

UNIVERSITÄT STUTTGART



INSTITUT FÜR STATIK UND DYNAMIK DER
LUFT- UND RAUMFAHRTKONSTRUKTIONEN
PROFESSOR DR.-ING. BERND-H. KRÖPLIN

STUDIENARBEIT

Harald von Rosenberg

**Entwicklung eines elektronischen
künstlichen Horizonts für kleine
unbemannte Fluggeräte**

Dezember 2005

Harald von Rosenberg , studienarbeit@vrosenberg.de

3DM-GX1 is a trademark of MicroStrain, Inc.

AVR and *ATmega* are registered trademarks of Atmel Corporation

I²C is a trademark of Philips Corps.

MATLAB and *Simulink* are registered trademarks of The MathWorks, Inc.

SPI is a trademark of Motorola Inc.

Vorwort

Ich danke Herrn Prof. Dr. Jörg Wagner (ISD) für die Ausgabe dieser Studienarbeit und seine Betreuung sowie für die Bereitstellung des institutseigenen Drehtisches.

Beim **STEINBEIS TRANSFERZENTRUM AERODYNAMIK, FLUGZEUG- UND LEICHTBAU** möchte ich mich für die Übernahme der Kosten dieses Projekts bedanken. Es stellte die Sensoren und Hardware bereit, die zum Aufbau des Prototyps notwendig waren. Danken möchte ich auch dem Zentrumsleiter Herrn Prof. Dipl.-Ing. Rudolf Voit-Nitschmann für den Messflug mit dem Piper-Flugzeug, Herrn Dipl.-Ing. Berthold Karrais für die Messflüge mit dem Flugzeug-Prototypen „Deltajet“, Herrn Dipl.-Ing. Markus Schlenker für die Beratungen bei Fragen zur Elektronik und Herrn Hans-Peter Otte für das Erstellen des Aluminium-Gehäuses, in das der IMU-Prototyp eingebaut wurde.

Firma **AUTOFLUG STEUERUNGS- UND SENSORTECHNIK GMBH** möchte ich für die Kalibriermessungen auf den temperaturgeregelten Drehtischen danken. Insbesondere Herren Hamann und Buchmann gilt mein besonderer Dank, die mir den Drehtisch vorbereitet und programmiert haben und mich bei den Messungen unterstützt haben.

Inhaltsverzeichnis

Vorwort	iii
Zusammenfassung	ix
Abkürzungsverzeichnis	x
Physikalische Einheiten und Größen	x
Mathematische Konventionen	xi
Aufstellung der mathematischen Symbole	xi
1 Einleitung	1
1.1 Übersicht über diese Arbeit	1
2 Projekt	3
2.1 Funktionsprinzip	3
2.2 Inertial-Measurement-Unit	4
2.3 Systemanforderungen	4
2.4 Referenzsysteme	5
2.4.1 Mechanischer Kreisel „Rosemount Vertical Gyro“	5
2.4.2 Microstrain 3DM-GX1 IMU	6
3 Sensoren	9
3.1 Drehratensensor	9
3.1.1 Elektrische und physikalische Eigenschaften	9
3.1.2 Fehlermodell	10
3.1.3 Gyroskop ADIS16100	11
3.2 Beschleunigungsmesser	12
3.2.1 Elektrische und physikalische Eigenschaften	12
3.2.2 Fehlermodell	13
3.3 Magnetfeldsensoren	14
3.4 Koordinatensysteme	15
4 Hardware	17
4.1 Analog-Digital-Wandler	18
4.2 Mikrocontroller	19
4.3 Interface-Board	20
4.3.1 Aufbau	20
4.3.2 Spannungsversorgung	23
4.3.3 Anschlüsse und Verbindungen	23
4.4 IMU-Board	24

5	Kalman-Filter	25
5.1	Systemmodelle	25
5.1.1	Systemmodell mit Euler-Winkeln	26
5.1.2	Systemmodell mit Quaternionen und Euler-Winkeln als Ausgang	28
5.1.3	Systemmodell mit Quaternionen	29
5.2	Verwendetes Systemmodell	30
5.3	Erweiterung des Systems	32
5.4	Analyse	33
5.4.1	Beobachtbarkeit	34
5.4.2	Steuerbarkeit	34
5.5	Weitere Modifikation und Adaption	35
5.6	Integrationsverfahren	36
5.7	Phasenverschiebung und Verzögerung	37
6	Mikrocontroller-Software	39
6.1	Software-Architektur	39
6.2	Software für Interface-Board	39
6.2.1	Gyroskope einlesen	39
6.2.2	Beschleunigungsmesser einlesen	40
6.2.3	Interrupt-Routinen	41
6.2.4	Hauptprogramm	43
7	Kalibrierung	45
7.1	Rauschen des AD-Wandlers	45
7.2	Kalibrierung der Beschleunigungsmesser	45
7.3	Kalibrierung der Gyroskope	49
7.4	Temperatur-Korrektur der Gyroskope	50
7.5	Temperatur-Korrektur der Beschleunigungsmesser	57
7.6	Ausrichtungsfehler	60
7.6.1	Gyroskope	61
7.6.2	Beschleunigungsmesser	62
7.7	Zentrifuge	63
7.8	Vibrationen	63
7.8.1	Gyroskope	63
7.8.2	Beschleunigungssensoren	63
8	Versuchsaufbau	65
8.1	Datenlogger	65
8.2	IMU-Prototyp und Microstrain-IMU	66
8.3	IMU-Prototyp, Microstrain-IMU und Kreisel	67
8.3.1	Messflug	67
8.3.2	Bodentest	69
8.3.3	Fahrversuch	69
9	Ergebnisse	71
9.1	Einfacher IMU-Algorithmus	71
9.2	Erweiterter IMU-Algorithmus	74
9.2.1	Bodentest	74
9.2.2	Messflug	74

9.2.3	Fahrversuch	76
9.2.4	Flugzeug-Prototyp	77
10	Diskussion	81
10.1	Hardwareaufbau	81
10.2	Sensoren	81
10.3	Filteralgorithmus	81
10.4	Ausblick	82
	Verzeichnis der Listings	83
	Literaturverzeichnis	85

Zusammenfassung

Diese Arbeit beschreibt die Entwicklung eines künstlichen elektronischen Horizonts für kleine unbemannte Fluggeräte. Basierend auf Gyroskopen und Beschleunigungssensoren in MEMS-Technologie wurde ein Hardware-Prototyp aufgebaut. Die eingesetzten Sensoren wurden im Prototyp vermessen und kalibriert.

Zur Fusion der verschiedenen Sensorsignale wurde ein Filteralgorithmus, basierend auf einem erweiterten KALMAN-Filter, implementiert, mit dem die aufgezeichneten Daten aus Messungen ausgewertet wurden. Die Ergebnisse wurden mit einem mechanischen Kreisel verglichen, der als Referenz diente. Zusätzlich wurde in die Tests und Versuche noch ein kommerziell erhältliches System, ebenfalls basierend auf MEMS-Sensoren, eingebunden.

Die Auswertung der Versuche ergab, dass die MEMS-Systeme nicht die Genauigkeit des mechanischen Systems erreichen. Vor allem in Phasen mit hoher Dynamik weichen sie teilweise erheblich von der Referenz ab.

Abkürzungsverzeichnis

ADC	Analog to D igital C onverter
BGA	B all G rid A rray
BNL	B asis- N ull- L age, d. h. $\phi = 0^\circ$ und $\theta = 0^\circ$
DGL	D ifferential g leichung
FS	F ull S cale
GND	G round (elektrisches Massepotential)
I ² C	I nter- I ntegrated C ircuit
IMU	I nertial M easurement U nit (Inertiale Messeinheit)
INS	I ntegrated N avigation S ystem
INT	ausgelöster I nterrupt im Microcontroller
ISR	I nterrupt S ervice R outine
LSB	L east S ignificant B yte
MEMS	M icro- E lectro- M echanical S ystem
MSB	M ost S ignificant B yte
PWM	P ulsweiten m odulation
RISC	R educed I nstruction S et C omputing
SMD	S urface M ounted D evice
SPI	S erial P eripheral I nterface
UART	U niversal A synchronous R eceiver T ransmitter
UAV	U nmanned A erial V ehicle (unbemanntes Flugobjekt)
VDC	Gleichspannung, V oltage D irect C urrent

Physikalische Einheiten und Größen

<i>A</i>	Ampere
<i>g</i>	Erdschwerebeschleunigung, $1g = -9,80665 \text{ m/s}^2$
<i>g</i>	Gramm
$^\circ$	Winkel in Grad
$^\circ\text{C}$	Temperatur in Grad Celsius
<i>Hz</i>	Hertz
<i>KiB</i>	K ilo b inary B yte
<i>m</i>	Meter
<i>mA</i>	Milli-Ampere
<i>rad</i>	Winkel im Bogenmaß
<i>s</i>	Sekunde
<i>V</i>	Volt

Mathematische Konventionen

\mathbf{x}	Zustand im System
$\hat{\mathbf{x}}$	Schätzung für \mathbf{x}
$\tilde{\mathbf{x}}$	Symbol im erweiterten System
\mathbf{x}^*	Wert, der zur Stützung in das System eingeht
$\dot{\mathbf{x}}$	Zeitliche Ableitung von \mathbf{x} ($\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$)
\mathbf{A}^T	Transponierte der Matrix \mathbf{A}
\mathbf{A}^{-1}	Inverse der Matrix \mathbf{A}
$\mathbf{I}^{3 \times 3}$	Einheitsmatrix der Dimension 3×3
$diag(\dots)$	Diagonalmatrix
$rank(A)$	Rang der Matrix \mathbf{A}
\bar{x}	Mittelwert über mehrere Werte von x
\bar{X}	Negation von X bezüglich des logischen Zustands

Aufstellung der mathematischen Symbole

Symbol	Beschreibung	Einheit
ϕ	Rollwinkel	°
θ	Pitchwinkel	°
ψ	Gierwinkel	°
ω_i	Drehratensignal des i. Gyroskops	°/s
Ω_i	Drehrate um Achse i, z. B. des Drehtisches	°/s
Ω	auf die IMU angewendeter Drehvektor mit $\Omega = [\Omega_p, \Omega_q, \Omega_r]^T$	
\mathbf{a}	Beschleunigungsvektor	g
a_i	Beschleunigungswert des i. Beschleunigungsmessers	g
b_i	Bias des i. Drehratensensors	
b_i^*	Statische Nullpunktkorrektur des i. Gyroskops	
\mathbf{C}	Steuerbarkeitsmatrix	
E	Erwartungswert	
\mathbf{f}	nichtlineare Dynamikfunktion der DGL	
\mathbf{F}	Jacobi-Matrix von \mathbf{f}	
g	Erdschwerebeschleunigung	
\mathbf{g}	nichtlineare Dynamikfunktion der DGL	
\mathbf{G}	Jacobi-Matrix von g	
\mathbf{H}	Jacobi-Matrix von h	
\mathbf{h}	nichtlineare Ausgangsfunktion des Systems	
\mathbf{I}	Einheitsmatrix	
\mathbf{K}	Verstärkungsmatrix für die Korrektur	
k	Diskreter Zeitschritt: $\mathbf{x}_k = \mathbf{x}(t = k)$, $\mathbf{x}_{k+1} = \mathbf{x}(t = k + \Delta t)$	
\mathbf{O}	Beobachtbarkeitsmatrix	
\mathbf{P}	Kovarianzmatrix des KALMAN-Filters	
p	Drehrate im IMU-Koordinatensystem um x-Achse	rad/s
\mathbf{Q}	Matrix der Intensitäten des Systemrauschens	

Fortsetzung auf nächster Seite ...

Symbol	Beschreibung	Einheit
q	Drehrate im IMU-Koordinatensystem um y-Achse	rad/s
q_i	Quaternionen	
\mathbf{R}	Matrix der Intensitäten des Messrauschens	
r	Drehrate im IMU-Koordinatensystem um z-Achse	rad/s
S_i	Wert in Stützstellen des Runge-Kutta Integrationsverfahrens	
t	Zeit	s
T	Temperatur	$^{\circ}C$
T_i	Temperatursignal des i. Gyroskops	
\mathbf{T}_{gyro}	Transformationsmatrix für Drehratenvektor	
\mathbf{T}_{acc}	Transformationsmatrix für Beschleunigungsvektor	
\mathbf{u}	Eingangsvektor in das System	
U	elektrische Spannung	V
v_i	Rauschen der Beschleunigungsmesser	
w_i	Rauschen des i. Drehratensensors	
\mathbf{x}	Zustandsvektor des Systems	
x	Längsachse des IMU-Koordinatensystems	
y	Querachse des IMU-Koordinatensystems	
\mathbf{z}	Ausgangsvektor des Systems	
z	Hochachse im IMU-Koordinatensystem	

1 Einleitung

Seit einigen Jahren steigt das Interesse und der Bedarf an kleinen unbemannten Fluggeräten stetig. Sie dienen der Überwachung und der Aufklärung, es bilden sich aber kontinuierlich weitere Einsatzmöglichkeiten heraus. Ihr Vorteil besteht in den geringen Abmessungen, flexiblen Einsatzmöglichkeiten und den niedrigen Stück- und Betriebskosten.

Damit das Flugzeug jedoch von einer Bodenstation aus bedient oder per Autopilot selbsttätig navigieren kann, muss das Fluggerät bzw. der in ihm eingesetzte Bordrechner und Flugregler die Lage des Fluggeräts im Raum kennen. Neben Informationen wie der Flughöhe oder auch der Fluggeschwindigkeit wird auch der Rollwinkel und der Nickwinkel (Pitchwinkel) des Fluggeräts ausgewertet. Diese Arbeit beschäftigt sich mit einer Sensorikeinheit, die auf elektronischem Weg den Rollwinkel und den Pitchwinkel ermittelt und bereitstellt. Die Darstellung dieser beiden Größen wird auch als „künstlicher Horizont“ bezeichnet.

Ziel dieser Arbeit ist die Entwicklung eines künstlichen Horizonts, der elektronisch, ohne mechanisch bewegte Teile, die Lage des Fluggeräts ermittelt. Die Entwicklung schließt sowohl die Hardware und Sensoren mit der Kalibrierung, als auch den Filteralgorithmus und die Software ein. Diese Studienarbeit entstand in Kooperation mit dem **STEINBEIS TRANSFER-ZENTRUM AERODYNAMIK, FLUGZEUG- UND LEICHTBAU (STZAF)**, an dem die Hardware aufgebaut wurde. Außerdem stellte das Steinbeis-Zentrum auch ein Flugzeug und Flugmodell für Messflüge zur Verfügung.

Die in dieser Arbeit entwickelte Software und Filteralgorithmen sollen im Postprocessing aus aufgezeichneten Messwerten die Ergebnisse berechnen. Dabei soll die Software derart gestaltet sein, dass eine Portierung auf ein Echtzeitsystem, das die Messwerte direkt zur Laufzeit auswertet, mit nur geringen Modifikationen möglich ist. Die Auswertung im Postprocessing bietet den Vorteil, verschiedene Filtertypen und Einstellungen über den gleichen Datensatz laufen lassen zu können und das Ergebnis daraufhin zu analysieren.

1.1 Übersicht über diese Arbeit

Im folgenden [Kapitel 2](#) stelle ich das Projekt und die Aufgabenstellung vor. Außerdem werden die Systeme, die als Referenz für den künstlichen Horizont dienten, beschrieben.

In [Kapitel 3](#) gehe ich auf die verwendeten Sensoren ein, die die messtechnische Grundlage für den künstlichen Horizont bilden.

Das [Kapitel 4](#) führt dann schließlich zum vollständigen Hardwareaufbau, der für die dann folgende Kalibrierung und die durchgeführten Versuche diente.

Einen eher theoretischen Einschub über den verwendeten Filteralgorithmus bildet das [Kapitel 5](#). Es werden verschiedene mögliche Versionen des Algorithmus vorgestellt und schließlich der verwendete Algorithmus beschrieben und untersucht.

1 Einleitung

Die Software, die auf dem Zielsystem läuft, wird in [Kapitel 6](#) entwickelt. Es ist das Bindeglied zwischen dem in [Kapitel 5](#) vorgestellten Algorithmus und der in [Kapitel 4](#) beschriebenen Hardware.

Das [Kapitel 7](#) widmet sich den Methoden zur Kalibrierung des künstlichen Horizonts und stellt die gewonnenen Ergebnisse dar.

Nach erfolgter Kalibrierung werden in [Kapitel 8](#) die Versuchsaufbauten beschrieben. Der künstliche Horizont wurde in realen Einsatzszenarien betrieben, und in [Kapitel 9](#) sind die gewonnenen Resultate aufgeführt, deren Qualität schließlich in [Kapitel 10](#) diskutiert wird.

In ?? finden sich dann die MATLAB-Programme, die für das Postprocessing im Einsatz waren. ?? stellt eine Methode zur Orthogonalisierung der Sensoren vor, und in ?? ist schließlich der Quellcode der Programme aufgeführt, die auf dem Mikrocontroller laufen.

2 Projekt

Die bisherige technische Entwicklung hat eine Vielzahl an Realisierungen und Funktionsweisen für künstliche Horizonte geliefert. Die ersten Systeme waren mechanische Gebilde, in denen das Verharren von rotierende Massen in einem kardanischen Rahmen ausgenutzt wurde. Die Verdrehung des kardanischen Rahmens lieferte somit die Lage im Raum.

Der nächste Schritt bestand darin, die rotierende Masse zu ersetzen und stattdessen eine Plattform im kardanischen Rahmen aufzuhängen, die elektronisch stabilisiert wird.

Die neuesten Verfahren verzichten auf diesen kardanischen Rahmen, wobei die Sensoren fest mit dem Fluggerät verbunden werden. Die Ausrichtung der Sensoren geschieht nun nicht mehr mechanisch, sondern nur noch per Software, weshalb dieses Verfahren als „*Strap-Down-System*“ bezeichnet wird. Die Sensoren bewegen sich körperfest mit dem Flugzeug mit, und alle makro-mechanischen¹ Bauteile können ersetzt werden. Diese Arbeit beschreibt die Entwicklung eines nach diesem Prinzip funktionierenden künstlichen Horizonts.

2.1 Funktionsprinzip

Die Lagewinkel werden durch Integration der Ausgangssignale von drei Drehratensensoren gebildet, die mit der Plattform fest verbunden sind. Die Umrechnung auf die Winkel erfolgt nur durch den in der Software realisierten Algorithmus.

Da jedoch die Drehratensensoren ein Messrauschen aufweisen und mit der Zeit vom Nullpunkt wegedriften, ist eine Korrektur erforderlich. Denn andernfalls liefert die Integration schon nach kurzer Zeit falsche Ergebnisse, die um Größenordnungen neben der tatsächlichen räumlichen Lage liegen. Dieser Fehler wird zusätzlich dadurch verstärkt, dass das Ausgangssignal der Sensoren von der sich verändernden Temperatur am Sensor abhängt. Zusätzlich gibt es eine weitere Reihe von Quellen, die das Ausgangssignal der Drehratensensoren verfälschen.

Damit das rechnerisch gewonnene Ergebnis langzeitstabil ist, müssen die Drehratensensoren gestützt werden. Diese Stützung findet jedoch nicht direkt im Sensor statt, sondern wird per Software im Algorithmus vorgenommen.

Zur Stützung benötigt man also eine Signalquelle, die langzeitstabil und nicht von der Drehrate abhängig ist. Zusätzlich muss die Stützung ortsungebunden zur Verfügung stehen, damit der künstliche Horizont weltweit und auch in größeren Höhen ohne äußeren Einfluss funktioniert.

Alle geforderten Eigenschaften erfüllt das Erdschwerefeld, und die Erdanziehung kann in den Einsatzräumen, in denen der künstliche Horizont eingesetzt wird, als konstant angesehen werden. Durch das Messen der Erdschwerebeschleunigung kann ein Lotvektor gewonnen werden.

¹Makro-mechanisch bedeutet, dass ein Bauteil mechanische Bewegungen ausführt, die makroskopisch messbar sind, wie sie z. B. bei einem Pendel auftreten.

2 Projekt

Die Richtung dieses Lotvektors bezüglich des körperfesten Flugzeug-Koordinatensystems gibt dann die Lage des Fluggeräts im Raum an.

Es gilt nun, einen Filteralgorithmus zu entwickeln, der die genauen und absoluten, aber nur kurzzeitstabilen Signale der Drehratensensoren mit den Sensoren, die die Erdschwerebeschleunigung messen, fusioniert. Dabei tritt jedoch das Problem auf, dass die Beschleunigungssensoren nicht nur die statische Erdschwerebeschleunigung, sondern auch dynamische Beschleunigungen erfassen und nicht zwischen diesen beiden Anregungen unterscheiden können.

Ziel dieser Arbeit ist es, ein Interface-Board mit den kalibrierten und korrigierten Sensoren und einen KALMAN-Filteralgorithmus zur Fusion und Auswertung der Eingangssensoren zu entwickeln.

2.2 Inertial-Measurement-Unit

Eine Inertial-Measurement-Unit (IMU) vereint alle oben aufgeführten Komponenten. Zusätzlich sind in der IMU die Signalaufbereitung und die Signalverarbeitung integriert. In dieser Arbeit wird auch das Gesamtsystem aus Sensoren, Hardware und der darauf laufenden Software als IMU bezeichnet. In der Literatur hat sich für dieses Gerät der Begriff *Attitude Heading Reference System* (AHRS) und *Integrated Navigation System* (INS) gebildet. Da der hier vorgestellte Prototyp jedoch nur als künstlicher Horizont ohne die Messung und Ausgabe des Gierwinkels fungiert, und auch keine navigatorischen Aufgaben bearbeitet werden, wird der Begriff *IMU* beibehalten.

2.3 Systemanforderungen

Typische unbemannte Fluggeräte sind häufig umgebaute oder vergrößerte Modellflugzeuge. Um eine aerodynamische Form zu erhalten, wird der Rumpfkörper häufig in seinen Abmessungen klein gehalten. Andererseits wird ein Großteil des Volumens für Treibstofftanks reserviert, um lange Missionsflüge zu ermöglichen. Aus diesen Gründen steht für die Elektronik an Bord nur wenig Raum zur Verfügung. Deshalb muss die zu entwickelnde IMU möglichst klein und leicht (Größenordnung 100 Gramm) sein, damit sie problemlos in UAVs eingesetzt werden kann.

Da die Kapazität der Batterien sehr beschränkt ist, bzw. für die Elektronikkomponenten aus Gewichtsgründen nur kleine Batterien eingesetzt werden, muss ein weiteres Ziel eine geringe Leistungsaufnahme sein. Durch den Einsatz weniger, hochintegrierter Schaltkreise und nicht-mechanischer Sensoren, die keine motorisch bewegten Teile besitzen, kann dieses Ziel erreicht werden.

Die mögliche Ausgaberate der Lagewinkel sollte mindestens 50 Hz betragen. Für den Einsatz in kleinen, wendigen UAVs werden aber deutlich höhere Raten gefordert. Um diese höheren Raten zu erreichen, müssen jedoch sowohl die Sensoren eine solche unterstützen, als auch der Mikrocontroller schnell genug sein, um in jedem Taktfenster den IMU-Algorithmus komplett abzuarbeiten.



Abbildung 2.1: Vertical Gyroscope von **ROSEMOUNT AEROSPACE INC.**

2.4 Referenzsysteme

Zum Einstellen des Filteralgorithmus wurde ein Referenzsystem herangezogen. Der mechanische Kreisel, der im folgenden Abschnitt vorgestellt wird, lieferte bei Messfahrten und Messflügen das Referenzsignal, das mit dem IMU-Prototyp erreicht werden sollte.

Neben dem Kreisel wurde auch noch eine andere kommerzielle IMU in die Versuche eingebunden. Die im [Unterabschnitt 2.4.2](#) vorgestellte IMU verwendet, wie der hier behandelte IMU-Prototyp, auch MEMS-Sensoren auf den Drehachsen und für die Beschleunigungen.

2.4.1 Mechanischer Kreisel „Rosemount Vertical Gyro“

Der „Vertical Gyro VG24-0825-1“ von **ROSEMOUNT AEROSPACE INC.** ist ein mechanischer Kreisel, wie er in großen UAVs eingesetzt wird. Ein elektrisch angetriebenes Kreiselsystem richtet sich beim Anschalten automatisch aus, und über ein mechanisches Abgriffsystem kann der Roll- und Nickwinkel bestimmt werden. [Abbildung 2.1](#) zeigt den Kreisel, und in [Tabelle 2.1](#) sind die wichtigsten Daten aufgelistet.

Der Kreisel gibt die Lagewinkel nicht direkt aus, er verstellt mit dem Abgriffsystem nur die Winkelpositionen zweier Potentiometer. Legt man an den einen äußeren Anschluss des Potentiometers eine Referenzspannungsquelle U_{ref} und den anderen an Masse an, so fungiert der Mittelabgriff als Spannungsteiler, dessen Spannung direkt proportional zum Auslenkungswinkel bezüglich der Roll- bzw. Nickachse ist. Die Gleichungen (2.1) und (2.2) beschreiben das Systemverhalten des Kreisels, woraus sich bei Messung der analogen Spannung direkt der Lagewinkel ergibt.

Tabelle 2.1: Technische Daten des ROSEMOUNT-Kreisels

Messbereich	Pitch: $\pm 60^\circ$ Roll: $\pm 90^\circ$
Abgreifsystem	Potentiometer $1500 \Omega \pm 100 \Omega$
Messauflösung	unbegrenzt
Genauigkeit	allgemein: $\pm 2,0^\circ$ Pitch: $\pm 0,5^\circ$
Statischer Fehler	Pitch: $\pm 1,25\%$ FS bei 0° , $\pm 2,08\%$ FS bei 60° Roll: $\pm 0,83\%$ FS bei 0° , $\pm 1,67\%$ FS bei 90°
Temperaturbereich	-54°C bis $+74^\circ\text{C}$
Betriebsspannung	28 VDC
Stromaufnahme	1,0 A beim Start, 0,5 A im Betrieb
Masse	1360 g
Shock Limit	30 g (ausgeschaltet); 15 g (im Betrieb)



Abbildung 2.2: 3DM-GX1 IMU von MICROSTRAIN INC.

$$U_{Roll} = \frac{U_{ref}}{2} \left(1 + \frac{\phi}{90^\circ} \right) \quad -90^\circ \leq \phi \leq +90^\circ \quad (2.1)$$

$$U_{Pitch} = \frac{U_{ref}}{2} \left(1 + \frac{\theta}{60^\circ} \right) \quad -60^\circ \leq \theta \leq +60^\circ \quad (2.2)$$

Dieser Kreisels dient bei der Auswertung und beim Einstellen des in Kapitel 5 vorgestellten Filters als Referenz, dessen Werte es zu erreichen gilt.

2.4.2 Microstrain 3DM-GX1 IMU

Neben dem oben vorgestellten mechanischen Kreisels wurde in die Versuche und Auswertung auch noch eine IMU der Firma MICROSTRAIN INC. einbezogen, die am STZAFL bisher für

Flugversuche und Messflüge mit unbemannten Fluggeräten eingesetzt wird.

Die „3DM-GX1“-IMU von **MICROSTRAIN INC.** vereint drei orthogonale Drehratensensoren, drei orthogonale Beschleunigungsmesser, drei orthogonale Magnetfeldsensoren, Temperaturfühler, Multiplexer, 16-bit-AD-Wandler und einen Mikrocontroller, um die Lage im Raum sowohl in statischer als auch in dynamischer Umgebung auszugeben. Alle Sensoren werden analog ausgelesen und mittels des eingesetzten Multiplexers und 16bit-AD-Wandlers dem Mikrocontroller zugeführt. Die [Abbildung 2.2](#) zeigt die kompakte MICROSTRAIN-IMU, die in ein mit Montagelaschen versehenes Plastikgehäuse eingebaut ist.

Wie auch die im Rahmen dieser Arbeit entwickelte IMU, basiert die MICROSTRAIN-IMU auf mikromechanischen Sensoren. Damit bietet sie sich besonders als Vergleichssystem an, da die arttypischen Fehler und Probleme, die sich aus der Verwendung der mikromechanischen Sensoren und deren späteren Fusion ergeben, in beiden Systemen wiederfinden müssten. Außerdem bietet die MICROSTRAIN-IMU durch die Ausgabe der skalierten Sensor-Rohwerte die Möglichkeit, den zu entwickelnden IMU-Algorithmus anhand einer bestehenden Hardware-Plattform zu testen und zu tunen. Die „3DM-GX1“-IMU verwendet zur Berechnung der Lagewinkel ein von MICROSTRAIN patentrechtlich geschütztes Verfahren, das der Hersteller nicht näher spezifiziert.

Weitere vorteilhafte Eigenschaften zeichnen diese IMU aus:

- Implementierte Temperatur- und Drift-Korrektur in der Software für alle 9 Sensoren über den vollen Temperaturbereich
- Jede IMU wurde auf einer Roboteranlage kalibriert.
- Software-Korrektur der Ausrichtungsfehler der Sensoren
- Korrektur der Fehler, die durch die Beschleunigungsabhängigkeit der Drehratensensoren entstehen.

Die wichtigsten technischen Daten sind in [Tabelle 2.2](#) aufgeführt.

Die MICROSTRAIN-IMU gibt an der seriellen RS-232-Schnittstelle je nach Eingangsbefehl verschiedene Ausgabeformate zurück:

- Die Rohwerte (*RAW-Data*) der Sensoren direkt vom AD-Wandler ohne jegliche Verarbeitung oder Filterung
- Die aufbereiteten Rohwerte (*Vectors*), die schon skaliert, orthogonalisiert und temperaturkorrigiert sind. Diese Rohwerte eignen sich gut, um sie einem eigenen Algorithmus zuzuführen.
- Die IMU berechnet intern über einen von MICROSTRAIN patentierten Algorithmus auch *Euler*-Winkel, die direkt den Rollwinkel ϕ , Pitchwinkel θ und Gierwinkel ψ angeben.
- Außerdem kann die IMU die Lagewinkel als Quaternionen ausgeben.

Die Formate *Euler* und *Vectors* wurden bei den Versuchen aufgezeichnet. Die aufbereiteten Rohwerte wurden dem hier vorgestellten Filteralgorithmus zugeführt, die Qualität der MICROSTRAIN-Eulerwinkel sind in den in [Kapitel 9](#) eingefügten Diagrammen ersichtlich.

Tabelle 2.2: Technische Daten der MICROSTRAIN-IMU

Orientation Range	360 °, all axes (Matrix, Quaternion modes)
Sensorbereich	Gyros: ± 300 °/s; Beschleunigungsmesser: $\pm 2,5$ g FS; Magnetometer: $\pm 1,2$ Gauss FS
ADC Auflösung	16 bit
Accelerometer Nonlinearity	0,2 %
Accelerometer Bias Stability	0,010 g
Gyro Nonlinearity	0,2 %
Gyro Bias Stability	0,7 °/s
Magnetometer Nonlinearity	0,4 %
Magnetometer Bias Stability	0,010 Gauss
Orientation Auflösung	< 0,1 ° minimum
Repeatability	0,20 °
Genauigkeit	$\pm 0,5$ ° typical for static test conditions ± 2 ° typical for dynamic (cyclic) test conditions and for arbitrary orientation angles
Output Modes	Matrix, Quaternion, Euler-Winkel und Vektro aus 9 skalierten Sensorwerten mit Temperatur
Digital Outputs	Serial RS-232
Digital Output Rates	100 Hz für Euler, Matrix, Quaternion 350 Hz nur für 9 orthogonale Sensorwerte
Serielle Datenrate	19.2/38.4/115.2 Kbaud
Versorgungsspannung	5,2 VDC – 12 VDC
Stromaufnahme	65 mA
Betriebstemperatur	-40 bis +70 °C mit Gehäuse; -40 to +85 °C ohne Gehäuse
Abmessung	mit Gehäuse: 65 x 90 x 25 mm ohne Gehäuse: 42 x 40 x 15 mm;
Masse	74,6 g mit Gehäuse, 25,8 g ohne Gehäuse
Shock Limit	1000 g (ausgeschaltet); 500 g (im Betrieb)

3 Sensoren

3.1 Drehratensensor

Drehratensensoren (Gyroskope) werden benutzt, um die Winkelgeschwindigkeit, mit der sich ein Objekt dreht, zu messen. Die Drehbewegung wird bezüglich einer festgelegten Achse gemessen. Aufgrund der guten Erfahrung mit Sensoren von **ANALOG DEVICES INC.** (a) fiel die Entscheidung auf das Gyroskop „ADXRS300“ von Analog.

Für den IMU-Prototyp werden drei ADXRS300-Sensoren verwendet, die zueinander bi-orthogonal montiert sind. Die Sensoren sind der leichteren Handhabung wegen auf sogenannten *ADXRS300-Evaluation-Boards* montiert. Es handelt sich dabei jeweils um einen ADXRS300-Sensor, der auf einem Standard-IC-Stiftsockel montiert ist. Zusätzlich sind die benötigten Kondensatoren schon fest aufgelötet. Somit ist der Sensor direkt betriebsbereit und benötigt keine weitere Beschaltung auf der Platine. Die [Abbildung 3.1](#) zeigt einen Sensor vor der Montage im IMU-Prototypen.

Der Sensor basiert auf der Micro-Electro-Mechanical-System-Technologie (MEMS). Die Geräte in MEMS-Technik zeichnen sich dadurch aus, dass sowohl der miniaturisierte Winkelgeschwindigkeits-Sensor, die Filterschaltungen, die Signalaufbereitung und weitere benötigte Elektronik in einem kleinen Gehäuse integriert sind. Dadurch kommt der Sensor mit einer minimalen Anzahl an externen Elektronikbauteilen aus, liefert aber trotzdem präzise Signale.

Der Sensor macht sich beim Messen der Drehrate die auftretende Coriolisbeschleunigung auf eine Masse zu nutzen. **GEEN UND KRAKAUER (2003)** beschreiben die Funktionsweise des Sensors und die Vorkehrungen, dass möglichst nur Drehraten in der gewünschten Achse gemessen werden. Im Sensor ist eine schwingende Masse an Federn aufgehängt, die aufgrund einer Führung nur entlang einer Richtung schwingen kann. Die Auslenkung der Masse wird über ein kapazitives Messsystem abgegriffen, und die integrierte Signalauswertung generiert daraus das Ausgangssignal, das direkt der anregenden Drehrate entspricht.

3.1.1 Elektrische und physikalische Eigenschaften

Der Sensor ist für einen Messbereich von $\pm 300^\circ/s$ ausgelegt. Der effektiv nutzbare Messbereich ist jedoch größer. In [Abschnitt 7.3](#) wird ein nutzbarer Bereich von $\pm 480^\circ/s$ bei der hier vorgestellten Konfiguration ermittelt. Der Sensor gibt sowohl die Drehrate als auch die Temperatur als analoge Spannung aus. Damit die Informationen dieses Sensors im Mikrocontroller verarbeitet werden können, müssen die analogen Spannungen zuvor von einem AD-Wandler in ein digitales Format überführt werden. In [Abschnitt 4.1](#) werden die Anforderungen an den AD-Wandler noch genauer spezifiziert und der verwendete Typ vorgestellt.

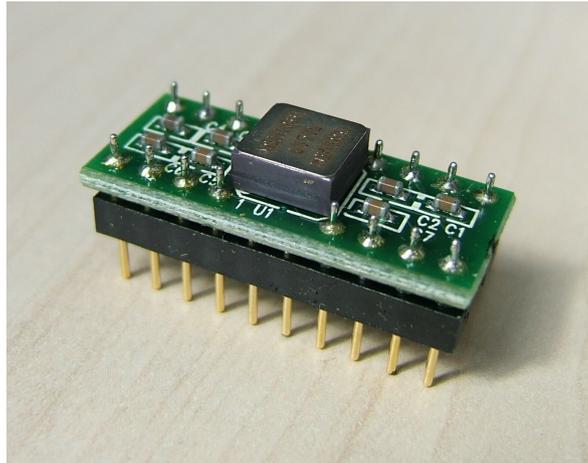


Abbildung 3.1: ADXRS300 Gyroskop aufgelötet auf Evaluationboard

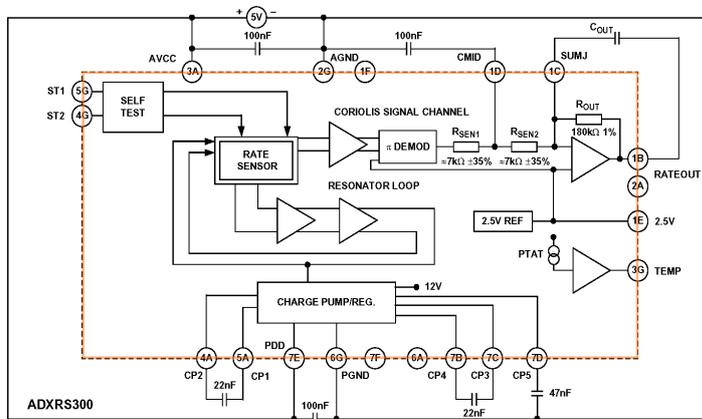


Abbildung 3.2: Blockschaltbild des ADXRS300, Quelle: [Datenblatt zum ADXRS300](#)

In der [Tabelle 3.1](#) sind die wichtigsten technischen Daten des Sensors aufgeführt. Die Daten wurden dem [Datenblatt zum ADXRS300](#) entnommen. Die Bandbreite ist durch das Evaluationboard mit 40 Hz fest vorgegeben. In Bezug auf das gewünschte Ausgangssignal und das Rauschverhalten stellt diese Einstellung einen guten Kompromiss für den hier vorgestellten Einsatzzweck dar.

Das Blockschaltbild des Sensors ist in [Abbildung 3.2](#) wiedergegeben.

3.1.2 Fehlermodell

Obwohl das Gyroskop schon im Werk kalibriert wird ([GEEN UND KRAKAUER, 2003](#)), weist der Sensor trotzdem eine Reihe von Fehlern auf, die das Ausgangssignal verfälschen.

Die größten Fehler verursacht die Temperaturabhängigkeit des Sensors und die Drift des Nullpunktes (②). Auch die Nichtlinearität des Sensors muss berücksichtigt werden. Die Ausrichtungsfehler und Einflüsse außerhalb der Messachse (④) addieren sich zusätzlich zum Signal.

Tabelle 3.1: Technische Daten des Gyroskops ADXRS300

Größe	Wert	Einheit
Betriebsspannung	5,0	V
Stromaufnahme	max. 8,0	mA
Messbereich	± 300	$^{\circ}/s$
Nullpunkts-Spannung Drehrate	2,5	V
Ausgangssignal Drehrate	5,0	$mv/^{\circ}/s$
Nichtlinearität	0,1	% FS
Beschleunigungseinfluss	0,2	$^{\circ}/s/g$
Resonanzfrequenz des Sensors	14	kHz
Temperatursignal bei 298°K	2,5	V
Ausgangssignal Temperatur	8,4	$mv/Kelvin$
Temperaturbereich	min. -55 bis max. +125	$^{\circ}C$
Abmessungen	$7 \times 7 \times 3$	mm^3
Gehäusetyp	BGA	
Masse	0,5	g

Auch die Einflüsse der Erdschwerebeschleunigung und Störungen, bedingt durch Beschleunigungen (①), überlagern den Ausgang. Die interne Referenzspannungsquelle und die Störeinflüsse durch die eingebaute Ladungspumpe (③) sind weitere Fehlerquellen. Schließlich enthält das Drehraten-Ausgangssignal auch noch ein zufälliges Rauschen (⑤). Ferner tritt auch ein Rauschen des Temperatur-Ausgangssignals auf, das allerdings vernachlässigt werden kann, da das Temperatursignal wie in [Abschnitt 7.4](#) gezeigt, nur geringen Einfluss auf das Drehratensignal hat.

Zusammenfassend kann das Ausgangssignal ω_i des Drehratensensors formuliert werden zu:

$$\omega_i = \underbrace{\mathbf{f}\left([a_x, a_y, a_z]^T\right)}_{\textcircled{1}} + \underbrace{\mathbf{f}(\Omega_i, T_i, b_i)}_{\textcircled{2}} + \underbrace{\mathbf{f}(V_{ref}, V_{Pumpe})}_{\textcircled{3}} + \underbrace{\mathbf{f}(\Omega_j, \Omega_k)|_{j,k \neq i \wedge j \neq k}}_{\textcircled{4}} + \underbrace{w_i}_{\textcircled{5}} \quad (3.1)$$

Im Verlauf dieser Arbeit werden die Komponenten ② und ④ des Fehlerterms in (3.1) genauer untersucht und deren Kompensation angestrebt.

3.1.3 Gyroskop ADIS16100

Als Alternative zum *ADXRS300* bietet sich der neue *ADIS16100* von **ANALOG DEVICES INC.** (a) an. Da dieser Sensor zum Zeitpunkt der Arbeit noch nicht verfügbar war, musste leider der oben aufgeführte Sensor benutzt werden. Der *ADIS16100* besitzt gegenüber dem *ADXRS300* bei gleichen technischen Daten folgende Vorteile:

- Der Sensor gibt seine Signale nicht als analoge Spannung aus, sondern direkt in einem digitalen Format. Der beim *ADXRS300* benötigte AD-Wandler (siehe [Abschnitt 4.1](#)) kann also entfallen. Dies senkt nicht nur die Kosten, sondern führt auch noch zu einer kleineren Baugröße der IMU.

3 Sensoren

- Der *ADIS16100* besitzt eine SPI-Schnittstelle, die kompatibel zum verwendeten Mikrocontroller ist.
- Der *ADIS16100* lässt sich einfacher einlöten, da er im Gegensatz zum *ADXRS300* kein BGA-Gehäuse¹ besitzt.

3.2 Beschleunigungsmesser

Beschleunigungssensoren werden benutzt, um das statische Erdschwerefeld zu messen. Außerdem können mit ihnen dynamische Beschleunigungen und Stöße gemessen werden. Der Sensor kann jedoch nicht zwischen der Beschleunigung, die die Erdanziehung erzeugt, und einer Beschleunigung, die das Fluggerät durch Luftturbulenzen, Schwingung oder Flugmanöver erfährt, unterscheiden.

Wie schon bei den Gyroskopen wurde auch bei den Beschleunigungssensoren auf Produkte von **ANALOG DEVICES INC.** (a) zurückgegriffen. Es wurden zwei Beschleunigungsmesser vom Typ „ADXL202E“ für den Prototypen vorgesehen. Jeder Sensor erfasst zwei orthogonal stehende Messachsen gleichzeitig, weshalb nur zwei Sensoren für den Prototyp benötigt werden, um alle drei Achsen zu erfassen. Dabei wurde ein horizontal montierter Sensor für die x und y-Achse vorgesehen, ein zweiter, der die z-Achse erfasst, wurde auf einem 90°-IC-Sockel montiert.

3.2.1 Elektrische und physikalische Eigenschaften

Der ADXL202E, dessen Blockschaltbild in [Abbildung 3.3](#) wiedergegeben ist, verfügt je Achse über einen Messbereich von $\pm 2 g$. Er besitzt einen PWM-Ausgang, dessen Signalform in [Abbildung 3.4](#) dargestellt ist. Das PWM-Signal besitzt bei der Weiterverarbeitung durch einen Mikrocontroller eine Reihe von Vorteilen:

- Es ist kein separater AD-Wandler zum Einlesen nötig (und der ATmega-Mikrocontroller hätte mit seinem eingebauten 10-bit AD-Wandler auch keine ausreichende Auflösung).
- Da das Signal nur zwei logische Zustände kennt, werden keine analogen Filter oder sonstige Schaltungen auf dem Interfaceboard benötigt.
- Die Länge des elektrischen Signalpfades ist unkritisch, solange die logischen Zustände richtig zugeordnet werden können.
- Das PWM-Signal kann direkt vom Mikrocontroller verarbeitet werden.
- Durch die Verwendung von Interrupts innerhalb des Mikrocontrollers beim Einlesen werden im Einleseprogramm keine speziellen Einlese-Routinen benötigt. Ferner muss nicht erst explizit eine Einleseroutine ausgeführt werden, sondern die aktuellen Messwerte liegen permanent vor.

¹BGA = Ball Grid Array – ist eine Gehäuseform von Integrierten Schaltungen, bei der die Anschlüsse für SMD-Bestückung kompakt auf der Unterseite des Chips liegen. Die Anschlüsse sind kleine Bälle aus Lötzinn, die nebeneinander in einem Feld (array) aus Spalten und Zeilen stehen. Diese Bälle werden beim Löten in einem Lötöfen aufgeschmolzen und verbinden sich mit dem Kupfer der Leiterplatte.

Aus den beiden Zeiten T_a und T_b des PWM-Signals läßt sich die gemessene Beschleunigung gemäß [Gleichung 3.2](#) berechnen.

$$a[g] = \frac{\left(\frac{T_a}{T_b} - \text{Offset}\right)}{\text{Skalierungsfaktor}} \quad (3.2)$$

Der Offset beträgt ungefähr 0,5 und der *Skalierungsfaktor* ca. 0,125 – wobei die genauen Werte erst bei der Kalibrierung in [Kapitel 7](#) ermittelt werden. Der Sensor ist so eingestellt, dass bei keiner einwirkenden Beschleunigung ($0g$) gilt: $\frac{T_a}{T_b} = 0,5$, T_a also halb so lang dauert, wie T_b .

Weitere wichtige technische Daten sind in der [Tabelle 3.2](#) zusammengestellt, deren Einträge aus dem [Datenblatt zum ADXL202E](#) entnommen sind.

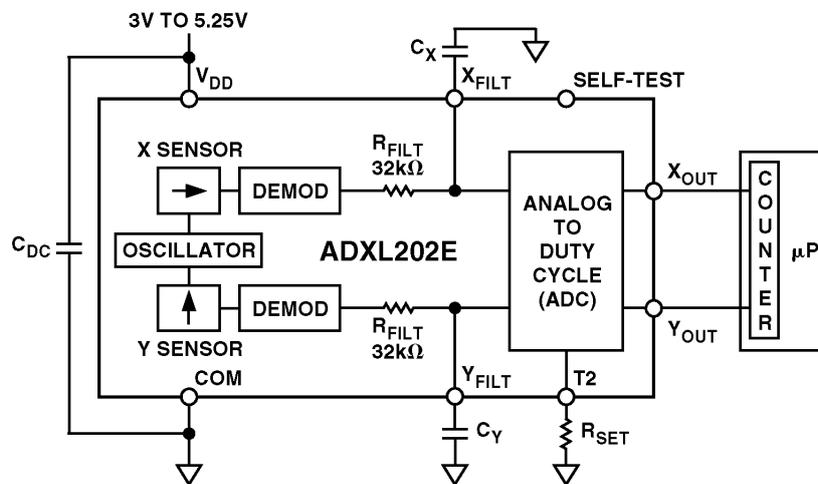


Abbildung 3.3: Blockschaltbild des ADXL202E-Sensors, Quelle: [Datenblatt zum ADXL202E](#)

3.2.2 Fehlermodell

Die bei MEMS-Sensoren typischen Fehlerquellen wie Temperaturabhängigkeit und Nichtlinearität finden sich auch bei den Beschleunigungssensoren. Auch die externen Bauteile, die die Bandbreite und Empfindlichkeit einstellen, haben temperaturabhängigen Einfluss auf das Sensorausgangssignal. Weitere Fehlerquellen stellen auch hier die Ausrichtungsfehler und die

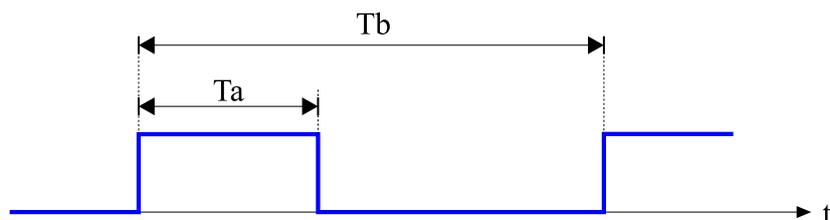


Abbildung 3.4: PWM-Signal des ADXL202E

Tabelle 3.2: Technische Daten des Beschleunigungssensors ADXL202E

Größe	Wert	Einheit
Versorgungsspannung	3 – 5,25	V
Stromaufnahme	max. 1,0	mA
Messbereich	± 2	g
Nichtlinearität	0,2	% FS
Auflösung bei 60 Hz	2	mg
Resonanzfrequenz	10	kHz
Temperaturdrift des PWM	$\pm 0,5$	%
Temperaturdrift von Tb	50	ppm/°C
Offset durch Temperatur	2,0	mg/°C
Flankensteilheit PWM	200	ns
Schock-Limit	1000	g
Temperaturbereich	0 – 70	°C
Abmessungen	$5 \times 5 \times 2$	mm ³
Gehäusetyp	8-lead LLC	

Einflüsse außerhalb der Messachse dar. Ferner folgen aus Drehungen durch die Zentripetalkraft Beschleunigungen, wenn der Sensor nicht in der Drehachse montiert ist. Die Beschleunigungssensoren können jedoch nicht örtlich konzentriert montiert werden und haben in der Regel einen Abstand zum Drehpunkt des Flugzeuges, bzw. die Drehachsen des Flugzeugs schneiden sich nicht in einem Punkt.

Somit kann auch hier das Fehlermodell für den Sensor i ($i = x, y, z$) beschrieben werden:

$$a_i = \underbrace{\mathbf{f}(a_{i_{real}}, b_i, T_i)}_{\textcircled{1}} + \underbrace{\mathbf{f}(a_{j_{real}}, a_{k_{real}})|_{j,k \neq i}}_{\textcircled{2}} + \underbrace{\mathbf{f}([\Omega_p, \Omega_q, \Omega_r]^T)}_{\textcircled{3}} + \underbrace{v_i}_{\textcircled{4}} \quad (3.3)$$

Der Fehlerterm $\textcircled{1}$ mit den Komponenten reale Beschleunigung in Richtung i ($a_{i_{real}}$), Offsetwert b_i und Temperatur T_i wird in [Abschnitt 7.5](#) behandelt. Die Ausrichtungsfehler, die den Term $\textcircled{2}$ bilden, werden in [Unterabschnitt 7.6.2](#) untersucht. Die Drehratenabhängigkeit in Teil $\textcircled{3}$ wurde nicht untersucht. Die Komponente $\textcircled{4}$ ist das zufällige Rauschen des Sensors, wobei, bedingt durch die Berechnungsformel (3.2) gilt:

$$v_i = \frac{v_i(T a_i)}{v_i(T b_i)} \quad (3.4)$$

3.3 Magnetfeldsensoren

Viele IMUs verwenden zur Stützung des Kurswinkels einen Kompass. In Strapdown-Systemen wird dieser aus drei jeweils orthogonal installierten Magnetfeldsensoren gebildet. Die Magnetfeldsensoren geben einen Vektor des Magnetfeldes aus, der bei Projektion in das erdfeste Koordinatensystem die Nordrichtung bestimmt.

Die UAVs, die als Zielplattformen für diese IMU dienen sollen, verfügen über leistungsstarke Elektroantriebe. Aufgrund der hohen Ströme, die durch die elektrischen Antriebsmotoren und in den Kabelbäumen des Fluggeräts fließen, entstehen starke Magnetfelder. Die kleine räumliche Ausdehnung sowie die Unmöglichkeit der Abschirmung und Kompensation verhindern einen sinnvollen Einsatz in kleinen unbemannten Fluggeräten.

Der Gierwinkel könnte zwar allein durch Integration aus den Drehratensensoren berechnet werden, jedoch fehlt in diesem Fall die Information über den Anfangswert des Gierwinkels, und es gibt keine verwertbare Stützung, die die Fehler der Gyroskope langfristig korrigiert.

Bei Bedarf könnte der Gierwinkel über eine Einkoppelung des GPS-Kurswinkels gestützt werden. Dies soll aber nicht Gegenstand dieser Arbeit sein.

3.4 Koordinatensysteme

Zur Beschreibung des künstlichen Horizonts werden verschiedene Koordinatensysteme eingeführt. Das erdfeste Koordinatensystem (x_E, y_E, z_E) bildet das Inertialsystem. Das IMU-System (x, y, z) ist das körperfeste Koordinatensystem, das sich mit dem Fluggerät bewegt. Die [Abbildung 3.5](#) stellt die Beziehung dieser Koordinatensysteme zueinander dar. Die Lage der beiden Koordinatensysteme wird durch die drei Drehwinkel ϕ , θ und ψ beschrieben, die auch *Euler-Winkel* genannt werden. Sie können als drei aufeinanderfolgende Rotationen ausgedrückt werden: Einer Drehung um die z_E -Achse um den Winkel ψ , einer Drehung mit Winkel θ um die neue Achse y_E^* und schließlich einer Drehung um die resultierende x -Achse ([STOVALL, 1997](#)). Das Einhalten dieser Reihenfolge ist wichtig, da die Drehungen ψ , θ und ϕ nicht kommutativ sind.

Der Gierwinkel ψ wird in dieser Arbeit jedoch nicht berücksichtigt, da die hierfür notwendigen Magnetfeldsensoren aus den oben genannten Gründen nicht benutzt wurden.

In den vorangehenden Abschnitten wurde dargelegt, dass sowohl die Gyroskope als auch die Beschleunigungsmesser Ausrichtungsfehler aufweisen. Neben dem IMU-Koordinatensystem (x, y, z) erhält man folglich noch zwei weitere Koordinatensysteme: Das Koordinatensystem $(x_{accel}, y_{accel}, z_{accel})$ repräsentiert die Ausrichtung der Messachsen der Beschleunigungssensoren, und das System $(x_{gyro}, y_{gyro}, z_{gyro})$ das der drei Drehratensensoren. Diese einzelnen Basisvektoren, die diese beiden Systeme aufspannen, stehen nun nicht mehr notwendigerweise paarweise orthogonal aufeinander. Zusätzlich kann die Länge dieser Vektoren aufgrund unterschiedlicher Skalierungsfaktoren voneinander abweichen.

Deshalb müssen die Messwerte der Gyroskope durch die Matrix \mathbf{T}_{gyro} (3.5) und jene der Beschleunigungsmesser mittels \mathbf{T}_{acc} (3.6) transformiert werden, bevor sie im IMU-Algorithmus verarbeitet werden können. Durch diese Transformation werden die Sensoren mit ihren Messachsen in das gemeinsame IMU-Koordinatensystem überführt. Mit diesem Verfahren können sowohl Ausrichtungsfehler bei der Montage als auch Fehler behoben werden, die durch die Nichtparallelität von Sensor-Messachse und IMU-Koordinatensystem entstehen.

$$\mathbf{T}_{gyro} = \begin{bmatrix} t_{11}^g & t_{12}^g & t_{13}^g \\ t_{21}^g & t_{22}^g & t_{23}^g \\ t_{31}^g & t_{32}^g & t_{33}^g \end{bmatrix} \quad (3.5)$$

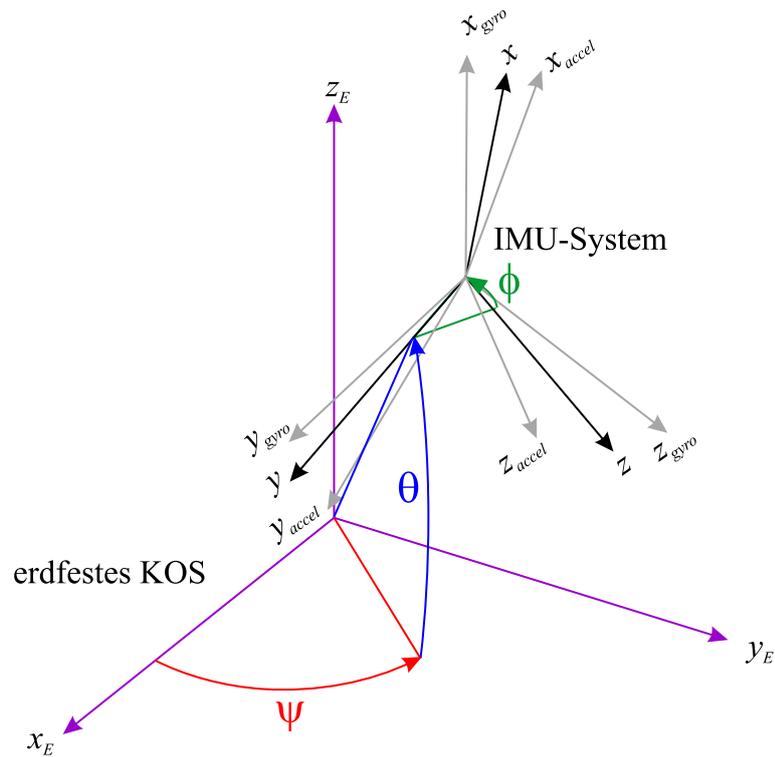


Abbildung 3.5: Erdfestes Koordinatensystem und körperfestes IMU-Koordinatensystem, sowie die Achsen der Sensoren

$$\mathbf{T}_{acc} = \begin{bmatrix} t_{11}^a & t_{12}^a & t_{13}^a \\ t_{21}^a & t_{22}^a & t_{23}^a \\ t_{31}^a & t_{32}^a & t_{33}^a \end{bmatrix} \quad (3.6)$$

Im Idealfall stellen die beiden Matrizen die Einheitsmatrix $\mathbf{I}^{3 \times 3}$ dar. In [Abschnitt 7.6](#) werden sie quantitativ bestimmt.

4 Hardware

Für diese Studienarbeit wurde ein Hardware-Prototyp aufgebaut, der zwei verschiedene Einsatzzwecke abdeckt.

Zum einen dient der Prototyp als *Interface-Board*, das lediglich die Sensoren ansprechen und deren Rohmesswerte an den Computer bzw. Datenlogger weitergeben soll. Diese Funktionsweise wird für die Kalibrierung der Sensoren (siehe [Kapitel 7](#)) und die Datenaufzeichnung der Messwerte für das Postprocessing in MATLAB (siehe [Kapitel 8](#) und [Kapitel 9](#)) verwendet.

Andererseits soll der im folgenden [Kapitel 5](#) entwickelte Algorithmus in Echtzeit auf dem Prototypen laufen, so dass die Lagewinkel nicht erst beim Postprocessing, sondern bereits zur Laufzeit verfügbar sind. Dies setzt voraus, dass der eingesetzte Mikrocontroller über ausreichende Rechenkapazitäten verfügt, um den IMU-Algorithmus in Echtzeit zu verarbeiten.

Die in beiden Varianten identischen Hardwarekomponenten und ihre Verschaltung stellt die [Abbildung 4.1](#) schematisch dar. Der Signalpfad „Ausgabe (MC)“ stellt die Verbindung zu einem externen Mikrocontroller her, auf dem der Filteralgorithmus implementiert werden könnte, falls die Rechenleistung des IMU-Mikrocontrollers nicht ausreichen sollte.

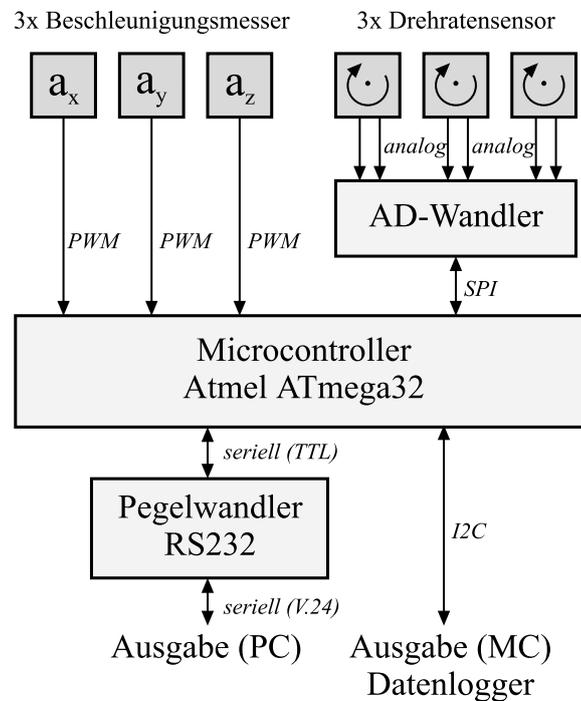


Abbildung 4.1: Hardwarekomponenten der IMU

4.1 Analog-Digital-Wandler

Während die Beschleunigungssensoren mit ihrem PWM-Ausgang direkt an den Mikrocontroller angeschlossen werden können, wird für die Gyroskope ein AD-Wandler benötigt, der die analogen Spannungen in digitale Werte umwandelt. Da der im Mikrocontroller eingebaute AD-Wandler mit seiner Auflösung von 10 bit für die Erfassung des Drehratensignals nicht ausreicht, musste ein separater, externer AD-Wandler angeschlossen werden.

An den AD-Wandler werden folgende Anforderungen gestellt:

- Die Auflösung des Wandlers sollte 16 bit betragen. Zwar würden auch 14 bit reichen, jedoch läuft die Kommunikation mit dem AD-Wandler immer Byte-orientiert ab, und im Mikrocontroller werden die Werte in Variablen des Typs 16 bit Integer gespeichert.
- Es werden mindestens 6 Eingangskanäle benötigt, die entweder per Multiplexer an das Sample-and-Hold weitergeleitet werden, oder die parallel dedizierten Sample-and-Hold Elementen zugeführt werden.
- Der Wandler soll es ermöglichen, die 6 Kanäle mit mindestens je 300 Hz einzulesen. Er muss über eine ausreichend schnelle Wandlungsrate verfügen, und die Zeit für die Kanalschaltung muss darin berücksichtigt sein.
- Das Protokoll der Datenübertragung soll vom Mikrocontroller per Hardware unterstützt werden. Außerdem muss es schnell genug sein, damit die Datenübertragungsrate keine Performance-Einbußen verursacht. Deshalb sollte der Wandler über eine SPI-Schnittstelle verfügen.
- Der Wandler sollte eine interne Referenzspannungsquelle bereitstellen. Es soll aber auch möglich sein, eine externe Spannung als Referenzspannung anzulegen.
- Damit zu den Fehlern der Gyroskope nicht noch weitere Störungen durch den AD-Wandler in das Signal gelangen, sollte der Wandler ein geringes Rauschen und eine gute Linearität aufweisen.
- Der Baustein sollte als SMD-Version verfügbar sein.

Aus der Vielzahl der am Markt angebotenen Bausteine wurde der „LTC1867A“ von **LINEAR TECHNOLOGY INC.** (a) ausgewählt, da dieser die gestellten Anforderungen optimal erfüllt.

Weitere wichtige technische Daten sind in [Tabelle 4.1](#) aufgeführt.

Der AD-Wandler wurde mit einer Schaltung aus Stützkondensatoren gemäß dem Datenblatt aufgebaut. Mittels zweier Steckbrücken (Jumper) kann zwischen der internen 4,096 V Referenzspannung und einer externen Referenzspannung gewählt werden. Um einen möglichst großen Messbereich zu erreichen, wurde die geglättete Betriebsspannung (5,0 V) an den Referenzspannungsanschluss des AD-Wandlers gelegt.

Tabelle 4.1: Technische Daten des AD-Wandlers LTC1867A, Quelle: [Datenblatt zum LTC1867A](#)

Wert	Größe	Einheit
Versorgungsspannung	5	V
Stromaufnahme	1,3	mA
AD-Kanäle	8	
Auflösung	16	bit
Wandlerrate	200	kHz
S/H-Acquisition Time	1,1	μs
Wandlungszeit	3,5	μs
SPI-Frequenz	max. 40	MHz
Linearitätsfehler	3	LSB
Power Supply Sensitivity	± 5	LSB
Signal-Rauschabstand	89	db
Gehäuse	16-pin Narrow SSOP	
Betriebstemperatur	-40 bis +85	$^{\circ}C$

4.2 Mikrocontroller

Als zentrale Steuerlogik wurde ein Mikrocontroller von **ATMEL** ausgewählt. Der *Atmel AVR ATmega32* ist ein RISC-Mikroprozessor, der über getrennten Programmspeicher (32KiB Flash) und Wertespeicher (2KiB SRAM) verfügt. Der Mikrocontroller kann mit bis zu 16 MHz getaktet werden.

Ein besonderes Merkmal dieses Mikrocontrollers sind die umfangreichen Schnittstellen und I/O-Funktionalitäten¹. Der ATmega32 stellt eine hardwaremäßige SPI-Schnittstelle und ein I²C-Interface bereit. Außerdem verfügt er über drei externe Interrupt-Eingänge, die zur Laufzeit in ihrem Verhalten verändert werden können. Es kann separat eingestellt werden, ob ein Interrupt bei einer steigenden oder fallenden Signalflanke am Eingang ausgelöst wird. Diese Funktionalität wird genutzt, um das PWM-Signal der Beschleunigungssensoren zu dekodieren.

Für zeitkritische Aufgaben und Funktionen, die zu bestimmten Zeitpunkten ausgelöst werden müssen, bietet der Mikrocontroller „Timer-Overflow-Interrupts“ an: Läuft ein eingebauter 8 bit bzw. 16 bit Timer über, so wird ein Interrupt ausgelöst und in die entsprechende ISR gesprungen. Diese Funktionalität ist nützlich, wenn der im nächsten Kapitel entwickelte Algorithmus auf den Mikrocontroller portiert werden soll und dort in einem festgelegten Zeitraster mit exklusiver Priorität ausgeführt wird.

Weiterhin bietet der Mikrocontroller eine UART-Schnittstelle zur Kommunikation mit einem PC, über die beim vorliegenden Prototypen jedoch keine Daten übertragen werden. Alle Messwerte und Signale der IMU werden über die I²C-Schnittstelle übertragen. Sollte der IMU-Algorithmus portiert werden, so könnte die IMU über diese Schnittstelle die Lagewinkel ausgeben.

¹Eingabe- und Ausgabe-Funktionen, z. B. für Funktionen, die direkt das Programm zur Laufzeit beeinflussen

Tabelle 4.2: Pin-Belegung am Mikrocontroller

Nr.	Pin	Belegung
3	INT2	externer Interrupt, PWM des z-Beschleunigungsmessers
4	PB3	Chip-Select-Leitung des AD-Wandlers
5	\overline{SS}	Ausgang muss für SPI-Master auf <i>high</i> liegen
6	MOSI	„Master out Slave in“ des SPI-Bus
7	MISO	„Master in Slave out“ des SPI-Bus
8	SCK	Taktsignal des SPI-Bus
9	\overline{Reset}	Reset des Mikrocontrollers
10	VCC	+5 V Versorgungsspannung
11	GND	Masse
12	XTAL2	externes Quarz
13	XTAL1	externes Quarz
14	RXD	Empfangskanal des UART
15	TXD	Sendekanal des UART
16	INT0	externer Interrupt, PWM des x-Beschleunigungsmessers
17	INT1	externer Interrupt, PWM des y-Beschleunigungsmessers
22	SCL	Taktsignal des I^2C -Bus
23	SDA	Datensignal des I^2C -Bus
31	GND	Masse

Für die Mikrocontroller der AVR-Reihe steht die Entwicklungsumgebung [WinAVR](#) und ein C-Compiler² zur Verfügung, die beide für diese Arbeit verwendet wurden. Die [Tabelle 4.2](#) zeigt die verwendete Pinbelegung am Mikrocontroller.

4.3 Interface-Board

Das Interface-Board vereint die beschriebenen Bauteile und die Sensoren auf einer Platine. Die [Abbildung 4.2](#) zeigt schematisch die Anordnung der Bauteile. Das Format des Prototyps wurde so gewählt, dass der Prototyp mit seiner Breite in y-Richtung von 6 cm auch in schlanken Rümpfen von unbemannten Fluggeräten montiert werden kann. Die Länge der Platine beträgt 14 cm, die Höhe ergibt sich durch die aus der Platine ragenden Bauteile zu 3 cm.

Der Prototyp wurde mit Gewindebolzen zwischen zwei 1-mm-Aluminiumplatten montiert. Diese Platten wurden derart hergestellt und montiert, dass ihre Außenkanten einen Quader mit parallelen Kanten und bi-orthogonalen Ecken bilden. Die Abmessungen des Prototyps betragen somit 60 mm × 41 mm × 150 mm. Die [Abbildung 4.3](#) zeigt den fertigen IMU-Prototyp.

4.3.1 Aufbau

Der Prototyp wurde auf einer einfachen 2,54-mm-Lochrasterplatine aufgebaut. Änderungen der Schaltung hätten so schnell vorgenommen werden können, und die Bauteile in herkömm-

²für den ATmega angepasste Version des GNU GCC Compilers

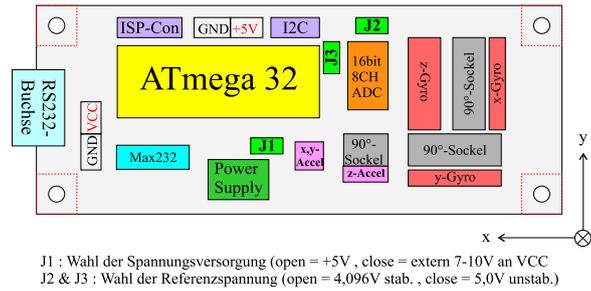


Abbildung 4.2: schematisches Board-Layout des IMU-Prototyps



Abbildung 4.3: IMU-Prototyp montiert im Rahmen aus parallelen Aluminiumplatten

licher Ausführung (kein SMD) lassen sich einfacher einlöten. Die Gyroskope wurden in spezielle 90°-IC-Sockel gesteckt (siehe [Abbildung 4.6](#)). Auch der Beschleunigungssensor, der die z-Achse erfasst, wurde mittels 90°-IC-Sockel montiert.

Die restlichen integrierten Bauteile wurden in spezielle Sockel gesteckt und nicht direkt eingelötet. Dies bietet den Vorteil, dass die Bauteile einfach ausgetauscht werden können. Der AD-Wandler, der nur in einer SMD-Version verfügbar ist, konnte über einen selbst erstellten Adapter ebenfalls in einen solchen Stecksockel installiert werden.

Die Beschleunigungssensoren verfügen über ein LLC-Gehäuse, das nicht auf einer Lochrasterplatine eingelötet werden kann. Deshalb musste hier ein Trick angewendet werden: Die Sensoren wurden „auf dem Rücken“ auf einen modifizierten IC-Sockel geklebt und dann wurden mit Bondingdrähten die Kontakte verbunden. Dabei wurde darauf geachtet, dass die Ausrichtung des Beschleunigungssensorgehäuses parallel zu den Kanten des IC-Sockels liegt, wie es die [Abbildung 4.4](#) veranschaulicht.

Die Bestückungsseite der Platine zeigt die [Abbildung 4.5](#), die Unterseite ist in [Abbildung 4.7](#) dargestellt.

Analoges Filter

Gemäß dem Nyquist-Shannon-Abtasttheorem benötigt man zwischen analoger Signalquelle und AD-Wandler ein analoges Filter (Anti-Aliasing-Filter), das die Bandbreite des analogen Signals begrenzt. Beim hier vorgestellten Prototyp wurde kein separates Filter vorgesehen,

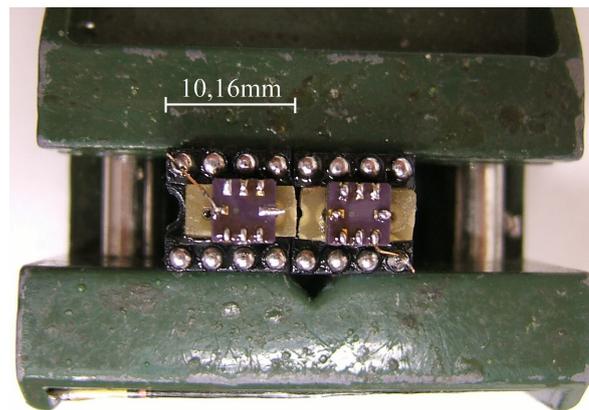


Abbildung 4.4: Montage der Beschleunigungssensoren auf den IC-Sockeln

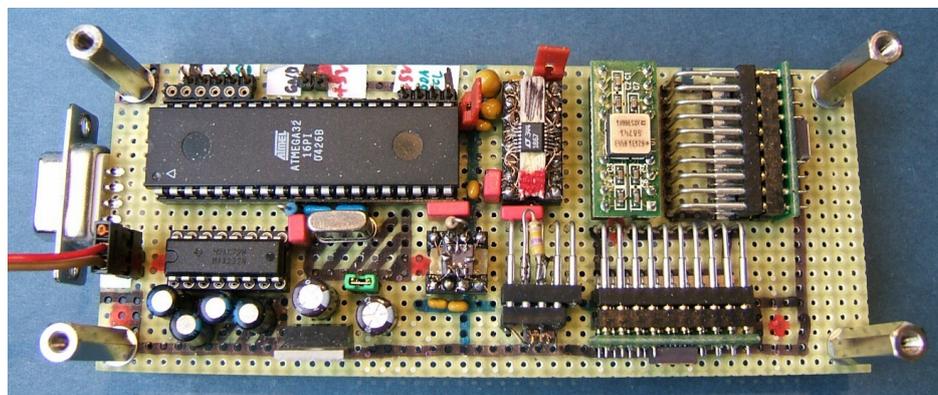


Abbildung 4.5: Bestückungsseite des IMU-Prototyps

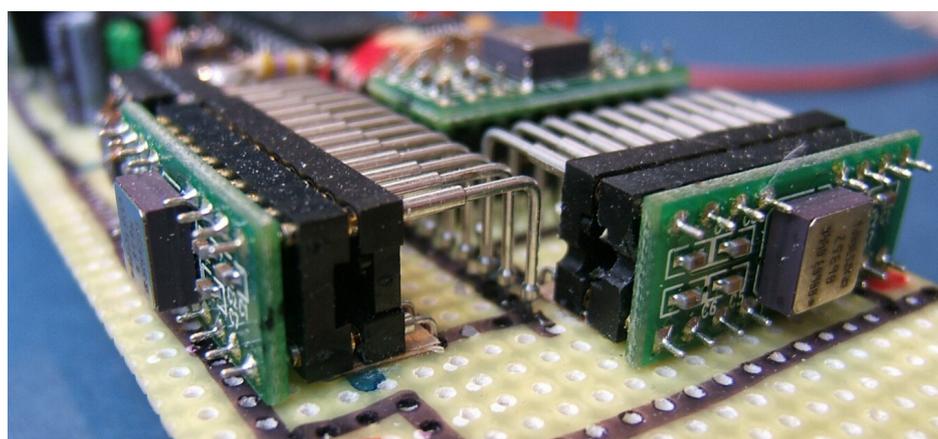


Abbildung 4.6: Detail des verwendeten 90°-IC-Sockels

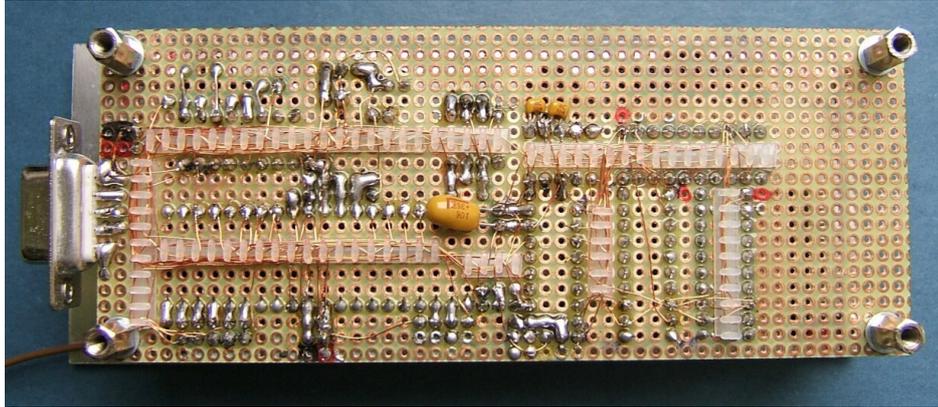


Abbildung 4.7: Unterseite des IMU-Prototyps. Die einzelnen Kontakte sind mit 0,2mm Kupferlackdraht verbunden, die in Fädelschienen geführt werden.

da die Gyroskope auf den Adapterplatinen schon ein solches implementiert haben, dessen Bandbreite bei 40 Hz liegt.

4.3.2 Spannungsversorgung

Der IMU-Prototyp kann auf zwei Arten mit der Betriebsspannung versorgt werden. Das Board kann direkt mit einer stabilisierten und gefilterten Spannung von 5,0 V betrieben werden. Alternativ kann der auf dem Board vorhandene 5-V-Festspannungsregler eingesetzt werden. Mit einem Jumper kann zwischen diesen beiden Quellen umgeschaltet werden.

Der Betrieb über den 5-V-Festspannungsregler ermöglicht eine unproblematische Spannungsversorgung aus verschiedenen Quellen, z. B. aus einem Akku mit 6 – 8 V oder einem Netzgerät.

Bei der Versorgung direkt mit 5,0 V muss darauf geachtet werden, dass diese Spannung in der vorliegenden Realisierung auch die Referenzspannung für den AD-Wandler darstellt.

4.3.3 Anschlüsse und Verbindungen

Neben der Spannungsversorgung sind auf dem Board noch weitere Anschlüsse vorgesehen.

ISP

Dieser fünfpolige Stecker dient zum Aufspielen der Firmware auf den Mikrocontroller. Die ISP-Programmierung benutzt, wie auch der AD-Wandler, die SPI-Schnittstelle mit den Leitungen MOSI, MISO und SCK. Dadurch, dass der verwendete AD-Wandler im Standby-Mode die Pins des SPI-Busses auf Tri-State („high Z“) legt, kann der Mikrocontroller ohne Änderung der Hardware neu geflasht werden.

I²C

Wie schon oben erwähnt, läuft der gesamte Datenstrom über diese Schnittstelle. Sie bildet den Kontakt zu anderen Geräten und dient auch zur Übertragung der Sensorwerte, wenn ein anderer Mikrocontroller die Lagewinkel oder Messwerte aufbereitet und über einen Filteralgorithmus berechnet. Vorteilhaft gegenüber der UART-Schnittstelle ist die höhere Datenrate von 400 kBit/s (im Vergleich zu 115,2 kBit/s bei UART). Außerdem wird die Abfrage und das Übertragen der Messwerte auf dem Mikrocontroller per Interrupt-Routine realisiert, so dass keine Eingliederung der Funktionalität in das Hauptprogramm notwendig ist.

UART

Die UART-Schnittstelle mit dem notwendigen Pegelwandler (MAX232) dient hauptsächlich zum Entwickeln und Debuggen, da sich darüber einfach Statusmeldungen an ein angeschlossenes Terminal übertragen lassen. Es wurde eine 9-polige Sub-D-Buchse zum Anschluss an einen PC direkt auf dem Prototyp vorgesehen.

4.4 IMU-Board

Aus dem Interface-Board kann durch Aufspielen einer anderen Software ohne Hardwareänderungen das IMU-Board entstehen. Den einzigen limitierenden Faktor stellt hierbei die Rechenleistung des Mikrocontrollers bzw. die Komplexität der Software dar: Nur wenn der Algorithmus so optimiert wird, dass er unter Echtzeitbedingungen und mit den begrenzten Hardware-Ressourcen läuft, kann dieser Wechsel stattfinden.

Andernfalls muss der IMU-Algorithmus auf einen separaten, leistungsstärkeren Mikrocontroller oder auf einen anderweitig vorhandenen Computer ausgelagert werden.

Die Implementierung des Filteralgorithmus auf den ATmega32-Mikrocontroller soll jedoch nicht Gegenstand dieser Arbeit sein.

5 Kalman-Filter

Für die IMU wird ein erweitertes KALMAN-Filter verwendet. Vom KALMAN-Filter gibt es verschiedene Formulierungen, hier soll die kontinuierliche Form mit diskreten Messungen implementiert werden.

Da die Rechenleistung eines Mikrocontrollers stark begrenzt ist, und der Algorithmus in Echtzeit laufen soll, muss eine möglichst optimale Systembeschreibung gefunden werden. Im Folgenden stelle ich die möglichen Systembeschreibungen vor und vergleiche ihre Vorteile und Nachteile bei der Implementierung auf dem Mikrocontroller.

5.1 Systemmodelle

Die generelle Systembeschreibung, die allen Verfahren zugrunde liegt, lautet:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (5.1)$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) \quad (5.2)$$

Damit ergibt sich dann das KALMAN-Filter nach **WAGNER (2003)** zu:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{g}(\mathbf{x}, \mathbf{w}) \quad (5.3)$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (5.4)$$

1. Schritt: „Kalman-Time-Update“

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) \quad (5.5)$$

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \quad (5.6)$$

2. Schritt: „Kalman-Measurement-Update“

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T (\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1} \quad (5.7)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K} (\mathbf{z}^* - \mathbf{h}(\hat{\mathbf{x}}_{k-1})) \quad (5.8)$$

$$\mathbf{P}_k = \mathbf{K}\mathbf{R}\mathbf{K}^T + (\mathbf{I} - \mathbf{K}\mathbf{H}) \mathbf{P}_{k-1} (\mathbf{I} - \mathbf{K}\mathbf{H})^T \quad (5.9)$$

In den oben aufgeführten Gleichungen stellt die Matrix \mathbf{F} die Jacobi-Matrix von $\mathbf{f}(\mathbf{x}, \mathbf{u})$, \mathbf{G} die Jacobi-Matrix von $\mathbf{g}(\mathbf{x}, \mathbf{w})$ und \mathbf{H} die Jacobi-Matrix von $\mathbf{h}(\mathbf{x})$ dar. Das Signal \mathbf{w} ist das weiße Systemrauschen und \mathbf{v} ist das weiße Messrauschen.

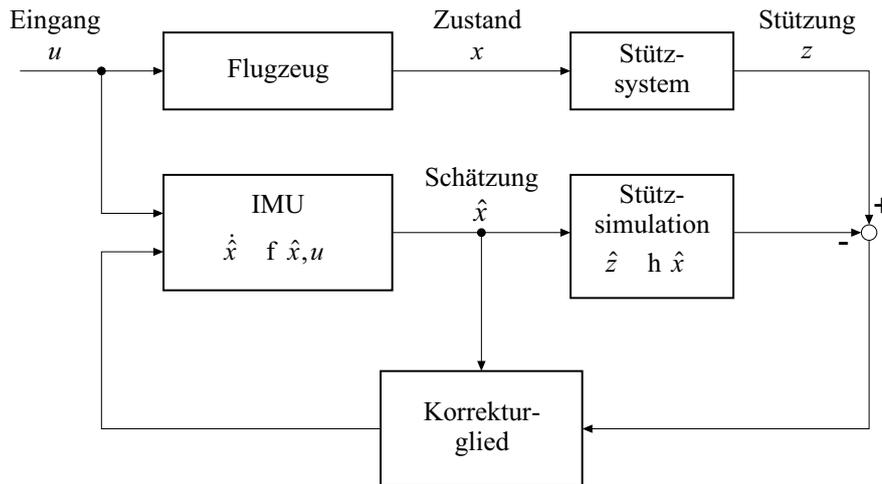


Abbildung 5.1: Blockschaltbild des Kalman-Filters

Die Gleichungen (5.5) und (5.6) werden bei der Auswertung sinnvollerweise zu einem Vektor-Matrix-System zusammengefasst und gemeinsam integriert.

Für das KALMAN-Filter soll gelten:

$$E[\mathbf{w}_t] \equiv 0 \quad (5.10)$$

$$E[\mathbf{w}_t \mathbf{w}_{t+\tau}^T] = \mathbf{Q} \cdot \delta(\tau) \quad (5.11)$$

$$E[\mathbf{w}_t \mathbf{v}_s] \equiv 0 \quad (5.12)$$

$$E[\mathbf{v}_t] \equiv 0 \quad (5.13)$$

$$E[\mathbf{v}_t \mathbf{v}_{t+\tau}^T] = \mathbf{R} \cdot \delta(\tau) \quad (5.14)$$

$$\mathbf{P} = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \quad (5.15)$$

Die [Abbildung 5.1](#) zeigt die Struktur des Filters als Blockschaltbild.

Damit das Filter stabil ist, gilt es, einige Voraussetzungen zu überprüfen: Das Matrix-Paar (\mathbf{F}, \mathbf{H}) muss entdeckbar¹ und (\mathbf{F}, \mathbf{G}) muss stabilisierbar² sein (siehe [Abschnitt 5.4](#)).

5.1.1 Systemmodell mit Euler-Winkeln

In der Flugmechanik werden die drei Lagewinkel im Raum auch Euler-Winkel genannt. Sie sind der Rollwinkel ϕ , der Pitchwinkel θ und der Gierwinkel ψ .

¹entdeckbar: Alle instabilen Moden sind beobachtbar, wobei aus der Beobachtbarkeit direkt die Entdeckbarkeit folgt.

²Ein System heißt *stabilisierbar*, wenn alle instabilen Moden steuerbar sind, bzw. wenn alle nicht steuerbaren Moden stabil sind.

Systembeschreibung

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}}_{=\dot{x}} = \underbrace{\begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix}}_A \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5.16)$$

$$\underbrace{\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}}_{=z} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_H \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (5.17)$$

Wobei p , q und r die drei Drehraten um die Hauptachsen der IMU im IMU-Koordinatensystem sind.

Der Gierwinkel ψ wird zwar nicht ausgewertet, da keine Sensoren dafür vorhanden sind (siehe [Abschnitt 3.3](#)), er wurde aber trotzdem für eine spätere Erweiterung hier aufgenommen.

Die Signale der drei Beschleunigungsmesser (a_x^* , a_y^* und a_z^*), die im KALMAN-Filter zur Stützung der Kreisel verwendet werden, werden mit folgender Beziehung in die Euler-Winkel ϕ^* , ψ und θ^* überführt:

$$\begin{bmatrix} \phi^* \\ \theta^* \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{a_y^*}{a_z^*}\right) \\ -\arctan\left(\frac{a_x^*}{\cos(\phi^*)a_z^* + \sin(\phi^*)a_y^*}\right) \\ \langle \text{nicht auswertbar} \rangle \end{bmatrix} \quad (5.18)$$

Da mit den Beschleunigungssensoren der Gierwinkel nicht ermittelt werden kann, und, wie in [Abschnitt 3.3](#) beschrieben, keine Sensoren für die Messung des Magnetfeldes eingebaut wurden, kann der Winkel ψ in [Gleichung 5.18](#) nicht ausgewertet und als Stützgröße herangezogen werden.

Vorteil

- Nur zwei (drei) Zustände. Die Matrix-Inversion ist also einfach.
- Zustand x ist direkt die Ausgangsgröße z . Damit ist H die Einheitsmatrix und braucht im Folgenden nicht mehr berücksichtigt zu werden.

Nachteil

- Mögliche Singularität in den Eulerwinkeln
- Für die Berechnung der Lagewinkel aus den Beschleunigungssensoren werden insgesamt vier Messwerte benötigt, die jeweils starkes Rauschen aufweisen. Das Stützsinal wird folglich ein extrem starkes Rauschen aufweisen.

- Bei jedem Integrationsdurchlauf und jeder Funktionsauswertung (Runge-Kutta 4. Ordnung) müssen vier mal trigonometrische Funktionen in der zeitvariablen Matrix A ausgewertet werden. Dies benötigt auf einem Mikrocontroller sehr viel Rechenleistung.

5.1.2 Systemmodell mit Quaternionen und Euler-Winkeln als Ausgang

Systembeschreibung

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \underbrace{\begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix}}_{=F(u)} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (5.19)$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \underbrace{\begin{bmatrix} \arctan\left(\frac{2(q_2q_3+q_0q_1)}{q_0^2-q_1^2-q_2^2+q_3^2}\right) \\ \arcsin(-2(q_1q_3-q_0q_2)) \\ \arctan\left(\frac{2(q_1q_2+q_0q_3)}{q_0^2+q_1^2-q_2^2-q_3^2}\right) \end{bmatrix}}_{=H(x)} \quad (5.20)$$

$$u = [p \quad q \quad r]^T \quad (5.21)$$

wobei q_0, q_1, q_2 und q_3 die Quaternionen sind, die die Fluglage beschreiben.

Vorteil

- Matrix $F(u)$ ist linear und stellt direkt die Jacobiform dar. Daher sind die Funktionsauswertungen $f(x, u) = F(u)x$ sehr einfach.

Nachteil

- Die Matrix $H(x)$, die in das KALMAN-Filter einfließt, ist nichtlinear, benötigt trigonometrische Funktionsauswertungen und enthält Brüche, die zu numerischer Instabilität führen können.
- Für die Stützwinkel aus den Beschleunigungssensoren gelten die gleichen Einschränkungen wie in [Unterabschnitt 5.1.1](#).

5.1.3 Systemmodell mit Quaternionen

Systembeschreibung

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \underbrace{\begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix}}_{=F(u)} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (5.22)$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{=H} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (5.23)$$

$$u = [p \quad q \quad r]^T \quad (5.24)$$

Vorteil

- $f(x, u)$ kann ohne explizite Linearisierung direkt als $f(x, u) = F(u)x$ geschrieben werden. Die Funktionsauswertungen sind sehr einfach.
- Zustand x ist direkt die Ausgangsgröße z . Damit ist H die Einheitsmatrix und braucht im Folgenden nicht mehr berücksichtigt zu werden.

Nachteil

- Bedingt durch die vier Zustände q_i wachsen die Matrizen in ihren Dimensionen, was deren Inversion schwieriger gestaltet.

In diesem Fall werden die Eulerwinkel nur dann berechnet, wenn die IMU eine Ausgabe ausführt. Die Winkel ϕ , θ und ψ berechnen sich dann gemäß Gleichung (5.20) aus den Quaternionen.

Die Euler-Winkel ϕ^* , θ^* und ψ , die aus den Beschleunigungsmessern nach Gleichung (5.18) gewonnen werden, transformieren sich dann in die Stützungs-Quaternionen nach WELL (2001):

$$\begin{aligned} q_0^* &= \cos\left(\frac{\phi^*}{2}\right) \cos\left(\frac{\theta^*}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi^*}{2}\right) \sin\left(\frac{\theta^*}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_1^* &= \sin\left(\frac{\phi^*}{2}\right) \cos\left(\frac{\theta^*}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi^*}{2}\right) \sin\left(\frac{\theta^*}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_2^* &= \cos\left(\frac{\phi^*}{2}\right) \sin\left(\frac{\theta^*}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi^*}{2}\right) \cos\left(\frac{\theta^*}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_3^* &= \cos\left(\frac{\phi^*}{2}\right) \cos\left(\frac{\theta^*}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi^*}{2}\right) \sin\left(\frac{\theta^*}{2}\right) \cos\left(\frac{\psi}{2}\right) \end{aligned} \quad (5.25)$$

Wie schon oben aufgeführt besteht jedoch auch in diesem Fall das Problem der verrauschten Stützwinkel aus den Beschleunigungssensoren.

5.2 Verwendetes Systemmodell

Aus den oben aufgeführten Varianten wurde das Modell mit den Euler-Winkeln als Zustand ausgewählt. Das Problem der Singularität kann für den geplanten Anwendungsfall vernachlässigt werden, da die unbemannten Fluggeräte in der Regel nur kleine Lagewinkel aufweisen und keine extremen Flugmanöver (Kunstflug, z. B. Looping) fliegen. Zudem bietet das Modell den Vorteil, nur zwei Zustandsgrößen statt vier bei Quaternionen zu haben. Dies führt zu Matrizen mit kleinerer Dimension und weniger Rechenschritten bei den Matrixoperationen.

Im Folgenden wird das vervollständigte und verfeinerte Systemmodell dargestellt:

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}}_{=\dot{\mathbf{x}}} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \end{bmatrix} \mathbf{T}_{gyro} \begin{bmatrix} p - b_p + w_p \\ q - b_q + w_q \\ r - b_r + w_r \end{bmatrix} \quad (5.26)$$

$$\underbrace{\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}}_{=\mathbf{z}} = g \underbrace{\begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \sin(\phi) \\ -\cos(\theta) \cos(\phi) \end{bmatrix}}_{=\mathbf{h}(\mathbf{x})} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (5.27)$$

mit:

$$p = p(T_x, \omega_x) \quad (5.28)$$

$$q = q(T_y, \omega_y) \quad (5.29)$$

$$r = r(T_z, \omega_z) \quad (5.30)$$

Wird mit Beschleunigungen in g statt in $\frac{m}{s^2}$ gerechnet, so lautet [Gleichung 5.27](#):

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (5.31)$$

Die Gleichungen (5.28)- (5.30) stellen die in [Abschnitt 7.4](#) aufgeführte Funktion ([Gleichung 7.5](#)) zur Aufbereitung der Gyrosignale dar.

Das Rauschen der Drehratensensoren wird durch w_i ($i = p, q, r$) abgebildet. Der Biaswert der Gyroskope wird durch b_i ($i = p, q, r$) beschrieben, und \mathbf{T}_{gyro} stellt die Transformationsmatrix dar, die in [Abschnitt 3.4](#) eingeführt wurde.

In den in [Abschnitt 5.1](#) aufgeführten Modellen werden aus den Beschleunigungswerten die Stützwinkel berechnet ([Gleichung \(5.18\)](#)) und dem KALMAN-Filter zugeführt. Die nun verwendete Beschreibung geht einen anderen Weg: Mit der Ausgangsfunktion $\mathbf{h}(\mathbf{x})$ wird aus den Lagewinkeln eine zu erwartende Beschleunigung errechnet. Unter Berücksichtigung der Transformationsmatrix der Beschleunigungssensoren folgt aus [Gleichung \(5.8\)](#) damit:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{K} (\mathbf{T}_{acc} \mathbf{z}^* - \mathbf{h}(\hat{\mathbf{x}}_{k-1})) \quad (5.32)$$

Der Vergleich der gemessenen Beschleunigungen mit den berechneten bietet den Vorteil, dass die geforderte stochastische Unabhängigkeit der Messfehler gewahrt bleibt. Bei der Berechnung der Stützwinkel aus den Beschleunigungen (Gl. (5.18)) kann dies nicht mehr angenommen werden. Außerdem sind die aus den Messungen ermittelten Beschleunigungen rauschärmer als die ihrerseits berechneten Lagewinkel zur Stützung.

Die Gleichung (5.26) kann dann umgeschrieben werden zu:

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}}_{=\dot{\mathbf{x}}} = \underbrace{\mathbf{f}(\mathbf{x})\mathbf{T}_{gyro}}_{=\mathbf{f}^*(\mathbf{x},\mathbf{u})} \begin{bmatrix} p - b_p \\ q - b_q \\ r - b_r \end{bmatrix} + \mathbf{g}(\mathbf{x})\mathbf{T}_{gyro} \begin{bmatrix} w_p \\ w_q \\ w_r \end{bmatrix} \quad (5.33)$$

Die Funktionen $\mathbf{f}(\mathbf{x})$ und $\mathbf{g}(\mathbf{x})$ lassen sich dann folgendermaßen aufstellen:

$$\mathbf{x} = \begin{bmatrix} \phi \\ \theta \end{bmatrix} \quad (5.34)$$

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \end{bmatrix} \quad (5.35)$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \end{bmatrix} \quad (5.36)$$

Die Jacobi-Matrix \mathbf{F} von $\mathbf{f}^*(\mathbf{x}, \mathbf{u})$ bestimmt sich im Fall $\mathbf{T}_{gyro} \equiv \mathbf{I}^{3 \times 3}$ zu:

$$\mathbf{F} = \frac{\partial \mathbf{f}^*(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \quad (5.37)$$

$$= \begin{bmatrix} \frac{\partial \mathbf{f}_1^*(\mathbf{x}, \mathbf{u})}{\partial \phi} & \frac{\partial \mathbf{f}_1^*(\mathbf{x}, \mathbf{u})}{\partial \theta} \\ \frac{\partial \mathbf{f}_2^*(\mathbf{x}, \mathbf{u})}{\partial \phi} & \frac{\partial \mathbf{f}_2^*(\mathbf{x}, \mathbf{u})}{\partial \theta} \end{bmatrix}$$

$$\Rightarrow \mathbf{F} = \begin{bmatrix} F_1 & F_2 \\ F_3 & 0 \end{bmatrix} \quad (5.38)$$

$$F_1 = \tan(\theta) \cos(\phi)(q - b_q) - \tan(\theta) \sin(\phi)(r - b_r)$$

$$F_2 = (1 + \tan(\theta)^2) \sin(\phi)(q - b_q) + (1 + \tan(\theta)^2) \cos(\phi)(r - b_r)$$

$$F_3 = -\sin(\phi)(q - b_q) - \cos(\phi)(r - b_r)$$

Die Jacobi-Matrix \mathbf{H} von $\mathbf{h}(\mathbf{x})$ wird analog zur Matrix \mathbf{F} gebildet:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial \theta} \\ \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial \theta} \\ \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial \theta} \end{bmatrix}$$

$$\Rightarrow \mathbf{H} = \begin{bmatrix} 0 & \cos(\theta) \\ -\cos(\theta) \cos(\phi) & \sin(\theta) \sin(\phi) \\ \cos(\theta) \sin(\phi) & \sin(\theta) \cos(\phi) \end{bmatrix} \quad (5.39)$$

Die Matrix \mathbf{G} ist die Jacobi-Matrix von $\mathbf{f}(\mathbf{x}, \mathbf{u})$, ausgewertet für die Rausch-Komponenten w_i ($i = p, q, r$). Die Gleichung 5.40 zeigt \mathbf{G} für den Fall $\mathbf{T}_{gyro} \equiv \mathbf{I}^{3 \times 3}$.

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} \frac{\partial \mathbf{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \mathbf{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \mathbf{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_r} \\ \frac{\partial \mathbf{f}_2(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \mathbf{f}_2(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \mathbf{f}_2(\mathbf{x}, \mathbf{u})}{\partial w_r} \end{bmatrix} \\ \Rightarrow \mathbf{G} &= \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \end{bmatrix} \end{aligned} \quad (5.40)$$

5.3 Erweiterung des Systems

Das bisher entwickelte Modell setzt voraus, dass der Nullpunkt der Drehratensensoren konstant ist. Die realen Sensoren weisen jedoch eine Drift auf, die mit analytischen Formeln nur schwer zu beschreiben ist. Deshalb wird das Modell nun um die drei weiteren Zustände b_p , b_q und b_r erweitert. Der *Bias* der Sensoren ist also nicht mehr konstant, sondern zeitvariabel. Der Zustandsvektor $\tilde{\mathbf{x}}$ setzt sich dann folgendermaßen zusammen:

$$\tilde{\mathbf{x}} = [\phi, \theta, b_p, b_q, b_r]^T \quad (5.41)$$

Damit kann das erweiterte Systemmodell aufgestellt werden:

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{b}_p \\ \dot{b}_q \\ \dot{b}_r \end{bmatrix}}_{=\dot{\tilde{\mathbf{x}}}} = \underbrace{\begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{=\tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u})} \mathbf{T}_{gyro} \begin{bmatrix} p - b_p + w_p \\ q - b_q + w_q \\ r - b_r + w_r \end{bmatrix} \quad (5.42)$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = g \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \sin(\phi) \\ -\cos(\theta) \cos(\phi) \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (5.43)$$

Analog zum einfachen System läßt sich auch hier die Jacobi-Matrix $\tilde{\mathbf{F}}$ (5.44), $\tilde{\mathbf{H}}$ (5.45) und $\tilde{\mathbf{G}}$ (5.46) bestimmen. Die aufgeführten Gleichungen gelten für den Fall $\mathbf{T}_{gyro} \equiv \mathbf{I}^{3 \times 3}$.

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_1 & \tilde{F}_2 & \tilde{F}_3 & \tilde{F}_4 & \tilde{F}_5 \\ \tilde{F}_6 & 0 & 0 & \tilde{F}_7 & \tilde{F}_8 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.44)$$

$$\begin{aligned} \tilde{F}_1 &= \tan(\theta) \cos(\phi)(q - b_q) - \tan(\theta) \sin(\phi)(r - b_r) \\ \tilde{F}_2 &= (1 + \tan(\theta)^2) \sin(\phi)(q - b_q) + (1 + \tan(\theta)^2) \cos(\phi)(r - b_r) \\ \tilde{F}_3 &= -1 \\ \tilde{F}_4 &= -\tan(\theta) \sin(\phi) \\ \tilde{F}_5 &= -\tan(\theta) \cos(\phi) \\ \tilde{F}_6 &= -\sin(\phi)(q - b_q) - \cos(\phi)(r - b_r) \\ \tilde{F}_7 &= -\cos(\phi) \\ \tilde{F}_8 &= \sin(\phi) \end{aligned}$$

$$\begin{aligned} \tilde{\mathbf{H}} &= \begin{bmatrix} \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial b_p} & \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial b_q} & \frac{\partial \mathbf{h}_1(\mathbf{x})}{\partial b_r} \\ \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial b_p} & \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial b_q} & \frac{\partial \mathbf{h}_2(\mathbf{x})}{\partial b_r} \\ \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial b_p} & \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial b_q} & \frac{\partial \mathbf{h}_3(\mathbf{x})}{\partial b_r} \end{bmatrix} \\ \Rightarrow \tilde{\mathbf{H}} &= \begin{bmatrix} 0 & \cos(\theta) & 0 & 0 & 0 \\ -\cos(\theta) \cos(\phi) & \sin(\theta) \sin(\phi) & 0 & 0 & 0 \\ \cos(\theta) \sin(\phi) & \sin(\theta) \cos(\phi) & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (5.45)$$

$$\begin{aligned} \tilde{\mathbf{G}} &= \begin{bmatrix} \frac{\partial \tilde{\mathbf{f}}_1(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \tilde{\mathbf{f}}_1(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \tilde{\mathbf{f}}_1(\mathbf{x}, \mathbf{u})}{\partial w_r} \\ \vdots & \vdots & \vdots \\ \frac{\partial \tilde{\mathbf{f}}_5(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \tilde{\mathbf{f}}_5(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \tilde{\mathbf{f}}_5(\mathbf{x}, \mathbf{u})}{\partial w_r} \end{bmatrix} \\ \Rightarrow \tilde{\mathbf{G}} &= \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (5.46)$$

5.4 Analyse

Für die folgenden Untersuchungen wird angenommen, dass \mathbf{T}_{gyro} und \mathbf{T}_{acc} identisch der Einheitsmatrix sind und somit vernachlässigt werden können. Sie können immer dann ganz außer Betracht bleiben, wenn sie vollen Rang besitzen, was bei den typischerweise nur kleinen Korrekturinträgen in diesen Matrizen erwartungsgemäß der Fall ist.

5.4.1 Beobachtbarkeit

Die Beobachtbarkeit eines Systems $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u}$, $\mathbf{z} = \mathbf{H}\mathbf{x}$ lässt sich mittels der Beobachtbarkeitsmatrix \mathbf{O}^T überprüfen. Das System (\mathbf{F}, \mathbf{H}) ist genau dann beobachtbar, wenn \mathbf{O}^T vollen Rang besitzt. Die Beobachtbarkeitsmatrix ist folgendermaßen definiert:

$$\mathbf{O}^T = \begin{bmatrix} \mathbf{H} \\ \mathbf{HF} \\ \mathbf{HF}^2 \\ \vdots \\ \mathbf{HF}^{n-1} \end{bmatrix} \quad (5.47)$$

Im Fall des einfachen Systems ($\mathbf{x} = [\phi, \theta]^T$) weist die Beobachtbarkeitsmatrix mit \mathbf{F} aus Gleichung (5.38) und \mathbf{H} aus Gleichung (5.39) stets vollen Rang auf: Für den möglichen Fall $\theta = \pi/2$ wird der Tangens ∞ , und die Bedingung $0 = \sin(\phi) \neq \cos(\phi) \quad \forall \phi$ garantiert vollen Rang. In allen anderen Konstellationen sichern die beiden spalten- und zeilenfremden Nulleinträge der Matrix \mathbf{O}^T die Beobachtbarkeit.

Beim erweiterten System ($\mathbf{x} = [\phi, \theta, b_p, b_q, b_r]^T$) findet man mit \mathbf{F} aus Gleichung (5.44) und \mathbf{H} aus Gleichung (5.45) die Beobachtbarkeitsmatrix.

Die für die Stabilität des Filters geforderte Entdeckbarkeit ergibt sich nach ALLGÖWER (2004) direkt aus der gezeigten Beobachtbarkeit.

Mit dem Computeralgebrasystem „Maple“ wurde die Beobachtbarkeit des erweiterten Systems analytisch untersucht. Es zeigte sich, dass die Beobachtbarkeitsmatrix einen Rangabfall aufweist: $\text{rank}(\mathbf{O}_{erw}^T) = 4$. Damit kann die Stabilität des Filters nicht mehr garantiert werden.

5.4.2 Steuerbarkeit

Eine der Voraussetzungen für das KALMAN-Filter ist die Stabilisierbarkeit von (\mathbf{F}, \mathbf{G}) . Die Stabilisierbarkeit wird mit der Steuerbarkeitsmatrix \mathbf{C} überprüft. Sie ist folgendermaßen definiert:

$$\mathbf{C} = [\mathbf{G} \quad \mathbf{FG} \quad \dots \quad \mathbf{F}^{n-1}\mathbf{G}] \quad (5.48)$$

Nach ALLGÖWER (2004) ist das System (\mathbf{F}, \mathbf{G}) dann steuerbar, wenn \mathbf{C} vollen Rang aufweist:

$$\text{rank}(\mathbf{C}) = n \quad (5.49)$$

Aus der Steuerbarkeit folgt dann die Stabilisierbarkeit.

Für das einfache System gibt „Maple“ den Rang der Steuerbarkeitsmatrix mit $\text{rank}(\mathbf{C}) = 2$ an. Damit ist das einfache System steuerbar.

Für das erweiterte System ergab die Analyse mit „Maple“ auch einen Rang der Steuerbarkeitsmatrix von $\text{rank}(\mathbf{C}_{erw}) = 2$. Damit ist das erweiterte System ($n = 5$) *nicht* mehr steuerbar,

was jedoch keinen negativen Einfluss auf die Stabilisierbarkeit haben muss. Auf eine Analyse der Eigenwerte und Eigenmoden wurde, bedingt durch die invariante Systemmatrix \mathbf{F} , verzichtet. Zur Bestimmung der Eigenwerte werden die Nullpunktwerte b_i benötigt, die jedoch für eine analytische Betrachtung nicht zur Verfügung stehen, da sie erst zur Laufzeit berechnet werden und nicht vorhergesagt werden können.

Im folgenden Abschnitt wird eine Modifikation gezeigt, die die Steuerbarkeit wieder herstellt.

5.5 Weitere Modifikation und Adaption

In einem weiteren Schritt wurde das KALMAN-Filter um Rauschkomponenten (w_1 , w_2 und w_3) der Nullpunktsfehler erweitert. Damit ergibt sich die Systembeschreibung zu:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{b}_p \\ \dot{b}_q \\ \dot{b}_r \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) & \vdots & \mathbf{0} \\ 0 & \cos(\phi) & -\sin(\phi) & \vdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \mathbf{I}^{3 \times 3} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{gyro} \begin{bmatrix} p - b_p + w_p \\ q - b_q + w_q \\ r - b_r + w_r \end{bmatrix} \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad (5.50)$$

Die Jacobi-Matrix $\tilde{\mathbf{G}}$ ergibt sich zu:

$$\begin{aligned} \tilde{\mathbf{G}} &= \begin{bmatrix} \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_r} & \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_1} & \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_2} & \frac{\partial \tilde{f}_1(\mathbf{x}, \mathbf{u})}{\partial w_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_p} & \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_q} & \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_r} & \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_1} & \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_2} & \frac{\partial \tilde{f}_5(\mathbf{x}, \mathbf{u})}{\partial w_3} \end{bmatrix} \\ \Rightarrow \tilde{\mathbf{G}} &= \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (5.51)$$

Diese Modifikation bietet den Vorteil, dass das System wieder steuerbar ist. Die Analyse mit „Maple“ ergab mit der neuen \mathbf{G} -Matrix einen Rang von $rank(\mathbf{C}_{erw}) = 5$ für die Steuerbarkeitsmatrix.

Die Matrix $\tilde{\mathbf{Q}}$, die das Rauschen des Systems beschreibt, wurde experimentell bestimmt zu:

$$\tilde{\mathbf{Q}} = \text{diag}(0, 0001; 0, 0001; 0, 0001; 10^{-9}; 10^{-9}; 10^{-9}) \quad (5.52)$$

Die Matrix \mathbf{R} bleibt unverändert.

Um einer Instabilität des Filters vorzubeugen und um die Qualität der Ergebnisse in Phasen mit hohen Beschleunigungen zu verbessern, wurde eine einfache Adaption integriert. Gemäß der Gleichung (5.43) kann die berechnete Beschleunigung maximal $\pm 1,0 g$ betragen. Die Beschleunigungssensoren liefern jedoch Signale von bis zu $\pm 2,0 g$. Deshalb wurde für die Gleichung (5.32) eine Fallunterscheidung im Algorithmus vorgesehen: Liefern alle drei Beschleunigungssensoren ein Signal vom Betrag kleiner oder gleich $1,02 g$, so wird die Gleichung (5.32) ausgewertet. Andernfalls wird sie übersprungen. Die Grenze wurde auf $1,02 g$ festgesetzt, damit ein minimales Rauschen der Beschleunigungssensoren vom Algorithmus noch toleriert wird.

5.6 Integrationsverfahren

Für alle Integrationen im Algorithmus wird ein Integrationsverfahren nach Runge-Kutta (4. Ordnung) verwendet.

Da in dem Integrationsschritt sowohl der Zustand \mathbf{x} als auch die Kovarianzmatrix \mathbf{P} integriert wird, wird ein neuer Vektor ξ eingeführt:

$$\mathbf{P} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \quad (5.53)$$

$$\Rightarrow \xi = [\phi, \theta, P_{11}, \dots, P_{1n}, \dots, P_{n1}, \dots, P_{nn}]^T \quad (5.54)$$

$$\Rightarrow \tilde{\xi} = [\phi, \theta, b_p, b_q, b_r, P_{11}, \dots, P_{1n}, \dots, P_{n1}, \dots, P_{nn}]^T \quad (5.55)$$

Die Gleichung (5.54) wird dabei für das normale Systemmodell und die Gleichung (5.55) für das um die drei Nullpunktfehler erweiterte System verwendet. Da die Matrix \mathbf{P} symmetrisch ist, würde auch die obere oder untere Dreiecksmatrix zur Integration ausreichen. Während der Entwicklung wurde der Einfachheit halber die ganze Matrix \mathbf{P} verarbeitet. Auf dem Mikrocontroller-Zielsystem würde aber aus Performancegründen nur eine Dreiecksform implementiert.

Nach **ZEITZ** ergibt sich folgendes Rechenschema für die Integration:

$$S_1 = \mathbf{f}(\xi_k, \mathbf{u}_{k-1}) \quad (5.56)$$

$$S_2 = \mathbf{f}\left(\xi_k + \frac{\Delta t}{2} S_1, \frac{1}{2}(\mathbf{u}_{k-1} + \mathbf{u}_k)\right) \quad (5.57)$$

$$S_3 = \mathbf{f}\left(\xi_k + \frac{\Delta t}{2} S_2, \frac{1}{2}(\mathbf{u}_{k-1} + \mathbf{u}_k)\right) \quad (5.58)$$

$$S_4 = \mathbf{f}(\xi_k + \Delta t S_3, \mathbf{u}_k) \quad (5.59)$$

$$\Rightarrow \xi_{k+1} = \xi_k + \frac{\Delta t}{6} (S_1 + 2S_2 + 2S_3 + S_4) \quad (5.60)$$

5.7 Phasenverschiebung und Verzögerung

Für die korrekte Funktionsweise des oben aufgeführten Algorithmus ist es essentiell, dass die Phasenverschiebungen der Sensoren minimal sind.

WEINBERG (a) untersuchte sowohl Frequenz- und Phasengang der ADXL-Drehratensensoren als auch der ADXRS-Gyroskope. Er stellte fest, dass die sensorinternen Filterschaltungen und die externe Beschaltung, die die Bandbreite festlegt, den größten Einfluss auf den Phasen- und Frequenzverlauf haben. Die ADXRS-Gyroskope verhielten sich dabei wie Filter mit zwei Polen, während sich die ADXL-Sensoren als RC-Glieder mit nur einem Pol beschreiben ließen.

Auf Verzögerungen, die durch die nachgeschaltete Messwerterfassung und Software entstehen, geht [Unterabschnitt 6.2.1](#) und [Unterabschnitt 6.2.2](#) noch genauer ein.

6 Mikrocontroller-Software

Dieses Kapitel beschreibt die Software, die auf dem Mikrocontroller läuft. Sie wurde in der Programmiersprache „C“ geschrieben und mithilfe des im **WINAVR**-Pakets enthaltenen AVR-GCC-Compilers übersetzt. Das so gewonnene Binary wurde mittels **PONYPROG** auf den Mikrocontroller übertragen.

6.1 Software-Architektur

Die Software wurde modular aufgebaut und in einzelne Funktionsbereiche unterteilt. Die [Abbildung 6.1](#) zeigt die wesentlichen Module der Software. Es gibt Funktionen, die einen Treiber für die Schnittstellen der Sensoren bereitstellen, Funktionen zur Kommunikation mit den Sensoren und Methoden zur Aufbereitung der Rohwerte. Diese Basisfunktionalität steht schließlich dem IMU-Algorithmus zur Verfügung, der mit den eingebundenen mathematischen Bibliotheken die Lagewinkel berechnet.

6.2 Software für Interface-Board

Die Software des Interface-Boards übernimmt keine Aufbereitung von Sensorrohdaten und führt auch keinen IMU-Algorithmus aus. Sie bedient sich aber der bereitgestellten Funktionen und Treiber. Sie stellt die eingelesenen Sensorsignale in aufbereiteter Form bereit, die ein Auslesen von definierten Messwertsätzen erlaubt.

Die Software für das Interface-Board wird bei folgenden Aufgaben benötigt:

- Sensoren ansprechen und versuchsweise auslesen.
- Sensoren kalibrieren.
- Temperatur-Korrekturfunktion ermitteln (siehe [Abschnitt 7.5](#) und [Abschnitt 7.4](#)).
- IMU kalibrieren (siehe Ausrichtungsfehler, [Abschnitt 7.6](#)).

6.2.1 Gyroskope einlesen

Die analogen Gyroskope werden über den 16-bit-AD-Wandler eingelesen. Die Routine liest nacheinander von jedem der drei Gyroskope sowohl den Temperaturwert als auch die Drehrate ein. In der [Abbildung 6.2](#) ist die Routine als UML-Diagramm dargestellt, im ?? wird der zugehörige C-Quellcode gezeigt.

Die Chipselect-Leitung \overline{SS} dient bei diesem AD-Wandler sowohl zum Aktivieren der SPI-Schnittstelle, als auch zum Anstoßen eines Wandlungszyklus. Somit wird im Zeitschritt k

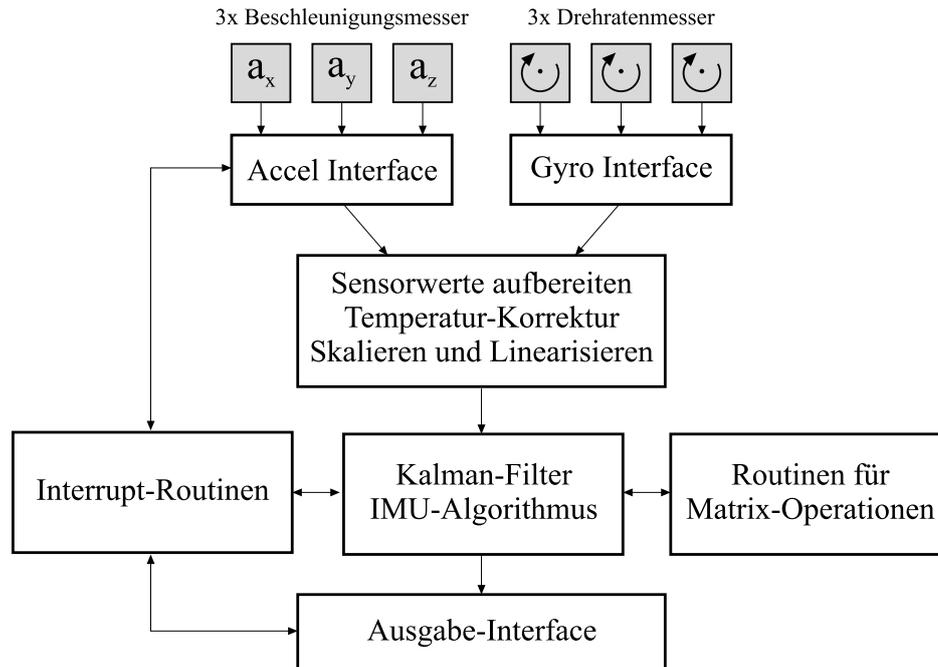


Abbildung 6.1: Module der IMU-Software

der AD-Wert aus Schritt $k - 1$ eingelesen, während gleichzeitig der AD-Kanal und Modus für die nächste Wandlung eingestellt wird. Nachdem der Wert ausgelesen wurde, wartet der Mikrocontroller $6,8 \mu s$, bis der AD-Wandler die nächste Wandlung ausgeführt hat.

Da der AD-Wandler nur über eine Sample-and-Hold-Einheit verfügt, können die drei Drehratensignale nicht gleichzeitig ausgelesen werden. Dieses verschobene und serielle Einlesen ist jedoch ursächlich für einen strukturellen Fehler und eine Verzögerung, die den Annahmen des KALMAN-Filter widerspricht. Die sequentiellen Temperaturinformationen sind, da die Änderungen aufgrund der Temperaturdrift nur klein sind, unproblematisch. Eine Modifikation der Einlese-Routine bestünde folglich darin, in jedem Schritt nur die drei Drehraten einzulesen, und die drei Temperatursignale nur in jedem n . Schritt.

Für die Auswertung im Filteralgorithmus wird angenommen, dass das Einlesen aller Sensoren zeitgleich erfolgt.

6.2.2 Beschleunigungsmesser einlesen

Das Einlesen der Beschleunigungssensoren wird über das Dekodieren des PWM-Signals in Interrupt-Routinen realisiert. Für jede Sensorachse steht eine eigene ISR zur Verfügung. Das PWM-Signal mit den auslösenden Interrupts und den Sprüngen in die ISR ist in [Abbildung 6.3](#) grafisch veranschaulicht. Die Interrupt-Routinen reagieren jeweils auf eine fallende bzw. steigende Flanke an den ihnen zugewiesenen Interrupt-Eingängen (INTx) des Mikrocontrollers.

Alle drei Routinen greifen zur Auswertung auf einen im Mikrocontroller frei laufenden 16 bit-Timer zurück. Dieser Timer zählt bis zu seinem Überlauf synchron zum Mikrocontroller mit gleicher Taktfrequenz hoch. Beim Überlauf fängt der Timer automatisch wieder bei Null an. Der Wert dieses Timers stellt somit eine sehr genaue Möglichkeit der Zeitmessung zur

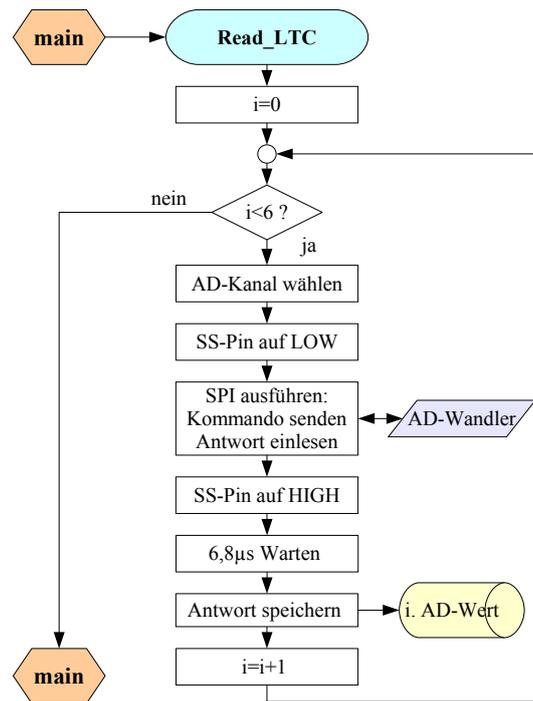


Abbildung 6.2: UML-Flussdiagramm der Routine, die den AD-Wandler ausliest und die Kommandos sendet

Verfügung. Der kleinste auflösbare Zeitschritt beträgt mit diesem Timer bei der gewählten Taktfrequenz des Mikrocontrollers $67,82 \text{ ns}$, und nach $4,444 \text{ ms}$ läuft der Timer jeweils über. Diese Zeit liegt über der maximalen Pulswiederhollänge Tb_{max} des PWM-Signals der Beschleunigungsmesser, das gemäß der Gleichung 6.1 mittels des Widerstandes $R_{set} = 470 \text{ k}\Omega$ auf $3,76 \text{ ns}$ eingestellt wurde. Dadurch ist sichergestellt, dass die Routine, die in Abbildung 6.4 als UML-Flussdiagramm dargestellt ist, das PWM-Signal sicher erkennen kann. Der C-Quellcode zur Routine ist im ?? aufgeführt.

$$Tb = \frac{R_{set}}{125M\Omega} \quad (6.1)$$

Die Beschleunigungssensoren weisen durch die Ausgabe des Pulsweitsignals eine systembedingte Verzögerung auf: Im Sensor muss aus der aktuell herrschenden Beschleunigung im ersten Schritt das PWM-Signal generiert werden. Damit der Mikrocontroller jedoch die Beschleunigung berechnen kann, muss er die Zeit Tb abwarten, bis der nächste Puls anliegt. Zuzüglich zu der Zeit, die der Mikrocontroller für die internen Rechenschritte benötigt, ergibt sich somit, dass die Phasenverschiebung mindestens Tb , unabhängig von der anliegenden Beschleunigung, beträgt. Ein Ausweg wäre die Berechnung der Beschleunigung aus Ta im Zeitschritt k und Tb aus dem Zeitschritt $k - 1$ unter der Annahme, dass $\Delta Tb \ll \Delta Ta$ gilt.

6.2.3 Interrupt-Routinen

Neben den drei oben aufgeführten Interrupt-Routinen zum Einlesen der PWM-Signale gibt es in der Software des Interface-Boards noch eine Routine, die die I^2C -Funktionalität bereit-

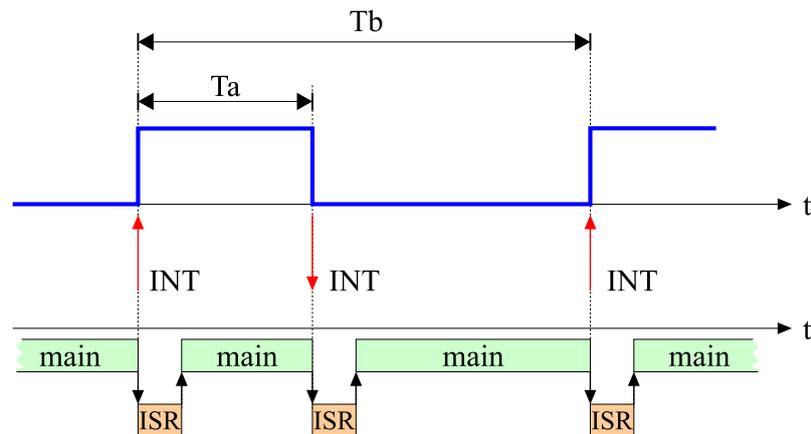


Abbildung 6.3: Schematische Darstellung der Einlese-Routine eines Beschleunigungssensors. Aus dem Hauptprogramm *main* springt das Programm bei einem Interrupt in die Interrupt-Routine *ISR*.

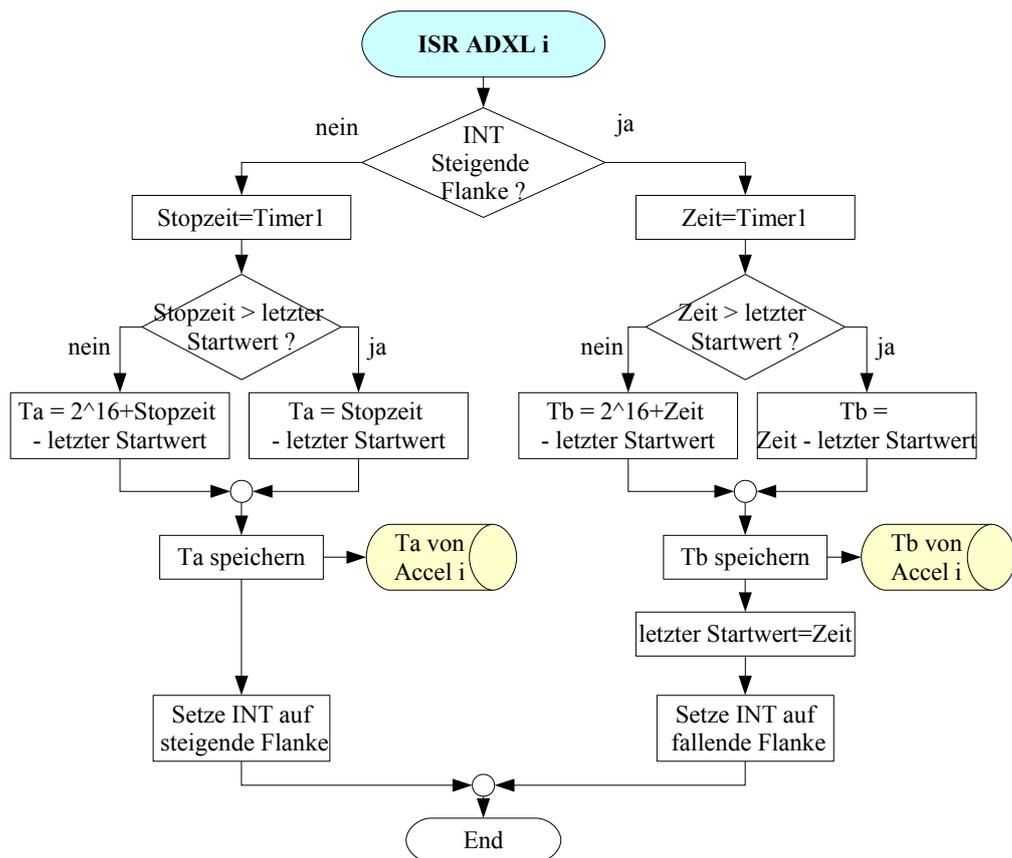


Abbildung 6.4: UML-Flussdiagramm der Interruptroutine, die einen Beschleunigungssensor einliest ($i = x, y, z$)

stellt.

Die IMU legt die in einzelne Bytes aufgeteilten Rohwerte in I^2C -Register bereit, die vom Datenlogger respektive Bordcomputer ausgelesen und wieder zusammengesetzt werden. Das Interface-Board dient in diesem Fall als „ I^2C -Slave“.

6.2.4 Hauptprogramm

Das Hauptprogramm initialisiert beim Start die Hardware des Mikrocontrollers, lädt die Treiberfunktionen und bereitet die Speicherplätze für Variablen und Werte vor.

Es fügt die einzelnen Module zusammen und führt die Einleseroutinen in einer Endlosschleife aus. Weitere Funktionalität besitzt es nicht.

7 Kalibrierung

7.1 Rauschen des AD-Wandlers

Tests mit dem AD-Wandler haben ein Rauschen von ca. 0,4 % FS ergeben. Es wurde eine geladene Zelle eines Nickel-Cadmium-Akkus an einen Eingang des AD-Wandlers angeschlossen und nur dieser eine Eingang aufgezeichnet. Dabei wurde die Spannung des Akkus als ideal geglättet und rauschfrei angesehen. Dieses Rauschen kann diverse Ursachen haben: Das Gehäuse der IMU ist nicht abgeschlossen und abgeschirmt. Elektromagnetische Störstrahlung kann das analoge Signal stören. Auch die Störstrahlung des Prototypen selbst, bedingt durch die verwendete Fädertechnik, kann Einflüsse haben. Außerdem wurden keine getrennten Spannungsversorgungen für den analogen (Sensoren und AD-Wandler) und den digitalen Teil (Microcontroller, Pegelwandler, Ladungspumpe) der Schaltung vorgesehen. Auch die Massepotentiale wurden nicht in GND_{analog} und $GND_{digital}$ aufgeteilt. Da der Messbereich mit der eingebauten 4,096 V Referenzspannungsquelle nicht ausreichend groß war, wurde die stabilisierte und geglättete Betriebsspannung als Referenzspannung verwendet. Hierdurch entstehen weitere Fehler, die jedoch nur schwer zu quantifizieren sind.

Eine Reduktion des Rauschens ließe sich wahrscheinlich durch eine separate 5-V-Präzisionsspannungsquelle als Referenzspannung erreichen.

7.2 Kalibrierung der Beschleunigungsmesser

Die drei Beschleunigungssensoren wurden anhand des Erdschwerefeldes und mittels einer schiefen Ebene vermessen. Zuerst wurde die IMU nacheinander auf die sechs äußeren Kanten gestellt, so dass für alle drei Sensoren jeweils die Signale bei $-1g$, $0g$ und $+1g$ verfügbar waren. Aus diesen Messergebnissen wurden der Offset und der Skalierungsfaktor ermittelt, die dann auf der schiefen Ebene validiert wurden. Die [Abbildung 7.1](#) zeigt den Versuchsaufbau mit dem als schiefe Ebene dienenden Vierkant-Stahlprofil, dessen Anstellwinkel mittels Spindeltrieb eingestellt werden kann. Der eingestellte Winkel wurde mit dem elektronischen Neigungsmesser „SOLA TRONIC“ abgelesen, dessen Anzeige eine Genauigkeit von $0,1^\circ$ besitzt. Die Versuche fanden bei einer Raumtemperatur von $20^\circ C$ statt.

Der IMU-Prototyp wurde in vier verschiedenen Positionen auf der schiefen Ebene montiert, um sowohl positive als auch negative Roll- und Nickwinkel erfassen zu können. Jeder Versuch wurde von der Basis-Null-Lage (BNL, $\phi = 0^\circ$, $\theta = 0^\circ$) aus gestartet. Bis 10° wurde in Schritten von 1° der Winkel erhöht, um eine feine Auflösung des im Flug relevanten Bereichs zu erhalten. Ab 10° wurde in Schritten von jeweils 10° der Winkel bis zur senkrechten Lage bei 90° erhöht. Für die x- und y-Achse wurden somit je eine Messkurve ermittelt, für die z-Achse zwei Kurven. Aus den Ergebnissen wurden die Offsetwerte und die Skalierungsfaktoren bestimmt. Dabei war eine gute Qualität bei kleinen Winkeln um die BNL wichtiger als bei

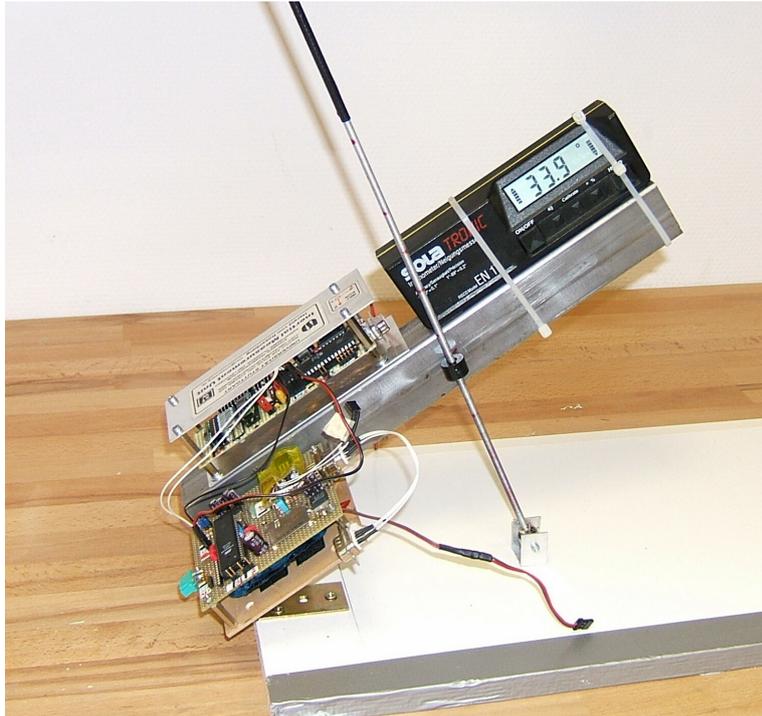


Abbildung 7.1: Kalibrierung der Beschleunigungssensoren auf der schiefen Ebene.

Tabelle 7.1: Kalibrierwerte der Beschleunigungssensoren

Sensor	Offset	Skalierungsfaktor	Bias
x	0,5164	-0,12860	0
y	0,5343	-0,12542	0
z	0,5600	0,12450	0,1

großen, da ein Flugzeug meistens nur mit kleinem Roll- und Nickwinkel fliegt. Die Flugphasen, in denen große Lagewinkel auftreten, sind generell durch starke dynamische Beschleunigungen gekennzeichnet, die nicht zur Stützung herangezogen werden können.

Die Abbildungen 7.2 und 7.3 zeigen die kalibrierten Messwerte bei der Drehung um die y-Achse (Nickwinkel) während 7.4 und 7.5 die kalibrierten Ergebnisse des Roll-Tests darstellen.

In der Tabelle 7.1 sind die gewonnenen Offset-Werte und Skalierungsfaktoren zusammengestellt, mit deren Hilfe aus Gleichung 3.2 die Beschleunigung berechnet werden kann.

Damit ergibt sich folgende Formel, die die Werte der Beschleunigungssensoren in g umrechnet:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \frac{(Ta_x/Tb_x) - 0,51640}{-0,12860} \\ \frac{(Ta_y/Tb_y) - 0,53430}{-0,12542} \\ \frac{(Ta_z/Tb_z) - 0,56000}{0,12450} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0,1 \end{bmatrix} \quad (7.1)$$

Eine einfach durchzuführende Kalibrierungsmethode stellt WEINBERG (c) vor. Außerdem

7.2 Kalibrierung der Beschleunigungsmesser

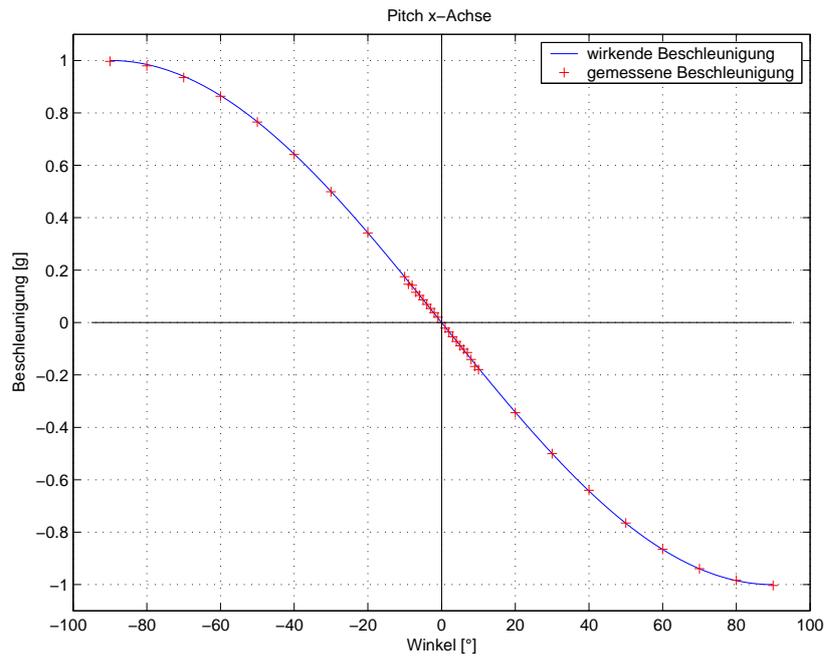


Abbildung 7.2: Kalibrierung Beschleunigungssensor x-Achse über Nickwinkel

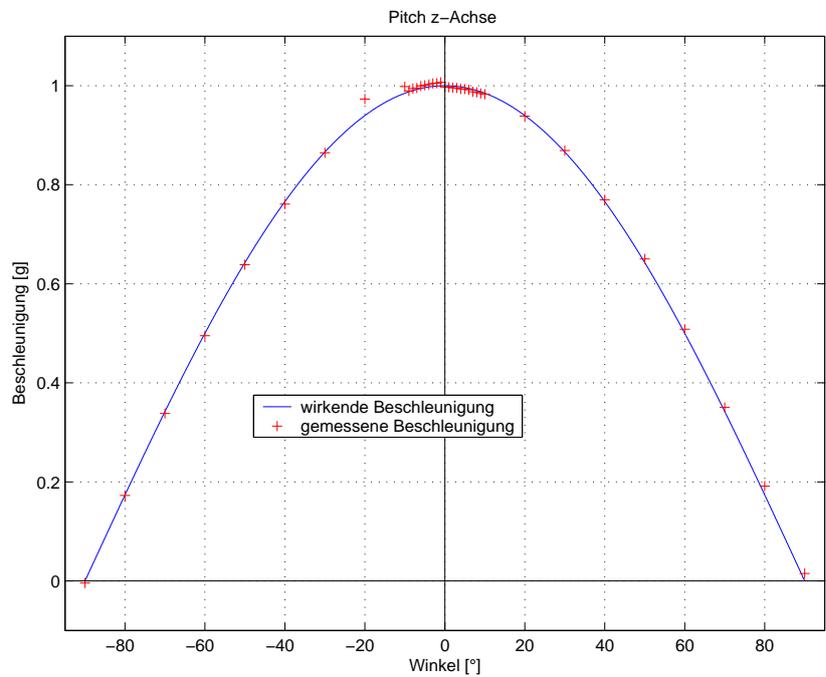


Abbildung 7.3: Kalibrierung Beschleunigungssensor z-Achse über Nickwinkel

7 Kalibrierung

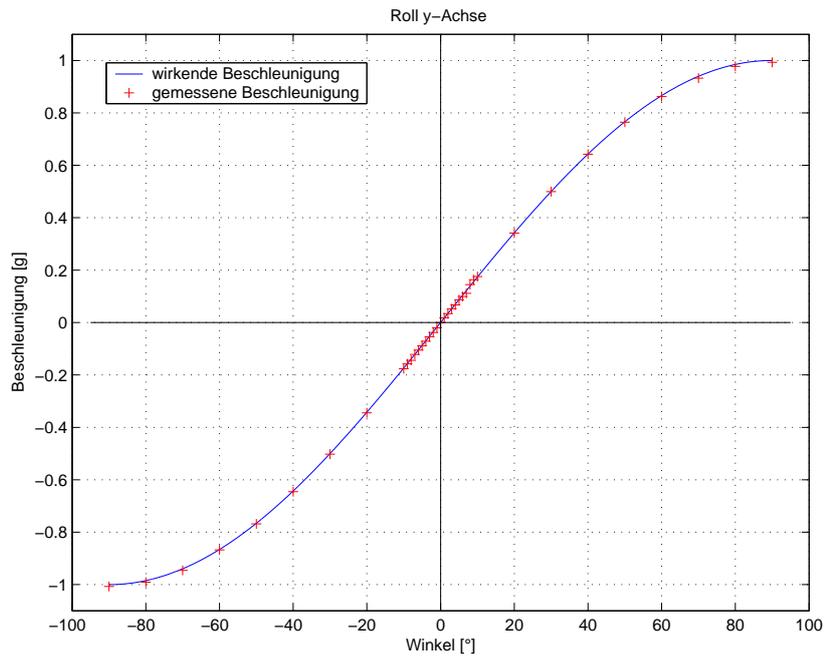


Abbildung 7.4: Kalibrierung Beschleunigungssensor y-Achse über Rollwinkel

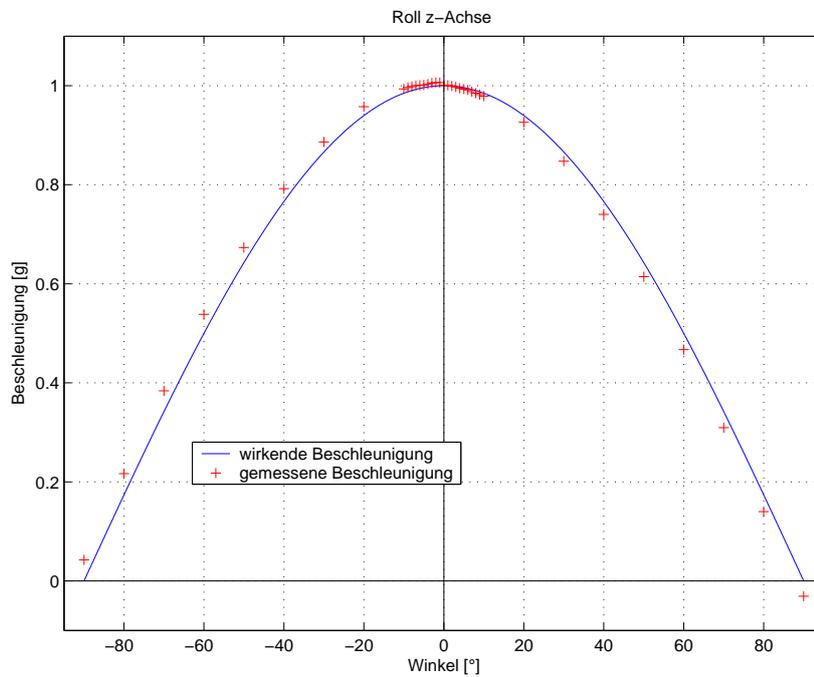


Abbildung 7.5: Kalibrierung Beschleunigungssensor z-Achse über Rollwinkel



Abbildung 7.6: Versuchsaufbau mit Drehtisch am ISD. Im Vordergrund ist der Drehtisch, rechts auf dem Tisch das Steuergerät zum Drehtisch, links daneben die Computer für die Auswertung und den Datenlogger zu sehen.

gibt er eine Methode an, wie man Änderungen von T_b aufgrund von Drift oder Jitter bei der Auswertung berücksichtigen kann.

7.3 Kalibrierung der Gyroskope

Die Gyroskope wurden in einem Vorversuch auf dem Drehtisch des ISD vermessen. Die [Abbildung 7.6](#) zeigt den Versuchsaufbau im Keller des ISD, in [Abbildung 7.7](#) ist der IMU-Prototyp auf dem Drehtisch montiert.

Der Prototyp wurde in allen drei Achsen vermessen. Dieser Versuch diente nicht nur der Bestimmung der Kalibrierungsfunktion, sondern auch dazu, die Ausrichtungsfehler zu bestimmen, deren Kompensation im [Unterabschnitt 7.6.1](#) angestrebt wird.

Pro Sensorachse wurden zwei Messreihen gefahren. Dabei wurde die Drehrate des Drehtisches von der Ruhelage jeweils in Schritten von $10^\circ/\text{s}$ bis zur Sensor-Nenn-drehrate von $300^\circ/\text{s}$ gesteigert. Jede Drehrate wurde für ca. 6 Sekunden gehalten, um eine stationäre Drehrate und ein stationäres Signal zu erhalten. Nachdem die $300^\circ/\text{s}$ anlagen, wurde der Drehtisch wieder gestoppt, um einen direkten Signalsprung von der Nenn-drehrate zur Ruhelage zu erhalten. Um den maximalen Messbereich und die Signalqualität außerhalb der Spezifikation zu überprüfen, wurde die Drehrate dann wiederum in $10^\circ/\text{s}$ -Schritten bis $600^\circ/\text{s}$ gesteigert.

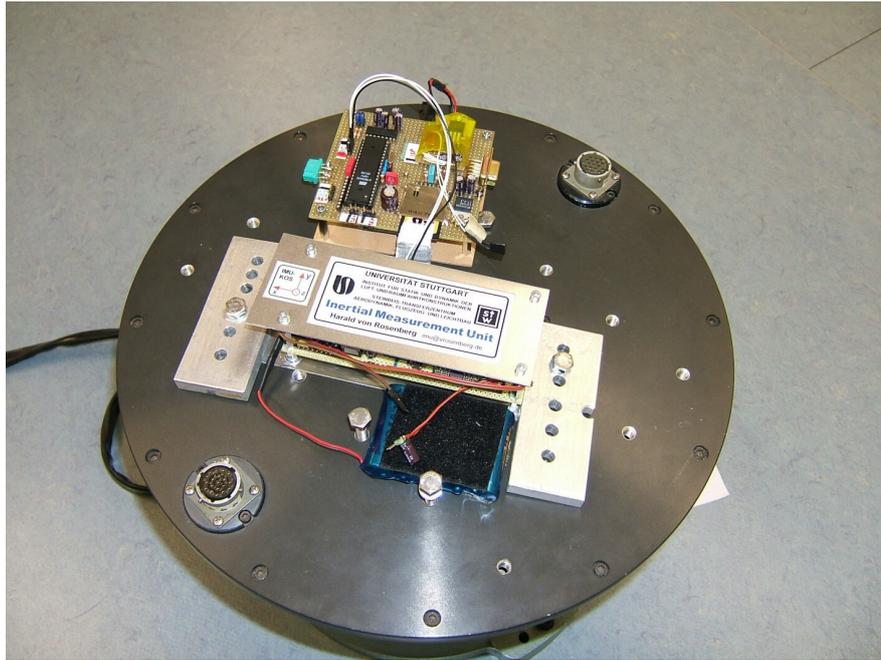


Abbildung 7.7: IMU-Prototyp montiert auf dem Drehtisch des ISD. Hinter der IMU ist der Datenlogger montiert, der die IMU-Signale aufzeichnet.

Dieser Versuch wurde sowohl für die positive Drehrichtung (im Uhrzeigersinn für die nach unten zeigende z-Achse) als auch für die negative durchgeführt.

Während der Versuche herrschte eine annähernd konstante Temperatur von ca. 18°C im Labor. Die maximal messbare Drehrate konnte mit $\pm 480^{\circ}/\text{s}$ für alle drei Achsen ermittelt werden.

Die Abbildungen 7.8 bis 7.10 zeigen die ermittelten Messpunkte und die gewonnenen Regressionsgeraden.

Damit ergibt sich die vorläufige Kalibrierungsfunktion (7.3) für die Drehratensensoren, die aus den Werten des AD-Wandlers die korrespondierende Drehrate bestimmt:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -66,885 \Omega_x \\ -65,371 \Omega_y \\ 67,437 \Omega_z \end{bmatrix} + \begin{bmatrix} 32844 \\ 32220 \\ 32498 \end{bmatrix} \quad (7.2)$$

$$\Rightarrow \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \frac{\pi}{180} \begin{bmatrix} (\omega_x - 32844) / (-66,885) \\ (\omega_y - 32220) / (-65,371) \\ (\omega_z - 32498) / 67,437 \end{bmatrix} \quad (7.3)$$

7.4 Temperatur-Korrektur der Gyroskope

Um eine Temperaturabhängigkeit der Drehraten zu prüfen, wurde der IMU-Prototyp bei Firma **AUTOFLUG STEUERUNGS- UND SENSORTECHNIK GMBH** auf einem Drehtisch „PARS3-MR“ von **RMS – REGELUNGS- UND MESSTECHNIK** mit Temperaturkammer (Fabrikat HERA-

7.4 Temperatur-Korrektur der Gyroskope

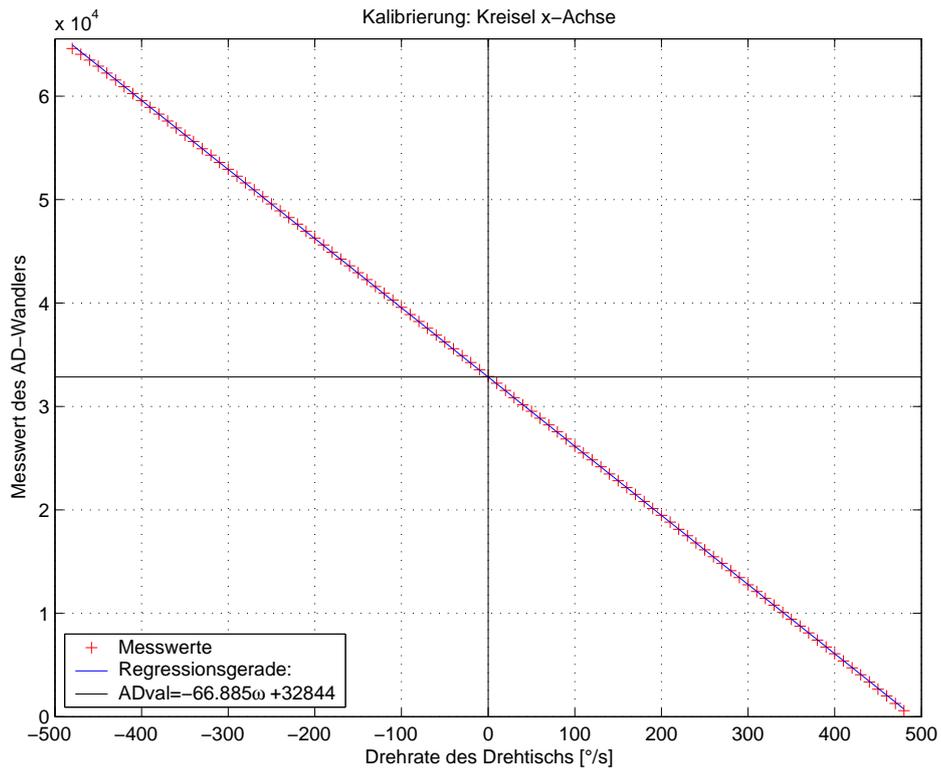


Abbildung 7.8: Kalibrierungsgerade des x-Gyros bei 18°C

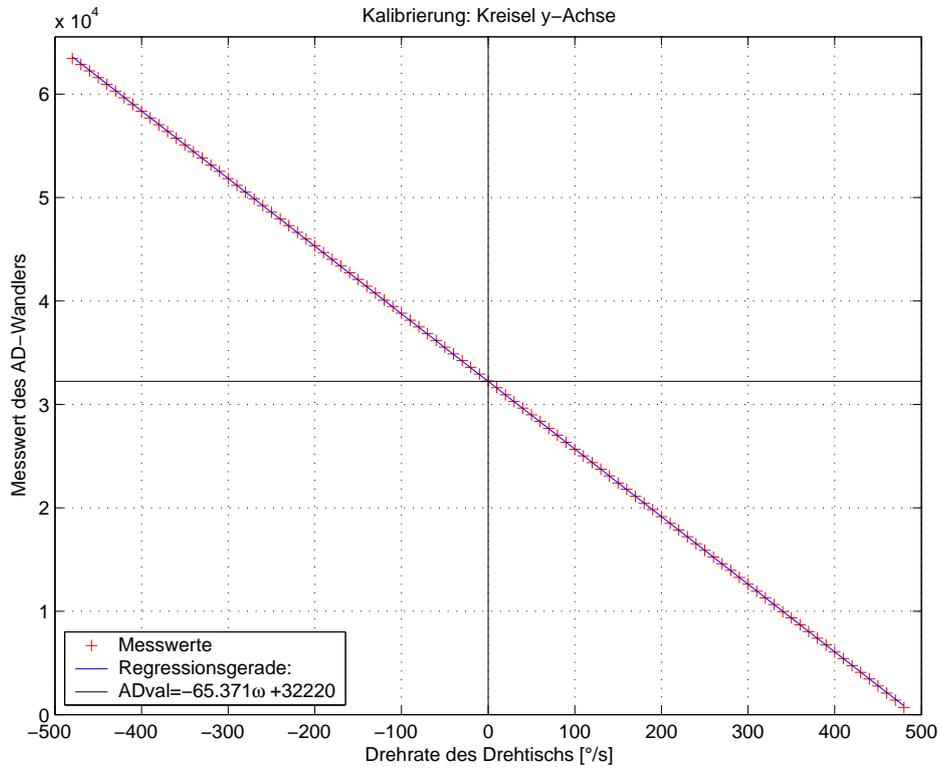


Abbildung 7.9: Kalibrierungsgerade des y-Gyros bei 18°C

7 Kalibrierung

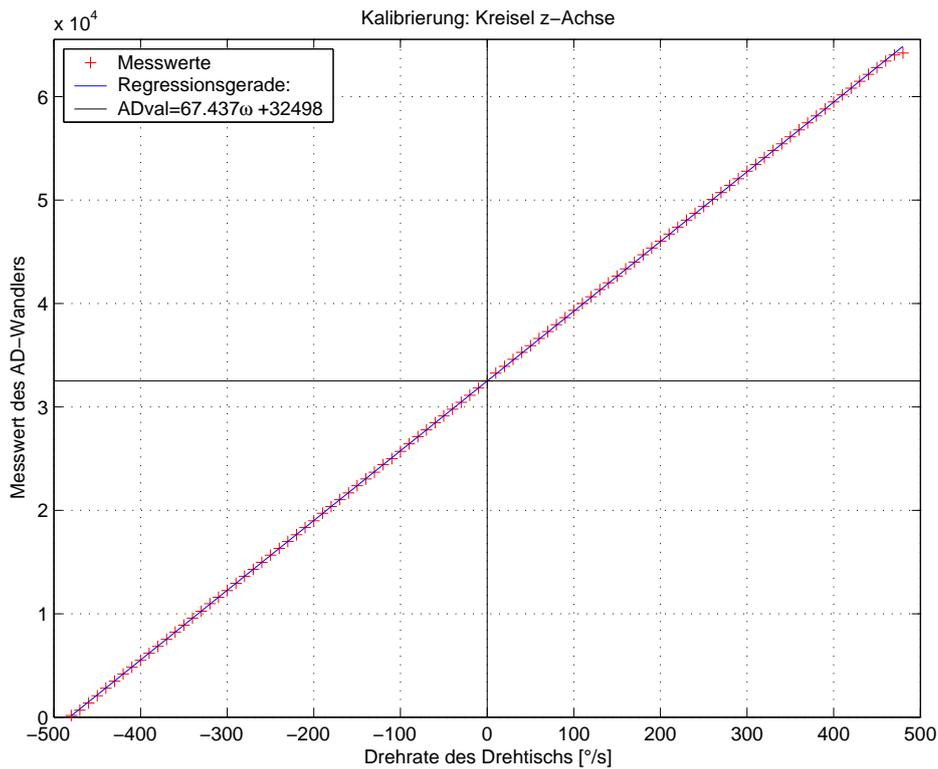


Abbildung 7.10: Kalibrierungsgerade des z-Gyros bei 18 $^{\circ}C$

EUS VÖTSCH) vermessen. Die [Abbildung 7.11](#) zeigt den Versuchsaufbau. Die Drehebene des Drehtischs befindet sich innerhalb der Temperaturkammer.

Für den Drehtisch wurde ein Programm erstellt, das bei verschiedene Temperaturen gefahren wurde. Die Sensoren wurden bei $-20,0^{\circ}C$, $0,0^{\circ}C$, $+20,0^{\circ}C$, $+40,0^{\circ}C$ und $+60,0^{\circ}C$ vermessen. Dabei bestand jede Messung aus folgendem Drehratenprofil:

1. Aufheizen bzw. Kühlen der Kammer und Stabilisierung der Temperatur
2. Von der Ruhelage aus in Schritten von je $10^{\circ}/s$ bis $+300^{\circ}/s$ beschleunigen, dabei jede Drehrate für 8 s halten
3. Mit Drehraten von $+310^{\circ}/s$ bis $+510^{\circ}/s$ in $10^{\circ}/s$ -Schritten fortfahren
4. Nachdem $+510^{\circ}/s$ anliegen, Drehtisch stoppen
5. Drehtisch wieder auf $+510^{\circ}/s$ beschleunigen, und dann in Schritten von $-10^{\circ}/s$ bis zur Ruhelage herunterfahren
6. Danach die Schritte 2 - 5 für die entgegengesetzte Drehrichtung bis $-300^{\circ}/s$ und $-510^{\circ}/s$ ausführen

Die x-Achse der IMU wurde bei allen oben genannten Temperaturen gemessen, die z-Achse wurde bei $+60,0^{\circ}C$ und $-20,0^{\circ}C$ gemessen, um eine Validierung der gewonnenen Temperaturkorrekturfunktion zu erhalten.

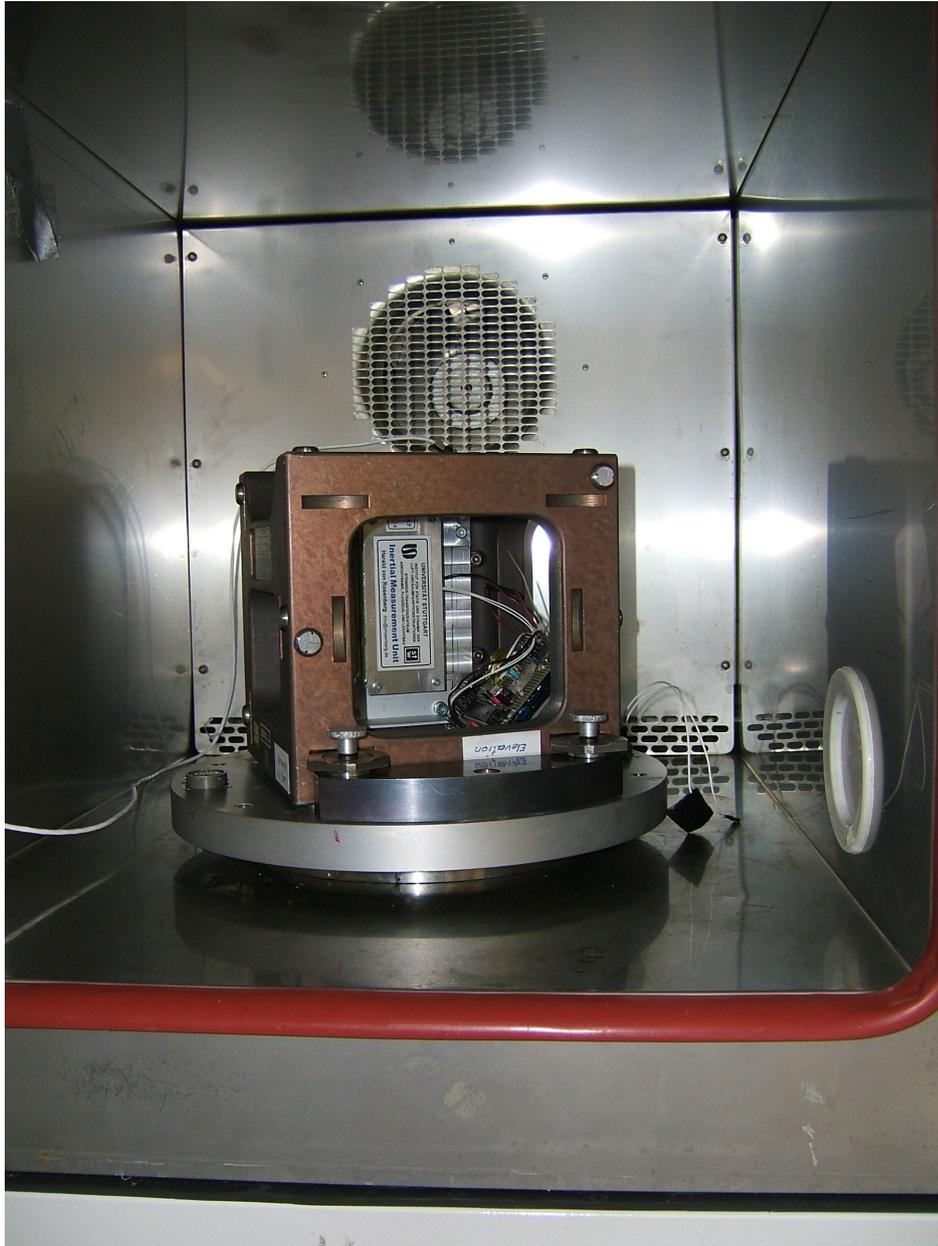


Foto mit freundlicher Genehmigung der Firma AUTOFLUG

Abbildung 7.11: Drehtisch der Firma **AUTOFLUG STEUERUNGS- UND SENSORTECHNIK GMBH**. Der IMU-Prototyp wurde in einem Präzisionswürfel befestigt, der seinerseits auf den Drehtisch aufgespannt wurde. Da die Messwerte nicht per I²C-Bus nach außen geführt werden konnten, wurde der Datenlogger ebenfalls an dem Würfel in der Temperaturkammer befestigt. Die Spannungsversorgung wurde von außen über die Schleifringe des Drehtischs der IMU und dem Datenlogger zugeführt.

7 Kalibrierung

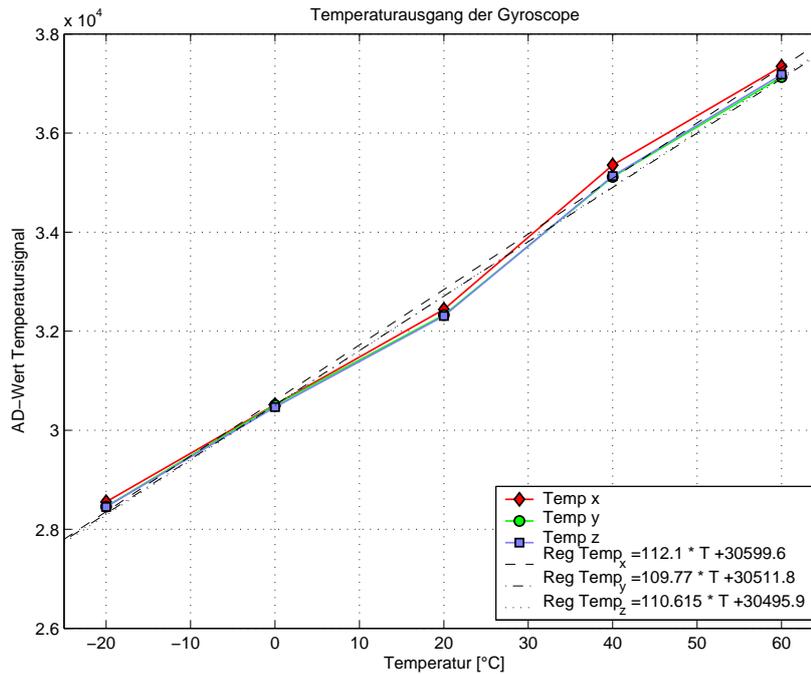


Abbildung 7.12: Temperatursignale der Beschleunigungssensoren und ermittelte Regressionsgeraden

Die Auswertung der Versuche, in denen die Drehraten nach dem Erhöhen bis zur Maximaldrehrate wieder bis zum Stillstand reduziert wurde, zeigte, dass die Sensoren keine Hysterese aufweisen. Alle Messwerte während des Dekrementierens stimmten mit denen der Inkrementierungsphase überein.

Das analoge Temperatursignal der drei Gyroscopes ist in [Abbildung 7.12](#) dargestellt. Mit den gewonnenen Regressionsgeraden erhält man die Umrechnungsformel (7.4). Eine lineare Regression wird für das Temperatursignal als ausreichend angesehen.

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} \frac{ADtemp_x - 30600}{112,1} \\ \frac{ADtemp_y - 30512}{109,8} \\ \frac{ADtemp_z - 30496}{110,6} \end{bmatrix} \quad (7.4)$$

Im Folgenden wird die Temperaturabhängigkeit des Drehratensignals untersucht. Die [Abbildung 7.13](#) zeigt die Messwerte des Gyroskops der x-Achse bei den verschiedenen Temperaturen. Es muss an dieser Stelle nochmals hervorgehoben werden, dass mit dem eingesetzten Messaufbau *nicht* die Fehler der Gyroscopes alleine, sondern die Fehler des *Gesamtsystems aus AD-Wandler und Gyroskop* gemessen wurden.

Dieser Sachverhalt mag auch das Verhalten aus Gyroskop und AD-Wandler erklären, dass die Kennlinie bei $+40^\circ\text{C}$ deutlich von jenen bei $+20^\circ\text{C}$ und $+60^\circ\text{C}$ abweicht. Die Reihenfolge der Kennlinien bei $-420^\circ/\text{s}$ lautet: $+20^\circ\text{C}$, $+60^\circ\text{C}$, 0°C , $+40^\circ\text{C}$ und -20°C . Die Offsetwerte in der Ruhelage bei $0^\circ/\text{s}$ erreichen mit (steigendem Wert) die Kennlinien: $+20^\circ\text{C}$, 0°C , $+60^\circ\text{C}$, $+40^\circ\text{C}$ und schließlich -20°C . Am oberen Drehratenende stimmt die Reihenfolge der

7.4 Temperatur-Korrektur der Gyroskope

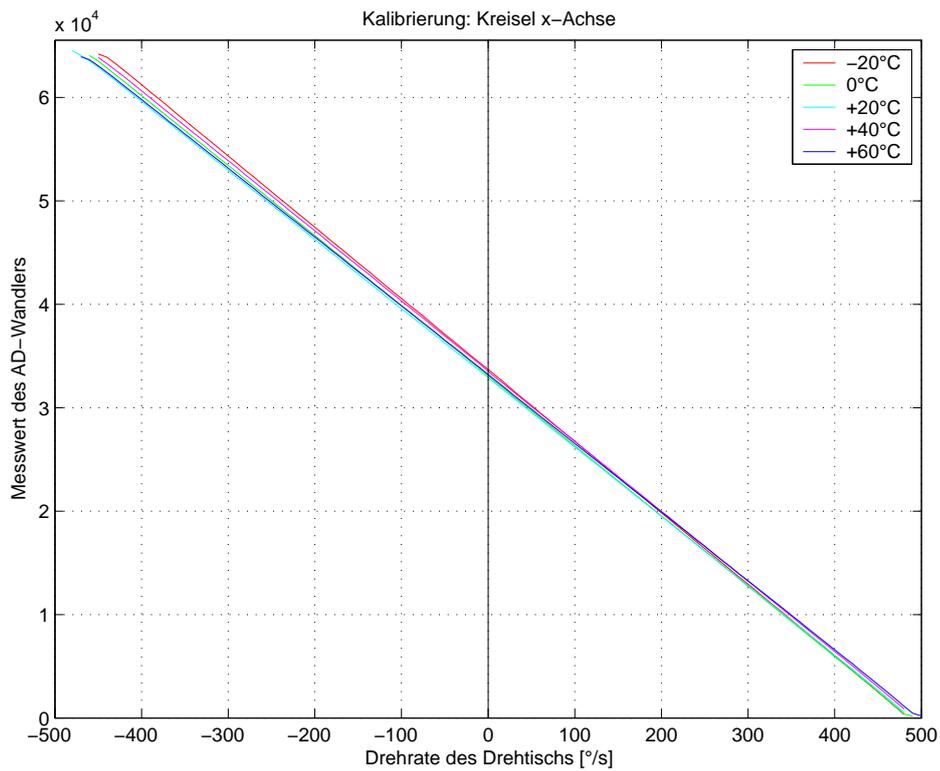


Abbildung 7.13: Messwerte des x-Achsen-Gyroskops bei verschiedenen Temperaturen

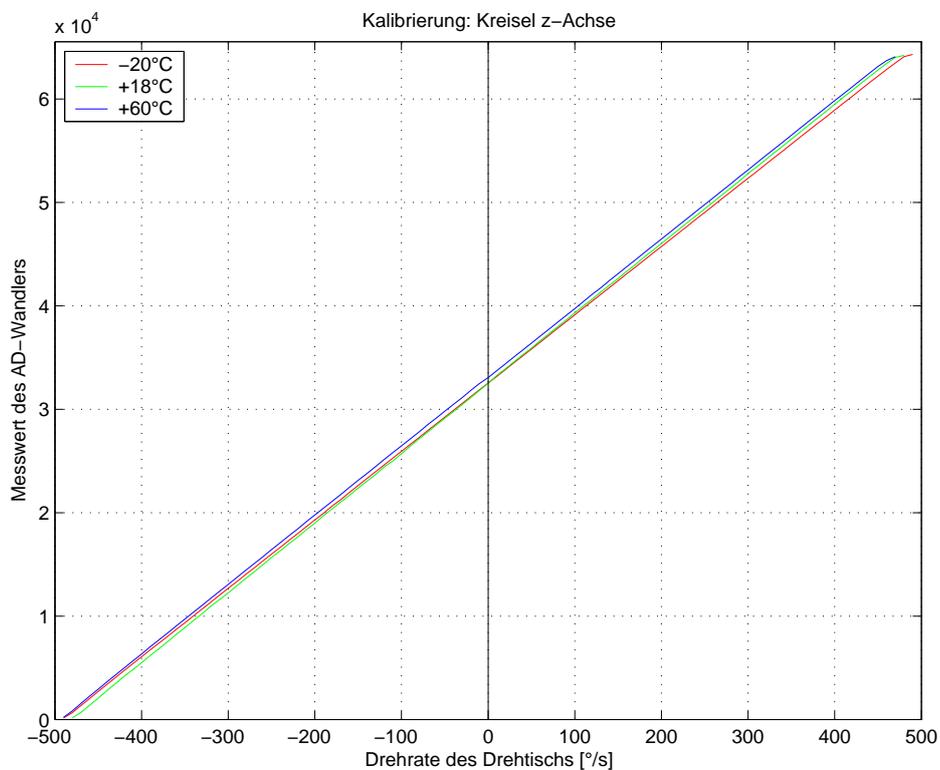


Abbildung 7.14: Messwerte des z-Achsen-Gyroskops bei verschiedenen Temperaturen. Die Kennlinie von $+18^{\circ}C$ ist von [Abbildung 7.10](#) übernommen.

7 Kalibrierung

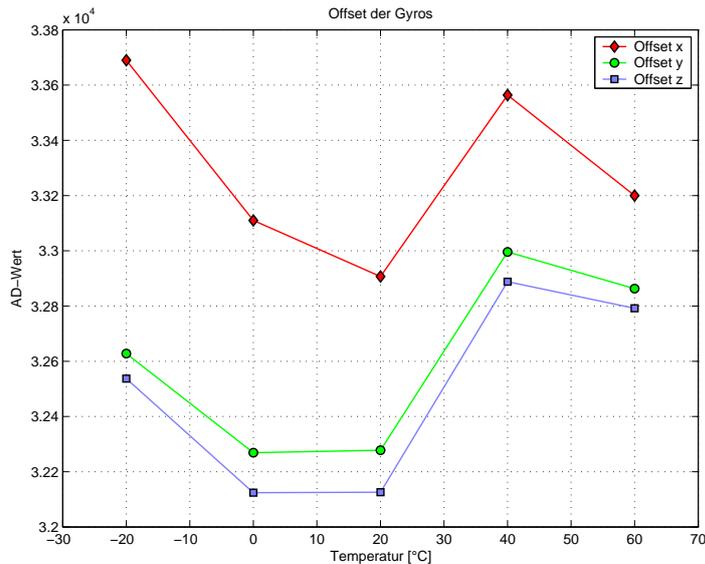


Abbildung 7.15: Werte des Nullpunkts der drei Gyroskope bei verschiedenen Temperaturen

Kennlinien mit den aufsteigenden Temperaturen überein. Die [Abbildung 7.14](#) zeigt das Verhalten des Gyroskops der z-Achse bei verschiedenen Temperaturen. Diese Messungen wurden aufgenommen, um die Temperaturkorrektionsfunktion validieren zu können.

Es ergibt sich also ein nichtlinearer Zusammenhang sowohl bezüglich des Offsetwerts, wie ihn die [Abbildung 7.15](#) zeigt, als auch bezüglich der Steigung der Kurven. Der temperaturabhängige Skalierungsfaktor ist für das Gyroskop der x-Achse in [Abbildung 7.16](#) dargestellt. Für den Offsetwert weisen die drei Sensoren qualitativ das gleiche Verhalten auf. Aus den abhängigen Nullpunkten und Faktoren folgt auch direkt, dass der Messbereich der Gyroskope nicht mehr bei $\pm 480^\circ/s$ liegt.

Verschiedene Messreihen bei Temperaturen von $15^\circ C$ – $25^\circ C$ haben ergeben, dass die Nullpunkte der Gyroskope starke Schwankungen aufweisen, die sich nicht durch einen einzigen temperaturabhängigen Zusammenhang beschreiben lassen. Die in [Abbildung 7.15](#) dargestellten Offsetwerte waren nicht mit hoher Genauigkeit reproduzierbar.

Aber schon geringe Abweichungen im Nullpunkt führen zu einem konstanten Drehratensignal, das durch die Aufintegration zu großen Fehlern führt. Für die Berechnungen in MATLAB wurde deshalb folgende Strategie verwendet:

Für die Temperatur-Korrektionsfunktion der Gyroskope wurde ein linearer Zusammenhang angenommen, wobei der Temperaturbereich im Intervall $0^\circ C$ – $20^\circ C$ die Basis für die lineare Funktion bildete.

Die Temperaturabhängigkeit des Skalierungsfaktors der Gyroskope wurde nicht berücksichtigt, da nicht alle drei Gyroskope temperaturabhängig vermessen wurden. Außerdem wirkt sich der Fehler durch die Skalierungen erst bei hohen Drehraten aus.

Mit den gewonnenen Kalibrierdaten und den getroffenen Annahmen ergibt sich somit die Kalibrierungsfunktion (7.5) für die Gyroskope, wobei b_i^* einen statischen Offset beschreibt.

7.5 Temperatur-Korrektur der Beschleunigungsmesser

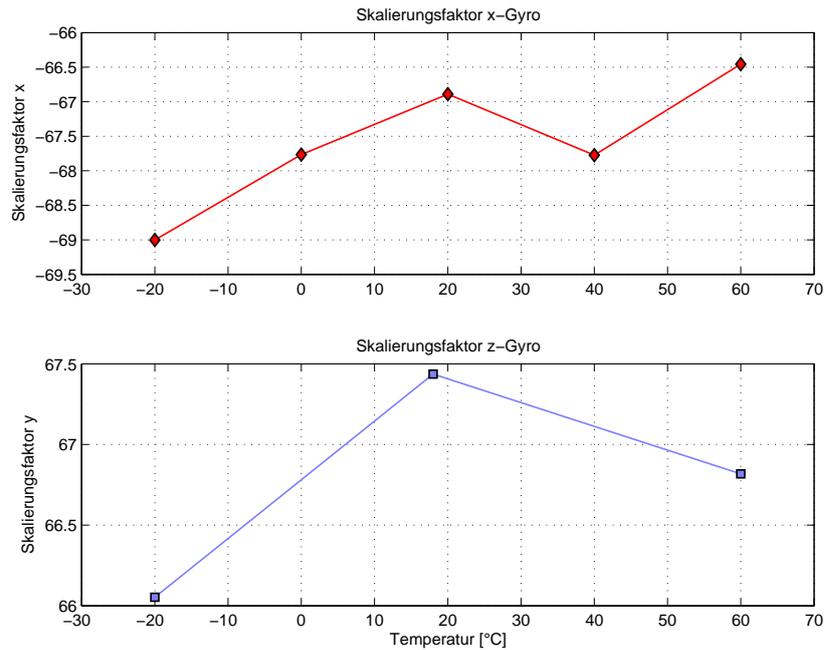


Abbildung 7.16: Skalierungsfaktor der Gyroskope (x-Achse oben, z-Achse unten) bei verschiedenen Temperaturen.

$$\begin{bmatrix} p(T_x, \omega_x) \\ q(T_y, \omega_y) \\ r(T_z, \omega_z) \end{bmatrix} = \frac{\pi}{180} \begin{bmatrix} (\omega_x - (32549 + 49,8 T_x)) / 66.885 \\ (\omega_y - (31742 + 50,4 T_y)) / 65.371 \\ (\omega_z - (30704 + 108,7 T_z)) / 67.437 \end{bmatrix} + \begin{bmatrix} b_p^* \\ b_q^* \\ b_r^* \end{bmatrix} \quad (7.5)$$

Die Werte b_i^* werden beim Einschalten der IMU, respektive anhand der ersten Messwerte der Datenaufzeichnung gewonnen. Unter der Annahme, dass sich die IMU beim Einschalten in Ruhe befindet und nicht bewegt wird, kann zwischen dem in [Gleichung 7.5](#) theoretischen Nullpunkt und den über einem Intervall (z.B. 100 Messungen, also 0,33s) gemittelten realen Messwerten eine Differenz gebildet werden, die direkt b_i^* entspricht.

7.5 Temperatur-Korrektur der Beschleunigungsmesser

Der in dieser Arbeit eingesetzte Beschleunigungssensor verfügt über keinen eigenen Temperaturfühler und dazugehörigen Ausgang. Es wäre jedoch möglich, einen Temperaturfühler separat auf dem Board in Nähe der Sensoren zu montieren oder hilfsweise als Näherung einen Temperaturfühler der Drehratensensoren zu verwenden. [WEINBERG \(b\)](#) schlägt weitere Methoden zur Kompensation vor. Er versucht, einen Fehler nicht rechnerisch zu minimieren, sondern der Fehlerursache am Sensor selbst durch Techniken wie Aufheizen auf konstante Temperatur entgegen zu wirken.

Aus den Messwerten, die bei den Versuchen in der Temperaturkammer im vorherigen Abschnitt gewonnen wurden, konnte auch ein Temperaturprofil der Beschleunigungssensoren gewonnen werden. Es wurde jeweils das Ausgangssignal der Beschleunigungssensoren in der

7 Kalibrierung

Tabelle 7.2: Pulslängen Ta und Tb der Beschleunigungssensoren in der Temperaturkammer für die Ruhelagen in Position $[\phi, \theta] = [0^\circ, 90^\circ]$

Temp. [$^\circ C$]	Ta_x	Tb_x	Ta_y	Tb_y	Ta_z	Tb_z
-20	30967	48054	25922	48034	27018	48629
0	30811	47765	25625	47768	26767	48421
20	30756	47542	25569	47564	26689	48368
40	30691	47339	25314	47354	26450	48209
60	30632	47067	24982	47066	26143	48085

Tabelle 7.3: Pulslängen Ta und Tb der Beschleunigungssensoren in der Temperaturkammer für die Ruhelagen in BNL $[\phi, \theta] = [0^\circ, 0^\circ]$

Temp. [$^\circ C$]	Ta_x	Tb_x	Ta_y	Tb_y	Ta_z	Tb_z
-20	24621	48025	25799	48051	33122	48632
20	24541	47583	25487	47568	32603	48361
60	24451	47071	24866	47087	32081	48067

Ruhelage vor Beginn einer Messreihe ausgewertet. Die in [Tabelle 7.2](#) aufgeführten Werte für Ta und Tb wurden ermittelt, die den in [Abbildung 7.18](#) gezeigten Verlauf des Quotienten Ta/Tb ergeben. Alle Werte repräsentieren die Lage $[\phi, \theta] = [0^\circ, 90^\circ]$ der IMU.

Aus den beiden Messungen in der BNL bei $60^\circ C$ und $-20^\circ C$ sowie der Kalibrierungsmessung aus [Abschnitt 7.2](#) bei $20^\circ C$ wurden die Werte in [Tabelle 7.3](#) gewonnen. Damit ergibt sich der in [Abbildung 7.18](#) dargestellte Verlauf für Ta/Tb .

Wie schon in [Abschnitt 7.2](#) dargelegt wurde, sind die Winkel ϕ und θ nahe BNL für die Stützung sehr wichtig. Daher wurde, aufbauend auf den in [Tabelle 7.3](#) aufgestellten Werten, eine Korrekturfunktion für den Offset der Beschleunigungssensoren ermittelt.

Die gewonnenen Regressionsgeraden stimmen in den in [Gleichung 7.1](#) gefundenen Werten überein und erweitern sinnvoll den kalibrierten Temperatureinsatzbereich.

Damit ergibt sich dann die folgende Kalibrierungsfunktion [\(7.6\)](#) für die Beschleunigungssensoren, die sich des in Grad Celsius umgerechneten Temperatursignals der Gyroskope bedient:

$$\begin{bmatrix} a_x(T) \\ a_y(T) \\ a_z(T) \end{bmatrix} = \begin{bmatrix} \frac{(Ta_x/Tb_x) - (0,5143 + 0,00008375T)}{-0,12860} \\ \frac{(Ta_y/Tb_y) - (0,5358 - 0,00011T)}{-0,12542} \\ \frac{(Ta_z/Tb_z) - (0,5634 - 0,0001713T)}{0,12450} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0,085 \end{bmatrix} \quad (7.6)$$

Zur Validierung der Gesamtfunktion wurde [\(7.6\)](#) für die verfügbaren Messpunkte ausgewertet. Das Resultat für die Position $[\phi, \theta] = [0^\circ, 90^\circ]$ ist in [Tabelle 7.4](#) dargestellt. Das Ergebnis für die BNL zeigt [Tabelle 7.5](#).

7.5 Temperatur-Korrektur der Beschleunigungsmesser

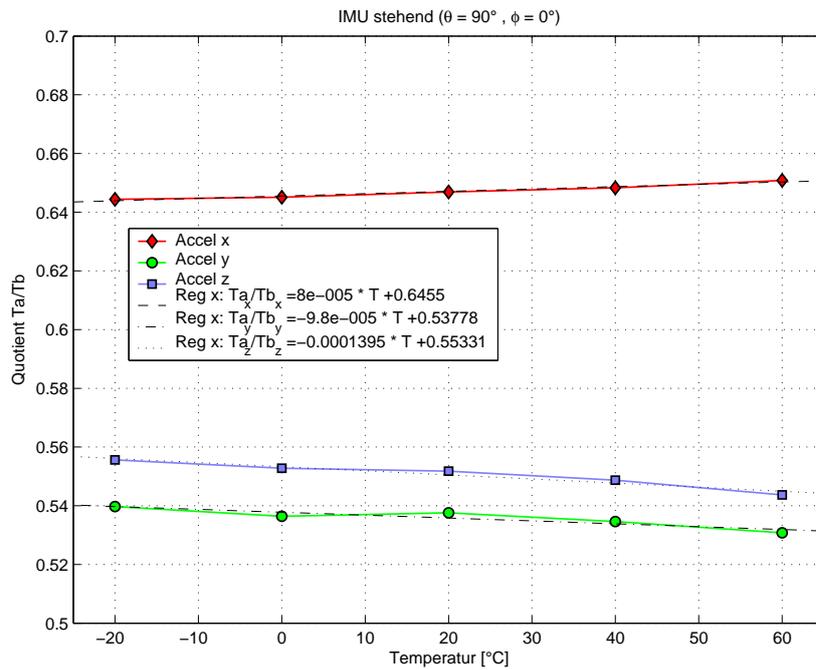


Abbildung 7.17: Verlauf des Quotienten Ta_i/Tb_i ($i = x, y, z$) der Beschleunigungssensoren bezüglich der Temperatur bei $[\phi, \theta] = [0^\circ, 90^\circ]$

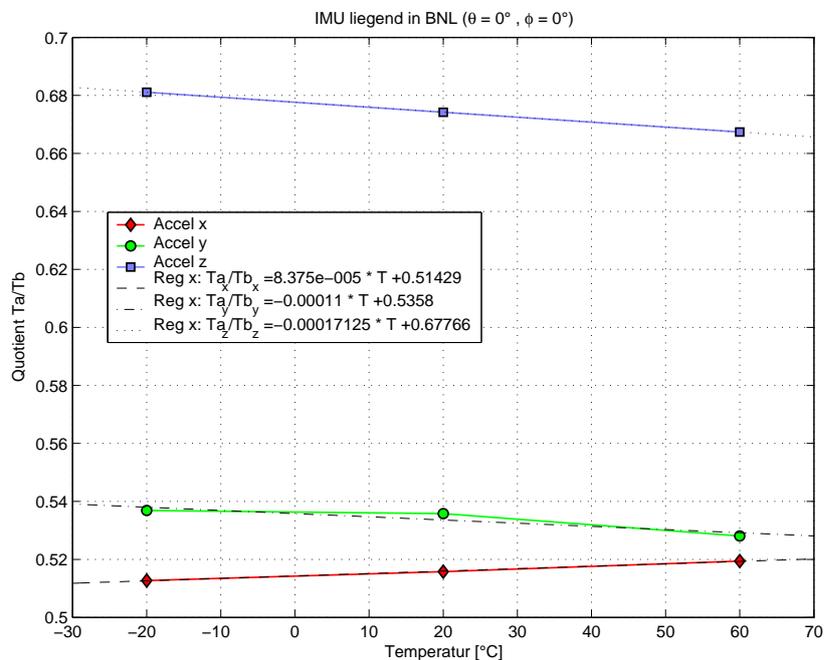


Abbildung 7.18: Verlauf des Quotienten Ta_i/Tb_i ($i = x, y, z$) der Beschleunigungssensoren bezüglich der Temperatur in BNL

7 Kalibrierung

Tabelle 7.4: Validierung der Funktion (7.6) zur Temperatur-Korrektur der Beschleunigungssensoren bei $\phi = 0^\circ$ und $\theta = 90^\circ$.

Temp. [$^\circ C$]	a_x [g]	a_y [g]	a_z [g]
-20	-1,0249	-0,0132	-0,0052
0	-1,0167	-0,0052	-0,0002
20	-1,0183	-0,0317	0,0193
40	-1,0161	-0,0253	0,0216
60	-1,0225	-0,0126	0,0092

Tabelle 7.5: Validierung der Funktion (7.6) zur Temperatur-Korrektur der Beschleunigungssensoren in BNL.

Temp. [$^\circ C$]	a_x [g]	a_y [g]	a_z [g]
-20	-0,0004	0,0087	1,0027
20	0,0017	-0,0176	1,0021
60	-0,0010	0,0089	1,0031

7.6 Ausrichtungsfehler

Für die Ausrichtungsfehler gibt es verschiedene Gründe. Zum einen ist das Die¹ im Sensorgehäuse nicht exakt ausgerichtet. Der Hersteller gibt einen Fehler von bis zu einem Grad an.

Zusätzlich kommt der Fehler hinzu, der durch das Auflöten des Sensors entsteht. Die hier vorgestellten Sensoren mit BGA-Gehäuse bzw. Löt pads an der Unterseite werden im Löt-ofenschmelzverfahren montiert. Hierbei wird der Sensor auf ein mit Löt zinn benetztes Feld gesetzt, und dann die Trägerplatine mitsamt den Sensoren soweit erhitzt, bis das Löt zinn zwischen Träger und Sensor schmilzt und diese sowohl elektrisch als auch mechanisch verbindet. Dabei kann es einerseits zu einem Verdrehen des Sensors bezüglich der Messachse kommen, da keine mechanische Führung vorhanden ist. Sollte der Sensor nicht gleichmäßig in das flüssige Löt zinn einsinken, so kann es andererseits passieren, dass der Sensor nicht plan auf dem Träger aufliegt. Für die Beschleunigungssensoren im LLC-8-Gehäuse gilt Ähnliches, jedoch weist GEITNER darauf hin, dass sich der Sensor beim Einlöten bedingt durch die Oberflächenspannung des Löt zinns selbsttätig gegenüber den auf der Platine vorgesehenen Löt pads ausrichtet.

Als dritte Fehlerquelle tritt bei dem vorgestellten IMU-Prototyp der Montagesockel bzw. die 90°-IC-Fassung auf, die in der Lochrasterplatine nur schwer exakt auszurichten sind, da die Löt beinchen einen wesentlich kleineren Durchmesser als die Löcher der Lochrasterplatine aufweisen. Die Abbildungen 4.4 und 4.6 demonstrieren diese Problematik. Schließlich kann sich auch noch die Platine selbst verziehen.

Im Folgenden wird ein Verfahren vorgestellt, mit dem diese Ausrichtungsfehler rechnerisch bestmöglich eliminiert werden können. Die Transformationsmatrizen T_{gyro} (Gleichung (3.5))

¹Auf dem aus Silizium hergestellten „Die“ ist der eigentliche Sensor aufgebaut. Das Die wird dann aus Schutzgründen im Gehäuse vergossen.

und T_{acc} (Gleichung (3.6)) werden in diesem Schritt quantitativ bestimmt, und das Ergebnis fließt in den IMU-Algorithmus in MATLAB (siehe ??) ein.

Wird das in [Abschnitt 4.4](#) eingeführte IMU-Board verwendet, so würden die gewonnenen Matrizen z. B. im EEPROM des Mikrocontrollers nach der Kalibrierung abgelegt, so dass das Programm nicht für jede IMU verändert werden muss.

7.6.1 Gyroskope

Methoden

Die Matrix T_{gyro} besitzt neun zu bestimmende Elemente. Für die folgenden Berechnungen kann angenommen werden, dass die Matrix eine dominante Diagonale und nur kleine Nebendiagonaleinträge aufweist, was sich aus der bi-orthogonalen Anordnung der Sensoren ergibt. Daraus wird geschlossen, dass T_{gyro} stets vollen Rang aufweist.

Montiert man die IMU auf einem Drehtisch, so dass die z-Achse der IMU mit der z-Achse des Drehtisches zusammenfällt, so erwartet man nur ein Signal des z-Gyroskops, während das x- und y-Gyroskop kein Ausgangssignal aufweisen sollen. Mittelt man nun über ausreichend viele Messwerte und bei gleichzeitiger Annahme, dass in diesem Zeitraum keine Drift auftritt, so lässt sich folgender Zusammenhang für die z-Achse aufstellen:

$$\begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = \mathbf{T}_{gyro} \begin{bmatrix} \bar{\omega}_x \\ \bar{\omega}_y \\ \bar{\omega}_z \end{bmatrix}_r \Rightarrow \Omega_r = \mathbf{T}_{gyro} \bar{\omega}_r \quad (7.7)$$

Führt man die Messung nun für die beiden weiteren Achsen durch, so ergibt sich folgendes Gleichungssystem:

$$\begin{bmatrix} \Omega_p \\ \Omega_q \\ \Omega_r \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{gyro} & 0 & 0 \\ 0 & \mathbf{T}_{gyro} & 0 \\ 0 & 0 & \mathbf{T}_{gyro} \end{bmatrix} \begin{bmatrix} \bar{\omega}_p \\ \bar{\omega}_q \\ \bar{\omega}_r \end{bmatrix} \quad (7.8)$$

Das lineare Gleichungssystem (7.8) besitzt 9 Unbekannte und 9 Gleichungen. Mit den oben getroffenen Annahmen lässt sich dieses System stets lösen und man erhält die Einträge von \mathbf{T}_{gyro} . Bedingt durch den konstanten Ausrichtungsfehler und den sich dadurch ergebenden linearen Zusammenhang reicht eine Messung je Messachse zur Bestimmung von \mathbf{T}_{gyro} aus.

Messreihe

Um die Ausrichtungsfehler quantifizieren zu können, wurden die Messwerte der Kalibrierungsreihe aus [Abschnitt 7.3](#) verwendet. Dabei wurde der Signalsprung von $300^\circ/\text{s}$ zur Ruhelage herangezogen. Um die Drehraten der nicht in der Messachse liegenden Sensoren zu bestimmen, wurde, um z. B. den Fehler des x-Gyros bei Messung der z-Achse zu finden, die Differenz $\omega_{x300} - \omega_{xRuhelage}$ an der Signalsprungstelle durch den in (7.3) gefundenen Skalierungsfaktor geteilt.

In [Tabelle 7.6](#) sind die gemittelten Messwerte aufgeführt.

7 Kalibrierung

Tabelle 7.6: Messreihe der Gyro-Ausrichtungsfehler

Mess-Achse i	Ω_i [°/s]	$\overline{\omega_x}$ [°/s]	$\overline{\omega_y}$ [°/s]	$\overline{\omega_z}$ [°/s]
x	300	300	-0,44	2,16
y	300	-5,40	300	0,15
z	300	-1,40	-3,18	300

Auswertung

Das ?? zeigt das Maple-Arbeitsblatt mit den ermittelten Messwerten und den ausgeführten Rechenschritten. Damit ergibt sich die Matrix² \mathbf{T}_{gyro} zu:

$$\mathbf{T}_{gyro} = \begin{bmatrix} 1,0000 & 0,1800 & 0,0049 \\ 0,1390 & 1,0000 & 0,1061 \\ -0,0072 & 0,0001 & 1,0000 \end{bmatrix} \quad (7.9)$$

7.6.2 Beschleunigungsmesser

Methode

Analog zu der oben vorgestellten Methode für die Drehratensensoren kann ein abgewandeltes Verfahren auch für die Beschleunigungssensoren verwendet werden. Anstatt einer Drehrate wird nun die Erdschwerebeschleunigung statisch in allen drei Achsen gemessen.

Messreihe

In [Tabelle 7.7](#) sind die Messungen der Beschleunigungssensoren dargestellt. Als Sollgröße wurde jeweils das Erdschwerebeschleunigungsfeld mit $\pm 1g$ verwendet.

Tabelle 7.7: Messreihe der Beschleunigungssensor-Ausrichtungsfehler, alle Werte nach temperaturkompensierter Kalibrierung bei 20 °C.

Achse	Beschleunigung [g]	$\overline{a_x}$ [g]	$\overline{a_y}$ [g]	$\overline{a_z}$ [g]
x	-1,0	-1,0183	0,0040	0,0193
y	1,0	-0,0224	0,9846	-0,0454
z	1,0	0,0017	0,0022	1,0021

Auswertung

Das ?? zeigt das Maple-Arbeitsblatt mit den ermittelten Messwerten und den ausgeführten Rechenschritten. Damit ergibt sich die Matrix \mathbf{T}_{acc} zu:

²auf vier Stellen gerundet, da durch den 16-bit-AD-Wandler keine höhere Genauigkeit garantiert werden kann. In Anbetracht des in [Abschnitt 7.1](#) gefundenen Rauschwertes von 0,4%FS wäre eine Rundung auf zwei Stellen auch korrekt.

$$\mathbf{T}_{acc} = \begin{bmatrix} 0,9821 & 0,0222 & -0,0017 \\ 0,0039 & 1,0156 & -0,0022 \\ 0,0191 & 0,0464 & 0,9978 \end{bmatrix} \quad (7.10)$$

7.7 Zentrifuge

Um die Fehler der Gyroskope noch weiter zu kompensieren, müsste noch der Einfluss der Beschleunigung auf die Drehratensensoren bestimmt werden. Auf dem Drehtisch erfährt der Sensor keine nennenswerte Beschleunigung, wenn er nahe der Drehachse des Drehtisches montiert ist. Damit sowohl eine konstante Drehrate als auch eine Beschleunigung auf den Sensor wirkt, verwendet man typischerweise eine Zentrifuge.

Nachdem die Drehratensensoren kalibriert und temperaturkompensiert sind, so dass sie auf dem Drehtisch das Wunschergebnis ausgeben, kann der Fehler bestimmt werden, der durch die Zentrifugalbeschleunigung induziert wird.

Wenn ein analytischer Zusammenhang gefunden wird, könnten die in der IMU eingebauten Beschleunigungssensoren aufgrund der räumlichen Nähe verwendet werden, um das Signal zu liefern, das über den gefundenen Zusammenhang das Korrektursignal liefert.

Der im Rahmen dieser Studienarbeit entwickelte Prototyp wurde nicht auf einer Zentrifuge vermessen. Diese genauere Untersuchung wäre Gegenstand einer weiterführenden Arbeit mit dem Prototypen.

7.8 Vibrationen

7.8.1 Gyroskope

GEEN UND KRAKAUER (2003) untersuchten das Verhalten der Gyroskope bei Vibrationen. Sie kamen zum Ergebnis, dass die ADXRS-Gyroskope weitgehend resistent gegen Vibrationen sind. Dieser Befund konnte bei den im Rahmen dieser Arbeit durchgeführten Kalibriermessungen und Versuchen bestätigt werden.

7.8.2 Beschleunigungssensoren

Die Beschleunigungssensoren reagieren hingegen sehr empfindlich auf Vibrationen. Vor allem ändert sich bei Vibrationen die Zeit T_b , obwohl diese laut Datenblatt konstant sein sollte. Die [Abbildung 7.19](#) zeigt beispielhaft einen Ausschnitt aus der Kalibriermessung aus [7.2](#). Damit wird die Berechnung des Quotienten aus T_a/T_b sehr schwierig. Es wäre sinnvoll, einen digitalen Tiefpass vor die Berechnung des Quotienten einzufügen, der T_a und T_b ausreichend glättet.

7 Kalibrierung

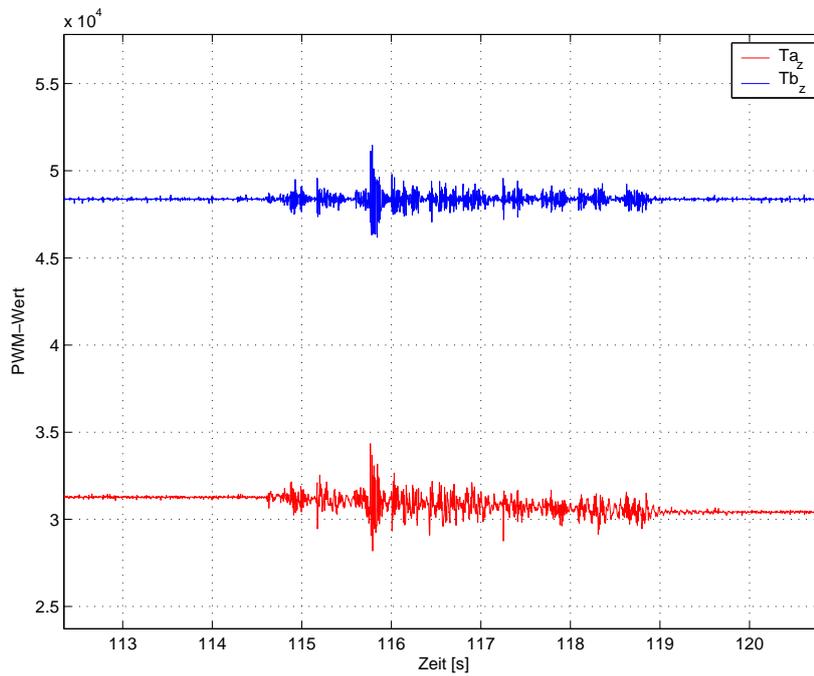


Abbildung 7.19: Pulsweitensignal des Beschleunigungssensors der z-Achsen bei Vibrationen. Bis $t=114,5$ s lag der Sensor statisch fest. Im Zeitintervall 114,4s – 119s wurde die Lage des Sensors auf der schiefen Ebene um 10° verändert. Ab $t=119$ s lag die nächste statische Lage vor.

8 Versuchsaufbau

In diesem Kapitel werden die Versuchsaufbauten und Konfigurationen zur Datengewinnung beschrieben. Die gewonnenen Daten dienen im Fall des ROSEMOUNT-Kreisels als Referenz, andererseits bilden sie die Datengrundlage für das Tuning des in [Kapitel 5](#) beschriebenen Filters.

Für alle im Folgenden beschriebene Versuche, bei denen die MICROSTRAIN-IMU beteiligt war, wurde an ihr das Datenformat „Euler-Winkel und Vektoren“ ausgewählt. Es liefert sowohl die von der MICROSTRAIN-IMU berechneten Eulerwinkel θ , ψ und ϕ , als auch die skalierten und temperaturkompensierten Rohwerte, aus denen unter Verwendung des in dieser Arbeit entwickelten Algorithmus im Postprocessing ebenfalls Eulerwinkel berechnet werden.

8.1 Datenlogger

Für alle Messungen, sowohl auf den Drehtischen, als auch in den Flugzeugen, wurde ein selbst entwickelter Datenlogger verwendet. Er basiert – wie auch der IMU-Prototyp – auf einem Atmel ATmega32 Mikrocontroller und zeichnet auf einer MMC-Speicherkarte auf. Die für die Messungen benutzte 128-MB-MMC-Karte ist für einen Einsatzbereich von -25°C bis $+70^{\circ}\text{C}$ ausgelegt.

Der Datenlogger liest die Messdaten aus dem IMU-Prototyp über die eingebaute I²C-Schnittstelle aus. Dabei fungiert der Datenlogger als I²C-Master, während der IMU-Prototyp die Messdaten als I²C-Slave in seinen I²C-Registern bereitstellt.

Ferner ist auf dem Datenlogger noch ein Pegelwandler für die serielle RS-232-Schnittstelle vorhanden. Über diese wurde bei den Messungen die MICROSTRAIN-IMU angeschlossen, wobei die maximale Datenrate (115,2 kBaud) verwendet wurde, die die MICROSTRAIN-IMU unterstützt.

Die beiden analogen Ausgangsspannungen des ROSEMOUNT-Kreisels wurden mittels des im Mikrocontroller eingebauten 10-bit-AD-Wandlers erfasst. Als Referenzspannung für den mechanischen Kresel und den AD-Wandler wurde die stabilisierte 5 VDC herangezogen, die den Datenlogger versorgt. Der AD-Wandler wurde auf einen freilaufenden Betrieb eingestellt, d. h. die Wandlungen laufen unabhängig vom Hauptprogramm und nach erfolgter Wandlung wird der Wert innerhalb einer Interrupt-Routine in eine globale Variable gespeichert, die vom Hauptprogramm neben den Werten der MICROSTRAIN-IMU und des IMU-Prototyps als ein Datensatz aufgezeichnet wird.

Die Aufzeichnungsrate liegt bei 300 Hz für den IMU-Prototyp und den mechanischen Kresel. Die MICROSTRAIN-IMU wird mit ihrer maximal möglichen Ausleserate betrieben: Nachdem jeweils das Startkommando an die MICROSTRAIN-IMU gesendet wurde, wird, während die

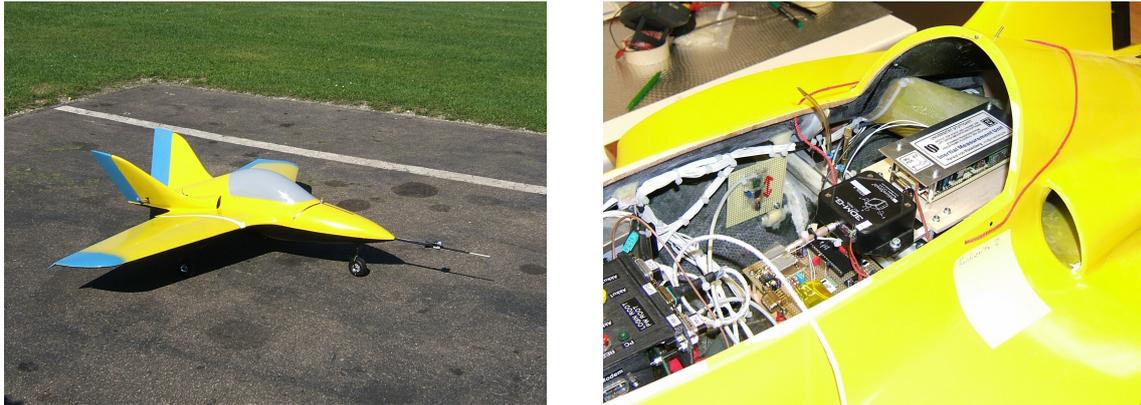


Abbildung 8.1: Flugzeug-Prototyp „Deltajet“ zur Flugerprobung mit im Rumpf montierter MICROSTRAIN-IMU und dem IMU-Prototyp

Aufzeichnung des Prototypen und des Kreisels weiterläuft, in einer Interrupt-Routine die Antwort der MICROSTRAIN-IMU eingelesen. Nachdem ein gültiger Datensatz empfangen wurde, wird dieser Wert synchron zum Prototypen und Kreisler aufgezeichnet, bis wiederum die nächste gültige Antwort eintrifft. In der Praxis ergibt sich somit eine nutzbare Signalrate von 75 Hz für die MICROSTRAIN-IMU.

Der Datenlogger zeichnet die Messwerte in einem selbst definierten Binärformat auf die Speicherkarte. Auf dem Computersystem, auf dem die Auswertung in MATLAB ausgeführt wird, liest ein selbst erstelltes Perl-Script diese Binärdatei ein und konvertiert die Binärdaten in ein ASCII-Format, das seinerseits von MATLAB importiert wird.

8.2 IMU-Prototyp und Microstrain-IMU

Zeitgleich mit der Vermessung des IMU-Prototyps entstand am Steinbeis-Zentrum ein Flugzeug-Prototyp. Freundlicherweise konnte ich den IMU-Prototyp in dieses Versuchsflugzeug einbauen und während zweier Messflüge Daten aufzeichnen. Die [Abbildung 8.1](#) zeigt das Erprobungsmodell und das darin eingebaute Messsystem sowie den IMU-Prototypen.

Die MICROSTRAIN-IMU, die Teil des Messsystems zur Flugerprobung war, wurde mit 84 Hz vom Bordcomputer aufgezeichnet, mit dem auch die anderen Messdaten des Flugzeug-Prototyps „Deltajet“ während des Versuchsflugs aufgezeichnet wurden. Damit dieses abgeschlossene Messsystem nicht modifiziert werden musste, wurde der IMU-Prototyp mit dem oben vorgestellten Datenlogger separat aufgezeichnet.

Die beiden Aufzeichnungen wurden synchron gestartet. Die Synchronität nach dem Start ist durch den Systemtakt der beiden Mikrocontroller in den beiden Messsystemen gewährleistet, deren interne Systemzeiten vom jeweiligen Logger mit aufgezeichnet wurden und für die spätere Auswertung zur Verfügung standen.

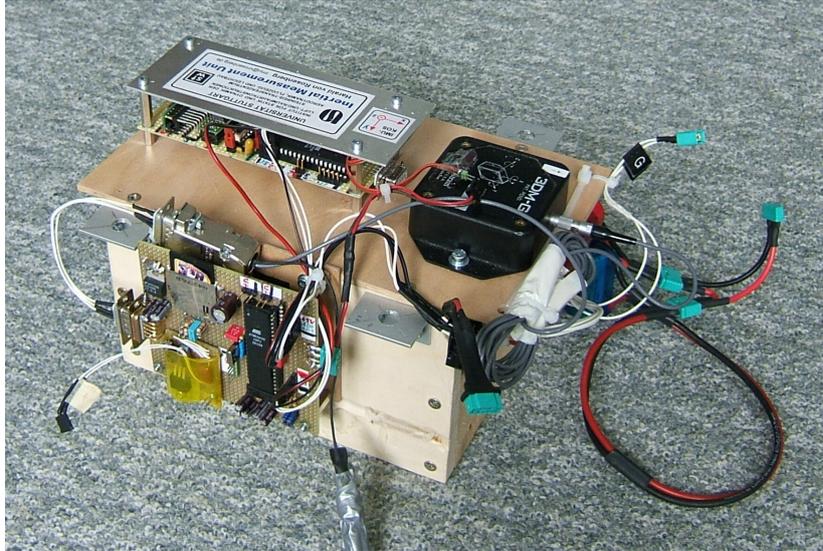


Abbildung 8.2: Versuchsaufbau mit ROSEMOUNT-Kreisel (innerhalb der Box eingebaut), MICROSTRAIN-IMU und IMU-Prototyp. An der Seite ist der Datenlogger montiert, auf der gegenüberliegenden Seite die Akkupacks, die die Betriebsspannung von 28 V DC für den Kreisel liefern.

8.3 IMU-Prototyp, Microstrain-IMU und Kreisel

Mit den oben vorgestellten Messaufbauten können nur Vergleiche zur MICROSTRAIN-IMU gewonnen werden, jedoch keine qualitativen Aussagen bezüglich des Referenzsystems. Deshalb wurden in einem weiteren Aufbau alle drei hier betrachteten Systeme auf einen Versuchsträger montiert. Die [Abbildung 8.2](#) zeigt, wie der mechanische ROSEMOUNT-Kreisel, der IMU-Prototyp und die MICROSTRAIN-IMU auf einer gemeinsamen Plattform montiert wurden. Mit Hilfe des oben vorgestellten Datenloggers werden die analogen Ausgangsspannungen des Kreisels und die digitalen Daten der MICROSTRAIN-IMU aufgezeichnet. Synchron dazu zeichnet der Datenlogger die Rohwerte der Sensoren auf, die der IMU-Prototyp liefert.

8.3.1 Messflug

Mit diesem Versuchsaufbau wurde ein Messflug mit einer *Piper PA-28 Cadet* ([Abbildung 8.3](#)) durchgeführt. Der mechanische Referenzkreisel, die MICROSTRAIN-IMU und der IMU-Prototyp wurden, wie in [Abbildung 8.4](#) gezeigt, im Flugzeug mitgeführt. Der Versuchsflug ging von Stuttgart/Pattonville nach Aalen und wieder zurück. Es wurde sowohl der Hinflug als auch der Rückflug mit dem Versuchsaufbau aufgezeichnet. Während des Flugs wurden jeweils drei verschiedene Manöver geflogen:

- Rollbewegungen mit einem Rollwinkel von ca. $\pm 20^\circ$
- Steigen und Sinken mit einem Nickwinkel von ca. $\pm 7^\circ$
- eine durch Gieren eingeleitete Dutch-Roll-Schwingung



Abbildung 8.3: Piper PA-28 Cadet, die für den Messflug verwendet wurde.

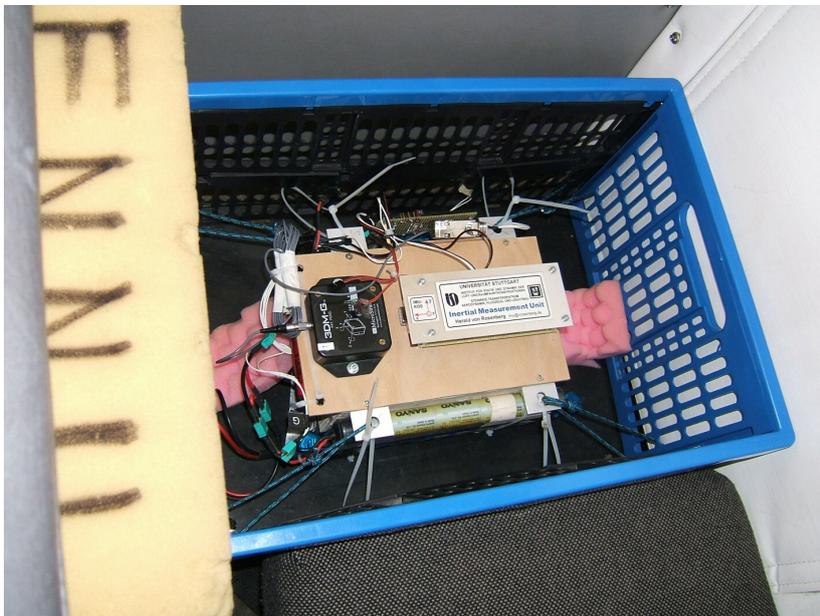


Abbildung 8.4: Der Messaufbau wurde in einer Transportbox mit Kabelbindern und Tam-
pen fixiert, die im Gepäckraum des Flugzeugs so arretiert wurde, dass eine
Eigenbewegung unmöglich war.

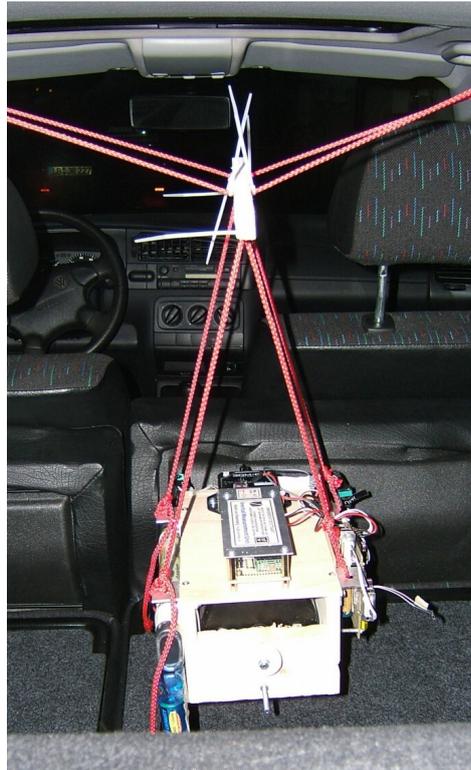


Abbildung 8.5: Versuchsaufbau beim Fahrversuch: Die Einheit aus mechanischem Kreisel, MICROSTRAIN-IMU, IMU-Prototyp und Datenlogger ist an einem Pendel in einem PKW aufgehängt.

Zusätzlich traten auf dem Rückflug Luftturbulenzen auf, die eine Beschleunigung in z-Achse bei gleichzeitiger Pitchänderung verursachten.

8.3.2 Bodentest

Neben dem Messflug wurde auch ein Bodentest durchgeführt. Dabei wurde der in [Abbildung 8.2](#) gezeigte Versuchsaufbau von Hand in verschiedene Lagen gebracht. Nach einer längeren stationären Phase (90 s), in der das Driftverhalten der Sensoren beobachtet werden konnte, wurde der Aufbau in verschiedenen Roll- und Pitchlagen gebracht, die jeweils einen Moment gehalten wurden. Danach folgte eine Phase, in der der Aufbau schnell um alle Achsen gedreht und bewegt wurde. In dieser Phase wurden auch lineare Beschleunigungen aufgebracht. Nach einer weiteren Phase langsamer Bewegungen folgte eine weitere Sequenz aus per Hand aufgebrachter Beschleunigungen und Rotationen. Abschließend wurde der Aufbau wieder in der Ausgangslage abgelegt.

8.3.3 Fahrversuch

Als dritte Versuchsart wurde noch ein Fahrversuch durchgeführt. Diese Messungen, den Versuchsaufbau zeigt [Abbildung 8.5](#), lieferte gänzlich verschiedene Rohdaten im Vergleich zu dem Flugversuch:

8 Versuchsaufbau

- Harte Stöße von der Strasse werden fast ungedämpft an den Messaufbau weitergereicht.
- Der Versuchsaufbau schwingt an dem Pendel mit einer viel höheren Frequenz und Amplitude, als diese im Flugzeug auftritt.
- Die Schwingung liegt nicht nur in einer Ebene. Es treten kombinierte Roll- und Pitch-Schwingungen auf, die von einer Schlingerbewegung um die Hochachse überlagert sind.

9 Ergebnisse

Das in [Kapitel 5](#) vorgestellte KALMAN-Filter muss für den jeweiligen Anwendungsfall eingestellt werden. Insbesondere müssen die beiden Matrizen \mathbf{Q} und \mathbf{R} den jeweiligen Sensoren und dem Systemverhalten angepasst werden. Das Tuning des Algorithmus wurde anhand der Daten aus dem Bodentest ([Unterabschnitt 8.3.2](#)) vorgenommen.

Die experimentell ermittelte Einstellung, bei der die Signale möglichst nahe an der Referenz liegen, wurde anschließend verwendet, um die Qualität der IMU-Signale bei den weiteren in [Kapitel 8](#) vorgestellten Messungen zu evaluieren. Die Ergebnisse wurden im Postprocessing unter MATLAB errechnet. Die Auswertung der Versuche aus [8.2](#) diente der qualitativen Beurteilung des IMU-Prototypen-Ergebnisses bezüglich der MICROSTRAIN-IMU.

Die nachstehenden Diagramme sind mit folgender Farbcodierung versehen, damit die Signalverläufe gut unterscheidbar sind:

	<i>(Kreisel)</i>	Lagewinkel des mechanischen ROSEMOUNT-Kreisels.
	<i>(3DMG)</i>	Eulerwinkel der MICROSTRAIN-IMU, wie sie die 3DM-GX1-IMU ausgibt.
	<i>(HvR-3DMG)</i>	Eulerwinkel, berechnet aus den MICROSTRAIN-Messwerten mittels des hier vorgestellten KALMAN-Filters.
	<i>(HvR-IMU)</i>	Winkel, berechnet aus den Messwerten des IMU-Prototyps, nachdem die Rohwerte wie in Kapitel 7 skaliert und normiert wurden.

9.1 Einfacher IMU-Algorithmus

Die nachfolgenden Diagramme stellen die Ergebnisse unter Verwendung des in [Abschnitt 5.2](#) vorgestellten Algorithmus dar. Die Matrizen \mathbf{Q} und \mathbf{R} wurden für die Messwerte des IMU-Prototyps und der MICROSTRAIN-Sensordaten zu $\mathbf{Q} = 0,05 \mathbf{I}^{3 \times 3}$ und $\mathbf{R} = 30 \mathbf{I}^{3 \times 3}$ angesetzt.

Die [Abbildung 9.1](#) und [9.2](#) zeigen die Ergebnisse des Bodentests. Die Auswertung des ersten Messflugs zeigen die beiden [Abbildungen 9.3](#) und [9.4](#).

Da der Algorithmus zu empfindlich auf Nullpunktfehler der Gyroskope reagiert, wurden die anderen Versuche (Flugzeug, Modell, Fahrversuch) nur mit der erweiterten Version ausgewertet. Die Ergebnisse sind im folgenden Abschnitt dargestellt.

9 Ergebnisse

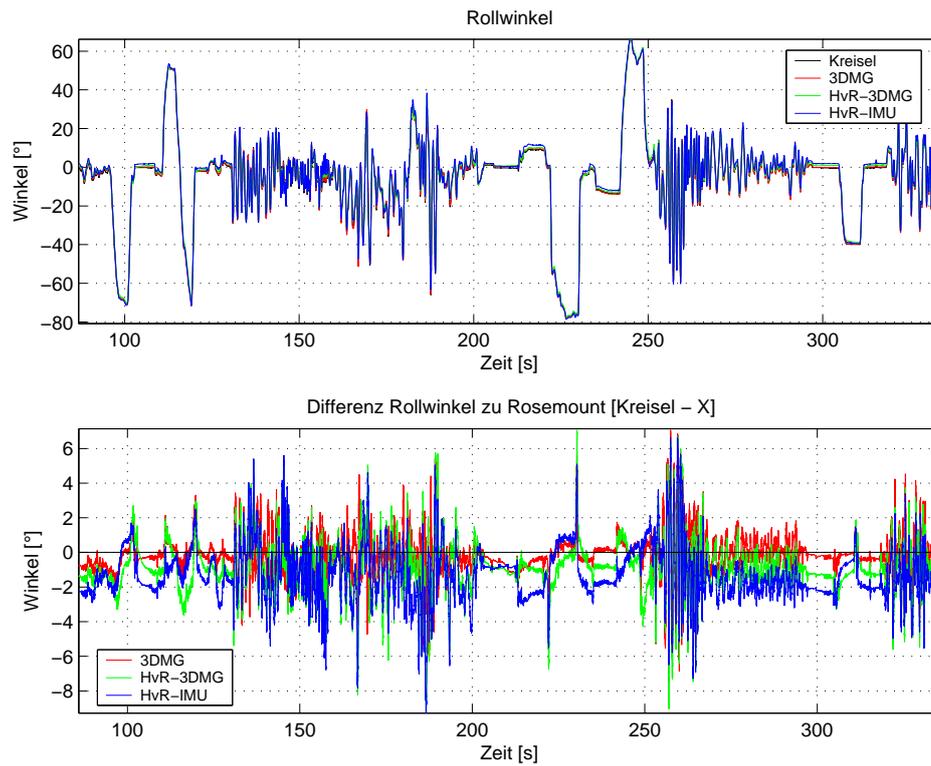


Abbildung 9.1: Vergleich der berechneten Roll-Winkel mit Referenz und MICROSTRAIN-IMU des Bodentests

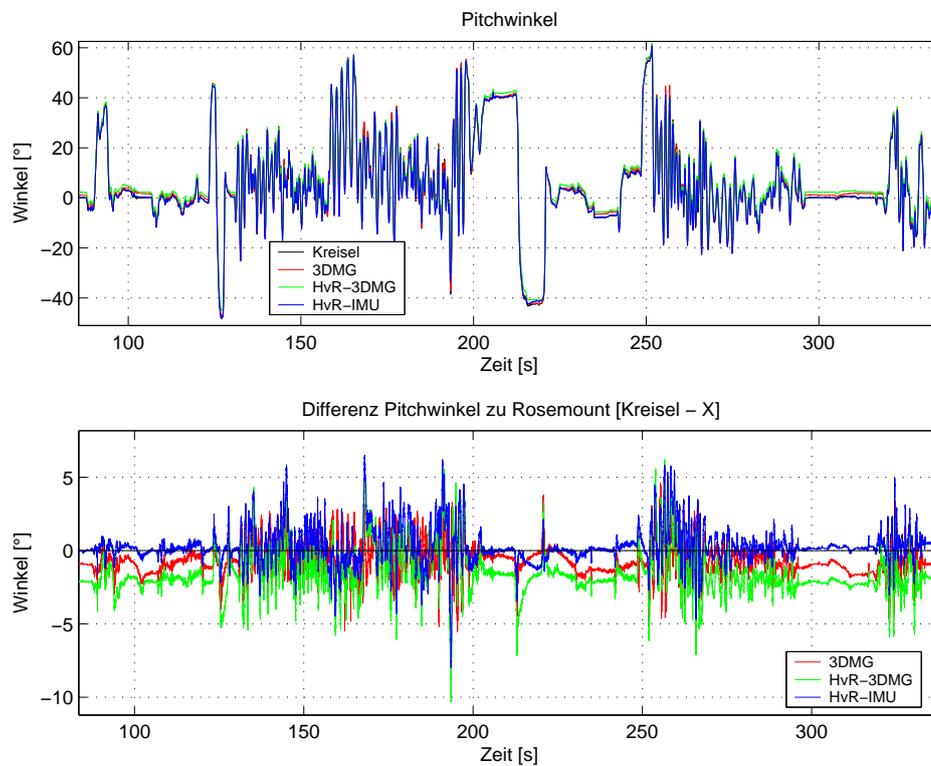


Abbildung 9.2: Vergleich der berechneten Pitch-Winkel mit Referenz und MICROSTRAIN-IMU des Bodentests

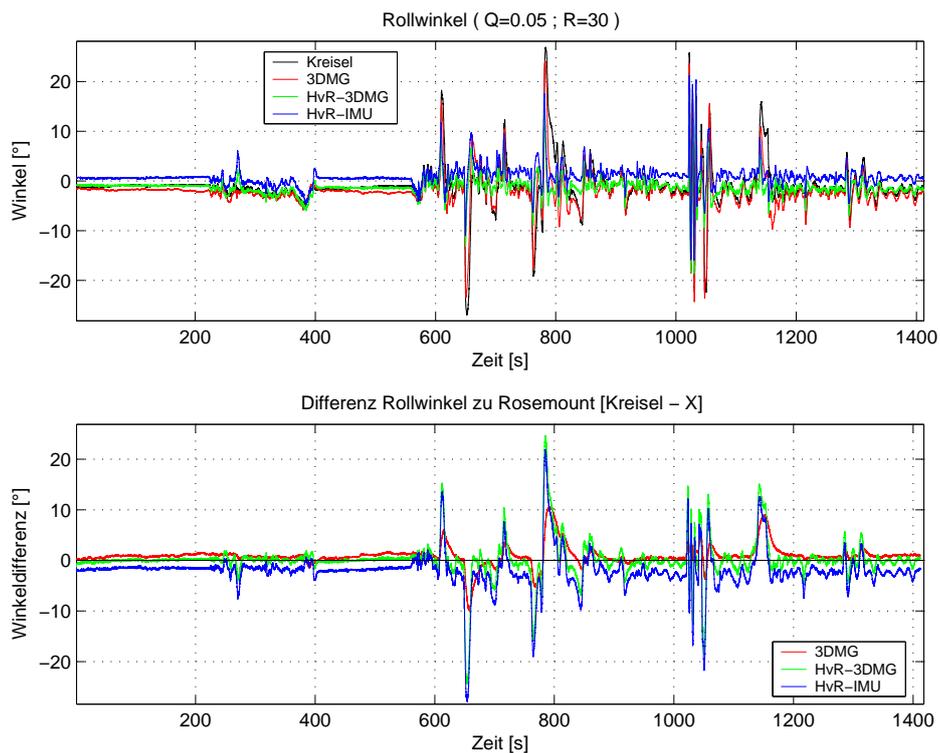


Abbildung 9.3: Vergleich der berechneten Roll-Winkel mit Referenz und MICROSTRAIN-IMU des ersten Messflugs

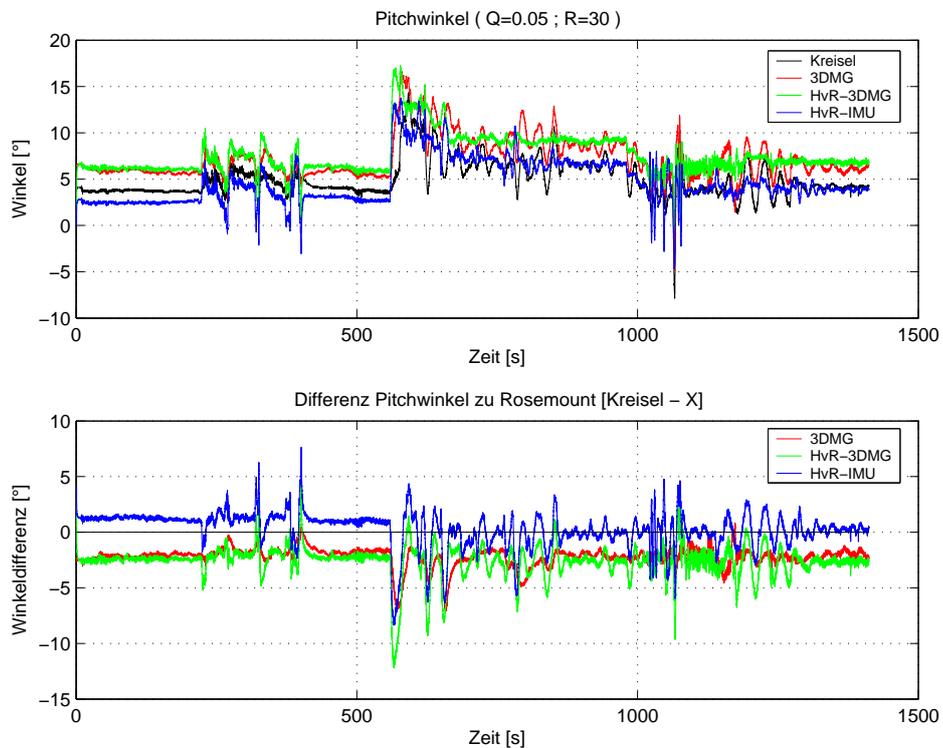


Abbildung 9.4: Vergleich der berechneten Pitch-Winkel mit Referenz und MICROSTRAIN-IMU des ersten Messflugs

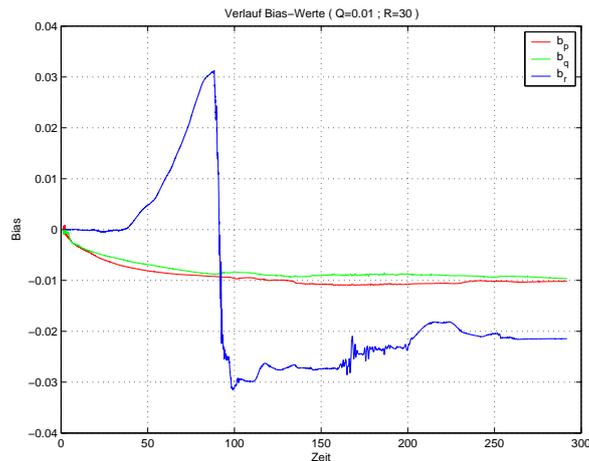


Abbildung 9.5: Verlauf der Nullpunktswerte b_p , b_q und b_r des erweiterten Systems beim Bodentest

9.2 Erweiterter IMU-Algorithmus

Die nachfolgenden Ergebnisse wurden mit dem um die drei Nullpunktsfehler erweiterten Algorithmus aus [Abschnitt 5.5](#) berechnet. Für die Matrix \mathbf{Q} und \mathbf{R} wurden andere Werte als im Fall des einfachen Systems verwendet. Der erweiterte Filter verfolgt zuverlässig den Nullpunkt der Gyroskope. Allerdings treten auf den Signalen b_p , b_q und b_r zum Teil beträchtliche Sprünge auf, die wohl nicht der eigentlichen Drift entsprechen: Sie resultieren aus der fehlenden Beobachtbarkeit des z -Achsen-Gyroskops, dessen Nullpunkt erst bei auftretender Bewegung um die z -Achse korrigiert werden kann. Dieses Verhalten tritt vor allem in der BNL auf, in der keine Stützung für das Gyroskop der z -Achse verfügbar ist. Ein exemplarisches Beispiel stellt die [Abbildung 9.5](#) dar, die den Verlauf von b_i , ausgewertet für den Bodentest, zeigt.

9.2.1 Bodentest

Die Auswertungen der Messdaten des Bodentests zeigen die beiden Abbildungen [9.6](#) und [9.7](#). Für die MICROSTRAIN-IMU und den IMU-Prototyp wurden verschiedene Werte für die Matrix \mathbf{R} verwendet: Die MICROSTRAIN-Werte konnten mit $\mathbf{R} = 100 \mathbf{I}^{3 \times 3}$ ausgewertet werden, während für die Sensorenwerte des IMU-Prototyps $\mathbf{R} = 0,5 \mathbf{I}^{3 \times 3}$ verwendet werden musste. Bei größerem \mathbf{R} wurde das Filter instabil und die Performance ließ deutlich nach. Die Matrix \mathbf{Q} wurde in beiden Fällen zu $\mathbf{Q} = \text{diag}(0,0001; 0,0001; 0,0001; 10^{-9}; 10^{-9}; 10^{-9})$ gesetzt.

In Phasen mit hoher Dynamik treten bei allen drei Ergebnissen Fehler von ungefähr 5° bezüglich der Referenz auf.

9.2.2 Messflug

Die Abbildungen [9.8](#) und [9.9](#) zeigen die Auswertungen des ersten Messflugs mit dem Piper-Flugzeug. Auffällig sind die großen Fehler von etwa 10° beim Rollwinkel.

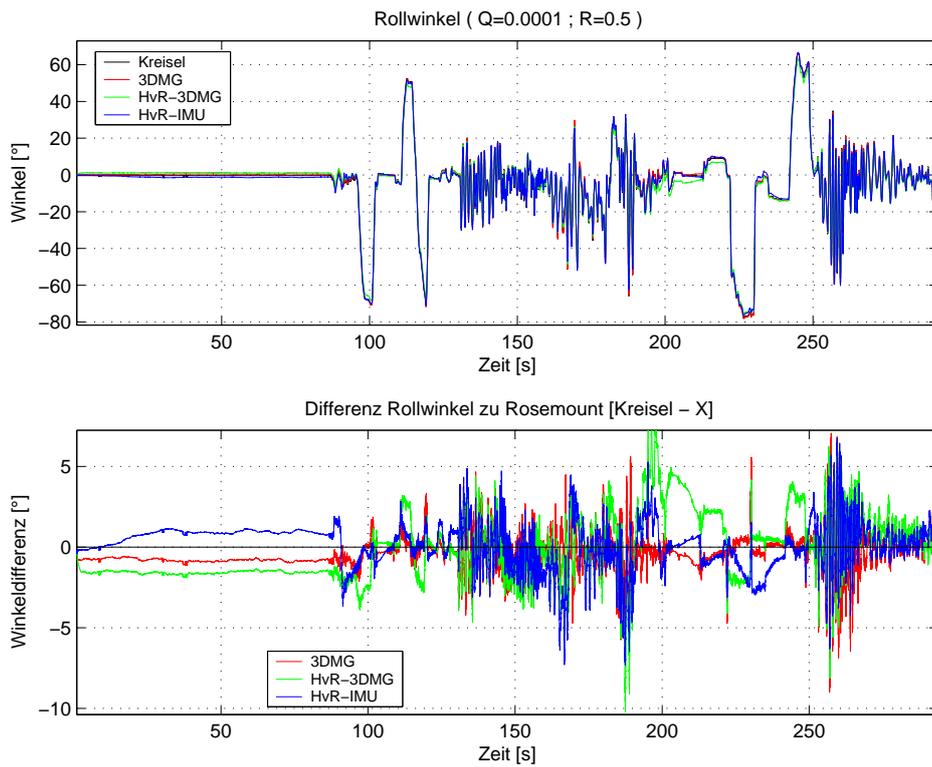


Abbildung 9.6: Vergleich der berechneten Roll-Winkel des erweiterten Systems mit Referenz und MICROSTRAIN-IMU des Bodentests

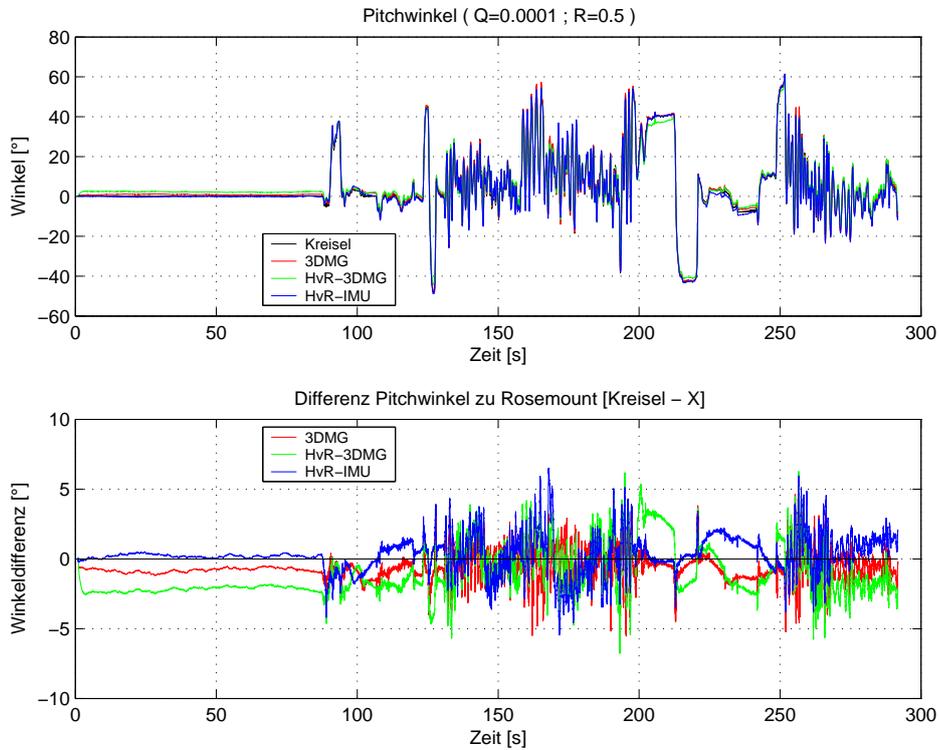


Abbildung 9.7: Vergleich der berechneten Nick-Winkel des erweiterten Systems mit Referenz und MICROSTRAIN-IMU des Bodentests

9 Ergebnisse

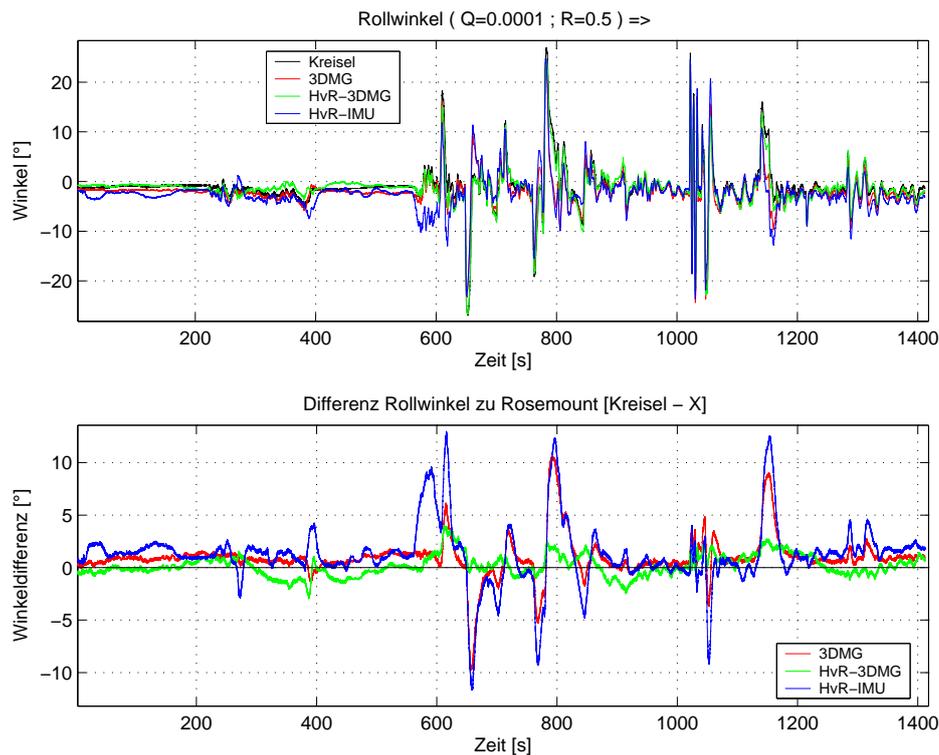


Abbildung 9.8: Vergleich der berechneten Roll-Winkel des erweiterten Systems mit Referenz und MICROSTRAIN-IMU des ersten Messflugs

Einen vergrößerten Bereich des Messflugs zeigt die [Abbildung 9.10](#), in der Phasen mit großen Rollwinkeln und großer Flugzeugdynamik gezeigt werden. Während die MICROSTRAIN-IMU zu früh (und dann zu stark) stützt, folgt der Algorithmus mit den MICROSTRAIN-Sensorwerten dem Kreisel bis in die Umkehrpunkte der Rollbewegung. Der IMU-Prototyp verhält sich mit den oben aufgeführten Einstellungen für \mathbf{Q} und \mathbf{R} ähnlich der MICROSTRAIN-IMU.

Die Nickwinkel des IMU-Prototyps in [Abbildung 9.9](#) sind sehr unruhig und weisen starke Sprünge und Fehler auf. Der aus den MICROSTRAIN-Sensorwerten berechnete Nickwinkel weist eine äquivalente Qualität wie die MICROSTRAIN-IMU auf, teilweise ist er auch erheblich besser.

9.2.3 Fahrversuch

Die Auswertung des Fahrversuchs ist in [Abbildung 9.11](#) und [9.12](#) dargestellt. Sowohl beim Rollwinkel als auch beim Nickwinkel wird das Filter beim IMU-Prototyp mit den Einstellungen für \mathbf{Q} und \mathbf{R} instabil. Das Resultat aus den MICROSTRAIN-Sensorwerten veranschaulicht, dass das Filter bei großem \mathbf{R} keine Instabilitäten aufweist. Auch bei diesem Versuch sind die aus den MICROSTRAIN-Sensorwerten berechneten Winkel äquivalent oder besser als die originalen MICROSTRAIN-Eulerwinkel.

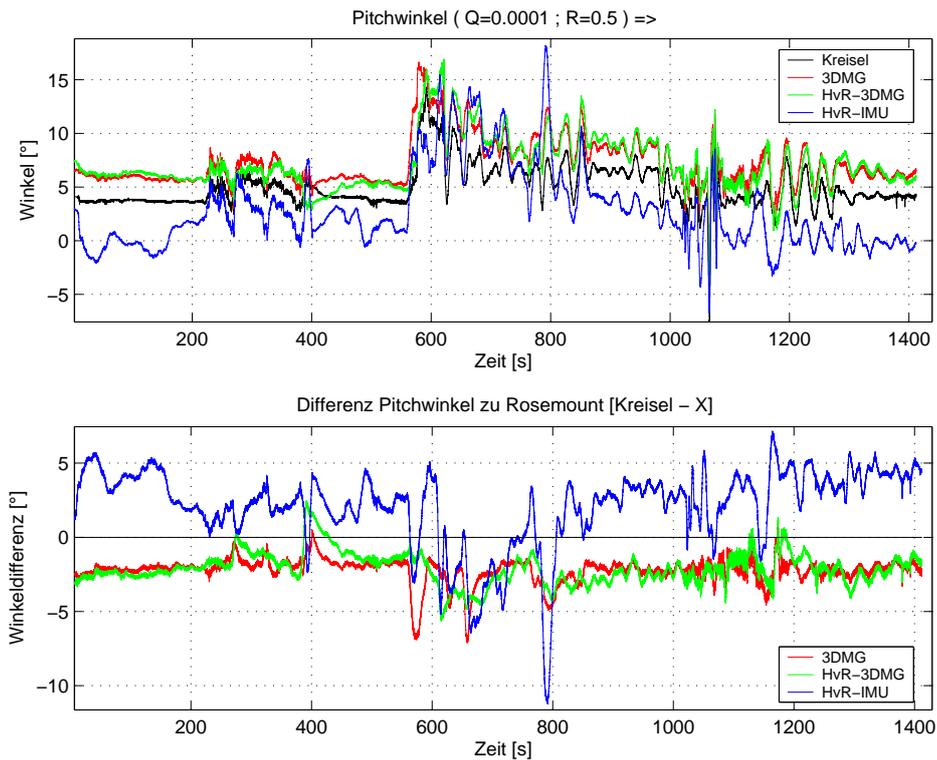


Abbildung 9.9: Vergleich der berechneten Nick-Winkel des erweiterten Systems mit Referenz und MICROSTRAIN-IMU des ersten Messflugs

9.2.4 Flugzeug-Prototyp

Die Ergebnisse des in [Abschnitt 8.2](#) vorgestellten Versuchsaufbaus mit dem Flugzeug-Prototyp sind in den Abbildungen [9.13](#) und [9.14](#) dargestellt. Da in dem Flugzeug-Prototyp der mechanische Kreisel nicht mitfliegen konnte, ist hier nur ein Vergleich zwischen den drei MEMS-Systemen ohne Referenz möglich. Die Auswertung der Messdaten, die mit dem im Fluggerät eingebauten Bordcomputer aufgezeichnet wurden, ergab, dass die MICROSTRAIN-IMU plausible Eulerwinkel liefert: Der Nickwinkel konnte über eine Vergleichsrechnung über die barometrische Höhe und der Rollwinkel über die Videoaufnahmen und GPS-Daten nachvollzogen werden.

Für die Auswertung der Flüge mit dem Flugzeug-Prototyp „Deltajet“ musste die Matrix \mathbf{Q} angepasst werden, da andernfalls das Filter instabil wurde und offensichtlich falsche Winkel liefert. Die Nickwinkel zeigen bei beiden Flügen ein sehr uneinheitliches Bild.

Beim zweiten Flug ([Abbildung 9.14](#)) wird das Filter mit den IMU-Prototyp-Daten sogar so instabil, dass es den Rollwinkel ab dem Zeitpunkt $t = 230 \text{ s}$ komplett verliert. Allerdings wird bei diesem Versuch auch das Filter bei Auswertung der MICROSTRAIN-Sensorwerte instabil. Im Zeitintervall $t = 500 - 620 \text{ s}$, in dem das Fluggerät nach der Landung am Boden stand, schwingt sowohl der Rollwinkel als auch der Nickwinkel mit bis zu 25° Amplitude.

Aufgrund dieser Beobachtungen muss die MICROSTRAIN-IMU (rote Kurve) als qualitativ bestes System für den Einsatz in dieser Konfiguration angesehen werden.

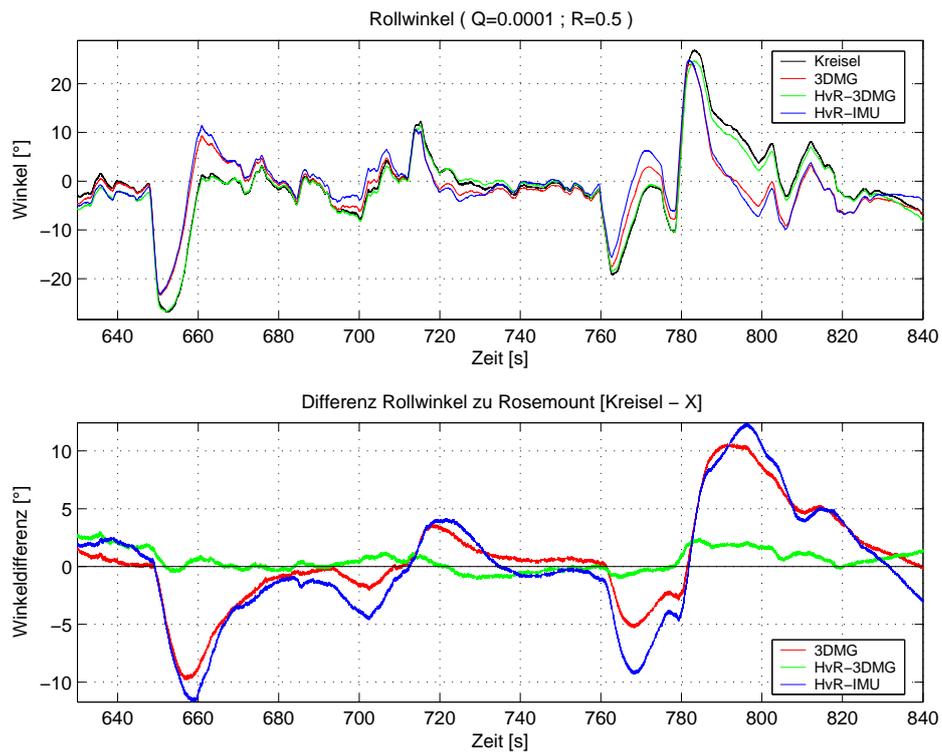


Abbildung 9.10: Bereich starken Rollens beim ersten Piper-Messflug. Deutlich ist die zu frühe und zu starke Korrektur des Signals bei *3DMG* und *HvR-IMU* zu erkennen, obwohl der Rollwinkel noch weiter ansteigt ($t=650 - 660$ s). Die grüne Kurve *HvR-3DMG* weist in diesem Intervall lediglich einen maximalen Fehler von 2° auf.

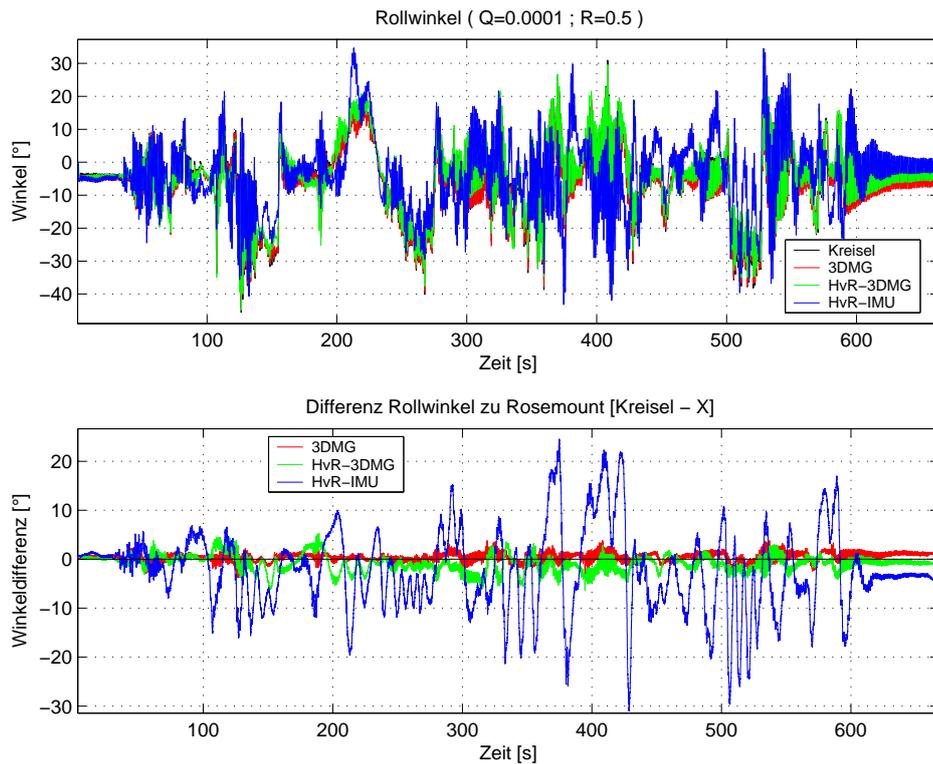


Abbildung 9.11: Vergleich der berechneten Roll-Winkel des erweiterten Systems mit der Referenz und MICROSTRAIN-IMU des Fahrversuchs.

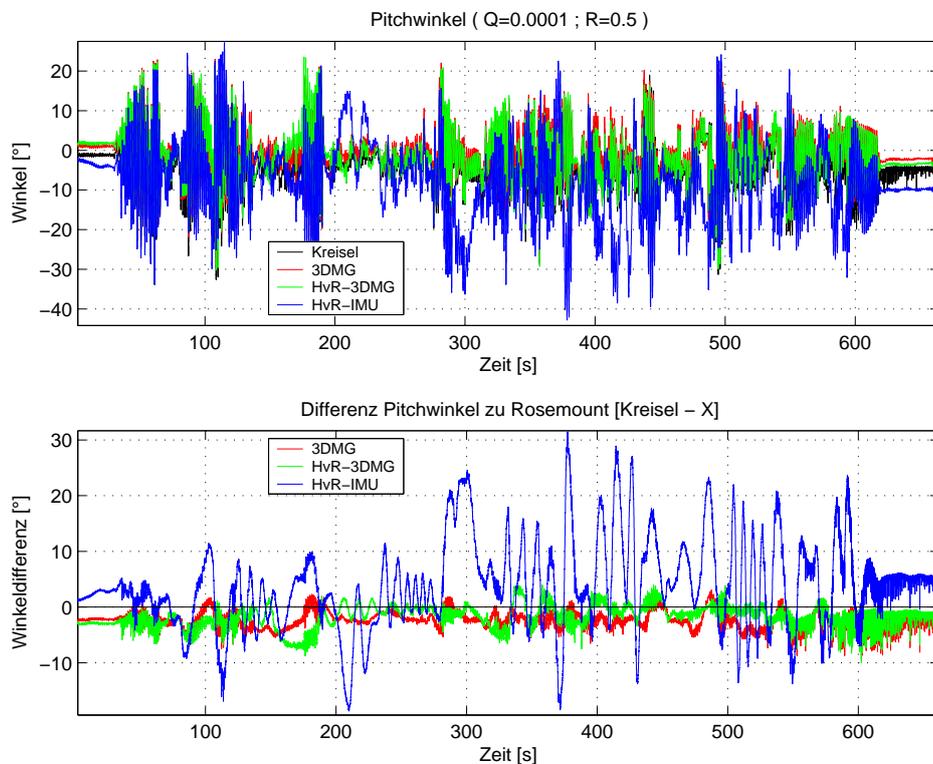


Abbildung 9.12: Vergleich der berechneten Nick-Winkel des erweiterten Systems mit der Referenz und MICROSTRAIN-IMU des Fahrversuchs.

9 Ergebnisse

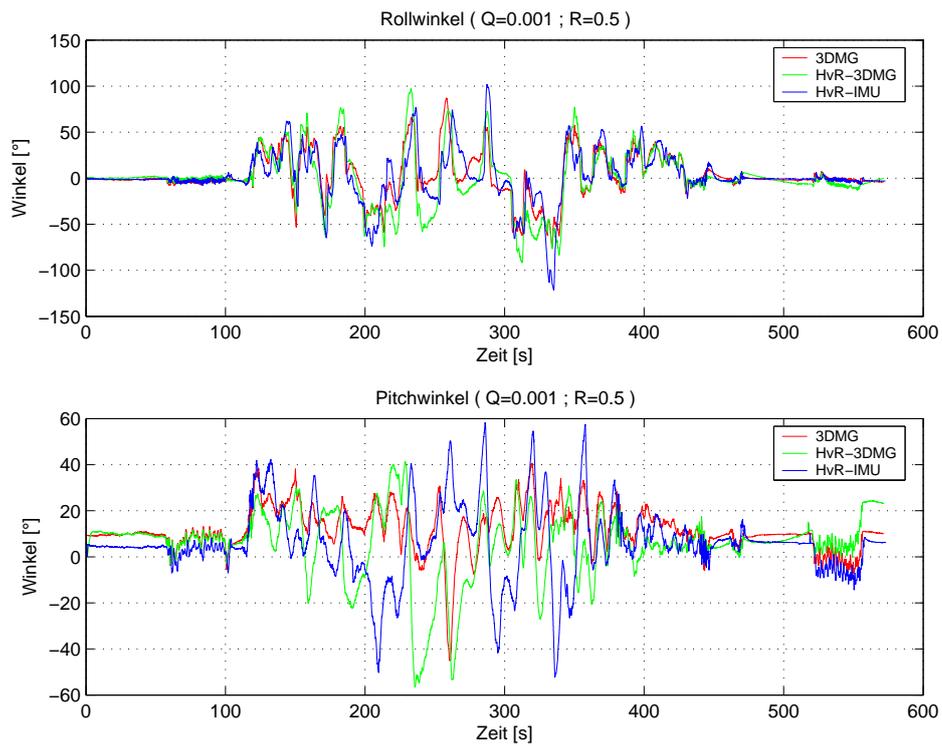


Abbildung 9.13: Auswertung des ersten Messflugs mit dem Flugzeug-Prototyp „Deltajet“.

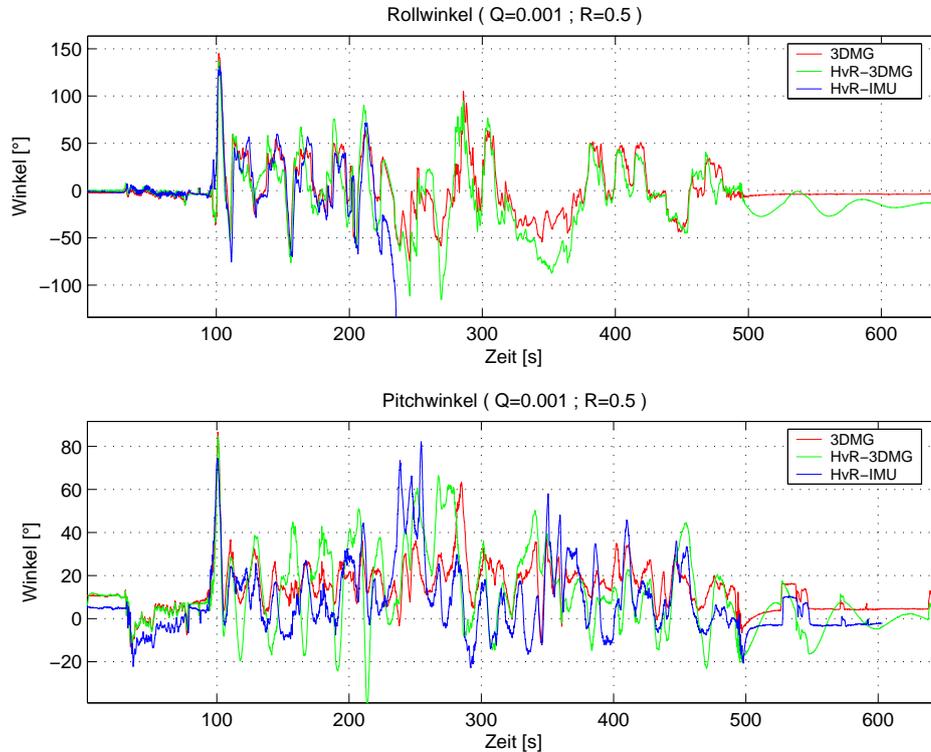


Abbildung 9.14: Auswertung des zweiten Messflugs mit dem Flugzeug-Prototyp „Deltajet“.

10 Diskussion

10.1 Hardwareaufbau

Die Qualität der Messsignale ließe sich durch eine Modifikation der Hardware verbessern. Positive Auswirkungen dürfte die Trennung der Spannungsversorgung in Digital- und Analogteil haben. Auch die räumliche Trennung der Signalfade auf einer geätzten Platine könnte zu einer besseren Signalqualität beitragen. Eine Abschirmung der IMU durch ein umschlossenes Metallgehäuse könnte Störstrahlung und Einwirkungen von der Schaltung und insbesondere vom AD-Wandler fernhalten.

10.2 Sensoren

Die Temperaturkorrektur der Sensoren führte zu einer deutlich gesteigerten Qualität der Messsignale. Während die Ergebnisse bei den Beschleunigungssensoren reproduzierbar waren, kann eine exakte Nachführung des Nullpunkts der Gyroskope nur durch den Filteralgorithmus realisiert werden, da das Temperaturverhalten der Gyroskope starken Schwankungen unterliegt. Das Verhalten der Sensoren bei Temperaturänderungen und die Nichtreproduzierbarkeit des exakten Gyroskop-Nullpunkts erschweren den seriennahen Einsatz dieser Sensoren. Insgesamt wäre eine Reduktion des Rauschens sowohl der Gyroskope als auch der Beschleunigungssensoren wünschenswert.

Es ist anzunehmen, dass die Gyroskope neben den in [Unterabschnitt 3.1.2](#) aufgezeigten Fehlerquellen noch andere Störeinflüsse aufweisen. So konnte ein Fehler, der aufgrund der auf das Gyroskop einwirkenden Winkelbeschleunigung entsteht, nicht ausgeschlossen werden.

10.3 Filteralgorithmus

Das Filter reagiert sehr empfindlich auf die Einstellungen der Matrizen \mathbf{Q} und \mathbf{R} . Außerdem ist die Qualität der Sensorwerte von entscheidender Bedeutung. Während die Sensorsignale der MICROSTRAIN-IMU schon einer internen Nullpunktkorrektur unterzogen wurden, muss das Filter beim IMU-Prototyp die Nullpunkte ohne vorherige Maßnahmen korrigieren. Daraus resultieren auch die unterschiedlichen Werte für \mathbf{R} (100 bei MICROSTRAIN zu 0,5 beim Prototyp).

Die Einstellung des KALMAN-Filters musste zwischen den Versuchen geändert werden. Dies ist insbesondere im Hinblick auf die Ergebnisse von [9.2.2](#) und [9.2.4](#) problematisch: Es konnte kein „Flug-Setup“ gefunden werden, das sowohl bei manntragenden als auch bei unbemannten Fluggeräten zuverlässig funktioniert. Die Ergebnisse des Messflugs mit dem Piper-Flugzeug konnten nicht auf den Flugzeug-Prototypen übertragen werden.

Ein Lösungsansatz wäre die Erweiterung des KALMAN-Filters um weitere adaptive Elemente, die die Stützung nicht nur an- und abschalten (siehe 5.5), sondern die Stärke der Stützung dynamisch anpassen. Ebenfalls denkbar wäre eine dynamische Anpassung von \mathbf{Q} und \mathbf{R} . Dies bedingt jedoch die Implementierung einer Erkennung der Flugdynamik. Ein sinnvoller Ansatz wäre die Einführung eines „Dynamikfaktors“, der sich aus der Norm der Drehratensignale und der Norm der Beschleunigungssensoren ableitet. Diese Größe könnte als veränderlicher Verstärkungsfaktor für \mathbf{Q} und \mathbf{R} dienen. Im eher stationären Flug, wie er beim Piper-Flugzeug herrschte, könnte \mathbf{Q} verkleinert werden, während die Matrix in Phasen hoher Dynamik vergrößert wird.

10.4 Ausblick

Für die Weiterentwicklung der IMU zu einem serienreifen Produkt bieten sich einige Erweiterungen an:

- Die Umstellung des Filteralgorithmus auf Quaternionen, damit die IMU in allen Lagen immer funktioniert und die Singularität der Euler-Winkel vermieden wird.
- Der Gierwinkel ψ könnte auch noch berücksichtigt werden. Eine Stützung wäre über eine Einkoppelung der Kurs-Signale eines GPS-Geräts oder eines Kompasses denkbar.
- Das GPS-Signal kann im KALMAN-Filter als Stützgröße berücksichtigt werden. Aus GPS-Geschwindigkeitsvektor und aus den aktuell gemessenen Drehraten kann der Radius der geflogenen Kreisbahn bestimmt werden. Damit kann dann präzise die Qualität der Beschleunigungsmesser beurteilt werden bzw. die Komponenten, die nicht zur Lageerkennung dienen (Corioliskraft, Zentripetalbeschleunigung), herausgerechnet werden. Damit erhält man dann einen unverfälschten Lotvektor. Aus der Kombination von Geschwindigkeit und projizierter Drehrate kann ein Anhalt für die Zentripetalkraft gefunden werden. Mit dem Lotvektor (bzw. den projizierten Beschleunigungen) kann damit ein Stützwert für den Rollwinkel ermittelt werden.

Listings

Listings

Literaturverzeichnis

- [Allgöwer 2004] ALLGÖWER, Frank:
Regelungstechnik II – Script zur Vorlesung.
Universität Stuttgart: Institute for Systems Theory in Engineering (IST) (Veranst.), 2004. –
URL <http://www.ist.uni-stuttgart.de>
- [Analog Devices Inc. a] ANALOG DEVICES INC.. –
URL <http://www.analog.com>. –
Hersteller der Beschleunigungsmesser und der Gyroskope
- [Analog Devices Inc. b] ANALOG DEVICES INC.:
ADXL202E – technical data sheet. –
URL http://www.analog.com/UploadedFiles/Data_Sheets/53728567227477ADXL202E_a.pdf. –
Datenblatt zum Beschleunigungssensor ADXL202E
- [Analog Devices Inc. c] ANALOG DEVICES INC.:
ADXRS300 – technical data sheet. –
URL http://www.analog.com/UploadedFiles/Data_Sheets/732884779ADXRS300_b.pdf. –
Datenblatt zum Gyroskop ADXRS300
- [Atmel] ATMEL. –
URL <http://www.atmel.com>. –
Hersteller des ATmega32-Mikrocontrollers
- [AUTOFLUG Steuerungs- und SensorTechnik GmbH] AUTOFLUG STEUERUNGS- UND SENSORTECHNIK
GMBH. –
URL <http://www.autoflug.de>. –
Neue Strasse 95, D-73230 Kirchheim unter Teck-Nabern
- [Geen und Krakauer 2003] GEEN, John und KRAKAUER, David:
New iMEMS Angular-Rate-Sensing Gyroscope.
In: *Analog Dialogue*
37 (2003)
- [Geitner] GEITNER, Hubert:
Considerations for Soldering Accelerometers in LLC-8 Packages onto printed circuit boards.
Application Note.
One Technology Way , Norwood, MA (USA): Analog Devices (Veranst.). –
AN-652
- [Linear Technology Inc. a] LINEAR TECHNOLOGY INC.. –
URL <http://www.linear.com>. –
Hersteller des AD-Wandlers
- [Linear Technology Inc. b] LINEAR TECHNOLOGY INC.:
LTC1867A – technical data sheet. –
Datenblatt zum AD-Wandler LTC1867A
- [Microstrain Inc.] MICROSTRAIN INC.. –

Literaturverzeichnis

- URL <http://www.microstrain.com>. –
Hersteller der 3DM-GX1 IMU – 310 Hurricane Lane, Suite 4 ,Williston, USA
- [Ponyprog] PONYPROG:
Programm zum Flashen des Mikrocontrollers. –
URL <http://www.lancos.com/prog.html>
- [RMS – Regelungs- und Messtechnik] RMS – REGELUNGS- UND MESSTECHNIK. –
URL <http://www.rms-testsystems.de>. –
Gutenbergstr. 27 , 21465 Reinbek / Hamburg
- [Rosemount Aerospace Inc.] ROSEMOUNT AEROSPACE INC.. –
Hersteller des mechanischen Kreisels – Burnsville, Minnesota USA
- [Steinbeis Transferzentrum Aerodynamik, Flugzeug- und Leichtbau] STEINBEIS TRANSFERZENTRUM AERODYNAMIK, FLUGZEUG- UND LEICHTBAU. –
URL <http://www.stzafl.de>. –
Baumwiesenweg 6b, 70569 Stuttgart
- [Stovall 1997] STOVALL, Sherry H.:
Basic Inertial Navigation.
China Lake, California: Naval Air Warfare Center Weapons Division (Veranst.), September 1997
- [Wagner 2003] WAGNER, J.:
Zur Verallgemeinerung integrierter Navigationssysteme auf räumlich verteilte Sensoren und flexible Fahrzeugstrukturen.
Düsseldorf : VDI, 2003
- [Weinberg a] WEINBERG, Harvey:
Phase and Frequency Response of iMEMS Accelerometers and Gyros.
Application Note.
One Technology Way , Norwood, MA (USA): Analog Devices (Veranst.). –
AN-688
- [Weinberg b] WEINBERG, Harvey:
Temperature Compensation Techniques for Low g iMEMS Accelerometers.
Application Note.
One Technology Way , Norwood, MA (USA): Analog Devices (Veranst.). –
AN-598
- [Weinberg c] WEINBERG, Harvey:
Using the ADXL202 Duty Cycle Output.
Technical Note.
One Technology Way , Norwood, MA (USA): Analog Devices (Veranst.)
- [Well 2001] WELL, Klaus H.:
Flugmechanik – Manuskript zur Vorlesung.
Universität Stuttgart: Institut für Flugmechanik und Flugregelung (IFR) (Veranst.), 2001. –
URL <http://www.uni-stuttgart.de/ifr>
- [WinAVR] WINAVR:
Entwicklungsumgebung für ATmega-Mikrocontroller in Programmiersprache C. –
URL <http://winavr.sourceforge.net/>
- [Zeititz] ZEITZ, M.:
Verfahren zur numerischen Integration - Hilfsblatt zur Vorlesung 'Simulationstechnik'.
Universität Stuttgart: Institut für Systemdynamik (ISYS) (Veranst.). –
URL <http://www.isys.uni-stuttgart.de>