

Automatisierung eines Kamera-Quadropters zur Objektverfolgung

Automation of a Camera Quadcopter for Object Tracking

Von der Fakultät für Maschinenwesen der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von

Thomas Engelhardt

Berichter: Univ.-Prof. Dr.-Ing. Dirk Abel
Univ.-Prof. Dr.-Ing. Dieter Moormann

Tag der mündlichen Prüfung: 7. Mai 2021

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Kurzfassung

Mit beeindruckenden Videoaufnahmen aus der Luft lassen sich in Film-, Fernseh- und Marketingproduktionen besondere filmische Akzente setzen. Solche Videosequenzen werden häufig mit Kamera-Quadrokoptern erstellt. Die Steuerung eines Quadrokopters per Fernsteuerung erfordert allerdings viel Erfahrung und Geschick, insbesondere wenn dynamische Aufnahmen von bewegten Motiven gedreht werden sollen. Es ist daher attraktiv, die Aufgaben des Piloten zu automatisieren und den Kamera-Operator durch Assistenzfunktionen bei der Kameraführung zu unterstützen.

Gegenstand dieser Arbeit ist daher die Automatisierung eines Kamera-Quadrokopters zur Verfolgung von bewegten Objekten für Filmzwecke. Um die nichtlineare Dynamik des Quadrokopters ausnutzen zu können, wurde eine flachheitsbasierte Regelung entwickelt. Diese berücksichtigt Störungen durch Wind, die von einem Störgrößen-Beobachter geschätzt werden. Neben dem Beweis der Flachheit wurde in der Arbeit die quasi-statische Zustandsrückführung hergeleitet, die die Systemdynamik eingangs-/zustandslinearisiert, und eine kaskadierte Trajektorienregelung entworfen. Für die unterschiedlichen Assistenzfunktionen wurden mehrere Methoden zur Trajektoriengenerierung implementiert. Als Sensorfusion zur Schätzung des Systemzustandes wurde ein Erweitertes Kalman-Filter ausgelegt, das als Sensorik eine IMU und einen GNSS-RTK-Empfänger nutzt. Zur Vorbereitung auf die Nutzung des europäischen Satellitennavigationssystems Galileo wurde im Testgebiet automotiveGATE ein experimenteller GPS- und Galileo-Empfänger auf dem Quadrokopter getestet.

Zur Erprobung der entwickelten Automatisierung wurde ein Quadrokopter-Prototyp aufgebaut. Außerdem wurde ein Handbediengerät mit Touchscreen und Joystick realisiert, das dem Operator erlaubt, das System zu kommandieren. In Flugversuchen wurde u. a. demonstriert, wie der Quadrokopter ein Zielobjekt automatisch verfolgt und seine Kamera eigenständig so ausrichtet, dass das Zielobjekt mittig im Kamerabild zu sehen ist.

Abstract

Stunning aerial videos can be used in movie, television, and marketing productions as special cinematic means. Such video footage is often created by camera quadcopters. Controlling a quadcopter by a radio controller however requires profound experience and skills, especially if dynamic video sequences of moving motifs are to be shot. It is therefore desirable to automate the pilot's tasks and to support the camera operator in camera guidance by assistance functions.

Thus, the topic of this thesis is the automation of a camera quadcopter for tracking of moving objects for cinematic purposes. A flatness-based controller was developed to leverage the nonlinear dynamics of the quadcopter. This controller also takes into account the disturbances by wind, which are estimated by a disturbance observer. Besides the proof of flatness, the quasi-static state feedback, which linearizes the input-state system dynamics, was derived and a cascaded trajectory controller was designed in this thesis. Several trajectory generation methods were implemented for the different assistance functions. As sensor fusion for system state estimation, an Extended Kalman-Filter was applied, which uses an IMU and a GNSS-RTK-receiver as sensor inputs. To prepare for the usage of the European navigation satellite system Galileo, an experimental GPS- and Galileo-receiver was tested on the quadcopter in the test facility automotiveGATE.

A quadcopter prototype was built to test the developed automation. Furthermore, a handheld device with touchscreen and joystick was realized, which allows the operator to command the system. It was demonstrated in flight tests, how the quadcopter e. g. automatically follows a target object and rotates its camera in such way that the target object appears centered in the camera image.

Vorwort

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Regelungstechnik der RWTH Aachen University.

Herrn Univ.-Prof. Dr.-Ing. Dirk Abel, Doktorvater und Leiter des Instituts für Regelungstechnik, möchte ich für die Betreuung der Arbeit und das mir entgegengebrachte Vertrauen ganz besonders danken. Herrn Univ.-Prof. Dr.-Ing. Dieter Moormann, Leiter des Instituts für Flugsystemdynamik, danke ich für die Übernahme des Korreferats. Mein Dank gilt auch Herrn Univ.-Prof. Dr.-Ing. Lutz Eckstein, Leiter des Instituts für Kraftfahrzeuge, für den Prüfungsvorsitz und Herrn Univ.-Prof. Dr.-Ing. Heinrich Rake, ehemaliger Leiter des Instituts für Regelungstechnik, als Prüfungsbeisitzer.

Meinen ehemaligen Kolleginnen und Kollegen am Institut für Regelungstechnik danke ich für die gute Zusammenarbeit und das angenehme Arbeitsklima – ich denke gerne an die Zeit am Institut und die gemeinsame Freizeit zurück. Insbesondere danke ich Herrn Dr.-Ing. Thomas Konrad für den intensiven fachlichen Austausch und die großartige Kooperation vor allem bei den Flugversuchen. Mein Dank geht auch an alle Mitarbeiterinnen und Mitarbeiter in Verwaltung und Werkstätten, die mich beim Aufbau des Quadropters und darüber hinaus tatkräftig unterstützt haben. Zudem bin ich den vielen Studentinnen und Studenten dankbar, die als studentische Hilfskräfte oder mit ihren Studien-, Diplom-, Bachelor- und Masterarbeiten zum Erfolg beigetragen haben.

Schließlich danke ich meiner Familie – ganz besonders meinen Eltern – sowie meinem Freundeskreis für die wertvolle Unterstützung während der Promotionszeit.

Böblingen, im Mai 2021

Thomas Engelhardt

Inhaltsverzeichnis

Kurzfassung	3
Abstract	5
Vorwort	7
Nomenklatur	12
Abkürzungen und Formelzeichen	12
Mathematische Notation	14
Verweise	14
Persönliche Veröffentlichungen	15
1 Einleitung	17
1.1 Einführung und Motivation	17
1.2 Stand der Technik	18
1.3 Zielsetzung, Schwerpunkte und Gliederung der Arbeit	19
2 Grundlagen	23
2.1 Koordinatensysteme, Erdmodell und Lagedarstellungen	23
2.1.1 Bezugs- und Koordinatensysteme	23
2.1.2 Erdmodell	24
2.1.3 Lagedarstellungen	28
2.2 Dynamik	29
2.2.1 Rotatorische Kinematik	29
2.2.2 Translatorische Kinematik	30
2.2.3 Kinetik	30
2.3 Kalman-Filter	31
2.3.1 Kalman-Filter für lineare Systeme	31
2.3.2 Erweitertes Kalman-Filter	34
2.3.3 Modelldiskretisierung	37
2.4 Flache Systeme	40
2.4.1 Flachheit	40
2.4.2 Exakte Eingangs-/Zustandslinearisierung	42
2.4.3 Trajektorienfolgeregelung	46
2.5 Satellitennavigation	47

2.5.1	Globale Satellitennavigationssysteme	48
2.5.2	Funktionsprinzip Satellitennavigation	51
2.5.3	Differentielle Korrekturverfahren	61
3	System	63
3.1	Systementwurf	63
3.2	Flugsystem	65
3.2.1	Hardwareaufbau	65
3.2.2	Softwarearchitektur	72
3.3	Benutzerschnittstelle	75
3.3.1	Bedienkonzept	75
3.3.2	Hardware	77
3.3.3	Software	79
3.4	Kommunikationsinfrastruktur	86
3.5	Testumgebungen	90
3.5.1	Simulationsumgebung	90
3.5.2	Indoor-Testumgebung	91
3.5.3	Outdoor-Testumgebung	92
3.5.4	Galileo-Testgebiet automotiveGATE	93
4	Sensorfusion	97
4.1	Sensoren	98
4.1.1	Inertiale Messeinheit (IMU)	98
4.1.2	GNSS-Empfänger	107
4.1.3	Magnetometer	112
4.1.4	Ultraschallsensor und Luftdrucksensor	112
4.2	Systemmodell für Sensorfusion	112
4.3	Propagation des Systemzustandes	117
4.4	Propagation der Schätzfehlerkovarianz	122
4.4.1	Schätzfehlermodell	123
4.5	Estimation	127
4.5.1	Messmodelle	128
4.5.2	Update	132
4.6	Ablauf	134
4.6.1	Initialisierung	134
4.6.2	Ablauf Propagation, Estimation	138
5	Regelung	143
5.1	Systemmodell für flachheitsbasierte Regelung	145
5.1.1	Systemzustand	145
5.1.2	Aerodynamik	147
5.1.3	Bewegungsgleichungen	153
5.1.4	Trägheitstensor	155
5.1.5	Störgrößen-Beobachter	158
5.2	Flachheitsbasierte Regelung	159

5.2.1	Beweis Flachheit	160
5.2.2	Flachheitsbasierte Zustandsrückführung	166
5.2.3	Trajektorienfolgeregelung	172
5.3	Trajektoriengenerierung	175
5.3.1	Verfolgung eines Zielobjekts	176
5.3.2	Trajektoriengenerierung mit Spline-Kurven	180
5.3.3	Trajektoriengenerierung mit Kaskadenregelung	182
5.3.4	Trajektoriengenerierung mit MPC	188
5.4	Ablaufsteuerung	195
5.5	Simulation	198
6	Erprobung	201
6.1	Sensorik-Erprobung	201
6.1.1	Sensorfusion	201
6.1.2	Experimenteller GPS- und Galileo-Empfänger im Galileo-Testgebiet	210
6.2	Flachheitsbasierte Regelung und Trajektoriengenerierung	213
6.2.1	Schubkalibrierung	214
6.2.2	Flachheitsbasierte Regelung	215
6.2.3	MPC-Trajektoriengenerierung	219
6.2.4	Flug mit Joystick	221
6.2.5	Objektverfolgung	223
7	Zusammenfassung und Ausblick	245
7.1	Zusammenfassung	245
7.2	Ausblick	247
	Anhang	249
	A Stochastik	250
	B Ergänzungen	256
B.1	Lagedarstellungen	256
B.2	Weitere mathematische Ergänzungen	258
B.3	Ergänzungen zum Kapitel Regelung	259
	C Parameter	261
	Abbildungsverzeichnis	263
	Tabellenverzeichnis	267
	Literaturverzeichnis	268

Nomenklatur

Abkürzungen und Formelzeichen

Abkürzungen und Akronyme

DOP	dilution of precision, S. 60
GBAS	bodengebundenes Augmentierungssystem (englisch: satellite-based augmentation system) für GNSS, S. 48
GLONASS	russisches globales Satellitennavigationssystem, S. 47
GNSS	Globales Satellitennavigationssystem (englisch: Global Navigation Satellite System), S. 47
GPS	US-amerikanisches globales Satellitennavigationssystem: Navigational Satellite Timing and Ranging – Global Positioning System, S. 47
IMU	Inertiale Messeinheit (englisch: Inertial Measurement Unit), S. 98
MEMS	Mikro-elektromechanisches System (englisch: micro-electro-mechanical system), S. 98
MPC	modellprädiktive Regelung (englisch: Model Predictive Control), S. 188
POI	Point-of-Interest: statischer Punkt, der gefilmt werden soll, S. 76
PRN	pseudo-zufällige Codesequenz (englisch: pseudo random noise), S. 53
PVT	Position, Geschwindigkeit, Zeit (englisch: position, velocity, time), S. 51
QC	Quadrocopter
RTK	Real Time Kinematic: GNSS Korrekturverfahren, S. 62
SBAS	satellitenbasiertes Augmentierungssystem (englisch: satellite-based augmentation system) für GNSS, S. 48

Lateinische Buchstaben

I	Einheitsmatrix, siehe <i>Einheitsmatrix</i> auf Seite 14
<i>a</i>	Beschleunigung
<i>b</i>	Bias
<i>c</i>	Lichtgeschwindigkeit im Vakuum, $c = 299\,792\,458$ m/s

f	spezifische Kraft
g	spezifische Gravitationskraft der Erde, siehe (2.5)
g_l	Schwerebeschleunigung, siehe (2.9)
h	Höhe im WGS84-Koordinatensystem, siehe Seite 24
k	Zeitindex, siehe <i>Zeitpunkt</i> auf Seite 14
n	Rauschen
p	Position
T	Zeitintervall
t	Zeit
u	Eingang
v	Geschwindigkeit
x	Systemzustand
y	Ausgang

Griechische Buchstaben

α	Drehbeschleunigung
λ	Längengrad im WGS84-Koordinatensystem, siehe Seite 24
ω	Drehrate
ψ	Gierwinkel, siehe (2.16a)
θ	Nickwinkel, siehe (2.16b)
φ	Breitengrad im WGS84-Koordinatensystem, siehe Seite 24
φ	Rollwinkel, siehe (2.16c)

Operatoren

\bullet	Quaternionen-Multiplikations-Operator, siehe (B.1)
\wedge	logische Und-Verknüpfung
\vee	logische Oder-Verknüpfung
\neg	logische Nicht-Verknüpfung
\times	Kreuzprodukt-Operator

Mathematische Notation

Skalare, Vektoren, Quaternionen, Matrizen Skalare Größen a werden ohne Zusätze, Vektoren \vec{a} mit einem Pfeil, Quaternionen \mathbf{q} durch fettgedruckte Kleinbuchstaben und Matrizen \mathbf{A} durch fettgedruckte Großbuchstaben dargestellt.

Vektor- und Matricelemente Das i -te Element eines Vektors \vec{a} wird durch einen Index a_i dargestellt. Die Schreibweise c_{lk} wird für das Element in der l -ten Zeile und k -ten Spalte einer Matrix \mathbf{C} verwendet.

Einheitsmatrix Die Einheitsmatrix wird durch den Buchstaben \mathbf{I} symbolisiert. Ein Index gibt die Zeilen- und Spaltenzahl an: \mathbf{I}_3 hat z. B. die Dimension 3×3 .

Zeitpunkt a_k ist der Wert der zeitdiskreten Größe zum Zeitpunkt $t = kT$.

Bezugs- und Koordinatensysteme Bezugssysteme einer Größe werden als nachgestellte Tiefstellungen dargestellt. So steht z. B. \vec{v}_{ab} für die Geschwindigkeit eines Bezugssystems B relativ zu einem Bezugssystem A . Soll zudem ein Koordinatensystem C für den Vektor spezifiziert werden, wird dieses durch eine nachgestellte Hochstellung \vec{v}^c angegeben.

zeitliche Ableitung n -ter Ordnung Die n -te zeitliche Ableitung einer Größe a wird durch eine hochgestellte Zahl in Klammern gekennzeichnet: $\frac{d^n a}{dt^n} = a^{(n)}$. Für die erste und zweite zeitliche Ableitung wird zudem die Schreibweise mit einem Punkt $\frac{da}{dt} = \dot{a}$ bzw. zwei Punkten $\frac{d^2 a}{dt^2} = \ddot{a}$ verwendet.

Bezugssystem zeitliche Ableitung Die zeitliche Ableitung der Vektorfunktion \vec{f} im Bezugssystem A wird geschrieben als $\frac{A d\vec{f}}{dt}$

Kreuzproduktbildende Matrix Die Notation für die kreuzproduktbildende Matrix eines Vektors \vec{a} ist $[\vec{a} \times]$, siehe (B.9).

Messgrößen, Schätzgrößen Eine Tilde kennzeichnet eine gemessene Größe \tilde{a} und ein Zirkumflex eine geschätzte Größe \hat{a} in Abgrenzung zur realen Größe a . Ein A-priori-Schätzwert vor dem Kalman-Estimationsschritt wird mit einem hochgestellten Minus \hat{a}^- , ein A-posteriori-Schätzwert nach dem Estimationsschritt mit einem hochgestellten Plus \hat{a}^+ geschrieben.

Beschränkung Um eine Beschränkung einer Variablen a auf die obere und untere Grenze u, l darzustellen, wird die Schreibweise $\text{sat}_l^u(a)$ verwendet, siehe (B.12).

Verweise

Literaturverweise werden mit eckigen Klammern [1] und Verweise auf Gleichungen mit der Gleichungsnummer in runden Klammern (5.87) ohne den Zusatz „Gleichung“ gekennzeichnet. Alle weiteren Verweise werden ausgeschrieben: z. B. Kapitel 5, Abschnitt 6.2.5, Abbildung 4.7, Tabelle 4.1, Seite 121. Im PDF-Dokument kann durch Anklicken der Ziffern den Verweisen gefolgt werden.

Persönliche Veröffentlichungen

Diese Arbeit basiert in Teilen auf folgenden Veröffentlichungen:

- Thomas Engelhardt, Thomas Konrad, Björn Schäfer und Dirk Abel. „Flatness-based control for a quadrotor camera helicopter using model predictive control trajectory generation“. In: *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, Seiten 852–859. DOI: 10.1109/MED.2016.7536036
- Thomas Engelhardt, Lothar Kurz, Tobias Noll und Dirk Abel. „Experimental Galileo-Receiver for Quadrocopter Navigation in automotiveGATE“. In: *European Navigation Conference (ENC) 2014*. Rotterdam, 2014
- Thomas Konrad, Thomas Engelhardt und Dirk Abel. „Propeller thrust identification and calibration for high-precision control of a quadrotor unmanned aerial vehicle“. In: *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017, Seiten 1225–1230. DOI: 10.1109/MED.2017.7984285
- Thomas Engelhardt, Alexander Katriniok, Björn Schäfer und Dirk Abel. „Fahrzeugführung mit Galileo – Anwendungen und Testgebiete“. In: *Deutscher Luft- und Raumfahrtkongress 2010, Hamburg, 31.08.-02.09.2010*. Deutsche Gesellschaft für Luft- und Raumfahrt (DGLR), 2010, Seiten 867–872. ISBN: 978-3-932182-68-5
- Thomas Engelhardt, Alexander Katriniok, Björn Schäfer und Dirk Abel. „automotiveGATE und railGATE: Galileo-basierte Fahrzeugführung für bodengebundenen Verkehr“. In: *E-Mobility: Technologien - Infrastruktur - Märkte. VDE-Kongress 2010, Leipzig, 08.-09.11.2010*. VDE-Verlag, 2010
- Thomas Engelhardt, René Rütters, Alexander Katriniok und Dirk Abel. „automotiveGATE und railGATE, Testgebiete für Galileo-basierte Fahrzeugführung auf Schiene und Straße“. In: *Automation 2011: Der 12. Branchentreff der Mess- und Automatisierungstechnik, Kongress Baden-Baden, 28.-29.06.2011*. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2011. ISBN: 978-3-18-092143-3

1 Einleitung

1.1 Einführung und Motivation

Luftbildaufnahmen faszinieren – sie eröffnen neue, außergewöhnliche Perspektiven auf die Welt. In Film- und Fotoproduktionen wird die Vogelperspektive daher als besonderes Gestaltungsmittel eingesetzt. Besonders beeindruckend und in der Erstellung herausfordernd sind dynamische Videoaufnahmen aus der Luft, sei es ein Werbefilm für einen Sportwagen oder die Liveübertragung eines Skiabfahrtsrennens.

Zahlreiche Branchen haben Bedarf an Video- und Fotoaufnahmen aus der Luft. Neben dem Einsatz im Film-, Fernseh- und Marketingbereich gibt es viele Anwendungen u. a. in Industrie und Landwirtschaft, im Vermessungswesen, Baugewerbe und Energiesektor sowie bei Polizei und Feuerwehr. Typisch sind dabei Aufgaben wie Inspektion, Kartierung, Überwachung und Dokumentation. So können z. B. Bauwerke wie Brücken, Oberleitungsmasten oder Windenergieanlagen per Kamera inspiziert werden, die sonst nur aufwendig von Industriekletterern untersucht werden können. Auch das Angebot an Produkten, mit denen Privatpersonen ihre Sport- und Freizeitaktivitäten filmen können, wächst stetig.

Luftbildaufnahmen, die von unbemannten Luftfahrzeugen aus erstellt werden, schließen eine Lücke zwischen Aufnahmen mithilfe bemannter Helikopter und bodengebundenen Aufnahmen. Als unbemannte Luftfahrzeuge werden dabei überwiegend Multikopter eingesetzt, die dank Lageregelung gut manövrierbar und mechanisch meist recht einfach konstruiert sind. Multikopter kommen überwiegend als Quadro-, Hexa- oder Oktokopter mit vier, sechs oder acht Rotoren vor. Je nach Nutzungsszenario werden für Luftbildaufnahmen aber auch andere unbemannte Fluggeräte wie Helikopter, Flugzeuge, Ballone und Luftschiffe eingesetzt.

Ein unbemanntes Fluggerät zu steuern erfordert neben der notwendigen Sachkunde vor allem viel Übung. Auch wenn viele Multikopter eine GNSS-basierte Positionsregelung mitbringen, muss die Pilotin oder der Pilot das Fluggerät auch unter schwierigen Bedingungen – bei Windböen, Hindernissen, schlechter Sicht oder GNSS-Störungen – sicher beherrschen. Insbesondere der Dreh von dynamischen Videosequenzen stellt höchste Anforderungen an die Pilotinnen und Piloten, wenn bei hohen Geschwindigkeiten und unter beschränkten Platzverhältnissen mitunter nahe an ein bewegtes Motiv geflogen werden soll. Unter diesen Umständen ist es häufig nicht möglich, gleichzeitig auch noch die Ausrichtung einer Kamera zu steuern. Daher wird die Kameraführung oft durch eine zweite Person übernommen.

Die Bewegungen von bodengebundenen Kameras sind in der Regel einfach und präzise steuerbar. Für dynamische Kamerafahrten wird die Kamera z. B. mithilfe eines Kamerawagens bewegt. Um von oben zu filmen, können Kamerakräne eingesetzt werden, an deren Ausleger die Kamera mit einem Gimbal schwenkbar montiert ist. Außerdem werden Roboterarme verwendet, mit denen die Kamera per Joystick in alle Richtungen verfahren und geschwenkt werden kann. Diese Roboterarme fahren auf Knopfdruck wiederholbare Kamerawendungen ab. Wenn größere Distanzen und Höhen abgedeckt werden sollen, kann eine sog. Spidercam eingesetzt werden, bei der die Kamera in einem Gimbal an vier Seilwinden hängt und so per Joystick z. B. im Innenraum eines Sportstadions bewegt werden kann.

Kamera-Multikopter so einfach und präzise wie solche bodengebundenen Kamerasysteme steuern zu können, ist sehr attraktiv für viele Anwendungen, da Kamera-Multikopter oftmals flexibler, schneller und preisgünstiger einsetzbar sind als bodengebundene Systeme. Um die Steuerung eines Kamera-Multikopters auch in herausfordernden Situationen zu erleichtern, sind Assistenzfunktionen erforderlich, die die Aufgaben der Pilotin oder des Piloten sowie der Kamerafrau oder des Kameramanns vollständig oder teilweise automatisieren. Eine Reduzierung der Komplexität ermöglicht es dem Kamera-Operator, sich auf die wesentlichen, gestalterischen Elemente der Dreharbeiten konzentrieren zu können.

Insbesondere bei der Verfolgung von bewegten Motiven ist eine Unterstützung des Operators vorteilhaft. Wenn das System ein Zielobjekt automatisch verfolgen kann, muss der Operator beispielsweise nur noch Abstand und Winkel vorgeben, aus denen das Objekt gefilmt werden soll. Die Einsatzszenarien für eine solche automatische Objektverfolgungsfunktion sind vielfältig. So können z. B. für eine Filmproduktion Schauspieler oder Fahrzeuge bei einer Verfolgungsjagd durch den Kamera-Multikopter gefilmt werden. Im Sportbereich ist die Verfolgung eines Kanuten im Wildwasser-Slalom oder eines Reiters mit Pferd im Geländeparcours möglich. Als industrielle Anwendung wäre u. a. die Überwachung eines Schiffs-Schleppvorganges im Hafenbereich aus verschiedenen Blickwinkeln denkbar.

1.2 Stand der Technik

Das Angebot an Multikoptern hat sich in den letzten zwei Jahrzehnten sowohl für private als auch für professionelle Kunden rasant entwickelt. Oftmals bringen die Flugsysteme bereits eine Kamera mit oder man kann häufig, wenn es die Tragfähigkeit zulässt, ein Gimbal mit Kamera montieren. Die Multikopter sind lagestabilisiert und ein Großteil wird konventionell per Modellbau-Fernsteuerung geflogen, über die Schub, Nickwinkel, Rollwinkel und Gierrate sowie die Drehraten der Kamera-Gimbal-Servomotoren vorgegeben werden. Wenn ein GNSS-Empfänger installiert ist, kann zusätzlich die Position stabilisiert werden, sodass der Multikopter im Schwebeflug nicht wegdriftet. Der lagestabilisierte und der zusätzlich positionsstabilisierte Flugmodus sind die typischen Betriebsmodi vieler Systeme.

Solche Multikopter eignen sich meist auch zum automatischen Abfliegen von zuvor definierten Wegpunkten. Dies wird bereits vielfach z. B. für Kartierungsaufgaben professionell genutzt. Die automatisierte Inspektion von dreidimensionalen Strukturen wie Windenergieanlagen stellt höhere Anforderungen hinsichtlich der Genauigkeit, um die Objekte vollständig abbilden und um Kollisionen vermeiden zu können. Hierzu gibt es bereits erste Ansätze [72, 43, 46]. Weitergehende Assistenzfunktionen insbesondere zur Aufnahme bewegter Objekte sind selten verfügbar. Eine Ausnahme sind die Produkte von DJI, siehe unten.

Für professionelle Filmaufnahmen per Multikopter für Kino-, Fernseh- und Werbeproduktionen setzen Dienstleister wie Skynamic [77], HD Skycam [36] oder Wild Rabbit Aerial Productions [85] überwiegend Multikopter von Freefly Systems und DJI ein.

Freefly Systems Inc. [33] ist eine US-amerikanische Firma, die sich auf Kamera-Bewegungssysteme für Filmproduktionen spezialisiert hat und die in ihrer „Alta“-Produktlinie Quadro- und Oktokopter anbietet, die mit einer Tragfähigkeit von bis zu 16 kg Kameras für Kinoproduktionen hochdynamisch bewegen können. Die Flugregelung der Freefly-Multikopter basiert auf der Auterion Enterprise PX4 Softwareplattform [8] und unterstützt wie viele Systeme lediglich einen lagestabilisierten und einen positionsstabilisierten Flugmodus sowie einen Wegpunkt-Flugmodus.

Das chinesische Unternehmen SZ DJI Technology Co. Ltd. [78] stellte 2012 den ersten Multikopter vor und hat sich inzwischen zum Marktführer für Kamera-Multikopter vor allem im Consumerbereich entwickelt. Mittlerweile hat DJI mit den „Inspire“- und „Matrice“-Produktlinien auch Kamera-Multikopter für semiprofessionelle und professionelle Kunden im Programm. Während die Inspire-Quadrokopter hochwertige Kameras mitbringen, können die Matrice-Multikopter u. a. professionelle Kinokameras anderer Hersteller tragen. Die DJI Produkte bringen je nach Modell zahlreiche Assistenzfunktionen wie z. B. Kollisionsvermeidung mit, wodurch sie sich von vielen Wettbewerbern unterscheiden. Die meisten Funktionen lassen sich über eine Smartphone-App steuern. So kann z. B. beim „Inspire 2“-Modell im Kamera-Videobild in der App eine Person markiert werden, die dann durch Fluggerät und Kamera-Gimbal verfolgt wird, solange die Bildverarbeitung das Zielobjekt erfassen kann. Diese Funktion wurde von den Consumer-Produkten übernommen. Für professionelle Produktionen fehlen ihr ggf. detailliertere Einstellmöglichkeiten und Robustheit in Szenen, in denen das Zielobjekt ganz oder teilweise verdeckt wird. DJI steht in der Kritik bezüglich des Datenschutzes. 2016 räumte das Unternehmen ein, dass die chinesische Regierung mitunter Zugriff auf die mit den Produkten weltweit aufgenommenen Fotos und Videos erhalten kann [32].

1.3 Zielsetzung, Schwerpunkte und Gliederung der Arbeit

Wie der Überblick über den Stand der Technik zeigt, steht die Automatisierung von Kamera-Multikoptern insbesondere für professionelle Filmaufnahmen noch am Anfang. Während der automatisierte Wegpunkt-Flug bereits recht verbreitet ist, gibt es für die

Verfolgung von bewegten Objekten nur einzelne Lösungen, die für professionelle Filmaufnahmen nur eingeschränkt nutzbar sind.

Das Ziel der Arbeit ist daher die Automatisierung eines Kamera-Quadropters zur Verfolgung von bewegten Objekten für Filmzwecke. Die Arbeit möchte damit einen Beitrag leisten, dass bei der Nutzung von Kamera-Multikoptern zukünftig die gewünschte Kameraführung im Vordergrund stehen kann, indem die Steuerung des Fluggerätes vom System übernommen wird.

Um die Aufgaben der Pilotin oder des Piloten vollständig zu automatisieren und den Kamera-Operator durch Assistenzfunktionen zu unterstützen, sind verschiedene Maßnahmen erforderlich. Für den Operator ist eine Benutzerschnittstelle zu realisieren, die es ihm erlaubt, das System in den verschiedenen Automatisierungsgraden zu kommandieren. Die Bewegung der Kamera muss gemäß den Vorgaben des Operators z. B. relativ zu einem Zielobjekt in Form einer Trajektorie geplant werden. Es ist eine Regelung zu entwickeln, die die Motoren des Multikopters und des Kamera-Gimbals so ansteuert, dass die Kamera präzise der geplanten Trajektorie folgt und auf das Zielobjekt ausgerichtet wird. Außerdem müssen für die Regelung Drehrate, Orientierung, Geschwindigkeit und Position des Flugsystems geschätzt werden.

Aus dieser Aufgabenstellung leiten sich folgende Schwerpunkte der Arbeit ab:

Als Benutzerschnittstelle für den Operator wird ein Handbediengerät mit Touchscreen und 6-Achs-Joystick entwickelt, das eine intuitive Bedienung des Systems in verschiedenen Betriebsmodi erlaubt. Über die App des Gerätes soll der Operator das System komfortabel kommandieren und überwachen können.

Zur Planung der Kamerabewegung gemäß den Vorgaben des Operators werden verschiedene Trajektoriengenerierungs-Methoden für die unterschiedlichen Betriebsmodi vorgestellt. Um mit dem unteraktuierten Quadropters dynamischen Trajektorien präzise folgen zu können, muss ein Regelungsverfahren eingesetzt werden, das die nichtlineare Systemdynamik des Quadropters berücksichtigt. Aufgrund der Flachheit des Systems, die in dieser Arbeit nachgewiesen wird, kann eine flachheitsbasierte Regelung genutzt werden, die die nichtlineare Systemdynamik explizit ausnutzt. Diese steuert den Quadropters so an, dass die überlagerte Trajektorienfolgeregelung nur noch ein lineares System regeln muss. Um die Auswirkungen von Störungen wie Wind etc. auf den Quadropters zu verringern, werden die Störgrößen beim Entwurf der flachheitsbasierten Zustandsrückführung berücksichtigt und ein Störgrößen-Beobachter entworfen. Außerdem werden die Rotorschübe zu Beginn jedes Fluges kalibriert, um die Genauigkeit der Trajektorienfolge weiter zu erhöhen.

Um für die Regelung den Zustand des Flugsystems zu schätzen, werden die onboard verfügbaren Sensorsignale mit einem Erweiterten Kalman-Filter fusioniert. Das Filter nutzt neben einer inertialen Messeinheit einen GNSS-RTK-Empfänger, um die zur Objektverfolgung benötigte hohe Positionsgenauigkeit zu erreichen.

Für den automatisierten Betrieb von unbemannten Luftfahrtsystemen sind globale Satellitennavigationssysteme (GNSS), wie in vielen anderen Bereichen, von entscheidender Bedeutung. Das europäische globale Satellitennavigationssystem Galileo wird mit dem Erreichen der vollständigen Betriebsfähigkeit als erstes ziviles GNSS neben den etablierten Systemen GPS und GLONASS weiter an Bedeutung gewinnen. Aus diesem Grund wird auf dem Quadrokopter neben dem GNSS-RTK-Empfänger auch ein experimenteller GPS- und Galileo-Empfänger erprobt. Dazu wird das Galileo-Testgebiet „automotive-GATE“ genutzt, das die Positionsbestimmung mit Galileo-Signalen erlaubt.

Das prototypische Flugsystem wird auf Basis eines Quadrokopters aufgebaut, der flexibel erweiterbar ist. So wird dieser für die Anwendung u. a. mit zusätzlichen Sensoren, einer Kamera im Gimbal und einem leistungsfähigen Computermodul ausgerüstet. Die für das Flugsystem entworfene Softwarearchitektur unterstützt die prototypische Anwendungsentwicklung. Die entwickelte Funk-Infrastruktur erlaubt eine Kommunikation zwischen Operator, Flugsystem und einer Ziel-Einheit, die die Position des zu verfolgenden Zielobjekts an den Quadrokopter sendet.

Zur Erprobung des Gesamtsystems werden diverse Flugversuche durchgeführt. Diese umfassen die Erprobung der Sensorfusion und der Automatisierung des Quadrokopters mit flachheitsbasierter Regelung und Trajektoriengenerierung in den verschiedenen Betriebsmodi. Dazu werden u. a. der Flug per Joystick und die Trajektorienfolge gezeigt sowie die Objektverfolgung beispielhaft anhand der Verfolgung eines Testfahrzeuges demonstriert.

Die Arbeit ist wie folgt gegliedert:

In Kapitel 2 werden die benötigten Grundlagen u. a. zum Kalman-Filter, zu flachen Systemen und zur Satellitennavigation eingeführt. Die verschiedenen Aspekte des Gesamtsystementwurfs wie der Hardwareaufbau und die Softwarearchitektur des Flugsystems, die Benutzerschnittstelle für den Operator und die Kommunikationsinfrastruktur sind in Kapitel 3 zu finden. Außerdem werden dort die genutzten Testumgebungen vorgestellt.

Die Darstellung der Sensorfusion erfolgt in Kapitel 4. Dies umfasst die Charakterisierung der verwendeten Sensoren, die Auslegung von Propagation und Estimation des Erweiterten Kalman Filters sowie den Filter-Ablauf. In Kapitel 5 wird der Entwurf der flachheitsbasierten Regelung dargestellt. Zu Beginn des Kapitels werden die kinematischen und aerodynamischen Modelle aufgestellt sowie die Schubkalibrierung und der Störgrößen-Beobachter eingeführt. Nach dem Beweis der Flachheit des Systems werden die flachheitsbasierte Zustandsrückführung hergeleitet und die überlagerte Trajektorienfolgeregelung entworfen. Im Anschluss werden die unterschiedlichen Trajektoriengenerierungsmethoden und die Ablaufsteuerung vorgestellt. Das Kapitel endet mit der Präsentation der simulativen Erprobung der Regelung.

Die Ergebnisse der Flugversuche zur Erprobung der Sensorik, der Trajektoriengenerierung und der flachheitsbasierten Regelung werden in Kapitel 6 präsentiert. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in Kapitel 7.

2 Grundlagen

2.1 Koordinatensysteme, Erdmodell und Lagedarstellungen

Im Folgenden werden die Bezugs- und Koordinatensysteme definiert sowie das Erdmodell und die Lagedarstellungen eingeführt, die für Sensorfusion und Regelung erforderlich sind. Zur Notation siehe auch Seite 14.

2.1.1 Bezugs- und Koordinatensysteme

Körperfeste Bezugs- und Koordinatensysteme

Das **körperfeste Bezugssystem** B ist im als starr angenommenen Körper des Quadropters fixiert.

Die Basisvektoren $\vec{b}_x, \vec{b}_y, \vec{b}_z$ des zugehörigen rechtshändigen, orthogonalen **Koordinatensystems** B zeigen im Quadropter nach vorne, rechts (Steuerbord) und unten. Für die Regelung liegt der Ursprung dieses Koordinatensystems im Punkt Z , dem „Zentrum“ des Quadropters, auf der vertikalen Symmetrieachse des Quadropters auf Höhe der Rotorarme. Der Schwerpunkt des Quadropters liegt auch in etwa auf dieser Höhe. Für die Sensorfusion ist es zweckmäßig, den Ursprung von B in die Position der Quadropter-IMU im Punkt U zu legen¹.

Außerdem wird das **körperfeste Koordinatensystem** C eingeführt, dessen Basisvektoren $\vec{c}_x, \vec{c}_y, \vec{c}_z$ entlang der Achsen der IMU ausgerichtet sind. Der Ursprung ist identisch zu dem von B für die Sensorfusion im Punkt U .

Navigationskoordinatensystem

Das **Navigationskoordinatensystem** N , das in der Sensorfusion verwendet wird, hat seinen Ursprung ebenfalls im Punkt U in der IMU und ist auch rechtshändig und orthogonal. Der Ursprung ist damit deckungsgleich zu denen der körperfesten Koordinatensysteme B für die Sensorfusion und C . Die Basisvektoren $\vec{n}_n, \vec{n}_e, \vec{n}_d$ zeigen an der aktuellen Position immer nach Norden, Osten und unten (north, east, down). \vec{n}_d zeigt vertikal nach unten in Richtung Erde, ist parallel zur Schwerebeschleunigung und steht

¹zur Motivation siehe Abschnitt 4.1.1.1, siehe auch (5.9)

orthogonal auf dem WGS84-Ellipsoid, siehe Abbildung 2.1 und Abschnitt 2.1.2. Die Ebene, die \vec{n}_n und \vec{n}_e aufspannen, ist damit parallel zu einer lokalen Tangentialebene am WGS84-Rotationsellipsoid.

Erdfeste Bezugs- und Koordinatensysteme

Für die Erde als **Bezugssystem** E werden drei Koordinatensysteme festgelegt:

Das **Koordinatensystem** E ist rechtshändig und orthogonal mit den Basisvektoren \vec{e}_x , \vec{e}_y , \vec{e}_z und wird ECEF-Koordinatensystem genannt (für earth centered, earth fixed). Der Ursprung dieses Koordinatensystems liegt im Mittelpunkt der Erde bzw. des WGS84-Rotationsellipsoids (siehe Abschnitt 2.1.2). Es ist fix gegenüber der Erde und rotiert mit ihr. Der Basisvektor \vec{e}_x zeigt vom Mittelpunkt auf den Punkt des Äquators, durch den der Nullmeridian läuft. Der Basisvektor \vec{e}_z zeigt zum Nordpol entlang der Symmetrieachse des Rotationsellipsoids, um die sich die Erde dreht.

Das **erdfeste WGS84-Koordinatensystem** W drückt eine Position auf der Erde mithilfe der Koordinaten Breitengrad φ und Längengrad λ sowie der Höhe h über dem WGS84-Rotationsellipsoid aus (siehe Abschnitt 2.1.2). Diese Koordinaten werden auch als LLH-Koordinaten bezeichnet für englisch: latitude φ , longitude λ , height h . Zu beachten ist dabei, dass es sich um geodätische Koordinaten handelt: Der Breitengrad φ ist der Winkel (in einer Meridianebene) zwischen der Äquatorialebene und dem Normalenvektor, der senkrecht auf dem Ellipsoid steht, siehe Abbildung 2.1. Die Verlängerung des Normalenvektors verläuft also im Allgemeinen nicht durch den Mittelpunkt des Ellipsoids.

Als drittes wird das **lokale, erdfeste Koordinatensystem** L eingeführt, das für die Regelung Verwendung findet. Dessen Basisvektoren \vec{l}_n , \vec{l}_e , \vec{l}_d zeigen wie beim Navigationskoordinatensystem N nach Norden, Osten und vertikal nach unten. Der Ursprung von L liegt in der Nähe zu dem von N und B , ist jedoch erdfest.

Inertialsystem

Im **inertialen Bezugssystem** I gelten die newtonschen Axiome. Die IMU misst z. B. Beschleunigungen und Drehraten bezüglich des Inertialsystems. Die Erde ist als beschleunigtes Bezugssystem u. a. durch ihre Rotation um sich selbst und um die Sonne kein Inertialsystem. Je nach Anwendung kann aber u. U. die Bewegung der Erde vernachlässigt werden.

2.1.2 Erdmodell

Das geodätische Referenzsystem WGS84 (World Geodetic System 1984) führt ein Rotationsellipsoid ein, um die Form der Erde anzunähern, siehe Abbildung 2.1. Die große Halbachse des Ellipsoids – der äquatoriale Radius – wird mit a , die kleine Halbachse – der Abstand vom Mittelpunkt des Ellipsoids zu den Polen – wird mit b bezeichnet, siehe

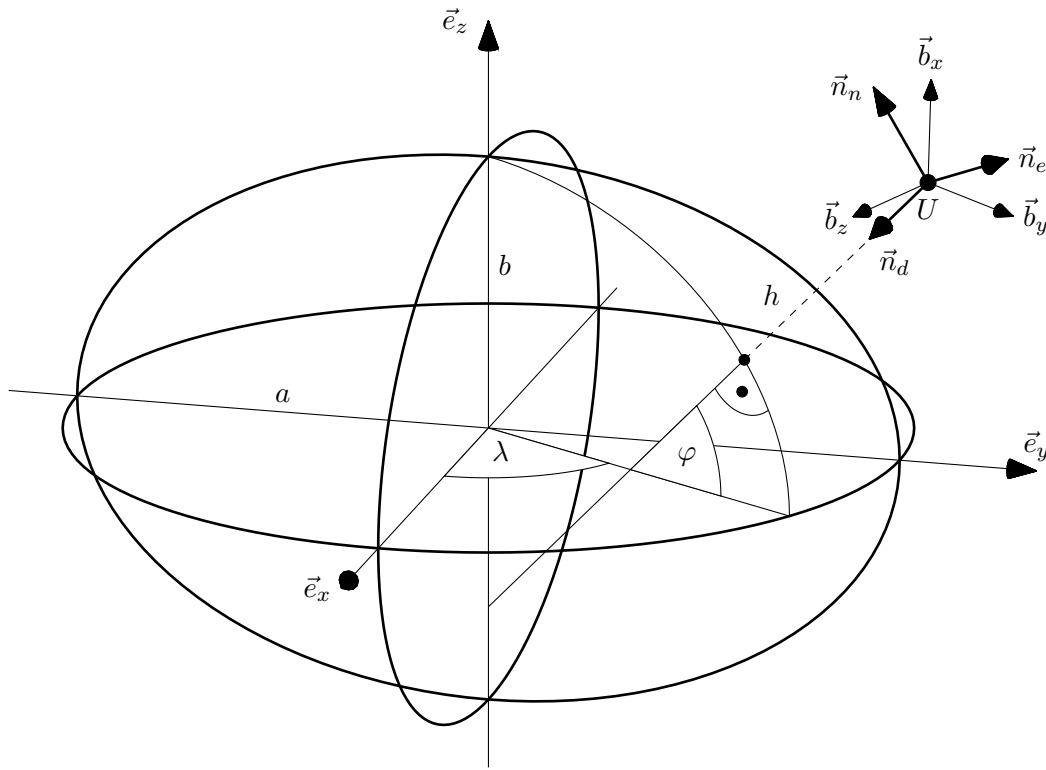


Abbildung 2.1: Erdellipsoid mit sensorfusionsrelevanten Koordinatensystemen: Die Position des Punktes U kann in den erdfesten Koordinatensystemen E und W (WGS84, geodätische Koordinaten φ , λ , h) ausgedrückt werden. Das Navigationskoordinatensystem N und das körperfeste Koordinatensystem B haben ihren Ursprung im Punkt U .

Tabelle 2.1. Die Achse durch die Pole entlang \vec{e}_z ist die Symmetrieachse des Rotationsellipsoids. Um diese Achse rotiert es mit der Erddrehrate Ω .

Der Nord-Süd-Krümmungsradius R_n , der Ost-West-Krümmungsradius R_e und der durchschnittliche Krümmungsradius R_0 an einem Punkt auf dem WGS84-Referenzellipsoid sind abhängig vom Breitengrad φ und den WGS84-Konstanten in Tabelle 2.1 [84, 28]:

$$R_n = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \varphi)^{3/2}} \quad (2.1)$$

$$R_e = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (2.2)$$

$$R_0 = \sqrt{R_n R_e} \quad (2.3)$$

Eine Position relativ zur Erde kann entweder im kartesischen ECEF-Koordinatensystem E oder im WGS84-Koordinatensystem W ausgedrückt werden, vgl. Abschnitt 2.1.1. Das WGS84-Koordinatensystem nutzt die Polarkoordinaten Breitengrad φ und Längengrad λ

Große Halbachse des Ellipsoids	a	6378137,0 m
Kehrwert der Abflachung des Ellipsoids	$\frac{1}{f}$	298,257223563
Kleine Halbachse des Ellipsoids	b	$a(1 - f)$
Exzentrizität des Ellipsoids	e	$\sqrt{f(2 - f)}$
Erddrehrate	Ω	$7,292115 \cdot 10^{-5} \frac{\text{rad}}{\text{s}}$
Geozentrische Gravitationskonstante	GM	$3,986\,004\,418 \cdot 10^{14} \text{ m}^3/\text{s}^2$

Tabelle 2.1: Parameter des WGS84-Referenzsystems

sowie die Höhe h über dem WGS84-Rotationsellipsoid, siehe Abbildung 2.1. Für die Koordinatentransformation der Position eines Punktes U von WGS84/LLH nach ECEF gilt [84]:

$$\vec{p}_U^e = \begin{pmatrix} (R_e + h) \cos \varphi \cos \lambda \\ (R_e + h) \cos \varphi \sin \lambda \\ (R_e(1 - e^2) + h) \sin \varphi \end{pmatrix} \quad (2.4)$$

Die inverse Transformation wird u. a. in [84, 28] dargestellt.

Auf einen Körper wirkt aufgrund der Anziehung der Erdmasse die Gravitationskraft \vec{F}_G (engl.: gravitational force). Die auf die Masse m des Körpers bezogene, spezifische Gravitationskraft der Erde wird mit \vec{g} bezeichnet:

$$\vec{g} = \frac{\vec{F}_G}{m} \quad , \quad g = |\vec{g}| \quad (2.5)$$

Gemäß dem newtonschen Gravitationsgesetz beträgt die spezifische Gravitationskraft der Erde auf einen Körper an der Position \vec{p} relativ zum Schwerpunkt der Erde näherungsweise²:

$$\vec{g} = g \frac{-\vec{p}}{p} \quad \text{mit} \quad g \approx \frac{GM}{p^2} \quad , \quad p = |\vec{p}| \quad (2.6)$$

G ist die Gravitationskonstante und M die Masse der Erde, siehe Tabelle 2.1. Die Gravitationskraft ist also proportional zum Kehrwert des quadrierten Abstandes p vom Erdschwerpunkt.

Die Erde rotiert mit der Erddrehrate Ω (Tab. 2.1) um die Achse \vec{e}_z , die durch die beiden Pole läuft:

$$\vec{\omega}_{ie} = \Omega \vec{e}_z \quad (2.7)$$

In diesem rotierenden Bezugssystem betrachtet, wirkt auf einen Körper, der relativ zur Erde ruht und sich mit ihr dreht, scheinbar eine Zentrifugalbeschleunigung \vec{a}_Z bzw. eine

²Bei dieser Näherung wird eine kugelsymmetrische Masseverteilung der Erde angenommen.

zugehörige Scheinkraft \vec{F}_Z :

$$\begin{aligned}\vec{a}_Z &= \frac{\vec{F}_Z}{m} = -\vec{\omega}_{ie} \times (\vec{\omega}_{ie} \times \vec{p}) \\ |\vec{a}_Z| &= \Omega^2 p_Z \quad \text{mit } p_Z = |\vec{e}_z \times (\vec{e}_z \times \vec{p})|\end{aligned}\tag{2.8}$$

Dabei ist \vec{p} die Position des Körpers relativ zum Mittelpunkt der Erde. $|p_Z|$ ist der Abstand des Körpers von der Rotationsachse. Die Zentrifugalbeschleunigung \vec{a}_Z ist also an den Polen null und am Äquator am größten. Dort wirkt sie genau entgegengesetzt zur spezifischen Gravitationskraft \vec{g} .

Die Summe aus Gravitationskraft \vec{F}_G und Zentrifugalkraft \vec{F}_Z wird als Schwerkraft bezeichnet³. Die auf die Masse m des Körpers bezogene, spezifische Schwerkraft ist daher:

$$\vec{g}_l = \vec{g} + \vec{a}_Z\tag{2.9}$$

\vec{g}_l ist die Beschleunigung des Körpers im freien Fall auf der Erde, daher wird \vec{g}_l auch als Schwerebeschleunigung (engl.: gravity) bezeichnet.

Die Form eines Rotationsellipsoids hat die Erde auch aufgrund der Schwerkraft als Summe von Gravitationskraft und Zentrifugalkraft [84]. So liegt der größte Erdradius a am Äquator, wo die Schwerebeschleunigung am geringsten ist; an den Polen ist die Erde abgeplattet – dort herrscht die höchste Schwerebeschleunigung.

Aus dem gleichen Grund stehen die Schwerebeschleunigung an jedem Punkt des Rotationsellipsoids in guter Näherung genau senkrecht auf der lokalen Tangentialebene [84]. Die Richtung der Schwerebeschleunigung definiert gleichzeitig die Richtung des Basisvektors \vec{n}_d des Navigationskoordinatensystems, vgl. Abbildung 2.1. Ein übliches, einfaches Schwerebeschleunigungsmodell gemäß [84] lautet:

$$\vec{g}_l^n \approx \begin{pmatrix} 0 \\ 0 \\ g_0 (1 + g_1 \sin^2 \varphi + g_2 \sin^2 2\varphi) \left(\frac{R_0}{R_0+h}\right)^2 \end{pmatrix} \quad \text{mit} \quad \begin{aligned} g_0 &= 9,780\,318 \text{ m/s}^2 \\ g_1 &= 5,3024 \cdot 10^{-3} \\ g_2 &= 5,9 \cdot 10^{-6} \end{aligned}\tag{2.10}$$

Tatsächlich treten lokal Abweichungen auf, da die Masse der Erde nicht gleichmäßig verteilt ist.

³Der Begriff „Schwerkraft“ wird nicht immer eindeutig verwendet. So wird er teilweise mit „Gravitationskraft“ gleichgesetzt. Hier soll jedoch die übliche Definition verwendet werden: Die Schwerkraft ist die Summe aus Gravitationskraft und Zentrifugalkraft.

2.1.3 Lagedarstellungen

Zur Darstellung der Lage (Orientierung) zweier Koordinatensysteme A und B zueinander stehen unterschiedliche Beschreibungsformen zur Verfügung.

Zunächst wird der Orientierungsvektor $\vec{\sigma}$ eingeführt als

$$\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)^T \quad \text{mit} \quad \sigma = |\vec{\sigma}| \quad , \quad (2.11)$$

der eine Drehung von B um σ Radianen um die durch $\vec{\sigma}$ beschriebene Achse definiert, durch die B mit A zur Deckung gebracht wird.

Das Lagequaternion \mathbf{q}_b^a wird mit diesem $\vec{\sigma}$ definiert als

$$\mathbf{q}_b^a = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \cos \frac{\sigma}{2} \\ \frac{\sigma_x}{\sigma} \sin \frac{\sigma}{2} \\ \frac{\sigma_y}{\sigma} \sin \frac{\sigma}{2} \\ \frac{\sigma_z}{\sigma} \sin \frac{\sigma}{2} \end{pmatrix} . \quad (2.12)$$

Durch diese Wahl von $q_1 \dots q_4$ ist \mathbf{q}_b^a normiert und hat die Länge $|\mathbf{q}_b^a| = 1$.

Alternativ zum Lagequaternion \mathbf{q}_b^a lässt sich die Lage als Richtungskosinusmatrix \mathbf{C}_b^a ausdrücken [28]:

$$\mathbf{C}_b^a = \begin{pmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_1q_2 + q_3q_4) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{pmatrix} \quad (2.13)$$

Für die inverse Transformation von \mathbf{C}_b^a nach \mathbf{q}_b^a siehe (B.5).

\mathbf{C}_b^a beschreibt die Koordinatentransformation bzw. die passive Rotation vom Koordinatensystem B nach A . \mathbf{C}_b^a transformiert die Koordinaten eines Vektors \vec{v} im Koordinatensystem B in Koordinaten bezüglich A :

$$\vec{v}^a = \mathbf{C}_b^a \vec{v}^b \quad (2.14)$$

Bei dieser Koordinatentransformation verändert sich die Richtung des Vektors nicht, lediglich die Darstellung bzw. das Koordinatensystem wird geändert.

Gleichzeitig beschreibt \mathbf{C}_b^a die aktive Rotation eines Vektors von A nach B :

$$\vec{v}'^a = \mathbf{C}_b^a \vec{v}^a \quad (2.15)$$

Dabei ändert der Vektor seine Richtung, das Koordinatensystem bleibt gleich.

Für eine Eulerwinkel-Darstellung soll die in der Luftfahrt gebräuchliche Form genutzt werden: Die drei Winkel werden als Gierwinkel ψ , Nickwinkel θ und Rollwinkel φ bezeichnet und die Drehungen erfolgen nacheinander gemäß der z - y' - x'' Kardan-Winkel-Konvention:

Das Koordinatensystem A wird in das Koordinatensystem B gedreht, indem die Rotationen in der Reihenfolge Gieren→Nicken→Rollen um die jeweils gedrehten Achsen ausgeführt werden. Die Winkel können aus \mathbf{C}_b^a und mit (2.13) aus \mathbf{q}_b^a wie folgt berechnet werden [28]:

$$\psi = \arctan2(c_{21}, c_{11}) \quad (2.16a)$$

$$\theta = \arcsin(-c_{31}) \quad (2.16b)$$

$$\varphi = \arctan2(c_{32}, c_{33}) \quad (2.16c)$$

Hierbei ist $\arctan2$ die Erweiterung des Arkustangens auf vier Quadranten. Für die inverse Transformation von Eulerwinkeln nach \mathbf{C}_b^a und \mathbf{q}_b^a siehe (B.4) und (B.6).

2.2 Dynamik

Die folgenden Grundlagen zur Kinematik und Kinetik bilden die Basis der Modellbildung für die Sensorfusion und die Regelung.

2.2.1 Rotatorische Kinematik

Die Drehrate (Drehgeschwindigkeit, Winkelgeschwindigkeit) eines Bezugssystems (oder eines Starrkörpers) B in einem Bezugssystem A ist definiert als [42]

$$\vec{\omega}_{ab} = \vec{b}_x \frac{{}^A d\vec{b}_y}{dt} \cdot \vec{b}_z + \vec{b}_y \frac{{}^A d\vec{b}_z}{dt} \cdot \vec{b}_x + \vec{b}_z \frac{{}^A d\vec{b}_x}{dt} \cdot \vec{b}_y \quad . \quad (2.17)$$

$\vec{b}_x, \vec{b}_y, \vec{b}_z$ sind die Basisvektoren des Koordinatensystems B gemäß Abschnitt 2.1.1.

Für die zeitliche Änderung des Lagequaternions \mathbf{q}_b^a aus (2.12) in Abhängigkeit von der Drehrate $\vec{\omega}_{ab}$ gilt [28]:

$$\dot{\mathbf{q}}_b^a = \frac{1}{2} \boldsymbol{\Omega}(\mathbf{q}_b^a) \vec{\omega}_{ab}^b \quad \text{mit} \quad \boldsymbol{\Omega}(\mathbf{q}_b^a) = \begin{pmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{pmatrix} \quad (2.18)$$

Die zeitliche Änderung der Richtungskosinusmatrix \mathbf{C}_b^a aus (2.13) lautet [84, 28]:

$$\dot{\mathbf{C}}_b^a = \mathbf{C}_b^a [\vec{\omega}_{ab}^b \times] \quad (2.19)$$

$[\vec{\omega}_{ab}^b \times]$ ist die kreuzproduktbildende Matrix (B.9).

Die Änderung des Orientierungsvektors $\vec{\sigma}$ aus (2.11) wird durch die Bortzsche Orientierungsvektordifferentialgleichung beschrieben [84]:

$$\dot{\vec{\sigma}} = \vec{\omega}_{ab}^b + \frac{1}{2}\vec{\sigma} \times \vec{\omega}_{ab}^b + \frac{1}{\sigma^2} \left(1 - \frac{\sigma \sin \sigma}{2(1 - \cos \sigma)} \right) \vec{\sigma} \times (\vec{\sigma} \times \vec{\omega}_{ab}^b) \quad (2.20)$$

Die drei Differentialgleichungen (2.18), (2.19), (2.20) beschreiben die gleichen Lageänderungen in den unterschiedlichen Darstellungen.

Die Drehbeschleunigung eines Bezugssystems (oder eines Starrkörpers) B in A ist die zeitliche Ableitung der Drehrate $\vec{\omega}_{ab}$ aus (2.17) in A :

$$\vec{\alpha}_{ab} = \frac{{}^A d\vec{\omega}_{ab}}{dt} \quad (2.21)$$

2.2.2 Translatorische Kinematik

Sei \vec{p}_{op} der Positionsvektor von einem beliebigen, im Bezugssystem A fixierten Punkt O zu einem Punkt P , der sich in A bewegt. Dann ist die Geschwindigkeit des Punktes P in A definiert als [42]:

$$\vec{v}_{ap} = \frac{{}^A d\vec{p}_{op}}{dt} \quad (2.22)$$

Die Beschleunigung des Punktes P in A ist die zeitliche Änderung der Geschwindigkeit \vec{v}_{ap} in A :

$$\vec{a}_{ap} = \frac{{}^A d\vec{v}_{ap}}{dt} \quad (2.23)$$

2.2.3 Kinetik

Für einen Starrkörper B gilt der Drehimpulssatz:

$$\vec{M}_S = \mathbf{J}_S \vec{\alpha}_{ib} + \vec{\omega}_{ib} \times \mathbf{J}_S \vec{\omega}_{ib} \quad (2.24)$$

\mathbf{J}_S ist der konstante Trägheitstensor des Starrkörpers bezüglich seines Schwerpunktes. \vec{M}_S bezeichnet die Summe aller auf den Körper wirkenden äußeren Momente bezüglich des Schwerpunktes, $\vec{\alpha}_{ib}$ die Drehbeschleunigung (2.21) und $\vec{\omega}_{ib}$ die Drehrate (2.17) von B in einem inertialen Bezugssystem I .

Gemäß dem zweiten newtonschen Gesetz gilt für einen Körper B mit konstanter Masse m :

$$\vec{F} = m \vec{a}_{ib} \quad (2.25)$$

Dabei ist \vec{F} die Summe aller auf den Körper wirkenden äußeren Kräfte und \vec{a}_{ib} die Beschleunigung (2.23) des Körperschwerpunktes von B in einem inertialen Bezugssystem I .

2.3 Kalman-Filter

Die Sensorfusion für den Quadrocopter (Kapitel 4) soll aus den zur Verfügung stehenden Sensorsignalen und unter Ausnutzung der bekannten nichtlinearen Systemdynamik den Systemzustand schätzen. Ein geeigneter Algorithmus ist das Erweiterte Kalman-Filter, das hier eingeführt wird. Zunächst wird das grundlegende Kalman-Filter für Systeme mit linearer Dynamik dargestellt, um es dann als Erweitertes Kalman-Filter auf Systeme mit nichtlinearer Dynamik zu übertragen. Im Abschnitt 2.3.3 wird schließlich die Diskretisierung eines kontinuierlichen Modells für die Kalman-Filter vorgestellt. Die Grundlagen der Kalman-Filter basieren auf [28, 84].

2.3.1 Kalman-Filter für lineare Systeme

Die Aufgabe des Kalman-Filters ist es, den Systemzustand \vec{x}_k eines linearen Systems anhand der Messgröße \vec{y}_k und dem Eingang \vec{u}_k zu schätzen. Die zeitdiskreten System- und Messmodelle haben die Form:

$$\vec{x}_{k+1} = \mathbf{\Phi}_k \vec{x}_k + \mathbf{B}_k \vec{u}_k + \mathbf{G}_k \vec{w}_k \quad (2.26)$$

$$\vec{y}_k = \mathbf{H}_k \vec{x}_k + \vec{v}_k \quad (2.27)$$

Das Systemmodell (2.26) propagiert den Systemzustand \vec{x}_k um einen Zeitschritt in Abhängigkeit vom bekannten Eingang \vec{u}_k und vom Systemrauschen \vec{w}_k anhand der Transitionsmatrix $\mathbf{\Phi}_k$ und der Eingangsmatrizen \mathbf{B}_k und \mathbf{G}_k .

Das Messmodell (2.27) beschreibt mit der Messmatrix \mathbf{H}_k die lineare Abhängigkeit der zur Verfügung stehenden Messgröße \vec{y}_k vom Systemzustand \vec{x}_k und vom Messrauschen \vec{v}_k .

Das Kalman-Filter schätzt den Erwartungswert $\hat{\vec{x}}_k$ des realen Systemzustandes \vec{x}_k sowie die Schätzgüte in Form der Schätzfehlerkovarianzmatrix \mathbf{P}_k :

$$\hat{\vec{x}}_k = E[\vec{x}_k] \quad (2.28)$$

$$\mathbf{P}_k = Cov(\vec{x}_k, \hat{\vec{x}}_k) \quad (2.29)$$

Für Erwartungswert und Kovarianzmatrix siehe (A.11) und (A.14). Ein Zirkumflex markiert im Folgenden Schätzwerte in Abgrenzung zum realen Wert. Eine Tilde kennzeichnet gemessene Größen. Zur verwendeten Notation siehe auch „Mathematische Notation“ auf Seite 14.

Das Systemrauschen \vec{w}_k modelliert Modellunsicherheiten und das Messrauschen der Eingangsgröße \vec{u}_k . Das Messrauschen \vec{v}_k beschreibt das Rauschen auf der Messgröße \vec{y}_k . \vec{w}_k und \vec{v}_k werden als zeitdiskrete weiße Rauschprozesse (A.29) mit mittelwertfreien, normalverteilten Wahrscheinlichkeitsdichtefunktionen (A.17) modelliert [28, 84].

Damit sind die Kovarianzmatrizen $Cov(\vec{v}_k, \vec{v}_k)$ und $Cov(\vec{w}_k, \vec{w}_k)$ gemäß (A.32) gleich den Diagonalmatrizen \mathbf{Q}_k und \mathbf{R}_k mit den spektralen Leistungsdichten der diskreten weißen Rauschprozesse \vec{w}_k und \vec{v}_k auf den Diagonalen:

$$E[\vec{w}_k] = \mathbf{0} \quad Cov(\vec{w}_k, \vec{w}_i) = \begin{cases} \mathbf{Q}_k & i = k \\ \mathbf{0} & i \neq k \end{cases} \quad (2.30)$$

$$E[\vec{v}_k] = \mathbf{0} \quad Cov(\vec{v}_k, \vec{v}_i) = \begin{cases} \mathbf{R}_k & i = k \\ \mathbf{0} & i \neq k \end{cases} \quad (2.31)$$

\mathbf{Q}_k und \mathbf{R}_k können gemäß (2.75)f. aus den spektralen Leistungsdichten eines kontinuierlichen weißen Rauschens berechnet werden.

Es wird angenommen, dass System- und Messrauschen unabhängig voneinander sowie unabhängig vom aktuellen Wert und von vorhergehenden Werten des Zustands sind. Dann lauten die Kreuzkovarianzfunktionen (A.20):

$$Cov(\vec{w}_k, \vec{v}_i) = \mathbf{0} \quad \forall k \geq 0, i \geq 0 \quad (2.32)$$

$$Cov(\vec{w}_k, \vec{x}_i) = \mathbf{0} \quad \forall i \leq k, \quad Cov(\vec{v}_k, \vec{x}_i) = \mathbf{0} \quad \forall i \leq k \quad (2.33)$$

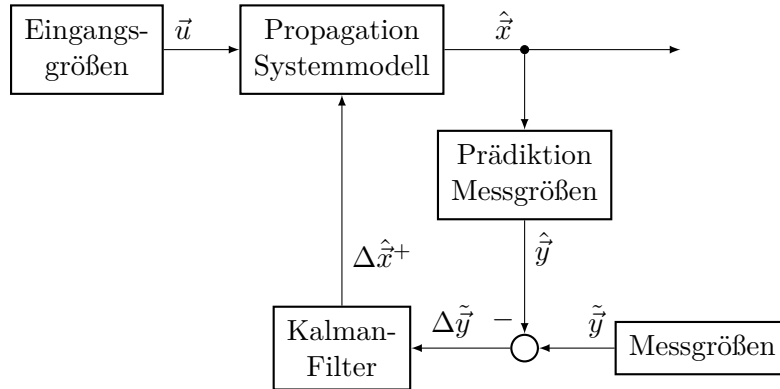


Abbildung 2.2: Algorithmus Kalman-Filter und Erweitertes Kalman-Filter

Der Algorithmus des Kalman-Filters ist in Abbildung 2.2 schematisch dargestellt. Die Gleichungen lauten [84, 28]:

$$\hat{\vec{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\vec{x}}_k^- + \mathbf{B}_k \vec{u}_k \quad (2.34)$$

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (2.35)$$

$$\hat{\vec{y}}_k = \mathbf{H}_k \hat{\vec{x}}_k^- \quad (2.36)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.37)$$

$$\hat{\vec{x}}_k^+ = \hat{\vec{x}}_k^- - \mathbf{K}_k (\hat{\vec{y}}_k - \tilde{\vec{y}}_k) \quad (2.38)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.39)$$

Während (2.34) und (2.35) die Propagation des Kalman-Filters darstellen, beschreiben (2.36) bis (2.39) die Estimation bzw. das Mess-Update.

Ein hochgestelltes Minus wie bei \hat{x}_k^- wird bei A-priori-Schätzwerten vor dem Estimationsschritt, ein hochgestelltes Plus wie \hat{x}_k^+ bei A-posteriori-Schätzwerten nach dem Estimationsschritt verwendet [84]. So ist \mathbf{P}_k^- die Kovarianzmatrix für den A-priori-Schätzwert \hat{x}_k^- und \mathbf{P}_k^+ die für \hat{x}_k^+ .

(2.34) ist die zeitliche Propagation des geschätzten Systemzustandes \hat{x}_k gemäß dem Systemmodell (2.26) mit bekannten Eingangsgrößen \vec{u}_k , jedoch ohne den stochastischen Eingang \vec{w}_k , dessen Realisierung zum Zeitpunkt k unbekannt ist. Neben dem geschätzten Systemzustand wird die Schätzfehlerkovarianzmatrix \mathbf{P}_k in (2.35) unter Nutzung der Matrizen des Systemmodells (2.26) und der Kovarianzmatrix des Systemrauschens \mathbf{Q}_k in der Zeit propagiert.

In (2.36) werden aus dem geschätzten Systemzustand \hat{x}_k gemäß dem Messmodell (2.27) ohne stochastischen Eingang \vec{v}_k die Messgrößen \hat{y}_k geschätzt.

Die Kalman-Gain-Matrix \mathbf{K}_k wird in (2.37) aus der Schätzfehlerkovarianz \mathbf{P}_k , der Messmatrix \mathbf{H}_k und der Kovarianzmatrix des Messrauschens \mathbf{R}_k berechnet.

In (2.38) wird mit dem Messwert \tilde{y}_k und der Kalman-Gain-Matrix \mathbf{K}_k der geschätzte Systemzustand \hat{x}_k^- zu \hat{x}_k^+ korrigiert. Die Gleichung lässt sich wie folgt in ihre Bestandteile zerlegen, die auch in der Abbildung 2.2 wiederzufinden sind:

$$\hat{x}_k^+ = \hat{x}_k^- - \Delta\hat{x}_k^+ \quad \Delta\hat{x}_k^+ = \mathbf{K}_k\Delta\tilde{y}_k \quad \Delta\tilde{y}_k = \hat{y}_k - \tilde{y}_k \quad (2.40)$$

Die Differenz zwischen geschätzten und gemessenen Messgrößen $\hat{y}_k - \tilde{y}_k$ wird als Residualmessgröße $\Delta\tilde{y}_k$ bezeichnet. Durch Multiplikation von $\Delta\tilde{y}_k$ mit \mathbf{K}_k wird die Korrektur $\Delta\hat{x}_k^+$ des Systemzustandes \hat{x}_k^- berechnet. Der Ausdruck $\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k$ in (2.37) ist die Prädiktion der Kovarianz der Residualmessgröße.

In (2.39) wird schließlich die Schätzfehlerkovarianzmatrix \mathbf{P}_k^- anhand \mathbf{K}_k und \mathbf{H}_k zu \mathbf{P}_k^+ korrigiert, sodass \mathbf{P}_k^+ dann die Güte von \hat{x}_k^+ beschreibt. (2.39) lässt sich vereinfachen zu:

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- \quad (2.41)$$

(2.39) und (2.41) sind mathematisch äquivalent. Die sogenannte Joseph-Form (2.39) ist jedoch aus numerischen Gründen zu bevorzugen [84].

Für das lineare System (2.26), (2.27) und unkorreliertes, normalverteiltes, mittelwertfreies weißes System- und Messrauschen ist das Kalman-Filter gemäß (2.34) bis (2.39) optimal mit dem kleinstmöglichen quadratischen Mittel der Schätzfehler [84].

2.3.1.1 Interpretation der Kalman-Gain-Matrix

Der Einfluss von \mathbf{P}_k und \mathbf{R}_k , also der geschätzten Güte von Schätzung und Messung, auf die Kalman-Gain-Matrix \mathbf{K}_k gemäß (2.37) wird besonders anschaulich bei einem System mit nur einem Zustand, der direkt gemessen werden kann [84]. Dann sind \mathbf{P}_k und \mathbf{R}_k skalare Größen und $\hat{y}_k = \hat{x}_k$ bzw. $\mathbf{H}_k = 1$. (2.37) lässt sich jetzt schreiben als

$$\mathbf{K}_k = \frac{\mathbf{P}_k^-}{\mathbf{P}_k^- + \mathbf{R}_k} \quad . \quad (2.42)$$

(a) Für einen perfekten Messwert gilt $\mathbf{R}_k \rightarrow 0$ und damit $\mathbf{K}_k \rightarrow 1$. (2.38) vereinfacht sich in dem Fall zu $\hat{x}_k^+ = \tilde{y}_k$ und der Messwert ersetzt den alten Schätzwert vollständig. Analog wird (2.41) zu $\mathbf{P}_k^+ = 0$, die neue Zustandsschätzung ist perfekt.

(b) Im gegenteiligen Fall, bei einem sehr schlechten Messwert, geht $\mathbf{R}_k \rightarrow \infty$. Mit $\mathbf{K}_k \rightarrow 0$ wird dann (2.38) zu $\hat{x}_k^+ = \hat{x}_k^-$, der alte Schätzwert bleibt erhalten und der Messwert nicht benutzt. Für (2.41) gilt dann $\mathbf{P}_k^+ = \mathbf{P}_k^-$, die Schätzgüte bleibt erhalten.

(c) Bei bereits perfekter Zustandsschätzung, d.h. verschwindender Schätzfehlervarianz $\mathbf{P}_k^- \rightarrow 0$ strebt $\mathbf{K}_k \rightarrow 0$. (2.38) wird wieder zu $\hat{x}_k^+ = \hat{x}_k^-$, der alte Schätzwert bleibt erhalten und ein fehlerbehafteter Messwert wird nicht benutzt. Die Schätzfehlervarianz gemäß (2.41) bleibt null $\mathbf{P}_k^+ = 0$.

(d) Zur Initialisierung des Kalman-Filters wird der Zustand auf den verfügbaren Messwert gesetzt: $\hat{x}_k^+ = \tilde{y}_k$. Dies entspricht (2.38) mit einem $\mathbf{K}_k = 1$. Wird dieses $\mathbf{K}_k = 1$ in (2.39) eingesetzt, wird der Term mit \mathbf{P}_k^- zu null und $\mathbf{P}_k^+ = \mathbf{R}_k$. Die Schätzfehlerkovarianz wird also mit der Messkovarianz initialisiert. Dieser Fall ist von dem Fall mit perfektem Messwert (a) zu unterscheiden: Dort gilt auch $\mathbf{K}_k \rightarrow 1$, aber mit $\mathbf{R}_k \rightarrow 0$ wird $\mathbf{P}_k^+ = 0$. Dies ist für eine Initialisierung ungeeignet, da dann zukünftige Messwerte verworfen werden, siehe (c).

2.3.2 Erweitertes Kalman-Filter

Während das Kalman-Filter den Zustand eines linearen Systems schätzt, berücksichtigt das Erweiterte Kalman-Filter (extended Kalman filter) nichtlineare Systemdynamik. Die zeitdiskreten, nichtlinearen System- und Messmodelle haben die allgemeine Form:

$$\vec{x}_{k+1} = \vec{f}_k(\vec{x}_k, \vec{u}_k) + \vec{g}_k(\vec{w}_k) \quad (2.43)$$

$$\vec{y}_k = \vec{h}_k(\vec{x}_k) + \vec{v}_k \quad (2.44)$$

Die Größen \vec{x}_k , \vec{u}_k , \vec{w}_k , \vec{y}_k und \vec{v}_k haben die gleiche Bedeutung wie in (2.26)f. Die Propagation des Systemzustandes und die Prädiktion der Messgrößen erfolgt im Erweiterten Kalman-Filter anhand dieser nichtlinearen System- und Messmodelle.

Die übrigen Schritte des Erweiterten Kalman-Filters – die Berechnung der Kalman-Gain-Matrix, die Propagation und Korrektur der Schätzfehlerkovarianzmatrix – sind wie beim

Kalman-Filter linear. Sie werden jedoch nicht mit einem geschätzten Systemzustand $\hat{\vec{x}}_k$, sondern mit einem geschätzten Fehler $\Delta\hat{\vec{x}}_k$ berechnet. Im Erweiterten Kalman-Filter ist quasi ein lineares Kalman-Filter für $\Delta\hat{\vec{x}}_k$ enthalten – dies wird unten weiter verdeutlicht, siehe „Propagation und Estimation des Schätzfehlerzustandes“ auf Seite 36.

Das Erweiterte Kalman-Filter schätzt $\Delta\hat{\vec{x}}_k$ als Erwartungswert des realen Schätzfehlers $\Delta\vec{x}_k$ sowie die Schätzgüte von $\Delta\hat{\vec{x}}_k$ in Form der Schätzfehlerkovarianzmatrix \mathbf{P}_k :

$$\Delta\hat{\vec{x}}_k = E[\Delta\vec{x}_k] \quad (2.45)$$

$$\mathbf{P}_k = Cov(\Delta\vec{x}_k, \Delta\hat{\vec{x}}_k) \quad (2.46)$$

Der reale Schätzfehler $\Delta\vec{x}_k$ wird gemäß seinem Namen zu null, wenn der geschätzte Systemzustand $\hat{\vec{x}}_k$ gleich dem wirklichen Systemzustand \vec{x}_k ist:

$$\Delta\vec{x}_k = 0 \quad \Leftrightarrow \quad \hat{\vec{x}}_k = \vec{x}_k \quad (2.47)$$

Eine mögliche Wahl für den Schätzfehler ist die Differenz zwischen geschätztem Systemzustand $\hat{\vec{x}}_k$ und dem wirklichen Systemzustand \vec{x}_k :

$$\vec{x}_k = \hat{\vec{x}}_k - \Delta\vec{x}_k \quad (2.48)$$

Es kann jedoch sinnvoll sein, den Schätzfehler in anderen Koordinaten und nicht einfach als Abweichungsgröße zum Systemzustand \vec{x}_k auszudrücken. Dies ist auch für das Schätzfehlermodell der vorliegenden Anwendung der Fall, siehe Abschnitt 4.4.1. Die Transformation zur Korrektur des Systemzustandes anhand des Schätzfehlers wird mit \vec{f}_c bezeichnet:

$$\vec{x}_k = \vec{f}_c(\hat{\vec{x}}_k, \Delta\vec{x}_k) \quad (2.49)$$

Für \vec{f}_c gilt ebenfalls (2.47): Für $\Delta\vec{x}_k = 0$ ist \vec{f}_c eine identische Abbildung: $\vec{x}_k = \vec{f}_c(\vec{x}_k, 0)$.

Für die Schritte des Erweiterten Kalman-Filters, die mit dem Schätzfehlerzustand $\Delta\hat{\vec{x}}_k$ rechnen, wird eine linearisierte Systembeschreibung benötigt. Diese wird hier als lineare Schätzfehler- und Messmodelle eingeführt:

$$\Delta\vec{x}_{k+1} = \mathbf{\Phi}_k \Delta\vec{x}_k + \mathbf{G}_k \vec{w}_k \quad (2.50)$$

$$\Delta\vec{y}_k = \mathbf{H}_k \Delta\vec{x}_k + \vec{v}_k \quad (2.51)$$

Sie haben die gleiche Form wie das lineare System- und Messmodell (2.26)f. des linearen Kalman-Filters, jedoch mit Schätzfehlerzustand $\Delta\vec{x}_k$ als Systemzustand und Messgröße $\Delta\vec{y}_k$. Neben dem Einfluss des Messrauschens $\mathbf{G}_k \vec{w}_k$ wird kein zusätzlicher Eingangsfehler $\mathbf{B}_k \Delta\vec{u}_k$ benötigt.

Der Algorithmus des Erweiterten Kalman-Filters lässt sich durch folgende Gleichungen beschreiben [84, 28], vgl. auch Abbildung 2.2:

$$\hat{x}_{k+1}^- = \vec{f}_k(\hat{x}_k, \vec{u}_k) \quad (2.52)$$

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (2.53)$$

$$\hat{y}_k = \vec{h}_k(\hat{x}_k^-) \quad (2.54)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.55)$$

$$\Delta \hat{x}_k^+ = \mathbf{K}_k \Delta \tilde{y}_k \quad \text{mit} \quad \Delta \tilde{y}_k = \hat{y}_k - \tilde{y}_k \quad (2.56)$$

$$\hat{x}_k^+ = \vec{f}_c(\hat{x}_k^-, \Delta \hat{x}_k^+) \quad (2.57)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.58)$$

Der Algorithmus des Erweiterten Kalman-Filters weist große Ähnlichkeit zu dem des linearen Kalman-Filters (2.34)ff. auf. Der wesentliche Unterschied ist, dass beim Erweiterten Kalman-Filter die Propagation des geschätzten Systemzustandes \hat{x}_k in (2.52) und die Prädiktion der Messgrößen \hat{y} in (2.54) nicht anhand der linearen System- und Messmodelle (2.26)f., sondern anhand der nichtlinearen Modelle (2.43)f. durchgeführt wird.

Die Propagation der Schätzfehlerkovarianzmatrix \mathbf{P}_k in (2.53), die Berechnung der Kalman-Gain-Matrix \mathbf{K}_k in (2.55) und die Korrektur der Schätzfehlerkovarianzmatrix zu \mathbf{P}_k^+ in (2.58) haben die gleiche Form wie im linearen Kalman-Filter, vgl. (2.35), (2.37), (2.39). Zu beachten ist jedoch, dass sich die Matrizen auf die linearisierten Systembeschreibungen beziehen: die Transitions- und Eingangsmatrizen $\mathbf{\Phi}_k$ und \mathbf{G}_k gehören zum linearisierten Schätzfehlermodell (2.50), die Messmatrix \mathbf{H}_k zum linearisierten Messmodell (2.51). Um den Linearisierungsfehler klein zu halten, werden $\mathbf{\Phi}_k$ und \mathbf{G}_k vor der Verwendung in (2.53) und \mathbf{H}_k vor der Berechnung von \mathbf{K}_k (2.55) und \mathbf{P}_k^+ (2.58) am Punkt (\hat{x}_k, \vec{u}_k) regelmäßig neu gebildet, siehe Abschnitt 2.3.3.

Das System- und Messrauschen im Zeitdiskreten \vec{w}_k, \vec{v}_k mit ihren Kovarianzmatrizen \mathbf{Q}_k und \mathbf{R}_k wird wie beim linearen Kalman-Filter modelliert, siehe (2.30) bis (2.33).

Aus der Residualmessgröße $\Delta \tilde{y}_k = \hat{y}_k - \tilde{y}_k$ berechnet das Erweiterte Kalman-Filter gemäß (2.56) durch Multiplikation mit \mathbf{K}_k dann den Schätzfehlerzustand $\Delta \hat{x}_k^+$, vgl. lineares Kalman-Filter in (2.40). Mit $\Delta \hat{x}_k^+$ wird schließlich in (2.57) der geschätzte Systemzustand \hat{x}_k^- gemäß (2.49) zu \hat{x}_k^+ korrigiert.

Propagation und Estimation des Schätzfehlerzustandes

Im Folgenden soll auf die Besonderheiten der Propagation und Estimation des Fehlerzustandes $\Delta \hat{x}_k$ im Erweiterten Kalman-Filter eingegangen werden.

Zum Zeitpunkt $k = 0$ wird der Schätzwert \hat{x}_0 mit dem Erwartungswert des realen Systemzustandes initialisiert: $\hat{x}_k^+ = E[\vec{x}_k]$. Damit ist $\Delta \hat{x}_k^+ = 0$ und die Propagation gemäß

linearem Kalman-Filter (2.34) anhand des linearisierten Systemmodells (2.50) wird trivial und ergibt sich zu null:

$$\Delta \hat{x}_{k+1}^- = \Phi_k \Delta \hat{x}_k^- = 0 \quad (2.59)$$

Die Propagation von $\Delta \hat{x}_k^-$ muss also neben der nichtlinearen Propagation von \hat{x}_k^- (2.52) nicht explizit durchgeführt werden.

Der Estimationsschritt (2.56) lässt sich ausführlich in der Form des Zustands-Updates des linearen Kalman-Filters (2.38) und (2.36) mit $\Delta \hat{x}_k^-$ anstatt \hat{x}_k^- schreiben als:

$$\Delta \hat{x}_k^+ = \Delta \hat{x}_k^- - \mathbf{K}_k \left(\mathbf{H}_k \Delta \hat{x}_k^- - \Delta \tilde{y}_k \right) \quad (2.60)$$

Mit $\Delta \hat{x}_{k+1}^- = 0$ als Ergebnis der Propagation (2.59) im vorhergehenden Zeitschritt wird hier (Indexverschiebung $k+1 \rightarrow k$)

$$\Delta \hat{x}_k^- = 0 \quad . \quad (2.61)$$

Damit vereinfacht sich (2.60) zur oben genannten Form (2.56)

$$\Delta \hat{x}_k^+ = \mathbf{K}_k \Delta \tilde{y}_k$$

Die Schreibweise (2.60) verdeutlicht außerdem den „Messwert“ des Erweiterten Kalman-Filters: während $\Delta \tilde{y}_k = \hat{y}_k - \tilde{y}_k$ der Messwert der Residualmessgröße ist, ist $\mathbf{H}_k \Delta \hat{x}_k^-$ ihr Schätzwert $\Delta \hat{y}_k$.

Nach Nutzung von $\Delta \hat{x}_k^+$ zur Korrektur von \hat{x}_k^- in (2.57) ist der geschätzte Systemzustand \hat{x}_k^+ wieder gleich dem Erwartungswert des realen Systemzustandes: $\hat{x}_k^+ = E[\vec{x}_k]$. Daher gilt jetzt:

$$\Delta \hat{x}_k^+ = 0 \quad (2.62)$$

Damit ist der gleiche Wert wie bei der oben genannten Initialisierung zum Zeitpunkt $k=0$ erreicht und die Zusammenhänge wiederholen sich bei jedem Zeitschritt k .

2.3.3 Modelldiskretisierung

Das lineare Kalman-Filter benötigt ein lineares, zeitdiskretes Modell (2.26)f. Für das erweiterte Kalman-Filter ist ein lineares, zeitdiskretes Modell als Schätzfehler- und Messmodell (2.50)f. erforderlich. Wenn das Systemmodell in zeitkontinuierlicher Form gegeben ist, muss es dazu diskretisiert werden. Diese Diskretisierung und die Diskretisierung des kontinuierlichen System- und Messrauschens $\vec{w}(t)$ und $\vec{v}(t)$ soll im Folgenden dargestellt werden.

Gegeben sei ein zeitkontinuierliches Modell in der Form:

$$\dot{\vec{x}}(t) = \mathbf{F}(t)\vec{x}(t) + \mathbf{B}(t)\vec{u}(t) + \mathbf{G}(t)\vec{w}(t) \quad (2.63)$$

$$\vec{y}(t) = \mathbf{H}(t)\vec{x}(t) + \vec{v}(t) \quad (2.64)$$

Die beiden Eingangsterme $\mathbf{B}(t)\vec{u}(t)$ und $\mathbf{G}(t)\vec{w}(t)$ werden in der folgenden Darstellung zu einem Term zusammengefasst:

$$\mathbf{B}^*(t)\vec{u}^*(t) = \begin{pmatrix} \mathbf{B}(t) & \mathbf{G}(t) \end{pmatrix} \begin{pmatrix} \vec{u}(t) \\ \vec{w}(t) \end{pmatrix} \quad (2.65)$$

Wenn $\mathbf{F}(t)$ im Zeitintervall $t \in [kT_s, (k+1)T_s]$ als konstant angenommen werden kann, lässt sich der zeitdiskrete Zustand $\vec{x}_{k+1} = \vec{x}((k+1)T_s)$ wie folgt aus dem vorhergehenden Zustand $\vec{x}_k = \vec{x}(kT_s)$ berechnen [28, 1]:

$$\vec{x}_{k+1} = e^{\mathbf{F}T_s}\vec{x}_k + \int_{kT_s}^{(k+1)T_s} e^{\mathbf{F}((k+1)T_s-\tau)}\mathbf{B}^*(\tau)\vec{u}^*(\tau) d\tau \quad (2.66)$$

Unter der Annahme, dass $\mathbf{B}^*(t)$ und $\vec{u}^*(t)$ im Zeitintervall $t \in [kT_s, (k+1)T_s]$ auch als konstant betrachtet werden können, kann (2.66) zum zeitdiskreten Modell vereinfacht werden [1]:

$$\vec{x}_{k+1} = \mathbf{\Phi}_k\vec{x}_k + \mathbf{B}_k^*\vec{u}_k^* \quad \text{mit} \quad \mathbf{\Phi}_k = e^{\mathbf{F}T_s}, \quad \mathbf{B}_k^* = (\mathbf{\Phi}_k - \mathbf{I})\mathbf{F}^{-1}\mathbf{B}^* \quad (2.67)$$

mit der Transitionsmatrix $\mathbf{\Phi}_k$ und der Eingangsmatrix \mathbf{B}_k^* . Das Matrixexponential $e^{\mathbf{F}T_s}$ ist definiert durch die Reihe:

$$e^{\mathbf{F}T_s} = \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T_s^k}{k!} \quad (2.68)$$

Es lässt sich zeigen, dass die Matrizen $\mathbf{\Phi}_k$ und \mathbf{B}_k^* durch die numerische Bildung des Matrixexponentials $e^{\mathbf{M}T_s}$ berechnet werden können:

$$\begin{pmatrix} \mathbf{\Phi}_k & \mathbf{B}_k^* \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = e^{\mathbf{M}T_s} \quad \text{mit} \quad \mathbf{M} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^* \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (2.69)$$

In der Berechnung des Matrixexponentials wird ein numerisch optimierter MATLAB-Algorithmus gemäß [37] genutzt.

Die Diskretisierung von (2.64) lautet also:

$$\vec{x}_{k+1} = \mathbf{\Phi}_k\vec{x}_k + \mathbf{B}_k\vec{u}_k + \mathbf{G}_k\vec{w}_k \quad \text{mit} \quad (\mathbf{B}_k, \mathbf{G}_k) = \mathbf{B}_k^* \quad (2.70)$$

Damit lassen sich nun aus den Matrizen des zeitkontinuierlichen Modells (2.64) zu jedem Zeitschritt die Systemmatrizen $\mathbf{\Phi}_k$, \mathbf{B}_k , \mathbf{G}_k des zeitdiskreten Systemmodells (2.26) für das lineare Kalman-Filter berechnen. Gleiches gilt für die Systemmatrizen $\mathbf{\Phi}_k$ und \mathbf{G}_k des zeitdiskreten Schätzfehlermodells (2.50) für das Erweiterte Kalman-Filter (dort nur mit Eingang \vec{w}).

Die Diskretisierung der Ausgangsgleichung (2.64) lautet [1]:

$$\vec{y}_k = \mathbf{H}_k\vec{x}_k + \vec{v}_k \quad \text{mit} \quad \mathbf{H}_k = \mathbf{H}(t) \quad (2.71)$$

Die Ausgangsmatrix \mathbf{H} und eine gedachte Einheitsmatrix \mathbf{I} vor $\vec{v}(t)$ bleiben also erhalten.

Die Messmatrix \mathbf{H}_k der zeitdiskreten Messmodelle des linearen Kalman-Filters (2.27) und des Erweiterten Kalman-Filters (2.51) sind mit (2.71) also gleich der Matrix $\mathbf{H}(t)$ der kontinuierlichen Ausgangsgleichung (2.64).

Im Folgenden wird die Diskretisierung des kontinuierlichen System- und Messrauschens $\vec{w}(t)$ und $\vec{v}(t)$ in (2.63)f. erläutert:

Das System- und Messrauschen $\vec{w}(t)$ und $\vec{v}(t)$ wird als kontinuierliches weißes Rauschen (A.25) mit mittelwertfreier, normalverteilter Wahrscheinlichkeitsdichtefunktion (A.17) modelliert [28, 84], vgl. Modell des diskreten System- und Messrauschens in (2.30)f.

Die Kovarianzfunktionen $Cov(\vec{w}(t), \vec{w}(t+\tau))$ und $Cov(\vec{v}(t), \vec{v}(t+\tau))$ können gemäß (A.28) als Funktion der Diagonalmatrizen \mathbf{Q} und \mathbf{R} geschrieben werden, deren Diagonale aus den spektralen Leistungsdichten der kontinuierlichen weißen Rauschprozesse $\vec{w}(t)$ und $\vec{v}(t)$ besteht. Erwartungswerte und Kovarianzfunktionen des kontinuierlichen System- und Messrauschens $\vec{w}(t)$ und $\vec{v}(t)$ können also geschrieben werden als:

$$\mu_{\vec{w}} = E[\vec{w}(t)] = \mathbf{0} \quad Cov(\vec{w}(t), \vec{w}(t+\tau)) = \mathbf{Q}\delta(\tau) \quad (2.72)$$

$$\mu_{\vec{v}} = E[\vec{v}(t)] = \mathbf{0} \quad Cov(\vec{v}(t), \vec{v}(t+\tau)) = \mathbf{R}\delta(\tau) \quad (2.73)$$

Zur Bildung des zeitdiskreten Rauschterms \vec{w}_k in (2.70) wird aufgrund der Integration von \vec{u}^* ($= \vec{w}$) in (2.66) das Integral des zeitkontinuierlichen Rauschterms $\vec{w}(t)$ (2.72) über einen Zeitschritt angesetzt [84]:

$$\vec{w}_k = \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} \vec{w}(\tau) d\tau \quad (2.74)$$

Mit diesem \vec{w}_k kann die Kovarianzmatrix \mathbf{Q}_k (2.30) des Systemrauschens im Zeitdiskreten wie folgt aus den spektralen Leistungsdichten \mathbf{Q} (2.72) des kontinuierlichen Systemrauschens berechnet werden [84]:

$$\mathbf{Q}_k = \frac{1}{T_s} \mathbf{Q} \quad (2.75)$$

Geht man für das diskrete Messrauschen \vec{v}_k auch von einer Mittelwertbildung wie in (2.74) aus, gilt für die Kovarianzmatrix \mathbf{R}_k (2.31) des Messrauschens im Zeitdiskreten als Funktion der spektralen Leistungsdichten \mathbf{R} (2.73) des kontinuierlichen Messrauschens:

$$\mathbf{R}_k = \frac{1}{T_s} \mathbf{R} \quad (2.76)$$

2.4 Flache Systeme

Der Kamera-Quadrokopter hat die Aufgabe, bewegte Objekte automatisch zu verfolgen, vgl. Abschnitt 1.3. Dazu wird eine Trajektorienfolgeregelung benötigt. Eine Klasse der nichtlinearen Systeme, die sich besonders für den Trajektorienfolgeregelungs-Entwurf eignet, ist die der flachen Systeme. Da die Quadrokopterdynamik flach ist (vgl. Beweis in Abschnitt 5.2.1), soll dies genutzt und in Kapitel 5 eine flachheitsbasierte Regelung entworfen werden.

Im Folgenden wird daher der Begriff der Flachheit eingeführt und auf die Eigenschaften von flachen Systemen eingegangen. Für die Trajektorienfolgeregelung wird der Zusammenhang zur exakten Eingangs-/Zustandslinearisierung hergestellt sowie der Steuerungs- und Trajektorienfolgeregelungs-Entwurf für flache Systeme vorgestellt.

2.4.1 Flachheit

Der Begriff der (*differentiellen*) *Flachheit* wurde 1992 von M. Fliess, J. Lévine, P. Martin und P. Rouchon zunächst im Rahmen der Differentialalgebra eingeführt [30, 29] und dann fortlaufend erweitert.

Nun folgt die Definition eines *flachen Systems* und eines *flachen Ausgangs* gemäß [66, 65, 67, 58, 29, 31]. Um die Anschaulichkeit zu erhöhen, wird für das System eine Zustandsraumdarstellung gewählt. Die Flachheitseigenschaft ist jedoch unabhängig von der Systemdarstellung und lässt sich auch z. B. für implizite Systemdarstellungen formulieren.

Definition 2.1 Flaches System

Ein nichtlineares System mit Zustand \vec{x} und Eingang \vec{u} in der Form

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}) \quad \text{mit} \quad \vec{x} = (x_1, \dots, x_n)^T, \quad \vec{u} = (u_1, \dots, u_m)^T \quad (2.77)$$

ist **differentiell flach** oder kurz **flach**, wenn es einen (fiktiven) Ausgang

$$\vec{y} = (y_1, \dots, y_m)^T \quad \text{mit} \quad \dim \vec{y} = \dim \vec{u} = m \quad (2.78)$$

gibt, der folgende Bedingungen, zumindest lokal um einen Punkt \vec{x} , erfüllt:

- 1) Der Ausgang \vec{y} lässt sich als Funktion des Zustands \vec{x} , des Eingangs \vec{u} und einer endlichen Anzahl von Zeitableitungen $u_i^{(k_i)}$ mit $k_i = 1, \dots, \alpha_i$ ausdrücken, d. h. es gilt:

$$\begin{aligned} \vec{y} &= \vec{\phi} \left(\vec{x}, u_1, \dot{u}_1, \dots, u_1^{(\alpha_1)}, \dots, u_m, \dot{u}_m, \dots, u_m^{(\alpha_m)} \right) \\ &= \vec{\phi} \left(\vec{x}, \vec{u}, \dot{\vec{u}}, \ddot{\vec{u}}, \dots, \vec{u}^{(\alpha)} \right) \end{aligned} \quad (2.79)$$

- 2) Die Zustandsvariable \vec{x} und der Eingang \vec{u} lassen sich als Funktion des Ausgangs \vec{y} und einer endlichen Anzahl von Zeitableitungen $y_i^{(k_i)}$ mit $k_i = 1, \dots, \beta_i$ ausdrücken, d. h. es gilt:

$$\vec{x} = \vec{\psi}_1 \left(y_1, \dot{y}_1, \dots, y_1^{(\beta_1-1)}, \dots, y_m, \dots, y_m^{(\beta_m-1)} \right) = \vec{\psi}_1 \left(\vec{y}, \dot{\vec{y}}, \dots, \vec{y}^{(\beta-1)} \right) \quad (2.80a)$$

$$\vec{u} = \vec{\psi}_2 \left(y_1, \dot{y}_1, \dots, y_1^{(\beta_1)}, \dots, y_m, \dots, y_m^{(\beta_m)} \right) = \vec{\psi}_2 \left(\vec{y}, \dot{\vec{y}}, \dots, \vec{y}^{(\beta)} \right) \quad (2.80b)$$

Sind diese Bedingungen erfüllt, dann ist \vec{y} ein **flacher Ausgang** des Systems (2.77).

Der Ausgang ist in (2.78) so gewählt, dass die Anzahl seiner Komponenten gleich der Anzahl m der differentiell unabhängigen Eingangsgrößen \vec{u} des flachen Systems ist. Aufgrund (2.80b) folgt daraus, dass die Komponenten von \vec{y} differentiell unabhängig sind [31, 65], d. h. es existiert keine Differentialgleichung der Form

$$\vec{\varphi}(\vec{y}, \dot{\vec{y}}, \ddot{\vec{y}}, \dots, \vec{y}^{(\gamma)}) = 0 \quad . \quad (2.81)$$

Aus der Definition flacher Systeme folgen direkt einige Eigenschaften. Gemäß (2.80) lassen sich sämtliche Zustands- und Eingangsgrößen durch den flachen Ausgang und dessen zeitliche Ableitungen darstellen⁴. Das bedeutet, dass durch die Trajektorien des flachen Ausgangs die gesamte Dynamik des Systems festgelegt ist. Der flache Ausgang beschreibt das dynamische Verhalten des Systems vollständig [66]. Insbesondere lassen sich aus den ausreichend oft differenzierbaren Solltrajektorien \vec{y}_s für den flachen Ausgang \vec{y} des nicht-linearen Systems diejenigen Trajektorien berechnen, die man auf den Eingang \vec{u} geben muss, damit der Ausgang exakt den Solltrajektorien folgt. Dabei lassen sich gemäß (2.81) die Komponenten von \vec{y} unabhängig voneinander vorgeben, sie sind differentiell unabhängig. (2.80b) führt also unmittelbar zu einer geeigneten Vorsteuerung zur Trajektorienfolge des flachen Ausgangs:

$$\vec{u} = \vec{\psi}_2(\vec{y}_s, \dot{\vec{y}}_s, \ddot{\vec{y}}_s, \dots, \vec{y}_s^{(\beta)}) \quad (2.82)$$

Siehe auch Abbildung 2.3. (2.82) wird auch als *inverses System* bezeichnet [66].

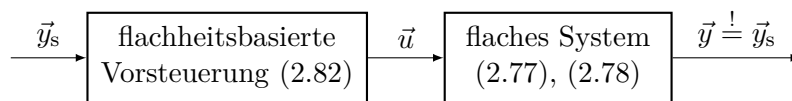


Abbildung 2.3: Flachheitsbasierte Vorsteuerung

Die Vorsteuerung (2.82) ist zur Trajektorienfolge geeignet, wenn das System (2.77) stabil ist, die Modellierungsfehler und Störungen ausreichend klein und die Anfangswerte des Systems hinreichend genau bekannt sind. Falls dies nicht der Fall ist, ist eine Trajektorienfolge-Regelung notwendig, siehe Abschnitt 2.4.3.

⁴Daher wird auch regelmäßig der Zusatz „*differentiell*“ im Zusammenhang mit *Flachheit* verwendet. Zur Berechnung der Systemgrößen wird die Differentiation, nicht aber die Integration benötigt [67].

Steuerbarkeit und Beobachtbarkeit Der Zusammenhang zur *Steuerbarkeit* [1] linearer Systeme ist offensichtlich: Flache nichtlineare Systeme können als Verallgemeinerung der linearen steuerbaren Systeme aufgefasst werden [66, 29]. Für lineare Systeme ist Flachheit und Steuerbarkeit identisch: Lineare Systeme sind genau dann flach, wenn sie steuerbar sind und umgekehrt [57, 1].

Flache Systeme sind zudem über den flachen Ausgang *beobachtbar*: Gemäß (2.80a) kann der zeitliche Verlauf des Zustands \vec{x} aus dem zeitlichen Verlauf des flachen Ausgangs \vec{y} rekonstruiert werden.

Bestimmung eines flachen Ausgangs Ein flacher Ausgang ist in der Regel nicht eindeutig und hat viele Alternativen, die ineinander umgerechnet werden können [66, 57]. Kandidaten für die Komponenten des flachen Ausgangs müssen meist heuristisch gesucht und die Flachheit muss dann durch Überprüfung der Bedingungen der Definition 2.1 bewiesen werden. Vielversprechend sind oft die Kandidaten, die inklusive ihrer Ableitungen das dynamische Verhalten des Systems vollständig beschreiben [66]. Idealerweise sind die Komponenten des flachen Ausgangs die Regelgrößen. Dann ist dieser flache Ausgang besonders geeignet für die Trajektorienfolgeregelung. Für einige nichtlineare Systemklassen gibt es notwendige und/oder hinreichende Kriterien für die Existenz eines flachen Ausgangs und zielgerichtete Verfahren für dessen Suche, siehe z. B. [65, 48].

2.4.2 Exakte Eingangs-/Zustandslinearisierung

Die flachen Systeme stehen im Zusammenhang mit der Klasse der exakt *eingangs-/zustandslinearisierbaren Systeme*. So lässt sich jedes flache System durch eine *quasi-statische* und eine *endogene Zustandsrückführung* exakt linearisieren [65]. Die quasi-statische und die endogene Zustandsrückführung gehören zu den *dynamischen Zustandsrückführungen*. Durch die dynamische Zustandsrückführung und eine Zustandstransformation wird das nichtlineare System in ein lineares System überführt, das das exakt gleiche dynamische Verhalten aufweist. Es erfolgt keine Näherung wie z. B. bei einer Linearisierung durch Taylor-Reihenentwicklung. Für das so exakt linearisierte System können dann die gängigen Regelungsverfahren für lineare Systeme verwendet werden, was z. B. auch die Trajektorienfolgeregelung vereinfacht. Von der Eingangs-/Zustandslinearisierung zu differenzieren ist die *Eingangs-/Ausgangslinearisierung*, bei der z. B. noch eine (ggf. instabile) interne Dynamik vorhanden sein kann.

Endogene Zustandsrückführung

Für die exakte Eingangs-/Zustandslinearisierung werden neue Eingänge $\vec{\xi}$ eingeführt:

$$\vec{\xi} = (\xi_1, \dots, \xi_m)^T \quad \text{mit} \quad \xi_i = y_i^{(\kappa_i)}, \quad i = 1, \dots, m, \quad \sum_{i=1}^m \kappa_i \geq n \quad (2.83)$$

Ersetzt man in (2.80) die höheren Ableitungen $y_i^{(\kappa_i)}, \dots, y_i^{(\beta_i)}$ durch die Zeitableitungen von (2.83),

$$\xi_i^{(k_i)} = y_i^{(\kappa_i + k_i)}, \quad k_i = 0, \dots, \beta_i - \kappa_i, \quad (2.84)$$

erhält man

$$\vec{x} = \vec{\alpha} \left(y_1, \dots, y_1^{(\kappa_1 - 1)}, \xi_1, \dots, \xi_1^{(\beta_1 - \kappa_1 - 1)}, \dots, \right. \\ \left. y_m, \dots, y_m^{(\kappa_m - 1)}, \xi_m, \dots, \xi_m^{(\beta_m - \kappa_m - 1)} \right) \quad (2.85a)$$

$$\vec{u} = \vec{\psi} \left(y_1, \dots, y_1^{(\kappa_1 - 1)}, \xi_1, \dots, \xi_1^{(\beta_1 - \kappa_1)}, \dots, \right. \\ \left. y_m, \dots, y_m^{(\kappa_m - 1)}, \xi_m, \dots, \xi_m^{(\beta_m - \kappa_m)} \right) \quad (2.85b)$$

(2.85a) kann als verallgemeinerte Zustandstransformation interpretiert werden [65], die den Zustand \vec{x} in einen neuen Zustand \vec{z} transformiert:

$$\vec{z} = (z_{1,1}, \dots, z_{1,\kappa_1}, \dots, z_{m,1}, \dots, z_{m,\kappa_m})^T \quad \text{mit} \quad (2.86) \\ z_{i,k_i} = y_i^{(k_i - 1)}, \quad i = 1, \dots, m, \quad k_i = 1, \dots, \kappa_i$$

Diese Zustandstransformation lässt sich unter Nutzung von (2.79), (2.77) darstellen als

$$\vec{z} = \tilde{\alpha} \left(\vec{x}, \vec{u}, \dot{\vec{u}}, \dots, \vec{u}^{(\kappa - 1)} \right) \quad (2.87)$$

Da der neue Zustand \vec{z} von den Zeitableitungen des Eingangs \vec{u} abhängen kann, wird er auch *verallgemeinerter Zustand* genannt [65]. Die Dimension der Zustände \vec{z} und \vec{x} muss nicht gleich sein, durch die Zustandstransformation können weitere Zustände eingeführt werden: $\dim \vec{z} \geq \dim \vec{x}$.

Mit der Zustandstransformation (2.86) stellt (2.85b) eine sogenannte *endogene Zustandsrückführung* der Form

$$\vec{u} = \vec{\psi} \left(\vec{z}, \vec{\xi}, \dot{\vec{\xi}}, \dots, \vec{\xi}^{(\beta - \kappa)} \right) \quad (2.88)$$

dar. Die Bezeichnung *endogene Zustandsrückführung* wird verwendet, da nur solche Zustände $z_{i,k}$ eingeführt werden, die ohne Integration, d. h. allein durch Differentiation und algebraische Umformungen aus den Systemgrößen (Zustand \vec{x} , Eingang \vec{u}), also aus den *im Inneren erzeugten*, den *endogenen* Größen, gebildet werden können [65].

Die Zusammenhänge (2.87), (2.88) lassen sich nach \vec{x} bzw. $\vec{\xi}$ auflösen [65]:

$$\vec{x} = \vec{\alpha} \left(\vec{z}, \vec{\xi}, \dot{\vec{\xi}}, \dots, \vec{\xi}^{(\gamma)} \right) \quad (2.89)$$

$$\vec{\xi} = \vec{\psi} \left(\vec{x}, \vec{u}, \dot{\vec{u}}, \dots, \vec{u}^{(\delta)} \right) \quad (2.90)$$

Quasi-statische Zustandsrückführung

Als Spezialfall der endogenen Zustandsrückführung soll nun die *quasi-statische Zustandsrückführung* dargestellt werden. Sie hat die Form [65]:

$$\vec{u} = \vec{\psi} \left(\vec{x}, \vec{\xi}, \dot{\vec{\xi}}, \dots, \vec{\xi}^{(s)} \right) \quad (2.91)$$

Für die neuen Eingänge $\vec{\xi}$ muss dabei

$$\vec{\xi} = \vec{\psi} \left(\vec{z}, \vec{u}, \dot{\vec{u}}, \dots, \vec{u}^{(s)} \right) \quad (2.92)$$

gelten. Die Zustandstransformation hat die Form:

$$\vec{z} = \vec{\alpha}(\vec{x}), \quad \vec{z} \in \mathbb{R}^n \quad (2.93)$$

Im Gegensatz zur endogenen Zustandsrückführung sind bei der quasi-statischen Zustandsrückführung die Dimensionen der Zustände \vec{z} und \vec{x} gleich. Daher kommt auch der Begriff *quasi-statische Zustandsrückführung*: es werden keine weiteren Zustände, also keine zusätzliche Dynamik, eingeführt. Es gilt $\dim \vec{z} = \dim \vec{x}$ und damit für die Summe der κ_i in (2.83):

$$\sum_{i=1}^m \kappa_i = n \quad (2.94)$$

Statische Zustandsrückführung

Die *statische Zustandsrückführung* ist wiederum ein Sonderfall der quasi-statischen Zustandsrückführung. Mit der statischen Zustandsrückführung lässt sich das flache System (2.77) exakt linearisieren, falls die Summe der β_i in (2.80b) gleich der Dimension von \vec{x} ist [57]:

$$\sum_{i=1}^m \beta_i = n \quad (2.95)$$

Die statische Zustandsrückführung und die zugehörige Zustandstransformation haben dann die Form [65]:

$$\vec{u} = \vec{\psi} \left(\vec{x}, \vec{\xi} \right), \quad \vec{\xi} \in \mathbb{R}^m, \quad \text{Rang} \frac{\partial \vec{\psi}(\vec{x}, \vec{\xi})}{\partial \vec{\xi}} = m \quad (2.96)$$

$$\vec{z} = \vec{\alpha}(\vec{x}), \quad \vec{z} \in \mathbb{R}^n \quad (2.97)$$

Im Unterschied zur quasi-statischen Zustandsrückführung ist die statische Zustandsrückführung also nicht von den Ableitungen der Eingänge $\vec{\xi}$ abhängig. Alle zustandslinearisierbaren Eingrößensysteme ($m = 1$) lassen sich durch eine statische Zustandsrückführung exakt linearisieren [65].

Lineares System

Durch die endogene und die quasi-statische Zustandsrückführung (2.91), (2.88) wird das nichtlineare System (2.77) in ein lineares System mit dem Zustand \vec{z} und Eingang $\vec{\xi}$ in Brunovský-Normalform transformiert:

$$\dot{z}_{i,k} = z_{i,k+1}, \quad k = 1, \dots, \kappa_i - 1, \quad i = 1, \dots, m \quad (2.98a)$$

$$\dot{z}_{i,\kappa_i} = \xi_i, \quad i = 1, \dots, m \quad (2.98b)$$

Dieses lineare System verfügt über die exakt gleiche Dynamik wie das nichtlineare System [65], siehe Abbildung 2.4. Die endogene und die quasi-statische Zustandsrückführung sind daher *exakt linearisierende Zustandsrückführungen*. Die Brunovský-Normalform besteht aus m Integrator-Ketten der Länge κ_i , wie im unteren Teil (b) von Abbildung 2.4 dargestellt. Die Integrator-Ketten sind nicht miteinander gekoppelt, die exakt linearisierenden Zustandsrückführungen führen zu einer vollständigen Entkopplung des Systems bezüglich seines Ausgangs. Wenn das nichtlineare System (2.77) durch die statische Zustandsrückführung (2.96) zustandslinearisierbar ist, führt auch dies zu einem System in Brunovský-Normalform.

Die Tatsache, dass für jedes flache System ein lineares System mit gleicher Dynamik existiert, veranschaulicht die Bezeichnung „*flaches System*“: Ein flaches System kann im transformierten linearen Zustand \vec{z} in linearen Räumen dargestellt werden. Die Koordinaten \vec{z} beschreiben dabei sämtliche Systemgrößen – die Zustände und die Eingänge des Systems, inklusive deren zeitliche Ableitungen – vollständig. Die neuen Koordinaten \vec{z} stehen mit den Originalkoordinaten in einem (formal eindeutig umkehrbaren) nichtlinearen Zusammenhang [65].

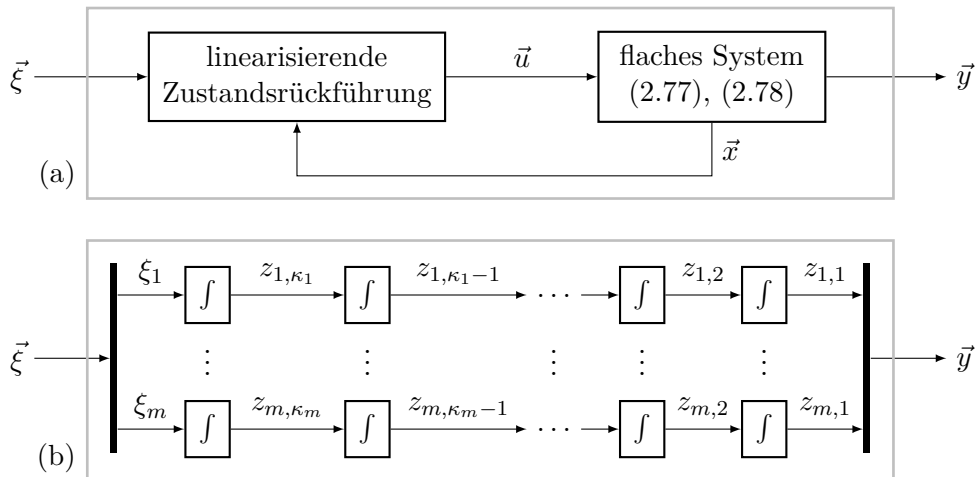


Abbildung 2.4: (a) Flaches System mit exakt linearisierender Zustandsrückführung und (b) lineares System in Brunovský-Normalform mit gleicher Dynamik

2.4.3 Trajektorienfolgeregelung

Aufgrund der exakten Linearisierbarkeit durch endogene und quasi-statische Zustandsrückführungen eignen sich flache Systeme besonders gut für die Trajektorienfolgeregelung. Für die resultierende lineare Systemdynamik (Abbildung 2.4) lässt sich ein Regler mit den üblichen Entwurfsverfahren für lineare Systeme auslegen, der das System stabilisiert sowie kleinere Störungen und Modellierungsfehler ausgleicht.

Eine mögliche Regelung hat folgende Struktur [65, 1, 57]:

$$\xi_i = y_{i,s}^{(\kappa_i)} - \sum_{j=0}^{\kappa_i-1} p_{i,j} \left(y_i^{(j)} - y_{i,s}^{(j)} \right) \quad , \quad i = 1, \dots, m \quad (2.99)$$

Sie erlaubt die Vorgabe der linearen Dynamik des Trajektorienfolgefehlers $\vec{e} = \vec{y} - \vec{y}_s$. Durch geeignete Wahl der Koeffizienten $p_{i,j}$ können die Eigenwerte bzw. die Nullstellen des charakteristischen Polynoms so eingestellt werden, dass die Fehlerdynamik

$$e_i^{(\kappa_i)} + \sum_{j=0}^{\kappa_i-1} p_{i,j} e_i^{(j)} = 0 \quad , \quad i = 1, \dots, m \quad (2.100)$$

asymptotisch stabil ist.

In (2.99) ist zu erkennen, dass der neue Eingang $\vec{\xi}$ primär durch den Sollwert der höchsten Ableitung des flachen Ausgangs $\vec{y}_s^{(\kappa_i)}$ vorgesteuert wird. Für die Regelung werden ebenfalls Solltrajektorien für die niedrigeren Ableitungen $y_{i,s}^{(j)}$ mit $j = 0, \dots, \kappa_i - 1$ benötigt. Wenn keine Störungen und Modellfehler vorhanden sind, wird die Summation in (2.99) zu null.

2.5 Satellitennavigation

GPS wird heutzutage oftmals als Synonym für Navigation verwendet. GPS als Kurzform von NAVSTAR GPS ist das Akronym für „Navigational Satellite Timing and Ranging – Global Positioning System“ und bezeichnet das US-amerikanische Satellitennavigationssystem. Das russische Pendant GLONASS⁵ ist weniger bekannt, wurde jedoch parallel zu GPS entwickelt und ist wie GPS seit den 1990er-Jahren betriebsbereit. Als Oberbegriff für globale Satellitennavigationssysteme hat sich GNSS als Akronym für „Global Navigation Satellite System“ im Englischen etabliert.

Sowohl GPS als auch GLONASS sind primär militärische Systeme. Mit Abschalten der künstlichen Signalverschlechterung (selective availability) von GPS im Jahr 2000 verbesserte sich die für die zivile Nutzung erreichbare Genauigkeit deutlich. Während Satellitennavigation zu Beginn vor allem in Luft- und Schifffahrt eingesetzt wurde, ist sie heutzutage aus vielen Bereichen des Geschäfts- und Privatlebens nicht mehr wegzudenken. Neben der Erweiterung um den Navigationsmassenmarkt im Automobil- und Mobiltelefonsektor hat die Nutzung von GNSS längst auch Geschäftsfelder wie Logistik, Landwirtschaft und Tourismus erreicht. Die genaue Positions-, Geschwindigkeits- und/oder Zeitbestimmung per GNSS ist oftmals ein entscheidender Faktor, der bestimmte Anwendungen erst realisierbar macht.

Aufgrund der großen Bedeutung, die Satellitennavigation mittlerweile für Gesellschaft, Wirtschaft und Verkehr sowie für öffentliche Aufgaben wie Rettungsdienste, Polizei und Verteidigung hat, bauen die Europäische Union und die Volksrepublik China mit „Galileo“ und „BeiDou“ jeweils eigene GNSS auf. Dadurch wird die Abhängigkeit von den USA und der Russischen Föderation verringert, die jederzeit die öffentlich nutzbaren Signale ihrer Systeme verschlechtern oder abschalten könnten. Mit Galileo und BeiDou gibt es vier globale Satellitennavigationssysteme. Sie sind interoperabel und können von GNSS-Empfängern gleichzeitig genutzt werden. Die genutzten Frequenzbänder der zivilen Signale sind in Abbildung 2.5 dargestellt.

Der GSA GNSS Market Report [25] prognostiziert für die Zahl aller jährlich verkauften GNSS-Empfänger einen kontinuierlichen Zuwachs von 1,8 Mrd. Geräten in 2019 auf 2,8 Mrd. in 2029. Unbemannte Luftfahrzeuge haben sich zu einem signifikanten GNSS-Marktsegment hinter Consumer-Geräten (mit Smartphones, Wearables) und Automotive-Applikationen entwickelt. Durch eine Verdreifachung der Zahl der verkauften Einheiten in den letzten drei Jahren hat das Segment der unbemannten Luftfahrzeuge klassische Segmente wie Schifffahrt, Luftfahrt und Landwirtschaft überholt und es wird im nächsten Jahrzehnt weiter wachsen.

Je nach Anwendung ergeben sich unterschiedliche Anforderungen an die Satellitennavigation [25]. Während es für manche professionellen Anwendungen vor allem auf Genauigkeit ankommt, sind für Massenmarktanwendungen eher Verfügbarkeit, Kontinuität und die Zeit bis zur ersten Positionsbestimmung wichtig. Für sicherheitskritische Anwendungen stehen wiederum Integrität und die Robustheit gegen Angriffe wie Spoofing und

⁵GLONASS ist ein Akronym für „Globalnaja nawigazionnaja sputnikowaja sistema“

Jamming im Vordergrund. Die Genauigkeit von Satellitennavigation kann z. B. durch die Nutzung mehrerer Frequenzen und durch differentielle Korrekturverfahren, für die satellitenbasierte oder bodengebundene Augmentierungssysteme genutzt werden, erhöht werden. Dadurch, dass immer mehr Empfänger mehrere GNSS unterstützen, verbessern sich für diese Empfänger unter schwierigen Empfangsbedingungen wie in der Stadt oder im Gebirge die Verfügbarkeit und Kontinuität der Positions- und Zeitbestimmung und die erreichbare Genauigkeit. Integritätsinformationen für sicherheitskritische Anwendungen können etwa durch das GNSS selbst oder durch Augmentierungssysteme zur Verfügung gestellt oder z. B. durch Plausibilitätschecks im Vergleich mit anderen Sensoren gewonnen werden.

Das japanische QZSS und das indische IRNSS/NavIC sind regionale Satellitennavigationssysteme, die in einem beschränkten Gebiet eine Positionsbestimmung ermöglichen und interoperabel zu den GNSS sind.

Satellitenbasierte Augmentierungssysteme (SBAS) wie das amerikanische WAAS, das europäische EGNOS, das japanische MSAS, das indische GAGAN und das russische SDCM senden mit geostationären Satelliten Korrekturdaten und Integritätsinformationen für die Satellitennavigationssysteme aus. Die Korrekturen (siehe Abschnitt 2.5.3) sind nur jeweils für den Kontinent bzw. die Region gültig. Die SBAS nutzen dabei die Frequenzen der Satellitennavigationssysteme, sodass die GNSS-Empfänger keinen separaten Empfangskanal benötigen.

Bodengebundene Augmentierungssysteme (GBAS) senden Korrekturdaten und Integritätsinformationen für die Satellitennavigationssysteme über eine Funk-, Internet- oder Telefonverbindung von Stationen auf der Erdoberfläche aus. Aufgrund ihres meist deutlich kleineren Gültigkeitsgebietes erlauben sie – verglichen mit SBAS – eine noch genauere Korrektur der GNSS-Positionsbestimmung. GBAS werden z. B. für Instrumentenlandesysteme für Verkehrsflugzeuge sowie für Fahrzeugsteuerungen im Bauwesen und in der Landwirtschaft eingesetzt. Mit den per GBAS ermöglichten RTK-Korrekturverfahren (siehe Seite 62), die u. a. in der Landesvermessung eingesetzt werden, können Positionsgenauigkeiten im Zentimeterbereich erreicht werden. Auch in der vorliegenden Anwendung werden RTK-Korrekturdaten genutzt, um eine hoch genaue Positionsbestimmung von Quadroptern und Zielobjekt zu ermöglichen, sodass der Quadropter das Objekt wie gewünscht verfolgen und die Kamera präzise darauf ausrichten kann.

2.5.1 Globale Satellitennavigationssysteme

GPS Die GPS-Satellitenkonstellation umfasst nominell 24 Satelliten auf sechs Erdumlaufbahnen [12], vgl. Abschnitt 2.5.2.1. Diese gehören mit einem Radius von 26 600 km zu den „mittleren Erdumlaufbahnen“⁶. Ein GPS-Satellit umkreist die Erde in knapp 12 Stunden. Die Inklination der Bahnebenen, die Neigung gegenüber der Äquatorebene, beträgt 55°. Aktuell sind mehr als 30 GPS-Satelliten in Betrieb [82] (Stand Oktober 2020). Die zusätzlichen Satelliten belegen weitere Slots auf den sechs Umlaufbahnen. Damit sind auf

⁶siehe Abschnitt 2.5.2.1

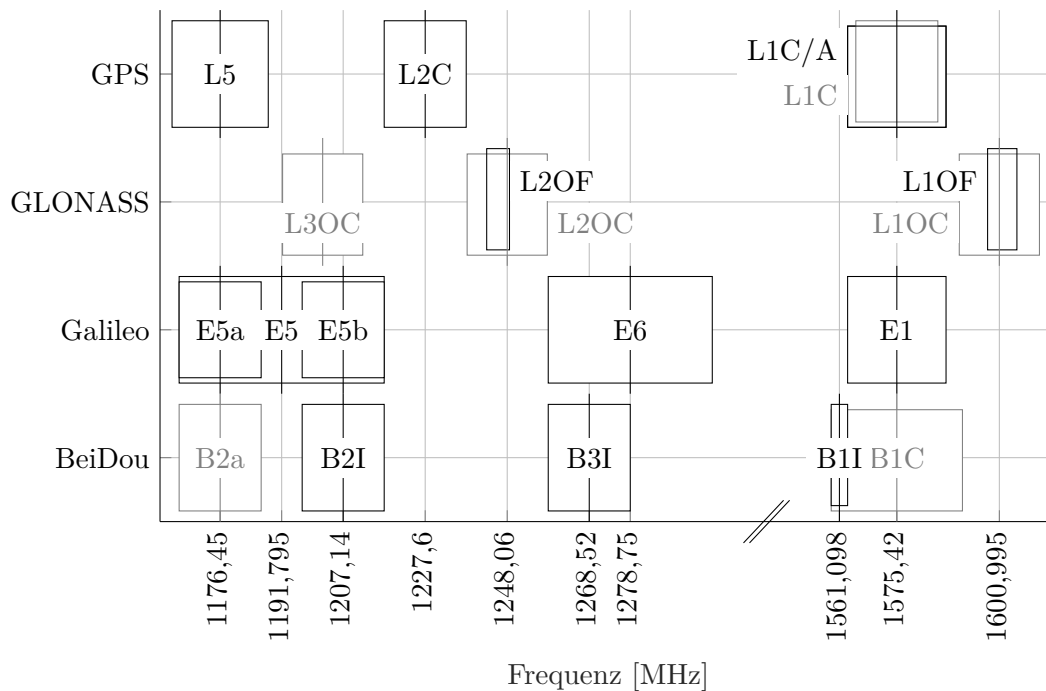


Abbildung 2.5: Zivile GNSS-Frequenzbänder

jedem Punkt der Erde ohne Hindernisse jederzeit ca. 7 bis 13 GPS-Satelliten zu sehen, wobei sie an Nord- und Südpol nur eine Elevation⁷ von knapp 50° erreichen [81].

Ursprünglich waren nur die Navigationsnachrichten des C/A-Signals im L1-Frequenzband (siehe Abschnitt 2.5.2.2) für zivile GPS-Empfänger nutzbar. Die militärischen P(Y)-Signale im L1- und L2-Band sind für zivile Empfänger nicht dekodierbar. Trotzdem kann ein Mehrfrequenzempfänger die Gleichheit der Y-Codesequenzen auf L1 und L2 ausnutzen und so den Signallaufzeitunterschied zwischen beiden Frequenzen berechnen, um den Ionosphärenfehler zu kompensieren (siehe Seite 56).

Im Zuge der Modernisierung von GPS werden mit L2C, L5 und L1C weitere zivil nutzbare Signale eingeführt, siehe Abbildung 2.5. Das L2C-Signal ist mit dem Start der IIR-M Satelliten seit 2005 und das L5-Signal mit den IIF-Satelliten seit 2010 verfügbar. L1C wird mit den IIIA-Satelliten kommen, von denen 2018 und 2019 jeweils einer gestartet wurde. Die Inbetriebnahme der neuen zivilen GPS-Signale ist noch nicht abgeschlossen und die Signale dürfen nur „auf eigenes Risiko“ genutzt werden [12] (Stand April 2020).

GLONASS Die GLONASS-Satellitenkonstellation ist mit 24 aktiven Satelliten auf 3 Erdumlaufbahnen aktuell vollständig [39] (Stand Oktober 2020). In der Vergangenheit war die Zahl der Satelliten nicht immer ausreichend, um eine vollständige Abdeckung

⁷Als Elevation wird der Höhenwinkel eines Himmelskörpers, also der Winkel z. B. eines Satelliten über der lokalen horizontalen Ebene, bezeichnet.

zu gewährleisten. Die Satelliten befinden sich auf mittleren Erdumlaufbahnen mit einem Radius von 25 500 km. Die Inklination der Bahnebenen ist mit $64,8^\circ$ größer als bei GPS, sodass sich die Abdeckung in hohen Breitengraden verbessert. So beträgt die Elevation⁷ der Satelliten an den Polen bis zu 58° . Die zivilen GLONASS-Frequenzbänder sind in Abbildung 2.5 dargestellt. Zum Unterschied der FDMA-Signale L1OF, L2OF und der CDMA-Signale L3OC, L2OC, L1OC siehe Abschnitt 2.5.2.2.

Galileo Das europäische globale Satellitennavigationssystem „Galileo“ ist das einzige GNSS, das unter ziviler Kontrolle steht. Im Dezember 2016 ging Galileo von der Test- in die initiale Betriebsphase über. Seitdem sind die „Galileo Initial Services“ offiziell nutzbar. Für den Vollausbau („Full Operational Capability“, FOC) sind eine Konstellation mit 24 Satelliten und 6 Reservesatelliten sowie weitere Dienste geplant [27]. Die drei Erdumlaufbahnen der Galileo-Satelliten haben einen Radius von 29 600 km und eine Inklination von 56° , sind also mit denen von GPS und GLONASS vergleichbar.

In den Jahren 2011 und 2012 wurden mit den vier In-Orbit-Validation-Satelliten (IOV) die ersten der zurzeit aktiven Galileo-Satelliten auf ihre Erdumlaufbahnen gebracht. Mit dem Start von 22 FOC-Satelliten von August 2014 bis Juli 2018 sind also 26 Galileo-Satelliten in Betrieb, von denen seit der Freigabe der letzten Charge im Februar 2019 insgesamt 22 nutzbar sind [23] (Stand Oktober 2020).

Zu den initialen Diensten, die seit Dezember 2016 zur Verfügung stehen, gehört der „Open Service“ (OS), der „Public Regulated Service“ (PRS) und der „Search and Rescue Service“ (SAR) [24]. Der Open Service, der die Frequenzbänder E1 und E5 nutzt (siehe Abbildung 2.5), kann weltweit kostenlos genutzt werden. Der PRS ist ein verschlüsselter Dienst mit höherer Kontinuität und Robustheit, der nur Institutionen mit hoheitlichen Aufgaben wie Rettungsdiensten und Polizei zur Verfügung steht. Der SAR-Dienst ist in das internationale Such- und Rettungssystem COSPAS-SARSAT integriert. Die Galileo-Satelliten empfangen und orten Notruf-Signale z. B. von Schiffen und Luftfahrzeugen und bestätigen dem Sender seit Anfang 2020 auch den Eingang des Notrufs bei der Leitstelle durch einen Rückkanal (RLS).

Als Erweiterung des Open Service ist geplant, dass dessen Navigationsnachrichten authentifiziert werden können (OS-NMA), sodass ein Empfänger feststellen kann, ob die Nachricht tatsächlich von einem Galileo-Satelliten und nicht von einem Angreifer gesendet wurde [27]. Außerdem sind mit dem „High Accuracy Service“ und dem „Commercial Authentication Service“ zwei weitere Dienste geplant. Der kostenlos nutzbare High Accuracy Service soll als Ergänzung zum Open Service im E6-Frequenzband Korrekturdaten für Precise-Point-Positioning (PPP) senden und so Positionsgenauigkeiten unter 20 cm ermöglichen [24]. Mit dem Commercial Authentication Service wird einem kontrollierten Nutzerkreis durch verschlüsselte Elemente der Navigationsnachrichten im E6-Band eine weitere Authentifikationsmöglichkeit zur Verfügung gestellt, mit der er die Vertrauenswürdigkeit der Navigationsnachrichten überprüfen kann.

BeiDou Das chinesische „BeiDou Navigation Satellite System“ (BeiDou/BDS) deckte in der zweiten Ausbaustufe BDS-2, die seit Dezember 2012 im regulären Betrieb war, mit Satelliten auf geostationären und geneigten geosynchronen Orbits⁸ (GEO, IGSO) die Region Asien-Pazifik ab. Die dritte Ausbaustufe BDS-3 umfasst mit 24 MEO⁸-, 3 GEO- und 3 IGSO-Satelliten insgesamt 30 Satelliten. Durch die höhere Zahl der MEO-Satelliten wird eine weltweite Abdeckung erreicht. Die zivilen BeiDou-Frequenzbänder sind in Abbildung 2.5 dargestellt. Seit Dezember 2018 ist BDS-3 im Basisbetrieb für die Nutzung freigegeben. Im Juni 2020 startete der letzte Satellit, sodass die BDS-3-Konstellation jetzt vollständig ist [59, 79].

2.5.2 Funktionsprinzip Satellitennavigation

Globale Satellitennavigationssysteme erlauben es einem GNSS-Empfänger, seine Position, Geschwindigkeit und Zeit (PVT: position, velocity, time) zu bestimmen. Anhand der von den GNSS-Satelliten ausgesandten Signale bestimmt der Empfänger ihre Laufzeit, um u. a. seine Entfernung zu den Satelliten zu ermitteln. Da sich die elektromagnetischen Wellen mit Lichtgeschwindigkeit ausbreiten, sind die Anforderungen an die Genauigkeit der Signallaufzeitmessung hoch. So führt z. B. ein Zeitfehler von 100 ns bereits zu einem Entfernungsfehler von 30 m. Die GNSS-Satelliten verfügen daher über Atomuhren, die eine hohe Zeitstabilität aufweisen. Da Atomuhren für die meisten GNSS-Empfänger aus Kostengründen keine sinnvolle Lösung sind, verwenden Empfänger meist Quarzoszillatoren. Der Empfängeruhrenfehler wird zusammen mit der Position geschätzt. Wenn ein Empfänger mindestens vier Satelliten empfängt, können bei bekannten Satellitenpositionen die 3-D-Position und das Uhren-Offset des Empfängers bestimmt werden.

2.5.2.1 Umlaufbahnen

Die Satelliten der GNSS bewegen sich aufgrund der Gravitationskraft der Erde (2.5) näherungsweise auf elliptischen Bahnen um die Erde. Diese Ellipsen können durch die große Halbachse a , die Exzentrizität e , drei Parameter für die Lage der Ellipse bezüglich des Schwerpunktes der Erde und eine Bezugszeit, die sog. Epoche, beschrieben werden⁹. Die tatsächlichen Satelliten-Bahnen weichen aufgrund der Unregelmäßigkeit des Gravitationsfelds der Erde, der Massenanziehung des Mondes und weiterer wirkender Kräfte von idealen Ellipsen ab. Jeder GNSS-Satellit sendet in seinen Navigationsnachrichten die Parameter seiner Umlaufbahn inklusive diverser Korrekturparameter [27, 12]. Diese Parametersätze, die sog. Ephemeriden, werden regelmäßig aktualisiert und anhand der Messungen der Referenzstationen am Boden an die tatsächlichen Satellitenbahnen angepasst. Neben den eignen Ephemeriden sendet jeder Satellit außerdem die Bahnparameter aller Satelliten der Konstellation in Form des weniger genauen sog. Almanachs.

⁸siehe Abschnitt 2.5.2.1

⁹Die nominelle Exzentrizität der Bahnellipsen der Satelliten von GPS, GLONASS, Galileo und der MEO-Satelliten von BeiDou ist null. Die nominellen Bahnen sind also kreisförmig mit a als Radius.

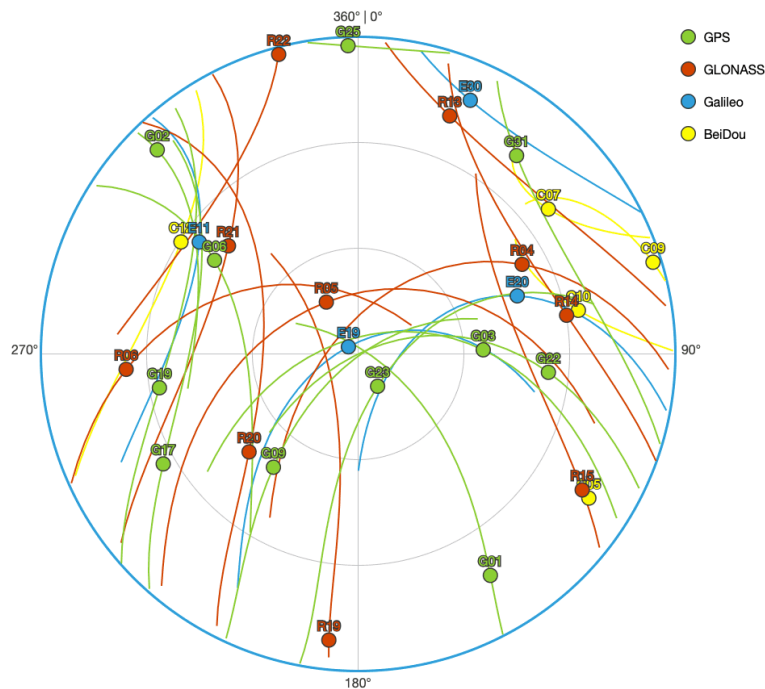


Abbildung 2.6: GNSS-Satellitenkonstellation während des Flugversuchs, der in Abschnitt 6.1.1 dargestellt ist (Quelle: [81])

Für die Satelliten von GPS, GLONASS, Galileo und BeiDou auf mittleren Erdumlaufbahnen (MEO) beträgt die Neigung der Bahnebenen gegenüber der Äquatorebene, die Inklination, zwischen 55° und 58° , siehe Abschnitt 2.5.1. Satelliten auf geosynchronen Orbits umkreisen die Erde in einer Höhe von 35 786 km über der Erdoberfläche innerhalb eines siderischen Tages (23 h, 56 min und 4 s). Dadurch ist ihr Orbit synchron mit der Erddrehung. Ein geostationärer Orbit (GEO) ist ein geosynchroner Orbit mit einer Inklination von 0° . Geostationäre Satelliten z. B. von BeiDou, EGNOS und WAAS bleiben daher scheinbar „stationär“ über einem Punkt des Erdäquators. Ein geneigter geosynchroner Orbit (IGSO), den z. B. die BeiDou-IGSO-Satelliten haben, ist ein geosynchroner Orbit mit einer von null verschiedenen Inklination. Die Bodenspur, die senkrechte Projektion der Bahn auf die Erdoberfläche, hat die Form einer Acht über dem Erdäquator.

Abbildung 2.6 zeigt exemplarisch die GNSS-Satelliten, die während des in Abschnitt 6.1.1 dargestellten Flugversuchs verfügbar waren. Die Grafik visualisiert Elevation⁷ und Azimut¹⁰ der Satelliten für den Zeitpunkt und den Ort des Flugversuchs sowie deren Pfad für 6 Stunden.

¹⁰Als Azimut wird der Horizontalwinkel eines Himmelskörpers gegenüber einer Himmelsrichtung bezeichnet – hier der Horizontalwinkel zwischen der Satellitenposition und Norden, siehe auch (5.121).

2.5.2.2 Signale

Die GNSS-Satelliten senden in verschiedenen Frequenzbändern mit Trägerfrequenzen zwischen 1,1 und 1,6 GHz, siehe Abbildung 2.5. Einfache Empfänger verarbeiten meist nur eine Trägerfrequenz. Ein Mehrfrequenz-Empfänger kann die frequenzabhängigen Laufzeitverzögerungen der Satelliten-Signale in der Ionosphäre (siehe Seite 56) kompensieren und so die Positionslösung verbessern.

Die Signalleistung der GNSS-Signale ist verglichen mit anderen Funkstandards sehr gering [27, 12]. Wenn die Signale den Empfänger erreichen, liegen sie deutlich unter dem Rauschlevel. Die Signalleistung wird oft als Trägerrauschverhältnis (C/N_0 , „carrier-to-noise-ratio“) ausgedrückt.

Um die Satellitensignale im Empfänger über das Rauschlevel heben zu können, werden verschiedene Codemultiplexverfahren (CDMA) eingesetzt. Dazu wird eine bekannte, pseudo-zufällige Codesequenz, die PRN („pseudo random noise“) oder „ranging code“ genannt wird, zusammen mit einer Navigationsnachricht im Satelliten auf die harmonische Schwingung des Trägers moduliert.

Bei GPS, Galileo und BeiDou senden die Satelliten jeweils auf den gleichen Trägerfrequenzen. Da jeder Satellit einen anderen PRN-Code verwendet, wird die Trennung der Signale mehrerer Satelliten im Empfänger erst durch CDMA möglich. Bei GLONASS kommt zur Unterscheidung der Satelliten ein Frequenzmultiplexverfahren (FDMA) zum Einsatz. Die GLONASS-Satelliten verwenden dabei eine gleiche PRN, aber unterschiedliche Frequenzen. Während ältere GLONASS-Satelliten nur FDMA-Signale senden, nutzen seit 2014 gestartete Satelliten neben FDMA auch CDMA in verschiedenen Frequenzbändern [39].

Die PRNs des ersten zivil nutzbaren GPS-L1-Signals, die C/A-Codes („coarse/acquisition“), werden nach der Exklusiv-Oder-Verknüpfung mit der Navigationsnachricht per binärer Phasenumtastung (BPSK) auf die Trägerfrequenz aufmoduliert. Die C/A-Codes haben eine Chiprate¹¹ von 1,023 MChips/s und wiederholen sich alle 1023 Chips, also jede Millisekunde [12]. Die Trägerfrequenzen sind exakte Vielfache dieser Chiprate. So passen z. B. genau 1540 Wellenlängen der L1-Trägerfrequenz 1575,42 MHz in einen Chip. Die Datenrate der Navigationsnachrichten auf dem L1-C/A-Signal liegt mit 50 Bit/s deutlich unter der Chiprate der PRN-Codes. Während eines Bits werden die PRN-Codes also 20-mal wiederholt.

Für die Galileo-Signale und für die neuen zivilen GPS-Signale wird als CDMA das BOC-Verfahren („Binary Offset Carrier“) mit unterschiedlichen Parametern genutzt [27]. Dies führt verglichen mit BPSK zu einer zusätzlichen Bandspreizung, wodurch die Signale robuster gegen Störungen werden. BeiDou und GLONASS nutzen als CDMA sowohl BPSK als auch BOC. Die Navigationsnachrichten werden bei Galileo jeweils nur in den In-Phase-Komponenten E5a-I, E5b-I, E1-B und E6-B der Signale übertragen. Die Quadratur-Komponenten E5a-Q, E5b-Q, E1-C und E6-C enthalten keine Navigationsnachrichten, sondern

¹¹Da die PRN-Codesequenzen keine Information enthalten, werden ihre Binärzahlen als „Chips“ und nicht als „Bits“ bezeichnet. Die Einheit für die Chiprate ist z. B. MChips/s also 10^6 Chips pro Sekunde.

nur die PRN-Codes. Dadurch wird die Akquisition und das Tracking der Signale erleichtert, siehe Abschnitt 2.5.2.3f. Die Galileo-PRNs haben Chipraten von 1,023 MChips/s (für E1) bis 10,23 MChips/s (E5) und Längen von 4092 (E1-B) bis 1 023 000 Chips (E5a-Q), wiederholen sich also alle 4 bis 100 ms [27]. Die Symbolraten der Navigationsnachrichten betragen 50 Symbole/s auf E5a-I, 250 Symbole/s auf E5b-I und E1-B sowie 1000 Symbole/s auf E6-B.

Durch die Relativbewegung zwischen Satelliten und Empfänger treten auf den Satellitensignalen Dopplerverschiebungen (2.113) der Sendefrequenzen in Abhängigkeit von der Relativgeschwindigkeit und von den Frequenzen auf. So kann z. B. bei einem auf der Erdoberfläche ruhenden Empfänger die Frequenzverschiebung der Trägerfrequenzen bei mehreren kHz liegen. Für die Chiprate des C/A-Codes beträgt sie mehrere Hz [84].

2.5.2.3 Akquisition

Durch die Dopplerverschiebung ist die genaue Trägerfrequenz, mit der das Satellitensignal den Empfänger erreicht, erst einmal unbekannt. Außerdem kennt der Empfänger die zeitliche Verschiebung, die Phase, des PRN-Codes nicht. Daher muss ein Empfänger einen Satelliten erst „suchen“, bevor er ihn nutzen kann. Diese Suche wird Akquisition genannt. Wenn der Empfänger über den aktuellen Almanach verfügt und die Zeit zumindest grob kennt, sind die Satelliten bekannt, die zu dem Zeitpunkt empfangbar sein müssten. Der Empfänger hat den aktuellen Almanach ggf. noch gespeichert, er empfängt ihn mit der Navigationsnachricht des ersten Satelliten, den er akquirieren konnte, oder er lädt den Almanach aus einer anderen Quelle, z. B. dem Internet. Je genauer die Uhrzeit bekannt ist, desto gezielter kann der Empfänger nach erwarteten Satelliten suchen. Der Empfänger generiert mit seinen Codegeneratoren die PRN-Codesequenz des zu suchenden Satelliten und korreliert sie mit dem empfangenen Signal, das mit einer vermuteten Trägerfrequenz demoduliert wurde [84]. Nur wenn die angenommene Verschiebung der Codesequenz und die Frequenzverschiebung in etwa stimmen, liegt das Korrelationsergebnis über dem Rauschlevel. Sonst wird die generierte Codesequenz leicht verschoben bzw. die angenommene Frequenzverschiebung angepasst. Die PRN-Codes sind dazu so entworfen, dass ihre Autokorrelationsfunktion (A.19) für eine Verschiebung um null Chips einen starken Peak aufweist und sonst nahezu null ist. Die PRN-Codes zweier Satelliten sind orthogonal, d. h. ihre Kreuzkorrelationsfunktion (A.18) ist für alle Verschiebungen nahe null.

2.5.2.4 Tracking

Nach der Akquisition eines Satelliten beginnt das Tracking seiner Signale. Die Empfänger-Korrelatoren führen dabei in Regelschleifen die generierten Signale den Satellitensignalen nach. Der Empfänger kennt dadurch die zeitliche Verschiebung der PRN-Codesequenz (Codephase), die Trägerfrequenzverschiebung und Trägerphase eines getrackten Satellitensignals. Die Codephase kann typischerweise auf unter 1% der PRN-Chiplänge genau getrackt werden [28]. 1% PRN-Chiplänge entsprechen z. B. beim C/A-Code ca. 10 ns bzw.

3 m. Durch höhere Chipraten z. B. beim Galileo E5-Signal (vgl. Abschnitt 2.5.2.2) werden diese Zahlen noch deutlich kleiner.

Wenn ein Satellit zu lange z. B. durch Signalabschattungen nicht empfangen werden kann, muss er reaktiviert werden. Die Zahl der Kanäle eines Empfängers bestimmt die maximale Zahl der Satelliten, die er gleichzeitig akquirieren bzw. tracken kann. Dabei benötigt der Empfänger pro Satelliten und Trägerfrequenz einen Kanal.

Die Navigationsnachrichten werden mit der Satellitenuhrzeit zeitgestempelt. Der Zeitstempel in einer Navigationsnachricht bezieht sich dabei auf den Anfang ihres ersten Bits. Dieser Anfang und der des ersten Chips einer PRN sind zeitlich synchron. Durch Mitzählen der vollständigen PRN-Sequenzen seit Beginn der Navigationsnachricht und der Chips der letzten begonnenen Sequenz sowie durch Ablesen des aktuellen Chip-Bruchteils vom PRN-Korrelator kann der Empfänger mit dem Zeitstempel in der Nachricht die Satellitenuhrzeit τ_i zum Sendezeitpunkt des aktuell empfangenen Signals bestimmen („time of arrival“).

Die Satellitenuhrzeit τ_i unterscheidet sich von der GNSS-Systemzeit t :

$$\tau_i = t + \Delta t_i + \delta t_i \quad (2.101)$$

Das Offset Δt_i kann ein Empfänger aus Parametern in den Navigationsnachrichten berechnen. Diese Parameter werden regelmäßig von den Bodenstationen angepasst. δt_i ist der verbleibende Restfehler.

2.5.2.5 Pseudorangemessung

Aus der aktuellen Empfängeruhrzeit τ'_e („time of measurement“), der Satellitenuhrzeit τ_i zum Sendezeitpunkt, die anhand der PRN-Codesequenzen bestimmt wurde („time of arrival“, s. o.), und dem Satellitenuhrenoffset Δt_i berechnet der Empfänger den Messwert \tilde{t}'_{li} der Signallaufzeit vom i -ten Satelliten zum Empfänger:

$$\tilde{t}'_{li} = \tau'_e - (\tau_i - \Delta t_i) \quad (2.102)$$

Wie die Satellitenuhren hat auch die Empfängeruhrzeit τ'_e ein Offset Δt_e gegenüber der Systemzeit t :

$$\tau'_e = t + \Delta t_e \quad (2.103)$$

Das Empfängeruhrenoffset Δt_e ist zum Messzeitpunkt nicht genau bekannt und wird zusammen mit der Position bestimmt, siehe Abschnitt 2.5.2.8.

Durch das Empfängeruhrenoffset handelt es sich bei \tilde{t}'_{li} (2.102) nicht um einen Messwert der tatsächlichen Signallaufzeit t_{li} , sondern einer „Pseudo“-Signallaufzeit t'_{li} .

Der Empfänger berechnet aus der Satellitenuhrzeit τ_i zum Sendezeitpunkt und dem aus der Navigationsnachricht ermittelten Satellitenuhrenoffset Δt_i die Systemzeit t zum Sendezeitpunkt als $t \approx \tau_i - \Delta t_i$, vgl. (2.101). Dies erlaubt dann, mit den Ephemeriden (siehe

Abschnitt 2.5.2.1) die geschätzte Position $\hat{\vec{p}}_i$ des Satelliten zum Sendezeitpunkt zu bestimmen.

Die geometrische Entfernung zwischen der Position \vec{p}_A der Empfängerantenne und der Position \vec{p}_i der Sendeantenne des i -ten Satelliten beträgt $|\vec{p}_A - \vec{p}_i|$. Als scheinbare Entfernung bzw. „Pseudorange“ wird die Summe aus dieser geometrischen Entfernung und dem Abstandsfehler bezeichnet, der durch das Empfängeruhrenoffset Δt_e verursacht wird [28]:

$$\rho_i(\vec{x}, \vec{p}_i) = |\vec{p}_A - \vec{p}_i| + c \Delta t_e \quad (2.104)$$

Dabei ist c die Lichtgeschwindigkeit, also die Ausbreitungsgeschwindigkeit der Satellitensignale im Vakuum¹². Die Positionen von Empfängerantenne und Satelliten werden im ECEF-Koordinatensystem angegeben (siehe Abschnitt 2.1.1). Jede Pseudorange ρ_i zum i -ten Satelliten ist also eine Funktion der Satellitenposition \vec{p}_i , für die der Schätzwert $\hat{\vec{p}}_i$ vorliegt, und einer Unbekannten \vec{x} :

$$\vec{x} = \left(\vec{p}_A^{e,T}, c\Delta t_e \right)^T \quad (2.105)$$

\vec{x} hat vier Komponenten: drei Koordinaten $(p_{A,x}, p_{A,y}, p_{A,z})^T$ der Empfänger-Position \vec{p}_A^e und das Empfängeruhrenoffset Δt_e .

Aus der Pseudo-Signallaufzeit t'_i (2.102) wird die Beobachtungsgröße $\tilde{\rho}_i$ der Pseudorange ρ_i berechnet:

$$\tilde{\rho}_i = c \tilde{t}'_i \quad (2.106)$$

Das Messmodell für die Pseudorange für die Trägerfrequenz f_j lautet [28]:

$$\begin{aligned} \tilde{\rho}_i &= \rho_i(\vec{x}, \vec{p}_i) + c \delta t_i + \frac{f_{j+1}}{f_j} I_i + T_i + M_{\rho_j, i} + \nu_{\rho_j, i} \\ &= \rho_i(\vec{x}, \hat{\vec{p}}_i) + \chi_i \quad \text{mit} \quad \chi_i = E_i + c \delta t_i + \frac{f_{j+1}}{f_j} I_i + T_i + M_{\rho_j, i} + \nu_{\rho_j, i} \end{aligned} \quad (2.107)$$

Der Pseudorangefehler χ_i , der auch „User Equivalent Range Error“ (UERE) genannt wird, setzt sich aus verschiedenen Komponenten zusammen:

Der Ephemeriden-Fehler $E_i = \rho_i(\vec{x}, \vec{p}_i) - \rho_i(\vec{x}, \hat{\vec{p}}_i)$ wird durch die Abweichung der unbekanntesten tatsächlichen Satellitenposition \vec{p}_i von der angenommenen $\hat{\vec{p}}_i$ verursacht. δt_i ist der verbleibende Satellitenuhrenfehler aus (2.101) nach der Korrektur von τ_i mit Δt_i . Sowohl E_i als auch δt_i wirken sich auf alle GNSS-Empfänger aus, die den i -ten Satelliten verwenden.

$f_{j+1}/f_j I_i$ beschreibt den Fehler, der durch den Einfluss der Ionosphären-Schicht der Atmosphäre auf die Satelliten-Signale auf dem Weg zum Empfänger entsteht. In der Ionosphäre in Höhen zwischen 50 und 1000 km über der Erdoberfläche wirken sich freie Elektronen und positiv geladene Moleküle auf die Phasen- und Gruppengeschwindigkeiten sowie auf

¹²In der Atmosphäre weichen die tatsächlichen Gruppen- und Phasengeschwindigkeiten der Signale von der Lichtgeschwindigkeit ab.

den Pfad der Satellitensignale aus [28]. Diese Effekte sind frequenzabhängig. Daher kann ein Mehrfrequenzempfänger den Fehlereinfluss dieser Effekte auf die Pseudorangemessung auf einer Trägerfrequenz f_j durch eine Pseudorangemessung auf einer zweiten Frequenz f_{j+1} eliminieren. Da der Fehler $f_{j+1}/f_j I_i$ durch die Ionosphäre i. Allg. den größten Anteil am Pseudorangefehler χ_i hat, können Mehrfrequenzempfänger deutlich bessere Genauigkeiten erreichen als Einfrequenzempfänger. Einfrequenzempfänger können zum Beispiel mithilfe des Klobuchar-Modells, dessen Parameter in den Navigationsnachrichten enthalten sind, etwa 50% des Ionosphärenfehlers kompensieren.

Der Fehler T_i wird durch Effekte in der Troposphären-Schicht der Atmosphäre unterhalb der Ionosphäre verursacht. Temperatur-, Druck- und Luftfeuchtigkeitsunterschiede durch das Wetter führen zu Luftdichtevariationen, die sich auf die Signallaufzeit auswirken. Diese Effekte sind für die GNSS-Signale frequenzunabhängig und können zum Teil durch Modelle kompensiert werden [28]. Bei Satelliten mit niedriger Elevation⁷ wirkt sich die Troposphäre stärker auf den Pseudorangefehler aus, da bei ihnen die Signallaufzeit durch die Troposphäre deutlich länger ist als bei Satelliten im Zenit.

Durch Signalreflexionen in der Nähe der Empfängerantenne, so genannte Mehrwege-Effekte („multi path“), kann ein weiterer Fehler $M_{\rho_j,i}$ verursacht werden. Diese Effekte können ggf. durch ein verändertes Antennendesign oder den Ausschluss von Satelliten mit niedriger Elevation⁷ verringert werden. Der verbleibende Fehler $\nu_{\rho_j,i}$ ist schließlich das Messrauschen.

Während $M_{\rho_j,i}$ und $\nu_{\rho_j,i}$ i. Allg. empfängerspezifisch sind, sind I_i und T_i für jeden anderen nahen Empfänger normalerweise ähnlich. Zusammen mit E_i und δt_i werden I_i und T_i daher „Common-Mode-Fehler“ genannt [28, 84].

2.5.2.6 Trägerphasenmessung

Neben der Pseudorangemessung auf Basis der PRN-Codesequenzen kann ein GNSS-Empfänger auch die Träger des Satellitensignals auswerten. Dazu werden diese mit den im Empfänger generierten Referenzträgern verglichen. Zum Zeitpunkt τ'_e hat der Empfänger für die Trägerfrequenz f_j mit der Wellenlänge $\lambda_j = c/f_j$ eine Referenzträgerphase $\tilde{\phi}_{i,j}$ mit N_t ganzen Wellenlängen und dem Bruchteil ϕ_t einer Wellenlänge erzeugt:

$$\tilde{\phi}_{i,j}(\tau'_e) = N_{t,j}(\tau'_e) + \phi_{t,j}(\tau'_e) \quad (2.108)$$

Das Messmodell der Trägerphasen-Beobachtungsgröße $\phi_{j,i}$ für den i -ten Satelliten und die Trägerfrequenz f_j kann formuliert werden als [28]:

$$\phi_{j,i} \lambda_j = \rho_i(\vec{x}, \hat{p}_i) + E_i + c \delta t_i - \frac{f_{j+1}}{f_j} I_i + T_i + M_{\phi_j,i} + \nu_{\phi_j,i} + N_{j,i} \lambda_j \quad (2.109)$$

$\rho_i(\vec{x}, \hat{p}_i)$ und die Common-Mode-Fehlerterme E_i , δt_i , I_i , T_i sind die gleichen wie in (2.107). Der empfängerspezifische Fehler – der Mehrwegefehler $M_{\phi_j,i}$ und das Messrauschen $\nu_{\phi_j,i}$ – unterscheiden sich von $M_{\rho_j,i}$ und $\nu_{\rho_j,i}$ der PRN-Code-basierten Pseudorangemessung

(2.107). $N_{j,i}$ ist die Differenz der vom Empfänger und der vom Satelliten generierten Träger-Wellenlängen. Die Sinusschwingungen des Trägers wiederholen sich im Abstand der Trägerwellenlänge λ_j (z. B. 19 cm für L1-Frequenz) und lassen sich nicht voneinander unterscheiden. Daher wird $N_{j,i}$ Trägerphasenmehrdeutigkeit („carrier phase ambiguity“) genannt. $N_{j,i}$ ist eine i. Allg. unbekannte, ganzzahlige Konstante.

Gelingt es, z. B. mit dem RTK-Verfahren (siehe Seite 62), die Common-Mode-Fehler E_i , δt_i , I_i , T_i zu eliminieren, sodass die Trägerphasenmehrdeutigkeit $N_{j,i}$ bestimmt werden kann, kann die Trägerphasenmessung wie die PRN-Code-basierte Pseudorangemessung (2.107) zur Bestimmung der Pseudorange verwendet werden:

$$\tilde{\rho}_{i,t} = \lambda_j(\phi_{j,i} - N_{j,i}) = \rho_i(\vec{x}, \hat{\vec{p}}_i) + M_{\phi_{j,i}} + \nu_{\phi_{j,i}} \quad (2.110)$$

Typischerweise sind die Fehler $M_{\phi_{j,i}} + \nu_{\phi_{j,i}}$ kleiner als 1% der Fehler $M_{\rho_{j,i}} + \nu_{\rho_{j,i}}$ der PRN-Code-basierten Pseudorangemessung (2.107). Daher können mit den Trägerphasen-basierten Pseudoranges Positionsgenauigkeiten im Zentimeter-Bereich erreicht werden [28].

Wenn die Trägerphasenmehrdeutigkeiten $N_{j,i}$ nicht exakt aufgelöst werden können, können die Trägerphasenmessungen trotzdem genutzt werden, um die PRN-Code-basierten Pseudorangemessungen (2.107) zu glätten („Träger-Glättung“, „carrier aided smoothing“) [28, 84]. Dabei werden die $N_{j,i}$ aus der Differenz von (2.109) und (2.107) geschätzt. Da die $N_{j,i}$ konstant sind, können durch Bildung des Mittelwertes aller Messungen die Schätzwerte für die $N_{j,i}$ geglättet werden und so (2.110) als Pseudorangemessung zur Positionsbestimmung genutzt werden, siehe Abschnitt 2.5.2.8.

2.5.2.7 Deltarangemessung

Zur Bestimmung der Empfänger-Geschwindigkeit wird die Deltarange-Beobachtungsgröße $\Delta\rho_i(\tau'_e)$ als Änderung der Pseudorange in einem kleinen Zeitintervall T definiert [28]:

$$\Delta\rho_i(\tau'_e) = \rho_i(\tau'_e) - \rho_i(\tau'_e - T) \quad (2.111)$$

Die mittlere Änderungsrate ist dann $\dot{\rho}_i = \Delta\rho_i(\tau'_e)/T$. Die Deltarange-Beobachtungsgröße wird nicht durch die Änderung der Pseudorange-Beobachtungsgröße $\tilde{\rho}_i$ (2.106) bestimmt, da zwei bessere Alternativen zur Verfügung stehen:

Wenn der Empfänger die Trägerphase trackt, wird $\Delta\rho_i(\tau'_e)$ aus der Änderung der Trägerphasen-Beobachtungsgröße $\phi_{j,i}$ (2.109) berechnet [28]:

$$\Delta\rho_i(\tau'_e) = \lambda_j(\phi_{j,i}(\tau'_e) - \phi_{j,i}(\tau'_e - T)) \quad (2.112)$$

Wenn die Trägerphase nicht getrackt werden kann, wird der Doppler-Effekt ausgenutzt: Aufgrund der Relativgeschwindigkeit v_{rel} zwischen Empfänger und Satellit weicht die Fre-

quenz $f_{e,j}$ des vom GNSS-Empfänger empfangenen Signals um die Dopplerverschiebung Δf von der Frequenz f_j des Satellitensignals ab:

$$f_{e,j} = f_j + \Delta f \quad \text{mit} \quad \Delta f = -f_j \frac{v_{\text{rel}}}{c}, \quad v_{\text{rel}} = \frac{d}{dt} |\vec{p}_A - \vec{p}_i| \quad (2.113)$$

Die Deltarange-Beobachtungsgröße $\Delta \rho_i(\tau'_e)$ wird dann als $T \lambda_j \Delta f$ aus der Dopplerverschiebung Δf des im Empfänger generierten Referenzträgers ermittelt.

Wenn die Deltarange-Beobachtungsgrößen für mindestens vier Satelliten zur Verfügung stehen, kann die Empfänger-Geschwindigkeit durch das Lösen eines linearen Gleichungssystems bestimmt werden (für Details siehe [28, 84]). Dabei werden die Satellitengeschwindigkeiten und die Satellitenuhrendrifts aus den Parametern der Navigationsnachrichten berechnet.

2.5.2.8 Positionslösung

Auf Basis der PRN-Code-basierten Pseudorangemessungen (2.107) bzw. der Trägerphasen-basierten Pseudorangemessungen (2.110) und den geschätzten Satellitenpositionen \hat{p}_i kann der GNSS-Empfänger seine 3-D-Position und sein Uhren-Offset bestimmen:

Setzt man (2.104) in die PRN-Code-basierten Pseudorangemessungen (2.107) ein, erhält man für die geschätzte Pseudorange $\hat{\rho}$ in Abhängigkeit von der gesuchten Empfängerantennenposition \hat{p}_A und des gesuchten Empfängeruhrenoffsets $\Delta \hat{t}_e$:

$$\hat{\rho}(\hat{x}, \hat{p}_i) = |\hat{p}_A - \hat{p}_i| + c \Delta \hat{t}_e + \hat{\chi}_i \quad (2.114)$$

$\hat{\chi}_i$ ist dabei der geschätzte Pseudorangefehler mit den Komponenten gemäß (2.107). Unbekannte Komponenten von $\hat{\chi}_i$ werden zu null gesetzt. Bei Nutzung der Trägerphasen-basierten Pseudorangemessungen $\tilde{\rho}_{i,t}$ (2.110) ist das Vorgehen gleich, dann entfällt $\hat{\chi}_i$.

Um die Empfängerantennenposition und das Empfängeruhrenoffset zu bestimmen, wird das \hat{x} gemäß (2.105) gesucht, das die Differenzen zwischen den gemessenen Pseudoranges $\tilde{\rho}_i$ aus (2.106) und den geschätzten Pseudoranges $\hat{\rho}(\hat{x}, \hat{p}_i)$ aus (2.114) minimiert. Als Kleinste-Quadrate- bzw. Least-squares-Problem lässt sich das schreiben als:

$$\min_{\hat{x}} \left| \tilde{\rho} - \hat{\rho}(\hat{x}) \right|^2 \quad \text{mit} \quad \tilde{\rho} = \begin{pmatrix} \tilde{\rho}_1 \\ \tilde{\rho}_2 \\ \vdots \\ \tilde{\rho}_n \end{pmatrix}, \quad \hat{\rho}(\hat{x}) = \begin{pmatrix} \hat{\rho}(\hat{x}, \hat{p}_1) \\ \hat{\rho}(\hat{x}, \hat{p}_2) \\ \vdots \\ \hat{\rho}(\hat{x}, \hat{p}_n) \end{pmatrix}, \quad n \geq 4 \quad (2.115)$$

Um die vier Komponenten von $\hat{\vec{x}}$ bestimmen zu können, müssen mindestens vier Pseudorangemesswerte $\tilde{\rho}_i$ vorliegen¹³. Die Lösung für $\hat{\vec{x}}$ wird iterativ mit dem Gauß-Newton-Verfahren ermittelt [28, 84]. Dazu wird $\hat{\rho}(\hat{\vec{x}})$ bezüglich $\hat{\vec{x}}$ am Punkt $\hat{\vec{x}}_k$ linearisiert:

$$\hat{\rho}(\hat{\vec{x}}) \approx \tilde{\rho}(\hat{\vec{x}}_k) + \mathbf{H}_k(\hat{\vec{x}} - \hat{\vec{x}}_k) \quad \text{mit} \quad \mathbf{H}_k = \left. \frac{\partial \hat{\rho}(\hat{\vec{x}})}{\partial \hat{\vec{x}}} \right|_{\hat{\vec{x}}=\hat{\vec{x}}_k} \quad (2.116)$$

\mathbf{H}_k ist die Jacobi-Matrix mit den partiellen Ableitungen der $\hat{\rho}(\hat{\vec{x}}, \hat{p}_i)$ aus (2.114) für alle $i \in \{1, 2, \dots, n\}$ nach den Komponenten von $\hat{\vec{x}}$.

Damit lautet der Iterationsschritt [28, 84]:

$$\hat{\vec{x}}_{k+1} = \hat{\vec{x}}_k + \Delta \hat{\vec{x}}_k \quad \text{mit} \quad \Delta \hat{\vec{x}}_k = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (\tilde{\rho} - \hat{\rho}(\hat{\vec{x}}_k)) \quad (2.117)$$

Beginnend mit einem geeigneten Startwert¹⁴ $\hat{\vec{x}}_0$ wird (2.117) so oft ausgeführt, bis $|\Delta \hat{\vec{x}}_k|$ ausreichend klein ist. Dann enthält $\hat{\vec{x}}_k$ die gesuchten Schätzwerte für die Empfängerantennenposition \hat{p}_A und das Empfängeruhrenoffset $\hat{\Delta t}_e$.

2.5.2.9 Dilution of Precision (DOP)

Die Geometrie der Satellitenkonstellation zum Empfangszeitpunkt hat Einfluss darauf, wie sich der Pseudorangefehler $\vec{\chi}$ auf den Fehler $\delta \vec{x}$ der Positions- und Uhrenoffsetlösung \vec{x} (2.105) auswirkt:

Nach Konvergenz der Iteration (2.117) ist $\Delta \hat{\vec{x}}_k \approx 0$. Mit (2.107) und der Linearisierung von (2.104) wie in (2.116) lässt sich damit für Abhängigkeit des Fehlers $\delta \vec{x}$ vom Pseudorangefehler $\vec{\chi}$ (2.107) zeigen (vgl. [28]):

$$\delta \vec{x} = \vec{x} - \hat{\vec{x}} = -(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \vec{\chi} \quad (2.118)$$

Unter der Annahme, dass der Pseudorangefehler $\vec{\chi}$ einen Mittelwert $\mu_{\vec{\chi}} = 0$ und eine Kovarianzmatrix $Cov(\vec{\chi}, \vec{\chi}) = \sigma_{\rho}^2 \mathbf{I}$ hat, gilt für die Kovarianzmatrix \mathbf{P}_x des Fehlers $\delta \vec{x}$ [84], vgl. (A.14):

$$\begin{aligned} \mathbf{P}_x &= Cov(\delta \vec{x}, \delta \vec{x}) = E[(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \sigma_{\rho}^2 \mathbf{I} \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1, T}] \\ &= \mathbf{G} \sigma_{\rho}^2 \quad \text{mit} \quad \mathbf{G}^T = (\mathbf{H}^T \mathbf{H})^{-1} \end{aligned} \quad (2.119)$$

¹³Werden die Pseudoranges mehrerer GNSS benutzt, müssen u. U. zusätzlich noch die genauen Zeitoffsets zwischen den GNSS geschätzt werden.

¹⁴Wenn kein besserer Startwert vorliegt, können $\hat{p}_{A,0} = 0$ und $\hat{\Delta t}_{e,0} = 0$ verwendet werden [28].

Der Hauptdiagonalen von \mathbf{G} kann entnommen werden, wie die Varianz σ_ρ^2 der Pseudorangefehler/UERE auf die Varianz von Position und Uhrenoffset auswirkt. Als Kennzahlen werden üblicherweise die sog. DOP-Faktoren („dilution of precision“) verwendet:

$$\text{Geometrie-DOP: } \text{GDOP} = \sqrt{\mathbf{G}_{11} + \mathbf{G}_{22} + \mathbf{G}_{33} + \mathbf{G}_{44}} \quad (2.120)$$

$$\text{Positions-DOP: } \text{PDOP} = \sqrt{\mathbf{G}_{11} + \mathbf{G}_{22} + \mathbf{G}_{33}} \quad (2.121)$$

$$\text{horizontaler DOP: } \text{HDOP} = \sqrt{\mathbf{G}'_{11} + \mathbf{G}'_{22}} \quad (2.122)$$

$$\text{vertikaler DOP: } \text{VDOP} = \sqrt{\mathbf{G}'_{33}} \quad (2.123)$$

Für den horizontalen und vertikalen DOP müssen die Positionselemente in \mathbf{G} mit der Rotationsmatrix \mathbf{C}_e^n vom ECEF ins Navigationskoordinatensystem gedreht werden:

$$\mathbf{G}' = \begin{pmatrix} \mathbf{C}_e^n & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{G} \begin{pmatrix} \mathbf{C}_e^n & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}^T \quad (2.124)$$

Bei DOP-Faktoren um die 1 werden die Pseudorangefehler durch eine günstige Geometrie der Satellitenkonstellation nicht verstärkt. In ungünstigen Geometrien, in denen die Zeilen von \mathbf{H} fast linear abhängig werden, wachsen die DOP-Faktoren stark an. Je mehr Satelliten verwendet werden können, desto kleiner werden die DOP-Werte.

2.5.3 Differentielle Korrekturverfahren

Durch die Verwendung von Korrekturdaten kann die Genauigkeit der GNSS-Positions- und Geschwindigkeitslösung erhöht werden. Dabei empfangen sog. Referenz- oder Basisstationen, deren Position bekannt sind, die Satellitensignale und messen Pseudorange, Deltarange und Trägerphasen der sichtbaren Satelliten. Durch Vergleich der aufgrund der Referenzstationsposition zu erwartenden Beobachtungsgrößen mit den gemessenen können deren Fehler ermittelt werden. Über einen Kommunikationskanal werden dann – je nach Verfahren – die Messungen der Referenzstation selbst, die bestimmten Fehler oder die Parameter von Fehlermodellen an die GNSS-Empfänger („rover“) gesendet.

Mit diesen Korrekturdaten kann ein GNSS-Empfänger dann die Common-Mode-Fehler seiner Pseudorange- (2.107) und Trägerphasen-Messungen (2.109) kompensieren. Dabei können die Ephemeridenfehler E_i und Satellitenuhrenfehler δt_i vollständig eliminiert werden, da sie sich identisch auf die Messungen von Referenzstation und GNSS-Empfänger auswirken. Je kleiner der Abstand zwischen Referenzstation und Empfänger ist, desto besser werden die anderen beiden Common-Mode-Fehler, die Ionosphären- und Troposphären-Fehler I_i und T_i , kompensiert. Die Korrektur der empfängerspezifischen Fehler – Mehrwegefehler $M_{\rho_j,i}$ und $M_{\phi_j,i}$ sowie Messrauschen $\nu_{\rho_j,i}$ und $\nu_{\phi_j,i}$ – ist auf diesem Wege nicht möglich.

Bei den satellitenbasierten Augmentierungssystemen (SBAS, siehe Seite 48) werden die Parameter von Fehlerkorrekturmodellen, aus denen sich die GNSS-Empfänger ihre Korrekturen berechnen können, auf den GNSS-Frequenzen versendet. So stellt z. B. EGNOS

Korrekturen für die Ephemeriden-, Satellitenuhren- und Ionosphären-Fehler zur Verfügung [22]. Für die Troposphären-Fehler, die sich lokal stärker unterscheiden, werden keine Modellparameter übermittelt. Dadurch lassen sich mit EGNOS Positionsgenauigkeiten im Bereich von 1 bis 3 m erreichen.

Bei bodengebundenen Augmentierungssystemen (GBAS, siehe Seite 48) wird entweder eine einzelne Basisstation oder ein Netzwerk von Referenzstationen genutzt. Die Korrekturdaten werden meist in den standardisierten RTCM-Formaten versendet [60]. Je nach Art des Augmentierungssystems und der Eigenschaften des GNSS-Empfängers werden entweder nur Pseudorange-Korrekturdaten oder auch Trägerphasen-Korrekturdaten übermittelt. Die höchste Genauigkeit kann beim RTK-Verfahren erreicht werden.

RTK Beim RTK-Verfahren („Real Time Kinematic“) werden die Pseudorange- und Trägerphasen-Messungen einer Referenzstation für zwei GNSS-Frequenzen an den GNSS-Empfänger übertragen. Dadurch kann der Empfänger bei geeigneten Empfangsbedingungen sämtliche Common-Mode-Fehler in seinen Messungen kompensieren und so die Trägerphasenmehrdeutigkeiten $N_{j,i}$ in (2.109) auflösen. Dies erlaubt ihm, die Trägerphasenmessungen zur Bestimmung der Pseudoranges zu verwenden, vgl. (2.110). Wenn für die Trägerphasenmehrdeutigkeiten $N_{j,i}$ ganze Zahlen als Lösung bestimmt werden können („RTK-Fix-Modus“), können Positionsgenauigkeiten im Bereich von 1 bis 2 cm erreicht werden. Wenn die Trägerphasenmehrdeutigkeiten nicht eindeutig aufgelöst und für die $N_{j,i}$ lediglich Fließkommazahlen bestimmt werden können, wird dies „RTK-Float-Modus“ genannt.

Die Referenzstation beim RTK-Verfahren kann eine tatsächliche Referenzstation sein, die sich möglichst nah am Empfänger befindet. Alternativ berechnet das Augmentierungssystem aus einem Referenzstationsnetzwerk eine virtuelle Referenzstation in der Nähe des Empfängers. Dazu muss der Empfänger dem Augmentierungssystem seine ungefähre Position mitteilen. Erfolgt die Verbindung zum Referenzstationsnetzwerk über das Internet, wird üblicherweise das auf HTTP basierende Ntrip-Protokoll (Networked Transport of RTCM via Internet Protocol) genutzt.

3 System

3.1 Systementwurf

In Abschnitt 1.3 wurde das Ziel dieser Arbeit – die Automatisierung eines Kamera-Quadrokopters zur Verfolgung von bewegten Objekten für Filmzwecke – und die resultierende Aufgabenstellung formuliert. Im Folgenden wird der daraus abgeleitete Systementwurf vorgestellt.

Das Flugsystem, das die Kamera trägt, ist das zentrale Element des Gesamtsystems. Als Fluggerät wird hier ein Quadrokofter, ein Multikofter mit vier Rotoren, genutzt. Multikofter gehören wie Helikofter zur Kategorie der Drehflügler, sind jedoch mechanisch deutlich einfacher aufgebaut als Helikofter. Während Flugzeuge Vorteile bei höheren Geschwindigkeiten und längeren Distanzen haben, sind Drehflügler auch bei niedrigen Geschwindigkeiten und im Schwebeflug dynamisch manövrierbar¹, was hier entscheidend ist.

Für den prototypischen Flugsystem-Aufbau wird der verwendete Quadrokofter mit einem Computer-Modul, zusätzlicher Sensorik und weiteren Hardwarekomponenten ausgestattet, die für die Anwendung benötigt werden, siehe Abschnitt 3.2.1. In Abschnitt 3.2.2 wird die Softwarearchitektur des Flugsystems vorgestellt, die die prototypische Anwendungsentwicklung unterstützt.

Fluggeräte für professionelle Filmaufnahmen werden bisher üblicherweise durch zwei Personen gesteuert. Während die Pilotin oder der Pilot den Quadrokofter fliegt, kümmert sich die Kamerafrau oder der Kameramann um die Steuerung der Kamera. Hier wird ein System entwickelt, bei dem die Aufgaben der Pilotin bzw. des Piloten automatisiert werden, sodass das System durch eine Person bedienbar ist. Diese Person, die im Folgenden als Operator bezeichnet wird, kann sich so ganz auf die Positionierung und Orientierung der Kamera konzentrieren, um das gewünschte Videobild einzufangen. Dabei wird der Operator durch verschiedene Assistenzfunktionen mit unterschiedlichen Automatisierungsgraden unterstützt. Im höchsten Automatisierungsgrad wird ein zu filmendes Zielobjekt vom Quadrokofter vollautomatisch verfolgt und die Kamera selbständig auf das Objekt ausgerichtet. Das Bedienkonzept und die Hard- und Software der entwickelten Benutzerschnittstelle für den Operator wird in Abschnitt 3.3 dargestellt.

Während der Entwicklung überwacht zusätzlich ein Sicherheitspilot die Flugversuche. Dieser kann den Quadrokofter jederzeit übernehmen und ihn dann konventionell steuern,

¹Es gibt auch Flugzeuge, die schweben und senkrecht starten und landen können (VTOL).

greift im Normalfall jedoch nicht ein. Für den Prototyp muss auf eine automatische Hinderniserkennung und Kollisionsvermeidung verzichtet werden, diese Aufgabe übernimmt der Sicherheitspilot.

Um ein Objekt automatisch verfolgen und die Kamera darauf ausrichten zu können, muss der Regelung auf dem Quadrocopter die aktuelle Position des Objekts zur Verfügung stehen. Dazu können folgende Alternativen eingesetzt werden:

- Das Objekt wird durch Bildverarbeitung im Kamerabild erkannt und verfolgt. Daraus wird die relative Position berechnet.
- Das Zielobjekt wird mit einem GNSS-Empfänger ausgestattet und sendet seine GNSS-Position per Funk an den Quadrocopter.
- Das Zielobjekt führt einen Transponder mit, dessen Entfernung und Richtung vom Quadrocopter bestimmt werden kann.

Alle drei Varianten haben Vor- und Nachteile, die für die konkrete Anwendung abgewogen werden müssen. Die Bildverarbeitung hat den Vorteil, dass das Zielobjekt nicht eigens ausgestattet werden muss, um es verfolgen zu können. Dies wäre bei Aufnahmen z. B. von Wildtieren oder polizeilichen/militärischen Zielen entscheidend. Nachteil bei der Nutzung des Kamerabildes ist jedoch, dass die Verfolgung scheitert, wenn die Kamera nicht schnell genug nachgeführt oder das Zielobjekt verdeckt wird oder das Zielobjekt im Bild nicht mehr eindeutig erkennbar ist. In diesen Fällen ist die Verfolgung per GNSS und Funk oder per Transponder im Vorteil, solange der GNSS-Empfang ausreichend ist und die Funkverbindung erhalten bleibt. Hier soll die zweite Lösung umgesetzt werden, bei dem das zu verfolgende Objekt eine Ziel-Einheit mit GNSS-Empfänger und Funkmodul mit sich trägt, denn dies ermöglicht eine zuverlässige Verfolgung.

Zur Übertragung der Position des Zielobjekts an den Quadrocopter sowie zur Kommandierung und Überwachung des Systems etc. müssen Quadrocopter, Ziel-Einheit und Benutzerschnittstelle miteinander kommunizieren können. Die dazu entwickelte Kommunikationsinfrastruktur wird in Abschnitt 3.4 dargestellt.

In Abschnitt 3.5 werden die verschiedenen Testumgebungen, die zur Erprobung des Systems zum Einsatz kommen, vorgestellt.

3.2 Flugsystem

3.2.1 Hardwareaufbau

Als Basis für das Flugsystem wird der „Pelican“ von Ascending Technologies² (Asc-Tec) [41] genutzt, der sich aufgrund seiner Flexibilität besonders für die Prototypentwicklung eignet. Dieser Quadrokoopter ist speziell für den Forschungsbereich konzipiert. Die Kommunikationsschnittstellen des Pelican lassen alle benötigten Steuerungsarten zu und stellen sämtliche Sensordaten zur Verfügung. Zudem lässt sich der Quadrokoopter in den Türmen ober- und unterhalb der Rotorarme leicht mit zusätzlichen Komponenten erweitern.

Zur Realisierung der Anwendung sind auf dem Pelican zusätzlich ein leistungsfähiges Computer-Modul, weitere Sensorik und ein Kamera-Gimbal installiert. In Abbildung 3.1 ist das voll aufgebaute Flugsystem zu sehen. Die Abbildung 3.2 zeigt die Hardwarearchitektur, deren Komponenten im Folgenden dargestellt werden.

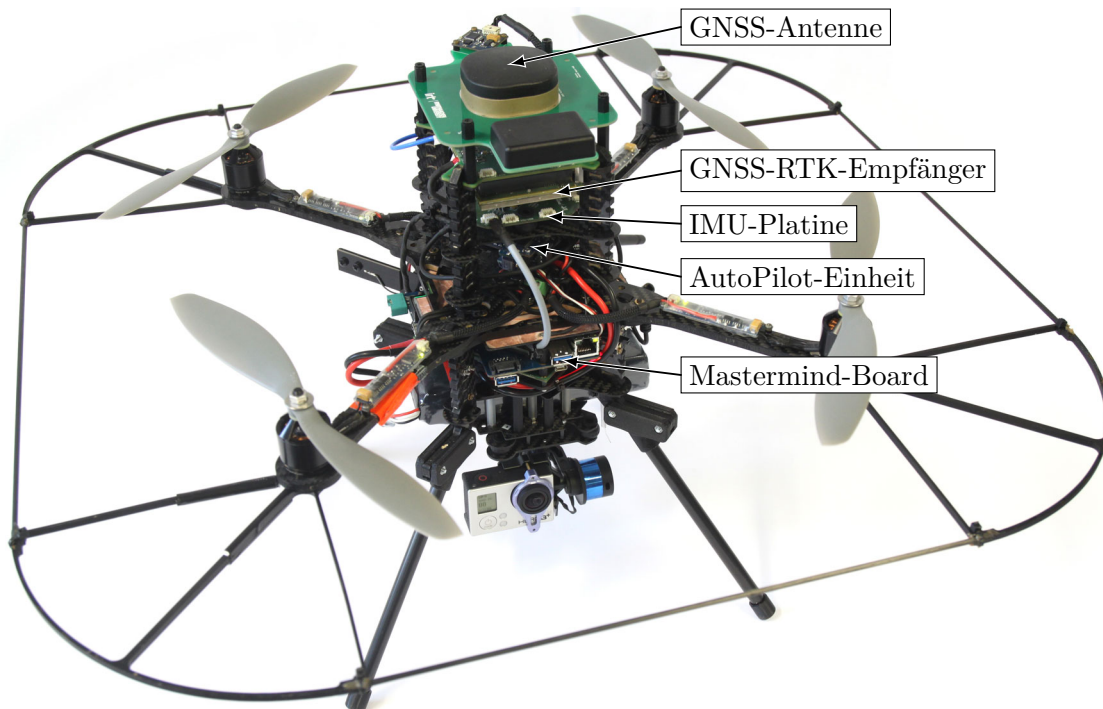


Abbildung 3.1: Quadrokoopter

²Ascending Technologies gehört seit Anfang 2016 zu Intel. Ende März 2018 wurde der Vertrieb der Multikoopter für den Forschungsbereich, inklusive der des Pelican, eingestellt.

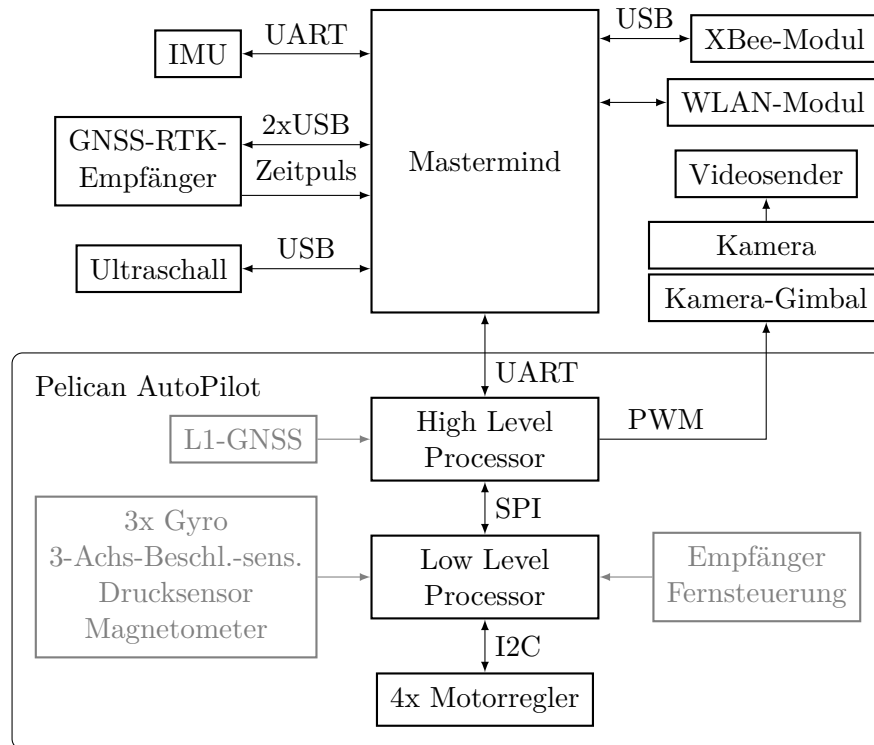


Abbildung 3.2: Hardwarearchitektur des Flugsystems

Quadrokopter „Pelican“ Der AscTec Pelican ist ein voll ausgestatteter, flugfähiger Quadrokopter. Die Flugregelungseinheit „AutoPilot“ des Pelican verfügt über zwei ARM7-Mikrocontroller, drei Drehratensensoren, einen 3-Achs-Beschleunigungssensor, einen Luftdrucksensor, ein Magnetometer und einen Einfrequenz-GNSS-Empfänger [7], vgl. Abbildung 3.2. Der Quadrokopter wird mit einem 3S-Lithium-Polymer-Akku versorgt. Während auf dem einen Mikrocontroller, dem „Low Level Processor“ (LLP), immer eine Flugregelung läuft, mit der der Quadrokopter per Modellbau-Fernsteuerung geflogen werden kann, ist der zweite Mikrocontroller, der „High Level Processor“ (HLP), frei programmierbar.

Die Flugregelung des LLP erlaubt neben einem manuellen, lagestabilisierten Flugmodus auch GPS-Flugmodi mit zusätzlicher Höhen- und Positionsstabilisierung. Die vier Motorregler für die Brushless-Motoren werden vom LLP über einen I2C-Bus angesteuert. Im Notfall kann der Sicherheitspilot mit der Fernsteuerung jederzeit vom HLP auf den LLP umschalten und den Quadrokopter dann manuell fliegen.

Der HLP wird mithilfe eines Software Development Kits per C-Code programmiert. Der funktionale Kern des Codes wird dabei aus einem Simulink-Modell generiert. Auf dem HLP lassen sich unterschiedliche Steuerungsmodi des LLP nutzen. So können entweder direkt Motordrehzahlen vorgegeben, die Flugmodi des LLP genutzt oder GPS-Wegpunkte kommandiert werden. Hier wird der Motordrehzahl-Modus genutzt (vgl. Abbildung 5.6).

Computer-Modul „Mastermind“ Auch wenn der HLP eine flexible Steuerung des Quadropters mit 1000 Hz zulässt, ist die Rechenleistung beschränkt. Um die für die vorgesehene Sensorfusion und Regelung ausreichende Rechenleistung bereitzustellen, wird das AscTec „Mastermind“ Computer-Modul als Erweiterung des Pelican genutzt. Dieser Computer verfügt über einen Intel Core i7 Prozessor mit 1,7 GHz, 4 GB Arbeitsspeicher, eine 64 GB SSD-Festplatte und ein WLAN-Modul. Weitere Hardware lässt sich per USB, Ethernet, UART, GPIO-Pins etc. anschließen. Als Betriebssystem läuft auf dem Computer Ubuntu 14.04.

Sensorik-Erweiterungen Neben den Sensoren, die die AutoPilot-Einheit mitbringt, wird das Flugsystem mit weiteren Sensoren ausgestattet. So wird für die Sensorfusion eine separate IMU genutzt, siehe Abschnitt 4.1.1. Da die Drifts der IMU-Sensoren geringer als die der AutoPilot-Sensoren sind, sind die IMU-Sensoren für die Sensorfusion besser geeignet. Die separate IMU verfügt zudem über ein weiteres Magnetometer.

Anstatt des Einfrequenz-GNSS-Empfängers kommt für die Anwendung ein GNSS-RTK-Empfänger mit einer deutlich höheren Positionsgenauigkeit zum Einsatz, siehe Abschnitt 4.1.2.1. Als weiterer GNSS-Empfänger wird ein experimenteller GPS- und Galileo-Empfänger erprobt, siehe Abschnitt 4.1.2.2.

Außerdem ist der Quadropters mit einem Devantech SRF10 Ultraschallsensor ausgestattet, um beim Landen die Höhe über dem Boden messen zu können, vgl. Abschnitt 5.4.

Kamera-Gimbal Als Videokamera wird eine GoPro HERO3+ Full-HD-Kamera eingesetzt, siehe Abbildung 3.3. Durch das Weitwinkel-Objektiv kann ein großer Bereich gefilmt werden, siehe Abbildung 3.4.

Die Kamera ist in einem Tarot TL68A00 2-Achs-Brushless-Gimbal unter dem Quadropters montiert, siehe Abbildung 3.3. So kann durch Schwenken des Kamera-Gimbals um



Abbildung 3.3: Kamera im 2-Achs-Gimbal



Abbildung 3.4: Kamera-Videobild

seine zwei Achsen und durch Drehen des Quadropters um die Hochachse die Kamera auf das zu filmende Objekt ausgerichtet werden, siehe Abschnitt 5.3.1.2.

In der Erprobung des Systems soll die Genauigkeit der Objektverfolgung anhand der Videobilder der Kamera ausgewertet werden. In Abschnitt 3.2.1.1 wird das dazu benötigte Projektionsmodell der verwendeten Kamera identifiziert.

Funkmodule Der Quadroptor verfügt über diverse Funkmodule für unterschiedliche Zwecke. Mit dem Empfänger der Modellbau-Fernsteuerung empfängt der LLP des Pelican die Befehle des Sicherheitspiloten. Das Mastermind-Board bringt ein WLAN-Modul mit. Da die Reichweite und Kontinuität der WLAN-Verbindung für die zuverlässige Kommunikation mit dem Quadroptor im Flug nicht ausreicht, wird zusätzlich ein XBee-Funkmodul genutzt, mit dem der Quadroptor Kommandos empfangen und Statusinformationen senden kann, siehe Abschnitt 3.4. Zur Überwachung des Kamera-Bildes ist zudem ein 5,8 GHz Videosender an der Kamera angeschlossen, siehe Abbildung 3.6.

Platinen Zur Integration einiger Komponenten werden maßgeschneiderte Platinen benötigt. So kann der GNSS-RTK-Empfänger nur über eine 80-polige Stiftleiste elektrisch angeschlossen werden. In Abbildung 3.5 ist die dazu in EAGLE entworfene Adapterplatine im CAD-Modell und als bestückte Platine zu sehen. Abbildung 4.5 auf Seite 108 zeigt die Adapterplatine im Quadroptor montiert.

Auf der Adapterplatine sind neben der Stiftleiste ein 3,3 V-Spannungswandler zur Versorgung des GNSS-RTK-Empfängers, ein Slot zur Aufnahme einer SD-Karte zum Loggen und zwei Seriell-zu-USB-Konverter installiert. Ein Reset-Taster und ein über einen Schmitt-Trigger entprellter Taster mit konfigurierbarer Funktion erlauben die direkte Kommandierung des Empfängers. Über LEDs kann der Systemstatus abgelesen werden. Zur bidirektionalen Kommunikation (GNSS-Daten, Korrekturdaten, Kommandierung) ist der AsterRx3 doppelt per USB (davon einmal über einen Seriell-zu-USB-Konverter) mit

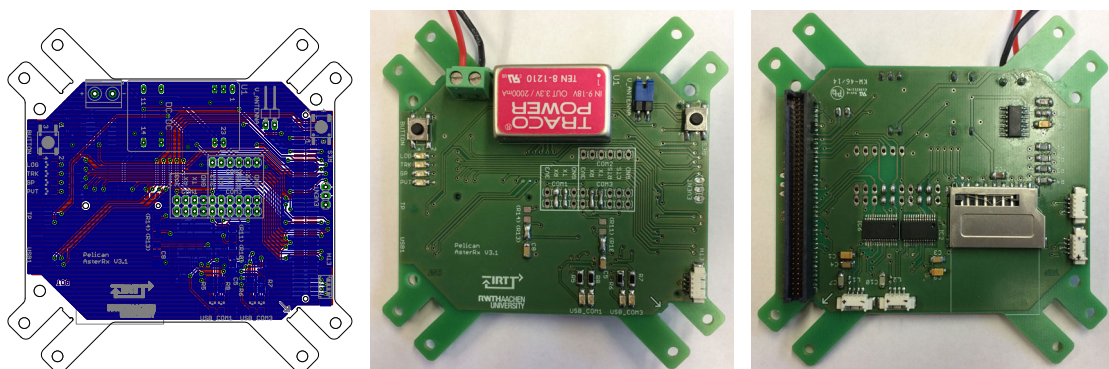


Abbildung 3.5: Adapterplatine für GNSS-RTK-Empfänger

dem Mastermind-Board verbunden. Außerdem ist der digitale Zeitpuls-Ausgang am Mastermind-Board angeschlossen, damit für die Sensorfusion das Alter der GNSS-Nachrichten berechnet werden kann, vgl. Seite 109.

Neben der Adapterplatine für den GNSS-RTK-Empfänger sind weitere Platinen für die IMU (Abbildung 4.1), die GNSS-Antenne (Abbildung 4.5) und das Kamera-Gimbal auf dem Quadrocopter installiert.

3-D-Druck-Bauteile Zur Befestigung von Komponenten wie den Ultraschallsensor werden 3-D-gedruckte Bauteile aus Kunststoff verwendet. Diese wurden im CAD-Programm CATIA entworfen und im FDM-Verfahren (Fused Deposition Modeling) gedruckt. In Abbildung 3.6 ist ein so hergestellter Adapter für das Landegestell zu sehen, in den ein Halter für das XBee-Modul integriert ist.

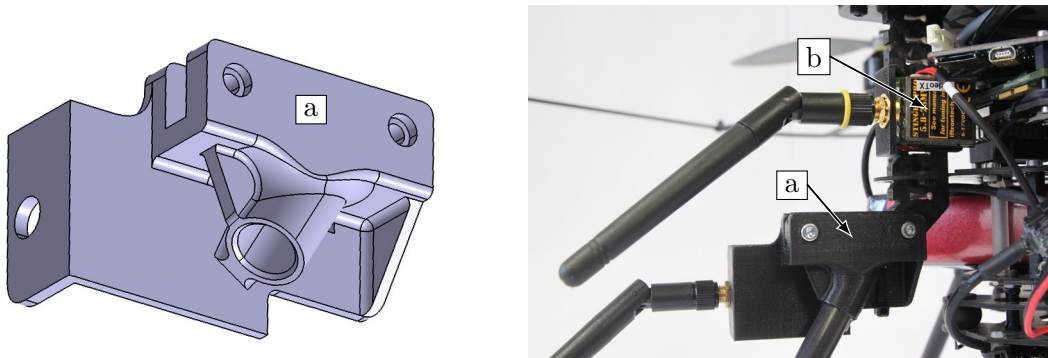


Abbildung 3.6: 3-D-gedruckter Adapter (a) für Landegestell und XBee-Modul (links im CAD-Modell und rechts montiert am Quadrocopter), Videosender (b) im 3-D-gedruckten Halter

3.2.1.1 Projektionsmodell Kamera

Um in der Erprobung in Abschnitt 6.2.5 die Genauigkeit der Objektverfolgung anhand der Videobilder bestimmen zu können, wird die Kamera-Abbildung von realen 3-D-Punkten auf die 2-D-Bildpunkte (Pixel) bestimmt.

Das Weitwinkel-Objektiv der Kamera führt zu einer Tonnenverzeichnung: gerade Linien werden gebogen dargestellt, was in den Abbildungen 3.4 und 3.7 gut zu erkennen ist. Das Projektionsmodell der Kamera wird anhand von mehreren Aufnahmen eines Schachbrettmusters (Abbildung 3.7) in verschiedenen Posen mithilfe der OCamCalib-Toolbox [69, 70] identifiziert.

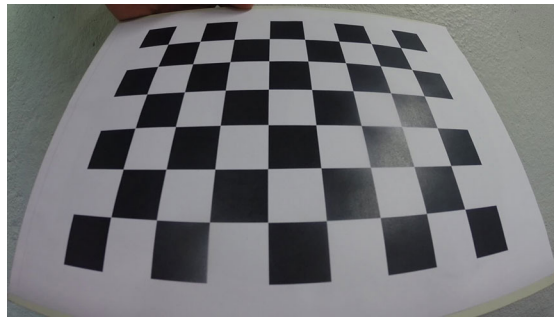


Abbildung 3.7: Schachbrettmuster

Gemäß dem Projektionsmodell der OCamCalib-Toolbox gilt für den 3-D-Einheitsvektor \vec{p} , der von der Kamera in Richtung des 3-D-Punktes zeigt, welcher im Bildpunkt (u, v) zu sehen ist:

$$\begin{pmatrix} u_e \\ v_e \end{pmatrix} = \begin{pmatrix} e_0 & e_1 \\ e_2 & 1 \end{pmatrix}^T \begin{pmatrix} u - e_3 \\ v - e_4 \end{pmatrix} \quad (3.1a)$$

$$\vec{p} = \begin{pmatrix} p_u \\ p_v \\ p_f \end{pmatrix} = \frac{\vec{p}^*}{|\vec{p}^*|} \quad \vec{p}^* = \begin{pmatrix} u_e \\ v_e \\ f(\rho) \end{pmatrix} \quad f(\rho) = \sum_{i=0}^4 a_i \rho^{(i)} \quad \rho = \sqrt{u_e^2 + v_e^2} \quad (3.1b)$$

Ein Bildpunkt (u, v) wird in (3.1a) verschoben und entzerrt. In (3.1b) wird (u_e, v_e) mithilfe des Polynoms $f(\rho)$ in Abhängigkeit vom Abstand zum Bildmittelpunkt ρ in einen 3-D-Vektor transformiert und auf die Länge 1 normiert. Die für die verwendete Weitwinkel-Kamera identifizierten intrinsischen Kameraparameter e_i und a_i sind in Tabelle 3.1 zu finden.

i	0	1	2	3	4
a_i	-833,50	0	$1,358 \cdot 10^{-4}$	$7,278 \cdot 10^{-7}$	$-4,931 \cdot 10^{-10}$
e_i	1,016	$-1,111 \cdot 10^{-3}$	$5,550 \cdot 10^{-4}$	529,17	952,55

Tabelle 3.1: Intrinsische Kamera-Parameter

Der Gierwinkel α_v und der Nickwinkel α_u , um die die Kamera gedreht werden muss, um einen Bildpunkt an der Position (u, v) in das Zentrum des Bildes zu bringen, lassen sich aus dem Einheitsvektor \vec{p} berechnen:

$$\alpha_v = \operatorname{atan} \left(\frac{p_v}{-p_f} \right) \quad \alpha_u = \operatorname{atan} \left(\frac{p_u}{\sqrt{p_v^2 + p_f^2}} \right) \quad (3.2)$$

Die Projektion der Winkel α_u und α_v auf die Bildpunkte (u, v) ist für die identifizierten Kameraparameter in Abbildung 3.8 dargestellt. Es ist die Innenansicht der Einheitskugel zu sehen, auf deren Hülle alle Punkte \vec{p} liegen. Die Kreise konstanter Winkel α_v und α_u stellen die Längen- und Breitengrade dieser Kugel dar.

In Abbildung 3.8 kann abgelesen werden, dass das Sichtfeld der Kamera horizontal ca. $\pm 62^\circ$ und vertikal ca. $\pm 35^\circ$ beträgt.

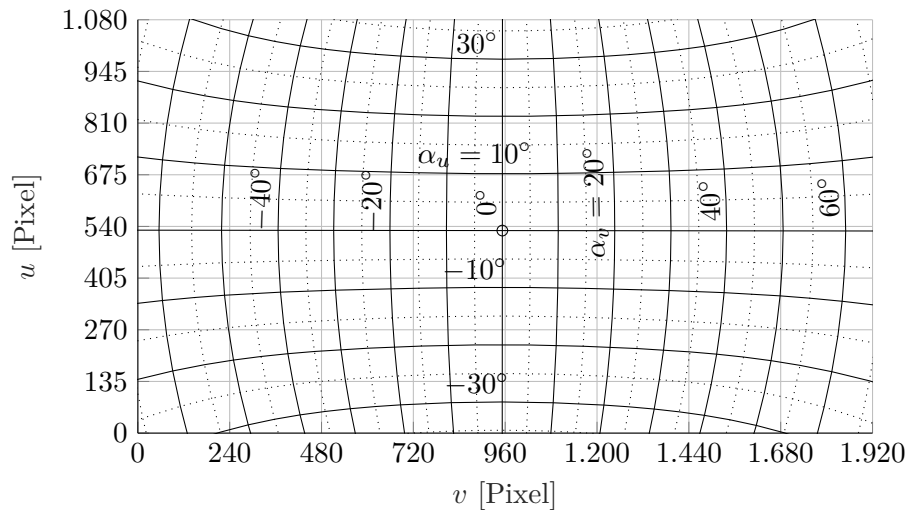


Abbildung 3.8: Weitwinkel-Kamera: Projektion Einheitskugel auf Bildpunkte (u, v)

3.2.2 Softwarearchitektur

Neben der Hardware, die zur Automatisierung des Kamera-Quadropters erforderlich ist (siehe Abschnitt 3.2.1), wird eine Software benötigt, die die Algorithmen der Sensorfusion und Regelung (vgl. Kapitel 4 und 5) sowie die Interaktion mit dem Operator (vgl. Abschnitt 3.3) auf dem Quadropters realisiert. In diesem Abschnitt wird die Architektur für diese Software dargestellt. Um die prototypische Anwendungsentwicklung zu unterstützen, erlaubt diese Architektur eine schnelle Anpassung der Funktionalsoftware und eine transparente Erprobung auf dem Quadropters. Einen Überblick über die entworfene Softwarearchitektur gibt Abbildung 3.9.

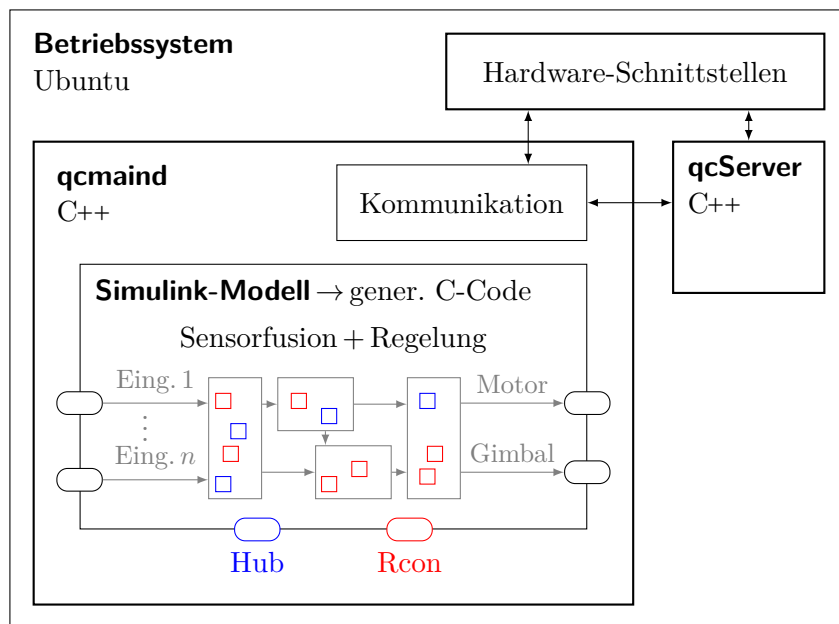


Abbildung 3.9: Architektur der Software auf dem Mastermind

Um die Funktionalsoftware einfach ändern und weiterentwickeln zu können, wird ein modellgetriebener Ansatz gewählt, bei dem aus einem graphischen Modell Programmcode generiert wird. Dies hat zugleich den Vorteil, dass dasselbe Modell, das zur Codegenerierung benutzt wird, auch direkt in einer Simulation des Quadropters verwendet werden kann, siehe Abschnitt 3.5.1. Als Tool kommt dabei MathWorks Simulink zum Einsatz.

Für eine zielgerichtete Erprobung ist es notwendig, dass Werte von Variablen der Funktionalsoftware direkt gelesen und Parameter kurzfristig geändert werden können. Die entwickelte Architektur unterstützt das Lesen von Variablen und das Ändern von Parametern zur Laufzeit – auch während des Fluges, siehe „Rcon“ unten.

Die Software auf dem Mastermind wird durch die zwei Programme „qcmaint“³ und „qcServer“ realisiert, die unter Ubuntu 14.04, dem Betriebssystem des Mastermind Compu-

³ „qcmaint“ steht für „Quadropters-Hauptprogramm“ und „d“ für „daemon“

ter-Moduls, ausführbar sind, sodass sie auf die Hardwareschnittstellen des Betriebssystems zugreifen können. Beide Programme sind als Hintergrundprozesse („daemon“) konzipiert, da im Flugbetrieb weder eine Ubuntu-Desktopoberfläche läuft noch ein Nutzer auf der Kommandozeile eingeloggt ist. Das Hilfsprogramm qcServer wird vom Betriebssystem bei Systemstart gestartet und erlaubt das Starten und Beenden des Hauptprogramms qcmaind auf Kommando des Operators. Damit ist ein Neustart des Hauptprogramms ohne Betriebssystemneustart möglich.

Während die beiden Programme qcmaind und qcServer objektorientiert in C++ entwickelt werden, erfolgt die Programmierung der Algorithmen zur Sensorfusion und Regelung graphisch in einem Simulink-Modell. Aus dem Modell wird C++-Code generiert, der von dem Hauptprogramm qcmaind eingebunden wird, siehe Abbildung 3.9. Dadurch, dass sowohl textuell in C++ als auch graphisch in Simulink programmiert wird, werden die Stärken der jeweiligen Programmierweise genutzt:

Die graphische Programmierweise mit Simulink ist vor allem bei signalorientierten Programmteilen wie der Regelung vorteilhaft. Auch lassen sich größere ereignisdiskrete Systeme, wie die Ablaufsteuerung des Quadropters (siehe Abbildung 5.15 auf Seite 196), gut graphisch z. B. in MathWorks Stateflow programmieren. Für vorwiegend prozedurale Algorithmen z. B. mit Verzweigungen oder Schleifen ist es meist sinnvoller, textuelle Programmiersprachen zu verwenden. So ist das Kalman-Filter für die Sensorfusion (vgl. Abbildung 4.7 auf Seite 139) innerhalb eines Simulink-Blockes in der textuellen MATLAB-Sprache geschrieben. Dabei kann nur der Teil der MATLAB-Sprache verwendet werden, der vom MATLAB Coder zur C/C++-Codegenerierung unterstützt wird. Die Programmierung mit den nutzbaren Sprachelementen ähnelt sehr der Programmierung in C.

Programmteile wie der Zugriff auf die Hardware-Schnittstellen und das Parsen von Kommunikationsprotokollen werden direkt in C++ erstellt. Hier würde eine Einbindung in das Simulink-Modell zusätzlichen Aufwand und keinen größeren Nutzen bedeuten. Bei der Programmierung in C++ können „Boost“-Bibliotheken, die den Umfang der C++ Standard-Bibliotheken erweitern, genutzt werden. Boost abstrahiert z. B. die Schnittstellen zur seriellen und IP-basierten Kommunikation, sodass im Anwendungsprogramm komfortabel auf die Schnittstellen zugegriffen werden kann.

Die primären Schnittstellen zwischen dem Programmteil, der aus dem Simulink-Modell generiert wird, und dem restlichen Programm werden in Form von Ein- und Ausgangs-Ports auf der obersten Ebene des Simulink-Modells angelegt. Die Eingangs-Ports werden u. a. für Sensorsignale und die Ausgangs-Ports für die Kommandierung von Motordrehzahlen und Gimbal-Winkeln verwendet. Zu Beginn eines Zeitschrittes des Simulink-Programmteils werden die Eingänge gelesen und am Ende des Zeitschrittes die Ausgänge geschrieben. Während des Zeitschrittes ist so keine Interaktion der Programmteile möglich. Um die Zugriffsmöglichkeiten zu erweitern, wurde eine zusätzliche Schnittstellenart entworfen, die im Folgenden „Simulink-Hub“ genannt wird. Mit diesem Simulink-Hub ist es möglich, mit Simulink-S-Function-Blöcken C++-Methoden des restlichen Programmteils aufzurufen. Während eines Zeitschrittes können auf diese Weise mehrfach Werte

gelesen und geschrieben werden. So wird z. B. erst die Zahl der vorliegenden IMU-Nachrichten abgefragt, um dann die Nachrichten nacheinander zu lesen. Ein weiterer Vorteil der Simulink-Hub-Schnittstellen ist, dass die Blöcke auf beliebigen Ebenen des Simulink-Modells platziert werden können.

Die Simulink-Hub-Blöcke werden auch verwendet, um die Kommandos des Operators ein- und Statusinformationen auszulesen. Die Kommandos und Statusinformationen werden von qcmaind als Nachrichten über das Kommunikationsnetzwerk per XBee empfangen bzw. versendet, siehe Abschnitt 3.4.

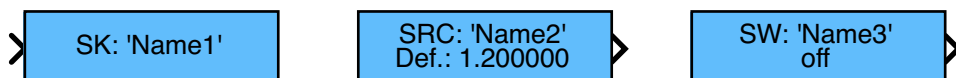


Abbildung 3.10: Rcon Simulink-Block-Typen: Senke, Quelle und Schalter (von links nach rechts)

Wie oben erläutert, ist es für die Erprobung entscheidend, dass Variablen der Funktionalsoftware während des Betriebes gelesen und geschrieben werden können. Nur so können z. B. interne Zustände analysiert, Reglerparameter effizient appliziert oder Teilfunktionen an- oder ausgeschaltet werden. Für diesen Zweck wurde neben dem Simulink-Hub eine weitere Schnittstellenart namens „Rcon“ („remote connection“) entwickelt. Rcon definiert drei Simulink-Block-Typen: eine Senke und eine Quelle für Fließkommazahlen sowie einen Schalter als binäre Quelle, die „wahr“ oder „falsch“ sein kann, siehe Abbildung 3.10. Für die Blöcke müssen eindeutige Namen und für die beiden Quellen zusätzlich Initialwerte

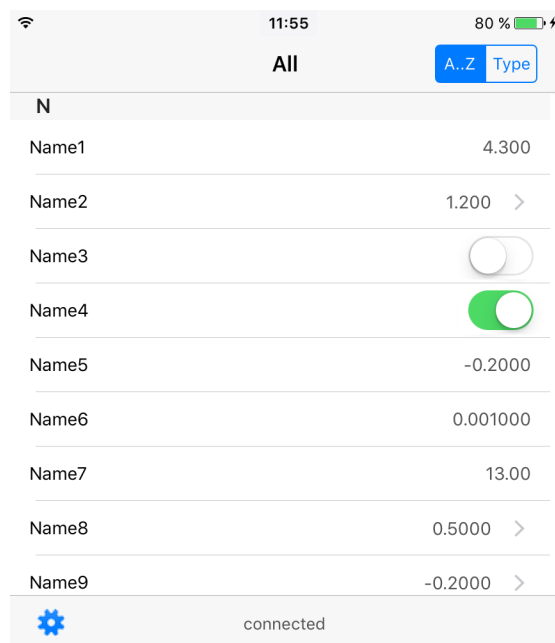


Abbildung 3.11: Rcon iOS-App

festgelegt werden. Wie die Simulink-Hub-Blöcke können die Rcon-Blöcke an beliebiger Stelle im Simulink-Modell platziert werden. Zu Programmstart werden alle Rcon-Blöcke automatisch registriert. Um die Werte dieser Blöcke zur Laufzeit aus der Ferne lesen bzw. setzen zu können, wurde u. a. eine iOS-App geschrieben, die auf einem iPhone oder iPad läuft, siehe Abbildung 3.11. Die App zeigt die Werte aller Blöcke live an und erlaubt das Setzen von Werten für die Fließkommazahl-Quellen bzw. das Betätigen der Schalter. Zur Kommunikationsinfrastruktur siehe Abschnitt 3.4.

Neben der Möglichkeit, per Rcon Signale live anzusehen, werden alle relevanten Signale aus dem Simulink-Modell auf dem Quadrokopter geloggt und im MAT-Dateiformat gespeichert. Dies erlaubt dann eine detaillierte Auswertung nach einem Flugversuch.

Die Regelung des Quadrokopters berechnet seine Stellgrößen in Form von Motor-Drehzahlen (5.112) und Gimbalwinkeln (5.128)f. Nach einer Skalierung der Kommandos sendet das qcmaind Programm sie über die UART-Schnittstelle an den HLP des AutoPilot (vgl. Abbildung 3.2). Der HLP leitet die Motordrehzahlkommandos direkt an den LLP und die Gimbalwinkelkommandos in Form von PWM-Signalen an das Gimbal weiter. Auf dem HLP ist sonst keine weitere Signal-Prozessierung notwendig.

3.3 Benutzerschnittstelle

3.3.1 Bedienkonzept

Die Aufnahme eines Videos von einem bewegten Objekt aus der Luft ist eine herausfordernde Aufgabe. Daher sollen dem Operator verschiedene Betriebsmodi mit mehreren Automatisierungsstufen zur Verfügung stehen, die ihn je nach Einsatzszenario bestmöglich unterstützen und ihm gleichzeitig die benötigte bildgestalterische Freiheit lassen. Ein geeignetes Bedienkonzept und eine intuitive Benutzerschnittstelle ist entscheidend für die Nutzbarkeit des Systems.

Die sechs Freiheitsgrade der Kamera – die Position und die Orientierung jeweils in drei Dimensionen – lassen sich durch Ansteuerung des Quadrokopters und des Kamera-Gimbals vollständig steuern⁴. Es ist i. Allg. sinnvoll, dass für die verschiedenen Freiheitsgrade der Kamera bzw. Steuerungskanäle des Systems unterschiedliche Betriebsmodi gewählt werden können. Fasst man die horizontale Position zusammen und formuliert die Orientierung der Kamera als Eulerwinkel (2.16), ergeben sich fünf Steuerungskanäle: horizontale 2-D-Position, vertikale Position, Gierwinkel, Nickwinkel, Rollwinkel. Dabei sind der Nick- und Rollwinkel durch das Gimbal beschränkt.

⁴Der Quadrokopter ist mit seinen vier Stellgrößen unteraktuiert. Er kann horizontal nicht unabhängig von seinen Nick- und Rollwinkeln beschleunigen, vgl. Abschnitt 5.1.2.1 und Abbildung 5.17. Dies wird durch das Kamera-Gimbal kompensiert (vgl. Seite 67 und Abschnitt 5.3.1.2). Es erlaubt, die Kamera unabhängig vom Quadrokopter nicken und rollen zu können, sodass die Nick- und Rollwinkel des Quadrokopters so angesteuert werden können, dass die gewünschte horizontale Beschleunigung erreicht wird.

		Steuerungskanäle				
		horizontale 2-D-Position	vertikale Position	Gierwinkel	Nickwinkel	Rollwinkel
Betriebsmodi	Objektverf.	folge Objekt		schaue auf Objekt		
	stat. Punkt	fliege zu Punkt		schaue auf Punkt (POI)		
	Joystick			Joystick		
	Abs.-winkel	–	–		Absolutwinkel	
	Trajektorien			Trajektorien		
	halten			halten		

Tabelle 3.2: Steuerungskanäle und Betriebsmodi

Die Betriebsmodi, die für den Prototyp umgesetzt werden, sind in Tabelle 3.2 bezogen auf die Steuerungskanäle dargestellt: Die Assistenzfunktion mit dem höchsten Automatisierungsgrad ist die Objektverfolgung: Der Quadrocopter verfolgt hierbei automatisch ein zu filmendes Zielobjekt und/oder richtet die Kamera selbständig auf das Objekt aus. Der Operator gibt lediglich über die Benutzerschnittstelle den gewünschten Abstand der Kamera und den Winkel zum Zielobjekt vor.

Neben der Verfolgung von bewegten Objekten kann es auch gewünscht sein, einen statischen (Weg-)Punkt anzufliegen oder einen statischen Punkt – einen Point-of-Interest (POI) – zu filmen. Dessen Position ist i. Allg. vorab bekannt, sodass die Position nicht per Funk übermittelt werden muss. Der Operator kann die Wegpunkte und POIs vorab abspeichern und zum gewünschten Zeitpunkt dann kommandieren, woraufhin die Assistenzfunktion den kommandierten Wegpunkt automatisch anfliegt bzw. die Kamera auf den kommandierten POI ausrichtet.

Für eine freie Bildgestaltung hat der Operator die Möglichkeit, einzelne Kanäle per Joystick zu steuern. Dies erlaubt dem Operator, beliebige Manöver und Kameraschwenks durchzuführen oder Zielobjekte manuell zu verfolgen. Die drei Kamerawinkel können alternativ absolut vorgegeben werden. Auf allen Steuerungskanälen lassen sich zudem Trajektorien kommandieren. Dies kann u. a. bei beschränkten Platzverhältnissen genutzt werden, indem vorab eine Trajektorie geplant wird, die an die örtlichen Gegebenheiten angepasst ist. Außerdem kann jeder Steuerungskanal „gehalten“ werden, wenn z. B. die vertikale Position oder der Kamera-Rollwinkel konstant bleiben soll.

Aus diesem Bedienkonzept lassen sich folgende Hardware- und Software-Anforderungen an die Benutzerschnittstelle ableiten:

- Wahl des Betriebsmodus pro Steuerungskanal
- Steuerung Betriebsmodus-Parameter (z. B. Abstand und Winkel zum Zielobjekt, Speichern und Kommandierung statischer Wegpunkte/POIs)

- Steuerung per Joystick
- Kommandierung von Trajektorien

Darüber hinaus soll die Benutzerschnittstelle folgende Anforderungen erfüllen:

- Kommandierung automatisches Starten/Landen, Reinitialisierung etc.
- Anzeige Statusinformationen: Systemzustand, Alarm, Batterieladezustand, Flugzeit, Positionen von Quadrokopter und Zielobjekt
- Funktionen für Erprobung (z. B. Logging starten/stoppen)
- geeignet für mobilen Outdoor-Einsatz, im Stehen bedienbar

Im Folgenden werden die Hardware und Software der Benutzerschnittstelle dargestellt, die diese Anforderungen realisieren.

3.3.2 Hardware

Für die Steuerung des Systems durch den Operator wurde ein „ControlPad“ genanntes Handbediengerät entwickelt, das mobil ist und sich z. B. im Stehen oder Gehen mit einer Hand halten und mit der anderen Hand bedienen lässt. Hauptbedienelemente des ControlPads sind ein Tablet-Computer mit Touchscreen und ein 6-Achs-Joystick, siehe Abbildung 3.12. Inspiriert wurde der Aufbau des ControlPads durch Handbediengeräte von Industrierobotern, die auch Touchscreen und 6-Achs-Joystick kombinieren.



Abbildung 3.12: ControlPad

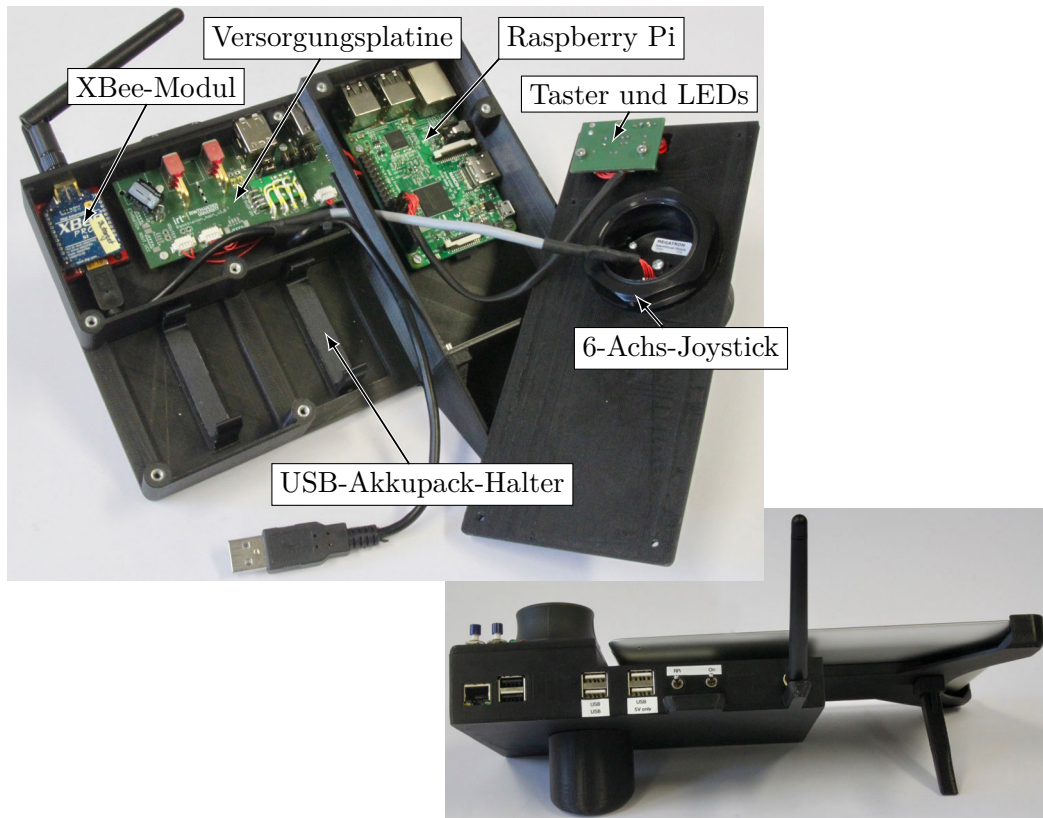


Abbildung 3.13: ControlPad Innen- und Rückansicht

Als Tablet-Computer wird ein Apple iPad genutzt, auf dem die App „QC Cam Ctrl“ läuft, die in Abschnitt 3.3.3 vorgestellt wird. Das iPad ist in ein Kunststoffgehäuse integriert, das wie einige Quadrocopter-Bauteile 3-D-gedruckt wurde (vgl. Seite 69).

Position und Orientierung der Kamera sollen in allen translatorischen und rotatorischen Achsen u. a. intuitiv von Hand steuerbar sein. Zwei klassische Joysticks bzw. Steuerknüppel wie z. B. auf einer Modellbau-Fernsteuerung bieten zusammen nur vier Steuerkanäle. Hier wird ein 6-Achs-Joystick genutzt, dessen drei rotatorischen und drei translatorischen Achsen sich mit einer Hand feinfühlig steuern lassen. Eine Zuordnung seiner Achsen zu denen des Flugsystems ist intuitiv möglich. Bei dem eingesetzten 6-Achs-Joystick handelt es sich um ein Megatron SpaceMouse Module, das neben dem Tablet in das ControlPad-Gehäuse integriert ist.

Das Gehäuse enthält weitere Komponenten, die zur Kommunikation und zum Anschluss des 6-Achs-Joysticks erforderlich sind. So sind ein Raspberry Pi 3, ein XBee-Modul, eine Versorgungsplatine und ein USB-Akkupack integriert, siehe Abbildung 3.13. Die Platine, die auch in der Bodenstation und der Ziel-Einheit verwendet wird (siehe Abschnitt 3.4), realisiert die Stromversorgung und enthält einen USB-Hub, über den der 6-Achs-Joystick am Raspberry Pi angeschlossen ist. Auf einer zweiten Platine befinden sich zwei Taster und LEDs, deren Funktion frei programmierbar ist. Das iPad ist über den im Raspberry Pi

integrierten WLAN-Chip verbunden. Um die RTK-Korrekturdaten für die GNSS-RTK-Empfänger des Quadropters und der Ziel-Einheit zu empfangen, wird über das LTE-Modem des iPads eine Mobilfunkverbindung mit einem Ntrip-Server hergestellt (zu RTK und Ntrip siehe Seite 62). Das XBee-Modul wird zur Kommunikation mit Quadropters, Bodenstation etc. verwendet, vgl. Abschnitt 3.4. Das Routing der Signale und Nachrichten zwischen den Komponenten übernimmt ein Programm auf dem Raspberry Pi.

3.3.3 Software

Die iOS-App „QC Cam Ctrl“, die auf dem iPad des ControlPads läuft, erlaubt dem Operator eine intuitive Kommandierung der verschiedenen Betriebsmodi des Systems (vgl. Tabelle 3.2) und einen Überblick über den Systemstatus. Das Icon der App, das einen stilisierten Kamera-Quadropters zeigt, ist in Abbildung 3.14 zu sehen.



Abbildung 3.14: Icon der iOS-App „QC Cam Ctrl“

Die App wurde in der Entwicklungsumgebung Xcode programmiert. Während die GUI⁵-nahen Programmteile in Objective C geschrieben sind, wird für die Programmlogik und die IP-Kommunikation C++ und Boost genutzt. Dadurch können Codeteile vom Hauptprogramm des Quadropters (siehe Abschnitt 3.2.2) und von der Software, die auf den Raspberry Pis von ControlPad, Bodenstation und Ziel-Einheit läuft (siehe Abschnitt 3.4), wiederverwendet werden.

In Abbildung 3.15 ist die Start-Ansicht der App zu sehen. In der linken Spalte befindet sich das Navigationsmenü, mit dem zwischen den Ansichten gewechselt werden kann. Die „Flight“-Ansicht auf der rechten Seite gibt einen Überblick über den Systemstatus und erlaubt die Kommandierung des Systems. Sie enthält neben den Elementen für den Operator auch Hilfsfunktionen für die Erprobung. In den ersten beiden Zeilen kann man den Verbindungsstatus zum Quadropters sehen. Sind beide Kreise grün, wie in der Abbildung, laufen die Programme qcServer und qcmaind auf dem Quadropters und es werden Daten empfangen.

Für Erprobungszwecke kann über die „Cmd“-Buttons u. a. qcmaind gestartet und gestoppt und der Mastermind neugestartet werden. Mit den Kommandos „Start all“ und „Stop all“ wird das Logging von Quadropters, Bodenstation, Ziel-Einheit und Control-Pad gestartet bzw. gestoppt. Über „Set all“ wird die Systemzeit auf allen Geräten gesetzt. In den folgenden Zeilen werden der Batterie-Ladezustand, die Flugzeit, der Alarm-Zustand und der GNSS-Empfangszustand des Quadropters angezeigt.

⁵GUI: graphical user interface, graphische Benutzerschnittstelle

3 System

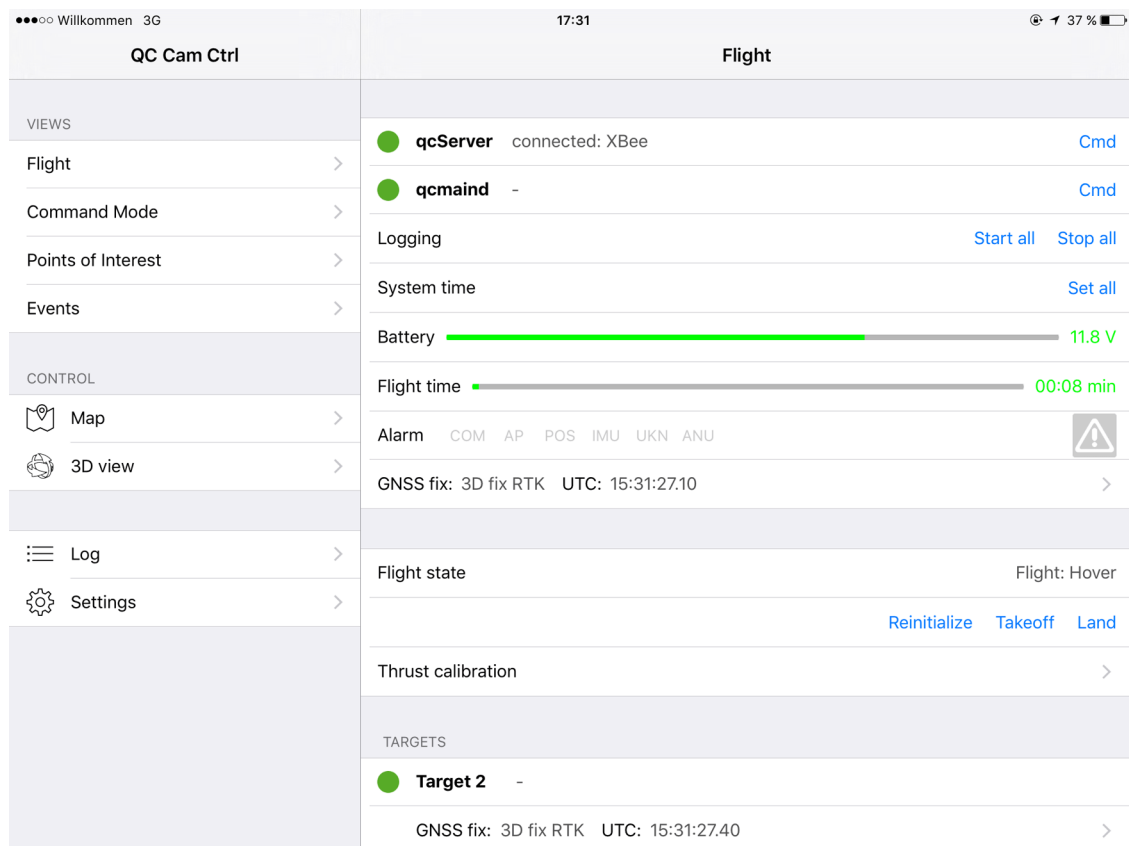


Abbildung 3.15: QC Cam Ctrl: Navigationsmenü und Ansicht „Flight“

In der zweiten Sektion sieht man unter „Flight state“ den Zustand der Ablaufsteuerung auf dem Quadrocopter (vgl. Abschnitt 5.4): Im dargestellten Fall befindet sich der Quadrocopter in der Flugphase. Mit den Buttons „Reinitialize“, „Takeoff“ und „Land“ können die Reinitialisierung, das Starten und das Landen kommandiert werden. Über „Thrust calibration“ ist die Ansicht für die Schubkalibrierung gemäß Abschnitt 5.1.2.2 erreichbar, siehe Abbildung 3.16.

Neben den visuellen Anzeigen werden Alarme und Zustandswechsel auch akustisch ausgegeben, sodass sie nicht verpasst werden können. Wenn ein Alarm vorliegt, z. B. ein Sensor ausgefallen oder die Verbindung unterbrochen ist, wird – je nach Fehler – ein Alarmton ausgegeben, sodass der Sicherheitspilot sofort mitbekommt, dass er den Quadrocopter übernehmen muss. Wichtige Zustandswechsel werden per Sprachausgabe angesagt. So hört man an Ansagen wie „Initialization“, „Ready for takeoff“, „Takeoff“ und „Landing“, dass der Quadrocopter den kommandierten Zustandswechsel bestätigt hat. Außerdem werden u. a. niedrige Batterie-Ladezustände, GNSS-Zustandswechsel und der Verlust von GNSS-Korrekturdaten angesagt.

In den Einstellungen der App können u. a. die Zugangsdaten zum Ntrip-Server gespeichert werden, sodass über die Mobilfunk-Internetverbindung des iPads RTK-Korrekturdaten

Thrust Calibration	
Front	0.870000
Back	0.870000
Left	0.870000
Right	0.870000
Status	
Received	21 sec
Start Stop Apply	

Abbildung 3.16: Schubkalibrierungs-Ansicht

Command mode	
CHANNELS	
QC horizontal	hold
QC vertical	hold
QC yaw	hold
Gimbal pitch	hold
Gimbal roll	hold
SET COMMAND MODE	
Hold all	
Hold >	
Joystick >	
Absolute angle >	
Target >	
Point >	

Abbildung 3.17: Betriebsmodus-Ansicht

abgerufen werden können, die dann an die GNSS-RTK-Empfänger des Quadropters und der Ziel-Einheit weitergeleitet werden.

Zur Anzeige und zum Wechsel des Betriebsmodus wählt man im Navigationsmenü die „Command mode“-Ansicht aus, siehe Abbildung 3.17. Oben in der Ansicht kann man den aktuellen Betriebsmodus für jeden Steuerungskanal, in diesem Fall „halten“ („hold“), ablesen. Über die blauen Buttons unten wird der Betriebsmodus gewechselt. Die möglichen Kombinationen zeigt Tabelle 3.2. Tippt man beispielsweise auf „Target“, gelangt man zur „Set Target Mode“-Ansicht, siehe Abbildung 3.18. Dort kann man auswählen, welchem Zielobjekt mit welchen Steuerungskanälen gefolgt werden soll. In diesem Fall ist ausgewählt, dass der Quadropters dem Objekt „Target 2“ horizontal folgen und durch Gieren des Quadropters und Nicken des Kamera-Gimbals („QC yaw“ und „Gimbal pitch“) die Kamera auf das Objekt ausrichten soll. Außerdem können die Start-Sollwerte für den Abstand δ (5.120) zum Zielobjekt und den Winkel α (5.121), aus dem das Zielobjekt gesehen werden soll, über zwei Schieberegler vorgegeben werden. Nach der Bestätigung durch „Activate“ richtet sich der Quadropters wie konfiguriert auf das Zielobjekt aus und beginnt mit der Verfolgung. Nach der Aktivierung können über Schieberegler der Abstand und der Winkel zum Zielobjekt jederzeit geändert werden. Diese Funktion wird z. B. in den Flugversuchen D1 und D2 genutzt, vgl. Seite 224ff.

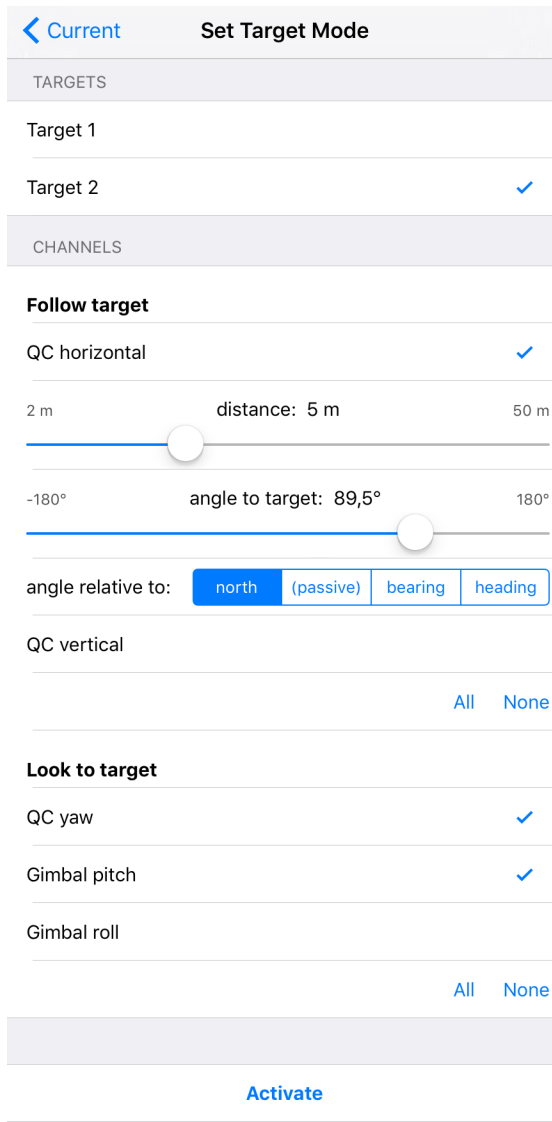


Abbildung 3.18: Aktivierung Objektverfolgung

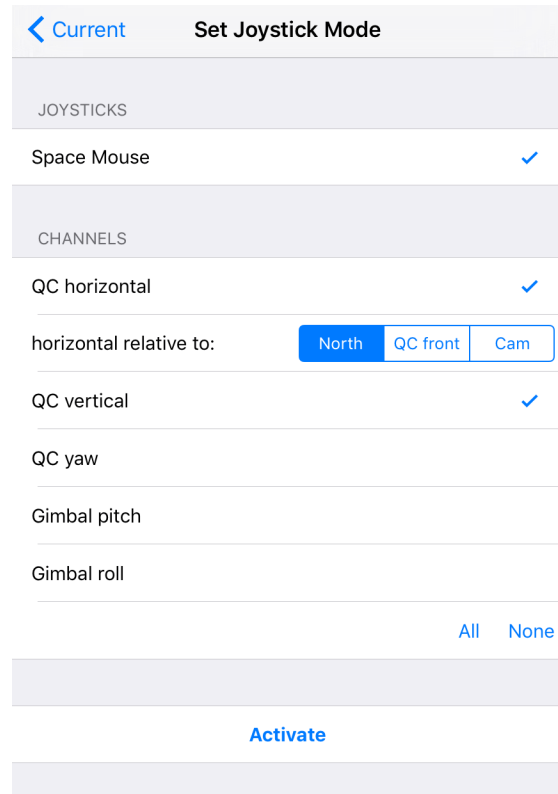


Abbildung 3.19: Aktivierung Joystick

In der „Set Joystick Mode“-Ansicht können die Kanäle ausgewählt werden, die mit dem 6-Achs-Joytick („Space Mouse“) des ControlPads (vgl. Abschnitt 3.3.2) gesteuert werden sollen. In Abbildung 3.19 sind z. B. die zwei Kanäle zur Steuerung der horizontalen und vertikalen Quadrocopterposition ausgewählt. Da „North“ selektiert ist, fliegt der Quadrocopter bei einer Joystickausrückung nach vorne in nördliche Richtung. Alternativ wird die Joystickausrückung bei „QC front“ und „Cam“ relativ zum Quadrocopter bzw. zur Kamera interpretiert, siehe Abschnitt 5.3.3.1. Neben der Quadrocopterposition können zudem der Quadrocopter-Gierwinkel sowie der Nick- und Rollwinkel des Kamera-Gimbals auf den Joystick gelegt werden, wenn diese nicht automatisch dem Zielobjekt nachgeführt

werden sollen. Zur Erprobung des Flugs mit Joystick siehe Abschnitt 6.2.4.

Die drei Winkel können alternativ auch absolut kommandiert werden. Wenn der „Absolutwinkel“-Modus aktiv ist, können die Sollwerte für Gier-, Nick- und Rollwinkel über Schieberegler gesteuert werden, siehe Abbildung 3.20.

In der „Set Static Point Mode“-Ansicht kann ein Wegpunkt ausgewählt werden, zu dem der Quadrokopter horizontal und/oder vertikal fliegen soll, siehe Abbildung 3.21. Außerdem kann ein Point-of-Interest gesetzt werden, auf den der Quadrokopter und das Gimbal die Kamera – durch Gieren, Nicken und/oder Rollen – ausrichten sollen. Die Positionen der Punkte werden zuvor in der App eingegeben.

Das Starten von Trajektorien, die im Simulink-Modell des Quadrokopters hinterlegt sind, erfolgt über die Kommandierung von Events mit zuvor festgelegten IDs. Dies wird z. B. in den Flugversuchen A1 und A2 in Abschnitt 6.2.2 benutzt.

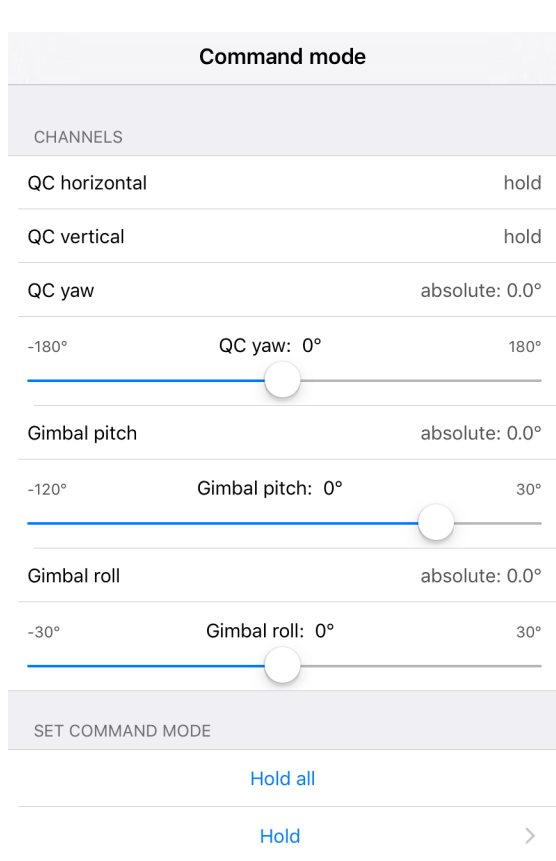


Abbildung 3.20: Schieberegler Absolutwinkel

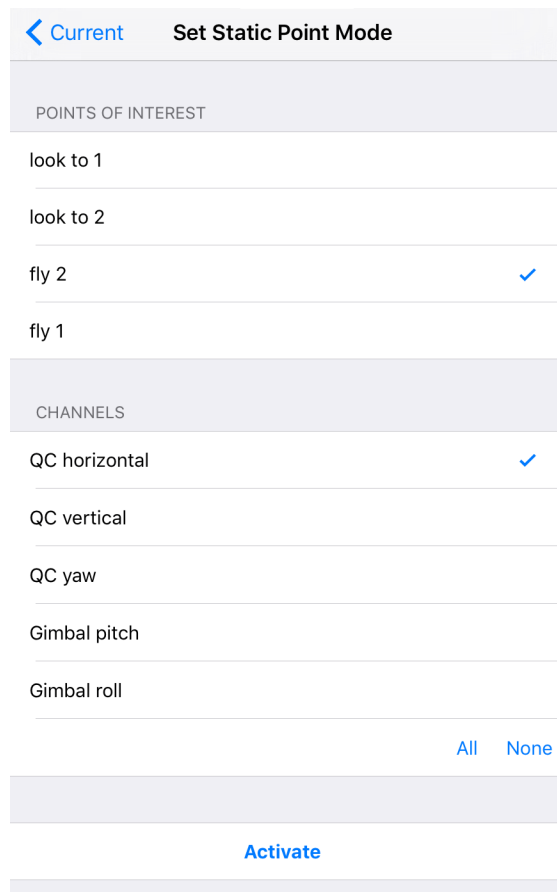


Abbildung 3.21: Aktivierung Modus „statischer Punkt“

Neben der Kommandierung über Schieberegler, Joystick etc. steht in der QC Cam Ctrl App für den Objektverfolgungs-Modus und den „Absolutwinkel“-Modus eine alternative, intuitive Steuerungsart zur Verfügung, die eine virtuelle 3-D-Ansicht nutzt, siehe Abbildung 3.22. Ist der Objektverfolgungs-Modus aktiviert, erscheint ein blauer Zylinder, der das Zielobjekt symbolisiert, auf einer gerasterten Bodenfläche. Der Operator kann dann die Sollwerte für den Winkel und den Abstand zum Zielobjekt sowie die Höhe über dem Zielobjekt durch Touch-Gesten intuitiv steuern. Wird horizontal über den Touchscreen gestrichen, dreht sich die Ansicht um die vertikale Achse des Zielobjekts und der Sollwert für den Winkel α (5.121), aus dem das Zielobjekt gesehen werden soll, wird entsprechend kommandiert. Bei einer vertikalen Streich-Geste verändert sich – soweit aktiviert – die Höhe über dem Zielobjekt in der 3-D-Ansicht und der neue relative Höhensollwert wird an den Quadrokopter gesendet. Der Sollwert für den Abstand δ (5.120) zum Zielobjekt wird durch eine Zoom-Geste verändert: Setzt man zwei Finger auf den Touchscreen und zieht sie auseinander, wird der Abstand verringert. Durch Aufeinanderzuziehen der Finger wird der Abstand entsprechend vergrößert.

Die 3-D-Ansicht kann auch zur Steuerung der Kameraausrichtung unabhängig von einem Zielobjekt genutzt werden („Absolutwinkel“-Modus, vgl. oben). Mit horizontalen und vertikalen Streichgesten wird dabei die Ansicht um die Gier- und Nickachse der virtuellen Kamera, die die 3-D-Ansicht filmt, geschwenkt. Entsprechende Gier- und Nickkommandos werden dann an den Quadrokopter gesendet, sodass sich die reale Kamera des Quadrokopfers genauso wie die virtuelle Kamera ausrichtet⁶. Denkbar wäre, den Kamera-Livestream als Hintergrund in der 3-D-Ansicht einzublenden⁷, sodass man das Live-Kamerabild „anfassen“ und – mit einem kleinen zeitlichen Verzug – auf den gewünschten Bildausschnitt „ziehen“ kann.

Alternativ zu den Streichgesten können die Winkel auch über die Orientierung des ControlPads gesteuert werden. Das iPad schätzt mithilfe von Inertialsensoren und einem Magnetometer die eigene 3-D-Orientierung und stellt sie der QC Cam Ctrl App zur Verfügung. Dadurch können durch Schwenken und Neigen des ControlPads die 3-D-Ansicht intuitiv gedreht und die entsprechenden Winkel kommandiert werden. Auf diese Weise kann die Ausrichtung der Kamera auf dem Quadrokopter so gesteuert werden, als ob man diese – und nicht das ControlPad – in der Hand halten würde. Wenn der Kamera-Livestream in der 3-D-Ansicht eingeblendet würde⁷, wäre es so, als ob man direkt durch die Quadrokopter-Kamera schauen würde und diese direkt – mit einem kleinen zeitlichen Verzug⁸ – schwenken könnte.

⁶Ein Rollen der Kamera kann – wenn gewünscht – durch eine Rotationsgeste mit zwei Fingern kommandiert werden.

⁷Der Kamera-Livestream wird in der realisierten Lösung auf einem separaten Videomonitor angezeigt, siehe Abbildung 3.27.

⁸Im Objektverfolgungs-Modus ist die Verzögerung i. Allg. größer, da der Quadrokopter nach einem Schwenken des ControlPads um die Hochachse neben dem Gieren auch eine neue Position anfliegen muss, um den neuen Azimut-Sollwert (5.121) zu erreichen.

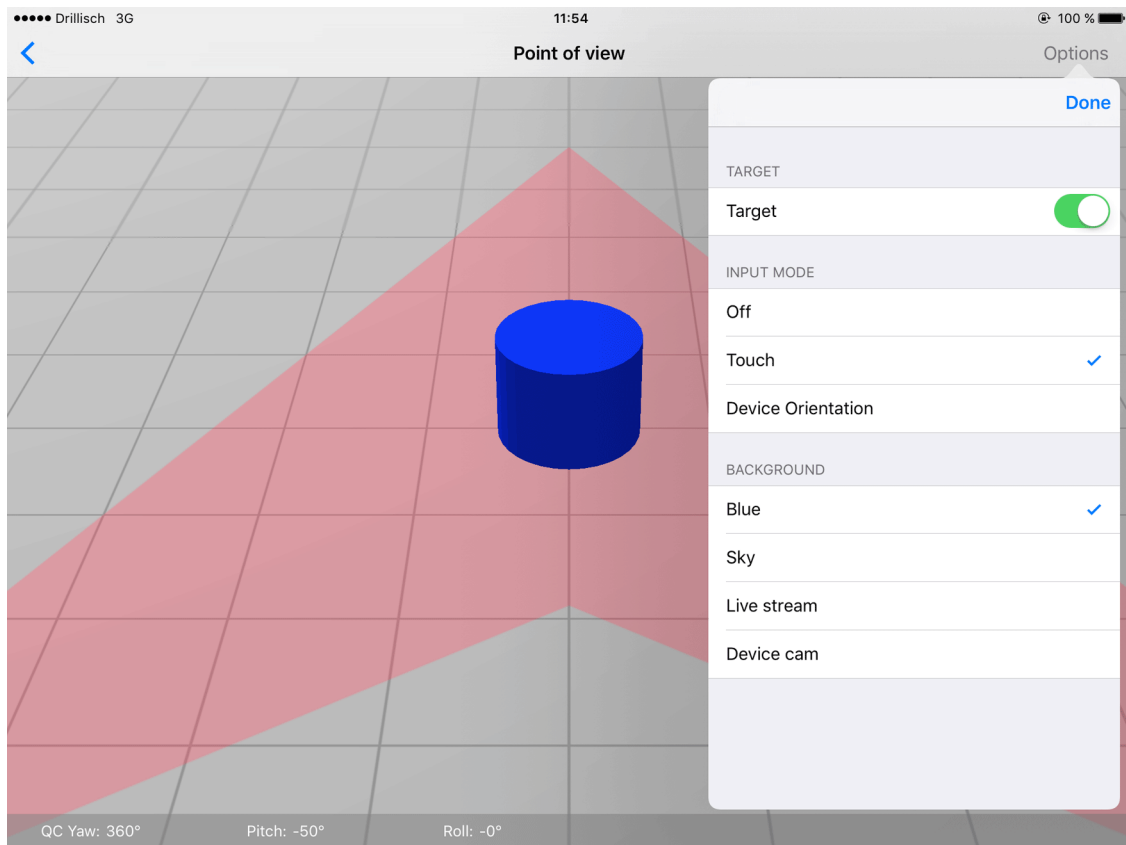


Abbildung 3.22: QC Cam Ctrl: Steuerung über virtuelle 3-D-Ansicht

3.4 Kommunikationsinfrastruktur

Das Gesamtsystem besteht aus vier Einheiten, die miteinander kommunizieren. Neben dem Flugsystem (siehe Abschnitt 3.2.1) und dem ControlPad als Benutzerschnittstelle (siehe Abschnitt 3.3.2) werden eine Bodenstation und eine Ziel-Einheit benötigt, die im Folgenden vorgestellt werden.

Die Bodenstation, die in Abbildung 3.23 zu sehen ist, erlaubt die Einbindung von weiteren Geräten per LAN und WLAN in die Kommunikationsinfrastruktur. Die Ziel-Einheit (siehe Abbildung 3.24), die auf dem zu verfolgenden Zielobjekt installiert wird, sendet die Zielposition über das Kommunikationsnetzwerk an den Quadrokoopter.

Die Bodenstation, die Ziel-Einheit und das ControlPad sind intern ähnlich aufgebaut, siehe Abbildungen 3.25, 3.26 und 3.13. Wie das ControlPad bestehen Bodenstation und Ziel-Einheit aus einem Raspberry Pi (a), einer Versorgungsplatine (b), einer Platine mit Tastern und LEDs (c) und einem XBee-Modul (d). Die Bodenstation verfügt zusätzlich über einen WLAN-Router und einen LAN-Switch (e), über den der Raspberry Pi und bis zu drei weitere Geräte an das IP-Netzwerk angeschlossen werden. Die Versorgungsplatine und die Taster-LED-Platine wurden so entworfen, dass in den drei Komponenten identische Platinen verwendet werden können, die sich nur in der Bestückung unterscheiden. So wie das ControlPad und einige Quadrokoopter-Bauteile (vgl. Seite 69) wurden die Ziel-Einheit und die Front der Bodenstation 3-D-gedruckt. Alle drei Komponenten werden durch einen USB-Akkupack versorgt.

In Abbildung 3.27 ist die Videomonitor-Einheit zu sehen, die das Live-Kamerabild vom Quadrokoopter empfängt und für den Operator anzeigt. Statt auf dem Videomonitor könnte das Kamerabild auch auf dem ControlPad dargestellt werden (vgl. Seite 84).

Der Quadrokoopter kann über sein WLAN-Modul in das Kommunikationsnetzwerk eingebunden werden. Dieser Kommunikationskanal eignet sich vorwiegend für den Indoor-Flug. Für Outdoor-Flüge hat sich WLAN nach diversen Tests mit unterschiedlichen WLAN- Routern als zu unzuverlässig herausgestellt. Für eine Kommandierung des Quadrokoopters durch das ControlPad, eine Live-Statusanzeige auf dem ControlPad und eine kontinuierliche Zielpositionsübertragung an den Quadrokoopter wird jedoch ein zuverlässiger Kommunikationskanal benötigt. Deutlich geeigneter als WLAN haben sich ZigBee-Funkmodule erwiesen. Deshalb werden hier Digi XBee Pro S1 Module verwendet, die auf 2,4 GHz untereinander ein Mesh-Netzwerk aufbauen und eine Reichweite von über 1 km ermöglichen.



Abbildung 3.23: Bodenstation



Abbildung 3.24: Ziel-Einheit

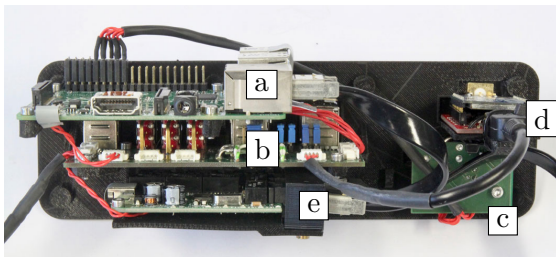


Abbildung 3.25: Bodenstation Innenansicht

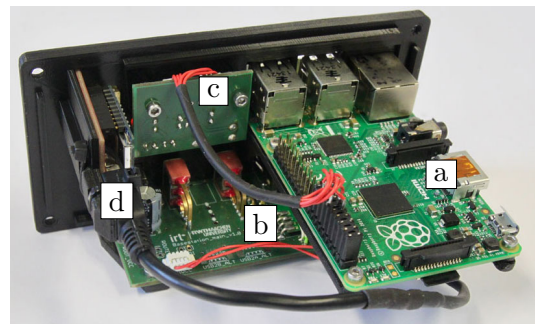


Abbildung 3.26: Ziel-Einheit Innenansicht



Abbildung 3.27: Videomonitor-Einheit

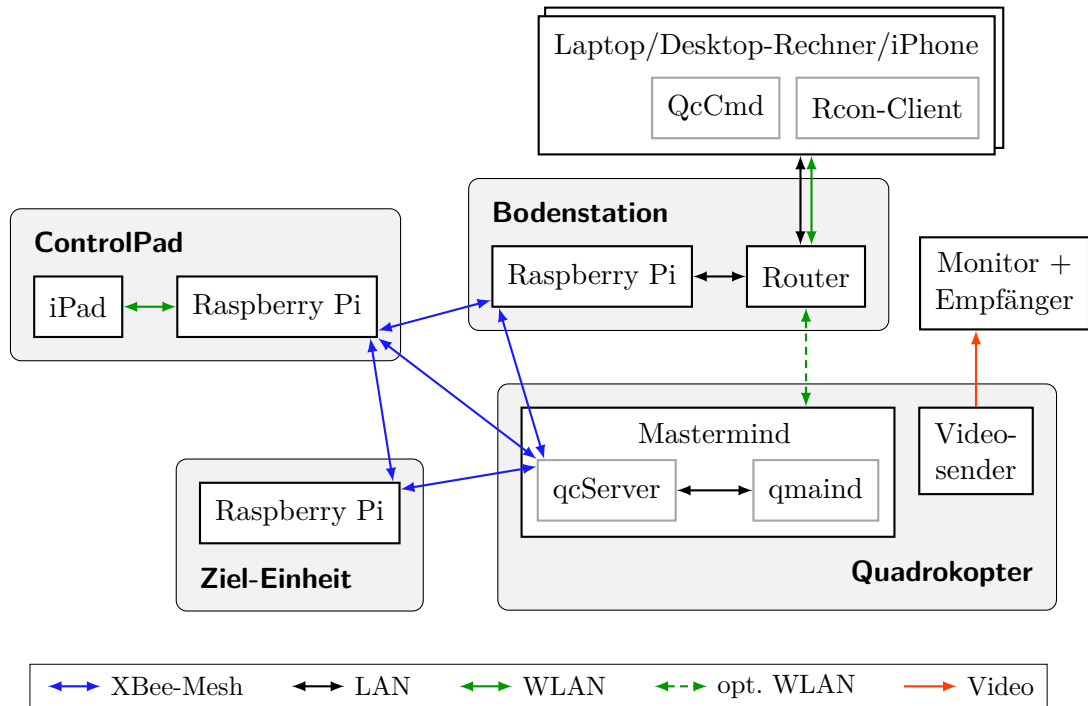


Abbildung 3.28: Kommunikationsarchitektur

Die Architektur des gesamten Kommunikationsnetzwerkes ist in Abbildung 3.28 dargestellt. Das zentrale XBee-Mesh-Netzwerk zwischen Quadrokopter, Ziel-Einheit, ControlPad und Bodenstation ist in Blau zu sehen. Über die Software auf den Raspberry Pis und das qcServer-Programm werden Nachrichten zwischen dem XBee-Mesh-Netzwerk und den IP-Netzwerken in Bodenstation, ControlPad und Mastermind geroutet. Die LAN- und WLAN-Verbindungen dieser IP-Netzwerke sind mit schwarzen und grünen Pfeilen dargestellt. Die Video-Funkübertragung (roter Pfeil) zwischen Videosender und -empfänger erfolgt auf 5,8 GHz.

Für die Kommunikation der Komponenten untereinander wurde ein binäres Nachrichtenformat „StdQCMessage“ mit variablem Dateninhalt definiert, siehe Abbildung 3.29. Jede Nachricht beginnt mit den zwei festen Bytes 0x71 0x63 – dem ASCII-Code für die beiden Buchstaben „qc“ – anhand derer der Parser den (möglichen) Nachrichtenanfang erkennen kann. Dann folgen zwei Bytes, in denen die Länge n des Daten-Feldes als 16-bit-unsigned-Integer im little-endian Format steht. Mit dem darauf folgenden Byte wird der Datentyp des Daten-Feldes festgelegt, z. B. ein bestimmter C-struct Datentyp. Im Daten-Feld sind

Sync „qc“	Länge n	Typ	Daten	CRC
2 Bytes	2 Bytes	1 Byte	n Bytes	2 Bytes

Abbildung 3.29: Nachrichtenformat StdQCMessage

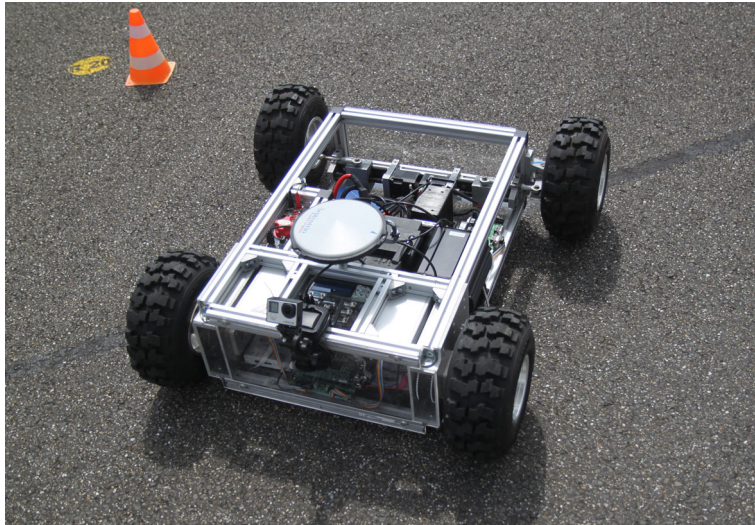


Abbildung 3.30: Testfahrzeug ausgerüstet mit Ziel-Einheit, GNSS-RTK-Empfänger und GNSS-Antenne

dann die n Byte des angegebenen Datentyps enthalten. Die Nachricht schließt mit zwei Bytes ab, die einen 16-bit-CRC-Prüfwert enthalten, anhand dessen Übertragungsfehler erkannt werden können. Außerdem wird durch Check des CRC-Prüfwertes verifiziert, dass die initialen zwei Bytes „qc“ tatsächlich der Nachrichtenbeginn sind.

Auf den drei Raspberry Pis von Bodenstation, Ziel-Einheit und ControlPad läuft die gleiche Software, die für den Einsatz in der jeweiligen Komponente konfigurierbar ist. Diese Software übernimmt in der Bodenstation und im ControlPad das Routing der StdQCMessages zwischen dem XBee-Mesh-Netzwerk und den IP-Netzwerken. So routet die Software auf dem Raspberry Pi des ControlPads zwischen dem per WLAN verbundenen iPad und dem XBee-Mesh. Dadurch kann die QC Cam Ctrl App z. B. Kommando-Nachrichten senden und Status-Nachrichten empfangen. Auf diese Weise werden auch die RTK-Korrekturdaten und die 6-Achs-Joysticks-Kommandos in das Kommunikationsnetzwerk weitergeleitet.

Zur Positionsbestimmung des Zielobjekts ist ein GNSS-RTK-Empfänger per USB an den Raspberry Pi der Ziel-Einheit angeschlossen. Dieser GNSS-RTK-Empfänger ist mit dem auf dem Quadrocopter genutzten vergleichbar, vgl. Abschnitt 4.1.2.1. In Abbildung 3.30 ist das etwa 1 m lange Testfahrzeug [63] zu sehen, auf dem die Ziel-Einheit mit GNSS-RTK-Empfänger und Antenne zur Erprobung der Objektverfolgung (siehe Abschnitt 6.2.5) installiert ist. Da die Ziel-Einheit bezüglich Stromversorgung etc. autark ist, kann sie auch für beliebige andere Zielobjekte genutzt werden. Die Software auf dem Raspberry Pi versendet die GNSS-RTK-Positionen als StdQCMessages an das Netzwerk.

Analog zum ControlPad routet die Software auf dem Raspberry Pi der Bodenstation die StdQCMessages zwischen dem XBee-Mesh und dem IP-Netzwerk des WLAN-Routers in der Bodenstation. Wenn z. B. eine Nachricht per XBee empfangen wird, wird diese

per UDP Broadcast über den WLAN-Router z. B. an einen verbundenen Laptop oder Desktop-Rechner gesendet. In der anderen Richtung werden per UDP empfangene Nachrichten an das XBee-Mesh weitergeleitet. So können u. a. Operator-Kommandos der Qt-Anwendung „QcCmd“, die auf einem Desktop-Rechner der Indoor-Testumgebung⁹ läuft, an den Quadrocopter übermittelt werden. Diese Anwendung wurde für den Einsatz bei Indoor-Flugversuchen geschrieben und bietet einen Teil der Funktionen der iOS-App „QC Cam Ctrl“¹⁰, die bei Outdoor-Flügen eingesetzt wird. Die Qt-Anwendung „QcCmd“ basiert auf der gleichen C++-Codebasis wie die iOS-App.

Bei Indoor-Flugversuchen kann – wenn ControlPad und Ziel-Einheit nicht benötigt werden – anstatt der XBee-Kommunikation die optionale WLAN-Kommunikation zwischen dem Quadrocopter und dem Router der Bodenstation (gestrichelter grüner Pfeil in Abbildung 3.28) genutzt werden.

Diese WLAN-Verbindung wird zudem indoor und outdoor genutzt, um zu Entwicklungszwecken per Rcon Parameter auf dem Quadrocopter zu ändern (vgl. Abbildung 3.10 auf Seite 74). Dazu wird z. B. ein iPhone, auf dem die Rcon iOS-App läuft (vgl. Abbildung 3.11 auf Seite 74), per WLAN mit dem Router verbunden. Alternativ kann ein Python Rcon-Client auf einem Laptop oder auf einem Desktop-Rechner in der Indoor-Testumgebung⁹ verwendet werden, der per LAN oder WLAN am Router angeschlossen ist.

Mit der dargestellten Kommunikationsarchitektur ist durch die Kombination von XBee-Mesh-Netzwerk und IP-Netzwerken eine robuste Kommunikation zwischen den benötigten Komponenten möglich, die bei Bedarf flexibel erweitert werden kann. So könnte ohne Weiteres z. B. ein zweites ControlPad eingebunden werden, damit eine zweite Person einen Teil der Steuerung übernehmen kann. Gleiches gilt für eine zweite Ziel-Einheit, sodass dann zwischen zwei Zielen umgeschaltet werden könnte.

3.5 Testumgebungen

Im Folgenden werden die verschiedenen Testumgebungen vorgestellt, die zur Erprobung des Systems benötigt werden. Dazu gehören zum einen die Simulationsumgebung, in der das System oder Teilsysteme simulativ erprobt werden können, und zum anderen die Indoor- und Outdoor-Testumgebungen, in denen die Flugversuche durchgeführt werden. Zudem wird das Galileo-Testgebiet automotiveGATE präsentiert, das für die Positionsbestimmung mit Galileo-Signalen genutzt wird.

3.5.1 Simulationsumgebung

Die Funktionalsoftware für den Quadrocopter mit Sensorfusion und Regelung wird graphisch in einem Simulink-Modell programmiert (vgl. Softwarearchitektur in Abbildung

⁹ vgl. Abschnitt 3.5.2

¹⁰ siehe Abschnitt 3.3.3

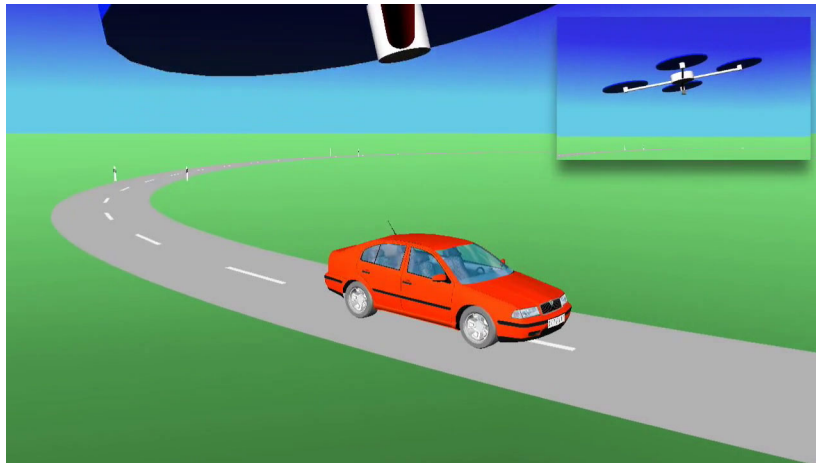


Abbildung 3.31: 3-D-Visualisierung in der Simulationsumgebung

3.9). Dies ist auch für die simulative Erprobung von Vorteil, da dieselben Simulink-Blöcke, aus denen C++-Code für den Quadroko­pter generiert wird, auch in einer Simulation mit Simulink verwendet werden können. So können sowohl das Gesamtsystem als auch Teilsysteme simuliert werden.

Zur Erprobung der flachheitsbasierten Regelung, die in Kapitel 5.2 entwickelt wird, werden z. B. die Simulink-Blöcke der Regelung mit einem Modell der Quadroko­pterdynamik (vgl. Abschnitt 5.1.3) verbunden. Die Ergebnisse einer solchen simulativen Erprobung werden in Abschnitt 5.5 präsentiert.

Die Simulationsumgebung ermöglicht auch eine 3-D-Visualisierung der Objektverfolgung: In Abbildung 3.31 ist z. B. die Sicht der Quadroko­pter-Kamera auf das Zielobjekt und eine Visualisierung der Quadroko­pter-Bewegungen dargestellt.

In der Simulationsumgebung werden auch andere Komponenten wie die Sensorfusion (Kapitel 4), die Trajektoriengenerierung (Abschnitt 5.3) und die Ablaufsteuerung (Abschnitt 5.4) vor dem Einsatz auf dem Quadroko­pter simuliert.

Dadurch, dass in der Simulation und auf dem Quadroko­pter die identische Funktionalsoftware verwendet wird, kann schnell zwischen Simulation und Flugversuch hin und hergewechselt und so die prototypische Software iterativ verbessert werden.

3.5.2 Indoor-Testumgebung

Für kleine Flugversuche steht im Labor eine Indoor-Testumgebung zur Verfügung. In Abschnitt 6.2.3 wird z. B. die Erprobung der MPC-Trajektoriengenerierung im Indoor-Flugversuch dargestellt.

Da im Labor kein GNSS-Empfang für die Sensorfusion möglich ist, kommt zur Positionsbestimmung stattdessen das Infrarot-Trackingsystem (Motion-Capture-System) ART

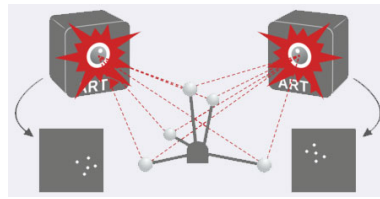


Abbildung 3.32: Funktionsprinzip Infrarot-Trackingsystem (Quelle: [6])

TrackPack [6] zum Einsatz. Die zwei Kamera-Einheiten des Systems senden synchronisierte Infrarotlicht-Blitze aus, sodass die Infrarot-Kameras reflektierende Marker-Kugeln erfassen können, siehe Abbildung 3.32.

Aus den zwei Bildern der Kameras berechnet das Trackingsystem dann die 3-D-Position der Infrarot-Marker im Raum. Dazu wird vorab ein Kalibrierungsprozess durchgeführt, in dem die Position und die Orientierung der Kameras bezüglich eines Raum-Koordinatensystems gelernt wird.

Am Quadrokopter wird ein Tracking-Baum mit fünf Infrarot-Markern verwendet, siehe Abbildung 6.17. Da die Abstände der fünf Marker voneinander bekannt sind, kann neben der 3-D-Position des Tracking-Baums auch die 3-D-Orientierung im Raum berechnet werden. Die Position und Orientierung des Tracking-Baums sendet das Trackingsystem mit einer Updaterate von 60 Hz per UDP über WLAN (vgl. Abbildung 3.28) an den Quadrokopter.

Neben dem Indoor-Einsatz kommt das Infrarot-Trackingsystem auch bei der Outdoor-Erprobung der Sensorfusion zum Einsatz, siehe Abbildung 6.1 in Abschnitt 6.1.1.

3.5.3 Outdoor-Testumgebung

Für Outdoor-Flugversuche sollte eine Testumgebung für den Quadrokopter ausreichend Platz ohne Hindernisse und eine gute GNSS-Abdeckung aufweisen. Zudem sollte die Outdoor-Testumgebung für die Experimente zur Objektverfolgung für das Zielobjekt geeignet sein.

Eine Testumgebung, die diese Voraussetzungen erfüllt, ist das Aldenhoven Testing Center (ATC) der RWTH Aachen University. Die wesentlichen Flugversuche, die in Kapitel 6 dargestellt werden, wurden im ATC durchgeführt. Für weitere Flugversuche wurde außerdem eine Freifläche auf dem Campus der RWTH Aachen University genutzt.

Das ATC verfügt als Automobiltestzentrum über verschiedene Streckenelemente wie Fahrdynamikfläche, Handlingkurs, Brems- und Schlechtwegstrecke, Steigungshügel und eine Ovalbahn mit Steilkurven [3]. 2018 wurde es um Kreuzungen und Straßen für innerstädtische Verkehrsszenarien erweitert. Außerdem kann die angrenzende Film Autobahn genutzt werden. Für die Quadrokopter-Flugversuche wurde vor allem die Fahrdynamikfläche mit 210 m Durchmesser genutzt, siehe Abbildung 3.33.

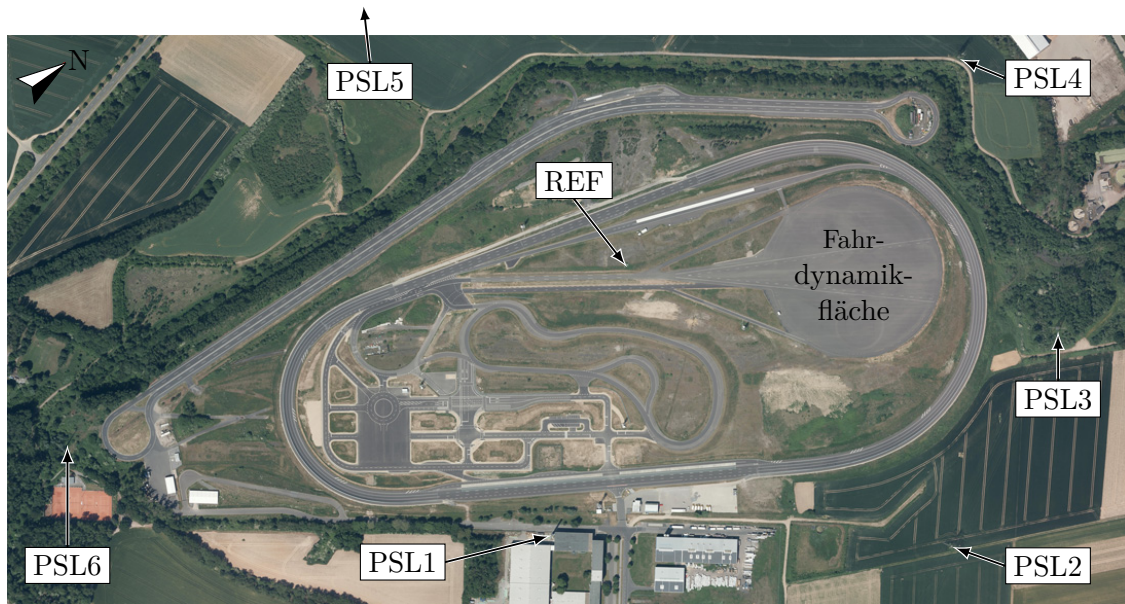


Abbildung 3.33: Luftbild des Aldenhoven Testing Centers (im Jahr 2019) mit den Standorten der Pseudoliten (PSL1 – PSL6) und der Referenzstation (REF) des Galileo-Testgebietes automotiveGATE

3.5.4 Galileo-Testgebiet automotiveGATE

Zum Zeitpunkt der Erprobung des experimentellen GPS- und Galileo-Empfängers (vgl. Abschnitt 4.1.2.2) befand sich die Galileo-Satellitenkonstellation noch im Aufbau (siehe Seite 50). Daher wurde für die Versuche zur Positionsbestimmung mit Galileo-Signalen das Galileo-Testgebiet „automotiveGATE“ genutzt, das in diesem Abschnitt vorgestellt wird.

Das automotiveGATE ist eines von fünf „Galileo Test- und Entwicklungsumgebungen“ (GATEs) in Deutschland [13], deren Aufbau vom Deutschen Zentrum für Luft- und Raumfahrt gefördert wurde, um schon vor der Einsatzbereitschaft der Satelliten die Entwicklung von Galileo-Anwendungen zu ermöglichen. Nach dem GATE im Berchtesgadener Talkessel, das 2008 eröffnet wurde, folgten 2010 das SEA Gate im Forschungshafen Rostock und das aviationGATE auf dem Forschungsflughafen Braunschweig für Galileo-Anwendungen speziell im Schiffs- und Luftverkehr [87]. Das automotiveGATE und das technisch identische railGATE, die beide von der RWTH Aachen University aufgebaut wurden und seit 2014 in Betrieb sind, sind auf die Verkehrsträger Straße und Schiene ausgerichtet. Das railGATE nutzt die Schienenfahrzeug-Testinfrastruktur des Prüf- und Validationcenters der Siemens AG in Wegberg-Wildenrath und macht mit acht bis zu 60 m hohen Sendemasten Galileo-Signale auf den Testgleisen verfügbar [21, 68, 71]. Das automotiveGATE wurde auf dem Gelände des Aldenhoven Testing Centers (siehe Abschnitt 3.5.3) errichtet und ermöglicht auf den verschiedenen Teststreckenelementen die Positionsbestimmung



Abbildung 3.34: Sendemast des Pseudoliten PSL6 im automotiveGATE (im Jahr 2011)

mit Galileo. Im SEA Gate wurde 2017 und im GATE in Berchtesgaden wurde 2019 der Betrieb eingestellt.

Im automotiveGATE werden die Galileo-Signale von sechs sog. Pseudoliten, terrestrischen Pseudo-Satelliten, ausgesendet [16, 17, 20]. Deren Sendeantennen sind in Höhen von 30 bis 50 m auf Hochspannungsmasten (Pseudoliten PSL2–PSL5), einem Schornstein (PSL1) und einem eigens dazu errichteten Mast (PSL6) installiert, siehe Abbildung 3.34. Die Standorte der Pseudoliten sind rund um eine Fläche von ca. 1 km² mit der Referenzstation in der Mitte angeordnet, siehe Abbildungen 3.33 und 6.10.

Die Pseudoliten strahlen Signale auf den Galileo-Trägerfrequenzen E1, E5a oder E5b aus, die konform zur Galileo-Spezifikation, dem „Galileo Open Service Signal in Space Interface Control Document“ [26] sind. Die BOC-Modulation, die PRN-Codesequenzen und das Nachrichtenformat (navigation message encoding, interleaving, synchronization pattern, CRC) entsprechen der spezifizierten Galileo-Signalstruktur [19], vgl. Abschnitt 2.5.

Die Genauigkeit der Positionsbestimmung im automotiveGATE ist mit 2,3 m im Base-Mode und 0,8 m im Assisted Mode, bei dem Korrekturdaten der Referenzstation vom Empfänger genutzt werden, spezifiziert [15, 14]. Tatsächlich erreichbar sind Genauigkeiten von 1,4 m im Base-Mode und 0,5 m im Assisted Mode [87]. Die Betriebsmodi des automotiveGATEs werden von der Masterkontrollstation gesteuert. Sie erlaubt u. a. auch die Definition und das Abspielen von Signalausfallszenarien [21, 68].

Im Testgebiet kann es vorkommen, dass sich ein Empfänger deutlich näher an einem Pseudoliten befindet als an den anderen. Um zu vermeiden, dass in diesem Fall der eine Pseudolit die Signale der anderen überstrahlt, senden die Pseudoliten im Gegensatz zu den Galileo-Satelliten in einem pseudo-zufälligen Pulsmuster, sodass jeder Pseudolit seinen Sendeslot hat.

Im Gegensatz zu den Galileo-Satelliten, die sich auf elliptischen Bahnen um die Erde bewegen (vgl. Abschnitt 2.5.2.1), sind die Positionen der Pseudoliten erdfest. Diese statischen Positionen lassen sich nicht in Form der Ephemeriden-Parameter in den Galileo-Navigationsnachrichten ausdrücken. Da in der Galileo-Spezifikation keine Navigationsnachricht vorgesehen ist, mit der statische Positionen von Pseudoliten übertragen werden können, senden die Pseudoliten Galileo-Dummy-Nachrichten (F/NAV page type 63, I/NAV word type 63) gemäß Galileo-Spezifikation [26]. Ein Empfänger muss also so konfiguriert werden, dass er die Dummy-Nachrichten auch auswertet. Die Positionen der Pseudoliten im automotiveGATE werden in den Empfängern hinterlegt [19].

Im GATE in Berchtesgaden kam mit dem Virtual-Satellite-Mode eine Alternative zum Einsatz [87]: Die Pseudoliten senden Navigationsnachrichten mit Ephemeriden-Parametern für eine virtuelle Satellitenkonstellation aus. Die Pseudolitensignale werden bezüglich Verzögerung, Frequenz und Sendeleistung so angepasst, als kämen die Signale tatsächlich von diesen Satelliten. Dies ist nur für einen einzelnen Empfänger möglich, dessen Position dazu bekannt sein muss. Da im automotiveGATE die meisten Nutzungsszenarien mehrere Nutzer vorsehen, wird der Virtual-Satellite-Mode dort nicht eingesetzt.

Die Galileo-Dummy-Nachrichten enthalten keine Zeitstempel. Daher kann auf Basis der Signale im automotiveGATE keine absolute Zeit bestimmt werden. Für die Positionsbestimmung ist jedoch ausreichend, die Signale der Pseudoliten zeitlich zueinander in Bezug setzen zu können. Dazu wird pro Pseudolit die Zeit seit Beginn der aktuellen PRN-Codesequenz bestimmt, vgl. Abschnitt 2.5.2.4. Für die geringen Entfernungen im automotiveGATE führt dies zu eindeutigen Laufzeitdifferenzen, da sich die PRN-Codesequenzen frühestens alle 4 ms (für die E1-B Signalkomponente) wiederholen, was multipliziert mit der Lichtgeschwindigkeit einer Entfernung von 1200 km entspricht. Für weitere Details siehe [19].

Die Erprobung des experimentellen GPS- und Galileo-Empfängers im automotiveGATE wird in Abschnitt 6.1.2 dargestellt.

4 Sensorfusion

Quadrokopter sind – wie andere Drehflügler – instabile Flugsysteme. Sie weisen im Gegensatz zu den meisten Flugzeugen mit starren Tragflächen keine intrinsisch stabilen Flugzustände auf. Um unbemannte Quadrokopter mit geringer Trägheit z. B. per Fernsteuerung fliegen zu können, verfügen sie meist über eine Drehraten- und Lageregelung. Während die Drehraten direkt gemessen werden können, müssen die Roll- und Nickwinkel für die Lageregelung geschätzt werden. Zur Automatisierung des Quadropters wird zudem eine Schätzung der restlichen Systemzustände wie 3-D-Geschwindigkeit und -Position sowie Gierwinkel benötigt. In diesem Kapitel wird eine Sensorfusion vorgestellt, die aus den onboard zur Verfügung stehenden Sensoren im Outdoor-Flug den Gesamtzustand des Quadropters für die Regelung in Kapitel 5 schätzt.

Ein GNSS-Empfänger erlaubt outdoor die Messung der Position und Geschwindigkeit, jedoch mit einer für die Regelung zu geringen Frequenz. Außerdem bietet er keine Information über die Lage des Quadropters. Die Inertialsensorik misst die Drehraten und Beschleunigungen mit einer hohen Frequenz. Daraus können die Lagewinkel – etwas vereinfacht – durch Integration der Drehraten, die Geschwindigkeit und Position durch einmalige bzw. doppelte Integration der Beschleunigungen berechnet werden. Jedoch werden diese durch Inertialsensor-Bias und -Drifts verfälscht.

GNSS-Empfänger und Inertialsensorik sind in ihren Eigenschaften komplementär, sie ergänzen sich gegenseitig. Um auch in Ruhe die Nordausrichtung bestimmen zu können, wird ein Magnetometer genutzt. Mit diesen Sensoren und einem geeigneten Systemmodell können im Flug alle Systemzustände geschätzt werden. Geeignet für die Sensorfusion von Inertialsensorik-, GNSS-Empfänger- und Magnetometer-Daten bei der zu berücksichtigenden nichtlinearen Systemkinematik ist ein Erweitertes Kalman-Filter, das in Abschnitt 2.3.2 eingeführt wurde.

Im Folgenden werden zunächst die für die Sensorfusion relevanten Sensoren charakterisiert. Dann erfolgt die Darstellung des nichtlinearen Systemmodells und die Auslegung der Propagation und Estimation des Erweiterten Kalman-Filters für die Sensorfusion auf dem Quadropter. Im Abschnitt 4.6 wird die Initialisierung des Kalman-Filters sowie der Ablauf von Propagation und Estimation vorgestellt. Die Erprobung der Sensorfusion ist in Abschnitt 6.1.1 zu finden.

4.1 Sensoren

4.1.1 Inertiale Messeinheit (IMU)

In diesem Abschnitt werden die Eigenschaften und Modelle der Inertialsensoren für die Sensorfusion sowie die Identifikation der Modellparameter dargestellt.

Die zentrale Sensoreinheit für die Sensorfusion bildet eine IMU, eine Inertiale Messeinheit (englisch: Inertial Measurement Unit). Zum Einsatz kommt mit der MicroStrain 3DM-GX3-25-OEM [50] eine „industrial grade“ IMU auf MEMS-Basis, die u. a. über dreiachsige Drehraten- und Beschleunigungssensoren verfügt. Die IMU ist zusammen mit einem USB-Hub und einem 5 V-Spannungswandler auf einer Platine montiert, siehe Abbildung 4.1. Die IMU-Platine wird auf Gummi-Dämpfern im Quadrokofter gelagert, um die auf die IMU wirkenden Vibrationen zu reduzieren (vgl. Abschnitt „Coning und Sculling“ auf Seite 121). Die Kommunikation zwischen IMU und Mastermind-Board erfolgt seriell. Dies hat sich bei der IMU als zuverlässiger erwiesen als die Kommunikation per USB.

Für die Drehratensensoren (Kreisel, Gyroskope) auf Basis von MEMS (mikro-elektromechanisches System, englisch: micro-electro-mechanical system) gibt es verschiedene Technologien. So werden die Drehraten z. B. anhand der Beschleunigungen bestimmt, die aufgrund der Coriolis-Kräfte auf schwingende Probmassen wirken [84]. Die IMU

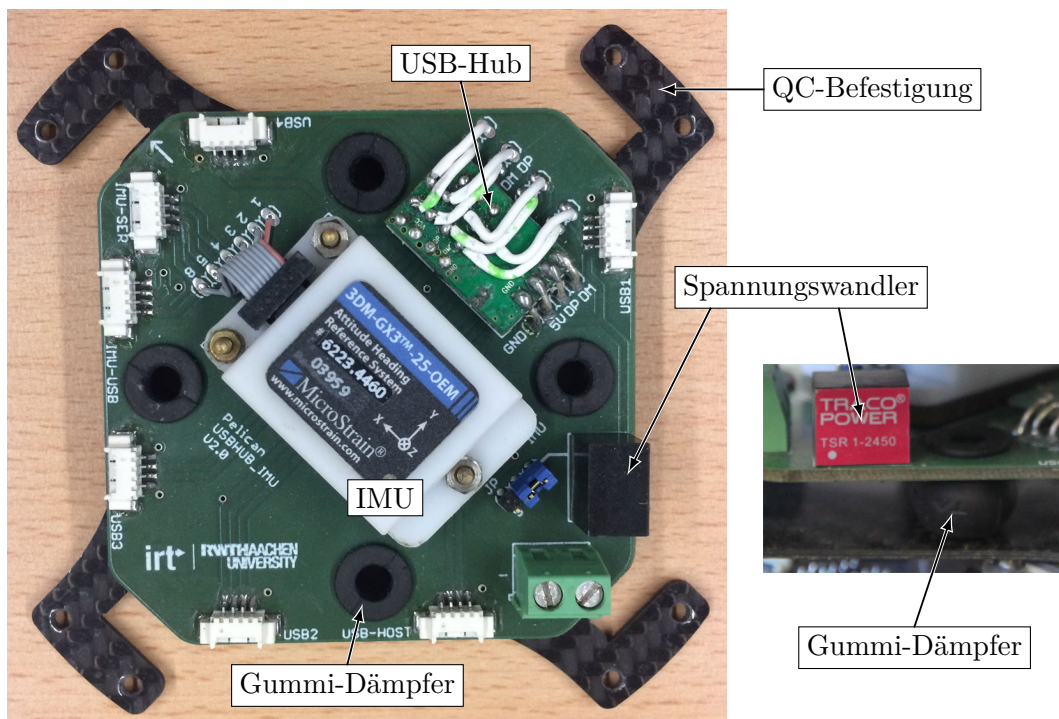


Abbildung 4.1: IMU-Platine auf Gummi-Dämpfern (links: Draufsicht, rechts: Seitenansicht)

ermittelt so die Drehraten $\vec{\omega}_{ib}$ zwischen sich (körperfestes Bezugssystem B) und einem initialen Bezugssystem I . Von der IMU ausgegeben werden die Komponenten von $\vec{\omega}_{ib}$ in ihrem Koordinatensystem C bezüglich ihrer orthogonalen Achsen x, y, z ¹:

$$\vec{\omega}_{ib}^c = (\tilde{\omega}_x \tilde{\omega}_y \tilde{\omega}_z)^T \quad (4.1)$$

Die MEMS-Beschleunigungssensoren messen je nach Technologie z. B. die Schwingungsfrequenzänderung von Quarzen durch die auf die Probemassen wirkenden Kräfte (Vibrating-Beam-Beschleunigungsmesser) oder kapazitiv die Auslenkung von Probemassen (Pendel-Beschleunigungsmesser) [84]. Bei allen Messprinzipien werden damit indirekt sämtliche auf die Probemassen bzw. die IMU wirkenden Kräfte bestimmt. Dazu gehört auch z. B. die Gravitationskraft der Erde, siehe (2.6). Die Wirkung der Gravitation auf die Probemasse ist nicht von einer Beschleunigung der IMU in die entgegengesetzte Richtung zu unterscheiden.

Es ist sinnvoll, die Kräfte als auf die Masse bezogene *spezifische Kräfte* $f = F/m$ (specific forces) auszudrücken, da dann die Masse m nicht bekannt sein muss. Die spezifischen Kräfte haben somit die Einheit von Beschleunigungen. Von der IMU werden die spezifischen Kräfte entlang der orthogonalen Achsen x, y, z ihres Koordinatensystems C ausgegeben:

$$\vec{f}_{ib}^c = (\tilde{f}_x \tilde{f}_y \tilde{f}_z)^T \quad (4.2)$$

Beschleunigungssensoren messen also nicht – wie der Name vermuten ließe – reine Beschleunigungen, sondern auch die Gravitationswirkung der Erde, was bei der Messwert-Verarbeitung berücksichtigt werden muss. Zur Kompensation muss die durch die Gravitation wirkende spezifische Kraft \vec{g} in Betrag und Richtung – im Koordinatensystem C der IMU – bekannt sein (siehe auch Abschnitt 4.6.1). Um die Beschleunigung der IMU \vec{a}_{ib} bezüglich eines initialen Bezugssystems I zu bestimmen, wird die spezifische Gravitationskraft der Erde \vec{g} (2.5) addiert:

$$\vec{a}_{ib} = \vec{f}_{ib} + \vec{g} \quad (4.3)$$

Das positive Vorzeichen von \vec{g} in diesem Zusammenhang wird deutlich, wenn man sich \vec{g} als virtuelle Beschleunigung in die entgegengesetzte Richtung vorstellt, die kompensiert werden muss.

¹Zur Notation siehe auch Seite 14, zu den Koordinatensystemen Seite 23

4.1.1.1 Modell IMU

Inertialsensoren weisen Messfehler auf, die im Allgemeinen nicht vernachlässigt werden können und korrigiert werden müssen. Üblicherweise werden sie wie folgt differenziert:

- Bias
- Messrauschen
- Skalenfaktorfehler
- Nichtlinearitäten
- Sensor-Misalignments

Nach der Vermessung eines Sensors wird eine Ausgleichsgerade durch die gemessene Sensorkurve gelegt, siehe Abbildung 4.2, und so die Abweichung von einem idealen Sensor – bei dem die Messwerte gleich den physikalischen Werten wären – bestimmt. Das Offset der Ausgleichsgeraden vom Nullpunkt ist das Bias, die Abweichung der Steigung von 1 ist der Skalenfaktorfehler. Die Differenzen der gemessenen Sensorkurve von der Ausgleichsgeraden werden als Nichtlinearitäten bezeichnet.

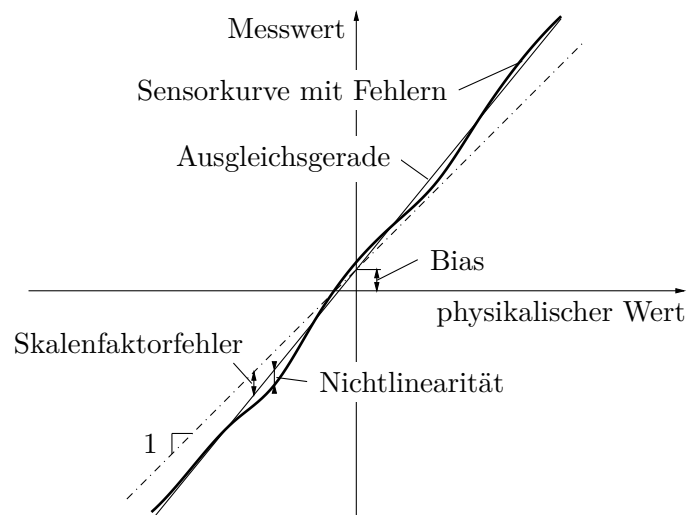


Abbildung 4.2: Typische Inertialsensorkurve mit Fehlern (in Anlehnung an [84])

Mehrere Inertialsensoren sind z. B. orthogonal zueinander angeordnet, um Beschleunigungen und Drehraten in verschiedene Richtungen messen zu können. Die Abweichungen von der nominalen relativen Lage zueinander werden als Sensor-Misalignments oder Nichtorthogonalitäten bezeichnet.

Das Bias von Beschleunigungs- und Drehratensensoren verändert sich mit der Zeit. Der Hauptunterschied zwischen den Güte-Klassen von Inertialsensoren ist die Geschwindigkeit der Bias-Änderung, der sog. Bias-Drift. So haben z. B. Faser- und Ringlaser-Gyros um Größenordnungen kleinere Bias-Drifts als MEMS-Gyros.

Skalenfaktorfehler, Nichtlinearitäten und Sensor-Misalignments weisen im Allgemeinen keinen Drift auf, können aber temperaturabhängig sein. Die vorliegende IMU wurde diesbezüglich durch den Hersteller über den gesamten Temperatur-Betriebsbereich vermessen. Da sie über Temperatursensoren verfügt, kann sie diese Fehler während des Betriebes kompensieren. Daher sollen im Folgenden Skalenfaktorfehler, Nichtlinearitäten und Sensor-Misalignments vernachlässigt werden.

Wenn das IMU-Koordinatensystem C nicht mit dem körperfesten Koordinatensystem B übereinstimmt, müssen die Messwerte $\tilde{\omega}_{ib}^c, \tilde{f}_{ib}^c$ vor der Weiterverarbeitung nach B transformiert werden. Für die rotatorische Transformation gilt:

$$\tilde{\omega}_{ib}^b = \mathbf{C}_c^b \tilde{\omega}_{ib}^c \quad \tilde{f}_{ib}^b = \mathbf{C}_c^b \tilde{f}_{ib}^c \quad (4.4)$$

mit der konstanten Rotationsmatrix \mathbf{C}_c^b , die nach Einbau der IMU bestimmt wird. Der Fehler bei der Ermittlung von \mathbf{C}_c^b soll als IMU-Misalignment – in Abgrenzung zu Sensor-Misalignments – bezeichnet werden.

Der Ursprung von B wird für die Sensorfusion so gewählt, dass er mit dem von C im Punkt U übereinstimmt (siehe Abschnitt 2.1.1 und 4.2), da sonst zur translatorischen Transformation von \tilde{f}_{ib}^c zu \tilde{f}_{ib}^b Drehbeschleunigungen benötigt würden, die aus der zeitlichen Differentiation von $\tilde{\omega}_{ib}^c$ gebildet werden müssten.

Das Messrauschen und der Bias-Drift der Inertialsensoren ist für die vorliegende Anwendung nicht zu vernachlässigen und soll nun modelliert werden: Die gemessene Drehrate $\tilde{\omega}_{ib}^b$ ergibt sich aus der Summe von realer Drehrate $\vec{\omega}_{ib}^b$, Bias \vec{b}_ω und Mess-Rauschen \vec{n}_ω . Die Beschleunigungssensoren werden analog modelliert [84]:

$$\tilde{\omega}_{ib}^b = \vec{\omega}_{ib}^b + \vec{b}_\omega + \vec{n}_\omega \quad (4.5)$$

$$\tilde{f}_{ib}^b = \vec{f}_{ib}^b + \vec{b}_a + \vec{n}_a \quad (4.6)$$

Der Bias-Drift ist gleich dem Rauschen \vec{n}_{b_ω} bzw. \vec{n}_{b_a} [84]:

$$\dot{\vec{b}}_\omega = \vec{n}_{b_\omega} \quad (4.7)$$

$$\dot{\vec{b}}_a = \vec{n}_{b_a} \quad (4.8)$$

Sowohl das Messrauschen \vec{n}_ω und \vec{n}_a als auch die Bias-Drifts \vec{n}_{b_ω} und \vec{n}_{b_a} werden als stochastische weiße Rausch-Prozesse (A.25) mit mittelwertfreien, normalverteilten Wahrscheinlichkeitsdichtefunktionen (A.17) modelliert [84]. Die Identifikation der spektralen Leistungsdichten (A.25) $R_\omega, R_a, R_{b_\omega}, R_{b_a}$ dieser weißen Rausch-Prozesse erfolgt im folgenden Abschnitt 4.1.1.2.

Anzumerken ist, dass die Sensoren bezüglich der gemäß (4.4) in das Koordinatensystem B gedrehten Größen modelliert werden – auch wenn die Modellierung im IMU-Koordinatensystem C intuitiver wäre. Dies macht aber keinen wesentlichen Unterschied, da zu erwarten ist, dass sich die orthogonalen Sensorelemente sehr ähnlich verhalten werden. Die Formulierung in B erleichtert die Nutzung der Modelle im Kalman-Filter.

Die Sendefrequenz, mit der die IMU Drehraten und spezifische Kräfte sendet, liegt oft unter der Frequenz, mit der die IMU die Daten intern verarbeitet. Wenn der Frequenzbereich der auftretenden Dynamik deutlich unter der Sendefrequenz liegt, muss das nicht weiter betrachtet werden. Wenn jedoch Schwingungen im Bereich der Sendefrequenz auftreten, kann es zu Aliasing-Effekten kommen, die als Coning und Sculling bezeichnet werden, siehe Seite 121. Um das zu vermeiden, kann die Sendefrequenz erhöht werden, was ggf. für die Kommunikation technisch problematisch sein kann. Alternativ können die Sensorsignale zwischen zwei Sendezeitpunkten IMU-intern mit der hohen Frequenz aufintegriert und dann die Zahlenwerte der Integrale als „Delta-Winkel“ $\Delta\vec{\sigma}$ und „Delta-Geschwindigkeiten“ $\Delta\vec{v}$ (inkl. Gravitationsanteil) ausgegeben werden:

$$\Delta\tilde{\sigma}_{ib,k}^c = \int_{t_{k-1}}^{t_k} \tilde{\omega}_{ib}^c dt \quad (4.9)$$

$$\Delta\tilde{v}_{ib,k}^c = \int_{t_{k-1}}^{t_k} \tilde{f}_{ib}^c dt \quad (4.10)$$

Diese Alternative wird hier gewählt, um den Einflüssen von Coning und Sculling Motion entgegenzuwirken, vgl. Seite 121.

Die Größen werden dann in das Koordinatensystem B gedreht:

$$\Delta\tilde{\sigma}_{ib,k}^b = \mathbf{C}_c^b \Delta\tilde{\sigma}_{ib,k}^c \quad \Delta\tilde{v}_{ib,k}^b = \mathbf{C}_c^b \Delta\tilde{v}_{ib,k}^c \quad (4.11)$$

Die Nutzung von $\tilde{\sigma}_{ib,k}^b$ und $\tilde{v}_{ib,k}^b$ in der Sensorfusion ist möglich, da die Sensorsignale in der Propagation integriert verarbeitet werden (siehe Abschnitt 4.3).

4.1.1.2 Identifikation IMU

Eine Möglichkeit, die stochastischen Prozesse von Inertialsensoren zu identifizieren, ist die Allan-Varianz [86, 40]. Die Methode wurde ursprünglich von David W. Allan eingeführt, um die Frequenzstabilität von Uhren zu beschreiben [5].

Hier soll die Allan-Varianz zur Bestimmung der stochastischen Sensorfehler \vec{n}_ω und \vec{n}_{b_ω} im Drehratensensormodell (4.5), (4.7) sowie \vec{n}_a und \vec{n}_{b_a} im Beschleunigungssensormodell (4.6), (4.8) verwendet werden.

Allan-Varianz Die Allan-Varianz $\sigma_y^2(\tau)$ eines Signals y ist eine Funktion der Länge τ eines Mittelungs-Zeitintervalls. Zur Berechnung des Funktionswertes für ein τ werden die M Messwerte \tilde{y}_k des Signals mit der Abtastzeit T in n Intervalle der Länge τ unterteilt:

$$\tau = \frac{MT}{n} \quad (4.12)$$

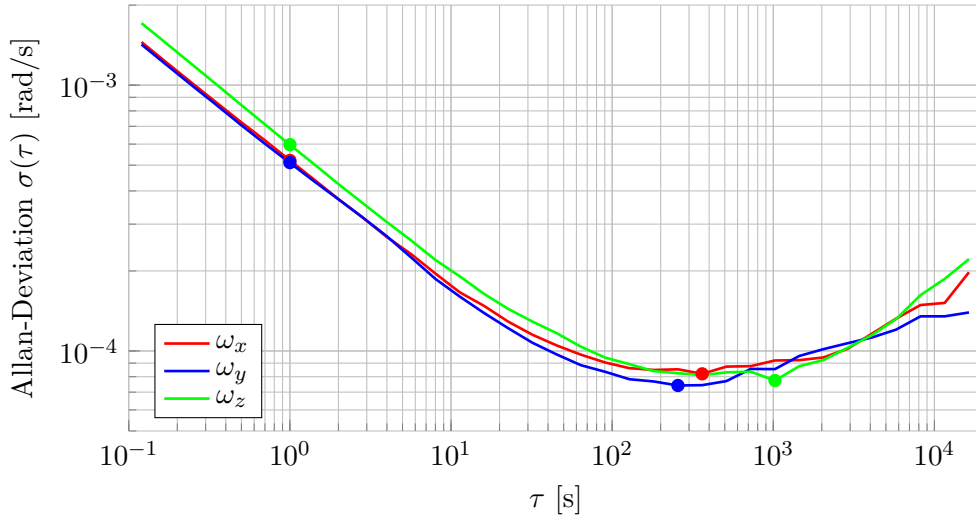


Abbildung 4.3: Allan-Deviation Drehratensensoren

Für jedes Intervall i wird das arithmetische Mittel $\bar{y}(\tau)_i$ der Messwerte im Intervall berechnet:

$$\bar{y}(\tau)_i = \frac{1}{M_\tau} \sum_{j=1}^{M_\tau} \tilde{y}_{(i-1)M_\tau+j} \quad \text{mit} \quad M_\tau = \frac{\tau}{T} = \frac{M}{n} \quad \forall i \in 1, \dots, n \quad (4.13)$$

Die Allan-Varianz $\sigma_y^2(\tau)$ ist definiert als die halbe Varianz der Mittelwerte $\bar{y}(\tau)_i$, jeweils bezogen auf den Mittelwert des Nachbarintervalls [86, 40]:

$$\sigma_y^2(\tau) = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} (\bar{y}(\tau)_{i+1} - \bar{y}(\tau)_i)^2 \quad (4.14)$$

Die Wurzel der Allan-Varianz wird als Allan-Deviation bezeichnet:

$$\sigma_y(\tau) = \sqrt{\sigma_y^2(\tau)} \quad (4.15)$$

Um für die vorliegende IMU die Allan-Deviationen der Sensorsignale zu bestimmen, wurden die von der IMU in Ruhe über einen Zeitraum von 48 Stunden ausgegebenen Delta-Winkel $\Delta\tilde{\sigma}_{ib,k}^c$ (4.9) und Delta-Geschwindigkeiten $\Delta\tilde{v}_{ib,k}^c$ (4.10) aufgezeichnet. In Abbildung 4.3 und 4.4 sind die aus diesen Messdaten berechneten Allan-Deviationen für verschiedene τ dargestellt. Zur Erhöhung der Anschaulichkeit sind die Delta-Winkel und Delta-Geschwindigkeiten in Drehraten und spezifische Kräfte umgerechnet.

Unterschiedliche stochastische Prozesse führen zu unterschiedlichen Steigungen der doppelt-logarithmischen Allan-Deviationen-Kurve in unterschiedlichen Bereichen von τ . Die Kennwerte dieser Prozesse können daher von der Kurve abgelesen werden. Für Inertialsensoren im Allgemeinen zu unterscheidende Prozesse sind: Quantisierungs-Rauschen,

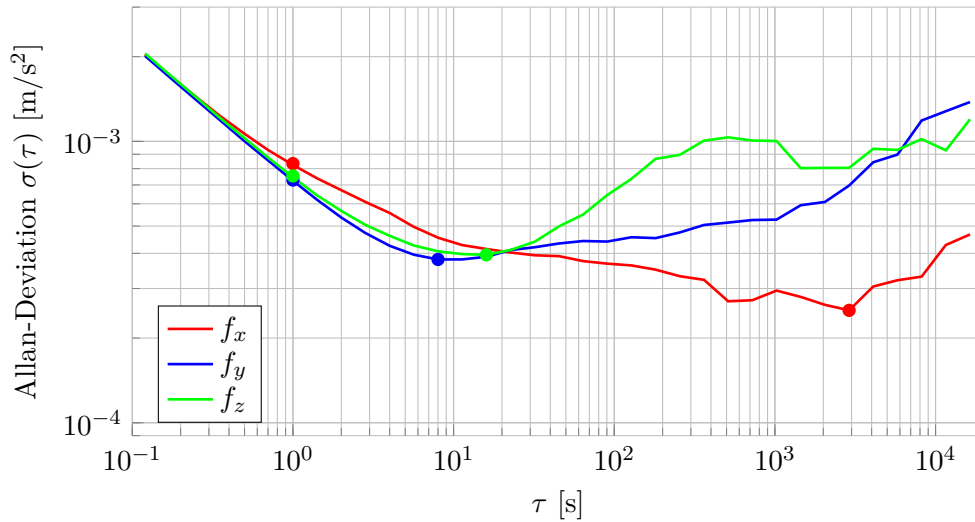


Abbildung 4.4: Allan-Deviation Beschleunigungssensoren

Winkel- bzw. Geschwindigkeits-Random-Walk, korreliertes Rauschen, sinusförmiges Rauschen, Bias-Instabilität, Drehraten- bzw. Beschleunigungs-Random-Walk und Drehraten- bzw. Beschleunigungs-Rampe. Eine ausführliche Darstellung ist in [40] zu finden.

Der Winkel- bzw. Geschwindigkeits-Random-Walk und die Bias-Instabilität von MEMS-Drehraten- und Beschleunigungssensoren liegen in einem Dynamikbereich, der sie für die vorliegende Anwendung zur Auslegung des Kalman-Filters relevant machen [86]. Daher sollen sie im Folgenden betrachtet werden.

Winkel- und Geschwindigkeits-Random-Walk Kontinuierliches weißes Rauschen $y(t)$ hat eine konstante spektrale Leistungsdichte von $S_y(j\omega) = R_y$, siehe (A.25). Integriert man dieses weiße Rauschen, erhält man einen Random-Walk-Prozess (A.33), dessen Standardabweichung durch den Random-Walk-Koeffizienten N_y beschrieben wird. Für ihn gilt gemäß (A.35):

$$N_y = \sqrt{R_y} \quad (4.16)$$

Die Allan-Deviation für weißes Rauschen ist gemäß [40]:

$$\sigma_y(\tau) = N_y \tau^{-0,5} \quad (4.17)$$

In doppelt-logarithmischer Darstellung ist $\sigma_y(\tau)$ also eine Gerade mit einer Steigung von $-0,5$. Der Random-Walk-Koeffizient N_y kann bei $\tau = 1$ von der Kurve abgelesen werden.

Die Steigung von $-0,5$ im Bereich um $\tau = 1$ ist in den Abbildungen 4.3 und 4.4 offensichtlich. Lediglich der Beschleunigungssensor in x -Richtung weist eine leicht flachere Steigung auf. Die bei $\tau = 1$ abgelesenen Random-Walk-Koeffizienten N_ω und N_a des

Winkel-Random-Walks der Drehratensensoren und des Geschwindigkeits-Random-Walks der Beschleunigungssensoren sind in Tabelle 4.1 dargestellt.

Achse	N_ω	N_a
x	$5,21 \cdot 10^{-4} \text{ rad}/\sqrt{\text{s}}$	$8,34 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}}$
y	$5,11 \cdot 10^{-4} \text{ rad}/\sqrt{\text{s}}$	$7,29 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}}$
z	$5,97 \cdot 10^{-4} \text{ rad}/\sqrt{\text{s}}$	$7,54 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}}$

Tabelle 4.1: Random-Walk-Koeffizienten der Drehraten- und Beschleunigungssensoren

Die quadrierten Random-Walk-Koeffizienten N_ω^2 , N_a^2 sind gemäß (4.16) die spektralen Leistungsdichten R_ω , R_a des weißen Rauschens auf den Drehraten bzw. Beschleunigungen und damit die spektralen Leistungsdichten des Messrauschens \vec{n}_ω , \vec{n}_a in den Sensormodellen (4.5) und (4.6). Als Parameter für die Sensorfusion werden die Zahlenwerte in Tabelle 4.2 für alle Sensorachsen verwendet.

Achse	$\sqrt{R_\omega} = N_\omega$	$\sqrt{R_a} = N_a$
x, y, z	$6,0 \cdot 10^{-4} \text{ rad}/\sqrt{\text{s}} = 2,1^\circ/\sqrt{\text{h}}$	$8,4 \cdot 10^{-4} \text{ m/s}/\sqrt{\text{s}} = 5,0 \text{ cm/s}/\sqrt{\text{h}}$

Tabelle 4.2: Spektrale Leistungsdichten R_ω , R_a des Messrauschens der Drehraten- und Beschleunigungssensoren

Diese Werte passen gut mit den im Datenblatt der IMU [50] angegebenen Werten zusammen: Dort sind umgerechnet $1,8^\circ/\sqrt{\text{h}}$ und $4,7 \text{ cm/s}/\sqrt{\text{h}}$ angegeben.

An den Random-Walk-Koeffizienten lässt sich die Größenordnung des zu erwartenden Winkel- und Geschwindigkeits-Random-Walks ablesen. Gemäß (A.34)f. gilt für die Standardabweichung $\sigma_y(t)$ und den Erwartungswert μ_y des Random-Walks als integriertes weißes Rauschen:

$$\sigma_y(t) = N_y \sqrt{t} \quad \mu_y = 0 \quad (4.18)$$

So sind durch das weiße Rauschen auf den Drehraten z. B. innerhalb einer Stunde Winkelfehler mit Standardabweichungen von 2° zu erwarten. Für die Geschwindigkeitsfehler ergeben sich im gleichen Zeitintervall Standardabweichungen von 5 cm/s . Die mittleren Winkelfehler und die mittleren Geschwindigkeitsfehler sind jeweils null.

Bias-Instabilität Die Bias-Instabilität der Drehraten- und Beschleunigungssensoren soll hier auch als Random-Walk der Bias \vec{b}_ω , \vec{b}_a modelliert werden [86]: Der Bias-Drift ist gleich dem weißen Rauschen mit den spektralen Leistungsdichten R_{b_ω} und R_{b_a} .

Die Bias-Instabilität tritt als flacher Bereich um das Minimum der Allan-Deviations-Kurve auf. Der Zahlenwert des Minimums der Allan-Deviations-Kurve entspricht $0,664 B$, mit B als Bias-Instabilitäts-Koeffizient [40]. Rechts vom Minimum steigt für größere τ

die Kurve wieder an. Der Grund dafür sind Prozesse wie der Drehraten- bzw. Beschleunigungs-Random-Walk und die Drehraten- bzw. Beschleunigungs-Rampe [40].

Die Drehratensensoren in Abbildung 4.3 weisen ein ausgeprägtes Minimum auf. Auch bei den Beschleunigungssensoren in y - und z -Richtung sind in Abbildung 4.4 die Minima gut zu erkennen. Der x -Beschleunigungssensor weicht in seiner Charakteristik – auch bei wiederholter Messung – deutlich ab. Hier scheinen noch weitere Effekte zu wirken. Daher ist seine so ermittelte Bias-Instabilität zu verwerfen. Bei den Drehratensensoren liegt das Minimum im Bereich von ca. $\tau_m = 250$ bis 1000 sec. Das Minimum der Beschleunigungssensoren in y - und z -Richtung tritt bei einem deutlich kleinerem τ_m um die 10 sec auf. Die abgelesenen Zahlenwerte für die Bias-Instabilitäts-Koeffizienten und die zugehörigen τ_m sind in Tabelle 4.3 zu finden.

Achse	B_ω	B_a
x	$1,24 \cdot 10^{-4}$ rad/s $\tau_m = 362$ s	$(3,78 \cdot 10^{-4}$ m/s ²) $\tau_m = 2896$ s
y	$1,12 \cdot 10^{-4}$ rad/s $\tau_m = 256$ s	$5,73 \cdot 10^{-4}$ m/s ² $\tau_m = 8$ s
z	$1,17 \cdot 10^{-4}$ rad/s $\tau_m = 1024$ s	$5,95 \cdot 10^{-4}$ m/s ² $\tau_m = 16$ s

Tabelle 4.3: Bias-Instabilitäts-Koeffizienten der Drehraten- und Beschleunigungssensoren

Im Datenblatt der IMU [50] sind für die Bias-Instabilitäten der Drehraten- und Beschleunigungssensoren umgerechnet $0,87 \cdot 10^{-4}$ rad/s und $3,92 \cdot 10^{-4}$ m/s² angegeben. Diese Werte sind leicht niedriger als die gemessenen.

Aus dem Bias-Instabilitäts-Koeffizienten B_y kann der Random-Walk-Koeffizient N_{b_y} für den Random-Walk auf dem Bias b_y berechnet werden als [86]:

$$N_{b_y} = \frac{B_y}{\sqrt{\tau_m}} \quad (4.19)$$

Für die Drehraten- und Beschleunigungssensoren ergeben sich damit die Bias-Random-Walk-Koeffizienten in Tabelle 4.4.

Achse	N_{b_ω}	N_{b_a}
x	$6,49 \cdot 10^{-6}$ rad/s/ \sqrt{s}	$(0,07 \cdot 10^{-6}$ m/s ² / \sqrt{s})
y	$6,98 \cdot 10^{-6}$ rad/s/ \sqrt{s}	$2,03 \cdot 10^{-6}$ m/s ² / \sqrt{s}
z	$3,64 \cdot 10^{-6}$ rad/s/ \sqrt{s}	$1,49 \cdot 10^{-6}$ m/s ² / \sqrt{s}

Tabelle 4.4: Bias-Random-Walk-Koeffizienten der Drehraten- und Beschleunigungssensoren

\vec{n}_{b_ω} und \vec{n}_{b_a} in den Sensormodellen (4.7) und (4.8) sind die kontinuierlichen weißen Rauschprozesse, die zu den Bias-Random-Walks führen. Die spektralen Leistungsdichten

R_{b_ω} und R_{b_a} dieser Rauschprozesse sind gemäß (4.16) gleich den quadrierten Random-Walk-Koeffizienten N_{b_ω} , N_{b_a} .

Für die Sensorfusion werden die spektralen Leistungsdichten in Tabelle 4.5 verwendet. Sie sind höher als die identifizierten in Tabelle 4.4, um etwaige weitere Fehlereinflüsse zu berücksichtigen.

Achse	$\sqrt{R_{b_\omega}} = N_{b_\omega}$	$\sqrt{R_{b_a}} = N_{b_a}$
x, y, z	$8,73 \cdot 10^{-6} \text{ rad/s}/\sqrt{s} = 0,03 \text{ }^\circ/\text{s}/\sqrt{\text{h}}$	$4,0 \cdot 10^{-5} \text{ m/s}^2/\sqrt{s} = 2,4 \cdot 10^{-3} \text{ m/s}^2/\sqrt{\text{h}}$

Tabelle 4.5: Spektrale Leistungsdichten R_{b_ω} , R_{b_a} der Bias-Drifts der Drehraten- und Beschleunigungssensoren

4.1.2 GNSS-Empfänger

Auf dem Quadrokopter kommen verschiedene GNSS-Empfänger zum Einsatz. Für die Automatisierung des Quadrokopters zur Objektverfolgung wird aufgrund der benötigten hohen Positionsgenauigkeit ein GNSS-RTK-Empfänger als Eingang in die Sensorfusion genutzt. Da das zu verfolgende Zielobjekt ebenfalls einen GNSS-RTK-Empfänger verwendet (vgl. Abschnitt 3.4), kann sich der Quadrokopter präzise relativ zum Zielobjekt positionieren und die Kamera darauf ausrichten (vgl. Abschnitt 5.3.1). Außerdem wird ein experimenteller GPS- und Galileo-Empfänger im Galileo-Testgebiet automotiveGATE erprobt, siehe Abschnitt 4.1.2.2.

Der GNSS-Empfänger des Pelican, ein u-blox LEA-6S Modul mit integrierter Patchantenne, ist ein Einfrequenz-GNSS-Empfänger mit einer spezifizierten Genauigkeit von 2,5 m CEP² (2,0 m CEP mit SBAS). Er wird für die Anwendung nicht eingesetzt.

4.1.2.1 GNSS-RTK-Empfänger

Als GNSS-RTK-Empfänger wird ein Septentrio AsterRx3 [75] verwendet. Dieser GNSS-Mehrfrequenz-Empfänger kann die Signale aller vier globalen Satellitennavigationssysteme GPS, Galileo, GLONASS und BeiDou (vgl. Abschnitt 2.5.1) verarbeiten. Mit GNSS-Korrekturdaten liegt seine spezifizierte Positionsgenauigkeit im RTK-Fix-Modus (vgl. Seite 62) bei 0,6 cm horizontal und 1 cm vertikal [75].

Um den AsterRx3 auf dem Quadrokopter nutzen zu können, wird die OEM-Variante ohne Kunststoffgehäuse verwendet. Diese wird mit einer Adapterplatine (Abbildung 3.5) im Quadrokopter installiert, siehe Abbildung 4.5. Als Antenne für den GNSS-RTK-Empfänger kommt die aktive Mehrfrequenz-Patchantenne Antcom G5Ant1AS4 zum Einsatz. Sie

²Streukreisradius (englisch: circular error probable): 50% der Messwerte liegen in einem Kreis mit diesem Radius

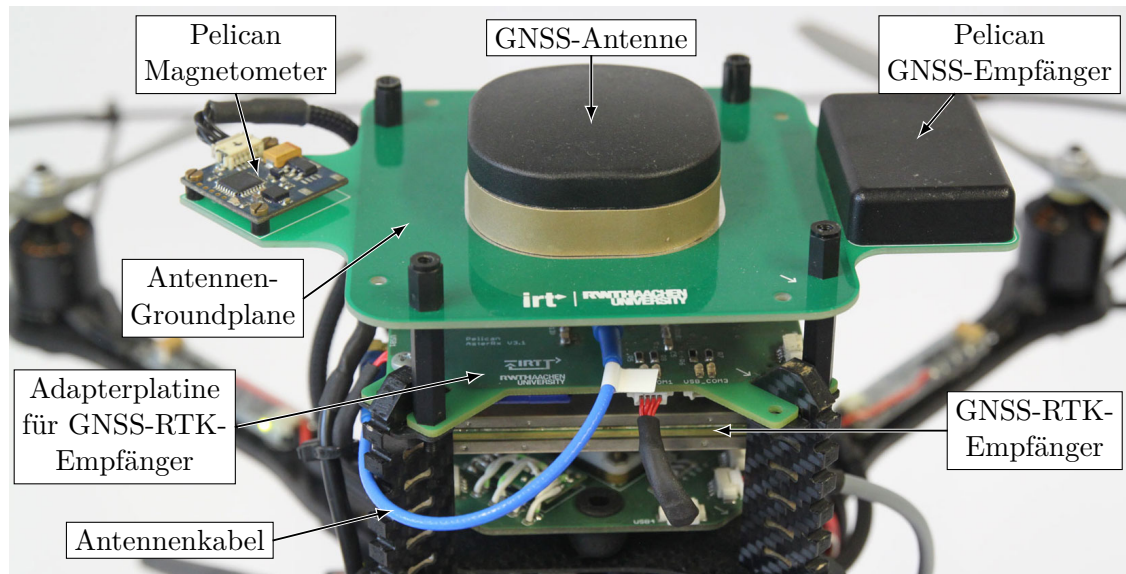


Abbildung 4.5: GNSS-RTK-Empfänger mit Adapterplatine und GNSS-Antenne sowie Pelican Magnetometer und GNSS-Empfänger montiert im Quadrokoopter

ist auf einer Platine installiert, die unterhalb der Antenne eine 10x10 cm große Groundplane (Massefläche) enthält. Diese Groundplane verbessert die Empfangseigenschaften der Patchantenne und schirmt sie gegen Einstrahlungen von unten ab. Neben der Patchantenne sind der GNSS-Empfänger und das Magnetometer des Pelican montiert, siehe Abbildung 4.5.

Erprobung GNSS-RTK-Empfänger Nach der Integration in den Quadrokoopter zeigte sich, dass der Empfänger zunächst nicht in der Lage war, eine Positionslösung im RTK-Fix-Modus zu berechnen, da die Störungen onboard zu stark waren. Diese Störungen konnten auf die Emissionen des nicht geschirmten Mastermind-Boards zurückgeführt werden. Die Taktrate des i7 Prozessors liegt bei 1,7 GHz. Es ist anzunehmen, dass dieser auch in die GNSS-Frequenzbänder ausstrahlt. Nach einigen Untersuchungen wurden zwei Maßnahmen getroffen, um das Problem in den Griff zu bekommen. Das Mastermind-Board ist jetzt nach oben hin mit einer Kupfer-Folie abgeschirmt, um die elektromagnetische Strahlung in Richtung des GNSS-Empfängers und der GNSS-Antenne zu reduzieren. Eine komplette elektromagnetische Schirmung des Boards konnte aufgrund der Vielzahl der Anschlüsse nicht umgesetzt werden. Als weitere Maßnahme wurden Tiefpassfilter in die Stromversorgungen von GNSS-RTK-Empfänger und Mastermind-Board eingesetzt, um die leitungsgebundenen Störungen abzuschirmen. Mit diesen zwei Maßnahmen erreicht der GNSS-RTK-Empfänger die gewünschte Performance im RTK-Fix-Modus.

Mess-Kovarianzmatrizen Der GNSS-RTK-Empfänger schätzt die Güte seiner Positions- und Geschwindigkeitsmessungen und gibt diese als Kovarianzmatrizen aus. Die

Schätzung basiert auf einem stochastischen Fehlermodell für die einzelnen Pseudorange-Messungen, in das u. a. das Trägerauschverhältnis, die Satellitenelevation und das Ionosphären-Modell eingehen [76], vgl. Abschnitt 2.5.2.

Für das Kalman-Filter werden Mess-Kovarianzmatrizen \mathbf{R}_k (2.31) benötigt, die die stochastischen Fehler der Messgrößen beschreiben. Dazu können die vom GNSS-Empfänger gelieferten Kovarianzmatrizen verwendet werden. Die Kovarianzmatrix für seine Positionen hat die Form:

$$\mathbf{R}_{\tilde{\mathbf{p}},k} = \begin{pmatrix} \sigma_{p,n,k}^2 & \sigma_{p,ne,k} & \sigma_{p,nd,k} \\ \sigma_{p,en,k} & \sigma_{p,e,k}^2 & \sigma_{p,ed,k} \\ \sigma_{p,dn,k} & \sigma_{p,de,k} & \sigma_{p,d,k}^2 \end{pmatrix} \quad (4.20)$$

Auf der Hauptdiagonalen stehen die geschätzten Varianzen $\sigma_{p,i,k}^2$ der Positionen in metrischen, kartesischen Koordinaten (Norden, Osten, unten) und auf den Nebendiagonalen die zugehörigen Kovarianzen $\sigma_{p,ij,k}$ (A.8). Die Wurzeln der Varianzen, die Standardabweichungen $\sigma_{p,n,k}$, $\sigma_{p,e,k}$, $\sigma_{p,d,k}$, liegen im RTK-Fix-Modus (siehe Seite 62) im Bereich von 1 bis 2 cm. Die Wurzeln der Beträge der Kovarianzen $|\sigma_{p,ij,k}|^{1/2}$ nehmen im RTK-Fix-Modus Werte in der Größenordnung von 1 bis 10 mm an.

Die Kovarianzmatrix mit den geschätzten Varianzen und Kovarianzen der Nord-, Ost- und Vertikal-Geschwindigkeiten hat die gleiche Form:

$$\mathbf{R}_{\tilde{\mathbf{v}},k} = \begin{pmatrix} \sigma_{v,n,k}^2 & \sigma_{v,ne,k} & \sigma_{v,nd,k} \\ \sigma_{v,en,k} & \sigma_{v,e,k}^2 & \sigma_{v,ed,k} \\ \sigma_{v,dn,k} & \sigma_{v,de,k} & \sigma_{v,d,k}^2 \end{pmatrix} \quad (4.21)$$

Im RTK-Fix-Modus betragen die üblichen Standardabweichungen $\sigma_{v,n,k}$, $\sigma_{v,e,k}$, $\sigma_{v,d,k}$ etwa 1 bis 10 cm/s und die $|\sigma_{v,ij,k}|^{1/2}$ etwa 1 bis 6 cm/s.

Zeitpuls Für die Berechnung der Positionen und Geschwindigkeiten benötigt der GNSS-Empfänger eine gewisse Zeit. Dadurch sendet er seine Nachrichten mit Verzögerung. Im rechenintensiven RTK-Modus (siehe Seite 62) beträgt diese durchschnittliche Verzögerung in Abhängigkeit von der CPU-Last zwischen 30 und 80 ms, vgl. Abbildung 6.4; sie kann aber auch kurzzeitig bei bis zu 150 ms liegen. Somit kann die Verzögerung nicht vernachlässigt und muss bestimmt werden, um sie in der Sensorfusion berücksichtigen zu können, siehe Abschnitt 4.6.2.

Der GNSS-Empfänger sendet alle 100 ms eine Nachricht mit Positions- und Geschwindigkeitsmessungen. Die Nachrichten enthalten einen präzisen Zeitstempel mit der GNSS-Zeit, zu der die Messung gültig war (vgl. Abschnitt 2.5.2.4). Die interne Uhr des Prozessors des Mastermind-Boards, auf dem die Sensorfusion läuft, ist nicht mit der GNSS-Zeit synchronisiert und driftet mit mehreren Millisekunden pro 100 s. Um aus dem Zeitstempel der GNSS-Nachrichten das Alter der Messungen berechnen zu können, muss das Offset der Uhren fortlaufend bestimmt werden. Dazu wird ein Zeitpuls genutzt, den der GNSS-Empfänger zu jeder vollen GNSS-Sekunde auf einem digitalen Ausgang generiert

(PPS: pulse per second). Wenn der Mikroprozessor die aufsteigende Flanke dieses Rechteck-Impulses detektiert, wird die interne Uhrzeit gespeichert. Wenn dann eine GNSS-Nachricht empfangen wird, kann aus der Differenz der verstrichenen Zeit Δt_P seit dem letzten Zeitpuls und den Sekunden-Nachkommastellen Δt_N des Nachrichten-Zeitstempels das Alter Δt_A der Nachricht bestimmt werden:

$$\Delta t_A = \begin{cases} \Delta t_P - \Delta t_N & \text{wenn } \Delta t_P - \Delta t_N \geq 0 \\ \Delta t_P - \Delta t_N + 1 \text{ s} & \text{sonst} \end{cases} \quad (4.22)$$

Wenn für eine GNSS-Sekunde bereits der Zeitpuls empfangen wurde, aber die Nachricht noch zur vorhergehenden GNSS-Sekunde gehört, ist $\Delta t_P - \Delta t_N < 0$. In diesem Fall muss eine Sekunde addiert werden, da der zur Nachricht gehörende Zeitpuls ja eine Sekunde älter ist.

4.1.2.2 Experimenteller GPS- und Galileo-Empfänger

Auf dem Quadrocopter wird alternativ zum GNSS-RTK-Empfänger auch ein experimenteller GNSS-Empfänger [19] erprobt, der GPS- und Galileo-Signale verarbeiten kann. Dieser wurde vom Lehrstuhl IDS (ehemals EECS) der RWTH Aachen aufgebaut und programmiert. Der Empfänger wird hier für die Navigation im Galileo-Testgebiet automotive-GATE (siehe Abschnitt 3.5.4) adaptiert und auch dort getestet, siehe Abschnitt 6.1.2.

Der experimentelle GNSS-Empfänger besteht aus einem konfigurierbaren Analog-Frontend-Evaluation-Board und einem FPGA-Development-Board, siehe Abbildung 4.6. Durch die volle Konfigurierbarkeit des analogen Frontends und die freie Programmierbarkeit des FPGA-Boards können die Signalverarbeitung und die Algorithmen zur Schätzung von Position, Geschwindigkeit und Zeit (PVT, vgl. Abschnitt 2.5.2) genau an die spezifische Anwendung angepasst werden. So ist für den Einsatz auf dem Quadrocopter z. B.

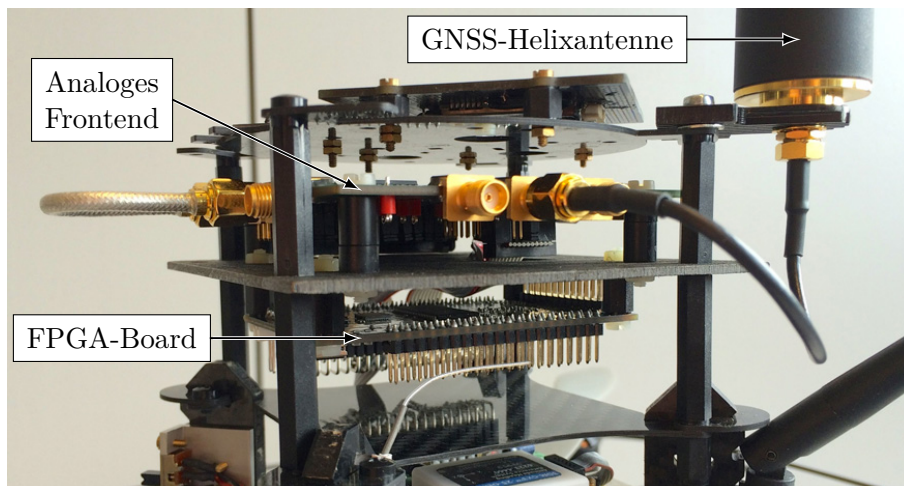


Abbildung 4.6: Experimenteller GNSS-Empfänger im Quadrocopter

die schnelle Reacquisition von Satelliten vorteilhaft, wenn bei starker Schrägstellung einzelne Satelliten kurzzeitig nicht mehr empfangen werden können. Außerdem können die Algorithmen und die Parametrierung von Sensorfusion und GNSS-Empfänger gut aufeinander abgestimmt werden. Sogar ein „deep coupling“, bei dem die Korrelatoren des GNSS-Empfängers bei kurzen Signalausfällen durch IMU-Signale gestützt werden, wäre denkbar.

Als analoges Frontend kommt der Maxim MAX2769B Chip in einem Evaluation-Kit zum Einsatz, bei dem viele Parameter wie die Abtast-Rate, die Filter-Bandbreite und die ADC-Word-Länge konfigurierbar sind. Das Frontend wird auf dem Quadrocopter mit einer AntCom 3D15A Helixantenne betrieben. Für den Empfang der Pseudolitensignale im Galileo-Testgebiet ist die Helixantenne besser geeignet als eine Patchantenne, da ihre Empfangscharakteristik besser zu den niedrigen Elevationen der Pseudoliten passt.

Für die digitale Signalverarbeitung wird ein Terasic DE0-nano Development-Board mit einem Altera Cyclone IV FPGA genutzt. Auf dem FPGA finden neben einem NIOS II Softcore-Prozessor z. B. 14 Korrelator-Module mit elf Speicher-basierten und drei LFSR-basierten Codegeneratoren oder 16 Korrelator-Module mit vier Speicher-basierten und zwölf LFSR-basierten Codegeneratoren Platz. Die Module mit den Speicher-basierten Generatoren können für die Verarbeitung von GPS-, Galileo- oder Pseudolitensignalen verwendet werden, während die LFSR-basierten Module nur GPS-Signalverarbeitung zulassen. Für die Galileo-Basisband-Verarbeitung werden aufgrund der BOC-Modulation des Galileo-Signals fünf anstatt drei komplexer Integratoren benötigt.

Der Softcore-Prozessor wird in der NIOS II/f Variante mit Hardware-Multiplikatoren und ohne Fließkommaeinheit bei einer Frequenz von 98,208 MHz betrieben, was dem sechsfachen der Abtastrate des Frontends entspricht. Die Basisbandverarbeitung, die Nachrichtendekodierung und die PVT-Algorithmen zur Schätzung von Position, Geschwindigkeit und Zeit werden auf dem Prozessor mit einer Rate von 5 Hz ausgeführt. Diese Rate wird durch eine Optimierung der PVT-Algorithmen erreicht [80, 47]. Es können Galileo-E1-B-Signale von Pseudoliten und Satelliten sowie GPS-L1-Signale gleichzeitig verarbeitet werden. Zur Berechnung der PVT-Lösung wird ein Least-Mean-Squares-Algorithmus mit Träger-Glättung („carrier aided smoothing“) eingesetzt, siehe Abschnitte 2.5.2.8, 2.5.2.6. Alternativ ist ein Ansatz mit Erweitertem Kalman-Filter verfügbar. Für die Navigation im Galileo-Testgebiet sind die Positionen der Pseudoliten im Empfänger hinterlegt, siehe Abschnitt 3.5.4.

Der experimentelle GNSS-Empfänger gibt die PVT-Lösung über eine serielle Schnittstelle als NMEA-Nachrichten RMC, GGA, GSA und GSV aus. Außerdem können RINEX-Dateien zum Postprocessing exportiert werden. Detaillierte interne Empfänger-Zustände können über ein binäres Datenformat ausgelesen und live auf einem Laptop dargestellt [19] und geloggt werden.

Die Erprobung des experimentellen GPS- und Galileo-Empfängers im Galileo-Testgebiet ist in Abschnitt 6.1.2 dargestellt.

4.1.3 Magnetometer

Im Quadrokofter sind zwei 3-Achs-Magnetometer verfügbar, mit der das lokale Magnetfeld gemessen werden kann. Zum einen bringt der Pelican ein Magnetometer mit (siehe Abbildung 4.5) und zum anderen verfügt die verwendete IMU (siehe Abschnitt 4.1.1) neben den Inertialsensoren auch über ein Magnetometer.

Das lokale Magnetfeld wird i. Allg. durch das Erdmagnetfeld dominiert, sodass eine Nordbestimmung mit dem Magnetometer möglich ist, siehe Abschnitt 4.5.1.3. Das Erdmagnetfeld hat an der Erdoberfläche annähernd die Form eines magnetischen Dipols, der leicht gegenüber der Erdrotationsachse gekippt ist. Der horizontale Anteil des Erdmagnetfeldvektors weist in die „magnetische“ Nordrichtung. Die Missweisung bezüglich der geographischen Nordrichtung, die Deklination δ_m , ist ortsabhängig. Der vertikale Anteil des Erdmagnetfeldvektors zeigt nach unten in Richtung Erdmittelpunkt. Der Winkel des Erdmagnetfeldvektors zur Horizontalen, die Inklination, nimmt mit dem Breitengrad zu; in Deutschland beträgt die Inklination etwa 60° .

Leichte, statische, körperfeste Störungen können durch eine Kalibrierung des Magnetometers kompensiert werden. Äußere Störer sind meist nicht kompensierbar.

4.1.4 Ultraschallsensor und Luftdrucksensor

Neben IMU, GNSS-Empfänger und Magnetometer verfügt der Quadrokofter mit Ultraschallsensor und Luftdrucksensor (vgl. Abschnitt 3.2.1) über zwei weitere Sensoren, die in der Sensorfusion verwendet werden könnten:

Der Ultraschallsensor ist so am Quadrokofter installiert, dass er die Höhe über Grund messen kann. Um die Ultraschallsensormessungen als Höheninformation in der Sensorfusion nutzen zu können, müsste ein detailliertes Geländemodell vorliegen, mit dem die gemessene Höhe über dem Grund in eine absolute Höhe umgerechnet werden kann. Der Ultraschallsensor wird daher nur zur Anpassung der Vertikalgeschwindigkeit der Landetrajektorie verwendet, vgl. Abschnitt 5.4.

Der Luftdrucksensor könnte zur Messung einer relativen Höhe verwendet werden, da sich der Luftdruck mit der Höhe ändert und der Luftdrucksensor dies auflösen kann. Die Höhenmessung wird durch Luftdruckschwankungen und Wind gestört. Da der GNSS-RTK-Empfänger eine präzise absolute Höhenmessung zur Verfügung stellt, wird auf die Nutzung des Luftdrucksensors in der Sensorfusion verzichtet.

4.2 Systemmodell für Sensorfusion

In diesem Abschnitt wird das Systemmodell des Quadrokofters für die Zustandsschätzung in der Sensorfusion zunächst in zeitkontinuierlicher Form

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), t) + \vec{g}(\vec{w}(t), t) \quad (4.23)$$

mit Systemzustand \vec{x} , bekanntem Eingang \vec{u} und Systemrauschen \vec{w} hergeleitet. Der Zeit-Parameter t wird im Folgenden weggelassen, um die Notation zu vereinfachen.

Auf Basis dieser Differentialgleichungen wird dann in Abschnitt 4.3 das zeitdiskrete, nicht-lineare Systemmodell (2.43) zur Propagation des Systemzustandes (2.52) im Erweiterten Kalman-Filter gebildet. Außerdem ist das kontinuierliche Systemmodell die Grundlage für das linearisierte Schätzfehlermodell in Abschnitt 4.4.1 zur Propagation der Schätzfehlerkovarianz (2.53).

Das Messmodell (2.44) für die unterschiedlichen Messgrößen, das für den Kalman-Estimations-Schritt benötigt wird, wird im Abschnitt 4.5 dargestellt.

Systemzustand

Der Punkt, dessen Position die Sensorfusion schätzt, soll mit U bezeichnet werden. Für die Sensorfusion ist es vorteilhaft, wenn dieser Punkt U , der Ursprung des IMU-Koordinatensystems C und der Ursprung des körperfesten Koordinatensystems B identisch sind³, da sonst zur translatorischen Transformation der spezifischen Kräfte der IMU Drehbeschleunigungen benötigt würden, siehe Abschnitt 4.1.1.1. Außerdem liegt der Ursprung des Navigationskoordinatensystems N im Punkt U , vgl. Abschnitt 2.1.1. Damit gilt für die Koordinaten des Punktes U in N , B und C :

$$\vec{p}_U^n = \vec{p}_U^b = \vec{p}_U^c = (0, 0, 0)^T \quad (4.24)$$

Relativ zum Erd-Bezugssystem E wird die Position des Punktes U im WGS84-Koordinatensystem W in den Polarkoordinaten Breitengrad φ , Längengrad λ und Höhe h dargestellt, siehe Abbildung 2.1 und Abschnitt 2.1.2:

$$\vec{p}_U^w = (\varphi, \lambda, h)^T \quad (4.25)$$

Die Geschwindigkeit des Quadropters bzw. des Punktes U bezüglich des Erd-Bezugssystems E in Koordinaten des Navigationskoordinatensystems N lautet:

$$\vec{v}_{eU}^n = \vec{v}_{eb}^n = \left(v_{eb,n}^n, v_{eb,e}^n, v_{eb,d}^n \right)^T \quad (4.26)$$

Die Lage des Quadropters gegenüber dem Navigationskoordinatensystem N soll als Lagequaternion \mathbf{q}_b^n dargestellt werden, vgl. (2.12).

Für die Inertialsensoren der IMU müssen außerdem Beschleunigungs- und Drehratensensor-Bias \vec{b}_a und \vec{b}_ω gemäß dem IMU-Modell (4.5)f. als Zustände eingeführt werden.

³Für die Regelung wird als Ursprung von B das Quadropter-Zentrum Z gewählt, da dort in etwa der Schwerpunkt liegt, siehe Abschnitte 2.1.1 und 5.1.1.

Der absolute Gesamtzustand \vec{x} für das Systemmodell (4.23) wird damit definiert als

$$\vec{x} = \left(\varphi, \lambda, h, \vec{v}_{eb}^{nT}, \mathbf{q}_b^{nT}, \vec{b}_a^T, \vec{b}_\omega^T \right)^T . \quad (4.27)$$

Bias

Die zeitliche Änderung der Beschleunigungs- und Drehratensensor-Bias \vec{b}_a und \vec{b}_ω ergibt sich aus den Sensormodellen (4.7)f.:

$$\dot{\vec{b}}_a = \vec{n}_{b_a} \quad (4.28)$$

$$\dot{\vec{b}}_\omega = \vec{n}_{b_\omega} \quad (4.29)$$

Lage

Für die Änderung der Lage \mathbf{q}_b^n gilt:

$$\dot{\mathbf{q}}_b^n = \frac{1}{2} \mathbf{\Omega}(\mathbf{q}_b^n) \vec{\omega}_{nb}^b \quad (4.30)$$

mit $\mathbf{\Omega}(\mathbf{q}_b^a)$ gemäß (2.18). Alternativ lässt sich die Änderung der Lage als Änderung der Richtungskosinusmatrix \mathbf{C}_b^n gemäß (2.19) oder als Änderung des Orientierungsvektors $\vec{\sigma}$ gemäß (2.20) formulieren.

Die benötigte Drehrate des Quadropters bezüglich des Navigationskoordinatensystems $\vec{\omega}_{nb}^b$ lässt sich aus der Drehrate im Inertialkoordinatensystem $\vec{\omega}_{ib}^b$, der Erddrehrate $\vec{\omega}_{ie}^n$ und der Transportrate $\vec{\omega}_{en}^n$ berechnen:

$$\vec{\omega}_{nb}^b = \vec{\omega}_{ib}^b - \mathbf{C}_b^{n,T} (\vec{\omega}_{ie}^n + \vec{\omega}_{en}^n) \quad (4.31)$$

Die Drehrate im Inertialkoordinatensystem $\vec{\omega}_{ib}^b$ ergibt sich gemäß (4.5) aus der von der IMU gemessenen Drehrate abzüglich des Drehraten-Bias und des Rauschterms:

$$\vec{\omega}_{ib}^b = \tilde{\vec{\omega}}_{ib}^b - \vec{b}_\omega - \vec{n}_\omega \quad (4.32)$$

Die Erddrehrate (2.7) lässt sich in Koordinaten des Navigationskoordinatensystems ausdrücken als

$$\vec{\omega}_{ie}^n = \begin{pmatrix} \Omega \cos \varphi \\ 0 \\ -\Omega \sin \varphi \end{pmatrix} . \quad (4.33)$$

Die Transportrate $\vec{\omega}_{en}^n$ beschreibt die notwendige Drehung des Navigationskoordinatensystems gegenüber dem erdfesten Koordinatensystem bei einer Bewegung gegenüber der

Erde, damit der Basisvektor \vec{n}_d immer nach unten in Richtung Erde zeigt, vgl. Definition des Navigationskoordinatensystems in Abschnitt 2.1.1:

$$\vec{\omega}_{en}^n = \begin{pmatrix} 0 & \frac{1}{R_e+h} & 0 \\ -\frac{1}{R_n+h} & 0 & 0 \\ 0 & -\frac{\tan \varphi}{R_e+h} & 0 \end{pmatrix} \vec{v}_{eb}^n \quad (4.34)$$

Hier und im Folgenden wird angenommen, dass die Geschwindigkeit des Quadropters im Verhältnis zum Erdradius klein ist, sodass die Änderung der Krümmungsradien R_n und R_e mit $\dot{\varphi}$ gemäß (2.1)f. vernachlässigt werden kann.

Geschwindigkeit

Nun wird die Differentialgleichung für die zeitliche Änderung des Geschwindigkeits-Zustands \vec{v}_{eb}^n hergeleitet. Mit \vec{p}_U als Positionsvektor vom Mittelpunkt der Erde zum Quadropter bzw. zum Punkt U folgt mit (2.22) und (B.10) für die Geschwindigkeit \vec{v}_{eb} :

$$\vec{v}_{eb} = \vec{v}_{eU} = \frac{E d\vec{p}_U}{dt} = \frac{I d\vec{p}_U}{dt} - \vec{\omega}_{ie} \times \vec{p}_U = \vec{v}_{ib} - \vec{\omega}_{ie} \times \vec{p}_U \quad (4.35)$$

Damit gilt für die Beschleunigung \vec{a}_{eb} gemäß (2.23):

$$\vec{a}_{eb} = \frac{E d\vec{v}_{eb}}{dt} = \frac{E d\vec{v}_{ib}}{dt} - \underbrace{\frac{E d\vec{\omega}_{ie}}{dt}}_0 \times \vec{p}_U - \vec{\omega}_{ie} \times \underbrace{\frac{E d\vec{p}_U}{dt}}_{\vec{v}_{eb}} \quad (4.36)$$

$$= \vec{a}_{ib} - \vec{\omega}_{ie} \times \vec{v}_{ib} - \vec{\omega}_{ie} \times \vec{v}_{eb} \quad (4.37)$$

$$= \vec{f}_{ib} + \vec{g} - \vec{\omega}_{ie} \times (\vec{v}_{eb} + \vec{\omega}_{ie} \times \vec{p}_U) - \vec{\omega}_{ie} \times \vec{v}_{eb} \quad (4.38)$$

Für die Umformung (4.37) wird $\vec{a}_{ib} = \frac{I d\vec{v}_{ib}}{dt} = \frac{E d\vec{v}_{eb}}{dt} + \vec{\omega}_{ie} \times \vec{v}_{ib}$ gemäß (2.23) und (B.10) sowie eine konstante Erddrehrate $\vec{\omega}_{ie}$ gemäß (2.7) ausgenutzt. (4.38) folgt mit (4.3) und (4.35).

Für die Ableitung der Geschwindigkeit \vec{v}_{eb} bezüglich des Navigationskoordinatensystems N gilt mit (B.10) und \vec{a}_{eb} aus (4.38):

$$\frac{N d\vec{v}_{eb}}{dt} = \frac{E d\vec{v}_{eb}}{dt} - \vec{\omega}_{en} \times \vec{v}_{eb} \quad (4.39)$$

$$= \vec{f}_{ib} + \underbrace{\vec{g} - \vec{\omega}_{ie} \times (\vec{\omega}_{ie} \times \vec{p}_U)}_{\vec{g}_i} - 2\vec{\omega}_{ie} \times \vec{v}_{eb} - \vec{\omega}_{en} \times \vec{v}_{eb} \quad (4.40)$$

mit der Schwerebeschleunigung \vec{g}_i gemäß (2.9), (2.8).

Damit lässt sich die Differentialgleichung für die Geschwindigkeit \vec{v}_{eb}^n im Navigationskoordinatensystem schreiben als

$$\dot{\vec{v}}_{eb}^n = \frac{N}{dt} d\vec{v}_{eb}^n = \mathbf{C}_b^n \vec{f}_{ib}^b + \vec{g}_l^n - (2\vec{\omega}_{ie}^n + \vec{\omega}_{en}^n) \times \vec{v}_{eb}^n \quad (4.41)$$

mit der Erddrehrate $\vec{\omega}_{ie}^n$ und der Transportrate $\vec{\omega}_{en}^n$ (4.33), (4.34). \vec{f}_{ib}^b ist gemäß (4.6) die von der IMU gemessene spezifische Kraft abzüglich Bias und Rauschterm:

$$\vec{f}_{ib}^b = \tilde{\vec{f}}_{ib}^b - \vec{b}_a - \vec{n}_a \quad (4.42)$$

Position

Für die Änderung der Position \vec{p}_U^w (4.25) in WGS84-Koordinaten gilt in Abhängigkeit von der Geschwindigkeit \vec{v}_{eb} gemäß [84, 28]:

$$\dot{\vec{p}}_U^w = \begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{1}{(R_e+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \vec{v}_{eb}^n \quad (4.43)$$

Zusammenfassung des Systemmodells

Die Gleichungen (4.43), (4.41), (4.30), (4.28) und (4.29) sind die kontinuierlichen Differentialgleichungen des Systemmodells (4.23) in der Form

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}(t), \vec{u}(t), t) + \vec{g}(\vec{w}(t), t)$$

mit Systemzustand $\vec{x}(t)$ (4.27). Der Eingang $\vec{u}(t)$ und das Systemrauschen $\vec{w}(t)$ für das Systemmodell lauten damit:

$$\vec{u}(t) = \left(\tilde{\vec{f}}_{ib}^{b,T}, \tilde{\vec{\omega}}_{ib}^{b,T} \right)^T \quad (4.44)$$

$$\vec{w}(t) = \left(\vec{n}_a^T, \vec{n}_\omega^T, \vec{n}_{b_a}^T, \vec{n}_{b_\omega}^T \right)^T \quad (4.45)$$

Systemrauschen

Zur Propagation der Schätzfehlerkovarianzmatrix \mathbf{P}_k in Abschnitt 4.4 wird die Kovarianzmatrix \mathbf{Q}_k (2.30) des diskreten Systemrauschens \vec{w}_k benötigt. Hier sollen dazu die spektralen Leistungsdichten des kontinuierlichen Systemrauschens $\vec{w}(t)$ in Form der Matrix \mathbf{Q} dargestellt werden, die dann diskretisiert werden kann.

Die Matrix \mathbf{Q} enthält gemäß (2.72) und (A.28) auf ihrer Diagonalen die spektralen Leistungsdichten des kontinuierlichen Systemrauschens $\vec{w}(t) = (\vec{n}_a^T, \vec{n}_{b_\omega}^T, \vec{n}_{b_a}^T, \vec{n}_\omega^T)^T$ (4.45):

$$\mathbf{Q} = \begin{pmatrix} R_a \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & R_\omega \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & R_{b_a} \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & R_{b_\omega} \mathbf{I}_3 \end{pmatrix} \quad (4.46)$$

Die Identifikation der spektralen Leistungsdichten $R_a, R_\omega, R_{b_a}, R_{b_\omega}$ des mittelwertfreien, normalverteilten, kontinuierlichen weißen Rauschens (A.25), (A.17) ist in Abschnitt 4.1.1.2 dargestellt. Für die in der Sensorfusion verwendeten Zahlenwerte siehe Tabellen 4.2 und 4.5.

4.3 Propagation des Systemzustandes

Nun wird auf Basis des in Abschnitt 4.2 hergeleiteten zeitkontinuierlichen Systemmodells das zeitdiskrete, nichtlineare Systemmodell (2.43) zur Propagation des Systemzustandes eingeführt. Die Propagation des Erweiterten Kalman-Filters (2.52) wird hier in der Form

$$\hat{\vec{x}}_k^- = \vec{f}_k(\hat{\vec{x}}_{k-1}, \vec{u}_k) \quad (4.47)$$

dargestellt. Im Zeitschritt k wird damit der aktuelle Zustand $\hat{\vec{x}}_k^-$ aus dem letzten $\hat{\vec{x}}_{k-1}$ und dem aktuellen Eingang \vec{u}_k berechnet (Rückwärts-Verfahren) und nicht der zukünftige Zustand $\hat{\vec{x}}_{k+1}$ aus $\hat{\vec{x}}_k$ und \vec{u}_k (Vorwärts-Verfahren) wie in (2.52).

Der Anteil des Systemrauschens $\vec{g}_k(\vec{w}_k)$ in (2.43) wird für die Propagation nicht benötigt und entfällt hier. Er findet im linearen Schätzfehlermodell zur Propagation der Schätzfehlerkovarianz in Abschnitt 4.4 Berücksichtigung.

Der Zeitschritt-Dauer T_s für die Propagation wird hier zu 0,01 sec gewählt. Der Schätzwert des Systemzustandes (4.27) zum Zeitpunkt $t = kT_s$ ist

$$\hat{\vec{x}}_k = \left(\hat{\varphi}_k, \hat{\lambda}_k, \hat{h}_k, \hat{v}_{eb,k}^n, \hat{\mathbf{q}}_{b,k}^n, \hat{b}_{a,k}^T, \hat{b}_{\omega,k}^T \right)^T. \quad (4.48)$$

Im vorliegenden System ist die IMU direkt auf dem Quadrokofter montiert und erfährt die gleichen Drehraten und Beschleunigungen wie der Quadrokofter. Ein solches Setup wird auch als „Strapdown“-System⁴ und die zugehörige Zustandspropagation als „Strapdown“-Rechnung bezeichnet [84, 28].

⁴Eine übliche Alternative ist, die IMU auf einer kardanisch gelagerten Plattform zu montieren, dessen Lage durch eine Regelung konstant gehalten wird [84, 28]. Solch ein mechanischer Aufbau ist aber für kleine Systeme wie den Quadrokofter im Allgemeinen nicht geeignet.

Propagation des Bias

Zur Propagation der Bias-Schätzwerte $\hat{\vec{b}}_{a,k}$ und $\hat{\vec{b}}_{\omega,k}$ werden $\dot{\vec{b}}_a$ und $\dot{\vec{b}}_\omega$ aus (4.28)f. über einen Zeitschritt $T_s = t_k - t_{k-1}$ integriert:

$$\vec{b}_{i,k} - \vec{b}_{i,k-1} = \int_{t_{k-1}}^{t_k} \dot{\vec{b}}_i dt = \int_{t_{k-1}}^{t_k} \vec{n}_{b_i} dt \quad \forall i \in \{a, \omega\} \quad (4.49)$$

Da die Realisierung des Rauschterms \vec{n}_{b_i} im Zeitintervall (t_{k-1}, t_k) unbekannt ist, wird er durch den Erwartungswert $E[\vec{n}_{b_i}]$ ersetzt. Dieser ist null, weil für \vec{n}_{b_a} und \vec{n}_{b_ω} Mittelwertfreiheit angenommen wird (siehe Seite 101). Damit wird auch das Integral in (4.49) zu null und die Bias werden als konstant propagiert:

$$\hat{\vec{b}}_{a,k}^- = \hat{\vec{b}}_{a,k-1} \quad (4.50)$$

$$\hat{\vec{b}}_{\omega,k}^- = \hat{\vec{b}}_{\omega,k-1} \quad (4.51)$$

Propagation der Lage

Zur Propagation der Lage wird das Integral über die Drehrate $\vec{\omega}_{ib}^b$ in einem Zeitschritt $T_s = t_k - t_{k-1}$ benötigt. Mit der Substitution von $\vec{\omega}_{ib}^b$ gemäß (4.32) gilt:

$$\begin{aligned} \int_{t_{k-1}}^{t_k} \vec{\omega}_{ib}^b dt &= \int_{t_{k-1}}^{t_k} \tilde{\vec{\omega}}_{ib}^b dt - \int_{t_{k-1}}^{t_k} \vec{b}_\omega dt - \int_{t_{k-1}}^{t_k} \vec{n}_\omega dt \\ &\approx \Delta \tilde{\vec{\sigma}}_{ib,k}^b - T_s \hat{\vec{b}}_{\omega,k-1} - \vec{0} = \Delta \hat{\vec{\sigma}}_{ib,k}^b \end{aligned} \quad (4.52)$$

In der zweiten Zeile ist das Integral über $\tilde{\vec{\omega}}_{ib}^b$ gemäß (4.9) und (4.11) durch den von der IMU ausgegebenen und nach B gedrehten Delta-Winkel $\Delta \tilde{\vec{\sigma}}_{ib,k}^b$ ersetzt. Die Update-Rate der Delta-Winkel-Messwerte $\Delta \tilde{\vec{\sigma}}_{ib,k}^b$ ist hier gleich der Propagations-Rate $1/T_s$. Außerdem wird genutzt, dass das Bias \vec{b}_ω über einen Zeitschritt näherungsweise konstant ist und das Integral über den Rauschterm \vec{n}_ω wegen $E[\vec{n}_\omega] = \vec{0}$ zu null wird. Der Bias-korrigierte Delta-Winkel soll mit $\Delta \hat{\vec{\sigma}}_{ib,k}^b$ bezeichnet werden.

Für die mit dem propagierten Bias-Schätzwert $\hat{\vec{b}}_{\omega,k-1}$ korrigierte Drehrate gilt damit:

$$\hat{\vec{\omega}}_{ib,k}^{b-} \approx \frac{\Delta \hat{\vec{\sigma}}_{ib,k}^b}{T_s} = \frac{\Delta \tilde{\vec{\sigma}}_{ib,k}^b}{T_s} - \hat{\vec{b}}_{\omega,k-1} \quad (4.53)$$

Zur Propagation der Lage wird – anstatt $\dot{\mathbf{q}}_b^n$ aus (4.30) direkt zu integrieren – gemäß [84] ein Korrektur-Quaternion gebildet, indem die Orientierungsvektor-Änderung $\vec{\sigma}_{nb}$ aus der Bortzschen Orientierungsvektordifferentialgleichung (2.20) integriert wird. (2.20) lässt sich für kleine Winkel vereinfachen: durch eine Reihenentwicklung des Faktors vor dem zweiten Kreuzproduktterm lässt sich dieser auf den konstanten Wert $1/12$ zurückführen.

Damit ist der zweite Kreuzproduktterm kleiner als der erste und wird vernachlässigt [84]. Die Vereinfachung von (2.20) für kleine Winkel σ lautet damit:

$$\dot{\vec{\sigma}}_{nb} \approx \vec{\omega}_{nb}^b + \frac{1}{2} \vec{\sigma} \times \vec{\omega}_{nb}^b \quad (4.54)$$

Für diese Vereinfachung muss die in einem Zeitschritt auftretende Lageänderung ausreichend klein bzw. die Updaterate ausreichend hoch sein. Die Lageänderung im Zeitintervall $T_s = t_k - t_{k-1}$ ist damit:

$$\Delta \vec{\sigma}_{nb,k} = \int_{t_{k-1}}^{t_k} \vec{\omega}_{nb}^b dt + \frac{1}{2} \int_{t_{k-1}}^{t_k} \left(\int_{t_{k-1}}^t \vec{\omega}_{nb}^b d\tau \right) \times \vec{\omega}_{nb}^b dt \quad (4.55)$$

Der zweite Term wird als Coning-Term bezeichnet. Er wird zu null, wenn sich die Richtung des Vektors $\vec{\omega}_{nb}^b$ im Zeitschritt näherungsweise nicht ändert. Dies kann hier angenommen werden, siehe Erläuterungen im Abschnitt „Coning und Sculling“ auf Seite 121. Damit ergibt sich mit der Substitution von $\vec{\omega}_{nb}^b$ gemäß (4.31):

$$\Delta \vec{\sigma}_{nb,k} \approx \int_{t_{k-1}}^{t_k} \vec{\omega}_{nb}^b dt = \int_{t_{k-1}}^{t_k} \vec{\omega}_{ib}^b dt - \int_{t_{k-1}}^{t_k} \mathbf{C}_b^{n,T} (\vec{\omega}_{ie}^n + \vec{\omega}_{en}^n) dt \quad (4.56)$$

Das Integral $\int_{t_{k-1}}^{t_k} \vec{\omega}_{ib}^b dt$ wird durch den Bias-korrigierten Delta-Winkel $\Delta \hat{\sigma}_{ib,k}^b$ aus (4.52) ersetzt. Der Integrand des letzten Integrals ist über einen Zeitschritt näherungsweise konstant. Damit gilt für die geschätzte Orientierungsänderung $\Delta \hat{\sigma}_{nb,k}$:

$$\Delta \hat{\sigma}_{nb,k} \approx \Delta \hat{\sigma}_{ib,k}^b - \hat{b}_{\omega,k-1} T_s - \mathbf{C}_{b,k-1}^{n,T} (\hat{\omega}_{ie,k-1}^n + \hat{\omega}_{en,k-1}^n) T_s \quad (4.57)$$

mit $\Delta \hat{\sigma}_{ib,k}^b$ aus (4.11). Die Größen $\mathbf{C}_{b,k-1}^{\hat{n}}$, $\hat{\omega}_{ie,k-1}^n$ und $\hat{\omega}_{en,k-1}^n$ ergeben sich anhand von (2.13), (4.33) und (4.34) aus den Schätzwerten des Zustands im letzten Zeitschritt $\hat{\mathbf{q}}_{b,k-1}^n$, $\hat{\varphi}_{k-1}$, \hat{h}_{k-1} und $\hat{v}_{eb,k-1}^n$.

Zur Propagation des Lagequaternions $\hat{\mathbf{q}}_b^n$ wird die Orientierungsänderung $\Delta \hat{\sigma}_{nb,k}$ wie in (2.12) in ein Korrektur-Quaternion $\mathbf{q}_c(\Delta \hat{\sigma}_{nb,k})$ umgerechnet, das diese Lageänderung beschreibt. Um eine Division durch null bei $|\Delta \hat{\sigma}_{nb,k}| = 0$ zu vermeiden, werden in der Implementierung von (2.12) Reihenentwicklungen für Kosinus und Sinus verwendet [84]:

$$\mathbf{q}_c(\vec{\sigma}) \approx \begin{pmatrix} 1 - \frac{1}{2! \cdot 2^2} \sigma^2 + \frac{1}{4! \cdot 2^4} \sigma^4 - \frac{1}{6! \cdot 2^6} \sigma^6 \\ \vec{\sigma} \left(\frac{1}{2} - \frac{1}{3! \cdot 2^3} \sigma^2 + \frac{1}{5! \cdot 2^5} \sigma^4 - \frac{1}{7! \cdot 2^7} \sigma^6 \right) \end{pmatrix} \quad \text{mit } \sigma = |\vec{\sigma}| \quad (4.58)$$

Die Propagation des Lagequaternions erfolgt schließlich durch eine rechtsseitige Quaternionen-Multiplikation (B.1) des Lagequaternions aus dem vorhergehenden Zeitschritt mit $\mathbf{q}_c(\Delta \hat{\sigma}_{nb,k})$:

$$\hat{\mathbf{q}}_{b,k}^n = \hat{\mathbf{q}}_{b,k-1}^n \bullet \mathbf{q}_c(\Delta \hat{\sigma}_{nb,k}) \quad (4.59)$$

Propagation der Geschwindigkeit

Wie bei der Propagation der Lage wird zur Propagation der Geschwindigkeit das Integral über die spezifische Kraft \vec{f}_{ib}^b in einem Zeitschritt $T_s = t_k - t_{k-1}$ benötigt. Mit \vec{f}_{ib}^b gemäß (4.42) gilt:

$$\begin{aligned} \int_{t_{k-1}}^{t_k} \vec{f}_{ib}^b dt &= \int_{t_{k-1}}^{t_k} \tilde{\vec{f}}_{ib}^b dt - \int_{t_{k-1}}^{t_k} \vec{b}_a dt - \int_{t_{k-1}}^{t_k} \vec{n}_a dt \\ &\approx \Delta \tilde{\vec{v}}_{ib,k}^b - T_s \hat{\vec{b}}_{a,k-1} - \vec{0} = \Delta \hat{\vec{v}}_{ib,k}^b \end{aligned} \quad (4.60)$$

Analog zu (4.52) wird in der zweiten Zeile das Integral über $\tilde{\vec{f}}_{ib}^b$ gemäß (4.10) und (4.11) durch die Delta-Geschwindigkeit $\Delta \tilde{\vec{v}}_{ib,k}^b$ und das Integral über den Rauschterm \vec{n}_a wegen $E[\vec{n}_\omega] = \vec{0}$ durch null ersetzt. Die Update-Rate der Delta-Geschwindigkeits-Messwerte $\Delta \tilde{\vec{v}}_{ib,k}^b$ ist hier auch gleich der Propagations-Rate $1/T_s$. Außerdem wird wieder angenommen, dass das Bias \vec{b}_a über einen Zeitschritt näherungsweise konstant ist. Die Bias-korrigierte Delta-Geschwindigkeit wird als $\Delta \hat{\vec{v}}_{ib,k}^b$ bezeichnet.

Damit gilt für die mit dem propagierten Bias-Schätzwert $\hat{\vec{b}}_{a,k-1}$ korrigierte spezifische Kraft $\hat{\vec{f}}_{ib,k}^{b-}$ analog zu (4.53):

$$\hat{\vec{f}}_{ib,k}^{b-} = \frac{\Delta \hat{\vec{v}}_{ib,k}^{b-}}{T_s} = \frac{\Delta \tilde{\vec{v}}_{ib,k}^b}{T_s} - \hat{\vec{b}}_{a,k-1} \quad (4.61)$$

Zur Propagation der Geschwindigkeit wird nun \dot{v}_{eb}^n aus (4.41) über einen Zeitschritt $T_s = t_k - t_{k-1}$ integriert:

$$\vec{v}_{eb,k}^{n-} = \vec{v}_{eb,k-1}^n + \int_{t_{k-1}}^{t_k} \dot{v}_{eb}^n dt \quad (4.62)$$

Die Schwerebeschleunigung und der Kreuzprodukt-Term in (4.41) können in der Regel als konstant über einen Zeitschritt angenommen werden, für den ersten Term in (4.41) gilt dies nicht. Als Approximation des Integrals über den ersten Term wird daher in [84] vorgeschlagen:

$$\begin{aligned} \int_{t_{k-1}}^{t_k} \mathbf{C}_b^n \vec{f}_{ib}^b dt &= \mathbf{C}_{b,k-1}^n \left(\int_{t_{k-1}}^{t_k} \vec{f}_{ib}^b dt + \frac{1}{2} \Delta \vec{\sigma}_{nb,k} \times \int_{t_{k-1}}^{t_k} \vec{f}_{ib}^b dt \right) \\ &\quad + \frac{1}{2} \mathbf{C}_{b,k-1}^n \int_{t_{k-1}}^{t_k} \left(\int_{t_{k-1}}^{t_k} \vec{\omega}_{nb}^b d\tau \times \vec{f}_{ib}^b - \vec{\omega}_{nb}^b \times \int_{t_{k-1}}^{t_k} \vec{f}_{ib}^b d\tau \right) dt \end{aligned} \quad (4.63)$$

Der Summand in der zweiten Zeile ist der Sculling-Term. Dieser kann hier vernachlässigt werden, siehe folgenden Abschnitt „Coning und Sculling“.

Mit der Substitution von $\int_{t_{k-1}}^{t_k} \vec{f}_{ib}^b dt$ in (4.63) durch $\Delta \hat{\vec{v}}_{ib,k}^b$ gemäß (4.60) ergibt sich für die Propagation der Geschwindigkeit gemäß (4.62):

$$\begin{aligned} \hat{\vec{v}}_{eb,k}^n = & \hat{\vec{v}}_{eb,k-1}^n + \mathbf{C}_{b,k-1}^{\hat{n}} \left(\Delta \hat{\vec{v}}_{ib,k}^b + \frac{1}{2} \Delta \hat{\vec{\sigma}}_{nb,k} \times \Delta \hat{\vec{v}}_{ib,k}^b \right) \\ & + T_s \hat{g}_l^n - T_s (2 \hat{\vec{\omega}}_{ie,k-1}^n + \hat{\vec{\omega}}_{en,k-1}^n) \times \hat{\vec{v}}_{eb,k-1}^n \end{aligned} \quad (4.64)$$

mit $\Delta \hat{\vec{v}}_{ib,k}^b$ aus (4.60) und $\Delta \hat{\vec{\sigma}}_{nb,k}$ aus (4.57). Die Größen $\mathbf{C}_{b,k-1}^{\hat{n}}$, $\hat{\vec{\omega}}_{ie,k-1}^n$, $\hat{\vec{\omega}}_{en,k-1}^n$ werden wieder aus den Schätzwerten des Systemzustandes im letzten Zeitschritt $\hat{\vec{x}}_{k-1}$ berechnet. \hat{g}_l^n ist die geschätzte Schwerkraft gemäß (4.133).

Coning und Sculling

In der Propagation von Lage und Geschwindigkeit werden die Coning- und Sculling-Terme in (4.55) und (4.63) vernachlässigt. Dies sollte erläutert werden.

Als „Coning Motion“ werden phasenverschobene Drehschwingungen um zwei orthogonale Achsen bezeichnet [84]. Dann beschreibt die Rotationsachse einen Kegel (englisch „cone“) [49]. Coning Motion, die z. B. durch Vibrationen verursacht werden kann, ist ein typisches Bewegungsmuster, bei dem sich die Richtung der Drehrate $\vec{\omega}_{nb}^b$ in einem Strapdown-Zeitschritt ändern kann. Wenn die Vibrationsfrequenz über der Abtastrate liegt, kann der Coning-Term in (4.55) einen signifikanten Beitrag zu $\Delta \hat{\vec{\sigma}}_{nb,k}$ liefern, sodass die Vernachlässigung des Coning-Terms zu einem Fehler in der Propagation der Lage führen würde.

Gleichzeitige rotatorische und translatorische Schwingungen bezüglich zweier orthogonaler Achsen werden als „Sculling Motion“ bezeichnet [84]. Das Bewegungsmuster gleicht der Bewegung der Ruder beim Rudern (Skullen, englisch „sculling“) [49]. Analog zur Coning Motion kann – wenn die Frequenz der Sculling Motion über der Abtastrate liegt – die Vernachlässigung des Sculling-Terms in (4.63) zu einem Fehler in der Propagation der Geschwindigkeit führen.

Auf dem Quadropter werden hochfrequente Schwingungen durch die drehenden Rotoren verursacht. Ursprünglich trat dadurch bei Experimenten mit hohen Rotordrehzahlen im Stand u. a. in der Beschleunigungsmessung der IMU eine vertikale Scheinbeschleunigung von bis zu 5 m/s^2 auf. Damit wäre die Sensorfusion unbrauchbar. Um dem entgegen zu wirken, wurden zwei Maßnahmen getroffen:

Zum einen ist die IMU-Platine auf Gummi-Dämpferelementen gelagert, wodurch die hochfrequenten Schwingungen, die auf die IMU wirken, reduziert werden, siehe Abbildung 4.1. Dabei muss darauf geachtet werden, dass durch die Dämpfer die für die Regelung benötigte Bewegungsschätzung des Quadropters nicht zu stark verfälscht wird.

Zum anderen wird die hohe interne Abtastung der IMU mit 1000 Hz genutzt. Die Drehraten und Beschleunigungen werden – über 10 Zeitschritte aufintegriert – von der IMU als Delta-Winkel $\Delta \hat{\vec{\sigma}}$ und Delta-Geschwindigkeiten $\Delta \hat{\vec{v}}$ ausgegeben [49], siehe (4.9)f.

Mit diesen Maßnahmen tritt die Scheinbeschleunigung nicht mehr auf. Die Sensorfusion ist in der Lage, u. a. Orientierung und Geschwindigkeit des Quadropters auch im dynamischen Flug präzise zu schätzen, siehe Erprobung in Abschnitt 6.1.1.

Propagation der Position

Die Propagation der Position ergibt sich aus (4.43) durch Integration über einen Zeitschritt $T_s = t_k - t_{k-1}$ zu

$$\vec{p}_{U,k}^{w-} = \begin{pmatrix} \hat{\varphi}_k^- \\ \hat{\lambda}_k^- \\ \hat{h}_k^- \end{pmatrix} = \begin{pmatrix} \hat{\varphi}_{k-1}^- \\ \hat{\lambda}_{k-1}^- \\ \hat{h}_{k-1}^- \end{pmatrix} + T_s \begin{pmatrix} \frac{1}{R_n + \hat{h}_{k-1}} & 0 & 0 \\ 0 & \frac{1}{(R_e + \hat{h}_{k-1}) \cos \hat{\varphi}_{k-1}} & 0 \\ 0 & 0 & -1 \end{pmatrix} \hat{v}_{eb,k-1}^n \quad (4.65)$$

Für R_n und R_e gemäß (2.1)f. genügt es, diese mit niedrigerer Rate in Abhängigkeit von $\hat{\varphi}_{k-1}$ zu aktualisieren [84].

Zusammenfassung der Systemzustandspropagation

Die Gleichungen (4.65), (4.64), (4.59), (4.50), (4.51) sind die Propagation des Systemzustandes \hat{x}_k (4.48) in der Form (4.47):

$$\hat{x}_k^- = \vec{f}_k(\hat{x}_{k-1}, \vec{u}_k)$$

Als Eingang \vec{u}_k werden hier – anstatt spezifischer Kräfte und Drehraten wie in (4.44) – direkt die nach B gedrehten Delta-Geschwindigkeiten und Delta-Winkel (4.11) der IMU verwendet:

$$\vec{u}_k = \begin{pmatrix} \Delta \tilde{\sigma}_{ib,k}^b \\ \Delta \tilde{\nu}_{ib,k}^b \end{pmatrix} \quad (4.66)$$

4.4 Propagation der Schätzfehlerkovarianz

Das Erweiterte Kalman-Filter schätzt die Schätzgüte als Schätzfehlerkovarianzmatrix \mathbf{P}_k (2.46). Die Propagation von \mathbf{P}_k erfolgt gemäß (2.53):

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_k \mathbf{G}_{k-1}^T \quad (4.67)$$

Der Zeitindex k ist gegenüber (2.53) wie in (4.47) um 1 verschoben: Die aktuelle Kovarianzmatrix \mathbf{P}_k^- wird aus \mathbf{P}_{k-1} des letzten Zeitschrittes berechnet.

Zur Propagation der Schätzfehlerkovarianzmatrix werden die Systemmatrizen $\mathbf{\Phi}_k$ und \mathbf{G}_k eines zeitdiskreten, linearen Schätzfehlermodells (2.50) zum Zeitpunkt $k - 1$ und die Kovarianzmatrix \mathbf{Q}_k (2.30) des Systemrauschens im Zeitdiskreten benötigt.

Zur Bestimmung des zeitdiskreten, linearen Schätzfehlermodells wird im Folgenden zunächst ein kontinuierliches, linearisiertes Schätzfehlermodell hergeleitet, das dann diskretisiert wird.

4.4.1 Schätzfehlermodell

Das kontinuierliche, linearisierte Schätzfehlermodell hat in Anlehnung an das nichtlineare Modell (4.23) die Form:

$$\Delta \dot{\vec{x}}(t) = \mathbf{F}(t)\Delta \vec{x}(t) + \mathbf{G}(t)\vec{w}(t) \quad (4.68)$$

Da der Eingang \vec{u} in (4.23) bekannt ist, tritt hier kein Eingangs-Fehler $\Delta \vec{u}(t)$ auf. Das Systemrauschen $\vec{w}(t)$ ist identisch zu dem in (4.45).

Im Folgenden werden der Schätzfehler-Zustand $\Delta \vec{x}$ und seine zeitlichen Änderungen dargestellt.

Schätzfehler-Zustand

Für den Schätzfehler-Zustand $\Delta \vec{x}$ werden der Geschwindigkeitsfehler und die Bias-Fehler als lineare Abweichungsgrößen vom Gesamtzustand (4.27) als $\hat{\vec{x}} - \vec{x}$ gemäß (2.48) definiert. Für den Positionsfehler und den Lagefehler sind abweichende Definitionen gemäß (2.49) besser geeignet [84].

Anstatt $\Delta \varphi$, $\Delta \lambda$, Δh wird hier ein Positionsfehler in den kartesischen Koordinaten des Navigationskoordinatensystems N gewählt:

$$\Delta \vec{p} = \hat{p}_U^n - \vec{p}_U^n = \begin{pmatrix} \Delta p_n \\ \Delta p_e \\ \Delta p_d \end{pmatrix} \quad (4.69)$$

mit $\vec{p}_U^n = (0, 0, 0)^T$ gemäß (4.24).

Der Geschwindigkeitsfehler $\Delta \vec{v}$ wird als lineare Abweichungsgröße formuliert:

$$\Delta \vec{v} = \hat{v}_{eb}^n - v_{eb}^n = \begin{pmatrix} \Delta v_{eb,n}^n \\ \Delta v_{eb,e}^n \\ \Delta v_{eb,d}^n \end{pmatrix} \quad (4.70)$$

Der Lagefehler wird als Koordinatentransformation zwischen der tatsächlichen Ausrichtung des Navigationskoordinatensystems N und der geschätzten Ausrichtung \hat{N} ausgedrückt. Das Quaternion $\mathbf{q}_n^{\hat{n}}$ (2.12) beschreibt diese Transformation. Für ein lineares Fehlermodell ist dieses Quaternion jedoch ungeeignet da $\mathbf{q}_b^{\hat{n}} \neq \mathbf{q}_b^n + \mathbf{q}_n^{\hat{n}}$. Daher soll hier der Lage-Schätzfehlerzustand als Winkelfehler α , β , γ eingeführt werden.

Die Richtungskosinusmatrix $\mathbf{C}_n^{\hat{n}}$ beschreibt ebenfalls die Koordinatentransformation von N nach \hat{N} , vgl. (2.14). Mit (2.13) kann $\mathbf{C}_n^{\hat{n}}$ aus $\mathbf{q}_n^{\hat{n}}$ berechnet werden. $\mathbf{C}_n^{\hat{n}}$ setzt die reale Lage \mathbf{C}_b^n zur geschätzten Lage $\mathbf{C}_b^{\hat{n}}$ wie folgt in Beziehung:

$$\mathbf{C}_b^{\hat{n}} = \mathbf{C}_n^{\hat{n}} \mathbf{C}_b^n \quad (4.71)$$

Die Richtungskosinusmatrix $\mathbf{C}_n^{\hat{n}}$ kann durch die Winkelfehler α, β, γ angenähert werden, solange diese klein sind:

$$\mathbf{C}_n^{\hat{n}} \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} = \mathbf{I} + [\Delta\vec{\psi} \times] \quad \text{mit} \quad (4.72)$$

$$\Delta\vec{\psi} = \vec{\psi}_n^{\hat{n}} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \quad (4.73)$$

$[\Delta\vec{\psi} \times]$ ist die kreuzproduktbildende Matrix (B.9). Die Winkel α, β, γ beschreiben kleine Drehungen um die drei Achsen des Navigationskoordinatensystems (Norden, Osten, unten). γ ist damit auch der Gierfehler $\Delta\psi$.

Die Beschleunigungs-Bias-Fehler $\Delta\vec{b}_a$ und Drehraten-Bias-Fehler $\Delta\vec{b}_\omega$ sind wie die Geschwindigkeitsfehler lineare Abweichungsgrößen vom Gesamtzustand:

$$\Delta\vec{b}_a = \hat{\vec{b}}_a - \vec{b}_a = \begin{pmatrix} \Delta b_{a,x} \\ \Delta b_{a,y} \\ \Delta b_{a,z} \end{pmatrix} \quad \Delta\vec{b}_\omega = \hat{\vec{b}}_\omega - \vec{b}_\omega = \begin{pmatrix} \Delta b_{\omega,x} \\ \Delta b_{\omega,y} \\ \Delta b_{\omega,z} \end{pmatrix} \quad (4.74)$$

Der Schätzfehler-Zustand $\Delta\vec{x}$ für (4.68) lautet damit:

$$\Delta\vec{x} = \left(\Delta\vec{p}^T, \Delta\vec{v}^T, \Delta\vec{\psi}^T, \Delta\vec{b}_a^T, \Delta\vec{b}_\omega^T \right)^T \quad (4.75)$$

Positionsfehler-Änderung

Für die Differentialgleichung des Positionsfehlers $\Delta\vec{p}$ (4.69) wird zunächst die Änderung des Fehlers des Breiten- und Längengrades sowie der Höhe aus der Linearisierung von (4.43) gebildet:

$$\begin{pmatrix} \Delta\dot{\varphi} \\ \Delta\dot{\lambda} \\ \Delta\dot{h} \end{pmatrix} = \frac{\partial\dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial\varphi, \lambda, h} \begin{pmatrix} \Delta\varphi \\ \Delta\lambda \\ \Delta h \end{pmatrix} + \frac{\partial\dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial\vec{v}_{eb}^n} \Delta\vec{v} \quad \text{mit} \quad (4.76)$$

$$\frac{\partial\dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial\varphi, \lambda, h} = \begin{pmatrix} 0 & 0 & \frac{-v_{eb,n}^n}{(R_n+h)^2} \\ \frac{v_{eb,e}^n \sin\varphi}{(R_e+h) \cos^2\varphi} & 0 & \frac{-v_{eb,e}^n}{(R_e+h)^2 \cos\varphi} \\ 0 & 0 & 0 \end{pmatrix} \quad (4.77)$$

$$\frac{\partial \dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial \vec{v}_{eb}^n} = \begin{pmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{1}{(R_e+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (4.78)$$

Dabei wird angenommen, dass die Krümmungsradien R_n und R_e für kleine Änderungen von φ konstant sind, vgl. oben.

Für den Zusammenhang zwischen Breitengrad-, Längengrad- sowie Höhenfehler und $\Delta \vec{p}$ aus (4.69) und für den Zusammenhang zwischen deren Änderungen gilt:

$$\begin{pmatrix} \Delta \dot{\varphi} \\ \Delta \dot{\lambda} \\ \Delta \dot{h} \end{pmatrix} = \frac{\partial \dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial \vec{p}_U^n} \Delta \dot{\vec{p}} \quad \text{mit} \quad \frac{\partial \dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial \vec{p}_U^n} = \frac{\partial \dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial \vec{v}_{eb}^n} \quad (4.79)$$

$$\begin{pmatrix} \Delta \varphi \\ \Delta \lambda \\ \Delta h \end{pmatrix} = \frac{\partial \varphi, \lambda, h}{\partial \vec{p}_U^n} \Delta \vec{p} \quad \text{mit} \quad \frac{\partial \varphi, \lambda, h}{\partial \vec{p}_U^n} = \frac{\partial \dot{\varphi}, \dot{\lambda}, \dot{h}}{\partial \vec{v}_{eb}^n} \quad (4.80)$$

Damit ergibt sich aus (4.76) für die Änderung des Positionsfehlers $\Delta \vec{p}$:

$$\Delta \dot{\vec{p}} = \underbrace{\begin{pmatrix} 0 & 0 & \frac{v_{eb,n}^n}{R_n+h} \\ \frac{v_{eb,e}^n \tan\varphi}{R_n+h} & 0 & \frac{v_{eb,e}^n}{R_e+h} \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{F}_{11}} \Delta \vec{p} + \underbrace{\mathbf{I}}_{\mathbf{F}_{12}} \Delta \vec{v} \quad (4.81)$$

Geschwindigkeitsfehler-Änderung

Zur Berechnung der Änderung des Geschwindigkeitsfehlers $\Delta \vec{v}$ (4.70) wird (4.41) linearisiert, vgl. [84]:

$$\begin{aligned} \Delta \dot{\vec{v}} = \Delta \dot{\vec{v}}_{eb}^n &= \underbrace{[\vec{v}_{eb}^n \times] \left(2 \frac{\partial \vec{\omega}_{ie}^n}{\partial \varphi, \lambda, h} + \frac{\partial \vec{\omega}_{en}^n}{\partial \varphi, \lambda, h} \right) \frac{\partial \varphi, \lambda, h}{\partial \vec{p}_U^n}}_{\mathbf{F}_{21}} \Delta \vec{p} \\ &+ \underbrace{\left([\vec{v}_{eb}^n \times] \frac{\partial \vec{\omega}_{en}^n}{\partial \vec{v}_{eb}^n} - 2 [\vec{\omega}_{ie}^n \times] - [\vec{\omega}_{en}^n \times] \right)}_{\mathbf{F}_{22}} \Delta \vec{v} - \underbrace{\left[(\mathbf{C}_b^{\hat{n}} \vec{f}_{ib}^b) \times \right]}_{\mathbf{F}_{23}} \Delta \vec{\psi} \\ &\underbrace{- \mathbf{C}_b^{\hat{n}} \Delta \vec{b}_a}_{\mathbf{F}_{24}} + \underbrace{\Delta \vec{g}_l^n}_{\approx 0} + \underbrace{\mathbf{C}_b^{\hat{n}} \vec{n}_a}_{\mathbf{G}_{21}} \end{aligned} \quad (4.82)$$

Der Schwerkraftfehler $\Delta \vec{g}_l^n$ kann vernachlässigt werden. \vec{f}_{ib}^b in \mathbf{F}_{23} ist die Bias-korrigierte spezifische Kraft, vgl. Schätzwert $\hat{\vec{f}}_{ib}^b$ in (4.61). Für \mathbf{F}_{24} und \mathbf{G}_{21} wird dabei das von (4.42) abgeleitete Beschleunigungssensor-Fehlermodell genutzt:

$$\Delta \vec{f}_{ib}^b = -\Delta \vec{b}_a + \vec{n}_a \quad (4.83)$$

Für die in (4.82) vorkommenden Ableitungsmatrizen von Erddrehrate (4.33) und Transportrate (4.34) gilt:

$$\frac{\partial \vec{\omega}_{ie}^n}{\partial \varphi, \lambda, h} = \begin{pmatrix} -\Omega \sin \varphi & 0 & 0 \\ 0 & 0 & 0 \\ -\Omega \cos \varphi & 0 & 0 \end{pmatrix} \quad \frac{\partial \vec{\omega}_{en}^n}{\partial \varphi, \lambda, h} = \begin{pmatrix} 0 & 0 & \frac{-v_{eb,e}^n}{(R_e+h)^2} \\ 0 & 0 & \frac{v_{eb,n}^n}{(R_n+h)^2} \\ \frac{-v_{eb,e}^n}{(R_e+h) \cos^2 \varphi} & 0 & \frac{v_{eb,e}^n \tan \varphi}{(R_e+h)^2} \end{pmatrix} \quad (4.84)$$

$$\frac{\partial \vec{\omega}_{ie}^n}{\partial \vec{v}_{eb}^n} = 0 \quad \frac{\partial \vec{\omega}_{en}^n}{\partial \vec{v}_{eb}^n} = \begin{pmatrix} 0 & \frac{1}{R_e+h} & 0 \\ -\frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{\tan \varphi}{R_e+h} & 0 \end{pmatrix} \quad (4.85)$$

Dabei wird wieder angenommen, dass die Krümmungsradien R_n und R_e für kleine Änderungen von φ konstant sind.

Lagefehler-Änderung

Die Änderung des Lagefehlers $\Delta \vec{\psi}$ (4.73) lässt sich aus (2.19) und (4.72) herleiten [84]:

$$\Delta \dot{\vec{\psi}} = -[\vec{\omega}_{in}^n \times] \Delta \vec{\psi} - (\Delta \vec{\omega}_{ie}^n + \Delta \vec{\omega}_{en}^n) + \mathbf{C}_b^{\hat{n}} \Delta \vec{\omega}_{ib}^b \quad (4.86)$$

Dies lässt sich unter Nutzung des aus (4.32) folgenden Drehratensensor-Fehlermodells

$$\Delta \vec{\omega}_{ib}^b = -\Delta \vec{b}_\omega + \vec{n}_\omega \quad (4.87)$$

und durch Rückführen von $\Delta \vec{\omega}_{ie}^n + \Delta \vec{\omega}_{en}^n$ auf Positions- und Geschwindigkeitsfehler umformen zu:

$$\begin{aligned} \Delta \dot{\vec{\psi}} = & - \underbrace{\left(\frac{\partial \vec{\omega}_{ie}^n}{\partial \varphi, \lambda, h} + \frac{\partial \vec{\omega}_{en}^n}{\partial \varphi, \lambda, h} \right) \frac{\partial \varphi, \lambda, h}{\partial \vec{p}}}_{\mathbf{F}_{31}} \Delta \vec{p} - \underbrace{\frac{\partial \vec{\omega}_{en}^n}{\partial \vec{v}_{eb}^n}}_{\mathbf{F}_{32}} \Delta \vec{v} \\ & - \underbrace{[(\vec{\omega}_{ie}^n + \vec{\omega}_{en}^n) \times]}_{\mathbf{F}_{33}} \Delta \vec{\psi} - \underbrace{\mathbf{C}_b^{\hat{n}}}_{\mathbf{F}_{35}} \Delta \vec{b}_\omega + \underbrace{\mathbf{C}_b^{\hat{n}}}_{\mathbf{G}_{32}} \vec{n}_\omega \end{aligned} \quad (4.88)$$

Für die Ableitungsmatrizen in \mathbf{F}_{31} und \mathbf{F}_{32} siehe (4.84)f.

Biasfehler-Änderung

Das Modell für die Änderungen der Sensor-Bias-Schätzfehler ergibt sich aus (4.28)f.:

$$\Delta \vec{b}_a = \underbrace{\mathbf{I}}_{\mathbf{G}_{43}} \vec{n}_{b_a} \quad \Delta \vec{b}_\omega = \underbrace{\mathbf{I}}_{\mathbf{G}_{54}} \vec{n}_{b_\omega} \quad (4.89)$$

Kontinuierliches Schätzfehlermodell

Damit ergibt sich aus den Gleichungen (4.81), (4.82), (4.88) und (4.89) das gesuchte kontinuierliche, lineare Schätzfehlermodell in der Form (4.68):

$$\underbrace{\begin{pmatrix} \Delta \dot{\vec{p}} \\ \Delta \dot{\vec{v}} \\ \Delta \dot{\vec{\psi}} \\ \Delta \dot{\vec{b}}_a \\ \Delta \dot{\vec{b}}_\omega \end{pmatrix}}_{\Delta \dot{\vec{x}}} = \underbrace{\begin{pmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \mathbf{F}_{23} & \mathbf{F}_{24} & \mathbf{0} \\ \mathbf{F}_{31} & \mathbf{F}_{32} & \mathbf{F}_{33} & \mathbf{0} & \mathbf{F}_{35} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\mathbf{F}} \underbrace{\begin{pmatrix} \Delta \vec{p} \\ \Delta \vec{v} \\ \Delta \vec{\psi} \\ \Delta \vec{b}_a \\ \Delta \vec{b}_\omega \end{pmatrix}}_{\Delta \vec{x}} + \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{32} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{43} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{54} \end{pmatrix}}_{\mathbf{G}} \underbrace{\begin{pmatrix} \vec{n}_a \\ \vec{n}_\omega \\ \vec{n}_{b_a} \\ \vec{n}_{b_\omega} \end{pmatrix}}_{\vec{w}} \quad (4.90)$$

Die Zahlenwerte der Matrizen $\mathbf{F}(t)$ und $\mathbf{G}(t)$ zu einem Zeitpunkt $t = kT_s$ können aus dem Zustandsschätzwert $\hat{\vec{x}}_{k-1}$ und dem Eingang \vec{u}_k berechnet werden. Für die spektralen Leistungsdichten \mathbf{Q} des kontinuierlichen Systemrauschens $\vec{w}(t)$ siehe (4.46).

Zeitdiskretes Schätzfehlermodell

Zur Bestimmung des zeitdiskreten, linearen Schätzfehlermodells (2.50)

$$\Delta \vec{x}_{k+1} = \Phi_k \Delta \vec{x}_k + \mathbf{G}_k \vec{w}_k$$

für die Propagation der Schätzfehlerkovarianzmatrix \mathbf{P}_k in (4.67) wird das kontinuierliche Modell (4.90) diskretisiert. Mit \mathbf{F} und \mathbf{G} aus (4.90) lassen sich mit (2.69)f. in jedem Zeitschritt die Systemmatrizen Φ_k und \mathbf{G}_k des linearen, zeitdiskreten Schätzfehlermodells berechnen.

Die Kovarianzmatrix \mathbf{Q}_k des diskreten Systemrauschens \vec{w}_k wird gemäß (2.75) aus den spektralen Leistungsdichten \mathbf{Q} des kontinuierlichen Systemrauschens aus (4.46) berechnet.

4.5 Estimation

In der Estimation des Erweiterten Kalman-Filters (siehe Abschnitt 2.3.2) wird der propagierte Schätzwert des Systemzustandes $\hat{\vec{x}}_k^-$ und die Schätzfehlerkovarianzmatrix \mathbf{P}_k^- mit den Messgrößen korrigiert. Dazu stehen als Messgrößengruppen die Positions- und Geschwindigkeitsmessung des GNSS-Empfängers sowie die Magnetometermessung zur Verfügung. Zudem können die Beschleunigungssensorsignale zur Stützung der Lage verwendet werden.

Zunächst wird auf Basis des geschätzten Systemzustandes $\hat{\vec{x}}_k$ (4.48) gemäß (2.54) mit einem Messmodell die Messgröße \hat{y}_k prädiziert. Das nichtlineare Messmodell einer Messgrößengruppe q hat gemäß (2.44) die Form:

$$\vec{y}_{q,k} = \vec{h}_{q,k}(\vec{x}_k) + \vec{v}_{q,k} \quad (4.91)$$

Aus der Residualmessgröße $\Delta\vec{y}_{q,k} = \hat{y}_k - \vec{y}_k$, der Differenz zwischen geschätzter und realer Messgröße, wird dann mit der Kalman-Gain-Matrix $\mathbf{K}_{q,k}$ pro Messgrößengruppe eine Korrektur $\Delta\hat{\vec{x}}_{q,k}^+$ berechnet (2.56), mit der der Schätzwert des Systemzustandes zu $\hat{\vec{x}}_k^+$ upgedated wird (2.57).

Zur Berechnung von \mathbf{K}_k (2.55) und zum Update von \mathbf{P}_k (2.58) werden zudem die Kovarianzmatrix $\mathbf{R}_{q,k}$ (2.31) des diskreten Messrauschens $\vec{v}_{q,k}$ und die Messmatrix $\mathbf{H}_{q,k}$ des linearisierten Messmodells gemäß (2.51)

$$\Delta\vec{y}_{q,k} = \mathbf{H}_{q,k}\Delta\vec{x}_k + \vec{v}_{q,k} \quad (4.92)$$

benötigt. $\Delta\vec{x}_k$ ist der Schätzfehler-Zustand aus (4.75).

Im Folgenden werden die Messmodelle (4.91) und (4.92), die Residualmessgrößen $\Delta\vec{y}_{q,k}$, die Messmatrizen $\mathbf{H}_{q,k}$ und die Kovarianzmatrizen $\mathbf{R}_{q,k}$ für die einzelnen Messgrößengruppen q eingeführt.

Das Update des geschätzten Systemzustandes und der Schätzfehlerkovarianzmatrix wird dann in Abschnitt 4.5.2 dargestellt.

4.5.1 Messmodelle

4.5.1.1 Positionsmessung GNSS

Der GNSS-Empfänger liefert Messungen der Position der GNSS-Antenne in WGS84-Koordinaten als Breitengrad, Längengrad und Höhe $\vec{p}_A^w = (\tilde{\varphi}_A, \tilde{\lambda}_A, \tilde{h}_A)^T$. Die Sensorfusion schätzt die Position des Punktes U : $\hat{\vec{p}}_U^w = (\hat{\varphi}, \hat{\lambda}, \hat{h})^T$ (4.25). Der Vektor vom Punkt U zur Position der GNSS-Antenne A soll mit \vec{d}_A bezeichnet werden:

$$\vec{p}_A = \vec{p}_U + \vec{d}_A \quad (4.93)$$

Im Allgemeinen ist $|\vec{d}_A|$ klein (siehe Tabelle C.1). Damit lässt sich unter Nutzung von (4.80) die Antennenposition als Funktion des Systemzustandes \vec{x} und \vec{d}_A ausdrücken:

$$\begin{pmatrix} \varphi_A \\ \lambda_A \\ h_A \end{pmatrix} = \begin{pmatrix} \varphi \\ \lambda \\ h \end{pmatrix} + \begin{pmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{1}{(R_e+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{C}_b^n \vec{d}_A^b \quad (4.94)$$

Als nichtlineares Messmodell (4.91) wird dieser Zusammenhang jedoch in den metrischen, kartesischen Koordinaten des Navigationskoordinatensystems ausgedrückt:

$$\vec{y}_{p,k} = \vec{p}_{A,k}^n = \vec{p}_{U,k}^n + \mathbf{C}_{b,k}^n \vec{d}_A^b + \vec{v}_{p,k} \quad (4.95)$$

$\vec{v}_{p,k}$ ist das diskrete Rauschen der Positions-Messungen, formuliert in den metrischen, kartesischen Koordinaten, mit der Kovarianzmatrix $\mathbf{R}_{p,k}$ (2.31). Für $\mathbf{R}_{p,k}$ wird die vom GNSS-Empfänger geschätzte Kovarianzmatrix $\mathbf{R}_{\hat{p},k}$ (4.20) genutzt:

$$\mathbf{R}_{p,k} = \mathbf{R}_{\hat{p},k} \quad (4.96)$$

Die Residualmessgröße $\Delta \vec{y}_{p,k}$ für das Erweiterte Kalman-Filter in (2.56) ist die Differenz zwischen erwarteter Antennenposition $\hat{p}_{A,k}^n$ aus der aktuellen Zustandsschätzung gemäß (4.95) und der gemessenen Antennenposition $\tilde{p}_{A,k}^n$:

$$\Delta \vec{y}_{p,k} = \hat{p}_{A,k}^n - \tilde{p}_{A,k}^n = \hat{p}_{U,k}^n + \mathbf{C}_{b,k}^{\hat{n}} \vec{d}_A^b - \tilde{p}_{A,k}^n \quad (4.97)$$

Dies lässt sich mit (4.94) umformen zu:

$$\Delta \vec{y}_{p,k} = \begin{pmatrix} (\hat{\varphi}_k - \tilde{\varphi}_{A,k})(R_n + \hat{h}_k) \\ (\hat{\lambda}_k - \tilde{\lambda}_{A,k})(R_e + \hat{h}_k) \cos \hat{\varphi}_k \\ -(\hat{h}_k - \tilde{h}_{A,k}) \end{pmatrix} + \mathbf{C}_{b,k}^{\hat{n}} \vec{d}_A^b \quad (4.98)$$

Zur Bildung der Messmatrix wird der Schätzfehler der Antennenposition $\Delta \vec{y}_{p,k} = \Delta \vec{p}_{A,k}^n$ als Differenz zwischen geschätzter und tatsächlicher Antennenposition in Abhängigkeit vom Schätzfehler-Zustand $\Delta \vec{x}_k$ (4.75) unter Nutzung von (4.72) gebildet:

$$\begin{aligned} \Delta \vec{p}_A^n &= \hat{p}_A^n - \tilde{p}_A^n = \hat{p}_U^n + \mathbf{C}_b^{\hat{n}} \vec{d}_A^b - \tilde{p}_U^n - \mathbf{C}_b^n \vec{d}_A^b = \Delta \vec{p} + (\mathbf{I} - \mathbf{C}_n^{\hat{n},T}) \mathbf{C}_b^{\hat{n}} \vec{d}_A^b \\ &\approx \Delta \vec{p} + [\Delta \vec{\psi} \times] \mathbf{C}_b^{\hat{n}} \vec{d}_A^b = \Delta \vec{p} - [\mathbf{C}_b^{\hat{n}} \vec{d}_A^b \times] \Delta \vec{\psi} \end{aligned} \quad (4.99)$$

Damit ergibt sich die Messmatrix $\mathbf{H}_{p,k}$ für das lineare Messmodell $\Delta \vec{y}_{p,k} = \mathbf{H}_{p,k} \Delta \vec{x}_k + \vec{v}_{p,k}$ gemäß (4.92):

$$\mathbf{H}_{p,k} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & -[\mathbf{C}_b^{\hat{n}} \vec{d}_A^b \times] & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (4.100)$$

Hier werden die Positions- (und Geschwindigkeits-)Messungen des GNSS-Empfängers als Messgrößen des Erweiterten Kalman-Filters verwendet. Ein solcher Ansatz wird als „loosely coupled“ bezeichnet [84]. Wenn stattdessen die Pseudorange- und Deltarange-Messungen (siehe Abschnitte 2.5.2.5 und 2.5.2.7) des GNSS-Empfängers im Erweiterten Kalman-Filter verwendet werden, wird dies „tightly coupled“ genannt. Eine entsprechende Implementierung ist z. B. in [44] dargestellt.

4.5.1.2 Geschwindigkeitsmessung GNSS

Neben der Position liefert der GNSS-Empfänger auch Messungen der Geschwindigkeit der GNSS-Antenne \tilde{v}_{eA}^n .

Die Geschwindigkeit an der Antennenposition lässt sich aus dem Systemzustand \vec{x} berechnen:

$$\vec{v}_{eA}^n = \vec{v}_{eb}^n + \mathbf{C}_b^n \left(\vec{\omega}_{eb}^b \times \vec{d}_A^b \right) \quad (4.101)$$

Die Erddrehrate $\vec{\omega}_{ie}$ soll hier vernachlässigt werden, sodass $\vec{\omega}_{eb}^b \approx \vec{\omega}_{ib}^b$ gilt. Als Messmodell (2.44) für die Geschwindigkeit ergibt sich damit:

$$\vec{y}_{v,k} = \vec{v}_{eA}^n \approx \vec{v}_{eb}^n + \mathbf{C}_b^n \left(\vec{\omega}_{ib}^b \times \vec{d}_A^b \right) + \vec{v}_{v,k} \quad (4.102)$$

Das diskrete Rauschen $\vec{v}_{v,k}$ der Geschwindigkeits-Messungen wird durch die Kovarianzmatrix $\mathbf{R}_{v,k}$ (2.31) beschrieben. Diese wird vom GNSS-Empfänger ebenfalls in geeigneter Form geschätzt, siehe (4.21):

$$\mathbf{R}_{v,k} = \mathbf{R}_{\tilde{v},k} \quad (4.103)$$

Die Differenz zwischen erwarteter Antennengeschwindigkeit $\hat{v}_{eA,k}^n$, basierend auf der aktuellen Zustandsschätzung, und der gemessenen Antennengeschwindigkeit \tilde{v}_{eA}^n ist die Residualmessgröße $\Delta\tilde{y}_{v,k}$ für das Erweiterte Kalman-Filter (2.56):

$$\Delta\tilde{y}_{v,k} = \hat{v}_{eA,k}^n - \tilde{v}_{eA,k}^n \approx \hat{v}_{eb,k}^n + \mathbf{C}_{b,k}^{\hat{n}} \left(\hat{\omega}_{ib,k}^b \times \vec{d}_A^b \right) - \tilde{v}_{eA,k}^n \quad (4.104)$$

Der Schätzfehler für die Antennengeschwindigkeit $\Delta\vec{y}_{v,k} = \Delta\vec{v}_{eA}^n$ ist die Differenz zwischen geschätzter und tatsächlicher Antennengeschwindigkeit. Ausgedrückt in Abhängigkeit vom Schätzfehler-Zustand $\Delta\vec{x}_k$ (4.75) ergibt sich [84]:

$$\Delta\vec{v}_{eA}^n = \hat{v}_{eA}^n - \vec{v}_{eA}^n \approx \Delta\vec{v} - \left[\mathbf{C}_b^{\hat{n}} \left[\hat{\omega}_{ib}^b \times \right] \vec{d}_A^b \times \right] \Delta\vec{\psi} + \mathbf{C}_b^{\hat{n}} \left[\vec{d}_A^b \times \right] \Delta\vec{b}_\omega \quad (4.105)$$

Daraus lässt sich die Geschwindigkeits-Messmatrix $\mathbf{H}_{v,k}$ für das lineare Messmodell $\Delta\vec{y}_{v,k} = \mathbf{H}_{v,k} \Delta\vec{x}_k + \vec{v}_{v,k}$ gemäß (4.92) ablesen:

$$\mathbf{H}_{v,k} = \left(\mathbf{0} \quad \mathbf{I} \quad - \left[\mathbf{C}_{b,k}^{\hat{n}} \left[\hat{\omega}_{ib,k}^b \times \right] \vec{d}_A^b \times \right] \quad \mathbf{0} \quad \mathbf{C}_{b,k}^{\hat{n}} \left[\vec{d}_A^b \times \right] \right) \quad (4.106)$$

4.5.1.3 Magnetometermessung

Das 3-Achs-Magnetometer (siehe Abschnitt 4.1.3) kann zu einer Bestimmung des Gierwinkels verwendet werden. Es misst das lokale Magnetfeld \vec{m} in körperfesten Koordinaten und ein Messrauschen \vec{n}_m :

$$\vec{m}^b = \mathbf{C}_b^{n,T} \vec{m}^n + \vec{n}_m \quad (4.107)$$

Es wird angenommen, dass äußere Störer nicht zu stark sind und statische, körperfeste Störungen durch eine Kalibrierung des Magnetometers kompensiert werden, sodass das lokale Magnetfeld durch das Erdmagnetfeld dominiert wird, vgl. Abschnitt 4.1.3.

Die Orientierung des Quadropters kann mit dem Magnetometer nicht vollständig bestimmt werden, da der Drehwinkel um die Achse, die parallel zum Magnetfeldvektor \vec{m} verläuft, nicht beobachtbar ist. Die anderen beiden Winkel gegenüber dem Magnetfeldvektor sind bestimmbar. Mit den von der Sensorfusion geschätzten Nick- und Rollwinkeln gegenüber der Horizontalen sind diese beiden Winkel der Gierwinkel in der Horizontalebene und der Drehwinkel um eine in östliche Richtung zeigende Achse. Für Letzteren muss jedoch die lokale Inklination ausreichend genau bekannt sein. Hier soll aus der Magnetometermessung nur der Gierwinkel bestimmt werden. Dazu wird die lokale Deklination benötigt.

Als Messgröße wird hier der Gierwinkel ψ verwendet. Das Messmodell (2.44) als Funktion des Systemzustandes \vec{x}_k lautet damit:

$$y_{m,k} = \psi_k + v_{m,k} \quad (4.108)$$

Der Gierwinkel ψ_k ist gemäß (2.16a), (2.13) eine Funktion des Zustandes $\mathbf{q}_{b,k}^n$. Der Rauschterm \vec{n}_m aus (4.107) wird hier als mittelwertfreies, diskretes weißes Winkelrauschen $v_{m,k}$ mit der Varianz $\sigma_{m,k}^2$ (A.31) und der Kovarianzmatrix $\mathbf{R}_{m,k}$ ausgedrückt:

$$\mathbf{R}_{m,k} = \sigma_{m,k}^2 \quad (4.109)$$

Um die körperfeste Magnetometer-Messung \tilde{m}_b zu horizontieren, wird ein Hilfskoordinatensystem H eingeführt, mit dem die Richtungskosinusmatrix \mathbf{C}_b^n in zwei Rotationen aufgeteilt wird: $\mathbf{C}_b^n = \mathbf{C}_h^n \mathbf{C}_b^h$. Gemäß der Eulerwinkeldefinition auf Seite 28 soll die Rotation um den Gierwinkel ψ durch \mathbf{C}_h^n sowie die Rotation um Nick- und Rollwinkel θ , φ durch \mathbf{C}_b^h darstellt werden. Damit lautet die horizontierte Magnetometer-Messung:

$$\begin{pmatrix} \tilde{m}_x^h \\ \tilde{m}_y^h \\ \tilde{m}_d^h \end{pmatrix} = \tilde{m}^h = \mathbf{C}_b^h \tilde{m}^b \quad (4.110)$$

mit \mathbf{C}_b^h gemäß (B.4) aus dem von der Sensorfusion geschätzten Nick- und Rollwinkel:

$$\mathbf{C}_b^h = \begin{pmatrix} \cos \hat{\theta} & \sin \hat{\varphi} \sin \hat{\theta} & \cos \hat{\varphi} \sin \hat{\theta} \\ 0 & \cos \hat{\varphi} & -\sin \hat{\varphi} \\ -\sin \hat{\theta} & \cos \hat{\theta} \sin \hat{\varphi} & \cos \hat{\varphi} \cos \hat{\theta} \end{pmatrix} \quad (4.111)$$

Der Arkustangens der horizontierten Komponenten \tilde{m}_x^h und \tilde{m}_y^h enthält den vom Magnetometer gemessenen „magnetischen“ Gierwinkel. Um die magnetische Missweisung zu

kompensieren, muss die lokale Deklination δ_m addiert werden. Der vom Magnetometer gemessene Gierwinkel lautet damit:

$$\tilde{\psi}_k = \arctan2(-\tilde{m}_{y,k}^h, \tilde{m}_{x,k}^h) + \delta_m + v_{m,k} \quad (4.112)$$

Die Residualmessgröße $\Delta\tilde{y}_{m,k}$ für das Erweiterte Kalman-Filter ist die Differenz zwischen erwartetem und gemessenem Gierwinkel:

$$\Delta\tilde{y}_{m,k} = \hat{\psi}_k - \tilde{\psi}_k = \hat{\psi}_k - \arctan2(\tilde{m}_{y,k}^h, \tilde{m}_{x,k}^h) - \hat{\delta}_m \quad (4.113)$$

$\hat{\delta}_m$ ist die geschätzte lokale, magnetische Deklination.

Der Gierwinkel-Schätzfehler $\Delta y_{m,k} = \Delta\psi_k$ ist gemäß (4.72) der Schätzfehler-Zustand γ :

$$\Delta y_{m,k} = \hat{\psi}_k - \psi_k = \gamma_k \quad (4.114)$$

Damit lautet die Messmatrix $\mathbf{H}_{m,k}$ für das lineare Messmodell $\Delta y_{m,k} = \mathbf{H}_{m,k} \Delta \vec{x}_k + v_{m,k}$:

$$\mathbf{H}_{m,k} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.115)$$

Bei der Wahl des Zahlenwertes für die Standardabweichung $\sigma_{m,k}$ zur Parametrierung von $\mathbf{R}_{m,k}$ (4.109) ist zu berücksichtigen, dass die Magnetometer-Messung zeitweise durch magnetische Störungen stark abweichen kann. Die Mittelwertfreiheit von $v_{m,k}$ ist dann nicht mehr gegeben. Wenn die Magnetometer-Messung in der Estimation verwendet werden soll, muss daher $\sigma_{m,k}$ recht groß gewählt werden (siehe Tabelle C.2), um die restliche Sensorfusion im Störungsfall nicht zu stark zu beeinflussen.

4.5.1.4 Messung Schwerkrafttrichtung mit Beschleunigungssensoren

Zur Stützung der Lage könnte als weitere Messgröße die auf Basis der Beschleunigungssensorsignale berechnete Richtung der Schwerkraft genutzt werden, da diese stets nach unten, parallel zum Basisvektor \vec{n}_d des Navigationskoordinatensystems, wirkt, vgl. (2.10). In Ruhe auf der Erdoberfläche bzw. im Schwebeflug sind die von den Beschleunigungssensoren gemessenen Bias-korrigierten spezifischen Kräfte gleich der negativen Schwerebeschleunigung \vec{g}_l (2.9) (vgl. auch (4.131)). Im Allgemeinen verfälschen jedoch die Beschleunigungen, die während des Fluges auftreten, die Schwerkraftmessung, vgl. (4.3) mit (2.9), (2.8). Daher wird hier auf die Verwendung der Schwerkrafttrichtung als Messgröße verzichtet.

4.5.2 Update

Im vorhergehenden Abschnitt wurden für die einzelnen Messgrößengruppen die Residualmessgrößen $\Delta\tilde{y}_{q,k}$, die Messmatrizen $\mathbf{H}_{q,k}$ und die Kovarianzmatrizen $\mathbf{R}_{q,k}$ hergeleitet.

Damit können jetzt der Schätzwert des Systemzustandes \hat{x}_k und die Schätzfehlerkovarianzmatrix \mathbf{P}_k gemäß (2.55) bis (2.58) korrigiert werden.

Die Korrektur kann für jede Messgrößengruppe getrennt erfolgen, wenn die Gruppen unkorreliert sind [28, 84]. Dass GNSS-Messungen und Magnetometermessungen nicht miteinander korreliert sind, ist offensichtlich. Für die GNSS-Geschwindigkeiten und -Positionen wird das im Allgemeinen auch angenommen. Die Gesamtmessmatrix \mathbf{R}_k hat somit Diagonalfom. Die getrennte Verarbeitung der Messgrößengruppen hat den Vorteil, dass die Dimensionen der benötigten Matrixoperationen verringert werden. So ist die Dimension der zu invertierenden Matrix zur Berechnung der Kalman-Gain-Matrix in (2.55) deutlich kleiner. Zudem treten die GNSS-Messungen nicht synchron zu den Magnetometermessungen auf, sodass für eine gemeinsame Verarbeitung alle Permutationen der Messgrößengruppen implementiert werden müssten. Bei der getrennten Verarbeitung der Gruppe wird die jeweilige Korrektur nur ausgeführt, wenn neue Messdaten vorliegen.

Mit der Messmatrix $\mathbf{H}_{q,k}$ und Kovarianzmatrix $\mathbf{R}_{q,k}$ einer Messgrößengruppe q sowie der propagierten Schätzfehlerkovarianzmatrix \mathbf{P}_k^- kann die jeweilige Kalman-Gain-Matrix $\mathbf{K}_{q,k}$ (2.55) berechnet werden:

$$\mathbf{K}_{q,k} = \mathbf{P}_k^- \mathbf{H}_{q,k}^T \left(\mathbf{H}_{q,k} \mathbf{P}_k^- \mathbf{H}_{q,k}^T + \mathbf{R}_{q,k} \right)^{-1} \quad q \in \{p, v, m\} \quad (4.116)$$

Der Schätzfehlerzustand $\Delta \vec{x}_{q,k}$ (4.75) wird gemäß (2.56) mit $\mathbf{K}_{q,k}$ und der Residualmessgröße $\Delta \vec{y}_{q,k}$ der Messgrößengruppe geschätzt:

$$\Delta \hat{x}_{q,k}^+ = \mathbf{K}_{q,k} \Delta \vec{y}_{q,k} \quad q \in \{p, v, m\} \quad (4.117)$$

Mit $\Delta \hat{x}_{q,k}^+$ kann dann für jede Messgrößengruppe q der geschätzte Systemzustand \hat{x}_k (4.48) gemäß (2.57) korrigiert werden:

$$\hat{x}_k^+ = \vec{f}_c(\hat{x}_k^-, \Delta \hat{x}_{q,k}^+) \quad q \in \{p, v, m\} \quad (4.118)$$

Die dazu notwendige Transformation \vec{f}_c (2.49) wird im Folgenden für die einzelnen Elemente des Schätzfehlerzustandes $\Delta \hat{x}_{q,k}^+$ (4.75) getrennt dargestellt:

$$\left(\Delta \hat{p}_{q,k}^{+T}, \Delta \hat{v}_{q,k}^{+T}, \Delta \hat{\psi}_{q,k}^{+T}, \Delta \hat{b}_{a,q,k}^{+T}, \Delta \hat{b}_{\omega,q,k}^{+T} \right)^T = \Delta \hat{x}_{q,k}^+ \quad q \in \{p, v, m\} \quad (4.119)$$

Der Index q wird im Folgenden weggelassen.

Aus der Positionskorrektur $\Delta \hat{p}_k^+$ wird gemäß (4.80), (4.78) die Korrektur von Breiten- und Längengrad sowie Höhe berechnet und auf den Gesamtzustand angewendet:

$$\begin{pmatrix} \hat{\varphi}_k^+ \\ \hat{\lambda}_k^+ \\ \hat{h}_k^+ \end{pmatrix} = \begin{pmatrix} \hat{\varphi}_k^- \\ \hat{\lambda}_k^- \\ \hat{h}_k^- \end{pmatrix} - \begin{pmatrix} \Delta \hat{\varphi}_k^+ \\ \Delta \hat{\lambda}_k^+ \\ \Delta \hat{h}_k^+ \end{pmatrix} \quad \text{mit} \quad \begin{pmatrix} \Delta \hat{\varphi}_k^+ \\ \Delta \hat{\lambda}_k^+ \\ \Delta \hat{h}_k^+ \end{pmatrix} = \begin{pmatrix} \frac{1}{R_n + \hat{h}_k^-} & 0 & 0 \\ 0 & \frac{1}{(R_e + \hat{h}_k^-) \cos \hat{\varphi}_k^-} & 0 \\ 0 & 0 & -1 \end{pmatrix} \Delta \hat{p}_k^+ \quad (4.120)$$

Das Lagequaternion $\hat{\mathbf{q}}_{b,k}^n$ wird durch eine linksseitige Quaternionen-Multiplikation (B.1) mit dem Korrekturquaternion \mathbf{q}_c korrigiert, das gemäß (4.58) aus $\Delta\hat{\psi}_k^+$ berechnet wird:

$$\hat{\mathbf{q}}_{b,k}^{n+} = \mathbf{q}_c(-\Delta\hat{\psi}_k^+) \bullet \hat{\mathbf{q}}_{b,k}^{n-} \quad (4.121)$$

Zur Korrektur von Geschwindigkeit und Bias vereinfacht sich die Transformation \vec{f}_c zu einer einfachen Subtraktion $\hat{x}_k^+ = \hat{x}_k^- - \Delta\hat{x}_k^+$ gemäß (2.48), vgl. (4.70) und (4.74):

$$\hat{v}_{eb,k}^{n+} = \hat{v}_{eb,k}^{n-} - \Delta\hat{v}_k^+ \quad \hat{b}_{a,k}^+ = \hat{b}_{a,k}^- - \Delta\hat{b}_{a,k}^+ \quad \hat{b}_{\omega,k}^+ = \hat{b}_{\omega,k}^- - \Delta\hat{b}_{\omega,k}^+ \quad (4.122)$$

Mit (4.120), (4.121) und (4.122) ist das Update (4.118) des Systemzustandes \hat{x}_k zum A-posteriori-Schätzwert \hat{x}_k^+ vollständig beschrieben.

Aus dem korrigierten Systemzustand \hat{x}_k^+ können dann weitere benötigte Größen, wie z. B. die korrigierte Richtungskosinusmatrix $\mathbf{C}_{b,k}^{\hat{n}+}$ mit (2.13) aus $\hat{\mathbf{q}}_{b,k}^{n+}$ berechnet werden.

Für die korrigierten Schätzwerte der spezifischen Kraft und der Drehrate, \hat{f}_{ib}^{b+} und $\hat{\omega}_{ib}^{b+}$, bietet sich an, die A-priori-Schätzwerte aus (4.61) und (4.53) parallel zu (4.122) mit den Bias-Schätzfehlerzuständen zu korrigieren:

$$\hat{f}_{ib,k}^{b+} = \hat{f}_{ib,k}^{b-} - \Delta\hat{b}_{a,k}^+ \quad (4.123)$$

$$\hat{\omega}_{ib,k}^{b+} = \hat{\omega}_{ib,k}^{b-} - \Delta\hat{b}_{\omega,k}^+ \quad (4.124)$$

Der Schätzwert für die Beschleunigung des Quadropters ist damit gemäß (4.41):

$$\hat{a}_{eb,k}^{n+} = \mathbf{C}_{b,k}^{\hat{n}+} \hat{f}_{ib,k}^{b+} - \left(2\hat{\omega}_{ie,k}^n + \hat{\omega}_{en,k}^n \right) \times \hat{v}_{eb,k}^{n+} + \hat{g}_l^n \quad (4.125)$$

mit der geschätzten Schwerkraft \hat{g}_l^n aus (4.133).

Mit jedem Update des Systemzustandes \hat{x}_k in (4.118) wird das zugehörige Schätzfehlerkovarianzmatrix-Update (2.58) durchgeführt:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_{q,k} \mathbf{H}_{q,k}) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_{q,k} \mathbf{H}_{q,k})^T + \mathbf{K}_{q,k} \mathbf{R}_{q,k} \mathbf{K}_{q,k}^T \quad \mathbf{q} \in \{p, v, m\} \quad (4.126)$$

4.6 Ablauf

4.6.1 Initialisierung

Vor dem Start der Sensorfusion muss der Schätzwert des Systemzustandes \hat{x} (4.48) und die Schätzfehlerkovarianzmatrix \mathbf{P}_k (2.46) initialisiert werden. Dazu werden vor Abflug im Stillstand mit ausgeschalteten Motoren in einer Initialisierungsphase (siehe Abschnitt 5.4) Startwerte für die Position, Geschwindigkeit, Lage und die Sensor-Bias bestimmt.

Während der Initialisierungsphase, die $T_{\text{init}} = 5 \text{ sec}$ dauert, werden Mittelwerte der Messwerte für Drehrate und spezifische Kraft aus den Delta-Winkeln und Delta-Geschwindigkeiten (4.11) der IMU bestimmt:

$$\bar{\bar{\omega}}_{ib}^b = \frac{1}{N} \sum_{k=1}^N \frac{\Delta \tilde{\sigma}_{ib,k}^b}{T_s}, \quad \bar{\bar{f}}_{ib}^b = \frac{1}{N} \sum_{k=1}^N \frac{\Delta \tilde{V}_{ib,k}^b}{T_s} \quad \text{mit} \quad N = \frac{T_{\text{init}}}{T_s} \quad (4.127)$$

Bei der Bildung der Mittelwerte wird durch Überwachung der spezifischen Kraft $|\tilde{f}_{ib,k}^b|$ sichergestellt, dass der Quadrocopter während der Initialisierungsphase nicht bewegt wird.

Mit der mittleren gemessenen Drehrate $\bar{\bar{\omega}}_{ib}^b$ kann das initiale Drehraten-Bias $\hat{b}_{\omega,0}^b$ gemäß (4.5) bestimmt werden:

$$\hat{b}_{\omega,0}^b = \bar{\bar{\omega}}_{ib}^b - \bar{\omega}_{ib}^b = \bar{\bar{\omega}}_{ib}^b - \underbrace{(\bar{\omega}_{ie}^b)}_{\approx \vec{0}} + \underbrace{(\bar{\omega}_{eb}^b)}_{=\vec{0}} \approx \bar{\bar{\omega}}_{ib}^b \quad (4.128)$$

Die Drehrate gegenüber dem Inertialsystem $\bar{\omega}_{ib}^b$ ist im Stillstand gleich der Erddrehrate $\bar{\omega}_{ie}^b$ (2.7), die hier vernachlässigt werden kann.

Im Stillstand ist die Beschleunigung gegenüber dem Inertialsystem \bar{a}_{ib}^b gleich der Zentripetalbeschleunigung $-\bar{a}_Z$ (2.8) durch die Erdrotation. Daher gilt für das Beschleunigungssensor-Bias \bar{b}_a^b gemäß (4.6), (4.3), (2.9):

$$\bar{b}_a^b = \bar{\bar{f}}_{ib}^b - \bar{f}_{ib}^b = \bar{\bar{f}}_{ib}^b - \underbrace{(\bar{a}_{ib}^b)}_{-\bar{a}_Z} - \bar{g}^b = \bar{\bar{f}}_{ib}^b + \bar{g}_l^b = \bar{\bar{f}}_{ib}^b + \mathbf{C}_n^b \bar{g}_l^n \quad (4.129)$$

Um das Beschleunigungssensor-Bias auf diese Weise zu bestimmen, müssen sowohl die Lage \mathbf{C}_b^n als auch die lokale Schwerebeschleunigung \bar{g}_l^n (vgl. 2.1.2 Erdmodell) durch externe Referenzen genau bekannt sein. Da dies im Allgemeinen nicht der Fall ist, wird das Beschleunigungssensor-Bias mit null initialisiert:

$$\hat{b}_{a,0}^b \approx \vec{0} \quad (4.130)$$

Mit dieser Annahme kann nun der Mittelwert der gemessenen spezifischen Kraft $\bar{\bar{f}}_{ib}^b$ genutzt werden, um den Betrag der lokalen Schwerebeschleunigung $|\bar{g}_l|$ zu schätzen. Gemäß (4.129) gilt mit (4.130):

$$\hat{g}_{l,0} = \left| \hat{g}_{l,0}^n \right| \approx \left| \mathbf{C}_b^n \bar{\bar{f}}_{ib}^b \right| = \left| \bar{\bar{f}}_{ib}^b \right| \quad (4.131)$$

Die Schwerebeschleunigung \bar{g}_l hat in guter Näherung im Navigationskoordinatensystem nur eine vertikale Komponente (siehe Erdmodell in 2.1.2 und Schwerebeschleunigungs-

modell (2.10)). Sie wird daher geschätzt als

$$\hat{g}_{l,0}^n = \begin{pmatrix} 0 \\ 0 \\ \hat{g}_{l,0} \end{pmatrix} \quad (4.132)$$

mit $\hat{g}_{l,0}$ aus (4.131). Die lokalen Änderungen der Schwerebeschleunigung sind gering. Daher ist der Schätzwert der Schwerebeschleunigung konstant:

$$\hat{g}_l^n = \hat{g}_{l,0}^n \quad (4.133)$$

Gemäß (4.129) gilt mit (4.130), (4.132) und (B.4)

$$-\bar{f}_{ib}^b = \mathbf{C}_b^{n,T} \bar{g}_l^n \approx \mathbf{C}_b^{n,T} \begin{pmatrix} 0 \\ 0 \\ \hat{g}_{l,0} \end{pmatrix} = \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \varphi \\ \cos \varphi \cos \theta \end{pmatrix} \hat{g}_{l,0} \quad (4.134)$$

Es können also auch der initiale Nick- und Rollwinkel aus der Messung \bar{f}_{ib}^b geschätzt werden:

$$\hat{\varphi}_0 = \text{atan2}(\bar{g}_y, \bar{g}_z) \quad \hat{\theta}_0 = \text{atan2}(-\bar{g}_x, \sqrt{\bar{g}_y^2 + \bar{g}_z^2}) \quad \text{mit} \quad \begin{pmatrix} \bar{g}_x \\ \bar{g}_y \\ \bar{g}_z \end{pmatrix} = -\bar{f}_{ib}^b \quad (4.135)$$

Der Gierwinkel ψ ist im Stand nur durch die Magnetometermessung beobachtbar. Daher wird der Messwert des Magnetometers \tilde{m}^b (4.107) im Zeitintervall T_{init} gemittelt:

$$\bar{\tilde{m}}^b = \frac{1}{N} \sum_{k=1}^N \frac{\tilde{m}^b}{T_s} \quad \text{mit} \quad N = \frac{T_{\text{init}}}{T_s} \quad (4.136)$$

Gemäß (4.112) und (4.110) wird daraus dann der initiale Gierwinkel geschätzt:

$$\hat{\psi}_0 = \arctan2(-\bar{m}_y^h, \bar{m}_x^h) + \hat{\delta}_m \quad \text{mit} \quad \begin{pmatrix} \bar{m}_x^h \\ \bar{m}_y^h \\ \bar{m}_d^h \end{pmatrix} = \mathbf{C}_{b,0}^h \bar{\tilde{m}}^b \quad (4.137)$$

$\mathbf{C}_{b,0}^h$ wird gemäß (4.111) aus den initialen Nick- und Rollwinkeln $\hat{\varphi}_0$ und $\hat{\theta}_0$ (4.135) bestimmt. $\hat{\delta}_m$ ist die geschätzte lokale magnetische Deklination.

Mit den initialen Eulerwinkeln $\hat{\varphi}_0$, $\hat{\theta}_0$ und $\hat{\psi}_0$ aus (4.135) und (4.137) kann dann mit (B.6) das initiale Lagequaternion $\hat{\mathbf{q}}_{b,0}^n$ und mit (B.4) die initiale Richtungskosinusmatrix $\mathbf{C}_{b,0}^n$ berechnet werden.

Die Geschwindigkeit gegenüber dem erdfesten Bezugssystem wird im Stillstand mit null initialisiert:

$$\hat{v}_{eb,0}^n = \vec{0} \quad (4.138)$$

Die Position in WGS84-Koordinaten $\hat{p}_{U,0}^w$ wird mit den Messwerten des GNSS-Empfängers $\tilde{p}_{A,0}^w$ gemäß (4.94) initialisiert:

$$\hat{p}_{U,0}^w = \begin{pmatrix} \hat{\varphi}_0 \\ \hat{\lambda}_0 \\ \hat{h}_0 \end{pmatrix} = \begin{pmatrix} \tilde{\varphi}_{A,0} \\ \tilde{\lambda}_{A,0} \\ \tilde{h}_{A,0} \end{pmatrix} - \begin{pmatrix} \frac{1}{R_n + \tilde{h}_{A,0}} & 0 & 0 \\ 0 & \frac{1}{(R_e + \tilde{h}_{A,0}) \cos \tilde{\varphi}_{A,0}} & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{C}_{b,0}^{\hat{n}} \vec{d}_A^b \quad (4.139)$$

Dabei ist $\mathbf{C}_{b,0}^{\hat{n}}$ die initiale Richtungskosinusmatrix, siehe oben.

Zur Initialisierung von \mathbf{P}_k (2.46) muss die Güte des initialen Schätzfehlerzustandes $\Delta \hat{x}_0$ (4.75) als Kovarianzmatrix $Cov(\Delta \vec{x}_0, \Delta \hat{x}_0)$ ausgedrückt werden. Oben wird der initiale Zustand \hat{x}_0 mit den zur Verfügung stehenden Messwerten \tilde{y}_0 initialisiert. Dies entspricht einem Kalman-Update-Schritt, bei dem der bisherige Zustand \hat{x}_k^- verworfen und nur der Messwert \tilde{y}_k benutzt wird. Beim linearen Kalman-Filter ist dies bei $\mathbf{K}_k \mathbf{H}_k = \mathbf{I}$ der Fall, vgl. (2.38), (2.36). Damit wird das entsprechende Update der Schätzfehlerkovarianzmatrix in (2.39) und (2.58) zu $\mathbf{P}_k^+ = \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$ bzw. $\mathbf{P}_k^+ = \mathbf{H}_k^* \mathbf{R}_k \mathbf{H}_k^{*,T}$ (mit \mathbf{H}_k^* als Pseudo-Inverse von \mathbf{H}_k). Die Schätzfehlerkovarianz wird also mit der Messkovarianz \mathbf{R}_k initialisiert, siehe auch Fall (d) in Abschnitt 2.3.1.1.

In (4.139) wird die initiale Positionsschätzung $\hat{p}_{U,0}^w$ mit der Positionsmessung $\tilde{p}_{A,0}^w$ berechnet. Die entsprechende Schätzfehlerkovarianzmatrix $\mathbf{P}_{\Delta \vec{p},0}$ ist also gleich der Messkovarianzmatrix $\mathbf{R}_{p,0}$ (4.96):

$$\mathbf{P}_{\Delta \vec{p},0} = \mathbf{R}_{p,0} \quad (4.140)$$

Dabei gilt $\mathbf{H}_{p,0}^* = \mathbf{I}_3$. Da die Lage nicht aus der Positionsmessung geschätzt wird, hat der Term $-[\mathbf{C}_{b,k}^{\hat{n}} \vec{d}_A^b \times]$ in der Messmatrix $\mathbf{H}_{p,k}$ (4.100) hier keine Relevanz.

Zur initialen Geschwindigkeit $\hat{v}_{eb,0}^n$ (4.138) wird die initiale Schätzfehlerkovarianzmatrix $\mathbf{P}_{\Delta \vec{v},0}$ wie bei der Position auf die Messkovarianzmatrix $\mathbf{R}_{v,0}$ (4.103) gesetzt, obwohl die GNSS-Geschwindigkeitsmessung nicht benutzt wird:

$$\mathbf{P}_{\Delta \vec{v},0} = \mathbf{R}_{v,0} \quad (4.141)$$

Dies entspricht einer $\mathbf{H}_{v,0}^* = \mathbf{I}_3$, vgl. 2. Element in $\mathbf{H}_{v,k}$ (4.106). Obwohl die Geschwindigkeit im Stillstand präzise bekannt ist, wird $\mathbf{P}_{v,0}$ nicht zu null gesetzt, da sonst zukünftige Geschwindigkeits-Updates aufgrund einer „perfekten“ Zustandsschätzung verworfen werden würden (siehe auch Fall (c) in Abschnitt 2.3.1.1).

Bei der Schätzung der initialen Nick- und Rollwinkel $\hat{\varphi}_0, \hat{\theta}_0$ (4.135) aus den Beschleunigungssensormessungen werden ein eventuell vorhandenes Sensor-Bias (4.130) und Abweichungen vom Schwerebeschleunigungsmodell (4.132) vernachlässigt. Die dadurch möglichen Fehler werden durch Varianzen $\sigma_{\alpha,0}^2$ und $\sigma_{\beta,0}^2$ der Lagewinkelfehler α_0 und β_0 (4.73) abgeschätzt. Die initiale Gierwinkelschätzung $\hat{\psi}_0$ (4.137) aus den Magnetometermessungen wird durch die nicht genau bekannte lokale magnetische Deklination $\hat{\delta}_m$, lokale magnetische Störquellen und ungenaue Nick- und Rollwinkel verfälscht. Der dadurch mögli-

che Gierwinkelfehler γ_0 (4.73) wird mit der Varianz $\sigma_{\gamma,0}^2$ abgeschätzt, vgl. $\sigma_{m,k}^2$ in (4.109). Die Schätzfehlerkovarianzmatrix $\mathbf{P}_{\Delta\vec{\psi},0}$ für den Lagefehler $\Delta\vec{\psi}_0 = (\alpha_0, \beta_0, \gamma_0)$ ist damit:

$$\mathbf{P}_{\Delta\vec{\psi},0} = \begin{pmatrix} \sigma_{\alpha,0}^2 & 0 & 0 \\ 0 & \sigma_{\beta,0}^2 & 0 \\ 0 & 0 & \sigma_{\gamma,0}^2 \end{pmatrix} \quad (4.142)$$

Die Schätzfehler der initialen Drehraten- und Beschleunigungssensor-Bias $\hat{b}_{\omega,0}^b, \hat{b}_{a,0}^b$ aus (4.128), (4.130) werden als Varianzen $\sigma_{\Delta b_{\omega},0}^2, \sigma_{\Delta b_a,0}^2$ ausgedrückt:

$$\mathbf{P}_{\Delta\vec{b}_{\omega},0} = \sigma_{\Delta b_{\omega},0}^2 \mathbf{I}_3 \quad \mathbf{P}_{\Delta\vec{b}_a,0} = \sigma_{\Delta b_a,0}^2 \mathbf{I}_3 \quad (4.143)$$

Dabei enthält $\mathbf{P}_{\Delta\vec{b}_a,0}$ die Abschätzung des vernachlässigten initialen Beschleunigungssensor-Bias $\vec{b}_{a,0}^b$ (4.130).

Die Zahlenwerte der Varianzen in (4.142) und (4.143) sind in Tabelle C.2 gegeben.

Der initiale Schätzwert des Systemzustandes und die initiale Schätzfehlerkovarianzmatrix lauten damit:

$$\hat{\vec{x}}_0 = \left(\hat{\varphi}_0, \hat{\lambda}_0, \hat{h}_0, \hat{v}_{eb,0}^{0T}, \hat{\mathbf{q}}_{b,k}^{nT}, \hat{b}_{a,0}^{b,T}, \hat{b}_{\omega,0}^{b,T} \right)^T \quad (4.144)$$

$$\mathbf{P}_0 = \begin{pmatrix} \mathbf{P}_{\Delta\vec{p},0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\Delta\vec{v},0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{\Delta\vec{\psi},0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{\Delta\vec{b}_a,0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{\Delta\vec{b}_{\omega},0} \end{pmatrix} \quad (4.145)$$

4.6.2 Ablauf Propagation, Estimation

In Abbildung 2.2 ist der grundsätzliche Ablauf eines Kalman-Zeitschrittes, der aus der Propagation und Estimation besteht, dargestellt. Da in der vorliegenden Anwendung die GNSS-Positions- und Geschwindigkeitsmessungen erst mit zeitlicher Verzögerung verfügbar sind, muss dies in der Implementierung berücksichtigt werden. Außerdem sind die IMU-Sendezeitpunkte nicht mit dem Mikroprozessor synchronisiert, auf dem die Sensorfusion läuft, sodass die Zahl der bei Ausführung eines Kalman-Schrittes verfügbaren IMU-Nachrichten variiert. Die dadurch notwendigen Anpassungen des Algorithmus sollen im Folgenden dargestellt werden:

Um die verzögerten GNSS-Messungen verarbeiten zu können, wird ein Ringspeicher implementiert, in dem empfangene IMU- und Magnetometer-Datensätze sowie Propagationsschritt-Ergebnisse gespeichert werden. Vor dem ersten Zeitschritt $k = 0$ wird der

Positionsindex des Ringspeichers auf $s = 0$ gesetzt sowie der Schätzwert des Systemzustandes \hat{x}_k und die Schätzfehlerkovarianzmatrix \mathbf{P}_k mit den Startwerten \hat{x}_0 und \mathbf{P}_0 aus (4.144) und (4.145) initialisiert.

Der dann folgende Algorithmus, der alle $T_s = 0,01$ sec ausgeführt wird, ist in Abbildung 4.7 dargestellt. Zu Beginn jedes Zeitschrittes k wird $i = 1$ gesetzt und die Zahl der seit dem letzten Zeitschritt neu empfangenen IMU-Datensätze n_c bestimmt. Da die IMU alle 0,01 sec sendet, ist n_c im Mittel 1. Es können jedoch auch 0 oder 2 Datensätze vorliegen, da die IMU-Sendezeitpunkte nicht synchronisiert sind.

Dann wird geprüft, ob eine GNSS-Messung mit Positions- und Geschwindigkeitsinformationen verfügbar ist. Dies ist im Mittel alle 10 Zeitschritte der Fall, da die Senderate des GNSS-Empfängers 10 Hz beträgt. Das Alter Δt_A (4.22) der ggf. verfügbaren GNSS-Messung wird als Vielfaches n_r der Dauer eines Propagationsschrittes T_s ausgedrückt:

$$n_r = \left\lfloor \frac{\Delta t_A}{T_s} \right\rfloor \quad (4.146)$$

Dabei wird auf die nächste ganze Zahl abgerundet. n_r ist mindestens 1, im Mittel 3–8, vgl. Seite 109.

Wenn keine GNSS-Messung verfügbar ist, müssen keine Strapdownschritte in der Vergangenheit n_p wiederholt werden, daher wird $n_p = 0$ gesetzt. In diesem Fall folgen dann die Propagationsschritte für die *neuen* IMU-Datensätze.

Für jeden *neuen* IMU-Datensatz (Anzahl n_c) wird ein Propagationsschritt durchgeführt (Abbildung 4.7, Block (a) mit $n_p < i \leq n_p + n_c$). Zunächst wird der nächste ($i - n_p$ -te) neue IMU-Datensatz $\vec{u}_{k-n_p-n_c+i}$ vom Eingang geladen und im Ringspeicher an der aktuellen Speicherposition s gespeichert. Mit diesem IMU-Datensatz wird dann der propagierte A-priori-Schätzwert $\hat{x}_{k-n_p-n_c+i}^-$ und die propagierte A-priori-Schätzfehlerkovarianzmatrix $\mathbf{P}_{k-n_p-n_c+i}^-$ gemäß (4.47) und (4.67) berechnet und ebenfalls an der Ringspeicherposition s gespeichert. Der Propagationsschritt entfällt, wenn keine neue IMU-Nachricht empfangen wurde ($n_c = 0$).

Nach jedem Propagationsschritt für die *neuen* IMU-Datensätze erfolgt das Update mit der entsprechenden *neuen* Magnetometer-Messung (Block (b) mit $n_p < i \leq n_p + n_c$). Da das Magnetometer der IMU genutzt wird, sind die Magnetometer-Messungen synchron zu den anderen IMU-Datensätzen. Falls keine neue Magnetometer-Nachricht empfangen wurde ($n_c = 0$) – oder das Magnetometer-Update deaktiviert ist –, entfällt der Magnetometer-Updateschritt (Block (b) mit $n_c = 0$, Pfad „[sonst]“). In einem Magnetometer-Updateschritt für einen *neuen* Magnetometer-Datensatz wird die nächste ($i - n_p$ -te) Magnetometer-Messung \tilde{m}^b vom Eingang geladen und an der Ringspeicherposition s gespeichert. Mit dieser Magnetometer-Messung wird dann in einem Kalman-Updateschritt gemäß (4.118) und (4.126) (mit $q = m$) der korrigierte A-posteriori-Zustand $\hat{x}_{k-n_p-n_c+i}^+$ und $\mathbf{P}_{k-n_p-n_c+i}^+$ berechnet. Dann wird i um 1 erhöht. s wird inkrementiert, falls ein Propagationsschritt stattgefunden hat ($i \leq n_p + n_c + 1$). Falls noch weitere neue IMU-

Datensätze vorliegen ($i \leq n_p + n_c$), folgt nun der nächste Propagationsschritt für den nächsten *neuen* IMU-Datensatz. Sonst ist der Kalman-Zeitschritt beendet.

Wenn zu Beginn des Kalman-Schrittes eine GNSS-Messung vorliegt, wird der Zustand entsprechend dem Alter der Messung um n_r Zeitschritte zurückgerollt, um die Messung „in der Vergangenheit“ verarbeiten zu können (Block (c)). Dazu wird die Ringspeicherposition auf $s = s - n_r$ gesetzt und der vergangene A-priori-Zustand $\hat{x}_{k-n_r}^-$ und $\mathbf{P}_{k-n_r}^-$ aus dem Ringspeicher geladen. Mit diesem propagierten Zustand werden dann für den Geschwindigkeits- und Positionsmesswert jeweils ein Updateschritt gemäß (4.118) und (4.126) ($q \in \{p, v\}$) durchgeführt. Dann erfolgt das Magnetometer-Update mit der ebenfalls aus dem Ringspeicher geladenen, vergangenen Magnetometer-Messung. Die Updates auf den zurückliegenden A-posteriori-Zustand $\hat{x}_{k-n_r}^+$ und $\mathbf{P}_{k-n_r}^+$ werden also so berechnet, als ob alle Messgrößen bereits zu diesem Zeitpunkt vorgelegen hätten. Nach dem Update müssen auf Basis des nun korrigierten Zustandes die vergangenen n_p Propagations- und Magnetometer-Updateschritte wiederholt werden. Dazu wird $n_p = n_r - 1$ gesetzt. n_p ist um 1 geringer als n_r , da der/die aktuelle(n) Propagations- und Updateschritt (e) mit aktuellen Daten separat ausgeführt werden.

Der Ablauf der zu wiederholenden, *vergangenen* n_p Propagationsschritte (Block (a) mit $i \leq n_p$) gleicht denen für die *neuen* IMU-Datensätze. Es wird lediglich die Propagation mit vergangenen IMU-Datensätzen durchgeführt, die aus dem Ringspeicher geladen wurden. Der A-priori-Schätzwert \hat{x}^- und \mathbf{P}^- aus der Propagation wird jeweils wieder im Ringspeicher abgelegt. Im Anschluss an die Wiederholung eines Propagationsschrittes wird das Magnetometer-Update mit der vergangenen, aus dem Ringspeicher geladenen Magnetometer-Messung durchgeführt (Block (b) mit $i \leq n_p$). Wenn alle vergangenen n_p Propagations- und Magnetometer-Updateschritte durchgeführt wurden, erfolgen die oben beschriebenen n_c Propagations- und Magnetometer-Updateschritte $n_p < i \leq n_p + n_c$ (jetzt ist $n_p > 0$) für die ggf. vorhandenen *neuen* IMU- und Magnetometer-Datensätze.

Am Ende des Kalman-Zeitschrittes werden \hat{x}_k und \mathbf{P}_k ausgegeben. In ihnen sind sämtliche IMU-Datensätze und Messgrößen verarbeitet, die zu diesem Zeitpunkt verfügbar sind. Für den nächsten Zeitschritt bleiben \hat{x}_k , \mathbf{P}_k , der Ringspeicherinhalt und der Positionsindex s gespeichert.

5 Regelung

Ein wesentlicher Baustein für die Automatisierung des Quadropters ist die Regelung, die in diesem Kapitel vorgestellt wird. Aufgabe der Regelung ist es zum einen, das instabile Flugsystem zu stabilisieren. Zum anderen muss sie die höheren Regelungsziele wie die Verfolgung eines bewegten Zielobjekts ermöglichen.

Im Forschungsbereich sind Quadropter ein beliebtes Fallbeispiel für instabile Systeme. Daher gibt es wahrscheinlich kein Regelungsverfahren, das nicht schon auf einen Quadropter angewendet wurde. Zu den vielfach genutzten Reglertypen und Entwurfsmethoden gehören PID-Regler [35, 9], modellprädiktive Regler [4, 61], der Backstepping- [9, 62, 73], Sliding-Mode- [83, 9] und \mathcal{H}_∞ -Regelungsentwurf [61, 62], die flachheitsbasierte Regelung [55, 11] sowie das Reinforcement Learning [83, 10].

Um die Lage eines Quadropters – oder eines anderen Multikopters – zu stabilisieren, benötigt man im einfachsten Fall nur lineare Proportional-Differential-(PD-)Regler, die aus den über Inertialsensoren geschätzten Nick- und Rollwinkeln Rotordrehzahl-Kommandos berechnen. Sollen zudem die horizontale Position und die Höhe geregelt werden, können weitere lineare Proportional-Differential-Regler verwendet werden, die die Positionsbestimmung eines GNSS-Empfängers nutzen. Solche linearen Regelungen reichen für viele Anwendungen aus.

Die Dynamik des Quadropters ist nichtlinear. So kann er aktiv horizontal nur beschleunigen oder verzögern, wenn zuvor die Rotorebene geneigt wurde. Wird beim Regelungsentwurf – ggf. implizit – ein um den Schwebeflug-Arbeitspunkt linearisiertes Dynamikmodell verwendet, weicht dieses z. B. bei größeren Lagewinkeln von dem realen Verhalten ab und es kommt zu Regelabweichungen. Beim manuellen Flug müssen diese Fehler von der Pilotin oder dem Piloten ausgeglichen werden.

Um ein bewegtes Zielobjekt automatisch zu verfolgen, ist eine Regelung erforderlich, die den zur Zielverfolgung geplanten, dynamischen Trajektorien präzise folgt. Ein Regelungsverfahren, das sich dazu besonders eignet, ist die flachheitsbasierte Regelung, da sie die nichtlineare Dynamik des Systems explizit ausnutzt und die Systemkanäle entkoppelt. In [55, 56] wird eine flachheitsbasierte Regelung z. B. erfolgreich im Indoor-Flug eingesetzt, um einen Quadropter dynamischen Trajektorien präzise folgen zu lassen, sodass dieser akrobatische Flugmanöver vollführt. Aufgrund der vielversprechenden Ergebnisse soll auch hier eine flachheitsbasierte Regelung zum Einsatz kommen, die an die Anforderungen einer Objektverfolgung im Outdoor-Flug angepasst wird.

Um einen flachheitsbasierten Regelungsansatz nutzen zu können, muss das zu regelnde System einen flachen Ausgang¹ besitzen. Dies wird für den Quadrokopter in Abschnitt 5.2.1 nachgewiesen. Somit lässt sich die Quadrokopter-Dynamik durch eine dynamische Zustandsrückführung exakt eingangs-/zustandslinearisieren². Eine überlagerte Trajektorienfolgeregelung muss sich dann lediglich um eine lineare Fehlerdynamik kümmern³, auch bei großen Lagewinkeln.

Abhängig davon, wie das Systemdynamik-Modell formuliert wird, kann die Ableitung der exakt linearisierenden Zustandsrückführung aufwendig sein. Wenn das Systemmodell mit Eulerwinkeln ausgedrückt wird, führen die trigonometrischen Funktionen zu großen Termen, wenn sie nicht geschickt umformuliert werden [2]. [11] umgeht das Problem durch eine Formulierung mit Quaternionen, betrachtet aber nur die translatorische Dynamik des Quadrokopters. In [55] wird eine sehr kompakte Form durch eine vektorielle Formulierung erreicht, ohne jedoch die resultierende Zustandsrückführung anzugeben. In dieser Arbeit wird ebenfalls eine vektorielle Notation verwendet. Die Rückführungsgleichungen werden hier vollständig hergeleitet und in der Form angegeben, wie sie für die Implementierung genutzt werden. Das Quadrokoptermodell berücksichtigt zusätzlich Störgrößeneingänge, um Störungen durch Wind im Outdoor-Flug ausgleichen zu können.

Für die Regelung müssen für die in Abschnitt 3.3.1 definierten Betriebsmodi geeignete Trajektorien erzeugt werden. Zur Objektverfolgung ist neben der geeigneten Kameraausrichtung die Trajektorie des Quadrokopters zu berechnen, mit der das Ziel aus dem gewünschten Abstand und Winkel gefilmt werden kann. Dazu kommt eine Trajektorien-generierung zum Einsatz, die auf einer Kaskadenregelung basiert. Diese Methode wird auch für den Flug per Joystick genutzt. Außerdem werden eine Trajektoriengenerierung mit Spline-Kurven zur Erzeugung von vorab planbaren Trajektorien sowie eine Trajektoriengenerierung auf Basis einer modellprädiktiven Regelung vorgestellt.

Das Kapitel ist wie folgt gegliedert: Zunächst werden in Abschnitt 5.1 das Systemmodell für die flachheitsbasierte Regelung, eine Schubkalibrierung und ein Störgrößen-Beobachter eingeführt. Außerdem wird auf die Identifikation der benötigten Modellparameter eingegangen. In Abschnitt 5.2 erfolgt dann der Beweis der Flachheit sowie der Entwurf der flachheitsbasierten Zustandsrückführung und der überlagerten Trajektorienfolgeregelung. Die Methoden zur Trajektoriengenerierung werden in Abschnitt 5.3 präsentiert. In Abschnitt 5.4 wird die Ablaufsteuerung des Systems und in Abschnitt 5.5 die Simulation des Quadrokopters mit flachheitsbasierter Regelung dargestellt.

¹vgl. Abschnitt 2.4

²vgl. Abschnitt 2.4.2

³vgl. Abschnitt 2.4.3

5.1 Systemmodell für flachheitsbasierte Regelung

5.1.1 Systemzustand

Zur Modellierung der Quadrocopter-Dynamik für die flachheitsbasierte Regelung wird im Folgenden der Systemzustand für das Modell eingeführt. Die Differentialgleichungen des Modells folgen dann in Abschnitt 5.1.3.

Die Position und die Geschwindigkeit des Quadrocopter-Zentrums Z im lokalen Koordinatensystem L (siehe Seite 24) haben die Koordinaten p_n, p_e, p_d und v_n, v_e, v_d (jeweils Norden, Osten, unten), siehe auch Abbildung 5.1:

$$\vec{p}_{lb} = \vec{p}_Z = p_n \vec{l}_n + p_e \vec{l}_e + p_d \vec{l}_d \quad (5.1)$$

$$\vec{v}_{lb} = \vec{v}_{lZ} = v_n \vec{l}_n + v_e \vec{l}_e + v_d \vec{l}_d \quad (5.2)$$

Die Orientierung des Quadrocopters wird mit dem Lagequaternion

$$\mathbf{q}_b^l = (q_1 \ q_2 \ q_3 \ q_4)^T \quad (5.3)$$

gemäß (2.12) beschrieben oder alternativ mit der entsprechenden Richtungskosinusmatrix \mathbf{C}_b^l (2.13).

Die Drehrate (2.17) des Quadrocopters bezüglich L hat die Koordinaten $\omega_x, \omega_y, \omega_z$ in B :

$$\vec{\omega}_{lb} = \omega_x \vec{b}_x + \omega_y \vec{b}_y + \omega_z \vec{b}_z \quad (5.4)$$

Der Systemzustand \vec{x} besteht aus der Position (5.1), der Geschwindigkeit (5.2), der Orientierung (5.3) und der Drehrate (5.4):

$$\begin{aligned} \vec{x} &= (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12} \ x_{13})^T \\ &= (p_n \ p_e \ p_d \ v_n \ v_e \ v_d \ q_1 \ q_2 \ q_3 \ q_4 \ \omega_x \ \omega_y \ \omega_z)^T \end{aligned} \quad (5.5)$$

Durch das normierte Lagequaternion \vec{q}_b^l haben die Zustände die Zwangsbedingung $q_1^2 + q_2^2 + q_3^2 + q_4^2 = x_7^2 + x_8^2 + x_9^2 + x_{10}^2 = 1$, vgl. (2.12).

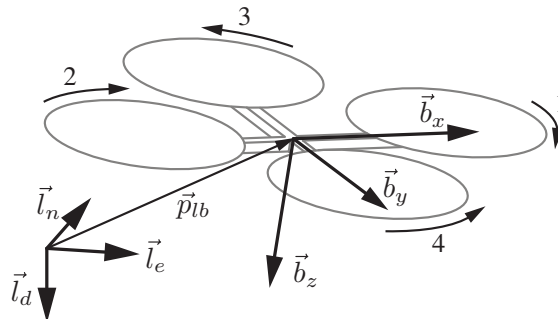


Abbildung 5.1: Position des Quadrocopters \vec{p}_{lb} , Koordinatensysteme B und L , Rotornummerierung

Im Folgenden wird auf einige Annahmen, die bei der Modellierung der Quadrokopter-Dynamik getroffen werden, eingegangen. Außerdem wird dargestellt, wie sich die oben eingeführten, für die Regelung benötigten Größen aus den Schätzgrößen der Sensorfusion berechnen lassen:

Für die Regelung werden das Bezugssystem E als inertial angenommen und die Erddrehrate $\vec{\omega}_{ie}$ (4.33) vernachlässigt. Außerdem wird angenommen, dass der Quadrokopter im Verhältnis zum Erdradius nur kleine Strecken zurücklegt; daher entfällt auch die Transportrate $\vec{\omega}_{en}$ (4.34) und L und N sind gleich orientiert, vgl. (4.31). Insgesamt gilt dann:

$$\vec{\omega}_{ie} \approx 0 \quad , \quad \vec{\omega}_{en} \approx 0 \quad , \quad \mathbf{C}_l^n \approx \mathbf{I} \quad , \quad \vec{\omega}_{nb} \approx \vec{\omega}_{lb} = \vec{\omega}_{eb} \approx \vec{\omega}_{ib} \quad (5.6)$$

Damit ergeben sich die Schätzwerte für die Orientierung \mathbf{q}_b^l (5.3) und die Drehrate $\vec{\omega}_{lb}$ (5.4) unmittelbar aus den Schätzwerten $\hat{\mathbf{q}}_b^n$ (4.121) und $\hat{\vec{\omega}}_{ib}$ (4.124) der Sensorfusion:

$$\hat{\mathbf{q}}_b^l \approx \hat{\mathbf{q}}_b^n \quad (5.7)$$

$$\hat{\vec{\omega}}_{lb} \approx \hat{\vec{\omega}}_{ib} \quad (5.8)$$

Die Regelung verwendet als Ursprung des körperfesten Koordinatensystems B das Quadrokopter-Zentrum Z , da dort in etwa der Schwerpunkt des Quadrokopters liegt (zur Definition der Bezugs- und Koordinatensysteme siehe Abschnitt 2.1.1). Für die Position \vec{p}_{lb} (5.1) des Quadrokopter-Zentrums Z gilt bezüglich der Position \vec{p}_U (4.24), die die Sensorfusion schätzt:

$$\vec{p}_{lb} = \vec{p}_Z = \vec{p}_U - \vec{d}_I \quad (5.9)$$

\vec{d}_I ist der Vektor vom Quadrokopter-Zentrum Z zum Punkt U , der Position der IMU.

Damit lässt sich der Schätzwert der Position \vec{p}_{lb} wie folgt aus den WGS84-Koordinaten $(\hat{\varphi}, \hat{\lambda}, \hat{h})$ (4.25) des Punktes U , die die Sensorfusion ausgibt (4.120), berechnen (vgl. (4.98)):

$$\hat{\vec{p}}_{lb}^l \approx \begin{pmatrix} (\hat{\varphi} - \varphi_L)(R_n + \hat{h}) \\ (\hat{\lambda} - \lambda_L)(R_e + \hat{h}) \cos \varphi_L \\ -(\hat{h} - h_L) \end{pmatrix} - \mathbf{C}_b^{\hat{n}} \vec{d}_I^b \quad (5.10)$$

$(\varphi_L, \lambda_L, h_L)$ sind die WGS84-Koordinaten des Ursprungs von L in W . $\mathbf{C}_b^{\hat{n}}$ lässt sich mit (2.13) aus $\hat{\mathbf{q}}_b^n$ berechnen.

Für die Geschwindigkeit \vec{v}_{lb} (5.2) des Quadrokopter-Zentrums Z bezüglich L folgt aus (5.9) mit (2.22) und (B.10):

$$\vec{v}_{lb} = \vec{v}_{lZ} = \vec{v}_{lU} - \vec{\omega}_{lb} \times \vec{d}_I \quad (5.11)$$

Die Geschwindigkeit \vec{v}_{lU} des Punktes U ist mit den Vereinfachungen (5.6) gleich der Geschwindigkeit $\vec{v}_{eU} = \vec{v}_{eb}$ (4.26), die die Sensorfusion schätzt (4.122). Für den Schätzwert

der Geschwindigkeit \vec{v}_{lb} (5.2) lässt sich damit in Abhängigkeit von den Sensorfusions-Schätzwerten \hat{v}_{eb}^n , $\hat{\mathbf{q}}_b^n$ und $\hat{\vec{\omega}}_{ib}^b$ schreiben:

$$\hat{v}_{lb}^l \approx \hat{v}_{eb}^n - \mathbf{C}_b^{\hat{n}} \left(\hat{\vec{\omega}}_{ib}^b \times \vec{d}_I^b \right) \quad (5.12)$$

Für die Beschleunigung \vec{a}_{lb} des Punktes Z bezüglich L folgt mit (2.23) und (B.10) aus (5.11):

$$\vec{a}_{lb} = \vec{a}_{lZ} = \vec{a}_{lU} - \vec{\omega}_{lb} \times (\vec{\omega}_{lb} \times \vec{d}_I) - \vec{\alpha}_{lb} \times \vec{d}_I \quad (5.13)$$

Da \vec{d}_I nur eine Länge von 6 cm hat (vgl. Tabelle C.1), können $\vec{\omega}_{lb} \times (\vec{\omega}_{lb} \times \vec{d}_I)$ und $\vec{\alpha}_{lb} \times \vec{d}_I$ bei den zu erwartenden Drehraten und Drehbeschleunigungen vernachlässigt werden. Dann gilt für den Schätzwert für \vec{a}_{lb} in Abhängigkeit von der Beschleunigung $\hat{a}_{eU} = \hat{a}_{eb}$ (4.125), die die Sensorfusion schätzt:

$$\hat{a}_{lb}^l \approx \hat{a}_{eb}^n \quad (5.14)$$

5.1.2 Aerodynamik

5.1.2.1 Rotorschub und -momente

Jeder der vier Rotoren des Quadropters (vgl. Abbildung 5.2) erzeugt eine Schubkraft F_{tk} , die in etwa proportional zum Quadrat der Rotordrehgeschwindigkeit ω_k ist. Gleiches gilt für das durch den Luftwiderstand erzeugte Moment M_{dk} um die Drehachse des Rotors, das auf den Quadropters wirkt [9, 54, 55]:

$$F_{tk} = k_t \omega_k^2 \quad (5.15)$$

$$M_{dk} = k_d \omega_k^2 \quad (5.16)$$

Die Faktoren k_t , k_d sind vor allem abhängig vom Rotordurchmesser, der Rotorgeometrie, der Luftdichte und der Anströmgeschwindigkeit der Luft.



Abbildung 5.2: Rotor mit Brushless-Motor

Im Schwebeflug bei Windstille ist die Anströmgeschwindigkeit weit oberhalb des Rotors null und es stellt sich eine konstante induzierte Luftströmung durch die Rotorebene ein. In Abhängigkeit von den Quadrokopter- und Windgeschwindigkeiten verändern sich die effektive Anströmgeschwindigkeit und damit die Faktoren k_t und k_d [38, 54, 34]. Bei einer seitlichen Anströmung des Rotors kommt es zudem – durch die unterschiedlichen Anströmgeschwindigkeiten des vor- und rücklaufenden Rotorblattes – zu asymmetrischem Auftrieb und Luftwiderstand und damit zu weiteren Kräften und Momenten, die auf den Quadrokopter wirken. Dies führt auch zu einer Auf- und Abbewegung der Rotorblätter (Schlagbewegung, blade flapping) [38, 54, 34]. Solche und weitere Effekte sollen hier vernachlässigt werden. Die resultierenden Kräfte und Momente müssen als Störungen durch die Regelung kompensiert werden.

Der Rotorschub-Faktor k_t in (5.15) wird in Abschnitt 5.1.2.2 indirekt für den Schwebeflug parametrisiert. Der Rotorwiderstandsmoment-Faktor k_d aus (5.16) wird zusammen mit dem Trägheitsmoment J_z bestimmt, siehe Abschnitt 5.1.4.

Die Schubkräfte F_{tk} addieren sich zu einer Gesamtschubkraft F_t , die körperfest nach oben, also entlang $-b_z$, wirkt:

$$F_t = \sum_{k=1}^4 F_{tk} \quad (5.17)$$

Die Schubkräfte F_{tk} und Widerstandsmomente M_{dk} erzeugen Momente um die drei Körperachsen:

$$\begin{aligned} \vec{M}_t = & (F_{t3} - F_{t4}) d_R \vec{b}_x + (F_{t1} - F_{t2}) d_R \vec{b}_y \\ & + (-M_{d1} - M_{d2} + M_{d3} + M_{d4}) \vec{b}_z \end{aligned} \quad (5.18)$$

d_R ist der Abstand von den Rotorachsen zur vertikalen Symmetrieachse im Zentrum des Quadrokopters. Für die Nummerierung der Rotoren siehe Abbildung 5.1.

Aus (5.18) wird ersichtlich, wie der Quadrokopter nicken, rollen und gieren kann: Für ein Rollen um die Längsachse \vec{b}_x wird ein Differenzschub der Rotoren 3 und 4 an Back- und Steuerbord benötigt. Der Quadrokopter nickt um die Querachse \vec{b}_y , wenn die Rotoren 1 und 2 vorne und hinten unterschiedlichen Schub erzeugen. Zum Gieren werden die Widerstandsmomente der Rotoren M_{dk} genutzt. Durch paarweises Erhöhen und Erniedrigen der Drehzahlen der jeweils gegenüberliegenden Rotoren, die sich in die gleiche Richtung drehen, wird ein Moment um die Hochachse \vec{b}_z erzeugt, ohne den Gesamtschub zu verändern (siehe (5.15) bis (5.18)).

Mit seinen vier Stellgrößen, den Rotordrehzahlen ω_k , ist der Quadrokopter bezüglich seiner sechs Starrkörper-Freiheitsgrade unteraktuiert: Er kann nicht in jede Raumrichtung aktiv translatorisch beschleunigen, sondern nur körperfest nach oben in Richtung $-b_z$, vgl. (5.17) und (5.39b). Es kann kein Schub körperfest seitlich in Richtung von b_x oder b_y erzeugt werden. Daher muss für eine horizontale Bewegung des Quadrokopters zunächst genickt bzw. gerollt werden, um den Schubvektor zu kippen, sodass dann ein Teil des

Schubes horizontal wirkt und der Quadrokopter horizontal beschleunigt. Zum horizontalen „Abbremsen“ muss der Quadrokopter in die entgegengesetzte Richtung kippen, siehe Simulation in Abschnitt 5.5 und Flugversuch A1 in Abschnitt 6.2.2.

Motordynamik Wird eine kommandierte Rotordrehzahl geändert, kann der Motorregler die Änderung aufgrund des Widerstandsmoments M_{dk} sowie des Trägheitsmoments von Rotor und Motor nicht instantan umsetzen. Zudem wird das Kommando durch die Kommunikation zwischen Mikroprozessor und Motorregler verzögert. Insgesamt kann das Verhalten als ein Verzögerungselement 1. Ordnung [52, 83] mit der Zeitkonstanten T_1 und einer zusätzlichen Totzeit T_t modelliert werden:

$$G(s) = \frac{1}{T_1 s + 1} e^{-T_t s} \quad (5.19)$$

Zur Identifikation der Parameter T_1 und T_t wurden die Sprungantworten für unterschiedliche Drehzahlsprünge aufgezeichnet; Details sind in [45] zu finden. Die mittleren, minimalen und maximalen Modellparameter sind in Tabelle 5.1 gegeben. Dabei ist zu berücksichtigen, dass die von den Motorreglern gemessenen, aufgezeichneten Drehzahlen durch die Kommunikation ebenfalls eine Totzeit in etwa der gleichen Größenordnung wie die Drehzahlkommandos aufweisen. Die tatsächliche Totzeit T_t beträgt also maximal die Hälfte der gemessenen.

	Mittelwert	Max.	Min.
T_1	30 ms	46 ms	24 ms
T_t	39 ms	42 ms	26 ms

Tabelle 5.1: Parameter des Motordynamikmodells

Es ist auffällig, dass die Variation in den Parametern recht groß ist. So ist der maximale Wert für T_1 fast doppelt so groß wie der minimale. Zudem wird meist eine Erhöhung der Drehzahl schneller umgesetzt als eine Reduzierung [52].

Bei Anwendungen mit höchsten dynamischen Anforderungen wie z. B. Kunstflug [52] sollte die Motordynamik – zusammen mit detaillierteren Rotordynamikmodellen – explizit berücksichtigt werden. Hier sind die dynamischen Anforderungen geringer. Daher wird die Motordynamik (5.19) im weiteren Regelungsentwurf vernachlässigt. Die Regler müssen dies als Störung kompensieren.

5.1.2.2 Schubkalibrierung

In diesem Abschnitt wird die statische Parametrierung des Rotorschubmodells (5.15) beschrieben und die Kalibrierung der individuellen Rotorschübe im Schwebeflug dargestellt.

Die Motorregler des Pelican erwarten als Stellgröße eine normierte Drehrate ω_{nsk} anstatt der physikalischen Drehrate ω_k . Der Zusammenhang zwischen ω_{nsk} und ω_k lässt sich als affine Funktion darstellen:

$$\omega_k = b \omega_{nsk} + c \quad (5.20)$$

Die normierte Motor-Soll-Drehrate ω_{nsk} ist beschränkt:

$$\omega_{nsk,\min} \leq \omega_{nsk} \leq \omega_{nsk,\max} \quad \text{mit} \quad \omega_{nsk,\min} = 0,005, \quad \omega_{nsk,\max} = 1 \quad (5.21)$$

Nach Einsetzen von (5.20) in (5.15) ergibt sich für den Rotorschub als Funktion der normierten Motordrehrate ein Polynom 2. Grades:

$$F_{tk} = a_{t2} \omega_{nsk}^2 + a_{t1} \omega_{nsk} + a_{t0} \quad (5.22)$$

$$\text{mit} \quad a_{t2} = k_t b^2, \quad a_{t1} = 2k_t b c, \quad a_{t0} = k_t c^2 \quad (5.23)$$

Die Zahlenwerte für die Koeffizienten a_{t2} , a_{t1} , a_{t0} in Tabelle 5.2 wurden experimentell auf einer Schubwaage bestimmt, für Details siehe [45].

Koeffizient	a_{t2}	a_{t1}	a_{t0}
Zahlenwert	14,73 N	0,32 N	0,38 N

Tabelle 5.2: Parameter des Rotorschubmodells

Das Polynom (5.22) mit den Parametern in Tabelle 5.2 approximiert die tatsächlichen Schubmessungen aller vier Rotoren recht gut mit Residuen von maximal 0,5 N, siehe Abbildung 5.3. Aufgrund der leicht unterschiedlichen individuellen Rotoreigenschaften, die sich auch mit der Zeit ändern, und variierendem Luftdruck wird zu Beginn jedes Fluges der Schub jedes Rotors kalibriert [51]. Dazu wird für das Verhältnis von kommandiertem Schub F_{tsk} und tatsächlichem Schub F_{tk} für jeden Rotor ein Kalibrierungsfaktor λ_k eingeführt:

$$F_{tk} = \lambda_k F_{tsk} \quad (5.24)$$

Im Schwebeflug muss die Gesamtschubkraft gleich der Gewichtskraft sein und die Momente müssen sich zu null ergeben:

$$F_t = m g \quad \vec{M}_t = \vec{0}$$

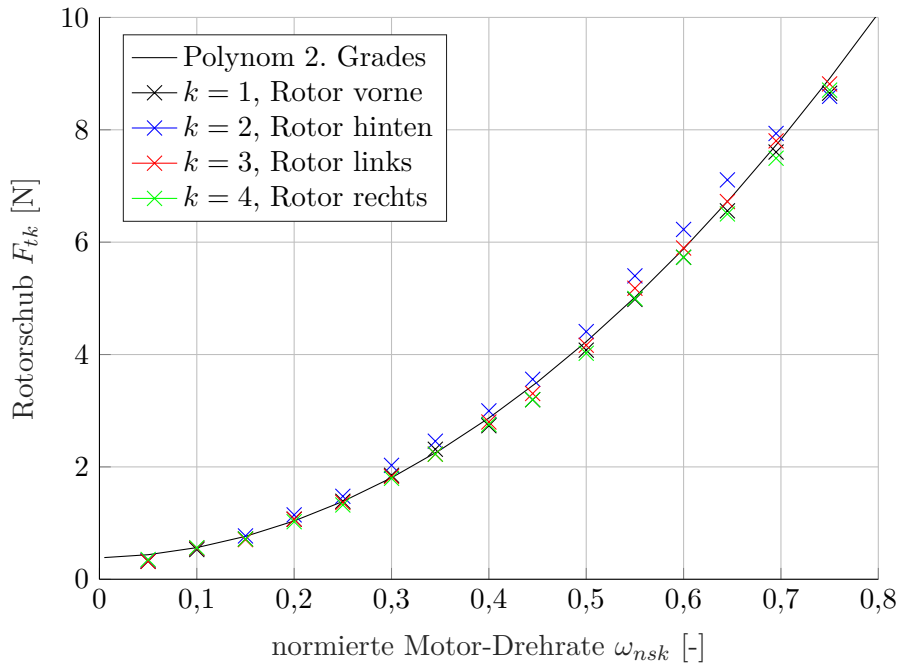


Abbildung 5.3: Rotorschub als Funktion der normierten Motordrehrate: Polynom 2. Grades (5.22) mit Parametern aus Tabelle 5.2, Rotor-Schubmessungen auf Schubwaage

Damit ergibt sich aus (5.17), (5.18), (5.15), (5.16), (5.24) ein lineares Gleichungssystem für die Rotorschub-Kalibrierungsfaktoren λ_k :

$$\begin{pmatrix} F_{ts1} & F_{ts2} & F_{ts3} & F_{ts4} \\ 0 & 0 & F_{ts3} & -F_{ts4} \\ F_{ts1} & -F_{ts2} & 0 & 0 \\ -F_{ts1} & -F_{ts2} & F_{ts3} & F_{ts4} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} = \begin{pmatrix} m g \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.25)$$

Die Lösung dieses Gleichungssystems lautet (für $F_{tsk} > 0$):

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} = \frac{m g}{4} \begin{pmatrix} 1/F_{ts1} \\ 1/F_{ts2} \\ 1/F_{ts3} \\ 1/F_{ts4} \end{pmatrix} \quad (5.26)$$

Durch Aufzeichnung des mittleren kommandierten Schubes F_{tsk} pro Rotor während des geregelten Schwebefluges können dann die Rotorschub-Kalibrierungsfaktoren λ_k bestimmt werden, siehe Abschnitt 6.2.1. Wenn das Rotorschubmodell sowie die Quadrokopter-Masse m passen und die Masseverteilung gleichmäßig ist, ist ein Wert von 1 für die λ_k zu erwarten. Ein $\lambda_k < 1$ bzw. $\lambda_k > 1$ bedeutet jeweils, dass der Rotorschub niedriger bzw. höher ist, als vom Modell (5.22) prädictiert.

5.1.2.3 Luftwiderstand

Auf einen Körper in einer Luftströmung wirkt die Kraft

$$F_w = c_w A \frac{\rho_L v_L^2}{2} \quad (5.27)$$

mit dem Luftwiderstandskoeffizienten c_w , zugehöriger Bezugsfläche A und der Luftdichte ρ_L . v_L ist die Relativgeschwindigkeit zwischen der Strömungsgeschwindigkeit der Luft und dem Körper. Die Luftdichte ρ_L ist abhängig von Luftdruck und Lufttemperatur (z. B. $1,225 \text{ kg/m}^3$ bei 1013 mbar und 15°C).

Die Bezugsfläche A für den Quadropter ist richtungsabhängig. Die seitliche Stirnfläche des Quadropters liegt in der Größenordnung 800 cm^2 . Der Luftwiderstandskoeffizient c_w bzw. das Produkt $c_w A$ ließe sich im Windkanal bestimmen. Für eine Beispielrechnung soll c_w mit 1 abgeschätzt werden.

Windstärke	Geschwindigkeit v_L		F_w	α
0 Bft, Windstille	0,5 m/s	1,8 km/h	0,01 N	0,03°
1 Bft, leiser Zug	2,0 m/s	7,4 km/h	0,21 N	0,53°
2 Bft, leichte Brise	3,6 m/s	13,0 km/h	0,64 N	1,6°
3 Bft, schwache Brise	5,7 m/s	20,3 km/h	1,6 N	4,0°
4 Bft, mäßige Brise	8,2 m/s	29,6 km/h	3,3 N	8,4°

Tabelle 5.3: Luftwiderstands-Kraft F_w und notwendige Schrägstellung α des Quadropters bei unterschiedlichen Windgeschwindigkeiten im Schwebeflug

In Tabelle 5.3 ist eine Abschätzung der auf den Quadropter wirkenden Kraft F_w für die oben genannten Parameter für Windstärken bis 4 Bft zu sehen. Außerdem ist der Neigungswinkel α angegeben, um den der Quadropter durch Nicken oder Rollen geneigt sein muss, um bei der genannten Windgeschwindigkeit die Position zu halten. Dabei ist zu beachten, dass die Geschwindigkeit v_L sowohl durch Wind als auch durch die Eigenbewegung des Quadropters verursacht werden kann. Die maximale Trajektoriengeschwindigkeit⁴ liegt mit 4 m/s deutlich unter den Windgeschwindigkeiten, die bei den zu erwartenden Windstärken von 3-4 Bft auftreten. Bei 4 m/s Eigengeschwindigkeit beträgt die auf den Quadropter wirkende Kraft durch den Luftwiderstand bei Windstille ca. $0,8 \text{ N}$. Der Quadropter muss sich also um etwa 2° zusätzlich neigen, um das zu kompensieren.

Zur Schätzung der auf den Quadropter wirkenden Kraft durch den Luftwiderstand wird in Abschnitt 5.1.5 ein Störgrößen-Beobachter vorgestellt.

⁴siehe Trajektoriengenerierung in Abschnitt 5.3.3 und $v_{ne,lim}$ in Tabelle C.6

5.1.3 Bewegungsgleichungen

In diesem Abschnitt werden die Bewegungsgleichungen des Systemmodells, die Differentialgleichungen für den Systemzustand \vec{x} (5.5), dargestellt. Wie bei der Sensorfusion wird der Quadrokopter als Starrkörper modelliert.

Auf die Quadrokoptermasse m wirken die Gravitationskraft der Erde $F_G = mg$ gemäß (2.5) entlang \vec{l}_d nach unten, die Gesamtschubkraft F_t aus (5.17) körperfest entlang $-\vec{b}_z$ und eine Störkraft \vec{F}_s :

$$\sum_i \vec{F}_i = mg \vec{l}_d - F_t \vec{b}_z + \vec{F}_s \quad (5.28)$$

Die Störkraft \vec{F}_s wird im Wesentlichen durch den Luftwiderstand bei Wind und Eigenbewegung verursacht und durch den Störgrößen-Beobachter in Abschnitt 5.1.5 geschätzt. Wie in Abschnitt 5.1.2.3 dargestellt, ist der Anteil der Luftwiderstands-Kraft, die durch die Eigengeschwindigkeit des Quadrokopters verursacht wird, im Mittel deutlich kleiner als der durch die Windgeschwindigkeit. Daher wird der eigengeschwindigkeitsabhängige Luftwiderstand hier nicht explizit gemäß (5.27) modelliert, da auch die Parameter $c_w A \rho_L$ nicht ausreichend genau bekannt sind. Der Störgrößen-Beobachter schätzt beide Anteile der Luftwiderstands-Kraft zusammen.

Mit den Kräften (5.28) folgt aus dem zweiten newtonschen Gesetz (2.25) für die Beschleunigung des Quadrokopters:

$$\vec{a}_{lb} = \frac{1}{m} \sum_i \vec{F}_i = g \vec{l}_d - u_1 \vec{b}_z - \vec{a}_s \quad (5.29)$$

u_1 ist die spezifische Gesamtschubkraft:

$$u_1 = \frac{1}{m} F_t \quad (5.30)$$

Und \vec{a}_s ist die auf die Masse bezogene spezifische Störkraft:

$$\vec{a}_s = \frac{1}{m} \vec{F}_s \quad \text{mit } \vec{a}_s = a_{sn} \vec{l}_n + a_{se} \vec{l}_e + a_{sd} \vec{l}_d \quad (5.31)$$

u_1 ist der erste Stellgrößen-Eingang; a_{sn} , a_{se} , a_{sd} sind die Störgrößen-Eingänge für das Systemmodell.

Aufgrund der Vereinfachung $\vec{\omega}_{ie} \approx 0$ in (5.6) muss in (5.29) nicht zwischen der spezifischen Gravitationskraft \vec{g} und der Schwerebeschleunigung \vec{g}_l (2.9) unterschieden werden. Für die Quadrokopter-Regelung wird ein konstanter Zahlenwert angenommen:

$$g = |\vec{g}| \approx |\vec{g}_l| \approx 9,81 \text{ m/s}^2 \quad (5.32)$$

Die Beschleunigung \vec{a}_{lb} kann als Ableitung der Geschwindigkeit (5.2) gemäß (2.23) und als zweifache Ableitung der Position (5.1) in den Koordinaten $\dot{v}_n, \dot{v}_e, \dot{v}_d$ und $\ddot{p}_n, \ddot{p}_e, \ddot{p}_d$ ausgedrückt werden:

$$\begin{aligned}\vec{a}_{lb} &= a_n \vec{l}_n + a_e \vec{l}_e + a_d \vec{l}_d = \frac{L d\vec{v}_{lb}}{dt} = \dot{\vec{v}}_{lb} = \dot{v}_n \vec{l}_n + \dot{v}_e \vec{l}_e + \dot{v}_d \vec{l}_d \\ &= \frac{L d^2 \vec{p}_{lb}}{dt^2} = \ddot{\vec{p}}_{lb} = \ddot{p}_n \vec{l}_n + \ddot{p}_e \vec{l}_e + \ddot{p}_d \vec{l}_d\end{aligned}\quad (5.33)$$

Für die rotatorischen Differentialgleichungen kommt der Drehimpulssatz (2.24) zum Einsatz: Demnach gilt für die Drehbeschleunigung $\vec{\alpha}_{lb}$:

$$\vec{\alpha}_{lb} = \mathbf{J}^{-1} \left(\sum_i \vec{M}_i - \vec{\omega}_{lb} \times (\mathbf{J} \vec{\omega}_{lb}) \right) \quad (5.34)$$

\mathbf{J} ist der Trägheitstensor des Quadropters bezüglich des Schwerpunktes, siehe Abschnitt 5.1.4. Auf den Quadropter wirken die Momente durch die Rotoren \vec{M}_t gemäß (5.18). Weitere Momente z. B. durch gyroskopische Effekte [9] oder Trägheit der Motoren werden vernachlässigt. Damit ist die Summe aller auf den Quadropter wirkenden Momente bezüglich des Schwerpunktes:

$$\sum_i \vec{M}_i = \vec{M}_t \quad (5.35)$$

Dabei wird implizit angenommen, dass der Schwerpunkt auf der senkrechten \vec{b}_z -Achse liegt, sodass d_R in (5.18) die Hebelarmlänge für die Schubkräfte F_{tk} bezüglich des Schwerpunktes ist. Die vertikale Lage des Schwerpunktes hat hier keinen Einfluss, da in \vec{M}_t keine Kräfte mit vertikalem Hebelarm vorkommen.

Die Drehbeschleunigung $\vec{\alpha}_{lb}$ (2.21) wird als Ableitung der Drehrate (5.4) durch die Koordinaten $\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$ beschrieben:

$$\vec{\alpha}_{lb} = \frac{L d\vec{\omega}_{lb}}{dt} = \frac{B d\vec{\omega}_{lb}}{dt} = \dot{\omega}_{lb} = \dot{\omega}_x \vec{b}_x + \dot{\omega}_y \vec{b}_y + \dot{\omega}_z \vec{b}_z \quad (5.36)$$

Es werden die Drehbeschleunigungs-Eingänge u_2, u_3, u_4 eingeführt, um die rechte Seite von (5.34) zu ersetzen:

$$\mathbf{J}^{-1} \left(\vec{M}_t - \vec{\omega}_{lb} \times (\mathbf{J} \vec{\omega}_{lb}) \right) = u_2 \vec{b}_x + u_3 \vec{b}_y + u_4 \vec{b}_z \quad (5.37)$$

Damit ist die Drehbeschleunigung eine direkte Funktion dieser Eingänge:

$$\vec{\alpha}_{lb} = u_2 \vec{b}_x + u_3 \vec{b}_y + u_4 \vec{b}_z \quad (5.38)$$

Insgesamt ergeben sich folgende Bewegungsgleichungen für den Quadropter:

$$\dot{\vec{p}}_{lb} = \vec{v}_{lb} \quad (5.39a)$$

$$\dot{\vec{v}}_{lb} = g \vec{l}_d - u_1 \vec{b}_z + \vec{a}_s \quad (5.39b)$$

$$\dot{\mathbf{q}}_b^l = \frac{1}{2} \mathbf{\Omega}(\mathbf{q}_b^l) \vec{\omega}_{lb} \quad (5.39c)$$

$$\dot{\vec{\omega}}_{lb} = u_2 \vec{b}_x + u_3 \vec{b}_y + u_4 \vec{b}_z \quad (5.39d)$$

Die Differentialgleichung für die Positionsänderung (5.39a) folgt aus (2.22). Die Geschwindigkeitsänderung (5.39b) entspricht (5.29) mit (5.33). Für die Änderung des Quaternions (5.39c) siehe (2.18). Die Differentialgleichung für die Drehratenänderung (5.39d) schließlich ergibt sich aus (5.38) mit (5.36). Siehe (B.14) im Anhang für eine Darstellung der Bewegungsgleichungen (5.39) in Koordinatenschreibweise.

Das Systemmodell (5.39) für die flachheitsbasierte Zustandsrückführung hat damit die Form

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}) \quad (5.40)$$

mit Zustand \vec{x} (5.5) sowie Stell- und Störgrößeneingang

$$\vec{u} = (u_1 \ u_2 \ u_3 \ u_4 \ a_{sn} \ a_{se} \ a_{sd})^T \quad (5.41)$$

Die Eingänge u_5, u_6, u_7 sind die Störgrößeneingänge a_{sn}, a_{se}, a_{sd} . Die Ausgangsgleichung $\vec{y} = \vec{h}(\vec{x}, \vec{u})$ wird in Abschnitt 5.2 festgelegt.

5.1.4 Trägheitstensor

Zur Nutzung des Drehimpulssatzes (5.34) bzw. zur Berechnung der notwendigen Momente für die gewünschten Drehbeschleunigungen (5.109) wird der Trägheitstensor \mathbf{J} des Quadropters bezüglich des Ursprungs von B benötigt. Während die Masse des Quadropters leicht mit einer Waage bestimmt werden kann, ist die Bestimmung der Zahlenwerte des Trägheitstensors aufwendiger. Zur Identifikation wird der Zusammenhang zwischen den auf den Quadropter wirkenden Momenten und den Drehbeschleunigungen gemäß dem Drehimpulssatz ausgenutzt.

Mit einem Shaker, der den Quadropter in rotatorische Schwingungen versetzt und die dazu notwendigen Momente misst, kann der Trägheitstensor präzise bestimmt werden. Da eine solche Apparatur nicht zur Verfügung steht, wird der Trägheitstensor im Flugversuch identifiziert. Dazu wird zunächst die Struktur von \mathbf{J} abgeschätzt und dann auf die Auswertung des Flugversuchs eingegangen.

Der Trägheitstensor \mathbf{J}_1 eines massiven homogenen Quaders der Masse m_1 mit den Seitenlängen l_x, l_y, l_z beträgt bezüglich eines Koordinatensystems B , das um $\vec{d} = (d_x \ d_y \ d_z)^T$ vom Schwerpunkt des Quaders verschoben ist:

$$\mathbf{J}_1 = \frac{m_1}{12} \begin{pmatrix} l_y^2 + l_z^2 & 0 & 0 \\ 0 & l_x^2 + l_z^2 & 0 \\ 0 & 0 & l_x^2 + l_y^2 \end{pmatrix} + m_1 \begin{pmatrix} d_y^2 + d_z^2 & -d_x d_y & -d_x d_z \\ -d_x d_y & d_x^2 + d_z^2 & -d_y d_z \\ -d_x d_z & -d_y d_z & d_x^2 + d_y^2 \end{pmatrix} \quad (5.42)$$

Der zweite Summand ist der Steiner-Anteil.

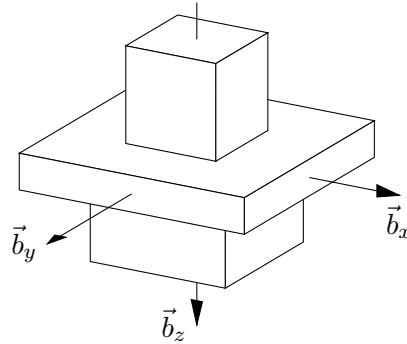


Abbildung 5.4: Geometrische Form des Quadropters zur Abschätzung der Struktur des Trägheitstensors

Die geometrische Form des Quadropters ist recht punktsymmetrisch bezüglich der \vec{b}_z -Achse, siehe Abbildung 3.1. Die Form lässt sich auf mehrere Quader reduzieren, die nur entlang der \vec{b}_z -Achse verschoben sind, sodass $d_{xi} = 0, d_{yi} = 0$. Die Seitenlängen dieser Quader entlang der \vec{b}_z - und \vec{b}_z -Achsen sind jeweils gleich: $l_{xi} = l_{yi}$. Die Quader sind punktsymmetrisch bezüglich der \vec{b}_z -Achse. Diese geometrische Form ist in Abbildung 5.4 dargestellt. Die Trägheitstensoren \mathbf{J}_i dieser Quader bezüglich B vereinfacht sich dadurch zu:

$$\mathbf{J}_i = \frac{m_i}{12} \begin{pmatrix} l_{xi}^2 + l_{zi}^2 & 0 & 0 \\ 0 & l_{xi}^2 + l_{zi}^2 & 0 \\ 0 & 0 & 2l_{xi}^2 \end{pmatrix} + m_i \begin{pmatrix} d_{zi}^2 & 0 & 0 \\ 0 & d_{zi}^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (5.43)$$

Der Gesamtträgheitstensor dieser Quader bezüglich B ist die Summe der einzelnen Trägheitstensoren bezüglich B . Dieser Gesamtträgheitstensor soll als Abschätzung der Struktur des Trägheitstensors \mathbf{J} des Quadropters dienen. \mathbf{J} hat also die Struktur:

$$\mathbf{J} \approx \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \quad \text{mit } J_x \approx J_y \quad (5.44)$$

Deviationsmomente auf den Nebendiagonalen von \mathbf{J} treten gemäß dem Steiner-Anteil in (5.42) durch Massen auf, deren Schwerpunkt in mindestens zwei Achsen gegenüber B verschoben ist. Derartige Anbauteile am Quadropters, deren Deviationsmomente nicht

durch punktsymmetrisch angeordnete Massen kompensiert werden, haben lediglich geringe Masse. Daher sollen die Deviationsmomente in \mathbf{J} vernachlässigt werden.

Unter Ausnutzung der Struktur von \mathbf{J} gemäß (5.44) lauten die für die Drehbeschleunigungen $\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$ erforderlichen Drehmomente \vec{M} gemäß dem Drehimpulssatz (2.24):

$$\vec{M}^b = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} \approx \begin{pmatrix} J_x \dot{\omega}_x \\ J_y \dot{\omega}_y \\ J_z \dot{\omega}_z \end{pmatrix} + \underbrace{\begin{pmatrix} (J_z - J_y) \omega_y \omega_z \\ (J_x - J_z) \omega_x \omega_z \\ 0 \end{pmatrix}}_{\approx 0} \approx \begin{pmatrix} J_x \dot{\omega}_x \\ J_y \dot{\omega}_y \\ J_z \dot{\omega}_z \end{pmatrix} \quad (5.45)$$

Dabei wird der zweite Summand vernachlässigt, weil er nur einen geringen Beitrag zu \vec{M} hat, da sowohl die Differenzen der Trägheitsmomente als auch die Drehratenprodukte klein sind.

Im Indoor-Flugversuch werden automatisch dynamische Trajektorien geflogen (vgl. Abschnitt 6.2.3). Position und Orientierung des Quadropters werden dabei mit dem Infrarot-Trackingsystem⁵ gemessen. Durch zweifache Differentiation der Orientierung können so die Drehbeschleunigungen $\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$ berechnet werden.

Da die Drehmomente \vec{M} nicht gemessen werden können, werden sie anhand der kommandierten normierten Rotordrehraten ω_{nsk} als die durch die Rotoren erzeugten Momente \vec{M}_t (5.18) abgeschätzt, vgl. (5.35). Damit folgt aus (5.45) unter Nutzung von (5.15)f, (5.20), (5.23), (5.24):

$$J_x \dot{\omega}_x \approx d_R \left(a_{t2} (\lambda_3 \omega_{ns3}^2 - \lambda_4 \omega_{ns4}^2) + a_{t1} (\lambda_3 \omega_{ns3} - \lambda_4 \omega_{ns4}) \right) \quad (5.46a)$$

$$J_y \dot{\omega}_y \approx d_R \left(a_{t2} (\lambda_1 \omega_{ns1}^2 - \lambda_2 \omega_{ns2}^2) + a_{t1} (\lambda_1 \omega_{ns1} - \lambda_2 \omega_{ns2}) \right) \quad (5.46b)$$

$$\begin{aligned} \frac{k_d}{k_t} J_z \dot{\omega}_z \approx & a_{t2} (-\lambda_1 \omega_{ns1}^2 - \lambda_2 \omega_{ns2}^2 + \lambda_3 \omega_{ns3}^2 + \lambda_4 \omega_{ns4}^2) \\ & + a_{t1} (-\lambda_1 \omega_{ns1} - \lambda_2 \omega_{ns2} + \lambda_3 \omega_{ns3} + \lambda_4 \omega_{ns4}) \end{aligned} \quad (5.46c)$$

Die Parameter auf der rechten Seite sind alle bekannt: a_{ti} sind die Rotorschub-Koeffizienten aus Tabelle 5.2, λ_k die zu Beginn des Versuchs bestimmten Rotorschub-Kalibrierungsfaktoren (5.26) und d_R die Länge der Rotorarme. Mit den aufgezeichneten kommandierten Motordrehzahlen ω_{nsk} und den berechneten Drehbeschleunigungen lassen sich durch eine lineare Regressionsrechnung J_x, J_y und $k_d/k_t J_z$ abschätzen. Das Trägheitsmoment J_z lässt sich dabei nicht unabhängig vom Quotienten der Schub- und Widerstandsmomentfaktoren k_d/k_t bestimmen. Die im Indoor-Flugversuch für eine Masse von $m = 2,075$ kg bestimmten Trägheitsmomente sind in Tabelle 5.4 dargestellt.

Mit einer höheren Masse steigen auch die Trägheitsmomente, siehe (5.42). Die in den Outdoor-Flugversuchen bei dem höheren Abfluggewicht verwendeten Trägheitsmomente sind in Tabelle C.1 zu finden.

⁵siehe Abschnitt 3.5.2

Trägheitsmoment	J_x	J_y	$k_d/k_t J_z$
Zahlenwert	0,03 kg m ²	0,03 kg m ²	0,9 kg m

Tabelle 5.4: Identifizierte Trägheitsmomente

5.1.5 Störgrößen-Beobachter

Wie in Abschnitt 5.1.2.3 dargestellt, muss sich der Quadropter bei Windstärken bis 4 Bft ca. 8° schräg stellen, um die Position halten zu können. Ohne Berücksichtigung einer Störkraft muss der Quadropter gemäß Modell (5.39b) jedoch waagrecht stehen, um die Position zu halten. Um zu erlauben, dass sich der Quadropter aktiv „in den Wind“ stellt, und um die Störgrößenkompensation nicht vollständig dem Regler zu überlassen, soll in diesem Abschnitt ein Störgrößen-Beobachter dargestellt werden, der die auf den Quadropter wirkende Störkraft F_s in (5.28) schätzt. Sie wird vor allem durch Luftwiderstand bei Wind und Eigenbewegung verursacht, siehe Abschnitt 5.1.2.3.

Es kommt ein lineares Kalman-Filter gemäß Abschnitt 2.3 zum Einsatz, das als Zustand die auf die Masse bezogene spezifische Störkraft \vec{a}_s (5.31) in drei Richtungen – Norden, Osten und unten – schätzt. Neben $\hat{\vec{a}}_{s,k}$ schätzt das Kalman-Filter die Schätzfehlerkovarianzmatrix $\mathbf{P}_{s,k}$ (2.29):

$$\hat{\vec{a}}_{s,k} = E[\vec{a}_{s,k}] \quad (5.47)$$

$$\mathbf{P}_{s,k} = Cov(\vec{a}_{s,k}, \hat{\vec{a}}_{s,k}) \quad (5.48)$$

Für die Schätzung wird der Umstand genutzt, dass die IMU die Summe der auf den Quadropter wirkenden spezifischen Kräfte messen kann. Die Differenz zu den bekannten Kräften wird durch die Störkraft verursacht. Der „Messwert“ für die spezifische Störkraft ist gemäß (5.39b):

$$\tilde{\vec{a}}_s = \hat{\vec{a}}_{lb} - g \vec{l}_d + u_1 \hat{\vec{b}}_z \quad (5.49)$$

Dabei ist $\hat{\vec{a}}_{lb}$ die geschätzte Beschleunigung des Quadropters (5.14), g der Wert aus (5.32) und u_1 die durch die Regelung gestellte spezifische Schubkraft. Die Richtung von $\hat{\vec{b}}_z$ wird aus der geschätzten Orientierung $\hat{\mathbf{q}}_b^l$ (5.7) berechnet.

Als Systemmodell wird angenommen, dass die Störkraft \vec{a}_s im Mittel konstant ist und die Änderung durch mittelwertfreies, zeitdiskretes weißes Rauschen $\vec{w}_{s,k}$ (A.29) modelliert werden kann:

$$\vec{a}_{s,k+1} = \Phi_s \vec{a}_{s,k} + \mathbf{G}_s \vec{w}_{s,k} \quad \text{mit} \quad \Phi_s = \mathbf{I}_3, \quad \mathbf{G}_s = \mathbf{I}_3 \quad (5.50)$$

Damit sind die Schätzwerte für die Änderung der Störkraft und für die höheren Ableitungen null:

$$\hat{\dot{\vec{a}}}_s = 0 \quad \hat{\ddot{\vec{a}}}_s = 0 \quad (5.51)$$

Die zur Messgröße $\tilde{y}_k = \tilde{a}_s$ passende, geschätzte Messgröße \hat{y}_k ist gleich dem geschätzten Systemzustand $\hat{a}_{s,k}$. Damit lautet das Messmodell (2.36) mit dem mittelwertfreien, diskreten weißen Messrauschen $\vec{v}_{s,k}$ (A.29):

$$\hat{y}_k = \mathbf{H}_s \hat{a}_{s,k} + \vec{v}_{s,k} \quad \text{mit} \quad \mathbf{H}_s = \mathbf{I}_3 \quad (5.52)$$

Zur Parametrierung der Kovarianzmatrizen $\mathbf{Q}_{s,k}$ (2.30) und $\mathbf{R}_{s,k}$ (2.31) werden die Varianzen des mittelwertfreien, diskreten weißen System- und Messrauschens $\sigma_{w_{s,k}}^2$ und $\sigma_{v_{s,k}}^2$ eingeführt, vgl. (A.32):

$$\mathbf{Q}_{s,k} = Cov(\vec{w}_{s,k}, \vec{w}_{s,k}) = \sigma_{w_{s,k}}^2 \mathbf{I}_3 \quad \mathbf{R}_{s,k} = Cov(\vec{v}_{s,k}, \vec{v}_{s,k}) = \sigma_{v_{s,k}}^2 \mathbf{I}_3 \quad (5.53)$$

Die Propagation des geschätzten Systemzustandes $\hat{a}_{s,k}$ gemäß (2.34) muss nicht explizit implementiert werden, da aus (5.50) $\hat{a}_{s,k+1}^- = \hat{a}_{s,k}$ folgt. Die Propagation der Schätzfehlerkovarianzmatrix $\mathbf{P}_{s,k}$ erfolgt gemäß (2.35) mit den oben eingeführten Matrizen.

Das Update des Schätzwertes $\hat{a}_{s,k}$ gemäß (2.38) lautet:

$$\hat{a}_{s,k}^+ = \hat{a}_{s,k}^- - \mathbf{K}_{s,k} (\mathbf{H}_s \hat{a}_{s,k}^- - \tilde{a}_{s,k}) \quad (5.54)$$

mit $\tilde{a}_{s,k}$ aus (5.49). Die Kalman-Gain-Matrix $\mathbf{K}_{s,k}$ wird gemäß (2.37) aus den oben eingeführten Matrizen berechnet. Das Update der Kovarianzmatrix $\mathbf{P}_{s,k}$ erfolgt gemäß (2.39).

Die in den Flugversuchen⁶ verwendeten Zahlenwerte für $\sigma_{w_{s,k}}$ und $\sigma_{v_{s,k}}$ wurden empirisch bestimmt und sind in Tabelle C.2 zu finden.

5.2 Flachheitsbasierte Regelung

In diesem Abschnitt wird eine flachheitsbasierte Regelung für die Quadropter-Dynamik (5.39) entworfen. Dazu muss zunächst ein Kandidat für den flachen Ausgang

$$\vec{y} = \vec{h}(\vec{x}, \vec{u}) \quad (5.55)$$

gemäß (2.78) definiert werden.

Als Komponenten des möglichen flachen Ausgangs werden hier gemäß [55, 56] die Positionskoordinaten aus (5.1) und der Gierwinkel (2.16a) sowie zusätzlich die Koordinaten der Störgröße \vec{a}_s (5.31) gewählt:

$$\begin{aligned} \vec{y} &= (y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7)^T \\ &= (p_n \ p_e \ p_d \ \psi \ a_{sn} \ a_{se} \ a_{sd})^T \end{aligned} \quad (5.56)$$

⁶Erprobung Störgrößen-Beobachter siehe Flugversuche C, D1 – D3 in Abschnitt 6.2, Details in Flugversuch D3 ab Seite 233

Die Störgröße \vec{a}_s wird mit in den Ausgang aufgenommen in der Annahme, dass diese mess- bzw. beobachtbar ist. Dies ist mit dem Störgrößen-Beobachter aus Abschnitt 5.1.5 der Fall. \vec{a}_s ist damit sowohl Eingang als auch Ausgang.

Im Folgenden wird zunächst gezeigt, dass das System (5.39) flach ist mit (5.56) als flacher Ausgang. Im Anschluss wird dann in Abschnitt 5.2.2 eine exakt linearisierende Zustandsrückführung entworfen und in Abschnitt 5.2.3 eine Trajektorienfolgeregelung vorgestellt.

5.2.1 Beweis Flachheit

In diesem Abschnitt wird nachgewiesen, dass das System (5.39) mit dem Ausgang (5.56) gemäß Definition 2.1 differentiell flach ist. Die Beweisführung erfolgt in Anlehnung an [55]. Dort wird der Beweis für ein ähnliches System – jedoch ohne Störgrößeneingänge – skizziert. Hier soll der Beweis vollständig dargestellt werden. Da eine vektorielle Schreibweise so lange wie möglich beibehalten wird, kann die Herleitung verhältnismäßig kurz gehalten werden.

Der Systemzustand \vec{x} hat gemäß (5.5) 13 Elemente. Durch die Zwangsbedingung des normierten Lagequaternions in den Zuständen x_7 bis x_{10} ist die Zahl der Freiheitsgrade $n = 12$.

Die Dimension des Ausgangs (5.56) ist gleich der des Eingangs (5.41) $\dim \vec{y} = \dim \vec{u} = 6$. Damit erfüllt der Ausgang die Form (2.78).

Zunächst wird Bedingung 1) von Definition 2.1 überprüft und gezeigt, dass sich der Ausgang \vec{y} als Funktionen des Zustands \vec{x} , des Eingangs \vec{u} und einer endlichen Anzahl von Zeitableitungen $u_i^{(k_i)}$ ausdrücken lässt:

Die Positionen p_n, p_e, p_d sind Teil des Zustands \vec{x} (5.5), der Gierwinkel ψ ist gemäß (2.16a), (2.13) eine statische nichtlineare Funktion von \vec{x} bzw. $(q_1 \ q_2 \ q_3 \ q_4)^T$ und die Störgrößen-Ausgänge sind gleich den Störgrößen-Eingängen:

$$y_1 = p_n = x_1, \quad y_2 = p_e = x_2, \quad y_3 = p_d = x_3 \quad (5.57)$$

$$y_4 = \psi = \arctan2(c_{21}, c_{11}) = f(x_7, x_8, x_9, x_{10}) \quad (5.58)$$

$$y_5 = a_{sn} = u_5, \quad y_6 = a_{se} = u_6, \quad y_7 = a_{sd} = u_7 \quad (5.59)$$

Damit ist $\vec{y} = \vec{f}(x_1, x_2, x_3, x_7, x_8, x_9, x_{10}, u_5, u_6, u_7)$ und Bedingung 1) ist erfüllt.

Um Bedingung 2) zu erfüllen, müssen sich die Zustandsvariablen \vec{x} und Eingänge \vec{u} als Funktion des Ausgangs \vec{y} und einer endlichen Anzahl von Zeitableitungen $y_i^{(k_i)}$ ausdrücken lassen:

Für die ersten drei Zustände folgt dies aus der Inversion von (5.57):

$$x_1 = p_n = y_1, \quad x_2 = p_e = y_2, \quad x_3 = p_d = y_3 \quad (5.60)$$

Der Zusammenhang zwischen den Geschwindigkeitszuständen und dem Ausgang \vec{y} ergibt sich aus der Differentiation von (5.60):

$$x_4 = \dot{p}_n = \dot{y}_1, \quad x_5 = \dot{p}_e = \dot{y}_2, \quad x_6 = \dot{p}_d = \dot{y}_3 \quad (5.61)$$

Die Zustände x_7 bis x_{10} sind die Elemente des Lagequaternions \mathbf{q}_b^l , die sich mit (B.5) aus der Richtungskosinusmatrix \mathbf{C}_b^l berechnen lassen. Die Richtungskosinusmatrix wiederum lässt sich gemäß (B.2) aus Skalarprodukten der Basisvektoren von B und L konstruieren:

$$\mathbf{C}_b^l = \begin{pmatrix} \vec{b}_x^l & \vec{b}_y^l & \vec{b}_z^l \end{pmatrix} \quad \text{mit} \quad \vec{b}_i^l = \begin{pmatrix} \vec{l}_n \cdot \vec{b}_i \\ \vec{l}_e \cdot \vec{b}_i \\ \vec{l}_d \cdot \vec{b}_i \end{pmatrix} \quad (5.62)$$

Nun werden die \vec{b}_i^l – die Repräsentationen der Basisvektoren \vec{b}_i in Koordinaten von L – als Funktionen des Ausgangs \vec{y} und dessen Ableitungen hergeleitet.

Gemäß (5.39b) gilt:

$$\vec{v}_{lb} = \ddot{p}_n \vec{l}_n + \ddot{p}_e \vec{l}_e + \ddot{p}_d \vec{l}_d = g \vec{l}_d - u_1 \vec{b}_z + a_{sn} \vec{l}_n + a_{se} \vec{l}_e + a_{sd} \vec{l}_d \quad (5.63)$$

Durch Umstellen nach \vec{b}_z und Projektion auf $\vec{l}_n, \vec{l}_e, \vec{l}_d$ ergibt sich:

$$\vec{b}_z^l = \begin{pmatrix} \vec{l}_n \cdot \vec{b}_z \\ \vec{l}_e \cdot \vec{b}_z \\ \vec{l}_d \cdot \vec{b}_z \end{pmatrix} = \frac{1}{u_1} \vec{t}^l \quad \text{mit} \quad \vec{t}^l = \begin{pmatrix} a_{sn} - \ddot{p}_n \\ a_{se} - \ddot{p}_e \\ a_{sd} - \ddot{p}_d + g \end{pmatrix} \quad (5.64)$$

Der Vektor $-\vec{t}$ kann als vektorielle spezifische Gesamtschubkraft interpretiert werden, die erzeugt werden muss, damit die gewünschte Beschleunigung \vec{p}_{lb} bei wirkender Schwerkraft und Störung \vec{a}_s erreicht wird. Um $-\vec{t}$ zu erzeugen, muss die Hochachse des Quadropters $-\vec{b}_z$ in die Richtung von $-\vec{t}$ zeigen, da der Quadropter nur Schub in Richtung $-\vec{b}_z$ erzeugen kann, vgl. (5.29). In (5.64) ist ersichtlich, dass u_1 keinen Einfluss auf die Richtung von \vec{b}_z^l hat. Damit kann die bekannte Länge des Basisvektors $|\vec{b}_z^l| = 1$ ausgenutzt werden, um die Abhängigkeit von u_1 zu eliminieren:

$$\vec{b}_z^l = \frac{\vec{t}^l}{|\vec{t}^l|} = \frac{1}{\sqrt{(a_{sn} - \ddot{p}_n)^2 + (a_{se} - \ddot{p}_e)^2 + (a_{sd} - \ddot{p}_d + g)^2}} \begin{pmatrix} a_{sn} - \ddot{p}_n \\ a_{se} - \ddot{p}_e \\ a_{sd} - \ddot{p}_d + g \end{pmatrix} \quad (5.65)$$

Mit $\ddot{p}_n = \ddot{y}_1, \ddot{p}_e = \ddot{y}_2, \ddot{p}_d = \ddot{y}_3$ kann \vec{b}_z^l als Funktion des Ausgangs \vec{y} und dessen Ableitungen dargestellt werden:

$$\vec{b}_z^l = f(\ddot{y}_1, \ddot{y}_2, \ddot{y}_3, y_5, y_6, y_7) \quad (5.66)$$

Zur Einbindung des Gierwinkels $\psi = y_4$ werden zwei Hilfskoordinatensysteme H und H' eingeführt, mit denen die Rotationsmatrix \mathbf{C}_b^l in drei Elementarrotationen aufgeteilt wird: $\mathbf{C}_b^l = \mathbf{C}_h^l \mathbf{C}_{h'}^h \mathbf{C}_b^{h'}$. Gemäß der Eulerwinkeldefinition auf Seite 28 ist \mathbf{C}_h^l die Gier-

Rotation ψ um $\vec{l}_d = \vec{h}_z$, $\mathbf{C}_{h'}^h$ die Nick-Rotation θ um $\vec{h}_y = \vec{h}'_y$ und $\mathbf{C}_b^{h'}$ die Roll-Rotation φ um $\vec{h}'_x = \vec{b}_x$. Mit (B.4) ist \vec{h}_y^l analog zu (5.62):

$$\vec{h}_y^l = (-\sin \psi, \cos \psi, 0)^T \quad (5.67)$$

und damit eine Funktion von y_4 . Der Basisvektor \vec{b}_x ($= \vec{h}'_x$) steht senkrecht auf einer Ebene, die durch die Basisvektoren $\vec{h}'_y = \vec{h}_y$, \vec{b}_y , \vec{h}'_z , \vec{b}_z aufgespannt wird und in der auch der Rollwinkel φ liegt. Damit lässt sich \vec{b}_x^l mit \vec{b}_z^l aus (5.64) und \vec{h}_y^l aus (5.67) konstruieren als

$$\vec{b}_x^l = \frac{\vec{h}_y^l \times \vec{b}_z^l}{|\vec{h}_y^l \times \vec{b}_z^l|} \quad (5.68)$$

Der verbleibende Basisvektor \vec{b}_y^l steht senkrecht auf \vec{b}_x^l und \vec{b}_z^l und ist damit

$$\vec{b}_y^l = \vec{b}_z^l \times \vec{b}_x^l \quad (5.69)$$

\vec{b}_x^l und \vec{b}_y^l sind somit auch eine Funktion von $\vec{y}^{(k)}$:

$$\begin{pmatrix} \vec{b}_x^l \\ \vec{b}_y^l \end{pmatrix} = \vec{f}(\vec{b}_z^l, y_4) = \vec{f}(\ddot{y}_1, \ddot{y}_2, \ddot{y}_3, y_4, y_5, y_6, y_7) \quad (5.70)$$

Mit (5.66), (5.70) und (5.62) lassen sich somit \mathbf{C}_b^l , über (B.5) \mathbf{q}_b^l und damit die Zustände $x_7 \dots x_{10}$ als Funktion des Ausgangs \vec{y} und dessen Ableitungen darstellen:

$$\mathbf{C}_b^l = \mathbf{F}(\ddot{y}_1, \ddot{y}_2, \ddot{y}_3, y_4, y_5, y_6, y_7) \quad (5.71)$$

$$(x_7, x_8, x_9, x_{10})^T = \mathbf{q}_b^l = \vec{f}(\ddot{y}_1, \ddot{y}_2, \ddot{y}_3, y_4, y_5, y_6, y_7) \quad (5.72)$$

Bevor die verbleibenden Zustände betrachtet werden, soll u_1 als Funktion von $\vec{y}^{(k)}$ dargestellt werden. Dazu wird (5.39b) auf \vec{l}_d projiziert:

$$\underbrace{\dot{v}_{lb} \cdot \vec{l}_d}_{\check{p}_d} = g \underbrace{\vec{l}_d \cdot \vec{l}_d}_1 - u_1 \underbrace{\vec{b}_z \cdot \vec{l}_d}_{c_{33}} + \underbrace{\vec{a}_s \cdot \vec{l}_d}_{a_{sd}} \quad (5.73)$$

Das Skalarprodukt $\vec{b}_z \cdot \vec{l}_d$ ist gemäß (B.3) das Element c_{33} von \mathbf{C}_b^l . Umstellen von (5.73) nach u_1 ergibt:

$$u_1 = \frac{g - \check{p}_d + a_{sd}}{c_{33}} \quad (5.74)$$

Hier muss $\vec{b}_z \cdot \vec{l}_d = c_{33} = \cos \varphi \cos \theta \neq 0$ gelten, was erfüllt ist, solange der Quadropter nicht 90° gekippt ist ($|\varphi| < \pi/2$ und $|\theta| < \pi/2$). (5.74) lässt sich mit $c_{33} = \vec{b}_z \cdot \vec{l}_d = \vec{b}_z^l$ gemäß (5.66) als Funktion von $\vec{y}^{(k)}$ darstellen:

$$u_1 = f(\ddot{y}_3, y_7, \vec{b}_z^l) = f(\ddot{y}_1, \ddot{y}_2, \ddot{y}_3, y_5, y_6, y_7) \quad (5.75)$$

Nun werden die Drehraten $\omega_x = x_{11}$ und $\omega_y = x_{12}$ als Funktion des Ausgangs \vec{y} dargestellt. Dazu wird die Ableitung von (5.39b) gebildet:

$$\ddot{v}_{lb} = \dot{g}\vec{l}_d + g\dot{\vec{l}}_d - \dot{u}_1\vec{b}_z - u_1\dot{\vec{b}}_z + \dot{\vec{a}}_s = 0 + 0 - \dot{u}_1\vec{b}_z - u_1\vec{\omega}_{lb} \times \vec{b}_z + \dot{\vec{a}}_s \quad (5.76)$$

Für die Ableitungen der Basisvektoren gilt dabei gemäß (B.11): $\dot{\vec{l}}_d = \frac{L_d}{dt}\vec{l}_d = 0$ und $\dot{\vec{b}}_z = \frac{L_d}{dt}\vec{b}_z = \vec{\omega}_{lb} \times \vec{b}_z$. Wird (5.76) nach $\vec{\omega}_{lb} \times \vec{b}_z$ aufgelöst, ergibt sich mit der Ableitung von (5.33):

$$\vec{\omega}_{lb} \times \vec{b}_z = \frac{1}{u_1} \left((\dot{a}_{sn} - p_n^{(3)})\vec{l}_n + (\dot{a}_{se} - p_e^{(3)})\vec{l}_e + (\dot{a}_{sd} - p_d^{(3)})\vec{l}_d - \dot{u}_1\vec{b}_z \right) \quad (5.77)$$

Durch Projektion auf \vec{b}_x und \vec{b}_y ergeben sich die gesuchten Drehraten als $\omega_x = -(\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{b}_y$ und $\omega_y = (\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{b}_x$:

$$\begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} = \frac{1}{u_1} \left((\dot{a}_{sn} - p_n^{(3)}) \begin{pmatrix} -c_{12} \\ c_{11} \end{pmatrix} + (\dot{a}_{se} - p_e^{(3)}) \begin{pmatrix} -c_{22} \\ c_{21} \end{pmatrix} + (\dot{a}_{sd} - p_d^{(3)}) \begin{pmatrix} -c_{32} \\ c_{31} \end{pmatrix} \right) \quad (5.78)$$

Hier muss $u_1 \neq 0$ gelten. In ω_x und ω_y tritt der Ausgang also als $y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_5, \dot{y}_6, \dot{y}_7$ auf. Die Abhängigkeit von der Lage kann mit (5.71) und die Abhängigkeit von u_1 mit (5.75) aufgelöst werden. Damit sind die Zustände x_{11} und x_{12} ebenfalls eine Funktion des Ausgangs und dessen Ableitungen:

$$\begin{aligned} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} &= \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} = \vec{f}(y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_5, \dot{y}_6, \dot{y}_7, \mathbf{q}_b^l, u_1) \\ &= \vec{f}(y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \ddot{y}_1, \ddot{y}_2, \ddot{y}_3, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7) \end{aligned} \quad (5.79)$$

Für die verbleibende Drehrate $\omega_z = x_{13}$ wird die Drehrate $\vec{\omega}_{lb}$ wie oben die Richtungskosinusmatrix in die Elementar-Drehraten bzw. Eulerwinkelraten aufgeteilt:

$$\vec{\omega}_{lb} = \vec{\omega}_{lh} + \vec{\omega}_{hh'} + \vec{\omega}_{h'b} = \dot{\psi}\vec{l}_d + \dot{\theta}\vec{h}'_y + \dot{\varphi}\vec{b}_x \quad (5.80)$$

Durch die Projektion auf \vec{b}_z ergibt sich das gesuchte ω_z :

$$\omega_z = \vec{\omega}_{lb} \cdot \vec{b}_z = \dot{\psi}\vec{l}_d \cdot \vec{b}_z + \dot{\theta}\vec{h}'_y \cdot \vec{b}_z + \dot{\varphi}\vec{b}_x \cdot \vec{b}_z \quad (5.81)$$

Mit $\vec{h}'_y{}^b = \mathbf{C}_{h'}^b \vec{b}_y^b = (0, \cos \varphi, -\sin \varphi)$, $\dot{\theta}$ aus (B.8), $\vec{l}_d \cdot \vec{b}_z = c_{33} = \cos \varphi \cos \theta$ (B.4) und $\vec{b}_x \cdot \vec{b}_z = 0$ kann (5.81) weiter aufgelöst werden:

$$\omega_z = \dot{\psi} \frac{\cos \theta}{\cos \varphi} - \omega_y \tan \varphi \quad (5.82)$$

Hier tritt der Ausgang als $\dot{y}_4 = \dot{\psi}$ auf. Die Drehrate ω_y kann mit (5.79) und die Eulerwinkel können über (2.16b)f. mit (5.71) in Abhängigkeit von \vec{y} ausgedrückt werden. Damit

ist auch der letzte Zustand x_{13} eine Funktion des Ausgangs und dessen Ableitungen:

$$x_{13} = \omega_z = f\left(\dot{y}_4, \omega_y, \mathbf{q}_b^l\right) = f\left(y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7\right) \quad (5.83)$$

Um die Abhängigkeit der Eingänge u_2, u_3 von $\vec{y}^{(k)}$ darzustellen, wird \dot{u}_1 benötigt. Dazu wird (5.76) auf \vec{l}_d projiziert, siehe (B.15), und nach \dot{u}_1 aufgelöst:

$$\dot{u}_1 = \frac{-p_d^{(3)} - u_1(c_{31}\omega_y - c_{32}\omega_x) + \dot{a}_{sd}}{c_{33}} \quad (5.84)$$

mit $c_{33} \neq 0$, vgl. oben. \dot{u}_1 ist also eine Funktion des Ausgangs $y_3^{(3)}$, der Lage, der Drehraten ω_x und ω_y sowie der Stell- und Störgrößeneingänge u_1 und \dot{y}_7 . Mit (5.71), (5.75) und (5.79) können diese Größen in Abhängigkeit von $\vec{y}^{(k)}$ ausgedrückt werden:

$$\dot{u}_1 = f\left(y_3^{(3)}, \dot{y}_7, \mathbf{q}_b^l, \omega_x, \omega_y, u_1\right) = f\left(y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7\right) \quad (5.85)$$

Nun werden die Ableitung von (5.76) und die zweite Ableitung von (5.33) gleichgesetzt:

$$\begin{aligned} \ddot{a}_{lb} &= -\ddot{u}_1 \vec{b}_z - 2\dot{u}_1 (\vec{\omega}_{lb} \times \vec{b}_z) - u_1 (\dot{\vec{\omega}}_{lb} \times \vec{b}_z) - u_1 (\vec{\omega}_{lb} \times (\vec{\omega}_{lb} \times \vec{b}_z)) + \ddot{a}_s \\ &= \vec{p}_{lb}^{(4)} = p_n^{(4)} \vec{l}_n + p_e^{(4)} \vec{l}_e + p_d^{(4)} \vec{l}_d \end{aligned} \quad (5.86)$$

Durch Projektion von \ddot{a}_{lb} auf \vec{b}_x bzw. \vec{b}_y erscheinen die benötigten u_2 und u_3 als $\dot{\omega}_x = -(\dot{\vec{\omega}}_{lb} \times \vec{b}_z) \cdot \vec{b}_y$ und $\dot{\omega}_y = (\dot{\vec{\omega}}_{lb} \times \vec{b}_z) \cdot \vec{b}_x$. Die Projektionen sind im Anhang in (B.16), (B.17) dargestellt. Durch Auflösen nach $\dot{\omega}_x = u_2$ und $\dot{\omega}_y = u_3$ ergibt sich:

$$\begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \frac{1}{u_1} \begin{pmatrix} (p_n^{(4)} - \ddot{a}_{sn})c_{12} + (p_e^{(4)} - \ddot{a}_{se})c_{22} + (p_d^{(4)} - \ddot{a}_{sd})c_{32} - 2\dot{u}_1\omega_x \\ -(p_n^{(4)} - \ddot{a}_{sn})c_{11} - (p_e^{(4)} - \ddot{a}_{se})c_{21} - (p_d^{(4)} - \ddot{a}_{sd})c_{31} - 2\dot{u}_1\omega_y \end{pmatrix} + \begin{pmatrix} \omega_y\omega_z \\ -\omega_x\omega_z \end{pmatrix} \quad (5.87)$$

Hier muss wieder $u_1 \neq 0$ gelten. Der Ausgang tritt als $y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \dot{y}_5, \dot{y}_6, \dot{y}_7$ auf. Die Abhängigkeiten von u_1, \dot{u}_1 , der Lage und den Drehraten können mit (5.75), (5.85), (5.71), (5.79) und (5.83) aufgelöst werden. Die Eingänge u_2, u_3 sind also eine Funktion des Ausgangs und dessen Ableitungen:

$$\begin{aligned} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} &= \vec{f}\left(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, \dot{y}_5, \dot{y}_6, \dot{y}_7, \mathbf{q}_b^l, \omega_x, \omega_y, \omega_z, u_1, \dot{u}_1\right) \\ &= \vec{f}\left(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_5, \dot{y}_6, \dot{y}_7, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7\right) \end{aligned} \quad (5.88)$$

Zur Darstellung der Abhängigkeit des Eingangs u_4 vom Ausgang $\vec{y}^{(k)}$ wird die Gierbeschleunigung $\dot{\psi}$ gebildet in der Erwartung, dass $\dot{\psi}$ vom Eingang $u_4 = \dot{\omega}_z$ abhängig ist. Dazu wird die zeitliche Ableitung der dritten Zeile von (B.8) berechnet und $\dot{\varphi}$ sowie $\dot{\theta}$

werden gemäß den ersten beiden Zeilen resubstituiert. Das Auflösen dieser Gleichung nach u_4 ergibt:

$$u_4 = \dot{\omega}_z = \underbrace{\ddot{\psi}}_{\ddot{y}_4} \frac{\cos \theta}{\cos \varphi} - \underbrace{\dot{\omega}_y}_{u_3} \tan \varphi + 2 \left(\omega_z^2 - \omega_y^2 \right) \sin \varphi \tan \theta - \omega_x \omega_y + \omega_x \omega_z \tan \varphi - 2 \omega_y \omega_z \tan \theta (\cos \varphi - \sin \varphi \tan \varphi) \quad (5.89)$$

für Rollwinkel $-90^\circ < \varphi < 90^\circ$ ist $\cos \varphi \neq 0$. In (5.89) tritt der Ausgang als $\ddot{y}_4 = \ddot{\psi}$ auf. Die Abhängigkeiten von dem anderen Eingang u_3 , der Lage und den Drehraten können mit (5.88), (5.71), (5.79) und (5.83) aufgelöst werden. Der Eingang u_4 ist also auch eine Funktion des Ausgangs und dessen Ableitungen:

$$\begin{aligned} u_4 &= \vec{f} \left(\ddot{y}_4, \mathbf{q}_b^l, \omega_x, \omega_y, \omega_z, u_3 \right) \\ &= \vec{f} \left(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7 \right) \end{aligned} \quad (5.90)$$

Für die verbleibenden Eingänge u_5, u_6, u_7 gilt gemäß (5.59):

$$u_5 = a_{sn} = y_5, \quad u_6 = a_{se} = y_6, \quad u_7 = a_{sd} = y_7 \quad (5.91)$$

Sie sind also ebenfalls eine Funktion des Ausgangs.

Mit (5.60), (5.61), (5.72), (5.79) und (5.83) sind alle Zustandsvariablen \vec{x} eine Funktion des Ausgangs \vec{y} und einer endlichen Anzahl von Zeitableitungen $y_i^{(k_i)}$:

$$\vec{x} = \vec{f} \left(y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7 \right) \quad (5.92)$$

Gemäß (5.75), (5.88), (5.90) und (5.91) lassen sich alle Eingänge \vec{u} sowohl als eine Funktion des Zustands \vec{x} , des Ausgangs und einer endlichen Anzahl von Zeitableitungen $y_i^{(k_i)}$

$$\vec{u} = \vec{f} \left(\vec{x}, y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, y_3^{(3)}, \dot{y}_3, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, \dot{y}_7, y_5, y_6, y_7 \right) \quad (5.93)$$

als auch als eine Funktion, die nur vom Ausgang und seinen Ableitungen $y_i^{(k_i)}$ abhängt, ausdrücken:

$$\vec{u} = \vec{f} \left(y_1^{(4)}, y_2^{(4)}, y_3^{(4)}, y_1^{(3)}, y_2^{(3)}, y_3^{(3)}, \dot{y}_1, \dot{y}_2, \dot{y}_3, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, \dot{y}_4, \dot{y}_5, \dot{y}_6, \dot{y}_7, y_4, y_5, y_6, y_7 \right) \quad (5.94)$$

Mit (5.92) und (5.94) ist auch Bedingung 2) von Definition 2.1 erfüllt.

Da also beide Bedingungen von Definition 2.1 erfüllt sind, ist **das System** (5.39) **flach mit dem flachen Ausgang** (5.56). \square

5.2.2 Flachheitsbasierte Zustandsrückführung

Gemäß dem Beweis im vorhergehenden Abschnitt ist die Quadropterdynamik (5.39) flach. Damit lässt sich das System durch eine quasi-statische oder endogene Zustandsrückführung exakt linearisieren (siehe Abschnitt 2.4.2), was die Trajektorienfolgeregelung vereinfacht.

Für die höchsten Ableitungsordnungen β_i gemäß (2.80) folgt aus (5.94):

$$\beta_1 = \beta_2 = \beta_3 = 4, \quad \beta_4 = \beta_5 = \beta_6 = \beta_7 = 2 \quad (5.95)$$

Damit ist $\sum_i \beta_i = 20 \neq n = 12$ und das System kann nicht durch eine statische Zustandsrückführung exakt linearisiert werden, vgl. (2.95).

Im Folgenden soll eine quasi-statische Zustandsrückführung entworfen werden. Gemäß (2.91) hat sie die Form $\vec{u} = \vec{\psi}(\vec{x}, \vec{\xi}, \dot{\vec{\xi}}, \dots, \vec{\xi}^{(s)})$. Damit lässt sich der neue Eingang $\vec{\xi}$ von (5.93) ablesen:

$$\vec{\xi} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \end{pmatrix} = \begin{pmatrix} y_1^{(4)} \\ y_2^{(4)} \\ \ddot{y}_3 \\ \ddot{y}_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} p_n^{(4)} \\ p_e^{(4)} \\ \ddot{p}_d \\ \ddot{\psi} \\ a_{sn} \\ a_{se} \\ a_{sd} \end{pmatrix} \quad (5.96)$$

Die quasi-statische Zustandsrückführung lässt sich damit schreiben als:

$$\vec{u} = \vec{\psi}(\vec{x}, \xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7, \dot{\xi}_3, \dot{\xi}_7, \ddot{\xi}_3, \ddot{\xi}_5, \ddot{\xi}_6, \ddot{\xi}_7) \quad (5.97)$$

Für die κ_i aus (2.84)ff. gilt dann hier mit den β_i aus (5.95):

$$\kappa_1 = \kappa_2 = 4, \quad \kappa_3 = \kappa_4 = 2, \quad \kappa_5 = \kappa_6 = \kappa_7 = 0 \quad (5.98)$$

Die Summe der κ_i ist damit gleich der Zahl der Systemfreiheitsgrade $\sum_{i=1}^m \kappa_i = 12 = n$, wie es gemäß (2.94) für eine quasi-statische Zustandsrückführung zu erwarten ist.

Die neuen Eingänge ξ_i sind gemäß (2.83) definiert als die κ_i -te Ableitung des flachen Ausgangs y_i . Durch Vorgabe der Eingänge $\xi_i = y_{i,s}^{(\kappa_i)}$ der flachheitsbasierten Vorsteuerung (5.97) kann das flache System – wie in Abbildung 2.3 dargestellt – vorgesteuert werden, sodass die flachen Ausgänge y_i den Solltrajektorien $y_{i,s}$ folgen. Die Trajektorienfolgeregelung wird in Abschnitt 5.2.3 hinzugefügt. ξ_1 und ξ_2 stehen für die Vorsteuerungs-Eingänge für die vierte Ableitung der Position in Nord- und Ostrichtung p_n, p_e . Die Vorsteuerungs-Eingänge für die zweiten Ableitungen der Vertikalposition p_d und des Gierwinkels ψ sind ξ_3 und ξ_4 . Gemäß (5.97) tritt ξ_3 auch in seiner ersten und zweiten Ableitung $\dot{\xi}_3, \ddot{\xi}_3$ auf. Alle drei Solltrajektorien $y_{1,s}, y_{2,s}, y_{3,s}$ für die Position müssen also 4-fach, die Solltrajektorie $y_{4,s}$ für den Gierwinkel muss 2-fach differenzierbar sein. Für die Ausgänge y_5, y_6, y_7

können keine Solltrajektorien vorgegeben werden, da es sich um die Störgrößen handelt, siehe folgenden Abschnitt.

Im Folgenden werden die Gleichungen der quasi-statischen Zustandsrückführung (5.97) im Detail dargestellt:

Störgrößen-Eingänge u_5 u_6 u_7

Die quasi-statische Zustandsrückführung (5.97) für die Eingänge u_5 , u_6 , u_7 würde gemäß (5.91) und (5.96) lauten:

$$\begin{pmatrix} u_5 \\ u_6 \\ u_7 \end{pmatrix} = \vec{a}_s = \begin{pmatrix} y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} \xi_5 \\ \xi_6 \\ \xi_7 \end{pmatrix} \quad (5.99)$$

Jedoch sind u_5 , u_6 , u_7 die Störgrößeneingänge der spezifischen Störkraft \vec{a}_s , die nicht vorgegeben werden können. Dies würde einer Steuerung des Quadropters durch den Wind entsprechen. Ebenso wenig können Solltrajektorien $y_{5,s}$, $y_{6,s}$, $y_{7,s}$ vorgegeben werden.

Entscheidend dafür, dass sich das Gesamtsystem trotzdem wie gewünscht verhält, ist, dass der Störgrößen-Beobachter aus Abschnitt 5.1.5 die spezifische Störkraft \vec{a}_s gut schätzt. Wenn gute Schätzwerte existieren, sind auch die ξ_5 , ξ_6 , ξ_7 bekannt, denen das System folgt, als ob (5.99) explizit implementiert wäre:

$$\begin{pmatrix} \xi_5 \\ \xi_6 \\ \xi_7 \end{pmatrix} = \begin{pmatrix} \hat{y}_5 \\ \hat{y}_6 \\ \hat{y}_7 \end{pmatrix} = \begin{pmatrix} \hat{a}_{sn} \\ \hat{a}_{se} \\ \hat{a}_{sd} \end{pmatrix} \quad (5.100)$$

Spezifischer Schub-Eingang u_1

Die Abhängigkeit des spezifischen Schub-Eingangs u_1 vom Zustand \vec{x} und vom Ausgang $y_i^{(\kappa_i)}$ ist bereits in (5.74) dargestellt. Darin tritt der neue Ausgang $\vec{\xi}$ aus (5.96) als $\ddot{p}_d = \ddot{y}_3 = \xi_3$ und $a_{sd} = y_7 = \xi_7$ auf. Mit $\xi_7 = \hat{a}_{sd}$ aus (5.100) ergibt sich als Zustandsrückführung für den spezifischen Schub-Eingang u_1 :

$$u_1 = \frac{g - \xi_3 + \hat{a}_{sd}}{c_{33}} \quad (5.101)$$

u_1 hängt also vom Sollgrößen-Eingang $\xi_3 = \ddot{p}_{d,s}$, vom Schätzwert der Störgröße $\xi_7 = \hat{a}_{sd}$ und vom Systemzustand \vec{x} in Form von c_{33} ab. Es gilt $c_{33} = \cos \varphi \cos \theta \neq 0$, solange der Quadropter nicht 90° gekippt ist ($|\varphi| < \pi/2$ und $|\theta| < \pi/2$). g ist der Wert aus (5.32).

Im Folgenden wird für u_2 und u_3 die Ableitung \dot{u}_1 benötigt. Aus (5.84) folgt mit $p_{d,s}^{(3)} = y_3^{(3)} = \dot{\xi}_3$ und $\dot{a}_{sd} = \dot{y}_7 = \dot{\xi}_7 = \hat{a}_{sd}$:

$$\dot{u}_1 = -\frac{\dot{\xi}_3 + u_1(c_{31}\omega_y - c_{32}\omega_x) + \hat{a}_{sd}}{c_{33}} \quad (5.102)$$

Drehbeschleunigungs-Eingänge u_2 u_3

Die Zustandsrückführung für die Drehbeschleunigungs-Eingänge u_2 und u_3 ergibt sich aus (5.87). Darin werden die Ausgangsableitungen $p_n^{(4)}$, $p_e^{(4)}$, $p_d^{(4)}$ durch die neuen Eingänge ξ_1 , ξ_2 , ξ_3 gemäß (5.96) ersetzt. Die \ddot{a}_{sn} , \ddot{a}_{se} , \ddot{a}_{sd} bzw. $\ddot{\xi}_5$, $\ddot{\xi}_6$, $\ddot{\xi}_7$ werden gemäß (5.100) durch die Schätzwerte ersetzt:

$$\begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \frac{1}{u_1} \begin{pmatrix} (\xi_1 - \hat{a}_{sn})c_{12} + (\xi_2 - \hat{a}_{se})c_{22} + (\ddot{\xi}_3 - \hat{a}_{sd})c_{32} - 2\dot{u}_1\omega_x \\ -(\xi_1 - \hat{a}_{sn})c_{11} - (\xi_2 - \hat{a}_{se})c_{21} - (\ddot{\xi}_3 - \hat{a}_{sd})c_{31} - 2\dot{u}_1\omega_y \end{pmatrix} + \begin{pmatrix} \omega_y\omega_z \\ -\omega_x\omega_z \end{pmatrix} \quad (5.103)$$

u_2 und u_3 hängen also von dem Systemzustand \vec{x} , den Vorsteuerungs-Eingängen $\xi_1 = p_{n,s}^{(4)}$, $\xi_2 = p_{e,s}^{(4)}$, $\xi_3 = p_{d,s}^{(4)}$ sowie von u_1 (5.101) und dessen Ableitung \dot{u}_1 (5.102) ab. Es muss durch Beschränkung der Eingangsgröße u_1 sichergestellt werden, dass $u_1 \neq 0$ ist (u_1 ist im Mittel bzw. im Schwebeflug gleich g).

Drehbeschleunigungs-Eingang u_4

Die Zustandsrückführung für den Drehbeschleunigungs-Eingang u_4 ergibt sich aus (5.89) nach Ersetzen von $\ddot{\psi} = \ddot{y}_4$ durch ξ_4 gemäß (5.96):

$$\begin{aligned} u_4 = \xi_4 \frac{\cos \theta}{\cos \varphi} - u_3 \tan \varphi + 2(\omega_z^2 - \omega_y^2) \sin \varphi \tan \theta - \omega_x\omega_y \\ + \omega_x\omega_z \tan \varphi - 2\omega_y\omega_z \tan \theta (\cos \varphi - \sin \varphi \tan \varphi) \end{aligned} \quad (5.104)$$

Für Rollwinkel $-90^\circ < \varphi < 90^\circ$ ist $\cos \varphi \neq 0$. Die Zustandsrückführung für den Drehbeschleunigungs-Eingang u_4 hängt also von ξ_4 , dem zuvor in (5.103) berechneten Eingang u_3 und dem Systemzustand \vec{x} ab.

Lineare Systemdynamik

Die quasi-statische Zustandsrückführung gemäß (5.99), (5.101), (5.103), (5.104) hat die erwartete Form (5.97). Wie in Abschnitt 2.4.2 dargestellt, wird das flache System (5.39), (5.56) durch diese quasi-statische Zustandsrückführung in ein lineares System überführt, das ein exakt gleiches dynamisches Verhalten hat. Das lineare System liegt in Brunovský-Normalform vor, besteht also aus mehreren parallelen Integrator-Ketten, die nicht miteinander verkoppelt sind. Gemäß (2.98) haben die Integrator-Ketten jeweils die Länge κ_i . Mit (5.98) besteht der Nord- und Ost-Kanal also aus 4 Integratoren, der Vertikal- und

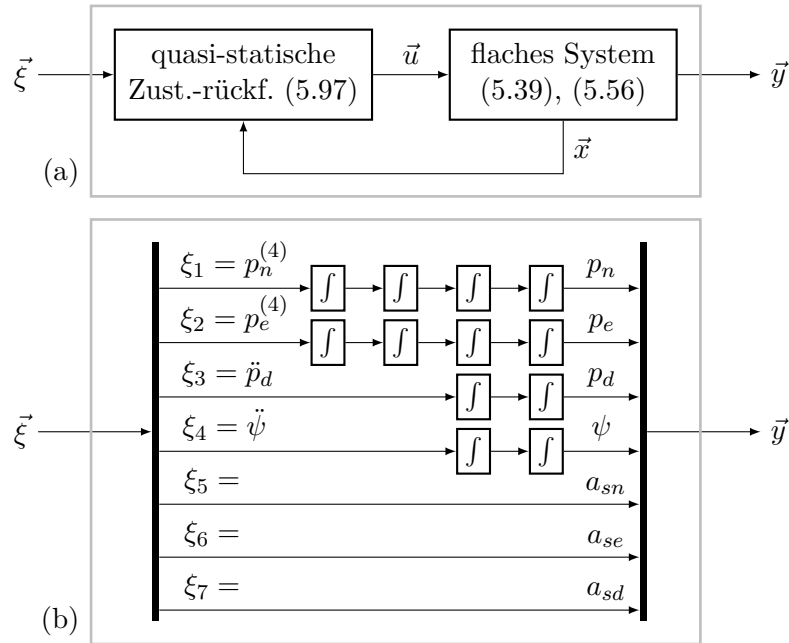


Abbildung 5.5: (a) Flaches System mit quasi-statischer Zustandsrückführung
 (b) lineares System in Brunovský-Normalf. mit exakt gleicher Dynamik

Gier-Kanal aus 2 und die Störgrößenkanäle aus 0 Integratoren, wie in Abbildung 5.5 dargestellt. Die Übertragungsfunktionen lassen sich in Laplace-Darstellung schreiben als:

$$\frac{Y_1(s)}{\Xi_1(s)} = \frac{Y_2(s)}{\Xi_2(s)} = \frac{1}{s^4} \quad \frac{Y_3(s)}{\Xi_3(s)} = \frac{Y_4(s)}{\Xi_4(s)} = \frac{1}{s^2} \quad \frac{Y_5(s)}{\Xi_5(s)} = \frac{Y_6(s)}{\Xi_6(s)} = \frac{Y_7(s)}{\Xi_7(s)} = 1 \quad (5.105)$$

Erforderliche Kräfte und Momente

Auf dem Quadropter werden die quasi-statischen Zustandsrückführungen als Vorsteuerung für die Stellgrößen-Eingänge u_1, u_2, u_3, u_4 realisiert. Eine Vorsteuerung der Störgrößen-Eingänge u_5, u_6, u_7 muss – wie oben erwähnt – entfallen, an ihre Stelle tritt die Störgrößenschätzung. Zudem muss berücksichtigt werden, dass der Störgrößen-Beobachter nur eine konstante Störkraft schätzt. Die in (5.102) benötigte Ableitung \hat{a}_{sd} und die in (5.103) erforderlichen zweiten Ableitungen $\hat{a}_{sn}, \hat{a}_{se}, \hat{a}_{sd}$ werden daher zu null gesetzt, vgl. (5.51). Bei sich änderndem Wind müssen die dadurch verursachten Fehler dann durch den Regler aus Abschnitt 5.2.3 ausgeglichen werden. Die implementierten Vorsteuerungen haben damit die Form:

$$u_1 = f(\vec{x}, \xi_3, \hat{a}_{sd}) \quad \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \vec{f}(\vec{x}, \xi_1, \xi_2, \xi_3, \dot{\xi}_3, u_1) \quad u_4 = f(\vec{x}, \xi_4, u_3) \quad (5.106)$$

Die Stellgrößeneingänge u_1, u_2, u_3, u_4 werden beschränkt, um sicherzustellen, dass für jeden Kanal ausreichend Reserve vorhanden ist, um Störungen ausregeln zu können:

$$u_{i,\text{sat}} = \text{sat}_{u_{i,\text{min}}}^{u_{i,\text{max}}}(u_i) \quad \forall i \in \{1, 2, 3, 4\} \quad (5.107)$$

Zur Auslegung der Beschränkungen siehe Tabelle 5.5 auf Seite 171 und zur Schreibweise $\text{sat}_{u_{\text{min}}}^{u_{\text{max}}}(u)$ siehe (B.12).

Die benötigte Gesamtschubkraft F_t wird aus $u_{1,\text{sat}}$ durch Auflösen von (5.30) nach F_t bei bekannter Masse m berechnet:

$$F_t = m u_{1,\text{sat}} \quad (5.108)$$

Durch Auflösen von (5.37) nach \vec{M}_t erhält man die von u_2, u_3, u_4 abhängigen, erforderlichen Drehmomente \vec{M}_t . Mit der Näherung (5.45) für \mathbf{J} gilt in Koordinatenschreibweise:

$$\vec{M}_t^b = \begin{pmatrix} M_{tx} \\ M_{ty} \\ M_{tz} \end{pmatrix} = \begin{pmatrix} J_x u_{2,\text{sat}} \\ J_y u_{3,\text{sat}} \\ J_z u_{4,\text{sat}} \end{pmatrix} \quad (5.109)$$

Um die individuellen Rotorschubkräfte F_{tk} zu ermitteln, werden (5.17) und (5.18) invertiert. Dazu werden in (5.18) die Widerstandsmomente M_{dk} gemäß (5.15)f. jeweils durch $k_d/k_t F_{tk}$ ersetzt und dann beide Gleichungen nach den F_{tk} aufgelöst. F_t und \vec{M}_t werden gemäß (5.108) und (5.109) ersetzt:

$$\begin{pmatrix} F_{t1} \\ F_{t2} \\ F_{t3} \\ F_{t4} \end{pmatrix} = \begin{pmatrix} 1/4 & 0 & 1/2 & -1/4 \\ 1/4 & 0 & -1/2 & -1/4 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & -1/2 & 0 & 1/4 \end{pmatrix} \begin{pmatrix} m u_{1,\text{sat}} \\ 1/d_r J_x u_{2,\text{sat}} \\ 1/d_r J_y u_{3,\text{sat}} \\ k_t/k_d J_z u_{3,\text{sat}} \end{pmatrix} \quad (5.110)$$

Die Identifikation der Trägheitsmomente J_x, J_y und $k_t/k_d J_z$ ist in Abschnitt 5.1.4 dargestellt.

Aus den Rotorschubkräften F_{tk} können dann die Motordrehzahl-Stellgrößen berechnet werden. Dazu werden zunächst mit den Rotorschub-Kalibrierungsfaktoren λ_k aus den benötigten Rotorschubkräften F_{tk} die zu kommandierenden Rotorschubkräfte F_{tsk} gemäß (5.24) berechnet:

$$F_{tsk} = \frac{1}{\lambda_k} F_{tk} \quad (5.111)$$

Um die Motordrehzahl-Stellgrößen ω_{nsk} zu erhalten, wird in (5.22) F_{tk} durch F_{tsk} ersetzt und dann nach ω_{nsk} aufgelöst. Die einzige positive, reellwertige Lösung für ω_{nsk} ist

$$\omega_{nsk} = -\frac{a_{t1}}{2a_{t2}} + \sqrt{\frac{a_{t1}^2}{4a_{t2}^2} - \frac{a_{t0}}{a_{t2}} + \frac{F_{tsk}}{a_{t2}}} \quad , \quad (5.112)$$

wenn $a_{t2} > 0$, $F_{tsk} > a_{t0} - a_{t1}^2/4a_{t2}$ und $F_{tsk} > a_{t0}$. Für die Parameter in Tabelle 5.2 ist dies erfüllt, wenn $F_{tsk} > a_{t0} = 0,38 \text{ N}$. Die normierte Motordrehrate für $F_{tsk} = 0,38 \text{ N}$ ist $\omega_{nsk} = 0,005$, was der Leerlaufdrehzahl $\omega_{nsk,\text{min}}$ entspricht, siehe (5.21).

In Abbildung 5.6 ist die Berechnung der Motordrehzahl-Stellgrößen $\omega_{ns1} \dots \omega_{ns4}$ aus den Stellgrößen $u_1 \dots u_4$ gemäß (5.107) bis (5.112) grafisch dargestellt.

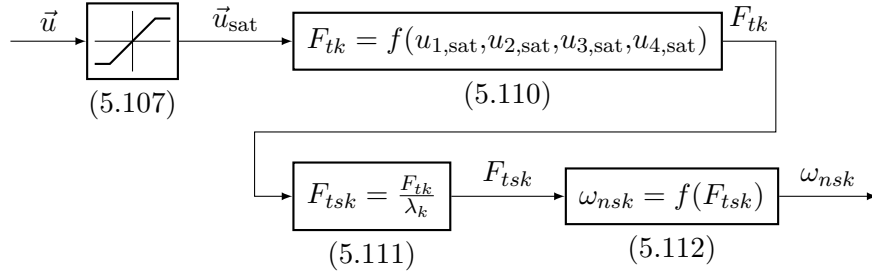


Abbildung 5.6: Berechnung der Motordrehzahl-Stellgrößen $\omega_{ns1} \dots \omega_{ns4}$ aus $u_1 \dots u_4$

Die kommandierbare Schubkraft jedes Rotors ist nach oben und unten beschränkt. Das Minimum und Maximum lässt sich rechnerisch aus (5.22) mit den Parametern in Tabelle 5.2 und $\omega_{nsk,\min}$, $\omega_{nsk,\max}$ aus (5.21) zu $F_{tsk,\min} = 0,38 \text{ N}$ und $F_{tsk,\max} = 15,4 \text{ N}$ berechnen. Der tatsächliche maximale Schub ist abhängig vom Batterieladestand. Die einzelnen Kanäle in u_i müssen daher auch geeignet beschränkt werden, sodass jeder Kanal über ausreichend Reserve zur Ausregelung von Störungen verfügt, auch wenn auf einem anderen Kanal Trajektorien eingeregelt werden. So wird vermieden, dass der Quadrocopter z. B. seine Höhe nicht mehr halten kann, weil Nick-, Roll- oder Giertrajektorien geflogen werden. In Tabelle 5.5 sind die Beschränkungen $u_{i,\min}$ und $u_{i,\max}$ der Kanäle sowie die dazu benötigten Änderungen der Rotorschubkräfte pro Rotor $\Delta F_{tk,\min}$ und $\Delta F_{tk,\max}$ gegeben. Die Zahlenwerte basieren auf den Quadrocopter-Parametern aus Tabelle C.1. Außerdem sind die Zahlenwerte für \bar{u}_i und $\Delta \bar{F}_{tk}$ im Schwebeflug aufgelistet.

u_i	\bar{u}_i	$u_{i,\min}$	$u_{i,\max}$	$\Delta \bar{F}_{tk}$	$\Delta F_{tk,\min}$	$\Delta F_{tk,\max}$
u_1	9,81 m/s ²	0,68 m/s ²	12 m/s ²	5,6 N	0,38 N	6,8 N
u_2	0	$\pm 40 \text{ rad/s}^2$		0	$\pm 3,6 \text{ N}$	
u_3						
u_4	0	$\pm 8 \text{ rad/s}^2$		0	$\pm 1,8 \text{ N}$	

Tabelle 5.5: Beschränkungen für \vec{u} und resultierende Rotorschubkräfte

Ein zu $u_{1,\max}$ symmetrischer Zahlenwert um den Schwebeflugwert \bar{u}_1 für $u_{1,\min}$ wäre 7,62 m/s² und das entsprechende $\Delta F_{tk,\min}$ dann 4,3 N. $\Delta F_{tk,\min}$ ist jedoch gleich dem Leerlauf-Schub $F_{tsk,\min}$, um beim Starten und Landen die Rotoren gleichmäßig hoch- und runterlaufen lassen zu können. $u_{1,\min}$ ist daher 0,68 m/s². Es ist zu beachten, dass die Kommandos in u_2 und u_3 jeweils immer nur auf zwei Rotoren bzw. zwei ΔF_{tk} wirken. Mit den in der Tabelle gegebenen Werten werden die oben angegebenen Rotorschubbeschränkungen normalerweise eingehalten.

5.2.3 Trajektorienfolgeregelung

Das flache System erhält durch die quasi-statische Zustandsrückführung eine lineare Dynamik in Form von Integrator-Ketten (5.105). Integrator-Ketten sind per se nicht stabil. Die Trajektorienfolgeregelung muss also diese Integrator-Ketten stabilisieren. Zudem sollte die Regelung Störungen unterdrücken und kleinere Modellierungsungenauigkeiten kompensieren. Es wird die Reglerstruktur mit proportionalen Rückführungen, die in (2.99) eingeführt und auch in [11] verwendet wird, genutzt. Hier wird jedoch eine kaskadierte anstatt einer parallelen Struktur gewählt, um Beschränkungen einführen zu können.

Die lineare Dynamik des Nord- und Ost-Kanals besteht von ξ_1 bis y_1 und von ξ_2 bis y_1 jeweils aus vier Integriern ($\kappa_1 = \kappa_2 = 4$), die des Vertikal- und Gier-Kanals von ξ_3 bis y_3 und von ξ_4 bis y_4 jeweils aus zwei ($\kappa_3 = \kappa_4 = 2$), siehe (5.105) und Abbildung 5.5. Daher werden für die Nord- und Ost-Kanäle Reglerkaskaden mit vier und für die Vertikal- und Gier-Kanäle Reglerkaskaden mit zwei Proportionalreglern benötigt. Da die quasi-statische Zustandsrückführung die Systemkanäle entkoppelt, können sie unabhängig voneinander geregelt werden. Die Regelstrecke und die kaskadierte Reglerstruktur sind für einen Kanal in Abbildung 5.7 dargestellt.

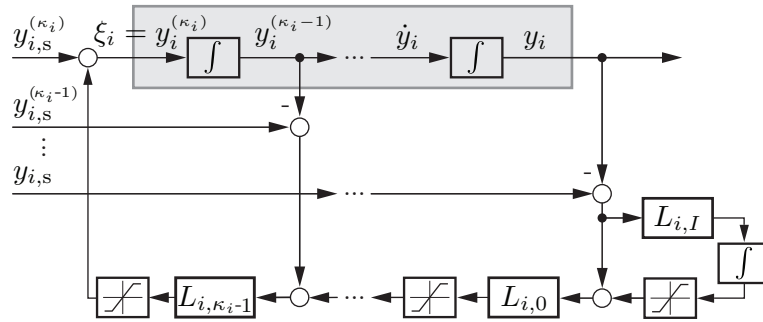


Abbildung 5.7: Stabilisierter Systemkanal bestehend aus Integrator-Ketten-Strecke (grauer Hintergrund) und kaskadiertem Regler mit Beschränkungen; nur der innere und äußere Regelkreis sind dargestellt.

Die Regler gemäß (2.99) lassen sich wie folgt schreiben mit $\kappa_i = 4$ für den Nord- und Ost-Kanal $i \in \{1, 2\}$ und $\kappa_i = 2$ für den Vertikal- und Gier-Kanal $i \in \{3, 4\}$:

$$\xi_i = y_{i,s}^{(\kappa_i)} + h_{i,\kappa_i,\text{sat}} \quad (5.113a)$$

$$h_{i,j+1} = L_{i,j}(y_{i,s}^{(j)} - \hat{y}_i^{(j)} + h_{i,j,\text{sat}}) \quad \forall j \in \{0, 1, \dots, \kappa_i - 1\} \quad (5.113b)$$

$$h_{i,0} = \int L_{i,I}(y_{i,s} - \hat{y}_i) \quad (5.113c)$$

$$h_{i,j,\text{sat}} = \text{sat}_{-h_{i,j,\text{lim}}}^{+h_{i,j,\text{lim}}}(h_{i,j}) \quad \forall j \in \{0, 1, \dots, \kappa_i\} \quad (5.113d)$$

Führungsgrößeneingänge für den Regler sind die Referenztrajektorien $y_{i,s}$ für die flachen Ausgänge und ihre Ableitungen bis zu den höchsten benötigten Ordnungen κ_i . Die

Führungsgrößen für die höchsten Ableitungen $y_{i,s}^{(\kappa_i)}$ wirken als Vorsteuerung direkt auf $\xi_1 \dots \xi_4$, siehe (5.113a). (5.113b) sind die vier bzw. zwei Proportionalregler pro Kanal mit Übertragungsbeiwerten $L_{i,j}$. Sie lassen sich mit $p_{i,j} = \prod_{k=1}^{\kappa_i-1} L_{i,k}$ in die Koeffizienten $p_{i,j}$ der parallelen Reglerstruktur (2.99) umrechnen. Die Führungsgrößen der Proportionalregler sind die niedrigeren Ableitungen der Referenztrajektorien und die Referenztrajektorien selbst.

Neben den proportionalen Rückführungen wird ein optionaler Integrierer mit Übertragungsbeiwerten $L_{i,I}$ in (5.113c) hinzugefügt, der eine bleibende Positions- oder Gierwinkel-Regelabweichung verhindern kann. Der Anti-Windup-Algorithmus ist hier nicht dargestellt.

Die (Hilfs-)Stellgrößen der Proportionalregler und des integrierenden Anteils $h_{i,j}$ werden in (5.113d) jeweils auf eine obere und untere Beschränkung $\pm h_{i,j,\text{lim}}$ begrenzt (zur Schreibweise siehe (B.12)). Durch die Einführung von Beschränkungen in die Rückführungen behalten die inneren Regelkreise eine Reserve zur Ausregelung von Störungen, wenn in den äußeren Regelkreisen für Position und Geschwindigkeit große Fehler auftreten. Sonst würden diese Fehler das ganze System sättigen. Im Fall von Gegenwind kann es vorkommen, dass Referenztrajektorien nicht eingehalten werden können. In diesem Fall wird das System den Regelfehler langsam reduzieren (mit den äußeren Beschränkungen in ihren Grenzen), während die inneren Regelkreise für Drehraten und Lage stabil bleiben. Die Zahlenwerte für die in den Flugversuchen (siehe Abschnitt 6.2) verwendeten Beschränkungen sind in Tabelle C.4 zu finden.

Bei idealer Systemdynamik – ohne Modellierungsfehler und Störungen – und konsistenten Führungsgrößen wäre die Rückführung aller Reglerkaskaden null. In diesem Fall hat das stabilisierte System die gleiche Dynamik wie die Regelstrecke (5.105).

Bei der Einstellung der Übertragungsbeiwerte $L_{i,j}$ ist zu beachten, dass im Modell der Regelstrecke z. B. die Motordynamik (vgl. Seite 149) und die Rotor-Aerodynamik nicht berücksichtigt wird. Daher lassen sich analytisch ausgelegte Übertragungsbeiwerte nicht direkt auf das reale System übertragen. Das Feintuning der Übertragungswerte der Regler erfolgt im Flugversuch.

Für die Berechnung von \dot{u}_1 gemäß (5.102) für (5.103) und (5.117)f. wird die Ableitung von ξ_3 benötigt. (5.103) erfordert außerdem die zweite Ableitung $\ddot{\xi}_3$. Dazu werden die erste und zweite Ableitung von (5.113) mit $i = 3$ und $\kappa_i = 2$ berechnet. Für die Ableitungen gilt mit $\gamma = 1$ für $\dot{\xi}_3$ und $\gamma = 2$ für $\ddot{\xi}_3$:

$$\xi_3^{(\gamma)} = y_{3,s}^{(2+\gamma)} + h_{3,2,\text{sat}}^{(\gamma)} \quad (5.114a)$$

$$h_{3,j+1}^{(\gamma)} = L_{3,j}(y_{3,s}^{(j+\gamma)} - \hat{y}_3^{(j+\gamma)} + h_{3,j,\text{sat}}^{(\gamma)}) \quad \forall j \in \{0,1\} \quad (5.114b)$$

$$h_{3,0}^{(\gamma)} = L_{3,I}(y_{3,s}^{(\gamma-1)} - \hat{y}_3^{(\gamma-1)}) \quad (5.114c)$$

$$h_{3,j,\text{sat}}^{(\gamma)} = \begin{cases} h_{3,j}^{(\gamma)} & \text{für } -h_{3,j,\text{lim}} \leq h_{3,j} \leq h_{3,j,\text{lim}} \\ 0 & \text{sonst} \end{cases} \quad \forall j \in \{0,1,2\} \quad (5.114d)$$

(5.114d) ist die erste und zweite Ableitung der Beschränkung (5.113d): Wenn eine der Beschränkungen aktiv ist, sind $\dot{h}_{3,j,\text{sat}}$ und $\ddot{h}_{3,j,\text{sat}}$ gleich null. Bei der zweiten Ableitung wird ein Impuls in $\ddot{h}_{3,j,\text{sat}}$ bei Aktivierung und Deaktivierung der Beschränkung vernachlässigt.

Als Führungsgrößen für den Vertikal-Kanal werden also nicht nur gemäß (5.113a) die Ableitungen der Trajektorien bis zur zweiten Ableitung $y_{3,s}^{(2)}$ benötigt, sondern mit (5.114a) auch die dritte und vierte Ableitung, $y_{3,s}^{(3)}$ und $y_{3,s}^{(4)}$. Damit lauten die Führungsgrößen für die vier Kanäle:

$$\vec{y}_{1,s} = \left(y_{1,s}, \dot{y}_{1,s}, \ddot{y}_{1,s}, y_{1,s}^{(3)}, y_{1,s}^{(4)} \right)^T = \left(p_{n,s}, \dot{p}_{n,s}, \ddot{p}_{n,s}, p_{n,s}^{(3)}, p_{n,s}^{(4)} \right)^T \quad (5.115a)$$

$$\vec{y}_{2,s} = \left(y_{2,s}, \dot{y}_{2,s}, \ddot{y}_{2,s}, y_{2,s}^{(3)}, y_{2,s}^{(4)} \right)^T = \left(p_{e,s}, \dot{p}_{e,s}, \ddot{p}_{e,s}, p_{e,s}^{(3)}, p_{e,s}^{(4)} \right)^T \quad (5.115b)$$

$$\vec{y}_{3,s} = \left(y_{3,s}, \dot{y}_{3,s}, \ddot{y}_{3,s}, y_{3,s}^{(3)}, y_{3,s}^{(4)} \right)^T = \left(p_{d,s}, \dot{p}_{d,s}, \ddot{p}_{d,s}, p_{d,s}^{(3)}, p_{d,s}^{(4)} \right)^T \quad (5.115c)$$

$$\vec{y}_{4,s} = \left(y_{4,s}, \dot{y}_{4,s}, \ddot{y}_{4,s} \right)^T = \left(\psi_s, \dot{\psi}_s, \ddot{\psi}_s \right)^T \quad (5.115d)$$

Die Ordnungen der höchsten Ableitungen $y_{i,s}^{(\beta_i)}$ mit $\beta_1 = \beta_2 = \beta_3 = 4$ und $\beta_4 = 2$ entsprechen den β_i aus (5.95).

Als (Hilfs-)Regelgrößen werden die gemessenen bzw. geschätzten flachen Ausgänge und deren Ableitungen bis zur dritten Ableitung $\hat{y}_i^{(3)}$ $i \in \{1,2,3\}$ für den Nord-, Ost- und Vertikalkanal sowie bis zur ersten Ableitung \hat{y}_4 für den Gierkanal benötigt, siehe (5.113b) und (5.114b). Die Positionen $(\hat{y}_1, \hat{y}_2, \hat{y}_3)^T = (\hat{p}_n, \hat{p}_e, \hat{p}_d)^T$ und Geschwindigkeiten $(\hat{y}_1, \hat{y}_2, \hat{y}_3)^T = (\hat{v}_n, \hat{v}_e, \hat{v}_d)^T$ können aus den Schätzwerten der Sensorfusion gemäß (5.10) und (5.11) berechnet werden. Das gleiche gilt für Gierwinkel $\hat{y}_4 = \hat{\psi}$ und Gierrate $\hat{y}_4 = \dot{\hat{\psi}}$, die unter Nutzung von (2.13), (2.16) und (B.8) in Form von (5.7) und (5.8) vorliegen.

Die zweiten zeitlichen Ableitungen der Positionsausgänge, die Beschleunigungen, werden gemäß (5.39b) geschätzt als:

$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{pmatrix} = \begin{pmatrix} \hat{p}_n \\ \hat{p}_e \\ \hat{p}_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - u_1 \begin{pmatrix} \hat{c}_{13} \\ \hat{c}_{23} \\ \hat{c}_{33} \end{pmatrix} + \begin{pmatrix} \hat{a}_{sn} \\ \hat{a}_{se} \\ \hat{a}_{sd} \end{pmatrix} \quad (5.116)$$

Es werden nicht die geschätzten Beschleunigungen (5.14) direkt benutzt, um die vom Störkraft-Beobachter (Abschnitt 5.1.5) geschätzten Störungen \hat{a}_s explizit berücksichtigen zu können.

Für den geschätzten Ruck in Nord- und Ostrichtung wird \ddot{a}_{lb} aus (5.76) analog zu (B.15) auf \vec{l}_n und \vec{l}_e projiziert, um $p_n^{(3)} = \ddot{a}_{lb} \cdot \vec{l}_n$ und $p_e^{(3)} = \ddot{a}_{lb} \cdot \vec{l}_e$ zu erhalten:

$$\begin{pmatrix} \hat{y}_1^{(3)} \\ \hat{y}_2^{(3)} \end{pmatrix} = \begin{pmatrix} \hat{p}_n^{(3)} \\ \hat{p}_e^{(3)} \end{pmatrix} = -\dot{u}_1 \begin{pmatrix} \hat{c}_{13} \\ \hat{c}_{23} \end{pmatrix} - u_1 \begin{pmatrix} \hat{c}_{11}\hat{\omega}_y - \hat{c}_{12}\hat{\omega}_x \\ \hat{c}_{21}\hat{\omega}_y - \hat{c}_{22}\hat{\omega}_x \end{pmatrix} + \begin{pmatrix} \hat{a}_{sn} \\ \hat{a}_{se} \end{pmatrix} \quad (5.117)$$

mit \dot{u}_1 aus (5.102). Die Ableitung der Störgrößen \hat{a}_{sn} , \hat{a}_{se} , \hat{a}_{sd} wird wieder zu null gesetzt, vgl. (5.51). Für den geschätzten Ruck in vertikaler Richtung gilt:

$$\hat{y}_3^{(3)} = \hat{p}_d^{(3)} = \dot{\xi}_3 \quad (5.118)$$

mit $\dot{\xi}_3$ aus (5.114a).

5.3 Trajektoriengenerierung

In diesem Abschnitt werden drei Methoden zur Trajektoriengenerierung für den Quadroptopter sowie die Steuerung des Kamera-Gimbals vorgestellt. Die Trajektoriengenerierungsmethoden eignen sich jeweils für einen oder mehrere der in Abschnitt 3.3.1 definierten Betriebsmodi des Systems.

Alle Trajektoriengenerierungsmethoden müssen folgende Anforderungen erfüllen:

- Erzeugung von Trajektorien sowie deren Ableitungen bis zur 4. bzw. 2. Ordnung für die Positionen p_n , p_e , p_d und den Gierwinkel ψ als Führungsgrößeneingang für die flachheitsbasierte Regelung aus Abschnitt 5.2, siehe (5.115)
- Berücksichtigung von Zustands- und Aktorbeschränkungen wie maximale Geschwindigkeiten, maximale Nick- und Rollwinkel, maximale Drehraten sowie maximaler und minimaler Schub der Rotoren
- Reaktion auf die Kommandos des Operators in den verschiedenen Betriebsmodi in Echtzeit

Für die Assistenzfunktion zur Objektverfolgung, bei dem ein Zielobjekt mit dem Quadroptopter und/oder mit der Kamera verfolgt werden soll, wird in Abschnitt 5.3.1 die Positionierung des Quadroptopters relativ zum Zielobjekt und die Ausrichtung der Kamera zum Zielobjekt beschrieben. Letzteres wird auch für die Ausrichtung der Kamera auf einen Point-of-Interest benutzt.

In Abschnitt 5.3.2 werden Trajektorien auf Basis von Spline-Kurven berechnet, die direkt auch die benötigten Ableitungen enthalten. Dabei werden die 4. Ableitungen der Trajektorien und damit die benötigten Rotorschubänderungen minimiert. Die so generierten Trajektorien können entweder direkt kommandiert oder auch für den Anflug von statischen Wegpunkten genutzt werden. Beim Wegpunkt-Flug wird die offline berechnete Trajektorie onboard skaliert, um einen vom Kamera-Operator kommandierten Punkt unter Berücksichtigung der Beschränkungen anzufliegen.

Die zweite Methode in Abschnitt 5.3.3 nutzt Integrator-Ketten, um die Trajektorien mit Ableitungen zu generieren. Geregelt werden die Integrator-Ketten mit einer kaskadierten Proportionalreglerstruktur mit Beschränkungen in Anlehnung an die Trajektorienfolgeregelung aus Abschnitt 5.2.3. Diese Methode wird sowohl für die Steuerung des Quadroptopters mit dem Joystick als auch zur Objektverfolgung mit dem Quadroptopter eingesetzt.

Als dritte Methode wird eine modellprädiktive Regelung vorgestellt, die durch Regelung von Integrator-Ketten die Trajektorien und Ableitungen erzeugt, siehe Abschnitt 5.3.4. Die so generierten Trajektorien werden mit den Spline-Kurven der ersten Methode verglichen („Szenario 1“). Außerdem wird die Methode eingesetzt, um Trajektorien für den Quadropter relativ zu einem Point-Of-Interest zu erzeugen („Szenario 2“).

5.3.1 Verfolgung eines Zielobjekts

Im Objektverfolgungs-Betriebsmodus (vgl. Abschnitt 3.3.1) soll der Kamera-Quadropter ein Zielobjekt verfolgen, um aus geeigneter Perspektive Filmaufnahmen davon machen zu können. Dabei ist zwischen der Positionierung des Kamera-Quadropters relativ zum Zielobjekt und der Kamera-Ausrichtung darauf zu unterscheiden. Die Position relativ zum Zielobjekt kann der Operator durch Vorgabe des Abstands und der Himmelsrichtung, aus der es gefilmt werden soll, steuern. Die diesbezüglichen Definitionen werden im folgenden Abschnitt 5.3.1.1 eingeführt. Um die Kamera auf das Zielobjekt auszurichten, müssen die dazu geeigneten Kamera-Gimbalwinkel und der Quadropter-Gierwinkel bestimmt werden, siehe Abschnitt 5.3.1.2.

Bei bewegten Zielen sendet das Zielobjekt seine GNSS-Position per Funk⁷ an den Quadropter. Bei einem statischen Ziel (POI) kann dessen Position auch manuell vorgegeben werden. Die 3-D-Position des Zielobjekts im Koordinatensystem L wird als \vec{p}_t^l und die horizontale 2-D-Position als $\vec{p}_{t,ne}$ geschrieben:

$$\vec{p}_t^l = (p_{t,n}, p_{t,e}, p_{t,d})^T, \quad \vec{p}_{t,ne} = (p_{t,n}, p_{t,e})^T \quad (5.119)$$

5.3.1.1 Position relativ zum Zielobjekt

Der horizontale Abstand δ zwischen dem Quadropter an der Position \vec{p}_{lb} (5.1) und dem Zielobjekt an der Position \vec{p}_t^l (5.119) ist gegeben durch:

$$\delta = \sqrt{(p_n - p_{t,n})^2 + (p_e - p_{t,e})^2} \quad (5.120)$$

Der Azimut α ist definiert als die Himmelsrichtung – ausgedrückt als Winkel gegenüber Nord – aus der das Zielobjekt gesehen wird:

$$\alpha = \text{atan2}(p_e - p_{t,e}, p_n - p_{t,n}) \quad (5.121)$$

Ist $\alpha = 0^\circ$, wird das Zielobjekt aus Norden, bei $\alpha = 90^\circ$ aus Osten usw. gesehen.

Außerdem ist eine vertikale Positionierung des Quadropters relativ zur Höhe $p_{t,d}$ des Zielobjekts denkbar. Hier soll die Höhe des Quadropters jedoch nicht relativ zum Zielobjekt, sondern durch den Joystick gesteuert werden, siehe Abschnitt 5.3.3.1.

⁷siehe Abschnitte 3.4 und 3.1

5.3.1.2 Kameraausrichtung zum Zielobjekt

Die Kamera ist mit einem lagestabilisierten Gimbal, mit dem sie in zwei Achsen geschwenkt werden kann, unter dem Quadrokoopter befestigt, siehe Seite 67. Zudem ist eine Drehung der Kamera um die Hochachse durch Gieren des Quadrokoopters möglich. Im Folgenden werden die Gimbalwinkel und der Quadrokoopter-Gierwinkel bestimmt, die geeignet sind, die Kamera auf das Zielobjekt auszurichten. Dazu muss die optische Achse der Kamera durch die Zielposition \vec{p}_t^l (5.119) laufen.

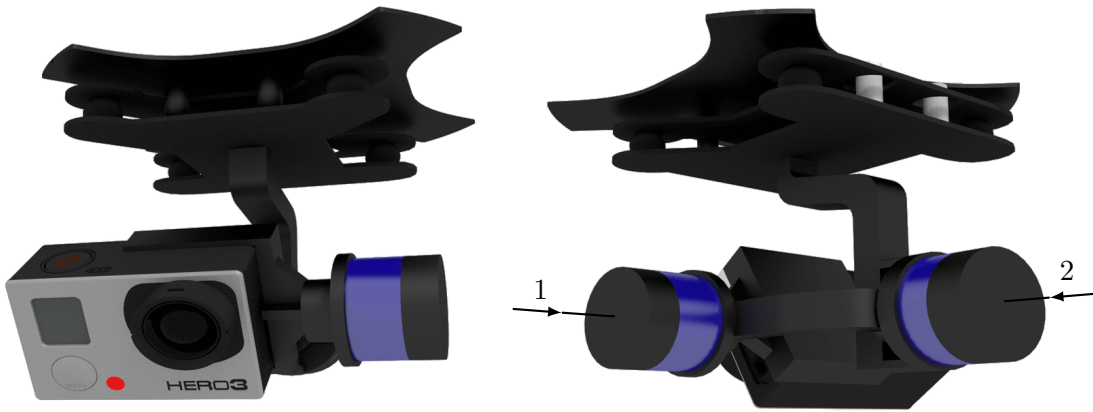


Abbildung 5.8: Vorder- und Rückansicht des Kamera-Gimbals. Die Drehachsen der beiden Servomotoren sind mit Vektorpfeilen gekennzeichnet.

In Abbildung 5.8 ist zu erkennen, dass die 1. Achse des Gimbals – von der Kamera aus gesehen – die Nickachse ist. Der Nick-Servomotor schwenkt daher direkt die Kamera. Die 2. Achse der kardanischen Aufhängung ist die Rollachse. Der Roll-Servomotor schwenkt neben der Kamera auch die Nickachse. Das Gimbal hat keinen Servomotor für die Gierachse. Daher muss der Quadrokoopter gieren, um die Kamera um die Gierachse zu schwenken. Das Gimbal ist mit einem festen Gierwinkel von

$$\psi_c = -45^\circ \quad (5.122)$$

am Quadrokoopter befestigt, die Kamera schaut also bei waagerechter optischer Achse nach „vorne links“.

Durch die Lagestabilisierung des Gimbals behält die Kamera ihre Lage gegenüber der Horizontalen, auch wenn der Quadrokoopter nickt oder rollt. Daher wird ein horizontiertes Hilfskoordinatensystem H eingeführt, das gegenüber dem erdfesten Koordinatensystem L um den aktuellen Quadrokoopter-Gierwinkel ψ und den festen Gimbal-Gierwinkel ψ_c gedreht ist:

$$\mathbf{C}_h^l = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \psi_c & -\sin \psi_c & 0 \\ \sin \psi_c & \cos \psi_c & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.123)$$

Um den Gimbal-Nickwinkel θ_c und den Gimbal-Rollwinkel φ_c zu definieren, werden zwei weitere Hilfskoordinatensysteme H' und H'' eingeführt:

$$\mathbf{C}_{h''}^h = \mathbf{C}_{h'}^h \mathbf{C}_{h''}^{h'} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi_c & -\sin \varphi_c \\ 0 & \sin \varphi_c & \cos \varphi_c \end{pmatrix} \begin{pmatrix} \cos \theta_c & 0 & \sin \theta_c \\ 0 & 1 & 0 \\ -\sin \theta_c & 0 & \cos \theta_c \end{pmatrix} \quad (5.124)$$

Das Koordinatensystem H wird mit der Rotation $\mathbf{C}_{h'}^h$ um die Achse $\vec{h}_x = \vec{h}'_x$ um den Rollwinkel φ_c in das Koordinatensystem H' gedreht. $\mathbf{C}_{h''}^{h'}$ beschreibt die Rotation des Koordinatensystems H' um die $\vec{h}'_y = \vec{h}''_y$ Achse um den Nickwinkel θ_c in das Kamerakoordinatensystem H'' . Der Basisvektor \vec{h}''_x des Kamerakoordinatensystems H'' zeigt parallel zur optischen Achse in Blickrichtung der Kamera. Die Basisvektoren \vec{h}''_y und \vec{h}''_z zeigen nach rechts und unten aus Sicht der Kamera.

Aufgrund der Gimbal-Lagestabilisierung erfolgt die Kommandierung des Gimbal-Nick- und Rollwinkels θ_c und φ_c gemäß der Definition (5.124) und nicht relativ zum Quadrocopter-Koordinatensystem B . θ_c und φ_c weichen daher auch von den durch die Servomotoren eingestellten Winkeln ab. Im Folgenden werden die zu kommandierenden Gimbal-Nick- und Rollwinkel sowie der zu kommandierende Quadrocopter-Gierwinkel bestimmt, die notwendig sind, um das Zielobjekt mit der Kamera zu verfolgen.

Der Vektor \vec{p}_{ct} von der Kamera zum Zielobjekt an der Position \vec{p}_t im lokalen, erdfesten Koordinatensystem L lautet:

$$\vec{p}_{ct}^l = \begin{pmatrix} p_{ct,n} \\ p_{ct,e} \\ p_{ct,d} \end{pmatrix} = \vec{p}_t^l - (\vec{p}_{tb}^l + \mathbf{C}_b^l \vec{d}_c^b) \quad (5.125)$$

Dabei ist \vec{p}_{tb}^b (5.1) die Position des Quadrocopter-Zentrums Z und \vec{d}_c^b der Vektor vom Quadrocopter-Zentrum zur Kamera (bei $\theta_c = 0$ und $\varphi_c = 0$), siehe Tabelle C.1.

Im Koordinatensystem H lautet der Vektor \vec{p}_{ct} :

$$\vec{p}_{ct}^h = \begin{pmatrix} p_{ct,x} \\ p_{ct,y} \\ p_{ct,z} \end{pmatrix} = \mathbf{C}_h^{l,T} \vec{p}_{ct}^l \quad (5.126)$$

ζ_c wird definiert als der Winkel zwischen der Horizontalen und dem Vektor \vec{p}_{ct} von der Kamera zum Zielobjekt:

$$\zeta_c = \text{atan} \left(\frac{-p_{ct,d}}{\sqrt{p_{ct,n}^2 + p_{ct,e}^2}} \right) \quad (5.127)$$

Aufgrund der Anordnung der Gimbalachsen kann das Zielobjekt nur dann sinnvoll durch Änderung des Gimbal-Rollwinkels φ_c verfolgt werden, wenn sich das Zielobjekt unterhalb des Quadrocopters befindet ($\zeta_c \rightarrow -90^\circ$). Dann kann durch Änderung des Rollwinkels

nach rechts und links geschaut werden. Ist das Zielobjekt hingegen am Horizont ($\zeta_c \rightarrow 0$), führt eine Änderung von φ_c zu einem Rollen der Kamera und damit zu einem schrägen Horizont im Kamerabild, was meist nicht erwünscht ist. Wenn das Zielobjekt am Horizont ist, kann das Zielobjekt horizontal durch Änderung des Quadrokopter-Gierwinkels verfolgt werden, siehe (5.130). Der Gimbal-Rollwinkel φ_c zur Verfolgung des Zielobjekts wird berechnet als:

$$\varphi_c = \text{sat}_{-\varphi_{c,\text{lim}}^*}^{+\varphi_{c,\text{lim}}^*} \left(\text{atan2} \left(\frac{-p_{ct,y}}{p_{ct,z}} \right) \right) \quad \varphi_{c,\text{lim}}^* = \varphi_{c,\text{lim}} \text{ sat}_0^1 \left(\frac{\zeta_c - \zeta_{c,\text{lim},\varphi,2}}{\zeta_{c,\text{lim},\varphi,1} - \zeta_{c,\text{lim},\varphi,2}} \right) \quad (5.128)$$

Wenn $\zeta_c \leq \zeta_{c,\text{lim},\varphi,1} = -60^\circ$ ist, sich das Zielobjekt also in etwa unterhalb des Quadrokopters befindet, wird der Rollwinkel φ_c auf die baulich bedingten Rollwinkel-Grenzen $\pm\varphi_{c,\text{lim}} = \pm 30^\circ$ beschränkt. Für $\zeta_c \geq \zeta_{c,\text{lim},\varphi,2} = -30^\circ$, wenn sich der Quadrokopter in der Nähe des Horizontes befindet, wird kein Rollen mehr zugelassen: $\varphi_c = 0$. Zwischen den beiden Grenzen, für $\zeta_{c,\text{lim},\varphi,1} < \zeta_c < \zeta_{c,\text{lim},\varphi,2}$, wird $\varphi_{c,\text{lim}}^*$ zwischen $\varphi_{c,\text{lim}}$ und 0 linear interpoliert.

Der zur Zielverfolgung notwendige Gimbal-Nickwinkel θ_c wird in Abhängigkeit vom kommandierten Gimbal-Rollwinkel φ_c berechnet als:

$$\theta_c = \text{sat}_{\theta_{c,\text{min}}}^{\theta_{c,\text{max}}} \left(\text{atan2} \left(\frac{-p_{ct,z}}{\cos \varphi_c}, p_{ct,x} \right) \right) \quad (5.129)$$

Dabei wird der Gimbal-Nickwinkel auf bauliche Grenzen von $\theta_{c,\text{min}} = -120^\circ$ und $\theta_{c,\text{max}} = 45^\circ$ beschränkt.

Der letzte Freiheitsgrad zur Verfolgung des Zielobjekts ist der Quadrokopter-Gierwinkel. Bei einem Quadrokopter-Gierwinkel von

$$\psi' = \text{atan2}(p_{ct,e}, p_{ct,n}) - \psi_c \quad , \quad (5.130)$$

einem Gimbal-Rollwinkel von $\varphi_c = 0$ und einem θ_c gemäß (5.129) zeigt die optische Achse der Kamera auf das Zielobjekt. Da die Verfolgung mit dem Gimbal-Rollwinkel schneller ist als die Quadrokopter-Gierdynamik, wird das Zielobjekt zuerst über den Gimbal-Rollwinkel verfolgt. Wenn der Quadrokopter-Gierwinkel dann eingeregelt wird, reduziert sich der Rollwinkel auf null (daher wird er hier zur Berechnung von ψ' vernachlässigt).

Es kann gewünscht sein, dass die Kamera nicht direkt auf das Zielobjekt gerichtet ist, sondern leicht links oder rechts vorbeischaute. Um dies zu beschreiben, wird der Winkel γ als horizontaler Winkel zwischen der optischen Achse und der Linie von der Kamera zum Zielobjekt definiert:

$$\gamma = \psi - \psi' = \psi + \psi_c - \underbrace{\text{atan2}(p_{ct,e}, p_{ct,n})}_{\approx \alpha + \pi} \quad (5.131)$$

Wenn $\gamma = 0$ ist, schaut die Kamera direkt zum Zielobjekt, für $\gamma > 0$ schaut sie rechts am Zielobjekt vorbei.

Wenn sich das Zielobjekt unter dem Quadrokopter befindet, kann es gut mit dem Gimbal verfolgt werden. Eine zusätzliche Verfolgung über den Quadrokopter-Gierwinkel bringt in diesem Fall keinen Vorteil. Bei Vorzeichenwechseln von $p_{ct,e}$ oder $p_{ct,n}$ in (5.130) würden 180° -Sprünge in ψ' auftreten. Daher wird das Zielobjekt durch Kommandierung des Quadrokopter-Gierwinkels nur verfolgt, wenn $\zeta_c \geq \theta_{\text{lim},\psi} = -70^\circ$, sonst wird der letzte kommandierte Gierwinkel gehalten. Der gewünschte Quadrokopter-Gierwinkel lautet daher:

$$\psi_{\text{des},k} = \begin{cases} \psi' + \gamma_{\text{des}} & \text{wenn } \zeta_c \geq \theta_{\text{lim},\psi} \\ \psi_{\text{des},k-1} & \text{sonst} \end{cases} \quad (5.132)$$

Dabei ist γ_{des} der Sollwert für den Offsetwinkel γ (5.131).

Bei der Berechnung der Roll-, Nick- und Gierwinkel werden Translationen der Kamera durch Änderungen der Gimbal-Winkel vernachlässigt. Da diese Translationen nur wenige Millimeter betragen, haben sie keinen relevanten Einfluss auf die Verfolgung des Zielobjekts.

Der Gimbal-Rollwinkel φ_c (5.128) und der Gimbal-Nickwinkel θ_c (5.129) werden an das Gimbal kommandiert. Aus dem gewünschten Quadrokopter-Gierwinkel $\psi_{\text{des},k}$ (5.132) wird in den folgenden Abschnitten eine geeignete Gier-Trajektorie berechnet.

5.3.2 Trajektoriengenerierung mit Spline-Kurven

Für die flachheitsbasierte Regelung aus Abschnitt 5.2 werden als Führungsgrößen (5.115) Trajektorien inklusive Ableitungen bis zur 4. bzw. 2. Ordnung benötigt. Eine Möglichkeit, Trajektorien zu bilden, deren Ableitungen analytisch berechnet werden können, sind Spline-Kurven [56]. Für die Trajektorien, bei denen die Ableitungen bis zur 4. Ableitung benötigt werden, sollen hier Splines 5. Grades verwendet werden, die 4-mal differenzierbar sind. Die m Zeitabschnitte dieser Splines werden aus Polynomen der Form

$$\begin{aligned} y_s(t + t_k) &= \sum_{j=0}^5 a_{kj} t^j & 0 \leq t < t_{k+1} & \quad \forall k \in \{1, \dots, m\} \\ &= a_{k5} t^5 + a_{k4} t^4 + a_{k3} t^3 + a_{k2} t^2 + a_{k1} t + a_{k0} \end{aligned} \quad (5.133)$$

zusammengesetzt. Es sind also $n_p = (5 + 1)m$ Polynom-Koeffizienten a_{kj} für die m Abschnitte zu bestimmen.

Die Zwangsbedingungen zwischen den Polynomen werden so gewählt, dass auch die 4. Ableitung der Splines stetig ist. Dadurch wird der kommandierte Schubverlauf ebenfalls stetig sein, sodass die Motoren dem Verlauf besser folgen können. Es ergeben sich also $n_v = 5(m - 1)$ Verbindungs-Zwangsbedingungen.

Auf diese Weise lassen sich Trajektorien durch mehrere Wegpunkte planen, die nacheinander angefliegen werden sollen. Dies lässt sich u. a. so erweitern, dass räumliche Beschränkungen durch Hindernisse o. Ä. beachtet werden [56].

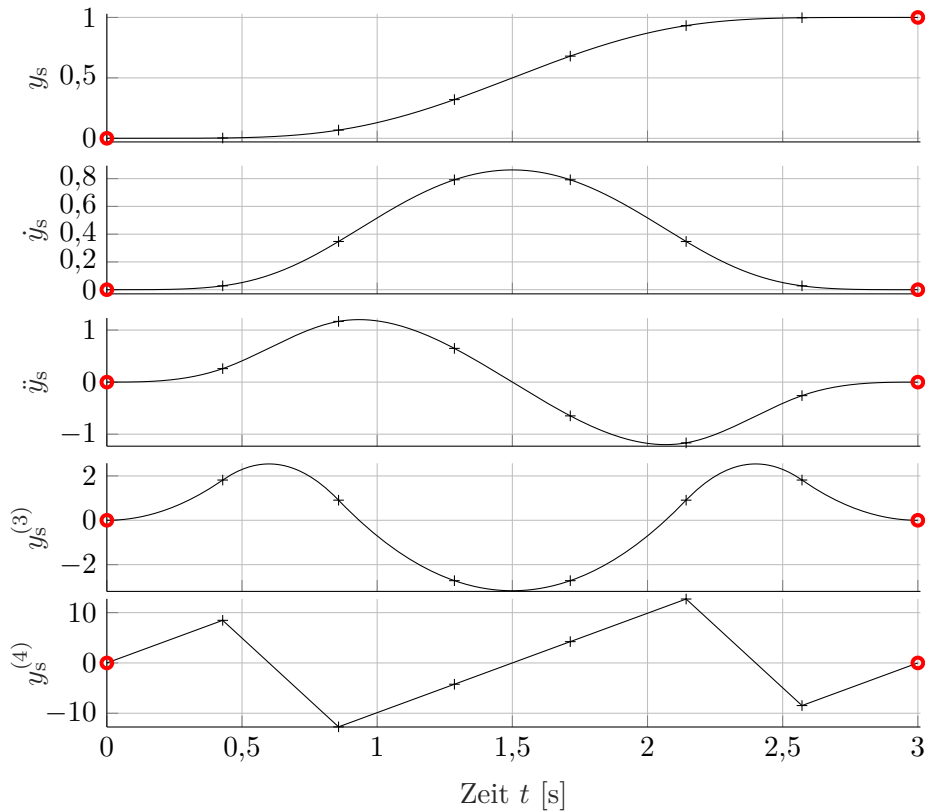


Abbildung 5.9: Spline 5. Grades inklusive Ableitungen. Die roten Kreise markieren Wert-Zwangsbedingungen, die schwarzen Kreuze Verbindungs-Zwangsbedingungen.

Hier soll als Beispiel eine Positionswechsel-Trajektorie zwischen zwei Punkten generiert werden, bei dem der Quadrocopter aus dem Schwebeflug heraus auf eine andere Position fliegt und dort dann wieder schwebt. Im Schwebeflug sind sämtliche Ableitungen der Trajektorie null: $y_s^{(j)} = 0$ für $j = 1 \dots 4$. Zusammen mit den beiden Positions-Vorgaben ergeben sich also als Anfangs- und Endbedingungen $n_w = 2 \cdot 5$ Wert-Zwangsbedingungen. Aus der Zahl der zu wählenden Koeffizienten n_p und der Zahl der Zwangsbedingungen $n_v + n_w$ ergeben sich $n_f = n_p - n_v - n_w = m - 5$ Freiheitsgrade, die hier dazu genutzt werden sollen, die 4. Ableitung $y_s^{(4)}$, also die Ableitung des Rucks (englisch snap) [56], zu minimieren. Dadurch wird zugleich der notwendige Differenz-Schub zwischen den Rotoren minimiert (vgl. (5.87), (5.109)f.). Der hier gewählte Spline mit $m = 7$ Elementen hat also $n_f = 2$ Freiheitsgrade. Die Koeffizienten a_{kj} werden durch Lösen des resultierenden quadratischen Optimierungsproblems mit den genannten Gleichheits-Nebenbedingungen bestimmt (vgl. (5.162)). Die berechnete Positionswechsel-Trajektorie $y_s(t)$ über eine Entfernung von einem Meter in einer Zeit von 3 s ist in Abbildung 5.9 zu sehen.

Die generierte Trajektorie kann u. a. dazu verwendet werden, einen statischen Punkt anzufliegen. Dazu werden zwei Trajektorien $\vec{y}_{1,s}$, $\vec{y}_{2,s}$ für die Nord- und Ost-Richtung gebildet

und passend zur Entfernung des Zielpunktes von der aktuellen Position skaliert. Durch eine Skalierung der Zeit wird sichergestellt, dass alle Beschränkungen für Geschwindigkeit $\dot{y}_s^{(j)}$, Beschleunigung $\ddot{y}_s^{(j)}$, Ruck $y_s^{(3)}$ und $y_s^{(4)}$ eingehalten werden. Dadurch werden indirekt auch die Beschränkungen von Nick- und Rollwinkel, Drehraten und Drehbeschleunigungen eingehalten. Die so generierten Trajektorien $\vec{y}_{1,s}$, $\vec{y}_{2,s}$ sind dann der gesuchte Führungsgrößeneingang in (5.115) für die flachheitsbasierte Regelung.

Solche Spline-Trajektorien werden u. a. in der Simulation in Abschnitt 5.5 und in den Flugversuchen A1 und A2 in Abschnitt 6.2.2 verwendet.

Für statische Szenarien sind offline berechnete Spline-Trajektorien gut geeignet. Durch örtliche und zeitliche Skalierung können sie im Flug an die konkrete Mission angepasst werden, ohne das quadratische Optimierungsproblem onboard lösen zu müssen.

Bei dynamischen Szenarien geht das im Allgemeinen nicht mehr. So müssen die Anfangswerte der Trajektorien zum aktuellen Systemzustand passen und der gewünschte zukünftige Trajektorienverlauf muss bekannt sein. Außerdem muss die Trajektorie regelmäßig neu geplant werden können, wenn der Quadrocopter durch Störungen von der geplanten Trajektorie abweicht oder z. B. sich das Zielobjekt anders bewegt als prädiziert. Für die vorliegende Anwendung wird davon abgesehen, die Spline-Trajektorien durch Lösen des quadratischen Optimierungsproblems onboard zu berechnen. Bei der Trajektoriengenerierung mit MPC wird ein quadratisches Optimierungsproblem hingegen onboard gelöst, siehe Abschnitt 5.3.4.

5.3.3 Trajektoriengenerierung mit Kaskadenregelung

In diesem Abschnitt wird die Trajektoriengenerierung vorgestellt, bei der eine kaskadierte Reglerstruktur zum Einsatz kommt. Dabei werden die als Führungsgrößen (5.115) für die flachheitsbasierte Regelung benötigten Trajektorien $y_{i,s}$ und ihre Ableitungen $y_{i,s}^{(j)}$ durch β_i -fache Integration der höchsten Ableitungen $y_{i,s}^{(\beta_i)}$ gebildet. Wie bei der Trajektorienfolgeregelung in Abschnitt 5.2.3 wird zur Stabilisierung und Regelung der Integrator-

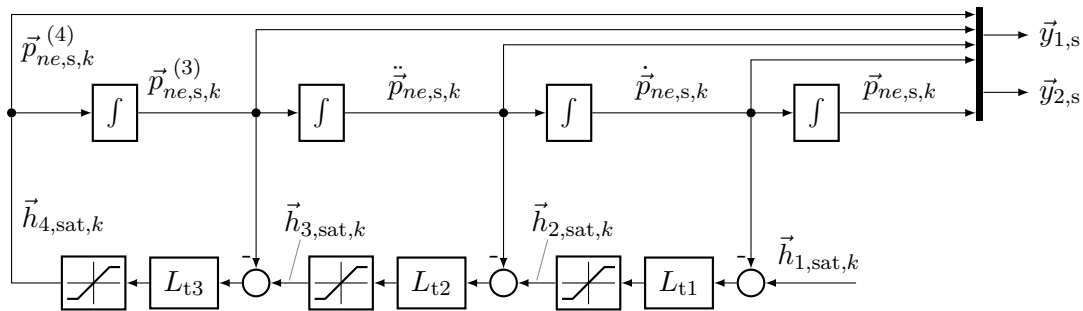


Abbildung 5.10: Trajektoriengenerierung horizontal (vgl. Abbildung 5.7 für die Trajektorienfolgeregelung)

Ketten eine kaskadierte Regelstruktur aus Proportionalreglern mit Beschränkungen genutzt. Die Regelstrecke ist hier nicht das reale System, es wird ein Systemmodell in Form der Integrator-Ketten geregelt. Wenn der Quadrokopter bei Störungen von den Trajektorien abweicht, sorgt die Trajektorienfolgeregelung aus Abschnitt 5.2.3 dafür, dass der Quadrokopter zu den Trajektorien zurückkehrt.

Zur Generierung der horizontalen Trajektorien $\vec{y}_{1,s}$, $\vec{y}_{2,s}$ in Nord- und Ostrichtung sind 4er Integrator-Ketten ($\beta_1 = \beta_2 = 4$) erforderlich. Die Struktur des Trajektoriengenerators ist in Abbildung 5.10 dargestellt. Die entsprechenden Gleichungen lauten (vgl. (5.113) für die Trajektorienfolgeregelung):

$$\vec{h}_{j+1,\text{sat},k} = \text{sat}_{-a_{\text{lim}}^{(j-1)}}^{+a_{\text{lim}}^{(j-1)}} \left(L_{tj} \left(\vec{h}_{j,\text{sat},k} - \vec{p}_{ne,s,k}^{(j)} \right) \right) \quad \forall j \in \{1, 2, 3\} \quad (5.134a)$$

$$\vec{p}_{ne,s,k}^{(4)} = \vec{h}_{4,\text{sat},k} \quad (5.134b)$$

$$\vec{p}_{ne,s,k+1}^{(j)} = \vec{p}_{ne,s,k}^{(j)} + \vec{p}_{ne,s,k}^{(j+1)} T \quad \forall j \in \{0, 1, 2, 3\} \quad (5.134c)$$

Der Sollgrößeneingang in die Trajektoriengenerierung ist die Geschwindigkeits-Hilfsstellgröße $\vec{h}_{1,\text{sat},k}$. Sie wird je nach Betriebsmodus für Joystick und Objektverfolgung unterschiedlich berechnet, siehe weiter unten in (5.139) und (5.146). (5.134a) sind die drei Proportionalregler pro Kanal mit den Übertragungsbeiwerten L_{tj} und der Beschränkung (B.13) der Hilfsstellgröße $\vec{h}_{j+1,\text{sat},k}$ auf $\pm a_{\text{lim}}^{(j-1)}$ für die nächsthöhere Kaskade. In (5.134c) sind die diskreten Integratoren dargestellt. Zur Berechnung der Position $\vec{p}_{ne,s,k}$ wird in allen Betriebsmodi derselbe Integrator benutzt, sodass es beim Umschalten der Modi nicht zu einem Positions-Sollwert-Sprung kommt. Bei der Initialisierung wird sein Startwert auf die aktuelle Ist-Position $\vec{p}_{ne,k}$ gesetzt. Die $\vec{p}_{ne,s,k}^{(j)}$ aus (5.134) bilden dann die Führungsgrößen $\vec{y}_{1,s}$, $\vec{y}_{2,s}$ in (5.115):

$$\begin{pmatrix} \vec{y}_{1,s} & \vec{y}_{2,s} \end{pmatrix} = \begin{pmatrix} y_{1,s} & y_{2,s} \\ \dot{y}_{1,s} & \dot{y}_{2,s} \\ \ddot{y}_{1,s}^{(3)} & \ddot{y}_{2,s}^{(3)} \\ y_{1,s}^{(4)} & y_{2,s}^{(4)} \\ y_{1,s} & y_{2,s} \end{pmatrix} = \begin{pmatrix} \vec{p}_{ne,s,k}^T \\ \dot{\vec{p}}_{ne,s,k}^T \\ \ddot{\vec{p}}_{ne,s,k}^T \\ \ddot{\vec{p}}_{ne,s,k}^{(3)T} \\ \vec{p}_{ne,s,k}^T \\ \ddot{\vec{p}}_{ne,s,k}^{(4)T} \\ \vec{p}_{ne,s,k}^T \end{pmatrix} \quad (5.135)$$

Für die vertikale Führungsgröße $\vec{y}_{3,s}$ (5.115c) und die Gier-Führungsgröße $\vec{y}_{4,s}$ (5.115d) kann der gleiche Ansatz wie für die horizontalen Größen $\vec{y}_{1,s}$ und $\vec{y}_{2,s}$ mit Integrator-Ketten der Länge $\beta_3 = 4$ und $\beta_4 = 2$ verwendet werden. Hier sollen die Geschwindigkeits-Führungsgrößen $\dot{y}_{3,s}$ und $\dot{y}_{4,s}$ jedoch direkt an die Geschwindigkeitskommandos $v_{d,\text{cmd}}$ und $\dot{\psi}_{\text{cmd}}$ angeschlossen werden. Die Positions- und Winkel-Führungsgrößen $\vec{p}_{d,s,k}$ und $\psi_{s,k}$ werden dann wieder durch Integration berechnet:

$$\dot{p}_{d,s,k} = v_{d,\text{cmd},k} \quad p_{d,s,k+1} = p_{d,s,k} + \dot{p}_{d,s,k} T \quad (5.136)$$

$$\dot{\psi}_{s,k} = \dot{\psi}_{\text{cmd},k} \quad \psi_{s,k+1} = \psi_{s,k} + \dot{\psi}_{s,k} T \quad (5.137)$$

Die Giergeschwindigkeits-Sollgröße $\dot{\psi}_{\text{cmd},k}$ wird je nach Betriebsmodus für Joystick und Objektverfolgung unterschiedlich berechnet, siehe (5.141) und (5.147). Der Sollgrößeneingang für die Vertikalgeschwindigkeit $v_{d,\text{cmd},k}$ wird bei der Steuerung der Quadrocopter-Höhe per Joystick verwendet, siehe (5.140). Eine vertikale Objektverfolgung erfolgt durch Steuerung des Pitchwinkels des Kamera-Gimbals, siehe Abschnitt 5.3.1.2.

Die höheren Ableitungen der Führungsgrößen $\vec{y}_{3,s}$ und $\vec{y}_{4,s}$ (5.115c)f. werden zu null gesetzt:

$$\vec{y}_{3,s} = \left(y_{3,s}, \dot{y}_{3,s}, \ddot{y}_{3,s}, y_{3,s}^{(3)}, y_{3,s}^{(4)} \right)^T = \left(\vec{p}_{d,s,k}, \dot{\vec{p}}_{d,s,k}, 0, 0, 0 \right)^T \quad (5.138a)$$

$$\vec{y}_{4,s} = \left(y_{4,s}, \dot{y}_{4,s}, \ddot{y}_{4,s} \right)^T = \left(\psi_{s,k}, \dot{\psi}_{s,k}, 0 \right)^T \quad (5.138b)$$

5.3.3.1 Betriebsmodus Joystick

Für die horizontale Steuerung des Quadrocopters mit dem 6-Achs-Joystick (siehe Abschnitt 3.3.2) wird der Geschwindigkeits-Eingang $\vec{h}_{1,\text{sat},k}$ von (5.134a) aus den horizontalen Joystick-Auslenkungen $c_{x,k}$, $c_{y,k}$ berechnet:

$$\vec{h}_{1,\text{sat},k} = \begin{pmatrix} v_{n,\text{joy},k} \\ v_{e,\text{joy},k} \end{pmatrix} = v_{ne,\text{lim}} \begin{pmatrix} \cos \psi_{c,k} & -\sin \psi_{c,k} \\ \sin \psi_{c,k} & \cos \psi_{c,k} \end{pmatrix} \begin{pmatrix} c_{x,k} \\ c_{y,k} \end{pmatrix} \quad \text{mit } c_{x,k}, c_{y,k} \in [-1, 1] \quad (5.139)$$

Dabei kann in der QC Cam Ctrl App zwischen drei Modi gewählt werden (siehe Abbildung 3.19): Im Modus „North“ wird $\psi_{c,k} = 0$ gesetzt, sodass sich der Quadrocopter mit dem Joystick – unabhängig vom Gierwinkel ψ_k des Quadrocopters – in Nord- und Ostrichtung steuern lässt. Im Modus „QC front“ ist $\psi_{c,k} = \psi_k$ und der Quadrocopter fliegt nach vorne oder nach rechts aus Sicht des Quadrocopters, wenn der Joystick nach vorne oder rechts gedrückt wird. Im dritten Modus „Cam“, mit $\psi_{c,k} = \psi_k + \psi_c$, erfolgt die Steuerung um den Gimbal-Gierwinkel ψ_c (5.122) gedreht, sodass die Joystickbefehle relativ zur Blickrichtung der Kamera interpretiert werden. Das Joystick-Kommando wird mit jeweils $v_{ne,\text{lim}}$ skaliert, sodass bei maximaler Joystick-Auslenkung die maximale Geschwindigkeit kommandiert wird.

Die Zahlenwerte von $v_{ne,\text{lim}}$, der Beschränkungen a_{lim} , \dot{a}_{lim} , \ddot{a}_{lim} in (5.134a) und der Übertragungsbeiwerte L_{tj} im Joystick-Betriebsmodus sind in den Tabellen C.6 und C.5 gegeben. Die Übertragungsbeiwerte L_{tj} sind so ausgelegt, dass die kaskadierten Regelkreise einen Dämpfungsgrad $D \geq 1$ haben. Auch wenn einzelne Beschränkungen aktiv sind, bleiben die anderen Regelkreise stabil und stark gedämpft.

In Abbildung 5.11 ist eine generierte Trajektorie für ein beispielhaftes Joystick-Kommando in Nord-Richtung $v_{n,\text{joy},k}$ gegeben. Man erkennt die Berücksichtigung der Beschränkung in allen Ableitungen und die gute Dämpfung aller Größen.

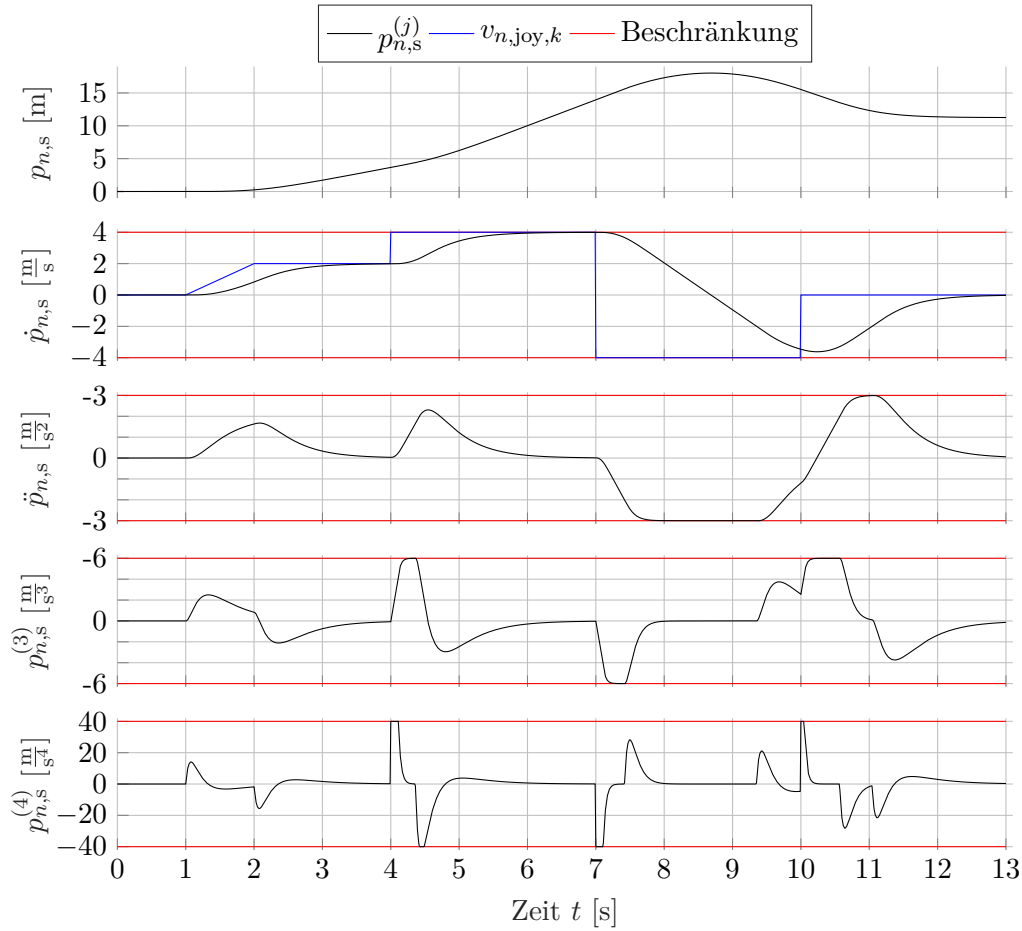


Abbildung 5.11: Generierte Trajektorie für Joystick-Kommandos

Um die Höhe und den Gierwinkel des Quadropters mit dem Joystick zu steuern, wird $v_{d,\text{cmd},k}$ in (5.136) und $\dot{\psi}_{\text{cmd},k}$ in (5.137) aus der vertikalen Joystick-Auslenkung $c_{z,k}$ und der Joystick-Drehung $c_{\psi,k}$ um die Hochachse berechnet:

$$v_{d,\text{cmd},k} = v_{d,\text{lim}} c_{z,k} \quad \text{mit } c_{z,k} \in [-1, 1] \quad (5.140)$$

$$\dot{\psi}_{\text{cmd},k} = \dot{\psi}_{\text{lim}} c_{\psi,k} \quad \text{mit } c_{\psi,k} \in [-1, 1] \quad (5.141)$$

Für die Zahlenwerte von $v_{d,\text{lim}}$ und $\dot{\psi}_{\text{lim}}$ siehe Tabelle C.6.

Die Erprobung des Joystick-Betriebsmodus im Flugversuch ist in Abschnitt 6.2.4 dargestellt.

5.3.3.2 Betriebsmodus Objektverfolgung

Im Objektverfolgungs-Betriebsmodus kann der Abstand δ (5.120) zum Zielobjekt und der Azimut α (5.121), aus dem das Zielobjekt gesehen wird, vorgegeben werden. Der Operator steuert die Sollwerte der Assistenzfunktion über die QC Cam Ctrl App, vgl. Abbildung 6.22 und Abschnitt 3.3.3.

Bei Variation des Abstands-Kommandos $\delta_{\text{cmd},k}$ fliegt der Quadrokofter auf das Zielobjekt zu bzw. entfernt sich von ihm. Bei Änderung des Azimut-Kommandos $\alpha_{\text{cmd},k}$ fliegt der Quadrokofter auf einer Kreisbahn mit aktuellem Radius δ_k um das Zielobjekt herum. Dazu müssen bei größeren Kommando-Sprüngen die Sollgrößen geeignet rampenförmig verändert werden, damit der Quadrokofter nicht auf kürzestem Weg zum neuen Zielpunkt fliegt, sondern auf einem Kreisbogen bleibt. Außerdem können durch die Sollwert-Rampen gewünschte Geschwindigkeitsbegrenzungen eingehalten werden.

Dazu werden die Kommandos $\delta_{\text{cmd},k}$ bzw. $\alpha_{\text{cmd},k}$ in ihrer Änderungsrate auf $\dot{\delta}_{\text{lim}}$ bzw. $\dot{\alpha}_{\text{lim},k}$ beschränkt. Zum Zeitpunkt k gilt für die resultierenden Kommandos $\delta_{r,k}$ und $\alpha_{r,k}$:

$$\delta_{r,k} = \begin{cases} \delta_{r,k-1} - \dot{\delta}_{\text{lim}} T & \Delta\delta < -\dot{\delta}_{\text{lim}} T \\ \delta_{\text{cmd},k} & \text{sonst} \\ \delta_{r,k-1} + \dot{\delta}_{\text{lim}} T & \Delta\delta > \dot{\delta}_{\text{lim}} T \end{cases} \quad \text{mit } \Delta\delta = \delta_{\text{cmd},k} - \delta_{r,k-1} \quad (5.142)$$

$$\alpha_{r,k} = \begin{cases} \alpha_{r,k-1} - \dot{\alpha}_{\text{lim},k} T & \Delta\alpha_w < -\dot{\alpha}_{\text{lim},k} T \\ \alpha_{r,k-1} + \Delta\alpha_w & \text{sonst} \\ \alpha_{r,k-1} + \dot{\alpha}_{\text{lim},k} T & \Delta\alpha_w > \dot{\alpha}_{\text{lim},k} T \end{cases} \quad \text{mit } \Delta\alpha_w = f_w(\alpha_{\text{cmd},k} - \alpha_{r,k-1}) \quad (5.143)$$

Bei der Berechnung von $\alpha_{r,k}$ muss beachtet werden, dass der Quadrokofter nicht unnötige Strecke um das Zielobjekt herum zurücklegt, um den Zielwert zu erreichen. Es ist nicht sinnvoll, dass der Quadrokofter z. B. 350° im Uhrzeigersinn um das Zielobjekt herum fliegt, wenn er auch 10° gegen den Uhrzeigersinn fliegen kann. Der Zielwert $\alpha_{\text{cmd},k}$ muss also als $\alpha_{\text{cmd},k} + 2\pi z$ interpretiert und $z \in \mathbb{Z}$ so gewählt werden, dass die Drehung minimal wird. Dazu wird in (5.143) $\Delta\alpha = \alpha_{\text{cmd},k} - \alpha_{r,k-1}$ mit der Funktion f_w (B.7) in einen äquivalenten Winkel $\Delta\alpha_w$ im Intervall $] -\pi; \pi]$ umgerechnet, wenn $\Delta\alpha$ außerhalb des Intervalls liegt.

Die Beschränkung der Azimutänderungsrate $\dot{\alpha}_{\text{lim},k}$ wird aus der Beschränkung $v_{\alpha,\text{lim}}$ der horizontalen Geschwindigkeit so berechnet, dass der Quadrokofter auf dem Kreisbogen um das Zielobjekt mit aktuellem Radius δ_k nicht schneller als $v_{\alpha,\text{lim}}$ fliegt:

$$\dot{\alpha}_{\text{lim},k} = \frac{v_{\alpha,\text{lim}}}{\delta_k} \quad (5.144)$$

Für die gewünschte Quadrokopter-Position $\vec{p}_{ne,des,k}$ zum Zeitpunkt k als Funktion der in der Änderungsrate beschränkten Kommandos $\delta_{r,k}$ und $\alpha_{r,k}$ sowie der Zielobjektposition $\vec{p}_{t,ne,k}$ gilt:

$$\vec{p}_{ne,des,k} = \vec{p}_{t,ne,k} + \delta_{r,k} \begin{pmatrix} \cos \alpha_{r,k} \\ \sin \alpha_{r,k} \end{pmatrix} \quad (5.145)$$

Um aus der gewünschten Quadrokopter-Position $\vec{p}_{ne,des,k}$ – wie für die Joystick-Kommandos – mit (5.134) eine Trajektorie zu berechnen, wird die Geschwindigkeits-Hilfsstellgröße $\vec{h}_{1,sat,k}$ als Eingang für (5.134a) wie folgt bestimmt:

$$\vec{h}_{1,k} = (h_{1,n,k}, h_{1,e,k})^T = L_{t0} (\vec{p}_{ne,des,k} - \vec{p}_{ne,s,k}) \quad (5.146a)$$

$$\zeta_k = \text{atan2}(h_{1,e,k}, h_{1,n,k}) \quad (5.146b)$$

$$\vec{h}_{1,sat,k} = \begin{cases} \vec{h}_{1,k} & \text{wenn } |\vec{h}_{1,k}| \leq v_{ne,lim} \\ v_{ne,lim} (\cos \zeta_k, \sin \zeta_k)^T & \text{sonst} \end{cases} \quad (5.146c)$$

Durch (5.146a) wird die kaskadierte Reglerstruktur in Abbildung 5.10 um einen weiteren Proportionalregler mit $\vec{p}_{ne,des,k}$ als Stellgrößeneingang erweitert. Die Beschränkung der Hilfsstellgröße auf die maximale Geschwindigkeit $v_{ne,lim}$ erfolgt nicht getrennt pro Kanal, sondern gleichmäßig auf beiden Kanälen, siehe (5.146c). Dadurch wird der Positionsfehler in direkter Linie zur gewünschten Position $\vec{p}_{ne,des,k}$ abgebaut.

Die Übertragungsbeiwerte L_{tj} der vier Proportionalregler sind wie bei der Joystick-Steuerung so ausgelegt, dass die kaskadierten Regelkreise einen Dämpfungsgrad $D \geq 1$ haben. Auch hier bleiben die Regelkreise stabil und stark gedämpft, wenn einzelne Beschränkungen aktiv sind. Die verwendeten Zahlenwerte der Übertragungsbeiwerte und der Beschränkungen δ_{lim} , $v_{\alpha,lim}$, $v_{ne,lim}$, a_{lim} , \dot{a}_{lim} , \ddot{a}_{lim} sind in den Tabellen C.5 bzw. C.6 angegeben.

Um den Quadrokopter zur Objektverfolgung um die Hochachse zu drehen, wird die Giergeschwindigkeits-Sollgröße $\dot{\psi}_{cmd,k}$ für (5.137) wie folgt berechnet:

$$\dot{\psi}_{cmd,k} = \text{sat}_{-\dot{\psi}_{lim}}^{+\dot{\psi}_{lim}} (L_{\psi} (f_w(\psi_{des,k} - \psi_{s,k}))) \quad (5.147)$$

Aus dem zur Objektverfolgung notwendigen Gierwinkel $\psi_{des,k}$ aus (5.132) und dem aktuellen Trajektorien-Gierwinkel $\psi_{s,k}$ (5.137) wird ein Gierwinkelfehler bestimmt und mit der Funktion f_w (B.7) in einen äquivalenten Winkel im Intervall $]-\pi; \pi]$ umgerechnet. Nach Multiplikation mit L_{ψ} wird die so ermittelte Giergeschwindigkeit auf $\pm \dot{\psi}_{lim}$ beschränkt (für $\dot{\psi}_{lim}$, L_{ψ} siehe Tabellen C.6, C.5).

Im Abschnitt 6.2.5 wird die Erprobung des Objektverfolgungs-Betriebsmodus im Flugversuch dargestellt.

5.3.4 Trajektoriengenerierung mit MPC

Als weitere Alternative zur Trajektoriengenerierung wird in diesem Abschnitt ein Ansatz auf Basis einer modellprädiktiven Regelung (MPC: Model Predictive Control) [53] vorgestellt.

Wie bei der Trajektoriengenerierung mit Kaskadenregelung in Abschnitt 5.3.3 werden auch hier die Trajektorien und deren Ableitungen, die als Führungsgrößeneingang (5.115) für die flachheitsbasierte Regelung benötigt werden, durch Integrator-Ketten gebildet. Hier regelt die MPC anstatt der kaskadierten Proportionalregler (5.134a) die Integrator-Ketten. Die MPC berücksichtigt ebenfalls Eingangs- und Zustandsbeschränkungen des Quadropters, sodass er den generierten Trajektorien folgen kann. Falls der Quadropter aufgrund von Störungen von den generierten Trajektorien abweicht, regelt auch hier der Trajektorienfolgeregler aus Abschnitt 5.2.3 die Störungen aus.

Der Vorteil der MPC ist, dass sie auch zukünftige Änderungen der Trajektorie berücksichtigen kann, solange diese prädiziert werden können. Alle $T_{\text{MPC}} = 0,05 \text{ s}$ wird der zukünftige Verlauf der Trajektorien optimiert.

Die Trajektoriengenerierung mit der MPC soll anhand von zwei Szenarien vorgestellt werden:

Im **Szenario 1** liegen geplante Trajektorien für die Position und den Gierwinkel vor. Der zukünftige Verlauf dieser Trajektorien ist der MPC hier bekannt. Die benötigten Ableitungen der Trajektorien müssen von der MPC berechnet werden.

Im **Szenario 2** gibt der Operator den Abstand δ zu einem Zielobjekt, den Azimut α , aus dem das Zielobjekt gesehen werden soll, und den Winkel γ vor. Hier sind die zukünftigen Sollgrößenänderungen unbekannt.

Linearisiertes Zustandsraummodell

Für die MPC wird ein zeitdiskretes, lineares Zustandsraummodell der Regelstrecke benötigt. Dazu wird im Folgenden das kontinuierliche, nichtlineare Modell der Regelstrecke aufgestellt, das linearisiert und diskretisiert wird.

Wie oben eingeführt, werden als Regelstrecke Integrator-Ketten verwendet, die die benötigten Trajektorien $y_{i,s}$ und Ableitungen $y_{i,s}^{(j)}$ durch β_i -fache Integration der höchsten Ableitungen $y_{i,s}^{(\beta_i)}$ berechnen. Die Integrator-Ketten haben wieder für den Nord-, Ost- und Vertikal-Kanal die Länge $\beta_1 = \beta_2 = \beta_3 = 4$ und für den Gier-Kanal die Länge $\beta_4 = 2$, vgl. (5.115). Die Integrator-Ketten werden als kontinuierliches System der Form

$$\dot{\vec{X}} = \mathbf{A} \vec{X} + \mathbf{B} \vec{U} \quad (5.148)$$

modelliert mit Eingang \vec{U} und Zustand \vec{X} :

$$\vec{U} = (U_1, U_2, U_3, U_4)^T = \left(p_{n,s}^{(4)}, p_{e,s}^{(4)}, p_{d,s}^{(4)}, \ddot{\psi}_s \right)^T \quad (5.149)$$

$$\vec{X} = \left(p_{n,s}, \dot{p}_{n,s}, \ddot{p}_{n,s}, p_{n,s}^{(3)}, p_{e,s}, \dot{p}_{e,s}, \ddot{p}_{e,s}, p_{e,s}^{(3)}, p_{d,s}, \dot{p}_{d,s}, \ddot{p}_{d,s}, p_d^{(3)}, \psi_s, \dot{\psi}_s \right)^T \quad (5.150)$$

Für die konstanten System- und Eingangsmatrizen \mathbf{A} und \mathbf{B} siehe (B.18).

Als Ausgänge \vec{Y} der Regelstrecke werden die Regelgrößen für Szenario 1 und 2 benötigt:

$$\vec{Y} = (p_{n,s}, p_{e,s}, p_{d,s}, \psi_s, \delta, \alpha, \gamma)^T \quad (5.151)$$

Neben den Zuständen $p_{n,s}$, $p_{e,s}$, $p_{d,s}$ und ψ_s enthält \vec{Y} für Szenario 2 auch die Größen δ , α und γ , die gemäß (5.120), (5.121) und (5.131) nichtlinear vom Zustand \vec{X} und von der Position des Zielobjekts \vec{p}_t (5.119) abhängen. Um den Ausgang \vec{Y} als lineare Funktion des Zustands \vec{X} auszudrücken, werden diese Zusammenhänge an einem Arbeitspunkt $(\vec{X}_A, \vec{U}_A, \vec{Y}_A)$ linearisiert. Dazu werden Abweichungsgrößen an diesem Arbeitspunkt definiert:

$$\vec{x} = \vec{X} - \vec{X}_A \quad \vec{u} = \vec{U} - \vec{U}_A \quad \vec{y} = \vec{Y} - \vec{Y}_A \quad (5.152)$$

Die partiellen Ableitungen von δ , α und γ bezüglich \vec{X} am Arbeitspunkt lauten:

$$\left[\frac{\partial \delta}{\partial p_{n,s}} \right]_A = \frac{p_{n,sA} - p_{t,nA}}{\delta_A} = c_{51} \quad \left[\frac{\partial \delta}{\partial p_{e,s}} \right]_A = \frac{p_{e,sA} - p_{t,eA}}{\delta_A} = c_{55} \quad (5.153)$$

$$\left[\frac{\partial \alpha}{\partial p_{n,s}} \right]_A = \frac{p_{t,eA} - p_{e,sA}}{\delta_A^2} = c_{61} \quad \left[\frac{\partial \alpha}{\partial p_{e,s}} \right]_A = \frac{p_{t,nA} - p_{n,sA}}{\delta_A^2} = c_{65} \quad (5.154)$$

$$\left[\frac{\partial \gamma}{\partial p_{n,s}} \right]_A = \frac{p_{e,sA} - p_{t,eA}}{\delta_A^2} = c_{71} \quad \left[\frac{\partial \gamma}{\partial p_{e,s}} \right]_A = \frac{p_{n,sA} - p_{t,n}}{\delta_A^2} = c_{75} \quad (5.155)$$

Damit gilt für die linearisierte Ausgangsgleichung am Arbeitspunkt:

$$\vec{y} = \mathbf{C} \vec{x} \quad \text{mit} \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ c_{51} & 0 & 0 & 0 & c_{55} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_{61} & 0 & 0 & 0 & c_{65} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_{71} & 0 & 0 & 0 & c_{75} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (5.156)$$

Das gesuchte zeitdiskrete Zustandsraummodell für die MPC hat die Form:

$$\vec{x}_{k+1} = \mathbf{A}_k \vec{x}_k + \mathbf{B}_k \vec{u}_k + \dot{\vec{X}}_A \quad (5.157)$$

$$\vec{y}_k = \mathbf{C}_k \vec{x}_k \quad (5.158)$$

\vec{x}_k , \vec{u}_k , \vec{y}_k sind wieder als Abweichungsgrößen gemäß (5.152) definiert. $\dot{\vec{X}}_A$ ist der Zahlenwert von (5.148) am Arbeitspunkt. Die Matrizen \mathbf{A}_k , \mathbf{B}_k werden gemäß Abschnitt 2.3.3

aus den konstanten Matrizen \mathbf{A} , \mathbf{B} (B.18) für die Abtastzeit T_{MPC} einmalig berechnet. Die Ausgangsmatrix \mathbf{C}_k des diskreten Modells ist gleich der Ausgangsmatrix \mathbf{C} (5.156) des kontinuierlichen Modells (vgl. (2.71)). Da die unteren drei Zeilen von \mathbf{C} und \mathbf{C}_k nicht konstant sind, werden sie zu jedem Zeitschritt neu berechnet.

Modellprädiktiver Regler

Das Ziel der modellprädiktiven Regelung ist es, die zukünftigen Abweichungen der Regelgrößen \vec{y}_k von den Referenzgrößen \vec{r}_k zu minimieren und dabei gleichzeitig die dazu notwendigen Stellgrößenänderungen $\Delta\hat{\vec{u}}_k$ gering zu halten. Dies wird durch die Kostenfunktion V_k gemäß [53] ausgedrückt als

$$V_k = \sum_{i \in H_y} \left\| \hat{y}_{k+i|k} - \vec{r}_{k+i|k} \right\|_{\mathbf{Q}(i)}^2 + \sum_{j \in H_u} \left\| \Delta\hat{u}_{k+j|k} \right\|_{\mathbf{R}(j)}^2 \quad (5.159)$$

mit der Prädiktion der Regelgrößen $\hat{y}_{k+i|k}$, Referenzgrößen $\vec{r}_{k+i|k}$ (relativ zum Arbeitspunkt \vec{Y}_A) und den berechneten Stellgrößenänderungen $\Delta\hat{u}_{k+j|k}$. Die Notation $\|\vec{a}\|_{\mathbf{M}}^2$ wird als Abkürzung für $\vec{a}^T \mathbf{M} \vec{a}$ verwendet. $\mathbf{Q}(i)$ ist die Gewichtungsmatrix, die die Abweichung der prädizierten Ausgänge von den Referenzgrößen zum Prädiktionsschritt i des Prädiktionshorizontes H_y bestraft. Die Gewichtungsmatrix $\mathbf{R}(j)$ bestraft die Stellgrößenänderungen zum Stellschritt j im Stellhorizont H_u . Beide Matrizen $\mathbf{Q}(i)$ und $\mathbf{R}(j)$ sind symmetrisch und positiv semidefinit. H_y und H_u müssen nicht alle Schritte zwischen den oberen und unteren Grenzen beinhalten [64]. Die untere Grenze von H_u ist normalerweise 0 (aktueller Zeitschritt).

Zur Berücksichtigung der Aktor- und Zustandsbeschränkungen werden Beschränkungen der Stellgröße \vec{U} und der Zustände \vec{X} eingeführt:

$$\vec{U}_{\min} \leq \vec{u}_{k+j} + \vec{U}_A \leq \vec{U}_{\max} \quad (5.160)$$

$$\vec{X}_{\min} - \vec{\epsilon}_x \leq \vec{x}_{k+i} + \vec{X}_A \leq \vec{X}_{\max} + \vec{\epsilon}_x \quad (5.161)$$

Während die Stellgrößenbeschränkungen harte Beschränkungen (hard constraints) sind, werden die Zustands-Beschränkungen durch Schlupfvariablen $\vec{\epsilon}_x$ (slack variables) als weiche Beschränkungen (soft constraints) implementiert, um die Lösbarkeit des Optimierungsproblems zu gewährleisten [53].

Die Kostenfunktion (5.159) sowie die harten und weichen Beschränkungen (5.160), (5.161) führen zu einem quadratischen Optimierungsproblem (quadratic program, QP) der Form [53]:

$$\min_{\vec{\theta}, \vec{\epsilon}} \frac{1}{2} \vec{\theta}^T \mathbf{\Phi} \vec{\theta} + \vec{\phi}^T \vec{\theta} + \rho_2 \|\vec{\epsilon}\|_2^2 + \rho_1 \|\vec{\epsilon}\|_1 \quad (5.162)$$

unter den Nebenbedingungen

$$\mathbf{\Omega}_h \vec{\theta} \leq \vec{\omega}_h \quad (5.163a)$$

$$\mathbf{\Omega}_s \vec{\theta} \leq \vec{\omega}_s + \vec{\epsilon} \quad (5.163b)$$

$$\vec{\epsilon} \geq \vec{0} \quad (5.163c)$$

Dieses quadratische Problem wird zu jedem Zeitschritt mit der Abtastzeit T_{MPC} durch einen „interior-point-convex“-Algorithmus gelöst. Die Ungleichung (5.163a) wird von den harten Stellgrößen-Beschränkungen (5.160) und die Ungleichung (5.163b) von den weichen Zustands-Beschränkungen (5.161) abgeleitet. Die Berechnung der Matrizen Φ , $\mathbf{\Omega}_h$, $\mathbf{\Omega}_s$ und der Vektoren $\vec{\phi}$, $\vec{\omega}_h$, $\vec{\omega}_s$ soll hier nicht dargestellt werden. Dazu wird auf [53] verwiesen.

$\vec{\epsilon}$ ist der Vektor aller Schlupfvariablen. Verletzungen der weichen Beschränkungen werden in (5.162) sowohl durch die quadrierte 2-Norm $\|\vec{\epsilon}\|_2$ als auch durch die 1-Norm $\|\vec{\epsilon}\|_1$ mit ausreichend hohen Gewichtungen ρ_2 and ρ_1 bestraft. Die Verwendung der 1-Norm führt zu einer exakten Bestrafung [74]. Wenn nur die 2-Norm genutzt würde, könnten Beschränkungen leicht verletzt werden, auch wenn die Verletzungen nicht notwendig sind [53].

Die optimale Lösung $\vec{\theta}$ des quadratischen Optimierungsproblems beinhaltet die optimale Stellgrößenänderung $\Delta\vec{u}_{k|k}$, aus der die Stellgröße \vec{U}_k (5.149) für den nächsten Stellschritt berechnet wird:

$$\vec{U}_k = \vec{U}_{k-1} + \Delta\vec{u}_{k|k} \quad (5.164)$$

Die Stellgröße \vec{U}_k ist der Eingang für die Integrator-Ketten (5.148), die die Trajektorien und deren Ableitungen erzeugen.

Der Zustand \vec{X}_k (5.150) und die Stellgröße \vec{U}_k bilden zusammen die Führungsgrößeneingänge (5.115) für die flachheitsbasierte Regelung:

$$\vec{y}_{1,s} = \left(p_{n,s}, \dot{p}_{n,s}, \ddot{p}_{n,s}, p_{n,s}^{(3)}, p_{n,s}^{(4)} \right)^T = (X_{1,k}, X_{2,k}, X_{3,k}, X_{4,k}, U_{1,k})^T \quad (5.165a)$$

$$\vec{y}_{2,s} = \left(p_{e,s}, \dot{p}_{e,s}, \ddot{p}_{e,s}, p_{e,s}^{(3)}, p_{e,s}^{(4)} \right)^T = (X_{5,k}, X_{6,k}, X_{7,k}, X_{8,k}, U_{2,k})^T \quad (5.165b)$$

$$\vec{y}_{3,s} = \left(p_{d,s}, \dot{p}_{d,s}, \ddot{p}_{d,s}, p_{d,s}^{(3)}, p_{d,s}^{(4)} \right)^T = (X_{9,k}, X_{10,k}, X_{11,k}, X_{12,k}, U_{3,k})^T \quad (5.165c)$$

$$\vec{y}_{4,s} = \left(\psi_s, \dot{\psi}_s, \ddot{\psi}_s \right)^T = (X_{13,k}, X_{14,k}, U_{4,k})^T \quad (5.165d)$$

Szenario 1: Positionswechsel-Trajektorie

Im ersten Szenario (vgl. Seite 188) liegt eine Solltrajektorie für die Position und den Gierwinkel vor. Die Ableitungen der Trajektorien werden von der MPC berechnet. Es wird eine Positionswechsel-Trajektorie wie in Abbildung 5.9 (Abschnitt 5.3.2), in Distanz und Zeit skaliert, als Ost-Solltrajektorie $p_{e,\text{ref}}$ genutzt. Als Gier-Solltrajektorie ψ_{ref} wird der gleiche Verlauf für eine Drehung um 45° vorgegeben. Da diese Trajektorien realisierbar sind, sollte die MPC sie ohne größere Abweichung umsetzen können. Die Sollwerte für $p_{n,\text{ref}}$ und $p_{d,\text{ref}}$ sind null, die Ausgänge δ , α und γ werden hier nicht genutzt. Der MPC

sind die zukünftigen Sollwerte als Referenztrajektorie $\vec{r}_{k+i|k}$ für einen Prädiktionshorizont von sechs Zeitschritten $H_y = \{1, \dots, 6\}$ bekannt:

$$\vec{r}_{k+i|k} = (p_{n,\text{ref},k+i}, p_{e,\text{ref},k+i}, p_{d,\text{ref},k+i}, \psi_{\text{ref},k+i})^T - \vec{Y}_{A|k} \quad \forall i \in H_y = \{1, \dots, 6\} \quad (5.166)$$

Bei einer Abtastzeit von $T_{\text{MPC}} = 0,05\text{s}$ prädiziert die MPC also einen Zeitraum von $0,3\text{s}$.

Die Elemente auf der Hauptdignalen von $\mathbf{Q}(i)$ werden für p_n, p_e, p_d für alle Prädiktions-schritte auf 100 gesetzt. Für ψ sind nur die Elemente für die ersten beiden Prädiktions-schritte 100, die restlichen sind 0. Für die Elemente auf der Hauptdignalen von $\mathbf{R}(j)$ wird 10^{-8} in einem Stellhorizont $H_u = \{0, 1\}$ gewählt. Die Eingangs- und Zustandsbeschränkungen (5.160) und (5.161) sind in diesem Szenario nicht aktiv.

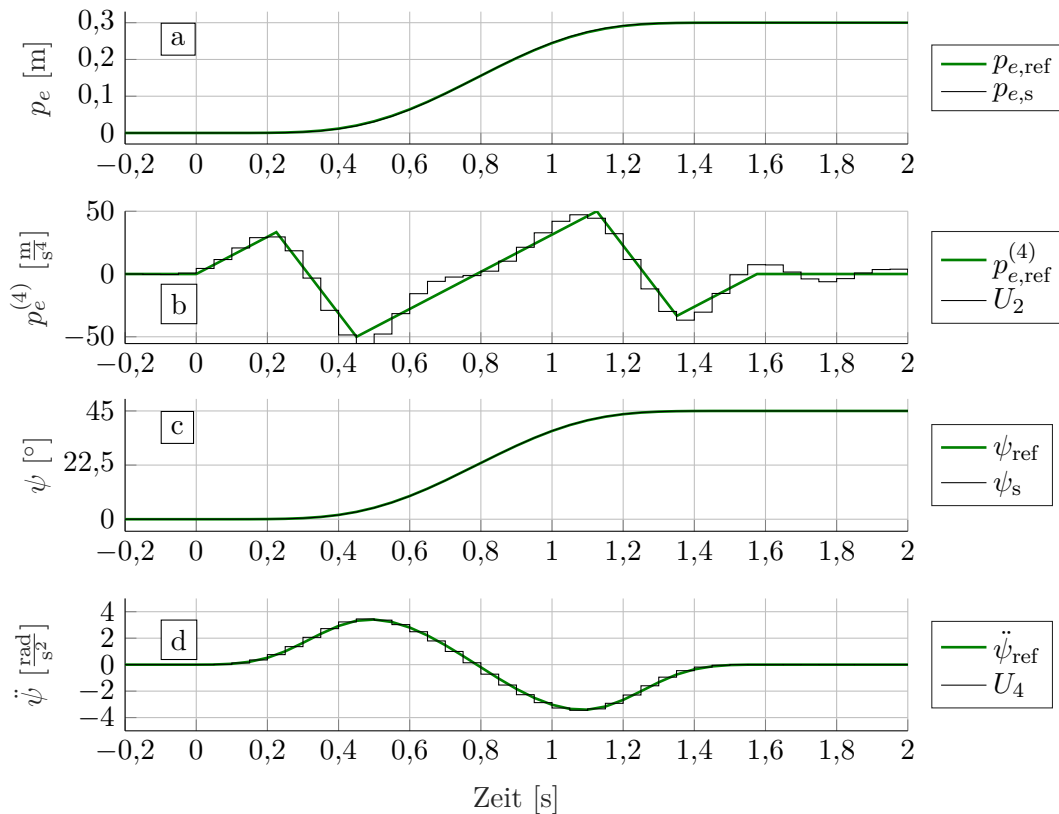


Abbildung 5.12: MPC-Trajektoriegenerierung: Positionswechsel

Die Ergebnisse sind in Abbildung 5.12 dargestellt: Die MPC folgt den Referenztrajektorien mit nur sehr kleinen Abweichungen; der maximale Fehler für die Ost-Trajektorie $p_{e,s}$ in Graph a liegt unter 2 mm. Die Giertrajektorie ψ_s in Graph c weicht um weniger als $0,05^\circ$ von der Referenz ψ_{ref} ab. Die Stellgröße U_2 in Graph b entspricht im Wesentlichen der 4. Ableitung der Ost-Referenztrajektorie $p_{e,\text{ref}}^{(4)}$, obwohl diese der MPC nicht zur Verfügung

steht. Das Gleiche gilt für die Stellgröße U_4 in Graph d, die mit der 2. Ableitung der Gier-Referenztrajektorie $\ddot{\psi}_{\text{ref}}$ übereinstimmt.

Die Ergebnisse der Erprobung dieses Szenarios im Indoor-Flugversuch sind in Abschnitt 6.2.3 zu finden.

Szenario 2: Point-of-Interest

Im zweiten Szenario (vgl. Seite 188) wird ein statischer Punkt, ein Point-of-Interest (POI), mit dem Kamera-Quadrokopter aus verschiedenen Distanzen und Richtungen gefilmt. Dazu gibt der Operator den Abstand δ (5.120), den Azimut α (5.121) und den Winkel γ (5.131) vor und die MPC plant die Trajektorien, um diese Sollgrößen anzufliegen.

Zu Beginn schwebt der Quadrokopter in einem Abstand δ von 2 m und in einem Azimut α von 180° zum Zielobjekt, siehe Abbildung 5.13. Bei $t = 1 \text{ sec}$ wird der Sollwert für α auf 135° und bei $t = 7 \text{ sec}$ auf 90° geändert (Abbildung 5.14, Graph a). Der Sollwert für δ wechselt bei $t = 3 \text{ sec}$ auf 3 m (Graph b). Die Sollwerte für γ und p_d sind 0° bzw. 0 m.

Der Stellhorizont H_u und die Gewichtungsmatrix $\mathbf{R}(j)$ sind gegenüber dem Szenario 1 unverändert. Der Prädiktionshorizont wird auf $H_y = \{1, \dots, 6, 10, 15\}$ erweitert, wodurch eine Prädiktion und eine Berücksichtigung der Zustandsbeschränkungen bis zu 0,75 s in die Zukunft möglich sind. Dies stellt eine frühzeitige Verzögerung vor einem Erreichen der Beschränkungen sicher und reduziert ein Überschwingen. In diesem Szenario sind die zukünftigen Sollgrößenänderungen unbekannt, sie werden als konstant prädiziert. Die

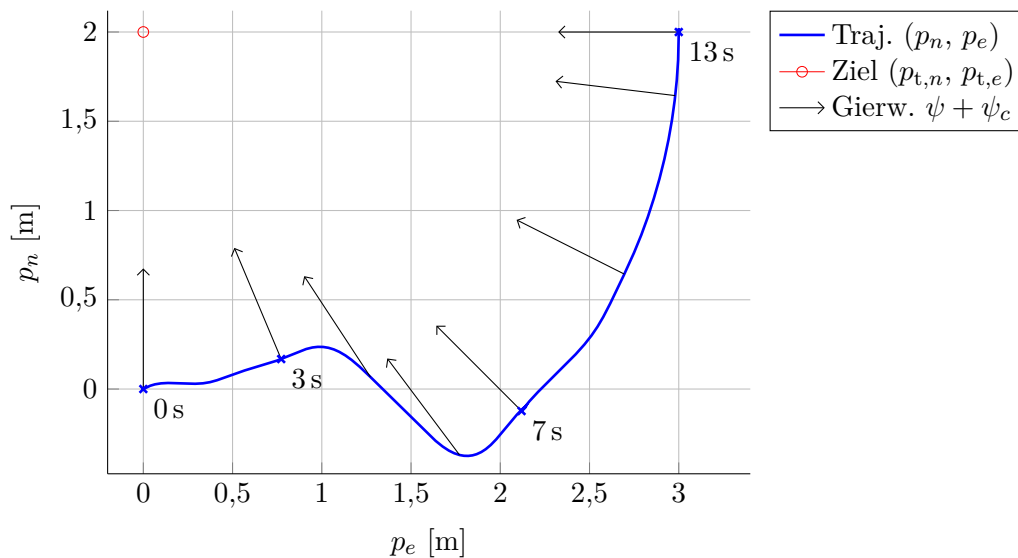


Abbildung 5.13: MPC-Trajektoriengenerierung POI-Szenario: Die Trajektorie des Quadrokkopters ist in Blau, das statische Zielobjekt in Rot und die Blickrichtung der Kamera als schwarze Pfeile dargestellt.

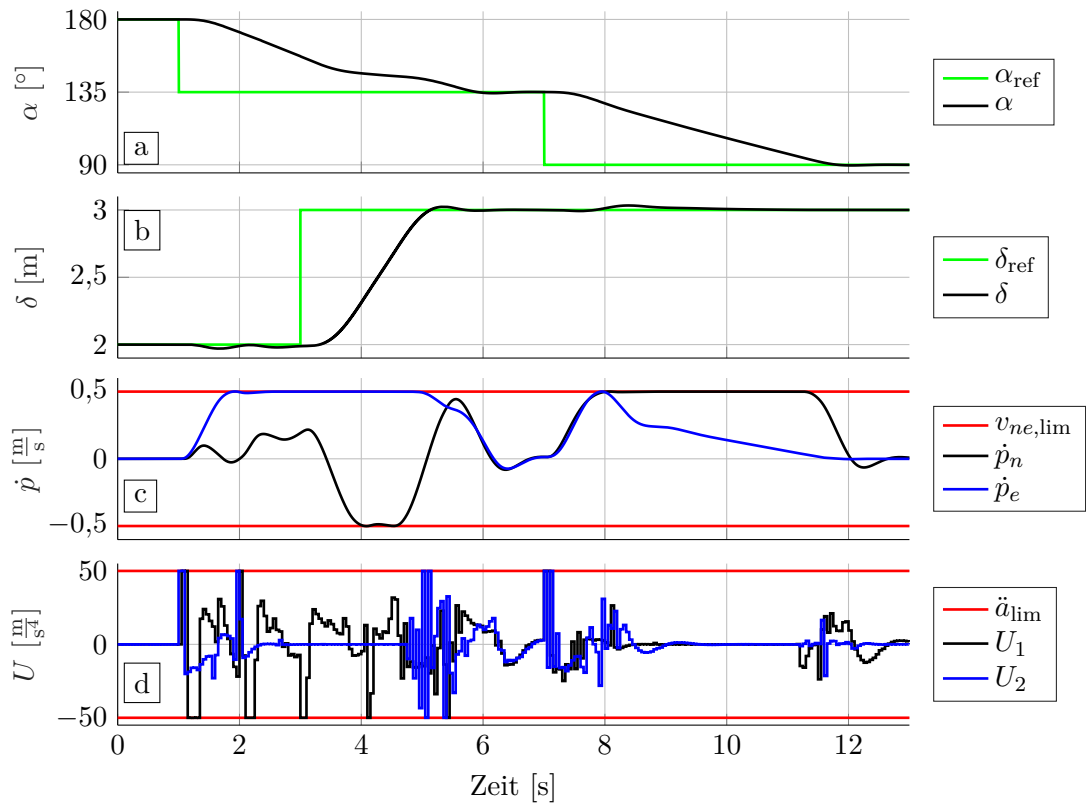


Abbildung 5.14: MPC-Trajektoriegenerierung POI-Szenario: Sollgrößen δ und α , Geschwindigkeiten \dot{p}_n und \dot{p}_e sowie Stellgrößen U_1 und U_2

Elemente auf der Hauptdiagonalen von $\mathbf{Q}(i)$ sind für alle Prädiktionsschritte für p_d und δ auf 100, für α auf 20 sowie für γ auf 10 gesetzt.

Für die Eingangs- und Zustandsbeschränkungen (5.160), (5.161) gilt hier

$$\vec{U}_{\max} = -\vec{U}_{\min} = (\ddot{a}_{\text{lim}}, \dot{a}_{\text{lim}}, \ddot{\psi}_{\text{lim}})^T \quad (5.167)$$

$$\vec{X}_{\max} = -\vec{X}_{\min} = (\infty, v_{ne,\text{lim}}, a_{\text{lim}}, \dot{a}_{\text{lim}}, \infty, v_{ne,\text{lim}}, a_{\text{lim}}, \dot{a}_{\text{lim}}, \infty, v_{d,\text{lim}}, a_{\text{lim}}, \dot{a}_{\text{lim}}, \infty, \dot{\psi}_{\text{lim}})^T \quad (5.168)$$

mit den Zahlenwerten aus Tabelle C.6. Die mit ∞ markierten Zustände werden nicht beschränkt.

Das Ergebnis ist in den Abbildungen 5.13 und 5.14 zu sehen: Das System stellt die Sollgrößenänderungen wie gewünscht ein und beachtet dabei Eingangsbeschränkungen (Abbildung 5.14, Graph d) und Zustandsbeschränkungen (u. a. Geschwindigkeit, siehe Graph c). Die neuen Sollgrößen werden mit nur kleinem Überschwingen erreicht. Die Graphen zeigen, dass die Gewichtungen für den δ -Fehler größer als die für den α -Fehler

sind: Beim Sollgrößensprung bei $t = 3$ sec reduziert der Regler die Änderungsrate von α , um bevorzugt den Fehler in δ verringern zu können, da durch die Beschränkungen nicht beide Fehler gleichzeitig reduziert werden können. Die Einregelung der Sollgröße für γ ist in Abbildung 5.13 in Form von Pfeilen für den resultierenden Kamera-Gierwinkel $\psi + \psi_c$ dargestellt: Die Kamera ist wie gewünscht immer auf das Zielobjekt, den statischen Point-of-Interest, ausgerichtet.

Szenario 1 und 2 zeigen, dass die MPC in der Lage ist, auf Basis unterschiedlicher Formulierungen der Regelgrößen geeignete Führungsgrößen (5.115) für die flachheitsbasierte Regelung zu berechnen. Im Szenario 1 wird ausgenutzt, dass die MPC zukünftige Sollwertverläufe explizit berücksichtigen kann. Dadurch kann die MPC den Referenztrajektorien ohne Zeitverzug folgen. Für die Objektverfolgung wäre es denkbar, die Bewegung des Zielobjekts anhand eines Modells zu prädictieren. Dies ist jedoch nur dann sinnvoll, wenn sich die Zielbewegung tatsächlich deterministisch vorhersagen lässt, was oftmals nur eingeschränkt oder gar nicht möglich ist. Im Szenario 2 ist die Prädiktion der Zielbewegung trivial, da sie bei einem statischen POI konstant ist. Die zukünftigen Kommandoänderungen können hier nicht prädictiert werden, auch wenn das vorteilhaft wäre.

Ein Nachteil der MPC ist, dass in jedem Zeitschritt ein rechenintensives quadratisches Optimierungsproblem (5.162) mit Nebenbedingungen (5.163) auf dem Quadrokopter gelöst werden muss. Die Trajektoriengenerierung mit Kaskadenregelung aus Abschnitt 5.3.3 erfordert wesentlich weniger Rechenaufwand. Bei der Trajektoriengenerierung mit Spline-Kurven in Abschnitt 5.3.2 kann die Lösung des Optimierungsproblems mitunter offline vorab berechnet werden. Bei der Wahl einer Trajektoriengenerierungsmethode für einen bestimmten Betriebsmodus müssen die Vor- und Nachteile der verschiedenen Methoden jeweils abgewogen werden.

5.4 Ablaufsteuerung

Das Simulink-Modell, aus dem der Code für Sensorfusion und Regelung generiert wird (siehe Abschnitt 3.2.2), enthält auch eine Ablaufsteuerung, um die Zustandsübergänge zwischen Initialisierung, Starten, Flug, Landen etc. zu steuern. Die Ablaufsteuerung ist in Abbildung 5.15 als Zustandsdiagramm dargestellt.

Bei Systemstart beginnt der Quadrokopter in der **Initialisierungsphase**. Dort wird die Initialisierung der Sensorfusion gemäß Abschnitt 4.6.1 gestartet (`initAktiv = wahr`), mit der in einem Zeitintervall von 5 sec die Startwerte für die Sensorfusion ermittelt werden. Wenn die Initialisierung erfolgreich ist und sich die Motoren im Leerlauf befinden (\neg MüL: nicht Motoren über Leerlauf), wechselt die Steuerung in den Zustand „Startklar“. Wenn hingegen während der Initialisierung ein Fehler auftritt (`initFehlgeschl`), weil z. B. sich der Quadrokopter bewegt hat, wechselt die Steuerung in den Zustand Initialisierungsfehler. Gleiches gilt aus Sicherheitsgründen auch, wenn die Initialisierung zwar erfolgreich war, aber gleichzeitig die Motoren bereits über ihrer Leerlaufdrehzahl rotieren. Die Initialisierung kann durch das Reinitialisierungskommando (`reinitKmd`) über die QC Cam Ctrl

App, die auf dem ControlPad läuft, neu gestartet werden, siehe Abbildung 3.15. Der Beginn der Initialisierung, die erfolgreiche Initialisierung und der Initialisierungsfehler sind durch unterschiedliche Piepsignale des Quadropters hörbar.

In der QC Cam Ctrl App wird jeweils der aktuelle Zustand der Ablaufsteuerung dargestellt, siehe Abbildung 3.15. Zudem werden die Zustandsübergänge per Sprache angesagt (vgl. Abschnitt 3.3.3).

Befindet sich der Quadropters im **Startklar**-Zustand, kann per Startkommando in der QC Cam Ctrl App in die Startphase gewechselt werden. Damit dieser Zustandsübergang möglich ist, müssen zugleich drei weitere Bedingungen erfüllt sein: Zum einen muss der Sicherheitspilot über einen Hebel an seiner Fernsteuerung die Steuerung des Quadropters durch das Mastermind-Board aktiviert (MmA) haben. Außerdem dürfen sich die Motoren nicht über Leerlauf drehen. Als letzte Bedingung darf kein kritischer Alarm, z. B. durch einen Sensorausfall, aktiv sein. Ein Alarm wird optisch in der App angezeigt und während des geregelten Fluges auch akustisch am ControlPad und am Quadropters ausgegeben.

Beim Übergang in die **Startphase** aktiviert die Ablaufsteuerung die Motordrehzahl-Steuerung (entry / motStrgAktiv = wahr) über den High Level Processor des Pelican (vgl. Seite 66). Dann wird die Drehzahl der Motoren gleichmäßig erhöht (entry / motDzErhAktiv = wahr). Wenn knapp 90% der zum Schwebeflug benötigten Drehzahl erreicht ist (motDzErreicht), wird in den Steigflug gewechselt. Dazu wird die flachheitsbasierte Flugregelung aktiviert (entry / flugRegAktiv = wahr) und die Starttrajektorie kommandiert (entry / startTrajAktiv = wahr). Die Starttrajektorie ist eine vertikale Spline-Kurve gemäß Abschnitt 5.3.2, die online skaliert wird (vgl. auch Flugversuch A2 „Spline-Trajektorie vertikal“ in Abschnitt 6.2.2). Ist die konfigurierte Starthöhe erreicht (StarthE), wechselt die Steuerung in die Flugphase.

In der **Flugphase** können in der QC Cam Ctrl App die Betriebsmodi pro Kanal zwischen Objektverfolgung, Joystick etc. umgeschaltet werden, siehe Abbildung 3.17. Zudem können die Schubkalibrierung gemäß Abschnitt 5.1.2.2 gestartet und gestoppt sowie die Kalibrierungsparameter angewendet werden. Die Unterzustände für die Betriebsmodi und die Schubkalibrierung sind in Abbildung 5.15 nicht dargestellt.

Um das Starten des Quadropters auch aus der Hand zu erlauben, kann die Flugphase auch ohne ein Starten vom Boden erreicht werden. Wenn sich die Ablaufsteuerung im Zustand Startklar befindet, die Motoren über der Leerlaufdrehzahl rotieren (MüL) und der Sicherheitspilot den Hebel umlegt (MmASteig: steigende Flanke auf dem Aktivierungssignal der Mastermind-Steuerung), wechselt die Steuerung in die Flugphase und aktiviert die Flugregelung etc. Entsprechendes gilt aus dem Zustand Manuell heraus, siehe unten.

Aus der Flugphase heraus kann per Landekommando (landeKmd) die **Landephase** eingeleitet werden. Die Steuerung aktiviert dann die vertikale Landetrajektorie (entry / landeTrajAktiv = wahr). Kurz vor Erreichen des Bodens, der per Ultraschallsensor erkannt wird, wird die Sinkgeschwindigkeit reduziert, um ein sanftes Aufsetzen zu ermöglichen.

Erkennt der Ultraschallsensor, dass der Boden erreicht ist (gelandet), wird die Flugregelung deaktiviert ($\text{entry} / \text{flugRegAktiv} = \text{falsch}$) und die gleichmäßige Reduzierung der Motordrehzahlen gestartet ($\text{entry} / \text{motDzRedAktiv} = \text{wahr}$). Wenn die Motoren die Leerlaufdrehzahl erreichen [motLeerlErreicht], wechselt die Steuerung wieder in den Startklar-Zustand. Die Landephase kann während des Sinkfluges abgebrochen werden, indem ein Startkommando gesendet wird. In dem Fall wechselt die Steuerung wieder in den Steigflug der Startphase. Entsprechend können der Steigflug und die Motordrehzahl-Erhöhung in der Startphase abgebrochen werden, indem ein Landekommando gesendet wird.

Der Sicherheitspilot kann zu jedem Zeitpunkt die Steuerung per Fernsteuerung übernehmen, indem er den entsprechenden Hebel an seiner Fernsteuerung umlegt. Dann schaltet der Low Level Processor des Pelican auf den manuellen Flug per Fernsteuerung um und deaktiviert die (über den High Level Processor des Pelican geroutete) Steuerung des Mastermind-Boards ($\neg \text{MmA}$), vgl. Seite 66f. Um dies zu spiegeln, wechselt die Ablaufsteuerung dann – wenn sie zuvor in der Start-, Flug- oder Landephase war – in den Zustand **Manuell** und deaktiviert aus Konsistenzgründen die Motorsteuerung ($\text{entry} / \text{motStrgAktiv} = \text{falsch}$) und die Flugregelung ($\text{entry} / \text{flugRegAktiv} = \text{falsch}$). Um den Zustand Manuell wieder zu verlassen, schaltet der Sicherheitspilot den Hebel an der Fernsteuerung wieder zurück und aktiviert so die Mastermind-Steuerung (MmA). Wenn die Motoren nicht über der Leerlaufdrehzahl rotieren ($\neg \text{MüL}$), wechselt die Ablaufsteuerung in den Zustand Startklar. Wenn die Motoren über der Leerlaufdrehzahl rotieren, wird direkt die Flugphase aktiv, z. B. für ein Starten aus der Hand, siehe oben.

5.5 Simulation

Vor der Erprobung im Flugversuch (siehe Abschnitt 6.2) werden die Eigenschaften der flachheitsbasierten Regelung aus Abschnitt 5.2 in der Simulation gezeigt. Dazu wird die Simulationsumgebung in Simulink genutzt, siehe Abschnitt 3.5.1. Die Dynamik des Quadropters wird durch das nichtlineare Modell (5.40) simuliert, dessen Stellgrößeneingänge $u_1 \dots u_4$ durch die flachheitsbasierte Zustandsrückführung (5.101), (5.103), (5.104) mit einer Rate von 100 Hz berechnet werden, vgl. Abbildung 5.5 (a). Der Eingang $\vec{\xi}$ (5.96) der flachheitsbasierten Zustandsrückführung wiederum wird durch die Trajektorienfolge-regelung aus Abschnitt 5.2.3 bestimmt, vgl. Abbildung 5.7.

Als Führungsgrößen $\vec{y}_{1,s} \dots \vec{y}_{4,s}$ (5.115) für die Trajektorienfolgeregelung werden hier die Spline-Trajektorien und deren Ableitungen aus Abschnitt 5.3.2 verwendet. Dabei ist die Ost-Solltrajektorie $p_{e,s}$ wie im MPC Szenario 1 (siehe Seite 191) auf eine Distanz von 30 cm skaliert, vgl. Abbildung 5.12. Die Simulation wird einmal als reiner Positionswechsel und ein zweites Mal mit einer Gier-Solltrajektorie ψ_s für eine gleichzeitige Drehung um 45° ausgeführt. Die Sollwerte $p_{n,s}$ und $p_{d,s}$ sind null.

Die Ergebnisse der Simulationen sind in Abbildung 5.16 zu sehen. Der reine Positionswechsel mit konstantem Sollgierwinkel $\psi_s = 0$ ist in Schwarz und der Positionswechsel mit

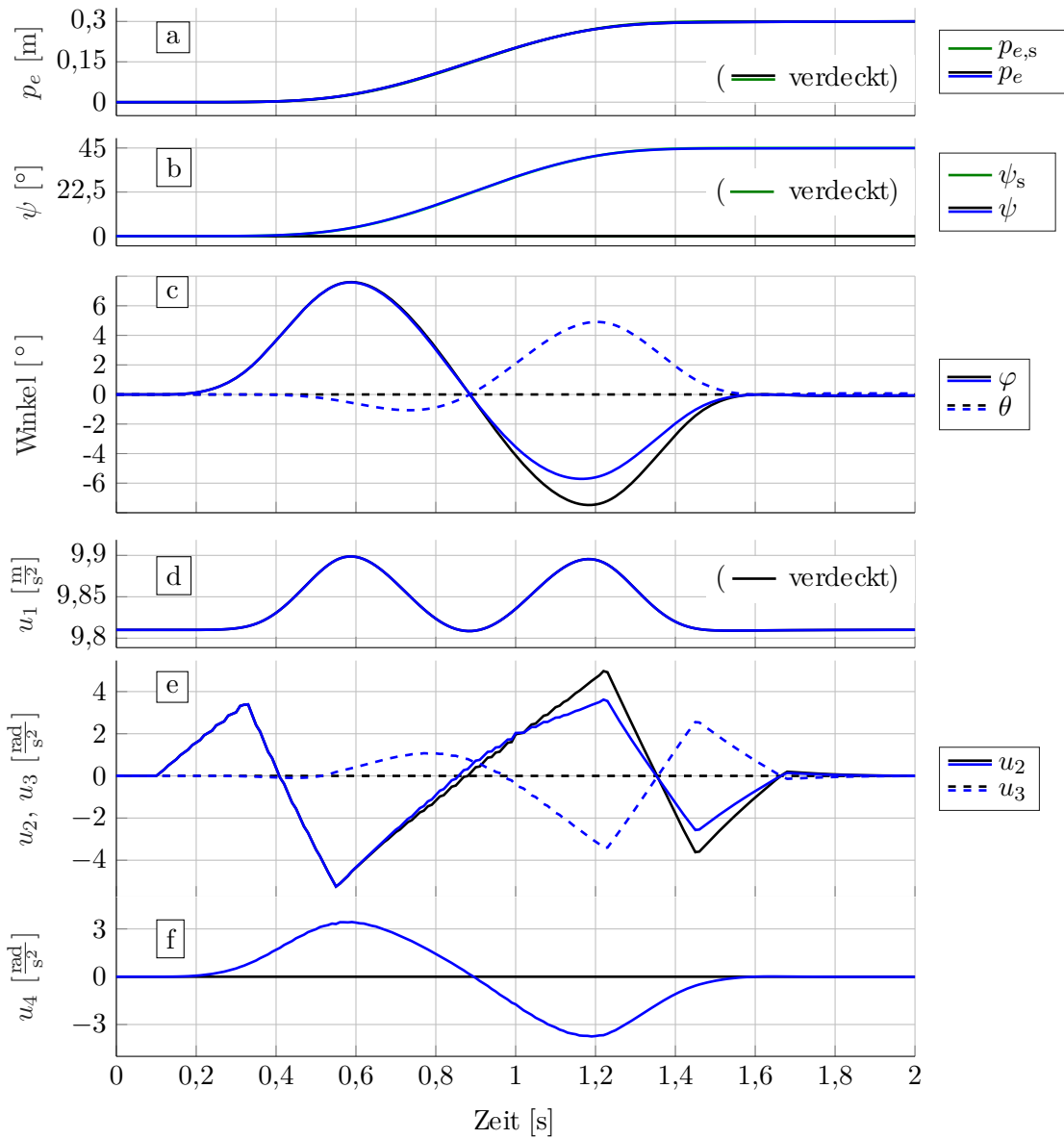


Abbildung 5.16: Simulation der nichtlinearen Systemdynamik mit flachheitsbasierter Zustandsrückführung und Trajektorienfolgeregelung; schwarze Linien: reiner Positionswechsel, blaue Linien: Positionswechsel mit gleichzeitigem Gieren

Gier-Trajektorie ψ_s gemäß Graph b ist in Blau dargestellt. Graph e zeigt die kommandierten Drehbeschleunigungen u_2 und u_3 . In Graph c ist die resultierende Lage des nichtlinearen Systems als Roll- und Nickwinkel gegeben. Die Position p_e in östlicher Richtung und ihre Solltrajektorie $p_{e,\text{ref}}$ stimmen sehr gut überein, siehe Graph a; der maximale Positionsfehler beträgt 2,6 mm in beiden Simulationen. Die kommandierte Drehbeschleunigung u_4 gemäß Graph f führt zu einer ebenfalls nur geringen Abweichung des Gierwinkels ψ von der Solltrajektorie ψ_s , siehe Graph b; der maximale Gierwinkelfehler ist $0,22^\circ$ in der zweiten Simulation. Durch den spezifischen Schub u_1 in Graph d wird die Höhe gehalten; die Fehler für p_d and p_n sind vernachlässigbar.

Insgesamt folgt der simulierte Quadrokopter den Trajektorien trotz der nichtlinearen Dynamik ohne nennenswerte Fehler. Im Flugversuch A1 in Abschnitt 6.2.2 wird das reale Gesamtsystem mit einer ähnlichen Trajektorie erprobt.

In Abbildung 5.17 ist die Bewegung des Quadrokopters für einen vergleichbaren Positionswechsel über 2 m visualisiert. Gut zu erkennen sind die Rollbewegungen zum horizontalen Beschleunigen und Verzögern am Beginn und Ende der Trajektorie. Zudem kommt es zu keinem Überschwingen am Endpunkt.

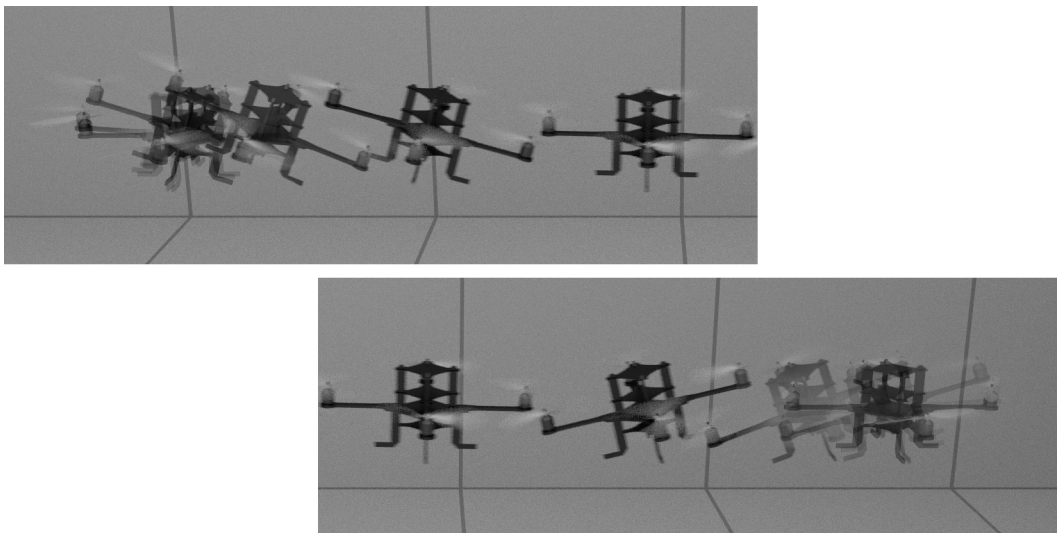


Abbildung 5.17: Visualisierung eines Positionswechsels über 2 m

6 Erprobung

Anhand diverser Flugversuche wird in diesem Kapitel die Erprobung des Gesamtsystems dargestellt. Zunächst werden in Abschnitt 6.1 die Validierung der Sensorfusion und die Tests des experimentellen GNSS-Empfängers im Galileo-Testgebiet gezeigt. In Abschnitt 6.2 wird dann die Erprobung der Automatisierung des Quadropters mit flachheitsbasierter Regelung und Trajektoriengenerierung in den verschiedenen Betriebsmodi vorgestellt.

6.1 Sensorik-Erprobung

6.1.1 Sensorfusion

Zur experimentellen Validierung der Sensorfusion aus Kapitel 4 wird ein Flugversuch dargestellt, bei dem das Infrarot-Trackingsystem (vgl. Abschnitt 3.5.2) genutzt wird, sodass in dessen Erfassungsbereich Referenzmessungen von Position und Orientierung des Quadropters zur Verfügung stehen. Damit die Erfassung der Infrarotlicht-reflektierenden Marker draußen funktioniert, wird der Flug bei vollständiger Bewölkung durchgeführt.

Auf dem Quadropter läuft die Sensorfusion, die den geschätzten Systemzustand \hat{x}_k (4.48) – wie in Abschnitt 4.3 beschrieben – auf Basis der Messdaten der IMU (Abschnitt 4.1.1) mit 100 Hz in der Zeit propagiert. Die Positions- und Geschwindigkeitsmessungen des GNSS-RTK-Empfängers (Abschnitt 4.1.2.1) werden im Estimationsschritt (Abschnitt 4.5) unter Berücksichtigung ihres Gültigkeitszeitpunktes (vgl. Ablauf in Abschnitt 4.6.2) zur Korrektur des geschätzten Systemzustandes verwendet. Die RTK-Korrekturdaten erhält der Empfänger über das Kommunikationsnetzwerk (vgl. Abbildung 3.28). Die Magnetometermessungen gemäß Abschnitt 4.5.1.3 finden lediglich in der Initialisierung Verwendung.

Um den von der Sensorfusion geschätzten Systemzustand mit den Messungen des Trackingsystems vergleichen zu können, wird das erdfeste Koordinatensystem des Trackingsystems so kalibriert, dass zwei Achsen horizontal nach Norden und Osten zeigen und die dritte Achse senkrecht nach unten weist. Der Koordinatensystem-Ursprung wird mithilfe des GNSS-RTK-Empfängers als $(\varphi_0, \lambda_0, h_0)$ eingemessen. Die Orientierung des Tracking-Ziels mit den fünf Infrarot-Markern am Quadropter (siehe Abbildung 6.1) wird so eingestellt, dass Lage- und Gierwinkel null sind, wenn die Quadropterarme horizontal



Abbildung 6.1: Flugversuch zur Validierung der Sensorfusion: Im Vordergrund sind die beiden Infrarotkameras des Tracking-Systems zu sehen. Die fünf hellgrauen Kugeln am Quadrokopter sind die Infrarot-Marker

ausgerichtet sind und der Arm mit Rotor 1 (siehe Abbildung 5.1) nach Norden zeigt. Außerdem wird die Position des Tracking-Ziels relativ zur IMU eingemessen, deren Position die Sensorfusion schätzt.

Zu Beginn des Versuchs wird die Sensorfusion gemäß Abschnitt 4.6.1 im Stillstand initialisiert, um die Startwerte des geschätzten Systemzustandes \hat{x}_0 (4.144) und der Schätzfehlerkovarianzmatrix \mathbf{P}_0 (4.145) zu setzen. Der Flug des Quadrokopters wird hier manuell per Fernsteuerung gesteuert. Nach dem Abheben werden zunächst einige Schleifen geflogen, um eine schnelle Konvergenz der Schätzfehlerkovarianzmatrix zu erlauben. Dann wird der Quadrokopter immer wieder durch den Erfassungsbereich des Trackingsystems gesteuert, bevor er gelandet wird. Die Trajektorie des Quadrokopters und der Erfassungsbereich, in dem die Referenzmessungen des Trackingsystems verfügbar sind, sind in Abbildung 6.2 zu sehen.

Die von der Sensorfusion geschätzte Position $(\hat{\varphi}_k^+, \hat{\lambda}_k^+, \hat{h}_k^+)^T$ (4.120) wird zum Vergleich mit dem Referenzsystem in kartesische Koordinaten umgerechnet (vgl. (4.80), (5.10)):

$$\begin{pmatrix} \hat{p}_n^+ \\ \hat{p}_e^+ \\ \hat{p}_d^+ \end{pmatrix} \approx \begin{pmatrix} (\hat{\varphi}^+ - \varphi_0)(R_n + \hat{h}^+) \\ (\hat{\lambda}^+ - \lambda_0)(R_e + \hat{h}^+) \cos \varphi_0 \\ -(\hat{h}^+ - h_0) \end{pmatrix} \quad (6.1)$$

Bezugspunkt $(\varphi_0, \lambda_0, h_0)$ ist dabei der eingemessene Ursprung des Referenzsystems, siehe oben. Die geschätzte Positionstrajektorie $(\hat{p}_n^+, \hat{p}_e^+)$ ist in Blau auf der Karte in Abbildung 6.2 dargestellt. In Abbildung 6.3 ist der Zeitverlauf der Positionsschätzung aus-

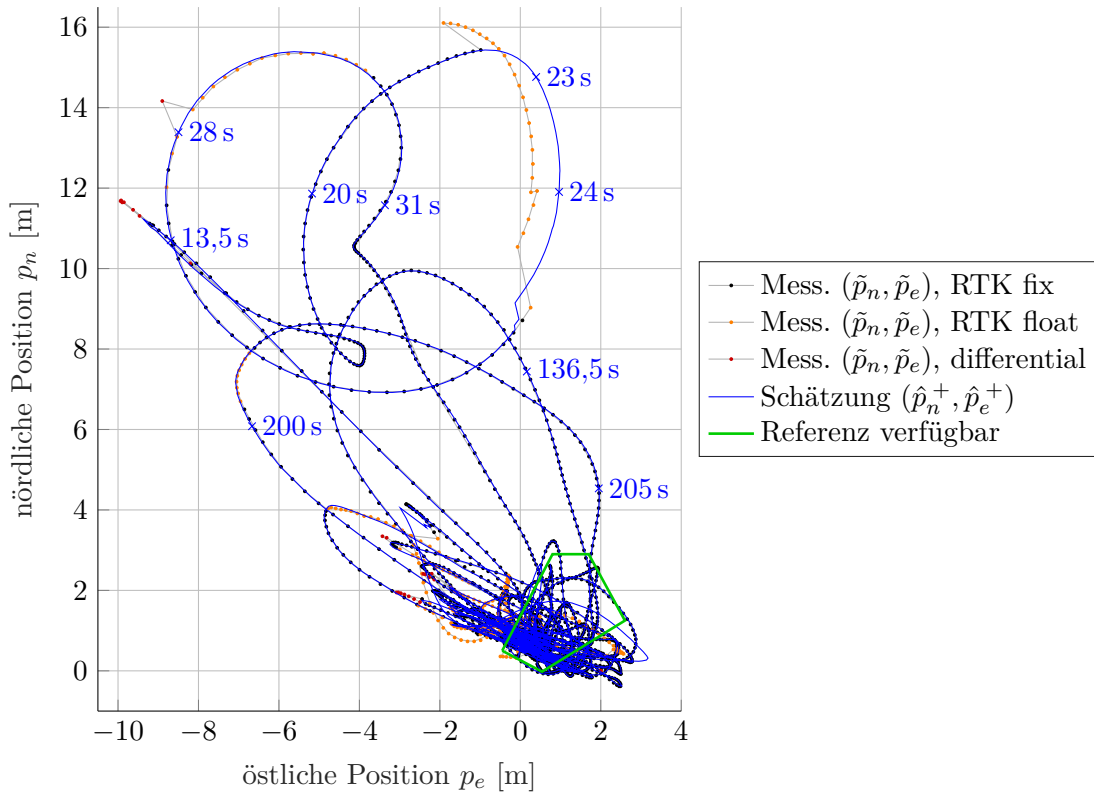


Abbildung 6.2: Karte Flugversuch zur Sensorfusionsvalidierung

schnittsweise für ein Zeitintervall, in dem sich der Quadrocopter überwiegend im Erfassungsbereich des Trackingsystems befindet, ebenfalls in Blau zu sehen.

In den beiden Abbildungen ist auch die vom RTK-GNSS-Empfänger gemessene Antennenposition $(\tilde{\varphi}_A, \tilde{\lambda}_A, \tilde{h}_A)$ dargestellt, umgerechnet in kartesische Koordinaten und in die Position der IMU, vgl. (4.94):

$$\begin{pmatrix} \tilde{p}_n \\ \tilde{p}_e \\ \tilde{p}_d \end{pmatrix} \approx \begin{pmatrix} (\tilde{\varphi}_A - \varphi_0)(R_n + \tilde{h}_A) \\ (\tilde{\lambda}_A - \lambda_0)(R_e + \tilde{h}_A) \cos \varphi_0 \\ -(\tilde{h}_A - h_0) \end{pmatrix} - \mathbf{C}_b^{\hat{n}} \vec{d}_A^b \quad (6.2)$$

Der Modus des GNSS-Empfängers ist an den Farben der Punkte zu erkennen. Während eines Großteils des Fluges wird im RTK-Fix-Modus (schwarze Punkte) die höchste Genauigkeit erreicht. Manchmal wechselt der Empfänger kurzzeitig in den RTK-Float-Modus (orangefarbene Punkte) oder sogar in den Differential-Modus (rote Punkte auf der Karte). Die Satellitenkonstellation für den Zeitpunkt und den Ort des Flugversuchs ist in Abbildung 2.6 dargestellt. Der GNSS-Empfänger trackt während des Fluges durchschnittlich 21 GPS-, GLONASS- und Galileo-Satelliten¹ auf den Frequenzbändern² L1,

¹Der Empfang von BeiDou-Satelliten ist in diesem Versuch deaktiviert.

²vgl. Abbildung 2.5

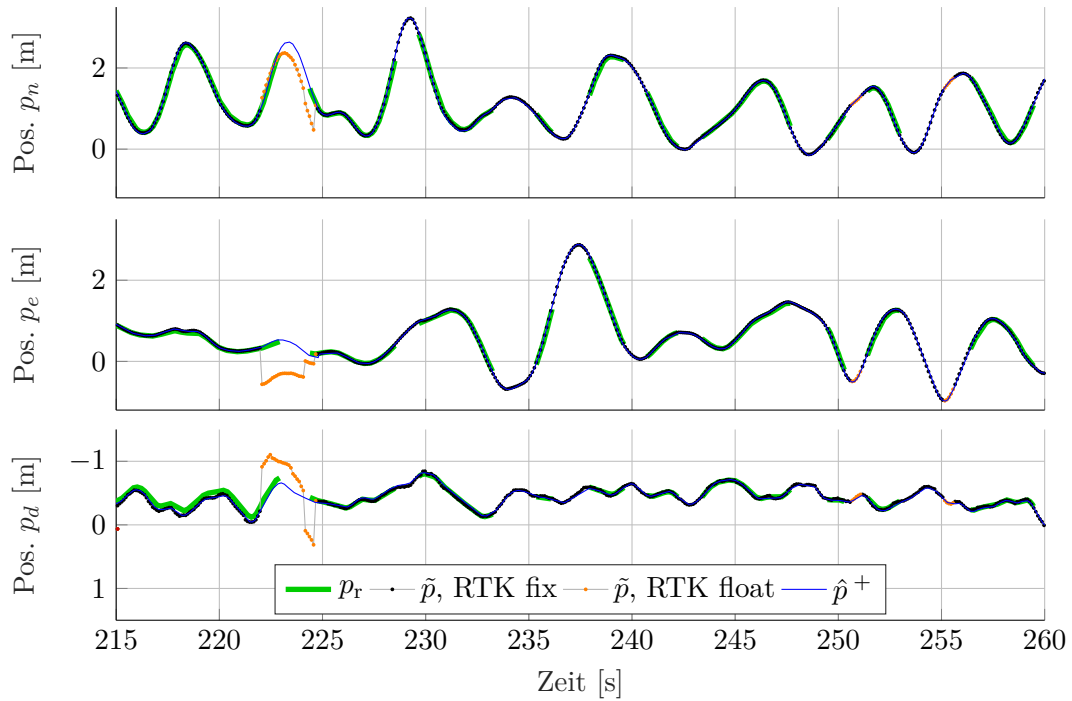


Abbildung 6.3: Positionsschätzung

L2, L2C, L1OF, E1. Zur Berechnung der Positionslösung verwendet der Empfänger im Mittel 7 GPS-Satelliten auf zwei Frequenzen.

Die Differenz Δt_A (4.22) zwischen Gültigkeits- und Empfangszeitpunkt der GNSS-Messungen ist in Abbildung 6.3 korrigiert. Die Messungen \tilde{p}_n , \tilde{p}_e , \tilde{p}_d sind also zu dem Zeitpunkt dargestellt, zu dem die Sensorfusion sie durch Zurückrollen um n_r Zeitschritte (4.146) verarbeitet. Das Alter Δt_A ist in Abbildung 6.4 für den Beispielzeitabschnitt dargestellt. Aufgrund der hohen CPU-Last des GNSS-Empfängers liegt das Alter im gesamten Flug zwischen 46 und 143 ms und im Mittel bei 70 ms. Der Zustand der Sensorfusion wird also im Mittel um $n_r = 7$ Zeitschritte zurückgerollt, bevor der Schätzwert des Systemzustandes mit den GNSS-Messungen korrigiert wird.

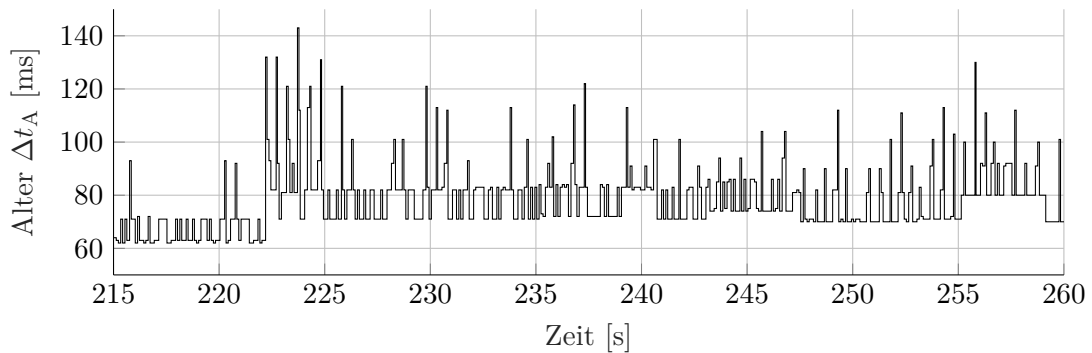


Abbildung 6.4: Alter Δt_A der Nachrichten des GNSS-Empfängers

Bei der oben beschriebenen Kalibrierung des Trackingsystems verbleiben Restfehler. Um die dynamischen Fehler der Sensorfusion besser analysieren zu können, wird von den Referenzmessungen der statische Anteil der Differenz zum Schätzwert der Sensorfusion abgezogen. Der Koordinatensystem-Ursprung des Trackingsystems wird durch eine Verschiebung um

$$\Delta\vec{p}_r = \begin{pmatrix} \Delta p_{n,r} \\ \Delta p_{e,r} \\ \Delta p_{d,r} \end{pmatrix} = \begin{pmatrix} -6 \text{ mm} \\ 13 \text{ mm} \\ 0 \text{ mm} \end{pmatrix} \quad (6.3)$$

so korrigiert, dass die Mediane der Positionsfehler der Sensorfusion (siehe unten in Abbildung 6.9) zu null werden. Die Zahlenwerte von $\Delta\vec{p}_r$ liegen in der Größenordnung, die durch die Einmessung des Koordinatensystem-Ursprungs per RTK-GNSS zu erwarten ist.

Die um $\Delta\vec{p}_r$ korrigierten und in die Position der IMU umgerechneten Referenzdaten $p_{n,r}$, $p_{e,r}$, $p_{d,r}$ sind in Abbildung 6.3 in Grün dargestellt. Sie haben eine Update-Rate von 50 Hz. In den Zeitabschnitten, in denen die Tracking-Marker nicht von den Infrarot-Kameras erkannt wurden, treten Lücken auf. Es ist erkennbar, dass die geschätzte Position $(\hat{p}_n^+, \hat{p}_e^+, \hat{p}_d^+)^T$ überwiegend sehr gut mit der Referenz übereinstimmt. Die Fehlerverteilung des Positionsfehlers über den gesamten Flug wird unten in Abbildung 6.9 dargestellt.

Zum Zeitpunkt $t = 222$ s springt die GNSS-Positionsmessung, da der Empfänger die Phasenmehrdeutigkeit nicht mehr eindeutig auflösen kann und vom RTK-Fix- in den RTK-Float-Modus wechselt (vgl. RTK auf Seite 62). Gleichzeitig erhöhen sich die vom GNSS-Empfänger geschätzten Positionsgenauigkeiten, die der Sensorfusion als $\mathbf{R}_{\hat{p},k}$ (4.20) zur Verfügung gestellt werden, z. B. in östlicher Richtung von $\sigma_{p,e} = 1,7$ cm auf 60 cm, für die Höhe sogar auf $\sigma_{p,d} = 1$ m. Die Sensorfusion ist in der Lage, die Verschlechterung der GNSS-Messungen wie gewünscht ohne großen Positionsdrift zu überbrücken.

Die von der Sensorfusion geschätzten Geschwindigkeiten \hat{v}_{eb}^{n+} (4.122) sind in Abbildung 6.5 für das Beispiel-Zeitintervall in Blau dargestellt. Die Referenzgeschwindigkeiten $v_{n,r}$, $v_{e,r}$, $v_{d,r}$, die aus der Differentiation der Positionsdaten des Trackingsystems berechnet werden, sind geglättet in Grün zu sehen. Es sind keine wesentlichen Abweichungen der geschätzten Geschwindigkeiten von den Referenzgeschwindigkeiten zu erkennen (Fehlerverteilung siehe unten in Abbildung 6.9).

Die GNSS-Geschwindigkeitsmessungen \tilde{v}_{eA}^{n+} sind wie die GNSS-Positionen oben, umgerechnet auf die Position der IMU (vgl. (4.101)) und um Δt_A verschoben, wieder als farbige Punkte in Abhängigkeit vom GNSS-Empfänger-Modus dargestellt.

In Abbildung 6.6 ist die geschätzte Lage $\hat{\mathbf{q}}_b^{n+}$ (4.121) umgerechnet in Eulerwinkel in Blau gegeben. Die Referenz in Grün ist die vom Trackingsystem gemessene Orientierung, die um

$$\Delta\varphi_r = 2,6^\circ \quad \Delta\theta_r = 0,2^\circ \quad \Delta\psi_r = -2,8^\circ \quad (6.4)$$

korrigiert ist. Durch diese Korrektur werden die Mediane der Differenzen zwischen den Eulerwinkeln von Sensorfusion und Referenz zu null, siehe unten in Abbildung 6.9.

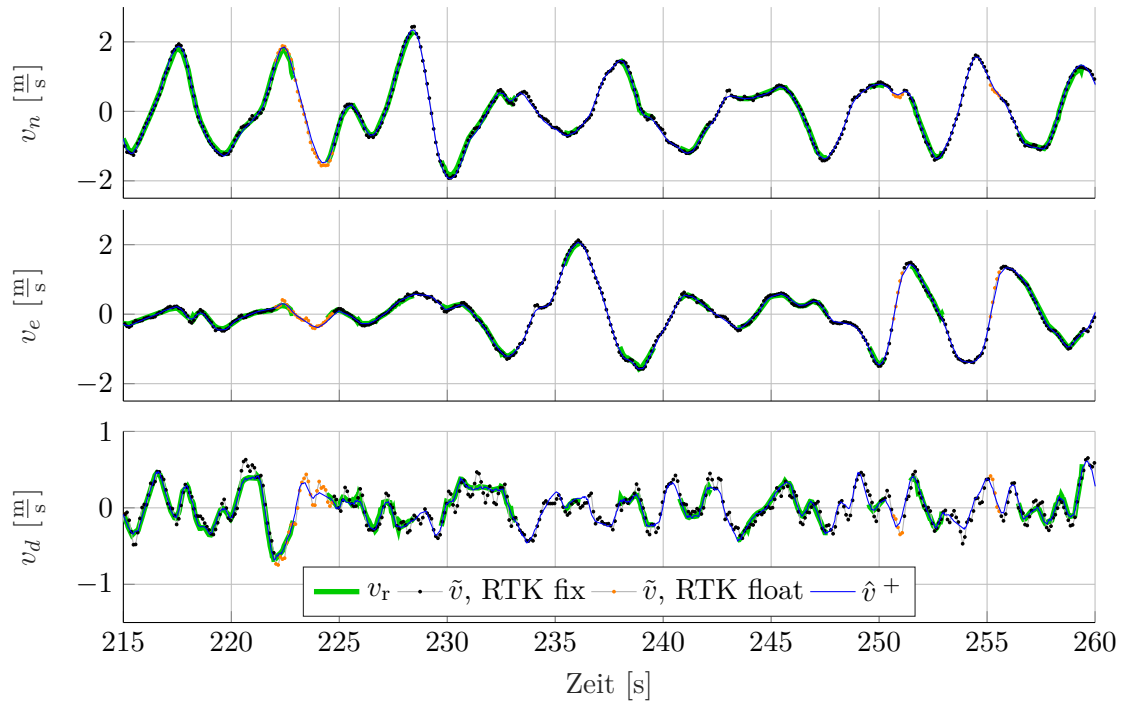


Abbildung 6.5: Geschwindigkeitsschätzung

Mit den entfernten Lage-Offsets ist in Abbildung 6.6 erkennbar, dass die dynamischen Abweichungen der geschätzten Lage von der Referenz minimal sind (Fehlerverteilung siehe unten in Abbildung 6.9). Alle Lageschätzungen bleiben also stabil; dies bestätigt die Annahme, dass weder das Magnetometer-Update gemäß Abschnitt 4.5.1.3 zur Stützung des Gierwinkels noch ein Update mit der geschätzten Schwerkrafttrichtung gemäß Abschnitt 4.5.1.4 zur Stützung von Nick- und Rollwinkel benötigt werden.

Als verbleibende Zustände, für die keine Referenzmessungen zur Verfügung stehen, sind in den Abbildungen 6.7 und 6.8 die geschätzten Beschleunigungs- und Drehratensensor-Bias, \hat{b}_a und \hat{b}_ω (4.122), in Schwarz dargestellt. Zusätzlich sind die geschätzten Standardabweichungen der zugehörigen Fehlerzustände $\hat{\sigma}_{\Delta b_{a,x}}$, $\hat{\sigma}_{\Delta b_{a,y}}$, $\hat{\sigma}_{\Delta b_{a,z}}$ bzw. $\hat{\sigma}_{\Delta b_{\omega,x}}$, $\hat{\sigma}_{\Delta b_{\omega,y}}$, $\hat{\sigma}_{\Delta b_{\omega,z}}$, die das Kalman-Filter quadriert als Diagonalelemente der Schätzfehlerkovarianzmatrix \mathbf{P}_k (2.46) ausgibt, in Grau zu sehen.

Zum Zeitpunkt $t = 0$ s ist die Initialisierung beendet, der geschätzte Zustand wird auf seinen Startwert \hat{x}_0 (4.144) gesetzt und der Kalman-Filter-Algorithmus startet. Das Kalman-Filter verändert sofort – ausgehend von den Startwerten $\hat{b}_{a,0}^b$ (4.130), $\hat{b}_{\omega,0}^b$ (4.128) – das Beschleunigungssensor-Bias in z-Richtung $\hat{b}_{a,z}$ und die Drehratensensor-Bias um die x- und y-Achse, $\hat{b}_{\omega,x}$ und $\hat{b}_{\omega,y}$. Die zugehörigen geschätzten Standardabweichungen der Fehlerzustände starten bei $\sigma_{\Delta b_{a,0}}$, $\sigma_{\Delta b_{\omega,0}}$ (4.143) und konvergieren sofort. Die anderen drei Bias $\hat{b}_{a,x}$, $\hat{b}_{a,y}$ und $\hat{b}_{\omega,z}$ hält das Kalman-Filter während der ersten 8 Sekunden, in denen der Quadrokopter noch steht, fast konstant. Erst als der Quadrokopter abhebt

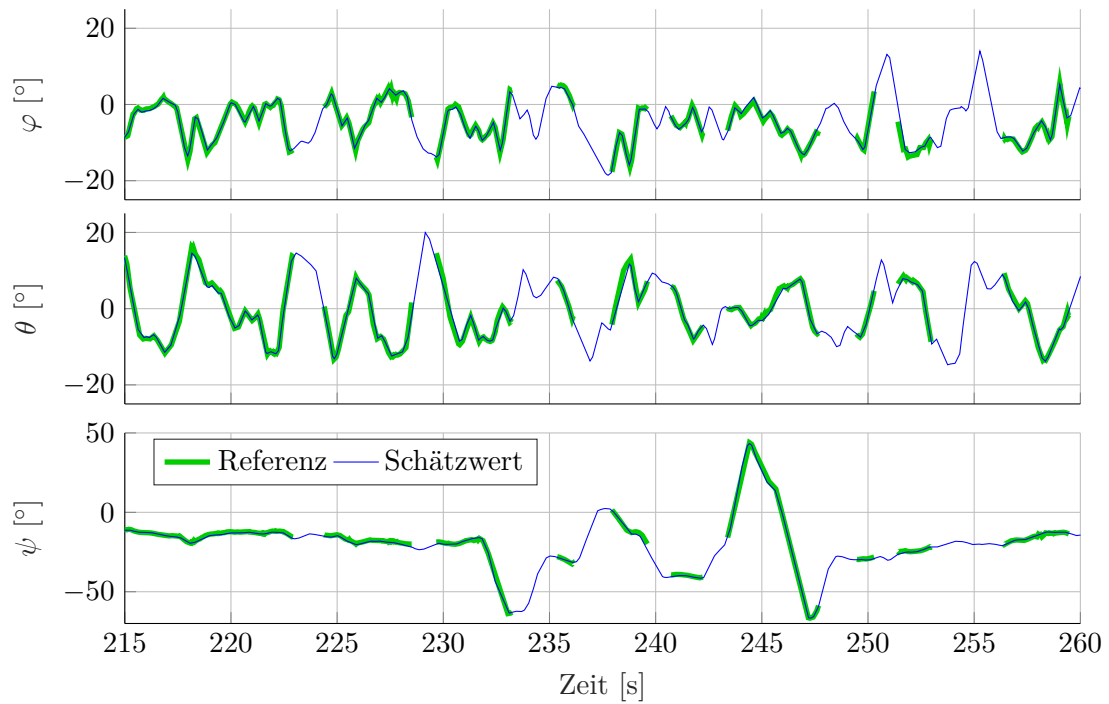


Abbildung 6.6: Lageschätzung

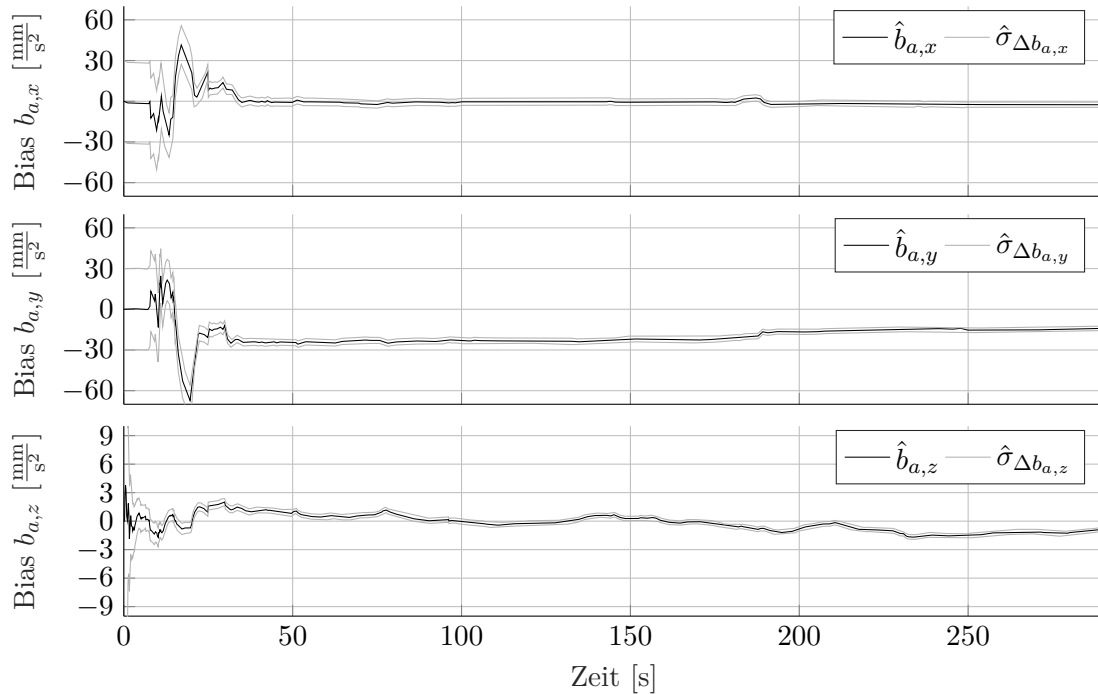


Abbildung 6.7: Beschleunigungssensor-Bias-Schätzung

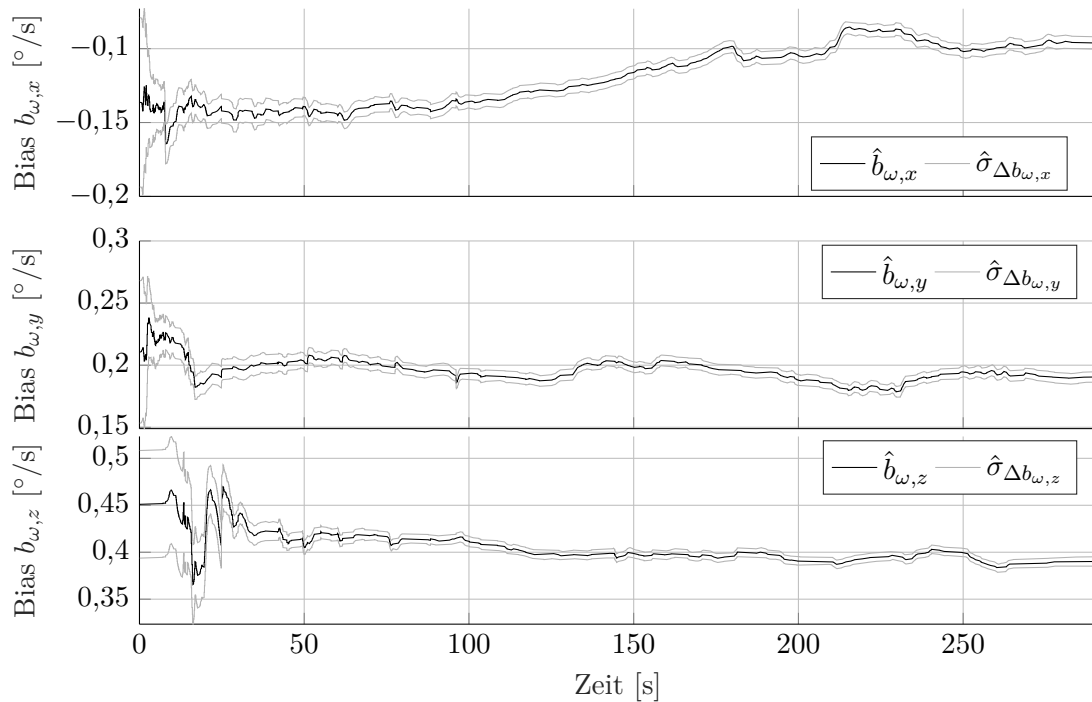


Abbildung 6.8: Drehratensensor-Bias-Schätzung

($t = 8$ s) und ausreichend Bewegung vorhanden ist, ändern sich die Schätzwerte und die geschätzten Standardabweichungen konvergieren.

Nach etwa 30 s Flug ist die Konvergenz der geschätzten Standardabweichungen der Beschleunigungssensor-Bias-Fehler weitestgehend abgeschlossen. Ab dann bleiben auch die geschätzten Beschleunigungssensor-Bias \hat{b}_a recht konstant. Die geschätzten Standardabweichungen der Drehratensensor-Bias-Fehler konvergieren bis ca. $t = 100$ s. Die geschätzten Drehratensensor-Bias \hat{b}_ω verändern sich in den ersten 30 s stärker und dann während des gesamten Fluges nur noch leicht.

In Abbildung 6.9 sind die kumulativen Häufigkeitsverteilungen der Positions-, Geschwindigkeits- und Lagewinkelfehler der Sensorfusionsschätzwerte gegenüber dem Referenzsystem dargestellt. Dabei sind die Referenzpositionen wie oben auch um $\Delta \vec{p}_r$ (6.3) und die Referenzlagewinkel um $\Delta \varphi_r$, $\Delta \theta_r$, $\Delta \psi_r$ (6.4) so korrigiert, dass hier die Mediane (die 50%-Perzentile) der Fehler null betragen. Es lässt sich erkennen, dass die Restfehler nur näherungsweise normalverteilt sind. Die Häufigkeitsverteilungen sind leicht unsymmetrisch.

Fasst man positive und negative Fehler zusammen, sind 68,3 % (entspricht 1σ bei der Normalverteilung) der dynamischen Positionsfehler in nördlicher, östlicher und vertikaler Richtung kleiner als $\pm 3,1$ cm und 95,4 % (2σ) kleiner als ± 7 cm. Bei den Geschwindigkeitsfehlern sind 68,3 % kleiner als $\pm 3,2$ cm/s und 95,4 % kleiner als $\pm 8,8$ cm/s. Es tritt kein statisches Offset zu den Geschwindigkeiten des Referenzsystems auf. 68,3 % der dy-

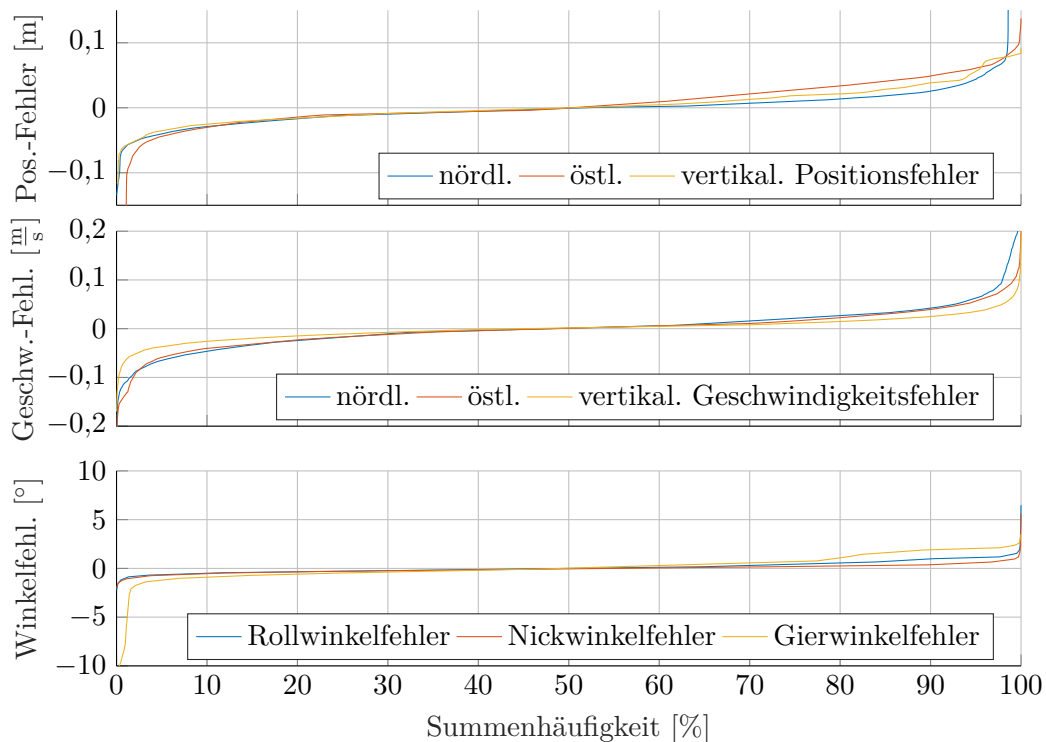


Abbildung 6.9: Kumulative Häufigkeitsverteilung der Sensorfusionsfehler

namischen Nick- und Rollwinkel Fehler sind kleiner als $\pm 0,5^\circ$ und 95,4% kleiner als $\pm 1,1^\circ$. Die Gierwinkel Fehler sind etwas größer mit 68,3% unter $\pm 0,9^\circ$ und 95,4% unter $\pm 2,1^\circ$.

Mit diesen Genauigkeiten ist die Sensorfusion gemäß Kapitel 4 gut geeignet für die entworfene Regelung und die vorliegende Anwendung. In Abschnitt 6.2 ist die Erprobung der flachheitsbasierten Regelung und der Trajektoriengenerierung auf Basis des von der Sensorfusion geschätzten Systemzustandes dargestellt.

Das hier eingesetzte Erweiterte Kalman-Filter verwendet einen „loosely coupled“ Ansatz, vgl. Seite 129. Eine Auswertung desselben Fluges für ein tightly-coupled System, bei dem anstatt der Positions- und Geschwindigkeitslösungen des GNSS-Empfängers die Pseudorange- und Deltarange-Messungen verwendet werden, ist in [44] zu finden. Der tightly-coupled Ansatz hat leichte Vorteile an den Stellen, wo die Positionslösungen des GNSS-Empfängers durch den Wechsel des Modus springen (siehe oben), da diese Sprünge auf den Pseudoranges nicht auftreten. Da in [44] die Pseudoranges und Deltaranges ohne GNSS-Korrekturdaten (vgl. Abschnitt 2.5.3) verwendet werden, sind die statischen Positionenfehler erwartungsgemäß hier deutlich kleiner als in [44]. Bei den dynamischen Positionen- und Geschwindigkeitsfehlern werden hier leicht bessere Ergebnisse als in [44] erreicht; die Lagewinkel Fehler sind in etwa gleich. In [44] wird vorgeschlagen, die loosely- und tightly-coupled Ansätze zu kombinieren, um sowohl die hohe Positionengenauigkeit durch RTK als auch die Robustheit bei schlechterem Satellitenempfang ausnutzen zu können.

6.1.2 Experimenteller GPS- und Galileo-Empfänger im Galileo-Testgebiet

Im Folgenden wird die Erprobung des experimentellen GNSS-Empfängers aus Abschnitt 4.1.2.2 im Galileo-Testgebiet automotiveGATE (Abschnitt 3.5.4) dargestellt [19]. Es wird in zwei Szenarien gezeigt, dass der Empfänger auf dem Quadrokopter funktioniert und in der Lage ist, sowohl GPS-Satelliten-Signale als auch Galileo-Pseudoliten-Signale zu verarbeiten.

Um auch interne Größen wie die Pseudoranges und Trägerphasenmessungen mit einem Laptop aufzeichnen zu können, wird der Empfänger im ersten Szenario in einem Pkw betrieben. Dies erlaubt zudem, größere Entfernungen bei höheren Geschwindigkeiten im Testgebiet zurücklegen zu können. Die GNSS-Helixantenne ist auf dem Fahrzeugdach installiert.

Bei dem Versuch startet der Pkw im Süden der Fahrdynamikfläche des ATC (siehe Abbildung 3.33) und fährt auf ihr dreimal gegen den Uhrzeigersinn einen Kreis mit etwa 200 m Durchmesser. Der Pkw verlässt dann die Fahrdynamikfläche im Westen und fährt südlich bis zum Beginn der Beschleunigungsspur, wo die Messung endet. Die vom experimentellen Empfänger mit 5 Hz berechnete Positionstrajektorie für diese Fahrt ist in Abbildung 6.10 in Blau zu sehen. Außerdem sind die Positionen der Pseudoliten im automotiveGATE eingezeichnet, vgl. Luftbild in Abbildung 3.33.

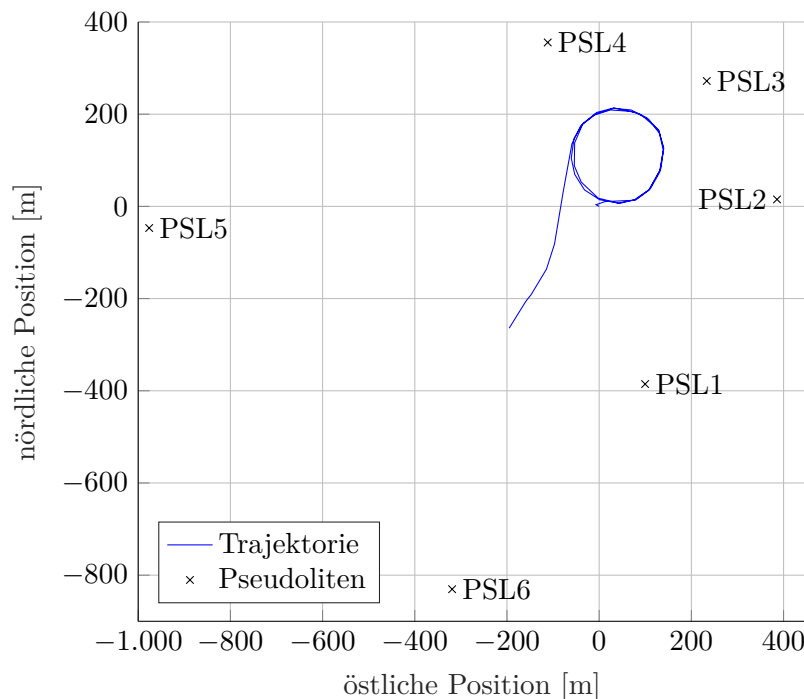


Abbildung 6.10: Pkw-Trajektorie und Pseudoliten-Positionen im automotiveGATE

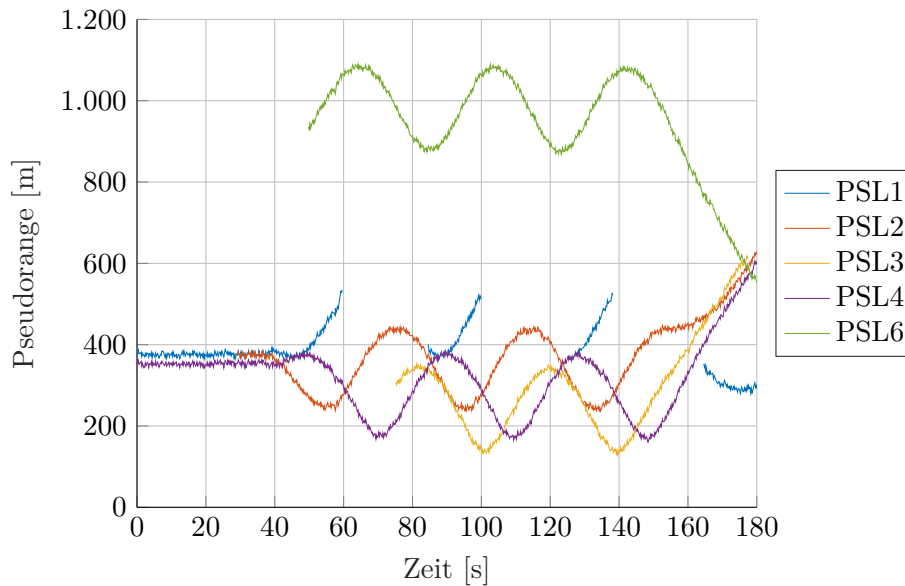


Abbildung 6.11: Offset-korrigierte Pseudoranges zu den Pseudoliten

In diesem Versuch nutzt der Empfänger sowohl Galileo-Pseudoliten- als auch GPS-Satelliten-Signale. Es ist aber auch eine Positionierung nur mit Galileo-Pseudoliten oder nur mit GPS-Satelliten möglich. Der Empfänger bestimmt die Pseudoranges zu den Pseudoliten und den GPS-Satelliten-Signalen (siehe Abschnitt 2.5.2.5). Die Pseudoliten-Pseudoranges enthalten zu dem Offset durch das Empfängeruhrenoffset außerdem ein gemeinsames Offset durch die unbekannte Pseudoliten-Systemzeit (siehe Abschnitt 3.5.4). Die Offsets lassen sich im Zuge der Positionslösung bestimmen. Die Zeitverläufe der mit den Offsets korrigierten Pseudoranges sind in Abbildung 6.11 für die Pseudoliten zu sehen. Sie entsprechen in etwa den tatsächlichen Entfernungen zwischen dem Empfänger und dem jeweiligen Pseudoliten.

Da sich die Pseudoliten nicht bewegen, ist die Bewegung des GNSS-Empfängers in den Pseudoranges gut zu erkennen. Die Sinus-Kurven zwischen 40 und 150s gehören zu den drei gefahrenen Kreisen auf der Fahrdynamikfläche. Die Bewegung in südliche Richtung nach Verlassen der Fahrdynamikfläche ist an der sich schnell verringernden Pseudorange zum Pseudoliten 6 (PSL6) zu erkennen, der sich im Süden befindet. Gleichzeitig vergrößern sich die Pseudoranges zu den nördlichen Pseudoliten 2, 3 und 4.

Das Signal des Pseudoliten 1 (Abbildung 6.11 in Blau) kann bei der Fahrt auf der Fahrdynamikfläche mehrmals etwa an der gleichen Stelle nicht mehr getrackt werden und wird dann wieder reacquiert. Dies ist auf eine zum Zeitpunkt des Versuchs vergleichbar schwach eingestellte Signalleistung der Pseudoliten zurückzuführen. Die Signalleistungen der Pseudoliten erreichen im Empfänger nur Trägerrauschverhältnisse (C/N_0), die nur leicht über dem im Empfänger konfigurierten Tracking-Schwellwert liegen. GPS-Satelliten-Signale werden mit einem um etwa 5 dB höheren C/N_0 empfangen. Für das Signal des



Abbildung 6.12: Erprobung des experimentellen GPS- und Galileo-Empfängers auf dem Quadroko­pter im automotiveGATE

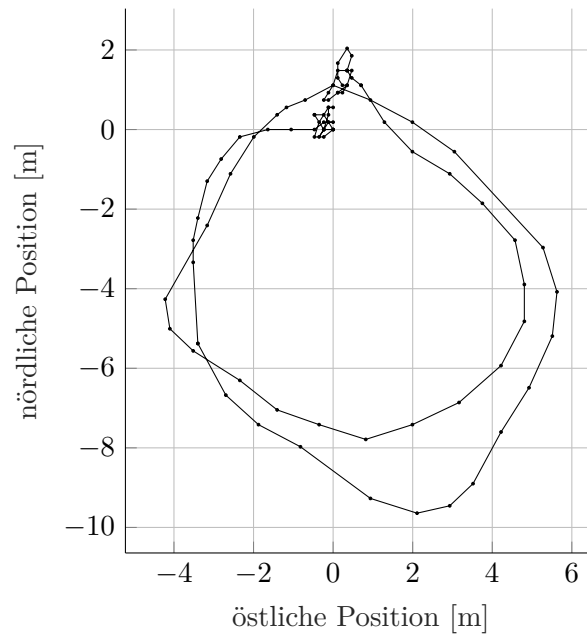


Abbildung 6.13: Positionslösung des experimentellen GNSS-Empfängers im Quadroko­pter-Flug

weit entfernten Pseudoliten 5 im Westen beginnt der Empfänger zwar mit der Akquisition, kann es jedoch nicht in ein Tracking überführen.

Im zweiten Szenario wird der experimentelle GNSS-Empfänger auf dem Quadrokofter getestet. In Abbildung 6.12 ist der Quadrokofter in einem frühen Aufbaustadium mit dem experimentellen GNSS-Empfänger (Komponenten vgl. Abbildung 4.6) während des Fluges im automotiveGATE zu sehen. Die vom Empfänger gesendeten NMEA-Nachrichten werden auf dem Quadrokofter aufgezeichnet. In Abbildung 6.13 ist die auf Basis der GPS-Satelliten-Signale berechnete Trajektorie, umgerechnet in kartesische Koordinaten, zu sehen.

Die Ergebnisse dieser ersten Versuche sind vielversprechend. Sie zeigen, dass der experimentelle GNSS-Empfänger auf dem Quadrokofter betrieben werden und sowohl Galileo-Pseudoliten- als auch GPS-Satelliten-Signale verarbeiten kann. Zur Nutzung des Empfängers in der Sensorfusion für den geregelten Flug sind weitere Untersuchungen notwendig.

6.2 Flachheitsbasierte Regelung und Trajektoriengenerierung

Zur experimentellen Validierung der Automatisierung des Quadrokofters in den verschiedenen Betriebsmodi³ werden hier die Ergebnisse mehrerer Flugversuche vorgestellt. Zum Einsatz kommt dabei die flachheitsbasierte Regelung gemäß Abschnitt 5.2 zusammen mit den verschiedenen Methoden zur Trajektoriengenerierung aus Abschnitt 5.3. Vorab wird in Abschnitt 6.2.1 die Rotorschubkalibrierung gemäß Abschnitt 5.1.2.2 vorgestellt.

In den Outdoor-Flugversuchen A1 und A2 wird die flachheitsbasierte Regelung anhand von horizontalen und vertikalen Spline-Trajektorien gemäß Abschnitt 5.3.2 getestet, siehe Abschnitt 6.2.2.

In Abschnitt 6.2.3 fliegt der Quadrokofter indoor eine gemäß Abschnitt 5.3.4 von der MPC generierte Trajektorie als reinen Positionswechsel (Flugversuch B1) und als Positionswechsel mit gleichzeitigem Gieren (Flugversuch B2).

Der Flug mit Joystick-Steuerung gemäß Abschnitt 5.3.3.1 unter Nutzung der Trajektoriengenerierung mit Kaskadenregelung wird in Abschnitt 6.2.4 als Outdoor-Flugversuch C vorgestellt.

In Abschnitt 6.2.5 wird die Objektverfolgung gemäß Abschnitt 5.3.3.2, bei der auch die Trajektoriengenerierung mit Kaskadenregelung zum Einsatz kommt, anhand dreier Outdoor-Flugversuche dargestellt. Im Flugversuch D1 wird ein statisches Zielobjekt aus unterschiedlichen Abständen und Himmelsrichtungen gefilmt, indem die Kommandos für den Abstand und den Azimut zum Zielobjekt verändert werden. Außerdem wird ein bewegtes Zielobjekt verfolgt: Im Flugversuch D2 fährt das Zielobjekt auf einer Geraden und das Azimut-Kommando wird variiert. Im Flugversuch D3 bewegt sich das Zielobjekt auf einem Halbkreis bei konstantem Abstands- und Azimut-Kommando.

³Betriebsmodi: siehe Abschnitt 3.3.1

Der Systemzustand wird während der Outdoor-Flüge durch die Sensorfusion gemäß Kapitel 4 geschätzt, deren Erprobung in Abschnitt 6.1.1 dargestellt ist.

6.2.1 Schubkalibrierung

Zu Beginn jedes Flugversuches werden die Rotorschübe gemäß Abschnitt 5.1.2.2 im Schwebeflug kalibriert. Dies wird über die QC Cam Ctrl App kommandiert, siehe Abbildung 3.16. Der Kalibrierungsvorgang wird hier anhand eines Outdoor-Fluges dargestellt. In [45] ist eine detaillierte Auswertung für eine Indoor-Kalibrierung zu finden. Abbildung 6.14 zeigt die Zeitverläufe für zwei Kalibrierungen direkt hintereinander während eines Outdoor-Fluges bei fast Windstille.

Zur Ermittlung der Rotorschub-Kalibrierungsfaktoren λ_k (5.24) werden die kommandierten Rotorschubkräfte F_{tsk} im Schwebeflug aufgezeichnet, siehe mittlerer Graph in Abbildung 6.14. Der Start und das Ende jedes Aufzeichnungsintervalls sind mit S und E markiert.

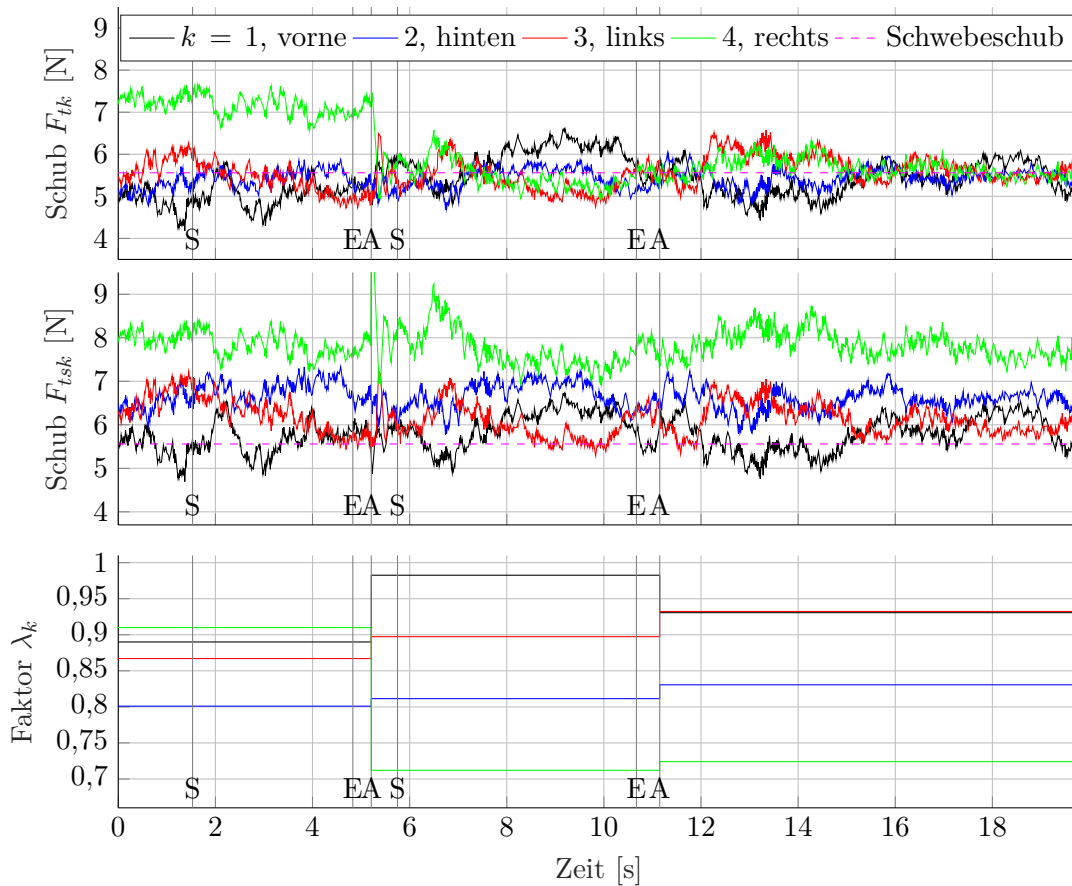


Abbildung 6.14: Schubkalibrierung während des Schwebefluges

Aus den im Intervall gemittelten F_{tsk} werden dann auf dem Quadrokopter gemäß (5.26) die Kalibrierungsfaktoren λ_k berechnet. Dabei wird die Quadrokopter-Masse m benutzt, die vor jedem Flug mit einer Waage bestimmt und per Rcon⁴ auf dem Quadrokopter eingestellt wird. Zu den mit A markierten Zeitpunkten werden jeweils die neu bestimmten Kalibrierungsfaktoren angewendet und die bis dahin aktiven überschrieben. Die jeweils aktiven Kalibrierungsfaktoren λ_k sind im unteren Graphen von Abbildung 6.14 zu sehen.

Der obere Graph von Abbildung 6.14 zeigt die vom Regler gestellten Rotorschubkräfte F_{tk} (5.110) vor der Division (5.111) durch λ_k . Im Schwebeflug sollte jeder Rotor im Mittel ein Viertel der Quadrokopter-Gewichtskraft als Schub aufbringen. Nach der ersten Kalibrierung zeigen die F_{tk} genau dies: sie bewegen sich gleichmäßig um den zu erwartenden Schwebeschub von $F_{tk} = mg/4$ (gestrichelte, magentafarbene Linie). Vorher weicht der Schub F_{t4} des rechten Rotors stark davon ab, der initiale Kalibrierungsfaktor λ_4 passte nicht. Nach den Kalibrierungen ist λ_4 mit ca. 0,72 der kleinste Kalibrierungsfaktor. Der effektive⁵ Schub des rechten Rotors ist also deutlich kleiner als vom Rotorschubmodell (5.22) prädiziert. Dies ist auch an F_{ts4} zu sehen: Der kommandierte Schub des rechten Rotors muss größer als der der anderen sein, um auf der Stelle zu schweben. Alle λ_k sind kleiner als eins. Die auf der Schubwaage gemessenen Rotorschübe, die Grundlage für das Rotorschubmodell mit den Polynom-Koeffizienten a_t gemäß Tabelle 5.2 sind, sind also größer als die im Flug.

6.2.2 Flachheitsbasierte Regelung

In diesem Abschnitt wird die flachheitsbasierte Regelung mit vorgegebenen Trajektorien als Führungsgrößen $\vec{y}_{i,s}$ (5.115d) im Outdoor-Flugversuch erprobt. Es werden die Spline-Trajektorien aus Abschnitt 5.3.2 im Flugversuch A1 als horizontale und im Flugversuch A2 als vertikale Solltrajektorien relativ zur initialen Quadrokopterposition verwendet. Die Kommandierung erfolgt über Events in der QC Cam Ctrl App (vgl. Abschnitt 3.3.3).

Die gleichen Spline-Trajektorien werden verwendet, um einen statischen Wegpunkt anzufliegen. Dazu werden die Spline-Trajektorien bei der Kommandierung onboard passend in Länge und Zeit skaliert, siehe Abschnitt 5.3.2. Skalierte vertikale Spline-Trajektorien werden zum automatischen Starten und Landen eingesetzt, vgl. „Startphase“ auf Seite 197.

Flugversuch A1 „Spline-Trajektorie horizontal“ Zur Validierung der horizontalen Regelung kommen Spline-Trajektorien 5. Grades zum Einsatz (vgl. Abbildung 5.9). Eine solche Trajektorie wurde auch für die simulative Erprobung in Abschnitt 5.5 verwendet. Die Positionswechsel-Trajektorien $y_{2,s}$ und ihre Ableitungen $\dot{y}_{2,s}$, $\ddot{y}_{2,s}$, $y_{2,s}^{(3)}$, $y_{2,s}^{(4)}$, die in zwei Skalierungen als Führungsgrößen auf den Ost-Kanal gegeben werden, sind in

⁴Rcon: siehe Abbildung 3.10

⁵Dies kann auch an einer ungleichmäßigen Masseverteilung liegen.

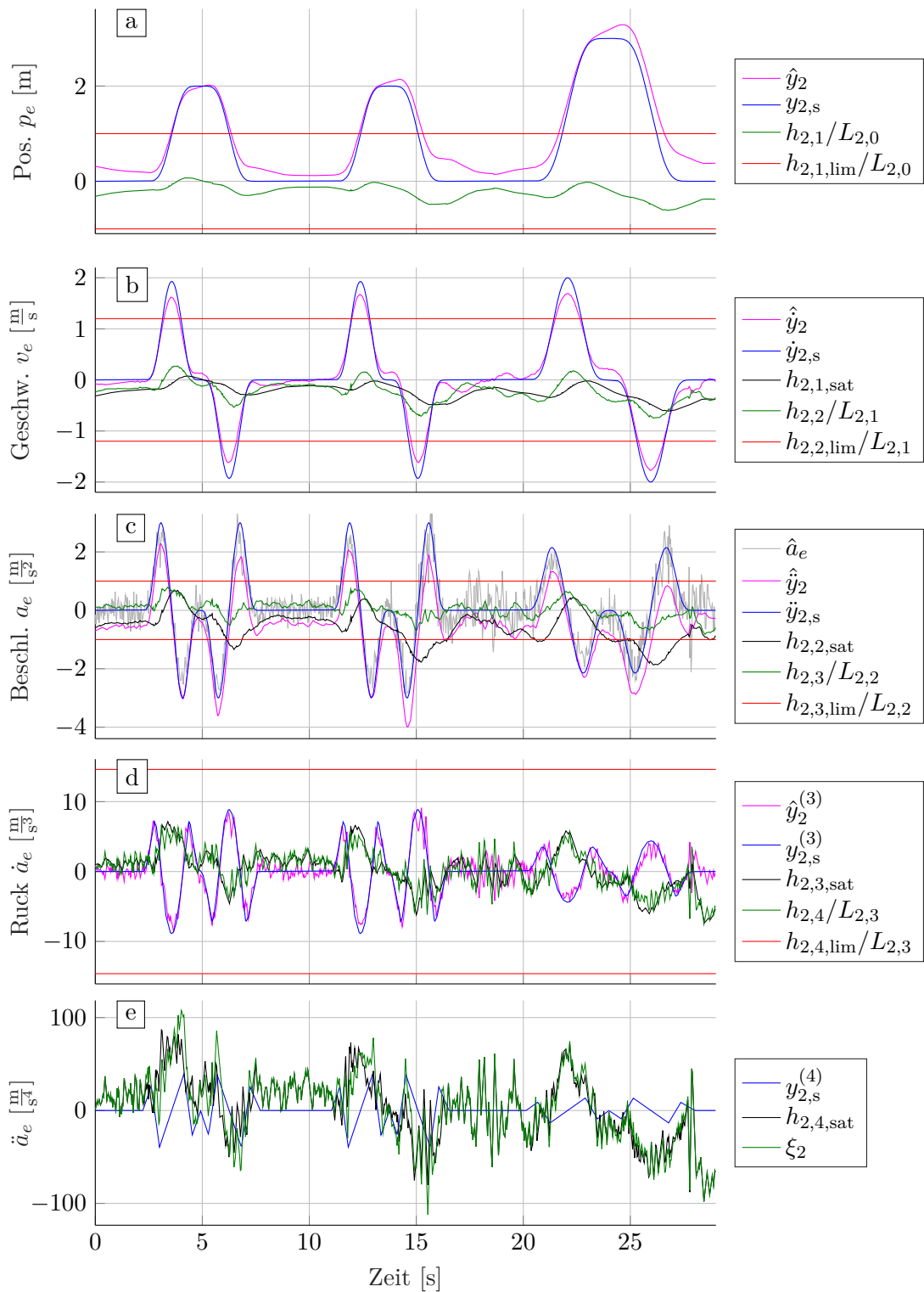


Abbildung 6.15: Flugversuch A1 „Spline-Trajektorie horizontal“: Trajektorienfolgeregelung, Ost-Kanal

Abbildung 6.15 als blaue Kurven dargestellt. Zu den Zeitpunkten $t = 2\text{ s}$ und 11 s wird jeweils ein Positionswechsel über 2 m (max. Geschwindigkeit $1,9\text{ m/s}$, max. Beschleunigung 3 m/s^2) und zum Zeitpunkt $t = 20\text{ s}$ ein Positionswechsel über 3 m (max. Geschwindigkeit 2 m/s , max. Beschleunigung $2,15\text{ m/s}^2$), jeweils hin und zurück, kommandiert.

Neben den Führungsgrößen sind in Abbildung 6.15 auch die anderen Größen der Reglerkaskaden (5.113) der Trajektorienfolgeregelung (vgl. Abbildung 5.7) für den Ost-Kanal gegeben. In Graph a ist zu sehen, dass die Position \hat{y}_2 des Quadropters in östlicher Richtung in Magenta recht gut mit der Positionstrajektorie $y_{2,s}$ übereinstimmt. Die maximale Abweichung beträgt 60 cm . Der Integrierer (5.113c) wird für die horizontalen Kanäle nicht verwendet, damit ist $h_{2,0,\text{sat}} = 0$. Die Hilfsstellgröße $h_{2,1,\text{sat}}$ in Grün ergibt sich daher gemäß (5.113b) aus dem negativen Positionsfehler $y_{2,s} - \hat{y}_2$ multipliziert mit dem Übertragungsbeiwert $L_{2,0}$ (Tabelle C.3). $h_{2,1,\text{sat}}$ überschreitet nicht die Beschränkung $h_{2,1,\text{lim}}$ (Tabelle C.4) in Rot, damit ist $h_{2,1,\text{sat}} = h_{2,1}$.

Die Extrema der Geschwindigkeit \hat{y}_2 in Graph b in Magenta erreichen nicht ganz die Extrema der Führungsgröße $y_{2,s}$ in Blau. Der Geschwindigkeitsfehler ergibt zusammen mit der Hilfsstellgröße $h_{2,1,\text{sat}}$ des Positionsreglers in Schwarz gemäß (5.113b) die Hilfsstellgröße $h_{2,2}$, die ihre Beschränkung $h_{2,2,\text{lim}} = 5\text{ m/s}^2$ (abweichend von Tabelle C.4) auch nicht überschreitet.

Der Beschleunigungsregler ist in Graph c dargestellt. In Magenta ist die im Regler verwendete, gemäß (5.116) geschätzte Beschleunigung \hat{y}_2 zu sehen. Hier wird kein Störgrößen-Beobachter verwendet, daher ist $\hat{a}_{se} = 0$ (die Flugversuche C, D1 – D3 verwenden den Störgrößen-Beobachter, für Details siehe Flugversuch D3 ab Seite 233). In Grau ist die geschätzte Beschleunigung \hat{a}_e (5.14) dargestellt, deren gleitender Mittelwert ein Offset zu \hat{y}_2 aufweist, das wahrscheinlich auf eine Störung durch Wind zurückzuführen ist. Die Hilfsstellgröße $h_{2,3}$ gemäß (5.113b) bleibt auch unter der Beschränkung $h_{2,3,\text{lim}} = 9\text{ m/s}^3$.

In Graph d für den Ruckregler stimmt der gemäß (5.117) geschätzte Ruck $\hat{y}_2^{(3)}$ in Magenta gut mit der Ruck-Führungsgröße $y_{2,s}^{(3)}$ in Blau überein, sodass sich die Hilfsstellgröße $h_{2,4}$ in Grün überwiegend aus $h_{2,3,\text{sat}}$ in Schwarz ergibt. Die Beschränkung $h_{2,4,\text{lim}} = 190\text{ m/s}^4$ wird von $h_{2,4}$ nicht überschritten.

Die Stellgröße ξ_2 ist in Graph e in Grün dargestellt. Sie setzt sich gemäß (5.113a) aus der Führungsgröße $y_{2,s}^{(4)}$ in Blau und der Hilfsstellgröße $h_{2,4,\text{sat}}$ in Schwarz zusammen.

Flugversuch A2 „Spline-Trajektorie vertikal“ Auf den Vertikalkanal wird eine Spline-Trajektorie 2. Grades gegeben, siehe blaue Kurven in Abbildung 6.16. Es wird zweimal eine Höhenänderung von 30 cm mit einer maximalen Geschwindigkeit von $0,4\text{ m/s}$ und einer maximalen Beschleunigung von 1 m/s^2 kommandiert. Der Quadropters folgte der Höhenvorgabe sehr gut; die vertikale Position \hat{y}_3 in Graph a in Magenta weicht kaum von der Führungsgröße $y_{3,s}$ ab. Der maximale Höhenfehler beträgt $3,5\text{ cm}$. Der Ausgang $h_{3,0}$ des Integrierers (5.113c), der für den Vertikal-Kanal aktiv ist und die Beschränkung $h_{3,0,\text{lim}}$ nicht überschreitet, ist in Schwarz dargestellt. Die resultierende Hilfsstellgröße $h_{3,1}$ in Grün hält wiederum ihre Beschränkung $h_{3,1,\text{lim}}$ in Rot mit großer Reserve ein.

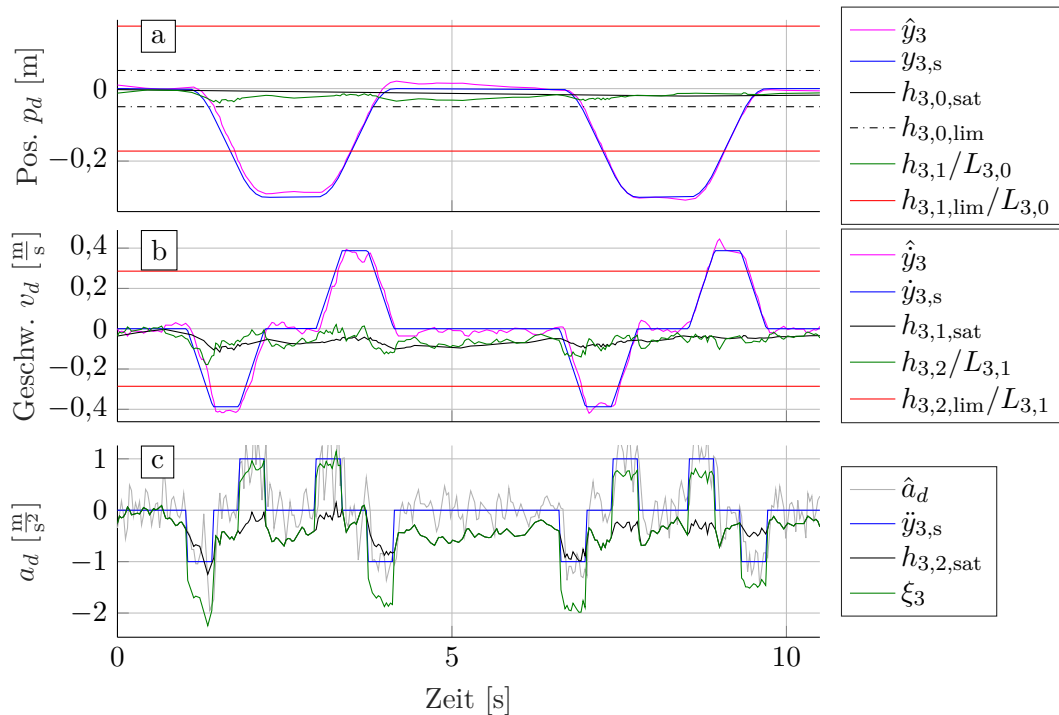


Abbildung 6.16: Flugversuch A2 „Spline-Trajektorie vertikal“: Trajektorienfolgeregelung, Vertikalkanal

Die vertikale Geschwindigkeit \hat{y}_3 in Graph b in Magenta stimmt ebenfalls gut mit der Führungsgröße $\dot{y}_{3,s}$ in Blau überein. Aus deren Differenz und $h_{3,1,sat}$ in Schwarz ergibt sich gemäß (5.113b) die Hilfsstellgröße $h_{3,2}$ des Geschwindigkeitsreglers in Grün, die ihre Beschränkung $h_{3,1,lim}$ in Rot nicht überschreitet.

In Graph c ist in Grau die geschätzte Beschleunigung \hat{a}_d gemäß (5.14) gegeben. Ihr gleitender Mittelwert stimmt sehr gut mit der vorgesteuerten Führungsgröße $\ddot{y}_{3,s}$ in Blau überein. Dies ist auch auf die Schubkalibrierung zurückzuführen. Die Stellgröße ξ_3 in Grün ergibt sich gemäß (5.113a) aus der Summe von $\ddot{y}_{3,s}$ und $h_{3,2,sat}$ in Schwarz. Aus ξ_3 wird gemäß (5.106) die Systemstellgröße u_1 berechnet, hier mit $\hat{a}_{sd} = 0$.

Fazit flachheitsbasierte Regelung Die flachheitsbasierte Regelung gemäß Abschnitt 5.2 konnte in den Outdoor-Flugversuchen A1 und A2 erfolgreich validiert werden. Das System folgt den Solltrajektorien trotz externer Störungen ohne signifikante Abweichungen. Es ist hervorzuheben, dass es durch die Vorsteuerung aller Ableitungen – wie schon in der Simulation (siehe Abschnitt 5.5) – zu keiner merklichen Verzögerung im Folgeverhalten kommt. Die entwickelte flachheitsbasierte Regelung ist somit wie erwartet für die Trajektorienfolge sehr gut geeignet.

6.2.3 MPC-Trajektoriengenerierung

Die Trajektoriengenerierung mit MPC aus Abschnitt 5.3.4 wird zusammen mit der flachheitsbasierten Regelung im Indoor-Flugversuch erprobt [18], siehe Abbildung 6.17.

Da der GNSS-Empfänger indoor nicht eingesetzt werden kann, wird das Infrarot-Trackingsystem genutzt, siehe Abschnitt 3.5.2. Das Trackingsystem bestimmt die Position und Orientierung des Tracking-Baums, der oben auf dem Quadrokopter montiert ist (siehe Abbildung 6.17). Daraus kann der Quadrokopter die Position des Ursprungs des körperfesten Koordinatensystems und dessen Orientierung berechnen. Für die Regelung ist die Updaterate von 60 Hz ausreichend hoch und die Latenz über WLAN klein genug. Daher wird hier beim Indoorflug auf eine Sensorfusion mit Kalman-Filter gemäß Kapitel 4 verzichtet. Die Geschwindigkeit wird durch Differentiation der Trackingsystem-Positionsdaten auf dem Quadrokopter berechnet. Als Drehraten werden die Bias-korrigierten Drehraten der IMU genutzt. Das Drehraten-Bias wird im Stillstand initialisiert (4.128) und dann konstant gehalten.

Wie bei den Flugversuchen A1 und A2 wird kein Störgrößen-Beobachter verwendet, d. h. es gilt $\hat{a}_s = 0$. Da die Beschleunigungen gemäß (5.116) geschätzt werden, werden die Beschleunigungssensoren hier nicht benötigt.

In den Flugversuchen umgesetzt wird das Szenario 1 von Seite 191 mit der Positionswechsel-Trajektorie gemäß Abbildung 5.12. Hierbei handelt es sich um die gleiche Spline-Trajektorie 5. Grades wie auch im Flugversuch A1, die aufgrund des beschränkten Erfassungsbereichs des Infrarot-Trackingsystems auf eine Distanz von 30 cm zwischen den beiden Positionen skaliert wird. Zu Beginn der Versuche werden die Rotorschübe gemäß Abschnitt 5.1.2.2 kalibriert.

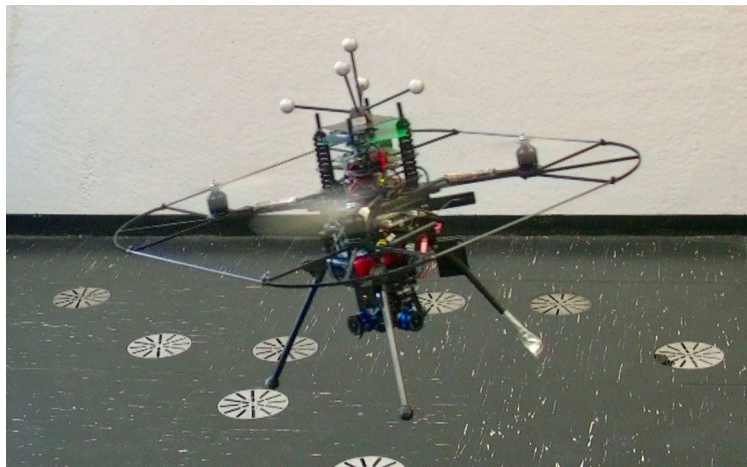


Abbildung 6.17: Quadrokopter im Indoor-Flugversuch

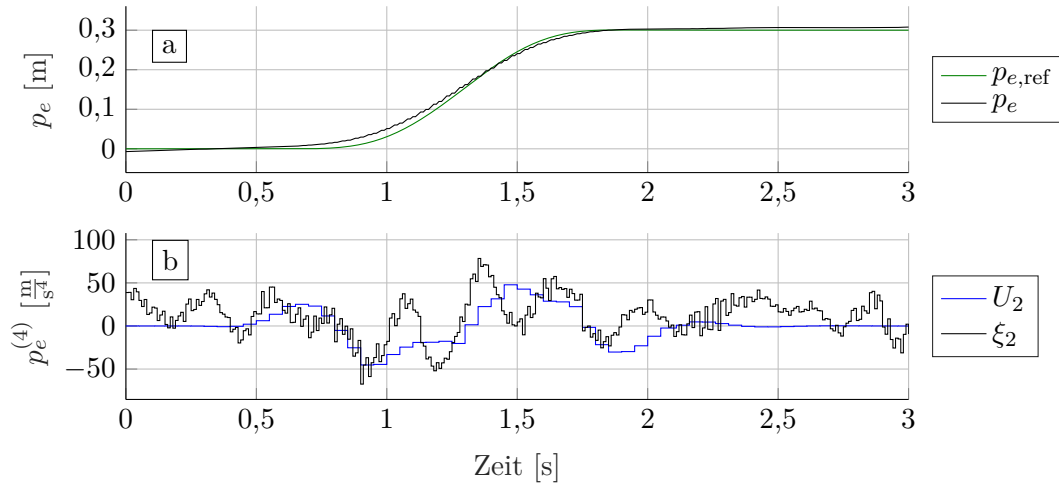


Abbildung 6.18: Flugversuch B1 „MPC-Trajektorie horizontal“

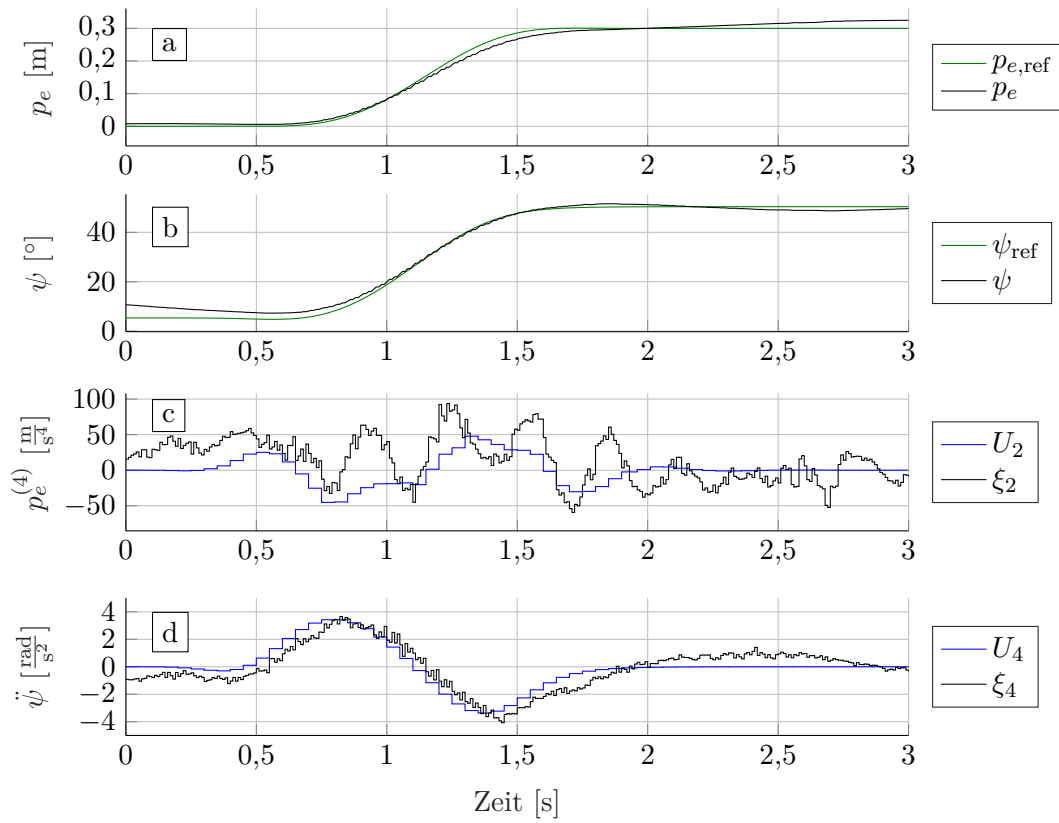


Abbildung 6.19: Flugversuch B2 „MPC-Trajektorie horizontal mit Gieren“

Flugversuche B1 „MPC-Trajektorie horizontal“ und B2 „MPC-Trajektorie horizontal mit Gieren“ Es werden zwei Flüge durchgeführt: im Flugversuch B1 wird ein reiner Positionswechsel und im Flugversuch B2 ein Positionswechsel mit gleichzeitigem Gieren geflogen. Die Ergebnisse sind in den Abbildungen 6.18 und 6.19 dargestellt.

Die unteren Graphen der beiden Abbildungen zeigen jeweils die von der MPC generierten Stellgrößen $U_2 = p_{e,s}^{(4)}$ (und $U_4 = \ddot{\psi}_s$) (5.165) und die Ausgänge ξ_2 (und ξ_4) (5.113a) der Trajektorienfolgeregelung.

In Graph a von Abbildung 6.18 ist zu sehen, dass die Ost-Solltrajektorie $p_{e,\text{ref}}$ beim reinen Positionswechsel wie gewünscht eingeregelt wird. Beim gleichzeitigen Gieren bleibt die Abweichung von der Ost-Solltrajektorie vergleichbar klein (Abbildung 6.19, Graph a). Der Gierwinkel folgt der Solltrajektorie ψ_{ref} ebenfalls gut, siehe Abbildung 6.19, Graph b.

Fazit MPC-Trajektoriengenerierung Die Flugversuche B1 und B2 demonstrieren, dass die MPC-Trajektoriengenerierung für den Betrieb auf dem Quadrokopter gut geeignet ist. Sie erzeugt Trajektorien, die von der flachheitsbasierten Regelung erfolgreich eingeregelt werden können. Die flachheitsbasierte Regelung zeigt auch hier im Indoorflug ein sehr gutes Trajektorienfolgeverhalten ohne wesentliche Regelabweichungen. Die Versuche verdeutlichen die Entkopplung der Systemkanäle durch die flachheitsbasierte Zustandsrückführung: Die translatorischen Kanäle und der Gier-Kanal lassen sich wie erwartet unabhängig voneinander einregeln und beeinflussen sich gegenseitig nicht.

6.2.4 Flug mit Joystick

Beim Flug mit Joystick gemäß Abschnitt 5.3.3.1 wird die Trajektoriengenerierung mit Kaskadenregelung und die flachheitsbasierte Regelung genutzt. Die Kommandierung erfolgt über den 6-Achs-Joystick des ControlPads (vgl. Abschnitt 3.3.2), siehe Abbildung 6.20. Mit der Joystick-Steuerung kann der Operator beliebige Manöver fliegen oder auch ein Zielobjekt manuell verfolgen. Hier wird exemplarisch die horizontale Steuerung mit Joystick im Outdoor-Flug gezeigt.



Abbildung 6.20: Steuerung des Quadrokopters mit dem 6-Achs-Joystick des ControlPads

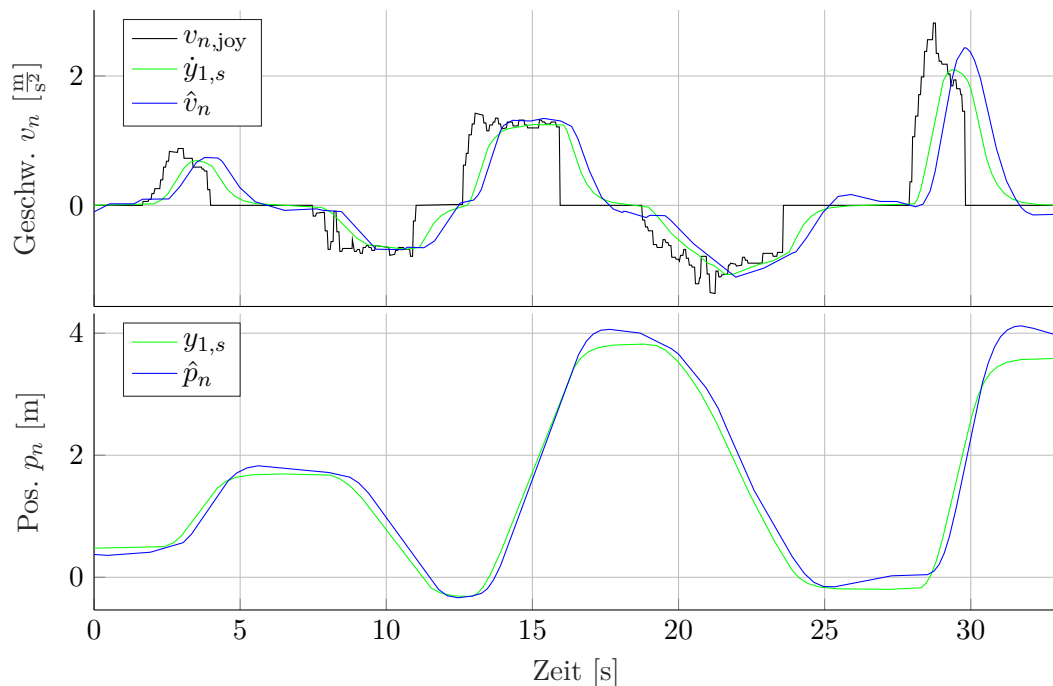


Abbildung 6.21: Flugversuch C „Joystick“: Geschwindigkeit und Position in nördlicher Richtung

Flugversuch C „Joystick“ In Abbildung 6.21 sind die Geschwindigkeit und die Position in nördlicher Richtung für den Flug mit Joystick-Kommandos dargestellt. Das Joystick-Geschwindigkeitskommando $v_{n,joy}$ gemäß (5.139) ist im oberen Graphen in Schwarz zu sehen. Aus diesem Kommando berechnet die Trajektoriengenerierung mit Kaskadenregelung in (5.134) mit den Parametern aus den Tabellen C.5 und C.6 die Führungsgrößen (5.135) u.a. für Position $y_{1,s}$ und Geschwindigkeit $\dot{y}_{1,s}$ in nördliche Richtung, die als grüne Kurven dargestellt sind. $\dot{y}_{1,s}$ folgt dem Joystick-Geschwindigkeitskommando $v_{n,joy}$ wie gewünscht ohne Überschwingen und unter Beachtung der Beschränkungen für Geschwindigkeiten, Beschleunigungen, Ruck etc., vgl. Entwurf in Abschnitt 5.3.3.1.

Die flachheitsbasierte Regelung aus Abschnitt 5.2 regelt die vorgegebene Geschwindigkeits- und Positionstrajektorie $\dot{y}_{1,s}$ und $y_{1,s}$ ein. Die höheren Ableitungen der Führungsgrößen (5.135) sind dabei als Vorsichtsmaßnahme ab der zweiten Ableitung $(\dot{y}_{1,s}\ddot{y}_{2,s}) = \ddot{\vec{p}}_{ne,s,k}^T$ zu null gesetzt. Die resultierende Geschwindigkeit \hat{v}_n und Position \hat{p}_n des Quadropters in nördlicher Richtung sind in Abbildung 6.21 blau eingezeichnet. Sie folgen den Führungsgrößen $\dot{y}_{1,s}$ und $y_{1,s}$ ohne größere Abweichungen.

Während der Quadropters in nördlicher Richtung den Joystick-Kommandos folgt, sind die Sollwerte für die östliche und vertikale Position sowie für den Gierwinkel konstant. Die maximale Positionsabweichung in östlicher Richtung beträgt 22 cm und die in vertikaler Richtung 4 cm. Der maximale Gierwinkelfehler liegt bei 8° .

Fazit Flug mit Joystick Wie in Flugversuch C zu sehen ist, lässt sich der Quadrocopter wie gewünscht mit dem 6-Achs-Joystick des ControlPads per Hand steuern. Die Trajektoriengenerierung mit Kaskadenregelung generiert aus den Joystick-Kommandos Trajektorien, die die eingestellten Beschränkungen berücksichtigen und von der flachheitsbasierten Regelung präzise umgesetzt werden. Der Joystick-Betriebsmodus erlaubt damit neben der automatischen Objektverfolgung, die im nächsten Abschnitt demonstriert wird, die manuelle Steuerung eines oder mehrerer Kanäle durch den Operator.

6.2.5 Objektverfolgung

In diesem Abschnitt werden die Outdoor-Flugversuche zur Erprobung der Objektverfolgung gemäß Abschnitt 5.3.3.2 dargestellt. Wie bei der Joystick-Steuerung finden die Trajektoriengenerierung mit Kaskadenregelung und die flachheitsbasierte Regelung Verwendung. Das Zielobjekt führt die Ziel-Einheit mit, die die GNSS-RTK-Position über das XBee-Mesh-Netzwerk an den Quadrocopter sendet, vgl. Abschnitt 3.4. Hier wird als Zielobjekt exemplarisch das etwa 1 m lange Testfahrzeug genutzt, das in Abbildung 3.30 zu sehen ist. Alternativ könnten natürlich auch z. B. andere Fahrzeuge oder Personen verfolgt werden.

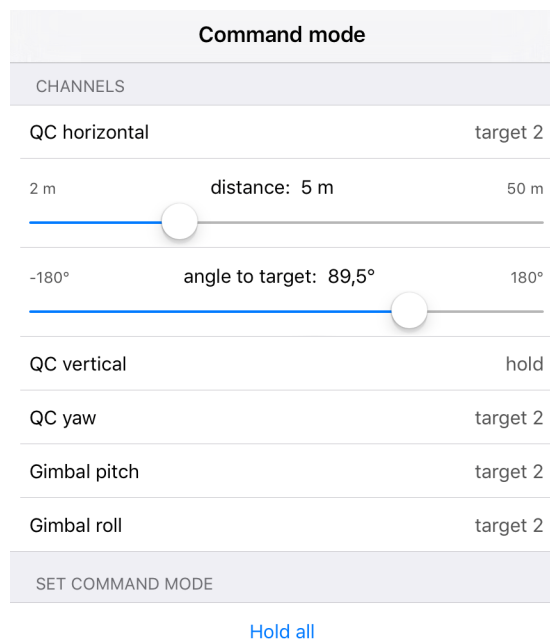


Abbildung 6.22: Kommandierung Abstand („distance“) und Azimut („angle to target“) zur Objektverfolgung in der QC Cam Ctrl App

Es werden drei verschiedene Flugversuche vorgestellt:

- **Flugversuch D1 „Statisches Ziel“:** Zielobjekt hält Position, Änderung des Abstands-Kommandos δ_{cmd} und des Azimut-Kommandos α_{cmd}
- **Flugversuch D2 „Objektverfolgung Gerade“:** bewegtes Zielobjekt auf Gerade, Änderung des Azimut-Kommandos α_{cmd} bei konstantem δ_{cmd}
- **Flugversuch D3 „Objektverfolgung Halbkreis“:** bewegtes Zielobjekt auf Halbkreis, Kommandos δ_{cmd} und α_{cmd} konstant

Flugversuch D1 „Statisches Ziel“ Es wird ein statisches Zielobjekt aus unterschiedlichen Abständen und Himmelsrichtungen gefilmt. Der Operator ändert dazu über die QC Cam Ctrl App auf dem ControlPad die Sollwerte für den Abstand δ (5.120) zum Zielobjekt und den Azimut α (5.121), aus dem das Zielobjekt gefilmt werden soll, siehe Abbildung 6.22. Die Höhe wird dabei vom Quadrokopter konstant gehalten.

In diesem Fall sendet die Ziel-Einheit die statische Position an den Quadrokopter. Alternativ könnte sie als „Point-Of-Interest“ über die QC Cam Ctrl App vorgegeben werden.

Die Verläufe der Kommandos für δ und α sind in Abbildung 6.23 dargestellt. Zunächst ist $\delta_{\text{cmd}} = 4,3\text{m}$ und der Operator ändert das Azimut-Kommando α_{cmd} von 159° kontinuierlich auf 86° . Dann bleibt α_{cmd} konstant und ab $t = 14\text{s}$ wird δ_{cmd} nach und nach auf $2,6\text{m}$ verringert. Schließlich wird nochmal α_{cmd} zunächst kontinuierlich und dann sprunghaft verändert. Bei $t = 40\text{s}$ ist der Unterschied zwischen α_{cmd} und α_r gut zu

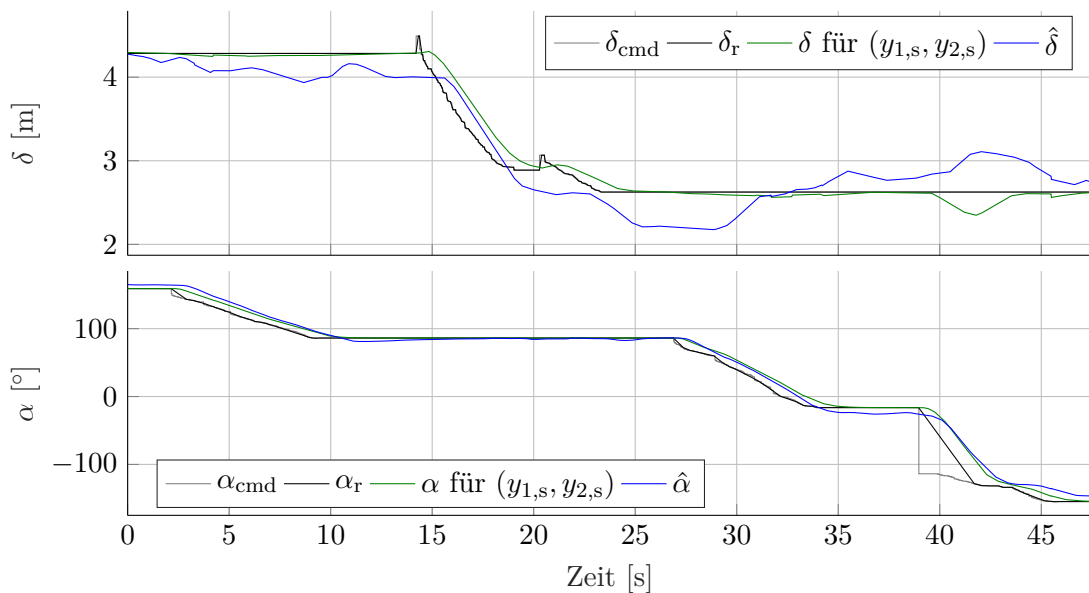


Abbildung 6.23: Flugversuch D1 „Statisches Ziel“: Abstand zum Zielobjekt δ und Azimut α

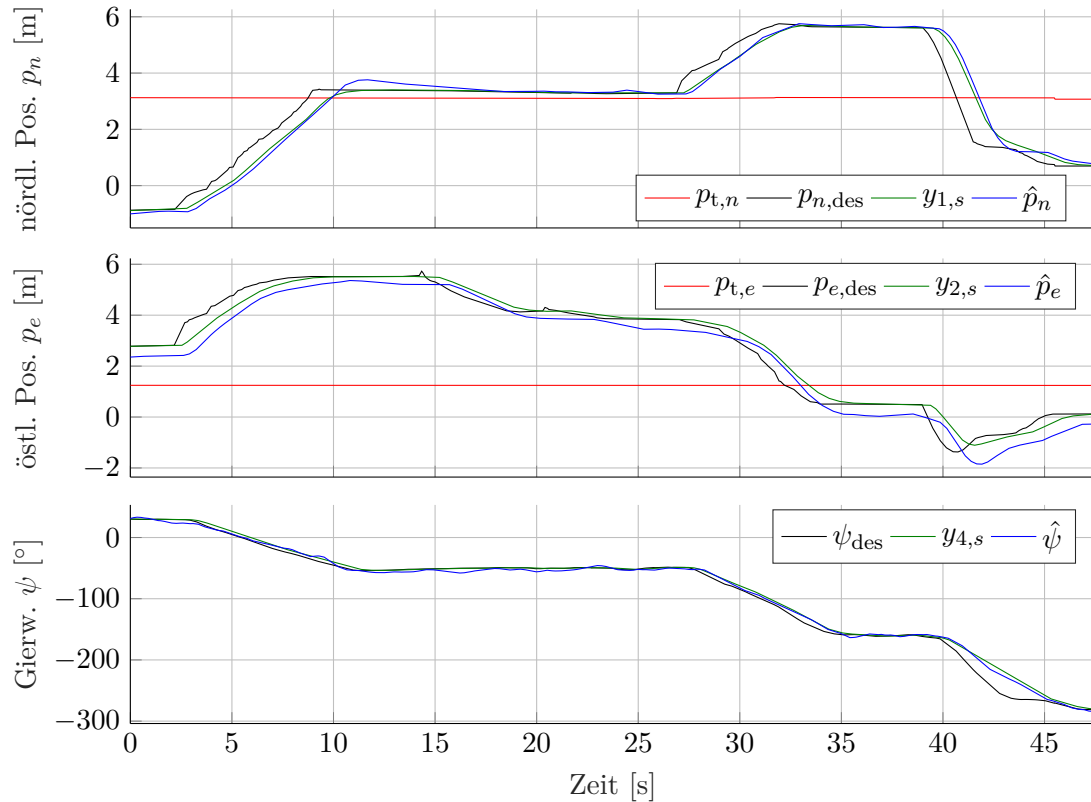


Abbildung 6.24: Flugversuch D1 „Statisches Ziel“: nördl. und östl. Position, Gierwinkel

erkennen: Während α_{cmd} durch das Kommando des Operators springt, verringert sich α_r gemäß (5.143) rampenförmig mit einer Rate von $\dot{\alpha}_{\text{lim},k} \approx 44^\circ$ (5.144). Auch die Änderungen von δ_r sind gemäß (5.142) rampenförmig mit der maximalen Änderungsrate $\dot{\delta}_{\text{lim}}$, dies ist jedoch in Abbildung 6.23 kaum zu erkennen, da in δ_{cmd} keine großen Sprünge vorkommen.

Die aus dem zeitlichen Verlauf von δ_r und α_r resultierende gewünschte Position $\vec{p}_{ne,\text{des}}$ (5.145) des Quadropters ist in Abbildung 6.24 und auf der Karte in Abbildung 6.25 als schwarze Kurve zu sehen. Auf der Karte erscheint $\vec{p}_{ne,\text{des}}$ als zwei Kreisbögen mit den Radien $\delta_{r,k}$ um die Position $\vec{p}_{t,ne}$ des Zielobjekts in Rot.

Als grüne Kurven sind in den Abbildungen 6.24 und 6.25 die durch die Trajektoriengenerierung mit Kaskadenregelung (5.146), (5.134) mit den Parametern aus Tabellen C.5 und C.6 berechneten Positions-Führungsgrößen $(y_{1,s}, y_{2,s})^T = (p_{n,s}, p_{e,s})^T$ (5.135) dargestellt. $y_{1,s}$ und $y_{2,s}$ folgen den gewünschten Positionen $p_{n,\text{des}}$ bzw. $p_{e,\text{des}}$ ohne Überschwingen und halten dabei die gewünschten Beschränkungen für Geschwindigkeiten, Beschleunigungen, Ruck etc. ein, vgl. Seite 187.

Die durch die flachheitsbasierte Regelung eingeregelt und durch die Sensorfusion geschätzten Positionen \hat{p}_{ne} des Quadropters sind in den Abbildungen 6.24 und 6.25 blau

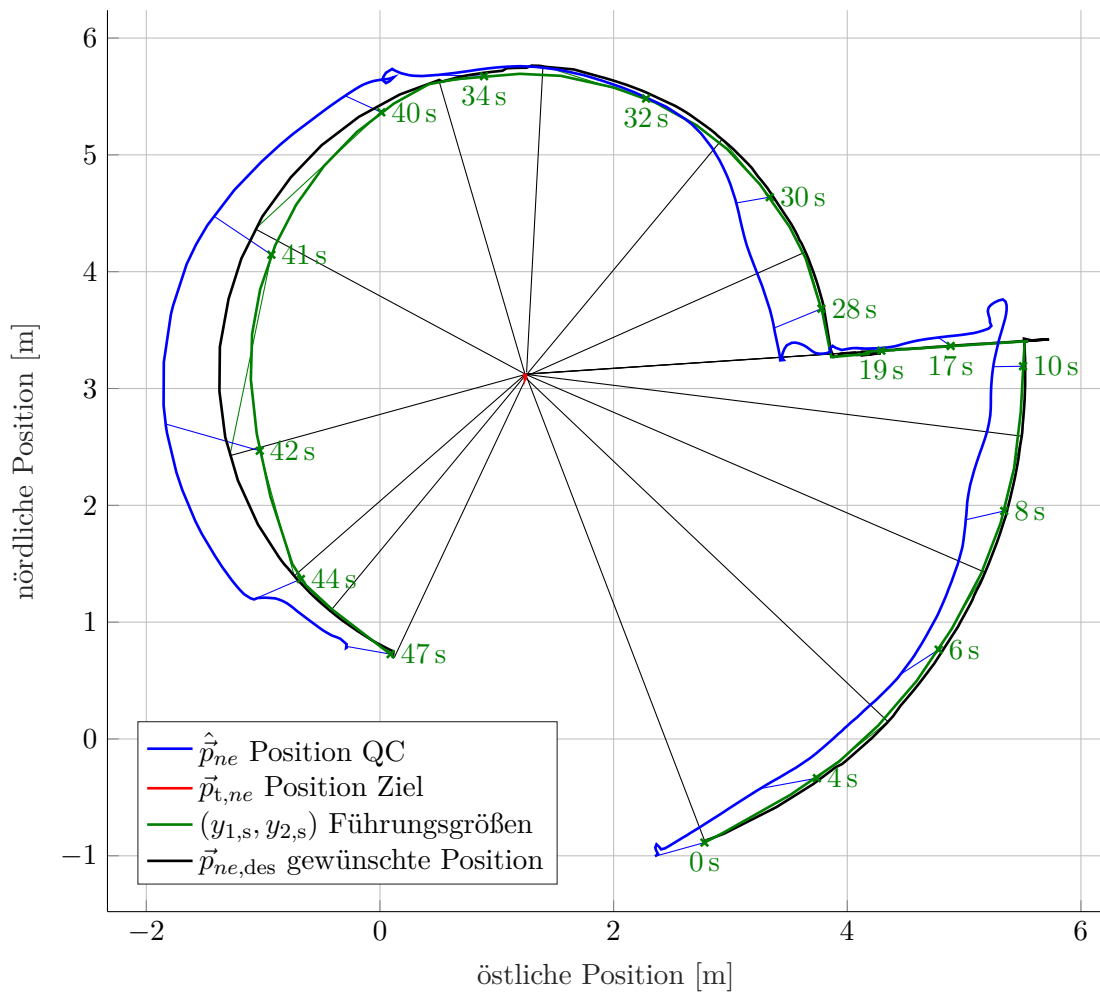


Abbildung 6.25: Karte des Flugversuchs D1 „Statisches Ziel“

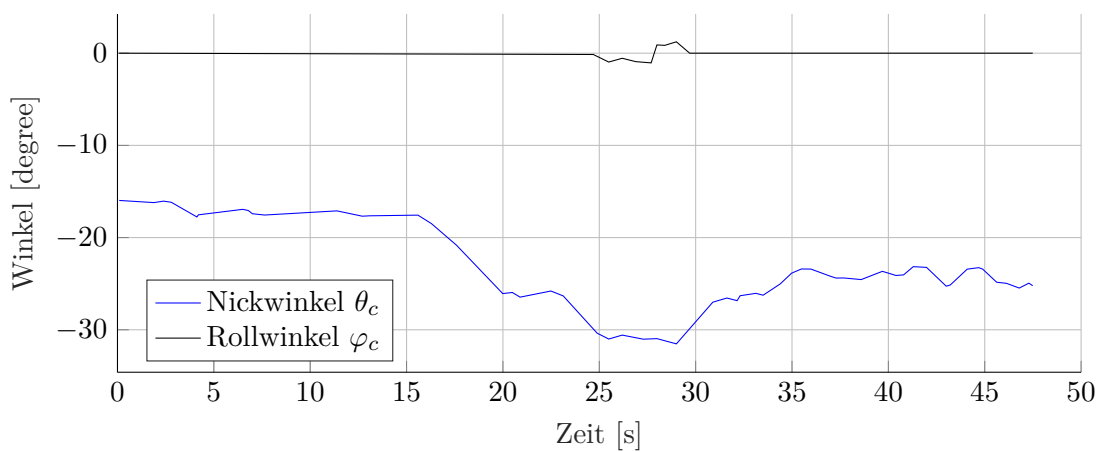


Abbildung 6.26: Flugversuch D1 „Statisches Ziel“: kommandierte Gimbal-Nick- und Rollwinkel

eingezeichnet. Man erkennt, dass die Ist-Positionen den Führungsgrößen in Grün gut folgen. Die maximale Abweichung beträgt unter 40 cm bei $t = 11,5$ s in nördlicher und ca. 80 cm bei $t = 42$ s in westlicher Richtung. Der aus den Ist-Positionen \hat{p}_{ne} resultierende Abstand $\hat{\delta}$ zum Zielobjekt und Azimut $\hat{\alpha}$ sind in Abbildung 6.23 in Blau dargestellt.

Im unteren Graphen von Abbildung 6.24 ist in Schwarz der zur Objektverfolgung notwendige Quadrokopter-Gierwinkel $\psi_{des,k}$, der gemäß (5.132) für die aktuelle Quadrokopterposition \hat{p}_{ne} und $\gamma_{des} = 0$ berechnet wird, dargestellt. Die grüne Kurve ist die daraus mit (5.147), (5.137) berechnete Gierwinkel-Trajektorie $y_{4,s} = \psi_s$. Zwischen $t = 40$ bis $t = 45$ s ist die Beschränkung der Gierrate auf $\dot{\psi}_{lim} = 20^\circ/s$ erkennbar. Der tatsächliche Gierwinkel $\hat{\psi}$ des Quadrokopters folgt der Führungsgröße $y_{4,s}$ ohne große Abweichungen mit einem maximalen Fehler von unter 9° .

In Abbildung 6.26 sind die während des Flugversuchs kommandierten Gimbal-Nickwinkel θ_c (5.129) und Gimbal-Rollwinkel φ_c (5.128) zu sehen. Der Nickwinkel θ_c korreliert erwartungsgemäß mit dem Abstand $\hat{\delta}$ zum Zielobjekt in Abbildung 6.23, da die Höhe von Zielobjekt und Quadrokopter in etwa konstant bleiben. Der Rollwinkel φ_c ist überwiegend null, da das Zielobjekt zu weit weg ist, als dass eine Verfolgung durch den Gimbal-Rollwinkel sinnvoll wäre, siehe Erläuterungen auf Seite 179. Erst wenn ζ_c (5.127) unter die Schwelle $\zeta_{c,lim,\varphi,2} = -30^\circ$ fällt, wird ein Rollen der Kamera kommandiert.

Bei der Verfolgung soll das Zielobjekt bestmöglich mittig im Kamerabild gehalten werden. Um zu analysieren, wie gut dies erreicht wird, wird das Video der Quadrokopter-Kamera

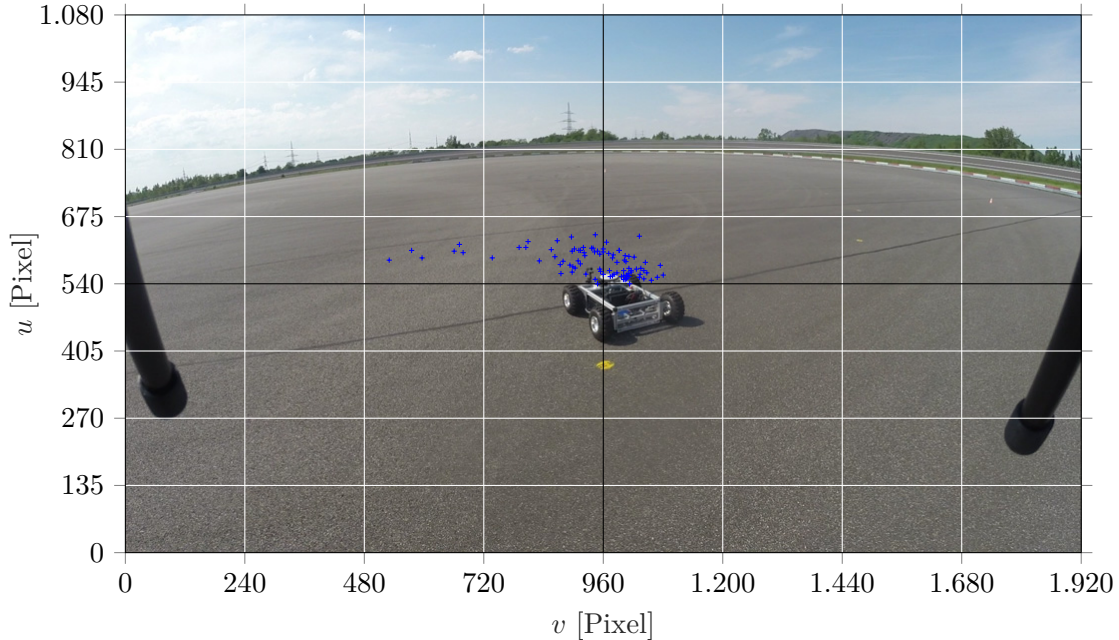


Abbildung 6.27: Flugversuch D1 „Statisches Ziel“: Positionen des Zielobjekts im Video (blaue Kreuze), Videobild zum Zeitpunkt $t = 5,2$ sec

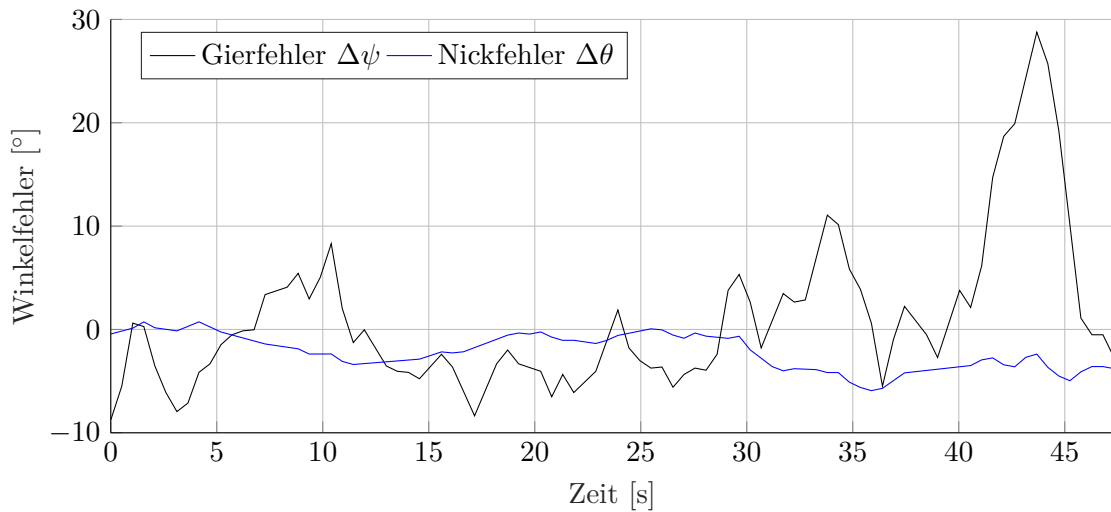


Abbildung 6.28: Flugversuch D1 „Statisches Ziel“: Gierwinkelfehler $\Delta\psi$ und Nickwinkelfehler $\Delta\theta$ aus Videoauswertung

ausgewertet. Dazu wird alle 0,5s die Position des Zielobjekts (GNSS-Antenne) im Videobild markiert, siehe blaue Kreuze in Abbildung 6.27. Zu erkennen ist, dass das Zielobjekt nicht ganz mittig (schwarzes Kreuz), sondern mit einem Offset nach oben gefilmt wird. Außerdem tritt ein Ausreißer nach links auf.

Mit (3.1), (3.2) können die Koordinaten (u, v) der Bildpixel in die Winkel α_u und α_v umgerechnet werden. Aus ihnen können der Gierwinkelfehler $\Delta\psi$ des Quadropters und der Nickwinkelfehler $\Delta\theta$ des Gimbals mit $\Delta\psi = -\alpha_v$ und $\Delta\theta = -\alpha_u$ bestimmt werden, die in Abbildung 6.28 über der Zeit aufgetragen sind. Der Gierwinkelfehler $\Delta\psi$ entspricht dem γ aus (5.131). Der Median des Gierwinkelfehlers beträgt $-0,7^\circ$, der des Nickwinkelfehlers $-2,2^\circ$. Das Maximum des Gierwinkelfehlers von 29° tritt bei $t = 44$ s zu einem Zeitpunkt auf, zu dem die Beschränkung der Gierwinkel-Trajektorie $y_{4,s}$ auf $\dot{\psi}_{\text{lim}}$ aktiv ist, vgl. Abbildung 6.24. Die Abweichung ist auch als Differenz zwischen der blauen und der schwarzen Kurve im unteren Graphen von Abbildung 6.24 zu erkennen.

Flugversuch D2 „Objektverfolgung Gerade“ Das Zielobjekt bewegt sich auf einer geraden Strecke und der Operator verändert das Azimut-Kommando α_{cmd} . Das Abstandskommando δ_{cmd} bleibt konstant und die Höhe wird gehalten.

Die gerade Trajektorie des Zielobjekts ist in Rot auf der Karte in Abbildung 6.29 dargestellt. Die Geschwindigkeit des Zielobjekts beträgt nach dem Beschleunigen jeweils 1,8 m/s. Das Abstandskommando δ_r ist durchgehend ca. 3 m. Der kommandierte Azimut α_{cmd} wird mehrfach durch den Operator verändert, vgl. Abbildung 6.30. Wenn die Änderungsrate von α_{cmd} über $\pm\dot{\alpha}_{\text{lim},k}$ (5.144) liegt, ändert sich α_r wie im vorhergehenden Flugversuch gemäß (5.143) maximal mit einer Rate von $\pm\dot{\alpha}_{\text{lim},k}$.

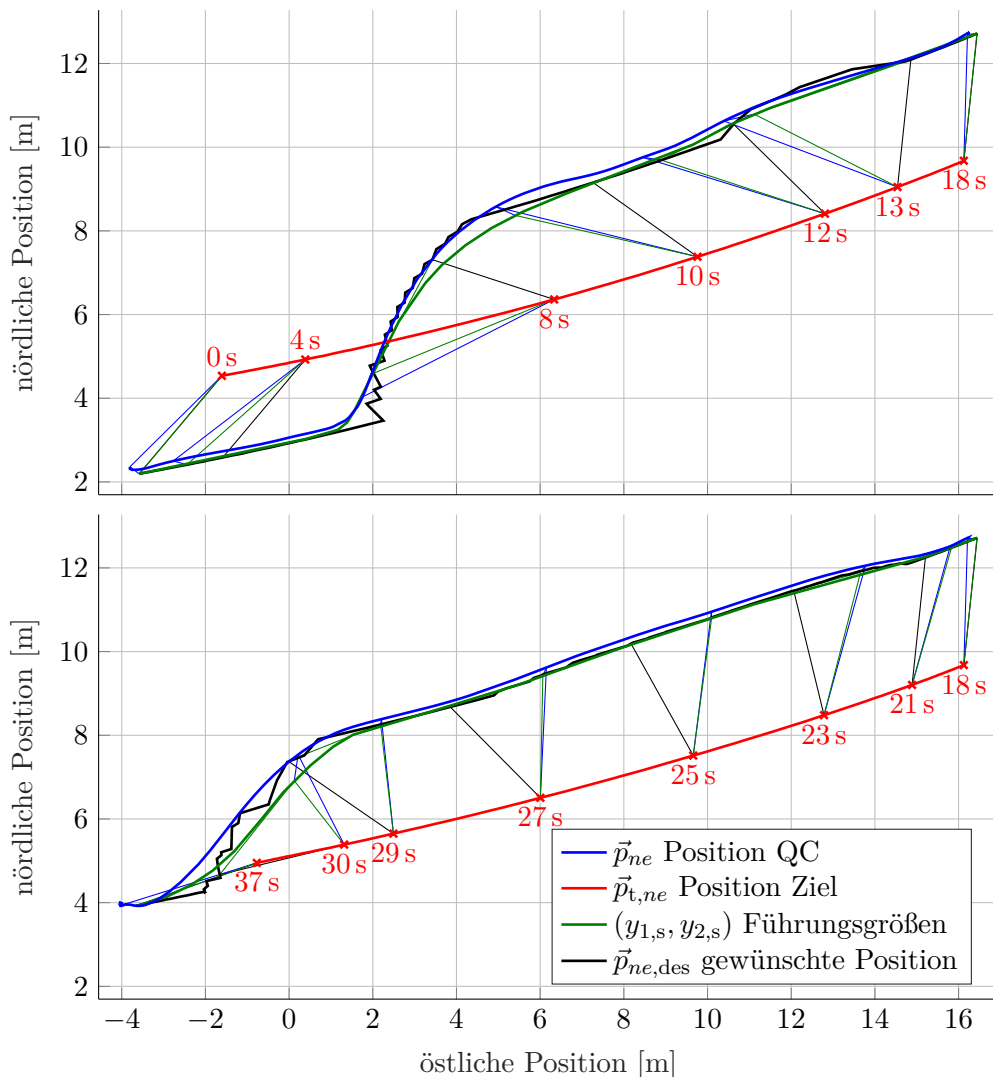


Abbildung 6.29: Karte des Flugversuchs D2 „Objektverfolgung Gerade“, oben: Hinfahrt, unten: Rückfahrt

Der zeitliche Verlauf der gewünschten Position $\vec{p}_{ne,des}$ (5.145) des Quadropters, die sich aus δ_r und α_r ergibt, ist in den oberen beiden Graphen von Abbildung 6.31 als schwarze Kurve dargestellt. Die Position des Zielobjekts $\vec{p}_{t,ne}$ ist jeweils als rote Kurve eingezeichnet. Die Ziel-Einheit sendet diese Position mit einer Update-Rate von 0,1 Hz. Die kleinen Stufen in den roten Kurven sind darauf zurückzuführen, dass einzelne Nachrichten mit der Zielposition bei der Übertragung im XBee-Mesh-Netzwerk⁶ verloren gehen. Zwischen zwei Updates wird die Position des Zielobjekts nicht prädiziert, sondern jeweils das letzte Update verwendet. Dies führt auch zu den Zacken in den schwarzen Kurven für $\vec{p}_{ne,des}$, die auch auf der Karte in Abbildung 6.29 vor allem in Bereichen, in denen sich α_r schnell

⁶XBee-Mesh-Netzwerk: siehe Abschnitt 3.4

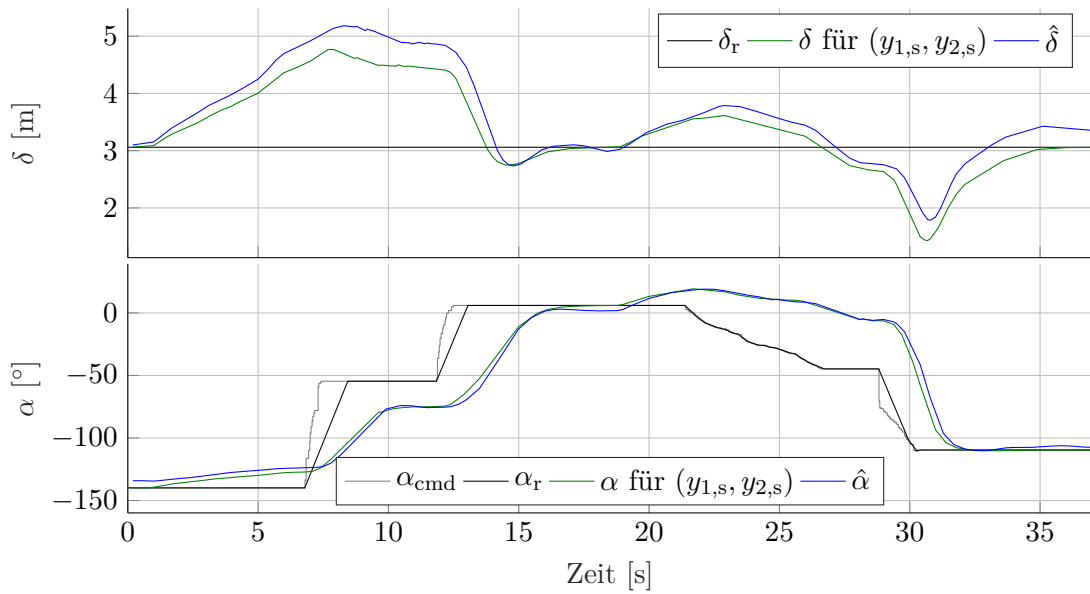


Abbildung 6.30: Flugversuch D2 „Objektverfolgung Gerade“: Abstand zum Zielobjekt δ und Azimut α

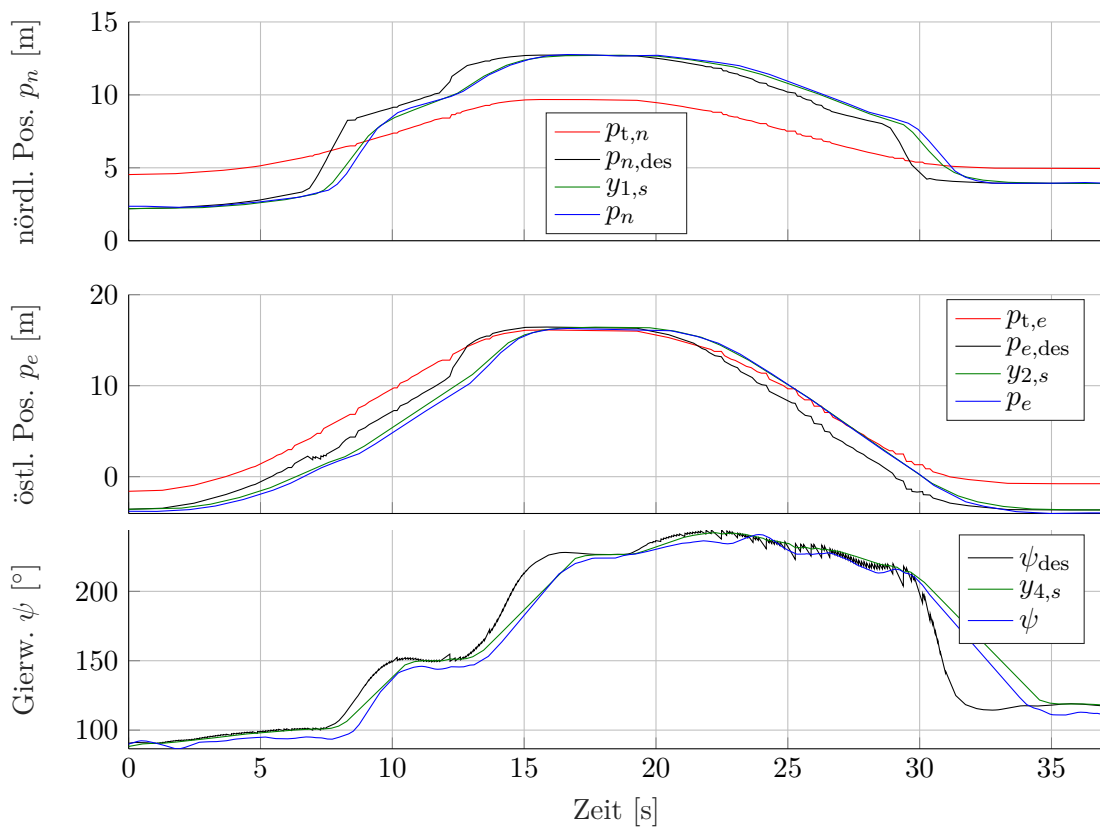


Abbildung 6.31: Flugversuch D2 „Objektverfolgung Gerade“: nördliche und östliche Position, Gierwinkel

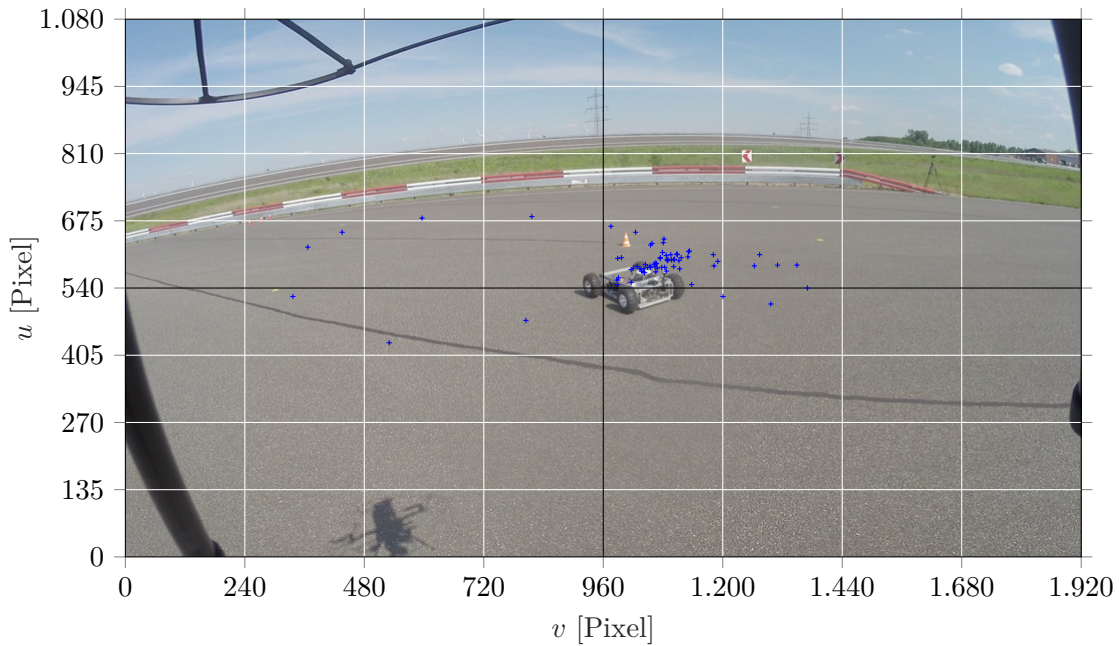


Abbildung 6.32: Flugversuch D2 „Objektverfolgung Gerade“: Positionen des Zielobjekts im Video (blaue Kreuze), Videobild zum Zeitpunkt $t = 13$ sec

ändert, sichtbar sind.

Die auf Basis von $\vec{p}_{ne,des}$ mit (5.146), (5.134) generierten Positions-Trajektorien $(y_{1,s}, y_{2,s})^T$ sind in den Abbildungen 6.29 und 6.31 wieder als grüne Kurven eingezeichnet. Wie im vorhergehenden Flugversuch folgen die Führungsgrößen $y_{1,s}$ und $y_{2,s}$ für die flachheitsbasierte Regelung wie beabsichtigt den gewünschten Positionen $\vec{p}_{ne,des}$ ohne Überschwingen und unter Beachtung der Beschränkungen.

Die blauen Kurven in den Abbildungen 6.29 und 6.31 zeigen wieder die Positionen \hat{p}_{ne} des Quadropters. Auch hier folgen die Ist-Positionen den Führungsgrößen in Grün ohne größere Abweichungen. Die Positionsfehler sind maximal während der Transienten von α_r und betragen dort maximal 75 cm in nördlicher und östlicher Richtung. In Abbildung 6.30 sind die aus den Führungsgrößen $y_{1,s}$ und $y_{2,s}$ und den Ist-Positionen \hat{p}_{ne} resultierenden Abstände δ und Azimute α in Grün bzw. Blau eingezeichnet.

Der mit (5.132) für die Objektverfolgung berechnete, gewünschte Quadropter-Gierwinkel $\psi_{des,k}$ ist im unteren Graphen von Abbildung 6.31 zu finden. Auch hier ist in Grün die daraus berechnete Gierwinkel-Trajektorie $y_{4,s}$ mit einer beschränkten Änderungsrate von $\dot{\psi}_{lim} = 20^\circ/s$ dargestellt. Der Quadropter folgt dieser Führungsgröße mit einem maximalen Gierwinkelfehler von unter 12° .

Die Positionen des Zielobjekts im Videobild sind in Abbildung 6.32 wieder als blaue Kreuze dargestellt. Der mit (3.1), (3.2) aus den Bildpunkten (u, v) berechnete Quadropter-

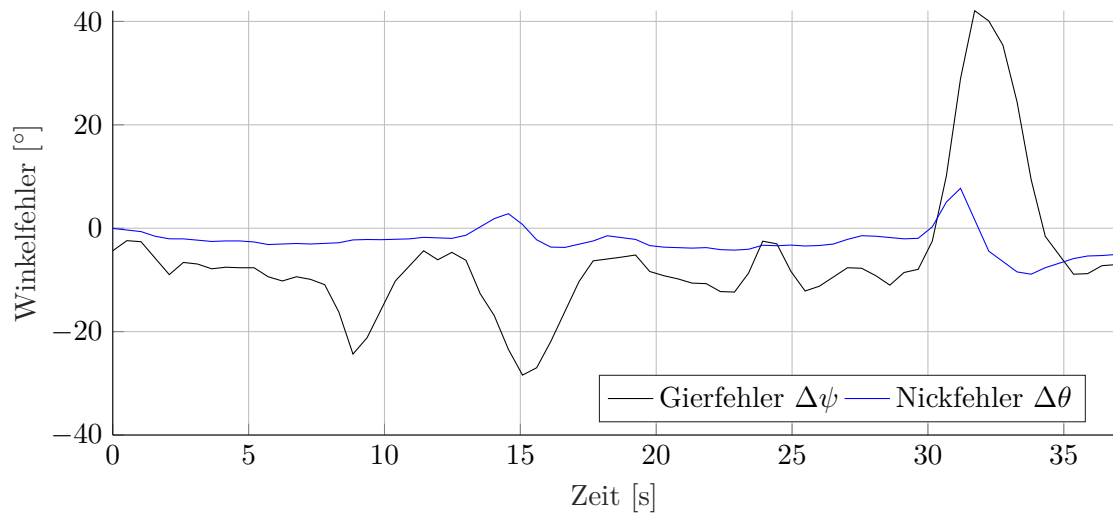


Abbildung 6.33: Flugversuch D2 „Objektverfolgung Gerade“: Gierwinkelfehler $\Delta\psi$ und Nickwinkelfehler $\Delta\theta$ aus Videoauswertung

Gierwinkelfehler $\Delta\psi = -\alpha_v$ und der Gimbal-Nickwinkelfehler $\Delta\theta = -\alpha_u$ sind in Abbildung 6.33 zu sehen. Bei diesem Flugversuch beträgt der Median des Gierwinkelfehlers $-8,2^\circ$ und der des Nickwinkelfehlers $-2,5^\circ$, vgl. Anhäufung der Positionen des Zielobjekts oben rechts der Mitte in Abbildung 6.32.

Die Extrema des Gierwinkelfehlers treten wie beim Flugversuch D1 in den Bereichen auf, in denen die Gierwinkel-Trajektorie $y_{4,s}$ durch die Beschränkung auf $\dot{\psi}_{\text{lim}}$ stärker von ψ_{des} abweicht, siehe Abbildung 6.31. Durch die Drehratenbeschränkung schafft es der Quadrocopter in diesen Situationen nur mit Verzögerung, sich so zum Zielobjekt zu drehen, dass es mittig im Bild ist. Auch beim größten Gierwinkelfehler von 42° (Ausreißer im Videobild nach links) bleibt das Zielobjekt aufgrund des Kamera-Sichtfeldes von $\pm 62^\circ$ (vgl. Abbildung 3.8) im Bild. Es ist zu erwarten, dass sich die Gierwinkelfehler durch eine Anhebung von $\dot{\psi}_{\text{lim}}$ reduzieren lassen.

Flugversuch D3 „Objektverfolgung Halbkreis“ Das Zielobjekt folgt einer in etwa halbkreisförmigen Trajektorie mit einer Geschwindigkeit von 1,2 m/s. Diese Trajektorie $\vec{p}_{t,ne}$ ist auf der Karte in Abbildung 6.34 in Rot dargestellt. Die Abstands- und Azimut-Kommandos, δ_r und α_r , sind konstant bei 3,2 m und -150° , siehe Abbildung 6.35. Der Quadrocopter hält seine Höhe.

Die Trajektorie $\vec{p}_{ne,des}$ ist hier aufgrund der konstanten Kommandos δ_r und α_r gleich der in etwa nach Südwesten (210° Nord) verschobenen Zielobjekttrajektorie $\vec{p}_{t,ne}$, vgl. rote und schwarze Kurven in Abbildung 6.34 und in den beiden oberen Graphen von Abbildung 6.36. Die Führungsgrößen $(y_{1,s}, y_{2,s})^T$ sind in diesen Abbildungen wieder in Grün und die Ist-Positionen \hat{p}_{ne} in Blau dargestellt.

Die aus den Führungsgrößen $y_{1,s}$ und $y_{2,s}$ und den Ist-Positionen \hat{p}_{ne} resultierenden Abstände δ und Azimute α sind in Abbildung 6.35 als grüne bzw. blaue Kurven eingezeichnet. Im unteren Graphen von Abbildung 6.36 sind der zur Objektverfolgung gewünschte

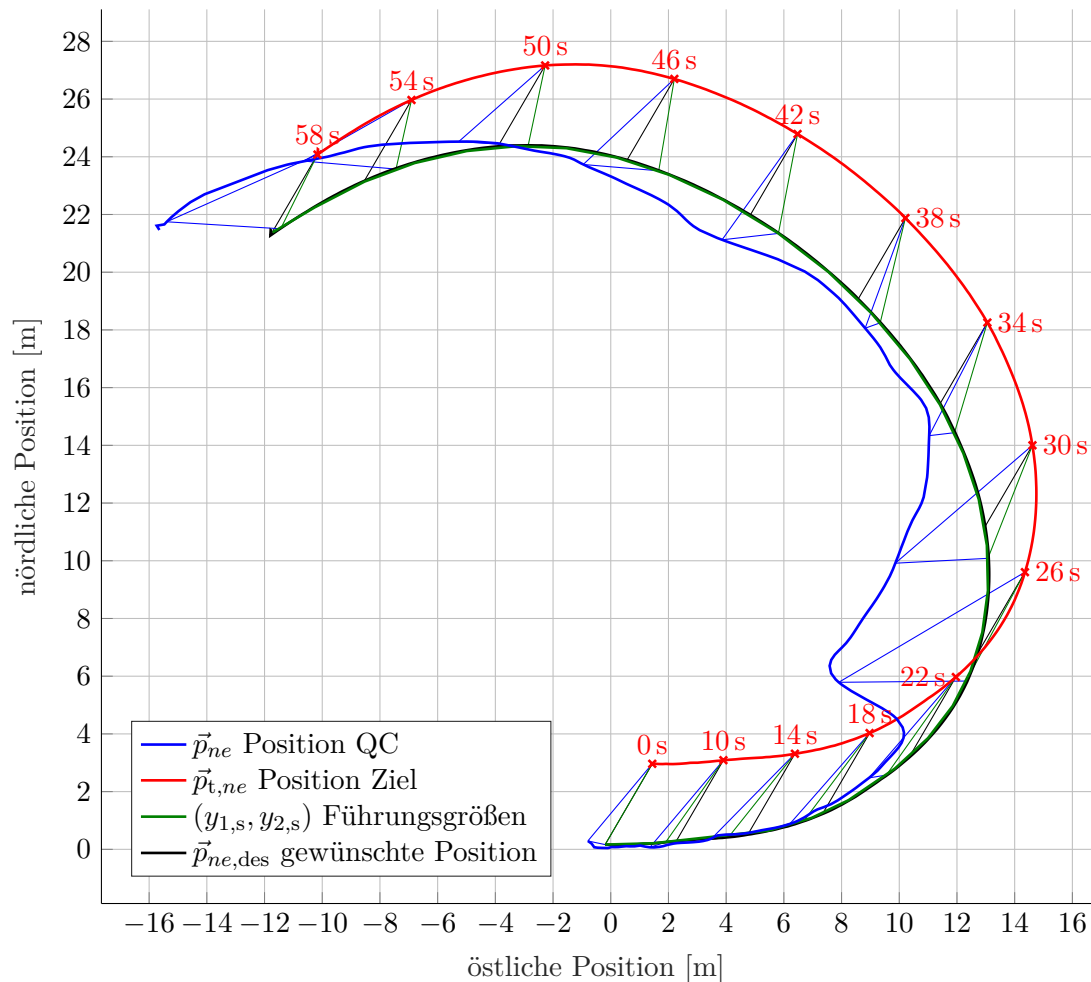


Abbildung 6.34: Karte des Flugversuchs D3 „Objektverfolgung Halbkreis“

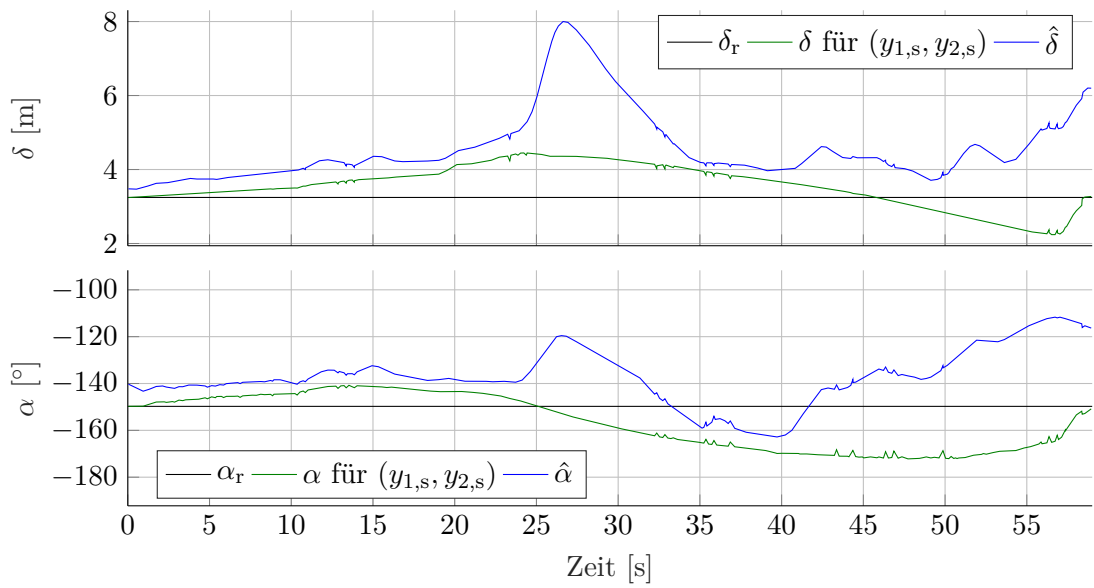


Abbildung 6.35: Flugversuch D3 „Objektverfolgung Halbkreis“: Abstand zum Zielobjekt δ und Azimut α

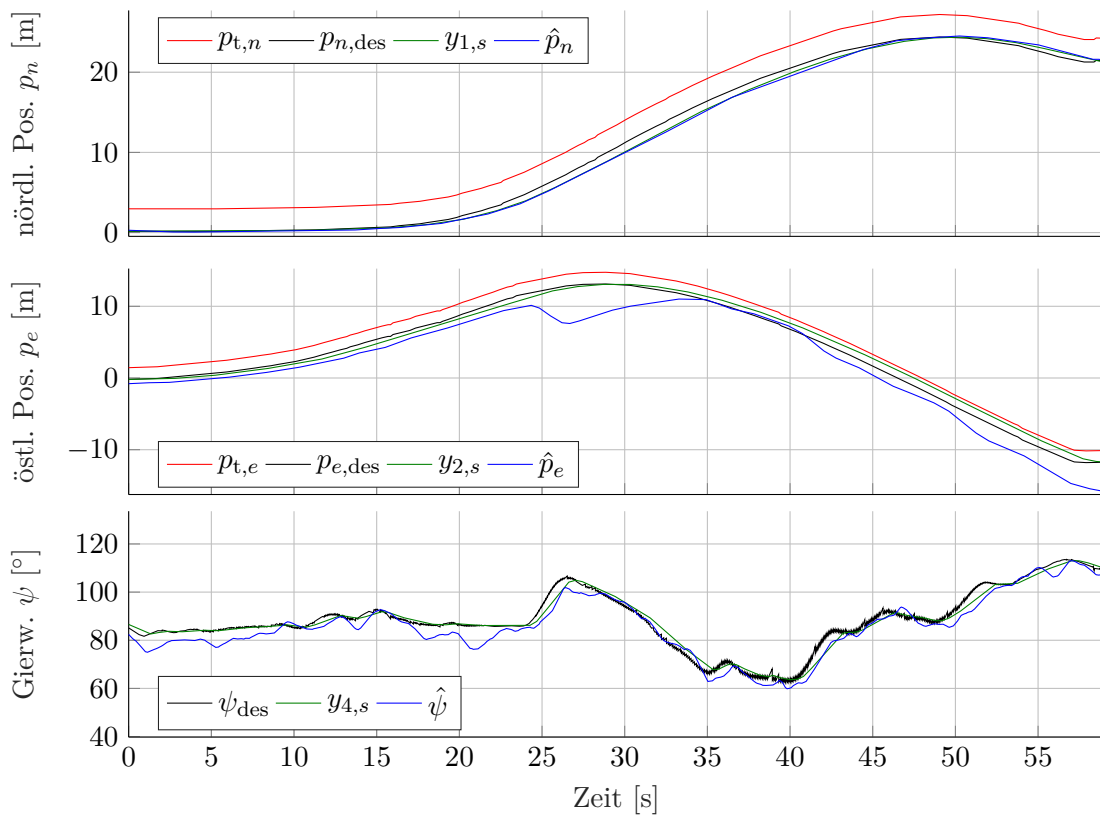


Abbildung 6.36: Flugversuch D3 „Objektverfolgung Halbkreis“: nördliche und östliche Position, Gierwinkel

Quadropter-Gierwinkel $\psi_{\text{des},k}$ in Schwarz, die Gierwinkel-Trajektorie $y_{4,s}$ in Grün und der Gierwinkel $\hat{\psi}$ des Quadropters in Blau zu finden.

Zur Analyse der Positionsabweichungen sind die Größen der Reglerkaskaden (5.113) der Trajektorienfolgeregelung für den Ost- und den Nord-Kanal in Abbildungen 6.37 und 6.38 dargestellt. Während der Positionsfehler in nördlicher Richtung maximal 30 cm beträgt (vgl. Abbildung 6.38, Graph a), weicht die östliche Position \vec{p}_e stärker von der Führungsgröße $y_{2,s}$ ab. Zum Zeitpunkt $t = 27$ s beträgt der Positionsfehler in östlicher Richtung kurzzeitig 5 m, zum Ende der Trajektorie 4 m und sonst ca. 50 cm (vgl. Abbildung 6.37, Graph a).

Da die größten Fehler in östlicher Richtung auftreten, wird Abbildung 6.37 näher betrachtet. In Graph a ist der Positionsfehler in östlicher Richtung $p_{e,s} - \hat{p}_e = y_{2,s} - \hat{y}_2 = h_{2,1}/L_{2,0}$ gemäß (5.113b) eingezeichnet. $h_{2,0,\text{sat}}$ ist hier null, da der Integrierer (5.113c) für den Nord- und Ost-Kanal nicht aktiv ist ($L_{1,I} = L_{2,I} = 0$). Im Bereich von $t = 24$ s bis 34 s und ab $t = 41$ s übersteigt $h_{2,1}$ die Beschränkung $h_{2,1,\text{lim}}$ (Tabelle C.4). Die Hilfsstellgröße $h_{2,1,\text{sat}}$ (5.113d), die in Graph b als schwarze Kurve erscheint, ist daher in diesen Bereichen auf $h_{2,1,\text{lim}}$ beschränkt.

Graph b stellt die Geschwindigkeiten dar, die multipliziert mit dem Übertragungsbeiwert $L_{2,1}$ (Tabelle C.3) in die Hilfsstellgröße $h_{2,2}$ eingehen. Die Ist-Geschwindigkeit \hat{y}_2 stimmt überwiegend gut mit der vorgesteuerten Geschwindigkeit $\dot{y}_{2,s}$ überein. Auffällig ist der negative Geschwindigkeitspeak bei $t = 25,5$ s, der zu dem Positionsfehlerpeak in westlicher Richtung führt. Zu der Differenz $\dot{y}_{2,s} - \hat{y}_2$ addiert sich die Hilfsstellgröße $h_{2,1,\text{sat}}$ des Positionsreglers und führt daher zu einer Hilfsstellgröße $h_{2,2}$, die mehrmals die Beschränkung $h_{2,2,\text{lim}}$ überschreitet. Dies führt zu einer in diesen Bereichen beschränkten Hilfsstellgröße $h_{2,2,\text{sat}}$ (5.113d), siehe schwarze Kurve in Graph c.

Die höheren Ableitungen der Führungsgrößen (5.135) sind in den Flugversuchen D1, D2 und D3 wie schon im Flugversuch C als Vorsichtsmaßnahme ab der zweiten Ableitung ($\ddot{y}_{1,s}, \ddot{y}_{2,s}$) = $\ddot{\vec{p}}_{ne,s,k}^T$ zu null gesetzt. Dies ist auch an den blauen Kurven in den Graphen c, d und e zu sehen.

Der Beschleunigungsregler ist in Graph c dargestellt. Die gemäß (5.116) geschätzte Ist-Beschleunigung $\hat{\ddot{y}}_2$ bewegt sich erwartungsgemäß im Mittel um null. Die größten Peaks von bis zu 2 m/s^2 führen zu dem oben genannten Geschwindigkeitspeak bei $t = 25,5$ s. Die Hilfsstellgröße $h_{2,3}$ des Beschleunigungsreglers wird von der Hilfsstellgröße $h_{2,2,\text{sat}}$ des Geschwindigkeitsreglers dominiert. Diese führt dazu, dass $h_{2,3}$ während des ganzen Flugversuchs nahe an der Beschränkung $h_{2,3,\text{lim}}$ ist und diese oft überschreitet. In diesen Fällen wird $h_{2,3,\text{sat}}$ abgeschnitten und es kann nicht mehr Beschleunigung bzw. Schräglage aufgebaut werden, die notwendig wäre, um durch eine Erhöhung der Geschwindigkeit den Positionsfehler zu reduzieren. Im Nord-Kanal wird die Beschränkung $h_{1,3,\text{lim}}$ nicht erreicht (siehe Abbildung 6.38, Graph c). Dies passt zu der Beobachtung, dass während des Flugversuches ein mäßiger Wind aus östlicher Richtung herrschte.

Die Hilfsstellgröße $h_{2,3,\text{sat}}$ des Beschleunigungsreglers geht in den Ruckregler ein, siehe schwarze Kurve in Graph d von Abbildung 6.37. Daraus ergibt sich im Wesentlichen die

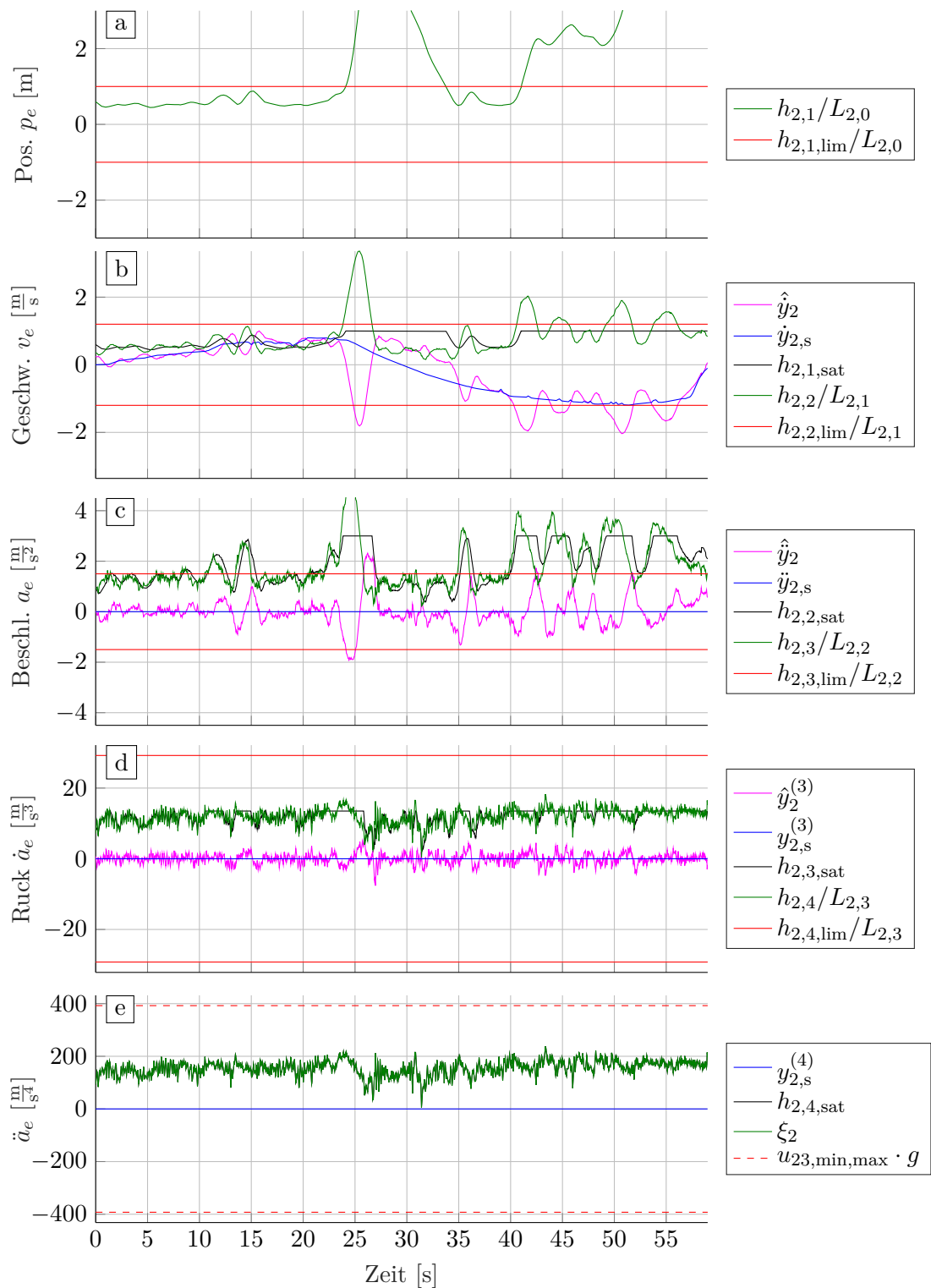


Abbildung 6.37: Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Ost-Kanal

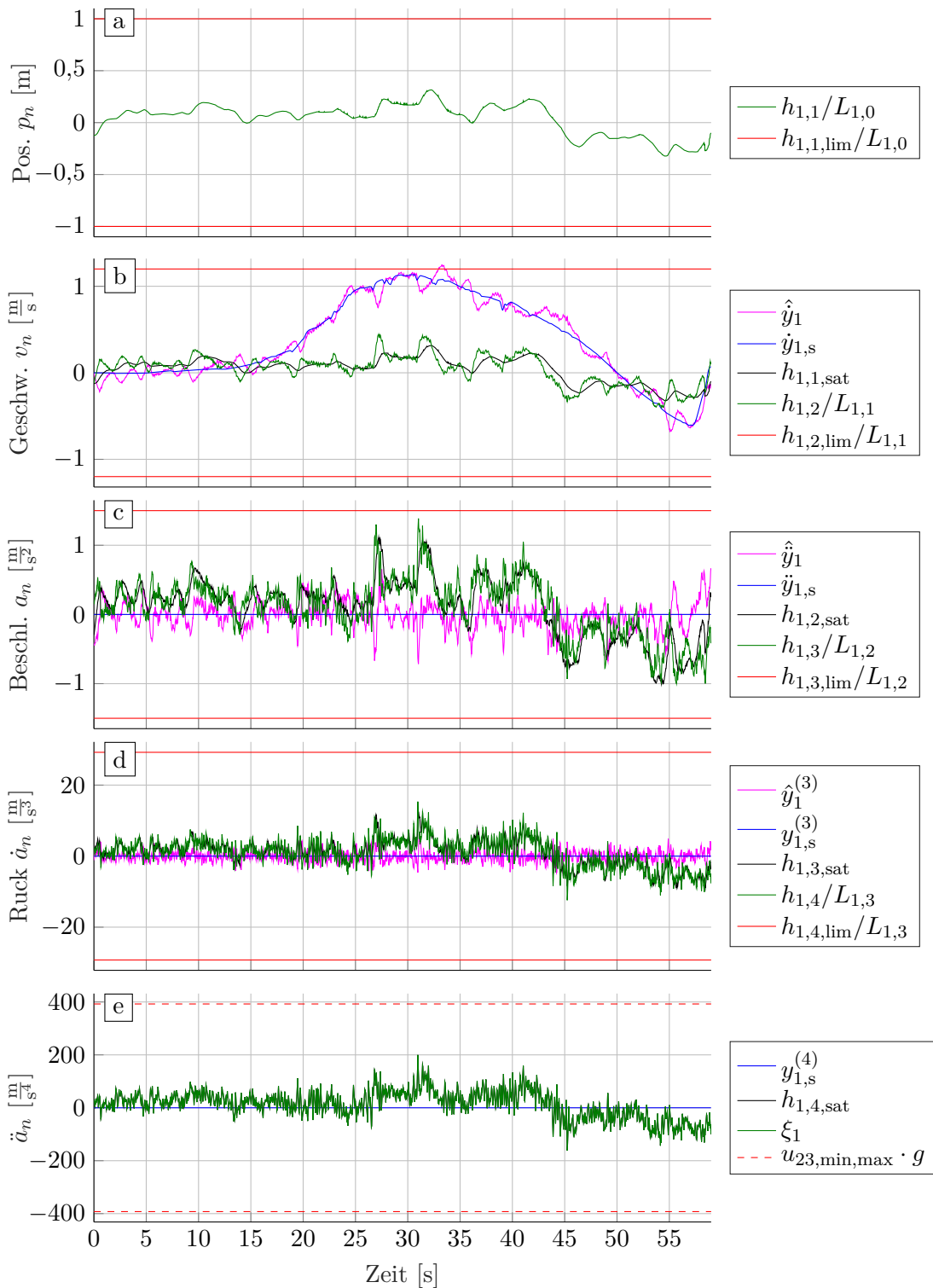


Abbildung 6.38: Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Nord-Kanal

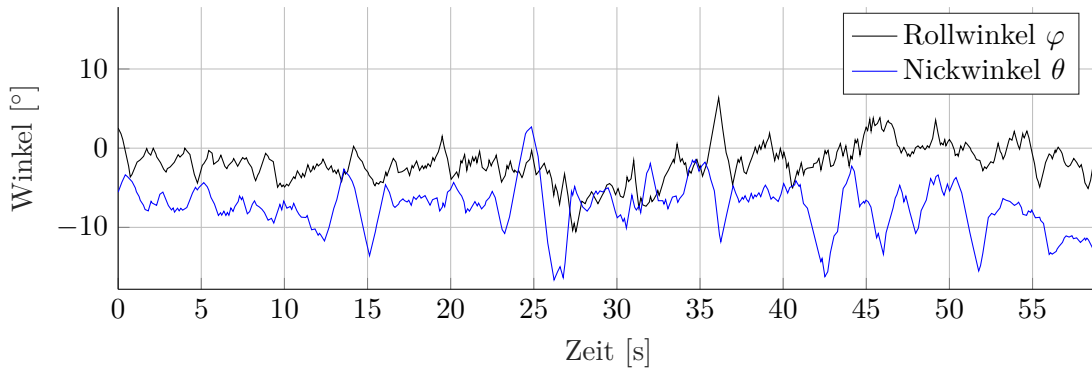


Abbildung 6.39: Flugversuch D3 „Objektverfolgung Halbkreis“: Roll- und Nickwinkel

Hilfsstellgröße $h_{2,4}$ des Ruckreglers. $h_{2,4}$ übersteigt $h_{2,4,\text{lim}}$ nicht, daher ist $h_{2,4,\text{sat}} = h_{2,4}$. Die Ruckschätzung $\hat{y}_2^{(3)} = \hat{p}_e^{(3)}$ gemäß (5.117) ist in Magenta dargestellt.

In Graph e von Abbildung 6.37 sind schließlich die Elemente von (5.113a) gegeben. Da $y_{2,s}^{(4)}$ zu null gesetzt ist (siehe oben), ist die Stellgröße ξ_2 gleich $h_{2,4,\text{sat}}$. Als Referenz sind die für den Schwebeflug umgerechneten, oberen und unteren Beschränkungen für u_2 und u_3 (5.107) als $u_{2,\text{max}} \cdot g = u_{3,\text{max}} \cdot g$ und $u_{2,\text{min}} \cdot g = u_{3,\text{min}} \cdot g$ eingezeichnet (vgl. (5.103)).

In Abbildung 6.39 sind die Roll- und Nickwinkel des Quadropters dargestellt. In diesem Flugversuch beträgt der mittlere Nickwinkel $\bar{\theta} = -7^\circ$ und der mittlere Rollwinkel $\bar{\varphi} = -2^\circ$. Zusammen mit einem Gierwinkel von etwa 90° (Abbildung 6.36 unten) lässt sich also ablesen, dass der Quadropter während des gesamten Manövers durch den Nickwinkel nach Osten und durch den Rollwinkel leicht nach Norden geneigt ist. Dies passt zu den Hilfsstellgrößen $h_{2,3}$ und $h_{1,3}$ der Beschleunigungsregler, siehe oben.

Nun wird betrachtet, wie der Störgrößen-Beobachter aus Abschnitt 5.1.5 die Störung durch den Wind schätzt. Dazu sind in Abbildung 6.40 die Komponenten der Beschleunigungsschätzung (5.116) und in Abbildung 6.41 die vom Beobachter geschätzte spezifische Störkraft \vec{a}_s (5.31) in nördlicher, östlicher und vertikaler Richtung dargestellt. Die in der Regelung benutzten Beschleunigungsschätzungen $(\hat{y}_1, \hat{y}_2, \hat{y}_3)^T$ gemäß (5.116) sind in Abbildung 6.40 als Kurven in Magenta visualisiert. Sie sind gleich den magentafarbenen Kurven in Graph c der Abbildungen 6.37 und 6.38. In Grau ist die geschätzte Beschleunigung \hat{a}_{lb} (5.14) dargestellt. Deren gleitender Mittelwert $(\bar{a}_n, \bar{a}_e, \bar{a}_d)^T$ über 10 Zeitschritte, also 0,1 s, in Schwarz stimmt gut mit $(\hat{y}_1, \hat{y}_2, \hat{y}_3)^T$ in Magenta überein. Der Modell-Anteil

$$\hat{a}^{*,l} = (\hat{a}_n^*, \hat{a}_e^*, \hat{a}_d^*)^T = g \vec{l}_d^l - u_1 \hat{b}_z^l \quad (6.5)$$

von (5.116) ist in Grün gegeben. In ihn geht die Schwerkraftbeschleunigung g , der spezifische Schub u_1 und die geschätzte Lage als \hat{b}_z^l ein. Am Modell-Anteil ist zu erkennen, dass ohne Berücksichtigung der Störung \vec{a}_s eine Beschleunigung \hat{a}^* vor allem in östlicher und leicht in nördlicher Richtung hätte stattfinden müssen.

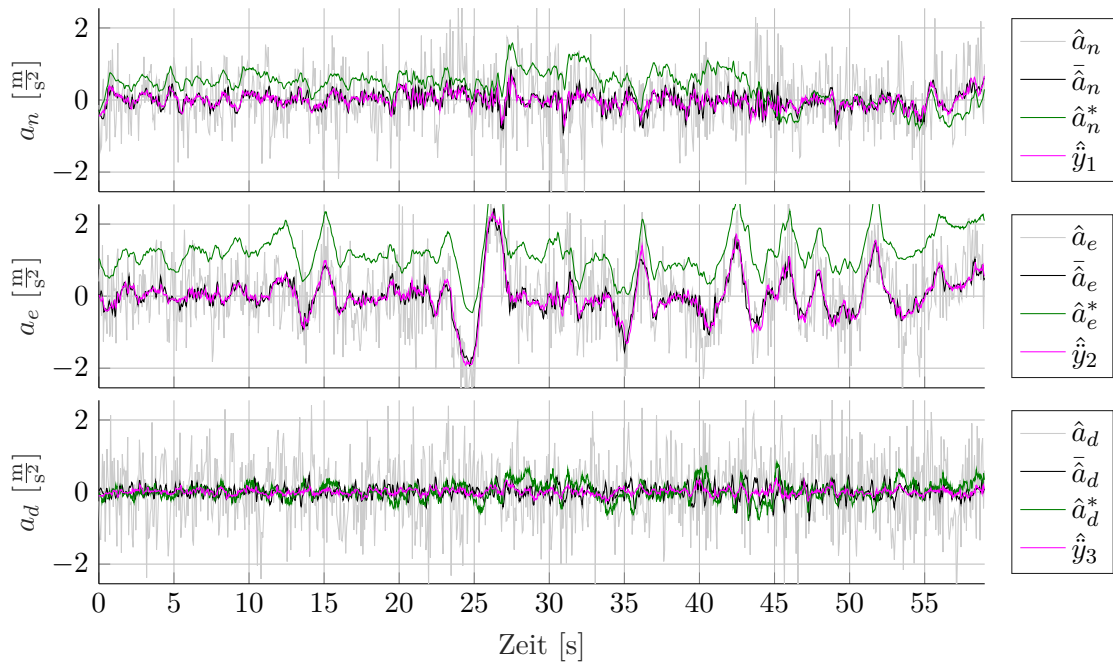


Abbildung 6.40: Flugversuch D3 „Objektverfolgung Halbkreis“: Beschleunigungsschätzung

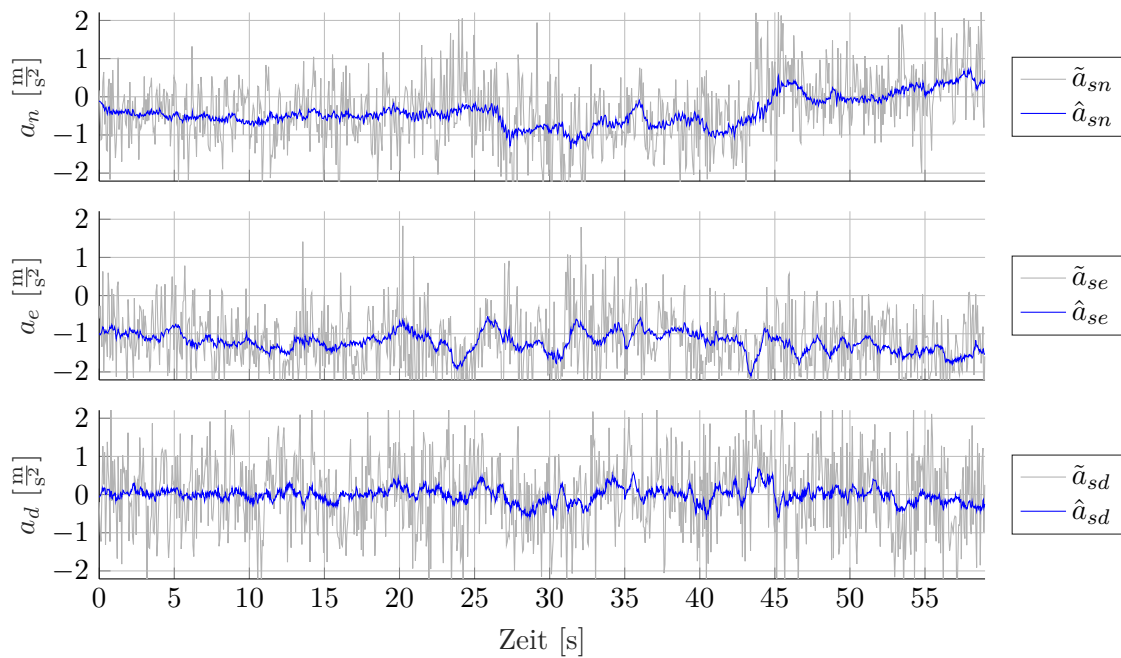


Abbildung 6.41: Flugversuch D3 „Objektverfolgung Halbkreis“: Schätzung spezifische Störkraft

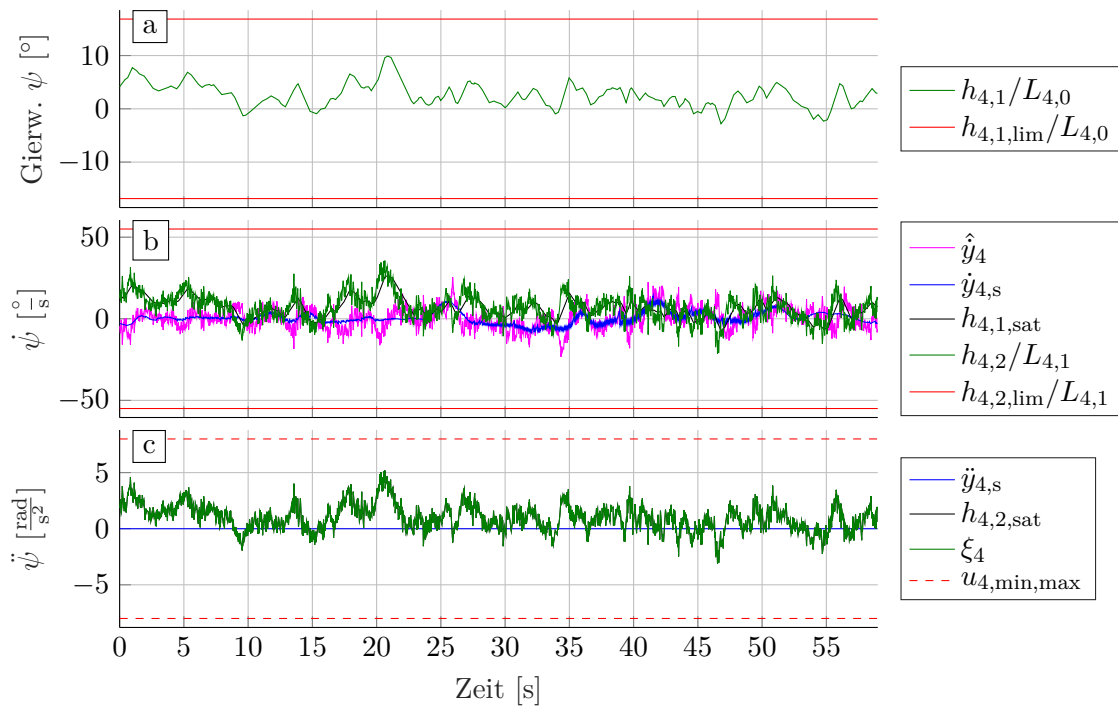


Abbildung 6.42: Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Gier-Kanal

Die Differenz von grüner und magentafarbener Kurve ist die vom Störgrößen-Beobachter geschätzte spezifische Störkraft \hat{a}_s , die in Abbildung 6.41 als blaue Kurven dargestellt ist. Zusätzlich ist in Abbildung 6.41 in Grau der Messwert \tilde{a}_s (5.49) gegeben. An \tilde{a}_s in Blau wird die Störung durch den Wind offensichtlich: Die spezifische Störkraft \hat{a}_{se} in westlicher Richtung beträgt im Mittel über 1 m/s^2 und in Spitzen bis 2 m/s^2 . Dies entspricht gemäß den Abschätzungen aus Abschnitt 5.1.2.3 einer relativen Windgeschwindigkeit von 7 m/s im Mittel und bis 10 m/s in den Spitzen. Wie schon in Graph c von Abbildung 6.37 erkennbar war, ist die Störung durch den Wind aus westlicher Richtung zu groß, als dass sie unter der Beschränkung $h_{2,3,lim}$ kompensiert werden könnte, um die Positionsfehler zu vermeiden. Zugleich zeigt sich aber auch, dass die kaskadierten Regler mit Beschränkungen geeignet sind, auch bei größeren Störungen die inneren Regelkreise für \dot{a} und \ddot{a} bzw. für Drehrate und Drehbeschleunigung stabil zu halten.

Nach dem Ost- und Nord-Kanal werden nun die verbleibenden zwei Kanäle, der Gier- und der Vertikal-Kanal, betrachtet. Die Größen der Reglerkaskade (5.113) für den Gier-Kanal sind in Abbildung 6.42 dargestellt. Der Gierwinkelfehler $\psi_s - \hat{\psi} = y_{4,s} - \hat{y}_4 = h_{4,1}/L_{4,0}$ gemäß (5.113b) ist in Graph a zu finden. Der Gierwinkelfehler beträgt maximal 10° . Wie beim Nord- und Ost-Kanal ist der Integrierer (5.113c) nicht aktiv ($L_{4,I} = 0$). Die Beschränkung $h_{4,1,lim}$ wird zu keinem Zeitpunkt überschritten, daher ist $h_{4,1,sat} = h_{4,1}$. Die Hilfsstellgröße $h_{4,1,sat}$ erscheint als schwarze Kurve in Graph b.

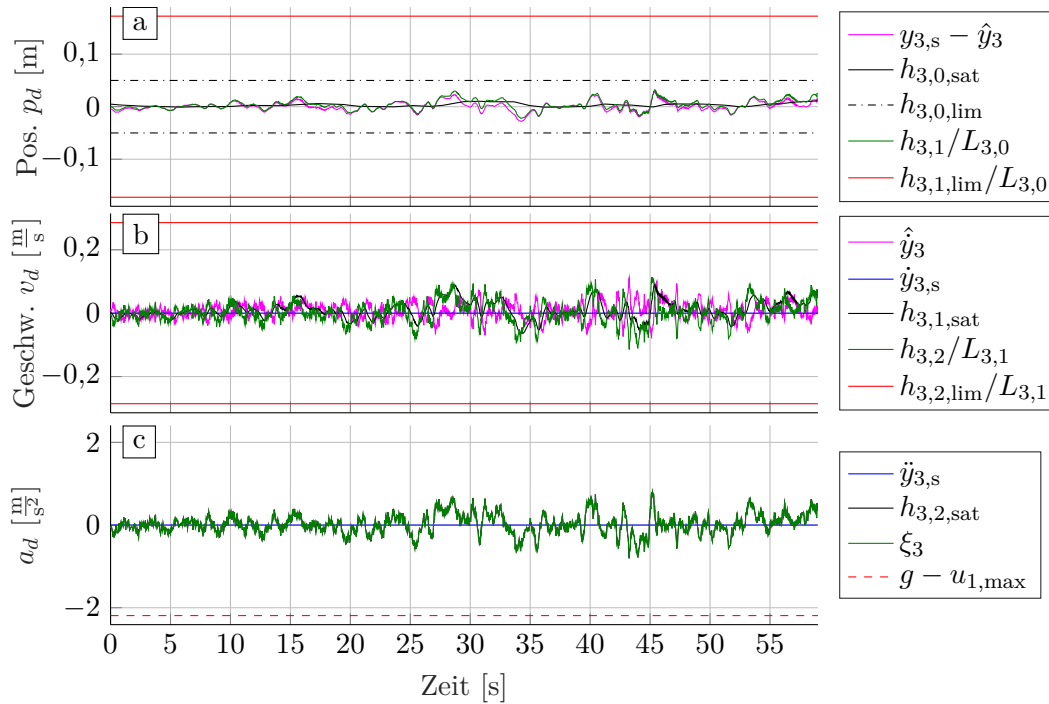


Abbildung 6.43: Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Vertikal-Kanal

Neben $h_{4,1,sat}$ stellt Graph b die weiteren Ein- und Ausgangsgrößen des Gierrgeschwindigkeitsregler dar. Die Ist-Gierrate $\hat{y}_4 = \hat{\psi}$ ist als magentafarbene und die Führungsgröße $\dot{y}_{4,s}$ als blaue Kurve gegeben. Aus $\dot{y}_{4,s}$, \hat{y}_4 und $h_{4,1,sat}$ ergibt sich mit dem Übertragungsbeiwert $L_{4,1}$ gemäß (5.113b) die Hilfsstellgröße $h_{4,2}$ in Grün. Auch $h_{4,2}$ überschreitet die Beschränkung $h_{4,2,lim}$ nicht, damit ist $h_{4,2,sat} = h_{4,2}$.

In Graph c ist die Berechnung der Stellgröße ξ_4 gemäß (5.113a) dargestellt. Wegen (5.138b) ist $\ddot{y}_{4,s} = 0$ und damit gilt $\xi_4 = h_{4,2,sat}$. Die Beschränkungen $u_{4,min}$ und $u_{4,max}$ (5.107) für u_4 können in Beschränkungen für ξ_4 umgerechnet werden. Im Schwebeflug gilt $u_4 = \xi_4$ gemäß (5.104), daher sind $u_{4,min}$ und $u_{4,max}$ als Beschränkungen von ξ_4 im Schwebeflug in Graph c eingezeichnet.

Die Reglerkaskade des Vertikal-Kanals ist schließlich in Abbildung 6.43 dargestellt. Der Höhenfehler $y_{3,s} - \hat{y}_3$ in Magenta in Graph a beträgt im Mittel 0 und maximal 3 cm. Beim Vertikal-Kanal ist der Integrierer aktiviert, dessen Ausgang $h_{3,0}$ in Schwarz erreicht die Beschränkung $h_{3,0,lim}$ nicht. Die Hilfsstellgröße $h_{3,1}$ in Grün liegt weit entfernt von ihrer Beschränkung $h_{3,1,lim}$ in Rot. Graph b zeigt den Vertikal-Geschwindigkeitsregler mit den üblichen Größen. Die Hilfsstellgröße $h_{3,2}$ ist ebenfalls weit entfernt von der Beschränkung $h_{3,2,lim}$. In Graph c ist schließlich ξ_3 zu sehen: Wegen $\ddot{y}_{3,s} = 0$ gemäß (5.138a) ist $\xi_3 = h_{3,2,sat}$. Im Schwebeflug gilt gemäß (5.101) $u_1 = g - \xi_3$. Daher ist die Beschränkung $u_{1,max}$ (5.107) hier für den Schwebeflug umgerechnet als $g - u_{1,max}$ eingezeichnet. Die Stellgröße

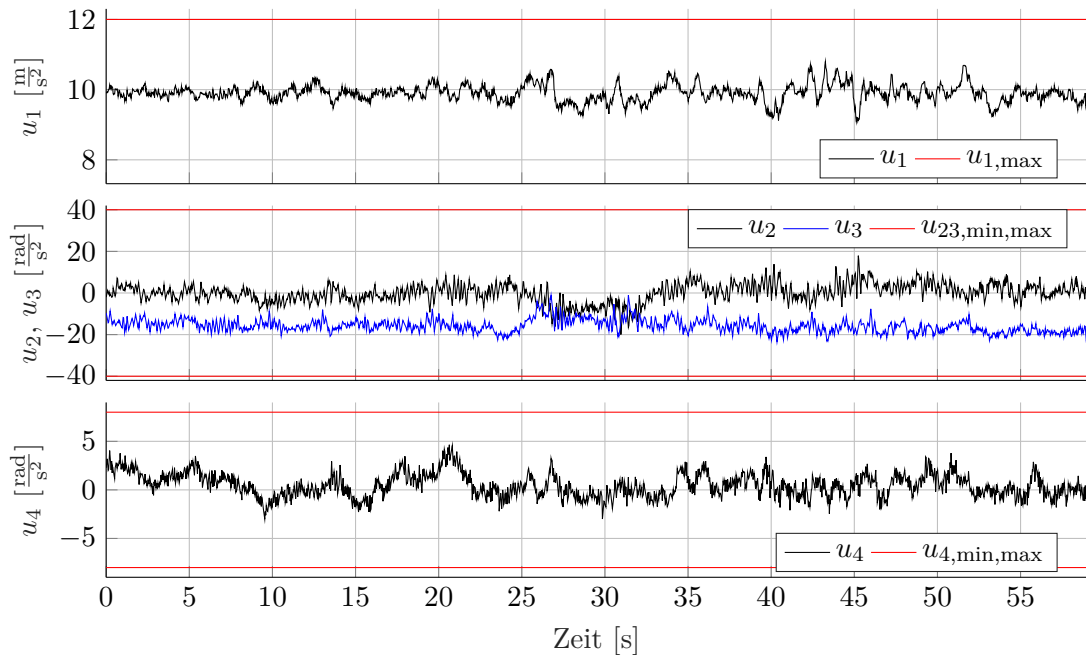


Abbildung 6.44: Flugversuch D3 „Objektverfolgung Halbkreis“: Stellgrößen u_1 , u_2 , u_3 , u_4

ξ_3 erreicht diese Beschränkung nicht. Die Berechnung von $\dot{\xi}_3$ und $\ddot{\xi}_3$ gemäß (5.114) soll hier nicht dargestellt werden.

Aus $\vec{\xi}$ und dem Systemzustand berechnet die flachheitsbasierte Zustandsrückführung gemäß (5.106) die Systemstellgrößen u_1 , u_2 , u_3 und u_4 . Diese sind in Abbildung 6.44 dargestellt. Die Stellgrößen bleiben jeweils innerhalb der Beschränkungen $u_{i,\min}$, $u_{i,\max}$ (5.107).

Die Videoauswertung der Quadropterkamera ist in Abbildung (6.45) zu sehen. Bereits auf den ersten Blick ist zu erkennen, dass die Abweichungen der Zielobjektpositionen von der Bildmitte deutlich kleiner sind, als in den Flugversuchen D1 und D2. Der daraus gemäß (3.1), (3.2) resultierende Quadropterk-Gierwinkelfehler $\Delta\psi = -\alpha_v$ und Gimbal-Nickwinkelfehler $\Delta\theta = -\alpha_u$ sind in Abbildung 6.46 dargestellt. Der Median des Gierwinkelfehlers beträgt hier -4° und der des Nickwinkelfehlers $-0,6^\circ$. In diesem Flugversuch erreicht die Gierwinkel-Trajektorie $y_{4,s}$ nicht die Beschränkung $\dot{\psi}_{\lim}$, sodass größere Gierwinkelfehler wie in den Flugversuchen D1 und D2 ausbleiben. Die betragsmäßig maximalen Gier- und Nickwinkelfehler sind hier 12° und 2° . Das Zielobjekt bleibt also während des gesamten Flugversuchs wie gewünscht sehr gut in der Mitte des Videobildes.

Fazit Objektverfolgung Mit den Flugversuchen D1, D2 und D3 wurde in verschiedenen Szenarien nachgewiesen, dass das Gesamtsystem in der Lage ist, ein bewegtes Zielobjekt mit Quadropterkopter und Kamera automatisch zu verfolgen.

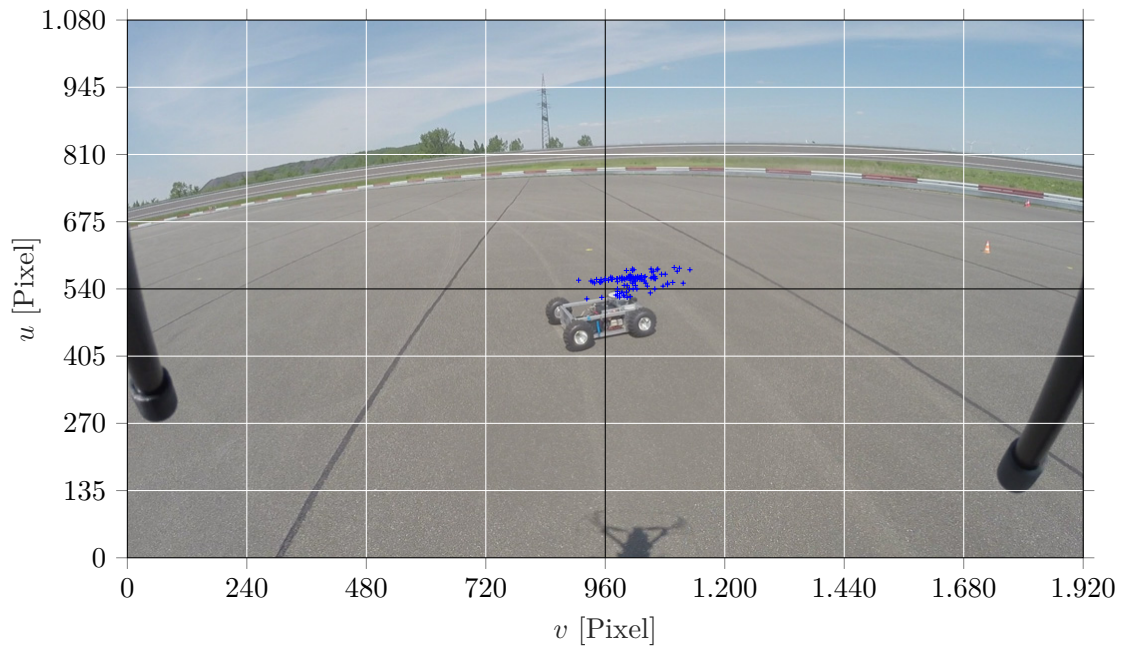


Abbildung 6.45: Flugversuch D3 „Objektverfolgung Halbkreis“: Positionen des Zielobjekts im Video (blaue Kreuze), Videobild zum Zeitpunkt $t = 45$ sec

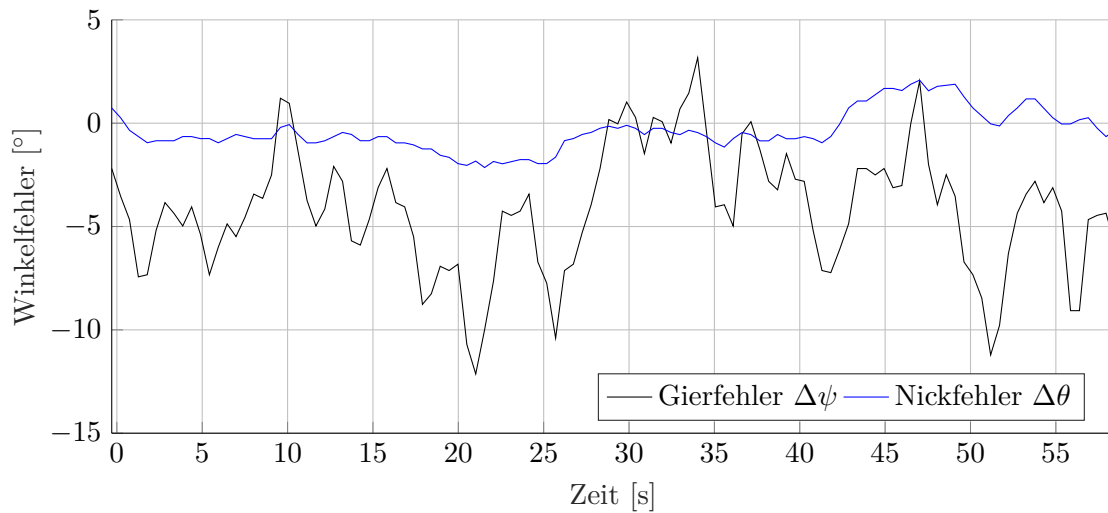


Abbildung 6.46: Flugversuch D3 „Objektverfolgung Halbkreis“: Gierwinkelfehler $\Delta\psi$ und Nickwinkelfehler $\Delta\theta$ aus Videoauswertung

Die Trajektoriengenerierung mit Kaskadenregelung und die flachheitsbasierte Regelung positionieren den Quadrokopter wie gewünscht gemäß den Kommandos des Operators relativ zum Zielobjekt. Auch bei größeren Störungen durch Wind, die der Quadrokopter bei den eingestellten Beschränkungen nicht vollständig durch Schrägstellung kompensieren kann, erlaubt die Struktur der Trajektorienfolgeregelung, dass die Lage des Quadrokopfers durchgehend stabil bleibt und die Positionsabweichung reduziert wird, sobald der Wind wieder abgenommen hat. Die Quadrokopter-Kamera wird – solange es die Dynamik des Zielobjektes erlaubt – durch die Gierregelung des Quadrokopfers und die Gimbalsteuerung stets so ausgerichtet, dass das Zielobjekt mittig im Kamerabild zu sehen ist.

Die experimentelle Validierung des Gesamtsystems kann also als erfolgreich bewertet werden. Die entworfene Automatisierung des Quadrokopfers bestehend aus Sensorfusion, flachheitsbasierter Regelung und Trajektoriengenerierung ist somit sehr gut zur Objektverfolgung geeignet.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Ziel dieser Arbeit war die Automatisierung eines Kamera-Quadropters zur Verfolgung von bewegten Objekten für Filmzwecke. Wie die Auswertung der Flugversuche zeigt, wurde dieses Ziel erreicht: Der Quadropter verfolgt das Zielobjekt automatisch gemäß den Vorgaben des Operators und richtet die Quadropter-Kamera so auf das Zielobjekt aus, dass dieses mittig im Videobild zu sehen ist. In weiteren Betriebsmodi kann der Operator u. a. Position und Orientierung der Quadropter-Kamera per 6-Achs-Joystick steuern sowie Trajektorien kommandieren.

Zur Automatisierung des Quadropters wurden eine Sensorfusion, eine flachheitsbasierte Regelung sowie Trajektoriengenerierungsmethoden für die verschiedenen Betriebsmodi entworfen und implementiert. Für die Erprobung im Flugversuch wurde ein Quadropter-Prototyp aufgebaut und mit den für die Anwendung benötigten Komponenten erweitert. Neben dem Flugsystem wurden eine Benutzerschnittstelle für den Operator, eine Ziel-Einheit, die die Position des Zielobjekts sendet, sowie eine Bodenstation realisiert, die über ein Funknetzwerk kommunizieren.

Für die flachheitsbasierte Regelung wurde in der Arbeit der Nachweis geführt, dass die Quadropter-Dynamik flach ist. Bei dem vollständig dargestellten Beweis wurden auch Störgrößen berücksichtigt, die durch Wind auf den Quadropter wirken. Für die flachen Ausgänge des Systems wurde eine quasi-statische Zustandsrückführung hergeleitet, die die Systemdynamik eingangs-/zustandslinearisiert und die Systemkanäle entkoppelt. Die überlagerte Trajektorienfolgeregelung muss damit lediglich lineare Systemkanäle stabilisieren und regeln.

Die flachheitsbasierte Regelung zeigt in den Flugversuchen ein sehr gutes Trajektorienfolgeverhalten. So folgt der Quadropter horizontalen und vertikalen Trajektorien sowie Gier-Trajektorien ohne Überschwingen und quasi ohne Verzögerungen. Durch die implementierte Schubkalibrierung, die es der Regelung erlaubt, den Quadropter genau zu beschleunigen, werden Abweichungen von den Trajektorien weiter reduziert. Wie in den Versuchen demonstriert wurde, kann die flachheitsbasierte Regelung translatorische Trajektorien und Gier-Trajektorien zudem ohne gegenseitige Beeinflussung gleichzeitig einregeln. Die Objektverfolgung profitiert unmittelbar von diesem präzisen Trajektorienfolgeverhalten.

Die auf den Quadropter wirkenden Störkräfte durch Wind werden vom Störgrößen-Beobachter geschätzt und von der Regelung durch Schrägstellung des Quadropters

kompensiert. Bei der in den Flugversuchen parametrisierten maximalen Schrägstellung des Quadropters konnte Wind bis zu einer Windgeschwindigkeit von etwa 7 m/s ausgeglichen werden. Auch bei Windböen, die der Quadropter nicht durch Schrägstellung vollständig kompensieren kann, zeigt das System das gewünschte Verhalten, indem es die Lage des Quadropters durchgehend stabilisiert und die Positionsabweichung reduziert, sobald die Windverhältnisse es zulassen.

Für die Betriebsmodi des Systems wurden drei verschiedene Methoden entwickelt, um die Solltrajektorien und deren Ableitungen für die flachheitbasierte Regelung zu generieren. Für den Flug zwischen statischen Wegpunkten, das Starten bzw. Landen etc. nutzt die erste Methode Spline-Trajektorien, die onboard zeitlich und räumlich skaliert werden. Die zweite Trajektoriengenerierungsmethode verwendet – in Anlehnung an die Trajektorienfolgeregelung – eine Kaskadenregelung zur Generierung der Trajektorien. Für die Trajektorien können Beschränkungen bezüglich Geschwindigkeit, Beschleunigung und ggf. Ruck vorgegeben werden. Diese Methode wird für den Flug per Joystick und zur Verfolgung von Objekten genutzt. Bei der dritten Methode kommt schließlich eine modellprädiktive Regelung zur Trajektoriengenerierung zum Einsatz, die die Integration von verschiedenen Optimierungskriterien erlaubt.

Wie die Erprobung der Sensorfusion gezeigt hat, ist das entworfene Erweiterte Kalman-Filter in der Lage, auf Basis der Signale von IMU und GNSS-RTK-Empfänger den Zustand des Quadropters für die Regelung ausreichend präzise zu schätzen. So wurden die Positionen im Flugversuch auf etwa 3 cm, die Geschwindigkeiten auf 3 cm/s und die Lagewinkel auf unter 1° genau geschätzt. Die Schätzwerte der Inertialsensor-Bias konvergieren wie erwartet. Die Sensorfusion kann außerdem kurzzeitige Verschlechterungen der GNSS-Messungen z. B. beim Wechsel des RTK-Modus überbrücken.

Neben dem GNSS-RTK-Empfänger wurde auch ein experimenteller GPS- und Galileo-Empfänger im Galileo-Testgebiet automotiveGATE erprobt, um der wachsenden Bedeutung des europäischen Satellitennavigationssystems Galileo Rechnung zu tragen. Es konnte gezeigt werden, dass der experimentelle Empfänger auch auf Basis der Galileo-Signale im Testgebiet seine Position berechnen kann und sich für den Betrieb auf dem Quadropter eignet.

Bei den Flugversuchen stellte es sich als essenziell heraus, dass der Operator mit dem System interagieren kann, um es in seinen verschiedenen Betriebsmodi nutzen zu können. Daher wurde als Benutzerschnittstelle auch ein Handbediengerät für den Operator – das „ControlPad“ – entwickelt. Über die graphische Darstellung in der iOS-App des ControlPads sowie durch die Ton- und Sprachausgaben kann der Operator den Systemstatus schnell erfassen. Der Touchscreen erlaubt die intuitive Kommandierung des Systems. So können z. B. die Betriebsmodi für jeden Steuerungskanal schnell gewechselt sowie die Parameter der Betriebsmodi angepasst werden.

Das vielfältige Potenzial eines solchen Handbediengeräts zeigt auch eine weitere implementierte, intuitive Steuerungsart, bei der zur Kommandierung des Systems eine virtuelle 3-D-Ansicht entweder durch Touch-Gesten oder durch Schwenken und Neigen des

ControlPads bewegt wird. Auf diese Weise kann man z. B. die Quadrokopter-Kamera so schwenken, als ob man sie selbst in der Hand halten würde.

Auch der im ControlPad integrierte 6-Achs-Joystick hat sich als nützlich herausgestellt, um ausgewählte Steuerungskanäle per Hand steuern zu können. Bei der Erprobung wurden die Geschwindigkeitskommandos des Joysticks wie gewünscht vom Quadrokopter umgesetzt. Die Trajektoriengenerierung berücksichtigt dabei eingestellte Beschleunigungs- und Ruck-Beschränkungen für eine gleichmäßige Bewegung der Kamera.

Die automatisierte Objektverfolgung wurde anhand eines Testfahrzeugs erprobt, dessen GNSS-Position die Ziel-Einheit per Xbee-Funk-Netzwerk an den Quadrokopter überträgt. Der Operator kommandiert über die App auf dem ControlPad Abstand und Winkel zum Zielobjekt. Die Regelung steuert den Quadrokopter und das Kamera-Gimbal so an, dass das Zielobjekt wie gewünscht in der Mitte des Kamerabildes gehalten wird. Der Ansatz, das Zielobjekt nur auf Basis der von der Ziel-Einheit gesendeten Position zu verfolgen, ohne eine Bildverarbeitung zu nutzen, war also zielführend.

Die in dieser Arbeit entwickelte Automatisierung übernimmt die Aufgaben eines Piloten und unterstützt den Operator bei der Kameraführung. Dies erleichtert die Erstellung von dynamischen Videoaufnahmen von bewegten Objekten mit dem Kamera-Quadrokopter. Der entwickelte Prototyp hat das Potenzial, als Basis für eine Produktentwicklung zu dienen. Mögliche Ansätze für eine Weiterentwicklung werden im anschließenden Ausblick gegeben. Mit dem großen Bedarf an attraktiven Luftbildaufnahmen in vielen Bereichen hätte ein solches Produkt sicherlich gute Vermarktungschancen.

7.2 Ausblick

Im Rahmen dieser Arbeit wurden zum einen eine Sensorfusion und eine flachheitsbasierte Regelung als Basis einer Multikopter-Automatisierung entwickelt. Zum anderen wurde eine spezifische Anwendung – die Verfolgung von bewegten Objekten für Filmzwecke – entworfen und prototypisch realisiert. Sowohl die Basiselemente der Automatisierung als auch die umgesetzte Anwendung bieten verschiedene Ansatzpunkte für weiterführende Arbeiten, auf die im Folgenden eingegangen werden soll.

Die entworfene Trajektorienfolgeregelung könnte in weiteren Flugversuchen so abgestimmt werden, dass durch eine größere Schrägstellung des Quadrokopters auch höhere Windgeschwindigkeiten über 7 m/s ausgeglichen werden können. Zudem ist eine Erprobung bei höheren Quadrokoptergeschwindigkeiten über 4 m/s und höheren Gierraten über 20°/s sinnvoll.

Für die Sensorfusion könnte in nachfolgenden Arbeiten die Robustheit des Erweiterten Kalman-Filters bei länger anhaltenden Verschlechterungen und Ausfällen der GNSS-Messungen untersucht werden. Ggf. ist es sinnvoll, weitere Messgrößen einzuführen, um die Stabilität der Lageschätzung unter diesen Umständen sicherzustellen. Außerdem könnte der experimentelle GPS- und Galileo-Empfänger zusammen mit der Sensorfusion getestet werden.

Für den Aufbau eines Prototyps hat sich der flexibel erweiterbare Pelican-Quadrokooper als gut geeignete Basis erwiesen. Durch die zusätzlich installierten Sensoren, das Computer-Modul etc. war der Pelican mit dem resultierenden Gesamtgewicht jedoch an seiner Leistungsgrenze. Für eine Weiterentwicklung und einen Produktiveinsatz sollte ein Multikooper mit einer höheren Traglast verwendet werden, sodass z. B. auch eine größere Kamera mit Zoomobjektiv genutzt werden könnte. Einige Multikooper bringen auch Sensorik zur Kollisionsvermeidung mit, was je nach Einsatzszenario sinnvoll sein kann.

Die im Rahmen dieser Arbeit implementierten und erprobten Betriebsmodi sind exemplarische Assistenzfunktionen, die den Operator bei den Dreharbeiten mit einem Kamera-Multikooper unterstützen. Durch eine Erweiterung der Trajektoriengenerierungsmethoden können leicht weitere Assistenzfunktionen ergänzt werden. Die flachheitsbasierte Regelung des Quadrokoopers und die Sensorfusion können dabei unverändert bleiben. Eine mögliche Erweiterung wäre z. B., die automatische Objektverfolgung mit vorgegebenen Trajektorien so zu kombinieren, dass der Quadrokooper wie eine Seilkamera einen vorgegebenen Pfad nicht verlässt und die Quadrokoopergeschwindigkeit entlang des Pfades der Bewegung des Zielobjekts automatisch angepasst wird. In diesem Zusammenhang wäre es sinnvoll, dass vor Ort ein Pfad eingelernt wird, z. B. indem er ggf. um Hindernisse herum zunächst per Joystick langsam abgeflogen wird. Für die vielfältigen Anwendungsszenarien eines solchen Systems sind zahlreiche Betriebsmodi denkbar, die auch beliebig miteinander kombiniert werden können.

Die Ergebnisse dieser Arbeit lassen sich auch auf andere Multikooper-Anwendungen übertragen. Insbesondere können die entwickelte flachheitsbasierte Regelung und die Sensorfusion, unabhängig von der hier realisierten Verwendung, als allgemeine Grundlage für andere Multikooper-Anwendungen genutzt werden, die vergleichbare Anforderungen bezüglich Flugregelungsgenauigkeit und Automatisierung haben.

Anhang

A Stochastik

In diesem Abschnitt werden die Grundlagen der Stochastik gemäß [28, 84] dargestellt, die für die Kalman-Filter-Grundlagen in Abschnitt 2.3 und für die Sensorfusion in Kapitel 4 benötigt werden.

Zufallsvariablen

Eine Zufallsvariable ist eine Funktion, die den möglichen Ergebnissen eines Zufallsexperiments die einzelnen Funktionswerte – die sog. Realisationen – zuordnet.

Die Wahrscheinlichkeit P , mit der eine Realisation einer skalaren **Zufallsvariablen** v kleiner als eine Schranke ξ ist, wird durch die **Verteilungsfunktion** $F_v(\xi)$ ausgedrückt:

$$F_v(\xi) = P(v \leq \xi) \quad (\text{A.1})$$

Die Ableitung der Verteilungsfunktion ist die **Wahrscheinlichkeitsdichte**:

$$p_v(\xi) = \frac{dF_v(\xi)}{d\xi}(\xi) \quad (\text{A.2})$$

Der **Erwartungswert** oder **Mittelwert** μ_v der Zufallsvariablen v ist

$$\mu_v = E[v] = \int_{-\infty}^{\infty} \xi p_v(\xi) d\xi \quad . \quad (\text{A.3})$$

Die **Varianz** σ_v^2 beschreibt die Streuung der Realisationen von v um den Mittelwert μ_v :

$$\sigma_v^2 = Var(v) = E[(v - \mu_v)^2] = \int_{-\infty}^{\infty} (\xi - \mu_v)^2 p_v(\xi) d\xi \quad (\text{A.4})$$

Die Wurzel der Varianz ist die **Standardabweichung** σ_v der Zufallsvariablen v .

Die **Verbundverteilungsfunktion** für zwei Zufallsvariablen v und w ist

$$F_{vw}(\xi, \eta) = P(v \leq \xi \wedge w \leq \eta) \quad (\text{A.5})$$

und die **Verbundwahrscheinlichkeitsdichte**:

$$p_{vw}(\xi, \eta) = \frac{d^2 F_{vw}(\xi, \eta)}{d\xi d\eta}(\xi, \eta) \quad . \quad (\text{A.6})$$

Die zwei Zufallsvariablen v und w sind **statistisch unabhängig**, wenn gilt:

$$p_{vw}(\xi, \eta) = p_v(\xi)p_w(\eta) \quad (\text{A.7})$$

Die gegenseitige Abhängigkeit kann als Kovarianz und Korrelationskoeffizient beschrieben werden. Die **Kovarianz** der Zufallsvariablen v und w ist definiert als

$$Cov(v, w) = E[(v - \mu_v)(w - \mu_w)] \quad (\text{A.8})$$

und ihr **Korrelationskoeffizient** lautet

$$\rho_{vw} = E[vw] = \frac{Cov(v, w)}{\sigma_v \sigma_w} \quad (\text{A.9})$$

Zufallsvektoren

Ein **Zufallsvektor** \vec{v} besteht aus den skalaren Zufallsvariablen $v_1 \dots v_n$:

$$\vec{v} = (v_1, \dots, v_n)^T \quad (\text{A.10})$$

Sein Erwartungswert lautet im Allgemeinen:

$$\vec{\mu}_v = E[\vec{v}] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \vec{\xi} p_{\vec{v}}(\vec{\xi}) d\xi_1 \dots d\xi_n \quad (\text{A.11})$$

Wenn die einzelnen Zufallsvariablen $v_1 \dots v_n$ statistisch unabhängig sind, vereinfacht sich der Erwartungswert des Zufallsvektors zu $\vec{\mu}_v = (E[v_1], \dots, E[v_n])^T$.

Die **Korrelationsmatrix** $Cor(\vec{v}, \vec{w})$ ist die Übertragung des Korrelationskoeffizienten (A.9) auf Zufallsvektoren:

$$Cor(\vec{v}, \vec{w}) = E[\vec{v}\vec{w}^T] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \vec{\xi} \vec{\eta}^T p_{\vec{v}\vec{w}}(\vec{\xi}, \vec{\eta}) d\vec{\xi} d\vec{\eta} \quad (\text{A.12})$$

Die **Kovarianzmatrix** $Cov(\vec{v}, \vec{w})$ ist die mehrdimensionale Kovarianz (A.8):

$$\mathbf{P}_{vw} = Cov(\vec{v}, \vec{w}) = E[(\vec{v} - \vec{\mu}_v)(\vec{w} - \vec{\mu}_w)^T] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\vec{\xi} - \vec{\mu}_v)(\vec{\eta} - \vec{\mu}_w)^T p_{\vec{v}\vec{w}}(\vec{\xi}, \vec{\eta}) d\vec{\xi} d\vec{\eta} \quad (\text{A.13})$$

Die **Auto-Kovarianzmatrix** \mathbf{P}_v beschreibt die Abhängigkeit des Zufallsvektors \vec{v} von sich selbst:

$$\mathbf{P}_v = Cov(\vec{v}, \vec{v}) = E[(\vec{v} - \vec{\mu}_v)(\vec{v} - \vec{\mu}_v)^T] \quad (\text{A.14})$$

Auf ihrer Hauptdiagonalen stehen die Varianzen $Var(v_i)$ der einzelnen skalaren Zufallsvariablen v_i des Zufallsvektors \vec{v} .

Gemäß dem **Verschiebungssatz** gilt für die Kovarianzmatrix (A.13) außerdem:

$$\mathbf{P}_{vw} = E[\vec{v}\vec{w}^T] - \vec{\mu}_v\vec{\mu}_w^T \quad (\text{A.15})$$

Damit sind für mittelwertfreie Zufallsvariablen Korrelationsmatrix und Kovarianzmatrix gleich:

$$\text{Cor}(\vec{v},\vec{w}) = E[\vec{v}\vec{w}^T] = \text{Cov}(\vec{v},\vec{w}) \quad \text{wenn } \vec{\mu}_v = \vec{0}, \vec{\mu}_w = \vec{0} \quad (\text{A.16})$$

Ein **normalverteilter Zufallsvektor** \vec{v} hat die mehrdimensionale Dichtefunktion

$$p_{\vec{v}}(\vec{\xi}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{P}_v)}} e^{-\frac{1}{2}(\vec{\xi}-\vec{\mu}_v)^T \mathbf{P}_v^{-1}(\vec{\xi}-\vec{\mu}_v)} \quad (\text{A.17})$$

mit dem Erwartungswert $\vec{\mu}_v$ und der Kovarianzmatrix \mathbf{P}_v .

Stochastische Prozesse

Ein **stochastischer Prozess** bzw. **Zufallsprozess** $\vec{v}(t)$ erweitert Zufallsvariablen \vec{v} um die Beschreibung des zeitlichen Verhaltens. Die Zeitabhängigkeit wird bei kontinuierlichen stochastischen Prozessen durch den Zeitparameter t , bei zeitdiskreten stochastischen Prozessen durch den Zeitindex k als $t = kT$ ausgedrückt.

Die **Kreuzkorrelationsfunktion** von zwei Zufallsprozessen $\vec{v}(t)$ und $\vec{w}(t)$ ist definiert als (vgl. Korrelationsmatrix (A.12)):

$$r_{\vec{v}\vec{w}}(t_1, t_2) = \text{Cor}(\vec{v}(t_1), \vec{w}(t_2)) = E[\vec{v}(t_1) \vec{w}(t_2)^T] \quad (\text{A.18})$$

Die **Autokorrelationsfunktion** beschreibt die Selbstähnlichkeit eines Zufallsprozesses zu zwei unterschiedlichen Zeiten:

$$r_{\vec{v}}(t_1, t_2) = \text{Cor}(\vec{v}(t_1), \vec{v}(t_2)) = E[\vec{v}(t_1) \vec{v}(t_2)^T] \quad (\text{A.19})$$

Die **Kreuzkovarianz-** und die **Autokovarianzfunktion** (vgl. Kovarianzmatrix und Auto-Kovarianzmatrix (A.13)f.) lauten unter Nutzung von (A.15):

$$\text{Cov}(\vec{v}(t_1), \vec{w}(t_2)) = E[\vec{v}(t_1)\vec{w}(t_2)^T] - E[\vec{v}(t_1)]E[\vec{w}(t_2)]^T \quad (\text{A.20})$$

$$\text{Cov}(\vec{v}(t_1), \vec{v}(t_2)) = E[\vec{v}(t_1)\vec{v}(t_2)^T] - E[\vec{v}(t_1)]E[\vec{v}(t_2)]^T \quad (\text{A.21})$$

Die Definitionen der Korrelations- und Kovarianzfunktionen sind unabhängig davon, ob t_1 und t_2 Zeitpunkte der kontinuierlichen Zeit t oder diskrete Zeitpunkte $t_k = kT$ beschreiben.

Für $t_1 = t_2$ gehen die Kreuzkorrelations- und Kreuzkovarianzfunktion in die Korrelationsmatrix (A.12) und Kovarianzmatrix (A.13) über. Entsprechendes gilt für die Autokorrelations- und Autokovarianzfunktion.

Für mittelwertfreie Prozesse sind gemäß (A.19) und (A.20) Korrelationsfunktion und Kovarianzfunktion gleich (vgl. (A.16)):

$$Cor(\vec{v}(t_1), \vec{w}(t_2)) = Cov(\vec{v}(t_1), \vec{w}(t_2)) \quad \text{wenn} \quad E[\vec{v}(t)] = \vec{0}, E[\vec{w}(t)] = \vec{0} \quad (\text{A.22})$$

Ein stochastischer Prozess $\vec{v}(t)$ ist **stationär im strengen Sinne**, wenn seine Wahrscheinlichkeitsdichte zeitunabhängig ist: $p_{\vec{v}}(\vec{\xi}, t) = p_{\vec{v}}(\vec{\xi})$

Ein stochastischer Prozess $\vec{v}(t)$ ist **stationär im weiteren Sinne**, wenn sein Erwartungswert und seine Varianz zeitunabhängig sind.

In beiden Fällen ist der Erwartungswert des stochastischen Prozesses zeitunabhängig und sowohl Autokorrelations- als auch Autokovarianzfunktionen sind nur von der Zeitdifferenz $\tau = t_2 - t_1$ abhängig: $r_v(t_1, t_2) = r_v(\tau)$ [28]

Ein normalverteilter Zufallsprozess (A.17) wird vollständig durch Erwartungswert und Varianz beschrieben, sodass aus der Stationarität im weiteren auch die Stationarität im strengen Sinne folgt.

Die **spektrale Leistungsdichte** eines im weiteren Sinne stationären, stochastischen Prozesses $v(t)$ ist die Fourier-Transformierte seiner Autokorrelationsfunktion (A.19):

$$S_v(j\omega) = \int_{-\infty}^{\infty} r_v(\tau) e^{-j\omega\tau} d\tau \quad (\text{A.23})$$

Die Autokorrelationsfunktion kann durch die inverse Fourier-Transformation aus der spektralen Leistungsdichte berechnet werden:

$$r_v(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_v(j\omega) e^{j\omega\tau} d\omega \quad (\text{A.24})$$

Weißes Rauschen

Zeitkontinuierliches weißes Rauschen ist ein kontinuierlicher stochastischer Prozess $v(t)$ mit konstanter spektraler Leistungsdichte (A.23):

$$S_v(j\omega) = R_v \quad (\text{A.25})$$

Der stochastische Prozess hat daher unendliche Leistung und ist somit nur ein theoretisches Modell, das nicht realisierbar ist. Die Autokorrelationsfunktion lautet unter Nutzung von (A.24):

$$r_v(\tau) = R_v \delta(\tau) \quad (\text{A.26})$$

mit $\delta(\tau)$ als Dirac-Impuls, der bei $\tau = 0$ unendlich wird.

Bei einem mittelwertfreien Prozess ist die Autokorrelationsfunktion bei $\tau = t_2 - t_1 = 0$ gemäß (A.22) gleich der Varianz:

$$\text{Var}(v) = r_v(0) = R_v \delta(0) \quad (\text{A.27})$$

Die Varianz von kontinuierlichem weißen Rauschen ist damit unendlich.

Für einen mehrdimensionalen Zufallsprozess $\vec{v}(t)$, bestehend aus n statistisch unabhängigen, mittelwertfreien, kontinuierlichen weißen Rauschprozessen $v_i(t)$ mit den spektralen Leistungsdichten R_{v_i} , lautet damit die Autokorrelationsfunktion und mit (A.22) auch die Autokovarianzfunktion:

$$\text{Cov}(\vec{v}(t_1), \vec{v}(t_1 + \tau)) = r_{\vec{v}}(\tau) = \begin{pmatrix} R_{v_1} & 0 & \cdots & 0 \\ 0 & R_{v_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_{v_n} \end{pmatrix} \delta(\tau) = \mathbf{R}_{\vec{v}} \delta(\tau) \quad (\text{A.28})$$

Die spektrale Leistungsdichte von **zeitdiskretem weißen Rauschen** v_k ist konstant im Frequenzintervall $[-\pi, \pi]$:

$$S_{v_k}(j\omega) = \begin{cases} R_{v_k} & \forall \omega \in [-\pi, \pi] \\ 0 & \text{sonst} \end{cases} \quad (\text{A.29})$$

Da die Leistungsdichte außerhalb des Intervalls $[-\pi, \pi]$ null ist, ist die Leistung des diskreten Prozesses beschränkt. Diskretes weißes Rauschen ist daher, im Gegensatz zu kontinuierlichem weißen Rauschen, technisch realisierbar.

Mit der diskreten Fourier-Transformation lässt sich die Autokorrelationsfunktion berechnen als:

$$\text{Cor}(v_k, v_{k+\kappa}) = r_{v_k}(\kappa) = R_{v_k} \delta_\kappa \quad \text{mit} \quad \delta_\kappa = \begin{cases} 1 & \text{wenn } \kappa = 0 \\ 0 & \text{sonst} \end{cases} \quad (\text{A.30})$$

δ_κ ist das Kronecker-Delta, das bei $\kappa = 0$ gleich 1 und nicht, wie der Dirac-Impuls in (A.26), unendlich wird.

Auch bei einem diskreten, mittelwertfreien Prozess ist die Autokorrelationsfunktion bei $t_2 - t_1 = \kappa T = 0$ gemäß (A.22) gleich der Varianz (A.4) bei skalaren und gleich der Autokovarianzmatrix (A.14) bei mehrdimensionalen Prozessen. Die Varianz $\sigma_{v_k}^2$ von mittelwertfreiem, diskretem weißen Rauschen v_k ist daher gleich der spektralen Leistungsdichte R_{v_k} :

$$\text{Var}(v_k) = \sigma_{v_k}^2 = r_{v_k}(0) = R_{v_k} \quad (\text{A.31})$$

Für einen mehrdimensionalen Zufallsprozess \vec{v}_k , bestehend aus n statistisch unabhängigen, mittelwertfreien, diskreten weißen Rauschprozessen $v_{i,k}$ mit den spektralen Leistungsdichten $R_{v_{i,k}}$ und Varianzen $\sigma_{v_{i,k}}^2$, lautet die Auto-Kovarianzmatrix damit:

$$\begin{aligned} \text{Cov}(\vec{v}_k, \vec{v}_k) &= \begin{pmatrix} \sigma_{v_{1,k}}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{v_{2,k}}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_{v_{n,k}}^2 \end{pmatrix} = \begin{pmatrix} R_{v_{1,k}} & 0 & \cdots & 0 \\ 0 & R_{v_{2,k}} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_{v_{n,k}} \end{pmatrix} = \mathbf{R}_{\vec{v}_k} \\ \text{Cov}(\vec{v}_k, \vec{v}_{k+\kappa}) &= \mathbf{0} \quad \text{für } \kappa \neq 0 \end{aligned} \tag{A.32}$$

Random-Walk

Integriert man normalverteiltes, kontinuierliches weißes Rauschen $v(t)$, erhält man einen stochastischen Prozess der Form:

$$x(t) = \int_0^t v(q) \, dq \tag{A.33}$$

Mit $P_x(0) = 0$ und R_v als spektraler Leistungsdichte (A.25) des kontinuierlichen weißen Rauschens ergeben sich für Erwartungswert und Standardabweichung von $x(t)$:

$$\mu_x(t) = E[x(t)] = 0 \quad \sigma_x(t) = \sqrt{\text{Var}(x(t))} = \sqrt{R_v t} \tag{A.34}$$

Die Standardabweichung von $x(t)$ wächst also mit der Wurzel der Zeit. Der Zahlenwert der Standardabweichung bei $t = 1$ ist der so genannte **Random-Walk-Koeffizient** N [86]:

$$N_x = \sigma_x(t = 1) = \sqrt{R_v} \tag{A.35}$$

B Ergänzungen

B.1 Lagedarstellungen

Quaternionen

Für das Produkt zweier Quaternionen $\mathbf{q} = (q_1, q_2, q_3, q_4)^T$ und $\mathbf{r} = (r_1, r_2, r_3, r_4)^T$ gilt [28, 84]:

$$\mathbf{q} \bullet \mathbf{r} = \begin{pmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} \quad (\text{B.1})$$

Damit lässt sich zeigen, dass die Quaternionen-Multiplikation nicht kommutativ $\mathbf{q} \bullet \mathbf{r} \neq \mathbf{r} \bullet \mathbf{q}$, aber assoziativ $\mathbf{q} \bullet (\mathbf{r} \bullet \mathbf{s}) = (\mathbf{q} \bullet \mathbf{r}) \bullet \mathbf{s}$ ist.

Richtungskosinusmatrix aus Basisvektoren

Die Richtungskosinusmatrix \mathbf{C}_b^a (2.13) zur Koordinatentransformation eines Vektors von B nach A (2.14) lässt sich auch aus den Basisvektoren von B und A konstruieren [28]:

$$\mathbf{C}_b^a = \begin{pmatrix} \vec{a}_1 \cdot \vec{b}_1 & \vec{a}_1 \cdot \vec{b}_2 & \vec{a}_1 \cdot \vec{b}_3 \\ \vec{a}_2 \cdot \vec{b}_1 & \vec{a}_2 \cdot \vec{b}_2 & \vec{a}_2 \cdot \vec{b}_3 \\ \vec{a}_3 \cdot \vec{b}_1 & \vec{a}_3 \cdot \vec{b}_2 & \vec{a}_3 \cdot \vec{b}_3 \end{pmatrix} \quad (\text{B.2})$$

Ein Element c_{ij} der Richtungskosinusmatrix ist also das Skalarprodukt der Basisvektoren \vec{a}_i und \vec{b}_j :

$$c_{ij} = \vec{a}_i \cdot \vec{b}_j \quad \forall i, j \in \{1, 2, 3\} \quad (\text{B.3})$$

Transformationen von Lagedarstellungen

Die Eulerwinkel gemäß (2.16) lassen sich folgendermaßen in die Richtungskosinusmatrix \mathbf{C}_b^a (2.13) umrechnen:

$$\mathbf{C}_b^a = \begin{pmatrix} \cos \psi \cos \theta & \cos \psi \sin \varphi \sin \theta - \cos \varphi \sin \psi & \sin \varphi \sin \psi + \cos \varphi \cos \psi \sin \theta \\ \cos \theta \sin \psi & \cos \varphi \cos \psi + \sin \varphi \sin \psi \sin \theta & \cos \varphi \sin \psi \sin \theta - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \varphi \cos \theta \end{pmatrix} \quad (\text{B.4})$$

Für die Umrechnung der Richtungskosinusmatrix \mathbf{C}_b^a in das Lagequaternion \mathbf{q}_b^a gilt als Inversion von (2.13) [28]:

$$\mathbf{q}_b^a = \begin{pmatrix} \zeta \\ \frac{1}{4\zeta}(c_{32} - c_{23}) \\ \frac{1}{4\zeta}(c_{13} - c_{31}) \\ \frac{1}{4\zeta}(c_{21} - c_{12}) \end{pmatrix} \quad \text{mit} \quad \zeta = \frac{1}{2}\sqrt{1 + c_{11} + c_{22} + c_{33}} \quad (\text{B.5})$$

Es muss $1 + c_{11} + c_{22} + c_{33} \neq 0$ gelten. Für Alternativen siehe [84].

Das Lagequaternion \mathbf{q}_b^a kann folgendermaßen aus den Eulerwinkeln berechnet werden [84]:

$$\mathbf{q}_b^a = \begin{pmatrix} \cos \frac{\varphi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\varphi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\varphi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\varphi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\varphi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\varphi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\varphi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\varphi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{pmatrix} \quad (\text{B.6})$$

Die Funktion $f_w(\alpha)$ rechnet einen Winkel α in einen äquivalenten Winkel im Intervall $]-\pi; \pi]$ um:

$$f_w(\alpha) = \alpha - 2\pi \left[\frac{\alpha}{2\pi} - \frac{1}{2} \right] \quad (\text{B.7})$$

Der Ausdruck zwischen den Klammern $[\]$ wird dabei auf die nächste ganze Zahl aufgerundet.

Eulerwinkelraten

Für die Eulerwinkelraten gilt als Funktion der Drehraten und Eulerwinkel [28]:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (\text{B.8})$$

für Nickwinkel $-90^\circ < \theta < 90^\circ$.

B.2 Weitere mathematische Ergänzungen

Kreuzproduktbildende Matrix

Das Kreuzprodukt zweier Vektoren \vec{a} und \vec{b} kann auch durch die Multiplikation mit einer schiefsymmetrischen Matrix $[\vec{a}\times]$ dargestellt werden:

$$\vec{a} \times \vec{b} = [\vec{a}\times] \vec{b} \quad \text{mit} \quad \vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{und} \quad [\vec{a}\times] = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad (\text{B.9})$$

Ableitung in zwei Bezugssystemen

Für die zeitliche Ableitung eines Vektors \vec{a} in den Bezugssystemen A und B gilt folgender Zusammenhang [42]:

$$\frac{{}^A d\vec{a}}{dt} = \frac{{}^B d\vec{a}}{dt} + \vec{\omega}_{ab} \times \vec{a} \quad (\text{B.10})$$

$\vec{\omega}_{ab}$ ist die Drehrate (2.17) von B in A

Für die zeitliche Ableitung eines körperfesten Basisvektors \vec{b}_m bezüglich Bezugssystem L folgt daraus beispielsweise:

$$\dot{\vec{b}}_m = \frac{{}^L d\vec{b}_m}{dt} = \underbrace{\frac{{}^B d\vec{b}_m}{dt}}_{=0} + \vec{\omega}_{lb} \times \vec{b}_m \quad \forall m \in \{x, y, z\} \quad (\text{B.11})$$

Beschränkungen

Eine Beschränkung einer Variablen a auf die obere Grenze u und die untere Grenze l wird geschrieben als

$$\text{sat}_l^u(a) = \begin{cases} u & \text{für } a > u \\ a & \text{sonst} \\ l & \text{für } a < l \end{cases} \quad (\text{B.12})$$

Für einen Vektor \vec{a} beschränkt $\text{sat}_l^u(\vec{a})$ jede Komponente a_i von \vec{a} :

$$\text{sat}_l^u(\vec{a}) = (a_{1,\text{sat}}, \dots, a_{n,\text{sat}})^T \quad \text{mit} \quad a_{i,\text{sat}} = \text{sat}_l^u(a_i) \quad \forall i \in \{1, \dots, n\} \quad (\text{B.13})$$

B.3 Ergänzungen zum Kapitel Regelung

Bewegungsgleichungen in Koordinatenschreibweise

Die Bewegungsgleichungen (5.39) des Systemmodells für die flachheitsbasierte Zustandsrückführung lauten in Koordinatenschreibweise:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} v_n \\ v_e \\ v_d \end{pmatrix} \quad (\text{B.14a})$$

$$\begin{pmatrix} \dot{v}_n \\ \dot{v}_e \\ \dot{v}_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - \mathbf{C}_b^l \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_1 \quad (\text{B.14b})$$

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \boldsymbol{\Omega}(\mathbf{q}_b^l) \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (\text{B.14c})$$

$$\begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} = \begin{pmatrix} u_2 \\ u_3 \\ u_4 \end{pmatrix} \quad (\text{B.14d})$$

Dabei ist \mathbf{C}_b^l und $\boldsymbol{\Omega}(\mathbf{q}_b^l)$ gemäß (2.13) und (2.18) jeweils eine Funktion von \mathbf{q}_b^l bzw. $(q_1 \ q_2 \ q_3 \ q_4)^T$.

Projektionen für den Beweis der Flachheit

Projektion von (5.76) auf \vec{l}_d zur Herleitung von (5.84):

$$\underbrace{\ddot{v}_{lb} \cdot \vec{l}_d}_{p_d^{(3)}} = -\dot{u}_1 \underbrace{\vec{b}_z \cdot \vec{l}_d}_{c_{33}} - u_1 \underbrace{(\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{l}_d}_{\omega_y c_{31} - \omega_x c_{32}} + \underbrace{\dot{a}_s \cdot \vec{l}_d}_{\dot{a}_{sd}} \quad (\text{B.15})$$

Projektion von (5.86) auf \vec{b}_y und \vec{b}_x zur Herleitung von (5.87):

$$\begin{aligned} & -\ddot{u}_1 \underbrace{\vec{b}_z \cdot \vec{b}_y}_0 - 2\dot{u}_1 \underbrace{(\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{b}_y}_{-\omega_x} - u_1 \underbrace{(\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{b}_y}_{-\omega_x = -u_2} - u_1 \underbrace{(\vec{\omega}_{lb} \times (\vec{\omega}_{lb} \times \vec{b}_z)) \cdot \vec{b}_y}_{\omega_y \omega_z} \\ & + \ddot{a}_{sn} \underbrace{\vec{l}_n \cdot \vec{b}_y}_{c_{12}} + \ddot{a}_{se} \underbrace{\vec{l}_e \cdot \vec{b}_y}_{c_{22}} + \ddot{a}_{sd} \underbrace{\vec{l}_d \cdot \vec{b}_y}_{c_{32}} = p_n^{(4)} \underbrace{\vec{l}_n \cdot \vec{b}_y}_{c_{12}} + p_e^{(4)} \underbrace{\vec{l}_e \cdot \vec{b}_y}_{c_{22}} + p_d^{(4)} \underbrace{\vec{l}_d \cdot \vec{b}_y}_{c_{32}} \end{aligned} \quad (\text{B.16})$$

$$\begin{aligned}
 & \ddot{u}_1 \underbrace{\vec{b}_z \cdot \vec{b}_x}_0 - 2\dot{u}_1 \underbrace{(\vec{\omega}_{lb} \times \vec{b}_z) \cdot \vec{b}_x}_{\omega_y} - u_1 \underbrace{(\dot{\vec{\omega}}_{lb} \times \vec{b}_z) \cdot \vec{b}_x}_{\dot{\omega}_y = u_3} - u_1 \underbrace{(\vec{\omega}_{lb} \times (\vec{\omega}_{lb} \times \vec{b}_z)) \cdot \vec{b}_x}_{\omega_x \omega_z} \\
 & + \ddot{a}_{sn} \underbrace{\vec{l}_n \cdot \vec{b}_x}_{c_{11}} + \ddot{a}_{se} \underbrace{\vec{l}_e \cdot \vec{b}_x}_{c_{21}} + \ddot{a}_{sd} \underbrace{\vec{l}_d \cdot \vec{b}_x}_{c_{31}} = p_n^{(4)} \underbrace{\vec{l}_n \cdot \vec{b}_x}_{c_{11}} + p_e^{(4)} \underbrace{\vec{l}_e \cdot \vec{b}_x}_{c_{21}} + p_d^{(4)} \underbrace{\vec{l}_d \cdot \vec{b}_x}_{c_{31}}
 \end{aligned} \tag{B.17}$$

Matrizen des Zustandsraummodells für die MPC-Trajektorien-generierung

Die Matrizen für das kontinuierliche, lineare Zustandsraummodell (5.148), (5.156):

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.18}$$

C Parameter

Parameter	Symbol	Zahlenwert	Einheit
Abstand Rotorachsen zur vertikalen Symmetrieachse des Quadropters	d_R	212,5	mm
Vektor von der IMU zur GNSS Antenne	\vec{d}_A^b	$(0; 0; -66)^T$	mm
Vektor vom Zentrum des Quadrok. zur IMU	\vec{d}_I^b	$(0; 0; -59,5)^T$	mm
Vektor vom Zentrum des Quadrok. zur Kamera	\vec{d}_c^b	$(25; -23; 166)^T$	mm
Gesamtmasse des Quadropters	m	2,267	kg
Trägheitsmoment des Quadrok. um die Rollachse	J_x	0,038	kg m ²
Trägheitsmoment des Quadrok. um die Nickachse	J_y	0,038	kg m ²
Produkt aus dem Trägheitsmoment des Quadrok. um die Hochachse und dem Quotienten aus Schub- und Widerstandsmoment-Faktoren der Rotoren	$J_z k_t / k_d$	0,9	kg m

Tabelle C.1: Parameter zur Geometrie und Masse des Quadropters

Standardabweichung	Symbol	Zahlenwert	Einheit
Magnetometer diskretes Winkelrauschen	$\sigma_{m,k}$	$\pi/2$	rad
Initialer Lagewinkelfehler	$\sigma_{\alpha,0}, \sigma_{\beta,0}$	2	deg
Initialer Gierwinkelfehler	$\sigma_{\gamma,0}$	5	deg
Initialer Beschleunigungsbiasfehler	$\sigma_{\Delta b_a,0}$	0,03	m/s ²
Initialer Drehratenbiasfehler	$\sigma_{\Delta b_\omega,0}$	0,001	rad/s
Störgrößen-Beobachter, disk. Systemrauschen	$\sigma_{w_s,k}$	0,01	m/s ²
Störgrößen-Beobachter, disk. Messrauschen	$\sigma_{v_s,k}$	0,2	m/s ²

Tabelle C.2: Zahlenwerte der Standardabweichungen für Sensorfusion und Störgrößen-Beobachter

$L_{i,j}$ [1/s]	$L_{i,3}$	$L_{i,2}$	$L_{i,1}$	$L_{i,0}$	$L_{i,I}$
Nord, Ost $i = 1, 2$	13	9	2,5	1	0
Vertikal $i = 3$	-	-	7	2,9	0,25
Gier $i = 4$	-	-	8,33	2,67	0

Tabelle C.3: Übertragungsfaktoren für stabilisierenden Regler

	$h_{i,4,lim}$	$h_{i,3,lim}$	$h_{i,2,lim}$	$h_{i,1,lim}$	$h_{i,0,lim}$
Nord, Ost $i = 1, 2$	380 m/s ⁴	13,5 m/s ³	3 m/s ²	1 m/s	-
Vertikal $i = 3$	-	-	2 m/s ²	0,5 m/s	0,05 m
Gier $i = 4$	-	-	8 rad/s ²	0,79 rad/s	-

Tabelle C.4: Obere und untere Beschränkungen für stabilisierenden Regler

Betriebsmodus	L_{t3}	L_{t2}	L_{t1}	L_{t0}	L_{ψ}
Joystick	24/s	6/s	1,5/s	-	-
Objektverfolgung	60/s	15/s	4/s	1/s	2/s

Tabelle C.5: Übertragungsfaktoren der kaskadierten Regler zur Trajektoriengenerierung

Method	Kaskadenregelung		MPC
Betriebsmodus	Joystick	Objektverfolgung	POI
$\dot{\delta}_{lim}$	-	2 m/s	-
$v_{\alpha,lim}$	-	2 m/s	-
$v_{ne,lim}$	4 m/s	4 m/s	0,5 m/s
a_{lim}	3 m/s ²	3 m/s ²	1,5 m/s ²
\dot{a}_{lim}	6 m/s ³	10 m/s ³	6 m/s ³
\ddot{a}_{lim}	40 m/s ⁴	100 m/s ⁴	50 m/s ⁴
$v_{d,lim}$	1 m/s	-	0,5 m/s
$\dot{\psi}_{lim}$	45 °/s	20 °/s	45 °/s
$\ddot{\psi}_{lim}$	(8 rad/s ²)	(8 rad/s ²)	4 rad/s ²

Tabelle C.6: Beschränkungen der Trajektoriengenerierung

Abbildungsverzeichnis

2.1	Erdellipsoid und Koordinatensysteme	25
2.2	Algorithmus Kalman-Filter und Erweitertes Kalman-Filter	32
2.3	Flachheitsbasierte Vorsteuerung	41
2.4	Flaches System mit exakt linearisierender Zustandsrückführung und lineares System in Brunovský-Normalform	45
2.5	Zivile GNSS-Frequenzbänder	49
2.6	GNSS-Satellitenkonstellation	52
3.1	Quadrocopter	65
3.2	Hardwarearchitektur des Flugsystems	66
3.3	Kamera im 2-Achs-Gimbal	67
3.4	Kamera-Videobild	67
3.5	Adapterplatine für GNSS-RTK-Empfänger	68
3.6	3-D-gedruckter Adapter	69
3.7	Schachbrettmuster	70
3.8	Weitwinkel-Kamera: Projektion Einheitskugel auf Bildpunkte (u, v)	71
3.9	Architektur der Software auf dem Mastermind	72
3.10	Rcon Simulink-Block-Typen	74
3.11	Rcon iOS-App	74
3.12	ControlPad	77
3.13	ControlPad Innen- und Rückansicht	78
3.14	Icon der iOS-App „QC Cam Ctrl“	79
3.15	QC Cam Ctrl: Navigationsmenü und Ansicht „Flight“	80
3.16	Schubkalibrierungs-Ansicht	81
3.17	Betriebsmodus-Ansicht	81
3.18	Aktivierung Objektverfolgung	82
3.19	Aktivierung Joystick	82
3.20	Schieberegler Absolutwinkel	83
3.21	Aktivierung Modus „statischer Punkt“	83
3.22	QC Cam Ctrl: Steuerung über virtuelle 3-D-Ansicht	85
3.23	Bodenstation	87
3.24	Ziel-Einheit	87
3.25	Bodenstation Innenansicht	87
3.26	Ziel-Einheit Innenansicht	87
3.27	Videomonitor-Einheit	87
3.28	Kommunikationsarchitektur	88

3.29	Nachrichtenformat StdQCMMessage	88
3.30	Testfahrzeug	89
3.31	3-D-Visualisierung in der Simulationsumgebung	91
3.32	Funktionsprinzip Infrarot-Trackingsystem	92
3.33	Luftbild Aldenhoven Testing Center	93
3.34	Sendemast des Pseudoliten PSL6 im automotiveGATE (im Jahr 2011) . .	94
4.1	IMU-Platine auf Gummi-Dämpfern	98
4.2	Typische Inertialsensorkurve mit Fehlern	100
4.3	Allan-Deviation Drehratensensoren	103
4.4	Allan-Deviation Beschleunigungssensoren	104
4.5	GNSS-RTK-Empfänger im Quadrokopter	108
4.6	Experimenteller GNSS-Empfänger im Quadrokopter	110
4.7	Ablauf eines Zeitschrittes des Kalman-Filters	139
5.1	Position des Quadrokopters, Koordinatensysteme B und L , Rotornummerierung	145
5.2	Rotor mit Brushless-Motor	147
5.3	Rotorschub als Funktion der normierten Motordrehrate	151
5.4	Geometrische Form des Quadrokopters zur Abschätzung der Struktur des Trägheitstensors	156
5.5	Flaches System mit quasi-statischer Zustandsrückführung, lineares System in Brunovský-Normalform	169
5.6	Berechnung der Motordrehzahl-Stellgrößen	171
5.7	Stabiler Systemkanal bestehend aus Integrator-Ketten-Strecke und kaskadiertem Regler mit Beschränkungen	172
5.8	Kamera-Gimbal	177
5.9	Spline 5. Grades	181
5.10	Trajektoriengenerierung horizontal	182
5.11	Generierte Trajektorie für Joystick-Kommandos	185
5.12	MPC-Trajektoriengenerierung: Positionswechsel	192
5.13	MPC-Trajektoriengenerierung: Trajektorie POI-Szenario	193
5.14	MPC-Trajektoriengenerierung POI-Szenario: Sollgrößen, Geschwindigkei- ten sowie Stellgrößen	194
5.15	Ablaufsteuerung	196
5.16	Simulation nichtlineare Systemdynamik mit flachheitsbasierter Zustands- rückführung und Trajektorienfolgeregelung	199
5.17	Visualisierung eines Positionswechsels über 2 m	200
6.1	Flugversuch zur Validierung der Sensorfusion	202
6.2	Karte Flugversuch zur Sensorfusionsvalidierung	203
6.3	Positionsschätzung	204
6.4	Alter Δt_A der Nachrichten des GNSS-Empfängers	204
6.5	Geschwindigkeitsschätzung	206
6.6	Lageschätzung	207

6.7	Beschleunigungssensor-Bias-Schätzung	207
6.8	Drehratensensor-Bias-Schätzung	208
6.9	Kumulative Häufigkeitsverteilung der Sensorfusionsfehler	209
6.10	Pkw-Trajektorie und Pseudoliten-Positionen im automotiveGATE	210
6.11	Offset-korrigierte Pseudorange zu den Pseudoliten	211
6.12	Erprobung des experimentellen GPS- und Galileo-Empfängers auf dem Quadrokopter im automotiveGATE	212
6.13	Positionslösung des experimentellen GNSS-Empfängers im Quadrokopter- Flug	212
6.14	Schubkalibrierung während des Schwebefluges	214
6.15	Flugversuch A1 „Spline-Trajektorie horizontal“: Trajektorienfolgeregelung, Ost-Kanal	216
6.16	Flugversuch A2 „Spline-Trajektorie vertikal“: Trajektorienfolgeregelung, Vertikalkanal	218
6.17	Quadrokopter im Indoor-Flugversuch	219
6.18	Flugversuch B1 „MPC-Trajektorie horizontal“	220
6.19	Flugversuch B2 „MPC-Trajektorie horizontal mit Gieren“	220
6.20	Steuerung des Quadrokopters mit dem 6-Achs-Joystick des ControlPads	221
6.21	Flugversuch C „Joystick“: Geschwindigkeit und Position	222
6.22	Kommandierung Abstand und Azimut zur Objektverfolgung	223
6.23	Flugversuch D1 „Statisches Ziel“: Abstand zum Zielobjekt δ und Azimut α	224
6.24	Flugversuch D1 „Statisches Ziel“: nördl. und östl. Position, Gierwinkel	225
6.25	Karte des Flugversuchs D1 „Statisches Ziel“	226
6.26	Flugversuch D1 „Statisches Ziel“: kommandierte Gimbal-Nick- und Roll- winkel	226
6.27	Flugversuch D1 „Statisches Ziel“: Positionen des Zielobjekts im Video	227
6.28	Flugversuch D1 „Statisches Ziel“: Gierwinkelfehler $\Delta\psi$ und Nickwinkelfeh- ler $\Delta\theta$ aus Videoauswertung	228
6.29	Karte des Flugversuchs D2 „Objektverfolgung Gerade“	229
6.30	Flugversuch D2 „Objektverfolgung Gerade“: Abstand zum Zielobjekt δ und Azimut α	230
6.31	Flugversuch D2 „Objektverfolgung Gerade“: nördliche und östliche Positi- on, Gierwinkel	230
6.32	Flugversuch D2 „Objektverfolgung Gerade“: Positionen des Zielobjekts im Video	231
6.33	Flugversuch D2 „Objektverfolgung Gerade“: Gierwinkelfehler und Nick- winkelfehler aus Videoauswertung	232
6.34	Karte des Flugversuchs D3 „Objektverfolgung Halbkreis“	233
6.35	Flugversuch D3 „Objektverfolgung Halbkreis“: Abstand zum Zielobjekt δ und Azimut α	234
6.36	Flugversuch D3 „Objektverfolgung Halbkreis“: nördliche und östliche Po- sition, Gierwinkel	234
6.37	Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Ost-Kanal	236

6.38	Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Nord-Kanal	237
6.39	Flugversuch D3 „Objektverfolgung Halbkreis“: Roll- und Nickwinkel . . .	238
6.40	Flugversuch D3 „Objektverfolgung Halbkreis“: Beschleunigungsschätzung	239
6.41	Flugversuch D3 „Objektverfolgung Halbkreis“: Schätzung spezifische Störkraft	239
6.42	Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Gier-Kanal	240
6.43	Flugversuch D3 „Objektverfolgung Halbkreis“: Trajektorienfolgeregelung, Vertikal-Kanal	241
6.44	Flugversuch D3 „Objektverfolgung Halbkreis“: Stellgrößen u_1, u_2, u_3, u_4 .	242
6.45	Flugversuch D3 „Objektverfolgung Halbkreis“: Positionen des Zielobjekts im Video	243
6.46	Flugversuch D3 „Objektverfolgung Halbkreis“: Gierwinkelfehler $\Delta\psi$ und Nickwinkelfehler $\Delta\theta$ aus Videoauswertung	243

Tabellenverzeichnis

2.1	Parameter des WGS84-Referenzsystems	26
3.1	Intrinsische Kamera-Parameter	70
3.2	Steuerungskanäle und Betriebsmodi	76
4.1	Random-Walk-Koeffizienten der IMU	105
4.2	Spektrale Leistungsdichten des IMU-Messrauschens	105
4.3	Bias-Instabilitäts-Koeffizienten der IMU	106
4.4	Bias-Random-Walk-Koeffizienten der IMU	106
4.5	Spektrale Leistungsdichten der IMU-Bias-Drifts	107
5.1	Parameter des Motordynamikmodells	149
5.2	Parameter des Rotorschubmodells	150
5.3	Luftwiderstands-Kraft und notwendige Schrägstellung des Quadropters	152
5.4	Identifizierte Trägheitsmomente	158
5.5	Beschränkungen für \vec{u} und resultierende Rotorschubkräfte	171
C.1	Parameter zur Geometrie und Masse des Quadropters	261
C.2	Zahlenwerte der Standardabweichungen für Sensorfusion und Störgrößen- Beobachter	261
C.3	Übertragungsfaktoren für stabilisierenden Regler	262
C.4	Obere und untere Beschränkungen für stabilisierenden Regler	262
C.5	Übertragungsfaktoren der kaskadierten Regler zur Trajektoriengenerierung	262
C.6	Beschränkungen der Trajektoriengenerierung	262

Literaturverzeichnis

- [1] Dirk Abel. *Regelungstechnik und Ergänzungen (Höhere Regelungstechnik)*. 43. Auflage. Verlag Mainz, 2018. ISBN: 978-3-8107-0067-4.
- [2] Carlos Aguilar-Ibáñez, Hebertt Sira-Ramírez, Miguel S. Suárez-Castañón, Edgar Martínez-Navarro und Marco A. Moreno-Armendariz. „The trajectory tracking problem for an unmanned four-rotor system: flatness-based approach“. In: *International Journal of Control*, Band 85, Nummer 1 (2012), Seiten 69–77. DOI: 10.1080/00207179.2011.638328.
- [3] Aldenhoven Testing Center of RWTH Aachen University GmbH. *Aldenhoven Testing Center (ATC)*. 2020. URL: <https://www.atc-aldenhoven.de> (besucht am 15.03.2020).
- [4] Kostas Alexis, George Nikolakopoulos und Anthony Tzes. „Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances“. In: *Control Engineering Practice*, Band 19, Nummer 10 (2011), Seiten 1195–1207. DOI: 10.1016/j.conengprac.2011.06.010.
- [5] David W. Allan. „Statistics of atomic frequency standards“. In: *Proceedings of the IEEE*, Band 54, Nummer 2 (1966), Seiten 221–230. DOI: 10.1109/PROC.1966.4634.
- [6] ART GmbH. *Advanced Realtime Tracking*. URL: <https://ar-tracking.com> (besucht am 25.05.2020).
- [7] Ascending Technologies GmbH. *AscTec Research Catalogue, V1.1*. 2013.
- [8] Auterion - the drone software platform built for enterprise. *Auterion*. URL: <https://auterion.com> (besucht am 25.05.2020).
- [9] Samir Bouabdallah. „Design and control of quadrotors with application to autonomous flying“. Dissertation. 2007. DOI: 10.5075/epfl-thesis-3727.
- [10] Haitham Bou-Ammar, Holger Voos und Wolfgang Ertel. „Controller design for quadrotor UAVs using reinforcement learning“. In: *IEEE International Conference on Control Applications*. IEEE, 2010, Seiten 2130–2135. DOI: 10.1109/CCA.2010.5611206.
- [11] Michael Buhl, Oliver Fritsch und Boris Lohmann. „Exakte Ein-/Ausgangslinearisierung für die translatorische Dynamik eines Quadrocopters“. In: *at - Automatisierungstechnik*, Band 59, Nummer 1 (2011), Seiten 374–381. DOI: 10.1524/auto.2011.0932.
- [12] Department Of Defense USA. *Global Positioning System Standard Positioning Service Performance Standard, 5th Edition*. 2020. URL: <https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf>.

-
- [13] Deutsches Zentrum für Luft- und Raumfahrt. *GATEs in Deutschland*. 2015. URL: <https://www.dlr.de/rd/GATEs> (besucht am 21.05.2020).
- [14] Thomas Engelhardt und Dirk Abel. „automotiveGATE and railGATE Test Areas for Galileo-based Vehicle Guidance“. In: *European Navigation Conference (ENC) 2011, London, 29.11. - 01.12.2011*. 2011.
- [15] Thomas Engelhardt und Dirk Abel. „Galileo-Testgebiete automotiveGATE und railGATE und erste Anwendungen“. In: *DGON Navigationskonvent 2011, Berlin, 7.-8.7.2011*. Deutsche Gesellschaft für Ortung und Navigation e.V., 2011.
- [16] Thomas Engelhardt, Alexander Katriniok, Björn Schäfer und Dirk Abel. „automotiveGATE und railGATE: Galileo-basierte Fahrzeugführung für bodengebundenen Verkehr“. In: *E-Mobility: Technologien - Infrastruktur - Märkte. VDE-Kongress 2010, Leipzig, 08.-09.11.2010*. VDE-Verlag, 2010.
- [17] Thomas Engelhardt, Alexander Katriniok, Björn Schäfer und Dirk Abel. „Fahrzeugführung mit Galileo – Anwendungen und Testgebiete“. In: *Deutscher Luft- und Raumfahrtkongress 2010, Hamburg, 31.08.-02.09.2010*. Deutsche Gesellschaft für Luft- und Raumfahrt (DGLR), 2010, Seiten 867–872. ISBN: 978-3-932182-68-5.
- [18] Thomas Engelhardt, Thomas Konrad, Björn Schäfer und Dirk Abel. „Flatness-based control for a quadrotor camera helicopter using model predictive control trajectory generation“. In: *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, Seiten 852–859. DOI: 10.1109/MED.2016.7536036.
- [19] Thomas Engelhardt, Lothar Kurz, Tobias Noll und Dirk Abel. „Experimental Galileo-Receiver for Quadrocopter Navigation in automotiveGATE“. In: *European Navigation Conference (ENC) 2014*. Rotterdam, 2014.
- [20] Thomas Engelhardt, René Rütters, Alexander Katriniok und Dirk Abel. „automotiveGATE und railGATE, Testgebiete für Galileo-basierte Fahrzeugführung auf Schiene und Straße“. In: *Automation 2011: Der 12. Branchentreff der Mess- und Automatisierungstechnik, Kongress Baden-Baden, 28.-29.06.2011*. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2011. ISBN: 978-3-18-092143-3.
- [21] Thomas Engelhardt und Björn Schäfer. „railGATE: Galileo-Testgebiet für Schienenfahrzeug-Anwendungen“. In: *15. Workshop des Fachausschusses 7.61 Automatisierung für Schienenverkehrssysteme in der VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 18.-19.09.2014, Wegberg-Wildenrath*. 2014.
- [22] European GNSS Agency. *EGNOS Open Service (OS) Service Definition Document*. 2.3. 2017.
- [23] European GNSS Agency. *European GNSS Agency*. 2020. URL: <https://www.gsa.europa.eu> (besucht am 18.10.2020).
- [24] European GNSS Agency. *European GNSS Service Centre*. 2020. URL: <https://www.gsc-europa.eu> (besucht am 25.05.2020).
- [25] European GNSS Agency. *GSA GNSS Market Report*. 6. 2019. ISBN: 978-92-9206-043-5. DOI: 10.2878/031762.

- [26] European Union. *European GNSS (Galileo) Open Service Signal In Space Interface Control Document*. 1.1. 2010. DOI: 10.2768/1968.
- [27] European Union. *European GNSS (Galileo) Open Service Signal-in-Space Interface Control Document*. 1.3. 2016.
- [28] Jay A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008. ISBN: 0-07-164266-8.
- [29] Michel Fliess, Jean Lévine, Philippe Martin und Pierre Rouchon. „Flatness and defect of non-linear systems: introductory theory and examples“. In: *International Journal of Control*, Band 61, Nummer 6 (1995), Seiten 1327–1361. DOI: 10.1080/00207179508921959.
- [30] Michel Fliess, Jean Lévine, Philippe Martin und Pierre Rouchon. „On Differentially Flat Nonlinear Systems“. In: *IFAC Proceedings Volumes*, Band 25, Nummer 13 (1992), Seiten 159–163. DOI: 10.1016/S1474-6670(17)52275-2.
- [31] Michel Fliess, Jean Lévine, Philippe Martin und Pierre Rouchon. „Some open questions related to flat nonlinear systems“. In: *Open Problems in Mathematical Systems and Control Theory*. Band 7. 5. 1999, Seiten 99–103. DOI: 10.1007/978-1-4471-0807-8_21.
- [32] Frankfurter Allgemeine. *Chinas Regierung bekommt Zugriff auf unsere Drohnen-Bilder*. 2016. URL: <https://www.faz.net/aktuell/wirtschaft/agenda/china-bekommt-zugriff-auf-daten-bilder-von-dji-drohnen-kunden-14190615.html>.
- [33] Freefly Systems Inc. *Freefly Systems - Camera Movement Systems for Filmmakers*. URL: <https://freeflysystems.com> (besucht am 25.05.2020).
- [34] Marco. B. Gerig. „Modeling, guidance, and control of aerobatic maneuvers of an autonomous helicopter“. Ph.D. Thesis. ETH Zurich, 2008. DOI: <https://doi.org/10.3929/ethz-a-005684509>.
- [35] Daniel Gurdan, Jan Stumpf, Michael Achtelik, Klaus-Michael Doth, Gerd Hirzinger und Daniela Rus. „Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz“. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. April. IEEE, 2007, Seiten 361–366. DOI: 10.1109/ROBOT.2007.363813.
- [36] HD Skycam GmbH. *HD Skycam*. URL: <https://hd-skycam.net> (besucht am 25.05.2020).
- [37] Nicholas J. Higham. „The Scaling and Squaring Method for the Matrix Exponential Revisited“. In: *SIAM Journal On Matrix Analysis and Applications*, Band 26, Nummer 4 (2005), Seiten 1179–1193. URL: <http://eprints.maths.manchester.ac.uk/634/>.
- [38] Gabriel M. Hoffmann, Haomiao Huang, Steven L. Waslander und Claire J. Tomlin. „Precision flight control for a multi-vehicle quadrotor helicopter testbed“. In: *Control Engineering Practice* (2011). DOI: doi:10.1016/j.conengprac.2011.04.005.

-
- [39] IAC PNT. *Information and Analysis Center for Positioning, Navigation and Timing*. 2020. URL: <https://www.glonass-iac.ru/en> (besucht am 18.10.2020).
- [40] IEEE Standards Board. *Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, Annex C*. 1997. URL: <https://standards.ieee.org/standard/952-1997.html>.
- [41] Intel Deutschland GmbH. *Ascending Technologies*. 2019. URL: <http://www.asctec.de> (besucht am 01.12.2019).
- [42] Thomas R. Kane und David. A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill, 1985. ISBN: 0-07-037846-0.
- [43] Thomas Konrad und Dirk Abel. „A Flatness-based Control Strategy for Multirotors in Industrial Applications“. In: *IFAC-PapersOnLine*, Band 51, Nummer 22 (2018), Seiten 453–458. DOI: 10.1016/j.ifacol.2018.11.592.
- [44] Thomas Konrad, Michael Breuer, Thomas Engelhardt und Dirk Abel. „State Estimation for a Multirotor using Tight-Coupling of GNSS and Inertial Navigation“. In: *IFAC-PapersOnLine*, Band 50, Nummer 1 (2017), Seiten 11683–11688. DOI: 10.1016/j.ifacol.2017.08.1684.
- [45] Thomas Konrad, Thomas Engelhardt und Dirk Abel. „Propeller thrust identification and calibration for high-precision control of a quadrotor unmanned aerial vehicle“. In: *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017, Seiten 1225–1230. DOI: 10.1109/MED.2017.7984285.
- [46] Thomas Konrad, Tobias Salesch und Dirk Abel. „Flatness-based model predictive trajectory optimization for inspection tasks of multirotors“. In: *2019 American Control Conference (ACC)*. IEEE, 2019, Seiten 2264–2270. DOI: 10.23919/ACC.2019.8815191.
- [47] Lothar Kurz, Hendrik Langner und Tobias G. Noll. „Performance Analysis of Advanced GNSS Position Estimation Algorithms“. In: *European Navigation Conference (ENC) 2011, London, 29.11. - 01.12.2011*. 2011.
- [48] Jean Lévine. „On flatness necessary and sufficient conditions“. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*, Band 37, Nummer 13 (2004), Seiten 123–128. DOI: 10.1016/S1474-6670(17)31210-7.
- [49] LORD MicroStrain. *3DM-GX3-25 Coning and Sculling, TN-I0019, Rev 2.0*. Technischer Bericht. 2010.
- [50] LORD MicroStrain. *3DM-GX3-25-OEM*. 2014. URL: <https://www.microstrain.com/inertial/3DM-GX3-25-OEM> (besucht am 21.05.2020).
- [51] Sergei Lupashin, Markus Hehn, Mark W. Mueller, Angela P. Schoellig, Michael Sherback und Raffaello D’Andrea. „A platform for aerial robotics research and demonstration: The Flying Machine Arena“. In: *Mechatronics*, Band 24, Nummer 1 (2014), Seiten 41–54. DOI: 10.1016/j.mechatronics.2013.11.006.

- [52] Sergei Lupashin, Angela Schöllig, Michael Sherback und Raffaello D’Andrea. „A simple learning strategy for high-speed quadcopter multi-flips“. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2010, Seiten 1642–1648. DOI: 10.1109/ROBOT.2010.5509452.
- [53] Jan M. Maciejowski. *Predictive Control with Constraints*. Pearson Education Limited, 2002. ISBN: 0-201-39823-0 PPR.
- [54] Robert Mahony, Vijay Kumar und Peter Corke. „Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor“. In: *IEEE Robotics & Automation Magazine*, Band 19, Nummer 3 (2012), Seiten 20–32. DOI: 10.1109/MRA.2012.2206474.
- [55] Daniel W. Mellinger. „Trajectory generation and control for quadrotors“. Ph.D. Thesis. Publicly Accessible Penn Dissertations, 2012. URL: <http://repository.upenn.edu/edissertations/547/>.
- [56] Daniel W. Mellinger und Vijay Kumar. „Minimum snap trajectory generation and control for quadrotors“. In: *IEEE International Conference on Robotics and Automation*. IEEE, 2011, Seiten 2520–2525. DOI: 10.1109/ICRA.2011.5980409.
- [57] Thomas Meurer. „Exakte Linearisierung und differenzielle Flachheit“. In: *Regelung nichtlinearer Systeme*. 2019. Kapitel 4, Seiten 93–140.
- [58] Thomas Meurer. *Regelung nichtlinearer Systeme*. Lecture Notes, Lehrstuhl für Regelungstechnik, Christian-Albrechts-Universität zu Kiel, 2014.
- [59] People’s Republic of China. *BeiDou Navigation Satellite System*. 2020. URL: <http://en.beidou.gov.cn> (besucht am 18. 10. 2020).
- [60] Radio Technical Commission For Maritime Services. *RTCM Standard 10403.3 for Differential GNSS (Global Navigation Satellite Systems) Services - Version 3*. 3.3. 2020.
- [61] Guilherme V. Raffo, Manuel G. Ortega und Francisco R. Rubio. „An integral predictive/nonlinear H-infinity control structure for a quadrotor helicopter“. In: *Automatica*, Band 46, Nummer 1 (2010), Seiten 29–39. DOI: 10.1016/j.automatica.2009.10.018.
- [62] Guilherme V. Raffo, Manuel G. Ortega und Francisco R. Rubio. „Backstepping/nonlinear H-infinity control for path tracking of a quadrotor unmanned aerial vehicle“. In: *2008 American Control Conference*. IEEE, 2008, Seiten 3356–3361. DOI: 10.1109/ACC.2008.4587010.
- [63] Matthias Reiter. „Environment and Tools for the Development of Navigation- and Map-based Driver Assistance and Vehicle Guidance Systems“. Dissertation. RWTH Aachen University, 2017. DOI: 10.18154/RWTH-2017-09313.
- [64] Matthias Reiter und Dirk Abel. „Two and a half carrots - A versatile and intuitive optimisation-based path-following approach for road vehicles“. In: *23rd Mediterranean Conference on Control and Automation, MED 2015*. 2015, Seiten 364–370. ISBN: 9781479999361. DOI: 10.1109/MED.2015.7158776.

- [65] Ralf Rothfuß. „Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme“. Dissertation. 1997. DOI: 10.18419/opus-4118.
- [66] Ralf Rothfuß, Joachim Rudolph und Michael Zeitz. „Flachheit: Ein neuer Zugang zur Steuerung und Regelung nichtlinearer Systeme“. In: *at - Automatisierungstechnik*, Band 45, Nummer 11 (1997), Seiten 517–525. DOI: 10.1524/auto.1997.45.11.517.
- [67] Joachim Rudolph. *Flachheitsbasierte Folgeregelung*. Lehrstuhl für Systemtheorie und Regelungstechnik, Universität des Saarlandes, 2011.
- [68] René Rütters, Björn Schäfer, Thomas Engelhardt und Dirk Abel. „Potential of a Galileo Test Environment for Rail Applications“. In: *European Navigation Conference (ENC) GNSS 2010, 19.-21.10.2010, Braunschweig*. Deutsche Gesellschaft für Ortung und Navigation e.V. (DGON), 2010.
- [69] Davide Scaramuzza. *OcamCalib: Omnidirectional Camera Calibration Toolbox for Matlab*. URL: <https://sites.google.com/site/scarabotix/ocamcalib-toolbox> (besucht am 04.04.2020).
- [70] Davide Scaramuzza, Agostino Martinelli und Roland Siegwart. „A toolbox for easily calibrating omnidirectional cameras“. In: *IEEE International Conference on Intelligent Robots and Systems* (2006), Seiten 5695–5701. DOI: 10.1109/IRoS.2006.282372.
- [71] Björn Schäfer, Thomas Engelhardt und Dirk Abel. „Galileo-Testgebiete für Schienen- und Straßenfahrzeuge“. In: *AMUS + ACMOTE 2012: Future Urban Mobility, Aachen, 13. - 14.09.2012*. RWTH Aachen University, 2012. ISBN: 978-3-88354-164-8. URL: <https://nbn-resolving.de/urn:nbn:de:kobv:109-opus-224249>.
- [72] Björn Schäfer, Davide Picchi, Thomas Engelhardt und Dirk Abel. „Multicopter unmanned aerial vehicle for automated inspection of wind turbines“. In: *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, Seiten 244–249. DOI: 10.1109/MED.2016.7536055.
- [73] Dario Schafroth, Christian Bermes, Samir Bouabdallah und Roland Siegwart. „Modeling, system identification and robust control of a coaxial micro helicopter“. In: *Control Engineering Practice*, Band 18, Nummer 7 (2010), Seiten 700–711. DOI: 10.1016/j.conengprac.2010.02.004.
- [74] Pierre O. M. Scokaert und James B. Rawlings. „Feasibility issues in linear model predictive control“. In: *AIChE Journal*, Band 45, Nummer 8 (1999), Seiten 1649–1659. DOI: 10.1002/aic.690450805.
- [75] Septentrio. *AsteRx3: GNSS Multi-frequency Receiver*. 2010.
- [76] Septentrio. *SBF Reference Guide, V1.15.1*. 2012.
- [77] Skynamic GmbH. *Skynamic - Close Range Aerial Filming*. URL: <https://www.skynamic.net> (besucht am 20.05.2020).
- [78] SZ DJI Technology Co. Ltd. *DJI - Da-Jiang Innovations Science and Technology Co.* URL: <https://www.dji.com/de> (besucht am 25.05.2020).

- [79] TARC. *Test and Assessment Research Center of China Satellite Navigation Office*. 2020. URL: <http://www.csno-tarc.cn/en> (besucht am 18.10.2020).
- [80] Emrah Tasdemir, Lothar Kurz und Tobias G. Noll. „Efficient computation of the position, velocity and time estimation algorithm on a GNSS processor using interpolation“. In: *6th ESA Workshop on Satellite Navigation Technologies NAVITEC 2012*. IEEE, 2012. DOI: 10.1109/NAVITEC.2012.6423061.
- [81] Trimble Inc. *GNSS Planning Online*. 2020. URL: <https://www.gnssplanning.com> (besucht am 25.05.2020).
- [82] U.S. Coast Guard Navigation Center. *Navigation Center*. 2020. URL: <https://www.navcen.uscg.gov> (besucht am 18.10.2020).
- [83] Steven L. Waslander, Gabriel M. Hoffmann und Claire J. Tomlin. „Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, Seiten 3712–3717. DOI: 10.1109/IRoS.2005.1545025.
- [84] Jan Wendel. *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. 2. München: Oldenbourg Wissenschaftsverlag GmbH, 2011. ISBN: 978-3-486-70439-6.
- [85] Wild Rabbit Aerial Productions. *Wild Rabbit Aerial*. URL: <http://www.wildrabbitaerial.com> (besucht am 25.05.2020).
- [86] Oliver J. Woodman. *An introduction to inertial navigation*. Technischer Bericht. University of Cambridge, Computer Laboratory, 2007. URL: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>.
- [87] René Zweigel, Matthias Hoppe, Thomas Engelhardt, Frank-Josef Heßeler, Oliver Funke, René Kleeßen und Dirk Abel. „Galileo-Testzentren GATEs in Deutschland: Fit für Europas zukünftiges Satellitennavigationssystem Galileo“. In: *ZfV - Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, Band 141, Nummer 4 (2016), Seiten 223–230. DOI: 10.12902/zfv-0124-2016.