

Modellbasierte Quadrocopter-Navigation mit Laserstützung

Zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

von der Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Dipl.-Ing. Philipp Crocoll

geb. in Leonberg

Tag der mündlichen Prüfung:

21. April 2015

Hauptreferent:

Prof. Dr.-Ing. Gert F. Trommer

Korreferent:

Prof. Dr.-Ing. Peter Vörsmann

Karlsruhe 2015

Inhalt

1	Einleitung	1
1.1	Motivation und Zielsetzung	1
1.2	Übersicht über die Arbeit	3
1.3	Bezug zu wissenschaftlichen Veröffentlichungen	5
1.4	Aufklärungsplattform AirQuad	6
1.4.1	Einordnung der Sensoren und Rechenleistung	7
2	Grundlagen der Navigation	9
2.1	Inertiale Navigation	9
2.1.1	Koordinatensysteme	9
2.1.2	Komponenten eines INS	10
2.1.3	Eigenschaften eines INS	12
2.2	Integrierte Navigation	12
2.2.1	Das Kalman-Filter	12
2.2.2	Multisensorielle Navigation mit Low-Cost-Sensoren	15
3	Bewegungsmodell für Quadrocopter	17
3.1	Funktionsprinzip AirQuad	17
3.2	Aktuatormodellierung	19
3.2.1	Drehzahl und Drehmoment	19
3.2.2	Auftriebskräfte	22
3.3	Strömungswiderstand	24

3.4	Translatorisches Modell	24
3.4.1	Identifikation der translatorischen Modellparameter	24
3.5	Rotatorisches Modell	28
4	Simulation	31
4.1	Softwarearchitektur	32
4.1.1	Grobstruktur der Simulation	32
4.1.2	Postprocessing	34
4.1.3	Schnittstellenentwurf	35
4.2	Simulation des Fluggerätes	36
4.2.1	Windeinfluss	36
4.3	Sensorsimulation	37
4.4	Validierung	39
4.4.1	Validierung des Simulationsmodells	40
4.4.2	Fazit	42
5	Navigation basierend auf Relativmessungen	45
5.1	Verwandte Arbeiten	46
5.1.1	Fazit	48
5.2	Posenänderungen als Messung im Kalman-Filter	48
5.2.1	Möglichkeiten der Messwertverarbeitung	50
5.2.2	Vergleich der Verarbeitungsmethoden	55
5.2.3	Verarbeitung bei verzögertem Eintreffen der Messwerte	60
5.3	Lasergestütztes Navigationssystem für ein unbemanntes Fluggerät	67
5.3.1	Scan-Matching	68
5.3.2	Verarbeitung der 2D-Posenänderungen	71
5.3.3	Verarbeitung der Höhe-über-Grund-Messungen	72
5.3.4	Ergebnisse	74
5.4	Fehlerreduktion durch Ortswiedererkennung	75

5.4.1	Ortswiedererkennung	76
5.4.2	Navigationsstützung	79
5.4.3	Ergebnisse	83
5.5	Kalman-Smoothing mit Relativmessungen	85
5.5.1	Algorithmen für Smoothing über feste Intervalle	86
5.5.2	RTS-Smoothing mit Relativmessungen	88
5.5.3	Berücksichtigung von Ortswiedererkennungen mit PRC-Smoothing	90
5.5.4	Ergebnisse	91
5.6	Fazit	91
6	Modellgestützte Navigation	93
6.1	Verwandte Arbeiten	94
6.1.1	Modellgestützte Navigation und Multiple-Model-Ansätze	94
6.1.2	Modellgestützte Navigation für Bodenfahrzeuge	94
6.1.3	Generelle Ansätze zur modellgestützten Navigation	95
6.1.4	Modellgestützte Navigation für Helikopter	97
6.1.5	Parameterschätzung für Quadropter aus Flugdaten	98
6.1.6	Windschätzung im Flug	98
6.1.7	Fazit	99
6.2	Unified Model: Methode zur Fusion von Propagationsmodellen	100
6.2.1	Kombinierte Filterstruktur	101
6.2.2	Zusammenfassen von Propagation und Pseudomessung	105
6.2.3	Reduktion des Zustandsvektors	107
6.2.4	Unified-Model-Gleichungen	108
6.3	Minimale modellgestützte Navigation für Quadropter	112
6.3.1	Prädiktionsmodelle	112
6.3.2	Fusion der Modelle	118
6.3.3	Theoretische Analyse	120

6.3.4	Experimentelle Analyse	125
6.3.5	Fazit	133
6.4	Modellgestützte Quadropter-Navigation mit Schätzung der Modellparameter	133
6.4.1	Methoden der Online-Parameterschätzung	134
6.4.2	Experimentelle Ergebnisse	140
6.5	Kompensation des Windeinflusses	145
6.5.1	Windschätzung im modellgestützten Navigationssystem	146
6.5.2	Eigenschaften des Navigationssystems mit Windschätzung	149
6.5.3	Experimentelle Analyse	149
6.5.4	Fazit zur Windkompensation	154
6.6	Fazit	154
7	Zusammenfassung	155
A	Herleitung der Unified-Model-Berechnung	159
B	Herleitung des Systemmodells mit v_{eb}^b	161
B.1	Geschwindigkeitsdifferentialgleichung für INS	161
B.1.1	Linearisierung	162
B.2	Positionsdifferentialgleichung	164
B.2.1	Linearisierung	164
	Literatur	165

Symbolverzeichnis

Verwendete Abkürzungen

ARM	Advanced RISC Machines
ARW	Angular Random Walk
CKF	Constrained-Kalman-Filter
EKF	Extended Kalman-Filter
FLIRT	Fast Laser Interest Region Transform
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICP	Iterative Closest Point
IIR	Infinite Impulse Response
IMM	Interacting Multiple Models
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LASER	Light Amplification by Stimulated Emission of Radiation
LIDAR	Light detection and ranging
MEMS	Micro-Electro-Mechanical System
MMAE	Multiple Model Adaptive Estimation
PRC	Place Recognition Corrections
RAM	Random-Access Memory (Arbeitsspeicher)
RANSAC	Random Sample Consensus
RISC	Reduced Instruction Set Computing
RMS	Root Mean Square
SLAM	Simultaneous Localization and Mapping
UKF	Unscented Kalman-Filter
WGS 84	World Geodetic System 1984
VDM	Vehicle Dynamics Model, also das Bewegungsmodell

Formelzeichen

$\mathbf{a}_{\alpha\beta}^\gamma$	Beschleunigung zwischen den Systemen α und β , in γ -Koordinaten
\mathbf{b}_a	Beschleunigungsmesserbiase
\mathbf{b}_w	Drehratensensorbiase
\mathbf{C}_β^α	Rotationsmatrix, zur Drehung eines Vektors von β -Koordinaten nach α -Koordinaten
\mathbf{C}_k	Smoother-Gain-Matrix
$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$	Systemfunktion
\mathbf{f}_{ib}^γ	Specific Force in γ -Koordinaten
$\mathbf{f}_{m,i}^b$	Auftriebskraft von Motor i in b-frame-Koordinaten
\mathbf{g}_b^n	Schwerebeschleunigung eines Körpers in n-frame-Koordinaten
$\mathbf{h}(\mathbf{x}_k)$	Messfunktion
\mathbf{H}	Messmatrix
\mathbf{K}_k	Kalman-Gain-Matrix
\mathbf{p}^γ	Position in Gamma-Koordinaten
\mathbf{P}_k	Kovarianzmatrix des Zustands \mathbf{x}_k oder Fehlers $\delta\mathbf{x}_k$
\mathbf{P}_k	Pose im Posengraph
\mathbf{q}	Lagequaternion
\mathbf{Q}_k	Kovarianzmatrix des Systemrauschens \mathbf{w}_k oder Varianz einer Pose im Posengraphen
\mathbf{R}_k	Kovarianzmatrix des Messrauschens \mathbf{v}_k
\mathbf{v}_k	Messrauschen
\mathbf{w}_k	Systemrauschen
$\mathbf{v}_{\alpha\beta}^\gamma$	Geschwindigkeit zwischen den Systemen α und β , in γ -Koordinaten
\mathbf{x}_k	Zustand zum Zeitpunkt k
$\delta\mathbf{x}_k$	Fehlervektor zum Zustand \mathbf{x}_k
$\delta\psi$	Lagewinkelfehler als Eulerwinkel
$\Delta\mathbf{P}(e_j)$	Posenänderung der Kante e_j im Posengraph
$\Delta\mathbf{Q}(e_j)$	Varianz der Posenänderung von Kante e_j im Posengraph

γ^γ	Beschleunigung aufgrund von Massenanziehung in γ -Koordinaten
Φ_k	Systemmatrix
ω_m	Motordrehzahl
$\omega_{\alpha\beta}^\gamma$	Drehrate zwischen den Systemen α und β , in γ -Koordinaten
$\Omega_{\alpha\beta}^\gamma$	Kreuzproduktbildende Matrix zu $\omega_{\alpha\beta}^\gamma$

Indizes und Sonderzeichen

$\hat{\square}$	Geschätzte Größe
$\tilde{\square}$	Gemessene Größe
\square^i	Größe angegeben im Inertialkoordinatensystem
\square^e	Größe angegeben im erdfesten, erdzentrierten Koordinatensystem
\square^b	Größe angegeben im körperfesten Koordinatensystem
\square^n	Größe angegeben im Navigationskoordinatensystem
$[\mathbf{a}\times]$	Kreuzproduktbildende Matrix zum 3×1 -Vektor \mathbf{a}

Kapitel 1

Einleitung

Kleine unbemannte Helikopter wurden in den vergangenen Jahren ständig weiterentwickelt und stehen inzwischen in verschiedensten Varianten zur Verfügung. Ihr Einsatz dient bereits heute vielfältigen Anwendungen, von der Fotografie, Anlagen- und Verkehrsüberwachung über Sicherheitsaufgaben bis hin zum Transport von Medikamenten. Insbesondere Quadrocopter und verwandte Gerätetypen finden hohe Verbreitung, da sie durch ihren einfachen Aufbau einen hohen Grad an mechanischer Robustheit aufweisen.

Je häufiger die Fluggeräte eingesetzt werden, desto wichtiger ist ihre Zuverlässigkeit während des Betriebs, um die Sicherheit von Menschen nicht zu gefährden. Gleichzeitig sind viele Anwendungen darauf angewiesen, dass ein Fluggerät auch außerhalb der Sichtweite des Operators eigenständig agieren kann, z. B. beim Einsatz als Transportdrohne oder bei der Suche nach Vermissten in einem größeren Gebiet. Da Funkverbindungen ausfallen können, benötigen unbemannte Helikopter autonome Fähigkeiten, z. B. um selbstständig zum Ausgangspunkt bzw. Ziel zu gelangen, ohne dabei eine Kollision zu verursachen.

Eine zentrale Anforderung, um autonomes Verhalten von Fluggeräten realisieren zu können, ist die ständige Verfügbarkeit einer Navigationsschätzung. Das Fluggerät muss also seine Position, Geschwindigkeit und Lage im Raum bestimmen können, möglichst ohne sich dabei auf externe Stützinformationen zu verlassen.

1.1 Motivation und Zielsetzung

Bisher eingesetzte Navigationssysteme sind meist stark abhängig von der Verfügbarkeit von GPS (Global Positioning System) oder anderen Satellitennavigationssystemen

(GNSS, engl. Global Navigation Satellite System): Zwar existiert i. d. R. ein Inertiales Navigationssystem (INS) basierend auf der Inertialsensorik an Bord des Fluggerätes, aber dieses kann schon nach wenigen Sekunden keine genaue Positionsschätzung mehr liefern, selbst wenn es mit Zusatzsensoren wie Baro-Altimeter oder Magnetometer kombiniert wird. Da Satellitennavigationssysteme mit sehr günstigen und leichten Geräten, sog. Jammern, gestört und lokal praktisch deaktiviert werden können, muss unbedingt eine zuverlässige Alternative gefunden werden.

Ein bildgestütztes Navigationssystem, wie es z. B. in [32] entwickelt wurde, kann eine Lösung darstellen, erfordert jedoch neben der Montage der Kamera hohe Onboard-Rechenleistung, die nicht immer verfügbar ist. Zudem ist die Navigationsschätzung nur möglich, wenn ausreichende Beleuchtung vorhanden ist, was gerade in Notfallsituationen in Gebäuden oder nachts nicht gewährleistet ist.

In dieser Arbeit sollen Methoden entwickelt, implementiert und untersucht werden, die auch unter ungünstigen äußeren Bedingungen zur Navigation für Quadropter geeignet sind. Diese sollen in die am Institut für Theoretische Elektrotechnik und Systemoptimierung (ITE) am Karlsruher Institut für Technologie (KIT) entwickelte Aufklärungsplattform AirQuad integriert werden.

Zum einen soll ein lasergestütztes Navigationssystem aufgebaut werden. Hierbei werden die Messungen eines Laserscanners, der Abstandsmessungen in verschiedene Raumrichtungen durchführt, so aufbereitet, dass sie als Navigationsstützung verwendet werden können. Laserscanner sind insofern eine geeignete Quelle für Stützinformationen, da sie für die autonome Hindernisvermeidung ohnehin auf einigen Plattformen vorhanden sind. Als aktive Sensoren sind sie nicht von Beleuchtungsbedingungen abhängig und sie liefern Messwerte insbesondere in kritischen Situationen, z. B. in Gebäuden oder in Nähe von Hindernissen.

Für das lasergestützte Navigationssystem soll in dieser Arbeit der Fokus darauf liegen, eine vollständige Navigationsschätzung liefern zu können, also neben der Position auch Geschwindigkeit und Lage. Diese soll an Bord des Fluggerätes in Echtzeit berechnet werden, und zwar mit der Messfrequenz der Inertialsensoren (ca. 300 Hz). Nur so können die berechneten Größen in der Regelschleife des Fluggerätes genutzt werden. Diese Anforderungen werden von vielen SLAM-Ansätzen (Simultaneous Localization and Mapping) nicht erfüllt, da diese nur eine horizontale Positionsschätzung berechnen, aktualisiert mit jeder Lasermessung. Die

Updaterate beträgt somit nur etwa 10 Hz–40 Hz, zudem sind zuverlässige SLAM-Ansätze nicht echtzeitfähig auf kleinen Embedded-Rechensystemen.

Das lasergestützte Navigationssystem soll außerdem dahin gehend weiter entwickelt werden, dass erkannt wird, wenn an bereits besuchte Orte zurückgekehrt wird. Damit soll dann die Positionsschätzung korrigiert werden. Dies ist für rechenstarke Plattformen oder Bodenstationen möglich. Außerdem soll so ein System geschaffen werden, das Operatoren nach dem Flug eine präzise Darstellung der Flugtrajektorie bietet, so dass unterwegs gesammelte Informationen genau verortet werden können.

In manchen Anwendungen ist der Einsatz eines Laserscanners zu teuer oder würde keinen Nutzen bringen, z. B. beim Medikamententransport über Wasser. Gerade hier ist es aber von zentraler Bedeutung, dass das Fluggerät selbstständig einen Landeplatz erreichen kann. Daher soll ein Navigationssystem entwickelt werden, das den Ausfall von Laser- oder Satellitenstützung dadurch überbrücken kann, dass Informationen über das Bewegungsmodell des Fluggerätes in der Navigation verwendet werden. Dies ist eine in der Vergangenheit meist nur theoretisch untersuchte Möglichkeit. Erste Simulationen mit Flugzeugmodellen wiesen zwar vielversprechende Ergebnisse auf, führten aber auch zu einer Reihe von Fragen, wie beispielsweise: Lässt sich dieser Grundgedanke auf Quadropter anwenden? Wie können diese Informationen optimal ins Navigationssystem integriert werden? Können zur Simulation vergleichbare Ergebnisse auch in realen Experimenten erzielt werden?

Diese Fragen sollen in der vorliegenden Arbeit umfassend beantwortet werden, indem ein geeignetes Navigationssystem für Quadropter entworfen, implementiert und in Flugtests evaluiert wird. Dieses soll zudem in der Lage sein, auf Änderungen am Fluggerät automatisch zu reagieren, z. B. wenn zusätzliche Nutzlast angebracht wird. Auch bei Wind, der Einfluss auf das Bewegungsmodell hat, soll die Funktionsfähigkeit des Navigationssystems erhalten bleiben.

1.2 Übersicht über die Arbeit

In dieser Arbeit werden zunächst grundlegende Definitionen erläutert und verwendete Algorithmen beschrieben (Kapitel 2).

In Kapitel 3 wird ein Bewegungsmodell für Quadropter beschrieben, das später sowohl für die Simulation der Fluggeräte als auch im bewegungsmodellgestützten Navigationssystem genutzt wird. Dieses Bewegungsmodell hat – im Vergleich zu detaillierten Flugzeug- oder Helikoptermodellen – eine eher geringe Komplexität, da

die im gewöhnlichen und autonomen Flug eingenommenen Flugzustände gut alleine durch Motorkräfte und Strömungswiderstand beschrieben werden können.

Für die Fluggeräte wurde eine Simulationsumgebung entwickelt, die in Kapitel 4 beschrieben wird. Diese Simulationsumgebung erlaubt die synthetische Generierung realistischer Flugtrajektorien und den Vergleich beliebiger Navigationsansätze. Sie dient gleichzeitig zum Software-In-The-Loop-Test aller im realen Flug eingesetzten Softwarekomponenten.

Kapitel 5 stellt die Entwicklung eines lasergestützten Navigationssystems vor. Hierzu wird zunächst simulativ untersucht, wie Relativmessungen möglichst gewinnbringend in einem Kalman-Filter verarbeitet werden können. Die dabei ausgewählte Methode wird erweitert, so dass auch verzögert eintreffende Relativmessungen verarbeitet werden können. Basierend auf diesen allgemeinen Betrachtungen wird dann ein konkretes Navigationssystem für den Online-Einsatz auf den AirQuad-Plattformen entwickelt, das Relativmessungen auch zur Stützung des Heading-Winkels und der Höhe nutzt (Unterkapitel 5.3). Eine Möglichkeit, die Positionsschätzung weiter zu verbessern, wird in Unterkapitel 5.4 beschrieben. Hierzu werden Ortswiedererkennungen ins Navigationssystem integriert, ohne die hohe Updaterate des Basissystems aufgeben zu müssen. Schließlich wird ein Smoothing-Ansatz entwickelt, der nach dem Flug eine Navigationsschätzung liefert, bei der die Ortswiedererkennungen auch rückwirkend ausgenutzt werden.

Kapitel 6 befasst sich mit Methoden zur bewegungsmodellgestützten Navigation. Nach einer Diskussion bestehender Ansätze wird zunächst das sog. Unified Model entwickelt. Hierbei handelt es sich um einen Ansatz, mit dem in einem Kalman-Filter zwei gleichermaßen gültige Prädiktionsmodelle zu einem Prädiktionsmodell vereinigt werden können. Dies ist notwendig, um das klassische Inertiale Navigationssystem mit dem Bewegungsmodell zu fusionieren (Unterkapitel 6.2). Das allgemein gültige Unified-Model-Verfahren wird zur Entwicklung eines Quadropter-Navigationssystems verwendet, was zu einem minimalistischen System führt, dessen Zustandsvektor durch die Modellstützung nicht anwächst. Dennoch weist es auch in realen Flügen gute Ergebnisse auf (Unterkapitel 6.3). Dieses Navigationssystem wird für den realen Einsatz mit veränderlichen Modellparametern (Unterkapitel 6.4) und Wind (Unterkapitel 6.5) erweitert.

Die Arbeit schließt mit einer Zusammenfassung der entwickelten Methoden und der erreichten Ergebnisse. Eine Diskussion zum Stand der Technik ist in den jeweiligen Kapiteln enthalten.

1.3 Bezug zu wissenschaftlichen Veröffentlichungen

Wesentliche Inhalte der vorliegenden Arbeit wurden bereits in wissenschaftlichen Fachzeitschriften oder auf wissenschaftlichen Konferenzen und Workshops veröffentlicht. Um eine durchgängige Darstellung in dieser Arbeit zu erreichen, sind diese Veröffentlichungen jedoch nicht immer als je ein Abschnitt oder Kapitel enthalten, sondern sind textuell wie folgt neu organisiert:

Das auf der European Navigation Conference vorgestellte [103] und im European Journal of Navigation erschienene Paper zu lasergestützter Navigation [105] ist in den Betrachtungen in den Unterkapiteln 5.2 und 5.3 wiedergegeben; Abschnitt 5.2.1.5 wurde aber noch ergänzt – dieser entspricht einer Beschreibung in [98]. Zudem wurden die Simulationen in Unterkapitel 5.2 neu durchgeführt, um diese Ergänzung zu berücksichtigen und eine modernere Inertiale Messeinheit (IMU, engl. Inertial Measurement Unit) repräsentieren zu können.

[98] führte zudem die Verarbeitung der Ortswiedererkennungen ein, die in dieser Arbeit im Unterkapitel 5.4 beschrieben wird.

Die allgemeine Unified-Model-Methode wurde in einer gemeinschaftlichen Arbeit mit dem Lehrstuhl für Flugsystemdynamik an der TU München entwickelt. Die Vorstellung fand zuerst auf dem ION International Technical Meeting 2013 statt [99], später erschien das Paper als gemeinschaftliche Veröffentlichung im Journal of Navigation [100]. In dieser Arbeit sind nur die Teile dieser Veröffentlichungen enthalten, an deren Entwicklung der Autor maßgeblich beteiligt war; ansonsten wird auf die Veröffentlichungen verwiesen. Der Inhalt ist in die Abschnitte 6.1.1 bis 6.1.3 sowie Unterkapitel 6.2 eingeflossen.

Die Anwendung dieser Methode auf Quadrocopter wurde auf dem ION International Technical Meeting 2014 präsentiert [107] und ebenfalls später im Journal of Navigation veröffentlicht [106]. Der Inhalt dieser Paper ist in Unterkapitel 6.3 beschrieben. Die experimentellen Ergebnisse wurden jedoch mit einer anderen IMU erstellt. Die nun verwendete IMU entspricht der aus Kapitel 5 und Unterkapitel 6.4. Zudem basiert die Beschreibung des translatorischen Quadrocoptermodells (Unterkapitel 3.4) auf diesen Veröffentlichungen.

Die Schätzung der Modellparameter während des Fluges (Unterkapitel 6.4) wurde in [109] vorgestellt, ist hier jedoch erweitert um eine Möglichkeit, mit der auf die Messung der Motordrehzahlen verzichtet werden kann. Diese ist erst in der zur Veröffentlichung eingereichten Paper [108] enthalten.

1.4 Aufklärungsplattform AirQuad

Am Institut für Theoretische Elektrotechnik und Systemoptimierung (ITE) des Karlsruher Institut für Technologie (KIT) entstand die Aufklärungsplattform „AirQuad“. Sie umfasst mehrere Fluggeräte, für die in früheren Arbeiten [60, 73] geeignete Hardware ausgewählt und verschiedene Steuerungs- und Navigationsalgorithmen entwickelt wurden; zudem wurde das System um einen Bodenroboter und eine Bodenstationssoftware erweitert [32, 73]. Mit dem AirQuad-Projekt soll es möglich sein, in Notfallszenarien schnell und einfach einen Überblick der Lage zu erhalten, wozu die Fluggeräte mit autonomen Fähigkeiten ausgestattet werden.

Die in dieser Arbeit entwickelten Algorithmen sollen die robuste Navigation der AirQuad-Fluggeräte sicherstellen. Gleichzeitig dienen die Fluggeräte zur experimentellen Evaluation der neuen Methoden.

In dieser Arbeit wurden zwei verschiedene Fluggeräte genutzt. Beide sind elektrisch angetrieben und mit dem selben „AirQuad-NavBoard“ ausgestattet. Ein auf diesem angebrachter Microcontroller liest verschiedene Sensoren aus, nimmt grundlegende Navigations- und Reglerfunktionen wahr und steuert so die Motoren an. Die verfügbaren Sensoren umfassen eine Inertiale Messeinheit, Barometer und Magnetometer, außerdem ein GPS-Receiver. Diese Sensoren sind in [60, 73] näher beschrieben. Die für diese Arbeit relevante IMU wird durch Tabelle 1.1 charakterisiert.

Für komplexere Navigationsaufgaben kommen weitere Sensoren und eine zusätzliche Recheneinheit mit Linux-Betriebssystem zum Einsatz. Die Sensoren unterscheiden sich zwischen den Fluggeräten aufgrund der unterschiedlichen Nutzlastkapazitäten. In beiden Fällen kommuniziert eine Embedded-PC-Einheit mit dem AirQuad-NavBoard. Sie empfängt sämtliche dort verfügbaren Sensorwerte und sendet ermittelte Navigationsgrößen und ggf. auch Stellgrößen für den Flugregler zurück.

Das Fluggerät „AirQuad 1“ (Abbildung 1.1 links) ist mit dem Laserscanner Hokuyo URG-04LX [95] ausgestattet. Als Rechenplattform dient ein Beagleboard [93] mit einem 720 MHz ARM-Prozessor. Es stehen 256 MB Arbeitsspeicher zur Verfügung.

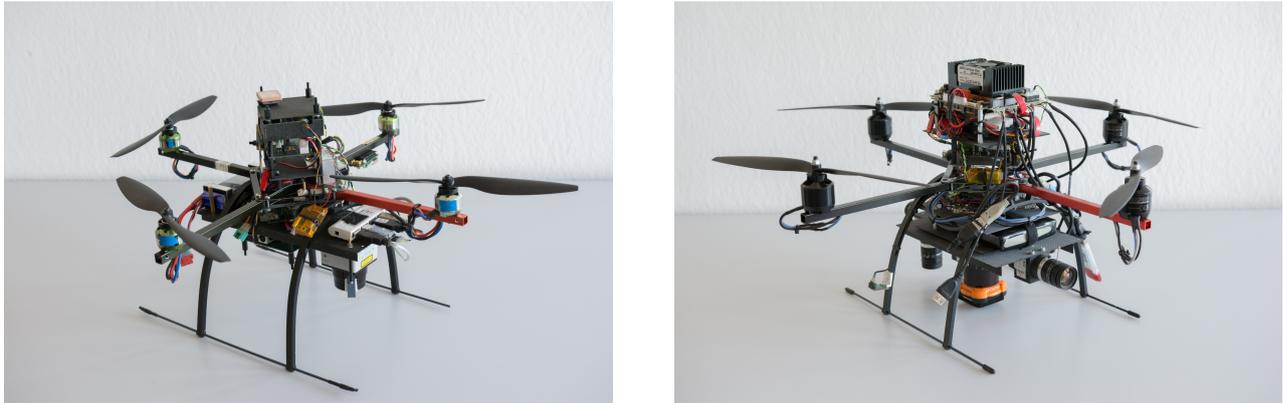


Abbildung 1.1: AirQuad 1 (links) und AirQuad 2 (rechts): Die beiden Fluggeräte unterscheiden sich in der Nutzlastkapazität, dem verfügbaren Laserscanner und der verfügbaren Rechenleistung, weisen aber dieselbe Modellstruktur auf.

Sensortyp	Bezeichnung	Wesentliche Kenngrößen
Beschleunigungs- messer	VTI SCA 3100 [97]	Dynamischer Bereich: ± 2 g Nichtlinearität: 1% des Messbereichs (bis 1 g) In-Run-Bias-Stabilität (temperaturbedingt): ± 30 mg Velocity Random Walk: $0,0658 \text{ m/s}/\sqrt{\text{h}}$ Genutzte Updaterate: 333 Hz
Drehratensensoren	Analog Devices ADIS 16265 [91]	Dynamischer Bereich: ± 320 °/s Nichtlinearität: 0,1% des Messbereichs In-Run-Bias-Stabilität: $25,2$ °/h Angular Random Walk: 2 °/ $\sqrt{\text{h}}$ Genutzte Updaterate: 333 Hz

Tabelle 1.1: Wesentliche Kenngrößen der auf dem AirQuad-NavBoard verfügbaren IMU

Eine weitere IMU vom Typ ADIS 16488A (s. Tabelle 1.2 und [92]) der Firma Analog Devices kann genutzt werden, um den Einfluss der IMU-Güte zu evaluieren.

„AirQuad 2“ (Abbildung 1.1 rechts) kann aufgrund leistungsfähigerer Motoren und größerer Rotoren weitere Nutzlast tragen. So ist es möglich, den Laserscanner Hokuyo UTM-30LX [96] mitzuführen; zur Berechnung steht ein Cool Xpressrunner-GS45 [94] zur Verfügung. Diese Rechenplattform stellt einen Intel® Core™2 Duo Prozessor mit 2,26 GHz bereit. Die Kameras an Bord von AirQuad 2 werden in dieser Arbeit nicht genutzt.

1.4.1 Einordnung der Sensoren und Rechenleistung

Die Sensoren auf dem AirQuad-NavBoard stehen für die auf Quadrokoptern typische Low-Cost-Sensorik. Wie üblich werden IMUs in MEMS-Technik (Microelectromechanical systems) verwendet, z. B. [10, 15, 42]. Die Laserscanner von Hokuyo finden

Sensortyp	Wesentliche Kenngrößen
Beschleunigungsmesser	Dynamischer Bereich: ± 18 g Nichtlinearität: 0,1% des Messbereichs (bis 10 g) In-Run-Bias-Stabilität: $\pm 0,07$ mg Velocity Random Walk: $0,029$ m/s/ \sqrt{h}
Drehratensensoren	Dynamischer Bereich: ± 480 °/s Nichtlinearität: 0,01% des Messbereichs In-Run-Bias-Stabilität: $5,1$ °/h Angular Random Walk: $0,26$ °/ \sqrt{h}

Tabelle 1.2: Wesentliche Kenngrößen der zusätzlich auf AirQuad-1 vorhandenen IMU ADIS 16488A von Analog Devices.

	Hokuyo URG-04LX	Hokuyo UTM-30LX
Max. Reichweite	4,1 m	30 m
Messbereich	240°	270°
Anzahl Messpunkte pro Scan	1024	1440

Tabelle 1.3: Kenngrößen der eingesetzten Laserscanner

vielfach in Robotikanwendungen Einsatz, aufgrund ihres geringen Gewichts gerade auch auf fliegenden Plattformen, z. B. [10, 42].

Die AirQuad-Plattform enthält also häufig eingesetzte Sensorkomponenten. Es kann somit davon ausgegangen werden, dass auf dieser Plattform evaluierte Algorithmen auch auf andere Flugsysteme übertragen werden können.

Die IMU ADIS 16488A gilt aktuell als hochwertige MEMS-IMU; sie erreicht bereits Kenngrößen sogenannter Tactical-Grade-IMUs. Trotz der recht hohen Kosten wird sie bereits in verschiedenen Projekten eingesetzt (z. B. [4, 51, 55]). Da die Entwicklung der MEMS-IMU-Technik weiter schnell voranschreitet, ist damit zu rechnen, dass in künftigen Jahren auch im Low-Cost-Bereich Sensoren mit ähnlichen Charakteristika verfügbar sein werden.

Die Rechenleistung auf dem Fluggerät AirQuad 1 ist relativ gering. Selbst ein mehrere Jahre alter Desktop-PC arbeitet typische Navigationsaufgaben ca. 20-mal schneller ab. Für diese Recheneinheit müssen also effiziente Algorithmen gefunden werden.

Die Rechenleistung auf AirQuad 2 ist deutlich höher, auch wenn sie nicht an die aktueller Desktop-Systeme heranreicht. Dennoch steht ausreichend Rechenleistung zur Verfügung, um auch komplexere Algorithmen einsetzen zu können.

Kapitel 2

Grundlagen der Navigation

In diesem Kapitel sollen einige grundlegende Begriffe und Verfahren der Navigation eingeführt werden. Zunächst wird die Inertiale Navigation beschrieben, die – geeignet genaue Sensoren vorausgesetzt – die Entwicklung eines eigenständigen Navigationssystems erlaubt. In vielen Fällen, z. B. auch im Einsatz auf Quadroptern, stehen solch hochwertige Sensoren aber nicht zur Verfügung. Dann muss ein integriertes Navigationssystem entworfen werden, das zusätzliche Messwerte nutzt, um das Inertiale Navigationssystem zu kalibrieren und sein Fehlerwachstum zu begrenzen. Dazu wird üblicherweise ein Kalman-Filter eingesetzt. Dessen Gleichungen und Eigenschaften werden in Abschnitt 2.2.1 diskutiert, bevor dann in Abschnitt 2.2.2 skizziert wird, wie basierend auf einem Inertialen Navigationssystem (INS) und dem Kalman-Filter-Algorithmus ein typisches Navigationssystem aufgebaut wird.

2.1 Inertiale Navigation

Das Grundprinzip der Inertialen Navigation ist die Bestimmung der Navigationsgrößen durch zeitliche Integration gemessener Drehraten und Beschleunigungen.

2.1.1 Koordinatensysteme

Bei der Formulierung der Gleichungen eines INS treten verschiedene Koordinatensysteme auf. Die wichtigsten sind

- das inertielle Koordinatensystem (genauer: erdzentriertes Inertialsystem, i-frame), dessen Ursprung im Zentrum der Erde liegt und dessen Achsen fest bezüglich der Fixsterne sind

- das erdfeste Koordinatensystem (e-frame), dessen Ursprung und z-Achse mit dem inertialen Koordinatensystem zusammenfallen, dessen x- und y-Achsen aber mit der Erde rotieren
- das Navigationskoordinatensystem (n-frame), dessen Ursprung im Ursprung des Fahrzeugs liegt; die Achsen sind nach Norden, Osten und Unten ausgerichtet
- das körperfeste Koordinatensystem (b-frame, von engl. body-frame), dessen Ursprung mit dem n-frame übereinstimmt, dessen Achsen aber in die Fahrzeugrichtungen Vorne, Rechts und Unten zeigen.

Ausführlichere Erläuterungen und Definitionen weiterer Systeme sind vielfach in der Literatur zu finden, z. B. [36, 80, 87].

2.1.2 Komponenten eines INS

Zum Aufbau eines INS sind folgende Komponenten erforderlich:

2.1.2.1 Sensoren

Zur Messung der Drehraten können hochwertige Laserkreisel, Faserkreisel oder auch günstige MEMS-Gyroskope eingesetzt werden. Beschleunigungen werden bestimmt, indem die Auslenkung einer Probemasse gemessen wird, z. B. in Vibrating-Beam- oder Pendel-Beschleunigungsmessern.

Diese Sensoren weisen verschiedene Fehler auf, u. a. Biase (ein Nullpunktsfehler beim Einschalten, überlagert von einer zeitabhängigen Drift), Skalenfaktorfehler und Messrauschen. Die Größe dieser Fehlerterme bestimmt wesentlich die Güte der Sensoren.

Bei Verwendung von Beschleunigungsmessern ist zu beachten, dass diese – anders als der Name suggeriert – keine tatsächlichen Beschleunigungen messen. Korrekter ist es, hier von „Specific Force“ zu sprechen. Dies ist die Beschleunigung des Körpers abzüglich der Gravitationsbeschleunigung; diese wird von den Beschleunigungsmessern nicht erfasst. Für ein am Boden ruhendes Objekt oder ein Fluggerät auf konstanter Höhe wird also eine Specific Force gemessen, die ca. $9,8 \text{ m/s}^2$ nach oben weist. Dies entspricht der vom Boden auf das Objekt ausgeübten Gegenkraft bzw. den aerodynamischen Kräften, allerdings normalisiert auf die jeweilige Masse, so dass sich wieder die Einheit einer Beschleunigung ergibt.

2.1.2.2 Erdmodell

Zur Umrechnung zwischen dem Navigationskoordinatensystem und erdfesten oder inertialen Koordinaten müssen die Parameter des Erdellipsoiden bekannt sein. Das sog. WGS-84-Modell, das ursprünglich für das U. S.-Militär entwickelt wurde, hat sich als Standard in der Navigation etabliert, nicht zuletzt da es auch im GPS Verwendung findet.

Um die gemessene Specific Force in eine tatsächliche Beschleunigung transformieren zu können, wird zudem ein Gravitationsmodell benötigt. Die Gravitation γ weist eine komplexe Ortsabhängigkeit auf, da sie u. a. vom Höhenprofil der Erde abhängig ist. Für die im Folgenden betrachteten Navigationsanwendungen ist es jedoch üblich und ausreichend, ein approximatives Modell zu verwenden. In dieser Arbeit wird das weit verbreitete Somigliana-Modell [36] verwendet; die Höhenabhängigkeit wird gemäß [2] ausgedrückt. Mit diesen Modellen, für die hier die WGS84-Parameter genutzt werden, erhält man die Schwerebeschleunigung an einem bestimmten Ort. Die Gravitation wird daraus durch Subtraktion der Zentrifugalbeschleunigung ermittelt.

2.1.2.3 Strapdown-Algorithmus

In heutigen Systemen sind die Inertialsensoren meist körperfest montiert und messen die Beschleunigungen und Drehraten des b-frames gegenüber dem i-frame. Gesucht sind die Navigationsgrößen oft in Navigationskoordinaten.

Der Strapdown-Algorithmus beschreibt die hierfür notwendigen Rechenschritte. Diese lassen sich basierend auf den Differentialgleichungen

$$\dot{\mathbf{C}}_b^i = \mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b = \mathbf{C}_b^i [\boldsymbol{\omega}_{ib}^b \times] \quad (2.1)$$

$$\dot{\mathbf{v}}_{ib}^i = \mathbf{a}_{ib}^i = \mathbf{f}_{ib}^i - \boldsymbol{\gamma}^i \quad (2.2)$$

herleiten, indem man sie in das gewünschte Koordinatensystem transformiert, so dass z. B. Differentialgleichungen für $\dot{\mathbf{C}}_b^n$ und $\dot{\mathbf{v}}_{eb}^n$ resultieren, also für die Drehmatrix vom b- ins n-frame und die Geschwindigkeit des Körpers gegenüber dem erdfesten Koordinatensystem. Solche Herleitungen werden in [36, 80] oder anderen Grundlagenbüchern für verschiedene Mechanisierungen präsentiert.

2.1.3 Eigenschaften eines INS

Ein Inertiales Navigationssystem ist ein Koppelnavigationssystem: Basierend auf der aktuellen Schätzung wird durch Zeitintegration die Schätzung für den nächsten Zeitpunkt berechnet. Dies führt zu einem mit der Zeit anwachsenden Fehler: Ein Nullpunktsfehler der Drehratensensoren führt zu einem linear mit der Zeit wachsenden Lagefehler. Durch die deswegen fehlerhafte Kompensation der Gravitation in den Beschleunigungsmesswerten entsteht ein linear wachsender Beschleunigungsfehler, der nach zweifacher Integration zu einem $\sim t^3$ wachsenden Positionsfehler führt.

Zwar ändert sich das langfristige Fehlerwachstum durch Effekte wie Schuleroszillationen oder die Höhenabhängigkeit der Gravitation, diese Effekte haben aber für kurze Flüge, wie sie bei Quadroptern üblich sind, keine praktische Relevanz.

Inertiale Navigationssysteme können in Flugzeugen nach geeigneter Kalibrierung eigenständig über mehrere Stunden arbeiten und können Fehler von weniger als einer nautischen Meile pro Stunde aufweisen. Dies erfordert aber hochwertige Sensoren, die sowohl aus Kosten- als auch aus Gewichtsgründen nicht auf Quadroptern eingesetzt werden. Bei den üblicherweise verfügbaren Sensoren liegt der Positionsfehler bereits nach ca. einer Minute im Bereich von hunderten Metern. Gleichzeitig erfordert der Einsatz nahe an Gebäuden eine deutlich höhere Positionsgenauigkeit als in Flugzeugen auf transatlantischen Routen. Aus diesen Gründen ist es notwendig, das Inertiale Navigationssystem so zu erweitern, dass mit Hilfe zusätzlicher Stützsensoren der Fehler reduziert werden kann.

2.2 Integrierte Navigation

Die Erweiterung des Inertialen Navigationssystems wird üblicherweise mit Hilfe stochastischer Filtertechniken vorgenommen, wobei für die überwiegende Mehrheit an Navigationsproblemen ein Kalman-Filter zum Einsatz kommt.

2.2.1 Das Kalman-Filter

Das von Rudolf Kálmán [48] formulierte und nach ihm benannte Kalman-Filter beschreibt einen Algorithmus, um den Zustand eines Systems zu schätzen, zu dem ein Systemmodell und Messungen vorliegen. Dies ist in der Navigation der Fall: Das Inertiale Navigationssystem kann als Systemmodell genutzt werden, das nun mit Hilfe zusätzlicher Messgrößen gestützt wird. Der Einsatz des Kalman-Filters bietet sich

an, da es sich dabei um ein Optimalfilter handelt, das den quadratischen Schätzfehler minimiert.

Das Kalman-Filter schätzt nicht nur den erwarteten Zustand des Systems, sondern auch die – als Normalverteilung modellierte – Wahrscheinlichkeitsverteilung des Zustands. Eine Normalverteilung ist vollständig über die ersten beiden Momente beschrieben, daher werden der Erwartungswert und die Kovarianzmatrix geschätzt. Mittels der Kovarianzmatrix kann das Filter die Schätzunsicherheit angeben.

Bei der Herleitung des Kalman-Filters geht man davon aus, dass das Systemrauschen \mathbf{w}_k und Messrauschen \mathbf{v}_k mittelwertfreies, weißes gaußsches Rauschen sind und dass sie untereinander und mit $\hat{\mathbf{x}}_k$ unkorreliert sind. Es muss also gelten:

$$E \{ \mathbf{w}_{k_1} \mathbf{w}_{k_2}^T \} = \begin{cases} \mathbf{Q}_k & k_1 = k_2 \\ \mathbf{0} & \text{sonst} \end{cases}, E \{ \mathbf{v}_{k_1} \mathbf{v}_{k_2}^T \} = \begin{cases} \mathbf{R}_k & k_1 = k_2 \\ \mathbf{0} & \text{sonst} \end{cases} \quad (2.3)$$

$$E \{ \mathbf{w}_k \mathbf{v}_k^T \} = \mathbf{0}, E \{ \mathbf{w}_k \hat{\mathbf{x}}_k^T \} = \mathbf{0}, E \{ \mathbf{v}_k \hat{\mathbf{x}}_k \} = \mathbf{0} \quad (2.4)$$

Der Algorithmus besteht aus einem sog. Prädiktions- oder Propagationsschritt, bei dem der Zustandsvektor $\hat{\mathbf{x}}_k$ und die Kovarianzmatrix \mathbf{P}_k in der Zeit propagiert werden, und aus einem Messschritt, in dem Messungen verarbeitet werden. Diese beiden Schritte werden rekursiv durchgeführt, d. h. es muss eine Initialschätzung $\hat{\mathbf{x}}_0$ mit \mathbf{P}_0 vorliegen.

Transformiert man die Strapdown-Gleichungen in eine zeitdiskrete Formulierung, erhält man das Systemmodell mit der Funktion $\mathbf{f}()$, die den Zustand \mathbf{x}_k basierend auf den Inertialmessungen \mathbf{u}_k in der Zeit propagiert:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{G}_k \mathbf{w}_k \quad (2.5)$$

Darin bezeichnet \mathbf{w}_k einen Rauschvektor, der Sensorrauschen und ggf. unmodellierte Effekte beschreibt. \mathbf{G}_k ist eine Matrix, die den Einfluss des Rauschvektors auf den Zustandsvektor beschreibt.

Messungen seien in Form von

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (2.6)$$

gegeben, wobei $\tilde{\mathbf{y}}_k$ den Messwert darstellt; \mathbf{v}_k ist das Messrauschen.

Die Funktionen $f()$ und $h()$ sind für Navigationssysteme meist nichtlinear, daher muss eine geeignete Formulierung des Kalman-Filters gewählt werden.

2.2.1.1 Extended Kalman-Filter

Beim Extended Kalman-Filter wird eine Linearisierung des Zustands- und Messmodells um den jeweiligen Schätzwert vorgenommen, so dass eine lineare Kovarianzpropagation erfolgen kann. Die Prädiktion der Zustandsschätzung erfolgt jedoch nichtlinear:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k) \quad (2.7)$$

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (2.8)$$

Hierbei bezeichnet $\mathbf{\Phi}_k$ die Jacobi-Matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ der Systemfunktion $\mathbf{f}()$.

Messungen werden verarbeitet, indem die Messfunktion linearisiert wird: $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$. Dann wird die Zustandsschätzung wie folgt aktualisiert:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (2.9)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- - \mathbf{K}_k (\mathbf{h}(\hat{\mathbf{x}}_k^-) - \tilde{\mathbf{y}}_k) \quad (2.10)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (2.11)$$

$$= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (2.12)$$

2.2.1.2 Error-State Kalman-Filter

Alternativ zum Extended Kalman-Filter, bei dem im Zustand direkt die Navigationsgrößen stehen, besteht die Möglichkeit, eine sog. Error-State-Formulierung zu wählen. Hierbei wird ebenfalls mittels der Systemfunktion die Zustandsschätzung propagiert:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k) \quad (2.13)$$

Im Kalman-Filter selbst wird aber nun der Fehler $\delta \mathbf{x}$ von $\hat{\mathbf{x}}$ bzgl. der wahren Größe \mathbf{x} geschätzt. Es gilt $\mathbf{x}_k = \hat{\mathbf{x}}_k - \delta \mathbf{x}_k$.

Da die Propagation außerhalb des Kalman-Filters erfolgt, ergibt sich

$$\delta \mathbf{x}_{k+1}^- = \delta \mathbf{x}_k^+ \quad (2.14)$$

Es lässt sich zeigen, dass die Linearisierung der Fehlerdifferentialgleichung nach den Fehlergrößen, $\left. \frac{\partial \delta \dot{\mathbf{x}}}{\partial \delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}$, gleich der im Extended Kalman-Filter genutzten

Linearisierung der Differentialgleichung nach den absoluten Größen, $\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}}$ ist. Daher entspricht die Fehlerkovarianzpropagation der Gleichung (2.8). Auch die Verarbeitung von Messungen entspricht der beim EKF.

Bei Low-Cost-Sensoren ist es üblich, die Fehlergröße $\delta \mathbf{x}$ nach jedem Messschritt auf die Zustandsschätzung zurückzuführen, also eine Korrektur wie folgt durchzuführen:

$$\hat{\mathbf{x}}_{k,\text{korrigiert}} = \hat{\mathbf{x}}_k - \delta \mathbf{x}_k, \quad \delta \mathbf{x}_k = 0 \quad (2.15)$$

Dadurch bleiben die Fehlerterme klein und die Linearisierung um $\mathbf{x} = \hat{\mathbf{x}}$ ist gültig. Aufgrund dieser Rückkopplung spricht man von einer sog. Closed-Loop-Error-State-Architektur.

In der bisherigen Darstellung besteht kein wesentlicher Unterschied zwischen der Formulierung als Extended Kalman-Filter und Error-State Kalman-Filter. Die Error-State-Formulierung ist dann vorteilhaft, wenn man den Vektor $\delta \mathbf{x}$ so definiert, dass die Lagefehler als Eulerwinkel enthalten sind. Die Positionsfehler können in Metern statt als Fehler in Längen- und Breitengrad ausgedrückt werden. Durch diese Umstellungen wird die Systemmatrix vereinfacht; außerdem wird die Dimension von $\delta \mathbf{x}$ um eins reduziert, da statt des Lagequaternions nur noch drei Einträge für die Lagefehler enthalten sind. Somit entfällt auch die sonst notwendige Normalisierung des Lagequaternions und die dadurch entstehende Singularität der Kovarianzmatrix. Die Korrektur wird durch Änderung des Fehlervektors allerdings nicht mehr entsprechend Gleichung (2.15) durchgeführt, sondern ist dann nichtlinear.

2.2.2 Multisensorielle Navigation mit Low-Cost-Sensoren

Viele in wissenschaftlichen Arbeiten zu autonomen Flugplattformen genutzte Navigationssysteme weisen einen identischen Aufbau auf. Da sich gezeigt hat, dass Filter höherer Ordnung, z. B. das Unscented Kalman-Filter, oder Partikelfilter für typische Navigationsanwendungen keine nennenswerten Vorteile bringen, wird fast immer ein EKF oder Error-State Kalman-Filter eingesetzt. Geschätzt wird der Zustand

$$\hat{\mathbf{x}}_k = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \mathbf{b}_a \\ \mathbf{b}_w \end{pmatrix} \quad (2.16)$$

bestehend aus Position, Geschwindigkeit, Lage und den Biasen der Inertialsensoren. Im Kalman-Filter lautet der geschätzte Zustandsvektor

$$\delta \mathbf{x} = \begin{pmatrix} \delta \mathbf{p} \\ \delta \mathbf{v} \\ \delta \boldsymbol{\psi} \\ \delta \mathbf{b}_a \\ \delta \mathbf{b}_w \end{pmatrix}, \quad (2.17)$$

wobei hier $\delta \boldsymbol{\psi}$ die Lagewinkelfehler als Eulerwinkel darstellen. Je nach Art der Stützsensoren kann der Zustandsvektor noch erweitert werden, z. B. um den Uhrenfehler des GPS-Empfängers oder einen Bias des barometrischen Höhenmessers zu schätzen.

Entsprechende Systemmodelle werden z. B. in [36, 87] hergeleitet.

Kapitel 3

Bewegungsmodell für Quadrocopter

In diesem Kapitel soll ein Bewegungsmodell für Quadrocopter vorgestellt werden. Dieses bildet die Grundlage sowohl für die Simulation des in dieser Arbeit verwendeten Quadrocopters (Kapitel 4) als auch für das Bewegungsmodell, welches im Rahmen der modellgestützten Navigation (Kapitel 6) zum Einsatz kommt.

Für das Modell wird angenommen, dass vier fest montierte Motoren vorhanden sind. Die Ausleger von Quadrocoptern sind i. d. R. steif ausgeführt und Nutzlasten werden fixiert. Daher kann in guter Näherung von einem Starrkörper ausgegangen werden. Um dessen Bewegung zu modellieren, müssen die wirkenden Kräfte und deren Angriffspunkte sowie die Trägheit des Fluggerätes bestimmt werden.

Da Start- und Landevorgänge nicht im Fokus dieser Arbeit stehen, wird davon ausgegangen, dass das Fluggerät entweder vollständig am Boden steht oder sich komplett in der Luft befindet. Der erste Fall (vollständig am Boden) ist trivial zu modellieren, da hier Geschwindigkeit und Drehrate gegenüber dem Navigationskoordinatensystem null sind. Sobald die Kraft der Motoren die Schwerkraft übersteigt, geht das Fluggerät in den freien Flug über. Eine Berücksichtigung von Stößen ist somit nicht erforderlich.

Kraftübertragung auf das Fluggerät geschieht demnach nur durch aerodynamische Kräfte. Diese sind insbesondere die Auftriebskräfte an den sich drehenden Rotorblättern sowie Strömungswiderstand, der ebenfalls an den Rotorblättern vorliegt, aber natürlich auch am restlichen Körper.

3.1 Funktionsprinzip AirQuad

Beim AirQuad handelt es sich um einen Quadrocopter, dessen Motoren wie in Abbildung 3.1 gezeigt angeordnet sind: Je ein Aktuator, d. h. ein Motor mit

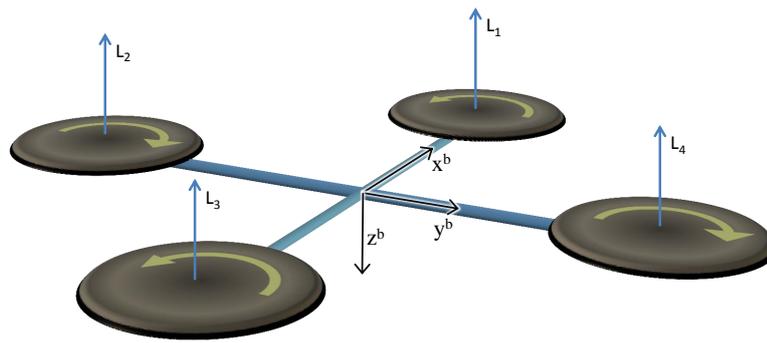


Abbildung 3.1: Anordnung der Motoren und ihre Drehrichtungen bei den AirQuad-Fluggeräten.

einem Rotor, ist vorne, hinten, links und rechts angebracht. Die Anstellwinkel der Rotorblätter können nicht wie bei Helikoptern mit Front- und Heckrotor angepasst werden. Änderungen des Auftriebs und der Momente sind also nur durch Drehzahländerungen möglich. Die eingesetzten bürstenlosen Gleichstrommotoren und leichten Rotorblätter lassen schnelle Änderungen der Drehzahlen zu, so dass auf die aufwändige Mechanik zur Anstellwinkelsteuerung verzichtet werden kann und dennoch ein stabiler Flug möglich ist.

Die Motorachsen und somit der Schub der Aktuatoren sind im Körperkoordinatensystem nach oben gerichtet. Horizontal kann nur beschleunigt werden, indem ein geeigneter Roll- oder Nickwinkel eingenommen wird.

Jeweils zwei Rotoren drehen im Uhrzeigersinn bzw. gegen den Uhrzeigersinn. Dadurch heben sich im Schwebeflug die Giermomente auf, wenn alle Rotoren den gleichen Auftrieb erzeugen. Beim Rollen und Nicken werden die Drehzahlen gegenüberliegender Motoren so geändert, dass die Summe der Auftriebskräfte konstant bleibt. Auch das Giermoment hebt sich dann weiterhin auf. Wenn eine Gierbewegung kommandiert wird, kann diese umgesetzt werden, indem die Drehzahlen der jeweils gleich drehenden Paare entgegengesetzt verändert werden.

Ein solcher Quadrokopter wird also mittels vier Stellgrößen, den Spannungen an den Motoren, gesteuert. Der Flugregler kann damit Roll-, Nick- und Gierwinkel sowie eine gewünschte Höhe einregeln. Die für den automatischen oder autonomen Flug berechneten Sollgeschwindigkeiten oder Sollpositionen werden von der Flugführungsalgorithmik in diese grundlegenden Sollwinkel und die Sollhöhe transformiert, bevor sie an den Flugregler gegeben werden. Eine getrennte Vorgabe von Geschwindigkeit und Lage ist prinzipbedingt nicht möglich, sondern erfordert

einen erweiterten Aufbau wie in [117] oder [69]. In dieser Arbeit sollen jedoch nur gewöhnliche Quadrocopter betrachtet werden.

3.2 Aktuatormodellierung

Bei der Systembildung der Aktuatoren geht es darum, die resultierenden Kräfte und Momente eines Motor-Rotor-Systems basierend auf den Eingangsspannungen zu bestimmen. Bei Multikoptern kommen fast ausschließlich bürstenlose Gleichstrommotoren zum Einsatz. Hierbei übt ein in den Statorspulen erzeugtes, rotierendes Magnetfeld eine Kraft auf die Permanentmagneten am Läufer aus, so dass dieser beschleunigt wird. Das rotierende Magnetfeld wird durch Kommutierung erzeugt, der Strom fließt also wechselnd durch unterschiedliche Stränge im Stator, wobei üblicherweise drei Stränge vorliegen. Der Motorcontroller detektiert anhand der in den Statorwicklungen induzierten Gegenspannungen die Rotorposition, um die Kommutierungszeitpunkte zu bestimmen. Es handelt sich also um ein drehzahlgeregeltes System.

3.2.1 Drehzahl und Drehmoment

Die aerodynamischen Kräfte an einem Rotorblatt werden maßgeblich von der relativen Geschwindigkeit zwischen Luft und Rotorblatt bestimmt. Diese Geschwindigkeit wiederum entsteht hauptsächlich aus der schnellen Drehbewegung des Rotorblattes um die Motorachse. Daher müssen für eine Quadrocopter-Simulation die Drehzahlen modelliert werden. In einigen Arbeiten, z. B. [15, 61] oder auch der bisherigen Modellierung des AirQuad [60], wird die statische Drehzahl-Spannungskennlinie der Aktuatoren ermittelt und das dynamische Verhalten als PT1-System modelliert. Diese Approximation ist aber nur in einem engen Drehzahlbereich gültig – bei dynamischen Manövern mit größeren Drehzahländerungen wird die tatsächliche Drehzahl nur bedingt abgebildet. Eine genauere Modellierung ist möglich, wenn ein Modell für den Motor kombiniert wird mit dem aerodynamisch erzeugten Lastmoment am Rotor. Eine solche Modellierung soll in den folgenden Abschnitten vorgenommen werden.

3.2.1.1 Motorgleichungen

Der bürstenlose Gleichstrommotor lässt sich in der Gleichung

$$U_A = R_A I_A + L \frac{dI_A}{dt} + \Psi \omega_m \quad (3.1)$$

beschreiben. Die verwendeten Motoren weisen nur kleine Induktivitäten L auf und die Zeitkonstante dieser Komponente ist deutlich kleiner als die Zeitkonstanten des mechanischen Systems. Daher wird wie üblich (z. B. [74]) die Spannungsgleichung zu

$$U_A = R_A I_A + \Psi \omega_m \quad (3.2)$$

vereinfacht.

Diese Gleichung kann umgewandelt werden in eine Differentialgleichung für die Motordrehzahl ω_m , indem das elektromagnetische Drehmoment $M_m = \Psi I_A$ mittels Gleichung (3.2) geschrieben wird als

$$M_m = \frac{\Psi}{R_A} (U_A - \Psi \omega_m). \quad (3.3)$$

3.2.1.2 Aerodynamisches Lastmoment

Die Drehrate ω_m wird neben dem Motormoment auch vom Lastmoment M_l beeinflusst:

$$\dot{\omega}_m = \frac{1}{J} (M_m(\omega_m) - M_l(\omega_m)) \quad (3.4)$$

Das Lastmoment M_l entsteht aufgrund der aerodynamischen Widerstandskräfte, die auf den sich in der Luft drehenden Rotor wirken. Für ein Tragflächenprofil gilt gemäß [9]

$$F_{D_r} = C_{D_r} \frac{A \rho v^2}{2}, \quad (3.5)$$

wobei F_{D_r} hier die Widerstandskraft (engl. drag) am Rotor bezeichnet. ρ ist die Dichte der Luft, A eine Referenzfläche, v die Anströmungsgeschwindigkeit und C_{D_r} ein einheitenloser Koeffizient. Nimmt man C_{D_r} als konstant entlang des Rotors an, lässt sich die Kraft auf den sich drehenden Rotor durch Integration berechnen: Der Rotor wird im Bereich $r \in [0..R]$ (mit R : Rotorradius) in infinitesimal kleine Rotorelemente zerlegt, für die dann $v = r\omega_m$ gilt. Für ein Rotorelement ergibt sich

$$dF_{D_r} = C_{D_r} \frac{dA \rho (r\omega_m)^2}{2}. \quad (3.6)$$

Das Gesamtmoment für zwei Rotorblätter ist

$$M_l = 2 \int_{r=0}^R r dF_{D_r}, \quad (3.7)$$

mit $dA = cdr$ (c : Profiltiefe):

$$\begin{aligned}
 M_l &= 2 \int_{r=0}^R r C_{D_r} \frac{dA \rho (r\omega_m)^2}{2} \\
 &= 2 \int_{r=0}^R r C_{D_r} \frac{\rho (r\omega_m)^2}{2} cdr \\
 &= C_{D_r} \frac{\rho c R^4 \omega_m^2}{4} \\
 &= C_{D_r} \frac{\rho c R^4 \omega_m^2}{4}
 \end{aligned} \tag{3.8}$$

3.2.1.3 Parameterbestimmung für die Drehzahlberechnung

Gemäß den oben hergeleiteten Zusammenhängen ergibt sich die Drehzahldifferentialgleichung

$$\dot{\omega}_m = \frac{1}{J} \left(\frac{\Psi}{R_A} (U_A - \Psi\omega_m) - C_{D_r} \frac{\rho c R^4 \omega_m^2}{4} \right). \tag{3.9}$$

Anstatt die darin enthaltenen Parameter einzeln zu identifizieren, wird die Gleichung abstrahiert betrachtet als

$$\dot{\omega}_m = aU_A + b\omega_m + c\omega_m^2, \tag{3.10}$$

wobei die Koeffizienten a , b und c die Parameter des Motors und der Aerodynamik zusammenfassen:

$$a = \frac{\Psi}{JR_A} \tag{3.11}$$

$$b = -\frac{\Psi^2}{JR_A} \tag{3.12}$$

$$c = -C_{D_r} \frac{\rho c R^4}{4J} \tag{3.13}$$

Zur Bestimmung dieser Koeffizienten ist es sinnvoll, auf Messdaten aus Flugversuchen zurückzugreifen, da nur dort reale Bedingungen, insbesondere bezüglich der vorherrschenden Luftströmungen, vorliegen. Messungen an einzelnen Motoren, z. B. auf einem Versuchsaufbau auf einem Tisch, könnten zu verfälschten Ergebnissen führen.

Um die tatsächlichen Drehzahlen zu messen, wurde die Motorcontroller-Software¹ so modifiziert, dass die exakten Zeiten der Kommutierungen erfasst wurden. Nach der Zeit \tilde{T} einer vollständigen Rotordrehung kann die gemessene Motordrehzahl $\tilde{\omega}_m = 2\pi/\tilde{T}$ ausgegeben werden.

Die Motorspannungen U_A sind bekannt, da diese als Stellgröße genutzt werden. Somit kann, wenn ein Parametervektor (a, b, c) vorliegt, durch Integration von Gleichung (3.10) mit $\hat{\omega}_0 = 0$ und $\hat{\omega}_{m,k+1} = \hat{\omega}_{m,k} + (aU_A + b\hat{\omega}_{m,k} + c\hat{\omega}_{m,k}^2) dt$ die modellierte Drehzahl $\hat{\omega}_{m,k}$ berechnet werden. Die Summe der Fehlerquadrate $\sum_{k=0}^N (\tilde{\omega}_{m,k} - \hat{\omega}_{m,k})^2$ ist ein Gütemaß für die Wahl des Parametervektors (a, b, c) . Der Parametervektor kann somit durch Minimierung des Gütemaßes bestimmt werden: Zuerst wird ein initialer Parametervektor ermittelt. Für diesen wird gefordert, dass er der bekannten statischen Kennlinie entspricht und dass die resultierende Sprungantwort zur Sprungantwort unter Laborbedingungen passt. Die Optimierung des Parametervektors erfolgt dann über ein Downhill-Simplex-Minimierungsverfahren [64]. Abbildung 3.2 zeigt, dass im Vergleich zu einer Modellierung als PT1-Glied insbesondere bei erhöhten oder reduzierten Drehzahlen die modellierte Drehzahl deutlich besser der gemessenen entspricht.

Die Abweichung der modellierten von der gemessenen Drehzahl kann als Root-Mean-Square-Fehler (RMS) quantifiziert werden, wobei ein nicht zur Parameterbestimmung verwendeter Zeitraum betrachtet wird: Bei Modellierung als PT1-Glied, ebenfalls mit Downhill-Simplex-Parameteroptimierung, liegt die Abweichung im quadratischen Mittel bei $\text{rms}_{\text{PT1}} = 50,5 U/\text{min}$. Durch die Modellierung mit aerodynamischem Lastmoment kann dieser Wert auf $\text{rms} = 30,5 U/\text{min}$, bzw. 0,6% der tatsächlichen Drehzahl, reduziert werden.

3.2.2 Auftriebskräfte

Analog zu Gleichung (3.6) für die Widerstandskraft kann auch die Auftriebskraft dF_L (engl. lift) eines infinitesimalen Elements aus einem Rotorblatt berechnet werden:

$$dF_L = C_L \frac{dA\rho (r\omega_m)^2}{2} \quad (3.14)$$

¹<http://svn.mikrocopter.de/listing.php?replname=BL-Ctrl&path=%2Ftags%2FV0.42%2F&>, Zugriff am 17.02.2015

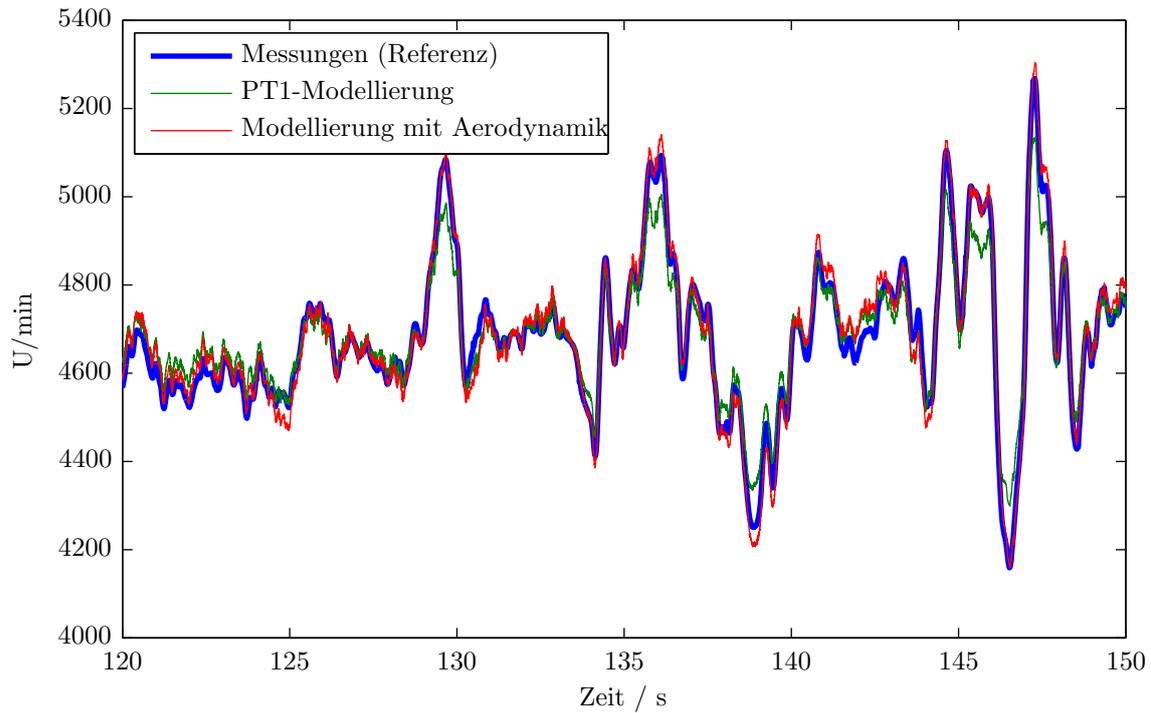


Abbildung 3.2: Validierung des Motormodells mit Berücksichtigung des aerodynamischen Lastmoments. Die Verbesserung gegenüber einer Modellierung als PT1-System ist insbesondere bei hohen und reduzierten Drehzahlen zu erkennen.

Die Gesamtkraft für zwei Rotorblätter ergibt sich dann zu $F_L = 2 \int_{r=0}^R dF_L$:

$$\begin{aligned}
 F_L &= 2 \int_{r=0}^R C_L \frac{dA \rho (r \omega_m)^2}{2} \\
 &= 2 \int_{r=0}^R C_L \frac{\rho (r \omega_m)^2}{2} c dr \\
 &= C_L \frac{\rho c r^3 \omega_m^2}{3}
 \end{aligned} \tag{3.15}$$

Die Auftriebskraft F_L eines Aktuators ist also proportional zu ω_m^2 . Daher kann die Auftriebskraft in Body-Koordinaten geschrieben werden als

$$F_L^b = m \omega_m^2. \tag{3.16}$$

Hierbei ist m ein Parametervektor. Dadurch kann auch eine Schrägstellung der Motoren, z. B. aufgrund von leicht verdrehtem Einbau, modelliert werden. Die Bestimmung von m wird in Abschnitt 3.4.1 beschrieben, da diese Konstanten für alle Motoren gemeinsam und auch zusammen mit den Strömungswiderstands-Parametern identifiziert werden müssen.

3.3 Strömungswiderstand

Bewegt sich ein Objekt, übt die umgebende Luft eine der Bewegung entgegengerichtete Kraft auf den Körper aus. Für Helikopter wirkt diese Kraft einerseits an den festen Komponenten, also dem Rahmen, den Sensoren und den Rechenplattformen. Andererseits kann auch die Drehung der Rotoren in der Luft zu einem weiteren Kraftanteil führen, der ebenfalls der Gesamtbewegungsrichtung entgegengerichtet ist. Der Widerstand am festen Körper steigt quadratisch mit der Geschwindigkeit, der Strömungswiderstand an den Rotoren dagegen linear zur Geschwindigkeit [17]. Während bei verschiedenen Experimenten zur Parameteridentifikation der quadratische Widerstandsterm eindeutig sichtbar wurde, wies der lineare Widerstandsterm aber keinen merklichen Beitrag auf.

Die durch den Strömungswiderstand verursachte Kraft am Fluggerät wird daher gemäß der Fluidodynamik modelliert als

$$F_{D_f} = C_{D_f} \frac{A \rho v_a^2}{2}. \quad (3.17)$$

Die Gleichung entspricht der Widerstandsgleichung für den Rotor (Gleichung (3.5)). C_{D_f} ist wieder ein einheitenloser Koeffizient, der von der Form des Körpers abhängig ist, A eine Bezugsfläche und ρ die Dichte der Luft. v_a ist die Anströmgeschwindigkeit.

3.4 Translatorisches Modell

Die Summe der am Fluggerät wirkenden Kräfte definiert die Bewegung des Schwerpunkts:

$$\mathbf{a}_{ib}^b = \frac{1}{m} \left(\sum_{i=1}^4 \mathbf{m}_i \omega_{mi}^2 + \mathbf{F}_{D_f}^b \right) + \mathbf{C}_n^b \mathbf{g}_b^n \quad (3.18)$$

Gemeinsam mit dem in Abschnitt 3.5 definierten rotatorischen Modell, aus dem die Drehraten in körperfesten Koordinaten resultieren, können durch Integration wie im Strapdown-Algorithmus die Geschwindigkeit und Position gewonnen werden.

3.4.1 Identifikation der translatorischen Modellparameter

Zur Modellierung der AirQuad-Fluggeräte wurden die translatorischen Parameter gemäß der im Folgenden beschriebenen Vorgehensweise bestimmt.

3.4.1.1 Masse

Die Masse m des Fluggerätes ist mit Hilfe einer Waage sehr einfach vor dem Flug zu bestimmen und bei elektrisch angetriebenen Helikoptern während des Fluges konstant. Eine Schätzung aus anderen Messgrößen ist daher nicht nötig.

3.4.1.2 Auftrieb und Strömungswiderstand

Zur Vereinfachung der folgenden Gleichungen wird zunächst ein neuer Parameter

$$k_D = \frac{1}{2} C_{D_f} A \rho \quad (3.19)$$

definiert, so dass $F_D = k_D v_a^2$ gilt. Für Identifikationsflüge in Innenräumen kann $v_{eb} = v_a$ angenommen werden, es wird also davon ausgegangen, dass kein Wind vorliegt.

Die oben beschriebenen Modellparameter m_i und k_D müssen nun so gewählt werden, dass die erwartete Specific Force \hat{f}_{ib}^b der tatsächlichen entspricht:

$$\mathbf{f}_{ib}^b \stackrel{!}{=} \hat{\mathbf{f}}_{ib}^b = \frac{1}{m} \left(\underbrace{\sum_{i=1}^4 m_i \omega_{m,i}^2}_{\text{Auftrieb}} \underbrace{- k_D |\mathbf{v}_{eb}^b| \mathbf{v}_{eb}^b}_{\text{Luftwiderstand}} \right) \quad (3.20)$$

Die Messung der Motordrehzahl $\omega_{m,i}^2$ erfolgt sehr präzise, die Geschwindigkeit \mathbf{v}_{eb}^b sollte mit einem genauen Referenznavigationssystem bestimmt werden, z. B. dem in Kapitel 5 entwickelten lasergestützten Navigationssystem mit RTS-Smoothing (Unterkapitel 5.5). Vorerst soll davon ausgegangen werden, dass die wahre Specific Force \mathbf{f}_{ib}^b bekannt sei (zur Ermittlung s. weiter unten). Dann können die Parameter mittels Least-Squares-Schätzung bestimmt werden:

- Für jedes $a \in [1..3]$ (entsprechend den Achsen x , y , und z):

- Definiere den Parametervektor $\mathbf{p}_a = \left(m_{1,a} \ m_{2,a} \ m_{3,a} \ m_{4,a} \ k_{D,a} \right)^T$, der vier Motorparameter sowie den Koeffizienten für den Strömungswiderstand in der jeweiligen Bewegungsrichtung enthält

- Berechne die Parameter über Least-Squares-Schätzung, um den Fehler von $\mathbf{X}\mathbf{p}_a = \mathbf{y}$ zu minimieren, wobei

$$\mathbf{y} = \begin{pmatrix} f_{ib,a,k=1}^b \\ \vdots \\ f_{ib,a,k=n}^b \end{pmatrix} \quad (3.21)$$

und

$$\mathbf{X} = \begin{pmatrix} \omega_{m,1,k=1}^2 & \cdots & \omega_{m,4,k=1}^2 & |v_{eb,k=1}^b|v_{eb,a,k=1}^b \\ \vdots & & \vdots & \vdots \\ \omega_{m,1,k=n}^2 & \cdots & \omega_{m,4,k=n}^2 & |v_{eb,k=n}^b|v_{eb,a,k=n}^b \end{pmatrix} \quad (3.22)$$

Natürlich ist die wahre Specific Force f_{ib}^b nicht exakt bekannt. Eine Approximation erhält man, indem man die Beschleunigungsmesswerte um den vom Referenznavigationssystem geschätzten Bias korrigiert. Wenn der verbleibende Fehler als weißes Rauschen beschrieben werden kann, kann die LS-Schätzung der Parameter direkt erfolgen. Bei den durchgeführten Messungen war aber die Vibration des Fluggerätes deutlich im Beschleunigungssignal enthalten, insbesondere in der einfachen und doppelten Frequenz der Motordrehzahlen, also 70 Hz–80 Hz und 140 Hz–160 Hz

Da diese Vibrationen nicht vernachlässigt werden konnten, wurde ein lineares Tiefpassfilter $TP(\cdot)$ auf beiden Seiten von $\mathbf{X}\mathbf{p}_a = \mathbf{y}$ angewendet. Aufgrund der Linearität der Gleichungen folgt $\bar{\mathbf{X}}\mathbf{p}_a = \bar{\mathbf{y}}$ mit

$$\bar{\mathbf{y}} = TP(\mathbf{y}) \quad (3.23)$$

und

$$\bar{\mathbf{X}} = \begin{pmatrix} \bar{\omega}_1 & \cdots & \bar{\omega}_4 & TP \begin{pmatrix} |v_{eb,k=1}^b|v_{eb,a,k=1}^b \\ \vdots \\ |v_{eb,k=n}^b|v_{eb,a,k=n}^b \end{pmatrix} \end{pmatrix} \quad (3.24)$$

mit

$$\bar{\omega}_i = TP \begin{pmatrix} \omega_{m,i,k=1}^2 \\ \vdots \\ \omega_{m,i,k=n}^2 \end{pmatrix}. \quad (3.25)$$

Die Lösung dieser drei Least-Squares-Probleme führt zu den gesuchten 15 Modellparametern.

3.4.1.3 Beobachtbarkeit der Parameter

Es soll nun geprüft werden, wann eine vollständige Beobachtbarkeit der Modellparameter für Auftriebskräfte und Strömungswiderstand vorliegt. Dazu betrachtet man die Messmatrix (3.22): Beobachtbarkeit liegt vor, wenn diese Matrix für die x-, y- und z-Parameter jeweils Rang 5 hat.

Dieser Matrixrang kann durch einige einfache Flugmanöver erreicht werden: Nach einer kurzen Phase im Schwebeflug wird eine Geschwindigkeit v_0 in jede Raumrichtung entlang der Körperachsen x^b , y^b und z^b aufgebaut, d.h. ein Flugmuster vor/zurück, links/rechts, auf/ab wird ausgeführt.

Um diese Geschwindigkeiten mit einem Quadropter zu erreichen, müssen Roll- und Nickwinkel kommandiert werden. Der Flugregler variiert dazu die jeweiligen Motordrehzahlen, also z. B. unterschiedliche Drehzahlen für den linken und rechten Motor für ein Rollmanöver.

Daraus folgt, dass die Matrix nach diesen Manövern mindestens Rang 4 hat:

$$\mathbf{X} = \begin{pmatrix} \omega_0^2 & \omega_0^2 & \omega_0^2 & \omega_0^2 & 0 \\ \omega_0^2 & \omega_0^2 & \omega_0^2 & \omega_0^2 & v_0^2 \\ (\omega_0 - \delta)^2 & \omega_0^2 & (\omega_0 + \delta)^2 & \omega_0^2 & 0 \\ \omega_0^2 & (\omega_0 - \delta)^2 & \omega_0^2 & (\omega_0 + \delta)^2 & 0 \end{pmatrix} \quad (3.26)$$

In dieser Matrix beschreibt die erste Zeile das Schweben (alle Motoren gleiche Drehzahl, keine Geschwindigkeit). Die zweite Zeile ist Flug mit konstanter Geschwindigkeit und die Zeilen 3 und 4 beschreiben Nicken und Rollen. δ ist die Variation der Motordrehzahlen zum Erreichen der Lageänderung.

Um eine weitere Kombination von Motordrehzahlen zu erhalten, kann ein Gieren kommandiert werden, das zu einer fünften linear unabhängigen Zeile und somit Rang 5 der Matrix führt:

$$\mathbf{X} = \begin{pmatrix} \omega_0^2 & \omega_0^2 & \omega_0^2 & \omega_0^2 & 0 \\ \omega_0^2 & \omega_0^2 & \omega_0^2 & \omega_0^2 & v_0^2 \\ (\omega_0 - \delta)^2 & \omega_0^2 & (\omega_0 + \delta)^2 & \omega_0^2 & 0 \\ \omega_0^2 & (\omega_0 - \delta)^2 & \omega_0^2 & (\omega_0 + \delta)^2 & 0 \\ (\omega_0 + \delta)^2 & (\omega_0 - \delta)^2 & (\omega_0 + \delta)^2 & (\omega_0 - \delta)^2 & 0 \end{pmatrix} \quad (3.27)$$

Höhere Geschwindigkeiten v_0 und stärkere Drehzahländerungen δ verbessern die Schätzgenauigkeit, wenn die Messwerte verrauscht sind. Die Ergebnisse in Kapitel 6 zeigen aber, dass für die Identifikation der Parameter zur Nutzung in einem modellgestützten Navigationssystem niedrige Geschwindigkeiten unter $2 \frac{\text{m}}{\text{s}}$ ausreichend sind. Die erforderlichen Manöver lassen sich innerhalb kurzer Zeit auf engem Raum, z. B. in einem Büroraum, durchführen.

3.5 Rotatorisches Modell

Zur Herleitung einer Gleichung für die rotatorische Bewegung des Fluggerätes betrachtet man den Eulerschen Drehimpulssatz im inertialen Koordinatensystem:

$$\dot{\mathbf{L}}^i = \sum \mathbf{M}_{\text{ext}}^i \quad (3.28)$$

Diese Gleichung besagt, dass die zeitliche Änderung des Drehimpulses gleich der Summe der äußeren Drehmomente ist.

Der Drehimpuls bei einem Quadrocopter ist die Summe aus dem Drehimpuls des Fluggerätes selbst zuzüglich der Rotordrehimpulse:

$$\mathbf{L}_{ges}^i = \mathbf{L}_f^i + \sum_{k=1}^4 \mathbf{L}_{r,k}^i \quad (3.29)$$

Die Trägheitstensoren der AirQuad-Fluggeräte und ihrer Rotoren lassen sich aufgrund der Symmetrien in Body-Koordinaten als zeitlich konstante, diagonale Matrizen \mathbf{J}_f^b bzw. $\mathbf{J}_{r,k}^b$ darstellen. Damit gilt:

$$\mathbf{L}_{ges}^i = \underbrace{\mathbf{C}_b^i \mathbf{J}_f^b \mathbf{C}_i^b}_{\mathbf{J}_f^i} \boldsymbol{\omega}_{ib}^i + \sum_{k=1}^4 \underbrace{\mathbf{C}_b^i \mathbf{J}_{r,k}^b \mathbf{C}_i^b}_{\mathbf{J}_{r,k}^i} \boldsymbol{\omega}_{r,k}^i = \mathbf{C}_b^i \mathbf{J}_f^b \boldsymbol{\omega}_{ib}^b + \sum_{k=1}^4 \mathbf{C}_b^i \mathbf{J}_{r,k}^b \boldsymbol{\omega}_{r,k}^b \quad (3.30)$$

Setzt man Gleichung (3.30) in Gleichung (3.28) ein, ergibt sich

$$\dot{\mathbf{C}}_b^i \mathbf{J}_f^b \boldsymbol{\omega}_{ib}^b + \mathbf{C}_b^i \mathbf{J}_f^b \dot{\boldsymbol{\omega}}_{ib}^b + \sum_{k=1}^4 \left(\dot{\mathbf{C}}_b^i \mathbf{J}_{r,k}^b \boldsymbol{\omega}_{r,k}^b + \mathbf{C}_b^i \mathbf{J}_{r,k}^b \dot{\boldsymbol{\omega}}_{r,k}^b \right) = \sum \mathbf{M}_{\text{ext}}^i. \quad (3.31)$$

Nutzt man $\dot{\mathbf{C}}_b^i = \mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b$, erhält man

$$\mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b \mathbf{J}_f^b \boldsymbol{\omega}_{ib}^b + \mathbf{C}_b^i \mathbf{J}_f^b \dot{\boldsymbol{\omega}}_{ib}^b + \sum_{k=1}^4 \left(\mathbf{C}_b^i \boldsymbol{\Omega}_{ib}^b \mathbf{J}_{r,k}^b \boldsymbol{\omega}_{r,k}^b + \mathbf{C}_b^i \mathbf{J}_{r,k}^b \dot{\boldsymbol{\omega}}_{r,k}^b \right) = \sum \mathbf{M}_{\text{ext}}^i. \quad (3.32)$$

Transformiert in b-frame-Koordinaten und aufgelöst nach $\dot{\boldsymbol{\omega}}_{ib}^b$ resultiert die gesuchte Differentialgleichung für die Drehrate des Fluggerätes:

$$\dot{\boldsymbol{\omega}}_{ib}^b = (\mathbf{J}_f^b)^{-1} \left(- \sum_{k=1}^4 (\boldsymbol{\omega}_{ib}^b \times \mathbf{J}_{r,k}^b \boldsymbol{\omega}_{r,k}^b + \mathbf{J}_{r,k}^b \dot{\boldsymbol{\omega}}_{r,k}^b) - \boldsymbol{\omega}_{ib}^b \times \mathbf{J}_f^b \boldsymbol{\omega}_{ib}^b + \sum \mathbf{M}_{\text{ext}}^b \right) \quad (3.33)$$

In dieser Gleichung sind bereits Kreiseffekte der Rotoren und des Fluggerätes enthalten. Auch eine Drehbeschleunigung des Fluggerätes bei einer Rotordrehzahländerung $\dot{\boldsymbol{\omega}}_{r,k}^b$ wird dadurch beschrieben. Die externen Drehmomente $\sum \mathbf{M}_{\text{ext}}^b$ entstehen aufgrund von aerodynamischen Kräften, nämlich den Auftriebskräften an den Rotoren, dem Strömungswiderstand an den Rotoren und dem Strömungswiderstand am Fluggerät. Es gilt

$$\sum \mathbf{M}_{\text{ext}}^i = \mathbf{M}_h + \mathbf{M}_a + \mathbf{M}_d, \quad (3.34)$$

wobei diese Momente im Folgenden detailliert beschrieben werden.

Aktuator-Momente Die an den Rotoren erzeugten Auftriebskräfte erzeugen über den Hebelarm \mathbf{l}_k zum Schwerpunkt ein Drehmoment \mathbf{M}_h

$$\mathbf{M}_h = \sum_{k=1}^4 \mathbf{l}_k \times \mathbf{f}_{m_k}^b. \quad (3.35)$$

Wenn die Motoren exakt auf den Achsen x^b und y^b und ideal senkrecht eingebaut sind, erzeugt dieses Moment nur Roll- und Nickbewegungen und lässt sich schreiben als

$$\mathbf{M}_{h, \text{senkr}} = l \cdot \begin{pmatrix} F_{L_2} - F_{L_4} \\ F_{L_1} - F_{L_3} \\ 0 \end{pmatrix}. \quad (3.36)$$

Neben diesem hebelarmbedingten Roll- und Nickmoment kann über die Aktoren ein Giermoment erzeugt werden. Dieses entsteht, wenn die aerodynamischen Lastmomente gemäß Gleichung (3.8) sich in Summe aller Motoren nicht aufheben:

$$\mathbf{M}_a = \sum_{k=1}^4 \mathbf{M}_{l,k} = \sum_{k=1}^4 (-1)^{k+1} C_{D_r} \frac{\rho c R^4 \omega_{r,k}^2}{4} \mathbf{e}_z \quad (3.37)$$

Bei senkrecht montierten Motoren kann wegen $M_l \sim \omega_m^2$ und $F_L \sim \omega_m^2$ auch geschrieben werden

$$\mathbf{M}_{a, \text{senkr}} = \begin{pmatrix} 0 \\ 0 \\ k_M (F_{L_1} - F_{L_2} + F_{L_3} - F_{L_4}) \end{pmatrix}, \quad (3.38)$$

wobei k_M dann der Proportionalität zwischen M_{lk} und F_{L_k} Rechnung trägt.

Moment aufgrund des Strömungswiderstands Der Angriffspunkt des Strömungswiderstands fällt im Allgemeinen nicht mit dem Schwerpunkt zusammen, so dass auch der Strömungswiderstand ein Moment erzeugt:

$$\mathbf{M}_d = \mathbf{l}_d \times \mathbf{F}_{D_f} \quad (3.39)$$

\mathbf{F}_{D_f} ist die Kraft aus dem Strömungswiderstand auf das gesamte Fluggerät aus Gleichung (3.17). Zur Stabilisierung der Flugeigenschaften wird ein Quadrokooper üblicherweise so entworfen, dass der Angriffspunkt knapp über dem Schwerpunkt liegt.

Kapitel 4

Simulation

Simulationen dienen dazu, Untersuchungen am PC durchzuführen, die ansonsten ein reales Experiment erfordern würden. Eine Fluggeräte-Simulation kann vielfältige Vorteile mit sich bringen:

- *Erhöhte Sicherheit:* Algorithmen, die Teil der Regelschleife sind, sollten nicht ohne vorherige ausgiebige Tests im realen Experiment eingesetzt werden. Nach entsprechenden Modultests kann eine Software-in-the-Loop-Simulation weitere Probleme aufdecken oder zum Einstellen von Parametern genutzt werden.
- *Beliebige Konfigurationen:* Die Parameter der simulierten Fluggeräte und Sensoren sind beliebig wählbar und jederzeit wieder abrufbar. Der zeitaufwändige Umbau eines realen Fluggerätes entfällt. Auch kann eine beliebige Anzahl von Entwicklern an verschiedenen Konfigurationen gleichzeitig arbeiten, ohne dass diese tatsächlich aufgebaut sein müssen. So können auch Teamingstrategien mit vielen autonomen Geräten getestet werden oder es kann erprobt werden, wie sich zukünftige Sensortechnologien auf die Güte der Navigation auswirken werden, ohne dass die entsprechende Hardware überhaupt vorliegen muss.
- *Reduzierter Zeitaufwand:* Eine Simulation ist auf aktueller Hardware sehr effizient durchführbar. Es können in kürzester Zeit stundenlange Flüge simuliert werden.
- *Erhöhte Vergleichbarkeit:* In der Simulation können Umgebungsbedingungen, z. B. Wind, in verschiedenen Durchläufen exakt reproduziert werden. Dadurch sind Unterschiede bei den Ausgangsgrößen eindeutig auf algorithmische Veränderungen zurückzuführen. Dies ist bei wiederholter Ausführung realer Experimente nur bedingt möglich.

- *Vorliegen einer Referenz:* Zur Beurteilung eines Navigationssystems muss eine Referenzlösung bekannt sein. Eine solche kann in realen Experimenten nur mit teils hohem Aufwand oder hohen Kosten für entsprechende Sensoren bestimmt werden.
- *Möglichkeit spezieller Untersuchungen:* Zur Verifikation neu geschriebener Softwarekomponenten ist es oft empfehlenswert, diese nicht sofort mit realistischen, verrauschten Sensordaten zu testen. Bestimmte Untersuchungen zum Aufzeigen von Fehlern, z. B. die Kovarianzanalyse, erfordern fehlerlose Eingangsdaten, also Messwerte ohne Rauschen etc. Diese können nur in der Simulation erzeugt werden, sind aber wichtig, um grundlegende Fehler frühzeitig aufzuzeigen.

Für die Aufklärungsplattform AirQuad wurde eine Simulationsumgebung entwickelt, die diese Vorteile in der Entwicklung nutzbar macht. Dabei wurde darauf Wert gelegt, dass

- Softwarekomponenten ohne Modifikationen sowohl in der Simulation als auch im realen Flug eingesetzt werden können,
- alle im Flug relevanten algorithmischen Komponenten wie Sensordatenverarbeitung, Navigation, Flugführung und Regelung in der Simulation entwickelt und untersucht werden können,
- sämtliche Anwendungsgebiete der AirQuad-Plattform durch Simulationen abgedeckt werden können.

4.1 Softwarearchitektur

Um die oben genannten Vorteile nutzbar zu machen, wurde eine geeignete Softwarearchitektur entworfen. Diese umfasst den Software-Aufbau sowohl für den im Flug eingesetzten Code wie auch für die eigentliche Simulation von Fluggeräten und den darauf eingesetzten Sensoren.

4.1.1 Grobstruktur der Simulation

Die Simulationsumgebung lässt die parallele Simulation mehrerer Objekte zu, wobei jedes der Objekte intern unterschiedlich aufgebaut und somit auch unterschiedlich simuliert werden kann. Damit wird es möglich, Multi-Agenten-Simulationen

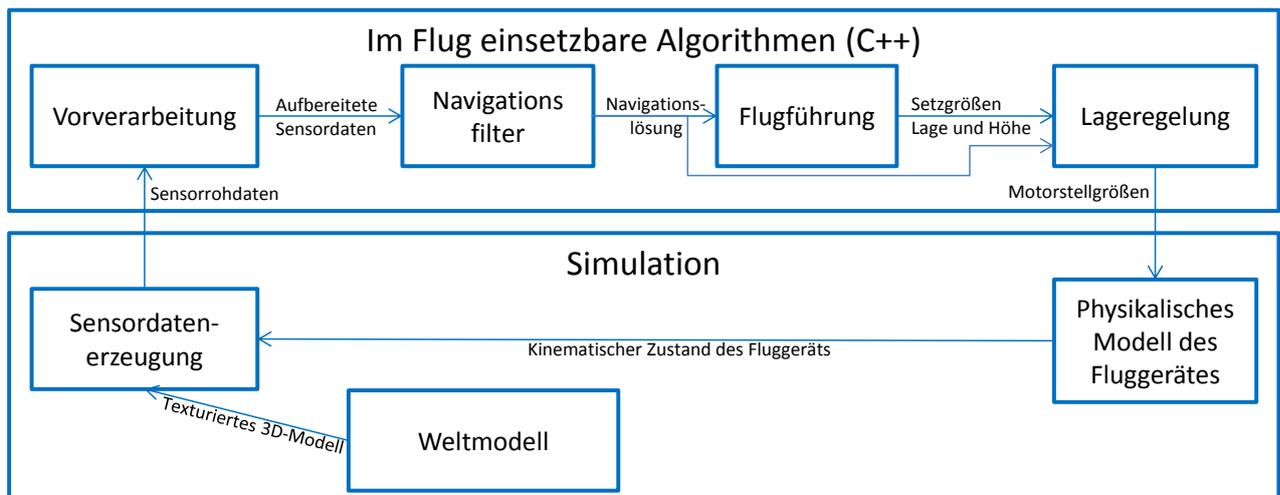


Abbildung 4.1: Struktur der Simulation für ein einzelnes simuliertes Flugobjekt. Die im realen Flug einsetzbare Algorithmik wird als Software-In-The-Loop eingebunden und kann so durch die Simulation getestet werden.

durchzuführen und die Interaktion zwischen mehreren autonomen Systemen zu testen. Da solche Untersuchungen nicht im Fokus dieser Arbeit stehen, wird im Folgenden eine vereinfachte Darstellung gewählt, bei der von der Simulation eines einzelnen Quadropters ausgegangen wird.

In jedem Zeitschritt der zeitdiskreten Simulation wird dabei zum einen der simulierte kinematische Zustand des Fluggerätes anhand des in Kapitel 3 beschriebenen Bewegungsmodells für die Simulationsschrittweite Δt propagiert. Für diesen Zustand werden dann synthetische Sensordaten generiert, die wiederum die Eingangsgröße für die zu testenden Softwaremodule darstellen. Die Datenverarbeitung, die auch im realen Flug eingesetzt wird, beginnt mit einer Vorverarbeitung von Sensordaten; hierbei können bereits komplexe und rechenaufwändige Algorithmen zum Einsatz kommen, um z. B. Bildmerkmale zu extrahieren und zuvor erkannten Merkmalen zuzuordnen. Nach der Sensordatenvorverarbeitung erfolgt die eigentliche Navigationsberechnung, i. d. R. mit Hilfe stochastischer Filtertechniken. Das Flugführungsmodul liefert Sollwerte für die Flugregelung, bestehend aus den Soll-Lagewinkeln und der Sollhöhe. Diese werden dann von der Flugregelung mittels geeigneter Regelungsalgorithmen in die Motorspannungen $U_{A,i}$ umgesetzt. Wie in Abbildung 4.1 gezeigt, werden diese bei der Simulation wieder an das Bewegungsmodell des Quadropters übergeben, so dass eine geschlossene Simulationsschleife entsteht.

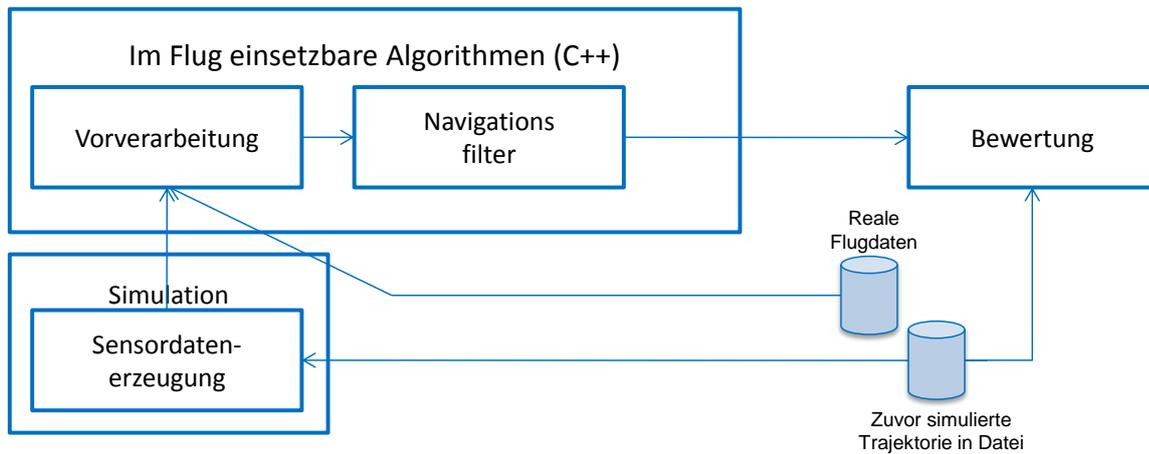


Abbildung 4.2: Struktur der Nachverarbeitung. Hier werden nur die Algorithmen für Sensordaten-Vorverarbeitung und Navigation eingesetzt, da keine geschlossene Regelschleife vorliegt. Dies ermöglicht aber umfassendere Auswertungen.

4.1.2 Postprocessing

Neben der Closed-Loop-Simulation kann es zum Test von Navigationsfiltern vorteilhaft sein, Open-Loop-Untersuchungen durchzuführen. Hierbei wird davon ausgegangen, dass eine Trajektorie vorliegt oder unabhängig von der zu untersuchenden Algorithmik generiert werden kann. Die zu dieser Trajektorie gehörenden Sensordaten können dann verschiedenen Navigationsfilter-Implementierungen als Eingangsgrößen übergeben werden; somit kann ein Vergleich der Navigationsfilter erfolgen. Eine Closed-Loop-Simulation wäre hierfür einerseits unnötig komplex und würde zudem dazu führen, dass für jedes Navigationsfilter auch eine andere Trajektorie generiert würde. Damit wären die Ergebnisse aber nicht mehr ohne weiteres vergleichbar.

Der in Abbildung 4.1 dargestellte Ablauf kann daher auch in reduzierter Form genutzt werden, wie Abbildung 4.2 zeigt. Basierend auf einer Trajektorie werden wieder synthetische Sensordaten erzeugt, die dann aber nur noch für Vorverarbeitung und Navigationsfilter verwendet werden. Die Ausgabe des Navigationsfilters kann direkt mit der tatsächlichen Trajektorie verglichen werden. Für jeden Monte-Carlo-Durchlauf wird der Zufallszahlengenerator unterschiedlich initialisiert, so dass unterschiedliche Sensorfehler generiert werden und der RMS-Wert der Navigationsfehler bestimmt werden kann. Zudem können auch experimentell aufgenommene Flugdaten mit verschiedenen Navigationsfiltern verarbeitet werden und die Ergebnisse der Filter verglichen werden.

4.1.3 Schnittstellenentwurf

Für die konkrete Umsetzung wurde ein objektorientierter Entwurf in C++ gewählt. Damit können Typsicherheit, Modularität und modernes Softwaredesign verbunden werden mit Hardware- und prozessnaher Programmierung, wo dies z. B. aufgrund von Performanceanforderungen notwendig ist. Damit die einzelnen Systemkomponenten austauschbar werden und in verschiedenen Projekten eingesetzt werden können, ist eine klare Definition von abstrakten Schnittstellen wesentlich: Diese Schnittstellen wurden dabei sehr allgemein formuliert, um die Implementierungsmöglichkeiten nicht einzuschränken. Dies wird am Beispiel der Schnittstelle für Navigationssysteme deutlich:

Deren `update()`-Methode nimmt nur die aktuellen Sensordaten entgegen. Zur Ausgabe gibt es `get()`-Methoden, die die aktuell geschätzte Navigationslösung und eine 9x9-Kovarianz für die Positions-, Geschwindigkeits- und Lagefehler liefern. Diese Schnittstelle erwartet also, dass eine Navigationslösung geschätzt wird und eine Unsicherheitsinformation darüber vorliegt. Ob zur Bestimmung dieser Größen intern z. B. ein EKF, Unscented Kalman-Filter (UKF) oder Partikel-Filter genutzt wird, ist der jeweiligen Implementierung freigestellt. Auch die Schätzung zusätzlicher Größen wie Inertialsensorbiase oder ganze Karten (im Zusammenhang mit SLAM) ist möglich, aber nicht von der Schnittstelle vorgeschrieben.

Ebenso allgemein ist die Schnittstelle für Vorverarbeitungsmodule gehalten: Diese erhalten die aktuell gemessenen Sensorwerte und geben sie in modifizierter Form zurück. Dies kann vom einfachen Filtern von IMU-Messwerten reichen bis hin zum Extrahieren neuer Informationen, z. B. der Posenänderung seit der letzten Lasermessung. Die Vorverarbeitung insgesamt ist als Pipeline aufgebaut, bei der eine beliebige Anzahl an Vorverarbeitungsmodulen hintereinander ausgeführt werden. Dabei wird die vorgegebene Reihenfolge stets beachtet, damit die Module aufeinander aufbauen können. Beispielsweise kann so zuerst ein Modul zur Transformation von Laserdaten in kartesische Koordinaten ausgeführt werden, dessen Ergebnis dann ein Modul zur Linienextraktion nutzt. Die gefundenen Linien können wiederum von einer Scanmatching-Komponente zur Bestimmung der Posenänderung verwendet werden. Wenn einzelne dieser Schritte nicht synchron in der Hauptregelschleife ausgeführt werden sollen, kann dies ebenfalls im Modul selbst entschieden und implementiert werden. Die Ausgabe steht dann erst in einem späteren Schleifendurchlauf zur Verfügung.

Durch die klaren Schnittstellen und getrennten Verantwortlichkeiten wird es möglich, die Module in vielfältiger Weise zu kombinieren und wiederzuverwenden. Hierfür ist es nicht einmal notwendig, neuen Code zu schreiben. Mit Hilfe des Factory-Entwurfsmusters wurde ein Konfigurationsdateiformat entwickelt, mit dem in übersichtlichen Textdateien Systeme aus den Modulen zusammengesetzt werden können. In diesen Dateien sind auch die verwendeten Parameter gespeichert. Modularität und Wiederverwendbarkeit sind auch hier gegeben, da solche Dateien zu größeren Systemen kombiniert werden können.

4.2 Simulation des Fluggerätes

Zur Simulation des Fluggerätes kommt das in Kapitel 3 vorgestellte Quadrocoptermodell zum Einsatz. Nicht modellierte Effekte wie Turbulenzen können durch additives weißes Rauschen auf den modellierten Kräften und Momenten ergänzt werden.

Gemäß der in [60] eingeführten Reglerschnittstellen gibt die Flugregelung nicht direkt Motorspannungen aus, sondern direkt den Tastgrad für die Pulsweitenmodulation, mit der die Spannung gestellt wird. Deswegen kann auch die Akkuspannung Einfluss auf das Flugverhalten haben. Um solche Effekte in der Simulation abbilden zu können, kann auch ein Verlauf der Akkuspannung über der Zeit vorgegeben werden.

4.2.1 Windeinfluss

Wind kann signifikanten Einfluss auf das Flugverhalten von Quadrocoptern haben und stellt für die später in dieser Arbeit entwickelte modellgestützte Navigation eine Herausforderung dar. Daher sollte dieser Effekt unbedingt auch in der Simulation abgebildet werden. In einer parallel laufenden Arbeit wurde ein Windmodell implementiert, bei dem einzelne an- und abschwellige Böen explizit vorgegeben werden können; ebenso wurde eine Variante des Dryden-Windmodells [1] angepasst auf Quadrocopteranwendungen [85] in die Simulation integriert.

Eine realistische Parameterwahl für diese Modelle ist jedoch nicht ohne weiteres möglich. Daher wurde in dieser Arbeit der Weg gewählt, mit einem Windmesser aufgezeichnete Daten in die Simulation zu integrieren. Liegt ein ausreichend langer Datensatz an Windmessungen vor, können für Monte-Carlo-Untersuchungen unterschiedliche Ausschnitte des Datensatzes abgespielt werden.

Ist die Windgeschwindigkeit \mathbf{v}_w in der Simulation bekannt, kann in Gleichung (3.17) die Anströmgeschwindigkeit $\mathbf{v}_a^b = \mathbf{v}_{eb}^b - \mathbf{v}_w^b$ eingesetzt werden. Dadurch wird der Einfluss des Windes auf das Fluggerät abgebildet.

4.3 Sensorsimulation

Neben dem Bewegungsmodell ist eine realitätsgetreue Abbildung der verwendeten Sensoren essentiell, damit die Algorithmik mit realitätsnahen Eingangsgrößen getestet werden kann. Daher können in der Simulationsumgebung alle zum Einsatz kommenden Sensoren simuliert werden. In den folgenden Abschnitten werden die verwendeten Modelle dargestellt.

4.3.0.1 Inertialsensorik

Die Messwerte der Beschleunigungs- und Drehratensensoren werden durch verschiedene Effekte verfälscht. Deren Einfluss lässt sich durch die Gleichung

$$\tilde{\mathbf{f}}_{ib}^b = (\mathbf{I}_3 + \mathbf{M}_a) \mathbf{N}_a(\mathbf{f}_{ib}^b) + \mathbf{b}_a(t) + \mathbf{w}_a(t) \quad (4.1)$$

$$\tilde{\boldsymbol{\omega}}_{ib}^b = (\mathbf{I}_3 + \mathbf{M}_g) \mathbf{N}_g(\boldsymbol{\omega}_{ib}^b) + \mathbf{b}_g(t) + \mathbf{w}_g(t) \quad (4.2)$$

abbilden. Hierbei sind $\mathbf{N}_a(\cdot)$ und $\mathbf{N}_g(\cdot)$ nichtlineare Funktionen, die die Nichtlinearität des Skalenfaktors darstellen. Hier wurde eine überlagerte Sinusfunktion zur Implementierung gewählt, mit einer Periodendauer im Bereich des Messbereichs. \mathbf{M}_a und \mathbf{M}_g sind Matrizen, die das sog. Misalignment und die Skalenfaktorfehler enthalten:

$$\mathbf{M} = \begin{pmatrix} s_x & m_{xy} & m_{xz} \\ m_{yx} & s_y & m_{yz} \\ m_{zx} & m_{zy} & s_z \end{pmatrix} \quad (4.3)$$

Das Misalignment kann hervorgerufen werden, wenn die drei sensitiven Achsen des Sensors nicht exakt orthogonal zueinander ausgerichtet sind.

Der Bias \mathbf{b}_a bzw. \mathbf{b}_g setzt sich aus einem statischen Teil zusammen, der bereits beim Einschalten vorliegt, und einem zeitvariablen Anteil, der bei MEMS-Sensoren insbesondere durch Temperatureinflüsse verursacht wird. Auch bei

temperaturkompensierten Sensoren verbleiben Restfehler, die durch den dynamischen Fehleranteil ausgedrückt werden.

Zusätzlich ist ein Rauschen in den gemessenen Größen vorhanden. Dieses entsteht einerseits im Sensor als weißes Rauschen; es wird in Datenblättern als Velocity Random Walk und Angular Random Walk bezeichnet. Beim Einsatz im Flug sind die Sensoren Vibrationen ausgesetzt, die zu weiteren Fehlern führen. Diese werden als additives farbiges Rauschen, welches in w_a und w_g enthalten ist, modelliert.

Der Quantisierungsfehler ist bei den auf der AirQuad-Plattform eingesetzten Sensoren gering und wird daher nicht modelliert.

4.3.0.2 Magnetometer

Die Fehler der Magnetfeldmessung des Magnetometers werden gemäß

$$\tilde{\mathbf{m}} = (\mathbf{I}_3 + \mathbf{M}_m)\mathbf{m} + \mathbf{b}_m + \mathbf{w}_m(t) \quad (4.4)$$

mit Misalignment \mathbf{M}_m , statischem Bias \mathbf{b}_m und weißem Rauschen \mathbf{w}_m abgebildet.

4.3.0.3 Baro-Altimeter

Mit einem barometrischen Höhenmesser kann anhand der Druckdifferenz der Höhenunterschied zum Startpunkt bestimmt werden. Eine absolute Schätzung der Höhe ist nur möglich, wenn der wetterbedingte Luftdruck bei einer definierten Höhe bekannt ist oder z. B. im Rahmen der Datenfusion geschätzt werden kann. Eine wesentliche Fehlergröße ist also der wetterbedingte, aufgrund der kurzen Simulationsdauern als konstant angenommene, Luftdruck. Zudem wurde experimentell festgestellt, dass der Aufwärmprozess im Sensor zu einem signifikanten Driftvorgang während der ersten Flugminuten führt. Dieser verursacht Höhenfehler im Bereich von etwa 4 m und muss daher modelliert werden. Auch weißes Rauschen ist deutlich in realen Messwerten vorhanden, so dass das Fehlermodell

$$\tilde{h}_{alti} = h_{alti} + A(1 - e^{-t/T}) + b_{alti} + w_{alti}(t) \quad (4.5)$$

verwendet wird, wobei h_{alti} die unverfälschte Messgröße darstellt, b_{alti} den Nullpunktsfehler und w_{alti} das weiße Rauschen. A ist die Amplitude und T die Zeitkonstante des temperaturbedingten Driftvorgangs.

4.3.0.4 Laser-Entfernungsmesser

In parallel laufenden Arbeiten wurde die Simulationsumgebung erweitert, so dass auch die Simulation von abbildenden Sensoren wie Kameras und Laserscannern möglich ist [32, 73]. Dazu wurde ein 3D-Umgebungsmodell in die Simulation integriert und abbildende Sensoren mittels OpenGL simuliert. Die von einem Laserscanner in einer Richtung gemessene Entfernung kann dann mit weißem Rauschen überlagert werden.

4.4 Validierung

Die mittels der vorgestellten Simulationsumgebung erhaltenen Ergebnisse können nur dann verwertet werden, wenn sichergestellt ist, dass die Simulation ausreichend genau die realen Gegebenheiten abbildet. Diese Überprüfung geschieht im sog. Validierungsprozess, der in diesem Unterkapitel beschrieben wird.

An erster Stelle muss dabei definiert werden, wofür die Simulationsumgebung eingesetzt werden soll und welche Anforderungen daraus resultieren [72]. Für diese Arbeit sind dies:

- Anwendung 1: Generierung von Trajektorien für spätere Nachverarbeitung. Die Trajektorien sollen insbesondere bzgl. Lagewinkeln und Geschwindigkeit realistische Verläufe aufweisen, da diese Größen wesentlichen Einfluss auf die Beobachtbarkeit der Navigationsgrößen haben.
- Anwendung 2: Nachverarbeitung generierter Trajektorien (vgl. Abschnitt 4.1.2). Hierbei müssen realistische Sensorwerte generiert werden.
- Anwendung 3 (nur Simulationsmodell): Das mathematische Quadroptermodell kommt neben der Simulation auch bei der Navigation im modellgestützten Navigationssystem aus Kapitel 6 zum Einsatz. Daher ist explizit die Gültigkeit des Simulationsmodells zu validieren.

In anderen Arbeiten, bei denen die AirQuad-Simulationsumgebung für ähnliche Zwecke eingesetzt wurde, wurden bereits Validierungen der Simulationsumgebung durchgeführt: Während bei Seibold [73] der Fokus der Validierung auf der Simulation eines Bodenroboters liegt, beschäftigt sich Frietsch [32] mit der Validierung der Simulation für Quadropter. Dabei wird zum einen gezeigt, dass die Simulation der Sensoren reale Effekte gut abbildet, u. a. durch Vergleich der Rauschspektren. Diese

Untersuchungen zeigen, dass die Simulationsumgebung für die o.g. Anwendung 2 (Nachverarbeitung) gültig ist.

Zudem wird in [32] gezeigt, dass eine Wegpunktliste in der Simulation ebenso abgearbeitet wird wie in einem realen Flug. Hierbei ist aber zu beachten, dass es sich dabei um Flüge handelt, bei denen die Positionsverläufe geregelt werden. Selbst bei einem ungenauen mathematischen Simulationsmodell könnten ähnliche Positionsverläufe entstehen, wenn der Regler die Modellungenauigkeiten ausgleicht. Insbesondere zur Validierung des Simulationsmodells, aber auch zur Validierung der Trajektoriengenerierung ohne Positionsregelung, wurden daher weitere Untersuchungen durchgeführt.

4.4.1 Validierung des Simulationsmodells

Zur Validierung des Simulationsmodells wurden zunächst Teilmodelle für die Aktuatoren und die wirkenden Kräfte validiert. Abschnitt 4.4.1.3 präsentiert dann eine geschlossene Simulation, in der das gesamte Simulationsmodell sowie die gesamte im Flug eingesetzte Algorithmik evaluiert wird.

4.4.1.1 Aktuatormodellierung

Die hohe Genauigkeit der Aktuatormodellierung, mit der bei gegebenen Motorspannungen die Motordrehzahlen simuliert werden können, wurde bereits in Abschnitt 3.2.1.3 gezeigt.

4.4.1.2 Kräftemodellierung

Anhand der Drehzahlen und der Geschwindigkeit des Fluggerätes werden gemäß Abschnitt 3.4 die wirkenden Kräfte bestimmt. Abbildung 4.3 zeigt für einen Testflug die Übereinstimmung der modellierten Kräfte mit den gemessenen Kräften. Der wesentliche Unterschied zwischen diesen ist, dass die Vibrationen des Fluggerätes im gezeigten Modell nicht enthalten sind. Dies ist aber gerechtfertigt, da durch die Vibrationen keine nennenswerten Änderungen der Lagewinkel oder der Geschwindigkeit entsteht. Relevant sind die Vibrationen nur dann, wenn sie von den Beschleunigungssensoren gemessen werden. Daher werden die Vibrationen in der Sensorsimulation berücksichtigt.

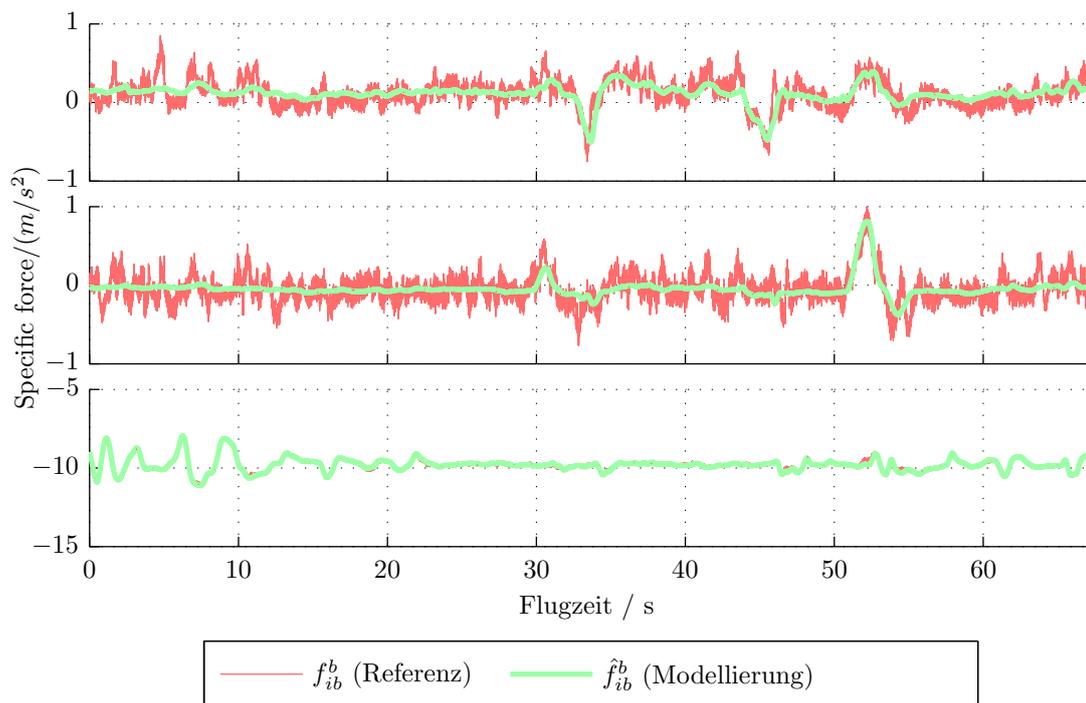


Abbildung 4.3: Vergleich der modellierten mit der gemessenen Specific Force für einen Testflug (von oben nach unten: x^b -, y^b - und z^b -Richtung). Bis auf die unmodellierten Vibrationen und vereinzelt Störeinflüsse stimmen diese gut überein.

4.4.1.3 Vergleich eines realen Flugs mit der Simulation

Um das gesamte Simulationsmodell zu validieren, wurde ein realer Flug durchgeführt und dabei neben den Sensordaten auch die Steuerkommandos aus der Fernbedienung aufgezeichnet. Dieselben Steuerkommandos dienen dann als Eingangsgrößen in der Simulation. Der Pilot gibt in diesem Flug Lagewinkel und Schub vor. Das in Kapitel 5 präsentierte lasergestützte Navigationssystem mit RTS-Smoothing liefert die tatsächlich geflogenen Lagewinkel, Geschwindigkeiten und Positionen. Diese können dann mit den simulierten Trajektorien verglichen werden. Dies ist einem Vergleich der jeweiligen Schätzgrößen aus dem im Flug eingesetzten Navigationssystem vorzuziehen, da sonst die Schätzunsicherheiten im Vergleich berücksichtigt werden müssten.

Aufgrund der im Flug eingesetzten Reglersoftware aus [60] haben stochastische Größen wie die Biase der Beschleunigungsmesser wesentlichen Einfluss auf die tatsächlich erzeugten Lagewinkel. In der Simulation wurden daher 100 Monte-Carlo-Durchläufe mit den gleichen Eingangsgrößen, aber unterschiedlichen Initialisierungen des Zufallszahlengenerators durchgeführt. Dadurch wird der Einfluss

dieser stochastischen Größen auch in der Simulation sichtbar. Abbildung 4.4 zeigt, dass die Lagewinkel des realen Fluges fast durchgängig innerhalb des Bereichs liegen, den die simulierten Flüge abdecken. Abweichungen kommen nur kurzzeitig vor und sind klein. Diese Abweichungen können durch unmodellierte Effekte und externe Störeinflüsse auftreten; sie sind in dieser Größe für die vorliegende Arbeit gut annehmbar.

Untersuchungen zum translatorischen Bewegungsmodell sind in Abbildung 4.5 dargestellt. Diese zeigt die simulierten und real geflogenen Geschwindigkeiten des Testflugs. Die Varianz innerhalb der simulierten Flüge ist bei den Geschwindigkeiten, insbesondere in horizontaler Richtung, relativ hoch. Das liegt daran, dass die Geschwindigkeit keine geregelte Größe ist, denn über die Fernbedienung werden nur Lagewinkel vorgegeben. Das erklärt auch, warum vereinzelt über einige Sekunden Abweichungen zwischen Simulation und Realität vorhanden sind: Wenn durch einen Störeinfluss eine Differenz entsteht, ist anschließend zwar das relative Verhalten der Geschwindigkeit ähnlich, der absolute Wert gleicht sich aber erst nach einer gewissen Zeit wieder an die Realität an. Externe Störeinflüsse sind hier in der Simulation aber nicht enthalten, da ansonsten die Varianz der Testläufe so stark steigen würde, dass praktisch jede real aufgenommene Trajektorie in die Simulation passen würde.

Unter diesen Gesichtspunkten entspricht das Simulationsmodell den aufgenommenen Daten.

4.4.2 Fazit

In der Simulation wird der komplette geschlossene Regelkreis abgebildet: Hierfür werden zum einen die Modelle für die Aktuatoren, die Fluggeräteebewegung und die Sensoren verwendet. Zudem sind auch die später real eingesetzten Softwarekomponenten für Navigation, Regelung und Flugführung Teil des simulierten Systems. Die Ergebnisse zeigen, dass sich sowohl Teilmodelle als auch die gesamte Bewegung des Fluggerätes in der Simulation sehr ähnlich wie im realen Flug verhalten. Somit können die in der Simulation erhaltenen Trajektorien für weitere Untersuchungen genutzt werden (Anwendung 1), Navigationssysteme mit synthetischen Sensordaten verglichen werden (Anwendung 2) und das Simulationsmodell als Grundlage für modellgestützte Navigation dienen (Anwendung 3).

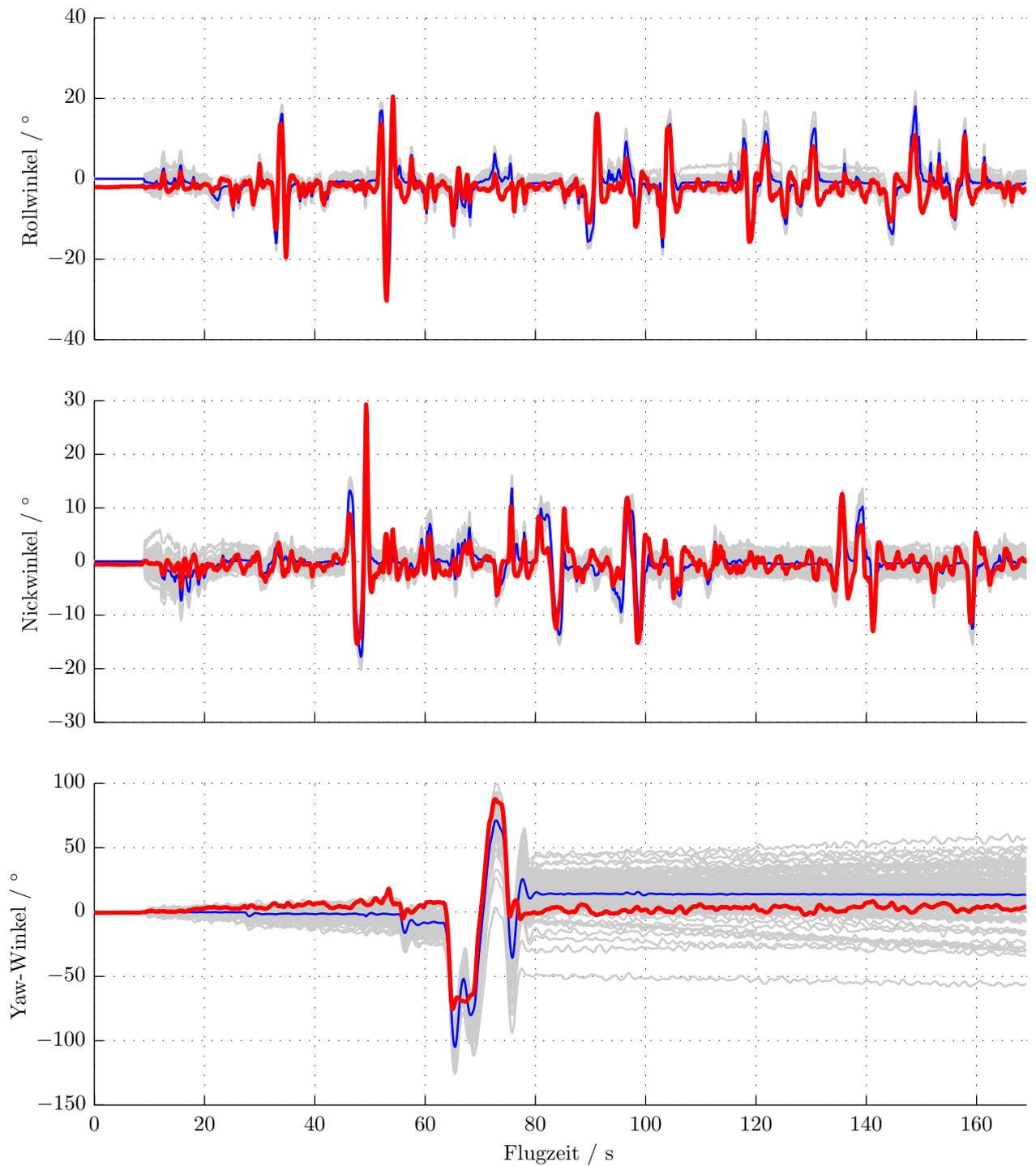


Abbildung 4.4: Vergleich der Lagewinkel von simulierten Flügen und einem realen Flug bei identischen Fernbedienungs-Eingaben. Grau: Einzelne Simulationsläufe mit unterschiedlichem Sensorrauschen; blau: Mittelwert der Simulationsläufe; rot: Realer Flug

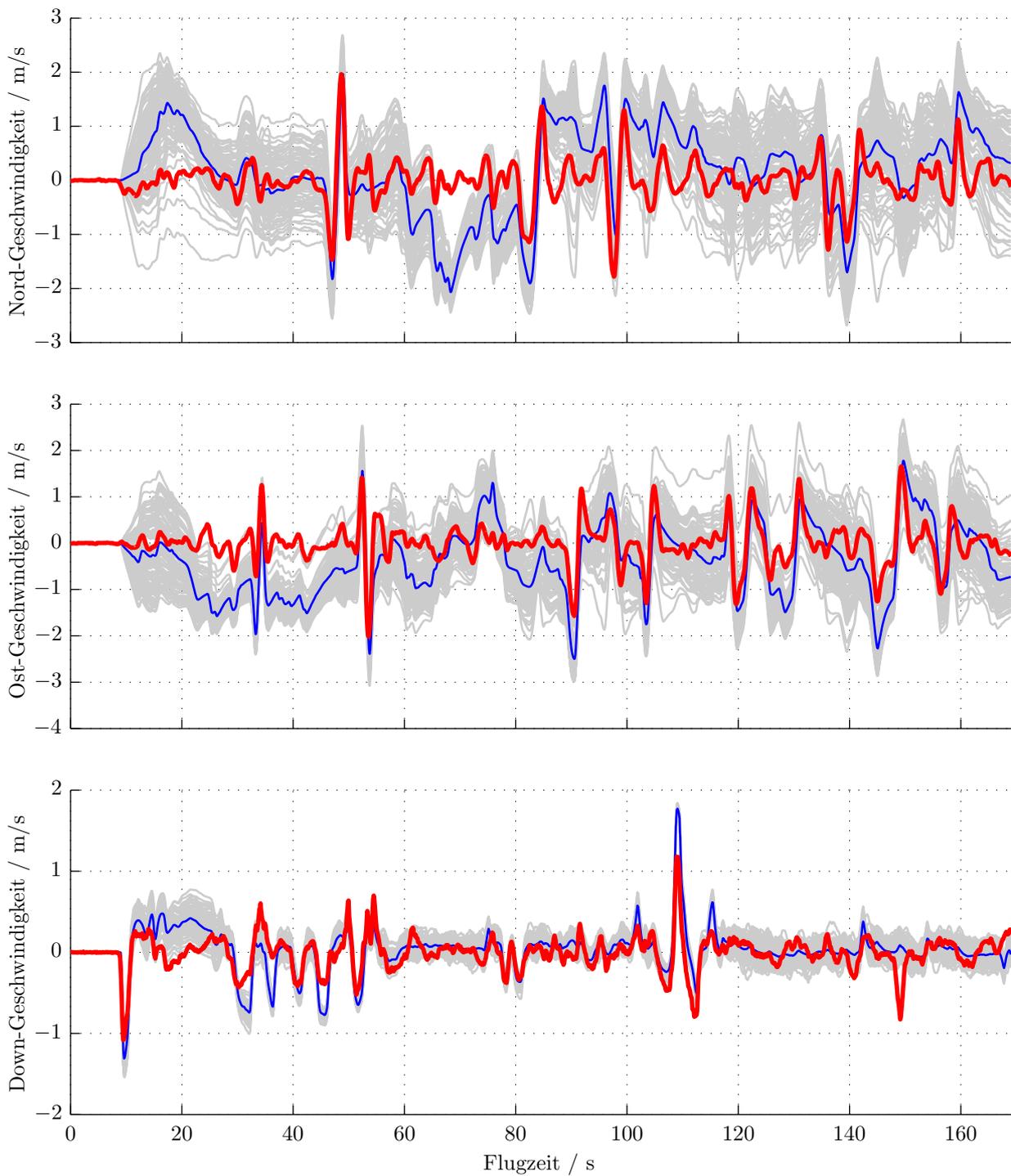


Abbildung 4.5: Vergleich der unregelmäßigen Geschwindigkeiten von simulierten Flügen und einem realen Flug bei identischen Fernbedienungs-Eingaben. Grau: Einzelne Simulationsläufe mit unterschiedlichem Sensorrauschen; blau: Mittelwert der Simulationsläufe; rot: Realer Flug

Kapitel 5

Navigation basierend auf Relativmessungen

Extended Kalman-Filter verarbeiten in Navigationssystemen üblicherweise absolute Größen. Dies sind beispielsweise Positions- und Geschwindigkeitsmessungen aus einem Satellitennavigationssystem. Im Gegensatz zu solchen Satellitennavigationssystemen können Laserscanner oder Kameras – ob in Mono- oder Stereokonfiguration oder gemeinsam mit 3D-Tiefenmessern – nur in Ausnahmefällen absolute Positionsinformationen liefern: Dazu muss nämlich bereits im Vorfeld eine detaillierte Datenbank des zu besuchenden Gebietes erstellt und die aktuelle Position zur Laufzeit darin wiedergefunden werden. Das Anlegen einer solchen Datenbank ist aber zeit- und kostenintensiv und daher oft unpraktikabel. Doch auch ohne dieses aufwendige Anlegen einer Datenbank kann aus den Messungen von Laserscannern und Kameras Information gewonnen werden, die zur Navigationsstützung dienen kann: Durch Vergleich von zeitversetzten Aufnahmen kann die Relativbewegung zwischen den Aufnahmezeitpunkten extrahiert werden.

Solche Posenänderungen stellen aber keine absolute Größe mehr dar. Eine direkte Verarbeitung im Kalman-Filter ist somit nicht möglich. Im Folgenden werden verschiedene Lösungsansätze zur Verarbeitung der Posenänderungen diskutiert und simulativ verglichen (Unterkapitel 5.2). Dies erlaubt es, die Charakteristika der Lösungen aufzuzeigen und die geeignetste Lösung zu identifizieren. Es wird gezeigt, dass diese Lösung beschränkte Geschwindigkeits- und Lagefehler aufweist. Dies ist beim Einsatz auf Quadroptern sehr wichtig, da die Flugregler die Geschwindigkeits- und Lageschätzung benötigen.

Anschließend wird die gewählte Methode noch erweitert, um auch eine zeitliche Verzögerung zwischen Gültigkeit und Verfügbarkeit der Relativmessung zu erlauben. Dies ist in Echtzeitsystemen notwendig, da die Berechnung der Positions- und

Lageänderung auf den heute eingesetzten mobilen Rechenplattformen nicht innerhalb eines Zyklus in der Programm-Hauptschleife erfolgen kann. Stattdessen läuft dieser Prozess im Hintergrund ab, was bereits eine Zeitverzögerung mit sich bringt, die zudem durch weitere Verzögerungen, z. B. im Sensortreiber, noch vergrößert wird. Das konkret entwickelte Navigationssystem wird in Unterkapitel 5.3 vorgestellt.

Navigation mittels Relativmessungen hat die inhärente Eigenschaft, dass die Positionsunsicherheit mit der Zeit tendenziell zunimmt. Eine Abnahme der Unsicherheit ist nur dann möglich, wenn ein voriger Anstieg rückwirkend reduziert werden kann. Eine Zielsetzung für die Navigation ist somit, den Anstieg bestmöglich zu begrenzen, um möglichst viele Anwendungsbereiche zu erschließen. Ein hierfür geeigneter Ansatz wird in Unterkapitel 5.4 beschrieben: Wenn nämlich erkannt wird, dass das Fahrzeug nach längerer Zeit an einen früheren Ort zurückkehrt (Schleifenschluss oder Loop Closure), kann die während der Schleife aufgebaute Unsicherheit wieder eliminiert werden. Dies ist aber in einem Kalman-Filter nicht in Echtzeit realisierbar. Daher wird im Unterkapitel 5.4.1 ein graphenbasierter Ansatz entwickelt, der für die Datenfusion auf einen Graphen mit Covariance Intersection zurückgreift.

Um nach dem Flug weiter verbesserte Ergebnisse berechnen zu können, wird in Unterkapitel 5.5 für das System mit Ortswiedererkennungen ein Smoother-Ansatz vorgestellt.

5.1 Verwandte Arbeiten

Umfelderfassende Sensoren werden auf kleinen unbemannten Fluggeräten sehr häufig eingesetzt. Zum Einsatz kommen Kameras in Mono- und Stereokonfiguration (s. z. B. [24, 52] für Mono, [31, 33] für Stereo), Tiefenbildkameras (z. B. [43, 49]), Laserscanner (z. B. [8, 10, 39]) und Kombinationen dieser Sensoren (z. B. [7, 76]). Auch wenn die Sensorkonfigurationen unterschiedlich sind, liefern sie doch immer eine Art von Merkmalsmessungen. Diese sollen genutzt werden, um eine volle Navigationslösung bestehend aus Position, Geschwindigkeit und Lage zu berechnen. Daher sind viele typische SLAM-Verfahren (vgl. [11, 27]) eher ungeeignet, da diese meist nur Posenschätzungen in niedriger Taktrate ausgeben, für die Flugregelung aber eine hochfrequente, volle Navigationslösung erforderlich ist.

Im Folgenden soll ein Überblick über einige der Methoden gegeben werden, die sich für unpräparierte und zuvor unbekannte Umgebungen eignen; dies muss für

eine Aufklärungsplattform vorausgesetzt werden, so dass andere Verfahren hier nicht weiter relevant sind.

Tomić präsentiert in [81] ein Kalman-Filter basiertes Navigationssystem, dessen Datenfusionsansatz dem weiter unten dargestellten und in [104] präsentierten System ähnelt. Durch den Einsatz der Stochastic-Cloning-Technik werden die Relativmessungen mathematisch korrekt verarbeitet, selbst wenn diese verzögert eintreffen. Das Verfahren nutzt jedoch keine intelligente Auswahl von Referenzzeitpunkten und behandelt das Problem des Schleifenschlusses nicht.

Achtelik [7] und Bachrach [10] nutzen Lasermessungen, um daraus mittels sog. Laser-Odometrie eine Geschwindigkeit zu berechnen. Das Kalman-Filter, in dem diese Geschwindigkeitsdaten fusioniert werden, läuft mit nur 10 bis 50 Hz. Da die aus dem Laser ermittelte Posenänderung in eine Geschwindigkeit umgerechnet wird, können keine älteren Referenzzeitpunkte verwendet werden, was das Fehlerwachstum erhöht. Die Kopplung mit einem ebenfalls vorhandenen SLAM-Verfahren wird in den Papern nicht diskutiert. Dadurch ist nicht klar, ob eine konsistente Positionsschätzung im Kalman-Filter gewährleistet werden kann, was aber eine wichtige Anforderung ist [20].

In anderen Arbeiten werden mehrere Filter genutzt, z. B. um Roll- und Nickwinkel, Geschwindigkeit und Pose getrennt voneinander zu schätzen. Dies erlaubt zwar eine modulare Softwarestruktur, verhindert aber die optimale Datenfusion, bei der alle Messungen alle Schätzgrößen verbessern. In [86] wird die Geschwindigkeitsschätzung von der Positionsschätzung getrennt. In [22, 31, 37, 38] werden Roll- und Nickwinkel direkt aus den IMU-Messwerten berechnet, was aber in der klassischen Navigation üblicherweise vermieden wird (vgl. Abschnitt 2.2.2), da die Lageschätzung im Navigations-Kalman-Filter präziser erfolgen kann.

Aus diesem Grunde wählt Chowdhary [24] explizit einen EKF-SLAM-Ansatz, bei dem das klassische Navigationsfilter um weitere Zustände zur Schätzung von Merkmalspositionen erweitert wird. Der EKF-SLAM-Ansatz, der auch in [44] eingesetzt wird, ist allerdings nicht gut geeignet, um das Schleifenschlussproblem zu lösen: Zum einen wächst der Rechenaufwand mit der Zeit drastisch an, wenn alte Merkmale im Zustandsvektor behalten werden, was für Verarbeitung von Ortswiedererkennungen aber nötig ist. Zweitens wurde gezeigt, dass EKF-SLAM aufgrund der Linearisierung inkonsistente Karten liefern kann [29, 45]. Daher überrascht es nicht, dass [24, 44] keine Ortswiedererkennungen verarbeiten.

Ein Verfahren zum Wiedererkennen von Orten mit dreidimensionalen Merkmalen wird in [39] präsentiert, allerdings ohne Konzept für die Datenfusion.

Leishman [57] nutzt ein Keyframe-basiertes Koordinatensystem, das es erlaubt, Relativmessungen zu verarbeiten, ohne die Stochastic-Cloning-Technik (oder eine weniger geeignete Alternative) anzuwenden. Dazu wird, wie auch in Unterkapitel 5.4, ein Posengraph genutzt. Dieser Ansatz ist in Umgebungen, wo gar keine Satellitenstützung möglich ist, gut geeignet. Allerdings können GPS-Messungen, wenn sie vorhanden sind, nicht im Kalman-Filter verarbeitet werden.

5.1.1 Fazit

Obwohl unterschiedliche Ansätze zur Verarbeitung von Relativmessungen implementiert wurden, fehlte in der Literatur ein Vergleich, der die Verfahren bezüglich der erzielten Genauigkeit gegenüberstellte. Ein solcher Vergleich soll in dieser Arbeit präsentiert werden.

Auch wenn das im vorangegangenen Vergleich ausgewählte Verfahren, Stochastic Cloning, bereits in der Literatur eingesetzt wird, sollen Wege zur besseren Ausnutzung der verfügbaren Informationen entwickelt und aufgezeigt werden.

Wenn Ortswiedererkennungen verarbeitet werden sollen, ist es bei keinem der in der Literatur gefundenen Verfahren möglich, diese so zu verarbeiten, dass weiterhin eine Kalman-Filter-zentrierte Schätzarchitektur verwendet werden kann, die eine volle Navigationslösung mit hoher Taktrate liefert. Eine solche Architektur soll daher entworfen werden, um eine optimale Schätzung der Navigationsgrößen und der inertialen Biase zu ermöglichen.

5.2 Posenänderungen als Messung im Kalman-Filter

Durch Vergleich von Bild- oder Laseraufnahmen aus verschiedenen Zeitpunkten kann die Änderung der Pose des Sensors bzw. der Sensorplattform ermittelt werden. Je nach Art des Sensors handelt es sich hier um eine 3D- oder 2D-Posenänderung. 3D-Posenänderungen, d.h. Änderungen von 3D-Position und 3D-Lage, können basierend auf dreidimensionalen Punktwolken gewonnen werden. Diese dreidimensionalen Punktwolken fallen beispielsweise bei Stereokameras, Kinect-Sensoren oder 3D-Laserscannern an. Aus den auf fliegenden Plattformen üblicherweise eingesetzten LIDAR-Systemen (engl. light detection and ranging) oder aus Daten einer nach unten gerichteten Kamera dagegen ist die Extraktion einer 2D-Posenänderung üblich, d.h.

einer Verschiebung in der Horizontalen und einer Änderung des Heading-Winkels. Im Folgenden wird davon ausgegangen, dass eine solche 2D-Posenänderung vorliegt. Eine Erweiterung der Gleichungen auf den 3D-Fall ist aber leicht möglich.

Eine Messung einer Posenänderung lässt sich mathematisch darstellen als

$$\tilde{\mathbf{y}}_k^{h_0} = \begin{pmatrix} p_{x_k}^{h_0} - p_{x_{k_0}}^{h_0} \\ p_{y_k}^{h_0} - p_{y_{k_0}}^{h_0} \\ \psi_k - \psi_{k_0} \end{pmatrix} + v_k. \quad (5.1)$$

Hierbei bezeichnet $\tilde{\mathbf{y}}_k^{h_0}$ den Messvektor. Die Indizes k_0 und k stehen für die Zeitpunkte k_0 und die aktuelle Zeit k . Zu diesen Zeitpunkten wurden die jeweiligen Messungen erfasst. Das hochgestellte h_0 definiert, dass die Messung im horizontalen Koordinatensystem zum Zeitpunkt k_0 vorliegt; dieses horizontale Koordinatensystem geht durch Drehung um den Heading-Winkel aus dem n-frame hervor. Die dritte Komponente des Messvektors beinhaltet die Änderung des Gierwinkels. Aus den ursprünglichen Bild- oder Laserdaten sollte neben dem Messvektor $\tilde{\mathbf{y}}_k^{h_0}$ auch die Kovarianz des Fehlers $R_k = E\{v_k v_k^T\}$ ermittelt werden. v_k wird hierbei als Zufallsvariable aus einem weißen gaußschen Rauschprozess angenommen.

Ein wie in Abschnitt 2.2.1 definiertes Error-State-Kalman-Filter schätzt den Zustand

$$\hat{\mathbf{x}}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{q}_k \\ \mathbf{b}_{a,k} \\ \mathbf{b}_{w,k} \end{pmatrix}, \quad (5.2)$$

der aber die in der Messgleichung (5.1) vorkommenden Größen $p_{x_{k_0}}$, $p_{y_{k_0}}$ und ψ_{k_0} nicht enthält. Die Messgleichung kann also nicht auf übliche Weise im Kalman-Filter implementiert werden. In der Literatur finden sich mehrere Ansätze, die Messung dennoch zu verarbeiten. Da ein Vergleich dieser Ansätze in der Literatur fehlte, wurde ein solcher im Rahmen dieser Arbeit durchgeführt [105]. Dieser wird im Folgenden präsentiert, erweitert um Ergänzungen, die in [98] vorgestellt wurden. Dazu werden zunächst die Methoden zur Verarbeitung von Posenänderungsmessungen dargestellt und diese dann simulativ verglichen.

5.2.1 Möglichkeiten der Messwertverarbeitung

5.2.1.1 Verarbeitung als Mittlere Geschwindigkeit

Ein üblicher Ansatz, die Messung $\tilde{\mathbf{y}}_k^{h_0}$ einem Kalman-Filter zuzuführen, besteht darin, den Messwert von einer Posenänderung in eine Geschwindigkeit zu transformieren. Die neue Messgröße berechnet sich als

$$\tilde{\mathbf{y}}_{k,velo}^n = \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix} \tilde{\mathbf{y}}_k^{h_0} / (t_k - t_{k_0}) \quad (5.3)$$

mit der zugehörigen Kovarianzmatrix

$$\mathbf{R}_{velo} = \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix} \mathbf{R} \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix}^T / (t_k - t_{k_0})^2. \quad (5.4)$$

Der Term $t_k - t_{k_0}$ bezeichnet hierbei die Zeitdifferenz in Sekunden zwischen den Aufnahmezeitpunkten k und k_0 . $\mathbf{C}_{h_0}^{ne}$ ist die Rotationsmatrix, die einen Vektor aus dem horizontalen Body-Koordinatensystem des Zeitpunkts k_0 in das aktuelle Nord-Ost-Koordinatensystem dreht.

Dieser neue Messvektor enthält nun die horizontale Geschwindigkeit und die Drehrate um die down-Achse. Diese Drehrate kann aber beim vorliegenden Zustandsvektor nicht verarbeitet werden, da nur Lagewinkel, aber keine Drehraten im Zustand enthalten sind. Daher werden nur die ersten zwei Komponenten des Messvektors berücksichtigt. Die Ableitung der Messgleichung nach den Zustandsgrößen ergibt die Messmatrix

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}. \quad (5.5)$$

Diese Art der Messwertverarbeitung ist einfach zu implementieren und stellt keine hohen Anforderungen an den Rechner. Allerdings wird dabei implizit angenommen, dass die aktuelle Geschwindigkeit der mittleren Geschwindigkeit im Zeitraum zwischen den Messungen entspricht. Ob diese Annahme gerechtfertigt ist, muss im Einzelfall geprüft werden. Wie stark diese vereinfachende Annahme verletzt wird, ist wesentlich von der Dynamik und Trajektorie des Fluggerätes sowie der

Sensorupdaterate abhängig. Dementsprechend sollte das Messrauschen \mathbf{R}_{velo} bei diesem Ansatz mit einem zusätzlichen Term beaufschlagt werden.

5.2.1.2 Verarbeitung als pseudo-absolute Pose

Anstatt die Posenänderung in eine Geschwindigkeitsmessung zu transformieren, besteht auch die Möglichkeit der Verarbeitung als pseudo-absolute Pose. Hierzu wird die gemessene Posenänderung auf die zum Zeitpunkt k_0 geschätzte Pose addiert, was die zum Zeitpunkt k erwartete Pose $\tilde{\mathbf{y}}_{k_{absolute}}^n$ ergibt:

$$\tilde{\mathbf{y}}_{k_{absolute}}^n = \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix} \tilde{\mathbf{y}}_k^{h_0} + \begin{pmatrix} \hat{p}_{x_{k_0}} \\ \hat{p}_{y_{k_0}} \\ \hat{\psi}_{k_0} \end{pmatrix} \quad (5.6)$$

Mit der Messmatrix

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix} \quad (5.7)$$

kann auch dieser Wert leicht als Messung im Kalman-Filter genutzt werden. Bei dieser Vorgehensweise werden allerdings die Korrelationen zwischen dem Messvektor $\tilde{\mathbf{y}}_{k_{absolute}}^n$ und dem Zustandsvektor $\hat{\mathbf{x}}_k$ ignoriert. Daher sollte wiederum die Kovarianzmatrix erhöht werden, z. B. durch Wahl eines skalaren Parameters $s \geq 1$:

$$\mathbf{R}_{absolute} = s \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix} \mathbf{R} \begin{pmatrix} \mathbf{C}_{h_0}^{ne} & 0 \\ 0 & 1 \end{pmatrix}^T \quad (5.8)$$

5.2.1.3 Stochastic Cloning

Das sog. Stochastic Cloning wurde in [70] vorgeschlagen, um relative Messungen zu verarbeiten. Erweiterungen dazu wurden in [62] beschrieben. Ebenfalls bekannt ist diese Technik als Delay States [34].

Der Grundgedanke hierbei ist es, den Zustandsvektor um einen sogenannten Zustandsklon zu erweitern:

$$\check{\mathbf{x}}_{k_0} = \begin{pmatrix} \mathbf{x}_{k_0} \\ \mathbf{x}_{k_0} \end{pmatrix}, \check{\mathbf{P}}_k = \begin{pmatrix} \mathbf{P}_{k_0} & \mathbf{P}_{k_0} \\ \mathbf{P}_{k_0} & \mathbf{P}_{k_0} \end{pmatrix} \quad (5.9)$$

Diese Erweiterung wird durchgeführt, wenn Kamera oder Laser die erste Messung zum Zeitpunkt k_0 durchführen. Der neue Zustandsvektor $\check{\mathbf{x}}$ wird fortan mit der ebenfalls erweiterten Systemmatrix

$$\check{\Phi} = \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (5.10)$$

und der erweiterten Rauscheingangsmatrix

$$\check{\mathbf{G}} = \begin{pmatrix} \mathbf{G} \\ \mathbf{0} \end{pmatrix} \quad (5.11)$$

propagiert. Dadurch ergibt sich zum Zeitpunkt k der Vektor

$$\check{\mathbf{x}}_k = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k_0} \end{pmatrix}, \quad \check{\mathbf{P}}_k = \begin{pmatrix} \mathbf{P}_{kk} & \mathbf{P}_{k_0k} \\ \mathbf{P}_{k_0k} & \mathbf{P}_{k_0k_0} \end{pmatrix}. \quad (5.12)$$

Messungen, die nur \mathbf{x}_k betreffen, werden wie üblich verarbeitet. Durch die in \mathbf{P}_{k_0k} ausgedrückte Korrelation zwischen \mathbf{x}_k und \mathbf{x}_{k_0} wird dabei jedoch auch der Zustandsklon \mathbf{x}_{k_0} aktualisiert. Dies ist eine wichtige Eigenschaft, wenn später Messungen verarbeitet werden, die auch \mathbf{x}_{k_0} betreffen; ohne diese Korrektur könnten sonst Inkonsistenzen im Filter auftreten.

Zur Verarbeitung der Messung einer Posenänderung, $\tilde{\mathbf{y}}_k^{h_0}$, wird die Messgleichung wie folgt umgeschrieben, um die auftretenden Größen durch die Schätzwerte und die zu schätzenden Fehler auszudrücken:

$$\tilde{\mathbf{y}}_k^{h_0} = \begin{pmatrix} p_{x_k}^{h_0} - p_{x_{k_0}}^{h_0} \\ p_{y_k}^{h_0} - p_{y_{k_0}}^{h_0} \\ \psi_k - \psi_{k_0} \end{pmatrix} + \mathbf{v}_k \quad (5.13)$$

$$= \begin{pmatrix} \mathbf{C}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{x_k}^n - p_{x_{k_0}}^n \\ p_{y_k}^n - p_{y_{k_0}}^n \\ \psi_k - \psi_{k_0} \end{pmatrix} + v_k \quad (5.14)$$

$$= \begin{pmatrix} \hat{\mathbf{C}}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\delta\hat{\psi}_{k_0}) & -\sin(\delta\hat{\psi}_{k_0}) & 0 \\ \sin(\delta\hat{\psi}_{k_0}) & \cos(\delta\hat{\psi}_{k_0}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{p}_{x_k}^n - \hat{p}_{x_{k_0}}^n - (\delta\hat{p}_{x_k}^n - \delta\hat{p}_{x_{k_0}}^n) \\ \hat{p}_{y_k}^n - \hat{p}_{y_{k_0}}^n - (\delta\hat{p}_{y_k}^n - \delta\hat{p}_{y_{k_0}}^n) \\ \hat{\psi}_k - \hat{\psi}_{k_0} - (\delta\hat{\psi}_k - \delta\hat{\psi}_{k_0}) \end{pmatrix} + \mathbf{v}_k \quad (5.15)$$

Zur Berechnung der Kalman-Filter-Messmatrix wird diese Gleichung approximiert durch die Taylorreihe erster Ordnung in den Fehlergrößen:

$$\begin{aligned} \tilde{\mathbf{y}}_k^{h_0} \approx & \begin{pmatrix} \hat{\mathbf{C}}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{p}_{x_k}^n - \hat{p}_{x_{k_0}}^n \\ \hat{p}_{y_k}^n - \hat{p}_{y_{k_0}}^n \\ \hat{\psi}_k - \hat{\psi}_{k_0} \end{pmatrix} \\ & + \begin{pmatrix} \hat{\mathbf{C}}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & -(\hat{p}_{y_k}^n - \hat{p}_{y_{k_0}}^n) \\ 0 & -1 & 0 & \hat{p}_{x_k}^n - \hat{p}_{x_{k_0}}^n \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} \delta \hat{p}_{x_k}^n - \delta \hat{p}_{x_{k_0}}^n \\ \delta \hat{p}_{y_k}^n - \delta \hat{p}_{y_{k_0}}^n \\ \delta \hat{\psi}_k \\ \delta \hat{\psi}_{k_0} \end{pmatrix} \quad (5.16) \end{aligned}$$

Von dieser Form lässt sich die Messmatrix leicht ablesen. Interessanterweise wird bei Verarbeitung dieser Gleichung auch der Zustandsklon \mathbf{x}_{k_0} aktualisiert, obwohl dies intuitiv falsch erscheinen mag: Eine Messung der Posenänderung zwischen den Zeitpunkten k_0 und k scheint keine neue Information über die Pose zum Zeitpunkt k_0 zu enthalten. Tatsächlich ist dies aber der Fall: Ähnlich wie eine Geschwindigkeitsmessung, z. B. aus GPS, eine Position im Kalman-Filter korrigieren kann, kann auch eine Messung der Posenänderung eine Pose korrigieren. Diese Analogie ist naheliegend, da für sehr kleine Δt die Posenänderungsmessung in eine Geschwindigkeitsmessung übergeht.

Die obige Herleitung zeigt, dass die Messgleichung nichtlinear ist und im Kalman-Filter nur die Approximation durch Linearisierung verwendet wird. Ebenso wie bei den anderen Ansätzen zur Messwertverarbeitung ist also auch hier eine Erhöhung des Messrauschens angebracht. Sonst würden die Linearisierungsfehler im Kalman-Filter nicht berücksichtigt.

In der gezeigten Herleitung wurde aus Gründen der Lesbarkeit davon ausgegangen, dass zum Zeitpunkt k_0 der gesamte Zustandsvektor geklont wird. Zur Verarbeitung von zweidimensionalen Posenänderungen ist es jedoch ausreichend, nur die Zustände p_x , p_y und ψ zu klonen, so dass die Größe des Zustandsvektors durch den Klonvorgang nur geringfügig steigt. Nach Verarbeitung einer Messung werden die nicht mehr benötigten Klone aus dem Zustandsvektor entfernt oder ggf. durch neue Klone ersetzt.

5.2.1.4 Wahl des Referenzzeitpunktes

Bisher wurde davon ausgegangen, dass die aktuelle Laser- oder Kameraaufnahme mit der vorherigen verglichen wird, um die Posenänderung zu bestimmen. Dies ist für

die spätere Verarbeitung als Mittlere Geschwindigkeit auch der einzig sinnvolle Ansatz. Für eine Verarbeitung als pseudo-absolute Pose oder mittels Stochastic Cloning ist es dagegen auch möglich, die aktuelle Aufnahme mit einer u. U. deutlich älteren Aufnahme zu vergleichen: Solange das Matching dann immer noch erfolgreich durchgeführt werden kann, wächst die Unsicherheit des Filters und der zugehörige Fehler deutlich langsamer an, da weniger inkrementelle Fehler gemacht werden. Im Extremfall, dass jede Aufnahme mit der ersten verglichen werden kann, ergibt sich sogar eine beschränkte Unsicherheit.

5.2.1.5 Mehrere Referenz-Kandidaten

Die Auswahl eines Referenzzeitpunktes ist nicht trivial: Länger zurückliegende Zeitpunkte sind insofern zu bevorzugen, als dass die Schätzunsicherheit zum Referenzzeitpunkt tendenziell geringer ist und somit auch die aktuelle Pose besser geschätzt werden kann. Allerdings ist zu beachten, dass das Matching von Aufnahmen, die zeitlich weit auseinander liegen, i. d. R. schwieriger durchzuführen ist als bei zeitlich sehr nahen Aufnahmen – schließlich nimmt die Ähnlichkeit mit wachsender Zeitdifferenz meist ab. Wenn dadurch das Matching fehlschlägt oder mit großer Unsicherheit behaftet ist, könnte es folglich in manchen Fällen sinnvoll sein, einen neueren Zeitpunkt als Referenz zu benutzen. Die Güte des Matchings kann vorher aber nur begrenzt geschätzt werden. Daher soll hier ein System vorgestellt werden, mit dem die Entscheidung für einen Referenzzeitpunkt erst dann getroffen werden muss, wenn alle nötigen Informationen vorliegen.

Der Grundgedanke dieses Ansatzes besteht darin, mehrere Kandidaten als Referenzzeitpunkte zu nutzen: Im Zustandsvektor ist also nicht nur ein Zustandsklon enthalten, sondern mehrere, z. B. drei. Somit existieren auch mehrere mögliche Referenzzeitpunkte. Bei Verfügbarkeit einer neuen Aufnahme wird diese dann mit den Aufnahmen aller möglichen Referenzzeitpunkte verglichen, d. h. es entstehen mehrere Posenänderungsmessungen. Diese können aber nicht alle verarbeitet werden, da sie alle von der aktuellen Aufnahme, d. h. auch den darin enthaltenen Messfehlern, abhängig sind. Die Messfehler sind also korreliert. Stattdessen wird die beste Messung ausgewählt, indem die entsprechende Kalman-Filter-Messgleichung für alle Messungen ausgewertet wird. Die Messung mit der geringsten a-posteriori-Kovarianz für die Positionsschätzung wird dann behalten, die anderen Messungen werden verworfen.

Je nach Art der Aufnahmen müssen geeignete Kriterien definiert werden, wann welcher Zustandsklon behalten oder erneuert wird. Insbesondere sollte natürlich sichergestellt werden, dass die Zustandsklone unterschiedliche Zeitpunkte repräsentieren und dass immer für mindestens einen Referenzzeitpunkt das Matching gelingt. Konkrete Kriterien für den Fall eines lasergestützten Navigationssystems sind im Abschnitt 5.3.1.6 beschrieben.

5.2.2 Vergleich der Verarbeitungsmethoden

Die oben vorgestellten Methoden zur Verarbeitung von Posenänderungen als Messung wurden in dieser Arbeit simulativ verglichen. Hierzu kam die in Kapitel 4 vorgestellte Simulationsumgebung zum Einsatz.

5.2.2.1 Trajektorien

Mittels manueller Steuerung wurden Trajektorien für mehrere Datensätze vorgegeben, die typisches Verhalten von Quadroptern und vergleichbaren Fluggeräten abbilden. Das Einstellen der Filterparameter und die Auswertung der Verfahren wurde dann anhand unterschiedlicher Datensätze durchgeführt. Der zur Auswertung genutzte Flug beginnt mit einer ruhigen Flugphase mit sehr geringer Dynamik und weist später hohe Dynamik und größere Lenkwinkel auf. Dadurch soll geklärt werden, inwiefern sich unterschiedliche Flugdynamik auf das Fehlerverhalten der Algorithmen auswirkt.

5.2.2.2 Simulation der Sensordaten

Für jeden der 100 durchgeführten Monte-Carlo-Läufe wurden – passend zur zuvor erstellten Trajektorie – folgende Sensordaten generiert: Die Inertialmessungen bilden das Fehlerverhalten der ADIS 16488A von Analog Devices [92] ab, wobei die auf Quadroptern vorkommenden Vibrationen berücksichtigt wurden. Zudem wurde ein barometrischer Höhenmesser und ein Magnetometer simuliert.

Da bei relativen Navigationssystemen selbst geringe Heading-Fehler zu signifikanten Positionsfehlern führen können, wurde ein perfektes Magnetometer simuliert. Dadurch kann der Heading-Fehler minimiert werden und die Positionsfehler resultieren tatsächlich aus den unterschiedlichen Verarbeitungsmethoden der 2D-Posenänderungsinformation. Um den eigentlichen Unterschied zwischen den Verarbeitungsmethoden weiter hervorzuheben, wurde sogar darauf verzichtet, die Heading-Änderung zu verarbeiten, weil dies nicht bei allen Verfahren möglich wäre.

Um die stochastischen Eigenschaften der Fehler in den Posenänderungsmessungen genau vorgeben zu können, wurden die simulierten Messgrößen direkt aus den tatsächlichen Posenänderungen generiert, überlagert mit einem Fehler, der als weißes Rauschen angenommen wurde. Bei Scan-Matching-Algorithmen nimmt die Güte der gelieferten Posenänderungsschätzung mit abnehmender Überlappung der verglichenen Laserscans ab. Dieser Effekt wurde nachgebildet, indem die Varianz des Fehlers proportional zur Entfernung zwischen aktueller Position und Referenzposition erhöht wurde. Die Posenänderungen wurden mit einer Rate von 10 Hz generiert, wobei jeweils eine Bestimmung der Posenänderung zum vorigen Referenzzeitpunkt sowie eine zu einem länger gehaltenen Referenzzeitpunkt simuliert wurde. Die untersuchten Filter erhalten dann entweder eine dieser Messungen oder beide, um daraus auswählen zu können.

Die so erzeugten Sensordaten wurden als Eingangsgrößen für alle zu testenden Navigationsfilter verwendet.

5.2.2.3 Bestimmung der Filterparameter

Um einen Einfluss von manuellem Filter-Tuning auf das Ergebnis zu vermeiden, wurden die Tuning-Parameter der zu vergleichenden Filter automatisiert bestimmt. Dazu wurde eine Parameterbestimmung gemäß [5] durchgeführt. Zuerst wurden mittels eines simulierten Fluges mit GPS-Stützung die Werte des Systemrauschens bestimmt. Diese Werte beschreiben das im Filter angenommene Fehlerverhalten der IMU. Sie wurden dann für alle zu untersuchenden Filter genutzt, die die unterschiedlichen Methoden zur Verarbeitung von Posenänderungen nutzen.

Die Messwert-Kovarianz wird zwar vom Matching-Algorithmus geliefert. Allerdings wurde in der Beschreibung der Ansätze deutlich, dass zusätzlich zum eigentlichen Messfehler auch Fehler bei der Verarbeitung der Messung gemacht werden, da Bedingungen des Kalman-Filters verletzt werden. Um dem Rechnung zu tragen, wurde ein Skalierungsfaktor für das Messrauschen mittels dem automatisierten Verfahren aus [5] erlernt. Als Gütemaß wurden zwei Kriterien verwendet: So sollte zum einen der RMS-Positionsfehler minimiert als auch das sog. Prediction Likelihood maximiert werden. Bei letzterem spielt neben dem Schätzfehler selbst auch die Schätzung der Kovarianzmatrix eine Rolle. Wenn die beiden Kriterien zu unterschiedlichen vorgeschlagenen Parametern führten, wurde der Parameterwert gewählt, der auch beim anderen Kriterium ein akzeptables Gütemaß lieferte.

Verfahren	Skalierungsfaktor für Messwert-Kovarianz
Mittlere Geschwindigkeit	14,34
Pseudo-absolute Position	6,20
Stochastic Cloning (ohne Halten des Referenz-Scans)	1,40
Stochastic Cloning (mit Halten des Referenz-Scans)	2,35

Tabelle 5.1: Mittels automatischem Filter-Tuning gelernte Parameter für die Messwertverarbeitung. Diese kompensieren die Verletzung von Kalman-Filter-Voraussetzungen.

Tabelle 5.1 listet die für die Skalierungsfaktoren bestimmten Werte auf. Bei den Verfahren „Mittlere Geschwindigkeit“ und „Pseudo-absolute Position“ muss die Kovarianz stark erhöht werden. Beim Einsatz von Stochastic Cloning genügt eine moderate Erhöhung der Kovarianz, wobei die Erhöhung etwas stärker ausfällt, wenn ältere Referenzzeitpunkte genutzt werden. Dieser Effekt ist darin begründet, dass bei Verarbeitung größerer Posenänderungen auch die Linearisierungsfehler größeren Einfluss haben.

5.2.2.4 Ergebnisse

Zunächst soll die Güte der Navigationsfilter betrachtet werden, die immer nur Posenänderungen zwischen direkt aufeinanderfolgenden Zeitpunkten verarbeiten. Dies kann als Geschwindigkeitsmessung gemäß Abschnitt 5.2.1.1 erfolgen, als Verarbeitung als pseudo-absolute Position (Abschnitt 5.2.1.2), oder tatsächlich als Verarbeitung der Posenänderung, wenn Stochastic Cloning gemäß Abschnitt 5.2.1.3 genutzt wird.

Die Entwicklung des RMS-Fehlers der horizontalen Positionsschätzung für diese Filter ist in Abbildung 5.1 dargestellt. Die Filter weisen deutlich unterschiedliche Verläufe auf: So ist bei Verarbeitung der Messung als Geschwindigkeit zu erkennen, dass ab 100 s, also wenn die Phase des dynamischen Fluges beginnt, die Fehler deutlicher schneller anwachsen als bei geringer Dynamik in der ersten Hälfte des Fluges (rote Kurve). Dieses Verhalten war aus theoretischer Sicht erwartet worden; die Simulation zeigt, dass der Fehlerzuwachs auch praktisch relevant ist. Bei dieser Art der Verarbeitung ist das System zu pessimistisch: Die erwarteten Positionsfehler sind größer als der tatsächliche Fehler.

Bei Verarbeitung der Posenänderung als pseudo-absolute Position (blaue Kurve) fällt das schnelle Anwachsen des Fehlers in den ersten Sekunden des Fluges, also während dem Einschwingvorgang des Kalman-Filters, auf. Dieses Anwachsen ist darauf zurückzuführen, dass Anfangsfehler in den IMU-Biasen zu einem verbleibenden

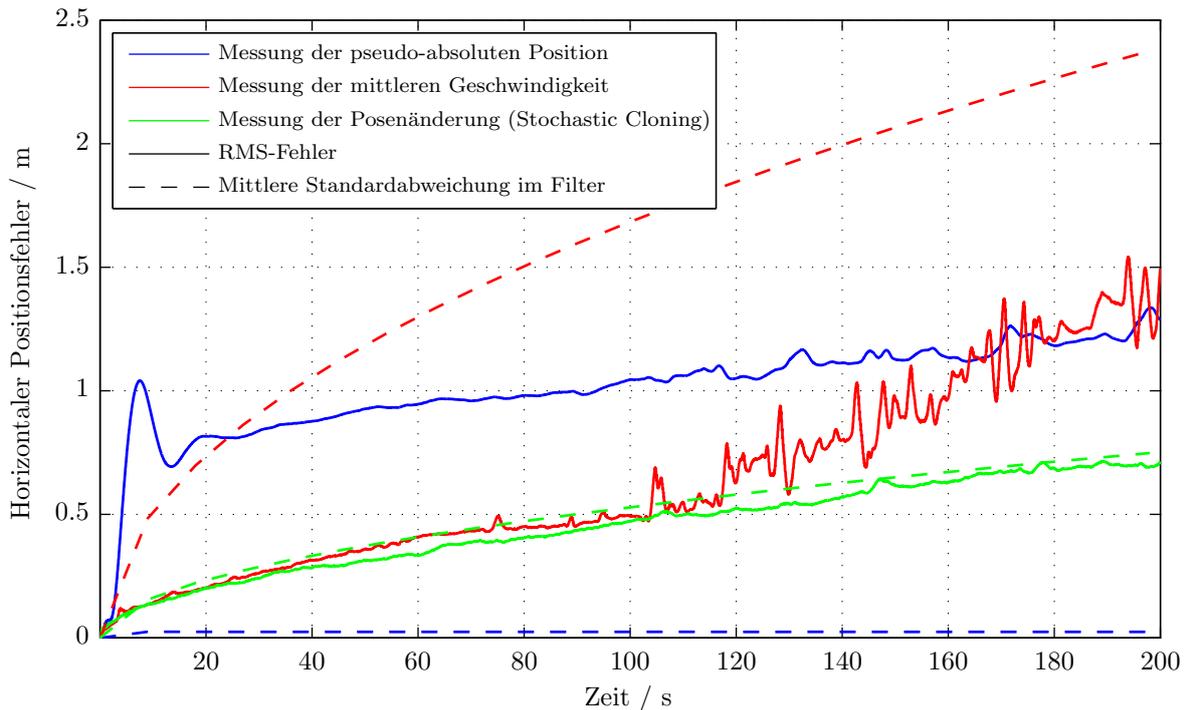


Abbildung 5.1: RMS-Fehler der horizontalen Positionsschätzung und geschätzte Standardabweichung bei ständigem Wechsel des Referenzzeitpunkts. Nur bei Einsatz von Stochastic Cloning entspricht der erwartete Fehler dem tatsächlichen Fehler, zudem ist der Fehler geringer als bei den anderen Verfahren.

Positionsfehler führen. Obwohl die Bias-Fehler beobachtbar sind, wird der durch sie anfangs verursachte Positionsfehler nicht nachträglich wieder korrigiert. Das gelingt nur bei den anderen Verfahren, ist hier dagegen nicht vollständig möglich. Die Ursache liegt darin, dass die im Filter geschätzten Kreuzkorrelationen zwischen Positionsfehler und IMU-Biasen nicht korrekt sind. Schließlich ist auch zu erkennen, dass das Filter extrem überoptimistisch bezüglich der erwarteten Positionsfehler ist: Die tatsächlichen Fehler steigen mit der Zeit immer weiter an, was durch die Kovarianz im Filter nicht widerspiegelt wird.

Wird die Posenänderung mittels Stochastic Cloning verarbeitet, ist der Fehler geringer als bei den beiden einfacheren Verfahren. Insbesondere fällt auch auf, dass hierbei die tatsächlichen Fehler gut zu den vom Filter erwarteten Fehlern passen. Bei Verarbeitung der Posenänderungen zwischen direkt aufeinanderfolgenden Zeitpunkten ist dieses Verfahren also zu bevorzugen.

Wenn nun, wie in Abschnitt 5.2.1.4 beschrieben, die aktuelle Aufnahme mit einer weiter zurückliegenden verglichen wird, können weitere Navigationsfilter realisiert und verglichen werden: Zum einen steht auch hier die Verarbeitung als pseudo-

absolute Position und mit Stochastic Cloning zur Wahl. Zudem kann mittels Stochastic Cloning, wie in Abschnitt 5.2.1.5 erläutert, auch aus den beiden zur Verfügung stehenden Posenänderungsmessungen die ausgewählt werden, die den minimalen Positionsfehler verspricht. Die Ergebnisse dieser Verfahren sind in Abbildung 5.2 aufgezeigt. Die grüne Kurve stellt hierbei als Referenz das beste Ergebnis aus Abbildung 5.1, also bei ständigem Wechsel des Referenzzeitpunkts, dar.

Es ist zu sehen, dass alle Verfahren geringere Fehler als diese Referenz aufweisen. Das Matching der aktuellen Aufnahme gegen weiter zurückliegende ist also gewinnbringend. Innerhalb der hier betrachteten Gruppe weist die Verarbeitung der pseudo-absoluten Position wieder den größten Fehler auf und zeigt auch wieder überoptimistisches Verhalten. Solange die Dynamik des Fluges gering ist (erste 100 s), besteht kaum ein Unterschied zwischen Stochastic Cloning mit Halten des Referenzzeitpunktes (lila Kurve) und der Erweiterung, bei der zwischen den beiden Posenänderungsmessungen ausgewählt wird (hellblau). Dies ist darauf zurückzuführen, dass bei geringer Dynamik das Matching auch mit zeitlich weiter entfernten Aufnahmen noch gut funktioniert und somit fast immer die Posenänderung bzgl. der älteren Referenzpose verarbeitet wird, was dem Verfahren „Stochastic Cloning, Referenzzeitpunkt halten“ entspricht. Wenn aber aufgrund höherer Dynamik die Posenänderungsmessung mit dem älteren Referenzzeitpunkt schlechter wird, kann es manchmal vorteilhaft sein, die Posenänderungsmessung bezogen auf den direkten Vorgänger-Zeitpunkt zu nutzen. Dies wird in dem vorgestellten Verfahren offensichtlich korrekt erkannt, so dass sich im zweiten Teil des Fluges der Positionsfehler und die zugehörige Positionsunsicherheit noch merklich reduzieren lassen im Vergleich zu dem Verfahren, bei dem immer der ältere Referenzzeitpunkt genutzt wird.

Abbildung 5.3 zeigt, dass die Lagefehler und die horizontalen Geschwindigkeitsfehler bei allen betrachteten Verfahren beschränkt sind. Die Lagefehler sind bei allen Verfahren ähnlich; bei den Geschwindigkeitsfehlern fällt auf, dass bei Verarbeitung der mittleren Geschwindigkeit teils deutlich erhöhte Fehler auftreten, nämlich dann, wenn hohe Beschleunigungen vorliegen.

5.2.2.5 Fazit

Die vorigen Untersuchungen zeigen, dass Posenänderungen im gegebenen Anwendungsfall mittels Stochastic Cloning verarbeitet werden sollten: Dadurch werden zum einen die Fehler gegenüber anderen Verfahren deutlich reduziert, zum anderen ist

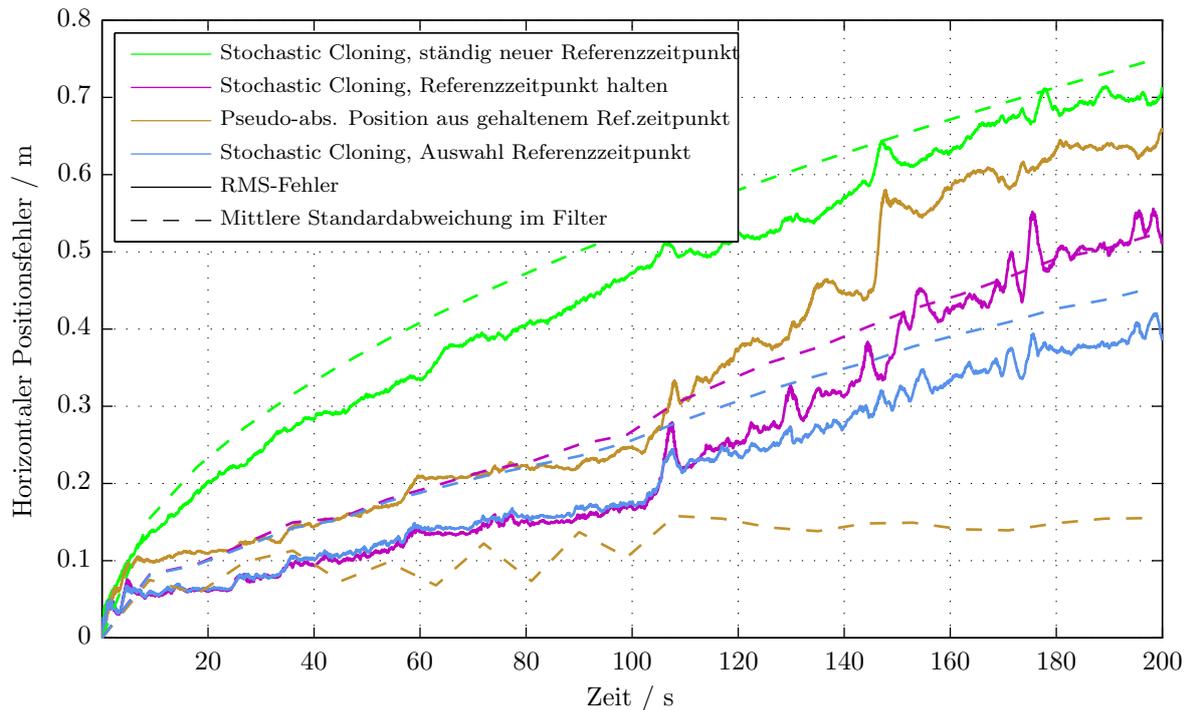


Abbildung 5.2: RMS-Fehler der horizontalen Positionsschätzung und geschätzte Standardabweichung mit Halten des Referenzzeitpunktes. Grün: Referenz ohne Halten des Referenzzeitpunktes, entspricht grüner Kurve aus Abbildung 5.1. Alle Verfahren weisen einen reduzierten Fehler auf. Die besten Ergebnisse werden mit Stochastic Cloning erzielt, wenn der Referenzzeitpunkt erst bei der Verarbeitung ausgewählt wird.

nur bei diesem Verfahren eine korrekte Kovarianzschätzung im Filter möglich. Die vorgestellte Erweiterung, dass mindestens ein weiteres Scan Matching durchgeführt wird und mittels Minimierung der a-posteriori-Kovarianz ein Matching-Ergebnis ausgewählt wird, verbessert die Positionsschätzung weiter – im hier untersuchten Szenario um ca. 20%. Dieses Verfahren ist also am geeignetsten, Posenänderungen im Kalman-Filter zu verarbeiten.

5.2.3 Verarbeitung bei verzögertem Eintreffen der Messwerte

Bisher wurde davon ausgegangen, dass die Messwerte in dem Moment vorliegen, in dem sie auch gültig sind. Dann ist eine synchrone Verarbeitung möglich. In realen Systemen kann aber eine deutliche Verzögerung zwischen der Gültigkeit und Verfügbarkeit der Posenänderungsinformation auftreten. Dabei spielen die Zeitdauer für die Messung, Verzögerungen im Treiber und der durchzuführende Matching-Vorgang eine Rolle. In der Summe können hierbei Verzögerungen auftreten,

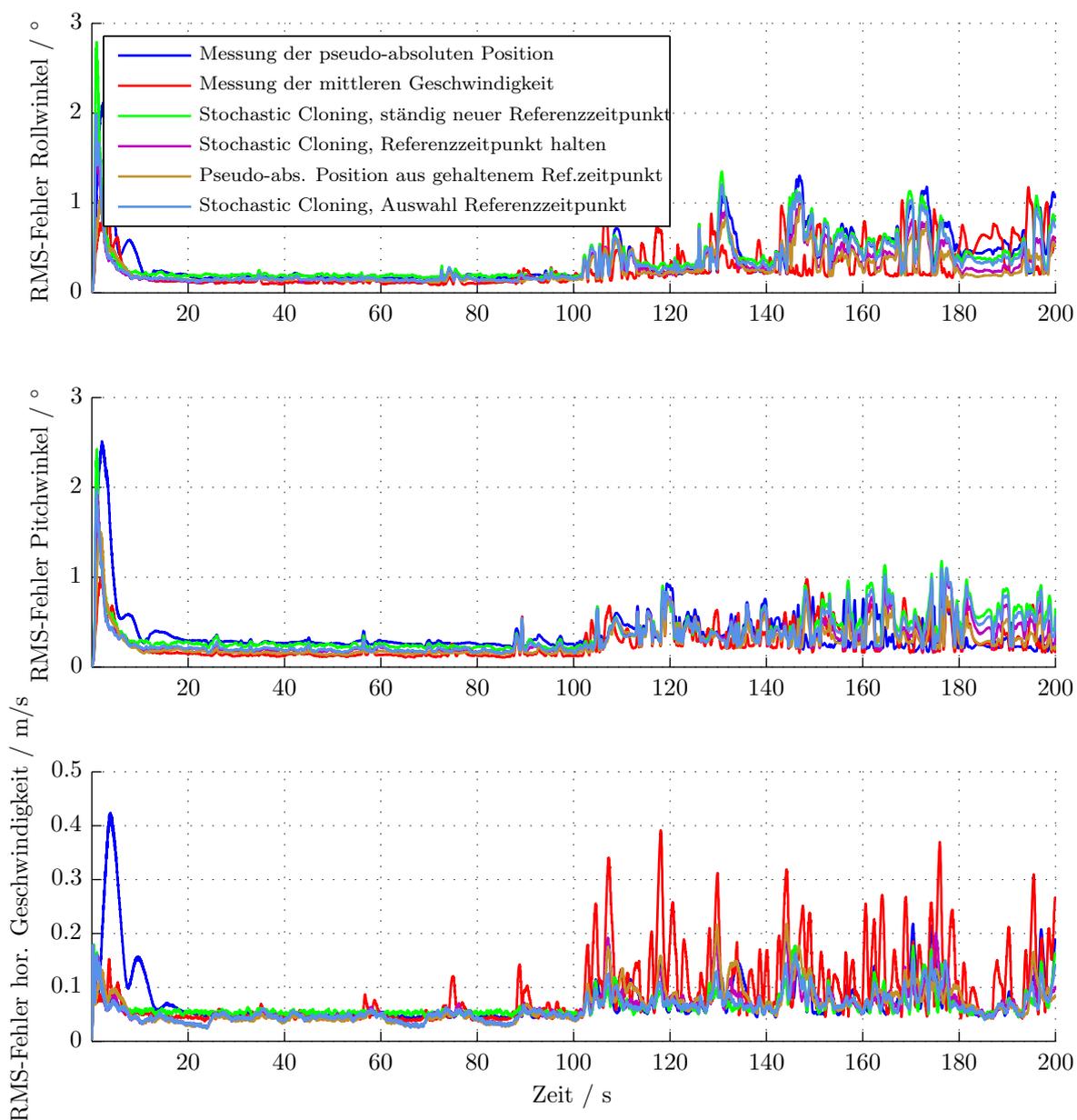


Abbildung 5.3: RMS-Fehler für Roll, Pitch und horizontale Geschwindigkeit: Die Lagefehler der unterschiedlichen Verfahren sind meist ähnlich groß. In der Flugphase ab 100 s mit höherer Dynamik sind die Fehler zwar größer als bei sehr ruhigem Flug, bleiben aber beschränkt.

die in einem Echtzeitsystem nicht tolerierbar wären. Daher muss eine asynchrone Verarbeitung im Navigationsfilter angestrebt werden.

Eine mögliche Lösung dieses Problems ist, den Filterzustand zum Gültigkeitszeitpunkt vollständig abzuspeichern. Der Filterzustand wird nun wie gewohnt weiterpropagiert und Messungen werden verarbeitet. Sobald die Posenänderungsmessung verfügbar ist, wird der gespeicherte Zustand geladen und die Posenänderungsmessung verarbeitet. Die zuvor durchgeführten Propagations- und Messschritte werden nun basierend auf dem neuen Filterzustand wiederholt. Dieser Ansatz wird z. B. in [10] genutzt. Bei dem in dieser Arbeit eingesetzten Fluggerät „AirQuad 1“ wird die Navigation allerdings innerhalb der Haupt-Regelschleife gerechnet, für deren gesamten Durchgang nur 3 ms zur Verfügung stehen. Auf der verfügbaren Rechenplattform ist dieser Ansatz daher nicht geeignet.

Stattdessen wird im Folgenden eine Erweiterung des Stochastic-Cloning-Ansatzes beschrieben und untersucht: Bei Gültigkeit der ersten Aufnahme (Zeitpunkt k_0) wird der Zustand wie zuvor geklont. Bei Gültigkeit der zweiten Aufnahme (Zeitpunkt k_1) kann noch keine Posenänderung verarbeitet werden, da diese erst später vorliegen wird. Stattdessen wird der Zustand ein weiteres Mal geklont. Solange die Posenänderung für das Intervall $[k_0, k_1]$ noch nicht verfügbar ist, kann es sogar zum Anlegen weiterer Klone kommen, falls bereits weitere Aufnahmen vorliegen (z. B. zum Zeitpunkt k_2). Wenn dann später zum Zeitpunkt k die Posenänderungsmessung für $[k_0, k_1]$ verfügbar ist, enthält der erweiterte Zustandsvektor die Werte

$$\check{\mathbf{x}}_k = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k_2} \\ \mathbf{x}_{k_1} \\ \mathbf{x}_{k_0} \end{pmatrix}, \quad k \geq k_2 > k_1 > k_0. \quad (5.17)$$

Nun kann die Messgleichung (5.16) angewendet werden, allerdings ist der Messwert hier eine Funktion von x_{k_1} und x_{k_0} , also von zwei bereits vergangenen Zuständen.

Somit ergibt sich die Gleichung

$$\begin{aligned} \tilde{\mathbf{y}}_{k_0, k_1}^{h_0} &\approx \begin{pmatrix} \hat{\mathbf{C}}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{p}_{x_{k_1}}^n - \hat{p}_{x_{k_0}}^n \\ \hat{p}_{y_{k_1}}^n - \hat{p}_{y_{k_0}}^n \\ \hat{\psi}_{k_1} - \hat{\psi}_{k_0} \end{pmatrix} \\ &+ \begin{pmatrix} \hat{\mathbf{C}}_{ne}^{h_0} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & -(\hat{p}_{y_k}^n - \hat{p}_{y_{k_0}}^n) \\ 0 & -1 & 0 & \hat{p}_{x_k}^n - \hat{p}_{x_{k_0}}^n \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} \delta \hat{p}_{x_{k_1}}^n - \delta \hat{p}_{x_{k_0}}^n \\ \delta \hat{p}_{y_{k_1}}^n - \delta \hat{p}_{y_{k_0}}^n \\ \delta \hat{\psi}_{k_1} \\ \delta \hat{\psi}_{k_0} \end{pmatrix}. \end{aligned} \quad (5.18)$$

Diese Messgleichung ist nun unabhängig vom aktuellen Zustand \mathbf{x}_k . Daher soll im Folgenden betrachtet werden, wie \mathbf{x}_k dadurch beeinflusst wird – denn nur dieser Teil des erweiterten Zustandsvektors $\check{\mathbf{x}}_k$ ist für den Regler des Fluggerätes relevant.

5.2.3.1 Theoretische Analyse

Wenn die zwei Sensoraufnahmen zu den Zeitpunkten k_0 und k_1 gemacht werden und die Messung der Posenänderung sofort zum Zeitpunkt $k_1 = k$ vorliegt – bzw. ein Warten möglich ist –, so kann die Messung synchron verarbeitet werden mit der Messmatrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix}. \quad (5.19)$$

Dies entspricht Gleichung (5.16), die ohne Berücksichtigung einer möglichen Verzögerung aufgestellt wurde. Mit dieser Messmatrix wird in diesem Fall der Zustand gemäß der Kalman-Filter-Messgleichung aktualisiert:

$$\begin{aligned} \check{\mathbf{x}}_{k_1}^+ &= \check{\mathbf{x}}_{k_1}^- - \mathbf{K} (\mathbf{H} \check{\mathbf{x}}_{k_1}^- - \tilde{\mathbf{y}}) \\ &= \begin{pmatrix} \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_0}^- \end{pmatrix} - \begin{pmatrix} \mathbf{P}_{k_1 k_1} \mathbf{H}_1 + \mathbf{P}_{k_1 k_0} \mathbf{H}_0 \\ \mathbf{P}_{k_1 k_0} \mathbf{H}_1 + \mathbf{P}_{k_0 k_0} \mathbf{H}_0 \end{pmatrix} (\mathbf{H} \check{\mathbf{P}}_{k_0}^- \mathbf{H}^T + \mathbf{R})^{-1} (\mathbf{H} \check{\mathbf{x}}_{k_1}^- - \tilde{\mathbf{y}}) \end{aligned} \quad (5.20)$$

Wenn nach dieser Messung noch Propagationen bis zum Zeitpunkt k_2 durchgeführt werden, ergibt sich bei k_2 der erweiterte Zustandsvektor

$$\check{\mathbf{x}}_{k_2} = \check{\Phi} \check{\mathbf{x}}_{k_1}^+ = \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \check{\mathbf{x}}_{k_1}^+ = \begin{pmatrix} \Phi \mathbf{x}_{k_1}^+ \\ \mathbf{x}_{k_0}^+ \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{k_2} \\ \mathbf{x}_{k_0}^+ \end{pmatrix}, \quad (5.21)$$

wobei die Matrix Φ das Produkt der Propagationsmatritzen von k_1 bis k_2 ist. Von besonderem Interesse ist dabei der Schätzwert des aktuellen Zustands:

$$\mathbf{x}_{k_2} = \Phi \left(\mathbf{x}_{k_1}^- - (\mathbf{P}_{k_1 k_1} \mathbf{H}_1 + \mathbf{P}_{k_1 k_0} \mathbf{H}_0) \left(\mathbf{H} \check{\mathbf{P}}_{k_0}^- \mathbf{H}^T + \mathbf{R} \right)^{-1} (\mathbf{H} \check{\mathbf{x}}_{k_1}^- - \tilde{\mathbf{y}}) \right) \quad (5.22)$$

Dieser Wert x_{k_2} ist optimal in dem Sinne, dass er berechnet wurde, ohne dass die Verzögerung berücksichtigt werden musste.

Im nun vorgestellten Ansatz wird die Messung nicht sofort zum Zeitpunkt k_1 , wenn die Messung gültig ist, verarbeitet. Stattdessen wird nur der Zustandsvektor erweitert zu

$$\check{\mathbf{x}}_{k_1, d}^- = \begin{pmatrix} \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_0}^- \end{pmatrix}. \quad (5.23)$$

Solange die Messung nicht verfügbar ist, wird dieser Zustandsvektor mit

$$\check{\Phi}_d = \begin{pmatrix} \Phi & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (5.24)$$

propagiert, so dass sich

$$\check{\mathbf{x}}_{k_2, d}^- = \check{\Phi}_d \check{\mathbf{x}}_{k_1, d}^- = \begin{pmatrix} \Phi \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_0}^+ \end{pmatrix} \quad (5.25)$$

und

$$\check{\mathbf{P}}_{k_2, d}^- = \begin{pmatrix} \Phi \mathbf{P}_{k_1 k_1} \Phi^T & \Phi \mathbf{P}_{k_1 k_1} & \Phi \mathbf{P}_{k_1 k_0} \\ \mathbf{P}_{k_1 k_1} \Phi^T & \mathbf{P}_{k_1 k_1} & \mathbf{P}_{k_1 k_0} \\ \mathbf{P}_{k_1 k_0} \Phi^T & \mathbf{P}_{k_1 k_0} & \mathbf{P}_{k_0 k_0} \end{pmatrix} \quad (5.26)$$

ergibt.

Wenn die verzögerte Messung nun vorliegt, kann diese mit der Messmatrix

$$\mathbf{H}_d = \begin{pmatrix} \mathbf{0} & \mathbf{H}_1 & \mathbf{H}_0 \end{pmatrix} \quad (5.27)$$

verarbeitet werden, so dass sich der aktualisierte Zustandsvektor wie folgt ergibt:

$$\begin{aligned}\check{\mathbf{x}}_{k_2,d}^+ &= \check{\mathbf{x}}_{k_2,d}^- - \mathbf{K} (\mathbf{H}_d \check{\mathbf{x}}_{k_2,d}^- - \tilde{\mathbf{y}}) \\ &= \begin{pmatrix} \mathbf{x}_{k_2}^- \\ \mathbf{x}_{k_1}^- \\ \mathbf{x}_{k_0}^- \end{pmatrix} - \begin{pmatrix} \Phi \mathbf{P}_{k_1 k_1} \mathbf{H}_1 + \Phi \mathbf{P}_{k_1 k_0} \mathbf{H}_0 \\ \mathbf{P}_{k_1 k_1} \mathbf{H}_1 + \mathbf{P}_{k_1 k_0} \mathbf{H}_0 \\ \mathbf{P}_{k_1 k_0} \mathbf{H}_1 + \mathbf{P}_{k_0 k_0} \mathbf{H}_0 \end{pmatrix} \left(\mathbf{H}_d \check{\mathbf{P}}_{k_2,d}^- \mathbf{H}_d^T + \mathbf{R} \right)^{-1} (\mathbf{H}_d \check{\mathbf{x}}_{k_2,d}^- - \tilde{\mathbf{y}})\end{aligned}\tag{5.28}$$

Vergleicht man dieses Ergebnis mit dem aus Gleichung (5.22), welches die synchrone Verarbeitung ohne Verzögerung beschreibt, wird deutlich, dass beide Ergebnisse identisch sind: Dazu sei zunächst festgestellt, dass $\mathbf{H}_d \check{\mathbf{P}}_{k_2,d}^- \mathbf{H}_d^T = \mathbf{H} \check{\mathbf{P}}_{k_0}^- \mathbf{H}^T$ und $\mathbf{H}_d \check{\mathbf{x}}_{k_2,d}^- = \mathbf{H} \check{\mathbf{x}}_{k_1}^-$. Dann folgt durch einfaches Ausmultiplizieren in Gleichung (5.22), dass die oberste Zeile in $\check{\mathbf{x}}_{k_2,d}^+$ aus Gleichung (5.28) dem vorigen Resultat entspricht.

Die obigen Betrachtungen zeigen also, dass die Schätzung des Zustands nicht verändert wird, unabhängig davon ob die Messung sofort oder mit Verzögerung verarbeitet wird. Dies gilt sogar, wenn zwischen Gültigkeit und Verfügbarkeit der Messung eine Messung eines weiteren Stützsensors verarbeitet wird. Diese Aussage kann leicht bewiesen werden, wenn man berücksichtigt, dass zwei zeitgleich gültige Messungen in beliebiger Reihenfolge verarbeitet werden können, ohne das Ergebnis im Kalman-Filter zu beeinflussen. Wenn aber der Zeitpunkt der Verarbeitung der verzögerten Messung ebenfalls irrelevant ist, kann diese also auch nach der zusätzlichen Messung verarbeitet werden.

In diesem Sinne ist Stochastic Cloning also optimal geeignet, verzögerte Messwerte zu verarbeiten. Lediglich zwei Einschränkungen sind – insbesondere bei langen Verzögerungen – zu beachten: Erstens ist natürlich die Schätzung *vor* Verarbeitung der verzögerten Messung i. d. R. schlechter, als wenn die Messung synchron verarbeitet wird. Zweitens werden genau dadurch in EKF- oder Error-State-Implementierungen zwischen Gültigkeit und Verfügbarkeit auch schlechtere Linearisierungspunkte zur Bestimmung der Systemmatrizen gewählt, da bei diesen ja um den aktuellen Schätzwert linearisiert wird. Dies kann zu einer weiteren Reduktion der Schätzgenauigkeit führen. Im folgenden Abschnitt soll untersucht werden, ab welcher Verzögerungsdauer dieser Effekt einen merklichen Einfluss hat.

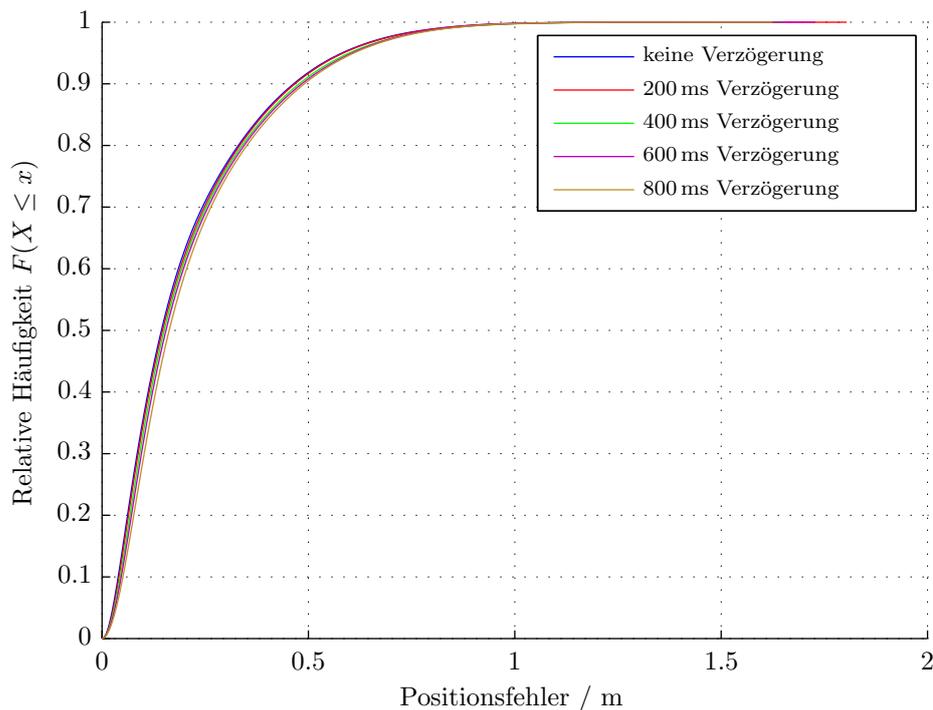


Abbildung 5.4: Einfluss von verzögerter Verarbeitung der Posenänderungsmessungen: Selbst große Verzögerungen wie z. B. 800 ms führen nur zu geringer Verschlechterung der Positionsschätzung.

5.2.3.2 Simulation: Einfluss der Verzögerung auf die Navigationslösung

In der in Abschnitt 5.2.2 genutzten Simulationsumgebung wurde das verzögerte Eintreffen der Posenänderungsmessungen simuliert. Hierzu wurden verschiedene, innerhalb eines Laufs aber konstante Verzögerungen vorgegeben. Die Messwerte waren identisch zu den unverzögerten, aber der Gültigkeitszeitpunkt wurde künstlich nach hinten verlagert. Abbildung 5.4 zeigt, dass der Einfluss der Verzögerung zwar vorhanden, aber sehr gering ist: Selbst bei 800 ms Verzögerung ist die Positionslösung kaum schlechter als ohne Verzögerung. In der tatsächlichen Realisierung kommen nur deutlich kleinere Verzögerungen vor (100 ms bis 150 ms), so dass die hier gezeigte Verarbeitung der verzögerten Messwerte problemlos genutzt werden kann. Bei den Lage- und Geschwindigkeitsfehlern sind die Unterschiede sogar noch kleiner und wären grafisch nicht wahrzunehmen, so dass hier auf die Darstellung dieser Fehlerverteilungen verzichtet wird.

5.3 Lasergestütztes Navigationssystem für ein unbemanntes Fluggerät

In diesem Kapitel wurden bisher Methoden vorgestellt, wie Relativmessungen in einem Navigationssystem verarbeitet werden können. Diese Methoden können nun dazu verwendet werden, ein konkretes Navigationssystem für die AirQuad-Fluggeräte zu entwerfen.

2D-Laserscanner zeichnen sich dadurch aus, dass sie präzise Abstandsinformationen in einem weiten Blickwinkelbereich liefern können. Dennoch ist die zu verarbeitende Datenmenge im Vergleich zu Kameras relativ gering. Sie sind auf autonomen Fluggeräten fast immer vorhanden, da sie auch für die Hindernisvermeidung eingesetzt werden. Als aktive Sensoren funktionieren Laserscanner auch im Dunkeln. Diese Eigenschaften machen Laserscanner zu geeigneten Sensoren für ein zuverlässiges Navigationssystem, das selbst auf Fluggeräten mit geringer Onboard-Rechenleistung eingesetzt werden kann.

Im Folgenden soll davon ausgegangen werden, dass ein 2D-Laserscanner so montiert ist, dass seine Scanebene durch x^b und y^b aufgespannt wird. Im Schwebeflug wird dann also in der horizontalen Ebene gemessen. Dies ist die übliche Ausrichtung, die auch für die Hindernisvermeidung sinnvoll ist. Zudem soll in einem kleinen Winkelbereich ein Spiegel die Laserstrahlen so ablenken, dass sie entlang z^b nach unten gerichtet werden; damit kann der Abstand zum Boden bestimmt werden.

Das Navigationssystem soll nicht an spezielle Laserscanner gebunden sein. Es wird aber davon ausgegangen, dass Scan-Daten mindestens mit 10 Hz vorliegen, ein Winkelbereich von 180° bei einer Winkelauflösung von $0,5^\circ$ abgedeckt wird und die Reichweite 4 m oder mehr beträgt. Diese Anforderungen werden von den Sensoren UTM-30LX und URG-04LX erfüllt, die auf den Fluggeräten der AirQuad-Plattform eingesetzt werden (s. Abschnitt 1.4). Der UTM-30LX ist aufgrund seiner größeren Reichweite und Messfrequenz dem URG-04LX prinzipiell vorzuziehen, sofern das höhere Gewicht und die höheren Kosten für die gewünschte Anwendung akzeptabel sind.

Im Folgenden wird zunächst der Scan-Matching-Prozess beschrieben, also die Gewinnung von Posenänderungen aus Laserdaten. Damit kann auf Basis der Methoden aus Abschnitt 5.2 ein Navigationssystem entworfen werden. Dieses wird

noch erweitert, um auch die Messungen der Höhe über Grund verarbeiten zu können und um das Anwachsen der Heading-Winkel-Fehler zu vermeiden.

Mittels experimenteller Ergebnisse werden die Eigenschaften des Navigationssystems aufgezeigt.

5.3.1 Scan-Matching

Aufgabe des Scan-Matching ist es, die Sensor-Posenänderung zwischen zwei verschiedenen Laseraufnahmen zu bestimmen. Diese Bestimmung kann einerseits direkt auf der vom Laser gelieferten Punktwolke durchgeführt werden, z. B. mittels dem ICP-Algorithmus (Iterative Closest Point) [12, 71]. Alternativ können aus der Punktwolke Merkmale wie Linien oder charakteristische Punkte extrahiert werden und die Merkmale aus den beiden Scans einander zugeordnet werden. Sowohl beim Matching der Punktwolken als auch der Merkmale wird ein Abstandsmaß und ein Algorithmus zur Minimierung dieses Abstandsmaßes benötigt. Für den Einsatz auf kleinen Flugplattformen ist die Wahl eines effizienten Verfahrens zwingend.

Im vorliegenden Anwendungsfall kann dabei davon ausgegangen werden, dass die gesuchte Posenänderung so klein ist, dass

1. sich die abgebildeten Raumbereiche überlappen und
2. die Größe der Posenänderung entweder geringer ist als der übliche Abstand zwischen den abgebildeten Strukturen wie z. B. Wänden, oder aber eine Initialschätzung der Posenänderung vorliegt, so dass der Restfehler entsprechend gering ist.

Diese Anforderungen sind erfüllt, weil nur zeitlich nahe beieinander liegende Scans verglichen werden sollen. Für den Einsatz zur Wiedererkennung von Orten (s. Abschnitt 5.4) kann der im Folgenden beschriebene Scan-Matching-Algorithmus dagegen nicht eingesetzt werden, da dort beide Annahmen nicht zwingend erfüllt sind.

Die AirQuad-Plattform wird insbesondere in Gebäuden oder in Gebäudenähe eingesetzt. Daher kann zudem davon ausgegangen werden, dass Wände senkrecht zueinander und senkrecht zur Bodenebene stehen (sog. Manhattan-Welt-Annahme). Diese Annahme erlaubt eine sehr recheneffiziente, robuste Umsetzung des Scan-Matching. Eine solche wurde in [113] als Teils eines SLAM-Systems vorgestellt und wird auch in dieser Arbeit genutzt. Anstatt wie in [113] die aktuelle Lasermessung

mit einem Kartenausschnitt zu vergleichen, werden aber immer zwei Lasermessungen verglichen. Das Verfahren besteht aus den im Folgenden erläuterten Teilkomponenten.

5.3.1.1 Detektion von Boden- und Decken-Messungen

Auf Grundlage der aktuellen Lagewinkelschätzung und der Messung der Höhe über Grund wird erkannt, ob ein Laserstrahl vom Boden reflektiert wurde. Wenn vom Operator Informationen bzgl. der zu erwartenden Deckenhöhe vorgegeben werden, können Reflexionen auch der Decke zugeordnet werden. Solche Messpunkte werden für das Scan-Matching verworfen, da nur vertikale Wände zur Bestimmung der 2D-Posenänderung genutzt werden können.

5.3.1.2 Liniextraktion

Wenn Lasermessungen nur noch von vertikalen Wänden in einer Manhattan-Welt stammen, ist es zweckmäßig, diese Messungen anhand der Lagewinkelschätzung in ein horizontales Koordinatensystem zu transformieren. Dann nämlich werden die Wände zu senkrecht aufeinander stehenden Linien. Um dies ausnutzen zu können, werden aus der Punktwolke Linien extrahiert. Damit wird die Komplexität und die Anzahl an Merkmalen reduziert, gleichzeitig kann hierbei Sensorrauschen unterdrückt werden. Linien, die der Orthogonalitätsannahme nicht entsprechen, werden als Ausreißer verworfen. Alle anderen Linien werden ggf. leicht gedreht, so dass nur exakt orthogonale Linien behalten werden. In diesem Schritt wird der Winkel des Laserscans bezüglich der Richtung der orthogonalen Linien bestimmt. Aus diesem kann später direkt die Winkeländerung zwischen zwei Scans berechnet werden.

Für das Scan-Matching ist es vorteilhaft, wenn Linien reproduzierbar extrahiert werden. Standardansätze wie der Iterative End Point Fit [66] weisen hierbei Probleme auf: Eine Wand wird damit aufgrund von leichten Unebenheiten oder Sensorrauschen in manchen Aufnahmen in mehrere Linien aufgeteilt, in anderen Aufnahmen aber als eine Linie erkannt. Das macht die Zuordnung schwerer. Bessere Ergebnisse können mit dem sog. Adaptive-Line-Extraction-Algorithmus [88] erzielt werden [113].

5.3.1.3 Abstandsmaß

Jeder der beiden Scans wird durch eine Liste orthogonalisierter Linien beschrieben. Um für diese ein Abstandsmaß zu definieren, wird zunächst ein Abstandsmaß zweier Linien benötigt.

Dieses besteht aus einem Anteil für den Abstand entlang der Normalenrichtung und einem Anteil entlang der Linienrichtung. Prinzipbedingt können Wandenden mit einem 2D-Laserscanner nicht genau bestimmt werden. Es ist nur ein Intervall für deren Position bekannt, dessen Ränder zudem dem Rauschen einzelner Messungen unterliegen. Dahingegen kann die Position entlang des Normalenvektors meist sehr gut bestimmt werden, weil viele Messpunkte zur Bestimmung genutzt werden und so Messrauschen unterdrückt werden kann. Diese Charakteristik wird im Abstandsmaß aus [113] berücksichtigt.

Um das Gesamtabstandsmaß zwischen den beiden Scans zu bestimmen, werden dann mittels Nearest-Neighbor-Suche passende Linien einander zugeordnet, wenn das Abstandsmaß eine Schwelle T nicht übersteigt. Das Gesamtabstandsmaß ist dann die Summe der Linien-Abstände zuzüglich $N \cdot T$, wobei N die Anzahl nicht zugeordneter Linien aus dem aktuellen Scan ist.

5.3.1.4 Minimierung des Abstandsmaßes

Das im vorigen Abschnitt skizzierte Abstandsmaß soll minimiert werden, indem der aktuelle Scan transformiert wird. Da durch die Orthogonalisierung der Linien bereits der Winkel zwischen den Scans bekannt ist, ist nur noch die optimale zweidimensionale Translation zu bestimmen.

Hierzu wird ein iteratives Verfahren verwendet: Ausgehend von einer Starttransformation werden Zuordnungen zwischen den Linien gefunden. Aus jeder Zuordnung können einer oder beide Translationsparameter bestimmt werden: Der Verschiebungsparameter in Normalenrichtung liegt immer vor; entlang der Linienrichtung beschränkt sich die Information auf ein Intervall, das bei Verdeckung der Linienenden auch unendlich groß sein kann. Wurden mehrere Linienzuordnungen gefunden, werden die bestimmten Parameter bzw. Parameterintervalle gewichtet gemittelt. Der zweite Scan wird transformiert und der Zuordnungsprozess wiederholt. Dies wird solange durchgeführt, bis das Gütemaß konvergiert oder eine maximale Zeitdauer überschritten ist.

Um zu verhindern, dass Fehler in der Starttransformation zu Falschzuordnungen führen, werden um die Starttransformation herum weitere Startpunkte ermittelt, für die ebenfalls der iterative Prozess durchgeführt wird.

Die finale gefundene Transformation zwischen den Scans ist diejenige, für die das Gütemaß am kleinsten ist.

5.3.1.5 Kovarianzschätzung

Für die Weiterverarbeitung der gefundenen Transformation im Kalman-Filter ist es wichtig, auch eine Abschätzung der Fehlerkovarianz liefern zu können. Diese wird für die Orthogonalitätsrichtungen getrennt anhand der zugeordneten Linien bestimmt. Hierbei ist zu beachten, dass die Likelihoodfunktion für die Linienassoziationen oft deutliche Nebenmaxima aufweist, wenn Mehrdeutigkeiten vorliegen. Die Kovarianzschätzung liefert aber die Fehlerkovarianz für den Fall, dass die korrekten Zuordnungen getroffen wurden. Ausreißer müssen daher in der weiteren Verarbeitung detektiert werden.

5.3.1.6 Wiederverwendung von Referenzscans

Nach dem Scan-Matching-Vorgang wird entschieden, ob der bisherige Referenzscan auch beim nächsten Scan-Matching als Referenzscan genutzt oder durch den aktuellen Scan ersetzt werden soll. Damit ein Referenzscan wiederverwendet wird, muss zum einen das aktuelle Matching ein geringes Abstandsmaß ermittelt haben. Zudem muss der Anteil der zugeordneten Linien über einem Schwellwert T_{match} liegen. Diese Prüfungen lassen erwarten, dass auch der Vergleich mit dem künftigen Laserscan erfolgreich sein wird.

Um mehrere Referenzkandidaten wie in Abschnitt 5.2.1.5 beschrieben nutzen zu können, werden bis zu drei unterschiedliche Werte für den Schwellwert T_{match} vorgegeben, so dass unterschiedlich lange gehaltene Referenzscans zur Verfügung stehen.

5.3.2 Verarbeitung der 2D-Posenänderungen

Gemäß dem Fazit in Abschnitt 5.2.2.5 werden die 2D-Posenänderungen in einem Kalman-Filter mit Stochastic Cloning verarbeitet. Da die Aufnahme von Lasermessungen je nach Sensor zwischen 20 ms und 80 ms benötigt und weitere Verzögerungen im Treiber sowie durch das asynchron durchgeführte Scan-Matching entstehen, muss auch die verzögerte Verarbeitung gemäß Abschnitt 5.2.3 implementiert werden. Eine Auswahl aus mehreren Referenzkandidaten wird ebenfalls integriert. Für die Online-Verarbeitung kann darauf jedoch nur auf der Plattform „AirQuad 2“ zurückgegriffen werden. Für „AirQuad 1“ kommt ein vereinfachtes System ohne die Erweiterung zum Einsatz, um die Rechenlast zu reduzieren.

Das bei der relativen Navigation übliche Anwachsen von Positionsfehlern mit der Zeit ist im Kalman-Filter nicht zu verhindern. Allerdings kann das Anwachsen des Heading-Winkel-Fehlers vermieden werden, wenn, wie ohnehin schon im Scan-Matching, von orthogonalen Wänden ausgegangen wird. Dann kann nämlich, sobald der Winkel der orthogonalen Wände bekannt ist, ein Zustandsklon des Heading-Winkels angelegt werden und in folgenden Zeitschritten die Winkeldifferenz zwischen diesem Heading-Referenzscan und den weiteren Scans bestimmt werden. Weil dieser Referenzscan nicht gewechselt werden muss, wächst der Heading-Winkel-Fehler somit nicht an. Das Scan-Matching vergleicht weiterhin zeitlich naheliegende Scans; es wird aber nur die daraus bestimmte Translation im Kalman-Filter verarbeitet.

Um Ausreißer in den Scan-Matching-Messungen und bei den Heading-Winkeln zu erkennen, wird die Mahalanobis-Distanz der Kalman-Filter-Innovationen betrachtet. Liegt diese über einem Schwellwert, wird die Messung verworfen. Der Schwellwert kann so eingestellt werden, dass z. B. 97% der Messwerte behalten werden, die der erwarteten Fehlerverteilung entsprechen. Ausreißer entsprechen dieser Verteilung üblicherweise nicht und werden verworfen.

5.3.3 Verarbeitung der Höhe-über-Grund-Messungen

Die alleinige Verwendung der barometrischen Messungen zur Höhenbestimmung führt oft zu unzureichenden Ergebnissen. Die Gründe liegen in der Temperaturdrift des Barometers, starkem farbigem Rauschen sowie Verfälschungen der Messungen in Bodennähe aufgrund erhöhten Druckes, der durch den Abwind der Rotoren erzeugt wird (sog. Bodeneffekt).

Die Messung der Höhe über Grund mittels abgelenkter Laserstrahlen kann diese Probleme lösen. Dazu wird – analog zur Verarbeitung von 2D-Posenänderungen – die Höhenänderung des Fluggerätes im Kalman-Filter verarbeitet. Dazu muss lediglich ein weiterer Zustandsklon, nämlich für die vorige Höhe, angelegt werden. Solange der Boden bei aufeinanderfolgenden Höhenmessungen auf der gleichen Höhe liegt, entspricht die Differenz der Höhenmessungen dann der Höhendifferenz. Eine Verletzung dieser Annahme kann analog zur Detektion von Ausreißern bei den 2D-Posenänderungen mit der Mahalanobis-Distanz detektiert werden. Damit können selbst kleine Stufen von 10 cm zu 99% detektiert werden, wenn 97% der korrekten Messungen akzeptiert werden.

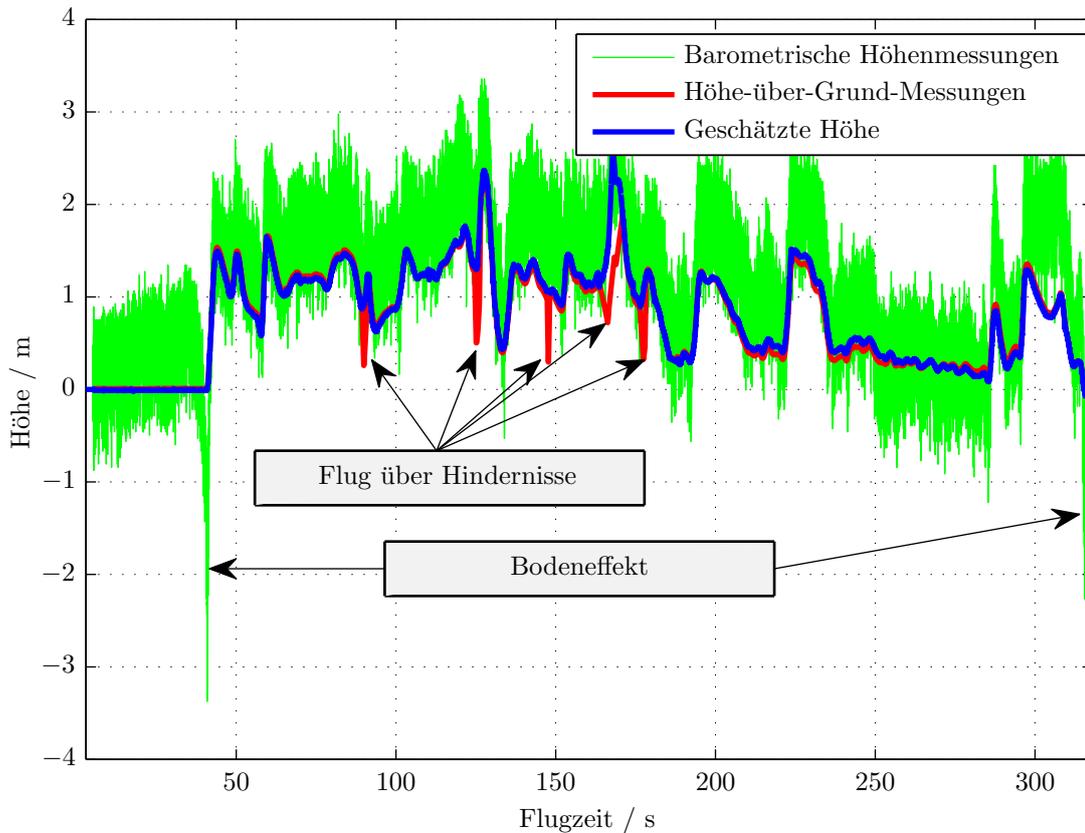


Abbildung 5.5: Höhenschätzung basierend auf barometrischen Höhenmessungen, Höhe-über-Grund-Messungen und Beschleunigungsdaten. Die jeweiligen Fehlereinflüsse werden gefiltert und die Daten zu einer präzisen Höhe fusioniert.

Ein kompletter Verzicht auf die Barometermessungen ist mit diesem System aber nicht empfehlenswert: Wenn für einige Sekunden keine Höhe über Grund gemessen wird, z. B. weil die Flughöhe höher als die Laserscanner-Reichweite ist, kann innerhalb dieses Zeitraums natürlich keine Stützung erfolgen. Auch besteht die Gefahr der Divergenz der Höhenschätzung, wenn wieder Messungen vorliegen. Daher sollten die Barometermessungen permanent verarbeitet werden. Eine hohe Messkovarianz, die auch dem farbigen Rauschen und möglichem Bodeneffekt Rechnung trägt, ist möglich. Dadurch entsteht ein robustes und genaues System zur Höhenbestimmung. Abbildung 5.5 zeigt exemplarisch das Verhalten für einen Flug mit verschiedenen Fehlereinflüssen. Es wird deutlich, dass trotz der einzelnen Sensorfehler eine zuverlässige und präzise Höhenschätzung erreicht wird.

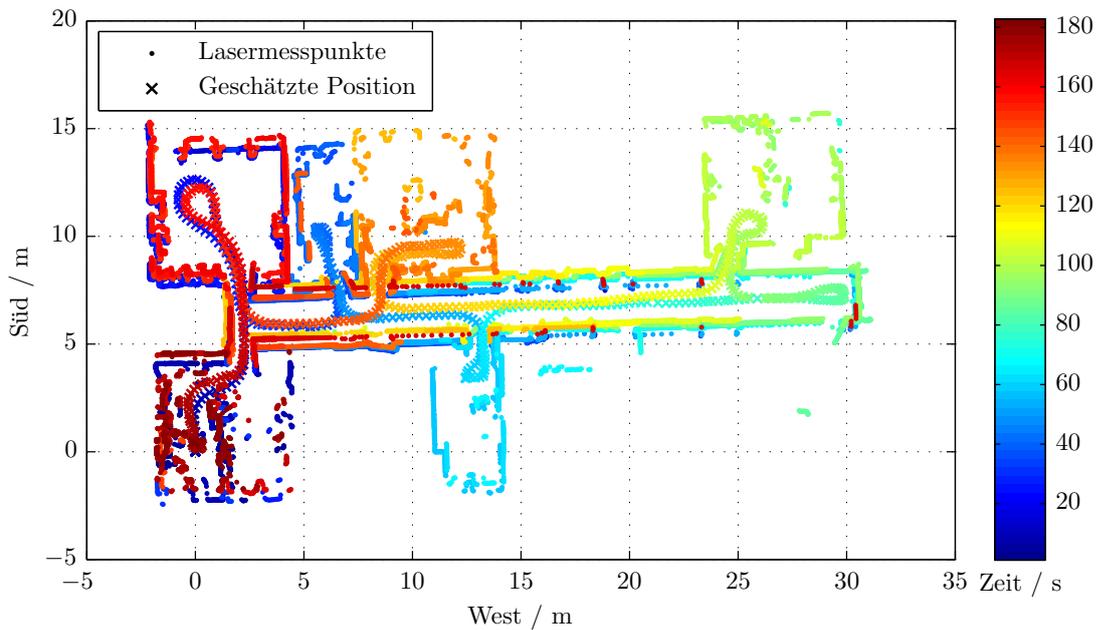


Abbildung 5.6: Ergebnis mit Realdaten auf der „AirQuad 2“-Plattform. Die Qualität der Navigationslösung ist so hoch, dass in Kombination mit den Lasermessungen eine leicht interpretierbare Karte erstellt werden kann.

5.3.4 Ergebnisse

Die Eignung des entworfenen Navigationssystems zum Einsatz auf den AirQuad-Fluggeräten wird beispielhaft anhand zweier Testdatensätze gezeigt. Der erste Datensatz wurde auf der „AirQuad 2“-Plattform aufgenommen, die mit dem Hokuyo UTM-30LX ausgerüstet ist und über vergleichsweise hohe Rechenleistung verfügt. So kann die Erweiterung „Mehrere Referenz-Kandidaten“ aus Abschnitt 5.2.1.5 aktiviert werden. Abbildung 5.6 zeigt das Ergebnis der Datenverarbeitung: Neben den Positionsschätzungen sind auch die – basierend auf der jeweiligen Pose dargestellten – Lasermessungen zu sehen. Die Trajektorie führt durch verschiedene Räume und wieder zurück. Die am Ende verzeichneten roten Lasermesspunkte liegen nur etwa 40 cm neben den entsprechenden Messpunkten vom Startzeitpunkt (blau). Da der Messfehler des Laserscanners sehr gering ist, entspricht diese Differenz dem relativen Positionsfehler der Navigationslösung. Das Navigationssystem liefert also eine präzise Positionsschätzung und kann, wie Abbildung 5.6 zeigt, sogar zur Kartierung eingesetzt werden.

Der zweite Datensatz stammt aus einem Flug mit dem Fluggerät „AirQuad 1“, das mit dem Laserscanner Hokuyo URG-04LX ausgerüstet ist. Die maximale

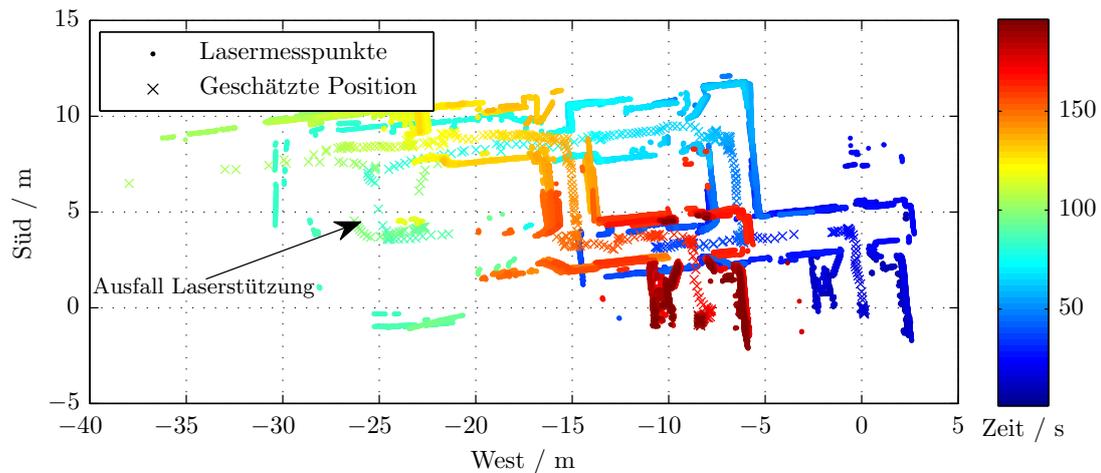


Abbildung 5.7: Ergebnis eines Fluges auf der „AirQuad 1“-Plattform mit Ausfall der Laserstützung. Nach Ende des Ausfalls verbleibt ein Positionsfehler, so dass Teile des Gebäudes versetzt nochmals kartiert werden.

Reichweite der Lasermessungen beträgt daher nur 4 m. Das Fluggerät kann zudem aufgrund der beschränkten Rechenleistung nicht auf die Erweiterung „Mehrere Referenz-Kandidaten“ zurückgreifen. Dennoch wird in der ersten Flughälfte eine gute Positionsschätzung geliefert. Dann aber kommt es zu einem kurzzeitigen Ausfall der Lasermessungen, da die Reichweite geringer ist als die Entfernung zu den nächsten Wänden; die Positionslösung driftet schnell weg. Die relative Navigation ist auf dem Rückweg zwar wieder zuverlässig, aber durch den zwischenzeitlichen Ausfall verbleibt ein deutlich erkennbarer Restfehler: Der rot dargestellte Bereich der Lasermessungen (Ende des Fluges) entspricht dem blau dargestellten Bereich, der zu Beginn kartiert wurde, ist aber ca. 8 m versetzt (Abbildung 5.7).

Gemäß Abbildung 5.8 gibt das Filter den so entstandenen Fehler aber in seiner Positionsunsicherheit wieder: Zu Beginn wächst diese nur leicht, während des Ausfalls der Laserstützung (grau hinterlegt) dann sehr schnell. Sobald wieder Lasermessungen verfügbar sind, kann ein Teil des Fehlers korrigiert werden und die Unsicherheit sinkt, aber entsprechend dem Restfehler bleibt eine Restunsicherheit.

5.4 Fehlerreduktion durch Ortswiedererkennung

Das im vorigen Abschnitt präsentierte Navigationssystem ist darauf ausgelegt, zuverlässige Geschwindigkeits- und Lageinformationen zu liefern. Auch das Anwachsen der Positionsfehler mit der Zeit soll klein gehalten werden. Dazu werden

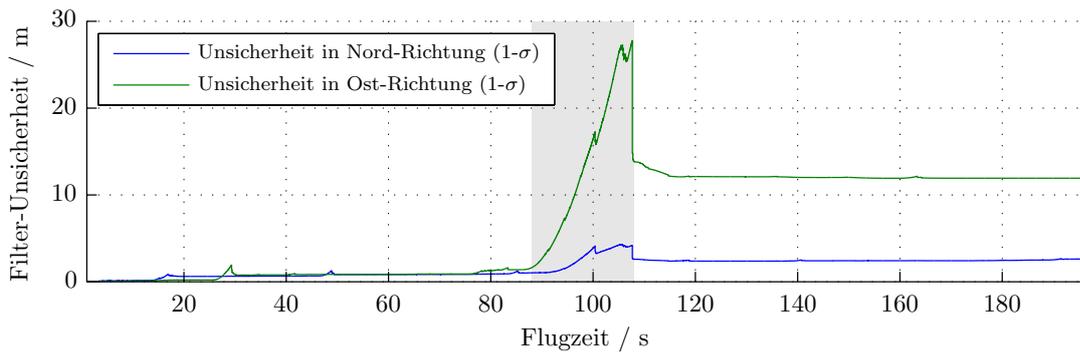


Abbildung 5.8: Verlauf der Filter-Unsicherheit bzgl. der Position aus Abbildung 5.7 über der Zeit. Der Verlauf entspricht dem Fehlerverlauf. Im grau hinterlegten Zeitraum des Laserausfalls wächst die Unsicherheit deutlich an.

die gemessenen Laserdaten so interpretiert, dass sie im Kalman-Filter möglichst gewinnbringend verarbeitet werden können. Dennoch wachsen die Positionsfehler – wie für ein Navigationssystem ohne globale Stützung üblich – an. Wenn das Fluggerät an bereits besuchte Orte zurückkehrt und dies erkennt, könnte aber der seit dem ersten Besuchen des Ortes zusätzlich entstandene Positionsfehler eliminiert werden. Dies stellt also eine vielversprechende mögliche Erweiterung dar, um das Anwachsen des Fehlers weiter zu reduzieren. Auch der in Abschnitt 5.3.4 gezeigte Effekt eines verbleibenden Restfehlers nach Stützensensorausfall könnte so vermieden werden.

Das bisher betrachtete Verfahren ist allerdings nicht geeignet, um Ortswiedererkennungen direkt zu nutzen: Mit der Stochastic-Cloning-Technik kann eine solche Information nur in Ausnahmefällen verarbeitet werden: Entweder muss im Voraus bekannt sein, welcher Ort später wiedererkannt wird, was aber im Allgemeinen nicht vorausgesetzt werden kann. Oder aber es müssten auf Verdacht regelmäßig Zustandsklone angelegt werden, die für lange Zeit im Zustandsvektor bleiben. Dies wiederum ist auf heutigen Rechenplattformen nicht für ein Echtzeitsystem geeignet. Daher wird im Folgenden eine Erweiterung des Kalman-Filter-basierten Ansatzes vorgestellt. Dabei wird das Ziel verfolgt, dass auch weiterhin mit voller IMU-Update rate eine komplette Navigationslösung geliefert werden kann. Bei Wiedererkennung bereits besuchter Orte soll aber der Positionsfehler und die damit verbundene Schätzunsicherheit sinken.

5.4.1 Ortswiedererkennung

In diesem Abschnitt wird beschrieben, wie basierend auf Laserdaten Hypothesen erzeugt werden können, ob der gerade erfasste Ort bereits besucht wurde. Dazu wird

ein Scan-Matching-Verfahren benötigt, welches zuvor gemachte Aufnahmen mit der aktuellen vergleicht.

5.4.1.1 FLIRT-basiertes Scan-Matching

Das in Abschnitt 5.3.1 vorgestellte Scan-Matching-Verfahren basiert auf der Annahme, dass sich die zu vergleichenden Aufnahmen überlappen, und nutzt wenn möglich zur Initialisierung sogar eine Schätzung der Transformation zwischen den Aufnahmen. Dann wird eine korrigierte Transformation geliefert. Zur Prüfung, ob der Ort der aktuellen Laseraufnahme bereits besucht wurde, muss jedoch zuerst die Frage geklärt werden, ob die aufgenommenen Szenen der Scans einander entsprechen. Dafür ist das Verfahren aus Abschnitt 5.3.1 nicht vorgesehen und auch nicht geeignet.

In [79] stellen Tipaldi und Arras ein Matching-Verfahren vor, das für den hier gegebenen Anwendungszweck entwickelt wurde: Mittels sogenannter FLIRT-Merkmale (engl. Fast Laser Interest Region Transform) werden Laserscans beschrieben und verglichen. Die FLIRT-Merkmale beschreiben eine lokale Umgebung innerhalb eines Scans. Sie sind daraufhin optimiert, reproduzierbar an der gleichen Stelle gefunden zu werden, wenn mehrere Aufnahmen vom gleichen Ort vorliegen: Dazu wird die Lasermessung im kartesischen Raum betrachtet und die Krümmung der Kurve bestimmt. Orte mit Maxima in der Krümmung werden als charakteristische Punkte detektiert. Diese Suche wird in verschiedenen Skalen der Kurve durchgeführt, so dass Punkte gefunden werden, die sowohl für die feine als auch für die grobe Struktur charakteristisch sein können. Zur Beschreibung der gefundenen charakteristischen Punkte werden dann sog. Beta-Grid-Deskriptoren benutzt. Diese beschreiben die Belegungswahrscheinlichkeit der Umgebung des charakteristischen Punktes. Dazu wird diese Umgebung in ein Linear-Polar-Raster aufgeteilt. Für jede Zelle wird nun die Belegungswahrscheinlichkeit geschätzt, indem betrachtet wird, ob ein Laserstrahl die Zelle durchquert hat oder ob darin ein Hindernis gefunden wurde. Mittels dieser Ereignisse können die Parameter α und β einer Beta-Verteilung geschätzt werden. Aus diesen wiederum lässt sich der Erwartungswert der Belegungswahrscheinlichkeit sowie die zugehörige Varianz berechnen.

Zu jedem Scan wird also eine Liste an charakteristischen Punkten gefunden, deren Umgebung jeweils durch einen Beta-Grid-Deskriptor beschrieben wird. Zwei Scans können verglichen werden, indem die jeweiligen Deskriptoren paarweise auf

Ähnlichkeit geprüft werden. Hierzu eignet sich die Chi-Quadrat-Distanz

$$D(x, y) = \frac{1}{2} \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}. \quad (5.29)$$

Wenn die so berechnete Distanz unter einem zuvor gewählten Schwellwert liegt, gelten die Punkte als ähnlich, es liegt eine mögliche Korrespondenz vor. Um Ausreißer in der Liste möglicher Korrespondenzen zu eliminieren, wird mittels eines RANSAC-Verfahrens (engl. random sample consensus) die maximale Menge der Korrespondenzen gesucht, die kompatibel zu einer gemeinsamen Transformationen zwischen den Scans sind.

Die Mächtigkeit dieser Menge sowie die Summe der quadratischen Abstände der korrespondierenden Punkte zueinander nach Durchführung der berechneten Transformation können genutzt werden, um zu entscheiden, ob die beiden Scans den gleichen Ort beschreiben.

5.4.1.2 Erzeugung von Schleifenschluss-Hypothesen

Mit dem FLIRT-basierten Scan-Matching-Verfahren wäre es möglich, einen neuen Laserscan mit jedem zuvor aufgenommenen Scan zu vergleichen und damit zu detektieren, ob der aktuelle Ort bereits früher besucht wurde. Allerdings ist ein Vergleich jedes Scans mit jedem vorigen Scan aus Gründen der dafür nötigen Rechenzeit nicht realisierbar. Es muss daher eine geeignete Untermenge der vorigen Laserscans ausgewählt werden, die dann in einer sog. Referenzdatenbank gespeichert wird. Die Referenzdatenbank sollte natürlich zu möglichst jedem bereits besuchten Ort mindestens einen Scan enthalten. Daher wurde die Referenzdatenbank so entworfen, dass der aktuelle Scan genau dann in die Referenzdatenbank aufgenommen wird, wenn er eine Mindestzahl an charakteristischen Punkten enthält und wenn eine Mindestzahl dieser Punkte in keinem bereits gespeicherten Referenzscan gefunden werden konnte. Dies stellt sicher, dass Scans an neu besuchten Orten aufgenommen werden, sofern sie ausreichend charakteristisch sind.

Gegen die so erstellte Referenzdatenbank kann der aktuelle Scan deutlich effizienter verglichen werden, da nur ein kleiner Teil aller Scans in die Referenzdatenbank aufgenommen werden muss.

Ein Vergleich des aktuellen Scans mit den in der Datenbank gespeicherten Scans kann mehrere Treffer gleichzeitig ergeben: Erstens, weil sich die Scans in der

Referenzdatenbank überlappen können, und zweitens aufgrund von falsch positiven Ergebnissen. Daher kann nach dem Vergleich der Scans nur von Schleifenschluss-Hypothesen gesprochen werden. Eine Überprüfung der Hypothesen erfordert weiteres Wissen, welches erst im Rahmen der Datenfusion zur Verfügung steht.

5.4.2 Navigationsstützung

Die Schleifenschluss-Hypothesen können nicht direkt im Kalman-Filter verarbeitet werden, wenn die Echtzeitfähigkeit des Systems erhalten bleiben soll. Diese Informationen würden ohnehin im Wesentlichen die Position und den Heading-Winkel korrigieren – der Einfluss auf die Geschwindigkeitsschätzung oder gar Biase ist dagegen gering, denn diese sind bereits im relativen lasergestützten Navigationssystem beobachtbar. Daher ist es ausreichend, wenn nur die horizontale Position und der Heading-Winkel, also die 2D-Pose, durch die Ortswiedererkennung verbessert wird. Dazu wird im Folgenden ein graphenbasiertes Verfahren beschrieben, das die 2D-Pose des Systems schätzt. Damit kann beim Ausgeben der Zustandsschätzung die 2D-Pose aus dem Kalman-Filter überschrieben werden, alle anderen Größen werden wie bisher dem Kalman-Filter entnommen.

5.4.2.1 Graphenstruktur

Zur Datenfusion wird ein Graph erstellt, in dem jeder Knoten einen Zeitpunkt repräsentiert. Neben der Zeit sind auch die geschätzte 2D-Pose und die zugehörige Unsicherheit bei jedem Knoten gespeichert. Neue Knoten werden angelegt, wenn neue Lasermessungen gültig sind. Die Kanten repräsentieren Informationen über die Posenänderung zwischen den Zeitpunkten, die durch die verbundenen Knoten beschrieben werden. Neben der Transformation, die die Posenänderung beschreibt, wird auch eine zugehörige Varianz der Transformationsparameter in der Kante abgelegt.

Zwei Arten von Kanten werden während der Datenprozessierung erstellt: Für die erste Art von Kanten wird das lasergestützte Kalman-Filter als sog. Basis-Navigationssystem verwendet. Es liefert Kanten zwischen aufeinanderfolgenden Knoten. In dieser Information enthalten sind sowohl die Laserstützung also auch die Messdaten der IMU und anderer möglicherweise verfügbaren Sensoren. Wegen der Nutzung von Stochastic Cloning kann dieses System die geforderte Varianz der Posenänderung liefern. Die zweite Art von Kanten basiert auf den Ortswiedererkennungen: Diese resultieren aus dem Vergleich des aktuellen Scans

mit der Referenzdatenbank. Sie verbinden üblicherweise auch zeitlich weiter auseinanderliegende Knoten.

5.4.2.2 Datenfusion im Graphen

Die Kanten des Graphen beschreiben also Posenänderungen zwischen den Knoten. Unter der Annahme, dass die Pose des Vorgängerknoten bekannt ist, kann aus dieser Pose und der durch die Kante beschriebenen Posenänderung eine Schätzung für die Pose des Folgeknotens und eine zugehörige Varianz ermittelt werden:

$$\mathbf{P}_k = \mathbf{P}(v_i) + \mathbf{C}\Delta\mathbf{P}(e_j) \quad (5.30)$$

$$\mathbf{Q}_k = \mathbf{Q}(v_i) + \mathbf{C}\Delta\mathbf{Q}(e_j)\mathbf{C}^T \quad (5.31)$$

Hierbei bezeichnet \mathbf{P}_k die aus der Kante e_j resultierende Schätzung für die Pose des Folgeknotens und \mathbf{Q}_k die Varianz dieser Schätzung. $\mathbf{P}(v_i)$ und $\mathbf{Q}(v_i)$ sind Pose und Varianz des Vorgängerknotens v_i . $\Delta\mathbf{P}(e_j)$ und $\Delta\mathbf{Q}(e_j)$ sind die Posenänderung und die Varianz der Kante e_j . Die Rotationsmatrix \mathbf{C} dreht die Posenänderung vom horizontalen Koordinatensystem des Knotens v_i in das Navigation-Frame. Für jede existierende Eingangskante kann also eine Schätzung für einen Knoten berechnet werden. Liegen mehrere solcher Schätzungen für den Knoten vor, müssen diese fusioniert werden. Allerdings sind die Fehler dieser Schätzungen möglicherweise stark korreliert, z. B. wenn zwei Kanten vom gleichen Vorgängerknoten ausgehen. Dann enthalten beide Schätzungen den Fehler in der Schätzung des Vorgängerknotens. Auch in anderen Fällen liegen solche Korrelationen vor, sind dann aber nicht mit vertretbarem Rechenaufwand zu ermitteln. Daher muss eine Datenfusion durchgeführt werden, die auch bei unbekanntem Korrelationen gültig ist. Eine geeignete Wahl ist das Covariance-Intersection-Verfahren [82]: Es liefert die Möglichkeit der optimalen Datenfusion für den Fall unbekannter Korrelation. Optimal bedeutet hier, dass die geschätzte Kovarianz immer konsistent ist [46], gleichzeitig aber die Kovarianz minimiert wird. Welches Maß für die Minimierung genutzt wird, also z. B. die Spur oder Determinante der Kovarianzmatrix, ist wählbar. Da im gegebenen Graphen die Covariance Intersection sehr häufig durchgeführt werden muss, ist ein effizientes Verfahren wichtig. Ein solches wird von Niehsen in [65] vorgestellt: Hierbei wird statt der iterativen Minimierung der Kovarianz eine nicht-iterative Berechnungsweise

für die Gewichtungskoeffizienten entwickelt, die zwar nicht mehr optimal, aber sehr effizient ist. Diese Einschränkung wird im Sinne der geforderten Echtzeitfähigkeit bei der Navigation in Kauf genommen.

Damit kann die fusionierte Schätzung $\mathbf{P}(v_k)$ aus den einzelnen Schätzungen \mathbf{P}_{k_n} wie folgt berechnet werden:

$$\mathbf{P}(v_k) = \mathbf{Q}(v_k) \sum_{n=1}^N \omega_n \mathbf{Q}_{k_n}^{-1} \mathbf{P}_{k_n} \quad (5.32)$$

$$\mathbf{Q}(v_k) = \left(\sum_{n=1}^N \omega_n \mathbf{Q}_{k_n}^{-1} \right)^{-1} \quad (5.33)$$

Die N Gewichtungskoeffizienten w_n werden nach [65] berechnet zu

$$\omega_n = \frac{1/\text{tr}(\mathbf{Q}_{k_n})}{\sum_{i=1}^N 1/\text{tr}(\mathbf{Q}_{k_n})} \quad \forall n : 1 \leq n \leq N. \quad (5.34)$$

In Gleichung (5.32) wird ein gewichteter Mittelwert aus 2D-Posen gebildet. Da der darin enthaltene Heading-Winkel eine zirkuläre Größe darstellt, muss für dieses Element eine geeignete Berechnung des gewichteten zirkulären Mittelwerts angewendet werden. Dazu wird zuerst

$$\bar{\Psi}_k = \text{atan2} \left(\frac{1}{N} \sum_{n=1}^N \sin \Psi_{k_n}, \frac{1}{N} \sum_{n=1}^N \cos \Psi_{k_n} \right) \quad (5.35)$$

berechnet. Dieser Wert wird dann vor Ausführen von Gleichung (5.32) von den Heading-Winkeln der einzelnen Posenschätzungen \mathbf{P}_{k_n} abgezogen und später wieder auf den Heading-Winkel der fusionierten Posenschätzung $\mathbf{P}(v_k)$ addiert.

5.4.2.3 Ausreißerererkennung

Bevor die Hypothesen der Ortswiedererkennung als Kanten im Graphen aufgenommen werden können, muss sichergestellt werden, dass diese korrekt sind. Sonst würde die Lösung des Navigationssystems wesentlich beeinträchtigt und könnte inkonsistent werden. Während die in Abschnitt 5.4.1.1 beschriebenen Kriterien lediglich die Ähnlichkeit der zugeordneten Laser-Aufnahmen sicherstellen, kann im Rahmen des Posengraphen zusätzlich geprüft werden, ob die aus der Ortswiedererkennungshypothese resultierende Posenschätzung plausibel ist.

Zuerst wird die resultierende Posenschätzung bestimmt. Diese ergibt sich analog zu Gleichung (5.30) zu

$$\mathbf{P}_H = \mathbf{P}(v_r) + \mathbf{C}\Delta\mathbf{P}_r, \quad (5.36)$$

wobei \mathbf{P}_H hier die resultierende Posenschätzung beschreibt; $\mathbf{P}(v_r)$ ist die Pose des wiedererkannten Ortes und $\Delta\mathbf{P}_r$ die von der Ortswiedererkennung gelieferte Transformation zwischen aktueller Pose und $\mathbf{P}(v_r)$.

Die Innovation der Hypothese der Ortswiedererkennung entspricht der Differenz aus der Posenschätzung anhand der Ortswiedererkennung und der aktuellen Posenschätzung:

$$\mathbf{z}_H = \mathbf{P}_H - \mathbf{P}(v_k) \quad (5.37)$$

Um für die Hypothese eine Plausibilitätsüberprüfung durchführen zu können, benötigt man die Kovarianzmatrix dieser Größe. Diese Kovarianzmatrix, im Folgenden relative Kovarianz \mathbf{Q}_{rel} genannt, beschreibt die Unsicherheit, die sich im Graphen zwischen dem wiedererkannten Ort $\mathbf{P}(v_r)$ und dem aktuellen Ort $\mathbf{P}(v_k)$ aufgebaut hat. Sie kann berechnet werden, indem ein Teilgraph betrachtet wird, der nur die Knoten seit v_r enthält sowie die zwischen diesen Knoten existierenden Kanten. Dann kann, beginnend von v_r gemäß Gleichung (5.31) die Kovarianz der anderen Knoten berechnet werden. Als Startknoten erhält v_r in diesem Fall die Kovarianz 0. Die Kovarianz, die dabei dem aktuellen Knoten v_k zugewiesen wird, entspricht der gesuchten relativen Kovarianz \mathbf{Q}_{rel} , mittels derer die Mahalanobis-Distanz berechnet werden kann:

$$M_{rel} = \sqrt{(\mathbf{P}_H - \mathbf{P}(v_k))^T \mathbf{Q}_{rel}^{-1} (\mathbf{P}_H - \mathbf{P}(v_k))} \quad (5.38)$$

Deren Quadrat M_{rel}^2 stellt eine Chi-Quadrat-verteilte Größe dar. Somit kann nach Festlegung eines Konfidenzniveaus ein Schwellwert M_{th} bestimmt werden, so dass Hypothesen nur akzeptiert werden, wenn $M_{rel} < M_{th}$ gilt. Die Überprüfung $M_{rel} < M_{th}$ ist dann ein wirksames Mittel zur Erkennung von Ausreißern.

Problematisch an dieser Überprüfung ist allerdings, dass die Bestimmung von \mathbf{Q}_{rel} einen recht hohen Rechenaufwand erfordert. Dieser kann aber reduziert werden. Es gilt nämlich $\mathbf{Q}(v_k) = \mathbf{Q}(v_r) + \mathbf{Q}_{rel}$, d. h. die absolute Kovarianz der aktuellen Posenschätzung, $\mathbf{Q}(v_k)$, ist immer größer oder gleich der relativen Kovarianz.

Berechnet man die Mahalanobis-Distanz anhand dieser absoluten Kovarianz

$$M_{abs} = \sqrt{(\mathbf{P}_H - \mathbf{P}(v_k))^T \mathbf{Q}(v_k)^{-1} (\mathbf{P}_H - \mathbf{P}(v_k))} \quad (5.39)$$

gilt $M_{abs} < M_{rel}$, weil die Kovarianzmatrizen $\mathbf{Q}(v_r)$ und \mathbf{Q}_{rel} positiv semidefinit sind. Mittels Überprüfung der notwendigen Bedingung $M_{abs} < M_{th}$ kann bereits eine signifikante Zahl von Ausreißern detektiert werden und somit die Berechnung von \mathbf{Q}_{rel} eingespart werden.

5.4.3 Ergebnisse

Zunächst soll noch einmal der Flug mit „AirQuad 1“ aus Abschnitt 5.3.4 betrachtet werden, bei dem es zu einem zwischenzeitlichen Ausfall der Laserstützung kam. Die Navigationslösung des lasergestützten Navigationssystems ohne Ortswiedererkennung weist für diesen Flug ab dem Ausfall einen signifikanten Positionsfehler auf. Betrachtet man nun die Positionslösung des um die Ortswiedererkennung erweiterten Systems (Abbildung 5.9), ist zu erkennen, dass die Positionslösung zwar ebenfalls abzudriften beginnt, allerdings fällt diese Drift geringer aus; bei Rückkehr in einen bereits besuchten Bereich wird der entstandene Positionsfehler dann sogar vollständig korrigiert. Dass bereits während der Driftphase ein geringeres Fehlerwachstum vorliegt, ist auf die zusätzlichen Stützinformationen aus dem FLIRT-Scan-Matching zurückzuführen.

Allerdings ist zu beachten, dass diese Ergebnisse nicht online auf der Flugplattform berechnet werden können, da zu wenig Rechenleistung bereitsteht. Dieses Verfahren eignet sich also – eine Funkverbindung vorausgesetzt – zum Einsatz auf einer Bodenstation oder auch für die Nachprozessierung.

Abbildung 5.10 stellt den Verlauf der Filterunsicherheit dar. Der Fehlerverlauf wird hierin gut wiedergegeben. Vorzuheben ist insbesondere die Reduktion der Unsicherheit zum Zeitpunkt 120 s, wo ein bereits besuchter Ort wiedererkannt wird. Die Positionsunsicherheit sinkt anschließend weiter, da weitere bereits früher besuchte Orte wiedererkannt werden.

Mit der „AirQuad 2“-Plattform wurde zudem ein Datensatz aufgenommen, bei dem mehrere große Schleifen geschlossen wurden. Damit wird demonstriert, dass auch in diesem sog. Loop-Closure-Szenario sehr gute Ergebnisse erzielt werden: Die

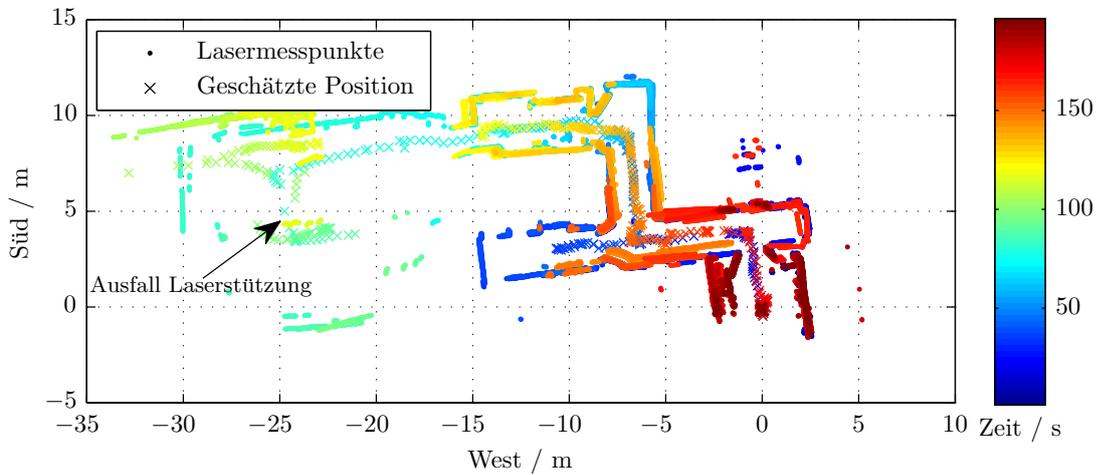


Abbildung 5.9: Ergebnis des Fluges auf der „AirQuad 1“-Plattform mit Ausfall der Laserstützung. Kurz nachdem die Laserstützung wieder vorhanden ist, wird der schon zuvor bereits besuchte Ort wiedererkannt. Dadurch kann der zwischenzeitlich entstandene Positionsfehler korrigiert werden, so dass eine dann wieder konsistente Karte des Gebäudes erstellt wird.

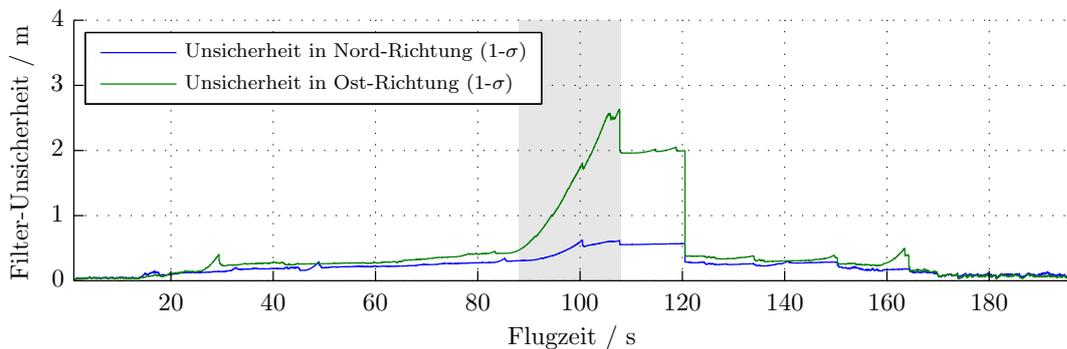


Abbildung 5.10: Verlauf der Filter-Unsicherheit bzgl. der Position aus Abbildung 5.7 über der Zeit. Der Verlauf entspricht dem Fehlerverlauf. Im grau hinterlegten Zeitraum des Laserausfalls wächst die Unsicherheit an, wird aber bei Ortswiedererkennung (120 s) deutlich reduziert.

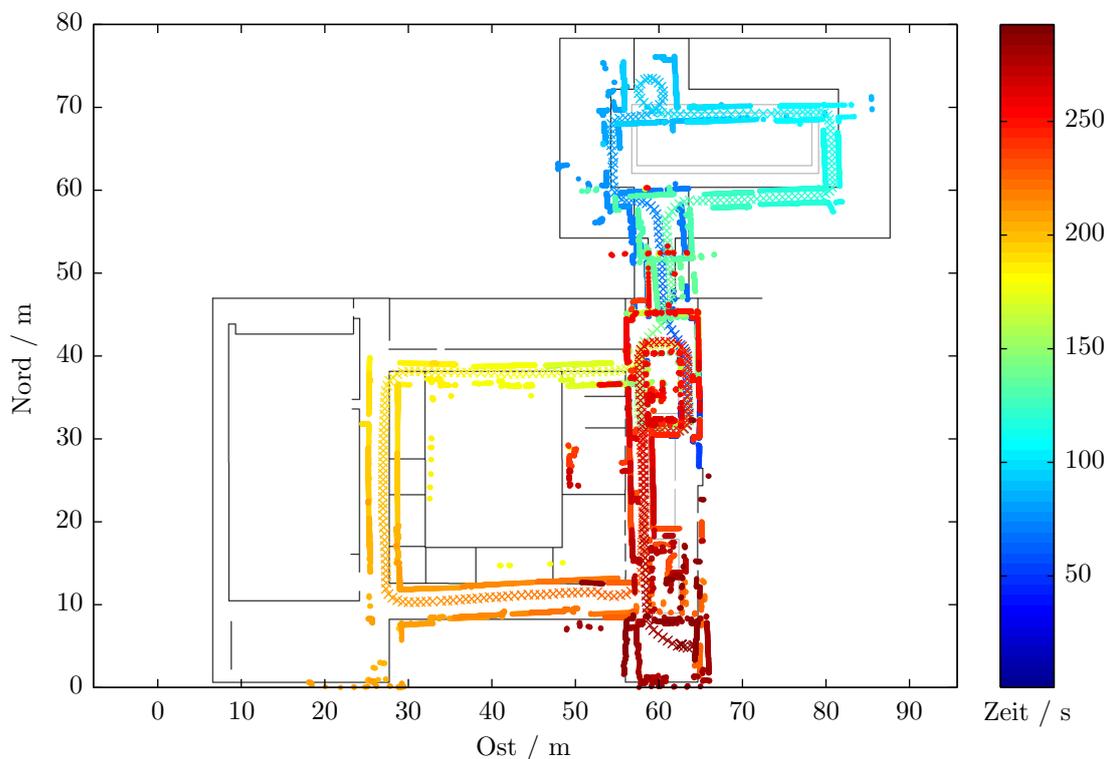


Abbildung 5.11: Ergebnis eines Datensatzes mit großen Schleifen. Bei ca. 210s wird eine große Schleife geschlossen und der bis dahin aufgebaute Positionsfehler korrigiert. Dadurch bleibt die aufgebaute Karte auch bei mehrfachem Durchqueren einiger Räume konsistent. Schwarze Linien: Referenzkarte.

Rückkehr an bereits besuchte Orte wird zuverlässig erkannt. Der seit dem ersten Erfassen entstandene Positionsfehler wird daher wieder korrigiert (Abbildung 5.11).

5.5 Kalman-Smoothing mit Relativmessungen

Kalman-Filter eignen sich sehr gut, um während der Laufzeit eines Systems Schätzungen über dessen jeweils aktuellen Zustand zu ermitteln. Für autonome Fluggeräte ist dies unverzichtbar, da eine solche Schätzung in der Flugführung und -regelung als Ist-Größe benötigt wird. Das Kalman-Filter ist das optimale Filter, wenn entsprechende Voraussetzungen erfüllt sind (s. Abschnitt 2.2.1): Zu einem Zeitpunkt t_k wird alle zwischen t_0 und t_k angefallene Information zur Zustandsschätzung $\hat{x}_{k|k}$ so fusioniert, dass die Quadrate der erwarteten Schätzfehler minimiert werden.

Trotz der Optimalität des Kalman-Filters können für gewisse Anwendungen noch bessere Zustandsschätzungen ermittelt werden. Dies ist möglich, wenn nicht nur die im Intervall von t_0 bis t_k angefallene Information berücksichtigt wird, sondern auch später verfügbar gewordene Messungen in die Schätzung einfließen. Dann wird also das Intervall t_0 bis t_N betrachtet, so dass die Zustandsschätzung $\hat{\mathbf{x}}_{k|N}$ ermittelt werden kann. Dieses Schätzproblem, bei dem das feste Intervall $[t_0, t_N]$ betrachtet wird, wird von der Klasse der sog. Fixed-Interval-Smoother-Algorithmen gelöst.

Auch wenn Smoother nur offline, also nach dem Flug ausgeführt werden können, sind solche Lösungen für die AirQuad-Plattform relevant: So können sie z. B. eine präzise Lage- und Positionsschätzung zum Zeitpunkt einer Bild- oder Laseraufnahme liefern, um nach dem Flug die Georeferenzierung zu verbessern und so eine präzisere Karte erstellen zu können. Nicht zuletzt kann eine Smoothing-Lösung als Referenz bei der Entwicklung von Online-Navigationssystemen verwendet werden.

5.5.1 Algorithmen für Smoothing über feste Intervalle

5.5.1.1 Vorwärts-Rückwärts-Smoothing

Es existieren verschiedene Herleitungen für Fixed-Interval-Smoother (vgl. [34, 77]): Eine gut nachvollziehbare Herleitung bietet das Vorwärts-Rückwärts-Smoothing [30], dessen Lösung aus dem Filterergebnis eines vorwärts und eines rückwärts laufenden Kalman-Filters gewonnen werden kann. Hierzu werden die beiden Lösungen, gewichtet über die jeweilige Kovarianz, gemittelt. Dies ist möglich, da die Lösungen als unabhängig gelten, da das vorwärts laufende Filter nur Informationen aus $[t_0, t_k]$, das rückwärts laufende Filter nur Informationen aus $(t_k, t_N]$ verwendet. Damit die Unabhängigkeit aber wirklich gilt, muss die Kovarianz des Rückwärts-Filters \mathbf{P}_{rN} mit unendlich großen Werten initialisiert werden. Dies ist aber in der Formulierung der Kalman-Filter-Gleichungen nicht möglich – zur Berechnung muss daher auf die Form des Informationsfilters ausgewichen werden, was die Implementierung erschwert.

Problematisch an diesem Ansatz bei Anwendung für die AirQuad-Plattform ist aber insbesondere die Verarbeitung von Relativmessungen im Rückwärts-Filter, da sich diese immer auf frühere Zustände beziehen, über die aber beim Rückwärts-Rechnen i. d. R. bei Verfügbarkeit der Messung noch gar kein Wissen vorliegt.

5.5.1.2 Modifizierter Bryson-Frazier-Smoother

Biermann stellte in [13] eine Smoother-Variante vor, die äquivalente Ergebnisse wie das Vorwärts-Rückwärts-Smoothing liefert, aber auf die Inversion von Zustandkovarianzmatrizen verzichtet. Auch hier werden aber im rückwärts laufenden Smoother-Durchgang die Messungen verwendet. Die Formulierung ist also für den Einsatz mit Relativmessungen ebenfalls nicht gut geeignet.

5.5.1.3 RTS-Smoother

Der von Rauch, Tung und Striebel vorgestellte Smoother-Algorithmus [67] basiert auf einem gewöhnlichen, vorwärts laufenden Filterdurchgang. Während diesem Durchgang werden neben dem Zustandsvektor und der Systemmatrix auch die a-priori- und a-posteriori-Kovarianzmatrizen abgespeichert. Dann kann nach dem Filtervorgang das Smoothing wie folgt ausgeführt werden:

Zunächst wird die Smoother-Lösung $\hat{\mathbf{x}}^s$ mit der a-posteriori-Filterlösung initialisiert:

$$\hat{\mathbf{x}}_N^s = \hat{\mathbf{x}}_N^+, \hat{\mathbf{P}}_N^s = \hat{\mathbf{P}}_N^+ \quad (5.40)$$

Dann wird iterativ für $k = N - 1 \dots 0$ die Filterlösung durch Addition eines Korrekturterms verbessert:

$$\hat{\mathbf{x}}_k^s = \hat{\mathbf{x}}_k^+ + \mathbf{C}_k (\hat{\mathbf{x}}_{k+1}^s - \hat{\mathbf{x}}_{k+1}^-) \quad (5.41)$$

$$\hat{\mathbf{P}}_k^s = \hat{\mathbf{P}}_k^+ + \mathbf{C}_k (\hat{\mathbf{P}}_{k+1}^s - \hat{\mathbf{P}}_{k+1}^-) \mathbf{C}_k^T \quad (5.42)$$

Hierbei wird die Smoother-Verstärkung

$$\mathbf{C}_k = \hat{\mathbf{P}}_k^+ \Phi_k^T (\hat{\mathbf{P}}_{k+1}^-)^{-1} \quad (5.43)$$

genutzt.

Im Gegensatz zu den beiden anderen Formulierungen werden hierbei die Messmatrizen oder Messwerte nicht benötigt, die Information über die Messungen ist in den a-posteriori-Schätzwerten $\hat{\mathbf{x}}_k^+$ und -Kovarianzen $\hat{\mathbf{P}}_k^+$ enthalten.

5.5.2 RTS-Smoothing mit Relativmessungen

Die Gleichungen des RTS-Smoother aus Abschnitt 5.5.1.3 sind zwar prinzipiell dazu geeignet, Relativmessungen zu verarbeiten. Allerdings ist eine direkte Anwendung auf das lasergestützte Navigationssystem aus Abschnitt 5.3 dennoch nicht möglich, weil beim Einsatz von Stochastic Cloning die Größe des Zustandsvektors nicht konstant ist. Die Gleichungen des RTS-Smoother gehen aber von einem gleichbleibenden Zustandsvektor aus.

5.5.2.1 Zustandsvektor konstanter Größe

Eine Möglichkeit, einen festen Zustandsvektor zu erhalten, besteht darin, dass bereits vor dem Filterdurchgang Zustände für alle künftigen Klone im Zustandsvektor reserviert werden und diese dann nach und nach gesetzt werden. Ein Löschen von Klonen erfolgt nicht. Es entsteht der Zustandsvektor $\hat{\mathbf{x}} = \left(\hat{\mathbf{x}}_b \quad \hat{\mathbf{x}}_{c_1} \quad \cdots \quad \hat{\mathbf{x}}_{c_M} \right)^T$, wobei $\hat{\mathbf{x}}_b$ die Basiszustände beschreibt, $\hat{\mathbf{x}}_{c_i}$ steht für einen Klonzustand, angelegt zum Zeitpunkt i . Auf ein Filter mit diesem Zustandsvektor kann das Smoothing gemäß den Gleichungen (5.41) bis (5.43) angewendet werden. Allerdings handelt es sich dabei eher um eine theoretische Option, da der Zustandsvektor in vielen Fällen einige tausend Zustände enthalten würde. Alleine das Abspeichern der nötigen Daten würde eine große oder gar unmögliche Herausforderung darstellen, die Verarbeitung entsprechend sehr lange dauern.

5.5.2.2 Reduzierter Smoother-Ansatz

Glücklicherweise kann eine deutlich speicher- und recheneffizientere Lösung implementiert werden. Hierzu nutzt man aus, dass keine Smoother-Lösung für die Klonzustände erforderlich ist: Die Klonzustände sind als Konstanten modelliert, d. h. $\Phi_c = \mathbf{I}$, $\mathbf{Q}_c = 0$. Es lässt sich zeigen, dass solche Zustände in der Smoother-Lösung dann auch exakt konstant sind, also $\hat{\mathbf{x}}_{c,i|N}^s = \hat{\mathbf{x}}_{c,j|N}^s \forall i, j \in [1, N]$. Dazu berechnet man die Filtergrößen für eine allgemeine Initialisierung mit \mathbf{x}_0 und \mathbf{P}_0 und nach einem allgemeinen Messupdate mit $(\tilde{\mathbf{y}}, \mathbf{R})$. Durch Berechnung eines Smoother-Schrittes mit den zuvor erhaltenen Ausdrücken folgt dann $\hat{\mathbf{x}}_{c,1|2}^s = \hat{\mathbf{x}}_{c,2|2}^s$. Zudem kann so gezeigt werden, dass $\hat{\mathbf{x}}_{b,p|N}^s = \hat{\mathbf{x}}_{c_p,i|N}^s = \hat{\mathbf{x}}_{c_p,N|N}^+$. Die Smoother-Lösung für die Basiszustände zum Zeitschritt p entsprechen also gerade dem Schätzwert des Filters für die zugehörigen Klonzustände am Ende des Filter-Durchgangs.

Die Klonzustände haben über die Systemmatrix keinen Einfluss auf die eigentlich gesuchten Basiszustände, da die Systemmatrix beim Klonvorgang gemäß Abschnitt 5.2.1.3 zu

$$\check{\Phi} = \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (5.44)$$

erweitert wird. Die rechte obere Blockmatrix, die einen Einfluss darstellen könnte, ist eine Null-Matrix. Daher kann ein effizienter Smoother-Algorithmus implementiert werden, indem bereits beim Speichern der Filtergrößen nur die linken oberen Submatrizen der Größe des Basiszustandes gespeichert werden, d. h. die Zeilen und Spalten bzgl. der Klonzustände werden gestrichen. Dies reduziert den Speicher- und Rechenaufwand erheblich und lässt eine problemlose Verarbeitung zu. Dass diese Modifikation eine gültige Näherung darstellt, wird im Folgenden an einem rechnerischen Beispiel verdeutlicht.

5.5.2.3 Vergleich Reduzierter Smoother mit vollem Smoother

Anhand eines vereinfachten Systems soll gezeigt werden, dass das oben beschriebene Verfahren mit reduzierten Matrizen für den Smoother-Durchgang nahezu identische Ergebnisse liefert wie der im allgemeinen nicht realisierbare vollständige Smoother. Dazu wird eine eindimensionale Bewegung betrachtet, als Basiszustände werden Position und Geschwindigkeit geschätzt, es gilt also: $\hat{\mathbf{x}}_b = \begin{pmatrix} p & v \end{pmatrix}^T$. In jedem vierten Zeitschritt, außer zwischen den Zeitschritten 50 und 100, sei eine Abstandsmessung gültig und verfügbar. Diese entspricht einer Lasermessung im realen System. Zu diesen Zeitpunkten wird ein Zustandsklon der Position angelegt und – sofern ein Klon für den Referenzzeitpunkt vorhanden ist – die Messung als Relativmessung verarbeitet. Für die kurze Simulationsdauer von 150 Zeitschritten kann der vollständige Smoother nach Abschnitt 5.5.2.1 implementiert und berechnet werden. Der reduzierte Smoother (Abschnitt 5.5.2.2) wird dann damit verglichen. Es zeigt sich, dass die Schätz- und Kovarianzverläufe (s. Abbildung 5.12) nahezu identische Werte liefern. Bei den Kovarianzen beträgt die maximale Differenz etwa $3,5 \cdot 10^{-4} \text{ m}^2$, $9,8 \cdot 10^{-6} \text{ m}^2/\text{s}^2$ und bei den Schätzgrößen $1,2 \cdot 10^{-4} \text{ m}$ bzw. $3,6 \cdot 10^{-4} \text{ m/s}$. Die Abweichungen sind also vernachlässigbar klein.

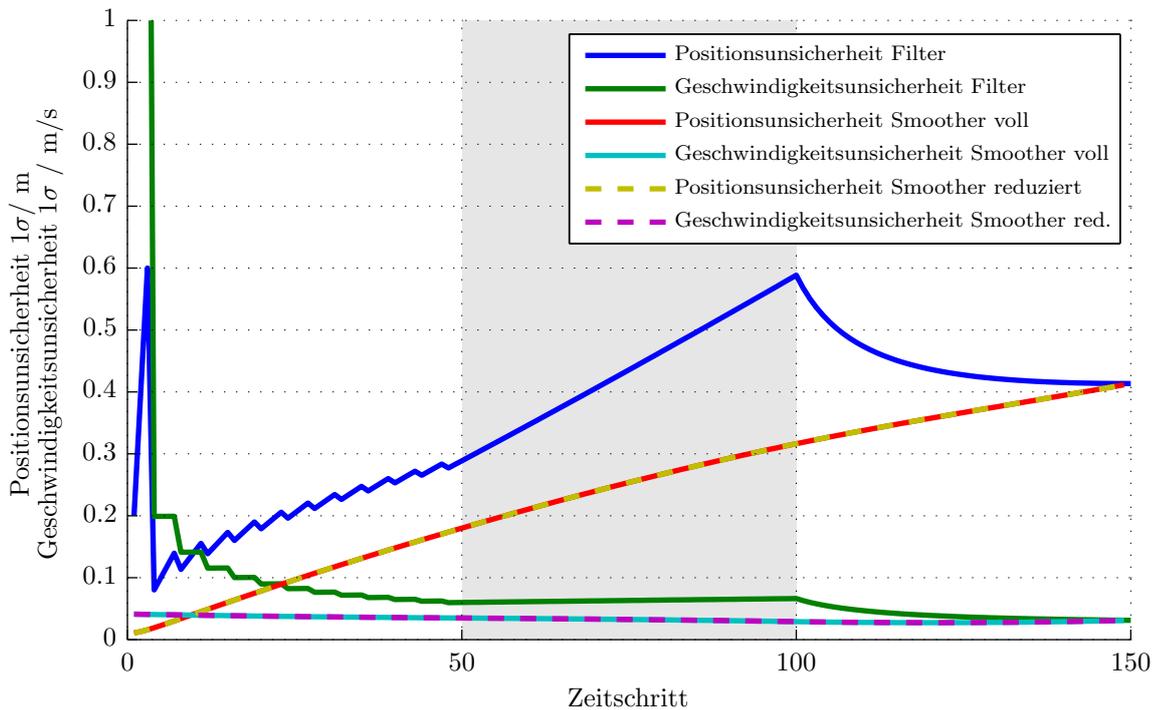


Abbildung 5.12: Verläufe der Positions- und Geschwindigkeitsunsicherheit für ein Kalman-Filter mit Stochastic Cloning und zwei RTS-Smoother – einmal ein vollständiger Smoother mit sämtlichen Klonzuständen, einmal der reduzierte Smoother mit nur zwei Zuständen (für Position und Geschwindigkeit). Die Reduktion hat keinen merklichen Einfluss.

5.5.3 Berücksichtigung von Ortswiedererkennungen mit PRC-Smoothing

Smoothing ist besonders wirksam, wenn zu einem späteren Zeitpunkt die Filterunsicherheit deutlich gesenkt werden kann. Dies ist bei GPS/INS-Systemen z. B. nach einem zeitweisen Ausfall der GPS-Stützung der Fall. Auch bei Ausfall von Laserstützung ist der Effekt vorhanden. Da es sich aber nur um Relativmessungen handelt, ist er dabei nicht so stark ausgeprägt. Der Gewinn von Kalman-Smoothing ist somit bei einem lasergestützten Navigationssystem eher gering. Dies ändert sich aber, wenn auch Ortswiedererkennungen, wie sie im Unterkapitel 5.4 beschrieben sind, berücksichtigt werden.

Im Unterkapitel 5.4 wurde argumentiert, dass Ortswiedererkennungen nicht sinnvoll im Kalman-Filter verarbeitet werden können, weil im Voraus nicht bekannt ist, auf welchen Ort sich diese beziehen werden. Diese Einschränkung liegt für die Offline-Prozessierung aber nicht mehr vor: Hier kann in einem ersten Verarbeitungsschritt das System nach Unterkapitel 5.4 den Datensatz vollständig prozessieren und so

alle Ortswiedererkennungen ermitteln. Im nächsten Schritt wird dann ein Kalman-Filter eingesetzt, das nicht nur die inkrementellen Relativmessungen verarbeitet, sondern zusätzlich auch die Ortswiedererkennungen einbezieht. Da diese nun von Beginn an bekannt sind, können zu den entsprechenden Bezugszeitpunkten Zustandsklone erstellt werden. Dann kann die Messwertverarbeitung identisch wie für die sonstigen Relativmessungen erfolgen. Werden die in diesem Filterdurchgang anfallenden Größen gespeichert, kann mit dem in Abschnitt 5.5.2.2 beschriebenen Verfahren eine Smoother-Lösung bestimmt werden. Dieses Verfahren wird als „PRC-Smoothing“ bezeichnet, da auch die Korrekturen durch die Ortswiedererkennung („Place Recognition Corrections“) einfließen.

Abhängig von der Anzahl der Ortswiedererkennungen und dem Zeitraum zwischen Referenzzeitpunkt und Wiedererkennungszeitpunkt muss das vorwärts rechnende Filter teilweise mit einem großen Zustandsvektor umgehen, so dass die Rechenzeit im Vergleich zum System nach Unterkapitel 5.4 deutlich steigt. Da es sich aber ohnehin um ein Offline-Verfahren handelt, wird dies in Kauf genommen. Die Implementierung einer intelligenten Auswahl der Ortswiedererkennungen stellt einen möglichen Optimierungsansatz dar, dessen Verfolgung aber hier nicht notwendig ist.

5.5.4 Ergebnisse

Wendet man das vorgestellte Verfahren auf die Daten des in Abbildung 5.9 gezeigten Fluges an, so erhält man weitere Verbesserungen: Dies ist anhand der noch dichter beieinander liegenden Lasermessungen, die die Wände beschreiben, zu erkennen (Abbildung 5.13). Insbesondere ist nun auch im Zeitraum von ca. 90 s bis 115 s, in dem ein größerer Sprung im Online-System in Abbildung 5.9 zu erkennen ist, eine glatte Trajektorie und korrekte Karte entstanden.

Das Verfahren eignet sich also gut zur Gewinnung einer Referenz-Navigationslösung, insbesondere da die Positionslösung durch die Berücksichtigung der Ortswiedererkennungen langzeitgenau ist.

5.6 Fazit

Die Untersuchungen zeigen, dass die in diesem Kapitel entwickelten Verfahren eine präzise Navigationsschätzung ermöglichen, wenn Lasermessungen vorliegen. Selbst die grundlegende Variante für den Onboard-Einsatz bei geringer Rechenleistung bietet in den meisten Fällen eine zuverlässige Positionsinformation. Der Ausfall der

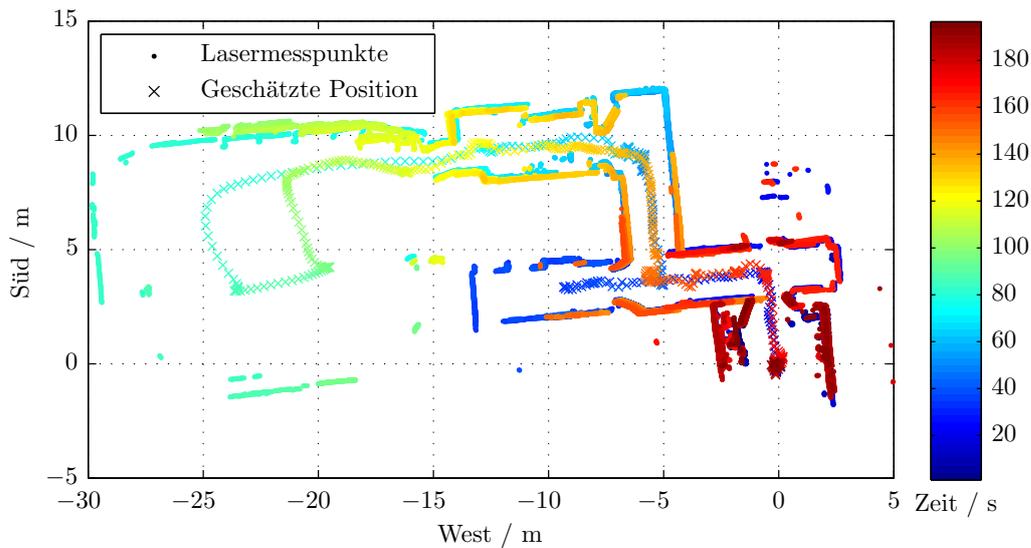


Abbildung 5.13: Flug mit realem Ausfall der Laserstützung. Durch das PRC-Smoothing verschwindet dessen Auswirkung praktisch vollständig: Es wird eine durchgehend präzise Positionsschätzung und eine vollständig konsistente Karte geliefert.

Laserstützung führt hierbei allerdings zu einem raschen Anwachsen des Positionsfehlers. Dieser kann aber nachträglich mittels der vorgestellten Ortswiedererkennung korrigiert werden, wenn an einen zuvor besuchten Ort zurückgekehrt wird. Die Umsetzung eines RTS-Smoothings für Relativmessungen und Ortswiedererkennungen erlaubt zudem eine weiter verbesserte Offlineschätzung der Navigationsgrößen.

Um die Drift der Schätzgrößen auch während des Fluges weiter zu beschränken, können dem System zusätzliche relative Stützmessungen hinzugeführt werden, z. B. aus Kameras, wie bereits in [115] demonstriert. Eine weniger kosten- und rechenintensive Ergänzung stellt die im folgenden Kapitel entwickelte modellgestützte Navigation dar.

Kapitel 6

Modellgestützte Navigation

Das in Kapitel 5 vorgestellte lasergestützte Navigationssystem kann in vielen Situationen eine präzise Navigationslösung liefern. Allerdings wird dazu ein Laserscanner benötigt, der hohe Kosten mit sich bringt. Zudem gibt es Situationen, in denen prinzipbedingt keine Laserstützung erfolgen kann, z. B. wenn keine Objekte im Erfassungsbereich des Scanners liegen oder starkes Sonnenlicht in den Sensor fällt. Der Einsatz von Kameras zusätzlich zum Laserscanner kann solche Probleme zwar reduzieren [115], löst sie aber nicht, da auch Kameras auf Umgebungsbedingungen, insbesondere geeignete Beleuchtung, angewiesen sind. Außerdem erfordert die Onboard-Verarbeitung das Mitführen einer leistungsstarken Rechenplattform, was auf kleineren Fluggeräten aus Gewichtsgründen nicht immer möglich ist.

Modellgestützte Navigation zielt darauf ab, mit zusätzlichem Wissen anstatt zusätzlicher Sensorik die Navigationslösung zu verbessern. Sie vermeidet also die zusätzlichen Belastungen bei Kosten und Gewicht. Dazu wird ein Bewegungsmodell des Fahrzeugs oder Fluggerätes aufgestellt, das dann auf geeignete Weise in das Navigationssystem integriert werden muss. Diese Integration des Bewegungsmodells in ein Navigationssystem stellt aber keine triviale Aufgabe dar. In Abschnitt 6.1 werden verschiedene bestehende Ansätze vorgestellt und deren Stärken und Schwächen beleuchtet. Anschließend wird in Abschnitt 6.2 ein neues Verfahren hergeleitet, das auch über das Gebiet der Navigation hinaus einsetzbar ist: Es erlaubt generell, Kalman-Filter einzusetzen, wenn zwei gleichermaßen gültige Prädiktionsmodelle vorliegen. Da dies bei modellgestützter Navigation der Fall ist, kann das Verfahren auf die Navigation von Quadrokoptern angewendet werden und somit auch die Eignung für eine konkrete Anwendung nachgewiesen werden (Abschnitt 6.3).

In den Abschnitten 6.4 und 6.5 wird das Navigationssystem erweitert, so dass während des Fluges Modellparameter und Wind geschätzt werden können. Dadurch kann das Navigationssystem in der Praxis einfach und zuverlässig eingesetzt werden.

6.1 Verwandte Arbeiten

6.1.1 Modellgestützte Navigation und Multiple-Model-Ansätze

Modellgestützte Navigation liegt vor, wenn neben der klassischen Kalman-Filter-Propagation mittels Strapdown-Algorithmus und inertialen Messungen auch ein Bewegungsmodell eingesetzt wird. Diese beiden Systemmodelle könnten die Vermutung nahelegen, dass Methoden wie Interacting Multiple Models (IMM, [14]) oder Multiple Model Adaptive Estimation (MMAE, [18]) geeignet wären, beide Modelle in einem System zu integrieren. Allerdings ist dies nicht der Fall: Bei IMM und MMAE wird davon ausgegangen, dass die Modelle unterschiedlich sind und zu einem gegebenen Zeitpunkt nur eines der Modelle gültig ist. Dass ein bestimmtes Modell gültig ist, stellt eine Hypothese dar. Die Wahrscheinlichkeit dieser Hypothesen kann geschätzt werden, wenn die Modelle unterschiedliche Charakteristiken aufweisen. Ein Mischen der Modelle ist in diesem Kontext zwar sinnvoll, um den Einfluss falscher Hypothesen zu minimieren. Die Lösung ist jedoch schlechter, als wenn nur das wahre Modell im Filter enthalten wäre.

Im Gegensatz hierzu wird bei modellgestützter Navigation davon ausgegangen, dass sowohl die inertielle Navigation als auch das Bewegungsmodell gültige Modelle sind: Sie beschreiben dasselbe System, aber in unterschiedlicher Form, so dass sie auch unterschiedliches Fehlerverhalten zeigen. Ein modellgestütztes Navigationssystem sollte die beiden Modelle also eng verknüpfen, um die jeweils enthaltene Information optimal zu nutzen.

Dies schließt eine Kombination aus modellgestützter Navigation und Multiple-Model-Ansatz aber nicht aus: In [19] wird eine Filterbank genutzt, um Hypothesen bzgl. Fehlern in GPS-Satelliten zu testen, wobei jedes dieser Filter ein modellgestütztes Filter ist, das inertielle Navigation mit einem Flugzeugmodell kombiniert.

6.1.2 Modellgestützte Navigation für Bodenfahrzeuge

Das Bewegungsmodell von Bodenfahrzeugen unterscheidet sich deutlich von dem von Flugzeugen oder Unterwasserfahrzeugen: Während die Bewegung von letzteren

gut durch Kräfte und Momente abgebildet werden kann, stehen bei Bodenfahrzeugen Beschränkungen im Vordergrund, da in gewöhnlichen Fahrsituationen keine Seitwärts-Geschwindigkeit vorliegt und das Fahrzeug sich nur auf der Bodenebene bewegt. Solche Zwangsbedingungen werden z. B. in [26, 50, 59] eingesetzt. Zudem können fahrzeugspezifische Informationen wie Lenkwinkel oder Raddrehzahlsensoren genutzt werden. Teilweise steht aus Kostengründen nur eine reduzierte IMU zur Verfügung, da in Pkws teilweise keine 3-Achs-Gyroskope vorhanden sind. Dies erfordert natürlich angepasste Datenfusionskonzepte, die sich von denen anderer Fahrzeugklassen unterscheiden, die hier aber nicht weiter betrachtet werden sollen.

6.1.3 Generelle Ansätze zur modellgestützten Navigation

Für den Einsatz auf Fluggeräten sind insbesondere solche Ansätze interessant, die ein allgemeines Kräfte- und Momentenmodell mit dem inertialen Navigationssystem fusionieren können. Hierzu existieren in der Literatur verschiedene Ansätze, die im Folgenden skizziert werden sollen.

Ein früherer Ansatz stammt von Koifman und Bar-Itzhack [53, 54], die mittels einer Pseudomessung im Kalman-Filter die Fusion der Informationen aus Bewegungsmodell und INS realisieren. Dazu sind die Navigationsgrößen Position, Geschwindigkeit und Lage doppelt im Zustandsvektor enthalten. Jeweils eine Kopie dieser Größen wird mit dem INS bzw. dem Bewegungsmodell in der Zeit propagiert. Die so entstehenden Schätzungen für die Navigationsgrößen driften mit der Zeit auseinander; sie können aber im Messschritt des Kalman-Filters auf einen identischen, fusionierten Wert gebracht werden: Hierzu wird in einer virtuellen Messung die Differenz der beiden Größen bestimmt. Diese ist im realen System immer gleich null, da der Zustand real ja nur einmal vorliegt. Die Stützinformation ist immer verfügbar und erfordert keinen Sensor, so dass von einer Pseudomessung gesprochen werden kann. Selbst ohne weiteren Stützsensoren kann dieses Navigationssystem Fehler sowohl im INS als auch im Bewegungsmodell schätzen. Dadurch werden die Fehler der Navigationslösung wesentlich reduziert im Vergleich zur Propagation mittels der einzelnen Modelle. Der Ansatz wird in [47] auch von Julier und Durrant-Whyte untersucht, wobei festgestellt wird, dass es sich um eine optimale Fusion der Teilsysteme handelt. Ein bedeutender Nachteil des Ansatzes besteht allerdings im deutlich vergrößerten Zustandsvektor, der sowohl die Rechenzeit als auch die Komplexität des Systems erhöht.

Daraufhin wurden verschiedene Arbeiten veröffentlicht, die die Fusion der Modelle mit einem reduzierten Zustandsvektor umsetzen:

Hegrenæs [40, 41] nutzt das Modell eines Unterwasserfahrzeuges, um basierend auf Stellgrößen und einer Strömungsschätzung eine modellbasierte Geschwindigkeit zu berechnen, mit der das INS gestützt wird. Dies ist möglich, da der Geschwindigkeitsfehler durch die hydrodynamische Dämpfung begrenzt ist. Der Fehler dieser Stützinformation wird als farbiges Rauschen modelliert. Dieser Ansatz bringt nur einen geringen Rechenaufwand mit sich, verzichtet aber auf die mögliche Stützung des Bewegungsmodells durch das INS und ist somit suboptimal.

In der Arbeit von Sendobry [74, 75] wird nicht das INS durch das Bewegungsmodell gestützt, sondern das Bewegungsmodell zum grundlegenden Modell erklärt und durch die Messungen der IMU gestützt. Dies entspricht der klassischen systemtheoretischen Betrachtung, da die Modellgleichungen zur Propagation und die Messungen zur Stützung genutzt werden. Dies ist insbesondere bei Ausfall der IMU vorteilhaft, weil dann das Navigationssystem direkt weiterlaufen kann. Um allerdings die Messwerte der Beschleunigungsmesser verarbeiten zu können, müssen entweder die geschätzten Beschleunigungen in den Zustand aufgenommen werden [74] oder – im Falle eines Quadropters – der Zustandsvektor um die Motordrehzahlen erweitert werden [75]. Dann kann die erwartete Beschleunigung als Funktion des Zustandsvektors berechnet werden. In beiden Fällen ist also eine Erweiterung des Zustandsvektors notwendig. Zudem muss eine vollständige Modellierung durchgeführt und in das Filter integriert werden: Wenn z. B. Motordrehzahlen geschätzt werden, muss auch deren Zeitverhalten in Form von Differentialgleichungen formuliert werden.

Bryson und Sukkarieh [21] schlagen vor, die von der IMU gemessenen Beschleunigungen und Drehraten direkt mit den vom Modell gelieferten Werten zu vergleichen, um so die Nullpunktsfehler beider Systeme in einem vorgeschalteten Filter zu schätzen. Der in [21] durchgeführte Vergleich zum Ansatz von Koifman und Bar-Itzhack zeigt aber, dass dieses vereinfachte System deutlich schlechter abschneidet als das von Koifman und Bar-Itzhack. Dies ist insofern leicht nachvollziehbar, da bei Bryson und Sukkarieh keine Stützung der Geschwindigkeit oder Lage stattfindet und die Fehler dieser Größen also unbeschränkt anwachsen.

Vasconcelos [83] beschreibt ein Navigationssystem, dessen Ziel die Reduktion des Zustandsvektors im Vergleich zu Koifman und Bar-Itzhack ist, wobei aber die

Schätzgenauigkeit beibehalten werden soll. Dies gelingt, indem die Modellgleichungen in das Filter integriert werden: Die rotatorische Dynamik wird genutzt, um die Drehrate im Zustand zu propagieren, die dann durch die Drehratensensoren gestützt wird; die translatorische Dynamik dagegen wird in einer Messgleichung verwendet, bei der die vom Modell gelieferte mit der von der IMU gemessenen Beschleunigung verglichen wird. Dadurch werden die Messungen der IMU zweifach verwendet: Zur Zustandspropagation wie in einem INS üblich und zur Fusion mit dem Modell. Dabei entstehen Korrelationen zwischen dem Zustand und den modellbasierten Stützmessungen. Um diese zu berücksichtigen, müssen die Kalman-Filter-Gleichungen gemäß [18] modifiziert werden. Die Komplexität der Filterstruktur wird noch weiter gesteigert, da der Zustandsvektor sowohl absolute Größen als auch Fehlergrößen beinhaltet.

6.1.4 Modellgestützte Navigation für Helikopter

Neben den im vorherigen Abschnitt beschriebenen allgemeinen Konzepten gibt es in der Literatur auch Vorschläge, die speziell auf Quadrocopter oder zumindest Drehflügler zugeschnitten sind:

Dadkhah nutzt das rotatorische Modell eines Helikopters zur Bestimmung der Lage [25]. Mit dieser Lage wird die von den Gyroskopen propagierte Lageschätzung im Kalman-Filter gestützt. Simulationen zeigen zwar, dass die Kovarianzen der Lage- und Gyroskop-Bias-Fehler dadurch beschränkt werden können; allerdings wird nicht gezeigt, ob dies auch für die tatsächlichen Fehler gilt. Die vom Modell gelieferte Lage wird direkt als Stützinformation im Kalman-Filter verwendet, ohne die Korrelation zwischen den Messwerten zu berücksichtigen. Daher muss erwartet werden, dass die Schätzungen überoptimistisch sind.

Vissière [84] zeigt eine Anwendung modellgestützter Navigation für Helikopter, die der von Sendobry [75] ähnelt. Trotz einer aufwändigen Modellierung und der Schätzung von 23 Zustandsgrößen treten teilweise große Fehler auf, wenn GPS nicht zur Stützung verwendet wird.

Bristeau verwendet ein aerodynamisches Quadrocoptermodell zur Bestimmung der Geschwindigkeit mit Hilfe der Beschleunigungsmesser [16]. Dies ist möglich, da die x- und y-Beschleunigungsmesser nur Beschleunigung durch Luftwiderstand, der am Körper des Fluggerätes sowie an den Rotoren auftritt, messen. Das hier vorgestellte Prinzip begrenzt zwar wirksam den Fehler der Geschwindigkeitsschätzung, aber

die Lageschätzung wird in einem eigenen Filter durchgeführt, so dass dieser vom Modellwissen nicht profitiert.

Dieser Nachteil wurde von Leishman et al. in [56] behoben, indem die Lageschätzung in das eigentliche Navigationsfilter integriert wurde. Die Propagation der Geschwindigkeit erfolgt über das Bewegungsmodell, die horizontalen Beschleunigungsmessungen werden in einer Stützmessung verarbeitet. Hierbei wirkt sich derselbe Effekt wie bei Bristeau aus, so dass die Geschwindigkeiten aus den Beschleunigungen beobachtet werden können. Auch wenn das vorgestellte Navigationssystem auf eine Schätzung der zeitveränderlichen Biase verzichtet, konnten hier gute Ergebnisse demonstriert werden.

6.1.5 Parameterschätzung für Quadrocopter aus Flugdaten

In diesem Kapitel soll auch betrachtet werden, wie Modellparameter aus Flugdaten geschätzt werden können. Eine solche Schätzung kann entweder nach dem Flug erfolgen [3, 78], oder auch mittels rekursiver Schätzung während des Fluges. Solch eine Online-Parameterschätzung kann als rekursiver Least-Squares-Schätzer realisiert werden [28, 63], oder aber durch Erweiterung des Zustandsvektors ins Navigationsfilter integriert werden [23]. Dies ist sowohl in einem Extended Kalman-Filter wie auch Unscented Kalman-Filter möglich, laut den Untersuchungen in [23] liefern diese eine ähnliche Genauigkeit. Welches dieser Verfahren für das hier entwickelte Navigationssystem am besten geeignet ist, soll später durch einen Vergleich untersucht werden.

6.1.6 Windschätzung im Flug

Als zusätzliche Erweiterung soll in diesem Kapitel auch die Möglichkeit zur Windschätzung im Flug aufgezeigt werden, um den Einfluss von Wind auf das Navigationssystem kompensieren zu können.

Ansätze zur Windschätzung aus Flugdaten werden häufig in Flugzeuganwendungen implementiert. Bei diesen Anwendungen kann davon ausgegangen werden, dass Staurohrmessungen verfügbar sind und somit die Geschwindigkeit des Flugzeugs relativ zur umgebenden Luft bekannt ist. In [68, 89, 90] erfolgt dann die Schätzung der Windgeschwindigkeit, indem die Geschwindigkeit über Grund von der gemessenen Geschwindigkeit durch die Luft subtrahiert wird. Die Geschwindigkeit über Grund wird dabei aus GPS-Messungen bzw. aus einem GPS- oder bildgestützten

Navigationssystem gewonnen. Bei Zhao [90] wird die Windgeschwindigkeit in einem EKF geschätzt, in dem der Wind modelliert ist als Überlagerung eines statischen Anteils mit einem abklingenden Anteil, der Böen repräsentiert.

Leutenegger [58] nutzt zwar auch ein Staurohr, verwendet in der Messgleichung aber das Bewegungsmodell des Fluggerätes, um dadurch die Windgeschwindigkeit im Navigationsfilter als Zustand schätzen zu können. Die Arbeit von Görcke [35] ist dem später verwendeten Ansatz sehr ähnlich, da hier ebenfalls der Zustandsvektor erweitert wird und die Unified-Model-Methode zum Einsatz kommt. Allerdings verwendet Görcke nur ein Gauß-Markov-Modell erster Ordnung zur Modellierung der Windgeschwindigkeit. Später wird gezeigt, dass für die Quadropteranwendung die erweiterte Modellierung nach [90] vorteilhaft ist.

Im Gegensatz zu Flugzeugen ist bei Quadroptern kein Staurohr zur Bestimmung der Geschwindigkeit relativ zur Luft vorhanden. Allerdings sind die Beschleunigungsmessungen aufgrund des Strömungswiderstands abhängig von dieser Relativgeschwindigkeit. Waslander [85] und Abeywardena [6] nutzen diesen Effekt, um die Windgeschwindigkeit auch auf Quadroptern zu ermitteln, wobei dazu von der Verfügbarkeit von präzisen DGPS-Messungen [85] oder optischer Stützung durch eine Kamera [6] ausgegangen wird. Der Einfluss des Windes auf die Beschleunigungsmessungen wird auch bei der in Unterkapitel 6.5 vorgestellten Realisierung implizit durch die Unified-Model-Methode ausgenutzt.

6.1.7 Fazit

Alle bestehenden generellen Ansätze, Modellstützung in ein Navigationssystem zu integrieren, weisen Nachteile auf. Nur die Ansätze nach Koifman [53], Sendobry [75] und Vasconcelos [83] können als optimal angesehen werden in dem Sinne, dass die verfügbare Information vollständig vom System genutzt wird. Der vergrößerte Zustandsvektor bei Koifman und Sendobry und die hohe Komplexität des Verfahrens nach Vasconcelos motivieren die Entwicklung eines neuen Verfahrens, das bei vergleichbaren Ergebnissen eine Reduktion des Zustandsvektors und der Filterkomplexität anstreben sollte.

Von den auf Helikopter angepassten Verfahren scheint das nach Leishman [56] am besten geeignet. Dieses wurde parallel zu dem weiter unten präsentierten Ansatz entwickelt und weist vergleichbare Ergebnisse auf. Allerdings handelt es sich um einen auf Helikopter beschränkten Ansatz, während das unten vorgestellte Unified

Model z. B. auch auf Starrflügler angewendet werden kann [35]. Der Einsatz für Quadrocopter ist nur ein mögliches Anwendungsfeld.

6.2 Unified Model: Methode zur Fusion von Propagationsmodellen

In diesem Abschnitt soll, basierend auf der ursprünglichen Methode von Koifman und Bar-Itzhack [53], ein neues Verfahren hergeleitet werden, mit dem ein Bewegungsmodell mit inertialer Navigation fusioniert werden kann. Diese Herleitung entstand im Rahmen einer gemeinschaftlichen Forschungsarbeit zwischen dem Institut für Theoretische Elektrotechnik und Systemoptimierung in Karlsruhe und dem Lehrstuhl für Flugsystemdynamik an der TU München und ist an [99] und [100] angelehnt.

Vor der eigentlichen Herleitung soll zunächst die Methode nach Koifman detailliert dargestellt werden, wobei sie im Vergleich zum ursprünglichen Paper [53] nicht auf die Anwendung im Flugzeug spezialisiert, sondern in allgemeiner Form präsentiert wird. Sowohl diese ursprüngliche Methode sowie das daraus neu abgeleitete Unified Model können nämlich generell dann eingesetzt werden, wenn zwei verschiedene aber kompatible Propagationsmodelle für ein Kalman-Filter vorliegen, d. h. auch außerhalb der Anwendung in der Navigation. Die ursprüngliche Methode wird in [47] daher auch als kombinierte Filterstruktur oder als „Horizontal Model Fusion“ bezeichnet, also eine Fusion von Modellen auf gleicher Ebene.

In der so entwickelten formalen Darstellung der Ursprungsmethode werden anschließend die Gleichungen für die Propagation und die Pseudomessung zusammengefasst. Das ermöglicht im nächsten Schritt die Elimination der doppelt enthaltenen Zustandsgrößen und daraufhin die Formulierung der eigentlichen Gleichungen der neuen Methode. Es zeigt sich, dass das resultierende System ein klassisches Kalman-Filter ist, bei dem aber die Propagation mittels einer optimalen Kombination der beiden Modelle erfolgt. Das so entstandene kombinierte Modell wird als Unified Model bezeichnet.

6.2.1 Kombinierte Filterstruktur

6.2.1.1 Partitionierung der Zustandsvektoren

Die von Koifman und Bar-Itzhack verwendete „Kombinierte Filterstruktur“ lässt sich allgemein für zwei Systemmodelle A und B formulieren. Es wird angenommen, dass diese die gleichen Zustandsgrößen propagieren, aber möglicherweise jeweils noch zusätzliche modellspezifische Zustandsgrößen enthalten können, die dann unterschiedlich sind. Ohne Beschränkung der Allgemeinheit ist es daher zweckmäßig, die Zustandsvektoren der beiden Modelle so zu ordnen, dass folgende partitionierte Zustandsvektoren entstehen:

$$\mathbf{x}_A = \begin{pmatrix} \mathbf{x}_{cA} \\ \mathbf{x}_{sA} \end{pmatrix} \quad (6.1)$$

$$\mathbf{x}_B = \begin{pmatrix} \mathbf{x}_{cB} \\ \mathbf{x}_{sB} \end{pmatrix} \quad (6.2)$$

Der Index s in \mathbf{x}_{sA} und \mathbf{x}_{sB} bezeichnet dabei die spezifisch zu A und B gehörenden Zustandsgrößen, während der Index c in \mathbf{x}_{cA} und \mathbf{x}_{cB} die gemeinsamen Zustände (engl. „common states“) beschreibt.

Im Rahmen der modellgestützten Navigation könnte Modell A das INS und Modell B das Bewegungsmodell beschreiben. Dann könnten die Zustandsvektoren beispielsweise wie folgt aussehen:

$$\mathbf{x}_A = \begin{pmatrix} \left. \begin{matrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^n \\ \mathbf{q}_b^n \end{matrix} \right\} \mathbf{x}_{cA} \\ \left. \begin{matrix} \mathbf{b}_a \\ \mathbf{b}_\omega \end{matrix} \right\} \mathbf{x}_{sA} \end{pmatrix} \quad (6.3)$$

$$\mathbf{x}_B = \begin{pmatrix} \left. \begin{matrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^n \\ \mathbf{q}_b^n \end{matrix} \right\} \mathbf{x}_{cB} \\ \left. \begin{matrix} \mathbf{v}_{wind} \\ \boldsymbol{\omega} \end{matrix} \right\} \mathbf{x}_{sB} \end{pmatrix} \quad (6.4)$$

Die Navigationsgrößen, also Position, Geschwindigkeit und Lage, sind in beiden Zustandsvektoren enthalten und bilden somit die gemeinsamen Zustände \mathbf{x}_c . In dem hier gezeigten Beispiel stellen die Inertialsensor-Biase die INS-spezifischen Zustände dar, während eine Windschätzung und die Drehrate des Flugzeugs spezifisch für das Bewegungsmodell sind.

Die Modelle weisen unterschiedliche Systemfunktionen $\mathbf{f}_A \neq \mathbf{f}_B$, Systemeingangsvektoren $\mathbf{u}_A \neq \mathbf{u}_B$ und Rauscheingangsmatrizen $\mathbf{G}_A \neq \mathbf{G}_B$ auf. Folglich sind auch die linearisierten Systemmatrizen Φ_A und Φ_B unterschiedlich.

Mit der oben eingeführten Schreibweise können auch diese Matrizen partitioniert dargestellt werden:

$$\Phi_A = \begin{pmatrix} \Phi_{cA} & \Phi_{csA} \\ \Phi_{scA} & \Phi_{sA} \end{pmatrix}, \Phi_B = \begin{pmatrix} \Phi_{cB} & \Phi_{csB} \\ \Phi_{scB} & \Phi_{sB} \end{pmatrix} \quad (6.5)$$

$$\mathbf{G}_A = \begin{pmatrix} \mathbf{G}_{cA} \\ \mathbf{G}_{sA} \end{pmatrix}, \mathbf{G}_B = \begin{pmatrix} \mathbf{G}_{cB} \\ \mathbf{G}_{sB} \end{pmatrix} \quad (6.6)$$

$$\mathbf{B}_A = \begin{pmatrix} \mathbf{B}_{cA} \\ \mathbf{B}_{sA} \end{pmatrix}, \mathbf{B}_B = \begin{pmatrix} \mathbf{B}_{cB} \\ \mathbf{B}_{sB} \end{pmatrix} \quad (6.7)$$

6.2.1.2 Kombination zu einem gemeinsamen System

Mit den so vorliegenden Matrizen wird nun ein kombiniertes System aufgestellt, das dann den Zustandsvektor

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_{cA} \\ \mathbf{x}_{sA} \\ \mathbf{x}_{cB} \\ \mathbf{x}_{sB} \end{pmatrix} \quad (6.8)$$

aufweist. Der erste Teil darin wird mit der Systemfunktion \mathbf{f}_A propagiert, der zweite Teil mit \mathbf{f}_B . Dadurch ergeben sich die Systemmatrix

$$\Phi = \begin{pmatrix} \Phi_A & \mathbf{0} \\ \mathbf{0} & \Phi_B \end{pmatrix} = \begin{pmatrix} \Phi_{cA} & \Phi_{csA} & \mathbf{0} & \mathbf{0} \\ \Phi_{scA} & \Phi_{sA} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi_{cB} & \Phi_{csB} \\ \mathbf{0} & \mathbf{0} & \Phi_{scB} & \Phi_{sB} \end{pmatrix} \quad (6.9)$$

und die kombinierten Eingangsmatrizen

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_A \\ \mathbf{B}_B \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{cA} \\ \mathbf{B}_{sA} \\ \mathbf{B}_{cB} \\ \mathbf{B}_{sB} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_a \\ \mathbf{G}_b \end{pmatrix} = \begin{pmatrix} \mathbf{G}_{cA} \\ \mathbf{G}_{sA} \\ \mathbf{G}_{cB} \\ \mathbf{G}_{sB} \end{pmatrix}. \quad (6.10)$$

Für diese Darstellung wurde angenommen, dass die Eingangsmatrizen bereits für kombinierte Eingangsvektoren \mathbf{u} und \mathbf{w} aufgestellt sind.

6.2.1.3 Zustandsgleichheit als Nebenbedingung

Im bisher betrachteten System werden die gemeinsamen Zustände \mathbf{x}_{cA} und \mathbf{x}_{cB} doppelt und unabhängig voneinander propagiert. Die Fusion der Modelle A und B kann nun durchgeführt werden: Dazu wird die Nebenbedingung genutzt, dass \mathbf{x}_{cA} und \mathbf{x}_{cB} gleich sein müssen, weil sie die selben physikalischen Zustände beschreiben. In einem Kalman-Filter kann eine solche Nebenbedingung in Form einer Messgleichung ausgedrückt werden. Es muss gelten:

$$\mathbf{h}(\mathbf{x}) = \mathbf{x}_{cA} - \mathbf{x}_{cB} = \mathbf{0} \quad (6.11)$$

Üblicherweise steht auf der rechten Seite einer solchen Messgleichung ein Messwert aus einem Sensor. Dieser wird hier ersetzt durch die geforderte Differenz $\tilde{\mathbf{y}}_k = \mathbf{0}$, es ist also kein Sensor notwendig. Daher spricht man bei dieser Stützinformation von einer Pseudomessung. Um diese verarbeiten zu können, ist neben dem Messwert auch eine Kovarianzmatrix des Messfehlers erforderlich.

Für die folgende Herleitung des Unified Models ist es wichtig festzuhalten, dass diese Kovarianzmatrix des Messfehlers

$$\mathbf{R} = \mathbf{0} \quad (6.12)$$

lautet. Dies folgt direkt aus der Nebenbedingung: Diese ist fehlerfrei, weil \mathbf{x}_{cA} und \mathbf{x}_{cB} exakt gleich sind.

Daraus darf natürlich nicht geschlossen werden, dass die Modelle A und B fehlerfrei sein müssten oder als solches betrachtet würden. Beide Modelle sind – über die Matrizen \mathbf{G}_a und \mathbf{G}_b – von einem Rauschen beeinflusst, wodurch der jeweilige Modellfehler abgebildet wird.

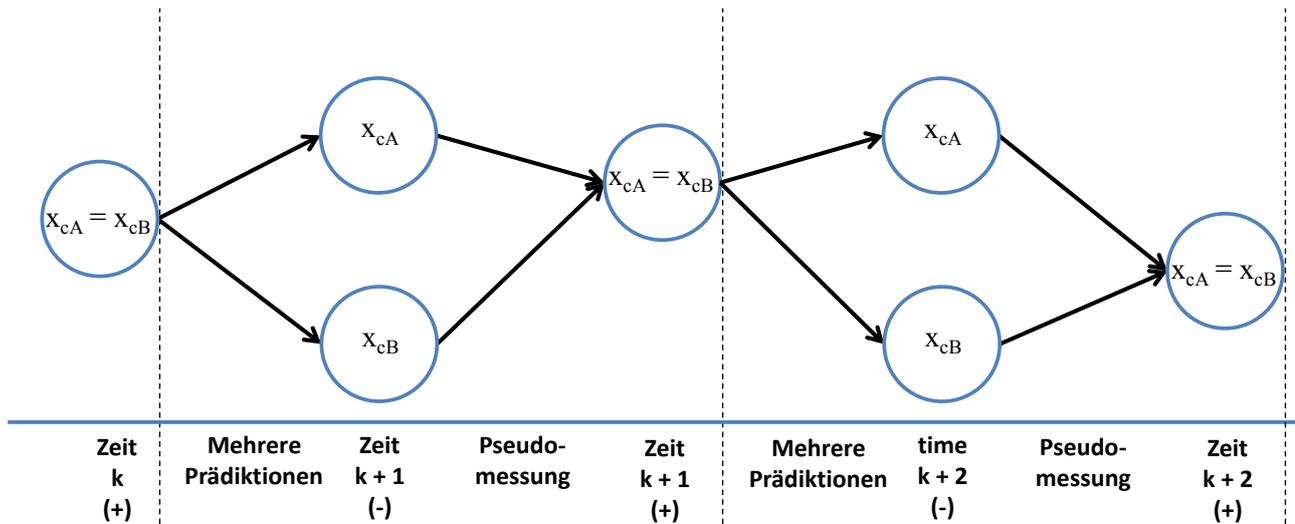


Abbildung 6.1: Verlauf der gemeinsamen Zustandsgrößen in der Kombinierten Filterstruktur nach Koifman und Bar-Itzhack: Nach den Prädiktionen mit den beiden Teilmodellen fusioniert die Pseudomessung x_{cA} und x_{cB} zu einer gemeinsamen Schätzung.

Die in Gleichung (6.11) dargestellte Messgleichung ist linear in den Zustandsgrößen und kann somit ohne Linearisierungsfehler auch geschrieben werden als

$$\mathbf{H}\mathbf{x} = \mathbf{x}_{cA} - \mathbf{x}_{cB} = \mathbf{0}, \quad (6.13)$$

mit der Messmatrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{pmatrix}. \quad (6.14)$$

Das so entstandene System wird in Abbildung 6.1 veranschaulicht: Die gemeinsamen Zustände werden zuerst unterschiedlich propagiert, mittels der Pseudomessung dann aber zu einer gemeinsamen Schätzung fusioniert, bevor wieder der nächste Prädiktions-Korrektur-Zyklus beginnt.

6.2.1.4 Kombinierte Filterstruktur in Error-State-Formulierung

Wenn das Kalman-Filter, wie bei Navigationsanwendungen oft üblich, nicht die absoluten Größen schätzt sondern die Fehler einer externen Navigationslösung, müssen folgende Anpassungen bei der Implementierung der diskutierten Filterstruktur vorgenommen werden:

Zum einen propagieren die Modelle A und B dann getrennte Kopien der externen Navigationslösung, der Zustandsvektor ist aufgrund des Rückkoppelns der Fehler $\mathbf{0}$. Wesentlich ist aber die Änderung in der Messgleichung. Es gilt weiterhin für die

wahren Größen $\mathbf{x}_{cA} - \mathbf{x}_{cB} = \mathbf{0}$. Daraus folgt dann

$$(\hat{\mathbf{x}}_{cA} - \delta\mathbf{x}_{cA}) - (\hat{\mathbf{x}}_{cB} - \delta\mathbf{x}_{cB}) = \hat{\mathbf{x}}_{cA} - \hat{\mathbf{x}}_{cB} - (\delta\mathbf{x}_{cA} - \delta\mathbf{x}_{cB}) = \mathbf{0} \quad (6.15)$$

und

$$\delta\mathbf{x}_{cA} - \delta\mathbf{x}_{cB} = H\delta\mathbf{x} = \hat{\mathbf{x}}_{cA} - \hat{\mathbf{x}}_{cB} = \delta\tilde{\mathbf{y}}. \quad (6.16)$$

Während die Messmatrix \mathbf{H} also gleich bleibt, entsteht ein Messwert, der nicht mehr 0 ist wie in Total-State-Formulierung. Stattdessen ist der Messwert nun die Differenz der geschätzten absoluten Größen.

Obwohl die Messgleichung linear ist, ist nach der Korrektur der absoluten Größen die Nebenbedingung unter Umständen nicht exakt erfüllt. Dies kann dann auftreten, wenn eine nichtlineare Korrektur der absoluten Größen durchgeführt wird, wie dies z. B. bei Schätzung von Lagewinkelfehlern der Fall ist. In [100] wird dieser Sachverhalt näher erläutert und begründet, dass die Anwendung der Pseudomessung einmal pro Zeitschritt ausreichend ist, damit hierdurch kein anwachsender Fehler entsteht.

6.2.2 Zusammenfassen von Propagation und Pseudomessung

Im Folgenden soll das bisherige System umgeformt werden in die neue Unified-Model-Darstellung.

Dazu betrachtet man die Kalman-Filter-Gleichungen für Prädiktion und Korrektur, die sich mit dem im vorherigen Unterabschnitt begründeten Wert von $\mathbf{R} = \mathbf{0}$ wie folgt ergeben:

$$\mathbf{x}_{k+1}^- = \Phi_k \mathbf{x}_k^+ + \mathbf{B}_k \mathbf{u}_k \quad (6.17)$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (6.18)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \quad (6.19)$$

$$\mathbf{x}_{k+1}^+ = \mathbf{x}_{k+1}^- + \mathbf{K}_{k+1} (\tilde{\mathbf{y}}_{k+1} - \mathbf{H} \mathbf{x}_{k+1}^-) \quad (6.20)$$

Diese Gleichungen beschreiben in zwei Schritten die Transformation des Zustands \mathbf{x}_k^+ nach \mathbf{x}_{k+1}^+ . Durch Einsetzen von Gleichung (6.17) in Gleichung (6.20) kann diese Transformation leicht in einem einzigen Schritt ausgeführt werden:

$$\begin{aligned} \mathbf{x}_{k+1}^+ &= \Phi_k \mathbf{x}_k^+ + \mathbf{B}_k \mathbf{u}_k + \mathbf{K}_{k+1} (\tilde{\mathbf{y}}_{k+1} - \mathbf{H} (\Phi_k \mathbf{x}_k^+ + \mathbf{B}_k \mathbf{u}_k)) \\ &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \Phi_k \mathbf{x}_k^+ + (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{B}_k \mathbf{u}_k + \mathbf{K}_{k+1} \tilde{\mathbf{y}}_{k+1} \end{aligned} \quad (6.21)$$

Die zugehörige Kovarianz berechnet sich in der Josephsform [34], die sich durch ihre Symmetrie auszeichnet, als

$$\begin{aligned}
\mathbf{P}_{k+1}^+ &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^-(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})^T \\
&= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})(\Phi_k\mathbf{P}_k^+\Phi_k^T + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T)(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})^T \\
&= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\Phi_k\mathbf{P}_k^+\Phi_k^T(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})^T \\
&\quad + (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})^T.
\end{aligned} \tag{6.22}$$

Führt man die Matrizen $\bar{\Phi}_k = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\Phi_k$, $\bar{\mathbf{G}}_k = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{G}_k$ und $\bar{\mathbf{B}}_k = \begin{pmatrix} (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{B}_k & \mathbf{K}_{k+1} \end{pmatrix}$ ein, ergibt sich mit dem erweiterten Eingangsvektor $\bar{\mathbf{u}}_k = \begin{pmatrix} \mathbf{u}_k & \tilde{\mathbf{y}}_{k+1} \end{pmatrix}^T$

$$\mathbf{x}_{k+1}^+ = \bar{\Phi}_k\mathbf{x}_k^+ + \bar{\mathbf{B}}_k\bar{\mathbf{u}}_k \tag{6.23}$$

$$\mathbf{P}_{k+1}^+ = \bar{\Phi}_k\mathbf{P}_k^+\bar{\Phi}_k^T + \bar{\mathbf{G}}_k\mathbf{Q}_k\bar{\mathbf{G}}_k^T. \tag{6.24}$$

Dies entspricht den Kalman-Filter-Gleichungen für den Prädiktionsschritt. Das System $(\bar{\Phi}_k, \bar{\mathbf{G}}_k, \bar{\mathbf{B}}_k)$ stellt also ein Prädiktionsmodell für ein gewöhnliches Kalman-Filter dar. Allerdings sind direkt nach dieser Prädiktion die Zustandsgrößen \mathbf{x}_{cA} und \mathbf{x}_{cB} identisch, ohne dass explizit eine Pseudomessung nötig wäre. Diese ist bereits in der Prädiktion enthalten.

Diese Darstellung hebt den für die Pseudomessung anfallenden Rechenaufwand allerdings nicht auf. Er wird nur in den Prädiktionsschritt verschoben. Dennoch ist diese Zwischenstufe ein wichtiger Schritt der Herleitung zum Unified Model: Betrachtet man nämlich nun den in Abbildung 6.2 dargestellten Verlauf der Zustandsgrößen, fällt auf, dass \mathbf{x}_{cA} und \mathbf{x}_{cB} *immer* identisch und vollständig korreliert sind. Damit ist offensichtlich überflüssige Information im Zustandsvektor enthalten: Der Zustandsvektor kann reduziert werden, indem \mathbf{x}_{cA} oder \mathbf{x}_{cB} eliminiert wird.

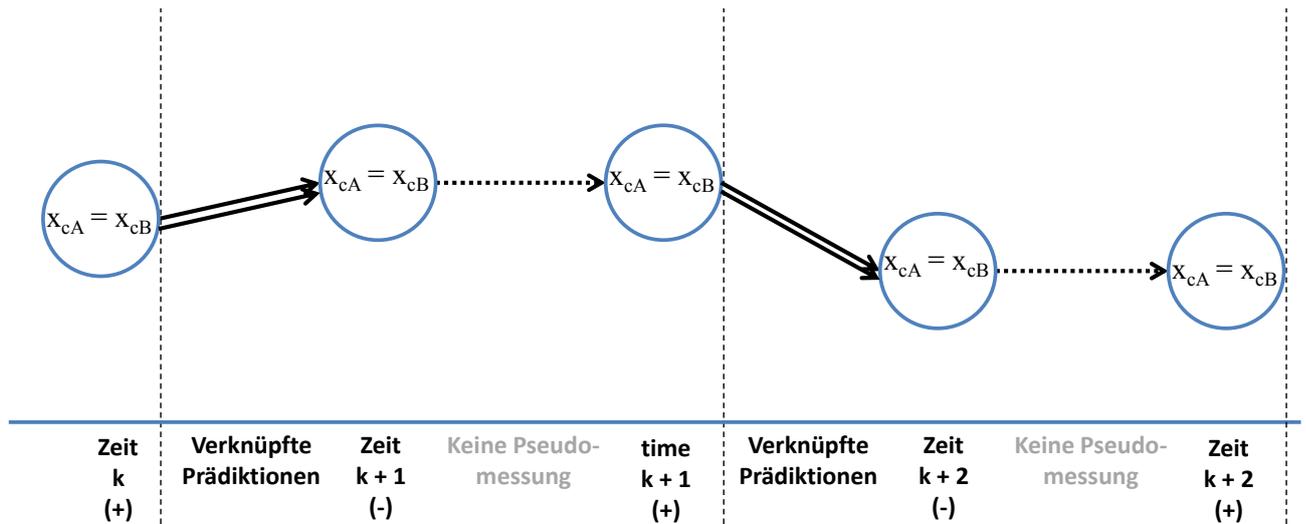


Abbildung 6.2: Verlauf der gemeinsamen Zustandsgrößen nach Zusammenfassen von Propagation und Pseudomessung. x_{cA} und x_{cB} werden im neuen Prädiktionsschritt direkt zur gemeinsamen Schätzung propagiert.

6.2.3 Reduktion des Zustandsvektors

Aufgrund der vorangegangenen Betrachtungen kann der Zustandsvektor nun unter Ausnutzung von $x_{cA} = x_{cB} = x_c$ verkleinert werden zu

$$\mathbf{x}_{\cup,k} = \begin{pmatrix} \mathbf{x}_{c,k} \\ \mathbf{x}_{sA,k} \\ \mathbf{x}_{sB,k} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\mathbf{T}_{3 \times 4}} \begin{pmatrix} \mathbf{x}_{cA} \\ \mathbf{x}_{sA} \\ \mathbf{x}_{cB} \\ \mathbf{x}_{sB} \end{pmatrix}. \quad (6.25)$$

Die Wahl der Matrix $\mathbf{T}_{3 \times 4}$, mit der der bisherige Zustandsvektor auf den neuen, verkleinerten Zustandsvektor abgebildet wird, ist nicht eindeutig. So wäre auch

$$\check{\mathbf{T}}_{3 \times 4} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (6.26)$$

eine gültige Wahl, die im Folgenden zu anderen, aber äquivalenten Gleichungen führen würde.

Mittels der Matrix $\mathbf{T}_{4 \times 3}$ lässt sich der reduzierte auf den bisherigen Zustandsvektor abbilden:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{x}_{cA} \\ \mathbf{x}_{sA} \\ \mathbf{x}_{cB} \\ \mathbf{x}_{sB} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\mathbf{T}_{4 \times 3}} \begin{pmatrix} \mathbf{x}_{c,k} \\ \mathbf{x}_{sA,k} \\ \mathbf{x}_{sB,k} \end{pmatrix} \quad (6.27)$$

Setzt man $\mathbf{x}_k = \mathbf{T}_{4 \times 3} \mathbf{x}_{\cup,k}$ in Gleichung (6.23) ein, ergibt sich

$$\mathbf{T}_{4 \times 3} \mathbf{x}_{\cup,k+1}^+ = \bar{\Phi}_k \mathbf{T}_{4 \times 3} \mathbf{x}_{\cup,k}^+ + \bar{\mathbf{B}}_k \bar{\mathbf{u}}_k, \quad (6.28)$$

das sich wegen $\mathbf{T}_{3 \times 4} \mathbf{T}_{4 \times 3} = \mathbf{I}_{3 \times 3}$ umformen lässt zu

$$\mathbf{x}_{\cup,k+1}^+ = \mathbf{T}_{3 \times 4} \bar{\Phi}_k \mathbf{T}_{4 \times 3} \mathbf{x}_{\cup,k}^+ + \mathbf{T}_{3 \times 4} \bar{\mathbf{B}}_k \bar{\mathbf{u}}_k, \quad (6.29)$$

so dass die Prädiktionsgleichung für \mathbf{x}_{\cup} vorliegt.

Wegen $\mathbf{x}_k = \mathbf{T}_{4 \times 3} \mathbf{x}_{\cup,k}$ folgt

$$\mathbf{P}_k = \mathbf{T}_{4 \times 3} \mathbf{P}_{\cup,k} \mathbf{T}_{4 \times 3}^T. \quad (6.30)$$

Multipliziert man Gleichung (6.24) von links mit $\mathbf{T}_{3 \times 4}$ und von rechts mit $\mathbf{T}_{3 \times 4}^T$ ergibt sich nach Einsetzen von Gleichung (6.30):

$$\mathbf{P}_{\cup,k+1}^+ = \mathbf{T}_{3 \times 4} \mathbf{P}_{k+1}^+ \mathbf{T}_{3 \times 4}^T = \mathbf{T}_{3 \times 4} \bar{\Phi}_k \mathbf{T}_{4 \times 3} \mathbf{P}_{\cup,k}^+ \mathbf{T}_{4 \times 3}^T \bar{\Phi}_k^T \mathbf{T}_{3 \times 4}^T + \mathbf{T}_{3 \times 4} \bar{\mathbf{G}}_k \mathbf{Q}_k \bar{\mathbf{G}}_k^T \mathbf{T}_{3 \times 4}^T. \quad (6.31)$$

Auch die Gleichungen (6.29) und (6.31) sind Prädiktionsgleichungen, und zwar für das System mit dem reduzierten Zustandsvektor \mathbf{x}_{\cup} . Die Systemmatrizen sind hierbei

$$\Phi_{\cup,k} = \mathbf{T}_{3 \times 4} \bar{\Phi}_k \mathbf{T}_{4 \times 3}, \quad \mathbf{G}_{\cup,k} = \mathbf{T}_{3 \times 4} \bar{\mathbf{G}}_k, \quad \mathbf{B}_{\cup,k} = \mathbf{T}_{3 \times 4} \bar{\mathbf{B}}_k. \quad (6.32)$$

6.2.4 Unified-Model-Gleichungen

Das in den vorherigen Abschnitten hergeleitete System $(\Phi_{\cup,k}, \mathbf{G}_{\cup,k}, \mathbf{B}_{\cup,k})$ mit dem reduzierten Zustandsvektor \mathbf{x}_{\cup} ist mathematisch äquivalent zum ursprünglichen System nach Koifman. Dennoch konnte der Zustandsvektor wesentlich verkleinert werden.

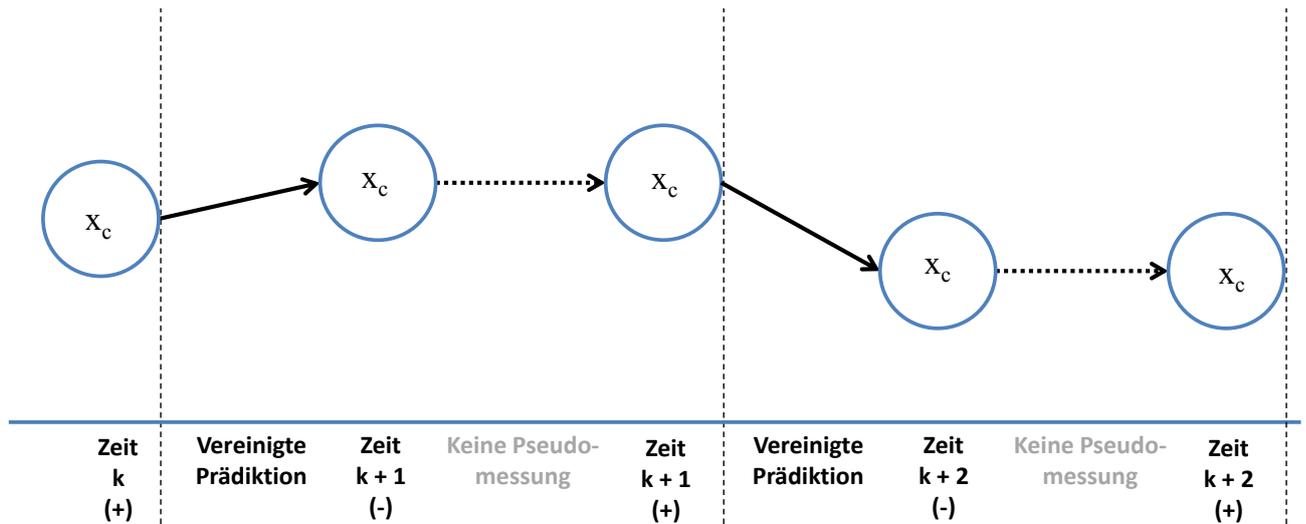


Abbildung 6.3: Verlauf der gemeinsamen Zustandsgrößen nach Reduktion des Zustandsvektors. Das Unified Model verknüpft die Teilmodelle bereits in der Prädiktion. Es sind keine duplizierten Zustände oder Pseudomessungen nötig.

Da die beiden Modelle A und B in der Berechnung des neuen Systems zu einem Modell fusioniert werden, wird dieses neue Modell als Unified Model bezeichnet. Der Ablauf ist in Abbildung 6.3 dargestellt, in der zu sehen ist, dass die Zustände x_{cA} und x_{cB} aus Abbildung 6.2 zu x_c verschmolzen sind, die Werte sind identisch.

In der bisherigen Herleitung ist zur Berechnung der neuen Systemmatrizen allerdings immer erst eine Formulierung des Ansatzes nach Koifman und Bar-Itzhack nötig. Dies ist nicht wünschenswert, weil dabei einige Matrizen mit der Größe des ursprünglichen Zustands berechnet und erst durch die Transformationsmatrix $\mathbf{T}_{3 \times 4}$ verkleinert werden. Gesucht ist stattdessen eine eigenständige, direkte Berechnung des neuen Systems ausgehend von den Systemmodellen A und B, also $(\Phi_A, \mathbf{G}_A, \mathbf{B}_A)$ und $(\Phi_B, \mathbf{G}_B, \mathbf{B}_B)$. Diese findet man, indem man die Definition der Matrizen $\bar{\Phi}_k$, $\bar{\mathbf{G}}_k$ und $\bar{\mathbf{B}}_k$ sowie von \mathbf{K}_{k+1} und \mathbf{P}_{k+1}^+ in Gleichung (6.32) einsetzt, die entstehenden Ausdrücke geeignet ausmultipliziert und dann wieder zusammenfasst. Diese Herleitung ist ausführlich in Anhang A aufgezeigt. Als Ergebnis resultieren die folgenden Gleichungen für das Unified Model:

Zuerst stellt man die Matrizen

$$\Phi_{T,k} = \begin{pmatrix} \Phi_{cA} & \Phi_{csA} & \mathbf{0} \\ \Phi_{scA} & \Phi_{sA} & \mathbf{0} \\ \Phi_{scB} & \mathbf{0} & \Phi_{sB} \end{pmatrix}, \quad (6.33)$$

$$\Phi_{H,k} = \begin{pmatrix} \Phi_{cA} & -\Phi_{cB} & \Phi_{csA} & -\Phi_{csB} \end{pmatrix} \quad (6.34)$$

und

$$\mathbf{G}_{T,k} = \begin{pmatrix} \mathbf{G}_{c,A} \\ \mathbf{G}_{sA} \\ \mathbf{G}_{sB} \end{pmatrix}, \quad \mathbf{G}_{H,k} = \mathbf{G}_{cA} - \mathbf{G}_{cB} \quad (6.35)$$

auf, indem die bekannten Submatrizen der Systemmodelle A und B entsprechend angeordnet werden.

Es zeigt sich, dass die Definition der Matrix

$$\mathbf{U}_k = \begin{pmatrix} \mathbf{U}_c \\ \mathbf{U}_{sA} \\ \mathbf{U}_{sB} \end{pmatrix}_k = \left(\Phi_{T,k} \mathbf{P}_{\cup,k} \Phi_{H,k}^T + \mathbf{G}_{T,k} \mathbf{Q}_k \mathbf{G}_{H,k}^T \right) \cdot \left(\Phi_{H,k} \mathbf{P}_{\cup,k} \Phi_{H,k}^T + \mathbf{G}_{H,k} \mathbf{Q}_k \mathbf{G}_{H,k}^T \right)^{-1} \quad (6.36)$$

eine sehr übersichtliche Berechnung der neuen Matrizen erlaubt:

$$\Phi_{\cup,k} = \Phi_{T,k} - \mathbf{U}_k \Phi_{H,k} \quad (6.37)$$

$$\mathbf{G}_{\cup,k} = \mathbf{G}_{T,k} - \mathbf{U}_k \mathbf{G}_{H,k} \quad (6.38)$$

$$\mathbf{B}_{\cup,k} = \mathbf{B}_{T,k} - \mathbf{U}_k \mathbf{B}_{H,k} \quad (6.39)$$

Der von diesem System propagierte Zustand

$$\mathbf{x}_{\cup,k} = \begin{pmatrix} \mathbf{x}_c \\ \mathbf{x}_{sA} \\ \mathbf{x}_{sB} \end{pmatrix} \quad (6.40)$$

enthält die sowohl in \mathbf{x}_A als auch \mathbf{x}_B gemeinsam enthaltenen Zustände nur noch einmal.

Bei gewöhnlichen Implementierungen von Error-State-Navigationsfiltern ist der Eingangsvektor \mathbf{u}_k leer, da die Eingangsgrößen in der nichtlinearen Prädiktion des absoluten, außerhalb des Filters gespeicherten Zustands einfließen. Daher ist der prädizierte Fehler $\delta \mathbf{x}_k^+ = 0$. In den oben hergeleiteten Gleichungen für das Unified Model wurde – im Falle einer Error-State-Implementierung – der Eingangsvektor aber erweitert, so dass hier eine Prädiktion gemäß

$$\delta \tilde{\mathbf{y}} = \mathbf{f}_A(\mathbf{x}_c, \mathbf{x}_{sA}, \mathbf{u}_A) - \mathbf{f}_B(\mathbf{x}_c, \mathbf{x}_{sB}, \mathbf{u}_B) \quad (6.41)$$

$$\delta \mathbf{x}_{\cup, k+1}^- = \underbrace{\Phi_{\cup, k} \delta \mathbf{x}_{\cup, k}^+}_{=0} + \mathbf{U}_k \delta \tilde{\mathbf{y}} = \mathbf{U}_k \delta \tilde{\mathbf{y}} \quad (6.42)$$

erfolgt. Daher sollte auch nach diesem Prädiktionsschritt eine Rückkopplung auf die absoluten Größen erfolgen.

Damit bei der Inversion für die Berechnung der Matrix \mathbf{U}_k gemäß Gleichung (6.36) keine numerischen Probleme auftreten, wurde die zu invertierende Matrix $\Phi_{H, k} \mathbf{P}_{\cup, k} \Phi_{H, k}^T + \mathbf{G}_{H, k} \mathbf{Q}_k \mathbf{G}_{H, k}^T$ in [100] genauer betrachtet. Es zeigt sich, dass diese insbesondere dann singular werden kann, wenn $\mathbf{G}_{H, k}$ Nullzeilen aufweist. Dies ist ein Ausdruck dafür, dass Teile der beiden Modelle identisch sind, z. B. wenn in beiden Modellen die Position durch Integration der Geschwindigkeit gewonnen wird. Ohne Informationsverlust können daher entsprechende Zeilen aus \mathbf{G}_H , Φ_H und $\delta \tilde{\mathbf{y}}$ gestrichen werden. Dadurch ist eine numerisch stabile Berechnung von \mathbf{U}_k möglich; gleichzeitig wird sogar noch die Rechenzeit reduziert.

Der Algorithmus des Unified-Model-Propagationsschritts lautet zusammengefasst also wie folgt:

1. Berechnung der Matrizen $\Phi_{T, k}$, $\Phi_{H, k}$, $\mathbf{G}_{T, k}$ and $\mathbf{G}_{H, k}$ (Gleichungen (6.33)-(6.35))
2. Propagation der absoluten Größen jeweils mit Modell A und Modell B. Daraus resultieren $\mathbf{x}_{A, k}^-$, $\mathbf{x}_{B, k}^-$ sowie $\delta \tilde{\mathbf{y}}$ (Gleichung (6.41))
3. Streichen der Nullzeilen aus \mathbf{G}_H und der entsprechenden Zeilen aus $\Phi_{H, k}$ und $\delta \tilde{\mathbf{y}}$
4. Berechnung der Matrix \mathbf{U}_k (Gleichung (6.36))
5. Berechnung der Systemmatrizen des Unified Models (Gleichungen (6.37) bis (6.39))
6. Propagation der Fehlerkovarianz mit den üblichen Kalman-Filter-Gleichungen und den Unified-Model-Systemmatrizen
7. Propagation von $\delta \mathbf{x}_{\cup}$ (Gleichung (6.42))
8. Aktualisierung der absoluten gemeinsamen Zustandsgrößen aus der Propagation mit Modell A ($\mathbf{x}_{cA, k}^-$) basierend auf $\delta \mathbf{x}_c$ in $\delta \mathbf{x}_{\cup}$. Aktualisierung der spezifischen Zustandsgrößen für beide Modelle basierend auf $\delta \mathbf{x}_{\cup}$.

6.3 Minimale modellgestützte Navigation für Quadrocopter

Die im vorherigen Unterkapitel hergeleitete Methode soll nun zur Synthese eines Quadrocopter-Navigationssystems angewendet werden. Dazu ist es empfehlenswert, zuerst die zwei einzelnen Modelle, das INS und das Bewegungsmodell, zu implementieren und unabhängig voneinander zu verifizieren. Rauschparameter der Modelle können dann bei Fusion mit dem Unified Model direkt übernommen werden, weitere Parameter kommen durch die Modell-Fusion nicht hinzu.

In den folgenden Abschnitten werden zunächst die Einzelmodelle vorgestellt und dann die Fusion diskutiert. Es entsteht dabei ein minimalistisches System: Der Zustandsvektor muss im Vergleich zu einem klassischen Inertialen Navigationssystem nicht erweitert werden, um die Modellstützung hinzuzufügen. Das so entstehende Navigationssystem wird daraufhin einer theoretischen und experimentellen Analyse unterzogen; darin wird gezeigt, dass bereits in diesem minimalen Ansatz wesentliche Vorteile gegenüber einem System ohne Modellstützung liegen. Erweiterungen, die Verbesserungen für die praktische Anwendbarkeit bringen, werden in späteren Unterkapiteln vorgestellt.

6.3.1 Prädiktionsmodelle

6.3.1.1 Modell A: Inertiale Navigation

Das erste der beiden hier verwendeten zwei Prädiktionsmodelle ist ein klassisches Inertiales Navigationssystem (INS). Dieses verwendet die inertialen Messungen der IMU als Eingangsgrößen \mathbf{u}_k . Die zeitkontinuierlichen Differentialgleichungen des INS in Strapdown-Formulierung werden in der Literatur ausführlich beschrieben [36, 87]. Im Folgenden sollen diese Gleichungen zusammengefasst werden:

Die Lagedifferentialgleichung für die Richtungskosinusmatrix \mathbf{C}_b^n lautet

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b = \mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^b - (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \mathbf{C}_b^n, \quad (6.43)$$

wobei $\boldsymbol{\Omega}_{ib}^b$ die Kreuzproduktmatrix zur Rotationsrate $\boldsymbol{\omega}_{ib}^b$ ist. Wenn eine gemessene Drehrate $\tilde{\boldsymbol{\omega}}_{ib}^b$ und eine Schätzung des Gyroskop-Bias $\hat{\mathbf{b}}_\omega$ vorliegen, wird die geschätzte

Richtungskosinusmatrix $\hat{\mathbf{C}}_b^n$ mittels

$$\dot{\hat{\mathbf{C}}}_b^n = \hat{\mathbf{C}}_b^n (\tilde{\boldsymbol{\Omega}}_{ib}^b - \hat{\mathbf{B}}_\omega) - (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \hat{\mathbf{C}}_b^n \quad (6.44)$$

propagiert. $\hat{\mathbf{B}}_\omega$ ist dabei die Kreuzproduktbildende Matrix zu $\hat{\mathbf{b}}_\omega$, $\hat{\mathbf{B}}_\omega = [\hat{\mathbf{b}}_\omega \times]$.

Geschwindigkeits- und Positionspropagation bei n-frame-Geschwindigkeitsfehlern Für die Quadrocopter-Navigation wird üblicherweise eine Formulierung im Navigationskoordinatensystem gewählt.

Die Geschwindigkeitsdifferentialgleichung erhält man durch Zeit-Differentiation von

$$\mathbf{v}_{eb}^n = \mathbf{C}_e^n \mathbf{v}_{eb}^e, \quad (6.45)$$

sowie einigen Umstellungen und nach Ersetzen der Beschleunigung durch die Specific Force \mathbf{f}_{ib}^b und Gravitation γ . Diese wird mit der zentrifugalen Beschleunigung zur Schwerebeschleunigung \mathbf{g}_b zusammengefasst (vgl. [36]). Man erhält

$$\dot{\mathbf{v}}_{eb}^n = -(\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n) \mathbf{v}_{eb}^n + \mathbf{g}_b^n + \mathbf{C}_b^n \mathbf{f}_{ib}^b. \quad (6.46)$$

Auch hier kann wieder zu einer Differentialgleichung für die Schätzgrößen übergegangen werden. In dieser wird, basierend auf der gemessenen Specific Force $\tilde{\mathbf{f}}_{ib}^b$ und dem geschätzten Beschleunigungsmesser-Bias $\hat{\mathbf{b}}_a$, die geschätzte Geschwindigkeit $\hat{\mathbf{v}}_{eb}^n$ wie folgt propagiert:

$$\dot{\hat{\mathbf{v}}}_{eb}^n = -(\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n) \hat{\mathbf{v}}_{eb}^n + \mathbf{g}_b^n + \hat{\mathbf{C}}_b^n (\tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a) \quad (6.47)$$

Die Propagation der Position kann mit

$$\dot{\hat{\mathbf{p}}}_{eb}^n \approx \hat{\mathbf{v}}_{eb}^n \quad (6.48)$$

angenähert werden, wenn die Transportrate vernachlässigt wird. Diese Approximation ist für Quadrocopter gültig, da bei diesen nur geringe Geschwindigkeiten auftreten.

Geschwindigkeits- und Positionspropagation bei b-frame-Geschwindigkeitsfehlern Während der Entwicklung des modellgestützten Navigationssystems wurde deutlich, dass es vorteilhaft ist, die Geschwindigkeitsfehler im körperfesten Koordinatensystem zu schätzen. Um diese Wahl später begründen zu können, soll hier auch das Systemmodell mit b-frame-Geschwindigkeitsfehlern vorgestellt werden.

Dessen Herleitung, die ausführlich in Anhang B durchgeführt wird, beginnt mit der Differentiation von

$$\mathbf{v}_{eb}^b = \mathbf{C}_e^b \mathbf{v}_{eb}^e \quad (6.49)$$

nach der Zeit und führt schließlich auf

$$\dot{\mathbf{v}}_{eb}^b = -(\boldsymbol{\Omega}_{ie}^b + \boldsymbol{\Omega}_{ib}^b) \mathbf{v}_{eb}^b + \mathbf{C}_n^b \mathbf{g}_b^n + \mathbf{f}_{ib}^b \quad (6.50)$$

für die wahren Größen und

$$\dot{\hat{\mathbf{v}}}_{eb}^b = -(\boldsymbol{\Omega}_{ie}^b + \boldsymbol{\Omega}_{ib}^b) \hat{\mathbf{v}}_{eb}^b + \hat{\mathbf{C}}_n^b \mathbf{g}_b^n + \tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a \quad (6.51)$$

für die geschätzten und gemessenen Größen.

Nur die Geschwindigkeitsfehler sollten in Body-Koordinaten ausgedrückt werden. Positionsfehler können weiterhin in Navigationskoordinaten geschätzt werden, so dass für die Positionsdifferentialgleichung

$$\dot{\mathbf{p}}_{eb}^n \approx \mathbf{C}_b^n \mathbf{v}_{eb}^b \quad (6.52)$$

zum Einsatz kommt, wobei hier dieselbe Approximation wie in Gleichung (6.48) vorgenommen wurde.

Modellierung der Inertial-Biase Neben den Navigationsgrößen Position, Geschwindigkeit und Lage schätzt das Kalman-Filter auch die Biase der Beschleunigungs- und Drehratensensoren. Unter der Annahme kurzer Flugzeiten können diese als

$$\dot{\mathbf{b}}_a = \mathbf{w}_{b_a} \quad (6.53)$$

$$\dot{\mathbf{b}}_\omega = \mathbf{w}_{b_\omega} \quad (6.54)$$

modelliert werden. Dies entspricht einem zufälligen konstanten Wert, der mit einem Random-Walk-Prozess überlagert wird. Dieser Prozess wird vom weißen, gaußschen Rauschen \mathbf{w}_{b_a} bzw. \mathbf{w}_{b_ω} getrieben.

Systemrauschen Das Systemrauschen des INS-Modells kann im Vektor

$$\mathbf{w}_{\text{INS}} = \begin{pmatrix} \mathbf{w}_{f_i} \\ \mathbf{w}_\omega \\ \mathbf{w}_{b_a} \\ \mathbf{w}_{b_\omega} \end{pmatrix} \quad (6.55)$$

zusammengefasst werden, wobei w_{f_i} das Rauschen der Beschleunigungsmesser, w_ω das Rauschen der Gyroskope bezeichnen und w_{b_a} bzw. w_{b_ω} die jeweilige Biasinstabilität beschreiben.

6.3.1.2 Modell B: Translatorisches Bewegungsmodell

In dem hier entwickelten Navigationssystem kommt nur der translatorische Teil des Quadrocopter-Bewegungsmodells zum Einsatz. Untersuchungen im Rahmen dieser Arbeit zeigten, dass das rotatorische Modell keine nennenswerten Vorteile bringt, den Implementierungs- und Rechenaufwand jedoch erheblich erhöht.

Dennoch muss auch das zweite, später mit dem INS zu fusionierende Modell ein vollständiges Propagationsmodell sein. Um auf das rotatorische Bewegungsmodell zu verzichten, kann entweder die Lagedifferentialgleichung $\dot{\mathbf{q}} = 0$ angesetzt werden, wobei der dabei gemachte Fehler über ein ausreichend hoch gewähltes Systemrauschen in der Matrix \mathbf{Q} ausgedrückt werden muss. Ein Einstellen dieses Wertes kann vermieden werden, indem auch für das Modell B die Lagepropagation basierend auf den Drehratenmessungen durchgeführt wird. Bei der späteren Fusion kann die Tatsache, dass dieselbe Information in beiden Modellen verwendet wird, ausgedrückt und berücksichtigt werden.

Das Modell B unterscheidet sich von Modell A, also dem INS, somit nur in der zur Geschwindigkeitspropagation verwendeten Specific Force \mathbf{f}_{ib}^b . Anstatt $\tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a$ wird hier die aus dem translatorischen Bewegungsmodell geschätzte Specific Force $\hat{\mathbf{f}}_{ib}^b$ verwendet. Diese wird als

$$\hat{\mathbf{f}}_{ib}^b = \sum_{i=1}^4 \hat{\mathbf{f}}_{m,i}^b - k_f |\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb}^b \quad (6.56)$$

modelliert. Sie entspricht somit der Summe der spezifischen Motorkräfte $\hat{\mathbf{f}}_{m,i}^b$ und einem quadratischen Strömungswiderstand $-k_f |\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb}^b$. Dabei wird also vorerst davon ausgegangen, dass kein Wind vorliegt. Eine entsprechende Erweiterung des Modells wird in Abschnitt 6.5 diskutiert.

Die spezifischen Motorkräfte werden als

$$\hat{\mathbf{f}}_{m,i}^b = \mathbf{m}_i \omega_{m,i}^2 \quad (6.57)$$

modelliert, wobei $\omega_{m,i}$ die Rotationsgeschwindigkeit des Motors bezeichnet; \mathbf{m}_i ist der Vektor der Motorparameter.

k_f ist der Koeffizient zur Berechnung der Specific Force anhand des Strömungswiderstandes. Dieser fasst gemäß $k_f = \frac{1}{m} C_{D_f} \frac{A\rho}{2}$ den Kehrwert der Masse und die Vorfaktoren des Strömungswiderstands aus Abschnitt 3.4.1.2 zusammen. Diese Zusammenfassung ist zweckmäßig, da die Größen m , ρ , A und C_{D_f} im Rahmen der Navigation nicht einzeln relevant sind, sondern nur ihr Produkt Einfluss auf die Bewegung hat. Aufgrund von üblicherweise vorliegenden Asymmetrien des Quadropter-Aufbaus ist k_f abhängig von der Richtung der Anströmung – sowohl bezüglich dem Elevations- als auch dem Azimuthwinkel. Eine umfangreiche Bestimmung einer entsprechenden Wertetabelle wäre zwar in Windkanaltests oder basierend auf einem zu erstellenden CAD-Modell möglich. Allerdings wäre eine solche Bestimmung sehr zeit- und kostenaufwändig und müsste für jeden neuen Fluggerätetyp erneut durchgeführt werden. Durch Anbringung neuer Nutzlasten würden die aufwändig gewonnenen Werte wertlos. Daher ist eine einfachere Bestimmung des Strömungswiderstandskoeffizienten k_f erstrebenswert. In dieser Arbeit wurde k_f getrennt für Geschwindigkeiten in x^b -, y^b - und z^b -Richtung bestimmt, die dann als k_{f1} , k_{f2} und k_{f3} bezeichnet werden. Daraus lässt sich dann für Anströmung aus einer beliebigen Richtung bzw. einer beliebigen Geschwindigkeit \mathbf{v}_{eb}^b der gesamte Koeffizient approximieren mittels

$$k_f = \frac{1}{|\mathbf{v}_{eb}^b|^2} \sum_{i=1}^3 (v_{eb,i}^b)^2 k_{fi}. \quad (6.58)$$

Wenn die Anströmung entlang einer Koordinatenrichtung i vorliegt, führt dies zu $k_f = k_{fi}$. Wenn $k_{f1} = k_{f2} = k_{f3}$, gilt $k_f = k_{f1} = k_{f2} = k_{f3}$. Im allgemeinen Fall wird dagegen zwischen den Koeffizienten k_{fi} interpoliert.

Sollte für ein spezielles Fluggerät der Aufwand von Windkanaltests oder einer CAD-Modellierung erfolgt sein, lässt sich natürlich auch das daraus gewonnene k_f im Navigationssystem nutzen. Die Quelle dieser Information ist für das Navigationssystem unerheblich.

Geschwindigkeitspropagation Mit dem oben beschriebenen Modell wird die Geschwindigkeit in Navigationskoordinaten mittels der folgenden Differentialgleichung

propagiert:

$$\dot{\hat{\mathbf{v}}}_{eb}^n = -(\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n)\hat{\mathbf{v}}_{eb}^n + \mathbf{g}_b^n + \hat{\mathbf{C}}_b^n \left(\sum_{i=1}^4 \hat{\mathbf{f}}_{m,i}^b - k_f |\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb}^b \right) \quad (6.59)$$

$$= -(\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n)\hat{\mathbf{v}}_{eb}^n + \mathbf{g}_b^n + \sum_{i=1}^4 \hat{\mathbf{C}}_b^n \hat{\mathbf{f}}_{m,i}^b - k_f |\hat{\mathbf{v}}_{eb}^n| \hat{\mathbf{v}}_{eb}^n \quad (6.60)$$

Bei Schätzung der Geschwindigkeit im Body-Frame ergibt sich

$$\dot{\hat{\mathbf{v}}}_{eb}^b = -(\boldsymbol{\Omega}_{ie}^b + \boldsymbol{\Omega}_{ib}^b)\hat{\mathbf{v}}_{eb}^b + \hat{\mathbf{C}}_n^b \mathbf{g}_b^n + \sum_{i=1}^4 \hat{\mathbf{f}}_{m,i}^b - k_f |\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb}^b. \quad (6.61)$$

Diese Differentialgleichungen enthalten den stabilisierenden Term $-k_f |\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb}^b$, der aus dem Strömungswiderstand resultiert. Dieser Term kann das Fehlerwachstum im Vergleich zum unbeschränkten Anwachsen beim INS wesentlich verringern. Er ist auch der Ursprung für die im fusionierten Navigationssystem später beschriebenen Eigenschaften, nämlich dass auch die Schätzung der IMU-Biase und sogar Roll- und Nickwinkel stabil wird. Wäre $k_f = 0$, dann hätten die Modelle A und B die gleiche Form. Eine Fusion würde der Fusion zweier Inertialer Navigationssysteme mit verschiedenen IMUs entsprechen. Bei einer solchen würde keine Stabilität ins System gebracht, lediglich das Rauschen könnte reduziert werden.

Systemrauschen Für das translatorische Modell werden die Modellierungsfehler und unberücksichtigte Einflüsse als additives, weißes gaußsches Rauschen modelliert: $\hat{\mathbf{f}}_{ib}^b = \mathbf{f}_{ib}^b + \mathbf{w}_{f_m}$. Eine solche Modellierung ist möglich, da die Korrelationszeiten dieser Fehler kurz sind.

Der Rauscheingangsvektor des gesamten Modells enthält zusätzlich noch das in Gleichung (6.55) eingeführte Rauschen der Drehratensensoren:

$$\mathbf{w}_{\text{VDM}} = \begin{pmatrix} \mathbf{w}_{f_m} \\ \mathbf{w}_{\omega} \\ \mathbf{w}_{b_{\omega}} \end{pmatrix} \quad (6.62)$$

6.3.2 Fusion der Modelle

Die im vorherigen Abschnitt vorgestellten Prädiktionsmodelle schätzen die Zustände

$$\mathbf{x}_{\text{INS}} = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^{n/b} \\ \mathbf{q}_b^n \\ \mathbf{b}_a \\ \mathbf{b}_\omega \end{pmatrix} \quad (6.63)$$

und

$$\mathbf{x}_{\text{VDM}} = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^{n/b} \\ \mathbf{q}_b^n \\ \mathbf{b}_\omega \end{pmatrix}. \quad (6.64)$$

Hierbei drückt „ n/b “ in $\mathbf{v}_{eb}^{n/b}$ aus, dass die Geschwindigkeiten im n -frame oder b -frame enthalten sein können. VDM steht für Vehicle Dynamics Model, also das Bewegungsmodell.

Betrachtet man die Überschneidungen der beiden Zustandsvektoren, stellt man fest, dass nicht nur die Navigationsgrößen \mathbf{p} , \mathbf{v}_{eb} und \mathbf{q}_b^n in beiden Modellen enthalten sind. Auch der Drehratensensorbias \mathbf{b}_ω ist eine gemeinsame Größe, also Teil der „common states“ \mathbf{x}_c . Für eine einfachere Darstellung werden daher die Zustandsgrößen in \mathbf{x}_{INS} neu sortiert:

$$\mathbf{x}_{\text{INS}} = \begin{pmatrix} \left. \begin{matrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^{n/b} \\ \mathbf{q}_b^n \\ \mathbf{b}_\omega \end{matrix} \right\} \mathbf{x}_{c,\text{INS}} \\ \left. \begin{matrix} \mathbf{b}_a \end{matrix} \right\} \mathbf{x}_{s,\text{INS}} \end{pmatrix} \quad (6.65)$$

Der Zustandsvektor des Bewegungsmodells enthält keine spezifischen Größen. Die Drehrate des Fluggerätes muss nicht geschätzt werden, weil kein rotatorisches Modell genutzt wird. In später vorgestellten Erweiterungen werden zwar weitere Zustände ergänzt. Vorerst soll aber mit diesem minimalen System gearbeitet und gezeigt werden, dass dieses für manche Anwendungen bereits ausreichend sein kann. Hier gilt also: $\mathbf{x}_{\text{VDM}} = \mathbf{x}_{c,\text{VDM}}$ und $\mathbf{x}_{s,\text{VDM}} = ()$.

Der kombinierte Vektor des Systemeingangsrauschens ist somit

$$\mathbf{w}_U = \begin{pmatrix} \mathbf{w}_{f_i} \\ \mathbf{w}_\omega \\ \mathbf{w}_{b_a} \\ \mathbf{w}_{b_\omega} \\ \mathbf{w}_{f_m} \end{pmatrix}, \quad (6.66)$$

mit den in den Gleichungen (6.55) und (6.62) eingeführten Größen.

Für den so definierten Rauschvektor \mathbf{w}_U kann die Matrix $\mathbf{Q} = E(\mathbf{w}_U \mathbf{w}_U^T)$ bestimmt werden.

Die Rauscheingangsmatrizen \mathbf{G}_{INS} und \mathbf{G}_{VDM} müssen entsprechend den Vektoren \mathbf{x}_{INS} , \mathbf{x}_{VDM} und \mathbf{w}_U aufgestellt werden. Für die Systeme mit n-frame-Geschwindigkeitsfehlern ergibt sich

$$\mathbf{G}_{\text{INS},n} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \hat{\mathbf{C}}_b^n & 0 & 0 & 0 & 0 \\ 0 & \hat{\mathbf{C}}_b^n & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 \end{pmatrix} \quad (6.67)$$

und

$$\mathbf{G}_{\text{VDM},n} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{\mathbf{C}}_b^n \\ 0 & \hat{\mathbf{C}}_b^n & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \end{pmatrix}. \quad (6.68)$$

Gemäß Abschnitt 6.2.4 müssen Nullzeilen in

$$\mathbf{G}_{H,k} = \mathbf{G}_{c,\text{INS}} - \mathbf{G}_{c,\text{VDM}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \hat{\mathbf{C}}_b^n & 0 & 0 & 0 & -\hat{\mathbf{C}}_b^n \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.69)$$

aus den Matrizen \mathbf{G}_H sowie Φ_H und $\delta\tilde{\mathbf{y}}$ gestrichen werden. Im hier vorliegenden Fall unterscheidet sich das Rauschen der Modelle nur im Einfluss auf die Geschwindigkeitszustände (zweite Blockzeile). Dies war zu erwarten, da sich die Modelle nur in der erwarteten Specific Force unterscheiden, die nur in die

Geschwindigkeitsdifferentialgleichung einfließt. Somit muss auch nur die Differenz der Geschwindigkeitspropagation betrachtet werden. Als positiver Nebeneffekt des Verkleinerns von \mathbf{G}_H ergibt sich, dass bei Berechnung der Matrix \mathbf{U}_k gemäß Gleichung (6.36) nur eine 3×3 -Matrix invertiert werden muss, was im Vergleich zur Inversion einer 12×12 -Matrix (für die Positions-, Geschwindigkeits-, Lagewinkel- und Biasfehler) erheblich weniger Rechenaufwand mit sich bringt.

Wenn die Geschwindigkeitsfehler im b-frame geschätzt werden, ergeben sich gemäß Anhang B die folgenden Matrizen:

$$\mathbf{G}_{\text{INS},b} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & [\hat{\mathbf{v}}_{eb}^b \times] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{C}}_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (6.70)$$

$$\mathbf{G}_{\text{VDM},b} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\hat{\mathbf{v}}_{eb}^b \times] & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \hat{\mathbf{C}}_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{pmatrix} \quad (6.71)$$

$$\mathbf{G}_{H,k} = \mathbf{G}_{c,\text{INS}} - \mathbf{G}_{c,\text{VDM}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad (6.72)$$

wobei $[\hat{\mathbf{v}}_{eb}^b \times]$ die Kreuzproduktmatrix zu $\hat{\mathbf{v}}_{eb}^b$ darstellt. Auch in dieser Formulierung treten die gleichen Nullzeilen wie in Matrix (6.69) auf, so dass dieselbe Verkleinerung durchgeführt wird.

Die in Abschnitt 6.2.4 dargestellten Gleichungen und der zugehörige Algorithmus für die Nutzung des Unified Models können nach dem Aufstellen dieser Matrizen ohne weitere Anpassung genutzt werden, um das Navigationsfilter mittels des fusionierten Systemmodells zu propagieren.

6.3.3 Theoretische Analyse

Durch die Fusion der beiden Teilmodelle zum Unified Model entsteht ein neues System mit völlig neuen Eigenschaften: Im Folgenden sollen diese Eigenschaften

aufgezeigt werden, insbesondere die neu eingeführte Stabilität für einige wichtige Fehlergrößen.

6.3.3.1 Eigenwerte und Stabilitätseigenschaften des Systems

Um die Systemmatrix des klassischen INS einer theoretischen Analyse zu unterziehen, kann diese für die Quadrocopter-Navigation wesentlich vereinfacht werden: Da hier nur geringe Geschwindigkeiten vorliegen, kann die Transportrate zu Null approximiert werden; auch Coriolis-Terme und Erddrehrate können aufgrund der geringen Sensorgüte und kurzer Flugdauern vernachlässigt werden [87]. Dadurch werden die Eigenwerte der Systemmatrix, die ohne die Vereinfachungen die Schulerschwingung und die Erddrehung beschreiben würden, ebenfalls zu null. Vernachlässigt man weiter den Einfluss des Höhenfehlers auf die Berechnung der Gravitation, werden auch die verbleibenden Eigenwerte der reduzierten INS-Systemmatrix zu null. Diese Vereinfachung dient der Übersichtlichkeit der folgenden Betrachtungen.

In der zeitdiskreten Formulierung weist die vereinfachte Systemmatrix $\Phi_{\text{INS,red}}$ somit nur Eigenwerte mit dem Wert 1 auf. Dies bedeutet, dass Einzelfehler konstant bleiben. Durch Integration solcher Fehler sowie aufgrund des Systemrauschens wachsen weitere Fehlergrößen aber mit der Zeit unbeschränkt an.

Betrachtet man hingegen die Systemmatrix des translatorischen Bewegungsmodells aus Abschnitt 6.3.1.2, lassen sich drei stabile Eigenwerte mit $|\lambda_i| < 1$ erkennen. Diese resultieren aus dem Strömungswiderstand, der die Geschwindigkeitsschätzung stabilisiert: Falls die Geschwindigkeit zu hoch geschätzt ist, wird auch der erwartete Strömungswiderstand höher als der tatsächliche. Dadurch reduziert sich der Geschwindigkeitsfehler. Alle anderen Eigenwerte liegen, wie im INS, bei $|\lambda_i| = 1$: Weder die Positions-, Lage- noch die Drehratensensorbiasfehler sind stabil.

Berechnet man nun mit den Gleichungen des Unified-Model-Ansatzes die Systemmatrix des fusionierten Systems, können auch deren Eigenwerte bestimmt werden. Für diese Berechnung der neuen Systemmatrix wird eine Kovarianzmatrix \mathbf{P} benötigt. Hierfür bietet sich die Kovarianzmatrix im eingeschwungenen Zustand aus einer Simulation oder einem Testflug an. Das Ergebnis zeigt, dass durch Fusion eines Systems ohne stabile Eigenwerte und einem System mit drei stabilen Eigenwerten ein neues System entsteht, das mehr als drei stabile Eigenwerte aufweist! Die genaue Anzahl an stabilen Eigenwerten ist abhängig von der verwendeten Kovarianzmatrix

\mathbf{P} sowie dem Systemzustand, bewegt sich aber zwischen sechs und acht stabilen Eigenwerten.

Dieses Ergebnis scheint überraschend, lässt sich aber erklären, wenn man das insgesamt vorhandene Wissen betrachtet: Wenn ein fehlerhafter Roll- oder Nickwinkel angenommen würde, würde dies gemäß den Modellgleichungen zu einer horizontalen Beschleunigung führen. Wird diese von den Beschleunigungsmessern aber nicht gemessen, kann der Lagefehler korrigiert werden. Dadurch entsteht eine Stabilität im Roll- und Nickwinkelfehler. Weitere stabile Eigenwerte resultieren, weil mit Hilfe der Modellgleichungen die Beschleunigungsmesserbiase stabilisiert werden können. Es sei jedoch darauf hingewiesen, dass solche Informationen nicht explizit im Navigationssystem implementiert werden müssen; stattdessen werden diese durch die optimale Modellfusion, die im Unified Model enthalten ist, automatisch ausgenutzt. Dies ist ein wesentlicher Unterschied zum Ansatz von Leishman [56] und auch Vasconcelos [83], wo explizit die vorliegenden Systemgleichungen analysiert werden mussten und für den jeweils speziellen Fall passend fusioniert wurden. Das Unified Model dagegen kann ebenso auf andere Probleme angewendet werden.

Bei einer konstanten Systemmatrix Φ_{\cup} würden die sechs bis acht stabilen Eigenwerte nicht ausreichen, um Geschwindigkeiten, Roll- und Nickwinkel sowie die IMU-Biase stabil zu schätzen. Betrachtet man aber das Produkt mehrerer zeitlich aufeinanderfolgender Systemmatrizen $\prod_{i=k}^{k+n} \Phi_{\cup i}$ während eines Flugs, lässt sich erkennen, dass aufgrund von Variationen in Lage und Geschwindigkeit die Anzahl an Eigenwerten weiter zunimmt. Die Eigenvektoren zeigen, dass weiterhin drei stabile Eigenwerte für die Geschwindigkeitspropagation vorliegen. Die anderen stabilen Eigenwerte lassen sich keinen einzelnen Zuständen zuordnen, so dass eine weitere Analyse mit Hilfe der Kovarianzbetrachtung erfolgt.

6.3.3.2 Vorteil der Formulierung „Geschwindigkeitsfehler im b-frame“

Ein Ergebnis der Kovarianzbetrachtung soll hier vorweg betrachtet werden: Wählt man den Zustandsvektor so, dass die Geschwindigkeitsfehler wie üblich in n-frame-Koordinaten geschätzt werden, resultiert ein überoptimistisches System: Die Unsicherheit der Heading-Winkel-Schätzung nimmt mit der Zeit immer mehr ab (s. Abbildung 6.4), was aber leider nicht dem tatsächlichen Fehler entspricht. In der Simulation und in realen Testflügen zeigt sich, dass dieser zunimmt oder im besten Falle konstant bleibt. Auch gibt es keine mögliche Begründung, warum die absolute Heading-Winkel-Schätzung mit der Zeit besser werden sollte: Weder INS

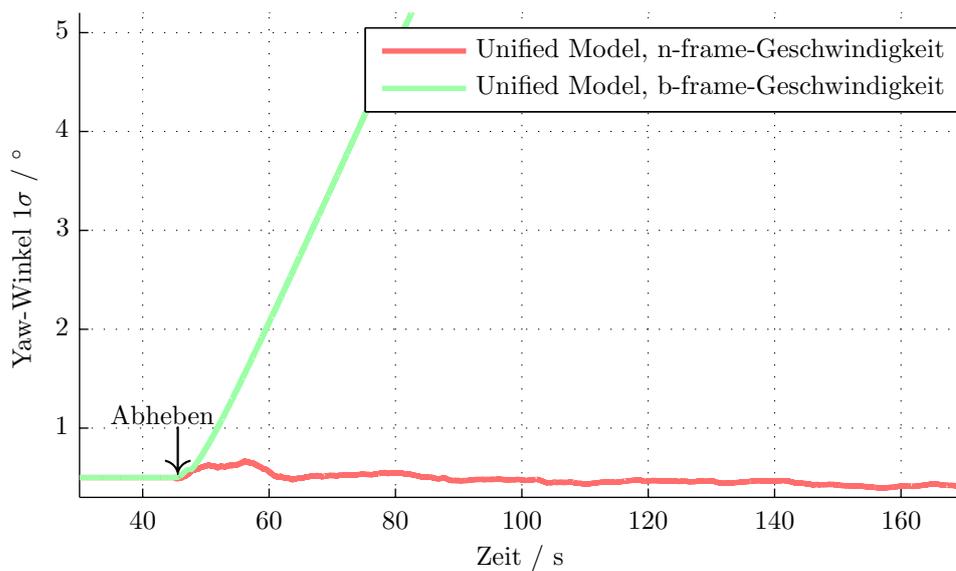


Abbildung 6.4: Schätzunsicherheit (1σ -Grenze) des Filters bzgl. des Yaw-Winkel-Fehlers: Die beiden Kurven unterscheiden sich darin, dass die Geschwindigkeitsfehler einmal in Navigationskoordinaten und einmal in körperfesten Koordinaten geschätzt wurden. Bei Verwendung der üblichen Navigationskoordinaten wächst die Unsicherheit fälschlicherweise nicht an. Wählt man stattdessen körperfeste Koordinaten, wächst die Yaw-Winkel-Unsicherheit an. Dies resultiert aus der Bias-Unsicherheit der Gyroskope

noch Bewegungsmodell enthalten Wissen über die absolute, d. h. auf die Nordrichtung bezogene Ausrichtung des Fluggerätes.

Dieses Problem wird aber nicht durch die Fusionsmethode des Unified Model verursacht, es ist vielmehr von allgemeiner Natur und kann auch in klassischen GPS/INS-Systemen auftreten: Wenn ein Navigationssystem still am Boden steht, messen die Beschleunigungsmesser aufgrund von Rauschen dennoch leichte Beschleunigungen. Dadurch entsteht eine vom Filter erwartete, aber dennoch nicht vorhandene Beobachtbarkeit des Heading-Winkels. Nur bei real vorliegenden Beschleunigungen wäre diese Beobachtbarkeit tatsächlich gegeben. Der hier im modellgestützten Navigationssystem auftretende Effekt ist ähnlich, aber nicht auf Stillstandsphasen beschränkt. Er ist zudem auch in Navigationssystemen für andere Flug- und Fahrzeugtypen zu erwarten, wenn dort ebenfalls eine Propagation von Geschwindigkeiten im n-frame vorgenommen wird und eine Fusion mit dem Unified Model oder dem Ansatz von Koifman und Bar-Itzhack durchgeführt wird. Daher ist eine genauere Betrachtung sinnvoll.

Die Geschwindigkeitspropagation in n-frame-Koordinaten lautet gemäß Gleichung (6.46):

$$\dot{\mathbf{v}}_{eb}^n = -(\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n)\mathbf{v}_{eb}^n + \mathbf{g}_b^n + \mathbf{C}_b^n \mathbf{f}_{ib}^b$$

Es ist zu erkennen, dass der Heading-Winkel im Term $\mathbf{C}_b^n \mathbf{f}_{ib}^b$ enthalten ist. Wenn sich die vom INS und Bewegungsmodell angenommenen horizontalen Beschleunigungen \mathbf{f}_{ib}^b unterscheiden (was aufgrund von Rauschen praktisch immer der Fall ist), erscheint der Heading-Winkel nach der Modellfusion beobachtbar. Es handelt sich aber nur um einen Linearisierungsfehler: Würden beide Modelle die um die wahren Werte linearisierte Systemmatrix liefern, würde der Effekt nicht auftreten.

Da die wahren Werte aber nicht vorliegen, muss ein anderer Weg gewählt werden, dieses ungewünschte Verhalten zu beseitigen. Das ist möglich, wenn man die Informationsfusion in b-frame-Koordinaten durchführt. Eine solche Wahl ist naheliegend, weil beide Teilmodelle die Specific Force im b-frame bestimmen.

Daher wurde die bereits in Gleichung (6.50) dargestellte Propagation gewählt:

$$\dot{\mathbf{v}}_{eb}^b = -(\boldsymbol{\Omega}_{ie}^b + \boldsymbol{\Omega}_{ib}^b)\mathbf{v}_{eb}^b + \mathbf{C}_n^b \mathbf{g}_b^n + \mathbf{f}_{ib}^b \quad (6.73)$$

In dieser Gleichung hat der Heading-Winkel keinen Einfluss: Er fließt zwar in die ersten beiden Spalten von \mathbf{C}_n^b ein, wird aber bei der Multiplikation mit \mathbf{g}_b^n eliminiert, da \mathbf{g}_b^n nur eine z-Komponente enthält.

Abbildung 6.4 zeigt, dass bei dieser Formulierung die Unsicherheit des Heading-Winkels korrekterweise ansteigt. Die sonstigen Filterkovarianzen und Schätzwerte sind zwischen beiden Formulierungen sehr ähnlich. Daher sollte die Formulierung mit b-frame-Geschwindigkeitsfehlern bevorzugt werden. Im Folgenden wird die Analyse daher auf das System in dieser Formulierung begrenzt.

6.3.3.3 Kovarianzbetrachtung

Um die Modellfusion näher zu beleuchten, werden die Filterkovarianzen der Einzelmodelle und des fusionierten Systems für einen Testflug dargestellt (Abbildungen 6.5, 6.6 und 6.7). Es ist zu erkennen, dass alle Kovarianzen des INS unbeschränkt anwachsen und beim reinen Bewegungsmodell nur die Unsicherheit der Geschwindigkeitsfehler beschränkt bleibt. Dahingegen weisen beim fusionierten Navigationssystem alle Zustandsgrößen außer der Position und dem Heading-Winkel

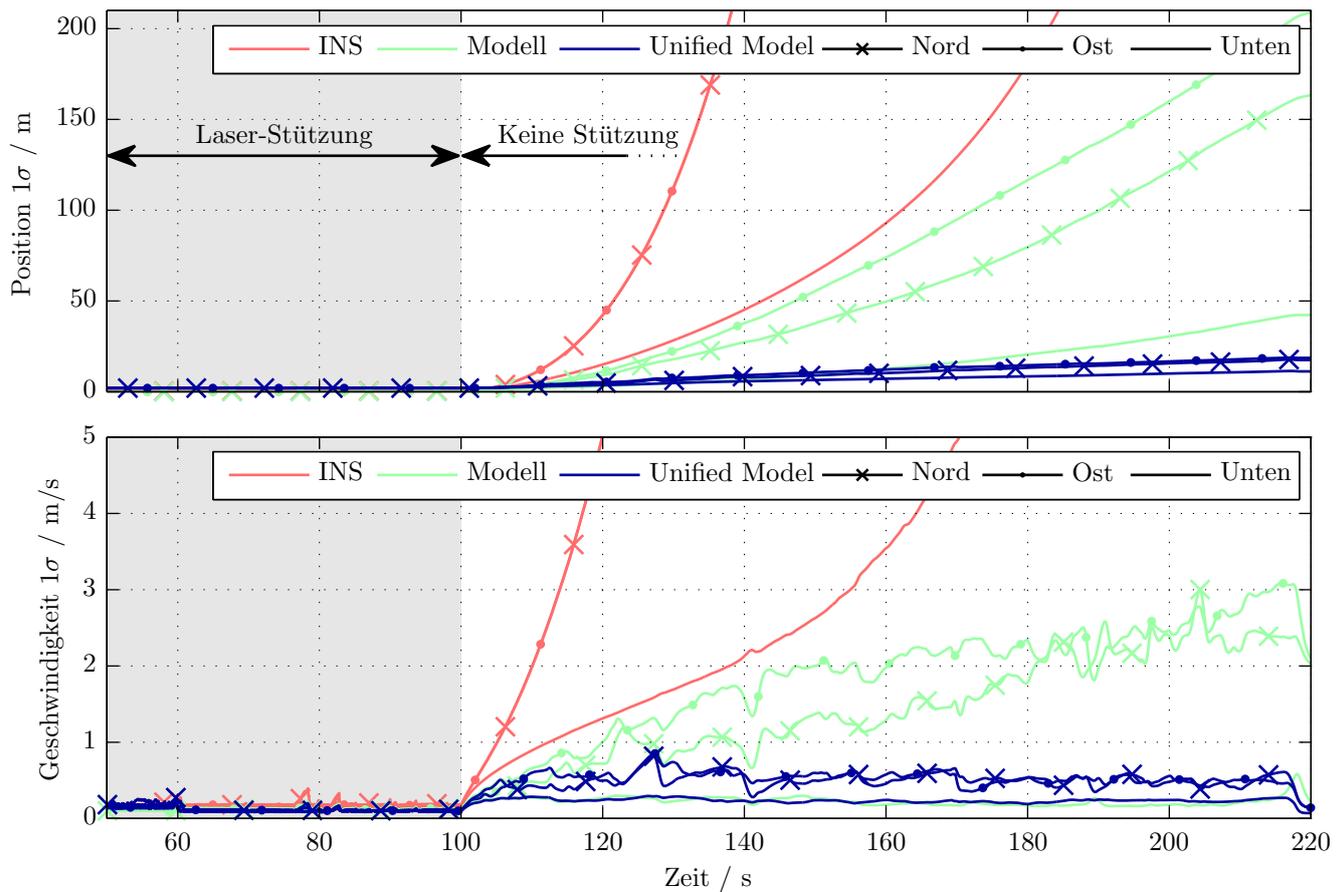


Abbildung 6.5: Schätzunsicherheit des Filters bzgl. Position und Geschwindigkeit (1σ -Grenze, anhand der Lageschätzung in Navigationskoordinaten transformiert): Die Geschwindigkeitsschätzung bei Modell und Unified Model ist stabil, wobei die Unsicherheit beim Unified Model deutlich kleiner ausfällt. Entsprechend wächst dort auch die Positionsunsicherheit deutlich langsamer.

beschränkte Fehlerkovarianzen auf, was die oben diskutierten Stabilitätseigenschaften bestätigt.

Die hier gezeigten Kovarianzen stellen aber natürlich nur die Unsicherheit des Filters dar; diese entsprechen nicht zwangsläufig den tatsächlichen Fehlern. Daher soll in einer experimentellen Untersuchung gezeigt werden, dass das Navigationssystem konsistent ist und die Fehlerverläufe den Kovarianzverläufen entsprechen.

6.3.4 Experimentelle Analyse

Bei modellgestützten Navigationssystemen ist eine Analyse von realen Flugdaten einer rein simulativen Untersuchung vorzuziehen, da einige Effekte wie z. B. Verwirbelungen nur mit hohem Aufwand simuliert werden können. Um dennoch statistisch aussagekräftige Schlüsse ziehen zu können, wird im Folgenden eine aus mehreren Flügen bestehende Versuchsreihe präsentiert.

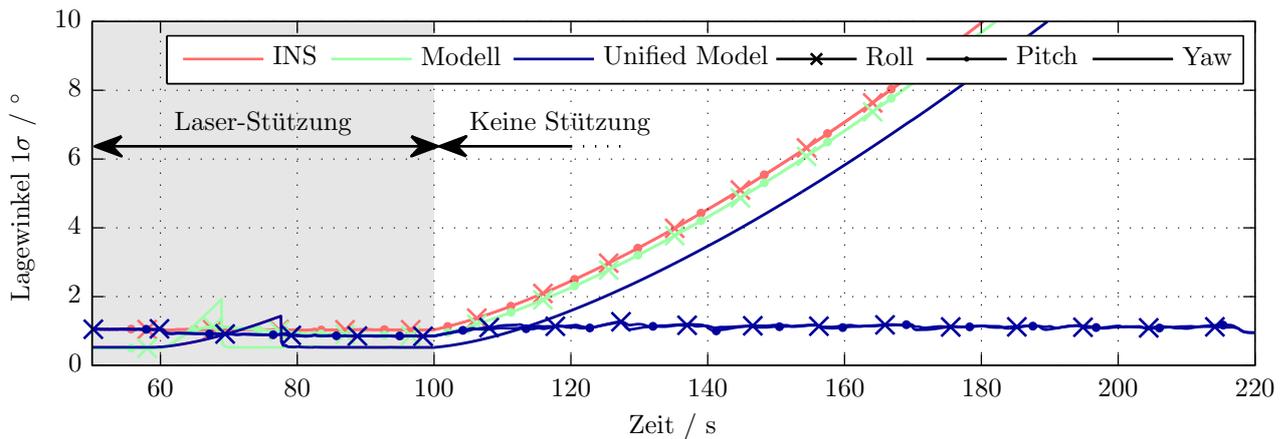


Abbildung 6.6: Schätzunsicherheit des Filters bzgl. der Lagewinkelfehler (1σ -Grenze): Das Unified Model weist begrenzte Kovarianzen für Roll- und Nickwinkel auf. Die Kovarianz des Yaw-Winkels ist beim Unified Model identisch zur INS-Yaw-Kovarianz (wenn beim Unified Model die Geschwindigkeitsfehler im b-frame geschätzt werden). Die Modellgleichungen sind erst ab dem Abheben gültig, daher ist eine andere Initialisierung nötig. Diese führt auch zu anderen Unsicherheiten der Lage, obwohl die gleiche Lagepropagation verwendet wird.

Verglichen werden drei verschiedene Navigationssysteme: Das klassische Navigationssystem ist ein INS mit Laserstützung. Das Modell-Filter nutzt das in Abschnitt 6.3.1.2 dargestellte translatorische Bewegungsmodell, erweitert um das Gyroskop-basierte rotatorische Modell. Schließlich wird das neu entworfene modellgestützte Filter, das Unified-Model-System, betrachtet, das die beiden einzelnen Modelle fusioniert.

Die Navigationssysteme verwenden zu Beginn der Flüge die Laser-Stützung gemäß Abschnitt 5.3. Später im Flug wird diese Informationsquelle deaktiviert, um das ungestützte Verhalten der Modelle vergleichen zu können. Als Referenznavigationssystem dient das permanent gestützte Laser/INS-Navigationssystem in Kombination mit RTS-Smoothing.

Die Parameter für das Bewegungsmodell wurden mit der in Abschnitt 3.4.1 beschriebenen Identifikation bestimmt.

6.3.4.1 Praktische Voraussetzungen

Inertialsensoren sind auf einem Quadrokopter starken Vibrationen ausgesetzt, da die verwendeten Motoren oft aus dem Modellbau stammen und nicht sehr ruhig laufen. Diese Vibrationen stellen für sich genommen bei klassischen Navigationssystemen kein Problem dar, solange die Inertialsensorik die Körperbewegung der Vibration messen kann. Bei Einsatz des Bewegungsmodell-gestützten Navigationssystems muss dagegen berücksichtigt werden, dass die Vibrationen nicht modelliert sind.

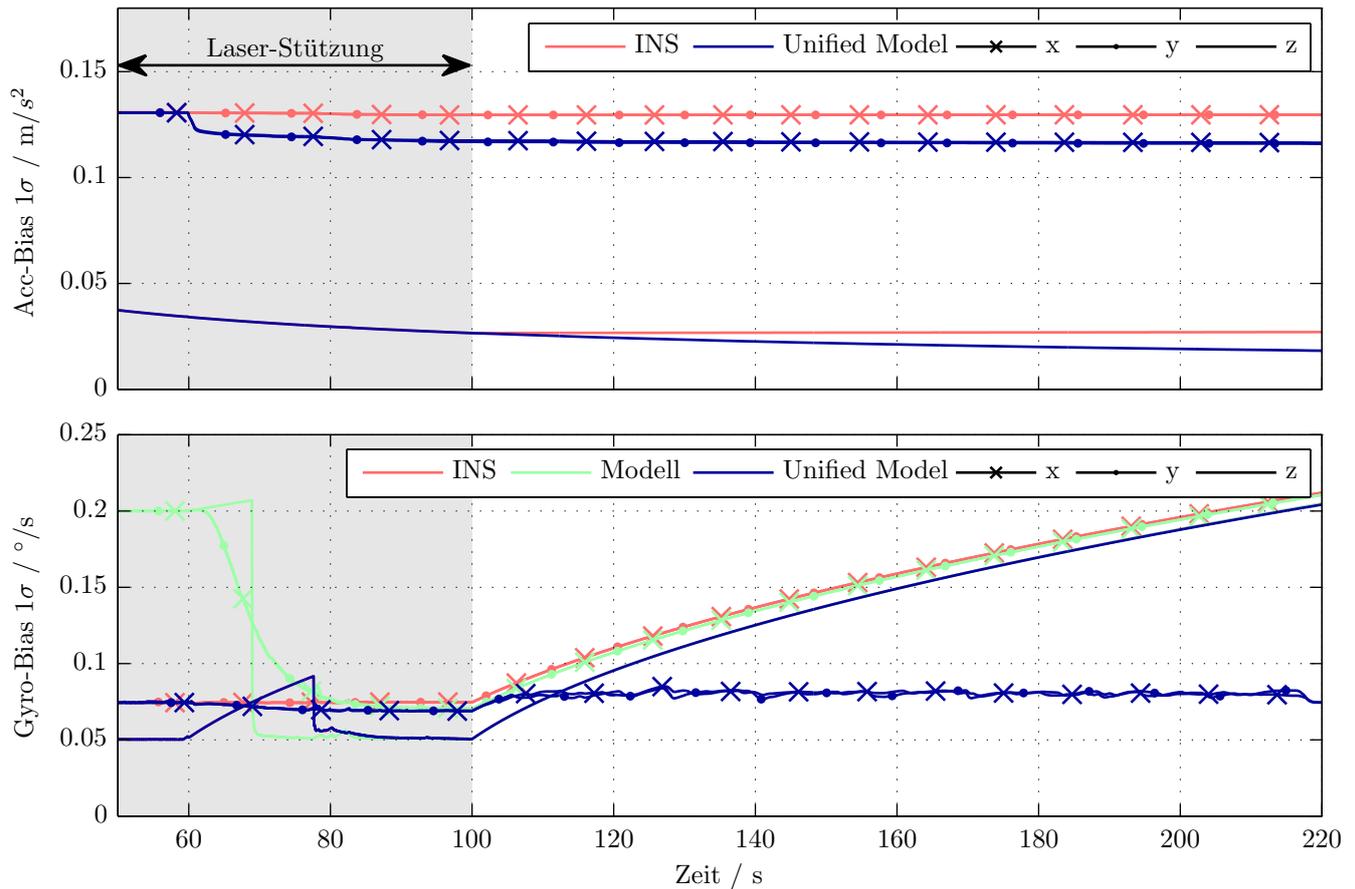


Abbildung 6.7: Schätzunsicherheit des Filters bzgl. der IMU-Bias (1 σ -Grenze): Die Gyroskop-Bias-Kovarianzen verhalten sich ähnlich wie die Lagewinkel-Kovarianzen: Beim Unified Model sind diese für Roll- und Nickwinkel beschränkt. Beim Yaw-Winkel ist die Unsicherheit des Unified Model (bei b-frame Geschwindigkeitsfehlern) identisch zur Yaw-Bias-Kovarianz des INS. Das Bewegungsmodell schätzt keine Beschleunigungsmesser-Bias. Dennoch entsteht auch bei dieser Schätzung im Unified Model zusätzliche Beobachtbarkeit: Während die Beschleunigungsmesser-Bias-Unsicherheit beim INS geringfügig wächst (bei der Darstellungsskala kaum zu erkennen), sinkt die Unsicherheit beim Unified Model etwas.

Da die Vibrationen aber Beschleunigungen im Bereich von mehreren m/s mit sich bringen, können sie deutliche Unterschiede zwischen gemessener und vom Modell vorhergesagter Beschleunigung verursachen. Dieser Unterschied zwischen Messung und Prädiktion ist jedoch proportional zur Innovation im Navigationssystem. Eine Verfälschung mit einem großen, zeitkorrelierten Fehler kann somit verheerende Auswirkungen auf die Schätzung haben. Um diese zu vermeiden sollte eine Reihe von Maßnahmen getroffen werden:

- Ideal ist die Beseitigung der Vibrationen selbst. Dies kann geschehen, indem kleine Massen (z. B. Klebestreifen) am Läufer des Motors angebracht werden, um den Motor auszuwuchten. Die Position und Größe dieser Massen kann iterativ bestimmt werden, wenn nach jeder Änderung die Vibrationsamplitude bestimmt wird. Sind die Motoren ausgewuchtet, sollten die Propeller balanciert werden, d. h. der Schwerpunkt sollte in der Achse liegen. Bei der Befestigung der Propeller im Aufnehmer stehen i. d. R. mehrere Positionen zur Auswahl. Auch hier sollte mittels Messungen eine Minimierung der Vibrationen angestrebt werden.
- Die verbleibenden Vibrationen am Rahmen des Fluggerätes können von der IMU entkoppelt werden, indem diese dämpfend gelagert wird, z. B. mittels O-Ringen.
- Trotz der oben genannten Maßnahmen sind oft noch merkliche Vibrationen im Messsignal vorhanden. Deren Frequenz entspricht der Motordrehzahl (ca. 70 Hz) und ihren Oberschwingungen. Somit können diese Vibrationen mittels eines Tiefpassfilters unterdrückt werden.

Die Bewegungsmodellgleichungen beschreiben die Bewegung des Quadropter-Schwerpunkts. Liegt die IMU nicht im Schwerpunkt, werden bei Rotationen und Rotationsänderungen zusätzliche Beschleunigungen gemessen. Diese müssen ebenfalls modelliert werden, auch wenn der Hebelarm nur kurz ist.

6.3.4.2 Bewertung der Navigationsergebnisse

Solange Laser-Stützmessungen genutzt werden, zeigen die untersuchten Navigationssysteme in Abbildung 6.8 nur geringe Unterschiede. Auffällig ist nur eine schlechtere Lagewinkelschätzung beim reinen Bewegungsmodell; diese ist darauf zurückzuführen, dass dessen Gleichungen erst nach Abheben gültig sind und somit im Stand keine Kalibrierung durchgeführt werden kann.

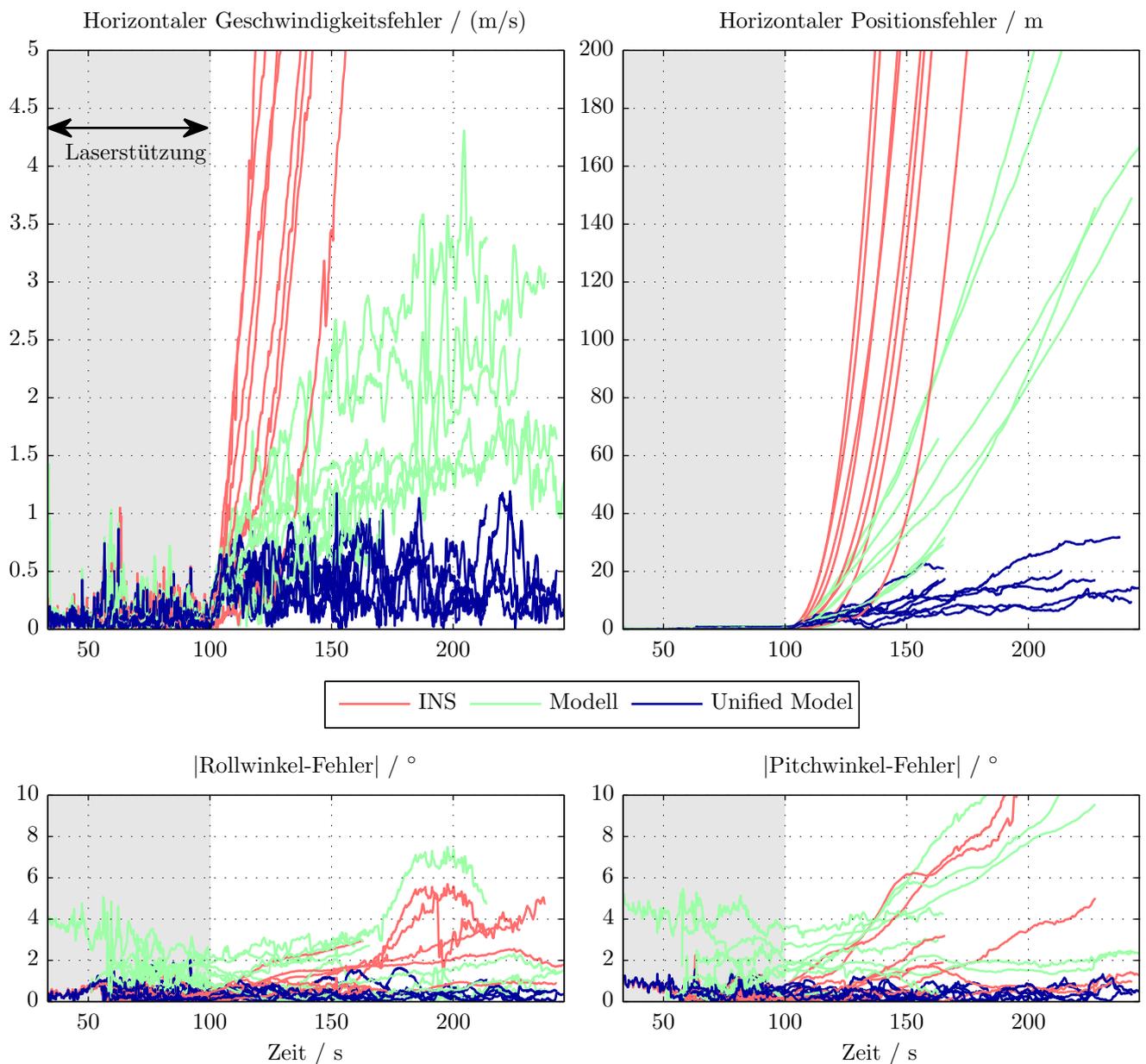


Abbildung 6.8: Verläufe der Navigationsfehler für acht Testflüge in einer Indoor-Umgebung. Die Fehler des klassischen Navigationssystems auf INS-Basis (rot) wachsen ohne Laserstützung sehr schnell an, das System wird nach kurzer Zeit unbrauchbar. Demgegenüber sind die Fehler von Roll- und Nickwinkel sowie der Geschwindigkeitsschätzung im neuen modellgestützten System beschränkt, der Positionsfehler wächst dementsprechend deutlich langsamer.

Um die Unterschiede der Navigationssysteme aufzuzeigen, wird ab 100 s die Laserstützung deaktiviert und das Verhalten ohne Stützung betrachtet:

Wie zu erwarten, wird die Navigationsschätzung des inertialen Navigationssystems („INS“) sehr schnell unbrauchbar, der Positionsfehler wächst innerhalb kurzer Zeit auf über 100 m an; auch die anderen Fehler wachsen hier unbegrenzt an. Demgegenüber

steht ein völlig anderes Verhalten bei Verwendung des Unified Models: Roll- und Nickwinkelfehler bleiben auch ohne Stützung klein, selbst der Geschwindigkeitsfehler ist ohne Stützung noch klein. Somit kann auch das Anwachsen des Positionsfehlers deutlich reduziert werden. Das Unified-Model-System stellt also eine Option dar, um längere Stützensensorausfälle zu überbrücken.

Der zur Parameterschätzung genutzte Flug ist nicht in Abbildung 6.8 enthalten. Die dargestellten Fehler wurden als Differenz aus Schätzung des jeweiligen Navigationssystems mit dem Referenznavigationssystem bestimmt. Da die Flüge unterschiedliche Längen haben und unterschiedliche Trajektorien beschreiben, wurde darauf verzichtet, einen gemittelten Fehler darzustellen. Die hier gezeigte Analyse betrachtet insbesondere die horizontalen Fehlergrößen bzw. Roll- und Nickwinkel. Die verbleibenden Größen, also die vertikale Bewegung und der Gierwinkel, können nämlich mittels Barometern und Magnetometern auch bei Ausfall anderer Stützensensorik weiter beobachtet werden.

Bewertung der Lageschätzung Sowohl INS als auch das Bewegungsmodell nutzen die Gyroskope zur Lagepropagation. Aufgrund von Fehlern in der Biasschätzung, der Biasinstabilität und dem Messrauschen (engl. Angular Random Walk, ARW) entsteht so ein mit der Zeit anwachsender, in Abbildung 6.8 unten deutlich zu erkennender Fehler. Obwohl diese beiden Systeme, die die Lage identisch propagieren, zum Unified Model fusioniert werden, ergibt sich ein System mit konstant geringem Lagefehler.

Ein beschränkter Lagefehler kann zwar auch erreicht werden, indem in einem reinen Lageschätzfilter die Beschleunigungsmesser als Stützsensoren genutzt werden. Hierbei wird aber die Annahme getroffen, dass sich das Fluggerät im unbeschleunigten Zustand befindet. Auch wenn diese Annahme oft gerechtfertigt ist, da der Arbeitspunkt der unbeschleunigte Schwebeflug ist, muss beachtet werden, dass insbesondere in kritischen Flugphasen mit großer Beschleunigung die Lageschätzung leiden kann. Auch ist dabei keine Bias-Schätzung möglich. Daher ist die Schätzung mittels Unified Model diesem klassischen Ansatz vorzuziehen.

Bewertung der Bias-Schätzung Der wahre IMU-Bias liegt in realen Flugversuchen nicht vor. In Abbildung 6.9 wird daher die Gyro-Bias-Schätzung des lasergestützten INS mit dem modellgestützten Navigationssystem verglichen. Bei diesem wird die Laserstützung wie auch in Abbildung 6.8 nach 100 s deaktiviert. Ab diesem Zeitpunkt dient nur noch die Modellstützung zur Biasschätzung. Das Ergebnis zeigt, dass die Bias-Schätzung äußerst ähnlich ist, selbst wenn im Unified-Model-System

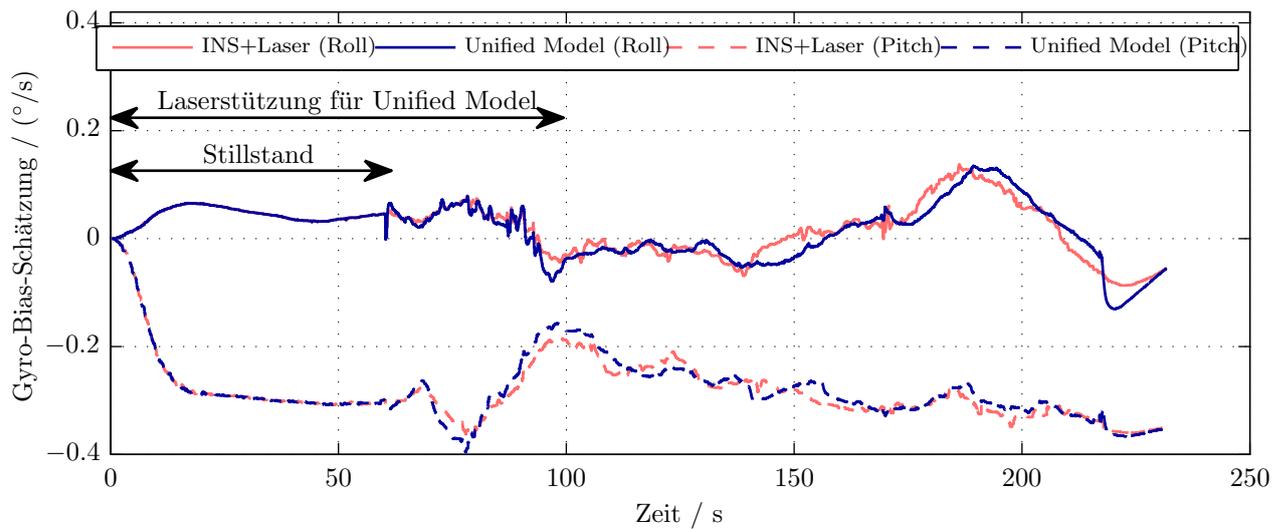


Abbildung 6.9: Vergleich der Gyro-Bias-Schätzung zwischen permanent lasergestütztem INS und Unified Model mit Phase ohne Laserstuetzung. Die Biasschätzung bleibt sehr ähnlich.

keine Laserstuetzung mehr vorliegt. Es kann also gefolgert werden, dass die Gyro-Bias-Schätzung im modellgestuetzten Navigationssystem nicht auf Lasermessungen angewiesen ist.

Die Schätzung der Beschleunigungsmesser-Biase zeigt deutlich geringere Ähnlichkeit. Hierfür können Fehler in den Modellparametern verantwortlich sein. Allerdings zeigen die Ergebnisse aus Abbildung 6.8, dass diese Biasschätzung offensichtlich nur einen kleinen Einfluss auf die Güte der Navigationslösung hat.

Bewertung der Positionsschätzung Für den praktischen Einsatz in einem autonomen Fluggerät kommt der Positionsschätzung eine wichtige Rolle zu. Während ein inertiales Navigationssystem mit kostengünstigen MEMS-Sensoren nur wenige Sekunden zur Positionsschätzung eingesetzt werden kann, zeigt Abbildung 6.8, dass das modellgestuetzte Navigationssystem auch nach ein oder gar zwei Minuten noch Fehler im Bereich von nur 10 m–20 m liefert. Das neu entworfene Navigationssystem kann also gerade bei Ausfall von Stuetzsensoren weiterarbeiten, wo herkömmliche Navigationssysteme ausfallen.

6.3.4.3 Parametersensitivität

Neben der erzielten Güte der Navigationsschätzung ist von Interesse, wie stark diese von der zuvor durchgeführten Parameterschätzung (vgl. Abschnitt 3.4.1) abhängt. Daher wurde eine Sensitivitätsanalyse durchgeführt, bei der die Parameter des Bewegungsmodells verfälscht wurden.

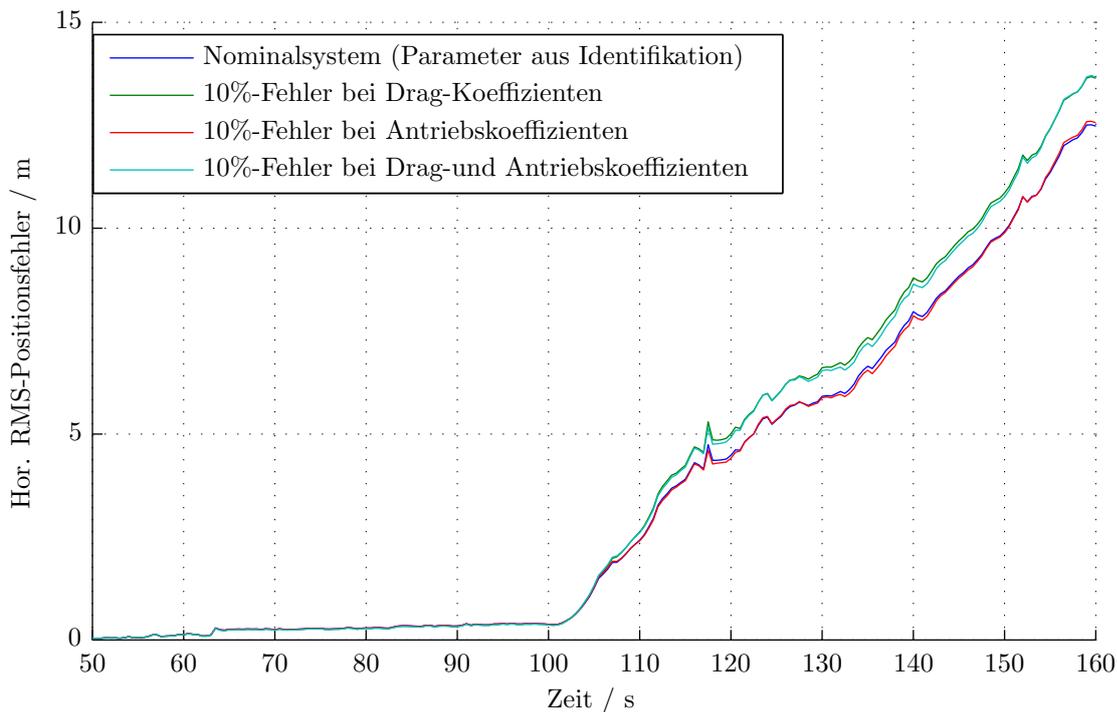


Abbildung 6.10: Sensitivität des Unified-Model-Navigationssystems bzgl. der Modellparameter: Selbst bei deutlichen Parameterfehlern kommt es nur zu einem mäßigen Anstieg des Positionsfehlers. Laserstützung ab 100 s deaktiviert.

Um eine übersichtliche Darstellung zu erhalten, wurden zunächst die RMS-Positionsfehler aus den einzelnen Testflügen in Abbildung 6.8 ermittelt. Dies bildet die Ergebnisse des Nominalsystems in Abbildung 6.10. Zum Vergleich sind die äquivalent gewonnenen Ergebnisse mit verfälschten Parametern aufgetragen. Die Fehlerverläufe werden bis 160 s gezeigt; in diesem Zeitraum liegen Daten aller verwendeten Testflüge vor.

Analysiert man die gezeigten Ergebnisse, wird klar, dass die Drag-Koeffizienten deutlich stärkeren Einfluss haben als die Koeffizienten für die Motorkräfte. Dies erklärt sich dadurch, dass die Motorkräfte eher für die vertikale Bewegung relevant sind, während bei der horizontalen Bewegung die Luftreibung dominiert. Sind alle Parameter um 10 % verfälscht, verschlechtert sich auch das Positionsergebnis um etwa 10 %. Gegenüber den Ergebnissen des INS in Abbildung 6.8 ist diese Genauigkeitsreduktion nahezu vernachlässigbar. Das modellgestützte Navigationssystem ist also robust gegenüber Ungenauigkeiten der Parameteridentifikation.

6.3.5 Fazit

Das entwickelte Navigationssystem zeigt die Vorteile des Unified-Model-Ansatzes: Der Entwicklungsprozess ist einfach, da nur die Teilmodelle entwickelt werden müssen; die Fusion zum Unified-Model beschränkt sich auf die Implementierung weniger Gleichungen. Die Flexibilität dieses Ansatzes wurde auch daran deutlich, dass problemlos auf das rotatorische Modell verzichtet werden und die Schätzung der Geschwindigkeitsfehler in einem beliebigen Koordinatensystem implementiert werden konnte. Das entstandene Navigationssystem ist minimalistisch in dem Sinne, dass der Zustandsvektor im Vergleich zum klassischen INS nicht vergrößert werden musste: es werden die gleichen Zustände geschätzt. Außerdem erlaubt die Beschränkung auf das translatorische Modell eine einfache Implementierung und Parameteridentifikation.

Die mit dem präsentierten Navigationssystem erreichten Ergebnisse weisen nach, dass Modellstützung im Anwendungsfall der Quadrokoetter-Navigation eine wesentliche Reduktion des Fehlerwachstums bewirken kann. Dieser Gewinn wird insbesondere dann deutlich, wenn andere Stützinformationen ausfallen.

6.4 Modellgestützte Quadrokoetter-Navigation mit Schätzung der Modellparameter

Das im vorherigen Unterkapitel entwickelte Navigationssystem kann eine deutlich verbesserte Navigationslösung im Vergleich zu rein inertialer Navigation liefern. Sein Einsatz ist daher wünschenswert. Dem steht allerdings entgegen, dass viele Fluggeräte oft für jeden Einsatz modifiziert werden: So wird z. B. oft die Nutzlast angepasst, aber auch Motoren und Rotoren werden regelmäßig getauscht. Solche Änderungen führen auch zu einer Änderung der Modellparameter. Daher ist es nicht möglich, sofort das modellgestützte Navigationssystem einzusetzen – zuerst müssen die neuen Parameter mit Hilfe von Testflügen bestimmt werden. Dies ist bei einer spontan einsetzbaren Aufklärungsplattform aber nicht möglich, hier muss eine anwenderfreundlichere Lösung entwickelt werden.

Dazu wird in diesem Unterkapitel eine Erweiterung des zuvor entwickelten Navigationssystems vorgestellt, mit der die nötigen Modellparameter während der ersten Sekunden des eigentlichen Fluges geschätzt werden können. Dadurch wird die praktische Anwendbarkeit der modellgestützten Navigation weiter erhöht. Es

wird zudem gezeigt, wie die Navigationslösung mit dieser Erweiterung noch weiter verbessert werden kann.

6.4.1 Methoden der Online-Parameterschätzung

Es soll nun untersucht werden, welche der Möglichkeiten zur Online-Parameterschätzung im Falle der modellgestützten Quadropternavigation am geeignetsten ist. Dazu wird zum einen eine externe Parameterschätzung, die außerhalb des Navigationssystems läuft, entwickelt. Diese ist angelehnt an die Offline-Parameterschätzung aus Abschnitt 3.4.1. Bei dieser Lösung kann die Unsicherheit der Parameter und deren Einfluss auf die Unsicherheit der Modellgleichungen nicht abgebildet werden. Daher wird zum anderen auch eine Erweiterung des Zustandsvektors vorgenommen, um die Modellparameter als konstante Zustandsgrößen zu schätzen. Hierbei werden in der Filterkovarianz die Zusammenhänge der Unsicherheiten abgebildet.

Die folgenden Abschnitte beschreiben diese Verfahren detailliert. Anschließend werden die Verfahren anhand von Testflügen verglichen.

6.4.1.1 Externe Parameterschätzung

Für die in Kapitel 3.4.1 beschriebene Identifikation der Modellparameter müssen u. a. die Geschwindigkeit des Fluggerätes und die Beschleunigungsmesser-Biase bekannt sein. Während diese Größen bei der Offline-Parameteridentifikation mittels Smoothing einer lasergestützten Navigationslösung gewonnen werden können, muss im Flug auf die Filterlösung zurückgegriffen werden, weil Smoothing erst nach dem Flug ausgeführt werden kann. Da zu Beginn des Fluges die Modellparameter noch nicht bekannt sind, kommt als Navigationssystem, das diese Größen bestimmt, kein modellgestütztes Navigationssystem in Frage. Es muss daher auf ein klassisches INS mit Stützmessungen, wie z. B. Laser oder GNSS, zurückgegriffen werden.

Anstatt der Least-Squares-Schätzung kann mit den so vorliegenden Größen zu einer iterativen Least-Squares-Schätzung übergegangen werden. Dadurch fällt in jedem Zeitschritt nur eine geringe Rechenlast an. Eine solche iterative Least-Squares-Schätzung wird z. B. von einem Kalman-Filter mit konstantem Messrauschen realisiert. Eine Modifikation des Kalman-Filters hin zu einem sog. Constrained-Kalman-Filter (CKF) erlaubt zudem, dass auch während des Einschwingvorgangs nur physikalisch plausible Werte erreicht werden können. Dadurch wird die Robustheit

des Systems erhöht: Wenn die Stützmessungen früh ausfallen und die Parameter noch nicht gut geschätzt sind, können durch den Einsatz des CKF Instabilitäten im modellgestützten Navigationssystem verhindert werden, die z. B. durch negative Werte für k_f entstehen würden. Solche Werte wurden bei manchen Tests bei Nutzung gewöhnlicher Kalman-Filter tatsächlich beobachtet und sollten daher ausgeschlossen werden.

Wie auch bei der Offline-Parameteridentifikation werden die Parameter für die drei Achsen im b-frame getrennt geschätzt. So entstehen drei CKFs mit je fünf Zustandsgrößen:

$$\mathbf{x}_{x/y/z} = \begin{pmatrix} k_{f,x/y/z} \\ m_{1,x/y/z} \\ m_{2,x/y/z} \\ m_{3,x/y/z} \\ m_{4,x/y/z} \end{pmatrix} \quad (6.74)$$

Es werden also die Strömungswiderstandskoeffizienten $k_{f,x/y/z}$ und die Motorkoeffizienten $m_{i,x/y/z}$ geschätzt.

Die Beschränkung auf physikalisch plausible Werte kann mittels einer Modifikation der Kalman-Filter-Gleichungen umgesetzt werden: Nach Berechnung der gewöhnlichen Kalman-Gain-Matrix \mathbf{K} lässt sich ein hypothetischer, korrigierter Zustand \mathbf{x}_k^h berechnen:

$$\mathbf{x}_k^h = \mathbf{x}_k^- - \mathbf{K}z \quad (6.75)$$

$$z = (\mathbf{H}\mathbf{x}_k^- - \tilde{y}) \quad (6.76)$$

Hierfür wird basierend auf der Modellgleichung

$$\mathbf{f}_{ib}^b = \sum_{i=1}^4 \mathbf{f}_{m,i}^b - k_f |\mathbf{v}_{eb}^b| \mathbf{v}_{eb}^b \quad (6.77)$$

und unter Annahme von $\mathbf{f}_{ib}^b \approx \tilde{\mathbf{f}}_{ib}^b - \hat{\mathbf{b}}_a$ der Messwert

$$\tilde{y} = \tilde{f}_{ib,x/y/z}^b - \hat{b}_{a,x/y/z} \quad (6.78)$$

und die linearisierte Messmatrix

$$\mathbf{H} = \left(|\hat{\mathbf{v}}_{eb}^b| \hat{\mathbf{v}}_{eb,x/y/z}^b \quad \omega_{m,1}^2 \quad \omega_{m,2}^2 \quad \omega_{m,3}^2 \quad \omega_{m,4}^2 \right) \quad (6.79)$$

verwendet.

Wenn $x_{k,1}^h < 0$ gilt, also der hypothetische neue Strömungswiderstandskoeffizient negativ ist, wird die Hypothese als ungültig angesehen. Die Messung wird aber nicht vollständig verworfen. Stattdessen wird das Kalman-Gain \mathbf{K} reduziert, so dass der neue Wert für $x_{k,1}^h$ gerade zu null wird, d.h. auf der Grenze des zulässigen Bereichs liegt:

$$\beta = \mathbf{x}_{k,1}^- / (K_1 z) \quad (6.80)$$

$$\mathbf{K}_r = \beta \mathbf{K} \quad (6.81)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- - \mathbf{K}_r z \quad (6.82)$$

Da $0 \leq \beta < 1$, ist das reduzierte Gain \mathbf{K}_r kleiner als das zuvor berechnete \mathbf{K} . Mit dieser Messung wird sichergestellt, dass zumindest ein Teil der vorliegenden Information in die verbesserte Schätzung der Motorkoeffizienten einfließt.

Natürlich darf die Parameterschätzung nur ausgeführt werden, wenn das Fluggerät in der Luft ist, da sonst die Modellgleichungen, die in die Messgleichung einfließen, nicht gültig sind. Eine entsprechende Überprüfung kann anhand der Höhenschätzung oder auch der Motordrehzahlen geschehen. Auch sollte zumindest ein Schwellwert für die Geschwindigkeitsunsicherheit definiert sein, über dem keine Schätzung stattfindet, weil die Linearisierung in der \mathbf{H} -Matrix sonst zu schlecht sein könnte. Um die unmodellierten Vibrationen zu berücksichtigen, werden sowohl die Beschleunigungsmessersignale wie auch alle anderen Eingangsgrößen, also z. B. die Geschwindigkeitsschätzung, mit einem IIR-Filter (engl. infinite impulse response) tiefpassgefiltert.

Problematik der externen Schätzung Schätzt man, wie hier gezeigt, die Modellparameter außerhalb des Kalman-Filters, entstehen mehrere miteinander interagierende Kalman-Filter: Die Navigationslösung aus dem Navigations-Kalman-Filter hat Einfluss auf die Parameterschätzung; sobald die Parameter für die modellgestützte Navigation verwendet werden, haben auch die Constrained-Kalman-Filter zur Parameterschätzung Einfluss auf die Navigation. Diese Beziehungen und die Fortpflanzung der Fehler wird aber nicht abgebildet: Die Filter zur Parameterschätzung gehen

von fehlerfreien Geschwindigkeits- und Bias-Schätzungen aus; das Navigations-Filter nimmt an, dass die Parameter perfekt geschätzt sind. Kleine Parameterfehler können zwar über das Systemrauschen berücksichtigt werden, aber während der Online-Identifikation können auch größere Fehler auftreten. Insbesondere ist dann die Annahme des weißen Rauschens im Systemrauschen verletzt, weil durch die Parameterfehler stark zeitlich korrelierte Fehler entstehen: Bei Geradeausflug ist z. B. die geschätzte Kraft aufgrund des Strömungswiderstands permanent zu klein geschätzt, wenn der entsprechende Widerstandskoeffizient zu klein geschätzt ist.

Diese Eigenschaften begründen verschiedene problematische Effekte, die mit diesem Verfahren auftreten können. So wurde festgestellt, dass die besten Ergebnisse erzielt werden, wenn das Navigationsfilter möglichst lange ohne Modellstützung läuft und erst beim Ausfall der Stützensensorik auf das modellgestützte System umstellt.

6.4.1.2 Interne Parameterschätzung

Um die Nachteile der externen Online-Parameterschätzung zu lösen, soll alternativ eine Parameterschätzung ins Navigations-Filter integriert werden. Dazu wird der Zustandsvektor des Bewegungsmodell-basierten Teilmodells erweitert um die zu schätzenden Parameter und lautet dann

$$\mathbf{x}_{\text{VDM,Int}} = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^b \\ \mathbf{q}_b^n \\ \mathbf{b}_\omega \\ \mathbf{k}_f \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \\ \mathbf{m}_4 \end{pmatrix}, \quad (6.83)$$

wobei \mathbf{k}_f und \mathbf{m}_i jeweils 3×1 Vektoren sind, die die Modellparameter bezeichnen.

Die Parameter werden als konstant modelliert:

$$\dot{\mathbf{k}}_f = 0 \quad (6.84)$$

$$\dot{\mathbf{m}}_i = 0, \forall i \in [1..4] \quad (6.85)$$

Diese hinzugefügten Modellparameter beeinflussen die Navigationsgrößen, da sie in der Berechnung der Specific Force genutzt werden. Dieser Einfluss muss durch entsprechende Einträge in der Systemmatrix ausgedrückt werden, die sich durch Berechnung der Jacobi-Matrix $\partial \dot{v}_{eb}^b / \partial \left(\mathbf{k}_f \quad \mathbf{m}_1 \quad \mathbf{m}_2 \quad \mathbf{m}_3 \quad \mathbf{m}_4 \right)^T$ ermitteln lassen.

Dadurch entstehen im Propagationsschritt automatisch Korrelationen zwischen Parameterfehlern und Geschwindigkeitsfehlern. Sobald eine Stützinformation vorliegt, z. B. aus einer Laser- oder GPS-Messung, aber auch bei Fusion mit dem INS im Unified Model, wird nicht nur die Geschwindigkeitsschätzung korrigiert, sondern auch die Schätzung der mit der Geschwindigkeit korrelierten Parameter. Gleichzeitig führt bei der Propagation mit diesem Ansatz die Unsicherheit eines Parameters korrekterweise zu einer erhöhten Unsicherheit der Geschwindigkeitsschätzung. Dies verbessert die relative Gewichtung zwischen Bewegungsmodell und INS oder auch Stützsensormessungen. Daher kann erwartet werden, dass dieser Ansatz der externen Online-Parameterschätzung überlegen ist.

Im Gegensatz zum minimalen modellgestützten Navigationssystem aus Abschnitt 6.3 treten nun Zustandsgrößen auf, die spezifisch für das Bewegungsmodell sind. Diese werden als spezifische Zustände \mathbf{x}_{sB} im Unified-Model-Ansatz behandelt. Sie vergrößern den kombinierten Zustandsvektor $\mathbf{x}_{U,Int}$ zwar, aber die Modellfusion kann unverändert gemäß Abschnitt 6.2.4 durchgeführt werden.

6.4.1.3 Reduzierte Modelle

Der Zustandsvektor in Gleichung (6.83) enthält neben den 12 bisherigen Zustandsgrößen noch 15 zu schätzende Parameter. Dies erhöht die Rechenlast natürlich spürbar und ist unter Umständen nicht mehr echtzeitfähig auf Embedded-Plattformen realisierbar. Daher sollen hier reduzierte Modelle definiert werden, die mit einer geringeren Anzahl an Parametern auskommen.

Vertikale Motorkräfte Bei der Offline-Parameterschätzung konnten die gemessenen Beschleunigungen besser vom Modell vorhergesagt werden, wenn die Motorkoeffizienten als Vektoren eingeführt wurden, d. h. ein Motor kann eine Kraft in horizontaler Richtung erzeugen. Allerdings ist die Summe der horizontalen Motorkräfte meist sehr klein, so dass untersucht werden kann, ob auch auf diese Kräfte verzichtet werden

kann. Dadurch reduziert sich der Zustandsvektor auf

$$x_{\text{VDM,Int,MV}} = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^b \\ \mathbf{q}_b^n \\ \mathbf{b}_\omega \\ \mathbf{k}_f \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{pmatrix}, \quad (6.86)$$

d. h. es müssen nur noch 7 statt 15 Parameter geschätzt werden.

Horizontales Drag-Modell Wenn die Motoren nur vertikale Kräfte erzeugen, wird die horizontale Bewegung nur noch von den Kräften aus dem Strömungswiderstand bestimmt. Eine modellbasierte Stützung der horizontalen Bewegung ist aber deutlich wichtiger als eine Stützung der vertikalen Bewegung, weil letztere bereits zuverlässig durch Baro-Altimeter oder auf den Boden gerichtete Abstandssensoren gestützt werden kann. Dadurch kann das Modell noch weiter reduziert werden, indem auf die Modellierung der in z^b -Richtung wirkenden Kräfte verzichtet wird und stattdessen – ähnlich wie beim Verzicht auf das rotatorische Modell – auf die gemessene z -Beschleunigung zurückgegriffen wird. Dadurch reduziert sich der Zustandsvektor weiter auf

$$x_{\text{VDM,Int,Hor}} = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{v}_{eb}^b \\ \mathbf{q}_b^n \\ \mathbf{b}_\omega \\ k_{fx} \\ k_{fy} \end{pmatrix}. \quad (6.87)$$

Es sind also nur noch zwei Modellparameter zu schätzen. Gleichzeitig bringt diese Reduktion einen weiteren, wesentlichen Vorteil: Die Modellstützung kann damit sogar für Quadrokoetter implementiert werden, auf denen keine Motordrehzahlmessung verfügbar ist.

6.4.1.4 Beobachtbarkeit der Modellparameter

Die externe Parameterschätzung entspricht bis auf die zusätzlichen Nebenbedingungen der offline Least-Squares-Schätzung, für die in Abschnitt 3.4.1.3 die Beobachtbarkeitsbedingungen untersucht wurden. Es gilt also auch hier, dass das Ausführen der grundlegenden Manöver Rollen, Nicken, Schweben, Gieren und Flug mit konstanter Geschwindigkeit in body-x- und body-y-Richtung zu einer vollständigen Beobachtbarkeit aller Parameter führt.

Dieselben Überlegungen können für die interne Schätzung angestellt werden, wenn als Stützmessung eine GPS- oder Lasermessung angenommen wird. Auch hier sind die genannten Manöver ausreichend. Wenn das Bewegungsmodell auf das Horizontale Modell reduziert wird, entfallen sogar noch einige nötige Manöver: Hier ist es ausreichend, Geschwindigkeiten in x^b - und y^b -Richtung aufzubauen. Die restlichen Manöver dienen dem Schätzen der Motorparameter oder dem $k_{f,z}$ -Wert, was in diesem reduzierten Modell entfällt.

6.4.2 Experimentelle Ergebnisse

Für die Untersuchung der Methoden zur Online-Parameterschätzung wurde eine weitere Versuchsreihe durchgeführt. Hierbei wurden acht Flüge mit mindestens drei Minuten Länge durchgeführt. Die Lasermessungen werden wieder wie in Abschnitt 6.3.4 zur Stützung zu Beginn der Flüge genutzt; außerdem wird daraus die Referenz-Navigationslösung berechnet.

6.4.2.1 Vergleich der Methoden zur Parameterschätzung

In diesem Abschnitt soll ein Vergleich zwischen den verschiedenen Methoden zur Parameterschätzung vorgestellt werden. Die untersuchten Methoden zur Online-Schätzung der Parameter sind

- die externe Schätzung, bei der zusätzliche Kalman-Filter außerhalb des Navigationsfilters zur Schätzung genutzt werden (Abschnitt 6.4.1.1)
- die interne Schätzung, bei der der Zustandsvektor des Navigationssystems um die zu schätzenden Parameter erweitert wird (Abschnitt 6.4.1.2)
- die interne Schätzung mit teilweise reduziertem Zustandsvektor, so dass keine Motorkräfte in der Horizontalen geschätzt werden (Abschnitt 6.4.1.3 „Vertikale Motorkräfte“)

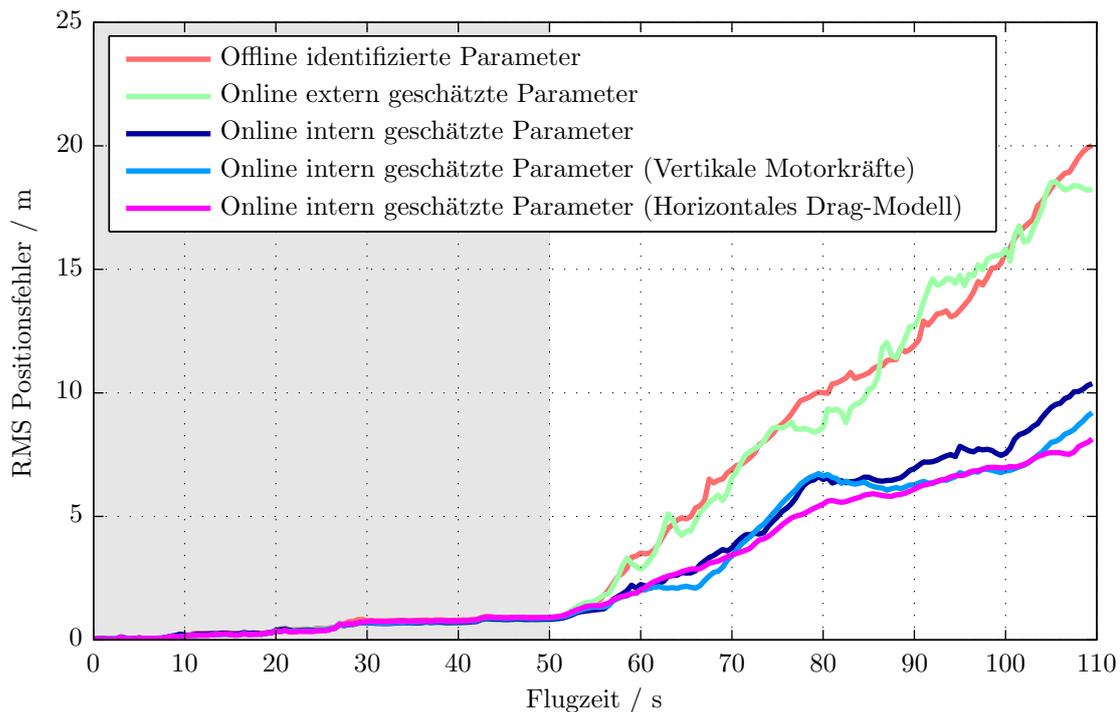


Abbildung 6.11: Horizontaler RMS-Positionsfehler (gemittelt über acht Testflüge) für verschiedene modellgestützte Navigationssysteme. Während der ersten 50 s Flugzeit sind alle Systeme mit Laser-Scanmatching-Ergebnissen gestützt, danach wird diese Stützung deaktiviert. Die interne Parameterschätzung liefert bessere Ergebnisse als die externe Schätzung und die Offline-Schätzung. Am besten verhält sich die Schätzung mit dem horizontalen Drag-Modell, bei dem sich der horizontale Positionsfehler nach 60 s ohne Laserstützung auf nur etwa 8 m beläuft.

- die interne Schätzung mit noch weiter reduziertem Zustandsvektor, so dass gar keine Motorkräfte geschätzt werden (Abschnitt 6.4.1.3 „Horizontales Drag-Modell“)

Zudem wird auch das Verfahren aus Abschnitt 6.3 in den Vergleich einbezogen, bei dem die Parameter offline, d. h. vor dem eigentlichen Flug geschätzt werden müssen. Auch beim Vergleich dieser Ansätze treten die Unterschiede erst dann deutlich zum Vorschein, wenn die Stützmessungen des Laser-Scanmatchings deaktiviert werden, also ein Ausfall simuliert wird. Ein solcher tritt auch in vielen realen Flügen auf, wenn z. B. keine Objekte im Erfassungsbereich des Laser-Sensors liegen. Es werden dann für jedes betrachtete Navigationssystem für jeden aufgenommenen Flug 60 s Flugdaten mit einem solchen Ausfall verarbeitet. Wann der Ausfall innerhalb des Fluges beginnt, ist frei wählbar. Abbildung 6.11 stellt die Ergebnisse dar, bei denen dieser ab 50 s Flugzeit auftritt. Dabei beschreibt jeder Graph den über alle acht Testflüge gemittelten RMS-Positionsfehler, um die Übersichtlichkeit zu erhöhen.

Die folgenden Schlüsse können aus dem Vergleich gezogen werden:

- Alle Verfahren der Online-Parameterschätzung liefern Ergebnisse, die mindestens so gut sind wie bei vorher durchgeführter Offline-Parameterschätzung.
- Die drei Verfahren mit interner Parameterschätzung weisen geringere Fehler auf als bei der externen Online- und Offline-Schätzung. Dies entspricht den Erwartungen, da bei der internen Schätzung Korrelationen zwischen Parameterfehlern und Navigations- sowie Bias-Fehlern aufgrund der zusätzlichen Zustände berücksichtigt werden können.
- Interessanterweise liefern die reduzierten Modelle noch bessere Ergebnisse als beim vollen Modell. Dies weist darauf hin, dass die Annahme, dass die Motorkräfte nur senkrecht wirken, gerechtfertigt ist und es mit den im vollen Modell zusätzlich enthaltenen Parametern eher zu einer Überanpassung kommt.

Da die reduzierten Modelle auch nur eine geringe Anzahl an Zuständen benötigen und somit effizienter sind, sind diese auf jeden Fall dem vollen Modell vorzuziehen. Die Verwendung des Horizontalen Drag-Modells ist für die praktische Anwendung besonders interessant, da hierfür keine Messung der Motordrehzahlen benötigt wird.

6.4.2.2 Dauer der Parameterschätzung

Die Modellparameter können in allen Ansätzen insbesondere dann gut geschätzt werden, wenn horizontale Laserstützung vorhanden ist. Im realen Flug kann diese aber aufgrund von Umgebungsbedingungen ausfallen. Es stellt sich daher die Frage, wie lange die Stützung notwendig ist, bevor auch ein Flug bei Laserausfall möglich ist.

Der Einfluss der Zeit bis zum Ausfall der Stützung wird wie folgt analysiert: Die Berechnungen, die für Abbildung 6.11 durchgeführt wurden, werden wiederholt für andere Zeitpunkte des horizontalen Laserausfalls. Die jeweils angefallenen Fehler nach 60 s ohne Laserstützung werden dann in Abbildung 6.12 als Funktion des Ausfallzeitpunkts aufgetragen.

Es lässt sich erkennen, dass bereits 30 s–40 s Flugzeit mit Stützung ausreichen, um die Parameter gut zu bestimmen, so dass Ergebnisse wie in Abbildung 6.11 erzielt werden können. Dieser Zeitraum reicht aus, um die Manöver durchzuführen, die zur Beobachtbarkeit der Parameter führen. Abbildung 6.12 bestätigt außerdem

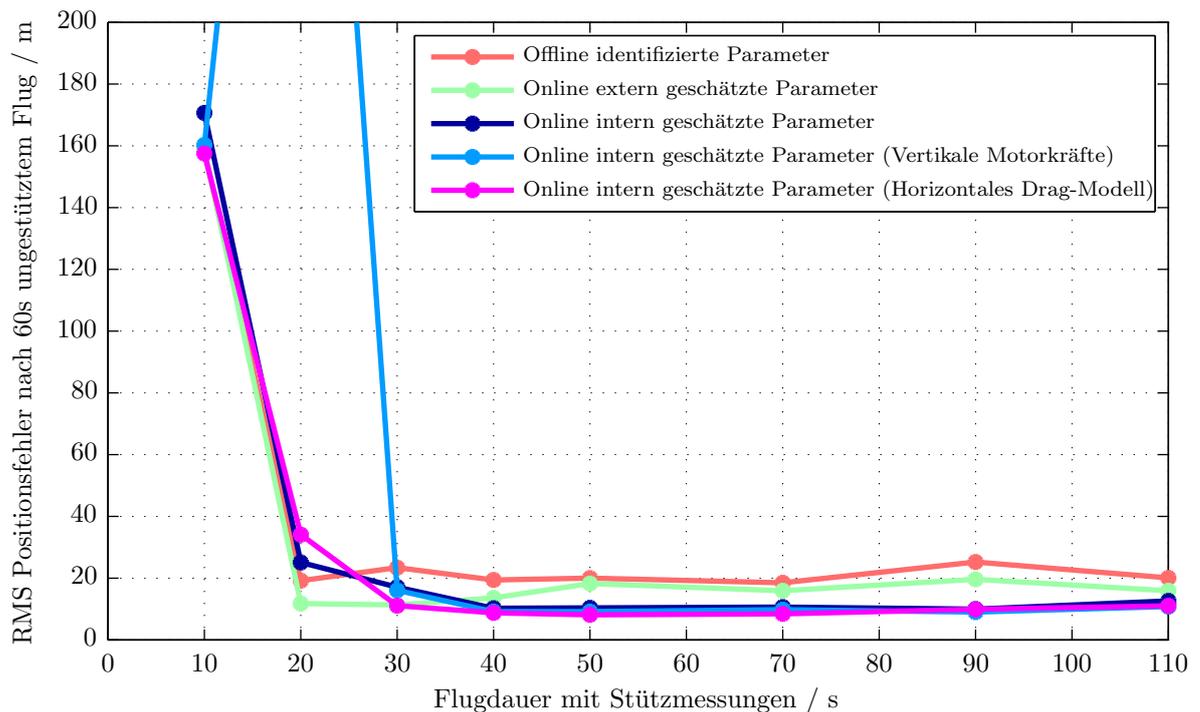


Abbildung 6.12: Horizontaler RMS-Positionsfehler nach 60 s Flug ohne horizontale Laserstützung aufgetragen über der zuvor für die Parameterschätzung verwendeten Flugdauer. Ab 30 s Parameterschätzung werden gute Ergebnisse erzielt.

das Ergebnis des vorherigen Abschnittes, dass das Navigationssystem mit dem Horizontalen Drag-Modell die besten Ergebnisse liefert.

6.4.2.3 Outdoor-Ergebnisse

Weitere drei Flüge wurden außerhalb von Gebäuden aufgenommen, um die Parameterschätzung anhand von GPS-Messungen bewerten zu können. Hierbei zeigte sich, dass etwas mehr Zeit für die Parameterschätzung nötig ist, in dem hier gezeigten Fall 60 s. Dies resultiert vermutlich daraus, dass GPS nur Messwerte in einer Frequenz von 4 Hz liefert, gegenüber 10 Hz beim Laser. Zudem ist die relative Genauigkeit der Lasermessungen höher.

Nach dieser Zeit für die Online-Parameterschätzung wurde die GPS-Stützung synthetisch deaktiviert, um erneut zu untersuchen, inwiefern ein Ausfall durch die Modellstützung kompensiert werden kann. Abbildung 6.13 zeigt die Ergebnisse für die Offline-Schätzung und die Online-Schätzung mit dem Horizontalen Drag-Modell. Eine Referenz wurde mit einem GPS/INS-Smoother erzeugt. Auch hier zeigt sich wieder, dass die Online-Parameterschätzung die Ergebnisse des Systems mit zuvor identifizierten Parametern weiter verbessert. Obwohl nur eine Low-Cost-IMU

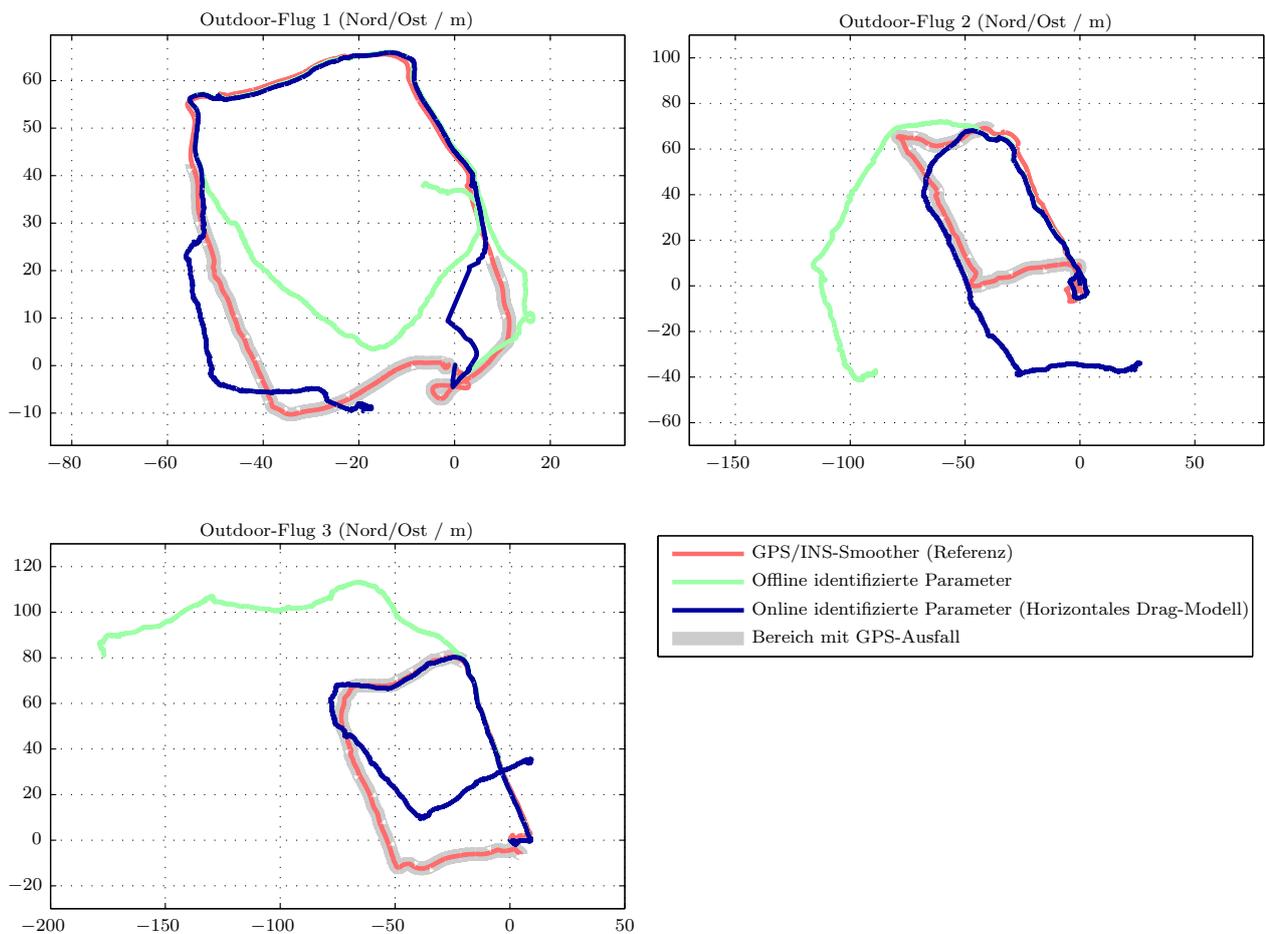


Abbildung 6.13: Ergebnisse des entwickelten modellgestützten Navigationssystems mit online geschätzten Parametern im Vergleich zum System mit offline geschätzten Parametern. Auch bei den Outdoor-Flügen zeigt das System mit Online-Parameterschätzung verbesserte Ergebnisse, die nun für die praktische Anwendung geeignet sind. Bei Flug 1 fielen kurz vor dem Start die GPS-Messungen aus, so dass die ersten 20 Sekunden Flug ohne Stützung erfolgen müssen. Abbildung bereits veröffentlicht in [109].

eingesetzt wird, lassen die geschätzten Trajektorien die wahren Trajektorien gut erkennen. Bei einem klassischen Navigationssystem beträgt der Positionsfehler nach dem GPS-Ausfall je nach Flug dagegen mehrere hundert oder gar tausend Meter.

6.4.2.4 Flug mit realem Laserausfall

In den bisher in diesem Kapitel gezeigten Ergebnissen wurden nur künstlich herbeigeführte Ausfälle der Stützensensorik betrachtet, da so die einzelnen Navigationssysteme gut verglichen werden konnten. Dass Sensorausfälle auch in realen Flügen auftreten, zeigt das Beispiel aus Abschnitt 5.3.4: Während des Fluges durch kleinere Räume und Flure steht zuverlässige Laserstützung zur Verfügung. Beim Weiterflug in einen

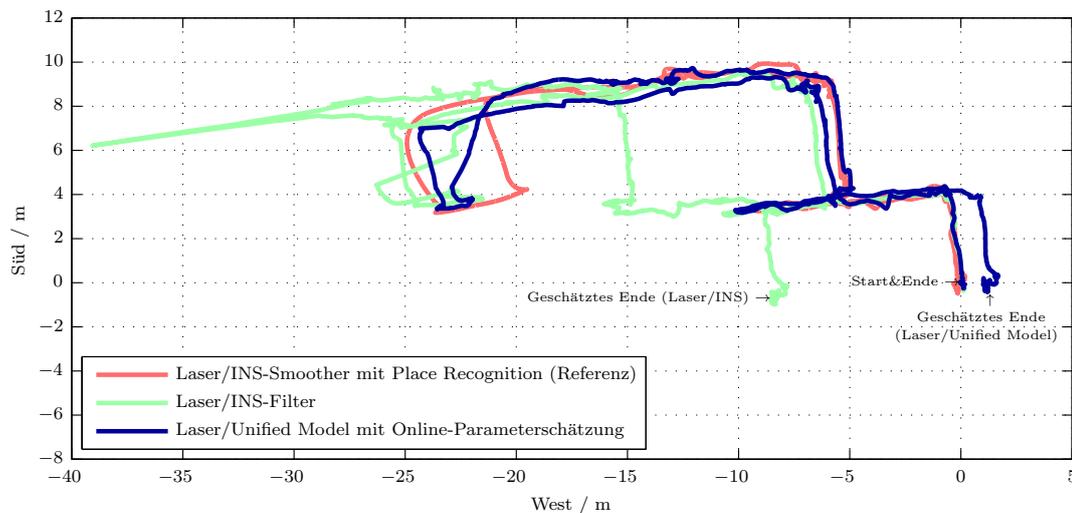


Abbildung 6.14: Indoor-Flug mit realem Sensorausfall: Während des Flugs in den Fluren werden vom modellgestützten System die Parameter geschätzt. Beim Eintritt in den Hörsaal (–25 m bis –20 m West) weist das Laser/INS-Filter ohne Modellstützung schnell große Fehler auf. Das modellgestützte Navigationssystem behebt dieses Problem. Bereits veröffentlicht in [109].

Hörsaal misst der Laserscanner aufgrund seiner auf 4 m begrenzten Reichweite jedoch schnell keine Objekte mehr, die Stützung fällt für einige Sekunden aus. In dieser Zeit wächst der Navigationsfehler des klassischen Laser/INS-Navigationssystems (Abschnitt 5.3.4) sehr schnell an. Erweitert man dieses Navigationssystem mit Hilfe des Unified Models um ein Bewegungsmodell, so lässt sich das Wegdriften wirksam begrenzen (vgl. Abbildung 6.14).

Zu Beginn des Flugs stand dem Navigationssystem keinerlei Information bzgl. der Modellparameter zur Verfügung; die Parameter wurden während des Fluges geschätzt. Im Vergleich zum Referenz-System aus Abbildung 6.14 kann das modellgestützte Navigationssystem die Navigationslösung bereits online während des Fluges liefern und läuft echtzeitfähig auf dem Fluggerät selbst, ist also nicht auf eine Funkverbindung zur Bodenstation angewiesen.

6.5 Kompensation des Windeinflusses

Im bisherigen Verlauf der Arbeit wurde davon ausgegangen, dass die das Fluggerät umgebende Luft sich nicht bewegt. Dann gilt $v_{eb}^b = v_{ab}^b$, d. h. die Geschwindigkeit des Fluggerätes gegenüber der Erde ist gleich der Geschwindigkeit gegenüber der Luft (Index a). In den gezeigten Indoor-Flügen ist diese Annahme sicherlich gerechtfertigt,

auch die Flüge im Außenbereich (Abbildung 6.13) fanden bei geringem Wind statt. Der Einsatz von Quadroptern ist allerdings nicht auf Situationen ohne Wind beschränkt; das Navigationssystem muss also auch bei Wind funktionsfähig sein. Dazu sind Erweiterungen gegenüber dem bisher vorgestellten System notwendig.

Um zu vermeiden, dass bei Wind der Positionsfehler sehr stark anwächst, muss die Windgeschwindigkeit geschätzt und der Windeinfluss kompensiert werden. Diese Kompensation erfolgt, indem Gleichung (6.56) ersetzt wird durch

$$\hat{\mathbf{f}}_{ib}^b = \sum_{i=1}^4 \hat{\mathbf{f}}_{m,i}^b - k_f |\hat{\mathbf{v}}_{ab}^b| \hat{\mathbf{v}}_{ab}^b. \quad (6.88)$$

Die erwartete Specific Force berechnet sich also anhand der geschätzten Anströmgeschwindigkeit $\hat{\mathbf{v}}_{ab}$, die gemäß $\hat{\mathbf{v}}_{ab} = \hat{\mathbf{v}}_{eb} - \hat{\mathbf{v}}_w$ als Differenz der Geschwindigkeit des Fluggerätes $\hat{\mathbf{v}}_{eb}$ und der Windgeschwindigkeit $\hat{\mathbf{v}}_w$ geschrieben werden kann.

6.5.1 Windschätzung im modellgestützten Navigationssystem

Die Schätzung der Windgeschwindigkeit sollte innerhalb des Navigationssystems erfolgen, da nur so die Unsicherheit dieser Schätzung korrekt zu einer Unsicherheit der Navigationsgrößen führen kann. Die Untersuchungen zur Schätzung der Modellparameter (Unterkapitel 6.4) zeigten, dass dies wichtig für eine genaue Navigation ist. Zudem kann dann die Windschätzung von allen verfügbaren Sensorinformationen profitieren.

Somit soll der Zustandsvektor des Navigationssystems um Größen zur Windschätzung erweitert werden. Dazu ist ein Propagationsmodell für die Windgeschwindigkeit in der Form $\dot{\mathbf{v}}_w = f(\mathbf{x}) + \mathbf{w}$ erforderlich. Ohne aufwändige Simulationen des gesamten Windfeldes, für die auch 3D-Modelle der Umgebung herangezogen werden müssten, kann aber keine präzise Vorhersage der Windgeschwindigkeit erfolgen. Auch das Dryden-Windmodell [1] ist als Spektralmodell nicht zur Prädiktion der Windgeschwindigkeit im Kalman-Filter geeignet.

Eine Möglichkeit besteht in der Modellierung $\dot{\mathbf{v}}_{w,s} = \mathbf{0} + \mathbf{w}_s$ mit ausreichend großem Rauschen \mathbf{w}_s , was einer Random-Walk-Modellierung entspricht. Diese führt jedoch zu einem raschen, unbegrenzten Anwachsen der Unsicherheit der Windgeschwindigkeit. Dadurch wird nicht ausreichend berücksichtigt, dass tatsächliche Windgeschwindigkeiten begrenzt sind. Um diese Tatsache auszudrücken,

kann eine Modellierung als Gauß-Markov-Prozess erster Ordnung vorteilhaft sein, wie sie z. B. auch in [35] genutzt wird. Die Differentialgleichung lautet dann $\dot{\mathbf{v}}_{w,a} = -1/\tau_a \mathbf{v}_{w,a} + \mathbf{w}_a$, wobei τ_a eine Zeitkonstante ist, die das Abklingen der Windgeschwindigkeit beschreibt. Problematisch an dieser Formulierung ist aber, dass eine konstante Windgeschwindigkeit damit nicht abgebildet werden kann. Folglich scheint es vorteilhaft, wie in [90] eine Überlagerung $\mathbf{v}_w = \mathbf{v}_{w,s} + \mathbf{v}_{w,a}$ als weitere Alternative in Betracht zu ziehen, bei der ein statischer Wind $\mathbf{v}_{w,s}$ mit geringem oder gar ohne Prozessrauschen \mathbf{w}_s mit einem abklingenden Anteil zur Beschreibung von Böen, $\mathbf{v}_{w,a}$, kombiniert wird.

Wind hat in Bodennähe kaum vertikale Anteile, was über entsprechende Parametrisierung der Down-Komponente des Prozessrauschens \mathbf{w}_s und \mathbf{w}_a ausgedrückt werden sollte.

Setzt man Gleichung (6.88) in die Geschwindigkeitsdifferentialgleichung (6.46) oder (6.50) ein, lässt sich die neue Jacobi-Matrix berechnen, die zur Systemmatrix führt. Je nach Art der Windmodellierung muss dabei der Zustandsvektor um drei Zustände (nur statisch oder nur abklingend) oder um sechs Zustände (Überlagerung statisch und abklingend) erweitert werden.

6.5.1.1 Vergleich von Windgeschwindigkeitsmodellen

Der Einfluss des im Kalman-Filter eingesetzten Windmodells wurde mittels der AirQuad-Simulationsumgebung untersucht. Basierend auf einer vorgegebenen Solltrajektorie wurden 100 Monte-Carlo-Wiederholungen durchgeführt. Für jede Wiederholung kamen unterschiedliches Sensorrauschen sowie unterschiedliche Ausschnitte aus aufgezeichneten Windmessungen zum Einsatz, so dass die realisierten Trajektorien zwar ähnlich, aber nicht identisch waren. In der Nachverarbeitung wurden dann die aufgezeichneten Sensordaten der 100 Simulationen mit verschiedenen Navigationsfilter-Varianten getestet. Diese basieren alle auf dem in den Unterkapiteln 6.3 und 6.4 entwickelten modellgestützten Navigationssystem, jeweils um eines der oben beschriebenen Windmodelle erweitert.

Abbildung 6.15 zeigt die Fehler der geschätzten Eigengeschwindigkeit des Fluggerätes (oben) und der geschätzten Windgeschwindigkeit (Mitte). Während die Fehler bei GPS-Stützung (bis 80 s) bei allen Windmodellen gering sind, wachsen die Fehler bei Ausfall der GPS-Stützung merklich an. Die Windmodelle „statischer Wind“ und „abklingender Wind“ weisen dabei deutlich höhere Fehler auf als die

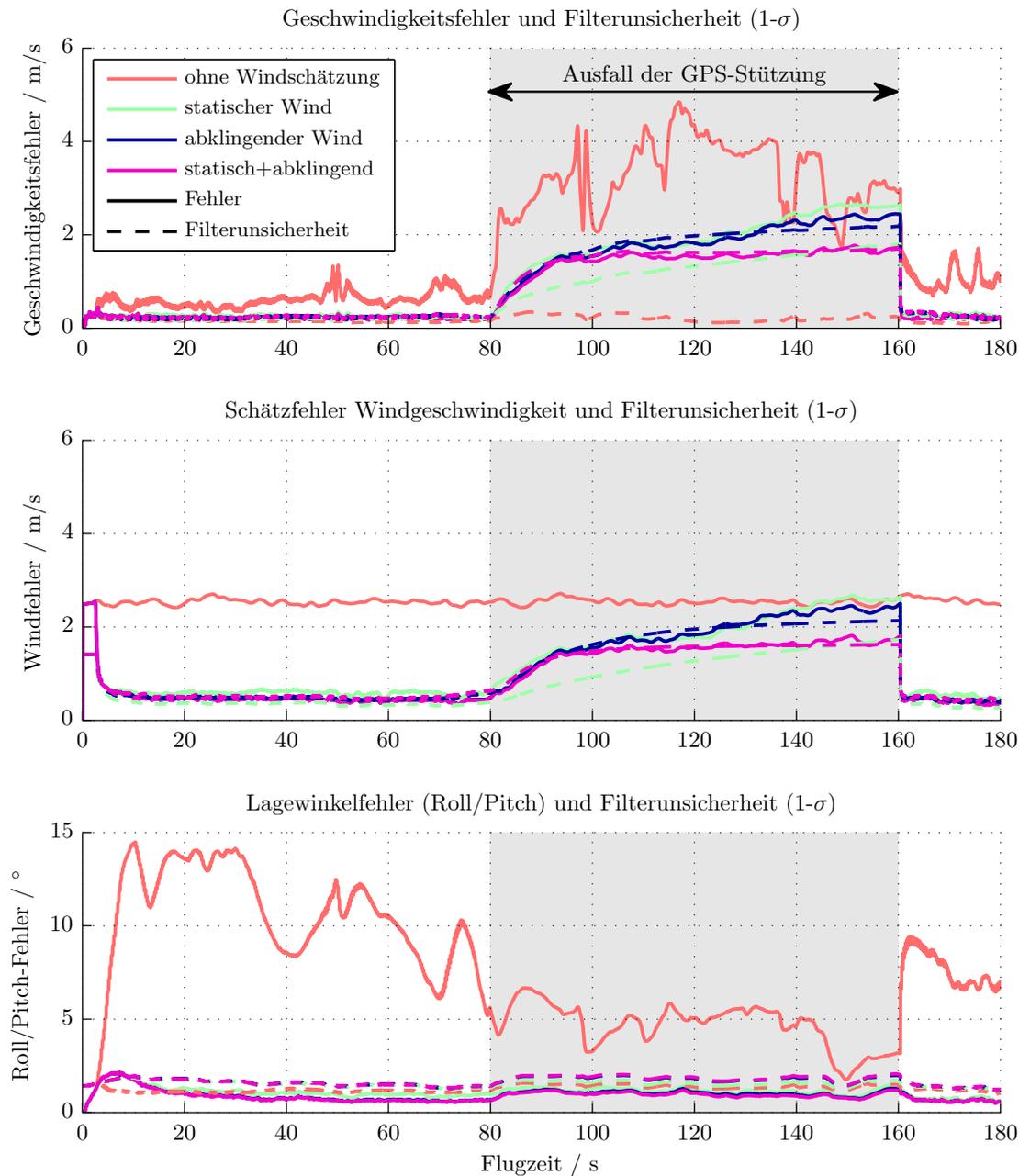


Abbildung 6.15: Simulationsergebnisse zum Vergleich verschiedener Windmodellierungen. Das Modell mit Überlagerung von statischem und abklingendem Anteil liefert die geringsten Fehler und die korrekteste Filterunsicherheit. Die Geschwindigkeitsfehler steigen ohne Stützung an, sind aber beschränkt; die Lagefehler bleiben auch ohne Stützung gering.

Überlagerung zu „statisch+abklingend“. Dementsprechend ist auch der Positionsfehler mit dem überlagerten Modell am geringsten. Für die folgenden Betrachtungen wird daher das Windmodell „statisch+abklingend“ genutzt.

6.5.2 Eigenschaften des Navigationssystems mit Windschätzung

Eine wesentliche Eigenschaft des modellgestützten Navigationssystems, wie es in Unterkapitel 6.3 entwickelt wurde, ist der beschränkte Lage- und Geschwindigkeitsfehler. In diesem Abschnitt soll betrachtet werden, inwiefern sich diese Eigenschaften bei Vorhandensein von Wind ändern.

Abbildung 6.15 oben zeigt, dass die Unsicherheit der Geschwindigkeitsschätzung sowie der zugehörige Fehler ohne Stützung mit der Zeit merklich ansteigen. Da die Unsicherheit des bisherigen Systems ohne Windschätzung nur geringfügig steigt, kann der Anstieg der Unsicherheit im erweiterten System wesentlich auf die Berücksichtigung des Windes zurückgeführt werden: Die Windgeschwindigkeit ist ohne GPS-Stützung nicht beobachtbar, das Wissen über die Windgeschwindigkeit nimmt somit nach dem Ausfallen der GPS-Stützung ab. Da im Bewegungsmodell nur die Geschwindigkeit relativ zur Luft beobachtbar ist, wächst in gleichem Maße auch die Unsicherheit der Eigenbewegung. Der Geschwindigkeitsfehler ist zwar weiterhin beschränkt, weil auch auftretende Windgeschwindigkeiten begrenzt sind, allerdings ist die Schranke damit höher als bei Flügen ohne Wind. Bei längeren Ausfällen der Stützensensorik muss also mit einem erhöhten Anwachsen der Schätzfehler gerechnet werden. Das Prozessrauschen des Windes sollte zumindest grob an die vorherrschenden Windverhältnisse angepasst werden. Je nach Einsatzgebiet könnte auch eine adaptive Schätzung realisiert werden.

In Abbildung 6.15 unten wird deutlich, dass auch bei Vorliegen von Wind selbst ohne GPS-Stützung der Lagefehler und die Lageunsicherheit weiterhin beschränkt sind; lediglich ein leichtes Anwachsen im Vergleich zu dem Zeitraum mit GPS-Stützung ist zu beobachten. Die Schätzung des für die Flugregelung wichtigen Lagewinkels wird also durch Wind nicht wesentlich beeinträchtigt.

6.5.3 Experimentelle Analyse

Um die Funktionsfähigkeit des Navigationssystems unter schwierigen Bedingungen nachzuweisen, wurden Flugtests bei böigem Wind durchgeführt. Während der Versuche wurden am Start- und Landeplatz Windgeschwindigkeiten von bis zu

7,5 m/s gemessen. Böen stellen nicht nur für den Flugregler sondern auch für das modellgestützte Navigationssystem eine deutlich größere Herausforderung dar als statischer Wind, da der statische Anteil mit GPS-Stützung geschätzt und kompensiert werden kann, auch nach einem möglichen GPS-Ausfall. Böen dagegen müssen auch während Ausfällen geschätzt werden. Die vorgefundenen Bedingungen sind also gut zur Bewertung der Algorithmen geeignet.

Im Gegensatz zu den Untersuchungen in Unterkapitel 6.4 wurde bei den hier gezeigten Versuchen davon ausgegangen, dass zumindest grundlegende Anfangsinformationen über die Modellparameter vorliegen. Diese können z. B. aus der Schätzung aus vorherigen Flügen stammen. Die zugehörige Anfangsunsicherheit der Modellparameter wurde auf $\pm 30\%$ (3σ -Bereich) der Anfangsschätzung angegeben. Damit bleiben die Parameter auch durch Tauschen von Rotorblättern oder durch Änderungen an der Nutzlast innerhalb des erwarteten Bereichs und können dann im Flug mit der weiterhin aktivierten Parameterschätzung auf den exakten Wert geschätzt werden.

6.5.3.1 Schätzung der Windgeschwindigkeit

Um die Windschätzung zu bewerten, wurde mit einer Messstation die Windgeschwindigkeit während der Versuche aufgezeichnet. Die Messungen können – wenn Fluggerät und Windmesser nahe beieinander sind – mit der im Filter geschätzten Windgeschwindigkeit verglichen werden. Daher wurde ein Flug nahe bei der Messstation durchgeführt; dabei wurde im Wesentlichen eine konstante Position gehalten.

Abbildung 6.16 zeigt die Messungen der Windgeschwindigkeit sowie die Schätzung im Navigationsfilter. Es werden Ergebnisse mit beiden auf dem Fluggerät „AirQuad 1“ verfügbaren IMUs dargestellt (vgl. Abschnitt 1.4). Dabei ist zu erkennen, dass die Schätzung der Windgeschwindigkeit bei Verfügbarkeit von GPS sowohl mit der preisgünstigen, als auch mit der höherwertigen IMU gelingt. Lediglich im Bereich 50 s–70 s weicht die Schätzung merklich von der Messung ab. Hier ist jedoch zu beachten, dass das Fluggerät teilweise recht nahe an Gebüsch flog, wo die Windgeschwindigkeit tatsächlich etwas reduziert gewesen sein könnte.

Wenn man ab 80 s auf Stützung durch GPS-Messungen verzichtet, wird der Unterschied zwischen den beiden Inertialmesseinheiten deutlich: Während bei der günstigen IMU auf dem AirQuad-NavBoard hauptsächlich das Abklingen des

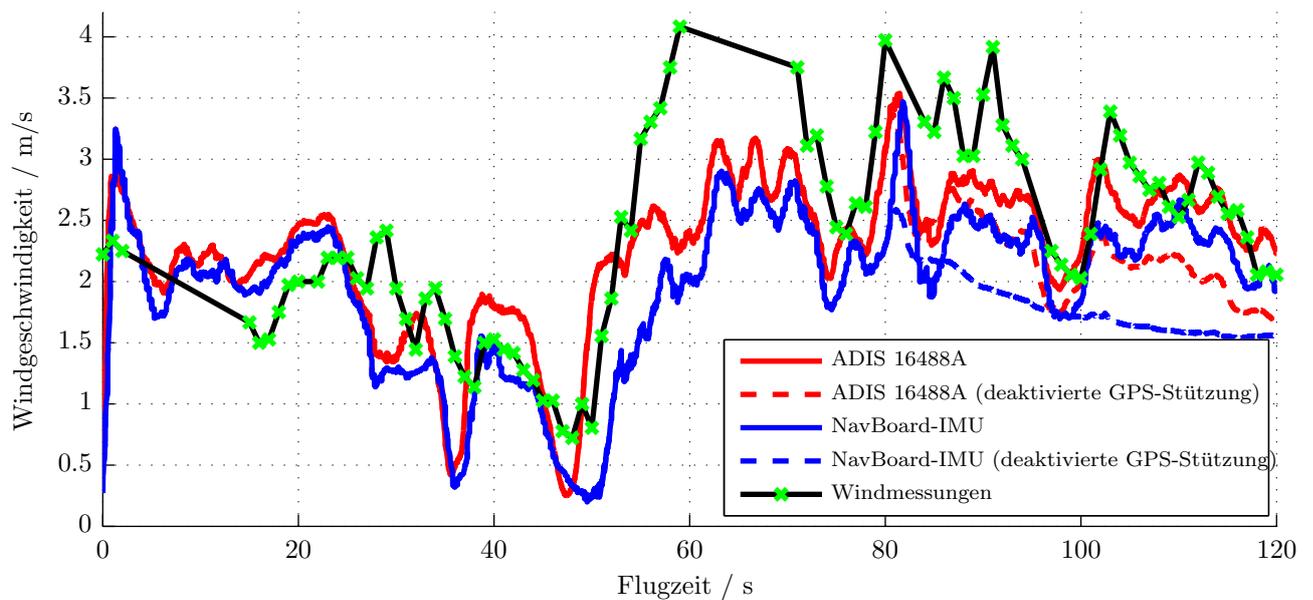


Abbildung 6.16: Vergleich der Windschätzungen basierend auf zwei verschiedenen IMUs mit Windmessungen einer in der Nähe platzierten Messstation. Mit der höherwertigen IMU ADIS 16488A lassen sich auch ohne GPS-Stützung (ab 80 s) noch Änderungen der Windgeschwindigkeit schätzen.

variablen Windanteils zu beobachten ist, können mit der höherwertigen IMU ADIS 16488A auch die Böen zumindest teilweise geschätzt werden.

6.5.3.2 Navigationsergebnisse

Zunächst sollen noch einmal die Flugversuche aus Abbildung 6.13 betrachtet werden. Dabei soll untersucht werden, ob trotz des nur relativ schwachen Windes die Windschätzung zu einem verbesserten Ergebnis führen kann. Dies ist tatsächlich der Fall: Wie Abbildung 6.17 zeigt, kann in allen Flügen, insbesondere im Flug 2, der Positionsfehler nochmals merklich reduziert werden.

Abbildung 6.18 zeigt die Navigationsergebnisse aus den bei stärkerem Wind durchgeführten Experimenten. Neben einer mit GPS/INS-Smoothing erzeugten Referenz sind die Positionsergebnisse basierend auf dem Inertialen Navigationssystem und dem modellgestützten Ansatz mit Windschätzung abgebildet. Beim INS und dem modellgestützten Navigationssystem wurde je zweimal pro Flug künstlich das GPS deaktiviert.

Das INS zeigt das erwartete Verhalten: Bereits nach kurzer Zeit wird die Positionsschätzung unbrauchbar. Ohne Windschätzung würde auch das modellgestützte Navigationssystem divergieren, da der nicht kompensierte Wind sehr starke Fehler verursachen würde. Dahingegen betragen die Fehler beim erweiterten

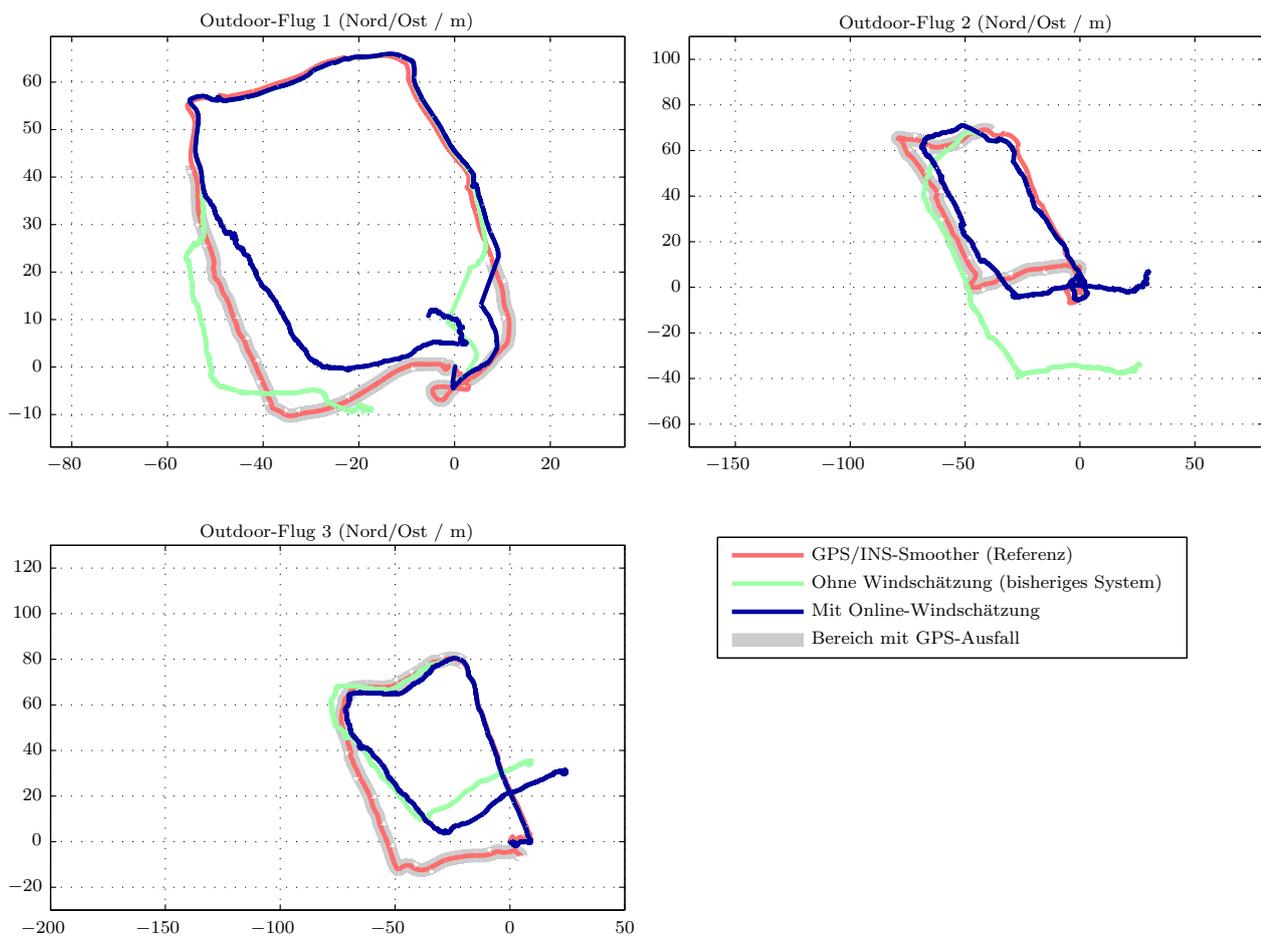


Abbildung 6.17: Verbesserte Positionsschätzung von Flügen im Außenbereich mittels Windschätzung. Obwohl nur geringer Wind vorlag, werden die bereits in Abbildung 6.13 gezeigten Ergebnisse durch Erweiterung um die Windschätzung genauer.

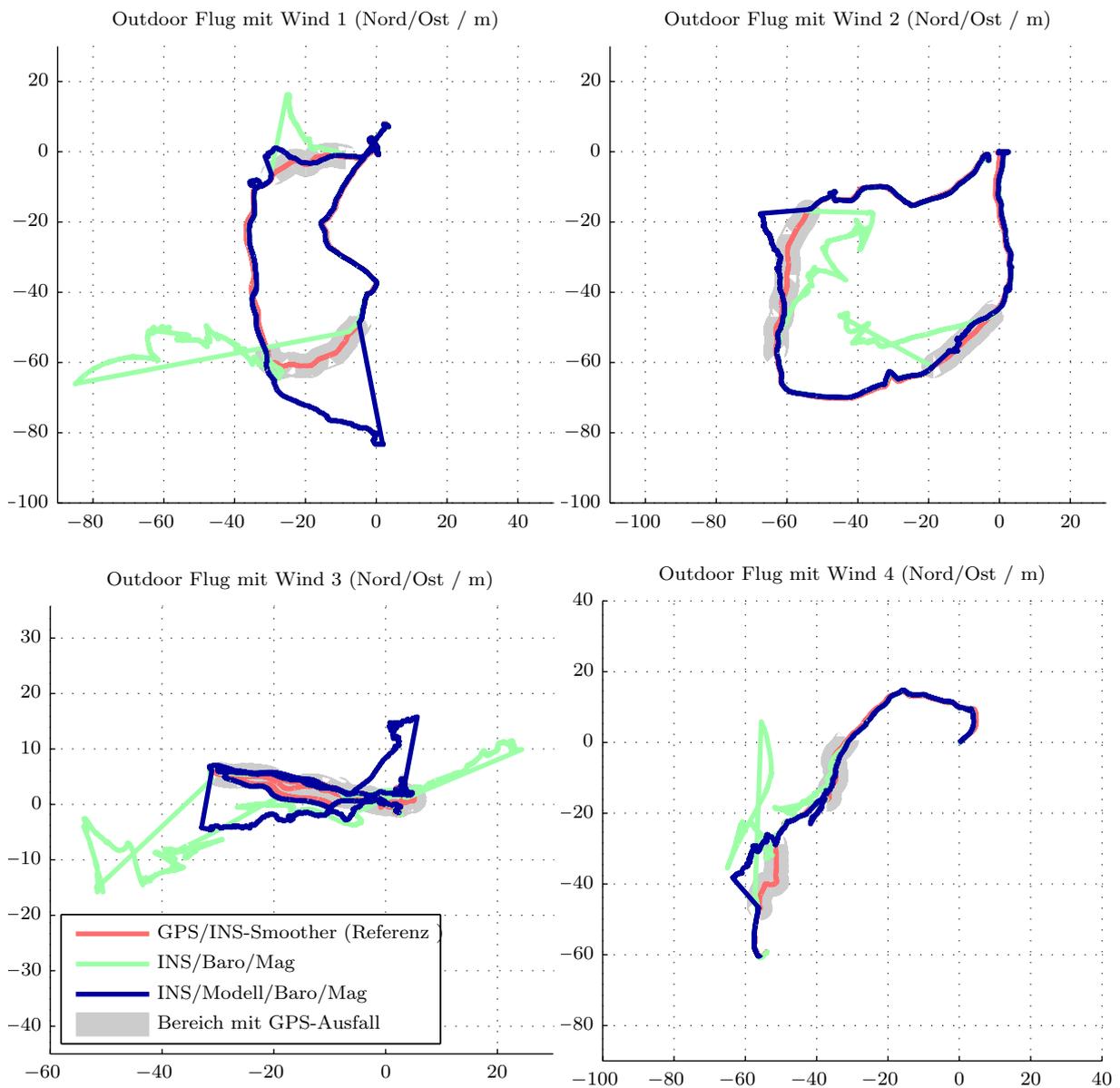


Abbildung 6.18: Ergebnisse von vier Flügen bei böigem Wind. Die Positionslösung des mit Barometer und Magnetometer gestützten INS driftet während der GPS-Ausfälle stark ab. Durch die Modellstützung mit Windschätzung kann dieses Abdriften deutlich reduziert werden.

Navigationssystem mit Windschätzung meist nur wenige Meter, obwohl für die hier gezeigten Ergebnisse die kostengünstige NavBoard-IMU genutzt wurde.

6.5.4 Fazit zur Windkompensation

In diesem Unterkapitel wurde untersucht, wie der Zustandsvektor des Navigationssystems um Zustände zur Windschätzung erweitert werden kann. Es wurde gezeigt, dass eine Überlagerung eines statischen und eines abklingenden Anteils ein geeignetes Windmodell darstellt. Damit wurde eine Windschätzung im Flug möglich. Diese funktioniert sehr gut bei Verfügbarkeit von Stützmessungen wie GPS und kommt ohne Zusatzsensorik wie einem Staurohr aus. Bei ausreichender Güte der IMU ist sogar ohne GPS-Stützung noch eine Windschätzung möglich.

Tests bei böigem Wind wiesen nach, dass mit dem so erweiterten modellgestützten Navigationssystem selbst bei Einsatz von kostengünstigen IMUs Ausfälle von GPS überbrückt werden können.

6.6 Fazit

Das in diesem Kapitel vorgestellte neuartige Navigationssystem stellt einen deutlichen Fortschritt gegenüber herkömmlichen Systemen dar. Durch die geeignete Fusion von Inertialmessungen mit dem Bewegungsmodell des Fluggerätes wird das Fehlerwachstum bei Ausfall von Stützsensoren wirksam begrenzt: Im Vergleich mit einem klassischen Inertialen Navigationssystem kann der Positionsfehler um ein bis zwei Größenordnungen reduziert werden. Durch die implementierte und erfolgreich demonstrierte Schätzung von Parametern während des Fluges entfällt der bei modellgestützter Navigation oft vorhandene zusätzliche Aufwand für spezielle Flüge zur Modellidentifikation; die implementierte Windschätzung erlaubt einen Einsatz auch im Außenbereich. Durch das Navigationssystem werden neue Anwendungen, die auf zuverlässige Onboard-Schätzung der Navigationsgrößen angewiesen sind, ermöglicht.

Die auf der AirQuad-Plattform erzielten Ergebnisse weisen zudem die Stärken des entwickelten, allgemeingültigen Unified-Model-Ansatzes nach: Dieser führt zu einer optimalen, flexibel anwendbaren Fusion zweier Prädiktionsmodelle. Diese ist leicht zu implementieren und effizient. Somit kann eine Anwendung auch auf anderen Plattformen oder gar außerhalb der modellgestützten Navigation, wenn zwei Prädiktionsmodelle vorliegen, empfohlen werden.

Kapitel 7

Zusammenfassung

Ziel dieser Arbeit war die Entwicklung eines robusten Navigationssystems für Quadrocopter, das den Ausfall von Satellitennavigationssystemen kompensieren kann. Dazu wurden Methoden entwickelt, mit denen Lasermessungen und Informationen aus einem Bewegungsmodell ins Navigationssystem integriert werden können.

Das lasergestützte Navigationssystem funktioniert besonders gut in der Nähe von Gebäuden und in Innenräumen – also dort, wo aufgrund von Mehrwegeausbreitung und Abschattungen mit verringerter Güte oder komplettem Ausfall der Satellitenstützung zu rechnen ist. Anhand von Monte-Carlo-Simulationen wurde gezeigt, dass die Stochastic-Cloning-Technik sich besser als andere, häufig verwendete Methoden eignet, um Relativmessungen in einem Kalman-Filter zu nutzen. Nur damit lässt sich eine hohe Schätzgenauigkeit und korrekte Unsicherheitsschätzung erreichen. Letztere ist außerordentlich wichtig für ein multisensorielles Navigationssystem, wenn also weitere Stützinformationen integriert werden sollen. Der Stochastic-Cloning-Ansatz wurde erweitert, so dass die Zeitverzögerung der Relativmessungen ohne nennenswerten Genauigkeitsverlust berücksichtigt werden konnte. Betrachtungen zur optimierten Wahl des Referenzscans und Erweiterungen zur Verarbeitung von Headingwinkel- und Höhenänderungen führten zu reduzierten Fehlern. Es wurde anhand von experimentellen Untersuchungen gezeigt, dass das so definierte Navigationssystem nur langsam anwachsende Positionsfehler aufweist.

Das Wachstum der Positionsfehler kann zudem dadurch verringert werden, dass das erneute Besuchen eines Ortes erkannt und geeignet verarbeitet wird. Diese Verarbeitung wird in einem neu entworfenen Fusionsgraphen durchgeführt. Dieser erlaubt die Berücksichtigung sowohl der inkrementellen Bewegungsschätzung aus dem zuvor entwickelten Navigationssystem als auch der Ortswiedererkennungen. Die

gelieferte Schätzung der Positionsunsicherheit stellt eine wichtige Information für Module zur autonomen Flugführung dar.

Nach dem Flug kann die geflogene Trajektorie noch genauer bestimmt werden, wenn der entwickelte PRC-Smoothing-Ansatz verwendet wird. Dieser beschreibt eine Möglichkeit, die Ortswiedererkennungen in einem RTS-Smoother zu verarbeiten.

In manchen Situationen kann aber auch ein Lasersensor keine Stützung bieten, z. B. wenn keine Objekte in Reichweite sind. Auch existieren viele Fluggeräte ohne Laserscanner, so dass das Ziel der robusten Navigation weitere Stützinformationen erforderlich machte. Hierzu wurde die Stützung durch ein Bewegungsmodell gewählt, da diese ebenso wie die Inertiale Navigation nicht auf äußere Bedingungen angewiesen ist. Um eine solche Stützung zu verwirklichen, mussten neuartige Verfahren and Navigationsansätze entwickelt und auf ihre Eignung hin untersucht werden.

Zunächst wurde die Unified-Model-Methode hergeleitet, die ganz allgemein die Fusion von zwei Prädiktionsmodellen im Kalman-Filter erlaubt. Im Fall der bewegungsmodellgestützten Navigation sind diese zwei Prädiktionsmodelle das Inertiale Navigationssystem und das Bewegungsmodell. Es wurde gezeigt, dass es für die Quadropter-Navigation ausreichend ist, nur den translatorischen Teil des Bewegungsmodells zu verwenden, der rotatorische Teil kann – wie im INS – durch die Gyroskopmessungen ersetzt werden. Dies führte zu einem minimalistischen Navigationssystem, da eben nur ein Teil des Modells benötigt wurde und zudem der Zustandsvektor im Vergleich zum INS nicht anwuchs. Bereits mit diesem Navigationssystem konnte anhand einer Versuchsreihe gezeigt werden, dass das Wachstum des Positionsfehlers bei Ausfall der Stützmessungen um ein bis zwei Größenordnungen gegenüber dem INS reduziert werden kann; die für die Regelung wichtigen Geschwindigkeits- und Lagefehler konnten mit dem neuen System beschränkt werden.

Um die praktische Anwendbarkeit zu erhöhen, wurde das Navigationssystem erweitert, so dass Modellparameter während des Fluges geschätzt werden. So können Modifikationen am Fluggerät oder der Nutzlast automatisch erkannt werden, es ist keine erneute Identifikation der Parameter vor dem Flug erforderlich. Auch eine Schätzung der Windgeschwindigkeit wurde integriert, so dass das Navigationssystem wetterunabhängig eingesetzt werden kann. Diese Erweiterungen wurden ebenfalls anhand von einigen Flugexperimenten getestet. Diese zeigten durchweg deutliche Verbesserungen gegenüber klassischen Navigationssystemen.

Bei allen entwickelten Navigationsmethoden wurde angestrebt, mit den mathematischen Formulierungen möglichst gut die vorliegenden Gegebenheiten zu berücksichtigen. So erlaubt z. B. die Stochastic-Cloning-Methode, Relativmessungen als solche zu verarbeiten, ohne sie in Messungen einer mittleren Geschwindigkeit transformieren zu müssen. Dadurch liefert sowohl das lasergestützte Navigationssystem als auch das modellgestützte Navigationssystem eine konsistente Unsicherheitsschätzung. Diese wiederum erlaubte direkt die Fusion der beiden Teilsysteme zu einem laser- und modellgestützten System. Es wurde an realen Flugdaten gezeigt, dass damit die Genauigkeit bei Laserstützung kombiniert werden kann mit der durch das Bewegungsmodell erreichten Robustheit, so dass bei Ausfall von Stützmessungen die Positionsfehler nur langsam wachsen.

Auch die Erweiterung des multisensoriellen Systems um weitere Sensoren ist problemlos möglich: Die Verarbeitung von GPS-Messungen, barometrischen Höhenmessungen oder Magnetfeldmessungen wurde bereits in dieser Arbeit genutzt. Eine Erweiterung um Relativmessungen aus Kamerabildern wurde in [115] durchgeführt.

Durch die Integration des Bewegungsmodells mit dem Unified-Model-Ansatz wurde ein Navigationssystem geschaffen, dessen Geschwindigkeits- und Lagefehler begrenzt sind, selbst wenn sämtliche Stützsensoren ausfallen. Auch die Positionsfehler unterliegen nur einem langsamen Wachstum. Durch die Integration der Lasermessungen und Ortswiedererkennungen kann in Gebäuden oder in der Nähe von Hindernissen sehr präzise navigiert werden. Störungen, Ausfälle und Abschattungen des GPS-Signals können so wirksam kompensiert werden und stehen dem zuverlässigen Betrieb von autonomen Fluggeräten nicht mehr im Wege.

Anhang A

Herleitung der Unified-Model-Berechnung

Ausgehend von Gleichung (6.32) und den Definitionen $\bar{\Phi}_k = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\Phi_k$, $\bar{\mathbf{G}}_k = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{G}_k$ und $\bar{\mathbf{B}}_k = \begin{pmatrix} (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{B}_k & \mathbf{K}_{k+1} \end{pmatrix}$ sowie mit $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1}$ lässt sich die Propagation des reduzierten Zustandsvektors schreiben als

$$\begin{aligned}
\mathbf{x}_{\cup,k+1}^+ &= \mathbf{T}_{3x4} \bar{\Phi}_k \mathbf{T}_{4x3} \mathbf{x}_{\cup,k}^+ + \mathbf{T}_{3x4} \bar{\mathbf{B}}_k \bar{\mathbf{u}}_k \\
&= \mathbf{T}_{3x4} (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}) \Phi_k \mathbf{T}_{4x3} \mathbf{x}_{\cup,k}^+ + \mathbf{T}_{3x4} \begin{pmatrix} (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{B}_k & \mathbf{K}_{k+1} \end{pmatrix} \bar{\mathbf{u}}_k \\
&= \mathbf{T}_{3x4} (\mathbf{I} - \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \mathbf{H}) \Phi_k \mathbf{T}_{4x3} \mathbf{x}_{\cup,k}^+ \\
&\quad + \mathbf{T}_{3x4} \begin{pmatrix} (\mathbf{I} - \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \mathbf{H})\mathbf{B}_k & \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \end{pmatrix} \bar{\mathbf{u}}_k \\
&= \begin{pmatrix} \underbrace{\mathbf{T}_{3x4} \Phi_k \mathbf{T}_{4x3}}_{:=\Phi_{T,k}} - \mathbf{T}_{3x4} \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \underbrace{\mathbf{H}\Phi_k \mathbf{T}_{4x3}}_{:=\Phi_{H,k}} \end{pmatrix} \mathbf{x}_{\cup,k}^+ \\
&\quad + \left(\mathbf{T}_{3x4} \begin{pmatrix} (\mathbf{I} - \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \mathbf{H})\mathbf{B}_k & \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \end{pmatrix} \right) \bar{\mathbf{u}}_k.
\end{aligned} \tag{A.1}$$

Nach der so erfolgten Definition von $\Phi_{T,k}$ und $\Phi_{H,k}$ soll nun zunächst der Term $\mathbf{T}_{3x4} \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1}$ betrachtet werden. Hier muss noch \mathbf{P}_{k+1}^- ersetzt werden gemäß Gleichung (6.20). Der Term soll hier als \mathbf{U}_k bezeichnet werden.

$$\begin{aligned}
\mathbf{U}_k &= \mathbf{T}_{3x4} \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \\
&= \mathbf{T}_{3x4} (\Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T) \mathbf{H}^T \\
&\quad \cdot (\mathbf{H} (\Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T) \mathbf{H}^T)^{-1} \\
&= \mathbf{T}_{3x4} (\Phi_k \mathbf{T}_{4x3} \mathbf{P}_{\cup,k}^+ \mathbf{T}_{4x3}^T \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T) \mathbf{H}^T \\
&\quad \cdot (\mathbf{H} (\Phi_k \mathbf{T}_{4x3} \mathbf{P}_{\cup,k}^+ \mathbf{T}_{4x3}^T \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T) \mathbf{H}^T)^{-1} \\
&= \underbrace{\mathbf{T}_{3x4} \Phi_k \mathbf{T}_{4x3}}_{\Phi_{T,k}} \mathbf{P}_{\cup,k}^+ \underbrace{\mathbf{T}_{4x3}^T \Phi_k^T \mathbf{H}^T}_{\Phi_{H,k}^T} + \underbrace{\mathbf{T}_{3x4} \mathbf{G}_k \mathbf{Q}_k}_{:=\mathbf{G}_{T,k}} \underbrace{\mathbf{G}_k^T \mathbf{H}^T}_{:=\mathbf{G}_{H,k}^T} \\
&\quad \cdot (\underbrace{\mathbf{H} \Phi_k \mathbf{T}_{4x3}}_{\Phi_{H,k}} \mathbf{P}_{\cup,k}^+ \underbrace{\mathbf{T}_{4x3}^T \Phi_k^T \mathbf{H}^T}_{\Phi_{H,k}^T} + \underbrace{\mathbf{H} \mathbf{G}_k \mathbf{Q}_k}_{\mathbf{G}_{H,k}} \underbrace{\mathbf{G}_k^T \mathbf{H}^T}_{\mathbf{G}_{H,k}^T})^{-1} \\
&= \left(\Phi_{T,k} \mathbf{P}_{\cup,k}^+ \Phi_{H,k}^T + \mathbf{G}_{T,k} \mathbf{Q}_k \mathbf{G}_{H,k}^T \right) \cdot \left(\Phi_{H,k} \mathbf{P}_{\cup,k}^+ \Phi_{H,k}^T + \mathbf{G}_{H,k} \mathbf{Q}_k \mathbf{G}_{H,k}^T \right)^{-1}
\end{aligned} \tag{A.2}$$

Damit lässt sich Gleichung (A.1) weiter umformen:

$$\begin{aligned}
\mathbf{x}_{\cup,k+1}^+ &= (\Phi_{T,k} - \mathbf{U}_k \Phi_{H,k}) \mathbf{x}_{\cup,k}^+ \\
&\quad + \left(\begin{array}{c} \underbrace{\mathbf{T}_{3x4} \mathbf{B}_k}_{:=\mathbf{B}_{T,k}} - \underbrace{\mathbf{T}_{3x4} \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{B}_k}_{\mathbf{U}_k \mathbf{B}_{H,k}} \\ \underbrace{\mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1}}_{\mathbf{U}_k} \end{array} \right)^T \bar{\mathbf{u}}_k \\
&= (\Phi_{T,k} - \mathbf{U}_k \Phi_{H,k}) \mathbf{x}_{\cup,k}^+ \\
&\quad + \left(\mathbf{B}_{T,k} - \mathbf{U}_k \mathbf{B}_{H,k} \quad \mathbf{U}_k \right) \bar{\mathbf{u}}_k
\end{aligned} \tag{A.3}$$

Zur Berechnung von \mathbf{G}_{\cup} betrachtet man

$$\begin{aligned}
\mathbf{G}_{\cup,k} &= \mathbf{T}_{3x4} \bar{\mathbf{G}}_k, \\
&= \mathbf{T}_{3x4} (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{G}_k \\
&= \underbrace{\mathbf{T}_{3x4} \mathbf{G}_k}_{\mathbf{G}_{T,k}} - \underbrace{\mathbf{T}_{3x4} \mathbf{P}_{k+1}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^- \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{G}_k}_{\mathbf{U}_k \mathbf{G}_{H,k}} \\
&= \mathbf{G}_{T,k} - \mathbf{U}_k \mathbf{G}_{H,k}.
\end{aligned} \tag{A.4}$$

Anhang B

Herleitung des Systemmodells mit \mathbf{v}_{eb}^b

B.1 Geschwindigkeitsdifferentialgleichung für INS

Es gilt:

$$\begin{aligned}
 \dot{\mathbf{v}}_{ib}^i &= \mathbf{f}_{ib}^i + \boldsymbol{\gamma}^i \\
 \mathbf{v}_{eb}^e &= \mathbf{C}_i^e (\boldsymbol{\Omega}_{ei}^i \mathbf{p}^i + \mathbf{v}_{ib}^i) \\
 \rightarrow \mathbf{v}_{ib}^i &= \mathbf{C}_e^i \mathbf{v}_{eb}^e - \boldsymbol{\Omega}_{ei}^i \mathbf{p}^i
 \end{aligned} \tag{B.1}$$

Damit kann die gesuchte Differentialgleichung hergeleitet werden:

$$\begin{aligned}
 \dot{\mathbf{v}}_{eb}^b &= \partial/\partial t(\mathbf{C}_e^b \mathbf{v}_{eb}^e) \\
 &= \dot{\mathbf{C}}_e^b \mathbf{v}_{eb}^e + \mathbf{C}_e^b \dot{\mathbf{v}}_{eb}^e \\
 &= \dot{\mathbf{C}}_e^b \mathbf{v}_{eb}^e + \mathbf{C}_e^b \left(\underbrace{\dot{\mathbf{C}}_i^e}_{=\mathbf{C}_i^e \boldsymbol{\Omega}_{ei}^i} (\boldsymbol{\Omega}_{ei}^i \mathbf{p}^i + \mathbf{v}_{ib}^i) + \mathbf{C}_i^e \underbrace{(\dot{\boldsymbol{\Omega}}_{ei}^i \mathbf{p}^i + \boldsymbol{\Omega}_{ei}^i \mathbf{v}_{ib}^i + \mathbf{f}_{ib}^i + \boldsymbol{\gamma}^i)}_{\approx 0} \right) \\
 &= \dot{\mathbf{C}}_e^b \mathbf{v}_{eb}^e + \mathbf{C}_i^b \boldsymbol{\Omega}_{ei}^i \boldsymbol{\Omega}_{ei}^i \mathbf{p}^i + 2\mathbf{C}_i^b \boldsymbol{\Omega}_{ei}^i \mathbf{v}_{ib}^i + \mathbf{f}_{ib}^b + \boldsymbol{\gamma}^b \\
 &= \mathbf{C}_e^b \boldsymbol{\Omega}_{be}^e \mathbf{v}_{eb}^e + \underbrace{\mathbf{C}_i^b \boldsymbol{\Omega}_{ei}^i \boldsymbol{\Omega}_{ei}^i \mathbf{p}^i}_* + 2\mathbf{C}_i^b \boldsymbol{\Omega}_{ei}^i \mathbf{C}_e^i \mathbf{v}_{eb}^e - \underbrace{2\mathbf{C}_i^b \boldsymbol{\Omega}_{ei}^i \boldsymbol{\Omega}_{ei}^i \mathbf{p}^i}_* + \mathbf{f}_{ib}^b + \underbrace{\boldsymbol{\gamma}^b}_* \\
 &= \boldsymbol{\Omega}_{be}^b \mathbf{v}_{eb}^b + \underbrace{\mathbf{C}_n^b \mathbf{g}_b^n}_* - 2 \underbrace{\mathbf{C}_i^b \boldsymbol{\Omega}_{ie}^i \mathbf{C}_e^i \mathbf{C}_b^e \mathbf{v}_{eb}^b}_{=\boldsymbol{\Omega}_{ie}^b} + \mathbf{f}_{ib}^b \\
 &= -(\boldsymbol{\Omega}_{eb}^b + 2\boldsymbol{\Omega}_{ie}^b) \mathbf{v}_{eb}^b + \mathbf{C}_n^b \mathbf{g}_b^n + \mathbf{f}_{ib}^b \\
 &= -(\boldsymbol{\Omega}_{ie}^b + \boldsymbol{\Omega}_{ib}^b) \mathbf{v}_{eb}^b + \mathbf{C}_n^b \mathbf{g}_b^n + \mathbf{f}_{ib}^b
 \end{aligned} \tag{B.2}$$

Die auf den Körper wirkende Schwerebeschleunigung \mathbf{g}_b (*) wurde dabei als Summe der Zentrifugalbeschleunigung und der Beschleunigung aufgrund der Massenanziehung (γ) berechnet:

$$\begin{aligned}
-\mathbf{C}_i^b \Omega_{ei}^i \Omega_{ei}^i \mathbf{p}^i + \gamma^b &= \mathbf{C}_n^b (\gamma^n - \mathbf{C}_i^n \underbrace{\mathbf{C}_n^i \Omega_{ie}^n \mathbf{C}_i^n}_{\Omega_{ie}^i} \underbrace{\mathbf{C}_n^i \Omega_{ie}^n \mathbf{C}_i^n}_{\Omega_{ie}^i} \mathbf{p}^i) \\
&= \mathbf{C}_n^b (\gamma^n - \Omega_{ie}^n \Omega_{ie}^n \mathbf{p}^n) \\
&= \mathbf{C}_n^b \mathbf{g}_b^n \\
&= \mathbf{g}_b^b
\end{aligned} \tag{B.3}$$

B.1.1 Linearisierung

Zuerst wird die Geschwindigkeitsdifferentialgleichung so umgeschrieben, dass wahre Größen durch zu schätzende Größen ersetzt werden. Hierzu wird das Lageinkrement ψ als Eulerwinkelvektor eingeführt. Dieser beschreibt die Drehung der geschätzten Lage $\mathbf{C}_b^{\hat{n}}$ in die wahre Lage \mathbf{C}_b^n , es gilt also

$$\mathbf{C}_{\hat{n}}^n = \mathbf{I} + [\psi \times] = \mathbf{I} + \Psi. \tag{B.4}$$

Damit folgt dann:

$$\begin{aligned}
\dot{\mathbf{v}}_{eb}^b &= -(\Omega_{ie}^b + \Omega_{ib}^b) \mathbf{v}_{eb}^b + \mathbf{C}_n^b \mathbf{g}_b^n + \mathbf{f}_{ib}^b \\
&= -(\mathbf{C}_{\hat{n}}^b \underbrace{\mathbf{C}_n^{\hat{n}} \Omega_{ie}^n}_{\mathbf{I} - \Psi} \underbrace{\mathbf{C}_{\hat{n}}^n}_{\mathbf{I} + \Psi} \mathbf{C}_b^{\hat{n}} + \tilde{\Omega}_{ib}^b - \mathbf{B}_\omega - \mathbf{W}_\omega) \mathbf{v}_{eb}^b + \mathbf{C}_{\hat{n}}^b \mathbf{C}_n^{\hat{n}} \mathbf{g}_b^n + \tilde{\mathbf{f}}_{ib}^b - \mathbf{b}_a - \mathbf{n}_a \\
&= \Omega_{ie}^n - \Psi \Omega_{ie}^n + \Omega_{ie}^n \Psi - \Psi \Omega_{ie}^n \Psi \\
&= -(\mathbf{C}_{\hat{n}}^b (\Omega_{ie}^n - \Psi \Omega_{ie}^n + \Omega_{ie}^n \Psi - \Psi \Omega_{ie}^n \Psi) \mathbf{C}_b^{\hat{n}} + \tilde{\Omega}_{ib}^b - \mathbf{B}_\omega - \mathbf{W}_\omega) \mathbf{v}_{eb}^b \\
&\quad + \mathbf{C}_{\hat{n}}^b (\mathbf{I} + \Psi) \mathbf{g}_b^n + \tilde{\mathbf{f}}_{ib}^b - \mathbf{b}_a - \mathbf{n}_a
\end{aligned} \tag{B.5}$$

Danach ist die Differentiation nach den absoluten, zu schätzenden Größen möglich:

$$\partial \dot{\mathbf{v}}_{eb}^b / \partial \mathbf{v}_{eb}^b \big|_{\mathbf{x}=\hat{\mathbf{x}}} = - \left(\mathbf{C}_{\hat{n}}^b \Omega_{ie}^n \mathbf{C}_b^{\hat{n}} + \tilde{\Omega}_{ib}^b - \hat{\mathbf{B}}_\omega \right) \tag{B.6}$$

$$\begin{aligned}
\partial \dot{\mathbf{v}}_{eb}^b / \partial \boldsymbol{\psi} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} &= \partial / \partial \boldsymbol{\psi} (\mathbf{C}_{\hat{n}}^b \boldsymbol{\Psi} \boldsymbol{\Omega}_{ie}^n \mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b - \mathbf{C}_{\hat{n}}^b \boldsymbol{\Omega}_{ie}^n \boldsymbol{\Psi} \mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b - \mathbf{C}_{\hat{n}}^b \boldsymbol{\Psi} \mathbf{g}_b^n) \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \\
&= \partial / \partial \boldsymbol{\psi} (\mathbf{C}_{\hat{n}}^b [\boldsymbol{\Omega}_{ie}^n \mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b \times] \boldsymbol{\psi} + \mathbf{C}_{\hat{n}}^b \boldsymbol{\Omega}_{ie}^n [\mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b \times] \boldsymbol{\psi} + \mathbf{C}_{\hat{n}}^b [\mathbf{g}_b^n \times] \boldsymbol{\psi}) \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \\
&= \mathbf{C}_{\hat{n}}^b (-[\boldsymbol{\Omega}_{ie}^n \mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b \times] + \boldsymbol{\Omega}_{ie}^n [\mathbf{C}_b^{\hat{n}} \mathbf{v}_{eb}^b \times] + [\mathbf{g}_b^n \times]) \quad (\text{B.7})
\end{aligned}$$

$$\partial \dot{\mathbf{v}}_{eb}^b / \partial \mathbf{b}_a \Big|_{\mathbf{x}=\hat{\mathbf{x}}} = -\mathbf{I} \quad (\text{B.8})$$

$$\begin{aligned}
\partial \dot{\mathbf{v}}_{eb}^b / \partial \mathbf{b}_\omega \Big|_{\mathbf{x}=\hat{\mathbf{x}}} &= \partial / \partial \mathbf{b}_\omega (\mathbf{B}_\omega \mathbf{v}_{eb}^b) \\
&= \partial / \partial \mathbf{b}_\omega (-[\mathbf{v}_{eb}^b \times] \mathbf{b}_\omega) \\
&= -[\mathbf{v}_{eb}^b \times] \quad (\text{B.9})
\end{aligned}$$

Zur Herleitung der G-Matrix benötigt man die Ableitungen nach den Rauschgrößen.

Wegen $\mathbf{x} = \hat{\mathbf{x}} - \delta \mathbf{x}$ folgt generell

$$\partial \dot{\mathbf{v}}_{eb}^b / \partial \boldsymbol{\omega} = -\partial (\delta \dot{\mathbf{v}}_{eb}^b) / \partial \boldsymbol{\omega}. \quad (\text{B.10})$$

Somit ergibt sich

$$\partial \dot{\mathbf{v}}_{eb}^b / \partial \boldsymbol{\omega}_a = -\mathbf{I} \quad (\text{B.11})$$

$$\rightarrow \partial (\delta \dot{\mathbf{v}}_{eb}^b) / \partial \boldsymbol{\omega}_a = \mathbf{I} \quad (\text{B.12})$$

$$\begin{aligned}
\partial \delta \dot{\mathbf{v}}_{eb}^b / \partial \boldsymbol{\omega}_\omega \Big|_{\delta \mathbf{x}=0} &= - \partial / \partial \boldsymbol{\omega}_\omega (\mathbf{W}_\omega \hat{\mathbf{v}}_{eb}^b) \\
&= \partial / \partial \boldsymbol{\omega}_\omega ([\hat{\mathbf{v}}_{eb}^b \times] \boldsymbol{\omega}_\omega) \\
&= [\hat{\mathbf{v}}_{eb}^b \times]. \quad (\text{B.13})
\end{aligned}$$

B.2 Positionsdifferentialgleichung

$$\begin{aligned}
 \dot{\mathbf{p}}_{eb}^n &= \partial/\partial t(\mathbf{C}_b^n \mathbf{p}_{eb}^e) \\
 &= \dot{\mathbf{C}}_b^n \mathbf{p}_{eb}^e + \mathbf{C}_b^n \mathbf{v}_{eb}^e \\
 &= \mathbf{C}_e^n \boldsymbol{\Omega}_{ne}^e \mathbf{p}_{eb}^e + \mathbf{C}_b^n \mathbf{v}_{eb}^b \\
 &= \mathbf{C}_e^n \boldsymbol{\Omega}_{ne}^e \mathbf{C}_n^e \mathbf{C}_e^n \mathbf{p}_{eb}^e + \mathbf{C}_b^n \mathbf{v}_{eb}^b \\
 &= - \underbrace{\boldsymbol{\Omega}_{en}^n}_{\approx \mathbf{0}} \mathbf{p}_{eb}^n + \mathbf{C}_b^n \mathbf{v}_{eb}^b
 \end{aligned} \tag{B.14}$$

$\approx \mathbf{0}$ für kleine Geschwindigkeiten

$$\approx \mathbf{C}_b^n \mathbf{v}_{eb}^b \tag{B.15}$$

B.2.1 Linearisierung

$$\partial \dot{\mathbf{p}}_{eb}^n / \partial \mathbf{v}_{eb}^b = \mathbf{C}_b^n \tag{B.16}$$

$$\begin{aligned}
 \partial \dot{\mathbf{p}}_{eb}^n / \partial \boldsymbol{\psi} &= \partial / \partial \boldsymbol{\psi} (\mathbf{C}_{\hat{n}}^n \mathbf{C}_{\hat{b}}^{\hat{n}} \mathbf{v}_{eb}^b) \\
 &= \partial / \partial \boldsymbol{\psi} ((\mathbf{I} + \boldsymbol{\Psi}) \mathbf{C}_{\hat{b}}^{\hat{n}} \mathbf{v}_{eb}^b) \\
 &= \partial / \partial \boldsymbol{\psi} (-[\mathbf{C}_{\hat{b}}^{\hat{n}} \mathbf{v}_{eb}^b \times] \boldsymbol{\psi}) \\
 &= -[\mathbf{C}_{\hat{b}}^{\hat{n}} \mathbf{v}_{eb}^b \times]
 \end{aligned} \tag{B.17}$$

Literatur

- [1] *U.S. Military Specification MIL-F-8785C*, Nov. 1980.
- [2] *Department of Defense World Geodetic System 1984 - TR8350.2*. National Imagery And Mapping Agency, 3. Aufl., 2000.
- [3] ABAS, N., A. LEGOWO und R. AKMELIAWATI: *Parameter identification of an autonomous quadrotor*. In: *4th International Conference On Mechatronics (ICOM)*, S. 17–19, Kuala Lumpur, Malaysia, 2011.
- [4] ABBASI, A., U. M. AL-SAGGAF und K. MUNAWAR: *Design of Unmanned Aerial Vehicle Autopilot based on Freescale Qorivva 32-bit Microcontroller*. *Life Science Journal*, 11(11):874–878, 2014.
- [5] ABBEEL, P., A. COATES, M. MONTEMERLO, A. Y. NG und S. THRUN: *Discriminative Training of Kalman Filters*. In: *Proceedings of Robotics: Science and Systems Conference*, 2005.
- [6] ABEYWARDENA, D., Z. WANG, G. DISSANAYAKE, S. L. WASLANDER und S. KODAGODA: *Model-aided state estimation for quadrotor micro air vehicles amidst wind disturbances*. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 4813–4818, Sep. 2014.
- [7] ACHELNIK, M., A. BACHRACH, R. HE, S. PRENTICE und N. ROY: *Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments*. In: *Proceedings of Robotics: Science and Systems Conference*, 2008.
- [8] ALPEN, M., K. FRICK und J. HORN: *Intelligent Robotics and Applications*, Bd. 7101 d. Reihe *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

- [9] ANDERSON, J. D.: *Fundamentals of Aerodynamics*. McGraw-Hill Education, 5. Aufl., 2010.
- [10] BACHRACH, A., S. PRENTICE, R. HE und N. ROY: *RANGE-Robust autonomous navigation in GPS-denied environments*. *Journal of Field Robotics*, 28(5):644–666, Sep. 2011.
- [11] BAILEY, T. und H. DURRANT-WHYTE: *Simultaneous localization and mapping (SLAM): part II*. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sep. 2006.
- [12] BESL, P. und H. MCKAY: *A method for registration of 3-D shapes*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [13] BIERMAN, G. J.: *Factorization methods for discrete sequential estimation*. Courier Corporation, 2006.
- [14] BLOM, H. und Y. BAR-SHALOM: *The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients*. *IEEE Transactions On Automatic Control*, 33(8):780–783, 1988.
- [15] BOUABDALLAH, S. und R. SIEGWART: *Full control of a quadrotor*. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, (1):153–158, Okt. 2007.
- [16] BRISTEAU, P.-J., F. CALLOU, D. VISSIÈRE und N. PETIT: *The Navigation and Control technology inside the AR.Drone micro UAV*. In: *18th IFAC World Congress*, S. 1477–1484. Mailand, Italien, 2011.
- [17] BRISTEAU, P.-J., P. MARTIN, E. SALAÜN und N. PETIT: *The Role of Propeller Aerodynamics in the Model of a Quadrotor UAV*. In: *European Control Conference*, S. 683–688, Budapest, Ungarn, 2009.
- [18] BROWN, R. und P. Y. C. HWANG: *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 3 Aufl., 1997.
- [19] BRUGGEMANN, T. S., D. G. GREER und R. A. WALKER: *GPS Fault Detection with IMU and Aircraft Dynamics*. *IEEE Transactions On Aerospace And Electronic Systems*, 47(1):305–316, 1996.

- [20] BRY, A., A. BACHRACH und N. ROY: *State estimation for aggressive flight in GPS-denied environments using onboard sensing*. In: *2012 IEEE International Conference on Robotics and Automation*, S. 1–8. IEEE, Mai 2012.
- [21] BRYSON, M. und S. SUKKARIEH: *Vehicle Model Aided Inertial Navigation for a UAV using Low-cost Sensors*. In: *Proceedings of the Australasian Conference on Robotics and Automation*, 2004.
- [22] CALDEIRA, T., L. SENEVIRATNE und J. DIAS: *Indoor Exploration Using a μ UAV and a Spherical Geometry Based Visual System*. In: CAMARINHA-MATOS, L. M., S. TOMIC und P. GRAÇA (Hrsg.): *Technological Innovation for the Internet of Things*, Bd. 394 d. Reihe *IFIP Advances in Information and Communication Technology*, S. 200–209. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [23] CHOWDHARY, G. und R. JATEGAONKAR: *Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter*. *Aerospace Science and Technology*, 14(2):106–117, März 2010.
- [24] CHOWDHARY, G., E. N. JOHNSON, D. MAGREE, A. WU und A. SHEIN: *GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft*. *Journal of Field Robotics*, 30(3):415–438, Mai 2013.
- [25] DADKHAH, N., B. METTLER und D. GEBRE-EGZIABHER: *A Model-Aided AHRS for Micro Aerial Vehicle Application*. In: *ION GNSS 21st International Technical Meeting of the Satellite Division*, S. 545–553, Savannah, Sep. 2008.
- [26] DISSANAYAKE, G., S. SUKKARIEH, E. NEBOT und H. DURRANT-WHYTE: *The Aiding of a Low-Cost Strapdown Inertial Measurement Unit Using Vehicle Model Constraints for Land Vehicle Applications*. *IEEE Transactions on Robotics and Automation*, 17(5):731–747, 2001.
- [27] DURRANT-WHYTE, H. und T. BAILEY: *Simultaneous localization and mapping: part I*. *IEEE Robotics & Automation Magazine*, 13(2):99–110, Juni 2006.
- [28] FLANZER, T. und S. A. NING: *Online Parameter Estimation for the Adaptive Control of Unmanned Aerial Vehicles*. <http://cs229.stanford.edu/proj2009/FlanzerNing.pdf>, 2009.

- [29] FOLKESSON, J. und H. I. CHRISTENSEN: *Closing the Loop With Graphical SLAM*. IEEE Transactions on Robotics, 23(4):731–741, Aug. 2007.
- [30] FRASER, D. und J. POTTER: *The optimum linear smoother as a combination of two optimum linear filters*. IEEE Transactions on Automatic Control, 14(4):387–390, Aug. 1969.
- [31] FRAUNDORFER, F., L. HENG, D. HONEGGER, G. H. LEE, L. MEIER, P. TANSKANEN und M. POLLEFEYS: *Vision-based autonomous mapping and exploration using a quadrotor MAV*. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 4557–4564. IEEE, Okt. 2012.
- [32] FRIETSCH, N.: *Bildbasiertes Navigationssystem eines unbemannten Mini-Helikopters*. Logos Verlag, Berlin, 2012.
- [33] GARCÍA CARRILLO, L. R., E. RONDON, A. SANCHEZ, A. DZUL und R. LOZANO: *Stabilization and Trajectory Tracking of a Quad-Rotor Using Vision*. Journal of Intelligent & Robotic Systems, 61(1-4):103–118, Okt. 2010.
- [34] GIBBS, B. P.: *Advanced Kalman Filtering, Least-Squares and Modeling*. Wiley, 2011.
- [35] GÖRCKE, L., J. DAMBECK und F. HOLZAPFEL: *Results of Model-Aided Navigation with Real Flight Data*. In: *Proceedings of the 2014 International Technical Meeting of The Institute of Navigation*, San Diego, CA, 2014.
- [36] GROVES, P. D.: *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
- [37] GRZONKA, S., G. GRISSETTI und W. BURGARD: *Towards a navigation system for autonomous indoor flying*. In: *2009 IEEE International Conference on Robotics and Automation*, S. 2878–2883, Kobe, Mai 2009. IEEE.
- [38] GRZONKA, S., G. GRISSETTI und W. BURGARD: *A Fully Autonomous Indoor Quadrotor*. IEEE Transactions on Robotics, 28(1):90–100, Feb. 2012.
- [39] GRZONKA, S., B. STEDER und W. BURGARD: *3D Place Recognition and Object Detection using a Small-sized Quadrotor*. In: *Robotics: Science and Systems (RSS), Workshop on 3D Exploration, Mapping, and Surveillance with Aerial Robots*, 2011.

- [40] HEGRENAES, O., E. BERGLUND und O. HALLINGSTAD: *Model-Aided Inertial Navigation for Underwater Vehicles*. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, S. 1069 – 1076, Pasadena, 2008.
- [41] HEGRENAES, O. und O. HALLINGSTAD: *Model-Aided INS With Sea Current Estimation for Robust Underwater Navigation*. *IEEE Journal of Oceanic Engineering*, 36(2):316–337, 2011.
- [42] HOFFMANN, G. M., H. HUANG, S. L. WASLANDER und C. J. TOMLIN: *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, S. 1–20, Hilton Head, SC, 2007.
- [43] HUANG, A. S., A. BACHRACH, P. HENRY, M. KRAININ, D. FOX und N. ROY: *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*. In: *International Symposium on Robotics Research (ISRR)*, S. 1–16, 2011.
- [44] HUH, S., D. H. SHIM und J. KIM: *Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs*. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 3158–3163. IEEE, Nov. 2013.
- [45] JULIER, S. und J. UHLMANN: *A counter example to the theory of simultaneous localization and map building*. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Bd. 4, S. 4238–4243. IEEE, 2001.
- [46] JULIER, S. und J. K. UHLMANN: *General Decentralized Data Fusion with Covariance Intersection (CI)*. In: *Handbook of Multisensor Data Fusion*, Kap. 12. CRC Press LLC, 2001.
- [47] JULIER, S. J. und H. F. DURRANT-WHYTE: *A Horizontal Model Fusion Paradigm*. In: *Proceedings of the SPIE AeroSense Conference*, Bd. 2738, S. 37–48, 1996.
- [48] KALMAN, R. E.: *A new approach to linear filtering and prediction problems*. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

- [49] KERL, C., J. STURM und D. CREMERS: *Robust odometry estimation for RGB-D cameras*. In: *2013 IEEE International Conference on Robotics and Automation*, S. 3748–3754, Karlsruhe, Mai 2013. IEEE.
- [50] KLEIN, I., S. FILIN und T. TOLEDO: *Vehicle Constraints for Supporting INS Navigation in Urban Environments*. NAVIGATION, 58(1):7–15, 2011.
- [51] KLINGBEIL, L., M. NIEUWENHUISEN, J. SCHNEIDER, C. ELING, D. DROESCHEL, D. HOLZ, T. LÄBE, W. FÖRSTNER, S. BEHNKE und H. KUHLMANN: *Towards autonomous navigation of an UAV-based mobile mapping system*. In: *Proc. of Int. Conf. on Machine Control & Guidance (MCG)*, Braunschweig, 2014.
- [52] KNEIP, L., M. CHLI und R. SIEGWART: *Robust Real-Time Visual Odometry with a Single Camera and an IMU*. In: *Proceedings of the British Machine Vision Conference*, S. 16.1–16.11, 2011.
- [53] KOIFMAN, M. und I. BAR-ITZHACK: *Inertial Navigation System Aided by Aircraft Dynamics*. IEEE Transactions on Control Systems Technology, 7(4):487–493, 1999.
- [54] KOIFMAN, M. und S. J. MERHAV: *Autonomously Aided Strapdown Attitude Reference System*. Journal of Guidance, Control, and Dynamics, 14(6):1164–1172, 1991.
- [55] KRÜGER, T., S. NOWAK, J. MATTHAEI und P. HECKER: *LiDAR-based Relative Positioning for Swarm Exploration on Mars*. In: *Proceedings of the 2014 International Technical Meeting of The Institute of Navigation*, S. 424–430, San Diego, CA, 2014.
- [56] LEISHMAN, R. C., J. C. MACDONALD, R. W. BEARD und T. W. MCLAIN: *Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model*. IEEE Control Systems, 34(1):28–41, Feb. 2014.
- [57] LEISHMAN, R. C., T. W. MCLAIN und R. W. BEARD: *Relative Navigation Approach for Vision-Based Aerial GPS-Denied Navigation*. Journal of Intelligent & Robotic Systems, S. 97–111, Okt. 2013.

- [58] LEUTENEGGER, S., A. MELZER, K. ALEXIS und R. SIEGWART: *Robust state estimation for small unmanned airplanes*. 2014 IEEE Conference on Control Applications (CCA), S. 1003–1010, Okt. 2014.
- [59] MA, X., S. SUKKARIEH und J.-H. KIM: *Vehicle Model Aided Inertial Navigation*. In: *Proceedings of the IEEE Intelligent Transportation Systems*, Bd. 2, S. 1004–1009, 2003.
- [60] MEISTER, O.: *Entwurf und Realisierung einer Aufklärungsplattform auf Basis eines unbemannten Minihelikopters mit autonomen Flugfähigkeiten*. Logos Verlag, Berlin, 2010.
- [61] MICHAEL, N., D. MELLINGER, Q. LINDSEY und V. KUMAR: *The GRASP Multiple Micro-UAV Testbed*. IEEE Robotics & Automation Magazine, 17(3):56–65, Sep. 2010.
- [62] MOURIKIS, A. I., S. I. ROUMELIOTIS und J. W. BURDICK: *SC-KF Mobile Robot Localization: A Stochastic Cloning Kalman Filter for Processing Relative-State Measurements*. IEEE Transactions on Robotics, 23(4):717–730, Aug. 2007.
- [63] MUHAMMAD, H., H. P. THIEN und T. MULYANTO: *Total Least Squares Estimation of Aerodynamic Parameter of Micro Coaxial Helicopter from Flight Data*. International Journal of Basic and Applied Sciences, 12(2):44–52, 2012.
- [64] NELDER, J. A. und R. MEAD: *A Simplex Method for Function Minimization*. The Computer Journal, 7(4):308–313, Jan. 1965.
- [65] NIEHSEN, W.: *Information fusion based on fast covariance intersection filtering*. In: *Proceedings of the Fifth International Conference on Information Fusion 2002*, Bd. 2, S. 901–904. Int. Soc. Inf. Fusion, 2002.
- [66] RAMER, U.: *An iterative procedure for the polygonal approximation of plane curves*. Computer Graphics and Image Processing, 1(3):244–256, Nov. 1972.
- [67] RAUCH, H. E., C. T. STRIEBEL und F. TUNG: *Maximum likelihood estimates of linear dynamic systems*. AIAA Journal, 3(8):1445–1450, Aug. 1965.
- [68] RODRIGUEZ, A., E. ANDERSEN, J. BRADLEY und C. TAYLOR: *Wind Estimation Using an Optical Flow Sensor on a Miniature Air Vehicle*. In: AIAA

- Guidance, Navigation and Control Conference and Exhibit*, S. 1–8, Reston, Virginia, Aug. 2007. American Institute of Aeronautics and Astronautics.
- [69] ROMERO, H., S. SALAZAR, A. SANCHEZ, P. CASTILLO und R. LOZANO: *Modelling and real-time control stabilization of a new VTOL aircraft with eight rotors*. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 147–152. IEEE, Okt. 2007.
- [70] ROUMELIOTIS, S. und J. BURDICK: *Stochastic cloning: a generalized framework for processing relative state measurements*. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation*, Bd. 2, S. 1788–1795. IEEE, 2002.
- [71] RUSINKIEWICZ, S. und M. LEVOY: *Efficient variants of the ICP algorithm*. In: *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, S. 145–152. IEEE Comput. Soc, 2001.
- [72] SARGENT, R. G.: *Verification And Validation Of Simulation Models*. In: *Proceedings of the 2004 Winter Simulation Conference*, Bd. 1, 2004.
- [73] SEIBOLD, J.: *Entwicklung eines heterogenen Multirobotersystems zur Aufklärung in unbekanntem Gelände*. Dissertation, Karlsruher Institut für Technologie, 2014.
- [74] SENDOBRY, A.: *A Model Based Navigation Architecture for Small Unmanned Aerial Vehicles*. In: *Proceedings of the European Navigation Conference 2011*, London, Dez. 2011.
- [75] SENDOBRY, A.: *Control System Theoretic Approach to Model Based Navigation*. Ingenieurwissenschaftlicher Verlag, Bonn, 2014.
- [76] SHEN, S., N. MICHAEL und V. KUMAR: *Autonomous multi-floor indoor navigation with a computationally constrained MAV*. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, S. 20–25, Mai 2011.
- [77] SIMON, D.: *Optimal State Estimation*. John Wiley & Sons, Inc., Hoboken, NJ, USA, Mai 2006.

- [78] STANCULEANU, I. und T. BORANGIU: *Quadrotor Black-Box System Identification*. World Academy of Science, Engineering and Technology, 5:300–303, 2011.
- [79] TIPALDI, G. D. und K. O. ARRAS: *FLIRT - Interest regions for 2D range data*. In: *2010 IEEE International Conference on Robotics and Automation*, S. 3616–3622, Anchorage, AK, Mai 2010. IEEE.
- [80] TITTERTON, D. und J. L. WESTON: *Strapdown inertial navigation technology*. The Institution of Electrical Engineers, Stevenage, UK, 2. Aufl., 2004.
- [81] TOMIĆ, T., K. SCHMID, P. LUTZ, A. DÖMEL, M. KASSECKER, E. MAIR, I. L. GRIXA, F. RUESS, M. SUPPA und D. BURSCHKA: *Toward a Fully Autonomous UAV*. IEEE Robotics & Automation Magazine, 19(3):46–56, 2012.
- [82] UHLMANN, J. K.: *Dynamic Map Building and Localization: New Theoretical Foundations*. Dissertation, University of Oxford, 1995.
- [83] VASCONCELOS, J., C. SILVESTRE, P. OLIVEIRA und B. GUERREIRO: *Embedded UAV model and LASER aiding techniques for inertial navigation systems*. Control Engineering Practice, 18(3):262–278, März 2010.
- [84] VISSIÈRE, D.: *Experimental autonomous flight of a small-scaled helicopter using accurate dynamics model and low-cost sensors*. In: *Proceedings of the 17th World Congress of The International Federation of Automatic Control*, S. 14642–14650, 2008.
- [85] WASLANDER, S. und C. WANG: *Wind Disturbance Estimation and Rejection for Quadrotor Position Control*. In: *AIAA Infotech@Aerospace Conference*, Seattle, USA, Apr. 2009. American Institute of Aeronautics and Astronautics.
- [86] WEISS, S., M. W. ACHELNIK, S. LYNEN, M. CHLI und R. SIEGWART: *Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments*. In: *2012 IEEE International Conference on Robotics and Automation*, S. 957–964. IEEE, Mai 2012.
- [87] WENDEL, J.: *Integrierte Navigationssysteme*. Oldenbourg Verlag, München, 2. Aufl., 2011.

- [88] YAGHOBI, M., M. JADALIHA, J. ZOLGHADR und M. NOROUZI: *Adaptive line extraction algorithm for SLAM application*. In: *2008 IEEE International Conference on Robotics and Biomimetics*, S. 1850–1855. IEEE, Feb. 2009.
- [89] ZACHARIAH, D. und M. JANSSON: *Self-motion and wind velocity estimation for small-scale UAVs*. In: *2011 IEEE International Conference on Robotics and Automation*, S. 1166–1171. IEEE, Mai 2011.
- [90] ZHAO, W., Z. FANG und P. LI: *Bridging GPS Outages for Fixed-wing Unmanned Aerial Vehicles*. *Navigation*, 68(02):308–326, Sep. 2014.

Datenblätter

- [91] *Analog Devices ADIS 16265*. http://www.analog.com/static/imported-files/data_sheets/ADIS16260_16265.pdf.
- [92] *Analog Devices ADIS 16488A*. http://www.analog.com/static/imported-files/data_sheets/ADIS16488A.pdf.
- [93] *Beagleboard Rev C4*. http://beagleboard.org/static/BBSRM_latest.pdf.
- [94] *Cool XpressRunner-GS45*. http://www.adlinktech.com/PD/marketing/Datasheet/CoolXpressRunner-GS45/CoolXpressRunner-GS45_Datasheet_en_2.pdf.
- [95] *Hokuyo URG-04LX*. https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html.
- [96] *Hokuyo UTM-30LX*. https://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html.
- [97] *VTI SCA-3100*. <http://www.murata.co.jp/products/sensor/pdf/sca3100-d04.pdf>.

Eigene Veröffentlichungen

- [98] CROCOLL, P., T. CASELITZ, B. HETTICH, M. LANGER und G. F. TROMMER: *Laser-aided navigation with loop closure capabilities for Micro Aerial Vehicles in indoor and urban environments*. In: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, S. 373–384. IEEE, Mai 2014.

- [99] CROCOLL, P., L. GÖRCKE, G. F. TROMMER und F. HOLZAPFEL: *Unified Model technique for inertial navigation aided by vehicle dynamics model*. In: *ION International Technical Meeting*, San Diego, CA, 2013.
- [100] CROCOLL, P., L. GÖRCKE, G. F. TROMMER und F. HOLZAPFEL: *Unified Model Technique for Inertial Navigation Aided by Vehicle Dynamics Model*. *Navigation*, 60(3):179–193, Sep. 2013.
- [101] CROCOLL, P., J. SEIBOLD, N. FRIETSCH und G. F. TROMMER: *Collision-free trajectory planning and local obstacle avoidance for MAVs*. In: *European Navigation Conference*, London, 2011.
- [102] CROCOLL, P., J. SEIBOLD, N. FRIETSCH und G. F. TROMMER: *Hindernisfreie Trajektorien und lokale Ausweichstrategien mit Hilfe eines Missionsplanungssystems*. In: *DGLR-Workshop „UAV-Regelung und Flugführung – Vom Missionsauftrag zur geflogenen Trajektorie“*, Braunschweig, 2011.
- [103] CROCOLL, P., J. SEIBOLD, M. POPP und G. F. TROMMER: *Indoor Navigation for a Micro Aerial Vehicle Aided by Laser Range Finder Measurements*. In: *European Navigation Conference*, Wien, 2013.
- [104] CROCOLL, P., J. SEIBOLD, M. POPP und G. F. TROMMER: *Indoor Navigation für einen unbemannten Helikopter basierend auf Laser-Stützung*. In: *DGLR-Workshop „Systemtechnologien für den autonomen Flug von UAVs“*, Stuttgart, 2013.
- [105] CROCOLL, P., J. SEIBOLD, M. POPP und G. F. TROMMER: *Laser Range Finder Aided Indoor Navigation for a Micro Aerial Vehicle*. *European Journal of Navigation*, 11(1):4–14, 2013.
- [106] CROCOLL, P., J. SEIBOLD, G. SCHOLZ und G. F. TROMMER: *Model-Aided Navigation for a Quadrotor Helicopter: A Novel Navigation System and First Experimental Results*. *Navigation*, 61(4):253–271, Dez. 2014.
- [107] CROCOLL, P., J. SEIBOLD, G. SCHOLZ und G. F. TROMMER: *Model-Aided navigation for a quadrotor helicopter: A novel navigation system and first experimental results*. In: *ION International Technical Meeting*, San Diego, CA, 2014.

- [108] CROCOLL, P. und G. F. TROMMER: *Quadrotor Inertial Navigation Aided by a Vehicle Dynamics Model with In-Flight Parameter Estimation*. Eingereicht bei Navigation, 2014.
- [109] CROCOLL, P. und G. F. TROMMER: *Quadrotor Inertial Navigation Aided by a Vehicle Dynamics Model with In-Flight Parameter Estimation*. In: *ION GNSS+*, S. 1784 – 1795, Tampa, FL, 2014.

Veröffentlichungen als Koautor

- [110] ASCHER, C., C. KESSLER, P. CROCOLL und G. F. TROMMER: *Real-time implementation of a Laser-based Graphical OrthoSLAM method coupled with a Dual IMU System*. In: *ION International Technical Meeting*, San Diego, CA, 2011.
- [111] ASCHER, C., C. KESSLER, A. MAIER, P. CROCOLL und G. F. TROMMER: *New Pedestrian Trajectory Simulator to study Innovative Yaw Angle Constraints*. In: *ION GNSS*, Portland, OR, 2010.
- [112] FRIETSCH, N., J. SEIBOLD, P. CROCOLL, M. WEISS und G. F. TROMMER: *Real Time Implementation of a Vision-Based UAV Detection and Tracking System for UAV-Navigation Aiding*. In: *AIAA Guidance, Navigation and Control Conference*, S. 1–11, Portland, OR, 2011.
- [113] KESSLER, C., C. ASCHER, P. CROCOLL und G. F. TROMMER: *A Laser-Based Localization and Mapping System for Pedestrian Indoor Navigation With Loop-Detection Capabilities*. In: *Proceedings of the National Navigation Conference NAV 10*, S. 1–16, London, 2010.
- [114] KIESEL, S., P. CROCOLL, T. MERKEL und G. F. TROMMER: *Filter Architectures for Non-Coherent GPS / INS Deep Integration*. In: *European Navigation Conference*, London, 2012.
- [115] POPP, M., P. CROCOLL, J. RUPPELT und G. F. TROMMER: *A Novel Multi Image Based Navigation System to Aid Outdoor-Indoor Transition Flights of Micro Aerial Vehicles*. In: *ION GNSS+*, S. 1595 – 1608, Tampa, FL, 2014.
- [116] POPP, M., J. SEIBOLD, P. CROCOLL, N. FRIETSCH und G. F. TROMMER: *Vision-Based Target Detection and Autonomous Target Approach Control for Unmanned MAVs*. In: *IMAV*, Braunschweig, 2012.

- [117] SCHOLZ, G., P. CROCOLL und G. F. TROMMER: *Nonlinear Inverse Dynamics Used to Control a Quadcopter with Tilttable Rotors*. In: *ION International Technical Meeting*, Dana Point, CA, 2015.
- [118] SEIBOLD, J., P. CROCOLL, N. FRIETSCH und G. F. TROMMER: *Mission planning tool for global path planning and local obstacle avoidance*. In: *UAS Conference*, Paris, 2011.
- [119] SEIBOLD, J., P. CROCOLL und G. F. TROMMER: *Cooperative Mission Planning for Search and Rescue Applications for Micro Aerial Vehicles Based on Artificial Potential Fields*. In: *European Navigation Conference*, Wien, 2013.

Betreute Arbeiten

- [120] CASELITZ, T.: *Wiedererkennung von Merkmalen in Laserdaten zur Navigationsstützung in Innenräumen*. Diplomarbeit, Karlsruher Institut für Technologie, 2013.
- [121] FOO, J. J.: *Modellierung und Behandlung von Sensoreffekten an Bord eines MAV in Boden- und Gebäudenähe*. Masterarbeit, Karlsruher Institut für Technologie, 2011.
- [122] FU, S.: *Entwicklung eines multisensoriellen Navigationssystems mit Stützung durch ein online identifiziertes Bewegungsmodell*. Diplomarbeit, Karlsruher Institut für Technologie, 2012.
- [123] HETTICH, B.: *Entwicklung und Test eines Indoor-Navigationssystems basierend auf Laser- und Kameramessungen*. Masterarbeit, Karlsruher Institut für Technologie, 2013.
- [124] MÜLLER, K.: *Entwurf und Implementierung eines Systems zur Reduktion von windbedingten Fehlern in einem Bewegungsmodell-gestützten Navigationssystem*. Masterarbeit, Karlsruher Institut für Technologie, 2015.
- [125] POLINSKI, V.: *Entwurf und Test eines Versuchsaufbaus zur Sensorfehler-Identifikation*. Bachelorarbeit, Karlsruher Institut für Technologie, 2011.
- [126] RUF, C.: *Laserbasiertes Scanmatching zur Bewegungsschätzung*. Studienarbeit, Karlsruher Institut für Technologie, 2013.

-
- [127] SIEBLER, B.: *Entwurf und Implementierung von Algorithmen zur Fehlerdetektion in Bewegungsmodell-gestützten Navigationssystemen*. Masterarbeit, Karlsruher Institut für Technologie, 2014.
- [128] TRAUTMANN, C.: *Laserbasierte Navigationsstützung für einen unbemannten Helikopter*. Studienarbeit, Karlsruher Institut für Technologie, 2012.
- [129] TÜRETKAN, V.: *Laufzeitoptimierung eines Pfadplanungs-Systems für einen unbemannten Helikopter*. Studienarbeit, Karlsruher Institut für Technologie, 2012.
- [130] ZABEL, O.: *Entwicklung eines Partikelfilter-basierten Navigationssystems für ein unbemanntes Fluggerät*. Diplomarbeit, Karlsruher Institut für Technologie, 2011.