Masterthesis

Precise Position Control of a Flapping-wing Micro Air Vehicle in a Wind-tunnel

Torbjørn Cunis, B.Sc. B.Sc.

342015

an der Fakultät für Maschinenwesen der

Rheinisch-Westfälischen Technischen Hochschule Aachen

Diese Arbeit wurde vorgelegt am

Institut für Flugsystemdynamik

durchgeführt am

Micro Air Vehicle Laboratory

13. Mai 2016

Betreuung:

Prof. Dr.-Ing. Dieter Moormann

Dr.-Ing. Norbert Siepenkötter

dr. Guido C.H.E. de Croon (TUDelft)

dr. ir. Matěj Karásek (TUDelft)

Eidesstattliche Versicherung

Name, Vorname	Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum	Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum	Unterschrift

Abstract

FLYING LIKE A BIRD has been both eternal dream and very first attempt of human flight; re-born in Micro Air Vehicles, flapping-wing propulsion made a comeback in today's aerospace engineering. The Delfly, a minimalistic representative developed at the Delft University of Technology, proves the success of an MAV miniaturization. However, the Delfly's full flight-envelope aerodynamics are still not understood completely.

In this Master's thesis, we focus on the high-precision position control problem of the Delfly flapping-wing MAV flying freely in a wind-tunnel, in order to enable future *in-flight* particle image velocimetry (PIV) investigation of the air-flow around the flapping wings. Both the limited model as well as the light-weight, simplistic character of the Delfly are challenging the control design.

We propose a hierarchical control scheme for the Delfly, which we implemented within the open-source paparazzi UAV autopilot software. Using a de-coupling, combined forward-backward control approach as core, we were able to achieve a precision much beyond the PIV requirements.

Kurzfassung

Wie ein Vogel zu fliegen, der ewige Menschheitstraum, ist gleichzeitig Vorbild für die ersten Flugversuche gewesen. Realisiert als Flugmodelle (Micro Air Vehicle) sind Ornithopter heute wieder im Fokus der Luftfahrtforschung. Dabei ist der von der TU Delft entwickelte Delfly, ein minimalistischer Vertreter, das Ergebnis einer erfolgreichen Minituarisierung; jedoch ist die Aerodynamik des Delflys noch nicht vollständig erforscht.

Im Rahmen dieser Masterarbeit wird das Problem einer präzisen Positionsregelung für den frei im Windtunnel fliegenden Delfly betrachtet; zukünftig soll auf diese Weise mit der Particle Image Velocimetry (PIV) Technik die Luftströme um die Flügel des Delfly *im freien Flug* untersucht werden. Besondere Herausforderungen stellen dabei sowohl das beschränkte Modellwissen als auch die leichtgewichtige und simple Struktur des Delfly dar.

Vorgestellt wird ein hierarchischer Reglerstruktur, implementiert in Rahmen der open-source UAV-Autopilotensoftware paparazzi. Mit einem entkoppelnden Regler mit Vorsteuerung als Hauptteil waren wir in der Lage Ergebnisse zu erzielen, die die Anforderungen von PIV hinsichtlich der Genauigkeit weit übertreffen.

Dedication

To all professors, teachers, assistants, and tutors of the universities of Hamburg, Würzburg, Aachen, and Delft who have prepared the ground of these Thesis for the last seven years.

To my friends, colleagues, and family – wherever they are – who accompanied, backed, and encouraged me.

Preface

This thesis has been completed at the Delft University of Technology, in the scope of the Delfly project of the Micro Air Vehicle Laboratory. I wish to thank everybody who contributed to this work and thesis, and in particular:

dr. Guido C.H.E. de Croon for his supervision and academic guidance.

dr. ir. Matěj Karásek for his daily supervision, team work in the wind-tunnel, and very helpful advices, like: “Quadrupel-check it!”

The technician in the MAVlab, Erik van der Horst, for his practical support for a theoretical control engineer.

The Delft Faculty of Aerospace Engineering in general for granting hospitality and coffee.

Sarah Gluschitz for her illustrations of the Delfly and the wind-tunnel.

Prof. Dieter Moormann and Dr.-Ing. Norbert Siepenkötter of the Aachen Institute of Flight System Dynamics for remotely supervising this thesis.

In order to successfully finish my Thesis, I took advantage of my recent and former studies in aerospace flight dynamics, control and systems theory, modular software development, and programming of embedded systems, to call a few only. I am grateful for having obtained a comprehensive education by the Universities of Hamburg, Würzburg, and Aachen.

Torbjørn Cunis, *Delft, March 2016*

Contents

Appendix

List of Tables

List of Figures

Nomenclature

Variables of flight dynamics		*Variables of control theory*
scalars		
	d	derivative gain
flapping frequency	f	frequency
	i	integral gain
	k	proportional gain
roll rate	p	
pitch rate	q	
yaw rate	r	
	s	*Laplace variable*
time	t	
velocity along 1st axis	u	control input
velocity along 2nd axis	v	
velocity along 3rd axis	w	
position coordinate, 1st axis	x	plant state
position coordinate, 2nd axis	y	plant output
position coordinate, 3rd axis	z	disturbance
force	F	
	G	(SISO) plant, system
height	H	
	$J(\cdot)$	cost function
	K	(SISO) controller
roll moment	L	
pitch moment	M	
yaw moment	N	
speed[1]	V	
force along 1st axis	X	
force along 2nd axis	Y	
force along 3rd axis	Z	

[1] The different speed variables – air-speed, path speed, wind speed, etc. – are distinguished by the respective subscripts (cf. Nomenclature: *superscripts, subscripts, and accents*).

Variables of flight dynamics		*Variables of control theory*
angle-of-attack	α	
side-slip angle	β	
angle-of-climb	γ	adaptation rate
rudder deflection	ζ	damping constant
elevator deflection	η	
	θ	adaptable gain
	ν	pseudo-input
	ς	pole
	τ	time constant
azimuth	χ	
	ω	angular frequency
change of, difference	Δ	error in, difference
pitch angle	Θ	
bank angle	Φ	
heading angle	Ψ	
throttle	$\digamma$	
vectors		
	$\mathbf{u}$	control inputs
position $\begin{bmatrix} x & y & z \end{bmatrix}^T$	$\mathbf{x}$	plant states
	$\mathbf{y}$	plant outputs
moments $\begin{bmatrix} L & M & N \end{bmatrix}^T$	$\mathbf{Q}$	
forces $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$	$\mathbf{R}$	
velocity $\begin{bmatrix} u & v & w \end{bmatrix}^T$	$\mathbf{V}$	
attitude $\begin{bmatrix} \Phi & \Theta & \Psi \end{bmatrix}^T$	$\boldsymbol{\Phi}$	
matrices		
	$\mathbf{A}$	state matrix
	$\mathbf{B}$	input matrix
	$\mathbf{C}$	output matrix
	$\mathbf{D}$	feed-through matrix
	$\mathbf{G}$	(MIMO) plant, system
	$\mathbf{I}$	identity matrix
	$\mathbf{K}$	feed-back matrix, (MIMO) controller
transformation matrix	$\mathbf{M}$	

Variables of flight dynamics		*Variables of control theory*
superscripts, subscripts, and accents		
equilibrium	$\odot_0$	initial value
air-path axis system	$\odot_a$	
	$\odot^c$	command variable
body axis system	$\odot_f$	
normal earth-fixed system	$\odot_g$	
flight-path axis system	$\odot_k$	discrete value at step k
	$\odot^{ref}$	reference variable
wind-tunnel axis system	$\odot_w$	
relative to air	$\odot_A$	
aerodynamic drag (force)	$\odot_D$	
thrust (force)	$\odot_F$	
weight (force)	$\odot_G$	
	$\odot^H$	conjugate matrix transponse
relative to ground	$\odot_K$	
lift (force)	$\odot_L$	
	$\odot^{SP}$	set-point variable
	$\odot^T$	matrix transponse
wind variables	$\odot_W$	
aerodynamic (force/moment)	$\odot_{Æ}$	
1st derivative to time	$\dot{\odot}$	
2nd derivative to time	$\ddot{\odot}$	
	$\tilde{\odot}$	value substitute
	$\hat{\odot}$	estimated value

Remark The overall structure of this nomenclature has been adopted from [1] in order to distinguish the symbols of aerodynamics and control theory, which partially coincide.

1. Introduction

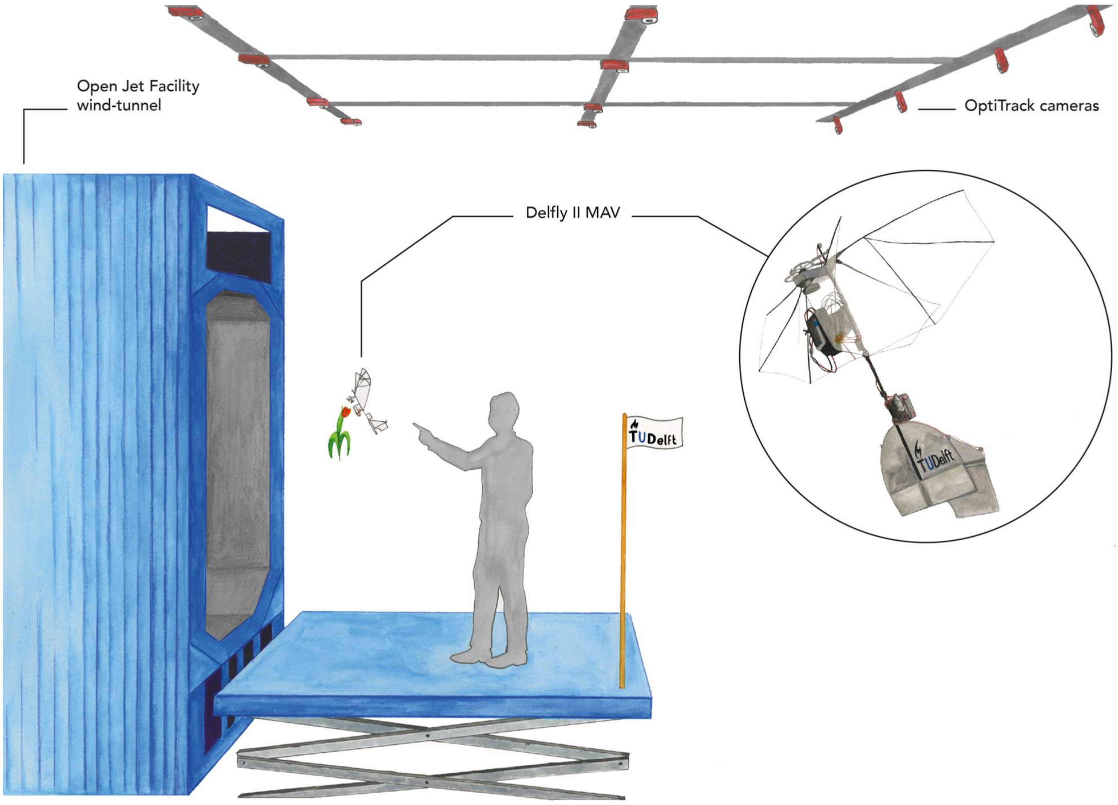
Ab initio, humankind has sought to equal flying animals challenging the ground-binding forces of gravity as easily as bird does. Learnèd men and women have tried to understand and mimick the avian flying techniques through the history: the ancient Greeks Daedalos and Ikaros are told to have flown from Minos' labyrinth in Crete by self-built avian wings [2]; and the visionary polymath Leonardo da Vinci discussed in *Codex on the Flight of Birds* [3] not only the structure and physics of birds' wings and flight but furthermore sketched a human-powered flying machine.(1)
In the later centuries, several daring aeronauts failed flying their – usually self-constructed – muscle-driven ornithopters [5]. When the first fixed-wing propulsion-equipped airplane was proposed by George Cayley [6], and Lilienthal and the Wright brothers succeeded in manned flight, gliding and engined respectively [7], the era of modern aircrafts had begun over-ruling flapping-wing as considered propulsion system [8].

However, birds and bird-like flight, respectively, have remained object of study of ecologists [9–14] and bio-inspired engineers [15–17]. This resulted amongst else in attempts of modeling [18–20] and successful replication [21] of flying animals. Meanwhile, the recent rise of Micro Air Vehicles (MAV) from first rubberband-driven, later radio-controlled model airplanes [22], to autonomously deciding platforms for several applications induced a re-birth of flapping-wing aircrafts in small-scale [23–25]: for a better lift-to-weight ratio and lower Reynolds numbers, miniaturized ornithopters are more easy to realise than their historical, full-scale ancestors [26]; while being smaller and lighter than competitive quadrocopters and potentially more agile than fixed-wing air vehicles, flapping-wing MAVs are now challenging conventional concepts.

Notable implementations are the Harvard University *RoboBee* [27–29], the AeroVironment Inc. *Nano Hummingbird* [30], and last but not least the TU Delft *Delfly II* [31, 32]. On the other hand, the flight performances of ornithopters are still limited by the lack of fully-understood aero-dynamics and complete models, complicating both the design of the aircraft itself and the development of advanced controllers.

(1) *cf.* as well [4] pp. 53–55, 111, 172, 271–273, 288–289, and 291.

Studies of flapping-wing aero-dynamics have covered measurement of forces (lift and drag) and moments induced by the wing flap of insects [33, 34] and ornithopters [35–38]; these experiments are common to examine fixed- and flexible-wing MAV aero-dynamics [39–41]. As well, the corresponding forces were estimated in free-flight experiments of ornithopters [37, 38, 42] and birds flying in a wind-tunnel [43, 44]. Thereby, both linear and non-linear models of basic flapping-wing kinematics were obtained [33, 34, 42]. But still, these models cover a certain condition of flight only: further techniques are needed to provide a full-envelope understanding of flapping-wing aero-dynamics.



The Delfly II MAV flying in the Open Jet Facility wind-tunnel. *Commissioned illustration.*

Nowadays, in order to study flapping-wing aerodynamics zoologists are imaging the so-called leading- and trailing-edge vortices, of flying animals using flow visualization techniques. Notably, particular image velocimetry (PIV) measures the velocity of the particles of the surrounding fluid and the vortex circulation induced by the wing flap can be calculated; additional lift is considered to be generated by the leading-edge vortex [45]. PIV has been applied for studies of desert locusts [46], various bird species [47–49], and lesser long-nose bats [50] flying in a wind-tunnel.

The kinematics of flexible- and flapping-wing MAVs has been investigated in similar manner [35, 51–53], tethered but not yet in free flight. Of the Delfly II MAV in particular, the aerodynamic forces have been measured statically and the air flow has been visualized using various

PIV techniques while tethered [54–57]; furthermore, a linear model of its aerodynamics was identified based on flight test data [58–60]. Now, the Delfly shall be examined using PIV but flying un-attached in the wind-tunnel, since the usual tethering is expected to influence the aero-dynamics significantly. As for effective PIV measurements the object of study is to hold a fixed position during the process of imaging, free-flying birds and bats are trained to fly in front of a feeder, providing a sweet syrup. However, for the Delfly ignores any kind of feed, it needs its position stabilized by an active controller. This is a novel approach to investigate the aerodynamics of a flapping-wing micro air vehicle.

So far, few wind-tunnel experiments have been performed with air vehicles flying loosely for the tethering provides as well measurement of the aero-dynamic forces and moments. Two, independent investigations of the US National Aeronautics and Space Administration (NASA) included models of forward-swept-wing [61] and blended-wing [62] un-tethered but remote-controlled. A closed-loop controlled wind-tunnel flight was performed by [63] at the RWTH Aachen University, who implemented adaptive identification and control algorithms for a fixed-wing aircraft model. Most recent, a quadrocopter flew controlled in a wind-tunnel at the Massachusetts Institute of Technology (MIT) in order to investigate its high-speed forward flight behaviour [64]. However, controlled flapping-wing free-flight in a wind-tunnel has only been achieved yet using the Delfly platform itself [65, 66].

Flapping-wing MAVs can basically be distinguished by either absence or existence of a tail, both stabilizing and steering the system. Insect-like vehicles like the RoboBee and the Nano Hummingbird have none, the Delfly and early prototypes of the Nano Hummingbird do have one.[(2)] So far, control techniques for the insect-like group focus mainly on attitude stabilization and control, where the necessary body forces and moments are induced through adjustable wing actuactors [67, 68] by either dynamic-inverting control [69–73] or state-optimal control [74–76] based on proposed or estimated time-averaged kinematics, partially using periodic control to cope the body's oscillations caused by wing flap [69, 75]. Except for [73], these were tested in simulation only; *in rerum natura* limited to the conditions wherefore the underlying models are derived and they are not robust against changing flight conditions – *i.e.*, for example increasing or decreasing pitch angles. Robust control techniques, e.g. mere static-gain control, as employed for fixed-wing or bird-like flapping-wing MAVs are not suitable for high-performance trajectory tracking in return.

Traditionally, aerospace flight control is divided in *partes tres*: the inner loop is damping, stabilizing, and controlling the attitude,[(3)] thus called *basic control*; next, the *guidance* loop

[(2)] Therefore, the Delfly can be referred as being *bird-like*, though actually it is neither.

[(3)] Historically, early flight control approaches only provided damping (of stable airplanes) and stabilization (of un-stable airplanes), respectively, in order to assist the human pilots.

ensures height, velocity, and heading tracking; and the outer-most *navigation* loop provides flight trajectories, optimisation against fuel-consumption or weather-influences, autonomous collision (TCAS) or obstacle [77, 78] avoidance, and so forth [1, 79]. This cascade control structure is common practice in the industry for independent design and, more important in human flight, for verification of each level control implementations [80]. In contrast, advanced and more-robust control approaches have been proposed, also bypassing the cascade structure [81].

The Delfly II flapping-wing MAV is a challenging platform for its light-weight and fragile structure, mass requirements, and limited flight time of the battery. This thesis presents the design and tests of a precise position control approach for the Delfly II in the Open Jet Facility wind-tunnel of the Delft University of Technology; here, the Delfly's position has been tracked using the camera-based OptiTrack motion tracking system. We will introduce the Delfly II and the fundamentals of flight control. After discussing basic control theory, we derive and explain a hierarchical position controller as well as its realisation within the open-source paparazzi autopilot software. Eventually, we present and discuss the test results in terms of steady-state precision, settling behaviour, and robustness against disturbances and wind speed. We will argue the novelty of our control solution and the benefits from its modular implementation with respect to succeeding work.

2. Delfly MAV, Wind-tunnel Set-up, and Preliminary Experiments

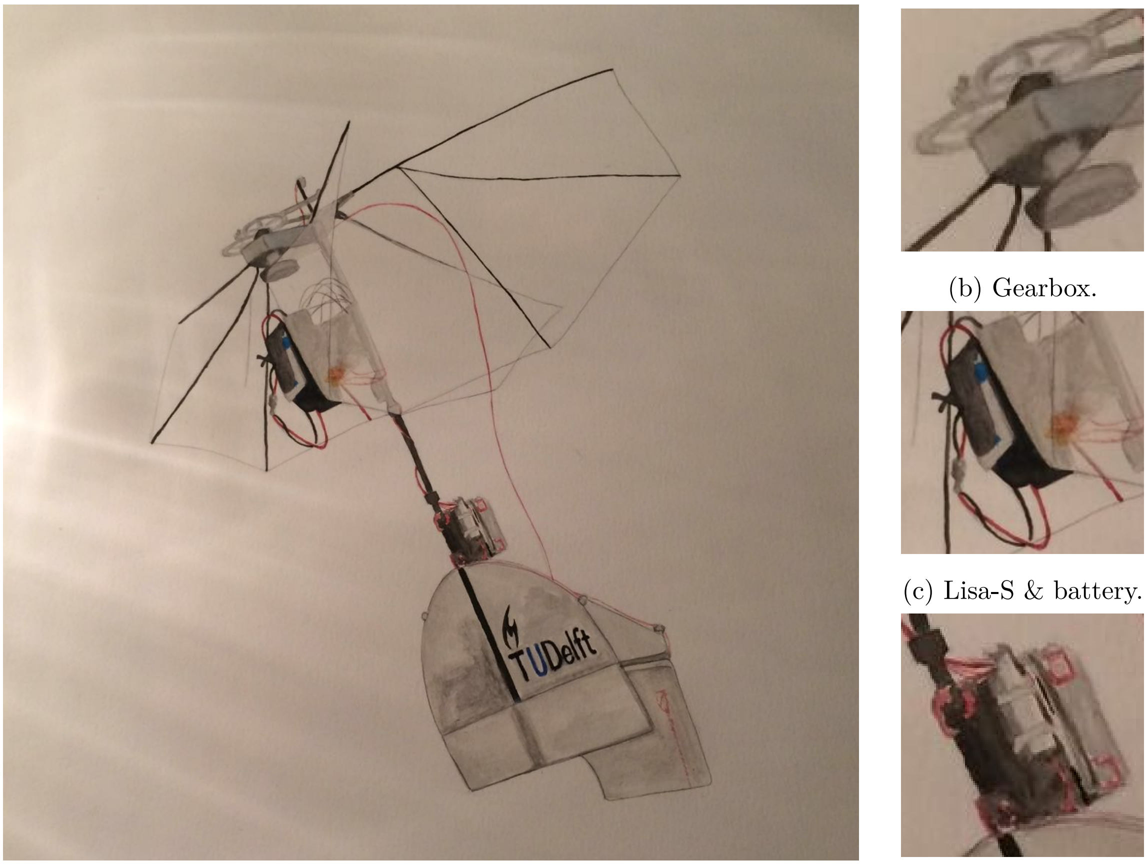
In this chapter, we introduce the Delfly II micro air vehicle (MAV). As well, we briefly present the wind-tunnel our test took place in and discuss the experimental set-up of our tests. We eventually recall preliminary experiments of the Delfly flying freely in a wind-tunnel.

2.1. The Delfly II MAV

The Delfly flapping-wing MAV has been developed by the TU Delft for over ten years. [32] In its different versions, the Delfly Micro is the smallest flapping-wing MAV carrying a camera [31]; the Delfly Explorer carries a stereo-vision camera in order to perform collision-free, autonomous flight [77]; and overall, the Delfly provides a platform to study flapping-wing aerodynamics (especially Delfly I and II). In this thesis, we employ the Delfly II, of which the configuration is shown in Fig. 2.1.

Structure and hardware As all Delfly versions have in common, the Delfly II MAV is equipped with two pairs of semi-rigid wings placed above each other; during one wing flapping cycle, the lower wing pair moves down and up again while the upper wing pair does the opposite concurrently. A full flapping cycle of the wings is shown in the Figures 2.2a to 2.2c.

The main fuselage, which will also define the body x_f-axis, is oriented orthogonal to the wings' leading edges: at its tip, the Delfly's nose, the gearing mechanism is placed which, with servo and gearing axles parallel to the fuselage, drives the wing flapping. Opposite, the tailplane and fin provide stabilization and control of the Delfly attitude; the elevator applies a pitch moment, the rudder a combined roll and yaw moment to the system.



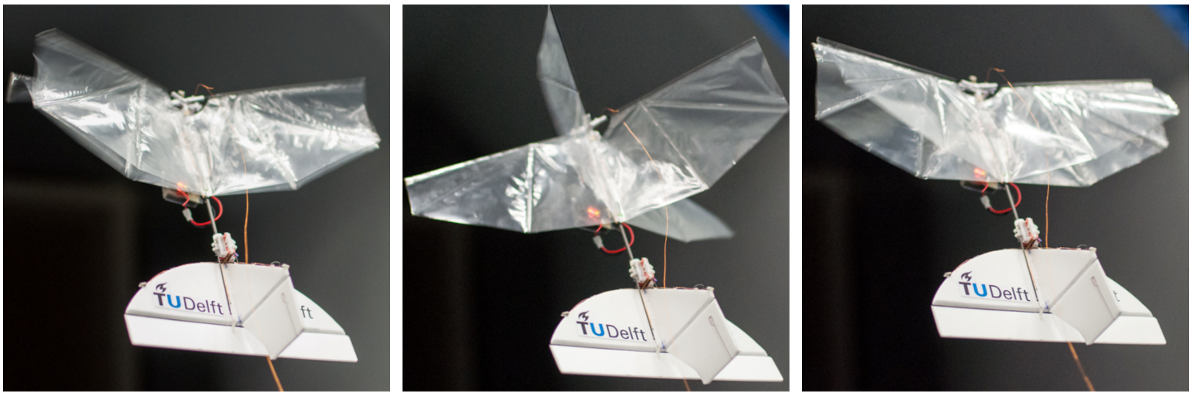
(b) Gearbox.

(c) Lisa-S & battery.

(a) Delfly II

(d) Tail servos.

Figure 2.1.: The Delfly II flapping-wing MAV. *Commissioned illustration.*



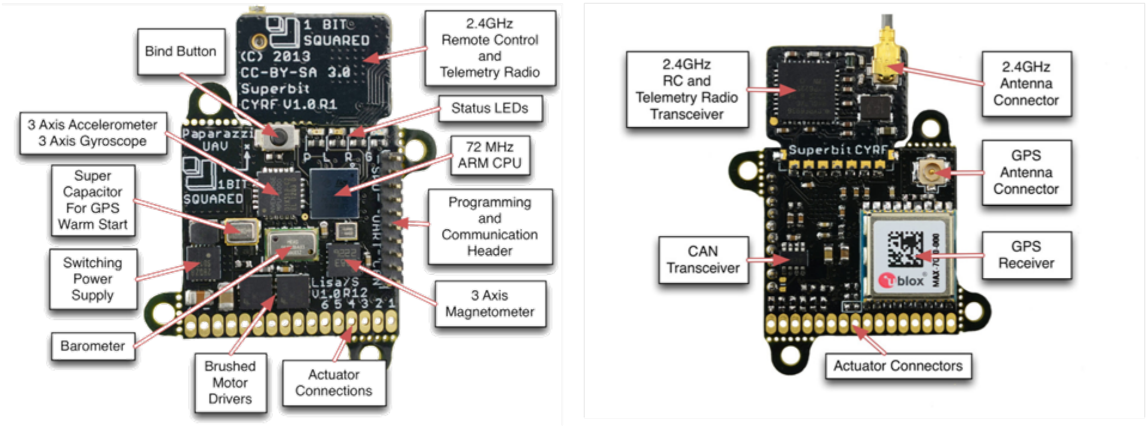
(a)

(b)

(c)

Figure 2.2.: The flapping cycle of the two Delfly wing pairs: opening (a), opened (b), closing (c), closed (*not shown*).

Autopilot board and link In order to run the flight control software, we employ a *Lisa-S* micro-controller board. With a weight of 2.8 g as well as a size of 2 cm × 2 cm only, this board includes gyroscope and accelerometer; furthermore, not used in this work, and magnetometer and GPS module. [82] The Lisa-S board is shown in Fig. 2.3 on page 7.



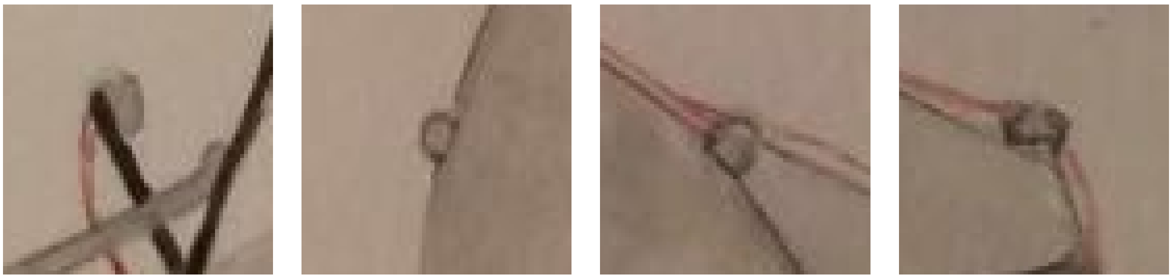
(a) Top view.

(b) Bottom view.

Figure 2.3.: The Lisa-S micro-controller board, from top (a) and bottom (b). *Remes et al. 2014 [82], Fig. 1 and 2.*

For telemetry, telecommands, and position uplink, the serial port of the Lisa-S board is bridged to a wireless network connection into a wireless network (WLAN). Messages to and from the ground control station are then sent using the *Transmission Control Protocol* (TCP/IP).

OptiTrack markers To be trackable by the OptiTrack system (*cf.* Sec. 2.2) the Delfly MAV is equipped with four active, infra-red LED markers. Those are placed at the nose, fin, and tailplane (one each left and right), forming a rotational asymmetric tetrahedron; thus, attitude and position of its geometrical centre[1] can be uniquely recognized.



(a) Nose.

(b) Tail starboard.

(c) Tail portboard.

(d) Fin.

Figure 2.4.: Placement of the OptiTrack markers (cf. Fig. 2.1). *Commissioned illustration.*

[1] This does not necessarily coincide with the centre of gravity.

2.2. Wind-tunnel experimental set-up

Open Jet Facility In order to perform our tests, we were gratefully able to use the *Open Jet Facility* of the Delft Faculty of Aerospace Engineering [83]. This low-speed wind-tunnel located in the facilities of the faculty's Aerodynamics Laboratories features an open test section, that is we could easily enter the flight area and directly interact with Delfly. The Open Jet Facility is capable of wind speeds up to $35\,\frac{\mathrm{m}}{\mathrm{s}}$ and offers a test section of nearly $3\,\mathrm{m} \times 3\,\mathrm{m}$. Thus, it is greatly suitable for micro air vehicle free-flight experiments.

The Open Jet Facility is build as closed-circuit wind-tunnel, *i.e.* the air which is pushed into the test section by a fan is fed back to the fan again. Fig. 2.5 schematically shows this closed circuit with the test section (in the front), two 90°-curves each before and after the test section (left and right) and the fan (in the rear). In front of the test section, five fine-mesh screens are installed in order to ensure a laminar airflow.

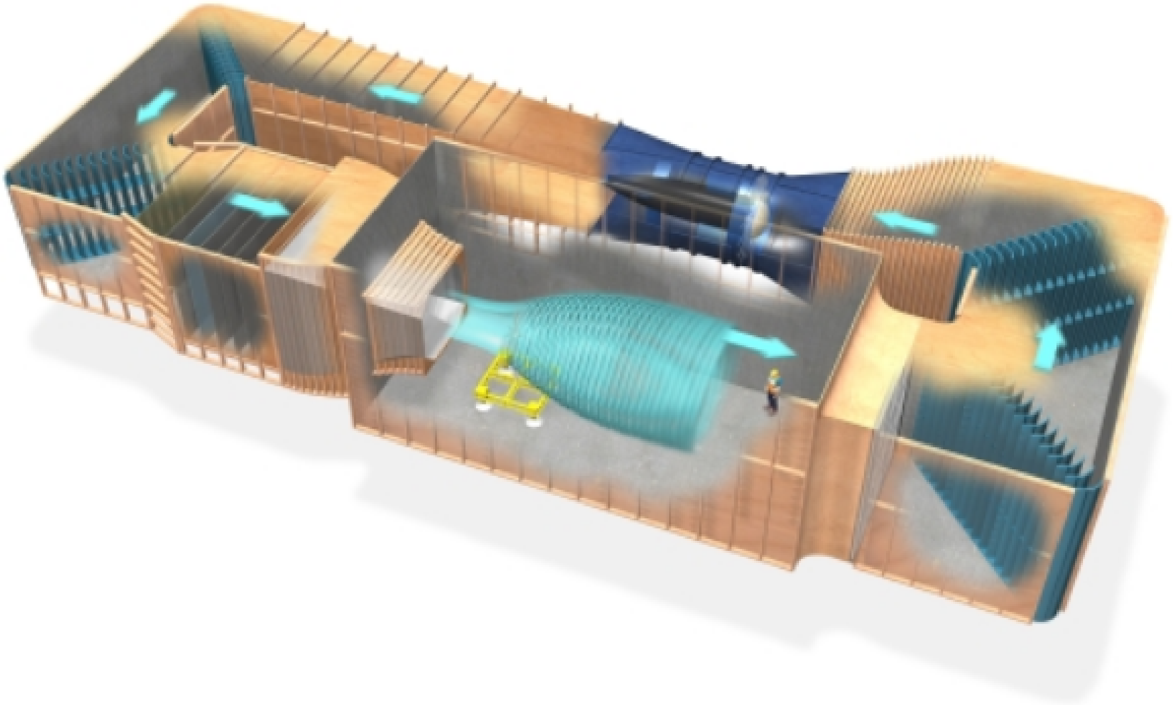
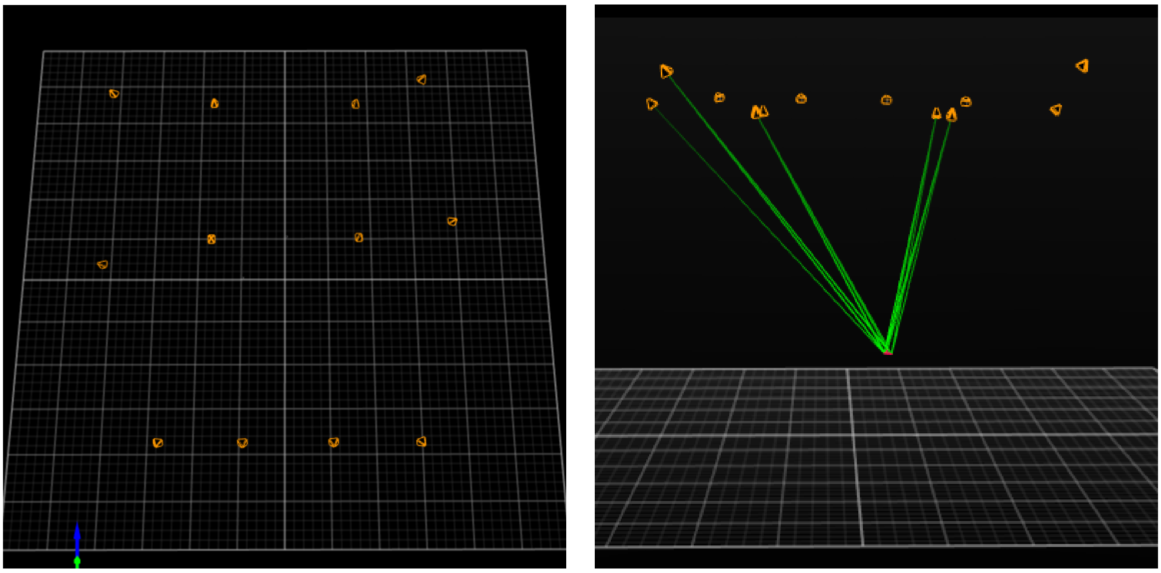


Figure 2.5.: Schematic representation of the Open Jet Facility closed-circuit. *Faculteit Luchtvaart- en Ruimtevaarttechniek 2016 [83], "Ins and outs of the OJF".*

By the control software of the wind-tunnel, the wind speed in the test section is controlled via the fan speed. However, as the wind speed is measured by a pitot tube, *i.e.* through static and dynamic pressure, we had to notice that the accuracy of the wind speed control is a matter of calibration and, for low wind speeds in particular, rather poor. In order to emphasize, we will state the wind speed set-point V_W^{SP} rather than the (unknown) actual speed V_W.

OptiTrack motion tracking The position of the Delfly has been tracked in three axes using twelve OptiTrack Flex 13 cameras [84] attached at the ceiling of the wind-tunnel, which are capturing the markers of the Delfly. From here, the proprietary OptiTrack *Motive* software is calculating off-board position and attitude of the Delfly in a rate up to 120 Hz. The configuration of the OptiTrack cameras is shown in Fig. 2.6. With this, the OptiTrack system establishes a mean error of $\approx 0.2\,\text{mm}$ *per marker*.



(a) Camera configuration from top.

(b) Camera configuration from below.

Figure 2.6.: Configuration of the 12 OptiTrack cameras in the wind-tunnel; looking from top (a) and from below (b).

Ground network, ground control station, and uplink From the OptiTrack Motive software, the Delfly's position and attitude is streamed into a wired ground network (wired LAN) including the ground control station. The latter, running the paparazzi Ground Control Station software [85], is forwarding this data to the Delfly autopilot through a wireless network connection (WLAN) at the rate of 30 Hz.

Fig. 2.7 on page 10 shows the full experimental set-up including the Delfly in the wind-tunnel, the OptiTrack system, the ground control station, and ground and uplink network.

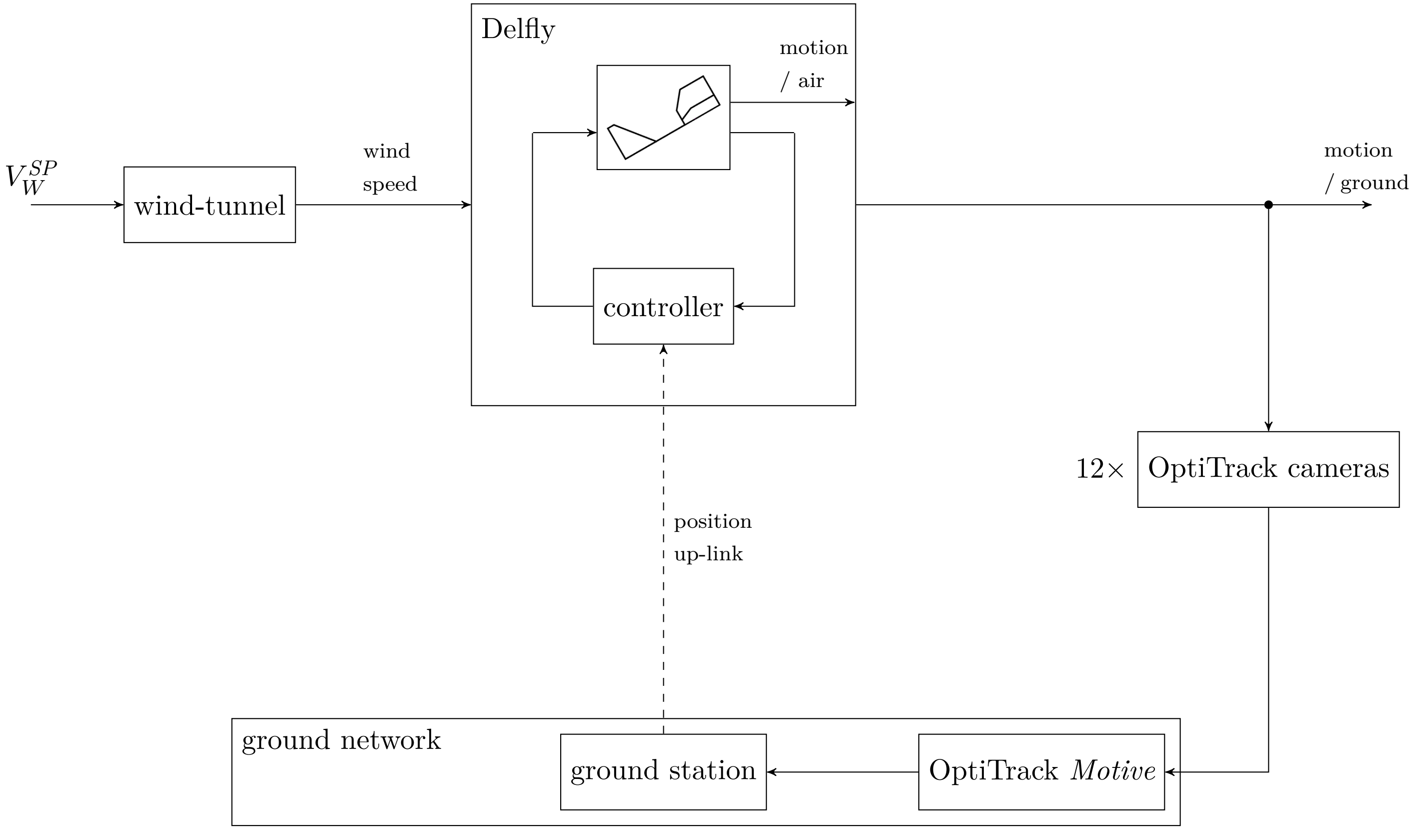


Figure 2.7.: Scheme of the experimental set-up: the motion of the Delfly flying in the wind-tunnel is captured by the OptiTrack system and broadcast into the ground network; the ground control station sends the position to the flight controller.

2.3. Preliminary wind-tunnel experiments

Beforehand, free-flight experiments of the Delfly in the Open Jet Facility wind-tunnel have been performed twice:

Koopmans 2012 First, by Andries Koopmans in 2012 [65, 66] in the scope of his Master's thesis. He tracked the position of the Delfly off-board using two WiiMote controllers and applied static-gain control laws (proportional, integral, and derivative gains) from the position error directly to throttle and the control surfaces. Therewith, he was able to achieve a precision of ±5 cm at the best which is, however, unsuitable for more ambitious research approaches like PIV in free flight.

This thesis Second, right in the beginning of this thesis. Here, we used the paparazzi autopilot software in order to control the Delfly for the first time but stressed the built-in control scheme for rotorcraft MAVs. Position and attitude of the Delfly have already been tracked by the OptiTrack system and send to the autopilot via SuperbitRF link. That is, static-gain control laws were applied from vertical, forward, and lateral position error to throttle command, desired pitch angle, and desired heading, respectively; the attitude of the Delfly was controlled in inner loop by further static-gain laws to elevator and rudder deflection. While the outer loop has been replaced by the control approach elaborately discussed in the next chapters, we kept but tuned the inner loop. Unfortunately, we cannot call on the result of this tests since the position up-link from the OptiTrack system to the autopilot has been disrupted by serious lacks of connection.

Figures 2.8 and 2.9 on page 12 show the control response in the wind-tunnel in vertical and forward axis, respectively. Good to see, the on-board position signal stays constant up to a couple of seconds, while the Delfly is obviously changing is position (tracked off-board). This is apparently affecting the control outcome to the worse.

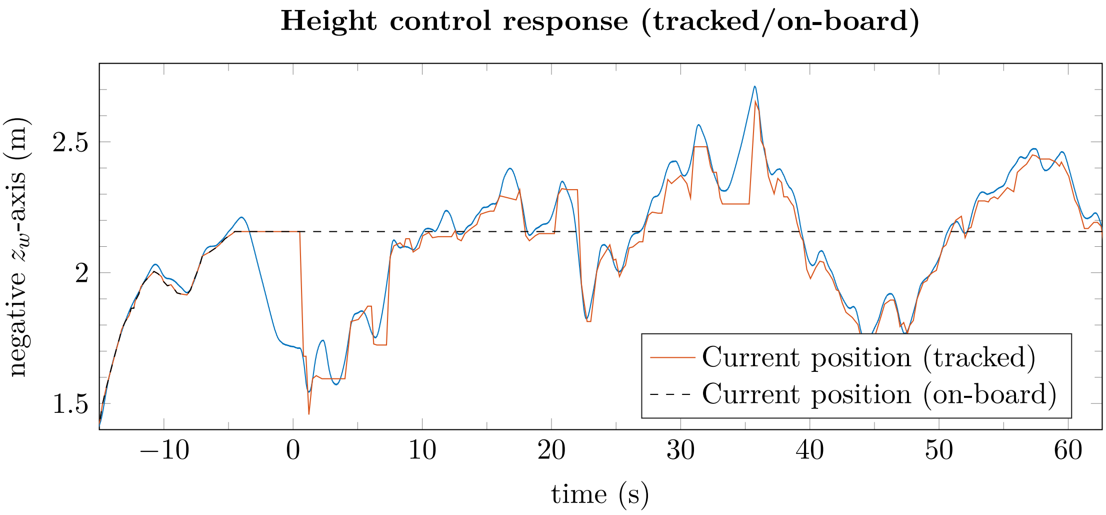


Figure 2.8.: Preliminary wind-tunnel flight results in the vertical $-z_w$-axis; due to uplink issues there is discrepancy between off-board tracked and on-board received position.

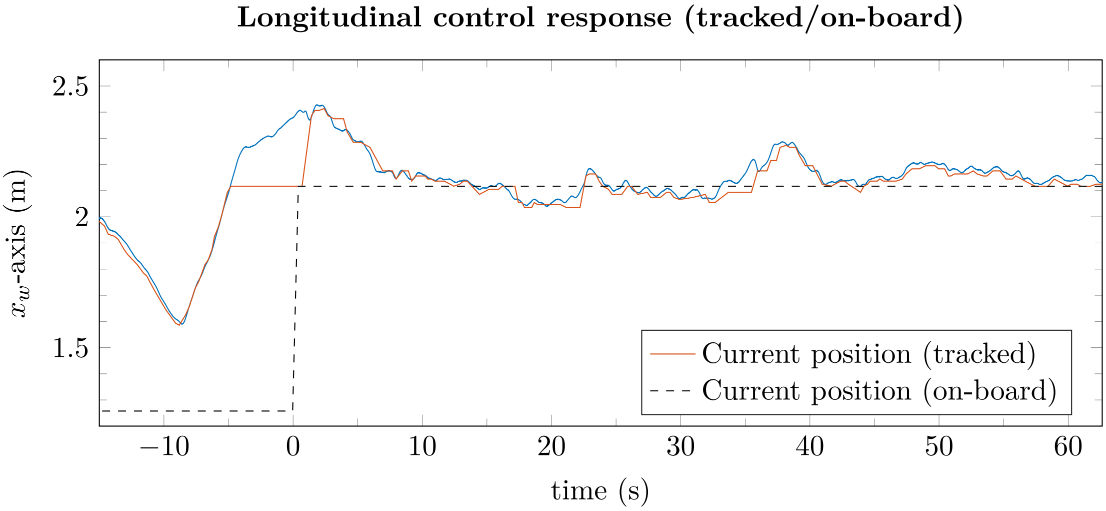


Figure 2.9.: Preliminary wind-tunnel flight results in the horizontal x_w-axis; due to uplink issues there is discrepancy between off-board tracked and on-board received position.

3. Fundamentals

We define subsequently the necessary reference frames, variables, and conventions used in this Thesis.

In flight dynamics, the nomenclature is obeying [DIN 9300-1] (axis systems, angles, and vectors of the aircraft relative to air) and [DIN 9300-2] (axis systems, angles, and vectors of aircraft and atmosphere relative to ground), yet extended as follows: the earth-fixed axis system $[x_0, y_0, z_0]$ ([DIN 9300-1]) is called *wind-tunnel axis system*, $[x_w, y_w, z_w]$, for unambiguity; forces are rather notated by $F_\wp$ where $\wp$ denotes the kind of force according to [DIN 9300-1]; suitable notations have been introduced in order to cover the characteristica of flapping-wing (micro) air vehicles.

In control, we adapt to [DIN IEC 60050-351] but drop some notations in order to ensure a more consistent termination through different control techniques and their applications to flight control. So, we prefer not to denote plant output, reference input, control error, and control input by x, w, e, and y as in [DIN IEC 60050-351] or e.q. found in [86]; instead, we use y, y^{SP}, Δy, and $u^{(c)}$, respectively, where y and u are either an abstract plant input or output or the respective variables of flight dynamics. We furthermore distinguish *reference*, *set-point*, and *command* variables: the command $\odot^c$ is given by an outer, superposing controller; the set-point $\odot^{SP}$ is a demand from either the user (top-level control loop) or a superior loop; while the reference $\odot^{ref}$ is a trajectory calculated from and substituting the set-point.

Transformation matrices, control gains, and transfer functions will be indicated using a double subscript, to be read *from right to left*; that is, the transformation from geographic into body-fixed coordinates is $\mathbf{M}_{fg}$, the proportional gain from pitch rate to elevator deflection is $k_{\eta q}$, and the transfer function from heading command to actual heading is $G_{\Psi\Psi^c}$.

As convenient, we denote linear aero-dynamic coefficients obtained by first-order Taylor series expansion by their respective variable in subscript.

3.1. Axis systems

In flight dynamics and control, there are several axis systems serving different purposes and at levels of the control hierarchy. The axis systems used in this thesis are well-defined by [DIN 9300-1] and [DIN 9300-2], and presented here. All systems in the longitudinal and horizontal plane are shown in Fig. 3.1 and 3.2 on page 15, respectively, including the relevant force and velocity vectors as well as the transferring angles. Projections of axes or quantities into the drawing layer are marked by $\odot'$.

Geographic axis system $\left[x_g, y_g, z_g\right]$ The "outer-most" axis system, defined along the earth surface: the x_g-axis points usually towards North, the z_g-axis points, by convention, down, and the y_g-axis completes the set-up. The system's origin is arbitrary but fixed. In our thesis, the origin and the direction of the x_g-axis are rather defined by the OptiTrack system.

Wind-tunnel reference axis system $[x_w, y_w, z_w]$ We are introducing the *wind-tunnel reference axis system* in addition to [DIN 9300-1] in order to discuss the (desired) motion in the wind-tunnel: the x_w-axis points opposite the wind velocity vector, z_w points down and y_w completes the set-up, too. As the origin is located at the geographic system's origin, the wind-tunnel reference system differs from the geographic only by rotation.

Flight-path axis system $[x_k, y_k, z_k]$ Used chiefly in guidance, the flight-path system is aligned to the motion *relative to ground*: the x_k-axis points along the flight-path velocity vector $\mathbf{V}_K$, the z_k-axis points down within the longitudinal plane, y_k again completes the set-up. The system's origin is located in the aircraft's centre-of-gravity.

Air-path axis system $[x_a, y_a, z_a]$ Analogue to the flight-path system but aligned to the motion *relative to air*: hence, the x_a-axis points along the aircraft velocity vector $\mathbf{V_A}$; z_a points down in the longitudinal plane, y_k completes the set-up, and the origin is in the centre-of-gravity, too.

Body-fixed axis system $\left[x_f, y_f, z_f\right]$ This last, rotating axis system is defined by the body's main axes: the x_f-axis points along the main fuselage towards the nose, the z_f-axis stands equidistant to the wing dihedrals pointing down, and the y_f completes the set-up pointing starboard.

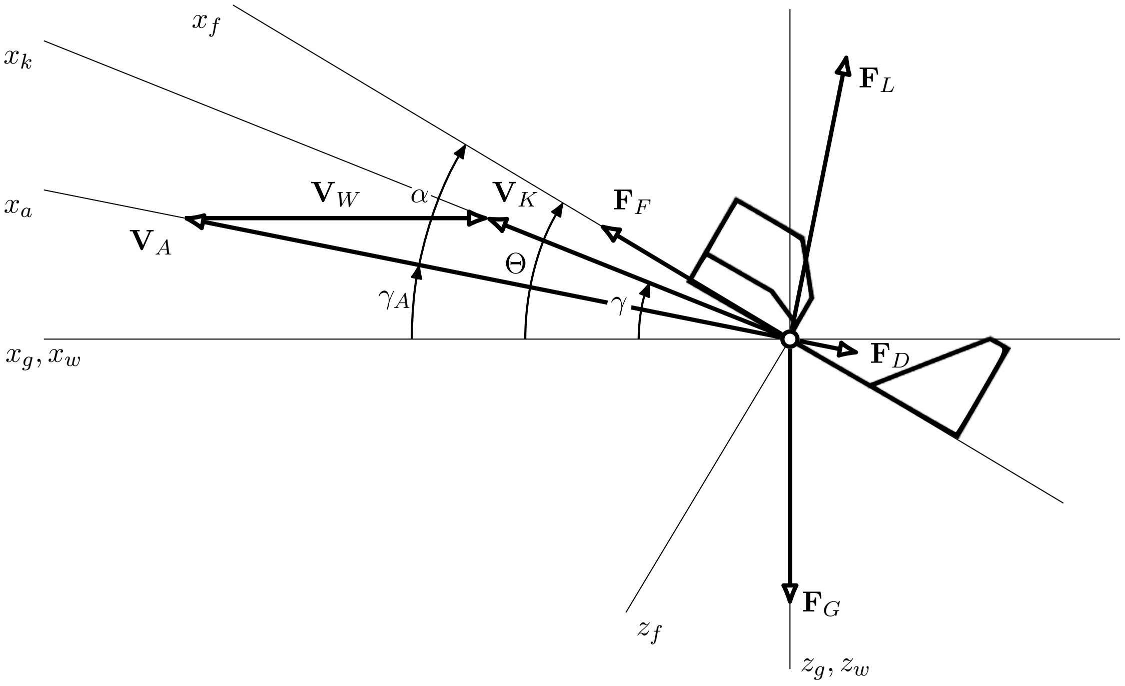


Figure 3.1.: Axes and descriptives of the longitudinal flight dynamics. *Based upon Brockhaus 2001 [1], Fig. 2.7.*

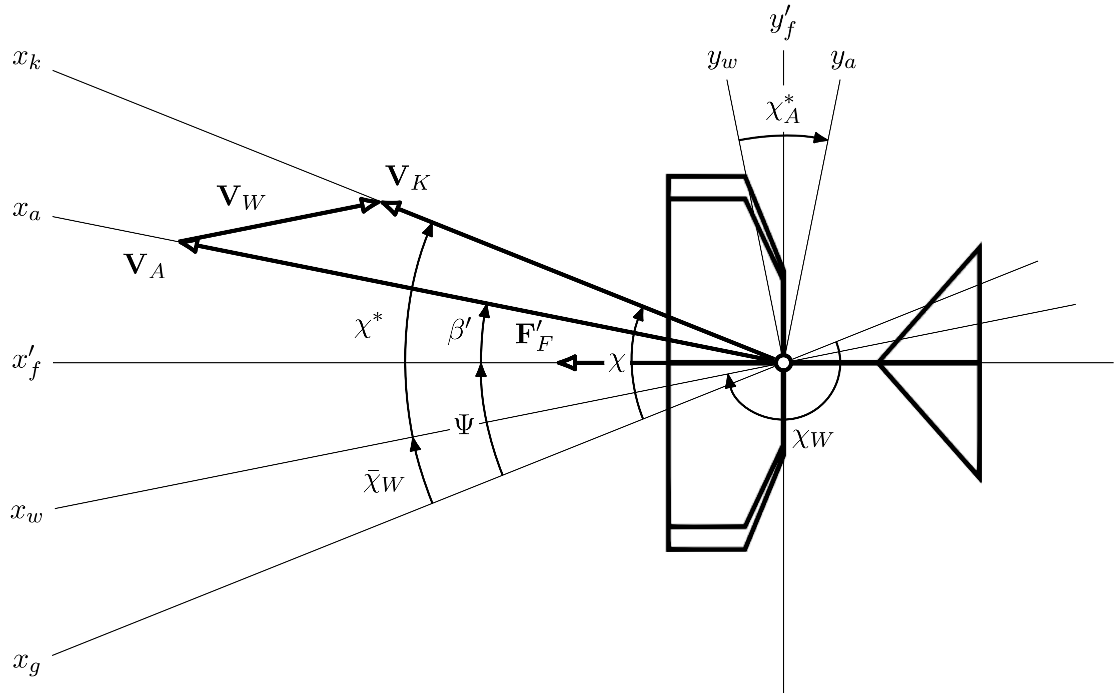


Figure 3.2.: Axes and descriptives of the horizontal flight dynamics. *Based upon Brockhaus 2001 [1], Fig. 2.8.* Projections of axes or quantities into the drawing layer are marked by $\odot'$.

3.2. Descriptives of the flight condition

While the aircraft's position is sufficiently expressed in coordinates of the axis systems, the orientation of the axis with respect to each other, and thus the attitude as well, are represented by different angles. We define those angles here, but first recall the relevant force and velocity vectors. For both force and velocity vectors, we denote the vector by $\mathbf{F}$ and $\mathbf{V}$, respectively, but the corresponding magnitude by F and V.

Velocities The important velocity vectors are given by the aircraft velocity $\mathbf{V}_A$ *relative to air*, the flight-path velocity $\mathbf{V}_K$ *relative to ground*, and the wind velocity $\mathbf{V}_W$. They complete to

$$\mathbf{V}_K = \mathbf{V}_A + \mathbf{V}_W \tag{3.1}$$

that is, the wind velocity is the difference of flight-path and aircraft velocity.

Forces Generated by the flapping wings itself, the thrust vector $\mathbf{F}_F$ points (approximately) along the body main axis x_f, towards the nose. As usual in flight dynamics, the lift vector $\mathbf{F}_L$ is orthogonal to the aircraft velocity vector, *i.e.* the air-path x_a-axis, and the body y_f-axis. The drag vector $\mathbf{F}_D$ opposites the aircraft velocity vector and the weight force vector $\mathbf{F}_G$ points down along the geographic z_g-axis.

Angles The orientation of the body axes with respect to the geographic axes are given by the *attitude*, that is roll angle Φ, pitch angle Θ, and heading angle Ψ. The orientation of the flight-path axis system to the geographic axes are represented by flight-path angle-of-climb γ and azimuth angle χ, respectively; similar, air-path angle-of-climb γ_A and azimuth angle χ_A give the orientation of the air-path axis system to the geographic axes. The orientation of the body with respect to the air-path axes are denoted by angle-of-attack α and side-slip angle β. Finally, the (horizontal) rotation of the wind-tunnel reference system is defined by the wind azimuth angle χ_W as well as its counter-part $\bar{\chi}_W = \chi_W - \pi$, and we note the flight-path and air-path azimuth angle with respect to the wind-tunnel reference system by χ^* and χ_A^*. In the same manner, the aircraft's heading in the wind-tunnel is labeled Ψ^*.

4. Control Theory Revisited

In this chapter, we review the basic definitions and approaches of control theory used for the control of the Delfly.

Basically, the aim of control theory is to design and apply a control law u in order to first either stabilize the (unstable) plant or improve the overall system's behaviour; second, to let the output y of the plant track a given target value y^c (either a constant set-point y^{SP} or time-varying reference y^{ref}) while negating any disturbances z. [86] The latter two control goals are called *reference tracking* and *disturbance rejection*.

The plant is usually represented by its transfer function, which is given by

$$G(s) = \frac{y(s)}{u(s)} \tag{4.1}$$

where $u(s)$ and $y(s)$ are the Laplace transform of plant input and output, respectively, in terms of the complex frequency $s = c + \jmath\omega$.

4.1. Representation of control systems

Pre-comment The terms *system* and *plant* are often used in alternating manner; hence, we introduce the following distinction: here and after, the plant describes the part *which is to be controlled*, while the system aggregates both plant and controller. Note that, if a system itself is controlled by a higher-level controller it becomes a plant in turn.

Various representations of plant and controller were developed in control theory in order to discuss different approaches and study the overall behaviour of their systems. Usually, the functional range of the resulting control is limited by the system representation chosen *ex ante*.

Feed-forward control The most simple representation and approach, feed-forward control, is shown by Fig. 4.1.

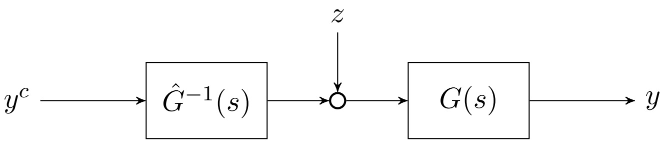


Figure 4.1.: Basic structure of a feed-forward control system.

Here, the controller, represented by its Laplace transformed $\hat{G}^{-1}$, has to invert the system behaviour of the plant G in order to achieve reference tracking. Easy to see, this can only be achieved if the plant behaviour is accurately known and no disturbances occur; *i.e.* $\hat{G}^{-1}(s) = G(s)^{-1}$ and $z(t) = 0$ for all frequencies s and times t.

Feedback control In order to react to disturbances and model uncertainties the plant output is fed back to the controller, resulting in the basic feedback control system of Fig. 4.2.

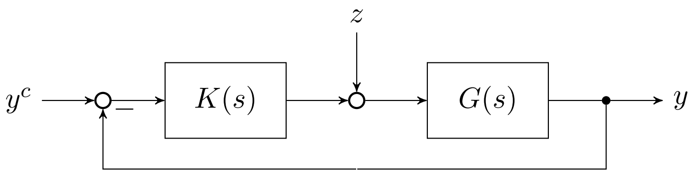


Figure 4.2.: Structure of a one-degree-of-freedom feedback control system.

Since the only input the controller deals with is the *control error* $\Delta y = y^c - y$, this control loop system is as well called *one degree-of-freedom control loop* in contrast to the system presented next.

Two-degree-of-freedom control By feeding the controller by both the commanded and the actual output value instead of their difference, the two degree-of-freedom control loop allows a more flexible controller design, while still covering the simple feedback control system aforementioned. Fig. 4.3 illustrates this system.

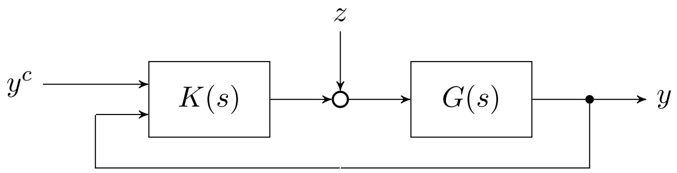


Figure 4.3.: Structure of a two-degree-of-freedom feedback control system.

State-space control Though rather a representation scheme of the plant itself than of the overall control loop, the state-space control system is basic system model affecting all three of plant model, control system structure, and control approach. The single-input single-output case of a state-space system with state feedback controller is given by Fig. 4.4.

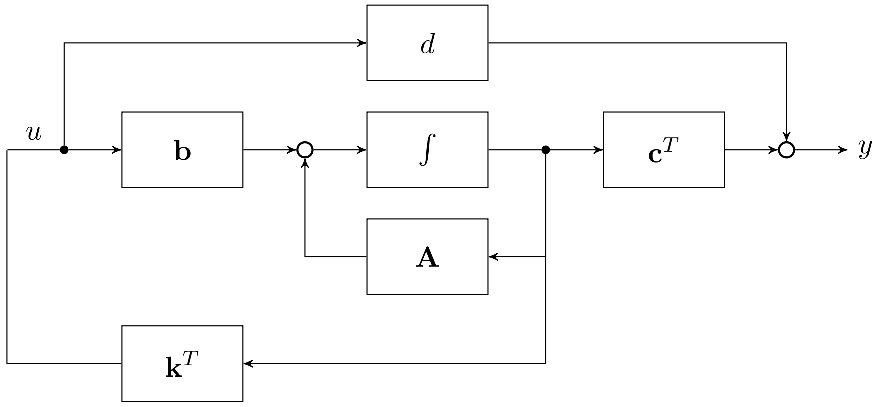


Figure 4.4.: State-space representation of a plant with state feedback control.

The state and output equations of the shown state-space model can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \tag{4.2}$$

$$y = \mathbf{c}^T\mathbf{x} + du \tag{4.3}$$

where $\mathbf{x}$ is the vector of states, u and y are the plant input and output, respectively, and $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}^T$, and d are the state-space matrices of appropriate dimensions. Then, the control law is given by

$$u = \mathbf{k}^T\mathbf{x} \tag{4.4}$$

where various approaches aim to derive the state-feedback matrix $\mathbf{k}^T$ (cf. Sec. 4.3). At this point, one will probably miss any command of the desired states values. In fact, while usually omitted, this can directly be obtained using the control law $u = \mathbf{k}^T(\mathbf{x}^c - \mathbf{x})$.

So far, we have only spoken about systems with one-dimensional input and output; however, all representations presented are simple to extend to the input, output, and disturbance vectors $\mathbf{u}$, $\mathbf{y}$, and $\mathbf{z}$, respectively, introducing matrix-valued transfer functions for plant and controller.

The system representations presented afore are not to be confused with the actual control approaches discussed later. Instead, they provide an overall scheme to design a controller within. This to see, one should consider that the very simple feed-forward control system can also be represented by two-degree-of-freedom controller, if the plant's output feedback is simply omitted by the control law $u_{\text{two}}(y^c, y) = \hat{G}^{-1}(s)\, y^c + 0 \cdot y$.

4.2. Static-gain control

If the control error and/or its derivatives or integrals are fed back to the plant via gain that is not changing over time, *i.e.* a static gain; therefore, this control approach is called *static-gain control*. Fig. 4.5 shows a P-controller, the most simple static-gain control approach, where the control error only is multiplied by the gain k.

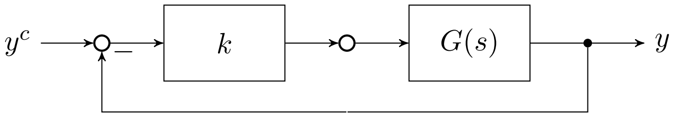


Figure 4.5.: Block diagram of a P-controller with static gain k.

The transfer function of the closed P-controlled system results to:

$$G_{yy^c}(s) = \frac{kG(s)}{1 + kG(s)} \tag{4.5}$$

As we can see from the closed-loop transfer function there is no steady-state error, *i.e.* $G_{yy^c}(t \to \infty) = 1$, if and only if there is integrating component in $G(s)$ and $G(s \to 0) \to \infty$; otherwise, it is $G(s \to 0) = g_\infty$, hence

$$G_{yy^c}(t \to \infty) = G_{yy^c}(s \to 0) = \frac{kg_\infty}{1 + kg_\infty} = \frac{1}{1 + (kg_\infty)^{-1}} \tag{4.6}$$

However, an integrator can be added artificially by feeding back the integrated control error; this is done for PI and PID control approaches. Fig. 4.6 thus shows a PID-controller, one of the control approaches mostly used in applied control theory. [**citation needed**]

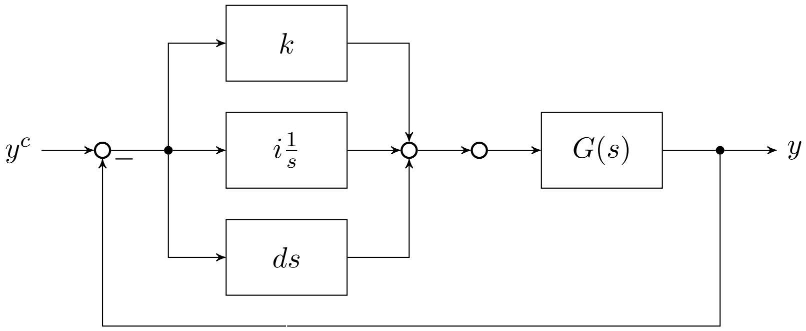


Figure 4.6.: Block diagram of a PID-controller with static gains k, i, and d.

Clearly, both performance and robustness of a PID-controlled system depend on well-chosen gains. Several tuning schemes have been proposed which are executed before hand either for a model of $G(s)$ or the real plant. [86, 87] Also, if the plant is non-linear, static gains derived for single point of linearisation may not be suitable for the full envelope.

4.3. State-space control

Based on the state-space system representation presented in Fig. 4.4, the state-feedback matrix $\mathbf{k}^T$ can derived using pole placement or linear-quadratic optimal control, to call just the two most important.

Pole placement By inserting Eq. 4.4 into Eq. 4.2 we can derive the closed-loop state equation to

$$\left(s\mathbf{I} - \mathbf{A} - \mathbf{k}^T\mathbf{b}\right)\mathbf{x} = 0 \tag{4.7}$$

where $\mathbf{I}$ is the identity matrix of suitable dimensions. Thus, the poles of the closed-loop system are given as roots of the matrix determinant

$$\left|s\mathbf{I} - \mathbf{A} - \mathbf{k}^T\mathbf{b}\right| = 0 \tag{4.8}$$

and $\mathbf{k}^T$ can be chosen in order to obtain the desired poles of the closed-loop.

Linear-quadratic optimal control Alternatively, the state-feedback matrix can be determined in an optimal way; namely, in order to minimize the quadratic cost function

$$J(\mathbf{k}) = \int_0^\infty \mathbf{x}^T(t)\,\mathbf{Q}\mathbf{x}(t) + Ru^2(t)\,\mathrm{d}t \tag{4.9}$$

where $\mathbf{Q}$ and R are (diagonal) weights.[(1)] A cost-optimal $\mathbf{k}_{\mathrm{opt}}$ is found in

$$\mathbf{k}_{\mathrm{opt}} = R^{-1}\mathbf{b}^T\mathbf{P} \tag{4.10}$$

where $\mathbf{P}$ solves the *algebraic Riccati equation*, [86]

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{b}R^{-1}\mathbf{b}^T\mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{4.11}$$

There are similar approaches minimizing varying cost functions, considering for example the weighted system output Ny^2 instead or in addition to the states. [88]

All state-space control approaches are commonly applying their feedback law to the system's states rather than output; thus, the values of states need to be known. If a state is not measurable or only with high noise in the signal, an *observer* can be employed; well known are the simple *Luenberger* observer or the optimal *Kalman filter*. [86]

[(1)] In systems of multiple input $\mathbf{u}$ a matrix-wise weight $\mathbf{Q}$ is used as well and the costs $\mathbf{u}^T\mathbf{R}\mathbf{u}$ are taken into account.

4.4. Adaptive control

In static-gain and state control, as discussed afore, the control law is designed, tuned, and optimised *before* the control application. However, this requires either an accurate model or a reproducible response of the plant. If this is not possible as the plant is unknown, not accessible beforehand, or needs to be controlled at very different conditions, these approaches do usually not yield a satisfying control result. Here, adaptive approaches have been proposed to cope with unknown factors: all in common, these approaches adapt their gains and/or structure to the actual, current plant behaviour.

Gain scheduling Originating from flight control at various Mach numbers and air pressures, [89] *gain scheduling* is basic approach, where gains of a non-adaptive control law are derived for selected environmental condition (e.g., the Mach number or the air pressure) and then changed in-control. That is, the system is controlled by a non-adaptive feedback law, while the control is adapted discretely and in feed-forward to the actual conditions.

Model reference adaptive systems Still rather simple, this approach is used when the structure of a plant's model is (roughly) known but not its parameters; then, the controller adapts itself to the plant's output in order to track a given model's behaviour. While there are different adaptation laws in literature, we will employ a model reference adaptive system in order to control the flapping-frequency of the Delfly in Sec. 7.2 using the so-called *MIT rule*.

Advanced adaptive control Further, more advanced adaptive approaches have been proposed, where the plant's behaviour is identified in-control and the control law is adapted based on the identification; e.g. by inversion of the plant in feed-forward [90]. The term *Dual adaptive control* has been introduced for approaches where the adaptation process is improved also during the application [91, 92].

*

4.5. Discussion

We presented afore basic static-gain and state control approaches as well as simple adaptive techniques. While an accurate model of the plant is required for state and state-optimal control, a static-gain control law can be designed and tuned to be sufficiently robust to model uncertainties and disturbances. However, a robust control outcome generally results into a less precise reference tracking [80].

In this thesis, we are interested in a control design for a plant without full-known model, the Delfly MAV, which should be both robust to disturbances and model uncertainties as well as precise in reference tracking. Obviously, this cannot be achieved by mere static-gain or state-optimal control approach. Thus, we will combine basic and adaptive techniques in order to control the Delfly.

5. Modelling and Control of the Delfly

For several reasons, position control of an aircraft flying freely in a wind-tunnel is hard in general; as well is control of the flapping-wing Delfly in particular:

First and in general, though based on the observer's point of view the position of the Delfly is controlled, on-board the story is different; actually, the velocity *relative to air* has to be adjusted in order to negate the facing wind-speed. The controller, however, must still maintain precise reference tracking of the aircraft's position *relative to ground.* Due to the OptiTrack tracking system we are rather able to measure the position and velocity relative to ground accurately than relative to air; that is, a proficient estimation of the wind velocity and velocity relative to air, respectively, will be necessary.

Second, the aero-dynamic equations of motion of the Delfly are both barely known and only derived for a certain flight condition. On the other hand, the flight condition is changing chiefly over the range of envisaged wind speeds. The pitch angle especially is varying, starting at nearly $90^{\circ\prime\prime\prime}$ at almost-hovering up to small angles at fast-forward flight, causing furthermore the difficulties discussed next. Thus, a control approach suitable over the whole flight envelope either is sufficiently robust,designed for different flight conditions, or able to adapt to a changing environment.

The generation of lift and thrust, third, through the flapping-wing propulsion is highly coupled: while for almost-hovering flight the lift is generated directly by the thrust pointing upwards, in forward flight the Delfly pitches down and additional lift is generated by the wings due to gliding effects. In contrast to quadrocopter control, where forward acceleration is achieved by varying the pitch angle and the thrust is to compensate the reduced lift, the control approach for the Delfly should consider the coupling effects as well. Fig. 5.1 shows the relation of lift and thrust for the Delfly.

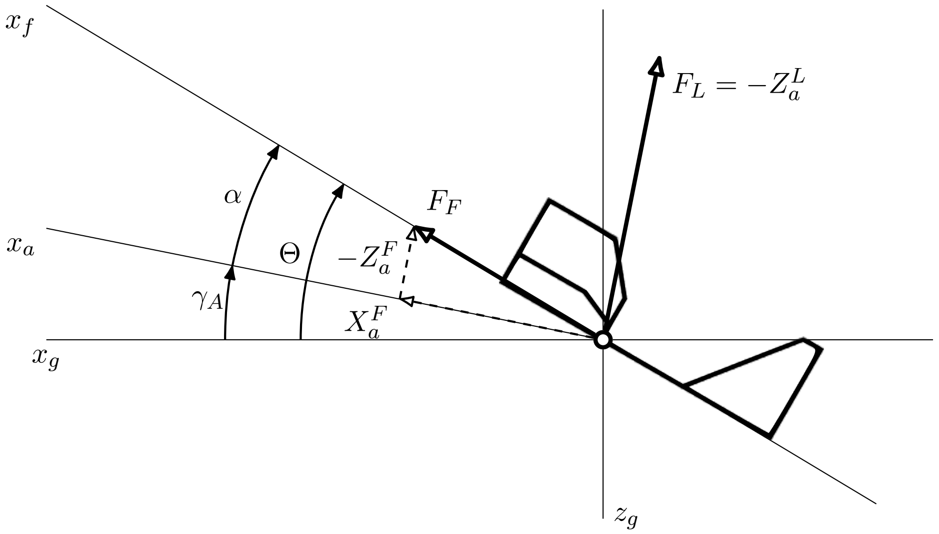


Figure 5.1.: The relation of lift and thrust generation of the Delfly: in forward flight, lift is generated both by gliding effects of the wings ($-Z_a^L$) and flapping-wing propulsion ($-Z_a^F$), while forward acceleration is due to thrust only (X_a^F).

5.1. Hierarchic control scheme

In order to tackle all three challenges mentioned, the hierarchic control scheme in Fig. 5.2 is proposed. The different levels of this control cascade shall be introduced briefly here and discussed in detail within the subsequent chapters.

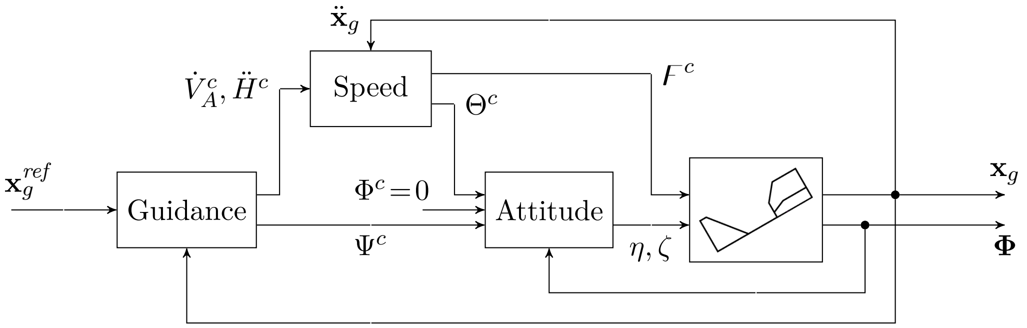


Figure 5.2.: The hierarchic control scheme for the Delfly control approach.

In contrast to the classical flight-control hierarchy, one first and foremost notices the *speed-thrust control* block additionally inserted between attitude and guidance control. From the inner loop to the most outer, these blocks are:

Attitude control The orientation of the vehicle (pitch angle; bank angle; heading) is controlled using elevator and rudder. The uncontrolled Delfly is in open-loop stable with respect to pitch and roll; thus, the task of attitude control is mainly to track the desired orientation without de-stabilizing the system. Note that for absence of ailerons, both roll and yaw rate are affected by the rudder; however, only the heading will be controlled. Finally, though the equations of motions are only captured for slow, near-hover forward flight, the attitude control laws derived has to be both precise and robust through changing flight conditions.

Speed-thrust control As aforementioned, at non-hovering flight additional lift is generated by the wings for angle-of-attacks ere stall; therewhile, thrust results into lift decreasingly from hover via forward flight to non-at-all at fast-forward flight. Equally, forward acceleration is achieved due to the thrust vector the more the pitch angle decreases. In order to track the forward and height acceleration commanded by guidance control, the coupling dynamics of pitch angle and throttle has to be faced. Measuring static forces at different wind-speeds, the relation of lift and drag depending on pitch angle and flapping frequency has been figured out by [93]; however, this is not considered to be a reliable model fully representing the dynamics, for the data has been obtained statically and the measurements are rather discrete (*cf.* Section 5.3).

Horizontal and vertical guidance The guidance's task is as usual: given a trajectory of waypoints – or, in the wind-tunnel, a single position relative to ground – the vehicle shall track these reference as accurate as possible. In order to increase performance, the velocity, *i.e.* magnitude and direction of the wind vector, shall be taken into account; that means, guidance rather refers to the trajectory relative to air than to ground. Therefore, naturally, the wind velocity vector has to be measured and estimated, respectively. Besides, the performance of guidance relies heavily on the performance of the underlying speed-thrust and attitude control.

5.2. Linear equations of motion

By cycle-averaging of real flight data, the linear equations of motion of the Delfly have been derived by [60] for a single linearization point. As usual in flight dynamics, the linear equations are separated into the longitudinal (Eq. 5.1) and lateral (Eq. 5.2) dynamics.

$$\begin{bmatrix} \dot{q} \\ \dot{u} \\ \dot{w} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{M_q}{I_{yy}} & \frac{M_u}{I_{yy}} & \frac{M_w}{I_{yy}} & 0 \\ \frac{X_q}{m} - w_0 & \frac{X_u}{m} & \frac{X_w}{m} & -g\cos\Theta_0 \\ \frac{Z_q}{m} - u_0 & \frac{Z_u}{m} & \frac{Z_w}{m} & -g\sin\Theta_0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta q \\ \delta u \\ \delta w \\ \delta\Omega \end{bmatrix} + \begin{bmatrix} \frac{M_\eta}{I_{yy}} \\ \frac{X_\eta}{m} \\ \frac{Z_\eta}{m} \\ 0 \end{bmatrix} \delta\eta \tag{5.1}$$

Longitudinal equations of motion.

The longitudinal and lateral equations of motion as presented are valid for small deviations ($\delta\mathfrak{x}$) around the point of linearization, denoted by Θ_0, u_0, w_0. The parameter of the equations has been estimated using the method of *maximum likelihood.* [60]

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{v} \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} \frac{I_z}{I_c}L_p + \frac{I_{xz}}{I_c}N_p & \frac{I_z}{I_c}L_r + \frac{I_{xz}}{I_c}N_r & \frac{I_z}{I_c}L_v + \frac{I_{xz}}{I_c}N_v & 0 \\ \frac{I_{xz}}{I_c}L_p + \frac{I_x}{I_c}N_p & \frac{I_{xz}}{I_c}L_r + \frac{I_x}{I_c}N_r & \frac{I_{xz}}{I_c}L_v + \frac{I_x}{I_c}N_v & 0 \\ \frac{Y_p}{m} + w_0 & \frac{Y_r}{m} + u_0 & \frac{Y_v}{m} & g\cos\Theta_0 \\ 1 & \tan\Theta_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta r \\ \delta v \\ \delta\Phi \end{bmatrix} + \begin{bmatrix} \frac{I_z}{I_c}L_\zeta + \frac{I_{xz}}{I_c}N_\zeta \\ \frac{I_{xz}}{I_c}L_\zeta + \frac{I_x}{I_c}N_\zeta \\ \frac{Y_\zeta}{m} \\ 0 \end{bmatrix} \delta\zeta \tag{5.2}$$

Lateral equations of motion.

5.3. Lift and drag forces

Beforehand, the generation of aerodynamic forces, *i.e.* lift and drag, has been measured statically by [93]: tethered, the Delfly has been “flying” at different pitch angles and throttle commands as well as at increasing wind velocities.

Given the mass m and a wind velocity V_W, we define the Delfly to be in an *equilibrium condition* for a pitch angle Θ_0, throttle command $\digamma_0$ if and only if both

$$F_F(\Theta_0, \digamma_0, V_W) = F_D(\Theta_0, \digamma_0, V_W) \tag{5.3}$$

$$F_L(\Theta_0, \digamma_0, V_W) = F_G = m * g \tag{5.4}$$

that is, the generated lift is compensating the Delfly’s weight and the drag force is just compensated by thrust. We will denote the forces at equilibria by F_{F0}, F_{D0}, and F_{L0}. For the wind velocities covered by [93], the equilibria conditions are given by Tab. 5.1a.

V_W	Θ_0	$\digamma_0$
$0.4\,\frac{\mathrm{m}}{\mathrm{s}}$	$74.40°$	$90.33\,\%$
$0.8\,\frac{\mathrm{m}}{\mathrm{s}}$	$65.85°$	$86.83\,\%$
$1.2\,\frac{\mathrm{m}}{\mathrm{s}}$	$47.23°$	$78.00\,\%$
$2.5\,\frac{\mathrm{m}}{\mathrm{s}}$	$30.51°$	$68.48\,\%$
$5.0\,\frac{\mathrm{m}}{\mathrm{s}}$	$11.90°$	$71.39\,\%$

(a) Equilibria conditions.

V_W	$F_{L\Theta}$	$F_{L\digamma}$	$F_{F\Theta}$	$F_{F\digamma}$
$0.4\,\frac{\mathrm{m}}{\mathrm{s}}$	$-0.1\,\mathrm{mN}/1^\circ$	$4.0\,\mathrm{mN}/1\,\%$	$-4.0\,\mathrm{mN}/1^\circ$	$1.0\,\mathrm{mN}/1^\circ$
$0.8\,\frac{\mathrm{m}}{\mathrm{s}}$	$0.8\,\mathrm{mN}/1^\circ$	$3.7\,\mathrm{mN}/1\,\%$	$-5.2\,\mathrm{mN}/1^\circ$	$1.4\,\mathrm{mN}/1^\circ$
$1.2\,\frac{\mathrm{m}}{\mathrm{s}}$	$0.8\,\mathrm{mN}/1^\circ$	$3.4\,\mathrm{mN}/1\,\%$	$-2.8\,\mathrm{mN}/1^\circ$	$2.4\,\mathrm{mN}/1^\circ$
$2.5\,\frac{\mathrm{m}}{\mathrm{s}}$	$4.9\,\mathrm{mN}/1^\circ$	$3.2\,\mathrm{mN}/1\,\%$	$-5.5\,\mathrm{mN}/1^\circ$	$2.2\,\mathrm{mN}/1^\circ$
$5.0\,\frac{\mathrm{m}}{\mathrm{s}}$	$19.3\,\mathrm{mN}/1^\circ$	$1.2\,\mathrm{mN}/1\,\%$	$-3.5\,\mathrm{mN}/1^\circ$	$1.6\,\mathrm{mN}/1^\circ$

(b) Aerodynamic force derivatives.

Table 5.1.: The equilibria conditions and aerodynamic force derivatives of the Delfly for different wind velocities based on [93].

For the sake of later control design, we linearise lift and drag force, respectively, for each wind velocity as functions of pitch angle Θ and throttle command $\digamma$. Using a first-order Taylor approximation at the respective equilibrium condition, we obtain ($\wp \in \{F, L\}$)

$$F_\wp(V_W) \approx F_\wp(\Theta_0, \digamma_0, V_W) + \frac{\partial}{\partial \Theta} F_\wp(\Theta, \digamma, V_W)\bigg|_{\Theta_0, \digamma_0} \partial\Theta + \frac{\partial}{\partial \digamma} F_\wp(\Theta, \digamma, V_W)\bigg|_{\Theta_0, \digamma_0} \partial\digamma \tag{5.5}$$

where $F_\wp(\Theta_0, \digamma_0, V_W)$ constitutes just the forces at equilibrium, *i.e.* $F_{F0} = F_{D0}$ and $F_{L0} = F_G$.

Substituting $\partial F_\wp$, $\partial\Theta$, and $\partial\digamma$ by $\Delta F_\wp$, $\Delta\Theta$, and $\Delta\digamma$, respectively, and referring to the longitudinal aerodynamic force vector, we get

$$\begin{bmatrix} \Delta F_F \\ \Delta F_L \end{bmatrix} = \mathbf{F}_{\AE}(V_W) \begin{bmatrix} \Delta\Theta \\ \Delta\digamma \end{bmatrix} \tag{5.6}$$

where $\mathbf{F}_{\mathit{\AE}}$ denotes the derivation of lift and drag forces with respect to pitch angle and throttle command,

$$\mathbf{F}_{\mathit{\AE}} = \begin{bmatrix} F_{F\Theta} & F_{F\varGamma} \\ F_{L\Theta} & F_{L\varGamma} \end{bmatrix} = \begin{bmatrix} \frac{\Delta}{\Delta\Theta} & \frac{\Delta}{\Delta\varGamma} \end{bmatrix} \begin{bmatrix} F_F \\ F_L \end{bmatrix} \tag{5.7}$$

which we will call the *aerodynamic force derivatives matrix*; its components are shown by Tab. 5.1b.

6. Guidance

While the control of attitude and speed-thrust, as afore discussed in the previous chapters, can be counted to be part of the basic control of an aircraft, the guidance control commands suitable control inputs to the first in order to achieve a long-term stable and reliable tracking of a given trajectory. Hence, its performance is both necessary for successful operation as well as dependent of the underlying control parts. As usual in fixed-wing and rotorcraft flight control, we separate the discussion of horizontal and vertical guidance; the coupling effects in between are taken into account by the speed-thrust control approach.

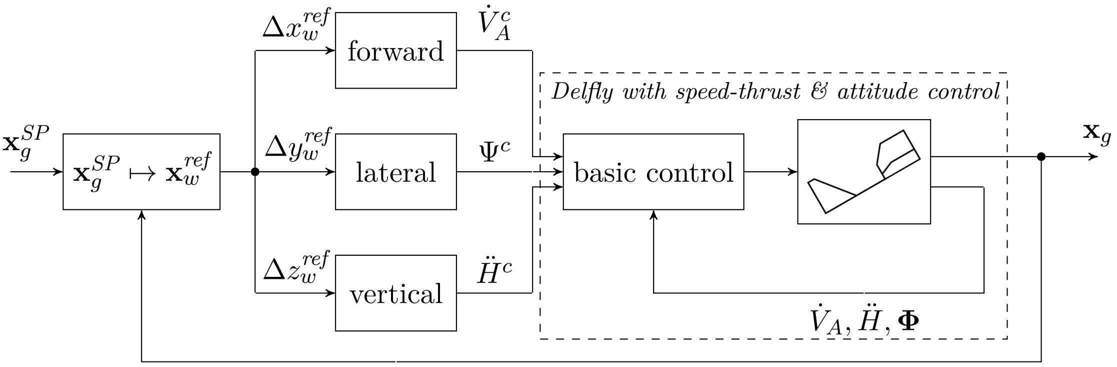


Figure 6.1.: The guidance control design within the Delfly control approach.

As acceleration and heading are controlled by the underlying speed-thrust and attitude controller, respectively, those can be commanded by the guidance control in order to achieve a desired approach to the position set-point. Fig. 6.1 outlines the control strategy as well as this chapter: the set-point $\mathbf{x}_g^{SP}$ given in normal earth-fixed coordinates is first transformed to get wind-tunnel reference coordinates; from the control error in position and velocity, respectively, the acceleration and heading commands are calculated – the approach used, non-linear dynamic control, is briefly present in Sec. 6.1 and applied in Sec. 6.2; acceleration and heading commands are then given as set-point to the basic control loops. Eventually, we argue that the guidance control can even handle non-nominal behaviour of the basic-controlled Delfly.

*

Horizontal guidance Basically, horizontal guidance control is responsible for the position and motion in the normal earth-fixed horizontal plane, *i.e.* the x_g-y_g-plane. Flying in the wind-tunnel, the direction of the aircraft velocity $\mathbf{V}_A$ is (in steady-state) pointing opposite the wind velocity vector $\mathbf{V}_W$, that is we have $\mathbf{V}_A = -\mathbf{V}_W$; thus there is no motion relative to ground and the flight-path velocity $\mathbf{V}_K = \mathbf{V}_A + \mathbf{V}_W = \mathbf{0}$. In order to discuss guidance control with respect to the desired air-path, we introduce the new, non-rotating[(1)] wind-tunnel reference axis system $[x_w, y_w, z_w]$, similar to the normal earth-fixed: the x_w-axis opposites the wind velocity vector; z_w points down; y_w complements the set-up. The wind-tunnel reference axis system is implementing an earth-fixed axis system according to [DIN 9300-1].

In [DIN 9300-1] the wind direction is called *wind azimuth* and denoted by χ_W. In contrast, we note the angle between normal earth-fixed x_g-axis and wind-tunnel reference x_w-axis, clockwise positive, by $\bar{\chi}_W = \chi_W - \pi$. From here, we can state the azimuth relative to wind-tunnel reference and note the difference by χ^*. The flight-path and air-path azimuth finally result into

$$\chi = \bar{\chi}_W + \chi^* \tag{6.1}$$

$$\chi_A = \bar{\chi}_W + \chi_A^* \tag{6.2}$$

We will discuss independently the control of position in opposite direction of wind (forward, x_w-axis) and orthogonal (lateral, y_w). Thus we neglect any coupling of forward and lateral motion – clearly, there is some –, since they are marginal in the steady-state wind-tunnel flight.

Vertical guidance Just as its name suggests, vertical guidance control takes care of the vertical, *i.e.* z_g-, axis of the geographic reference frame. In flight control it is convenient to define the *height* above a certain ground reference, $H = z_{g0} - z_g$,[(2)] and refer to the height rather than to the z_g-axis in vertical guidance. For the motion in the vertical axis is not depending neither on direction nor on speed of the wind, we directly set

$$z_w = z_g \tag{6.3}$$

and have $H = z_{g0} - z_g = z_{w0} - z_w$.

[(1)] If the wind-direction is constant.

[(2)] Note that, while the geographic z_g-axis is defined pointing *towards* ground, height is positive above ground level.

6.1. Non-linear inversion control

In their paper in 2010, Sieberling, Chu, and Mulder described the idea of non-linear dynamic inversion as "*to find a direct relation between the desired output and the input and invert it.*" [94]

Given the non-lineary dynamics of a system,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\,\mathbf{u} \tag{6.4}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{6.5}$$

they find this relation using multiple order *Lie derivation* and under certain conditions to be [94]

$$\mathbf{y}^{(n)} = L_f^n \mathbf{h}(\mathbf{x}) + L_g L_f^{n-1} \mathbf{h}(\mathbf{x})\,\mathbf{u} \tag{6.6}$$

and invert this to obtain

$$\mathbf{u} = \Big(L_g L_f^{n-1} \mathbf{h}(\mathbf{x})\Big)^{-1} \Big(\boldsymbol{\nu} - L_f^n \mathbf{h}(\mathbf{x})\Big) \tag{6.7}$$

where the n-th derivative of the desired output $\mathbf{y}^{(n)}$ is replaced by the so-called pseudo-input $\boldsymbol{\nu}$. If one calculates the pseudo-input as linear combination of the desired output and its derivatives,

$$\boldsymbol{\nu} = \boldsymbol{\lambda}\Big(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(n-1)}\Big) \tag{6.8}$$

the closed-loop system of Eq. 6.6 and 6.7 results into

$$\mathbf{y}^{(n)} = \boldsymbol{\nu} = \boldsymbol{\lambda}\Big(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(n-1)}\Big) \tag{6.9}$$

Thus we get the closed-loop system in time and frequency domain:

$$\mathbf{y}^{(n)} - \lambda_{n-1}\mathbf{y}^{(n-1)} - \cdots - \lambda_1 \dot{\mathbf{y}} - \lambda_0 \mathbf{y} = 0 \tag{6.10}$$

$$\Big(s^n - \lambda_{n-1} s^{n-1} - \cdots - \lambda_1 s - \lambda_0\Big)\,\mathbf{y}(s) = 0 \tag{6.11}$$

and the system behaviour can be determined by the coefficients $\lambda_0, \ldots, \lambda_{n-1}$ of the characteristic polynomial Eq. 6.11.

Apparently, non-linear dynamic inversion control enjoys the benefits of directly injecting the desired closed-loop system behaviour (*i.e.* the coefficients of the characteristic polynomials) via the control law without inevitably compensating the poles of the controlled plant, as in classical control. On the other hand, an accurate model of the plant dynamic is obviously required.

6.2. Forward, vertical, and lateral guidance

In the nominal case, the Delfly is flying steadily, *i.e.* along the x_w-axis of the wind-tunnel reference axis system, and the underlying basic and speed-thrust control is ideal in closed-loop; that is

$$\dot{x}_w = V_A - V_W \tag{6.12}$$

$$\dot{z}_w = -\dot{H} \tag{6.13}$$

and

$$\dot{V}_A = \dot{V}_A^c \tag{6.14}$$

$$\ddot{H} = \ddot{H}^c \tag{6.15}$$

$$\Psi = \Psi^c \tag{6.16}$$

From derivation, and since the wind-speed is constant ($\dot{V}_W = 0$), we get

$$\ddot{x}_w = \dot{V}_A^c \tag{6.17}$$

$$\ddot{z}_w = -\ddot{H}^c \tag{6.18}$$

Hence, the inversion in x_w- and z_w-axes yields

$$\dot{V}_A^c = \nu_x \tag{6.19}$$

$$\ddot{H}^c = -\nu_z \tag{6.20}$$

and proposing the pseudo-input control law

$$\nu_x = d_{\dot{V}_A x_w} \Delta\dot{x}_w + k_{\dot{V}_A x_w} \Delta x_w = d_{\dot{V}_A x_w}\left(\dot{x}_w^{SP} - \dot{x}_w\right) + k_{\dot{V}_A x_w}\left(x_w^{SP} - x_w\right) \tag{6.21}$$

$$\nu_z = d_{\ddot{H} z_w} \Delta\dot{z}_w + k_{\ddot{H} z_w} \Delta z_w = d_{\ddot{H} z_w}\left(\dot{z}_w^{SP} - \dot{z}_w\right) + k_{\ddot{H} z_w}\left(z_w^{SP} - z_w\right) \tag{6.22}$$

the closed-loop guidance yields a second-order system with coefficients k and d.

While the forward and vertical guidance can be designed in this manner, the lateral guidance needs a slightly different approach, which we discuss separately.

*

6.2.1. Forward guidance

From the proposed control law (Eq. 6.19 and 6.21),

$$\dot{V}_A^c = d_{\dot{V}_A x_w} \left(\dot{x}_w^{SP} - \dot{x}_w \right) + k_{\dot{V}_A x_w} \left(x_w^{SP} - x_w \right) \tag{6.23}$$

where the velocity set-point $\dot{x}_w^{SP}$ is zero (no motion of the Delfly relative to ground) and the position set-point x_w^{SP} can be assumed to be zero, too. The closed-loop forward motion then results to (Eq. 6.17 and 6.23)

$$\ddot{x}_w = d_{\dot{V}_A x_w} \left(-\dot{x}_w \right) + k_{\dot{V}_A x_w} \left(-x_w \right) \tag{6.24}$$

which is, re-written and in frequency-domain, equivalent to

$$\begin{aligned} &\left(s^2 + d_{\dot{V}_A x_w} s + k_{\dot{V}_A x_w} \right) x_w(s) = \\ &\qquad (s - \varsigma_{x1}) (s - \varsigma_{x2}) \, x_w(s) = 0 \end{aligned} \tag{6.25}$$

where, if $\varsigma_{x1,2}$ are the (desired) poles of the closed-loop system, $d_{\dot{V}_A x_w} = -\varsigma_{x1} - \varsigma_{x2}$ and $k_{\dot{V}_A x_w} = \varsigma_{x1}\varsigma_{x2}$.

6.2.2. Vertical guidance

Applying Eq. 6.18, 6.20, and 6.22, we obtain for the closed-loop vertical motion:

$$\ddot{z}_w = -\ddot{H}^c = \nu_z = d_{\ddot{H} z_w} \left(\dot{z}_w^{SP} - \dot{z}_w \right) + k_{\ddot{H} z_w} \left(z_w^{SP} - z_w \right) \tag{6.26}$$

which simplifies with $z_w^{SP} = \dot{z}_w^{SP} = 0$ (cf. forward guidance) to

$$\ddot{z}_w + d_{\ddot{H} z_w} \dot{z}_w + k_{\ddot{H} z_w} z_w = 0 \tag{6.27}$$

and in frequency-domain,

$$\begin{aligned} &\left(s^2 + d_{\ddot{H} z_w} s + k_{\ddot{H} z_w} \right) z_w(s) = \\ &\qquad (s - \varsigma_{z1}) (s - \varsigma_{z2}) \, z_w(s) = 0 \end{aligned} \tag{6.28}$$

That is, $\varsigma_{z1,2}$ are the poles of the closed-loop system and the gains $d_{\ddot{H} z_w}$, $k_{\ddot{H} z_w}$ can be calculated by $d_{\ddot{H} z_w} = -\varsigma_{z1} - \varsigma_{z2}$ and $k_{\ddot{H} z_w} = \varsigma_{z1}\varsigma_{z2}$ for desired poles.

6.2.3. Lateral guidance

While the thrust force vector $\mathbf{F}_F$ is pointing towards the body x_f-axis, the drag force $\mathbf{F}_D$ is applied opposing the air-speed V_A, *i.e.* against the air-path x_a-axis. As long as in steady flight and zero side-slip, thrust is compensating drag both in angle and magnitude and there is no motion lateral to the body x_f-axis. If, however, the angle of thrust (heading Ψ^*) deviates from the angle of air-speed (azimuth χ_A^*) by yawing – implying a side-slip angle $\beta = \chi_A^* - \Psi^*$ –, we have for the lateral motion:

$$\begin{aligned} m\ddot{y}_w &= F_F \sin(\Psi - \bar{\chi}_W) - F_D \sin(\chi_A - \bar{\chi}_W) \\ &= F_F \sin\big(\Psi^*\big) - F_D \sin\big(\chi_A^*\big) \end{aligned} \tag{6.29}$$

where Ψ^*, χ_A^* are the heading and air-path azimuth in the wind-tunnel reference frame. Since for appropriate wind-velocities V_W the air-path azimuth is comparatively small in contrast to heading and flight-path azimuth, we can neglect the drag term in near-steady flight and Eq. 6.29 simplifies to

$$\ddot{y}_w = m^{-1} F_F \sin\big(\Psi^*\big) \tag{6.30}$$

which we will use for the lateral guidance design.

Proposing the control law

$$\Psi^{*c} = d_{\Psi y_w}\left(\dot{y}_w^{SP} - \dot{y}_w\right) + k_{\Psi y_w}\left(y_w^{SP} - y_w\right) \tag{6.31}$$

thus we get the closed-loop lateral motion for $\dot{y}_w^{SP} = 0$ (no lateral motion desired) and $y_w^{SP} = 0$ (no lateral offset desired)

$$\ddot{y}_w = m^{-1} F_F \sin\big(-d_{\Psi y_w}\dot{y}_w - k_{\Psi y_w} y_w\big) \tag{6.32}$$

which can be simplified for small angles Ψ^* to

$$\ddot{y}_w = -m^{-1} F_F \left(d_{\Psi y_w}\dot{y}_w + k_{\Psi y_w} y_w\right) \tag{6.33}$$

resulting, in frequency-domain, to

$$\begin{aligned} \left(s^2 + m^{-1} F_F d_{\Psi y_w} s + m^{-1} F_F k_{\Psi y_w}\right) y_w(s) &= \\ \left(s - \varsigma_{y1}\right)\left(s - \varsigma_{y2}\right) y_w(s) &= 0 \end{aligned} \tag{6.34}$$

where, in contrast to the forward and vertical guidance, the poles $\varsigma_{y1,2}$ are dependent on the thrust force and mass, too.

Lateral integral gain In order to compensate a steady-state error we encounter during the wind-tunnel flight tests, we will expand the control law of Eq. 6.31 by an additional integral term, *i.e.*

$$\Psi^{*c} = d_{\Psi y_w}\left(\dot{y}_w^{SP} - \dot{y}_w\right) + k_{\Psi y_w}\left(y_w^{SP} - y_w\right) + i_{\Psi y_w}\int\left(y_w^{SP} - y_w\right)\mathrm{d}t \tag{6.35}$$

where $i_{\Psi y_w}$ is a suitable integral gain.

7. Speed-Thrust Control

Controlling the behaviour in the longitudinal plane to achieve the requested movement, is the core task of the Delfly control approach. While the generation of lift and thrust is highly coupled, the speed-thrust control law is asked to command a suitable pitch angle and flapping frequency. In this chapter, we therefore propose an overall control strategy and discuss different approaches to fulfil the task. To keep things simple first, we consider flapping frequency and throttle command to be interchangeable, as there is a quasi-proportional relation between them.[1]

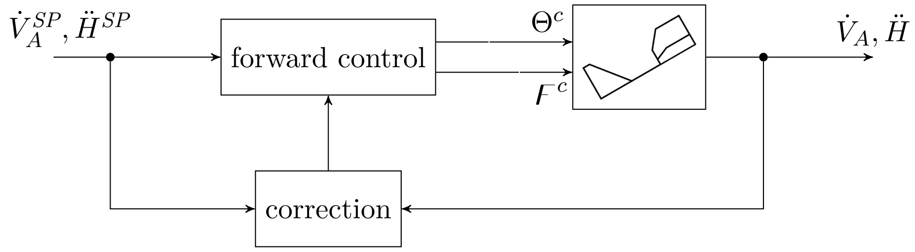


Figure 7.1.: The overall control strategy for speed-thrust control.

In the longitudinal x_a-z_a-plane, the motion is determined by the (horizontal-forward component of the) air-speed V_A and the change in height $\dot{H}$. For the speed-thrust controller, we make the following assumptions about the flight condition of the Delfly:

- the air-speed vector lays in the horizontal plane;
- the thrust vector lays in the longitudinal plane.

These assumptions are fully satisfied in steady-state flight in the wind-tunnel, and sufficiently met for slight deviations from the set-point. Thus, a couple of simplifications hold:

1. The lift vector F_L is orthogonal to the horizontal plane, and $F_L - F_G = m * \ddot{H}$. That is, pitch angle and angle-of-attack are remaining the same.

[1] In Section 7.2, we discuss this relation and propose a flapping-frequency control approach.

2. Both horizontal acceleration and air-speed are aligned, *i.e. there is no side-slip angle*; Hence, the thrust leads “directly” to a change of the air-speed and $F_F - F_D = m * \dot{V}_A$.

We can control lift and thrust of the Delfly via pitch angle and throttle – we have discussed the coupled relation in Chapter 5 and Section 5.3, respectively –, to follow the horizontal and vertical accelerations set-points from the higher-level guidance control loop. In order to design and develop a control approach, we propose the overall control strategy shown by Fig. 7.1: pitch angle and throttle are commanded by a forward control block from the forward and vertical acceleration set-points, while adjusted by the feed-back loop.

This control scheme, which resembles technically a two-degree-of-freedom controller, allows various control laws obtained by different approaches: a naïve solution, for example, would be a static-gain controller in feed-back loop without any forward control. However, in combination with an *ex ante* plant model we get a *Combined forward-backward controller*, which is a common approach in process control engineering [95]. On the other hand, the proposed scheme is flexible enough to realise advanced adaptive control approaches such as *Adaptive inverse control* [90] or *Adaptive dual control* [91], where an initial or *a priori* model estimation is adjusted over time.

7.1. Combined forward-backward control

A feedback-only controller with static-gains tends to respond heavily to a changing input reference, since it is fed directly by the difference of set-point and plant output. While this reaction is reasonable if disturbance rejection is the main task and the reference input is constant, it becomes less admissible as inner part of a hierarchic control approach where the input reference changes presumably in each control loop execution: here, the controller cannot distinguish an error due to deviating plant output and from errors due to varying control input. The control commands, however, can be improved in forward loop, taking into account information (or educated guesses) about the plant behaviour.

The combined forward-backward control approach is based on the lift and drag model from Section 5.3, where, for different wind velocities, the pitch angle Θ_0 and throttle command F_0 in equilibrium condition, as well as the aerodynamic derivatives matrix $\mathbf{F}_{Æ}$ were obtained. Applying first-order Taylor linearization, we have got:

$$\begin{bmatrix} F_F \\ F_L \end{bmatrix} = \begin{bmatrix} F_F(\Theta_0, \mathit{\Gamma}_0, V_W) \\ F_L(\Theta_0, \mathit{\Gamma}_0, V_W) \end{bmatrix} + \mathbf{F}_{Æ}(V_W) \begin{bmatrix} \Delta\Theta \\ \Delta\mathit{\Gamma} \end{bmatrix} \tag{7.1}$$

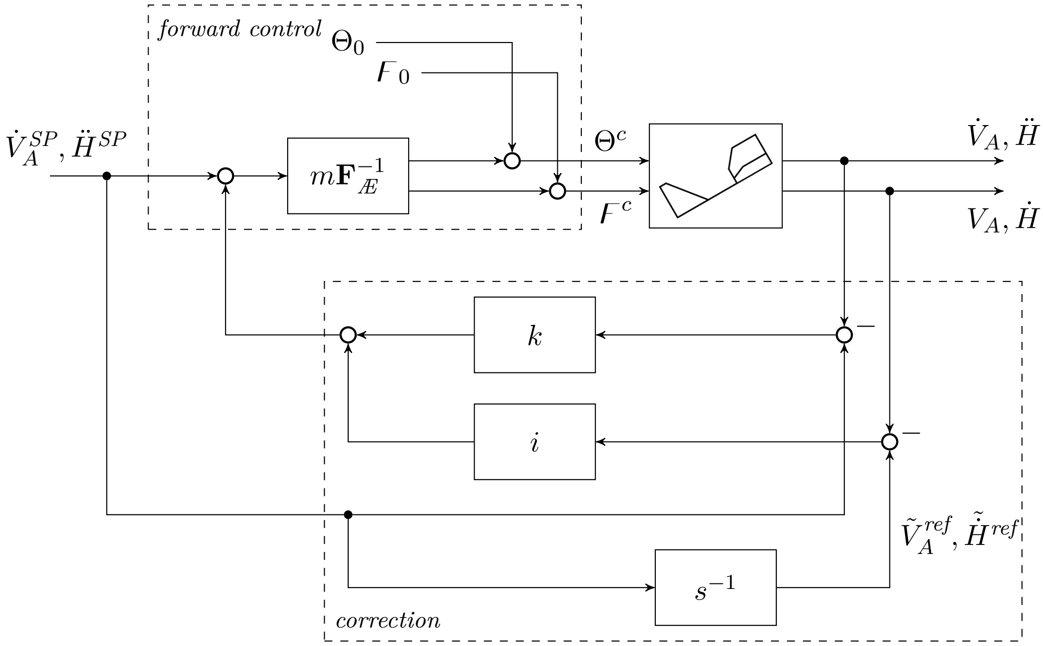


Figure 7.2.: The combined forward-backward speed-thrust control realisation.

Inverting the lift and drag model, a feed-forward control law can be found, where the gains depend on certain wind velocities; that is, the feed-forward control is *gain-scheduling*. In order to correct model uncertainties as well as any (unexpected) disturbances, a static-gain feed-back control loop is added. The resulting controller is shown in Fig. 7.2.

Feed-forward control law Re-writing Eq. 7.1, we get the change of thrust and lift force from the equilibrium force

$$\begin{bmatrix}\Delta F_F\\ \Delta F_L\end{bmatrix} = \begin{bmatrix}F_F\\ F_L\end{bmatrix} - \begin{bmatrix}F_{T0}\\ F_{L0}\end{bmatrix} = \mathbf{F}_{\text{Æ}}(V_W)\begin{bmatrix}\Delta\Theta\\ \Delta F\end{bmatrix} = \mathbf{F}_{\text{Æ}}(V_W)\left(\begin{bmatrix}\Theta\\ F\end{bmatrix} - \begin{bmatrix}\Theta_0\\ F_0\end{bmatrix}\right) \tag{7.2}$$

where $F_{T0} = F_{D0}$ and $F_{L0} = mg$, *i.e.* $\dot{V}_{A0} = \ddot{H}_0 = 0$. Eq. 7.2 results into (Newton's second law of motion)

$$\begin{bmatrix}\Delta F_F\\ \Delta F_L\end{bmatrix} = m\begin{bmatrix}\dot{V}_A\\ \ddot{H}\end{bmatrix} = \mathbf{F}_{\text{Æ}}(V_W)\left(\begin{bmatrix}\Theta\\ F\end{bmatrix} - \begin{bmatrix}\Theta_0\\ F_0\end{bmatrix}\right) \tag{7.3}$$

and by inversion we obtain a wind velocity-dependent forward control law:

$$\begin{bmatrix}\Theta^c\\ F^c\end{bmatrix} = \begin{bmatrix}\Theta_0\\ F_0\end{bmatrix} + m\mathbf{F}_{\text{Æ}}^{-1}\begin{bmatrix}\dot{V}_A^{SP}\\ \ddot{H}^{SP}\end{bmatrix} \tag{7.4}$$

Feed-back control law To correct model uncertainties as well as reject disturbances, the control approach is completed by a static-gain feed-back control loop with proportional and integral gains, k and i. As it is customary, the basic control law states

$$\begin{bmatrix} \dot{V}_A^{\mathrm{fb}} \\ \ddot{H}^{\mathrm{fb}} \end{bmatrix} = k \begin{bmatrix} \Delta\dot{V}_A \\ \Delta\ddot{H} \end{bmatrix} + i \int \begin{bmatrix} \Delta\dot{V}_A \\ \Delta\ddot{H} \end{bmatrix} \mathrm{d}t \tag{7.5}$$

Note, that we add the feed-back correction before the forward control block rather than after in order to use the aerodynamic derivation matrix featuring a direction guess. While the proportional error can be derived directly, namely

$$\begin{bmatrix} \Delta\dot{V}_A \\ \Delta\ddot{H} \end{bmatrix} = \begin{bmatrix} \dot{V}_A^{SP} \\ \ddot{H}^{SP} \end{bmatrix} - \begin{bmatrix} \dot{V}_A \\ \ddot{H} \end{bmatrix} \tag{7.6}$$

for the integrated error we notice

$$\int \begin{bmatrix} \Delta\dot{V}_A \\ \Delta\ddot{H} \end{bmatrix} \mathrm{d}t = \int \left(\begin{bmatrix} \dot{V}_A^{SP} \\ \ddot{H}^{SP} \end{bmatrix} - \begin{bmatrix} \dot{V}_A \\ \ddot{H} \end{bmatrix} \right) \mathrm{d}t = \int \begin{bmatrix} \dot{V}_A^{SP} \\ \ddot{H}^{SP} \end{bmatrix} \mathrm{d}t - \int \begin{bmatrix} \dot{V}_A \\ \ddot{H} \end{bmatrix} \mathrm{d}t = \begin{bmatrix} \tilde{V}_A^{ref} \\ \tilde{\dot{H}}^{ref} \end{bmatrix} - \begin{bmatrix} V_A \\ \dot{H} \end{bmatrix} \tag{7.7}$$

where $\tilde{V}_A^{ref}$, $\tilde{\dot{H}}^{ref}$ denote the reference velocities obtained by integration of the set-point acceleration, whereas V_A, $\dot{H}$ are the true velocities in the longitudinal plane.

Eventually, the combined forward-backward control law results into

$$\begin{bmatrix} \Theta^c \\ \digamma^c \end{bmatrix} = \begin{bmatrix} \Theta_0 \\ \digamma_0 \end{bmatrix} + m\mathbf{F}_{\AE}^{-1} \begin{bmatrix} \dot{V}_A^{SP} \\ \ddot{H}^{SP} \end{bmatrix} + m\mathbf{F}_{\AE}^{-1} k \begin{bmatrix} \Delta\dot{V}_A \\ \Delta\ddot{H} \end{bmatrix} + m\mathbf{F}_{\AE}^{-1} i \left(\begin{bmatrix} \tilde{V}_A^{ref} \\ \tilde{\dot{H}}^{ref} \end{bmatrix} - \begin{bmatrix} V_A \\ \dot{H} \end{bmatrix} \right) \tag{7.8}$$

as outlined by Fig. 7.2.

7.2. An adaptive flapping-frequency controller

The relation between throttle command $\digamma^c$ and resulting flapping frequency f of the Delfly can be approximated to be quasi-proportional with

$$f = \kappa_{f\digamma} \digamma^c \tag{7.9}$$

and the proportionality constant $\kappa_{f\digamma}$ is a function of the weight, battery voltage, and airspeed. While weight and air-speed are (more or less) constant in steady-state flight, the battery voltage decreases when the battery is discharged. Assuming $\kappa_{f\digamma}$ is estimated, we can design a open-loop control law:

$$\digamma^c = \hat{\kappa}_{f\digamma}^{-1} f = \theta_{\digamma f} f \tag{7.10}$$

If the proportionality constant is not estimated closely, and since it is varying over the battery voltage, the feed-forward gain, θ_{Ff}, needs to be updated accordingly. A suitable mechanism is given by the *Model reference adaptive system*; here, the feed-forward gain is adapted depending on the error between the plant output and a reference model output. Among different adaptation laws, the "MIT rule" is commonly used, [89] proposing the change of θ_{Ff} over time by the reference error:

$$\dot{\theta}_{Ff} = \gamma_{\dot{\theta}f} \left(f^{ref} - f \right) \tag{7.11}$$

where $\gamma_{\dot{\theta}f}$ is, simply spoken, the adaptation rate and the reference flapping frequency is clearly given by

$$f^{ref} = f^{SP} \tag{7.12}$$

The adaptive flapping frequency controller is shown by Fig. 7.3.

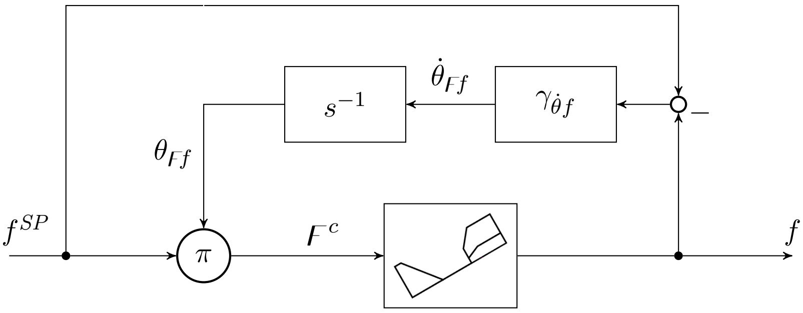


Figure 7.3.: A model reference adaptive system to control flapping frequency. *Note that the multiplication block denoted by* "π" *is not part of* [DIN IEC 60050-351].

7.3. Semi-adaptive control approach

Though the combined forward-backward controller clearly has advantages compared to a feedback-only controller while being much simpler than sophisticated adaptive approaches, it also suffers from the need of the integral feedback to compensate model errors in both aerodynamic derivation matrix and equilibrium condition. In order to reduce the latter, we enhance the control approach by a quasi-adaptation stage using a second-order feedback executed *once and before* the already described controller takes place; thus, we call this extended approach *semi-adaptive.*

First, notice we can use a position feedback to find the "true" equilibrium condition without de-stabilizing the system, as proven in Lemma 7.1.

Lemma 7.1. *Let $\hat{\Theta}_0$, $\hat{\digamma}_0$, $\hat{\mathbf{F}}_{Æ}$ be close guesses of the equilibrium pitch angle, throttle command, and aerodynamic derivation matrix, respectively; given an initial longitudinal position (x_{w0}, H_0) and a feedback law*

$$\begin{bmatrix} \Theta^c \\ \digamma^c \end{bmatrix} = \begin{bmatrix} \hat{\Theta}_0 \\ \hat{\digamma}_0 \end{bmatrix} + m\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x_{w0} - x_w \\ H_0 - H \end{bmatrix} \tag{7.13}$$

for a suitable gain $\gamma \in \mathbb{R}_{>0}$ it holds

1. *there is a position $\left(x'_w, H'\right)$ in the neighbourhood of (x_{w0}, H_0) such that the equilibrium condition is achieved;*
2. *in the neighbourhood of $\left(x'_w, H'\right)$ this position is approached.*

Proof. Let Θ_0, $\digamma_0$, $\mathbf{F}_{Æ}$ be the true pitch angle, throttle command, and aerodynamic derivation matrix in equilibrium;

1. if the equilibrium condition is achieved, *i.e.* $\Theta^c = \Theta_0$, $\digamma^c = \digamma_0$, Eq. 7.13 yields

$$\begin{bmatrix} x'_w \\ H' \end{bmatrix} = \begin{bmatrix} x_{w0} \\ H_0 \end{bmatrix} + m^{-1}\hat{\mathbf{F}}_{Æ}\gamma^{-1} \begin{bmatrix} \Theta_0 - \hat{\Theta}_0 \\ \digamma_0 - \hat{\digamma}_0 \end{bmatrix} \tag{7.14}$$

   and $\left|x_{w0} - x'_w\right|$, $\left|H_0 - H'\right|$ are small for a suitable γ.

2. Let (x_w, H) be a position in the neighbourhood of the equilibrium $\left(x'_w, H'\right)$; rearranging Eq. 7.14 yields

$$\begin{bmatrix} \Theta^c \\ \digamma^c \end{bmatrix} = \begin{bmatrix} \hat{\Theta} \\ \hat{\digamma} \end{bmatrix} + m\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x_{w0} - x'_w \\ H_0 - H' \end{bmatrix} = \begin{bmatrix} \Theta_0 \\ \digamma_0 \end{bmatrix}$$

   and inserting into Eq. 7.13, we get

$$\begin{bmatrix} \Theta^c \\ \digamma^c \end{bmatrix} = \begin{bmatrix} \hat{\Theta} \\ \hat{\digamma} \end{bmatrix} + m\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x_{w0} - x'_w \\ H_0 - H' \end{bmatrix} + m\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x'_w - x_w \\ H' - H \end{bmatrix} = \begin{bmatrix} \Theta_0 \\ \digamma_0 \end{bmatrix} + m\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x'_w - x_w \\ H' - H \end{bmatrix} \tag{7.15}$$

   Finally, inserted into Eq. 7.3, assuming $\Theta = \Theta^c, \digamma = \digamma^c$ we result in

$$\begin{bmatrix} \dot{V}_A \\ \ddot{H} \end{bmatrix} = m^{-1}\mathbf{F}_{Æ}\left(\begin{bmatrix} \Theta^c \\ \digamma^c \end{bmatrix} - \begin{bmatrix} \Theta_0 \\ \digamma_0 \end{bmatrix}\right) = \mathbf{F}_{Æ}\hat{\mathbf{F}}_{Æ}^{-1}\gamma \begin{bmatrix} x'_w - x_w \\ H' - H \end{bmatrix} \tag{7.16}$$

As $\hat{\mathbf{F}}_{\AE}$ is a close guess of $\mathbf{F}_{\AE}$, we can assume $\mathbf{F}_{\AE}\hat{\mathbf{F}}_{\AE}^{-1}$ is diagonal and positive; thus we have

$$\begin{aligned} \dot{V}_A &= \lambda_1 \gamma \left(x'_w - x_w\right) \\ \ddot{H} &= \lambda_2 \gamma \left(H' - H\right) \end{aligned}$$

where $\lambda_1\gamma, \lambda_2\gamma > 0$ and the equilibrium position $\left(x'_w, H'\right)$ is stable in its neighbourhood.

□

That is, the stable position $\left(x'_w, H'\right)$ is approached and Eq. 7.13 yields the true pitch angle $\Theta_0 = \Theta^c\left(x'_w, H'\right)$ and throttle command $\digamma_0 = \digamma^c\left(x'_w, H'\right)$.

Taking advantage of Lemma 7.1 we can now split our control approach into two subsequent, disjunct stages: *adaptation* and *correction.* Fig. 7.4 is showing the basic idea of the two stages; after the adaptation is completed, the controller switches to the correction stage and remains here. We denote the time of switching by t_A.

Adaptation stage (Fig. 7.4a) The position feedback law of Eq. 7.13 is utilized to find the stable position; to do so, the acceleration set-point and feedback gains except for γ are set to zero, $\dot{V}_A^{SP} = \ddot{H}^{SP} = k = i = 0$. As soon as the position is stabilized by the adaptation feedback law, we have found by Lemma 7.1 the equilibrium condition and

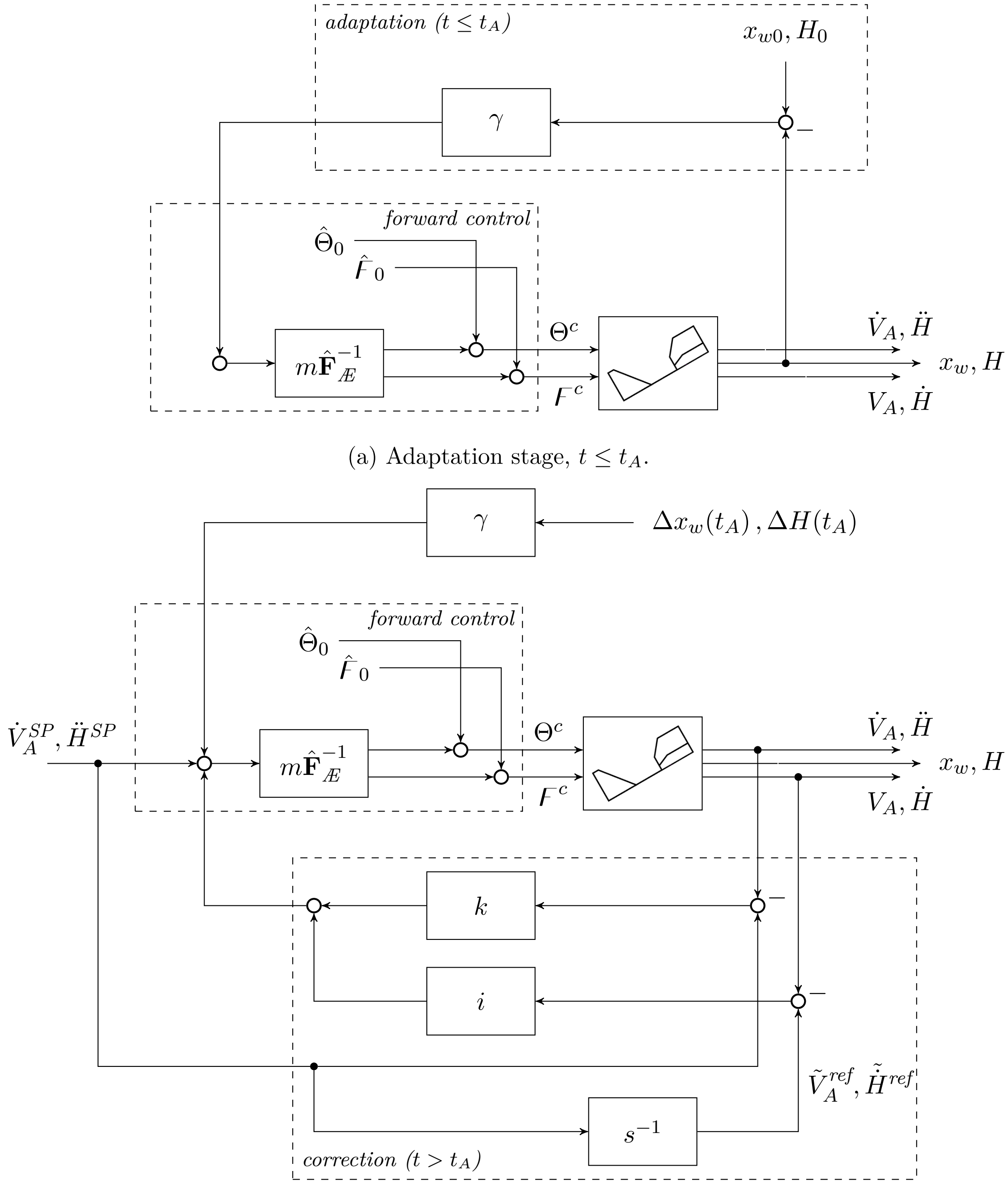
$$\begin{bmatrix} \Theta_0 \\ \digamma_0 \end{bmatrix} = \begin{bmatrix} \hat{\Theta}_0 \\ \hat{\digamma}_0 \end{bmatrix} + m\hat{\mathbf{F}}_{\AE}^{-1}\gamma \begin{bmatrix} \Delta x_w(t_A) \\ \Delta H(t_A) \end{bmatrix} \tag{7.17}$$

i.e. model uncertainties in the equilibrium condition are compensated. Now, in order to counteract the steady-state position error left, the correction stage takes over.

Correction stage (Fig. 7.4b) Afterwards, the previous adaptation is taken into account by Eq. 7.17, considering Δx_w, ΔH to be constantly assigned with the final values of the adaptation stage. From here on the control law equals the combined forward-backward control described in Section 7.1 except for the adapted pitch and throttle equilibria which yields now

$$\begin{bmatrix} \Theta_{0,t>t_A} \\ \digamma_{0,t>t_A} \end{bmatrix} = \begin{bmatrix} \hat{\Theta}_0 \\ \hat{\digamma}_0 \end{bmatrix} + m\hat{\mathbf{F}}_{\AE}^{-1}\gamma \begin{bmatrix} \Delta x_w(t_A) \\ \Delta H(t_A) \end{bmatrix} \tag{7.18}$$

where $\hat{\Theta}_0$, $\hat{\digamma}_0$ denotes the initially estimated equilibria while $\Theta_{0,t>t_A}$, $\digamma_{0,t>t_A}$ the equilibria after adaptation.



(a) Adaptation stage, $t \leq t_A$.

(b) Correction stage, $t > t_A$.

Figure 7.4.: The two-stages semi-adaptive control approach: at time $t = t_A$, the semi-adaptive controller switches from the adaptation (a) to the correction (b) stage.

8. Implementation

The paparazzi UAV autopilot provides an open-source implementation of a flight controller hierarchy enabling both radio-controlled and autonomous flying. The autopilot subsystems are programmed in C and can be adapted to suit specific needs of the autopilot design; furthermore, modules can be developed, distributed, and added through a simplified interface. We will briefly present in this chapter the overall structure of the paparazzi autopilot with its subsystems and modules used; the basic control implementation provided by paparazzi and used to control the Delfly's attitude; and the discrete-time realisation of an on-board state filter as well as of the Delfly control hierarchy discussed before. Eventually, we present an integer calculation error affecting our results in the wind-tunnel and how we solved it.[1]

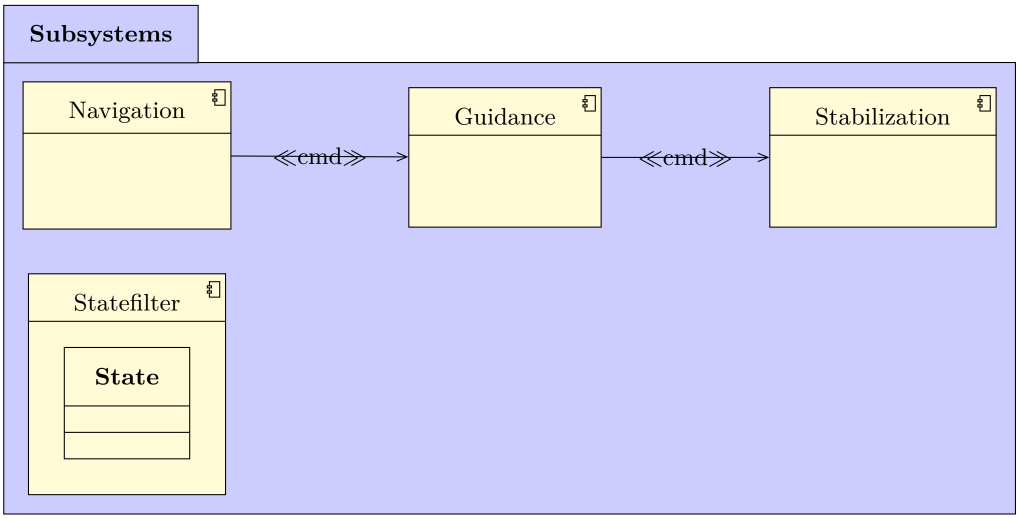


Figure 8.1.: [UML 2.5] Components Diagram of the basic paparazzi UAV autopilot.

8.1. The paparazzi autopilot and Delfly control module

The mandatory flight control components are provided by paparazzi as built-in *subsystems*; optional *modules* complete the autopilot implementation. The important subsystems with respect to the this thesis are shown in Fig. 8.1 on page 44.

[1] For the impact of the integer calculation error and how we found it, see Section 9.4.

The Navigation component commands position and/or velocity set-points to the Guidance in order to maintain a certain trajectory, flight-plan, or mission goal; the Guidance component itself commands a desired attitude to the Stabilization as well as a throttle command; the Stabilization component eventually stabilizes and controls the aircraft's attitude. In the meantime, the Statefilter component estimates the aircraft attitude, position, and change of both based on inertial sensors (accelerometer and gyroscopes) and position tracking (here, OptiTrack).

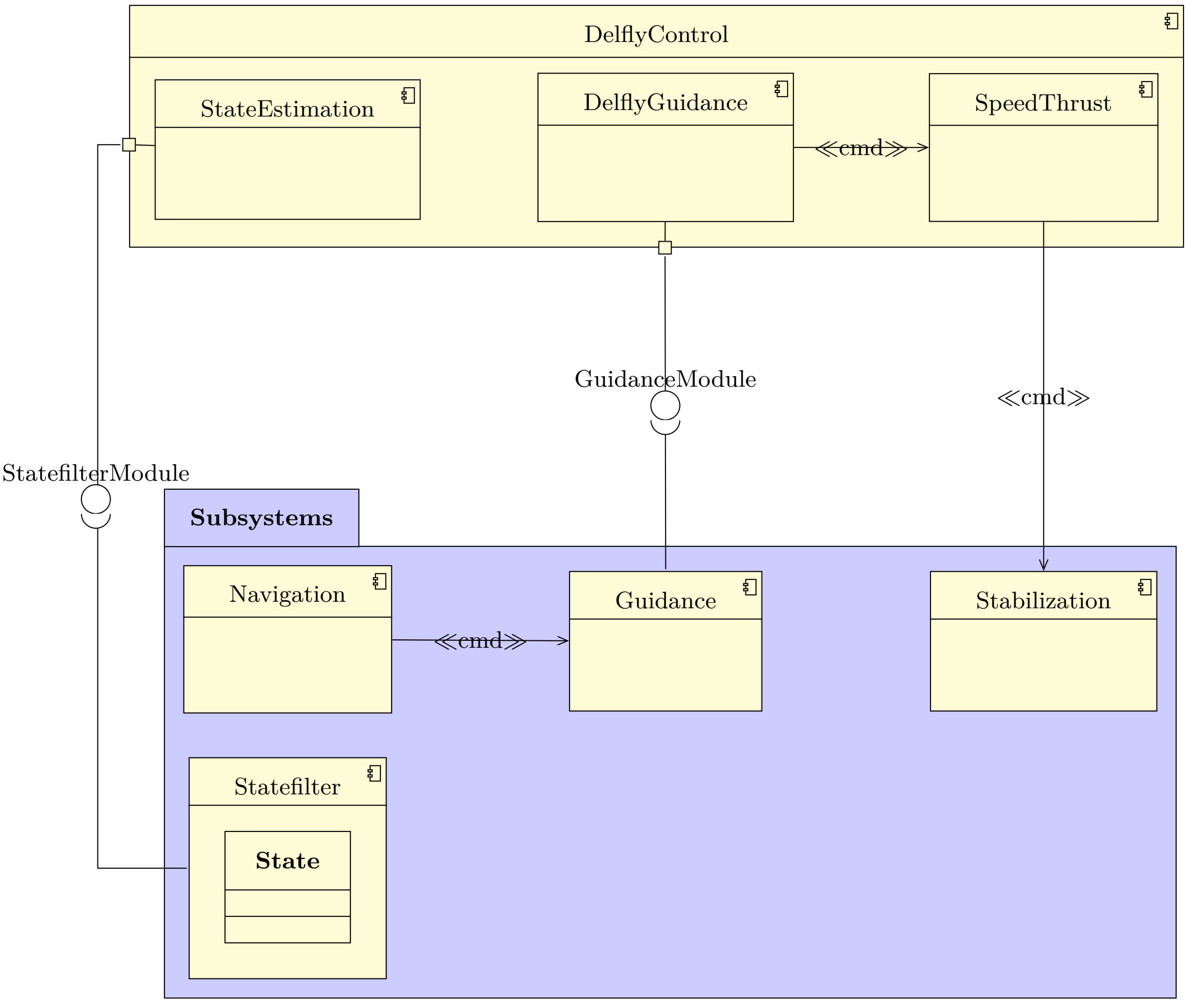


Figure 8.2.: [UML 2.5] Components Diagram of the Delfly control module.

Both the Guidance and Statefilter components can be extended by module implementations in order to suit particular needs. We do so to guide the Delfly through the wind-tunnel and estimate the position, velocity, and acceleration from the OptiTrack system. The task of the Navigation in the wind-tunnel is rather simple, as just a single waypoint to be kept is given; the basic control implementation of the Stabilization is presented in Sec. 8.2.

We implemented the designed control hierarchy and its auxiliaries as subcomponents of the `DelflyControl` module. Fig. 8.2 on page 45 shows the interconnection of this module's subcomponents to the paparazzi built-in components and each other: The `DelflyGuidance` subcomponent, implementing Chapter 6, extends the paparazzi `Guidance` component through the provided interface; it commands the desired acceleration to the speed-thrust controller (Chapter 7) by the `SpeedThrust` subcomponent, bypassing paparazzi's `Guidance`-to-`Stabilization` connection; the latter is then commanding the desired attitude to the paparazzi `Stabilization` component; finally, the `StateEstimation` subcomponent implements an on-board state filter (Section 8.3) extending the `Statefilter` component.

8.2. Basic control by paparazzi

To control the Delfly's attitude, we simply re-used the attitude control provided by the paparazzi `Stabilization` component. This component is implementing a static-gain feedback control law including integration and derivation of the control error (PID), where rather the error in rates is used for derivation. The resulting feedback loop is shown in Fig. 8.3.

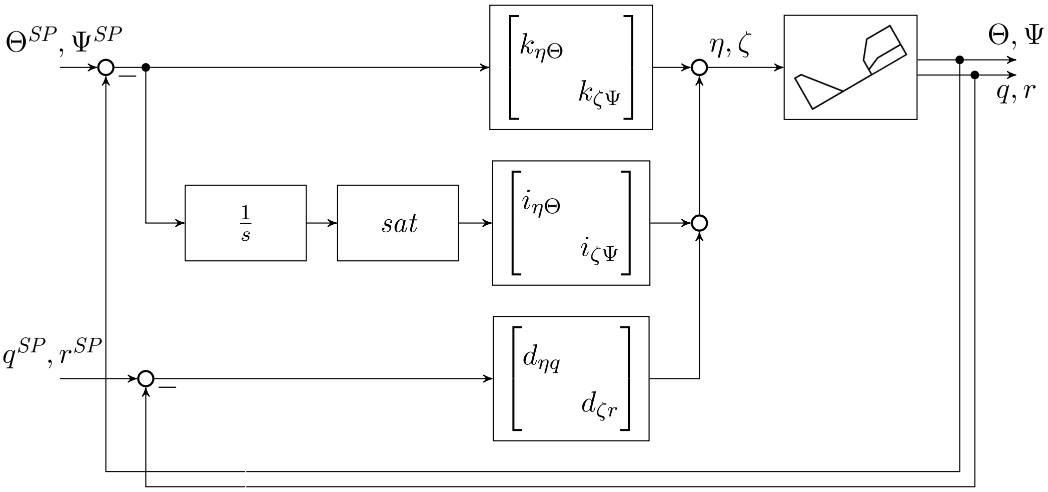


Figure 8.3.: The attitude control loop implemented in paparazzi.

Since the Delfly is statically stable around the roll axis, although affected by the rudder as well, and we do not control roll angle nor rate, the roll loop is dismissed here. The desired pitch and heading angle Θ^{SP}, Ψ^{SP} are given by the overlaying speed-thrust controller; the pitch and heading rate set-points are zero (no attitude change in steady-state; $q^{SP} = r^{SP} = 0$).

We have tuned the gains of the basic control component in order to achieve a suitable control response while flying in the wind-tunnel, preceding the free-flight tests. The resulting gain values used for the tests are presented in App. B.2.

8.3. On-board state filter

We extended the paparazzi statefilter component in order to determine the Delfly's velocity and acceleration by the OptiTrack position rather than the accelerometers of the Lisa-S's inertial measurement unit, as the wing flaps introduce high frequency accelerations.

Simply spoken, we can calculate the velocity as discrete derivation of the position,

$$\dot{\hat{\mathbf{x}}}_g(k) = \frac{\hat{\mathbf{x}}_g(k) - \hat{\mathbf{x}}_g(k-1)}{T} \tag{8.1}$$

where k denotes a discrete execution step and T is the period of position updates at 30 Hz. The acceleration is calculated from the velocity in the same manner:

$$\ddot{\hat{\mathbf{x}}}_g(k) = \frac{\dot{\hat{\mathbf{x}}}_g(k) - \dot{\hat{\mathbf{x}}}_g(k-1)}{T} \tag{8.2}$$

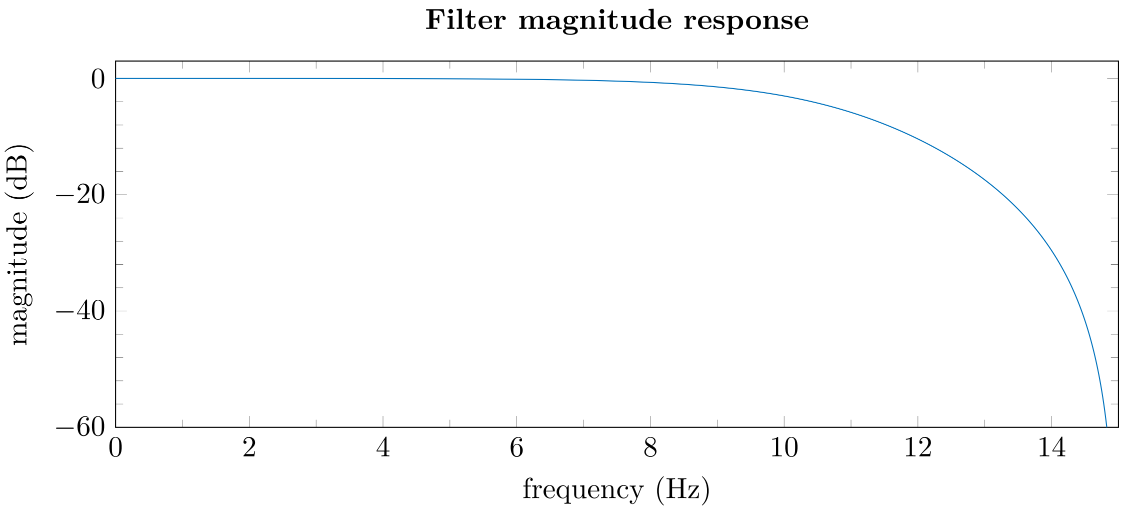


Figure 8.4.: Bode magnitude response of the Butterworth filter.

During each wing flap, both velocity and acceleration are oscillating as the body rocks up and down. As we are rather interested in the "overall" velocity and acceleration, we introduce a low-pass filter in order to smooth the position signal before derivation:

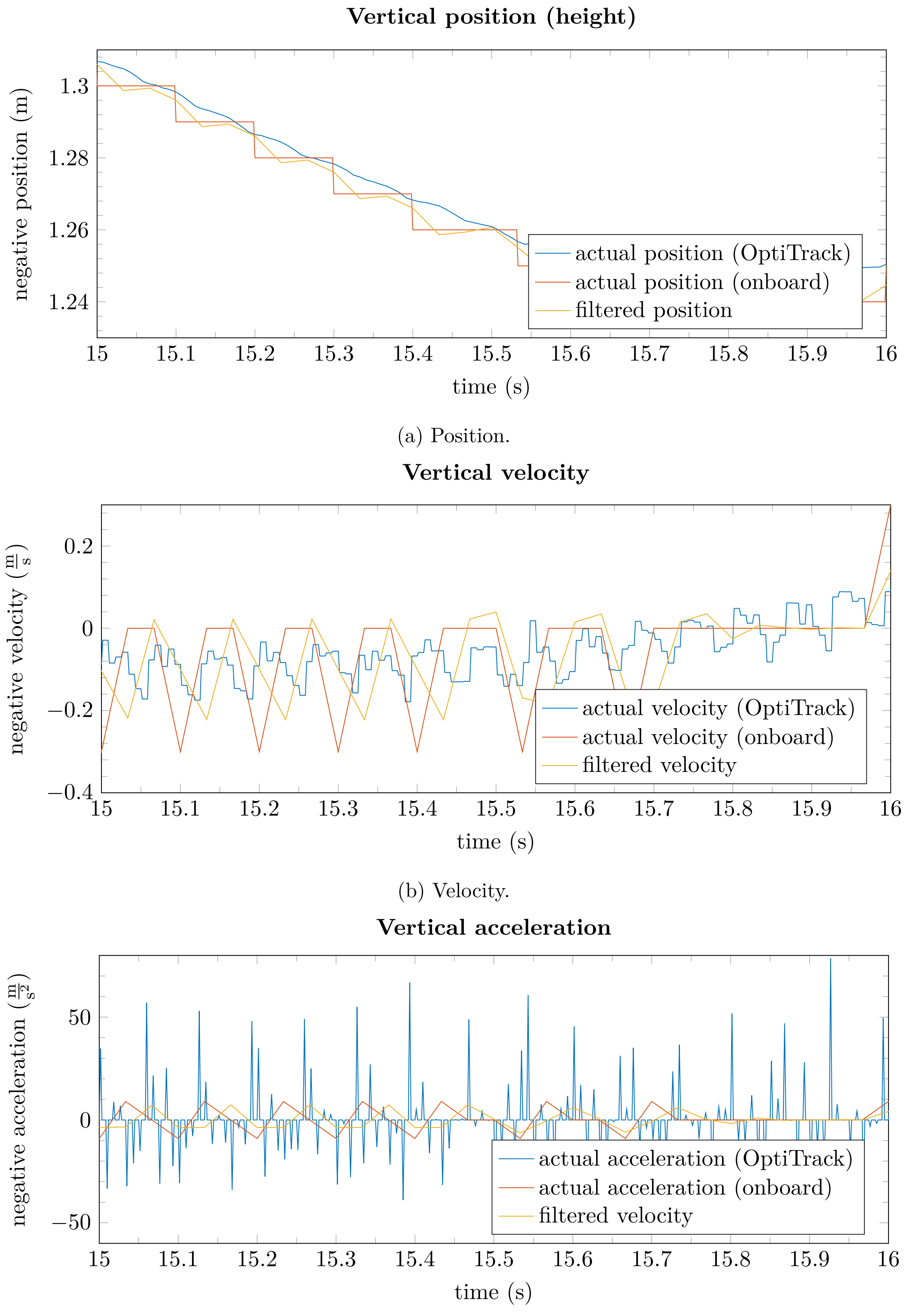
$$\hat{\mathbf{x}}_g^{\text{filt}}(s) = G_{\text{filt}}(s)\,\hat{\mathbf{x}}_g \tag{8.3}$$

$$\dot{\hat{\mathbf{x}}}_g^{\text{filt}}(k) = \frac{\hat{\mathbf{x}}_g^{\text{filt}}(k) - \hat{\mathbf{x}}_g^{\text{filt}}(k-1)}{T} \tag{8.4}$$

$$\ddot{\hat{\mathbf{x}}}_g^{\text{filt}}(k) = \frac{\dot{\hat{\mathbf{x}}}_g^{\text{filt}}(k) - \dot{\hat{\mathbf{x}}}_g^{\text{filt}}(k-1)}{T} \tag{8.5}$$

As for the low-pass filter, we employed a second-order Butterworth filter with a cut-off frequency $f_{\text{cut}} = 10\,\text{Hz}$ and a sample frequency $f_{\text{sample}} = 30\,\text{Hz}$, where the first corresponds to the flapping-wing frequency and the latter to the position updates. The Bode magnitude response of the Butterworth filter used is shown in Fig. 8.4; the filtered position signal as well as the velocity and acceleration after filtering is shown in Fig. 8.5, too.

We controlled the Delfly based on the un-filtered position, while the filtered position was used to estimate velocity and acceleration only.



(a) Position.

(b) Velocity.

(c) Acceleration.

Figure 8.5.: Comparison of position, velocity, and acceleration as recorded by OptiTrack, sampled on-board, and filtered.

8.4. Digital control realisation

In control theory, two basic assumptions are made about the signals in the control loop: their values are *continuous*, that is each signal has a distinct value at any time $t \in \mathbb{R}$ and its value can change due to a infinitesimal timestep $\mathrm{d}t \longrightarrow 0$; and the values are *real* numbers, *i.e.* a signal can take a value which is either rational or irrational. In digital computation, however, signals are both discrete-time and discrete-valued; this discrepancy needs to be taken into account both in the control design and implementation.

Quasi-continuous loop execution Opposite to the continuous assumption, a digital control implementation is executed with non-infinite frequency T^{-1} and properties of continuous control hold not. If T is sufficiently small, on the other hand, the control loop is considered to be *quasi-continuous*; that is, it may still be represented in continuous-time. [86] Thus, a property derived for a continuous control approach *does* apply here.

Representation of real numbers In modern binary computers, numbers are represented by sequences of *bits*, which are either 0 or 1, and with fixed length.(2) While this is fine for integer numbers, in order to approximate a real number a higher section of the bit sequence represents the integer part and a lower section the fractional places; this is comparable to the scientific notation:

$$x \approx P \cdot 2^{-M} \tag{8.6}$$

where P is a integer number, $P \in \mathbb{Z}$, and M is the integer *mantissa* $M \in \mathbb{N}$. Obviously, the range of representable numbers and the precision of the approximation is limited by the number of bits used to display P and M, respectively.

Now, the value of the mantissa M is either constant (*fixed-point* representation) or varying (*floating-point* representation). In comparison, the floating-point representation needs to store each variable's actual value of M in addition, but provides a wider range of representable numbers though decreasing the accuracy of larger ones. While floating-point data types are now customary in all higher programming languages (cf. `float` and `double` in C or Java), in the early days of computation they have caused serious errors and thus have been objected by programmers for a long time [96]. Today still, efficient floating-point calculation requires a particular processing unit and this kind of data types is avoided for safety-related application.

(2) In today's higher programming languages common data types span bit sequences of 8 (`byte`, `char`), 16 (`short`), 32 (`int`, `float`), or 64 (`long`, `double`) bits.

Due the latter, we have implemented our control loop based on the fixed-point representations. In order to represent values while taking into account the accuracy typically need, paparazzi provides different mantissas for the miscellaneous types of quantities. A few exeamplary quantities and the mantissas used are shown in Tab. 8.1.

Quantity	Mantissa
Position	13
Velocity	19
Acceleration	10

Quantity	Mantissa
Angle	12
Rate	12
Trigonometry	14

Table 8.1.: Exemplary quantities and their mantissas in the autopilot implementation.

Note finally, that the maximal (and minimal) value of a variable in binary fixed-point representation depends now on the number of bits N, the mantissa M, and whether the variable is signed or unsigned.

8.5. Wireless telemetry and telecommand link

In order to communicate with the aircraft, that is telecommand and position upload as well as telemetry download, a wireless link between the paparazzi Ground Control Station (GCS) and the autopilot is established. After the former SuperbitRF link turned out to be unreliable (cf. Sec. 2.3), we implemented an IP-based communication bridging the serial port of the Lisa-S board to a Wireless LAN adapter connected to the extended ground network.

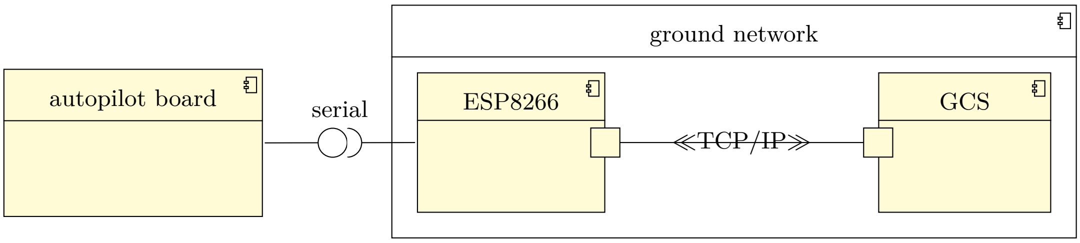


Figure 8.6.: [UML 2.5] Components Diagram of the wireless communication link.

This serial-to-wifi bridge is established by an EspressIf ESP8266/ESP09 module [97] running the open-source esp-link firmware [98]. Data packages from and to the autopilot are now sent using the *Transmission Control Protocol* (TCP/IP). The components of the wireless communicaiton link are shown in Fig. 8.6. TCP/IP, notably, establishes a hand-shaking mechanism to ensure the message reception.

9. Test Results and Discussion

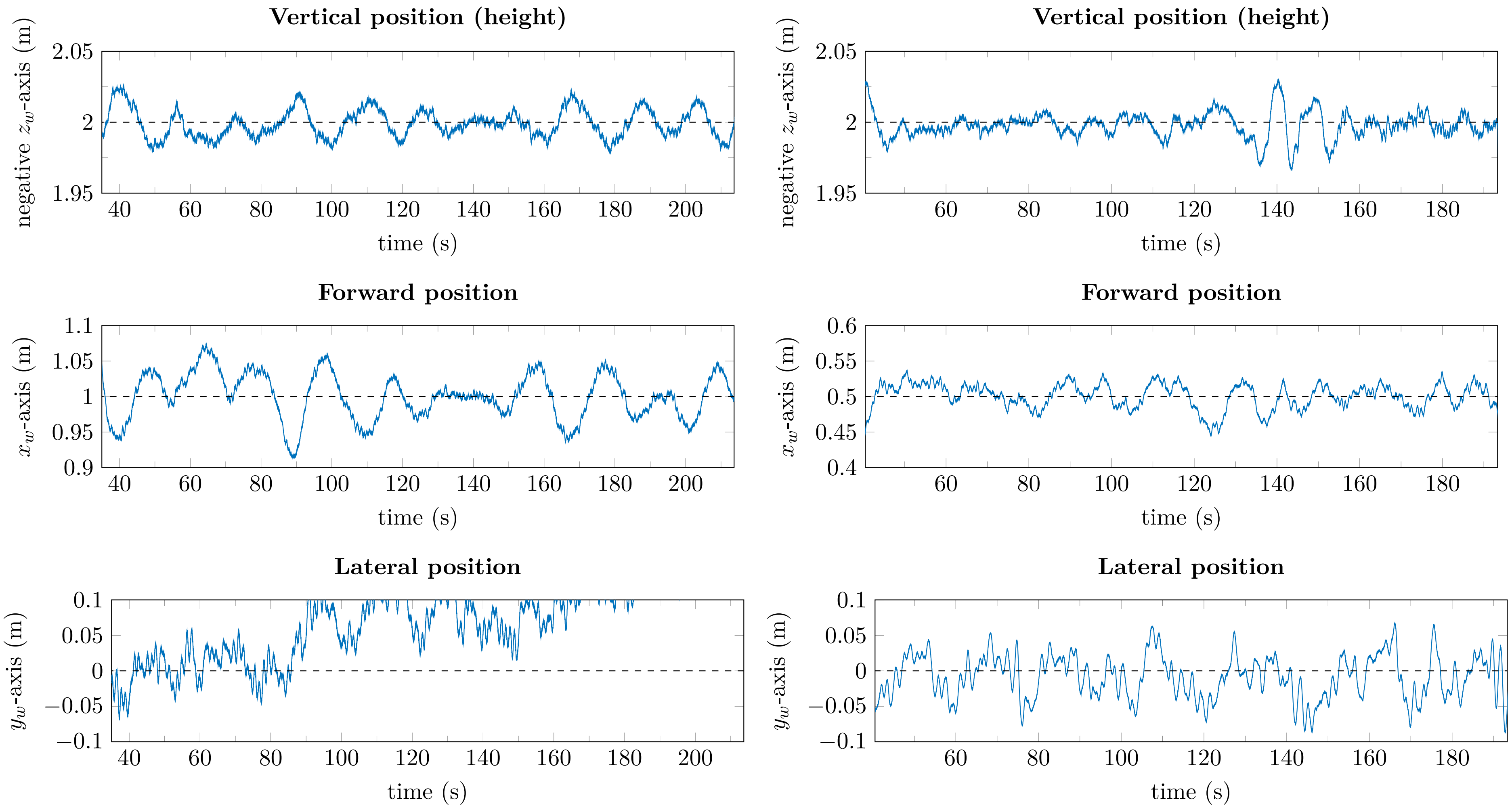
In this chapter, we present the results of the free-flight tests in the TU Delft Open Jet Wind-tunnel Facility [83]. For the tests and discussion, we focus on precision and robustness in the steady-state flight, as well as on settling and step responses.

Fig. 9.1a shows the first flight in the wind-tunnel by the control design of this thesis: the speed-thrust controller is not executing the adaptation phase here, and the lateral guidance works without integral gain. The settling response in the vertical and horizontal axis shows a huge overshot; once settled, while the vertical response achieves a satisfying precision (within ±5 cm) around the set-point, the horizontal response still shows a low frequency oscillation up to about ±20 cm. In the lateral axis there is no settling at all and the system is jittering at high frequency.

We were able to improve the steady-state precision in horizontal and vertical axis as well as to remove the divergence in the lateral axis by adding the speed-thrust adaptation phase and a lateral guidance integral gain, respectively, during the wind-tunnel tests (Fig. 9.1b). However, we were not able to solve the overshot but reduced that to an error in the control programming code: due to an integer calculation error in the speed-thrust control implementation, the settling response during the flight-tests in the wind-tunnel is heavily impaired and thus not representative of the control approach.

We will show the settling response in the wind-tunnel, though; after correction, the proper step responses have been tested outside the wind-tunnel and are presented finally.

If not stated otherwise, the wind-tunnel wind speed was set to $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; we observed a precision of the built-in speed controller about $\pm 0.2\,\frac{\mathrm{m}}{\mathrm{s}}$. For the forward and vertical guidance, we chose the desired poles to be $\varsigma_{1,2} = -1$ leading to a non-oscillating second-order response of the ideal system; the lateral guidance gains were set equally. Except for the first flight (Fig. 9.1a), the speed-thrust controller was executing the adaptation stage as well (*semi-adative speed-thrust control*), with an adaptation rate $\gamma = 250\,\%$, and the lateral integral gain was introduced (Fig. 9.1b).



(a) Combined speed-thrust control. ($k = 0$, $i = 200\,\%$; *no lateral integral gain.*)

(b) Semi-adaptive speed-thrust control. ($k = 0$, $i = 300\,\%$.)

Figure 9.1.: Comparing results of the wind-tunnel flights; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$.)

9.1. Precision in steady-state

We have tested the Delfly control approach with the semi-adaptive speed-thrust controller for different integral gains i; most remarkable are $i = 200\,\%$ and $i = 300\,\%$ – the proportional gain p has not notably affected the precision at all. As both gains have in common, the vertical control is more precise than the forward, while the lateral is much worse. The results of the different gain settings tests are presented in App. C.

We were able to improve the results to its best using an integral gain $i = 300\,\%$ only; the results are shown in Fig. 9.3 on page 55: the vertical axis achieves a precision of ±1.0 cm except for a disturbance around $t = 140\,\mathrm{s}$. The precision in the forward axis has been improved to ±2.5 cm, except for a disturbance around $t = 130\,\mathrm{s}$. Both axes show a high-frequency oscillations which superpose the outcome. The performance in the lateral axis is not influenced by the integral gain and is oscillating less but fully in the range of it's precision; thus, a lateral precision of ±5.0 cm is achieved most of the time.

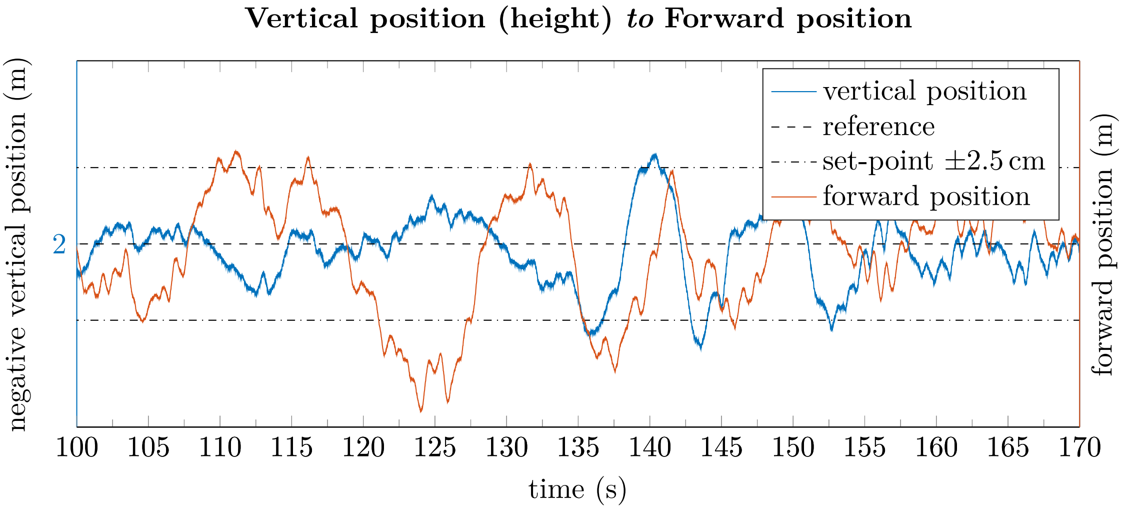
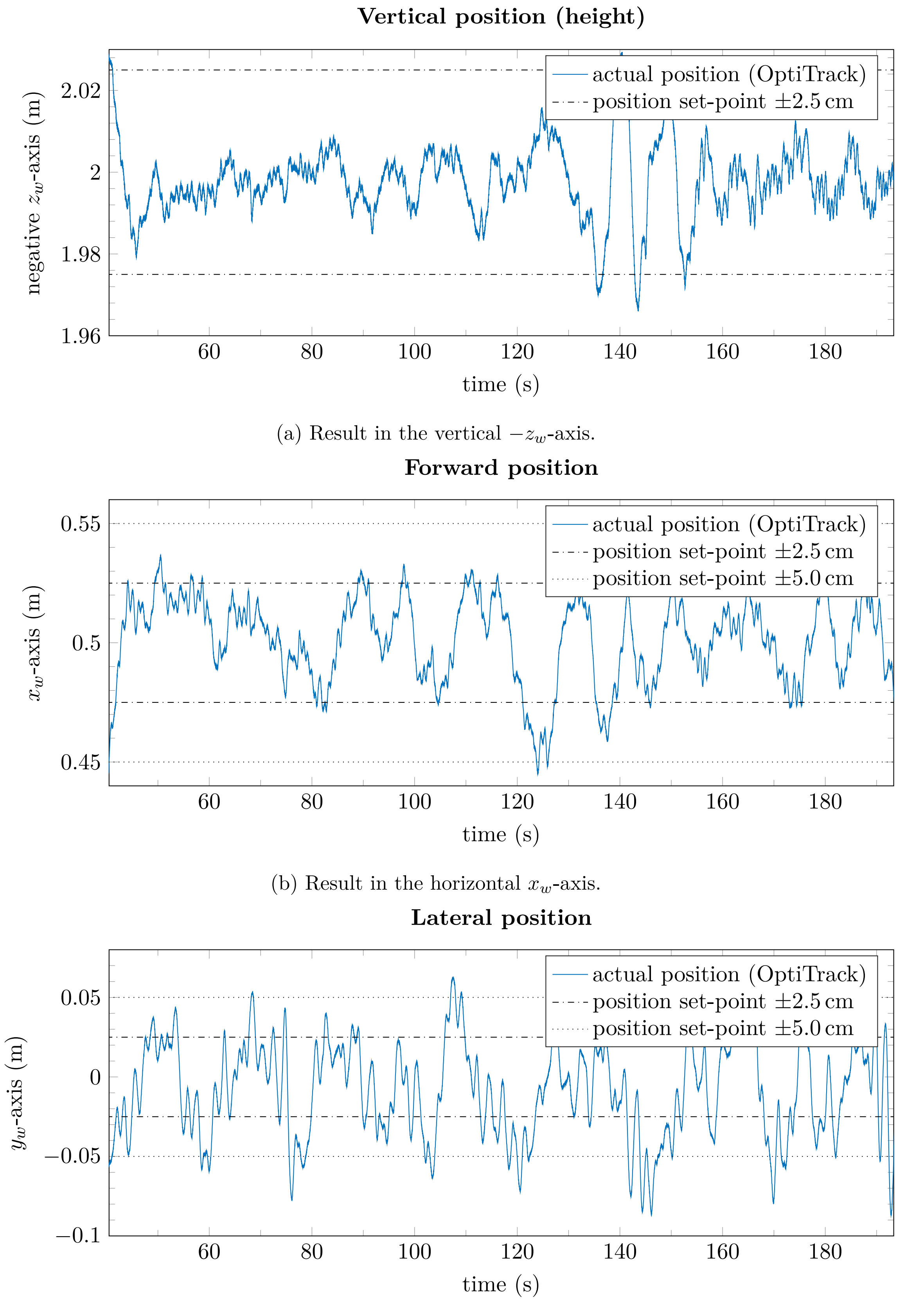


Figure 9.2.: Comparing behaviour in vertical and forward position for $i = 300\,\%$ in the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$*; semi-adaptive speed-thrust control with* $k = 0$, $i = 300\,\%$.)

There are variations in both the vertical and forward axes for $t \geq 120\,\mathrm{s}$, leading to a notable disturbance in the vertical axis at $t = 140\,\mathrm{s}$; Fig. 9.2 shows this part. The forward axis here shows imbalances in the beginning, leading to a deviation in the vertical axis then. The settling of the vertical axis then is seen in the results. The unsteady forward position is probably caused by variances in the wind speed, combined with the increased integral gain resulting into a stronger response to errors.



(a) Result in the vertical $-z_w$-axis.

(b) Result in the horizontal x_w-axis.

(c) Result in the lateral z_w-axis.

Figure 9.3.: Steady-state results of the wind-tunnel flight with $i = 300\,\%$; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$*; semi-adaptive speed-thrust control with* $k = 0$, $i = 300\,\%$.)

9.2. Settling behaviour (wind-tunnel)

The settling response of the Delfly control approach in the implementation of the wind-tunnel tests, for all gains of the semi-adaptive speed-thrust controller, showed a quite large overshoot: in the vertical axis, the overshot yields 50 % of the position set-point or 1 m in absolute; the forward overshoot advances or exceeds 0.3 m, *i.e.* 60 % of the forward position set-point. The lateral axis, however, does not show a notable overshoot beyond the accuracy.

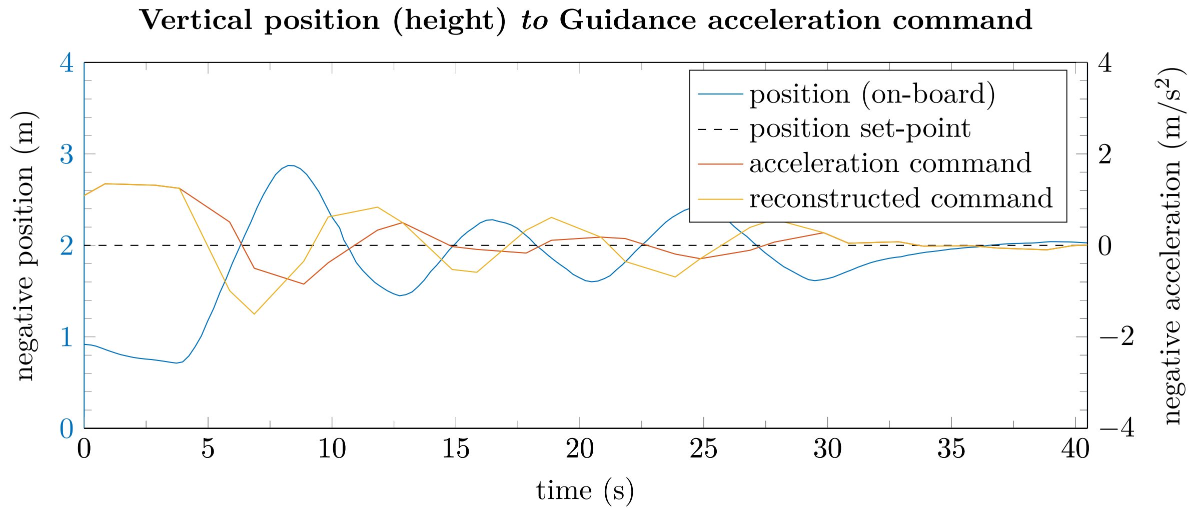


Figure 9.4.: Comparing vertical position z_w and vertical guidance acceleration command $\ddot{H}^c$ as well as the *a posteori* reconstructed acceleration command $\ddot{H}^{c\prime}$.

From the settling response in vertical position, the guidance control and the underlying speed-thrust control seem to counteract each other, resulting into an under-damped system behaviour. Fig. 9.4 shows the vertical acceleration command $\ddot{H}^c$ of the guidance control in comparison to the actual vertical position z_w: position and acceleration command are obviously misaligned as their extrema rather coincide and the acceleration command zeroes only if the set-point position is reached. This is matching the observed system's behaviour in position lacking the desired damping. *A posteori*, the vertical guidance acceleration command has been reconstructed from the vertical position and velocity measured and is shown in Fig. 9.4, too. In opposite to the *recorded* command, the reconstructed command $\ddot{H}^{c\prime}$ meets the trajectory required for a second-order system: $\ddot{H}^{c\prime}$ crosses zero when z_w is approaching z_w^{SP}, *i.e.* before the set-point is reached, so the system can be de-accelerated in time.

This discrepancy of recorded and reconstructed acceleration command was caused by an integer calculation error in the on-board flight control implementation: the relevant programming code implementing the guidance control laws as described in Sec. 6.2 is given by

```
acc_cmd = err_pos*kgain + err_vel*dgain;
```

where the position and velocity error as well as the acceleration command is stored with their respective mantissas given by Tab. 8.1; the gains have a mantissa of 14 each.

We are able to show by re-calculation that due to the mantissas of gains and velocity, for rather high errors in velocity an integer overflow was likely to cause a random command: reconstructing the overflow of 32-bit binary fixed-point representation in MATLAB, given the mantissas above, we created models of the erroneous closed-loop behaviour as well as the desired, accurate closed-loop.

In Fig. 9.5 we compare the settling response of a wind-tunnel flight with the desired closed-loop behaviour (*ideal closed-loop*) and taking into account integer overflows (*erroneous closed-loop*). Clearly, the response of the erroneous model is similar to the behaviour observed in the wind-tunnel; the erroneous model is even unmuted here and the settling of the real system was rather damped due to drag effects.

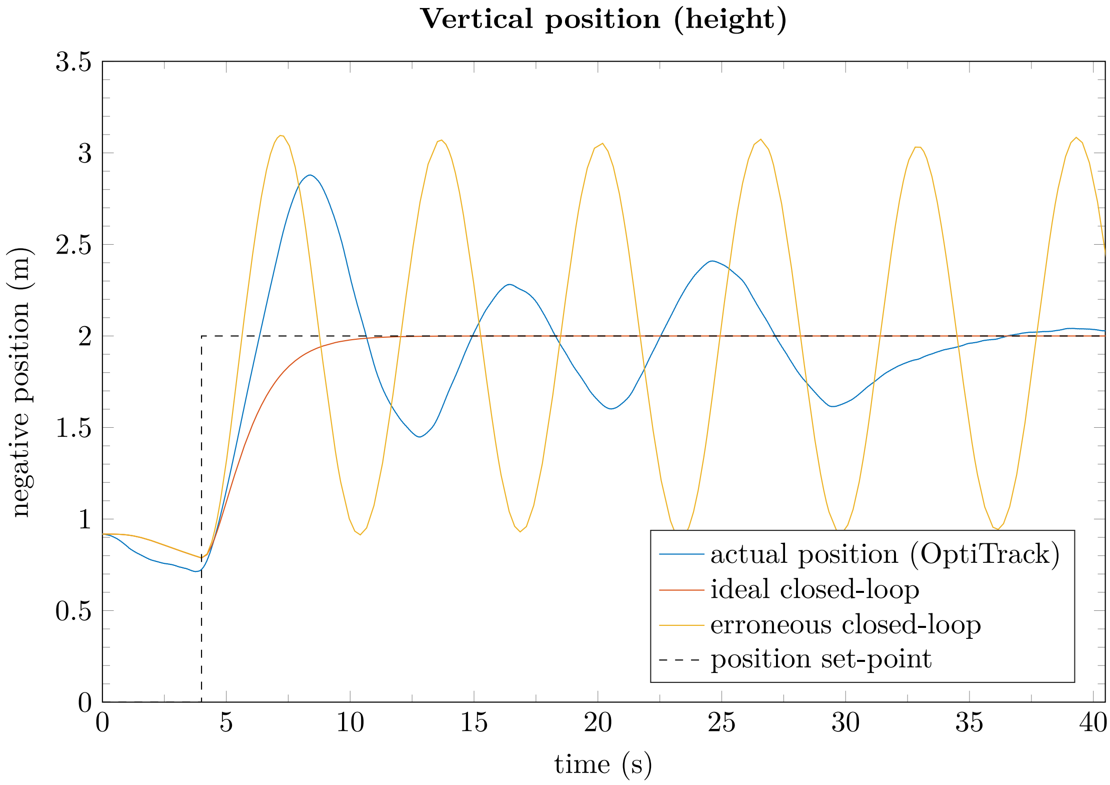


Figure 9.5.: Comparing the settling response in the wind-tunnel with an ideal and the erroneous closed-loop model.

The system's response after correction is discussed in the end of this chapter.

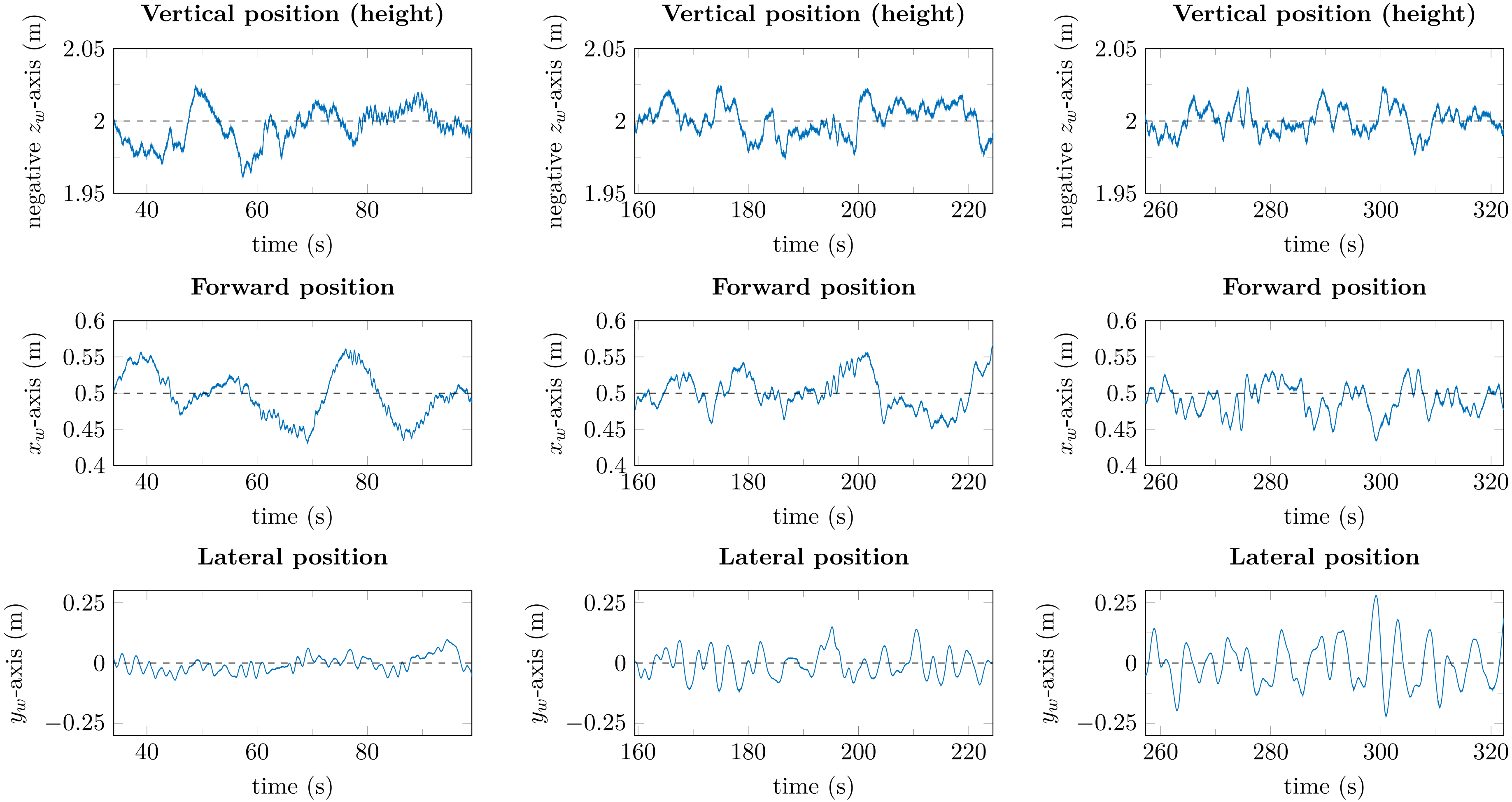
9.3. Robustness to wind speed

Eventually, we tested our control approach for different wind speeds as well. While flying, we increased the wind speed set-point V_W^{SP} stepwise from $0.7\,\frac{\mathrm{m}}{\mathrm{s}}$ up to $1.3\,\frac{\mathrm{m}}{\mathrm{s}}$; neither feed-forward nor feedback gains of the control hierarchy were changed during this test. The results of the Delfly flying at different wind speeds – *i.e.* with different air-speed, too – are shown in Fig. 9.6a to 9.6c on page 59.

In the vertical and forwards axes, the precision in steady-state is clearly unaffected by the change of the wind speed. That is, our control approach is robust to the speed of the wind.

The lateral axis, however, gets worse for increasing wind speed; while the precision is below 10 cm for lower speed, it deteriorates to 25 cm and worse for higher speed. This is caused by the decreased pitch angle in order to obtain higher air speed: first, rudder deflections are more effective here and couplings between yaw and roll are introduced; second, for a constant heading angle the lateral component of the air speed is similarly increased; thus, the lateral control loop as third-order integrator system from yaw moment to position is more sensitive to air speed.

In order to improve the lateral control outcome for changing wind speed, the lateral guidance control as well needs to take into account the air speed as the attitude control must be aware of the pitch angle, for example by gain scheduling, dynamic inversion control, or adaptive approaches.



(a) Wind speed $V_W^{SP} = 0.7\,\frac{\mathrm{m}}{\mathrm{s}}$.

(b) Wind speed $V_W^{SP} = 1.0\,\frac{\mathrm{m}}{\mathrm{s}}$.

(c) Wind speed $V_W^{SP} = 1.3\,\frac{\mathrm{m}}{\mathrm{s}}$.

Figure 9.6.: Results for different wind speeds; in the axes of the wind-tunnel reference system. (*semi-adaptive speed-thrust control with* $k = 0$, $i = 200\,\%$.)

9.4. Step responses (*in postero*)

After correction of the flight control implementation as aforementioned, we tested the step response of the resulting system *in postero*, that is afterwards and outside the wind-tunnel. Instead, the Delfly was flying in the Institute's flight area, emulating the conditions of the wind-tunnel as far as possible. In order to test the height step response, the Delfly flew in constant circles with radii of 2 m controlling chiefly height. However, the precision of the flight controller is slightly worse for a couple of reasons: first, the air-flow outside the wind-tunnel is less laminar; second, due to the circular trajectory, the Delfly's attitude is unsteady as the changing heading set-point implies extensive steering of the rudder.

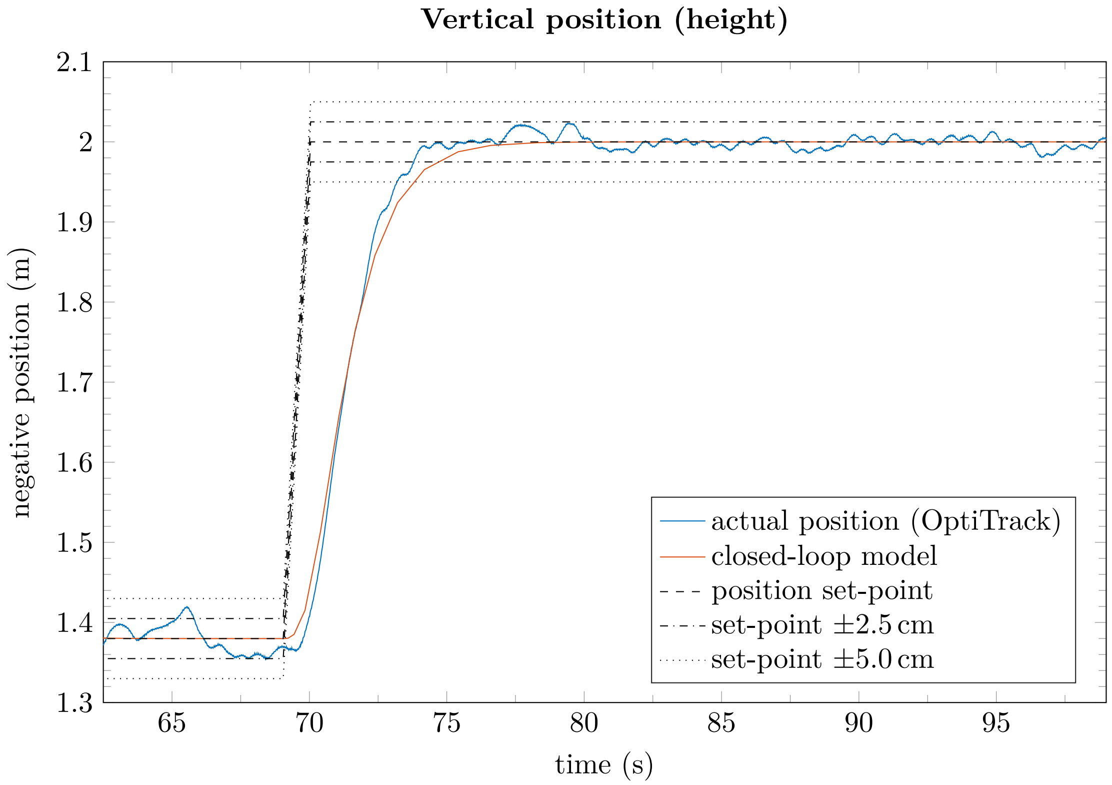


Figure 9.7.: Height step response with respect to set-point and closed-loop model.

Fig. 9.7 shows the system response to a height step in flight. Obviously, there is no overshoot beyond the accuracy at all. The system response is furthermore compared to the step response of the closed-loop model, that is the response of the second-order system obtained by guidance control, Eq. 6.28, given an ideal speed-thrust control where

$$\ddot{H} = \ddot{H}^{SP} \tag{9.1}$$

i.e. the acceleration command of the guidance control would be exactly achieved by the speed-thrust controller. By comparing the closed-loop model and the actual step response in Fig. 9.7, we can state that the speed-thrust control approach is realizing the vertical acceleration set-point almost perfectly, even for major changes in the input due to a height step.

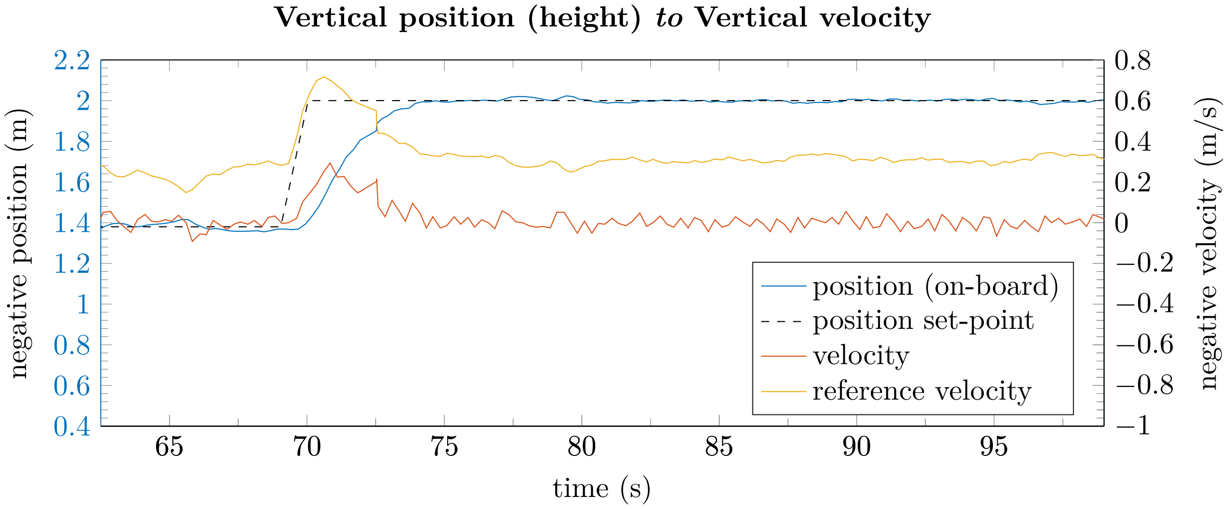


Figure 9.8.: Comparing vertical position z_w, vertical velocity w_w, and vertical speed-thrust reference velocity $\tilde{w_w}^{ref}$.

For the sake of completeness, the height step response with respect to the vertical velocity is shown in Fig. 9.8. In contrast to the response in the wind-tunnel, the vertical velocity is increased by the height step until the position set-point is approached half-way and decreased afterwards. As soon as the position set-point is reached, the vertical velocity remains about zero. All the time, the actual and reference velocity establish a certain offset equivalent to the integrated vertical acceleration error fed-back by the speed-thrust controller.

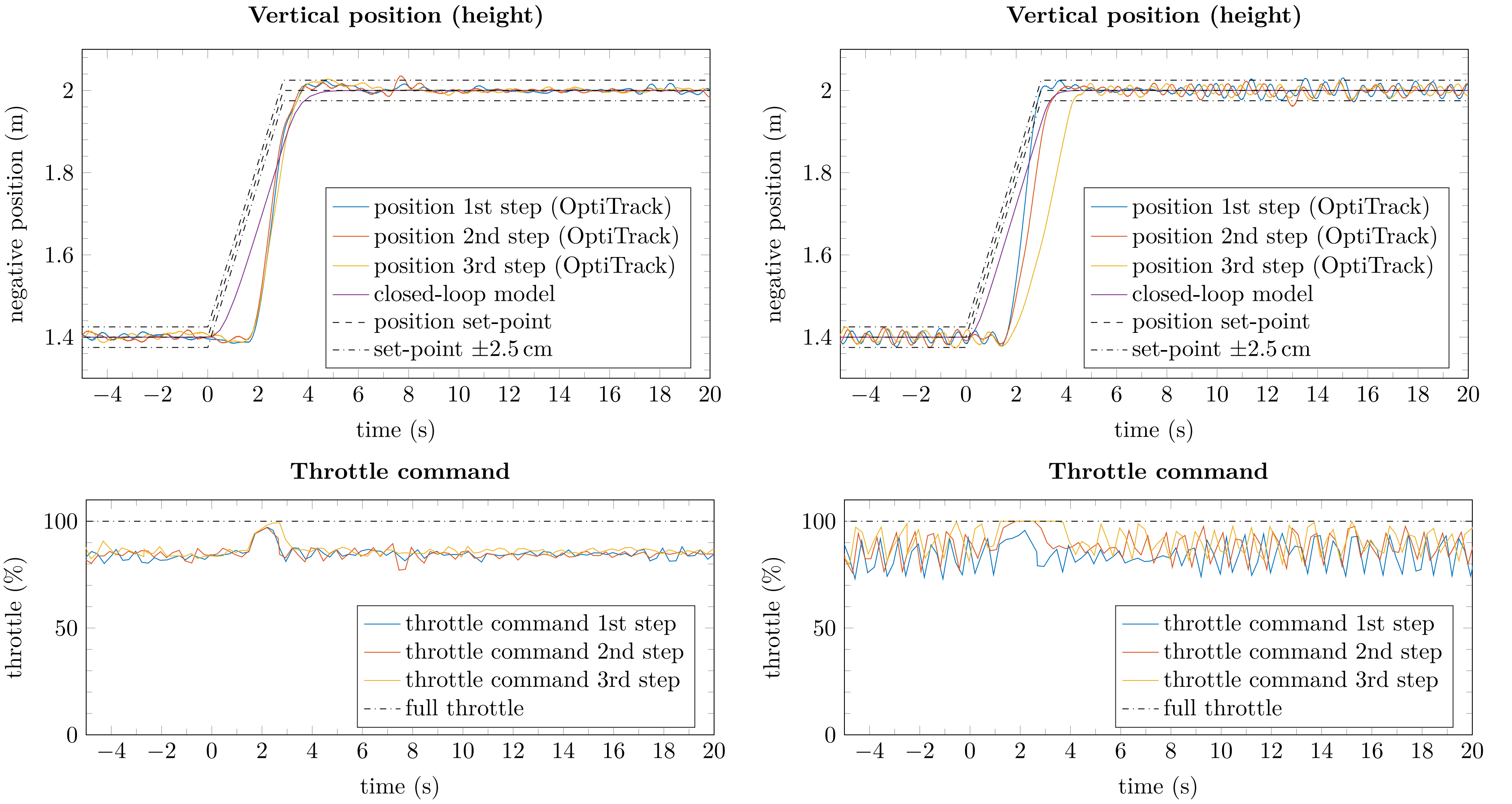
Influence of guidance poles Eventually, we tested the influence of different poles to the guidance control to the vertical step reponse: the vertical guidance control law is given by Eq. 6.22,

$$-\ddot{H}^c = \nu_z = d_{\ddot{H}z_w}\left(\dot{z}_w^{SP} - \dot{z}_w\right) + k_{\ddot{H}z_w}\left(z_w^{SP} - z_w\right)$$

where the gains $d_{\ddot{H}z_w}, k_{\ddot{H}z_w}$ are determined by the desired poles of the closed loop system $\varsigma_{z1,2}$ to (cf. Eq. 6.28)

$$\begin{aligned} d_{\ddot{H}z_w} &= -\varsigma_{z1} - \varsigma_{z2} \\ k_{\ddot{H}z_w} &= \varsigma_{z1}\varsigma_{z2} \end{aligned}$$

Desiring a “smooth” approach without overshot, we chose pole pairs on the left-hand real axis, namely $\varsigma_{z1,2} \in \{-3, -5\}$. The step responses for these pole pairs are compared in Fig. 9.10 on page 63.



(a) Vertical position and throttle command for $\varsigma_{1,2} = -3$.

(b) Vertical position and throttle command for $\varsigma_{1,2} = -5$.

Figure 9.9.: Height step responses in position z_w and throttle command F^c for different guidance poles, $\varsigma_{z1,2} \in \{-3, -5\}$.

As discussed, the initial pole pair $\varsigma_{z1,2} = -1$ shows no overshot and the underlying speed-thrust controlled system is able to follow the commanded acceleration well. For the pole pairs $\varsigma_{z1,2} = -3$ (Fig. 9.9a) and $\varsigma_{z1,2} = -5$ (Fig. 9.9b), there is a slight overshot within the accuracy of ±2.5 cm and the system responds as fast as the desired closed system; on the other hand, there is a notable deadband of the step response of about 2 s and 2.5 s for pole pairs $\varsigma_{z1,2} = -3$ and $\varsigma_{z1,2} = -5$, respectively, indicating a delayed reaction of the integral feedback of the speed-thrust controller to the increased acceleration command of the guidance. Apparently, the underlying controller is not able to achieve the acceleration commands of higher closed-loop guidance poles in the feed-forward instance. In feedback, that is delayed, the speed-thrust controller increases the acceleration *a fortiori*.

Furthermore, increasing poles introduces an oscillation of the Delfly's position around the set-point – still within a desirable accuracy. This is expectable as due to the increased gains the guidance control command reacts more drastic to errors both in position and velocity and, the while, both errors are affected by the flapping-wing oscillations of the body.

In the later steps of Fig. 9.9b, in the last step in particular, the effects of a drained battery are observable: here, the slopes are less step than the first – the system is not able to keep pace anymore but the throttle command is obviously saturated during the step response. In the first step however, the command stays unsaturated. One can also notice an overall increased throttle command for the different steps.

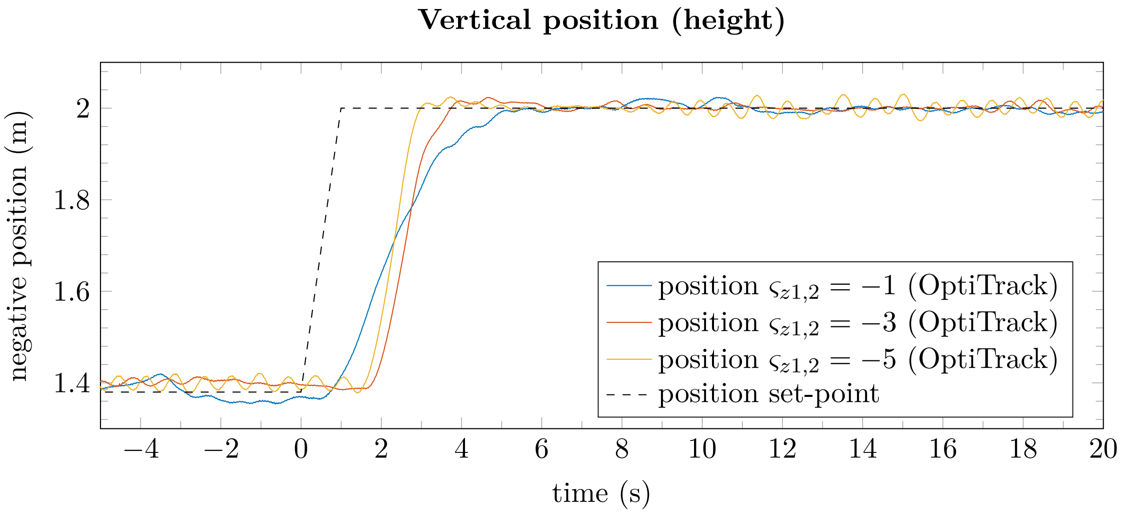


Figure 9.10.: Height step response for different guidance poles, $\varsigma_{z1,2} \in \{-1, -3, -5\}$.[1]

Finally, in Fig. 9.10 we compare the step responses for the different poles directly: for higher poles the steep of the response is increased but so is the deadband, too. Although, the response is faster overall, as one would expect for higher poles.

[1] For $\varsigma_{z1,2} \in \{-3, -5\}$, the height set-point at $t < 0$ has been $H^{SP} = 1.4$ m.

10. Conclusion

In this thesis, we have formulated the control problem to precisely control position of a flapping-wing micro air vehicle in the wind-tunnel. In particular, precise position control is required in order to investigate the aerodynamics of the Delfly II MAV using particle image velocimetry; however, as presented here, the Delfly is a challenging platform to control, since there is no accurate model valid over the full flight envelope.

Therefore, we discussed usual control techniques and presented a hierarchical control approach, combining basic and advanced flight control techniques. Outer-most, there is a dynamic-inverting guidance block to approach and track the position set-point. An intermediate semi-adaptive, combined forward-backward controller based but not relying on static force measurements was designed to achieve commanded accelerations in the longitudinal plane, which has been newly presented here. A classic PID basic controller, well-established in flight control, takes care of the inner-most attitude and stabilization loop.

We implemented our control hierarchy within the open-source paparazzi UAV autopilot software, re-using its attitude control laws after tuning. Eventually, we tested the Delfly position control approach in the TU Delft Open Jet Facility, a low-speed wind-tunnel with an open test section, allowing us to interact with the MAV. While the precision in steady-state has been very satisfying, the position control showed a high overshot in vertical and forward position which we deduced to an integer calculation; after correction, we showed that our control approach is actually able to maintain a given second-order position dynamic. Furthermore we demonstrated that the controller is robust to disturbances and changing wind speed, too.

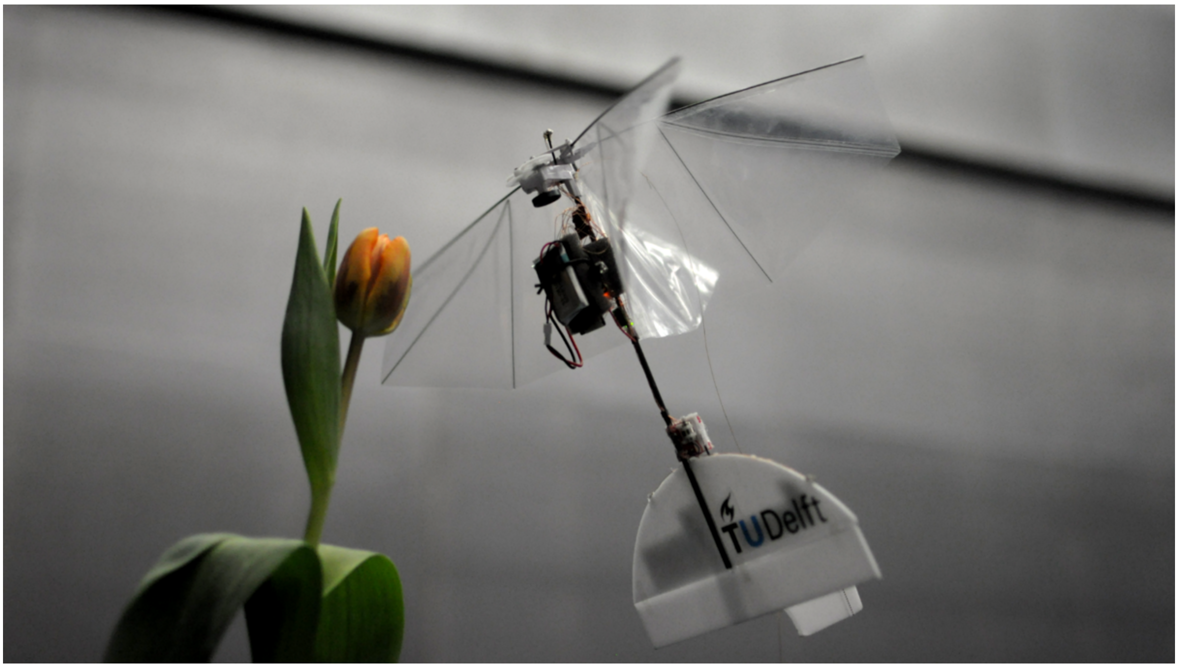
Thus, we performed the first high-precision flight of a flapping-wing micro air vehicle in a wind-tunnel, maintaining an accuracy of $\pm 5\,\mathrm{cm}$ in all axes at worst; in the vertical axis in particular, we achieved an accuracy of $\pm 2.5\,\mathrm{cm}$ and below, up to $\pm 1\,\mathrm{cm}$ for tens of seconds. This performance is over-fulfilling the requirements of PIV in position accuracy and duration, promising future investigations of the Delfly aerodynamics.

In order to solve the formulated control problem, we realised an elaborated control approach though not relying on a closed-loop model of neither a single flight condition nor the full envelope. A compatible approach for flapping-wing MAV has not been presented in literature yet. Therewith we are now able to control the forward and vertical acceleration of the MAV in closed-loop directly. Likewise, the control implementation is build modular and thus easy to adapt and extend, allowing future improvements as well as re-employments in different applications.

On the other hand, we still propose future work on the control hierarchy: First and foremost, the attitude control should be re-designed in order to suit the special needs of a flapping-wing MAV; in particular, for heading control the effectiveness of the rudder due to the actual angle-of-attack of the Delfly needs to be considered. In the guidance control, second, the horizontal guidance should take into account both heading angle and motion in body x_f-axis rather than separately control forward and lateral position. Finally, the semi-adaptive speed-thrust controller should be evolved to a fully adaptive control approach measuring and inverting the true equilibrium and aerodynamic gradient in-flight for the respective flight condition.

Besides control, the performance of the closed system can be increased by improving the on-board state data; using the *User Data Protocol* (UDP/IP) to uplink the tracked position instead of TCP, *i.e.* there is no re-sent of “old,” unreceived position messages, allows a higher frequency of the position uplink and thus of the control loop execution. Additionally, the off-board tracked data can be augmented by on-board acceleration measurements, and the Delfly’s attitude can be tracked off-board, too. Suitable filter methods like complementary or Kalman filtering then ensure a sophisticated estimation of the “true” states.

Summarising, we claim to solve the flapping-wing position control problem successfully and, furthermore, in a new and advanced manner. While our results in accuracy and performance raised the bar to be competed by future work, the modular design of the control hierarchy and its implementation benefit further improvements as well.



The Delfly II MAV approaching a tulip.

Standards

[DIN 9300-1] *Begriffe, Größen und Formelzeichen der Flugmechanik: Bewegung des Luftfahrzeuges gegenüber der Luft, ISO 1151-1 : 1988 modifiziert.* DIN 9300-1. Berlin, DE: Normenstelle Luftfahrt im DIN Deutsches Institut für Normung e.V., 1990.

[DIN 9300-2] *Begriffe, Größen und Formelzeichen der Flugmechanik: Bewegung des Luftfahrzeuges und der Atmosphäre gegenüber der Erde, ISO 1151-2 : 1987 (Stand 1985) modifiziert.* DIN 9300-2. Berlin, DE: Normenstelle Luftfahrt im DIN Deutsches Institut für Normung e.V., 1990.

[DIN IEC 60050-351] *Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik (IEC 60050-351:2013).* DIN IEC 60050-351. Berlin, DE: DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN Deutschen Institut für Normung e.V. und VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V., 2014.

[UML 2.5] *OMG Unified Modeling Language (OMG UML) Version 2.5.* formal/2015-03-01. Object Management Group, 2015. URL: `http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF`.

References

[1] Rudolf Brockhaus. *Flugregelung.* 2nd, revis. Berlin, DE: Springer, 2001.

[2] Gustav Schwab. *Griekse Mythen en Sagen.* translated from the German by J. K. van den Brink. Utrecht, NL: Het Spectrum, 1970, pp. 88–90.

[3] National Air and Space Museum. *Leonardo da Vinci's Codex on the Flight of Birds.* URL: `http://airandspace.si.edu/exhibitions/codex/codex.cfm` (visited on Nov. 7, 2015).

[4] V P Zubov. *Leonardo da Vinci.* translated from the Russian by David H. Kraus. Cambridge, US-MA: Harvard University Press, 1968.

[5] John W R Taylor and Kenneth Munson. *History of Aviation.* London, GB: New English Library, 1975.

[6] Thomas C Parramore. *First to Fly: North Carolina and the Beginning of Aviation.* Chapel Hill, US-NC: The University of North Carolina Press, 2002.

[7] R Douglas Archer and Maido Saarlas. *An Introduction to Aerospace Propulsion.* Upper Saddle River, US-NJ: Prentice-Hall, 1996.

[8] Thomas J Mueller and James D DeLaurier. „An Overview of Micro Air Vehicle Aerodynamics". In: *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications.* Ed. by Thomas J Mueller. Vol. 195. Progress in Astronautics and Aeronautics. Reston, US-VA: American Institute of Aeronautics and Astronautics, 2001. Chap. 1, pp. 1–10.

[9] David E Alexander. „Wind Tunnel Studies of Turns by Flying Dragonflies". In: *The Journal of Experimental Biology* 122 (1986), pp. 81–98.

[10] J J Videler et al. „Indoor Flight Experiments with Trained Kestrels II: The Effect of Added Weight on Flapping Flight Kinematics”. In: *The Journal of Experimental Biology* 134 (1988), pp. 185–199.

[11] C J Pennycuick et al. „Wingbeat Frequency and the Body Drag Anomaly: Wind-tunnel Observations on a Thrush Nightingale (*luscinia luscinia*) and a Teal (*anas crecca*)”. In: *The Journal of Experimental Biology* 199 (1996), pp. 2757–2765.

[12] Adrian L R Thomas and Graham K Taylor. „Animal Flight Dynamics I: Stability in Gliding Flight”. In: *Journal of Theoretical Biology* 212 (2001), pp. 399–424.

[13] Graham K Taylor and Adrian L R Thomas. „Animal Flight Dynamics II: Longitudinal Stability in Flapping Flight”. In: *Journal of Theoretical Biology* 214 (2002), pp. 351–370.

[14] Sridhar Ravi et al. „Hummingbird Flight Stability and Control in Freestream Turbulent Winds”. In: *Journal of Experimental Biology* 218.9 (2015), pp. 1444–1452.

[15] Che-Shu Lin, Chyanbin Hwu, and Wen-Bin Young. „The Thrust and Lift of an Ornithopter’s Membrane Wings with Simple Flapping Motion”. In: *Aerospace Science and Technology* 10 (2006), pp. 111–119.

[16] John M Dietl and Ephrahim Garcia. „Stability in Ornithopter Longitudinal Flight Dynamics”. In: *Journal of Guidance, Control, and Dynamics* 31.4 (2008), pp. 1157–1162.

[17] G Iosilevskii. „Forward Flight of Birds Revisited – Part 1: Aerodynamics and Performance”. In: *Royal Society Open Science* 1 (2014).

[18] J D DeLaurier and J M Harris. „A Study of Mechanical Flapping-wing Flight”. In: *The Aeronautical Journal* (1993), pp. 277–286.

[19] J D DeLaurier. „An Aerodynamic Model for Flapping-wing Flight”. In: *The Aeronautical Journal* (Apr. 1993), pp. 125–130.

[20] Joydeep Bhowmik, Debopam Das, and Saurav Kumar Gosh. „Aerodynamic Modelling of Flapping Flight Using Lifting Line Theory”. In: *International Journal of Intelligent Unmanned Systems* 1.1 (2013), pp. 36–61.

[21] A N Brooks et al. „Development of a Wing-Flapping Flying Replica of the Largest Pterosaur”. In: *21st Joint Propulsion Conference*. American Institute of Aeronautics and Astronautics. Monterey, US-CA, July 1985, pp. 59–67.

[22] Thomas J Mueller. „On the Birth of Micro Air Vehicles“. In: *International Journal of Micro Air Vehicles* 1.1 (2009), pp. 1–12.

[23] Emmanuel de Margerie et al. „Flapping-wing Flight in Bird-sized UAVs for the ROBUR Project: from an Evolutionary Optimization to a Real Flapping-wing mechanism“. In: *3rd US-European Competition and Workshop on Micro Air Vehicle Systems & European Micro Air Vehicle Conference and Flight Competition*. Toulouse, FR, Sept. 2007.

[24] T Nakata et al. „Aerodynamics of a Bio-inspired Flexible Flapping-wing Micro Air Vehicle“. In: *Bioinspiration and Biomimetics* 6 (2011).

[25] Lindsey L Hines, Domenico Campolo, and Metin Sitti. „Liftoff of a Motor-Driven, Flapping-Wing Microaerial Vehicle Capable of Resonance“. In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 220–232.

[26] Thomas J Mueller, ed. *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applicatios*. Vol. 195. Progress in Astronautics and Aeronautics. Reston, US-VA: American Institute of Aeronautics and Astronautics, 2001.

[27] Kevin Y Ma et al. „Controlled Flight of a Biologically Inspired, Insect-Scale Robot“. In: *Science* 340.6132 (2013), pp. 603–607.

[28] Pakpong Chirarattananon, Kevin Y Ma, and Robert J Wood. „Adaptive Control for Takeoff, Hovering, and Landing of a Robotic Fly“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Institute of Electrical and Electronics Engineers. Tokyo, JP, Nov. 2013, pp. 3808–3815.

[29] Pierre-Emile J Duhamel et al. „Biologically Inspired Optical-Flow Sensing for Altitude Control of Flapping-Wing Microrobots“. In: *IEEE/ASME Transactions of Mechatronics* 18.2 (2013), pp. 556–568.

[30] Matthew Keennon et al. „Development of the Nano Hummingbird: A Tailless Flapping Wing Micro Air Vehicle“. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics. Nashville, US-TN, Jan. 2012.

[31] Guido C.H.E. de Croon et al. „Design, Aerodynamics, and Vision-based Control of the DelFly“. In: *International Journal of Micro Air Vehicles* 1.2 (2009), pp. 71–97.

[32] Guido C.H.E. de Croon et al. *The DelFly: Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot.* Dordrecht, NL: Springer, 2016.

[33] Graham K Taylor and Adrian L R Thomas. „Dynamic Flight Stability in the Desert Locust *schistocerca gregaria*". In: *The Journal of Experimental Biology* 206 (2003), pp. 2803–2829.

[34] Graham K Taylor and Rafał Żbikowski. „Nonlinear Time-periodic Models of the Longitudinal Flight Dynamics of Desert Locusts *schistocerca gregaria*". In: *Journal of the Royal Society Interface* 2 (2005), pp. 197–221.

[35] K D Jones et al. „Bio-inspired Design of Flapping-wing Micro Air Vehicles". In: *The Aeronautical Journal* (2005), pp. 385–393.

[36] Hui Hu et al. „An Experimental Investigation on the Aerodynamic Performances of Flexible Membrane Wings in Flapping Flight". In: *Aerospace Science and Technology* 14 (2010), pp. 575–586.

[37] Jun-Seong Lee and Jae-Hung Han. „Experimental Study on the Flight Dynamics of a Bioinspired Ornithopter: Free Flight Testing and Wind Tunnel Testing". In: *Smart Materials and Structures* 21 (2012).

[38] Cameron Rose and Ronald S Fearing. „Comparison of Ornithopter Wind Tunnel Force Measurements with Free Flight". In: *IEEE International Conference on Robotics and Automation.* Institute of Electrical and Electronics Engineers. Hong Kong, CN, 2014, pp. 1816–1821.

[39] Thomas J Mueller. „Aerodynamic Measurements at Low Reynolds Numbers for Fixed Wing Micro-Air Vehicles". In: *RTO Applied Vehicle Panel Special Course on Development and Operation of UAVs for Military and Civil Applications.* NATO Research and Technology Organisation. Saint-Genèse, BE, Sept. 1999.

[40] C Thipyopas and N Intaratep. „Aerodynamics Study of Fixed-Wing MAV: Wind Tunnel and Flight Test". In: *Proceedings of the International Micro Air Vehicles conference 2011 summer edition.* Delft, NL, Sept. 2011, pp. 100–107.

[41] R Albertani et al. „Aerodynamic Coefficients and Deformation Measurements on Flexible Micro Air Vehicle Wings". In: *Experimental Mechanics* 47 (2007), pp. 625–635.

[42] Jared Grauer, Evan Ulrich, and James Hubbard Jr. „Testing and System Identification of an Ornithopter in Longitudinal Flight”. In: *Journal of Aircraft* 48.2 (2012), pp. 660–667.

[43] C J Pennycuick, Thomas Alerstam, and Anders Hedenström. „A New Low-turbulence Wind Tunnel for Bird Flight Experiments at Lund University, Sweden”. In: *The Journal of Experimental Biology* 200 (1997), pp. 1441–1449.

[44] Kirsty J Park, Mikael Rosén, and Anders Hedenström. „Flight Kinematics of the Barn Swallow (*hirundo rustica*) Over a Wide Range of Speeds in a Wind Tunnel”. In: *The Journal of Experimental Biology* 204 (2001), pp. 2741–2750.

[45] C W Pitt Ford and H Babinsky. „Lift and the Leading-edge Vortex”. In: *Journal of Fluid Mechanics* 720 (2013), pp. 280–313.

[46] Per Henningsson et al. „The complex aerodynamic footprint of desert locusts revealed by large-volume tomographic particle image velocimetry.” In: *Journal of the Royal Society Interface* 12.108 (2015).

[47] P Henningsson, G R Spedding, and A Hedenström. „Vortex Wake and Flight Kinematics of a Swift in Cruising Flight in a Wind Tunnel”. In: *The Journal of Experimental Biology* 211 (2008). *errata ibid.* [99], pp. 717–730.

[48] Roi Gurka et al. „PIV-based Study of the Gliding Osprey Aerodynamics in a Wind Tunnel”. In: *62nd Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society. Minneapolis, US-MN, Nov. 2009.

[49] Hadar Ben-Gida et al. „Estimation of Unsteady Aerodynamics in the Wake of a Freely Flying European Starling (*sturnus vulgaris*)”. In: *PLoS ONE* 8.11 (2013). Ed. by Assad Anshuman Oberai.

[50] Florian T Muijres et al. „Leading Edge Vortices in Lesser Long-nosed Bats Occurring At Slow But Not Fast Flight Speeds”. In: *Bioinspiration and Biomimetics* 9 (2014).

[51] K D Jones, S J Duggan, and M F Platzer. „Flapping-Wing Propulsion for a Micro Air Vehicle”. In: *39th Aerospace Sciences Meeting & Exhibit*. American Institute of Aeronautics and Astronautics. Reno, US-NV, Jan. 2001.

[52] Peter G Ifju et al. „Flexible-Wing-Based Micro Air Vehicles”. In: *40th AIAA Aerospace Sciences Meeting & Exhibit*. American Institute of Aeronautics and Astronautics. Reno, US-NV, Jan. 2002, pp. 1–11.

[53] Dae-Kwan Kim, Jae-Hung Han, and Ki-Jung Kwon. „Wind Tunnel Tests for a Flapping Wing Model with a Changeable Camber using Macro-fiber Composite Actuators". In: *Smart Materials and Structures* 18 (2009).

[54] Kristien M E De Clercq et al. „Flow Visualization and Force Measurements on a Hovering Flapping-Wing MAV 'DelFly II' ". In: *Proceedings of the 39th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics. San Antonio, US-TX, June 2009.

[55] Kristien M.E. De Clercq et al. „Aerodynamic Experiments on DelFly II: Unsteady Lift Enhancement". In: *International Journal of Micro Air Vehicles* 1.4 (2009), pp. 255–262.

[56] M. Percin et al. „Three-dimensional vortex wake structure of a flapping-wing micro aerial vehicle in forward flight configuration". In: *Experiments in Fluids* 55.1806 (2014).

[57] Shuanghou Deng, Mustafa Percin, and Bas van Oudheusden. „Experimental Investigation of Aerodynamics of Flapping-Wing Micro-Air-Vehicle by Force and Flow-Field Measurements". In: *AIAA Journal* 54.2 (2016), pp. 588–602.

[58] J V Caetano et al. „Controlled Flight Maneuvers of a Flapping Wing Micro Air Vehicle: a Step Towards the Delfly II Identification". In: *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics. Boston, US-MA, Aug. 2013.

[59] J V Caetano et al. „Linear Aerodynamic Model Identification of a Flapping Wing MAV Based on Flight Test Data". In: *International Journal of Micro Air Vehicles* 5.4 (2013), pp. 273–286.

[60] S F Armanini et al. „Time-varying model identification of flapping-wing vehicle dynamics using flight data". In: *Journal of Guidance, Control, and Dynamics* (2015).

[61] Daniel G Murri, Luat T Nguyen, and Sue B Grafton. *Wind-Tunnel Free-Flight Investigation of a Model of a Forward-Swept-Wing Fighter Configuration*. NASA Technial Paper 2230. Hampton, US-VA: Langley Research Center, 1984.

[62] E Bruce Jackson and C W Buttrill. *Control Laws for a Wind Tunnel Free-Flight Study of a Blended-Wing-Body Aircraft*. NASA Technical Memorandum NASA/TM-2006-214501. Hampton, US-VA: Langley Research Center, 2006.

[63] Jan Nowak. „Windkanal-Freiflugmessungen zur Bestimmung flugmechanischer Kenngrößen". PhD thesis. Aachen, DE: RWTH Aachen University, 2010.

[64] James Neil Wiken. „Analysis of a Quadrotor in Forward Flight". Master's thesis. Cambridge, US-MA: Massachusetts Institute of Technology, 2015.

[65] J Andries Koopmans. „Delfly Freeflight – Autonomous Flight of the Delfly in the Wind Tunnel using Low-Cost Sensors". Master's thesis. Delft, NL: Delft University of Technology, 2012.

[66] Christophe De Wagter et al. „Autonomous Wind Tunnel Free-Flight of a Flapping Wing MAV". In: *Advances in Aerospace Guidance, Navigation and Control.* Berlin, DE: Springer, 2013, pp. 603–621.

[67] Haithem E Taha, Muhammad R Hajj, and Ali H Nayfeh. „Flight Dynamics and Control of Flapping-wing MAVs: a Review". In: *Nonlinear Dynamics* 70 (2012), pp. 907–939.

[68] Michael L Anderson and Richard G Cobb. „Toward Flapping Wing Control of Micro Air Vehicles". In: *Journal of Guidance, Control, and Dynamics* 35.1 (2012), pp. 296–308.

[69] Xinyan Deng, Luca Schenato, and S Shankar Sastry. „Flapping Flight for Biomimetic Robotic Insects: Part II – Flight Control Design". In: *IEEE Transactions on Robotics* 22.4 (2006), pp. 789–803.

[70] Zaeem A Khan and Sunil K Agrawal. „Control of Longitudinal Flight Dynamics of a Flapping-Wing Micro Air Vehicle Using Time-Averaged Model and Differential Flatness Based Controller". In: *Proceedings of the 2007 American Control Conference.* Institute of Electrical and Electronics Engineers. New York, US-NY, July 2007, pp. 5284–5289.

[71] David B Doman, Michael W Oppenheimer, and David O Sigthorsson. „Wingbeat Shape Modulation for Flapping-Wing Micro-Air-Vehicle Control During Hover". In: *Journal of Guidance, Control, and Dynamics* 33.3 (2010), pp. 724–739.

[72] David O Sigthorsson, Michael W Oppenheimer, and David B Doman. „Flapping Wing Micro-Air-Vehicle 4-DOF Controller Applied to a 6-DOF Model". In: *AIAA Guidance, Navigation, and Control Conference.* American Institute of Aeronautics and Astronautics. Toronto, CA, Aug. 2010.

[73] Sawyer B Fuller et al. „Controlling Free Flight of a Robotic Fly Using an Onboard Vision Sensor Inspired by Insect Ocelli". In: *Journal of the Royal Society Interface* 11 (2014).

[74] Xinyan Deng, Luca Schenato, and Shankar S Sastry. „Model Identification and Attitude Control for a Micromechanical Flying Insect Including Thorax and Sensor Models". In: *Proceedings of the 2003 IEEE International Conference on Robotics & Automation.* Institute of Electrical and Electronics Engineers. Taipei, TW, Sept. 2003, pp. 1152–1157.

[75] John M Dietl and Ephrahim Garcia. „Ornithopter Control with Periodic Infinite Horizon Controllers". In: *Journal of Guidance, Control, and Dynamics* 34.5 (2011), pp. 1412–1422.

[76] Manav Bhatia et al. „LQR Controller for Stabilization of Flapping Wing MAVs in Gust Environment". In: *AIAA Atmospheric Flight Mechanics Conference.* American Institute of Aeronautics and Astronautics. Minneapolis, US-MN, Aug. 2012.

[77] C De Wagter et al. „Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System". In: *IEEE International Conference on Robotics and Automation.* Institute of Electrical and Electronics Engineers. Hong Kong, CN, 2014, pp. 4982–4987.

[78] Sjoerd Tijmons et al. „Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV". In: *IEEE Transactions on Robotics* (2016). submitted on April 4, 2016. arXiv: `arXiv:1604.00833v1`.

[79] Rudolf Brockhaus. *Flugregelung II: Entwurf von Regelsystemen.* Methoden der Regelungstechnik. München, DE: R. Oldenbourg Verlag, 1979.

[80] Declan Bates and Ian Postlethwaite. *Robust Multivariable Control of Aerospace Systems.* Control and Simulation. Delft, NL: Delft University Press, 2002.

[81] Kevin A Wise. „Applied Controls Research Topics in the Aerospace Industry". In: *Proceedings of the 34th Conference on Decision and Control.* Vol. 1. Institute of Electrical and Electronics Engineers. New Orleans, US-LA, Dec. 1995, pp. 751–756.

[82] B D W Remes et al. „Lisa-S 2.8g Autopilot for GPS-based Flight of MAVs”. In: *IMAV 2014: International Micro Air Vehicle Conference and Competition*. Delft, NL, Aug. 2014, pp. 280–285.

[83] Faculteit Luchtvaart- en Ruimtevaarttechniek. *Aerospace Engineering: The Open Jet Facility*. Magazine. Delft, NL: Technische Universiteit Delft, 2016.

[84] NaturalPoint. *OptiTrack – Flex 13 Camera*. Tech. rep. 2012.

[85] paparazzi UAV. *Ground Control Station*. 2015. URL: `http://wiki.paparazziuav.org/wiki/GCS` (visited on Apr. 11, 2016).

[86] Dirk Abel. *Umdruck zur Vorlesung Regelungstechnik und Ergänzugen (Höhere Regelungstechnik)*. 4th. Aachen, DE: Verlagshaus Mainz, 2013.

[87] P Cominos and N Munro. „PID controllers: recent tuning methods and design to specification”. In: *IEE Proceedings - Control Theory and Applications* 149.1 (2002), pp. 46–53. ISSN: 09168508.

[88] Sigurd Skogestad and Ian Postlethwaite. „Multivariable Feedback Control: Analysis and design”. In: (2005), p. 592.

[89] Karl Johan Åström and Björn Wittenmark. *Adaptive Control*. reprinted, Addison-Wesley Series in Electrical and Computer Engineering: Control Engineering. Boston, US-MA: Addison-Wesley, 1989.

[90] Bernard Widrow and Eugene Walach. *Adaptive Inverse Control*. Prentice Hall Information & System Sciences Series. Upper Saddle River, US-NJ: Prentice-Hall, 2008.

[91] Björn Wittenmark. „Adaptive Dual Control Methods: An Overview”. In: *5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*. International Federation of Automatic Control. Budapest, HU, 1995, pp. 67–72.

[92] Nikolai Michailovich Filatov and Heinz Unbehauen. *Adaptive Dual Control*. Lecture Notes in Control and Information Sciences. Berlin, DE: Springer, 2003.

[93] J. Andries Koopmans and Matěj Karásek. „Static force measurement of the Delfly flapping-wing MAV (*working title*)”. unpublished. Delft, NL, 2015.

[94] S Sieberling, Q P Chu, and J A Mulder. „Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction". In: *Journal of Guidance, Control, and Dynamics* 33.6 (2010), pp. 1732–1742.

[95] J Raisch and E D Gilles. „Regelungskonzepte". In: *Prozeßleittechnik*. Ed. by Martin Polke and Ulrich Epple. 2nd, revis. Munic, DE: R. Oldenbourg Verlag, 1994. Chap. 4.2.

[96] David Goldberg. „What every computer scientist should know about floating-point arithmetic". In: *ACM Computing Surveys* 23.1 (1991), pp. 5–48.

[97] Neil Kolban. *Kolban's Book on ESP8266*. Jan. 2016. 2016. URL: `http://neilkolban.com/tech/esp8266/`.

[98] JeeLabs. *esp-link: WiFi-Serial Bridge*. URL: `https://github.com/jeelabs/esp-link` (visited on May 5, 2016).

[99] P Henningsson, G R Spedding, and A Hedenström. „Vortex Wake and Flight Kinematics of a Swift in Cruising Flight in a Wind Tunnel". In: *The Journal of Experimental Biology – Corrigendum* 214 (2011), p. 697.

Appendix

A. Delfly System Identification

The quantities and dimensions of the Delfly II micro air vehicle are given by Tab. A.1. Below, we state the aerodynamic coefficients of the linear model.

Quantity	Variable	Value
mass	m	17.4 g
length	(l)	23.3 cm
wing span	(b)	27.5 cm

(a) Dimensions.

Quantity	Variable	Value
inertia	I_x	$0.10\,\mathrm{g}\cdot\mathrm{m}^2$
	I_y	$1.58\,\mathrm{g}\cdot\mathrm{m}^2$
	I_z	$0.96\,\mathrm{g}\cdot\mathrm{m}^2$

(b) Moments of inertia. *Caetano et al. 2013 [58], Tab. 1, "#2".*

Table A.1.: Data sheet of the Delfly II micro air vehicle.

A.1. Coefficients of the linear model

A linear model of the Delfly flight dynamic has been presented in [60] and the derived longitudinal and lateral equations of motion were recalled in Sec. 5.2. The aerodynamic coefficients of these equations are given by Tab. A.2a and A.2b, respectively.

	q	u	w	η
$M_{(\cdot)}$	$-1.03 \cdot 10^{-3}$	$+3.90 \cdot 10^{-3}$	$+2.59 \cdot 10^{-3}$	$-6.96 \cdot 10^{-3}$
$X_{(\cdot)}$	$+3.05 \cdot 10^{-3}$	$-3.39 \cdot 10^{-2}$	$+1.81 \cdot 10^{-2}$	$+2.53 \cdot 10^{-2}$
$Z_{(\cdot)}$	$-1.31 \cdot 10^{-2}$	$-3.21 \cdot 10^{-2}$	$-7.74 \cdot 10^{-2}$	$-9.67 \cdot 10^{-2}$

(a) Coefficients of the longitudinal equations. *Armanini et al. 2015 [60], Tab. 1.*

	p	r	v	ζ
$L_{(\cdot)}$	$-7.14 \cdot 10^{-6}$	$+3.09 \cdot 10^{-5}$	$-4.47 \cdot 10^{-5}$	$+6.89 \cdot 10^{-5}$
$N_{(\cdot)}$	$-1.98 \cdot 10^{-5}$	$-4.79 \cdot 10^{-4}$	$-1.45 \cdot 10^{-3}$	$-2.10 \cdot 10^{-3}$
$Y_{(\cdot)}$	$+6.54 \cdot 10^{-4}$	$-2.59 \cdot 10^{-3}$	$-9.92 \cdot 10^{-2}$	$-6.96 \cdot 10^{-2}$

(b) Coefficients of the lateral equations. *Armanini et al. 2015 [60], Tab. 4.*

Table A.2.: Aerodynamic coefficients of the longitudinal and lateral equations of motion.

B. Delfly Control Implementation

B.1. Speed-thrust control forward gains

Tab. B.1 gives the feed-forward gains of the speed-thrust control implementation, *i.e.* the inverted aerodynamic force derivatives and pitch and throttle equilibria.

V_W	$\hat{\Theta}_0$	$\hat{\digamma}_0$
$0.4\,\frac{\mathrm{m}}{\mathrm{s}}$	74.40°	90.33 %
$0.8\,\frac{\mathrm{m}}{\mathrm{s}}$	65.85°	86.83 %
$1.2\,\frac{\mathrm{m}}{\mathrm{s}}$	47.23°	78.00 %
$2.5\,\frac{\mathrm{m}}{\mathrm{s}}$	30.51°	68.48 %
$5.0\,\frac{\mathrm{m}}{\mathrm{s}}$	11.90°	71.39 %

(a) Pitch and throttle equilibria.

V_W	$\Delta\hat{\Theta}_{F_L}$	$\Delta\hat{\Theta}_{F_F}$	$\Delta\hat{\digamma}_{F_L}$	$\Delta\hat{\digamma}_{F_F}$
$0.4\,\frac{\mathrm{m}}{\mathrm{s}}$	0.06°/mN	−0.25°/mN	0.25 %/mN	−0.01 %/mN
$0.8\,\frac{\mathrm{m}}{\mathrm{s}}$	0.07°/mN	−0.18°/mN	0.25 %/mN	0.04 %/mN
$1.2\,\frac{\mathrm{m}}{\mathrm{s}}$	0.21°/mN	−0.30°/mN	0.25 %/mN	0.07 %/mN
$2.5\,\frac{\mathrm{m}}{\mathrm{s}}$	0.08°/mN	−0.11°/mN	0.19 %/mN	0.17 %/mN
$5.0\,\frac{\mathrm{m}}{\mathrm{s}}$	0.05°/mN	−0.04°/mN	0.10 %/mN	0.55 %/mN

(b) Inverted aerodynamic force derivatives.

Table B.1.: Estimated pitch and throttle equilibria and inverted aerodynamic force derivatives, based on Sec. 5.3

For their respective wind-velocities V_W the pitch and throttle equilibria are directly taken from Tab. 5.1a. The inverted aerodynamic force derivatives have been derived from the *aerodynamic force derivative matrix* $\mathbf{F}_{Æ}$ by

$$\begin{bmatrix} \Delta\hat{\Theta}_{F_F} & \Delta\hat{\Theta}_{F_L} \\ \Delta\hat{\digamma}_{F_F} & \Delta\hat{\Theta}_{F_L} \end{bmatrix} = \mathbf{F}_{Æ}^{-1}(V_W) \tag{B.1}$$

such that

$$\begin{bmatrix} \Delta\Theta \\ \Delta\digamma \end{bmatrix} = \mathbf{F}_{Æ}^{-1} \begin{bmatrix} F_F \\ F_L \end{bmatrix} = \mathbf{F}_{Æ}^{-1} \cdot m \begin{bmatrix} \dot{V}_A \\ \ddot{H} \end{bmatrix} \tag{B.2}$$

where m denotes the mass of the Delfly, and $\Delta\Theta$ and $\Delta\digamma$ are the change in pitch angle and throttle command, respectively, with respect to their equilibria. Thus, the inverted aerodynamic force derivatives contribute to the speed-thrust forward control law stated in Eq. 7.4.[1]

[1] Under the assumptions made in Chap. 7.

B.2. Attitude control feedback gains

The feedback gains of the paparazzi basic controller after tuning in the wind-tunnel are given by Tab. B.2.

gain	$k_{\eta\Theta}$	$i_{\eta\Theta}$	$d_{\eta\Theta}$
value	2024	800	200

(a) Pitch.

gain	$k_{\zeta\Psi}$	$i_{\zeta\Psi}$	$d_{\zeta\Psi}$
value	850	225	100

(b) Heading.

Table B.2.: Feedback gains of the basic controller with respect to pitch (a) and heading (b).

B.3. Settings of the paparazzi `DelflyControl` module

Tab. B.3 on page 82 shows the settings of the `DelflyControl` module implemented in paparazzi.

Setting	Variable	Unit	see...	Description
`lat_ratio`	–	$\frac{\mathrm{rad}}{\mathrm{m/s^2}}$	Eq. 6.31	Ratio between lateral acceleration pseudo-command ν_y and heading angle command Ψ^c.
`lat_i`	$i_{\Psi y_w}$	%	Eq. 6.35	Integrator feedback gain of the lateral guidance control law.
`ver_pole`	$\varsigma_{z1,2}$	–	Eq. 6.28	Pole-pair of the desired closed-loop vertical guidance.
`flap_gamma`	$\gamma_{\dot{\theta} f}$	%	Fig. 7.3	Adaptation rate of the adaptive flapping-frequency controller.
`flap_ratio`	$\theta_{F f}$	$\frac{\%}{\mathrm{Hz}}$	Fig. 7.3	Adaptable ratio between flapping frequency f and throttle command F^c.
`theta_off`	–	°	–	Pitch angle offset between body-fixed axis system $\left[x_f, y_f, z_f\right]$ and the internal paparazzi body frame.
`fb_fwd_kp` `fb_ver_kp`	k	%	Fig. 7.2	Proportional feedback gain of the speed-trust control law, with respect to forward and vertical acceleration.
`fb_fwd_ki` `fb_ver_ki`	i	%	Fig. 7.2	Integrator feedback gain of the speed-trust control law, with respect to forward and vertical acceleration.
`fb_adapt`	γ	%	Fig. 7.4a	Adaptation rate of the semi-adaptive speed-thrust controller (during the adaptation stage).
`type`	–	–	–	Type of the commanded variable of the speed-thrust controller, either throttle (`THROTTLE`) or flapping frequency (`FLAPFREQ`, via adaptive flapping frequency controller).
`wind_speed`	V_W	$\frac{\mathrm{m}}{\mathrm{s}}$	Fig. 2.7	Wind speed set-point to the wind-tunnel speed controller.
`est_mode`	–	–	–	Estimation mode of the implemented state filter (Sec. 8.3 describes `GPS_FILTER` mode).

Table B.3.: Settings of the `DelflyControl` module implemented in paparazzi.

C. Wind-tunnel Test Results

The results of the wind-tunnel flights for different gain settings, both settling and in steady-state, are presented here. We used the gain settings as shown in Tab. C.1.

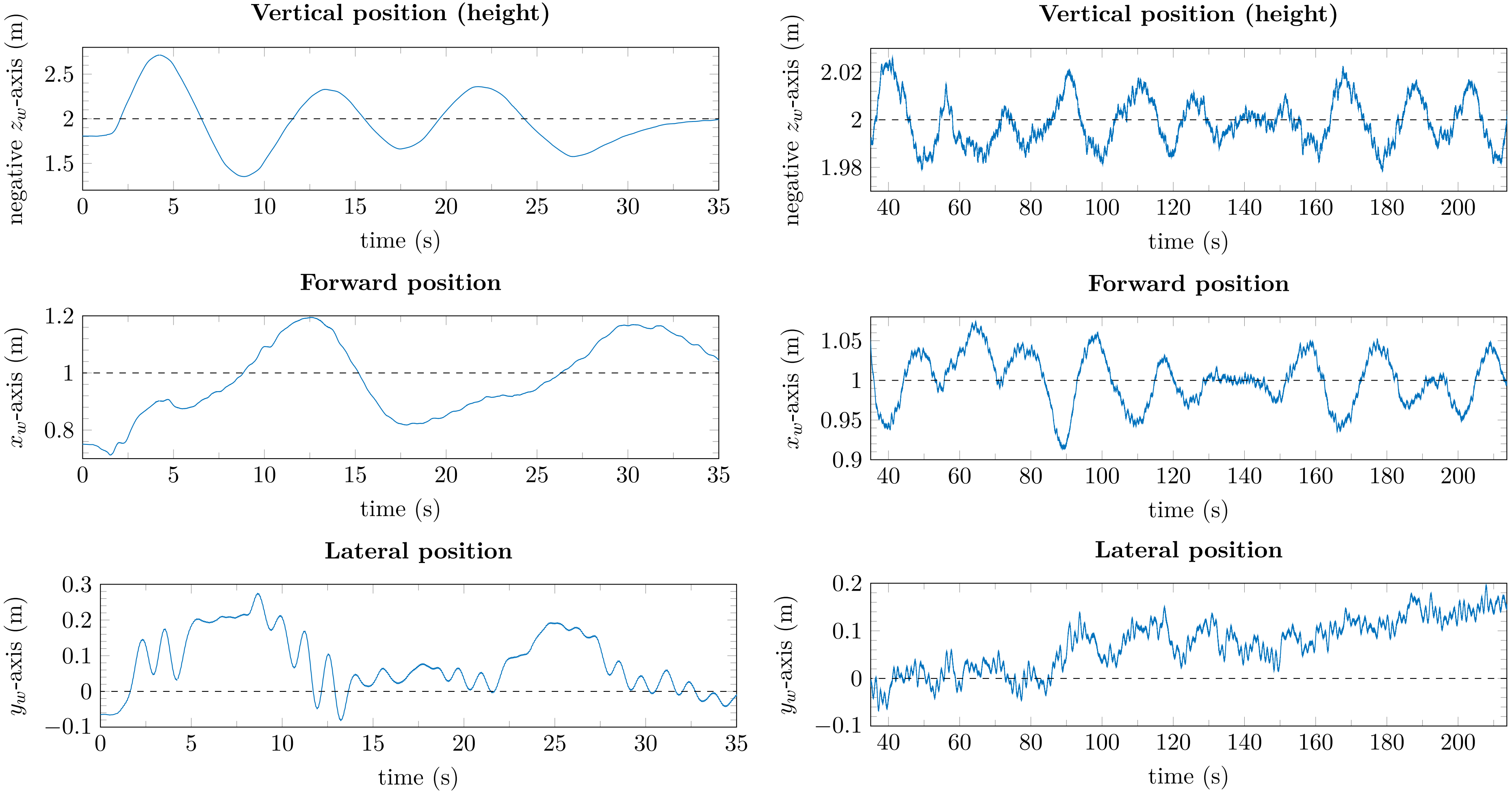
Test №	gain i	gain p	Figure	on page	Notes
1	200 %	0	C.1	84	Combined speed-thrust control; no lateral integral gain.
9	–	–	C.2	85	Build-in paparazzi PID position control.
18	200 %	0	C.3	86	Semi-adaptive speed-thrust control; increasing wind speed $0.7\,\frac{\mathrm{m}}{\mathrm{s}} \leq V_W^{SP} \leq 1.3\,\frac{\mathrm{m}}{\mathrm{s}}$
24	300 %	0	C.4	87	Semi-adaptive speed-thrust control.
25	100 %	0	C.5	88	Semi-adaptive speed-thrust control.
27	200 %	0	C.6	89	Semi-adaptive speed-thrust control.
28	200 %	100 %	C.7	90	Semi-adaptive speed-thrust control.
30	200 %	300 %	C.8	91	Semi-adaptive speed-thrust control.

Table C.1.: Gain settings of the wind-tunnel fights.

For comparison, we employed as well the already implemented position control approach of paparazzi; here, all three axes were controlled independently by PID feedback laws. After tuning, the gains were given by:

$$\begin{aligned} k_{Fz} &= 60 & i_{Fz} &= 35 & d_{Fz} &= 20 \\ k_{\Theta x} &= 1900 & i_{\Theta x} &= 10 & & \\ k_{\Psi y} &= 200 & & & & \end{aligned}$$

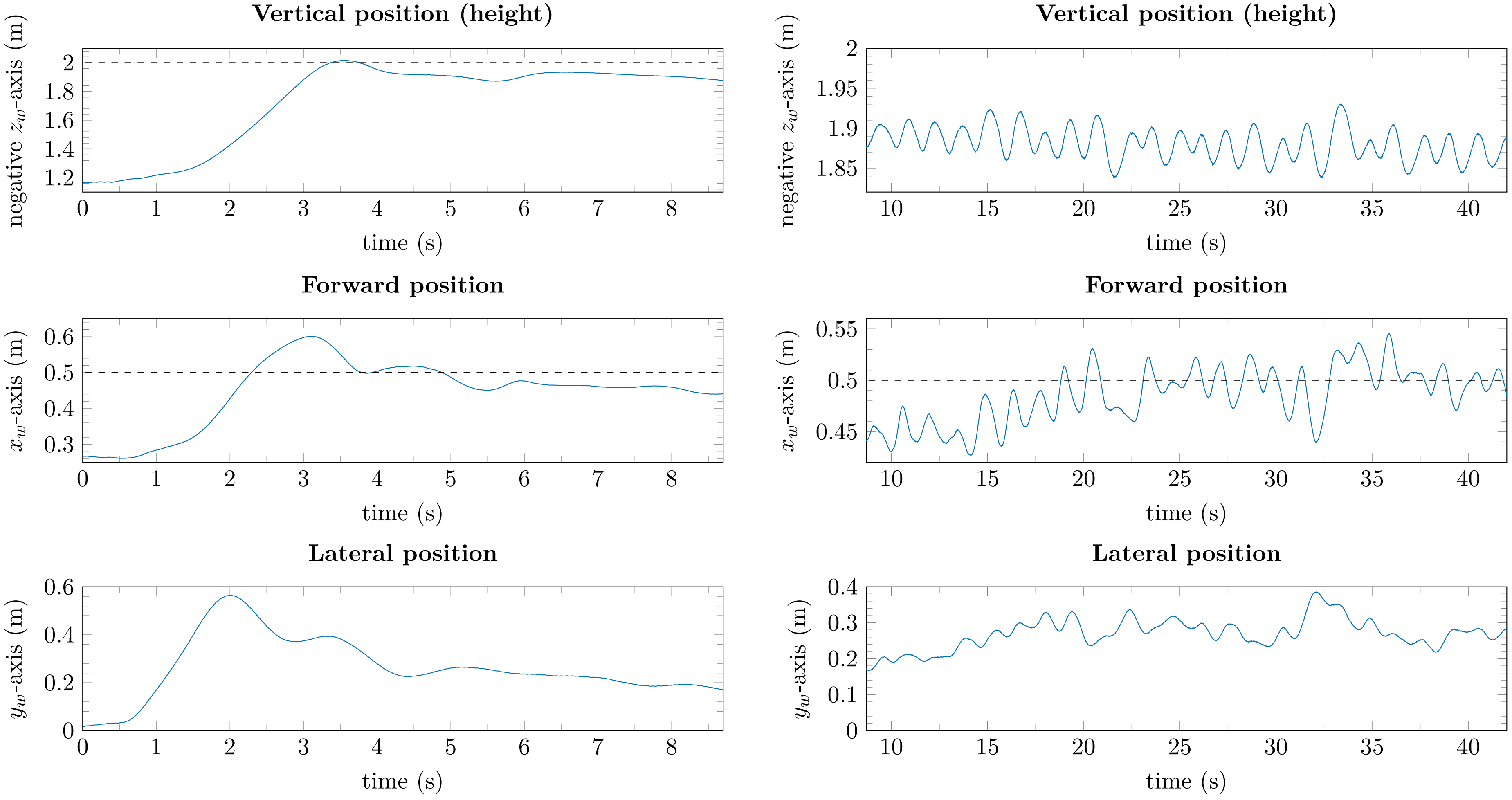
The PID position control was tested in Flight № 9, shown in Fig. C.2 on page 85.



(a) Settling response.

(b) Steady-state.

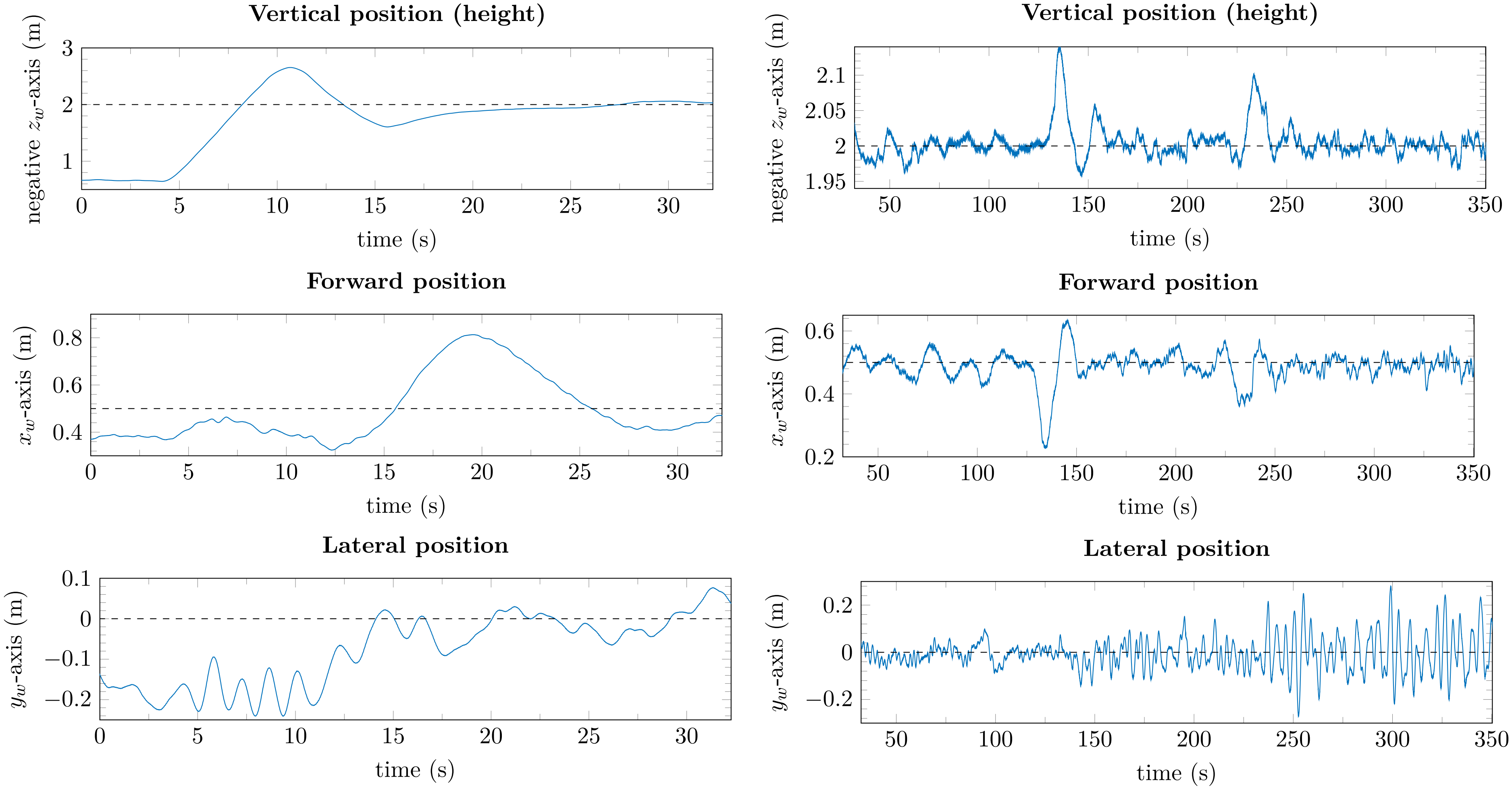
Figure C.1.: Flight № 1: combined speed-thrust control; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\text{m}}{\text{s}}$; *combined speed-thrust control with* $k = 0$, $i = 200\,\%$.)



(a) Settling response.

(b) Steady-state.

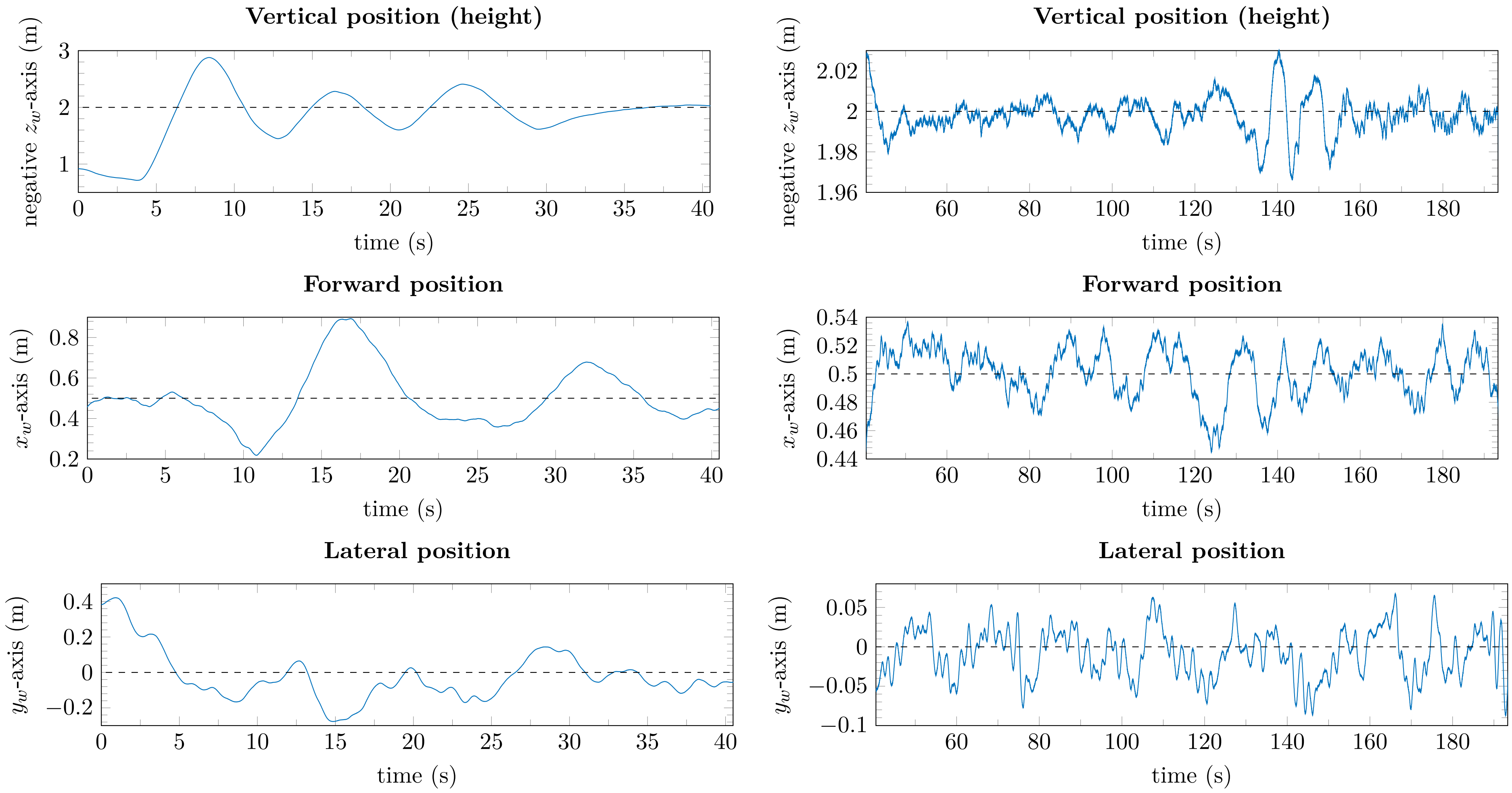
Figure C.2.: Flight № 9: paparazzi PID position control; in the axes of the wind-tunnel reference system.
(*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; *PID position control.*)



(a) Settling response.

(b) Steady-state.

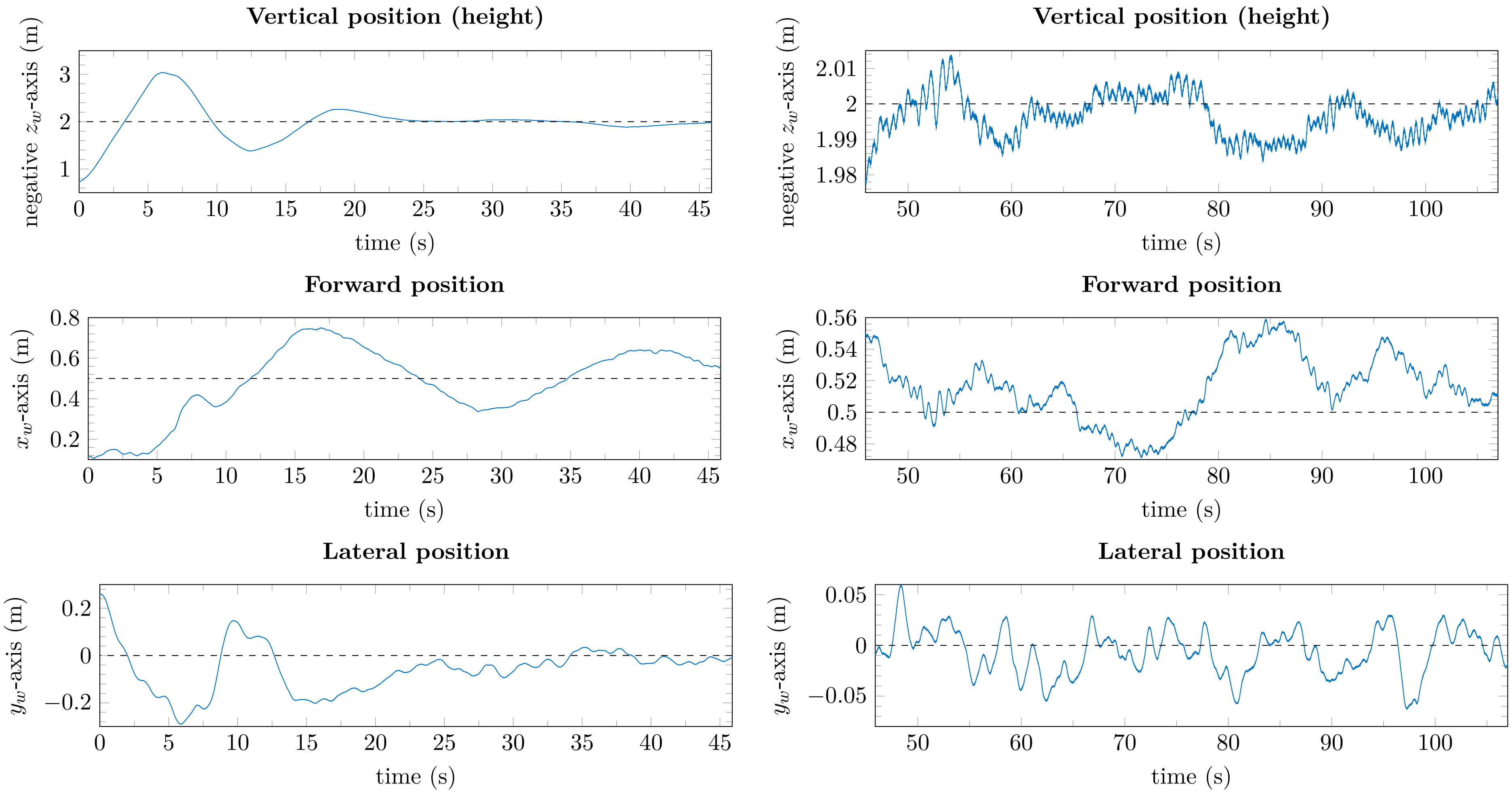
Figure C.3.: Flight № 18: semi-adaptive speed-thrust control for increasing wind speed; in the axes of the wind-tunnel reference system. (*wind speed* $0.7\,\frac{\mathrm{m}}{\mathrm{s}} \leq V_W^{SP} \leq 1.3\,\frac{\mathrm{m}}{\mathrm{s}}$; *semi-adaptive speed-thrust control with* $k = 0$, $i = 300\,\%$.)



(a) Settling response.

(b) Steady-state.

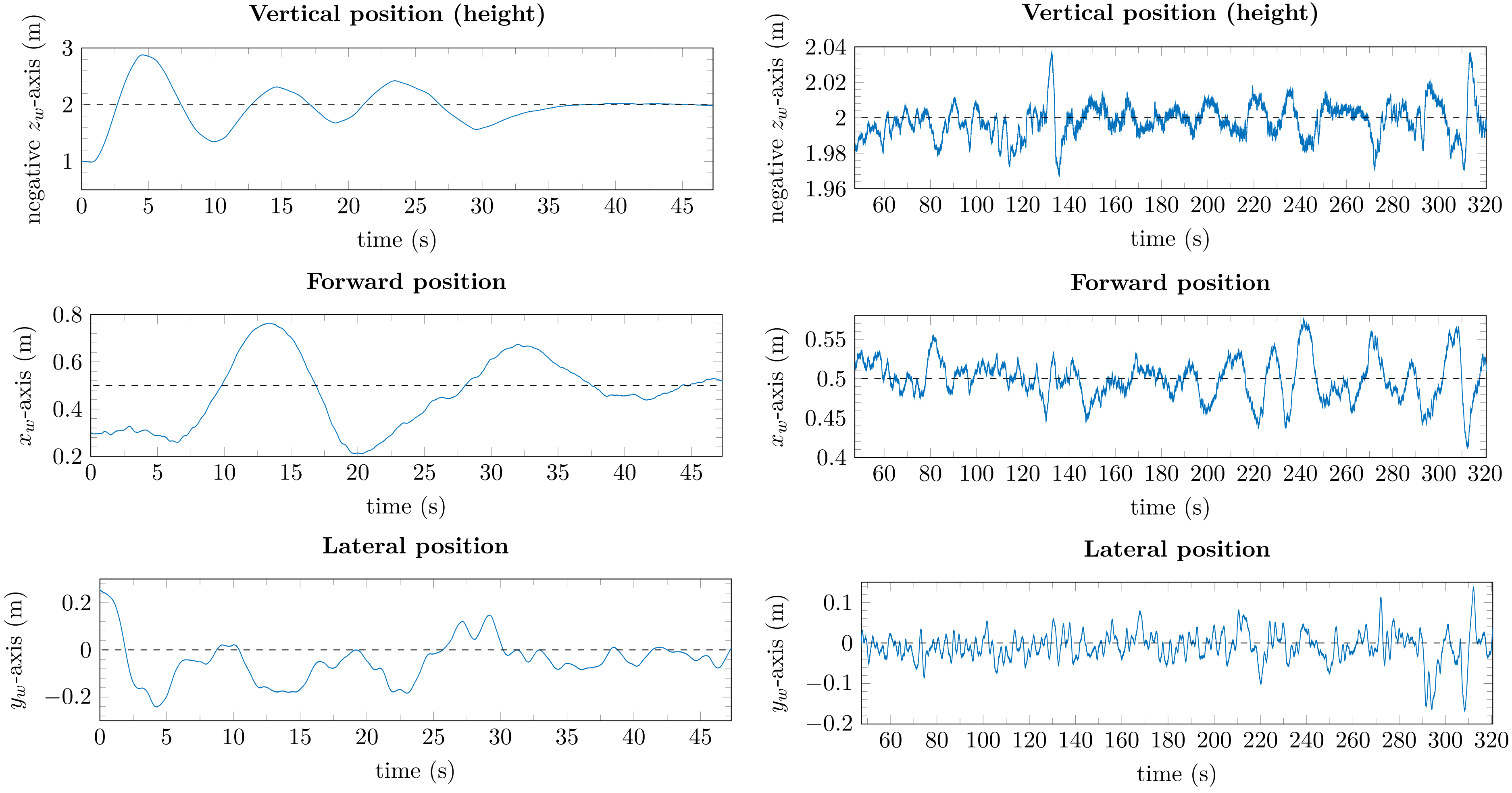
Figure C.4.: Flight № 24: semi-adaptive speed-thrust control with $i = 300\,\%$; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; *semi-adaptive speed-thrust control with* $k = 0$, $i = 300\,\%$.)



(a) Settling response.

(b) Steady-state.

Figure C.5.: Flight № 25: semi-adaptive speed-thrust control with $i = 100\,\%$; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; *semi-adaptive speed-thrust control with* $k = 0$, $i = 100\,\%$.)

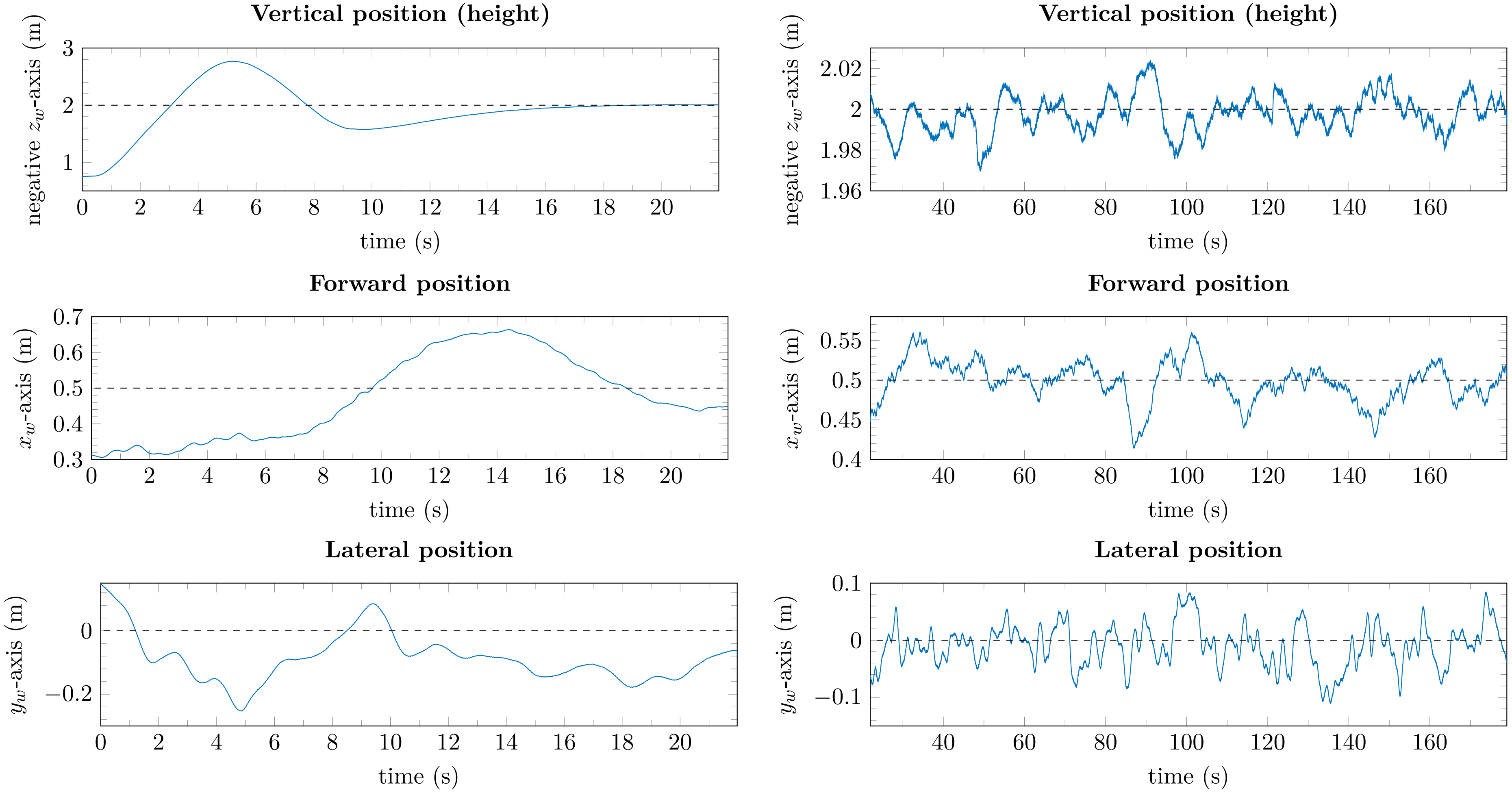


(a) Settling response.

(b) Steady-state.

Figure C.6.: Flight № 27: semi-adaptive speed-thrust control with $i = 200\,\%$; in the axes of the wind-tunnel reference system.

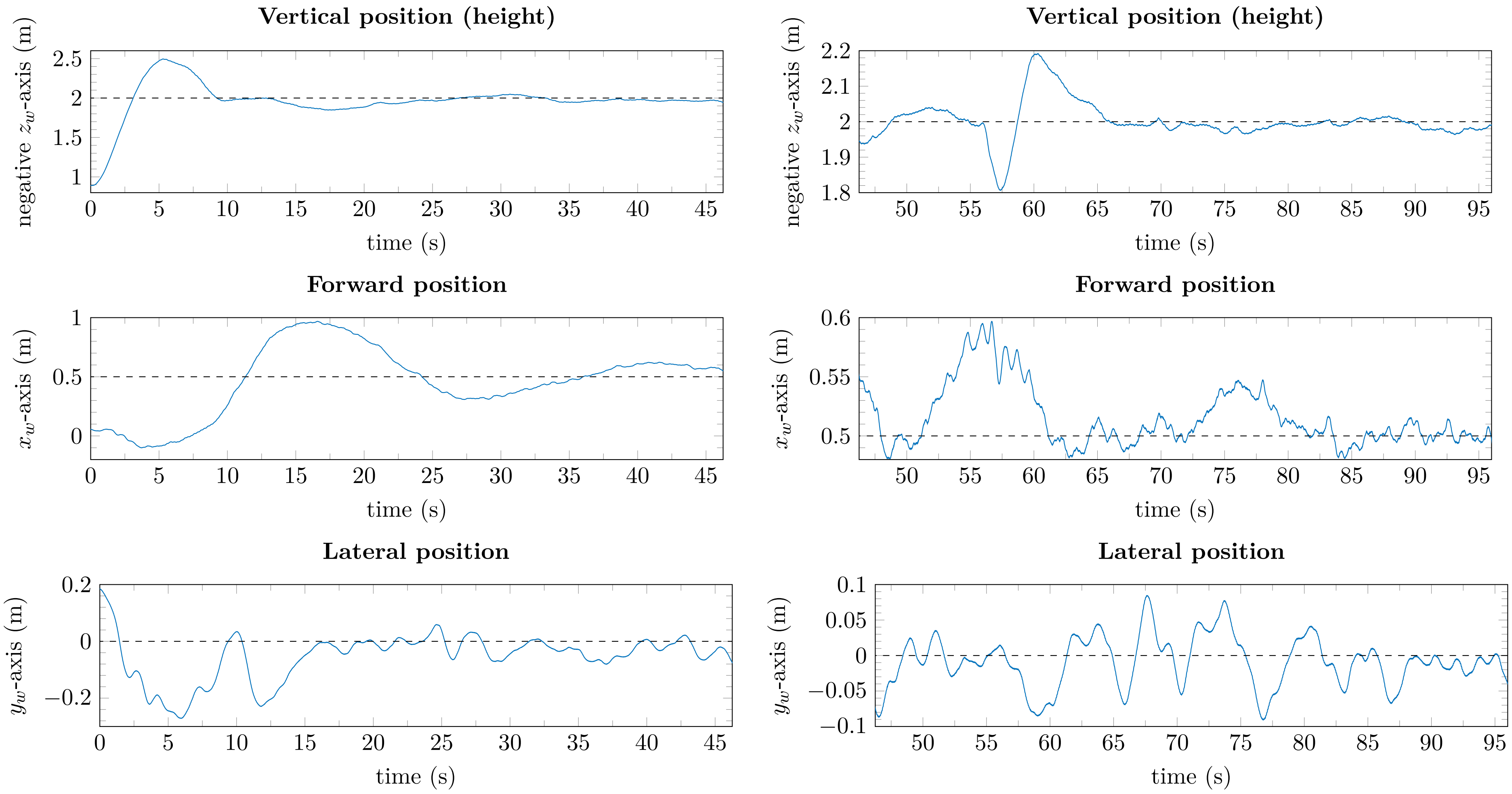
(*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; *semi-adaptive speed-thrust control with* $k = 0$, $i = 200\,\%$.)



(a) Settling response.

(b) Steady-state.

Figure C.7.: Flight № 28: semi-adaptive speed-thrust control with $k = 100\,\%$; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\text{m}}{\text{s}}$; *semi-adaptive speed-thrust control with* $k = 100\,\%$, $i = 200\,\%$.)



(a) Settling response.

(b) Steady-state.

Figure C.8.: Flight № 30: semi-adaptive speed-thrust control with $k = 300\,\%$; in the axes of the wind-tunnel reference system. (*wind speed* $V_W^{SP} = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$; *semi-adaptive speed-thrust control with* $k = 300\,\%$, $i = 200\,\%$.)

D. DVD Content

```
/
├─ MA_TorbjoernCunis.pdf ........................ Thesis
├─ flightlogs
│  ├─ windtunnel ................................ Test flights in the OJF wind-tunnel
│  │  └─ (...)
│  └─ cyberzoo .................................. Test flights in postero
│     └─ (...)
├─ matlab
│  ├─ model ..................................... MATLAB Simulink model of linear flight dynamics by [60]
│  │  └─ (...)
│  └─ speedthrust ............................... MATLAB script and data deriving speed-thrust control
│     └─ (...)
└─ paparazzi .................................... Delfly control implementation in paparazzi
   ├─ conf_delfly.xml ........................... paparazzi configuration for the Delfly II
   ├─ conf
   │  ├─ airframes
   │  │  └─ delfly_optitrack_wifi.xml ........... Definition of the Delfly II airframe
   │  ├─ flight_plans
   │  │  ├─ rotorcraft_delfly2_ojf.xmlFlight-plan for tests in the OJF wind-tunnel
   │  │  └─ rotorcraft_delfly2.xml .............. Flight-plan for tests in postero
   │  ├─ modules
   │  │  ├─ delfly_control.xml .................. Definition of the DelflyControl module
   │  │  └─ delfly_utils.xml .................... Definition of the DelflyUtility module
   │  └─ settings
   │     └─ modules
   │        ├─ delfly_control_settings.xml ...... DelflyControl module settings
   │        └─ delfly_utils_settings.xml ........ DelflyUtility module settings
   └─ ...
```

```
⋮
└─ paparazzi
   └─ sw
      ├─ airborne
      │  └─ modules
      │     ├─ delfly_control ........ Implementation of the DelflyControl module
      │     │  └─ (...)
      │     └─ delfly_utils........... Implementation of the DelflyUtility module
      │        └─ (...)
      └─ ground_segment
         └─ misc
            └─ natnet2ivy.c ................. Script for position uplink to the aircraft
```