a very good basis for the low-level control task. The key issue encountered with this controller is its behaviour in the presence of actuator saturation. If the desired actuation forces and moments issued by the controller cannot be achieved, the airship motion encounters an upset. While the controller is well capable of attenuating the upset, once the actuator saturation has ended, an approach to avoid the initial upset would be desirable.

Modifying the control law such that the actuator saturation is explicitly considered may allow a reduction of the strength of the motion upset. However, the actuation saturation limits depend on the current thruster forces and tilt angles, making this task very challenging.

7.1.3 Path tracking controller

The current design of the high-level controller presented in Chapter 5 uses a cascaded control architecture with the low-level controller providing the inner loop and the high-level controller the outer loop. The stability of each loop has been established separately, but the overall closed-loop behaviour was only analyzed in simulation and flight test.

Integrating the low-level control law into the high-level controller analysis may allow an overall stability proof to be devised and may provide constraints on the controller parameters that need to be fulfilled for guaranteed stability. Due to the complexity of the low-level control law, this is not a straight forward task.

The path planning used for testing the high-level controller assembled a trajectory consisting of straight lines. This led to a sudden increase in position error at each waypoint as the airship cannot follow the sudden change in direction. Using a more advanced path planning technique that considers physical constraints of the vehicle will allow smoother tracking of the desired trajectory.

7.1.4 Wind and parameter estimation

The key issue identified for the wind estimation is the lack of precision with which the sensor and actuators are mounted to the flexible airship hull. This issue can either be solved by modifying the airship design, which will be discussed in the next section, or by implementing a procedure that allows to determine the misalignment prior to take-off.

If the wind estimation works well after elimination of the alignment issues, it may be desirable to extend the estimation to other vehicle parameters. For example, the heaviness and the CG location are difficult to determine, as both are functions of the inflation pressure of the hull. Estimating these parameters in real time may allow adjustment of the respective controller parameters, thus improving control performance.

7.2 Transfer to other vehicles

The transfer of the controller algorithms designed in this thesis to other vehicles than the Quanser ALTAV MkII could be investigated. Quadrocopters also allow travel in all directions, using differential thrust for attitude and velocity control. The control algorithms presented in this work may hence be easily transferable to this type of vehicles. Larger scale airships that use differential thrust for motion control provide another opportunity for the use of the control algorithms derived in this work.

The wind estimation developed in Chapter 6 can also be used on other vehicles that are sensitive to wind gusts by adapting the equations of motion employed in the algorithm. Other types of airships and quadrocopters are again the ideal candidates for such a transfer, but even a transfer to fixed wing vehicles could be studied. The fact that fixed wing aircraft are limited in their directions of travel might facilitate wind estimation in this case.

7.3 Vehicle modification recommendations

Based on the experience collected during the flight tests with the current vehicle, the following improvements to the vehicle design are suggested.

A frequently encountered issue with the current airship design is the lack of downward thrust, which leads to reduced control authority in airship roll and pitch, especially for low airship heavinesses. Modifying the thruster arrangement such that downward thrust can be generated, will significantly improve the controllability of the vehicle. Downward thrust can be achieved either by increasing the tilt angle range to 360°, installing reversible pitch props, using motors with reversible direction of rotation, or by installing additional thrusters that are oriented downwards.

The alignment issue with the IMU and thrusters could be addressed by building a rigid, light-weight airship frame on which the IMU and the thrusters will be mounted. The frame will allow to align the thrusters and the IMU in the laboratory and fix their alignment with respect to each other. It will also inhibit changes of the thruster orientation during the flight, as they are currently encountered due to the flexible airship hull. A first prototype of such a structure has already been built, but could not yet be tested.

APPENDIX A Frequently used mathematical relations

A.1 Coordinate transformations

The following relationships describe the transformations between inertial and body-fixed velocities. They are identical for the airship velocity and the wind speed.

$$oldsymbol{v} = oldsymbol{R}oldsymbol{v}_I$$
 $oldsymbol{v}_w = oldsymbol{R}oldsymbol{v}_{w,I}$
(A.1)

The first derivative of this equation gives the relationship to transform the change of velocities between the frames:

$$\dot{\boldsymbol{v}} = \dot{\boldsymbol{R}} \boldsymbol{v}_I + \boldsymbol{R} \dot{\boldsymbol{v}}_I$$

$$\dot{\boldsymbol{v}}_w = \dot{\boldsymbol{R}} \boldsymbol{v}_{w,I} + \boldsymbol{R} \dot{\boldsymbol{v}}_{w,I}$$
(A.2)

The direction cosine matrix \mathbf{R} for the Euler angle sequence $\psi \to \theta \to \phi$ can be found in [71]. It can be computed from the Euler angles as

$$\boldsymbol{R} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta\\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi\\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$
(A.3)

and from the quaternion as

$$\boldsymbol{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}.$$
 (A.4)

The transformation from Euler angles to quaternions is

$$q_{0} = \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)$$

$$q_{1} = \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)$$

$$q_{2} = \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)$$

$$q_{3} = \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)$$
(A.5)

and the reverse transformation is given by

$$\phi = \operatorname{atan2} \left(2(q_0 q_1 + q_2 q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2 \right)$$

$$\theta = \operatorname{arcsin} \left(2q_0 q_2 - 2q_1 q_3 \right)$$

$$\psi = \operatorname{atan2} \left(2(q_0 q_3 + q_1 q_2), q_0^2 + q_1^2 - q_2^2 - q_3^2 \right).$$
(A.6)

APPENDIX B External forces and moments acting on the airship

B.1 Gravity and buoyancy forces

The computation of $f_{b,g}$ and n_g is a straight forward task depending on airship attitude, displaced air and location of the centre of gravity. The displaced mass of air m_D is

$$m_D = \rho V \tag{B.1}$$

with ρ being the density of the fluid surrounding the vehicle and V being the volume of the vehicle hull. The body fixed frame has its origin at the centre of buoyancy, hence the buoyancy does not generate a moment. Using the earth acceleration g, gravity and buoyancy forces can be computed to

$$\boldsymbol{f}_{b,g} = (m - \rho V) \boldsymbol{R} \begin{bmatrix} 0\\0\\g \end{bmatrix}.$$
(B.2)

and the gravity moment is

$$\boldsymbol{n}_{g} = m\boldsymbol{C}^{\times}\boldsymbol{R} \begin{bmatrix} 0\\0\\g \end{bmatrix}.$$
(B.3)

B.2 Viscous forces

The viscous forces due to the translational motion of the airship have been derived in detail in [16] and have been adapted to the properties of the Quanser MkII in [1]. The derivation is based on the methods described in [21]. The viscous forces are assumed to act at the aerodynamic centre which is located at x_{ac} on the axis of revolution of the airship hull. The y_{ac} and z_{ac} coordinates in the body-frame are respectively zero. The relevant airspeed v_{ac} at this point is

$$\boldsymbol{v}_{ac} = \boldsymbol{v} - \boldsymbol{v}_w + \boldsymbol{\Omega}^{\times} \begin{bmatrix} x_{ac} & 0 & 0 \end{bmatrix}^T.$$
(B.4)

Based on this velocity, the dynamic pressure q_0 can be computed based on the assumption of incompressible flow to

$$q_0 = \frac{1}{2} \rho \boldsymbol{v}_{ac}^T \boldsymbol{v}_{ac}. \tag{B.5}$$

The angle of attack α is the angle between the direction of v_{ac} and the axis of revolution of the airship hull. It is given by

$$\alpha = \tan^{-1} \left(\frac{\sqrt{v_{ac}^2 + w_{ac}^2}}{u_{ac}} \right) \tag{B.6}$$

In [1], the equation for the total viscous force N acting perpendicular to the body x axis, given in [16], has been reduced by the potential flow terms and the integrals have been solved analytically. This yields

$$N = \eta C_{D_n} A_p q_0 \sin^2 \alpha \tag{B.7}$$

The quantities η , C_{D_n} , and A_p are vehicle-specific parameters: η represents the crossflow efficiency factor, A_p the planform area of the airship hull, and C_{d_n} the crossflow drag coefficient. The first two are constant for a given hull geometry, the latter depends on the crossflow Reynolds number. This dependency is described in detail in [1] and will be omitted here for brevity. All of the aerodynamic parameters of the QuanserMkII as determined in [1] are shown in Table B–1.

As N acts perpendicular to the body x axis, at a point that is x_{ac} away from the body frame origin, it generates a viscous moment M of

$$M = \eta C_{D_n} A_p x_{ac} q_0 \sin^2 \alpha \tag{B.8}$$

The axial drag D of the airship is also given in [16] for small angles α . In [1] it has been extended to the full range of 0 to 180 degrees for α . The associated equation is

$$D = \begin{cases} -q_0 C_A A \cos^2 \alpha & \text{if } u_{ac} \ge 0\\ q_0 C_A A \cos^2 \alpha & \text{if } u_{ac} < 0 \end{cases}$$
(B.9)

with A being the frontal area of the airship and C_A being the axial drag coefficient.

Depending on the hull shape, the axial drag coefficient C_A may be different for $u_{ac} \ge 0$ and $u_{ac} < 0$. However, for the vehicle studied here, it is acceptable to assume the same value for both cases.

Parameter	Symbol	Value		
Airship mass (incl. Helium)	m	6.346kg		
CG-Position	c	$[0.032 \ 0 \ 0.1165]^T \ [m]$		
Airship Volume	V	$4.765 \ [m^3]$		
Added mass matrix	$oldsymbol{A}_m$	diag(0.638, 4.693, 4.693) [kg]		
Added inertia matrix	$oldsymbol{A}_J$	diag $(0, 3.389, 3.389)$ [kg m ²]		
Crossflow efficiency factor	η	0.5921 [-]		
Airship planform area	A_p	$5.229 \ [m^2]$		
Airship frontal area	A	$1.740 \ [m^2]$		
Aerodynamic centre location	x_{ac}	-0.076 [m]		
Crossflow drag coefficient	C_{D_n}	0.26-1.2 [-]		
Axial drag coefficient	C_A	0.041 [-]		
Inertia values				
Inertia about x axis	I_{xx}	$3.038 [kg m^2]$		
Inertia about y axis	I_{yy}	$7.627 \; [\mathrm{kg} \; \mathrm{m}^2]$		
Inertia about z axis	I_{zz}	$8.665 \; [\mathrm{kg} \; \mathrm{m}^2]$		
xz product of inertia	I_{xz}	$-0.0815 \ [kg m^2]$		

Table B–1: Airship properties as determined in [1].

The force N acts in the yz plane. It can be decomposed into components in y and z direction using the relation of v_{ac} and w_{ac} to the total crossflow airspeed $\sqrt{v_{ac}^2 + w_{ac}^2}$. This is shown in detail in [16], yielding

$$\boldsymbol{f}_{v} = \begin{bmatrix} \pm q_{0}C_{A}A\cos^{2}\alpha \\ -\eta q_{0}C_{D_{n}}A_{p}\sin^{2}\alpha \frac{v_{ac}}{\sqrt{v_{ac}^{2}+w_{ac}^{2}}} \\ -\eta q_{0}C_{D_{n}}A_{p}\sin^{2}\alpha \frac{w_{ac}}{\sqrt{v_{ac}^{2}+w_{ac}^{2}}} \end{bmatrix}.$$
 (B.10)

The split of M into its y- and z- components is done analogously giving

$$\boldsymbol{n}_{v} = \begin{bmatrix} 0 \\ \eta q_{0} C_{D_{n}} x_{ac} A_{p} \sin^{2} \alpha \frac{w_{ac}}{\sqrt{v_{ac}^{2} + w_{ac}^{2}}} \\ -\eta q_{0} C_{D_{n}} x_{ac} A_{p} \sin^{2} \alpha \frac{v_{ac}}{\sqrt{v_{ac}^{2} + w_{ac}^{2}}} \end{bmatrix}.$$
 (B.11)

Note that there is no viscous moment around the x axis as the viscous force N acts along the x axis.

The current model does not include any rotational viscous damping terms. At the time of this work, no model of these forces and moments was available for the Quanser MkII. However, since these terms will have a damping, i.e. stabilizing effect, they can be safely omitted from the controller design without risking instability.

B.3 Propulsion forces

The thruster forces \boldsymbol{f}_p and moments \boldsymbol{n}_p are very particular to each vehicle as they depend on the type of propulsion system installed, as well as the location and number of individual thrusters.

The propulsion system for the Quanser MkII consists of 4 electrical thrusters. The thrusters are installed roughly in the xy plane of the vehicle, in order not to keep the CG close to the centre of volume. The thrusters are installed symmetrically to the xz plane, with the forward thrusters about 1m forward of the body frame origin, and the rear thrusters about 1m aft of the origin. The distance of the thrusters from the xz plane is also about 1m for all thrusters.

The thrusters are tiltable by a range of 180 degrees, enabling them to produce thrust forward, upwards and backwards, but not downwards. The thrust force F_i and the tilt angle μ_i can be individually adjusted for each thruster, giving the actuation 8 degrees of freedom. The maximum thrust of each thruster has been determined in [1] to be 11N.

The dynamics of the thrusters have been investigated in detail in [1] for the thrust at zero airspeed. This work has then been extended in [81] to include studies on the thrust for non-zero airspeeds. Both references come to the conclusion that the transient behaviour of the thrusters can be described with a first order low-pass filter with variable gains and time constants. In [81], the effect of non-zero airspeeds was found to be negligible for the speeds at which the Quanser MkII will be operated.

The transient dynamics of the thrusters have been modelled in [1] using a discrete transfer function of the type

$$G(z) = \frac{a_1 z^{-1}}{1 + b_1 z^{-1}} \tag{B.12}$$

with a sampling time of 2.5ms. Equation (B.12) relates the thrust generated by the thrusters to the dimensionless thrust control input, which ranges from 0.19 at idle to 0.5 for maximum thrust. The values for a_1 and b_1 for a given command input as well as the associated stationary thrust are given in [1]. In addition to the transient dynamics, a time delay in the thruster speed controller has been identified in [81]. Accordingly, a delay of 85ms was included in the thruster model.

The tilting actuators used onboard the Quanser MkII are Hitec HS-322HD servos. In [1], the dynamics of the servos have also been investigated experimentally. The servo specifications indicate a tilting speed of 60° in 0.19s, which corresponds to 316° /s, but the experiments showed a slightly slower speed of 287° /s. Also, a time delay of approximately 50ms was observed. The servo modelling in the simulation hence features a rate limiter to reflect the maximum tilting speed of 287° /s as well as the observed delay of 50ms.

The above covers the dynamics of each individual thruster based on the desired thrust and tilt angle values. These need to be transferred into the respective total forces and moments acting on the airship in order to determine the terms f_p and n_p .

The convention for the tilt angle is that 0° is thrust vertically upwards, 90° is horizontal thrust forward and -90° is horizontal thrust backwards. With this convention the total propulsion force generated by the thrusters is

$$\boldsymbol{f}_{p} = \begin{bmatrix} \sum_{i=1}^{4} \sin \mu_{i} F_{i} \\ 0 \\ -\sum_{i=1}^{4} \cos \mu_{i} F_{i} \end{bmatrix}.$$
 (B.13)

To compute the moments generated by the thrusters, the location at which each thrust force acts needs to be known. The vector $\mathbf{r}_{T,i}$ describes the intersection of the tilting axis with the propeller axis of rotation for the *i*th thruster. This definition ensures that the thrust force acts on this point at all tilt angles μ_i . The numerical values for $\mathbf{r}_{T,i}$ for the current configuration of the Quanser MkII are given in Table B–2.

Thruster number i	$r_{T,i,x}$	$r_{T,i,y}$	$r_{T,i,z}$
1	1.1721	0.9935	0
2	-1.0245	0.9774	0
3	-1.0245	-0.9774	0
4	1.1721	-0.9935	0

Table B–2: Thruster location as given in [1].

The moments generated by the thrusters can now be calculated to be

$$\boldsymbol{n}_{p} = \begin{bmatrix} -\sum_{i=1}^{4} r_{Ti,y} \cos \mu_{i} F_{i} \\ \sum_{i=1}^{4} (r_{Ti,x} \cos \mu_{i} + r_{Ti,z} \sin \mu_{i}) F_{i} \\ -\sum_{i=1}^{4} r_{Ti,y} \sin \mu_{i} F_{i} \end{bmatrix}$$
(B.14)

with the terms related to $r_{T,i,z}$ omitted as these are zero.

Besides the actual thrust forces discussed above, the thrusters also generate a reaction torque. Also when changing the axis of rotation for the motors, gyroscopic effects may create additional moments. At the time of this work, no model for either of these effects was available. Due to the light weight of the thrusters and the low thrust levels, it is hence assumed that they are negligible.

$\begin{array}{c} \textbf{APPENDIX C} \\ \textbf{Control error weighting function } \boldsymbol{W}_{e}(s) \end{array}$

The control error weighting function $W_e(s)$ is used to penalize the tracking error of the H_{∞} controller designed in chapter 3. The function represents a diagonal state-space system with the state equations

$$\dot{\boldsymbol{x}}_e = \boldsymbol{A}_e \boldsymbol{x}_e + \boldsymbol{B}_e \boldsymbol{u}_e \tag{C.1}$$

$$\boldsymbol{y}_e = \boldsymbol{C}_e \boldsymbol{x}_e + \boldsymbol{D}_e \boldsymbol{u}_e.$$
 (C.2)

The variable u_e represents the input to the control error weighting function, the variable y_e represents its output. The internal states of the function are given by x_e . The state matrices are:

$$\boldsymbol{A}_{e} = \begin{bmatrix} -0.4772 & 0 & 0 & 0 & 0 \\ 0 & -0.906 & 0 & 0 & 0 \\ 0 & 0 & -9.975 & 0 & 0 \\ 0 & 0 & 0 & -4.403 & 0 \\ 0 & 0 & 0 & 0 & -4.024 \end{bmatrix}$$
(C.3)

$$\boldsymbol{B}_{e} = \begin{bmatrix} 1.909 & 0 & 0 & 0 & 0 \\ 0 & 4.53 & 0 & 0 & 0 \\ 0 & 0 & 598.5 & 0 & 0 \\ 0 & 0 & 0 & 264.2 & 0 \\ 0 & 0 & 0 & 0 & 241.4 \end{bmatrix}$$
(C.4)
$$\boldsymbol{C}_{e} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(C.5)
$$\boldsymbol{D}_{e} = \begin{bmatrix} 0.008 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0.12 & 0 \\ 0 & 0 & 0 & 0 & 0.12 \end{bmatrix}$$
(C.6)

Details on the derivation of these matrices are given in section 3.2.1

APPENDIX D Sensor dynamics model

The Quanser MkII is equipped with 3 different devices to measure its motion. An "Inertial Measurement Unit" (IMU) is used to measure the airship's attitude, angular rates and linear accelerations. A "Global Positioning System Receiver" GPS measures the airship velocity and position. A sonar installed below the airship measures the distance to the nearest surface. In level flight this corresponds to the height of the gondola above the ground.

All of these measurements are prone to noise, which needs to be represented in the simulation for a realistic evaluation of the controller performance.

The sonar height measurement is not used in the low-level control algorithms used to stabilize the airship attitude and velocity. Hence, no noise model has been developed. The onboard signal is filtered using a first order low-pass filter with a time constant of 0.18s. This filter is also present in the simulation results presented here.

D.1 IMU noise modelling

The noise characteristics of the IMU have been determined by recording 30 minutes of IMU data at a sampling rate of 64Hz with the IMU stationary on a horizontal flat surface. Under these conditions, perfect IMU readings would provide angular rates of zero around all three axes (neglecting the rotation of the earth), and constant Euler angles with roll and pitch close to zero. The measured accelerations

would be 1g along the vertical axis and zero along the other two axes. Deviations from these ideal measurements represent noise and possibly bias terms.

The accelerations measured by the IMU are actually not used by the onboard control algorithms. Hence, we will limit the IMU noise modelling to the Euler angles ϕ , θ , and ψ as well as the angular rates p, q, r.

To verify whether bias terms need to be considered in the sensor noise model, the average value over the entire timespan was calculated for each of the six parameters. The averages are shown in Table D–1. The average values for the angular rates were very close to zero, indicating that the gyro bias is sufficiently small to be neglected. The average roll and pitch values were less than 1 degree; their deviation from zero may be due to misalignment of the flat surface the IMU was lying on. The heading value was clearly different from zero, because the IMU was not aligned with magnetic north during the experiment.

Parameter	Average Value		
ϕ	-0.778 deg.		
θ	0.052 deg.		
ψ	-102.9 deg.		
p	$-2.1 \cdot 10^{-5} \text{ rad/s}$		
q	$6.8 \cdot 10^{-6} \text{ rad/s}$		
r	$-2.1 \cdot 10^{-5} \text{ rad/s}$		

Table D–1: Average values during stationary IMU test.

For further processing of the noise data, the samples have been corrected by the average value, giving 6 sets of samples with an average of zero each. This allows computation of the 3×3 covariance matrices Σ of the noise on the Euler angle
measurement and the angular rates measurement. The covariance matrix for the attitude measurement is

$$\Sigma_{Euler} = \begin{bmatrix} 0.0220 & 0.00207 & 1.80 \cdot 10^{-5} \\ 0.00207 & 0.0315 & 0.00917 \\ 1.80 \cdot 10^{-5} & 0.00917 & 0.0590 \end{bmatrix}$$
(D.1)

The corresponding matrix for the angular rates is

$$\Sigma_{\omega} = 10^{-5} \begin{bmatrix} .787 & -2.664 & 4.295 \\ -2.664 & 8.2171 & -3.252 \\ 4.295 & -3.252 & 11.157 \end{bmatrix}$$
(D.2)

The cross-terms for the angular rate covariance matrix are significant and hence will have be considered in the noise generation process. This is a surprising result, as one would expect the three rate gyros to provide measurement information independently. The cross-terms for the Euler angle measurement are one order of magnitude less than the diagonal values and could be neglected, if desired. However, the noise generation proposed below will take them into account also.

The second important characteristic of the noise is its frequency spectrum. The IMU internally combines the data from different sensors to compute the Euler angles which may lead to non-white noise due to filters used in the unknown IMU internal algorithms. To determine the frequency characteristics of the noise, a discrete Fourier analysis was performed on the zero-mean sample sets using the function fft from MatLab[83].



Figure D–1: Frequency spectrum of the Figure D–2: Frequency spectrum of the roll rate p measurement noise. roll angle ϕ measurement noise.

The results of the Fourier transformation are very similar for each of the three Euler Angles as well as each of the angular rates. The noise of the angular rates measurement appears to have the same intensity at all frequencies, as shown for the roll rate noise in Figure D–1. Hence, the angular rates measurement noise can be modelled as white noise using the covariance matrix calculated earlier.

The frequency spectrum of the roll angle noise is shown in Figure D–2 as an example of the characteristics of the Euler angle measurement noise. Interestingly, the noise on the Euler angle measurements drops at 20dB per decade at high frequencies, requiring a more complex noise generation setup. SimuLink provides only white noise generators. Generating white noise with the appropriate covariance matrix and passing it through a low-pass filter will give noise with a frequency spectrum similar to the one shown in Figure D–2. The cut-off frequency of the low-pass filter was determined graphically from the frequency spectrum plots for each Euler angle.

The intensity values in Figure D–2 have been scaled using the roll angle variance, so that the intensity of very low frequencies is expected to be at 0dB. The cut-off frequency can then be determined by finding the intersection of the straight line representing the upper limit of the noise spectrum with the 0dB level. The process for pitch and yaw is analogous. This gives for roll and pitch a cut-off frequency of 0.01585Hz and 0.01259Hz respectively, whereas for yaw it is approximately 0.00631Hz.

The IMU uses the rate gyros in combination with the accelerometers to calculate the roll and pitch angle, whereas it uses the rate gyros in combination with the magnetometers to calculate the yaw angle. In that respect, it appears sensible that the cut-off frequency for roll and pitch should be the same, whereas the cut-off frequency for yaw is different. Based on this, the time constants for the low-pass filters to represent these cut-off frequencies are $\tau_{\phi} = \tau_{\theta} = 11.3$ s and $\tau_{\psi} = 25.2$ s.

The application of the low-pass filter significantly reduces the overall intensity of the noise despite the use of the appropriate covariance matrix in the white noise generator. Hence, the noise has been scaled up after the filter to fit the intensity of the measured noise. This factor has been determined by comparing the variance of the simulated noise with the variance of the measured noise and changing the scale factor until both are identical. This procedure yields the gains $G_{\phi} = G_{\theta} = 37$ and $G_{\psi} = 53$.

Figures D–3 and D–4 show a comparison of the measured zero-mean noise and the noise modelled as described above for the measurement of pitch θ and yaw ψ .





Figure D–3: Modeled and measured zeromean noise on the pitch angle measurement over a period of 600 seconds.

Figure D–4: Modeled and measured zeromean noise on the yaw angle measurement over a period of 600 seconds.

Both figures indicate that the modelled noise is statistically representative of the actual noise.

D.2 GPS noise modelling

The GPS information required for the low-level controller is the velocity measurement. The GPS noise modelling will hence be limited to this quantity. The GPS can be operated in two different configurations either as standalone GPS or in a differential GPS setup. In standalone configuration, the GPS processes the satellite data it receives via its dedicated antenna and directly computes velocity and position.

In the differential configuration, the onboard GPS transmits the raw-data it receives via its dedicated antenna to a ground station. This data transmission can for example be a wifi 802.11g connection. Besides the data from the onboard GPS, the ground station also receives raw data from a second GPS, called the base station, that is directly connected to the station. The ground station compares the data received from both receivers which allows to compute a very precise position of the onboard unit with respect to the base station. The precision of the position information is in the range of centimeters in this case.

The performance of the velocity measurement is shown in Figures D–5 for the standalone GPS configuration and D–6 for the differential configuration. The GPS antenna was at rest during the time of the experiment, hence the actual velocity is zero in all three directions. The noise level in differential configuration is obviously less than in standalone configuration. However, the noise level even in standalone configuration is still very low with all peaks being less than 0.1m/s.



Figure D–5: Stationary velocity measure- Figure D–6: Stationary velocity measurement in standalone GPS configuration. ment in differential GPS configuration.

For the standalone configuration, the variance of the horizontal velocities is $2.69 \cdot 10^{-4} \text{m}^2/\text{s}^2$ in the North direction and $2.18 \cdot 10^{-4} \text{m}^2/\text{s}^2$ in the East direction. The variance in the vertical direction is about 4 times larger at $1.06 \cdot 10^{-3} \text{m}^2/\text{s}^2$. The covariances are one order of magnitude less than the variances and hence considered

negligible. The specifications of the GPS receiver indicate a standard deviation for the velocity measurement of 0.03m/s. This corresponds to a variance of $9 \cdot 10^{-4}$ m²/s² and hence represents a noise level somewhat larger than the level actually observed.

In this work, the GPS noise had been modelled using the specification values, as (a) the stationary measurements were available too late to incorporate the actual noise characteristics in the simulation, (b) the specification values may be more representative in situations with less satellite coverage, and (c) the specification values lead to a more conservative evaluation of the controller performance.

The raw GPS velocity measurement is passed through a low-pass filter before being fed to the controller in order to reduce noise-induced jitter of the controller output. Different time constants in the range of 0.5s to 1s have been tried to verify the performance with the different constants.

APPENDIX E Jacobian matrices of the linearized equations of motion

The Jacobian matrices in (3.6) have the structure

				•					(1) (1)
$\cos \phi_e \tan \theta_e$	$-\sin\phi_e$	$\frac{\cos \phi_e}{\cos \theta_e}$	$\left. \frac{\partial f_x}{\partial r} \right _e$	$\left. \frac{\partial f_y}{\partial r} \right _e$	$\left. \frac{\partial f_z}{\partial r} \right _e$	$\left. \frac{\partial n_x}{\partial r} \right _e$	$\frac{\partial n_y}{\partial r}\Big _e$	$\frac{\partial n_z}{\partial r}\Big _e$	Ū
$\sin\phi_e \tan\theta_e$	$\cos \phi_e$	$\frac{\sin \phi_e}{\cos \theta_e}$	$\left. \frac{\partial f_x}{\partial q} \right _e$	$\left. \frac{\partial f_y}{\partial q} \right _e$	$\left. \frac{\partial f_z}{\partial q} \right _e$	$\left. \frac{\partial n_x}{\partial q} \right _e$	$\left. \frac{\partial n_y}{\partial q} \right _e$	$\frac{\partial n_z}{\partial q}\Big _e$	
1	0	0	$\frac{\partial f_x}{\partial p}\Big _e$	$\left. \frac{\partial f_y}{\partial p} \right _e$	$\left. \frac{\partial f_z}{\partial p} \right _e$	$\left. \frac{\partial n_x}{\partial p} \right _e$	$\left. \frac{\partial n_y}{\partial p} \right _e$	$\frac{\partial n_z}{\partial p} \Big _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial w} \right _e$	$\left. \frac{\partial f_y}{\partial w} \right _e$	$\left. \frac{\partial f_z}{\partial w} \right _e$	$\left. \frac{\partial n_x}{\partial w} \right _e$	$\left. \frac{\partial n_y}{\partial w} \right _e$	$\left. \frac{\partial n_z}{\partial w} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial v} \right _e$	$\left. \frac{\partial f_y}{\partial v} \right _e$	$\left. \frac{\partial f_z}{\partial v} \right _e$	$\left. \frac{\partial n_x}{\partial v} \right _e$	$\frac{\partial n_y}{\partial v}\Big _e$	$\left. \frac{\partial n_z}{\partial v} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial u} \right _e$	$\left. \frac{\partial f_y}{\partial u} \right _e$	$\left. \frac{\partial f_z}{\partial u} \right _e$	$\left. \frac{\partial n_x}{\partial u} \right _e$	$\left. \frac{\partial n_y}{\partial u} \right _e$	$\left. \frac{\partial n_z}{\partial u} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial \psi} \right _e$	$\left. \frac{\partial f_y}{\partial \psi} \right _e$	$\left. \frac{\partial f_z}{\partial \psi} \right _e$	$\frac{\partial n_x}{\partial \psi} \Big _e$	$\left. \frac{\partial n_y}{\partial \psi} \right _e$	$\frac{\partial n_z}{\partial \psi} \Big _e$	
$\frac{\sin\phi_e}{\cos^2\theta_e}q_e + \frac{\cos\phi_e}{\cos^2\theta_e}r_e$	0	$\frac{\tan \theta_e}{\cos \theta_e} (\sin \phi_e q_e + \cos \phi_e r_e)$	$\left. \frac{\partial f_x}{\partial \theta} \right _e$	$\left. \frac{\partial f_y}{\partial \theta} \right _{e}$	$\left. \frac{\partial f_z}{\partial \theta} \right _e$	$\left. rac{\partial n_x}{\partial heta} ight _e$	$\left. \frac{\partial n_y}{\partial \theta} \right _e$	$\left. \frac{\partial n_{z}}{\partial \theta} \right _{e}$	
Γ tan $ heta_e(\cos\phi_e q_e - \sin\phi_e r_e)$	$-\sin\phi_e q_e -\cos\phi_e r_e$	$\frac{\cos\phi_e}{\cos\theta_e}q_e - \frac{\sin\phi_e}{\cos\theta_e}r_e$	$\left. rac{\partial f_x}{\partial \phi} ight _e$	$\left. rac{\partial f_y}{\partial \phi} \right _e$	$\left. \frac{\partial f_z}{\partial \phi} \right _e$	$\left. \frac{\partial n_x}{\partial \phi} \right _e$	$\left. rac{\partial n_y}{\partial \phi} ight _e$	$\frac{\partial n_z}{\partial \phi} \Big _e$	
				$\bar{\boldsymbol{J}}_x =$					

				(E.2)					
0	0	0	$\left. \frac{\partial f_x}{\partial \dot{w}_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial \dot{w}_{w,I}} \right _e$	$\left. \frac{\partial f_z}{\partial \dot{w}_{w,I}} \right _e$	$\left. \frac{\partial n_x}{\partial \dot{w}_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial \dot{w}_{w,I}} \right _e$	$\left. \frac{\partial n_z}{\partial \dot{w}_{w,I}} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial \dot{v}_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial \dot{v}_{w,I}} \right _e$	$\left. \frac{\partial f_z}{\partial \dot{v}_{w,I}} \right _e$	$\left. \frac{\partial n_x}{\partial \dot{v}_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial \dot{v}_{w,I}} \right _e$	$\left. \frac{\partial n_z}{\partial \dot{v}_{w,I}} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial \dot{u}_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial \dot{u}_{w,I}} \right _e$	$\left. \frac{\partial f_z}{\partial \dot{u}_{w,I}} \right _e$	$\left. \frac{\partial n_x}{\partial \dot{u}_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial \dot{u}_{w,I}} \right _e$	$\left. \frac{\partial n_z}{\partial \dot{u}_{w,I}} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial w_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial w_{w,I}} \right _e$	$\left. \frac{\partial f_z}{\partial w_{w,I}} \right _e$	$\left. \frac{\partial n_x}{\partial w_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial w_{w,I}} \right _e$	$\left. \frac{\partial n_z}{\partial w_{w,I}} \right _e$	
0	0	0	$\left. \frac{\partial f_x}{\partial v_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial v_{w,I}} \right _e$	$\left. \frac{\partial f_z}{\partial v_{w,I}} \right _e$	$\left. \frac{\partial n_x}{\partial v_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial v_{w,I}} \right _e$	$\left. \frac{\partial n_z}{\partial v_{w,I}} \right _e$	•
0	0	0	$\left \frac{\partial f_x}{\partial u_{w,I}} \right _e$	$\left. \frac{\partial f_y}{\partial u_{w,I}} \right _e$	$\left \frac{\partial f_z}{\partial u_{w,I}} \right _e$	$\left \frac{\partial n_x}{\partial u_{w,I}} \right _e$	$\left. \frac{\partial n_y}{\partial u_{w,I}} \right _e$	$\left \frac{\partial n_z}{\partial u_{w,I}} \right _e$	
				$\bar{\bm{J}}_w =$					
				and					
				<u>`</u>					,
. 0	0	0	0	0	0	0	0		
0 (0	0 (0 (0	0 () 1	0 (
) () C) C) C) C	1	C) C) (
0	0	0	0	1	0	0	0	0	
	0	0		0	0	0	0	Ο,	
				$ar{J}_u =$					i

The sum of the forces and moments as given on the right hand side of (3.3) can be written in component rather than vector form to simplify the computation of the partial derivatives. The component form of the

forces is

$$f_{x} = -m_{a3}wq + m_{a2}vr + m((q^{2} + r^{2})c_{x} - pqc_{y} - rpc_{z}) - (m - \rho V)g\sin\theta + (m_{Da3} - m_{Da1})qw_{w} + \dots$$

$$\dots + (m_{Da1} - m_{Da2})rw_{w} + m_{Da1}\dot{w}_{w}^{*} + f_{v,x} + f_{p,x}$$

$$f_{y} = -m_{a1}wr + m_{a3}wp + m((p^{2} + r^{2})c_{y} - pqc_{x} - rqc_{z}) + (m - \rho V)g\cos\theta\sin\phi + (m_{Da1} - m_{Da2})rw_{w} + \dots$$

$$\dots + (m_{Da2} - m_{Da3})pw_{w} + m_{Da2}\dot{w}_{w}^{*} + f_{v,y}$$

$$f_{z} = -m_{a2}vp + m_{a1}wq + m((p^{2} + q^{2})c_{z} - prc_{x} - rqc_{y}) + (m - \rho V)g\cos\theta\cos\phi + (m_{Da2} - m_{Da3})pw_{w} + \dots$$

$$\dots + (m_{Da3} - m_{Da1})qw_{w} + m_{Da3}\dot{w}_{w}^{*} + f_{v,z} + f_{p,z}$$

The component form of the moments is

(E.3)
The component form of the moments is

$$n_x = qr(J_{a2} - J_{a3}) + m((ru - pw - g\cos\theta\sin\phi)c_z - (pv - qu - g\cos\theta\cos\phi)c_y) - vw(m_{Da3} - m_{Da2}) + \dots + (m_{Da3} - m_{Da2})(wv_w + vw_w - v_ww_w) + n_{v,x} + n_{p,x}$$

$$n_y = pr(J_{a3} - J_{a1}) + m((pv - qu - g\cos\theta\cos\phi)c_x - (qw - rv + g\sin\theta)c_z) - uw(m_{Da1} - m_{Da3}) + \dots + (m_{Da1} - m_{Da3})(ww_w + ww_w) + n_{v,y} + n_{p,y}$$

$$n_z = pq(J_{a1} - J_{a2}) + m((qw - rv + g\sin\theta)c_y - (ru - pw - g\cos\theta\sin\phi)c_x) - wv(m_{Da2} - m_{Da1}) + \dots + (m_{Da2} - m_{Da1})(vu_w + ww_w - u_wv_w) + n_{v,z} + n_{p,z}$$

The viscous force equations (B.10) and (B.11) can be reorganized to contain only parameters and state variables using the relationships

$$\sin^2 \alpha = \frac{\tan^2 \alpha}{1 + \tan^2 \alpha} \quad \text{and} \quad \cos^2 \alpha = \frac{1}{1 + \tan^2 \alpha}.$$
 (E.5)

Using (E.5) on (B.6) and (B.5) yields

$$\cos^{2} \alpha = \frac{u_{ac}^{2}}{u_{ac}^{2} + v_{ac}^{2} + w_{ac}^{2}} = \frac{1}{2} \rho \frac{u_{ac}^{2}}{q_{0}}$$
(E.6)
$$\sin^{2} \alpha = \frac{v_{ac}^{2} + w_{ac}^{2}}{u_{ac}^{2} + w_{ac}^{2}} = \frac{1}{2} \rho \frac{v_{ac}^{2} + w_{ac}^{2}}{q_{0}}$$
(E.7)

These relationships allow to rewrite (B.10) and (B.11) as

$$\begin{split} \boldsymbol{f}_{v} &= \begin{bmatrix} -\frac{1}{2}\rho C_{A}A|u_{ac}|u_{ac}\\ -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}v_{ac}\sqrt{v_{ac}^{2}+w_{ac}^{2}}\\ &-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}w_{ac}\sqrt{v_{ac}^{2}+w_{ac}^{2}}\\ &-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}w_{ac}w_{ac}\sqrt{v_{ac}^{2}+w_{ac}^{2}}\\ &0\\ &0\\ &-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}w_{ac}\sqrt{v_{ac}^{2}+w_{ac}^{2}}\\ &-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}w_{ac}\sqrt{v_{ac}^{2}+w_{ac}^{2}}\\ \end{split}$$

(E.8)

(E.9)

The final step to achieve a direct representation of the states is to apply (B.4) to these relationships

-

giving

$$\mathbf{f}_{v} = \begin{bmatrix}
-\frac{1}{2}\rho C_{A}A|(u-u_{w})|(u-u_{w}) \\
-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}(v-v_{w}+rx_{ac})\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}} \\
-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}(w-w_{w}-qx_{ac})\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}} \\
\mathbf{n}_{v} = \begin{bmatrix}
-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}(w-w_{w}-qx_{ac})\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}} \\
-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}(v-v_{w}+rx_{ac})\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}} \\
-\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}(v-v_{w}+rx_{ac})\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}} \\
\end{bmatrix},$$
(E.10)

These two equations in combination with (E.3) and (E.4) allow for easy differentiation of the force terms with respect to the vehicle states and hence facilitate the computation of the individual entries of the Jacobian matrices. The results of these computations are given below.

E.1 Entries of the state Jacobian matrix

The derivatives of the force in x-direction with respect to the states are:

$$\begin{array}{lcl} \left. \frac{\partial f_x}{\partial u} \right|_e &=& -\rho C_A A |u_e - u_{w,e}| \\ \left. \frac{\partial f_x}{\partial v} \right|_e &=& m_{a2} r_e \\ \left. \frac{\partial f_x}{\partial w} \right|_e &=& -m_{a3} q_e \end{array}$$

$$\begin{aligned} \frac{\partial f_x}{\partial p}\Big|_e &= -m(q_e c_y + r_e c_z)\\ \frac{\partial f_x}{\partial q}\Big|_e &= -m_{a3} w_e + m(2q_e c_x - p_e c_y) + (m_{Da3} - m_{Da1}) w_{u,e}\\ \frac{\partial f_x}{\partial r}\Big|_e &= -m_{a3} w_e + m(2r_e c_x - p_e c_y) + (m_{Da1} - m_{Da2}) v_{w,e}\\ \frac{\partial f_x}{\partial r}\Big|_e &= (m_{Da3} - m_{Da1}) q_e \frac{\partial w_w}{\partial \phi}\Big|_e + (m_{Da1} - m_{Da2}) r_e \frac{\partial v_w}{\partial \phi}\Big|_e + \rho C_A A |u_e - u_{w,e}| \frac{\partial u_w}{\partial \phi}\Big|_e\\ \frac{\partial f_x}{\partial \theta}\Big|_e &= (m_{Da3} - m_{Da1}) q_e \frac{\partial w_w}{\partial \theta}\Big|_e + (m_{Da1} - m_{Da2}) r_e \frac{\partial v_w}{\partial \theta}\Big|_e + \rho C_A A |u_e - u_{w,e}| \frac{\partial u_w}{\partial \phi}\Big|_e\\ \frac{\partial f_x}{\partial \theta}\Big|_e &= (m_{Da3} - m_{Da1}) q_e \frac{\partial w_w}{\partial \theta}\Big|_e + (m_{Da1} - m_{Da2}) r_e \frac{\partial v_w}{\partial \theta}\Big|_e + \rho C_A A |u_e - u_{w,e}| \frac{\partial u_w}{\partial \theta}\Big|_e\end{aligned}$$

The derivatives of the force in y-direction with respect to the states are:

$$\begin{aligned} \frac{\partial f_{y}}{\partial u} \bigg|_{e} &= -m_{a1}r_{e} \\ \frac{\partial f_{y}}{\partial v} \bigg|_{e} &= -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}\frac{2(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} \\ \frac{\partial f_{y}}{\partial w} \bigg|_{e} &= m_{a3}p_{e} - \frac{1}{2}\rho\eta C_{D_{n}}A_{p}\frac{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} \\ \frac{\partial f_{y}}{\partial p} \bigg|_{e} &= m_{a3}w_{e} + m(2p_{e}c_{y}-q_{e}c_{x}) + (m_{Da2}-m_{Da3})w_{w}e} \\ \frac{\partial f_{y}}{\partial q} \bigg|_{e} &= -m(p_{e}c_{x}+r_{e}c_{z}) + \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}\frac{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} \\ \frac{\partial f_{y}}{\partial r} \bigg|_{e} &= -m_{a1}u_{e} + m(2r_{e}c_{y}-q_{e}c_{z}) + (m_{Da1}-m_{Da2})u_{w,e} - \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}\frac{2(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} \end{aligned}$$

$$\frac{\partial f_y}{\partial \phi}\Big|_e = (m - \rho V)g\cos\theta\cos\phi + (m_{Da2} - m_{Da3})p_e \frac{\partial w_w}{\partial \phi}\Big|_e + (m_{Da1} - m_{Da2})r_e \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\partial \phi}\Big|_e = (m - \rho V)g\cos\theta\cos\phi + (m_{Da2} - m_{Da3})p_e \frac{\partial w_w}{\partial \phi}\Big|_e + (m_{Da1} - m_{Da2})r_e \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \phi}\Big|_e + (m_{Da1} - m_{Da2})r_e \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \phi}\Big|_e + (m_{Da1} - m_{Da2})r_e \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \phi}\Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}{\partial \phi}\Big|_e \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})^2}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \phi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \psi}\Big|_e + \cdots$$

$$\cdots + \frac{1}{2}\rho\eta C_{D_n}A_p\left[\frac{2(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}{\partial \psi}\Big|_e + \cdots$$

$$\frac{\partial f_y}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial \psi}\Big|_e + \cdots$$

The derivatives of the force in z-direction with respect to the states are:

$$\begin{aligned} \frac{\partial f_{z}}{\partial u} \bigg|_{e} &= m_{a1}qe \\ \frac{\partial f_{z}}{\partial v} \bigg|_{e} &= -m_{a2}pe - \frac{1}{2}\rho\eta C_{D_{n}}A_{p} \frac{(v - v_{w} + rx_{ac})(w - w_{w} - qx_{ac})}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial f_{z}}{\partial w} \bigg|_{e} &= -\frac{1}{2}\rho\eta C_{D_{n}}A_{p} \frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} + rx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial f_{z}}{\partial p} \bigg|_{e} &= -m_{a2}v_{e} + m(2p_{e}c_{z} - r_{e}c_{x}) + (m_{Da3} - m_{Da1})u_{w,e} + \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}\frac{2(w - w_{w} - qx_{ac})^{2} + (v - w_{w} - qx_{ac})^{2}}{\sqrt{(v - w_{w} + rx_{ac})^{2}}} \end{aligned}$$

$$\begin{split} \frac{\partial f_{z}}{\partial r}\Big|_{e} &= -m(p_{e}c_{x} + q_{e}c_{y}) - \frac{1}{2}\rho\eta CD_{n}Apx_{ex}\frac{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \phi}\Big|_{e} &= -(m - \rho V)g\cos\theta\sin\phi + (m_{Da2} - m_{Da3})p_{e}\frac{\partial v_{w}}{\partial \phi}\Big|_{e} + (m_{Da3} - m_{Da1})q_{e}\frac{\partial u_{w}}{\partial \phi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \phi}\frac{\partial w_{w}}{e}\Big|_{e} + (m_{Da3} - m_{Da1})q_{e}\frac{\partial u_{w}}{\partial \phi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \theta}\frac{\partial w_{w}}{e}\Big|_{e} + (m_{Da3} - m_{Da1})q_{e}\frac{\partial u_{w}}{\partial \theta}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \theta}\frac{\partial w_{w}}{e}\Big|_{e} + (m_{Da3} - m_{Da1})q_{e}\frac{\partial u_{w}}{\partial \theta}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\frac{\partial w_{w}}{e}\Big|_{e} + (m_{Da3} - m_{Da1})q_{e}\frac{\partial u_{w}}{\partial \theta}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\frac{\partial w_{w}}{e}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \cdots \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + (w - w_{w} - w_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \frac{(w - w_{w} - w_{ac})(w - w_{w} - qx_{ac})}{\partial \psi}\Big|_{e} \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \frac{(w - w_{w} - w_{w} - qx_{ac})}{\partial \psi}\Big|_{e} + \frac{(w - w_{w} - w_{w} - qx_{ac})}{\partial \psi}\Big|_{e} \\ \cdots + \frac{1}{2}\rho\eta CD_{n}Ap\left[\frac{2(w - w_{w} - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_{e} + \frac{(w - w_{w} - w_{w} - qx_{ac})^{2}}{\partial \psi}\Big|_$$

The derivatives of the moment in x-direction with respect to the states are:

$$\frac{\partial n_x}{\partial u}\Big|_e = m(r_ec_z + q_ec_y)$$
$$\frac{\partial n_x}{\partial v}\Big|_e = -mp_ec_y - (m_{Da3} - m_{Da2})(w_e - w_{w,e})$$
$$\frac{\partial n_x}{\partial w}\Big|_e = -mp_ec_z - (m_{Da3} - m_{Da2})(v_e - v_{w,e})$$
$$\frac{\partial n_x}{\partial p}\Big|_e = -m(w_ec_z + v_ec_y)$$

$$\begin{aligned} \frac{\partial n_x}{\partial q}\Big|_e &= (J_{a2} - J_{a3})r_e + mu_e c_y\\ \frac{\partial n_x}{\partial r}\Big|_e &= (J_{a2} - J_{a3})q_e + mu_e c_z\\ \frac{\partial n_x}{\partial r}\Big|_e &= (J_{a2} - J_{a3})q_e + mu_e c_z\\ \frac{\partial n_x}{\partial \phi}\Big|_e &= -mg\cos\theta_e(c_z\cos\phi_e + c_y\sin\phi_e) + (m_{Da3} - m_{Da2})\left[(w_e - w_{w,e})\frac{\partial v_w}{\partial \phi}\Big|_e + (v_e - v_{w,e})\frac{\partial w_w}{\partial \phi}\Big|_e\\ \frac{\partial n_x}{\partial \theta}\Big|_e &= mg\sin\theta_e(c_z\sin\phi_e - c_y\cos\phi_e) + (m_{Da3} - m_{Da2})\left[(w_e - w_{w,e})\frac{\partial v_w}{\partial \theta}\Big|_e + (v_e - v_{w,e})\frac{\partial w_w}{\partial \theta}\Big|_e \right]\end{aligned}$$

The derivatives of the moment in y-direction with respect to the states are:

$$\begin{aligned} \frac{\partial n_{y}}{\partial u}\Big|_{e} &= -mq_{e}c_{x} - (m_{Da1} - m_{Da3})(w_{e} - w_{u,e}) \\ \frac{\partial n_{y}}{\partial v}\Big|_{e} &= m(p_{e}c_{x} + r_{e}c_{z}) + \frac{1}{2}\rho mC_{D_{n}}A_{p}x_{ac}\frac{(v - v_{w} + rx_{ac})(w - w_{w} - qx_{ac})}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{y}}{\partial w}\Big|_{e} &= -mq_{e}c_{z} - (m_{Da1} - m_{Da3})(u_{e} - u_{w,e}) + \frac{1}{2}\rho mC_{D_{n}}A_{p}x_{ac}\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{y}}{\partial p}\Big|_{e} &= (J_{a3} - J_{a1})r_{e} + mv_{e}c_{x} \\ \frac{\partial n_{y}}{\partial p}\Big|_{e} &= -m(u_{e}c_{x} + w_{e}c_{z}) - \frac{1}{2}\rho mC_{D_{n}}A_{p}x_{ac}^{2}\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{y}}{\partial r}\Big|_{e} &= (J_{a3} - J_{a1})p_{e} + mv_{e}c_{z} + \frac{1}{2}\rho mC_{D_{n}}A_{p}x_{ac}^{2}\frac{2(w - w_{w} - qx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{y}}{\partial r}\Big|_{e} &= mgc_{x}\cos\theta_{e}\sin\phi_{e} + (m_{Da1} - m_{Da3})\left[(u_{e} - u_{w,e})\frac{\partial w_{w}}{\partial \phi}\Big|_{e} + (w_{e} - w_{w,e})\frac{\partial u_{w}}{\partial \phi}\Big|_{e}\right] + \dots \end{aligned}$$

$$\begin{array}{rcl} \cdots & -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac} \left[\frac{2(w-w_{w}-qx_{ac})^{2}+(v-w_{w}+rx_{ac})^{2}}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} & \frac{\partial w_{w}}{\partial \phi} \right|_{e} + \frac{(v-v_{w}+rx_{ac})(w-w_{w}-qx_{ac})}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} & \frac{\partial w_{w}}{\partial \phi} \right|_{e} \\ \frac{\partial h_{y}}{\partial \theta} \bigg|_{e} & = mg(c_{x}\sin\theta_{e}\cos\phi_{e}-c_{z}\cos\theta_{e}) + (m_{Da1}-m_{Da3}) \left[(u_{e}-u_{w,e}) & \frac{\partial w_{w}}{\partial \theta} \right]_{e} + (w_{e}-w_{w,e}) & \frac{\partial u_{w}}{\partial \theta} \bigg|_{e} + (w_{e}-w_{w,e}) & \frac{\partial u_{w}}{\partial \theta} \bigg|_{e} + \cdots \\ \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac} \left[\frac{2(w-w_{w}-qx_{ac})^{2}+(w-w_{w}+rx_{ac})^{2}}{\sqrt{(v-v_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} & \frac{\partial u_{w}}{\partial \theta} \bigg|_{e} + \frac{(v-w_{w}+rx_{ac})(w-w_{w}-qx_{ac})}{\sqrt{(v-w_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} & \frac{\partial v_{w}}{\partial \theta} \bigg|_{e} \\ \cdots & -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac} \left[\frac{2(w-w_{w}-qx_{ac})^{2}+(w-w_{w}+rx_{ac})^{2}}{\partial \psi} & \frac{\partial w_{w}}{\partial \psi} \bigg|_{e} + \cdots \\ \cdots & -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac} \left[\frac{2(w-w_{w}-qx_{ac})^{2}+(w-w_{w}+rx_{ac})^{2}}{\partial \psi} & \frac{\partial w_{w}}{\partial \psi} \bigg|_{e} + \cdots \\ \cdots & -\frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac} \left[\frac{2(w-w_{w}-qx_{ac})^{2}+(w-w_{w}+rx_{ac})^{2}}{\partial \psi} & \frac{\partial w_{w}}{\partial \psi} \bigg|_{e} + \frac{(v-w_{w}+rx_{ac})(w-w_{w}-qx_{ac})}{\sqrt{(v-w_{w}+rx_{ac})^{2}+(w-w_{w}-qx_{ac})^{2}}} & \frac{\partial w_{w}}{\partial \psi} \bigg|_{e} \right] \\ \end{array}$$

The derivatives of the moment in z-direction with respect to the states are:

$$\begin{aligned} \frac{\partial n_{2}}{\partial u}\Big|_{e} &= -mrec_{x} - (m_{Da2} - m_{Da1})(v_{e} - v_{w,e}) \\ \frac{\partial n_{z}}{\partial v}\Big|_{e} &= -mrec_{y} - (m_{Da2} - m_{Da1})(u_{e} - u_{w,e}) - \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{z}}{\partial v}\Big|_{e} &= m(q_{e}c_{y} + p_{e}c_{x}) - \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}\frac{(v - v_{w} + rx_{ac})(w - w_{w} - qx_{ac})}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{z}}{\partial p}\Big|_{e} &= (J_{a1} - J_{a2})q_{e} + mw_{e}c_{x} \\ \frac{\partial n_{z}}{\partial q}\Big|_{e} &= (J_{a1} - J_{a2})q_{e} + mw_{e}c_{x} + \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}^{2}\frac{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \\ \frac{\partial n_{z}}{\partial r}\Big|_{e} &= -m(u_{e}c_{x} + v_{e}c_{y}) - \frac{1}{2}\rho\eta C_{D_{n}}A_{p}x_{ac}^{2}\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \end{aligned}$$

$$\begin{array}{lcl} \frac{\partial h_{z}}{\partial \phi}\Big|_{e} &=& mgc_{x} \cos \theta_{e} \cos \phi_{e} + \left(m_{Da2} - m_{Da1}\right) \left[\left(u_{e} - u_{w,e}\right) \frac{\partial v_{w}}{\partial \phi}\Big|_{e} + \left(v_{e} - v_{w,e}\right) \frac{\partial u_{w}}{\partial \phi}\Big|_{e} \right] + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial w_{w}}{\partial \theta}\Big|_{e} + \left(v_{e} - v_{w,e}\right) \frac{\partial u_{w}}{\partial \theta}\Big|_{e} \right] + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \theta}\Big|_{e} + \left(v_{e} - v_{w,e}\right) \frac{\partial u_{w}}{\partial \theta}\Big|_{e} \right] + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \theta}\Big|_{e} + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \theta}\Big|_{e} + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \theta}\Big|_{e} + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \psi}\Big|_{e} + \cdots \\ &\cdots + \frac{1}{2}\rho\eta C_{Dn}Apx_{ac} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}{\sqrt{\left(v - v_{w} + rx_{ac}\right)^{2} + \left(w - w_{w} - qx_{ac}\right)^{2}}} \frac{\partial v_{w}}{\partial \psi}\Big|_{e} \right] \\ \end{array}$$

_

The derivatives of the wind speeds with respect to the attitude are given in section E.3.

E.2 Entries of the wind disturbance Jacobian matrix

The derivatives of the force in x-direction with respect to the inertial wind speed components are:

$$\begin{aligned} \frac{\partial f_x}{\partial u_{w,I}}\Big|_e &= (m_{Da3} - m_{Da1})q_e \left.\frac{\partial w_w}{\partial u_{w,I}}\Big|_e + (m_{Da1} - m_{Da2})r_e \left.\frac{\partial v_w}{\partial u_{w,I}}\Big|_e + \rho C_A A |u_e - u_{w,e}| \left.\frac{\partial u_w}{\partial u_{w,I}}\Big|_e \\ \frac{\partial f_x}{\partial v_{w,I}}\Big|_e &= (m_{Da3} - m_{Da1})q_e \left.\frac{\partial w_w}{\partial v_{w,I}}\Big|_e + (m_{Da1} - m_{Da2})r_e \left.\frac{\partial v_w}{\partial v_{w,I}}\Big|_e + \rho C_A A |u_e - u_{w,e}| \left.\frac{\partial u_w}{\partial v_{w,I}}\Big|_e \\ \frac{\partial f_x}{\partial w_{w,I}}\Big|_e &= (m_{Da3} - m_{Da1})q_e \left.\frac{\partial w_w}{\partial w_{w,I}}\Big|_e + (m_{Da1} - m_{Da2})r_e \left.\frac{\partial v_w}{\partial v_{w,I}}\Big|_e + \rho C_A A |u_e - u_{w,e}| \left.\frac{\partial u_w}{\partial v_{w,I}}\Big|_e \\ \frac{\partial f_x}{\partial w_{w,I}}\Big|_e &= (m_{Da3} - m_{Da1})q_e \left.\frac{\partial w_w}{\partial w_{w,I}}\Big|_e + (m_{Da1} - m_{Da2})r_e \left.\frac{\partial v_w}{\partial w_{w,I}}\Big|_e + \rho C_A A |u_e - u_{w,e}| \left.\frac{\partial u_w}{\partial w_{w,I}}\Big|_e \end{aligned}$$

$$\left. \frac{\partial f_x}{\partial \dot{v}_{w,I}} \right|_e = m_{Da1} \left. \frac{\partial \dot{u}_w^*}{\partial \dot{v}_{w,I}} \right|_e \\ \left. \frac{\partial f_x}{\partial \dot{w}_{w,I}} \right|_e = m_{Da1} \left. \frac{\partial \dot{u}_w^*}{\partial \dot{w}_{w,I}} \right|_e$$

The derivatives of the force in y-direction with respect to the inertial wind speed components are:

$$\begin{array}{lcl} \frac{\partial f_{y}}{\partial u_{wl}} \Big|_{e} &= & (m_{Dal} - m_{Da2}) r_{e} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + (m_{Da2} - m_{Da3}) p_{e} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \frac{(v - v_{w} + rx_{ac})(w - w_{w} - qx_{ac})}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial v_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial v_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial v_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial u_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial u_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial u_{wl} H} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} + \cdots \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\partial u_{wl} H} \Big|_{e} + \frac{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} \\ & \cdots + \frac{1}{2} \rho \eta C_{D_{n}} A_{P} \left[\frac{2(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \frac{\partial u_{w}}{\partial u_{wl}} \Big|_{e} \\ & \frac{\partial f_{w}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}}{\partial u_{wl}} \Big|_{e} + \frac{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}{\sqrt{(v - v_{w} + rx_{ac})^{2} + (w - w_{w} - qx_{ac})^{2}}} \frac{\partial u_{w}}{\partial u_{w}} \Big|_{e} \\ & \frac{\partial f_{w}}{\partial u_{w}} \Big|_{e} = m_{Da2} \frac{\partial \tilde{u$$

 $\dots + \frac{1}{2}\rho\eta C_{D_n}A_p\left[\frac{2(w-w_w-qx_{ac})^2 + (v-v_w+rx_{ac})^2}{\sqrt{(v-v_w+rx_{ac})^2 + (w-w_w-qx_{ac})^2}}\frac{\partial w_w}{\partial w_{y,I}}\right]_e + \frac{(v-v_w+rx_{ac})(w-w_w-qx_{ac})}{\sqrt{(v-v_w+rx_{ac})^2 + (w-w_w-qx_{ac})^2}}\frac{\partial v_w}{\partial w_{y,I}}\Big|_e\right]$ $\dots + \frac{1}{2}\rho\eta C_{D_n}A_p\left[\frac{2(w-w_w-qx_{ac})^2 + (v-v_w+rx_{ac})^2}{\sqrt{(v-v_w+rx_{ac})^2 + (w-w_w-qx_{ac})^2}} \left.\frac{\partial w_w}{\partial u_{w,I}}\right|_e + \frac{(v-v_w+rx_{ac})(w-w_w-qx_{ac})}{\sqrt{(v-v_w+rx_{ac})^2 + (w-w_w-qx_{ac})^2}} \left.\frac{\partial v_w}{\partial u_{w,I}}\right|_e\right]$ $\dots + \frac{1}{2}\rho\eta C_{D_n}A_p\left[\frac{2(w-ww-qx_{ac})^2 + (v-vw+rx_{ac})^2}{\sqrt{(v-vw+rx_{ac})^2 + (w-ww-qx_{ac})^2}} \left.\frac{\partial w_w}{\partial v_{w,I}}\right|_e + \frac{(v-vw+rx_{ac})(w-ww-qx_{ac})}{\sqrt{(v-vw+rx_{ac})^2 + (w-ww-qx_{ac})^2}} \left.\frac{\partial v_w}{\partial v_{w,I}}\right|_e\right]$ $= \left(m_{Da2} - m_{Da3}\right)p_e \left.\frac{\partial v_w}{\partial w_{w,I}}\right|_e + \left(m_{Da3} - m_{Da1}\right)q_e \left.\frac{\partial u_w}{\partial w_{w,I}}\right|_e + \dots$ $= \left. (m_{Da2} - m_{Da3}) p_e \left. \frac{\partial v_w}{\partial v_{w,I}} \right|_e + (m_{Da3} - m_{Da1}) q_e \left. \frac{\partial u_w}{\partial v_{w,I}} \right|_e + \dots$ $\frac{\partial f_z}{\partial u_{w,I}}\Big|_e = (m_{Da2} - m_{Da3})p_e \left.\frac{\partial v_w}{\partial u_{w,I}}\Big|_e + (m_{Da3} - m_{Da1})q_e \left.\frac{\partial u_w}{\partial u_{w,I}}\Big|_e + \dots$ $= m_{Da3} \frac{\partial \dot{w}_{w}^{*}}{\partial \dot{u}_{w,I}} \Big|_{e}$ $= m_{Da3} \frac{\partial \dot{w}_{w}}{\partial \dot{v}_{w,I}} \Big|_{e}$ $= m_{Da3} \frac{\partial \dot{w}_{w}^{*}}{\partial \dot{w}_{w,I}} \Big|_{e}$ $\left. \frac{\partial f_z}{\partial w_{w,I}} \right|_e$ $\left. \frac{\partial f_z}{\partial v_{w,I}} \right|_e$ $\left. \frac{\partial f_z}{\partial \dot{u}_{w,I}} \right|_e \\ \frac{\partial f_z}{\partial \dot{v}_{w,I}} \left|_e \\ \frac{\partial f_z}{\partial \dot{w}_{w,I}} \right|_e$

The derivatives of the force in z-direction with respect to the inertial wind speed components are:

The derivatives of the moment in x-direction with respect to the inertial wind speed components are:

$$\begin{array}{lcl} \displaystyle \frac{\partial m_{w,I}}{\partial u_{w,I}} \bigg|_{e} &= & \left(m_{Da3} - m_{Da2} \right) \left[\left(w_{e} - w_{w,e} \right) \left. \frac{\partial v_{w}}{\partial u_{w,I}} \right|_{e} + \left(v_{e} - v_{w,e} \right) \left. \frac{\partial w_{w}}{\partial u_{w,I}} \right|_{e} \right] \\ \displaystyle \frac{\partial m_{w}}{\partial v_{w,I}} \bigg|_{e} &= & \left(m_{Da3} - m_{Da2} \right) \left[\left(w_{e} - w_{w,e} \right) \left. \frac{\partial v_{w}}{\partial v_{w,I}} \right|_{e} + \left(v_{e} - v_{w,e} \right) \left. \frac{\partial w_{w}}{\partial v_{w,I}} \right|_{e} \right] \\ \displaystyle \frac{\partial m_{w,I}}{\partial v_{w,I}} \bigg|_{e} &= & \left(m_{Da3} - m_{Da2} \right) \left[\left(w_{e} - w_{w,e} \right) \left. \frac{\partial v_{w}}{\partial v_{w,I}} \right|_{e} + \left(v_{e} - v_{w,e} \right) \left. \frac{\partial w_{w}}{\partial v_{w,I}} \right|_{e} \right] \\ \displaystyle \frac{\partial m_{w,I}}{\partial v_{w,I}} \bigg|_{e} &= & 0 \\ \displaystyle \frac{\partial m_{w,I}}{\partial v_{w,I}} \bigg|_{e} &= & 0 \\ \displaystyle \frac{\partial m_{w}}{\partial v_{w,I}} \bigg|_{e} &= & 0 \\ \displaystyle \frac{\partial m_{w}}{\partial v_{w,I}} \bigg|_{e} &= & 0 \\ \end{array}$$

The derivatives of the moment in y-direction with respect to the inertial wind speed components are:

$$\frac{\partial n_y}{\partial u_{w,I}}\Big|_e = (m_{Da1} - m_{Da3}) \left[(u_e - u_{w,e}) \frac{\partial w_w}{\partial u_{w,I}} \Big|_e + (w_e - w_{w,e}) \frac{\partial u_w}{\partial u_{w,I}} \Big|_e \right] - \cdots \\ \cdots - \frac{1}{2} \rho \eta C_{D_n} A_p x_{ac} \left[\frac{2(w - w_w - qx_{ac})^2 + (w - w_w - qx_{ac})^2}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial u_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - w_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial u_{w,I}} \Big|_e \\ \cdots - \frac{1}{2} \rho \eta C_{D_n} A_p x_{ac} \left[\frac{2(w - w_w - qx_{ac})^2 + (w - w_w - qx_{ac})^2}{\partial v_{w,I}} \frac{\partial u_w}{|_e} \right] - \cdots \\ \cdots - \frac{1}{2} \rho \eta C_{D_n} A_p x_{ac} \left[\frac{2(w - w_w - qx_{ac})^2 + (w - w_w - qx_{ac})^2}{\partial v_{w,I}} \frac{\partial u_w}{|_e} \right] - \cdots \\ \cdots - \frac{1}{2} \rho \eta C_{D_n} A_p x_{ac} \left[\frac{2(w - w_w - qx_{ac})^2 + (w - w_w - qx_{ac})^2}{\partial v_{w,I}} \frac{\partial w_w}{|_e} \right] + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \frac{\partial w_w}{\partial v_{w,I}} \Big|_e + \frac{(v - v_w + rx_{ac})^2 + (w - w_w - qx$$

$$\begin{split} \frac{\partial h_{u}}{\partial u_{u,l}} &= (m_{Dal} - m_{Dal}) \left[(u_e - u_{uc}) \frac{\partial u_{u}}{\partial u_{u,l}} \right|_e + (u_e - u_{uc}) \frac{\partial u_{u}}{\partial u_{u,l}} \right|_e \\ & \cdots - \frac{1}{2} \rho \eta C D_n A_p x_e \left[\frac{2(u - u_u - qx_e)^2 + (v - u_v - rx_a)^2}{\sqrt{(v - u_v + rx_a)^2}} \frac{\partial u_{u}}{\partial u_{u,l}} \right|_e + \frac{(v - u_v + rx_a)^2 + (u - u_u - qx_a)}{\sqrt{(v - u_v - rx_a)^2}} \frac{\partial u_{u}}{\partial u_{u,l}} \right|_e \\ & \frac{\partial h_u}{\partial u_{u,l}} \Big|_e &= 0 \\ \frac{\partial h_u}{\partial u_{u$$

 $\dots + \frac{1}{2}\rho\eta C_{D_n}A_p \left[\frac{(w - w_w - qx_{ac})^2 + 2(v - v_w + rx_{ac})^2}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \left. \frac{\partial v_w}{\partial w_{u,I}} \right|_e + \frac{(v - v_w + rx_{ac})(w - w_w - qx_{ac})}{\sqrt{(v - v_w + rx_{ac})^2 + (w - w_w - qx_{ac})^2}} \left. \frac{\partial w_w}{\partial w_{u,I}} \right|_e \right]$

$$\begin{array}{lll} \displaystyle \frac{\partial n_z}{\partial \dot{u}_{w,I}} \bigg|_e & = & 0 \\ \displaystyle \frac{\partial n_z}{\partial \dot{v}_{w,I}} \bigg|_e & = & 0 \\ \displaystyle \frac{\partial n_z}{\partial \dot{w}_{w,I}} \bigg|_e & = & 0 \end{array}$$

E.3 Subderivatives

This section gives the derivatives not explicitly given in the representation of the Jacobian matrix entries above.

The derivatives of the wind speeds with respect to the attitude angles are

$$\begin{aligned} \frac{\partial u_w}{\partial \phi} \bigg|_e &= 0\\ \frac{\partial v_w}{\partial \phi} \bigg|_e &= 0\\ \frac{\partial v_w}{\partial \phi} \bigg|_e &= (\cos \psi_e \sin \theta_e \cos \phi_e + \sin \psi_e \sin \phi_e) u_{w,I,e} + (\sin \psi_e \sin \theta_e \cos \phi_e - \cos \psi_e \sin \phi_e) v_{w,I,e} + \cos \theta_e \cos \phi w_{w,I,e}\\ \frac{\partial v_w}{\partial \phi} \bigg|_e &= (-\cos \psi_e \sin \theta_e \sin \phi_e + \sin \psi_e \cos \phi_e) u_{w,I,e} + (-\sin \psi_e \sin \theta_e \sin \phi_e - \cos \psi_e \cos \phi_e) v_{w,I,e} - \cos \theta_e \sin \phi w_{w,I,e}\\ \frac{\partial u_w}{\partial \theta} \bigg|_e &= -\sin \theta_e (\cos \psi_e u_{w,I,e} + \sin \psi_e \cos \theta_e u_{w,I,e}) - \cos \theta_e \sin \theta_e w_{w,I,e}\\ \frac{\partial u_w}{\partial \theta} \bigg|_e &= \cos \psi_e \cos \theta_e \sin \phi_e u_{w,I,e} + \sin \psi_e \cos \theta_e \sin \phi_e w_{w,I,e} - \sin \theta_e \sin \phi_e w_{w,I,e}\\ \frac{\partial u_w}{\partial \theta} \bigg|_e &= \cos \psi_e \cos \theta_e \sin \phi_e u_{w,I,e} + \sin \psi_e \cos \theta_e \sin \phi_e v_{w,I,e} - \sin \theta_e \sin \phi_e w_{w,I,e} \end{aligned}$$

$$\begin{aligned} \frac{\partial u_w}{\partial \psi} \bigg|_e &= -\sin\psi_e \cos\theta_e u_{w,I,e} + \cos\psi_e \cos\theta_e v_{w,I,e} \\ \frac{\partial v_w}{\partial \psi} \bigg|_e &= (-\sin\psi_e \sin\theta_e \sin\theta_e - \cos\psi_e \cos\phi_e) u_{w,I,e} + (\cos\psi_e \sin\theta_e \sin\phi_e - \sin\psi_e \cos\phi_e) v_{w,I,e} \\ \frac{\partial w_w}{\partial \psi} \bigg|_e &= (-\sin\psi_e \sin\theta_e \cos\phi_e + \cos\psi_e \sin\phi_e) u_{w,I,e} + (\cos\psi_e \sin\theta_e \cos\phi_e + \sin\psi_e \sin\phi_e) v_{w,I,e}. \end{aligned}$$

The derivatives of the inertial wind speed rate of change in body frame coordinates with respect to the rate of change in inertial coordiantes are identical to the derivatives of the wind speeds in body frame with respect to the wind speeds in the inertial frame.

$$\begin{aligned} \frac{\partial \dot{u}_{w,I}^{*}}{\partial \dot{u}_{w,I}} \bigg|_{e} &= \frac{\partial u_{w}}{\partial u_{w,I}} \bigg|_{e} &= \frac{\partial u_{w}}{\partial u_{w,I}} \bigg|_{e} &= \frac{\partial u_{w}}{\partial v_{w,I}} \bigg|_{e} &= \frac{\partial u_{w}}{\partial v_{w,I}} \bigg|_{e} &= \sin\psi_{e}\cos\theta_{e} \\ \frac{\partial \dot{u}_{w}^{*}}{\partial \dot{v}_{w,I}} \bigg|_{e} &= \frac{\partial u_{w}}{\partial w_{w,I}} \bigg|_{e} &= -\sin\theta_{e} \\ \frac{\partial \dot{v}_{w}^{*}}{\partial \dot{u}_{w,I}} \bigg|_{e} &= \frac{\partial v_{w}}{\partial u_{w,I}} \bigg|_{e} &= \cos\psi_{e}\sin\theta_{e}\sin\phi_{e} - \sin\psi_{e}\cos\phi_{e} \\ \frac{\partial \dot{v}_{w}^{*}}{\partial \dot{v}_{w,I}} \bigg|_{e} &= \frac{\partial v_{w}}{\partial v_{w,I}} \bigg|_{e} &= \cos\theta_{e}\sin\theta_{e}\sin\phi_{e} + \cos\psi_{e}\cos\phi_{e} \\ \frac{\partial \dot{v}_{w}^{*}}{\partial \dot{v}_{w,I}} \bigg|_{e} &= \frac{\partial v_{w}}{\partial w_{w,I}} \bigg|_{e} &= \cos\theta_{e}\sin\theta_{e}\sin\phi_{e} \\ \frac{\partial \dot{v}_{w}^{*}}{\partial \dot{u}_{w,I}} \bigg|_{e} &= \frac{\partial v_{w}}{\partial w_{w,I}} \bigg|_{e} &= \cos\theta_{e}\sin\theta_{e}\cos\phi_{e} \\ \frac{\partial \dot{v}_{w}^{*}}{\partial \dot{u}_{w,I}} \bigg|_{e} &= \frac{\partial w_{w}}{\partial u_{w,I}} \bigg|_{e} \\ \end{array}$$

$$\frac{\partial \dot{w}_{w}^{*}}{\partial \dot{v}_{w,I}}\Big|_{e} = \frac{\partial w_{w}}{\partial v_{w,I}}\Big|_{e} = \sin\psi_{e}\sin\theta_{e}\cos\phi_{e} - \cos\psi_{e}\sin\phi_{e}$$
$$\frac{\partial \dot{w}_{w}^{*}}{\partial \dot{w}_{w,I}}\Big|_{e} = \frac{\partial w_{w}}{\partial w_{w,I}}\Big|_{e} = \cos\theta_{e}\cos\phi_{e}$$

References

- P. Peddiraju. Development and validation of a dynamics model for an unmanned, finless airship. Master's thesis, Department of Mechanical Engineering, McGill University, 2010.
- [2] J. R. Azinheira; E. C. de Paiva; S. S. Bueno. Influence of wind speed on airship dynamics. AIAA Journal of Guidance, Control and Dynamics, 25(6), 2002.
- [3] Desktop Nexus Aircraft. Zeppelin nt on a mast truck blimps wallpaper 74435. http://aircraft.desktopnexus.com/wallpaper/74435/.
- [4] Zukunft in Brand. Der cargolifter cl160. http://www.zukunft-in-brand.de/ index.php?id=cl160_1.
- [5] E. Earon; C. A. Rabbath; J. Apkarian. Design and control of a novel hybrid vehicle concept. In AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2007.
- [6] Defence Industry Daily. Boeing's skyhook shot: Redefining aerial heavy-lift market? http://www.defenseindustrydaily.com/ Boeings-Skyhook-Shot-Redefining-the-Aerial-Heavy-Lifting-Market-04970/.
- [7] Draganfly Innovations Inc. Picture 38, photo gallery draganflyer x4 uav helicopter aerial video platform. http://www.draganfly.com/uav-helicopter/ draganflyer-x4/gallery/pictures/picture-38.php.
- [8] Y. Li; M. Nahon; I. Sharf. Airship dynamics modelling: A literature review. Progress in Aerospace Sciences, 2011.
- [9] M. M. Munk. Some tables of the factors of apparent additional mass. NACA TR 197, July 1924.
- [10] M. M. Munk. Notes on aerodynamic forces 3: The aerodynamic forces on airships. NACA TN 106, 1922.
- [11] M. M. Munk. Aerodynamics of airships. *Aerodynamics Theory*, 6, 1936.

- [12] H. B. Freeman. Force measurement on a 1/40-scale model of the u.s. airship akron. NACA TR 443, 1933.
- [13] H. Lamb. Hydrodynamics. 6th edition. New York: Dover, 1945.
- [14] H. J. Allen; E. W. Perkins. Characteristics of flow over inclined bodies of revolution. NACA RM A50L07, 1951.
- [15] S. F. Hoerner. *Fluid-dynamic drag.* NJ: Midland Park, 1958.
- [16] Y. Li; M. Nahon. Modeling and simulation of airship dynamics. Journal of Guidance, Control and Dynamics, 30(6), November-December 2007.
- [17] S. B. V. Gomes; J. J. G. Ramos. Airship dynamic modeling for autonomous operation. In *IEEE International Conference on Robotics and Automation*, May 1998.
- [18] P. G. Thomasson. Equations of motion of a vehicle in a moving fluid. AIAA Journal of Aircraft, 37(4), 2000.
- [19] J. R. Azinheira; A. Moutinho; E. C. de Paiva. Erratum influence of wind speed on airship dynamics. AIAA Journal of Guidance, Control and Dynamics, 31(2), 2008.
- [20] T. I. Fossen. Guidance and Control of Ocean Vehicles. Wiley, United Kingdom, 1994.
- [21] L. Jorgensen. Prediction of static aerodynamic characteristics for space-shuttlelike and other bodies at angles of attack from 0° to 180°. NASA TN D-6996, 1973.
- [22] D.-A. Wimmer; K. H. Well. Instrumentation, identification and control of airship "lotte.". In 14th AIAA Lighter-Than-Air-Systems Technology Conference, July 2001.
- [23] E. C. de Paiva; S. S. Bueno; S. B. V. Gomes; J. J. G. Ramos; M. Bergerman. A control system development environment for aurora's semi-autonomous robotic airship. In *IEEE International Conference on Robotics and Automation*, 1999.
- [24] J. R. Azinheira; E. C. de Paiva; J. R. H. Carvalho; J. J. G. Ramos; S. S. Bueno; M. Bergermann; P. A. V. Ferreira. Lateral/directional control for an

autonomous, unmanned airship. Aircraft Engineering and Aerospace Technology, 73, 2001.

- [25] E. C. de Paiva; J. R. H. Carvalho; P. A. V. Ferreira; J. R. Azinheira. An h_2/h_{∞} pid heading controller for aurora-i semi-autonomous robotic airship. In 14th AIAA Lighter-Than-Air-Systems Technology Conference, July 2001.
- [26] E. A. Kulczycki; S. S. Joshi; R. A. Hess; A. Elfes. Towards controller design for autonomous airships using slc and lqr methods. In AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006.
- [27] A. Elfes; S. S. Bueno; M. Bergermann; E. C. de Paiva; J. J. G. Ramos; J. R. Azinheira. Robotic airships for exploration of planetary bodies with an atmosphere: Autonomy challenges. *Journal of Autonomous Robots*, 14, 2003.
- [28] J. R. Azinheira; E. C. de Paiva; J. J. G. Ramos; S. S. Bueno. Mission path following for an autonomous unmanned airship. In *IEEE International Conference* on Robotics and Automation, 2000.
- [29] J. J. G. Ramos; E. C. de Paiva; J. R. Azinheira; S. S. Bueno; S. M. Maeta; L. G. B. Mirisola; M. Bergerman; B. G. Faria. Autonomous flight experiment with a robotic unmanned airship. In *IEEE International Conference on Robotics and Automation*, 2001.
- [30] E. C. de Paiva; J. R. Azinheira; J. J. G. Ramos; A. Moutinho; S. S. Bueno. Project aurora: Infrastructure and flight control experiments for a robotic airship. *Journal of Field Robotics*, 23, 2006.
- [31] G. F. Silveira; J. R. H. Carvalho; P. Rives; J. R. Azinheira; S. S. Bueno; M. K. Madrid. Optimal visual servoed guidance of outdoor autonomous robotic airships. In *American Control Conference*, 2002.
- [32] W. Xiao-Liang; S. Xue-Xiong. Airship attitude tracking system. Applied Mathematics and Mechanics, 27, 2006.
- [33] S.-J. Lee; D.-M. Kim; H.-C. Bang. Feedback linearisation controller for semi station keeping of the unmanned airship. In AIAA 5th Aviation, Technology, Integration and Operations Conference, September 2005.
- [34] C.-S. Park; H. Lee; M.-J. Tahk; H. Bang. Airship control using neural network augmented model inversion. *IEEE Conference on Control Applications*, 2003.

- [35] W. Yongmei; Z. Ming; Z. Zongyu; Z. Zewei. Trajectory tracking of a high altitude unmanned airship based on adaptive feedback linearization. In International Conference on Mechatronic Science, Electrical Engineering and Computer, 2011.
- [36] A. Moutinho; J. R. Azinheira. Stability and robustness analysis of the aurora airship control system using dynamic inversion. In *IEEE International Confer*ence on Robotics and Automation, 2005.
- [37] J. R. Azinheira; A. Moutinho; E. C. de Paiva. Airship hover stabilization using a backstepping control approach. AIAA Journal of Guidance, Control and Dynamics, 29(4), 2006.
- [38] J. R. Azinheira; A. Moutinho. Hover control of an uav with backstepping design including input saturations. *IEEE Transactions on Control Systems Technology*, 2008.
- [39] J. R. Azinheira; A. Moutinho; E. C. de Paiva. A backstepping controller for path-tracking of an underactuated autonomous airship. *International Journal* of Robust and Nonlinear Control, 19, 2009.
- [40] L. Beji; A. Abichou. Tracking control of trim trajectories of a blimp for ascent and descent flight manoeuvres. *International Journal of Control*, 78, 2005.
- [41] L. Beji; A. Abichou; Y. Bestaoui. Stabilization of a nonlinear underactuated autonomous airship - a combined averaging and backstepping approach. In International Workshop on Robot Motion and Control, 2002.
- [42] F. Repoulias; E. Papadopoulos. Robotic airship trajectory tracking control using a backstepping methodology. In *IEEE International conference on Robotics and Automation*, 2008.
- [43] S. Bennaceur; A. Abichou; N. Azouz. Modelling and control of flexible airship. In AIP Intelligent Systems and Automation, 1st Mediterranean Conference, 2008.
- [44] E. Hygounenc; P. Soueres; S. Lacroix. Developments of autonomous airship control at laas/cnrs. In AIAA Lighter-Than-Air Conference and Exhibit, 2001.
- [45] E. Hygounenc; P. Soueres. Automatic airship control involving backstepping techniques. In *IEEE International Conference on System, Man, and Cybernetics*, October 2002.

- [46] E. Hygounenc; P. Soueres. Lateral path following gps-based control of a smallsize unmanned blimp. In *IEEE International Conference on Robotics and Au*tomation, 2003.
- [47] E. Hygounenc; P. Soueres; S. Lacroix; I.-K. Jung. The autonomous blimp project at laas-cnrs: Achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 2004.
- [48] Y. Zhang; W.-D. Qu; Y.-G. Xi. Adaptive stabilization and trajectory tracking of airship with neutral buoyancy. ACTA Automatica Sinica, 34(11), 2008.
- [49] M. Yamada; Y. Taki; A. Katayama; Y. Funahashi. Robust global stabilization and disturbance rejection of an underactuated nonholonomic airship. In *IEEE International Conference on Control Applications*, 2007.
- [50] M. Yamada; H. Adachi; Y. Funahashi. Robust control of an uncertain underactuated airship with asymptotic rejection against wind disturbance. In *IEEE International Conference on Control Applications*, 2010.
- [51] G. Xia; D. A. Corbett. Cooperative control system of searching targets using unmanned blimps. In World Congress on Intelligent Control and Automation, 2004.
- [52] G. Jian-Guo; Z. Jun. Lateral path controller design for autonomous airship. In International Conference on Intelligent Human-Machine Systems and Cybernetics, 2010.
- [53] T. Puls; M. Kemper; R. Kuke; A. Hein. Gps-based position control and waypoint navigation system for quadrocopters. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2009.
- [54] E. D. B. Medagoda; P. W. Gibbens. Synthetic waypoint guidance algorithm for following a desired flight trajectory. AIAA Journal of Guidance, Control and Dynamics, 33(2), 2010.
- [55] M. Niculescu. Lateral track control law for aerosonde uav. In AIAA Aerospace Sciences Meeting and Exhibit, 2001.
- [56] S. Park; J. Deyst; J. P. How. A new nonlinear guidance logic for trajectory tracking. In AIAA Guidance, Navigation, and Control Conference and Exhibit, 2004.

- [57] S. Park; J. Deyst; J. P. How. Performance and lyapunov stability of a nonlinear path-following guidance method. AIAA Journal of Guidance, Control and Dynamics, 30(6), 2007.
- [58] D. A. Lawrence; E. W. Frew; W. J. Pisano. Lyapunov vector fields for autonomous uav flight control. In AIAA Guidance, Navigation, and Control Conference and Exhibit, 2007.
- [59] D. R. Nelson; D. B. Barber; T. W. McLain; R. W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3), 2007.
- [60] R. Cunha; D. Antunes; P. Gomes; C. Silvestre. A path-following preview controller for autonomous air vehicles. In AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006.
- [61] D. J. Gates. Nonlinear path following method. AIAA Journal of Guidance, Control and Dynamics, 33(2), 2010.
- [62] W. Adamski; P. Herman; Y. Bestaoui; K. Kozlowski. Control of airship in case of unpredictable environment conditions. In *Conference on Control and Fault Tolerant Systems*, 2010.
- [63] N. Rooz; E. N. Johnson. Design and modeling of an airship station holding controller for low cost satellite operations. In AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2005.
- [64] A. Katz; M. Sharma. Estimation of wind from airplane states in coordinated flight. AIAA Journal of Aircraft, 35, 1998.
- [65] M. Kumon; I. Mizumoto; Z. Iwai; M. Nagata. Wind estimation by unmanned air vehicle with delta wing. In *IEEE International Conference on Robotics and Automation*, 2005.
- [66] M. Kumon; M. Nagata; R. Kohzawa; I. Mizumoto; Z. Iwai. Flight path control of small unmanned air vehicle. *Journal of Field Robotics*, 23, 2006.
- [67] N. de Divitiis. Wind estimation on a lightweight vertical-takeoff-and-landing uninhabited vehicle. AIAA Journal of Aircraft, 40, 2003.
- [68] B. L Stevens; F. L. Lewis. Aircraft Control and Simulation. John Wiley and Sons, Inc., 2004.

- [69] W. F. Phillips; C. E. Hailey. Kalman filtering for spacecraft attitude estimation. *AIAA Journal of Aircraft*, 38, 2001.
- [70] A. Tayebi. Unit quaternion-based output feedback for the attitude tracking problem. *IEEE Transactions on Automatic Control*, 53, 2008.
- [71] J. B. Kuipers. Quaternions and rotation sequences. Princeton University Press, 1999.
- [72] B. Etkin. Dynamics of Atmospheric Flight. Wiley, New York, 1972.
- [73] M. Nahon; G. Gilardi; C. Lambert. Dynamics/control of a radio telescope receiver supported by a tethered aerostat. AIAA Journal of Guidance, Control, and Dynamics, 25(6):1107–1115, 2002.
- [74] P. Coulombe-Pontbriand. Modeling and experimental characterization of a tethered spherical aerostat. Master's thesis, McGill University, 2005.
- [75] P. Peddiraju; T. Liesk; M. Nahon. Dynamics modeling for an unmanned, unstable, fin-less airship. In 18th AIAA Lighter-Than-Air Systems Technology Conference, May 2009.
- [76] J. C. Doyle K. Zhou. *Essentials of Robust Control*. Prentice Hall, 1998.
- [77] T. F. Coleman; Y. Li. Reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. SIAM Journal on Optimization, 6, 1996.
- [78] M. Krstic; I. Kanellakopoulus; P. Kokotovic. Nonlinear and Adaptive Control Design. John Wiley and Sons, Inc., 1995.
- [79] Y. Tan; J. Hu; J. Chang; H. Tan. Adaptive integral backstepping motion control and experiment implementation. In *IEEE Industry Applications Society Annual Meeting*, 2000.
- [80] M. Nahon. Optimization of Force Distribution in Redundantly-Actuated Robotic Systems. PhD thesis, McGill University, Montreal, Canada, 1991.
- [81] S. Joassard. Controle d'un dirigeable sans gouverne. Technical report, INSA Lyon, 2010.
- [82] B. Singh. Online aerodynamic parameter estimation for a fault tolerant flight control system. Master's thesis, Wichita State University, 2005.

[83] The MathWorks. Discrete fourier transform. http://www.mathworks.com/ help/techdoc/ref/fft.html.