



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Dimitrios Papadopoulos

Untersuchung von low-cost Micro-Electro-Mechanical-Systems
zur Lagebestimmung

Dimitrios Papadopoulos

Untersuchung von low-cost Micro-Electro-
Mechanical-Systems zur Lagebestimmung

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Hans-Jürgen Hotop
Zweitgutachter : Prof. Dr.-Ing. Karl-Ragmar Riemschneider

Abgegeben am 20. Januar 2016

Dimitrios Papadopoulos

Thema der Bachelorthesis

Untersuchung von low-cost Micro-Electro-Mechanical-Systems zur
stimmung

Stichworte

MEMS, Kalman, INS, Navigation, Lagebestimmung, Gyroskop,
gungsmesser, Arduino, MPU6050

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Zusammenstellung und Aufbau einer
umgebung zur Untersuchung der Lagebestimmung von low-cost micro-elektro-
mechanischen Systemen mit Inertialsensoren.

Dimitrios Papadopoulos

Title of the paper

Investigation of low-cost micro-electro-mechanical systems for orientation

Keywords

MEMS, Kalman, INS, Navigation, Orientation, Determination, Gyroscope, Ac-
celerometer, Arduino, MPU6050

Abstract

This paper deals with the composition and implementation of a test environ-
ment to examine the determination of the orientation of low-cost micro-electro-
mechanical systems with inertial sensors.

Inhaltsverzeichnis

1	Einführung.....	1
2	Grundlagen.....	2
2.1	Inertialnavigation.....	2
2.1.1	Stable Platform System.....	2
2.1.2	Strapdown System.....	3
2.2	Bezugssysteme.....	3
2.3	Anwendungsbereiche.....	4
2.4	Navigationsgleichungen.....	5
2.5	Sensoren.....	6
2.5.1	Gyroskop.....	6
2.5.2	Beschleunigungsmesser.....	11
3	Theoretischer Hintergrund.....	13
3.1	Rotation.....	13
3.1.1	Rotationsmatrix.....	13
3.1.2	Quaternionen.....	15
3.1.3	Vergleich von Quaternionen und Rotationsmatrizen.....	16
4	Filter.....	17
4.1	Kalman Filter.....	18
4.1.1	Einführung.....	18
4.1.2	Varianz und Mittelwert.....	20
4.1.3	Prozessschritte ‚Prediction‘ und ‚Update‘.....	22
4.2	Mehrdimensionaler Kalman Filter.....	27
4.2.1	Kalman-Filter Gleichungen.....	28
4.3	Extended Kalman Filter.....	30
5	Testumgebung und Umsetzung.....	31

5.1	Hardware	31
5.1.1	MEMS-Chip	31
5.1.2	Mikroprozessor.....	32
5.1.3	Bluetooth	33
5.1.4	Befestigung des Chips	34
5.2	Software.....	35
5.2.1	Arduino IDE, Visual Studio	35
5.2.2	Empfang, Verarbeitung und Speicherung	36
5.3	Diagramm	38
5.4	Umsetzung	38
5.4.1	Algorithmus und Kalman Filter	38
6	Tests	40
6.1	Offsets.....	41
6.1.1	Offset Gyroskope	41
6.1.2	Offset Beschleunigungsmesser	44
6.2	Lagebestimmung	47
7	Fazit und Ausblick	50
8	Quellen	51
	Anhang	54

Abbildungsverzeichnis

Abbildung 1 - Plattform System.....	3
Abbildung 2 - Fehlerfortpflanzung bei Bestimmung von Geschwindigkeit und Position.....	6
Abbildung 3 - kardanisch aufgehängtes Gyroskop	7
Abbildung 4 - Drehratensensor in traditioneller Bauweise	8
Abbildung 5 - Aufbau eines Lasergyroskops	9
Abbildung 6 - vibrierendes Piezoelektrikum in Vibrationsgyroskop.....	10
Abbildung 7 - rotierendes Vibrationsgyroskop	10
Abbildung 8 - Funktionsprinzip eines Beschleunigungsmessers	11
Abbildung 9 - kapazitiver Beschleunigungssensor durch ein Rasterelektronenmikroskop	12
Abbildung 10 - Piezoelektrischer Beschleunigungsmesser	13
Abbildung 11 - Rotation im zweidimensionalen Raum	14
Abbildung 12 - Quaternionen und Rotation	16
Abbildung 13 - Gimbal Lock	16
Abbildung 14 - Kalman Filter, Ablauf allgemein	18
Abbildung 15 - Verrauschte Werte vom Gyroskop; Varianz Rauschen $r=0.02$	19
Abbildung 16 - Integration der verrauschten Sensordaten.....	19
Abbildung 17 - Verteilungsdichtefunktion mit Mittelwert und Varianzen.....	20
Abbildung 18 - Normalverteilung DEG nach Initialisierung.....	21
Abbildung 19 - Ablauf 1D - Kalman-Filter	22
Abbildung 20 - Normalverteilung nach dem Prediction-Schritt	23
Abbildung 21 - Darstellung der Normalverteilungen während einer Iteration	24
Abbildung 22 - 1D-KF, Sensordaten mit Rauschvarianz = 0.1, Systemvarianz = 0.1.....	25
Abbildung 23 - 1D-KF, Sensordaten mit Rauschvarianz = 0.4, Systemvarianz = 0.1, Messvarianz = 5	25
Abbildung 24 - 1D-KF, Sensordaten mit Rauschvarianz = 0.4, Systemvarianz = 0.01, Messvarianz = 5	26
Abbildung 25 - 1D-KF mit Rauschvarianz = 0.4, Systemvarianz = $1e-5$, Messvarianz = 1000	27
Abbildung 26 - MPU6050 auf einem GY-521 breakout Board.....	32
Abbildung 27 - Arduino Due	33
Abbildung 28 - SparkFun Bluetooth Modem - BlueSMiRF Gold und CSL-USB nano Bluetooth Adapter	34
Abbildung 29 - Befestigung des MPU6050	35

Abbildung 30 - ZedGraph	37
Abbildung 31 - Diagramm, Aufbau	38
Abbildung 32 - Algorithmus und Kalman Filter	38
Abbildung 33 - Bedienelemente für Filterparameter	40
Abbildung 34 - Messung des Offsets vom Gyroskop, während Abkühlung.....	42
Abbildung 35 - Gyroskop-Offset über Temperatur.....	43
Abbildung 36 - Kamerawasserwaage zum Leveln des MPU6050.....	44
Abbildung 37 - genauere Offsetkorrektur	45
Abbildung 38 - Messung des Offsets vom Beschleunigungsmesser, während Abkühlung.....	46
Abbildung 39 - Beschleunigungsmesser-Offset über Temperatur	46
Abbildung 40 - Roll 30°, fest	47
Abbildung 41 - Roll 45° fest	49

Tabellenverzeichnis

Tabelle 1 - Übersicht von Standard und Extended Kalman Filter	30
Tabelle 2 - MEMS-Chips Datenvergleich.....	32
Tabelle 3 - Ergebnisse, Rollwinkel 30°, fest.....	48
Tabelle 4 - Ergebnisse , Rollwinkel 30°, fest, neukalibriert	48
Tabelle 5 - Ergebnisse , Rollwinkel 45°, fest.....	49

1 Einführung

Der Mensch besitzt allgemeinbekannt fünf Sinne: Sehen, Hören, Tasten, Schmecken und Riechen. Allerdings reichen diese nicht aus um die Umwelt ausreichend zu erfassen. Weitere Sinne sind nötig: Temperatursinn, Gleichgewichtssinn, Körperposition, Beschleunigungen und mehr. Die Informationen dieser Sinne werden im Gehirn verarbeitet und geben dem Menschen ein Gesamtbild seines aktuellen Daseins in der Umwelt.

Dies passiert durchgehend, ohne dass der Mensch aktiv diese Verarbeitung anregen muss, auch wenn es natürlich Ausnahmen gibt, wie z.B. beim Sport oder in Notsituationen.

Werden die genannten Termini in ein technisches Umwelt übertragen, dann können die Sinne als Sensoren angesehen werden und das Gehirn als eine Recheneinheit. Mit dem heutigen Stand der Technik sind fast alle oben erwähnten Sinne gewissermaßen als einzelne Sensoren oder einer Zusammensetzung dieser vorhanden: Sehen/Kamera, Hören/Mikrofon, Tasten/Drucksensor. Selbst an Sensoren für Geschmack und Geruch wird geforscht. Dabei handelt es sich allerdings um Interpretationen des Gehirns, weswegen, z.B. beim Versuch Gerüche zu erfassen, Gassensoren zum Einsatz kommen, die Gaskonzentrationen messen.¹

Diese Thesis befasst sich mit der Berechnung einer Orientierung, also der Lage eines Objekts in einem definierten Umfeld, z.B. die Erde. Während beim Menschen die Bestimmung seiner Lage mehr oder weniger ein Automatismus ist, ist es im technischen Bereich eine eher nichttriviale Problemstellung. Zur Verfügung steht eine begrenzte Art von Sensoren, die verrauschte Signale ausgeben. Auch müssen diese Signale zunächst korrigiert werden. So ist es z.B. zu beachten ob die Gravitation gebraucht wird (Orientierung) oder heraus gerechnet werden muss (Navigation). Eine Navigation kann wiederum nicht ohne bekannte Orientierung stattfinden.

In dieser Thesis werden Gyroskope und Beschleunigungsmesser aus dem low-cost-Segment benutzt. Ein Gyroskop misst die Winkel- bzw. Drehgeschwindigkeit um eine Achse. Der Beschleunigungsmesser misst die Beschleunigung, die auf eine Achse wirkt. Beschreibung und Funktionsweise der Sensoren wird in Kapitel 2.5 behandelt.

¹ [5]

In der inertialen Navigation werden Orientierung und Position eines Objektes mithilfe der Beschleunigung und Drehrate von diesem, ermittelt. Dafür muss die Anfangsposition, -geschwindigkeit, -beschleunigung und -orientierung bekannt sein. Das System, mit den nötigen Sensoren und Recheneinheit, heißt *inertiales Navigationssystem* (Kapitel 2.1).

2 Grundlagen

2.1 Inertialnavigation

In der *inertialen Navigation*² werden Orientierung und Position eines Objektes mithilfe der Beschleunigung und Drehrate von diesem, ermittelt. Dafür muss die Anfangsposition, -geschwindigkeit, -beschleunigung und -orientierung bekannt sein. Das System, mit den nötigen Sensoren und Recheneinheit, heißt *inertiales Navigationssystem* (INS).

Ein inertiales Navigationssystem misst mithilfe von Gyroskopen und Beschleunigungsmessern Drehrate und Beschleunigung eines Objekts. Ausgehend von den Anfangsbedingungen ist es durch Integration der Sensordaten möglich die Position und Orientierung des Objektes zu bestimmen.

Die Navigation im dreidimensionalen Raum erfordert, dass das inertielle Navigationssystem sechs Freiheitsgrade (DOF: Degree Of Freedom) besitzt. Ein DOF entspricht dabei einer möglichen Bewegung im Raum: ein Objekt kann um seine drei Achsen rotieren → Gieren, Nicken, Rollen (Yaw, Pitch, Roll) und im Raum bewegt werden (Translation).

Es ist möglich inertielle Navigationssysteme in zwei Gruppen zusammenzufassen: *Stable Platform System* und *Strapdown System*.

2.1.1 Stable Platform System

Die Ursprungsform der Inertialnavigation ist das *Stable Platform System*. Auf einer kardinalisch aufgehängten Plattform (Abbildung 1), werden drei orthogonal angeordnete Beschleunigungsmesser so ausgerichtet, dass sie, zum Beispiel, nach Norden, Osten und Richtung Erdmittelpunkt zeigen. Um zu verhindern, dass die Plattform bei einer Rotation des zu beobachteten Objekts mit rotiert, wird jede Achse durch ein Gyroskop stabilisiert.

² [6] S. 5

Da die Lage der inneren Plattform immer gleich bleibt, kann die Orientierung des Objekts an der kardanische Aufhängung abgelesen werden. Die Daten des Beschleunigungsmessers werden zweifach integriert und geben somit die Position des Objekts wieder.

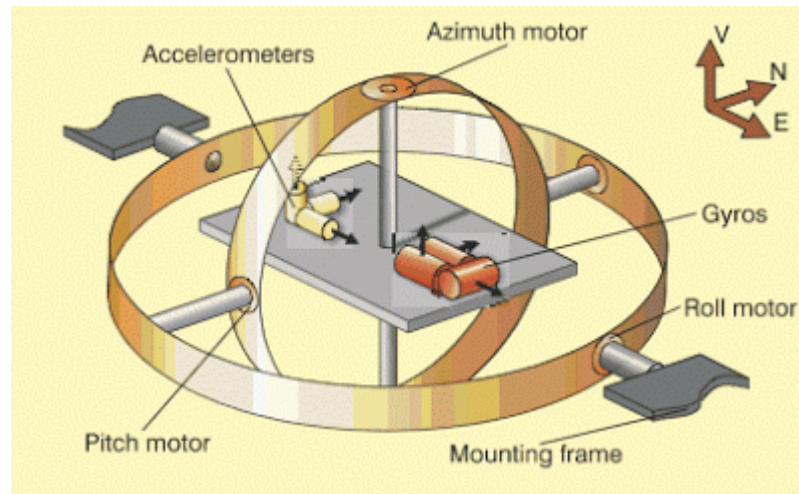


Abbildung 1 - Plattform System³

2.1.2 Strapdown System

Beim *Strapdown System* werden die Beschleunigungsmesser und Gyroskope, direkt am Objektkörper befestigt und bewegen sich somit mit. Die Messdaten sind Körperbezogen und müssen zunächst vom body-frame ins navigation-frame (Kapitel 2.1.1) umgerechnet werden, bevor eine Positionsbestimmung möglich ist.

2.2 Bezugssysteme

Um Navigieren bzw. eine Orientierung zu bestimmen, müssen Koordinatensysteme bestimmt werden, die als Bezugssysteme dienen.

Anmerkung zur Notation: Die Achsen der Koordinatensysteme werden mit x,y und z benannt. Die Zugehörigkeit der Achse zu einem System, wird mit einem hochstehenden Kürzel (z^e) gekennzeichnet. Außerdem gilt $z^e \triangleq z^e$ -Achse.

- ***inertial-frame***

Das *inertial-frame* (*i-frame*) hat seinen Koordinatenursprung im Mittelpunkt der Erde. Die Ausrichtung ist anhand der Fixsterne definiert und rotiert nicht mit der Erde. Dabei

³ [8]

entspricht z^i der Polarachse der Erde. Im inertial-frame gilt das erste Newton'sche Gesetz: solange keine Kraft auf einen Körper einwirkt, behält dieser seine Geschwindigkeit und Richtung bei.

- ***earth-frame***

Das *earth-frame* (*e-frame*) hat ebenfalls seinen Koordinatenursprung im Erdmittelpunkt mit $z^e = z^i$. Die x^e -Achse zeigt stets zum Nullmeridian und rotiert somit mit der Erde.

- ***navigation-frame***

Das *navigation-frame* (*n-frame*) hat seinen Ursprung im Mittelpunkt des Messsystems. Er bewegt sich gemeinsam mit dem INS. Die x^n -Achse zeigt nach Norden und y^n nach Osten.

- ***body-frame***

Das *body-frame* (*b-frame*) ist an den roll-, pitch- und yaw-Achsen des Objekts ausgerichtet. Der Ursprung kann beliebig im Objekt gewählt werden. Oft wird der Schwerpunkt des Objekts oder das INS selbst gewählt.

2.3 Anwendungsbereiche

Inertiale Navigation wird so gut wie in allen Fortbewegungsmitteln genutzt. Ihr großer Vorteil ist, dass sie vollkommen autonom arbeiten, d.h. keine Verbindung zu externen Hilfsmitteln, wie z.B. einem Satelliten, brauchen. Im Kraftfahrzeug unterstützt es das Navigationsgerät (also das GPS-Signal). Als klassischer Fall gilt die Einfahrt in einen Tunnel. Das Navigationsgerät funktioniert trotz Abbruch des GPS-Signals, da Anfangsposition, -orientierung und -beschleunigung zum Zeitpunkt des Signalabbruchs bekannt waren und während der Tunnelfahrt mithilfe der inertialen Navigation aktualisiert wird.

In der Luftfahrt wird INS ebenfalls zur Unterstützung des GPS-System genutzt. Da hier höhere Genauigkeit gefordert wird als beim Automobil, muss das INS zunächst ausgerichtet werden, was etwa 20-30 Minuten dauert. Während der Ausrichtung muss das Luftfahrzeug komplett unbewegt bleiben. Das INS kann somit die Erddrehung erfassen und daraus

den geographischen Nordpol bestimmen. Die Ausgangsposition wird dabei zusätzlich als Koordinatenangabe (z.B. über das GPS) eingegeben.

Um die verrauschten und fehlerbehafteten Sensordaten richtig zu verarbeiten, müssen Filter und entsprechende Algorithmen zur Ermittlung der Lage in einen Prozessor implementiert werden. Als Filter werden Kalman-Filter verwendet, welche das Signalrauschen vermindern sollen (Kapitel 4). Die sogenannten Navigationsgleichungen beinhalten die notwendigen Gleichungen zur Bestimmung der Orientierung eines Objekts (Kapitel 2.4).

2.4 Navigationsgleichungen

In der Inertialnavigation werden die sogenannten *Navigationsgleichungen* eingesetzt. Sie werden in inertialen Navigationssystemen eingesetzt um Lage, Geschwindigkeit, Position in der Ebene und Höhe zu bestimmen. Parameter sind Drehrate und Beschleunigung des Objekts.

$$p = \int v dt = \iint a dt dt \quad (2-1)$$

Die Integration der Beschleunigung führt zur Geschwindigkeit. Integration der Geschwindigkeit führt zur Position. Die doppelte Integration fordert eine hohe Genauigkeit der Messwerte. Wie in Abbildung 2 zu erkennen ist, kann ein kleiner Messfehler von $0,1m/s^2$ nach 10s zu einem Positionsfehler von etwa $5,5m$ führen. Diese Simulation bezieht sich nur auf den zweidimensionalen Raum. Ein Messfehler beim Beschleunigungsmesser, führt im dreidimensionalen Raum zu einem Fehler in der Lagebestimmung, was die Positionsbestimmung zusätzlich verfälscht.

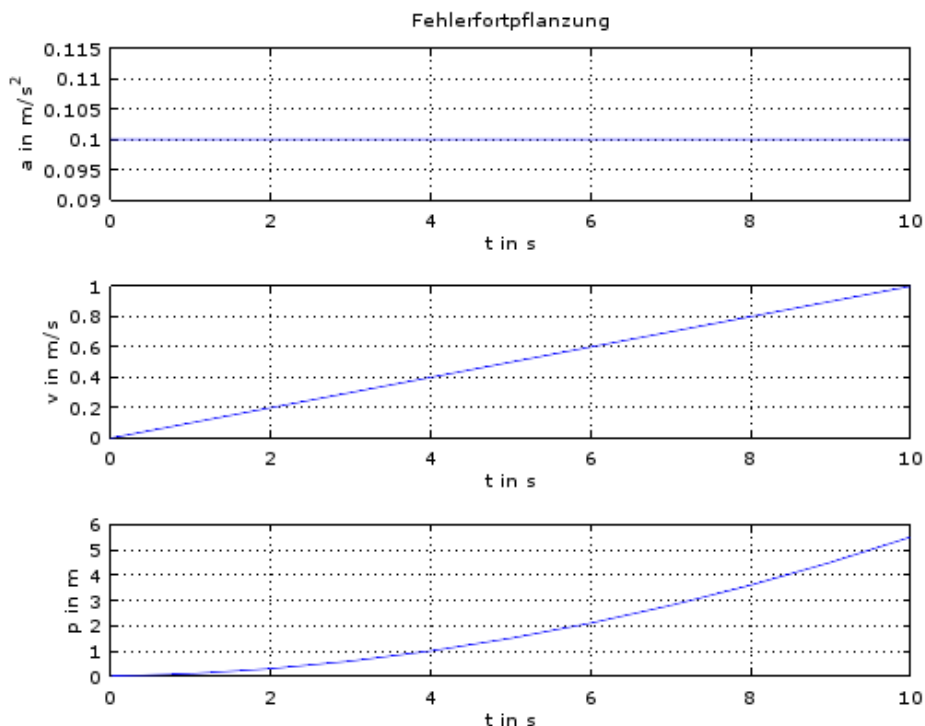


Abbildung 2 - Fehlerfortpflanzung bei Bestimmung von Geschwindigkeit und Position

Über der Integration der Drehrate, kann die Lage bestimmt werden. Ohne Lagebestimmung ist eine Navigation nicht möglich. Sie wird benötigt um Beschleunigungswerte vom körperfesten System (b-frame) ins Navigationssystem (n-frame) zu berechnen.

Dafür wird die DCM (Direct Cosine Matrix) aufgestellt (Anhang).

2.5 Sensoren

Inertiale Navigationssysteme verfügen grundsätzlich über Beschleunigungsmesser und Gyroskope. Manchmal werden diese noch durch Magnetometer, welche, anhand der Erdmagnetfelder, einen zusätzlichen Bezugspunkt zur Bestimmung des Kurses geben können, unterstützt.

2.5.1 Gyroskop

Ein *Gyroskop* (griech. γύρος ‚Drehung‘, altgriech. σκοπεῖν ‚betrachten‘), auch Kreiselinstrument, ist ein Kreisel, welcher sich in einem bewegenden Lager (z.B. kardanischer Rahmen) dreht⁴. Ein Gyroskop ist bestrebt seine Lage beizubehalten (Präzession). Eine

⁴ [10]

Krafteinwirkung auf die Rotationsachse führt dazu, dass das Gyroskop senkrecht dieser ausweicht. Dies ist auf die Drehimpulserhaltung zurückzuführen auf welche in dieser Arbeit nicht weiter eingegangen wird.⁵

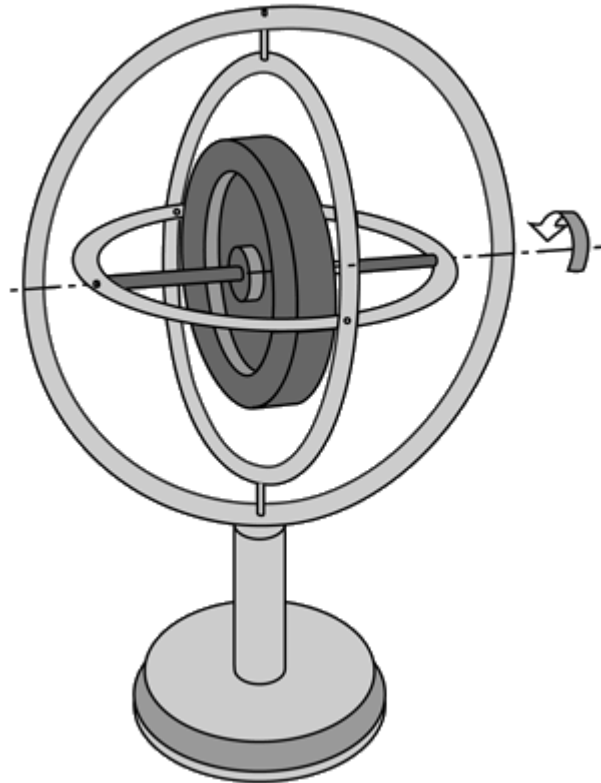


Abbildung 3 - kardanisch aufgehängtes Gyroskop⁶

2.5.1.1 Gyroskop als Sensor

Ein Gyroskop wird oft als Bestandteil von Lage- und Drehratesensoren genutzt. Beim Lagesensor wird die Fähigkeit zur Beibehaltung der Ursprungslage genutzt.

Wenn das Gyroskop kardanisch an einem Objekt aufgehängt ist, kann am kardanischen Rahmen die Lage des Objekts abgelesen werden (*Lagesensor*). Dieses Prinzip kommt im Stable Plattform System (Kapitel 2.1.1) vor, bei dem je ein Gyroskop für eine Achse verbaut ist.

Beim *Drehratesensor* (auch Gyrometer) wird das Gyroskop so an ein Objekt befestigt, dass es senkrecht zur bezogenen Achse ist und nur noch zwei Freiheitsgrade hat (Abbildung 4)

⁵ [12]

⁶ nach [11]

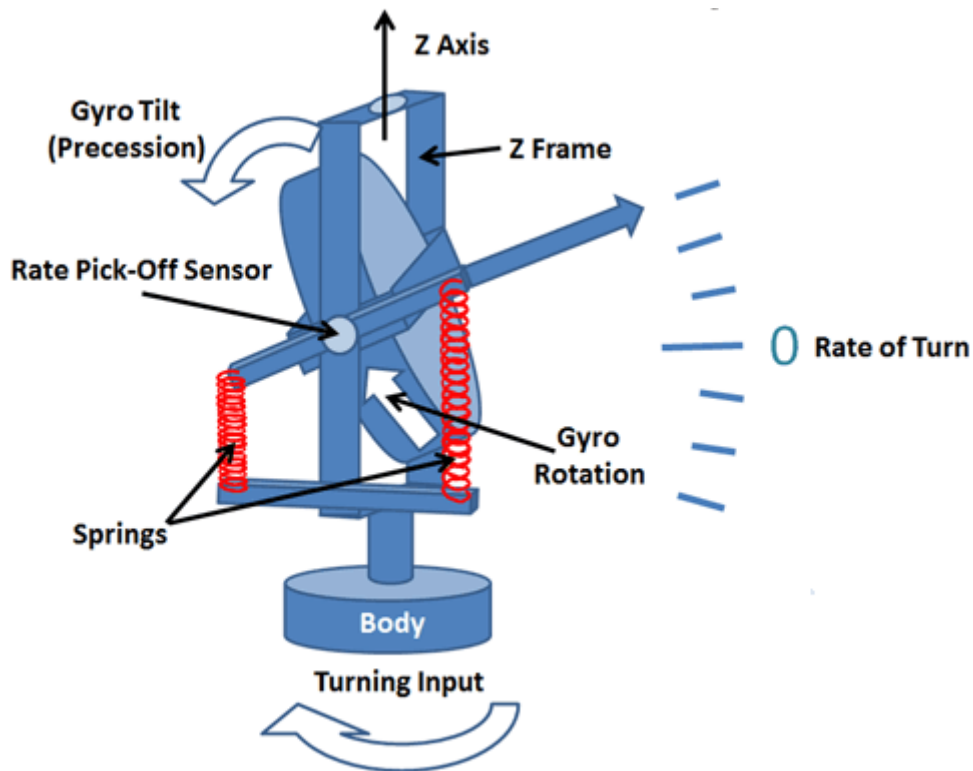


Abbildung 4 - Drehratensensor in traditioneller Bauweise⁷

Eine Drehung um die Achse (in Abbildung 4 die z-Achse) führt dazu, dass sich das Gyroskop neigt. Je schneller die Drehung, desto höher das Präzessionsverhalten des Gyroskops und desto mehr neigt es sich senkrecht zur Drehrichtung.

2.5.1.2 Laser- und Vibrationsgyroskope

Gyroskope können in drei Kategorien eingeteilt werden: mechanisch (wie oben beschrieben), optisch und vibrierend. Neben vielen anderen Punkten unterscheiden sie sich hauptsächlich in

- Bauart
- Volumen
- Kosten
- Genauigkeit
- Stabilität.

Das *Lasergyroskop*⁸ gibt mit einer sehr hohen Genauigkeit die Drehrate an. Zwei gegenläufige Laserstrahlen werden in eine geschlossene, dreiecksförmige Bahn geschickt

⁷ [12]

(Abbildung 5). Die Ablenkung der Strahlen erfolgt durch Spiegel. Es ist auch möglich die Strahlen ringförmig durch Glasfaserleitungen zu senden. Wenn sich das Lasergyroskop in Ruhelage befindet, dann haben beide Laserstrahlen die gleiche Frequenz. Bei einer Rotation des Sensors verkürzt sich für einen Laserstrahl bzw. verlängert sich die zurückzulegende Strecke für den gegenläufigen Strahl. Durch den Sagnac-Effekt ändern sich dadurch die Frequenzen der beiden Strahlen. Während bei einem Laserstrahl die Frequenz steigt, sinkt sie beim Gegenläufer. Um die Differenz der Frequenzen zu messen werden an einem Punkt beide Laserstrahlen überlagert und bilden somit ein Interferenzmuster. Durch die Auswertung dieses Patterns ist eine genaue Drehratenbestimmung möglich.

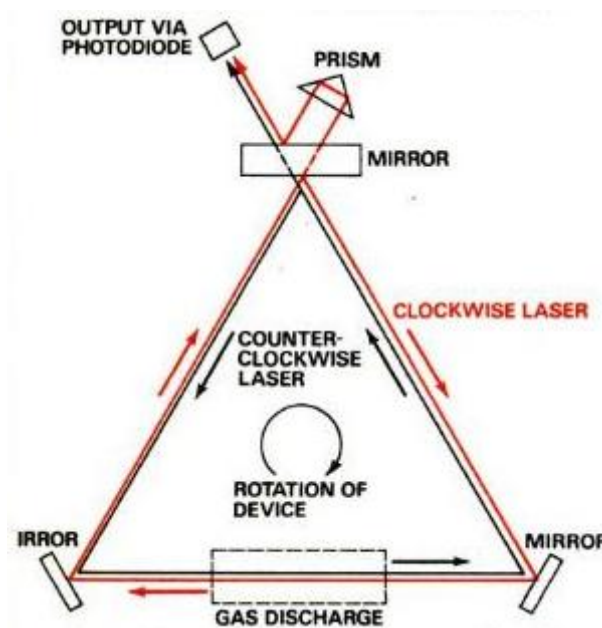


Abbildung 5 - Aufbau eines Lasergyroskops⁹

Vibrationsgyroskope werden häufig in der Mikrotechnik eingesetzt. Im Inneren wird ein Kristall zum vibrieren gebracht. Kristalle induzieren einen Strom, wenn sie zusammengedrückt oder gestreckt werden. Das ist der sogenannte Piezoeffekt (griech $\pi\epsilon\sigma\eta$, Druck). Dieser Effekt ist auch umkehrbar: Das Kristall bzw. piezoelektrische Material kann zum Vibrieren gebracht werden, wenn es mit einer Wechselspannung verbunden wird.

⁸ [13]

⁹ [13]

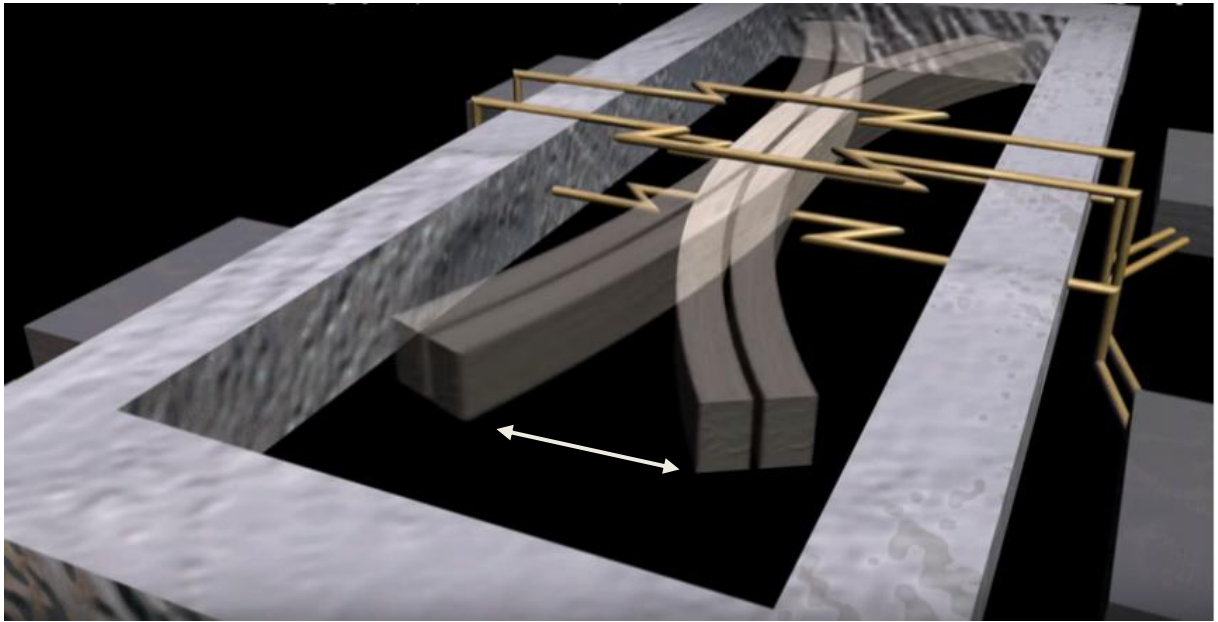


Abbildung 6 - vibrierendes Piezoelektrikum in Vibrationsgyroskop¹⁰

Abbildung 6 zeigt (vereinfacht) zwei Piezoelektrika im Vibrationsgyroskop. Durch Dehnung und Kontraktion dieser Piezoelektrika, werden sie zum Vibrieren gebracht. Bei einer Drehung des Gyroskops werden die Piezoelektrika durch die Coriolisbeschleunigung und ihre Trägheit verformt (Abbildung 7). Diese Verformung induziert eine Spannung, welche in eine Drehrate übersetzt werden kann. Das Vorzeichen gibt die Drehrichtung an.

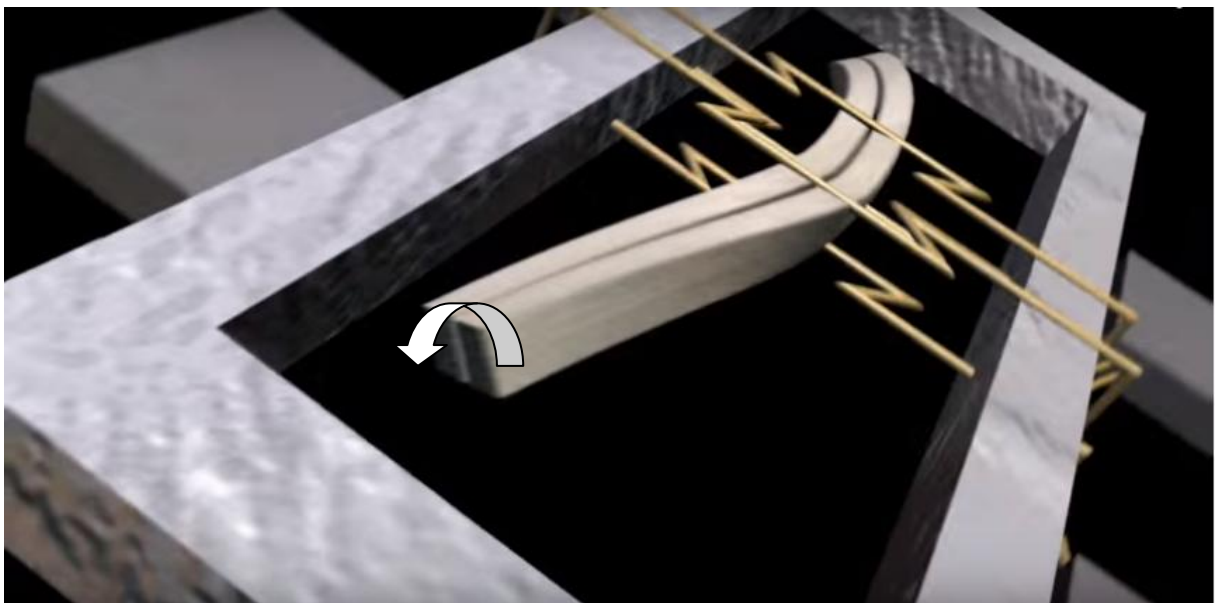


Abbildung 7 - rotierendes Vibrationsgyroskop

¹⁰ nach [14]

2.5.1.3 Anwendung

Das Gyroskop kommt bevorzugt in der Navigation zum Einsatz. Dort kann durch Integration der Drehraten, die Orientierung des Objekts bestimmt werden. Im Kraftfahrzeug unterstützt es das Navigationssystem, in der Luftfahrt kommt es in einer Vielzahl von Instrumenten vor, zum Beispiel im künstlichen Horizont oder im Wendekoordinator. Außerdem werden Gyroskope unterstützend in der Unterhaltungsindustrie eingesetzt. Smartphone-Hersteller stellen Gyroskope zur Verfügung, die zur Steuerung von mobilen Applikationen genutzt werden. Erstmals wurden 2009 in den WiiMotion-Controllern von Nintendo Gyroskope in der Spielindustrie eingesetzt¹¹.

2.5.2 Beschleunigungsmesser

Ein Beschleunigungsmesser, auch Beschleunigungssensor oder Accelerator, misst die Beschleunigungskraft, die auf ein Objekt wirkt. Das Funktionsprinzip eines Beschleunigungsmessers ist in Abbildung 8 zu sehen.

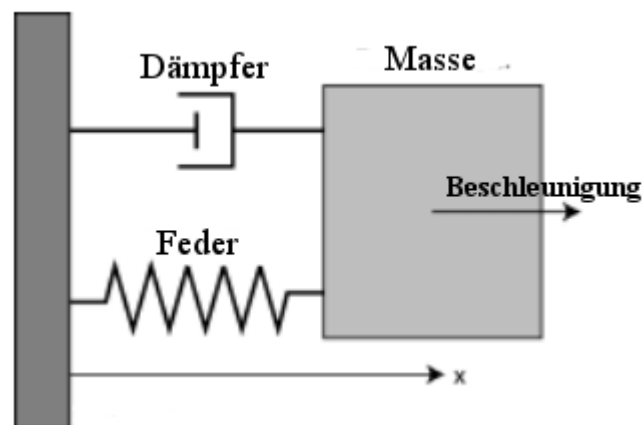


Abbildung 8 - Funktionsprinzip eines Beschleunigungsmessers¹²

Wenn eine Beschleunigung auf den Sensor wirkt, bewegt sich die Masse entsprechend in die Gegenrichtung. Das liegt daran, dass Körper dazu bestrebt sind trotz Krafteinfluss, ihre Bewegung beizubehalten (Trägheitsprinzip). Wenn die Federkonstante und die Masse bekannt sind, ist es möglich nach Newtons zweitem Gesetz der Bewegung $\vec{F} = m\vec{a}$ die Beschleunigung \vec{a} zu bestimmen. Die Feder wird außerhalb ihres Resonanzbereichs betrie-

¹¹ [4]

¹² nach [15] S.19

ben. Allerdings gibt es hochpräzise Beschleunigungssensoren, bei denen die Feder innerhalb des Resonanzbereichs betrieben wird, um somit die Bewegung der Masse zu verstärken¹³.

2.2.2.1 MEMS-Beschleunigungssensor

In der MEMS-Technologie kommen bevorzugt zwei Arten von Beschleunigungssensoren vor: kapazitive Beschleunigungssensoren und Piezo-Beschleunigungssensoren.

Es gibt viele Bauweisen von *kapazitiven Beschleunigungssensoren*, das Prinzip ist jedoch bei allen gleich: Eine Masse ist zwischen zwei Kondensatorplatten befestigt. Wirkt eine Beschleunigung auf die Masse, bewegt sie sich und ändert somit die Kapazität des Kondensators. Da das verwendete Plattenkondensatormodell bekannt ist, kann die Sensorfunktion daraus abgeleitet werden und die Beschleunigung bestimmt werden¹⁴.

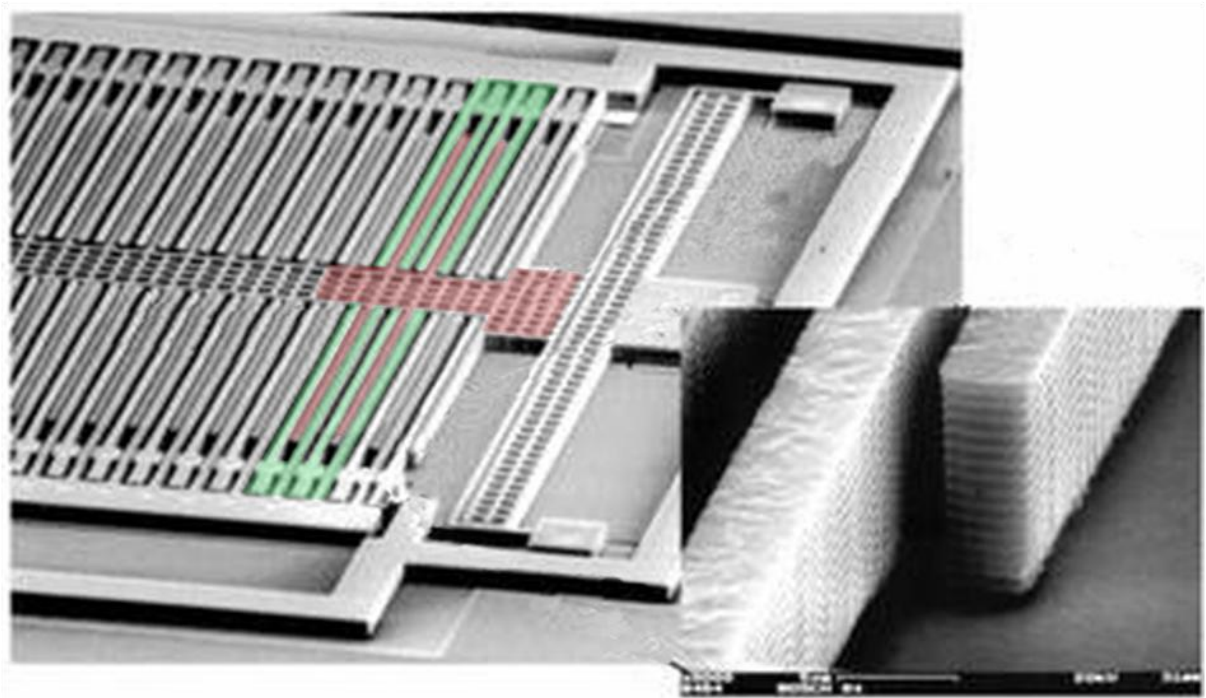


Abbildung 9 - kapazitiver Beschleunigungssensor durch ein Rasterelektronenmikroskop¹⁵

In Abbildung 9 wurden einige Bereiche farblich hervorgehoben. Die rot gefärbten Stifte, welche am Gitter hängen, ist die Masse, welche sich zwischen zwei Platten (grüner Bereich) bewegt und dadurch die Kapazität ändert.

¹³ [15]

¹⁴ [16]

¹⁵ nach [17]

Beim *piezoelektrischen Beschleunigungsmesser* wird, wie beim Vibrationsgyroskop (s. Kapitel 2.5.1.2), der Piezo-Effekt genutzt. Eine Masse „liegt“ dabei auf einem piezoelektrischen Material. Wirkt eine Beschleunigung auf die Masse, staucht oder dehnt sie das Piezoelektrikum. Daraus kann die Beschleunigung ermittelt werden.

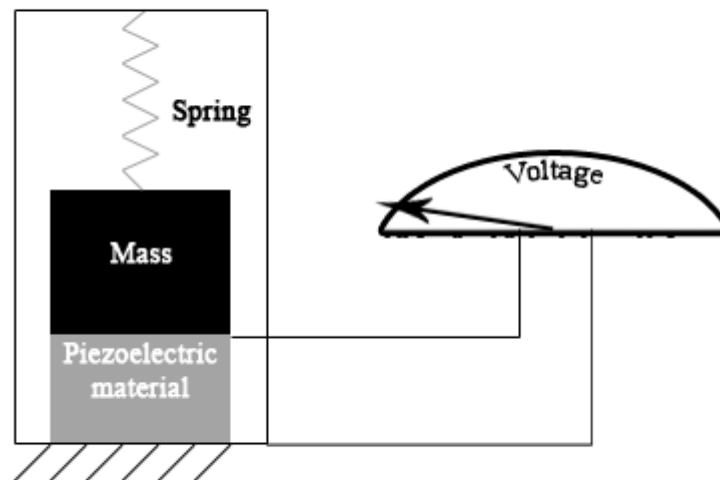


Abbildung 10 - Piezoelektrischer Beschleunigungsmesser¹⁶

3 Theoretischer Hintergrund

3.1 Rotation

Eine Rotation ist die Drehung eines Vektors oder eines Körpers um eine (Rotations-) Achse. In diesem Kapitel wird bei Rotationen von Koordinatensystemen ausgegangen. Mit Bezug zu dieser Thesis, kann beim Referenzkoordinatensystem vom n-frame ausgegangen werden. Rotiert wird der körperbezogene body-frame (s. Kapitel 2.2).

3.1.1 Rotationsmatrix

Ein einfaches Beispiel einer Rotationsmatrix im zweidimensionalen Raum:

¹⁶ [25]

$$R = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (3-1)$$

Im zweidimensionalen Raum kann nur um eine Achse bzw. um den Ursprung rotiert werden. Demnach wird nur eine Winkelangabe benötigt um eine Rotation zu beschreiben. Winkel, die die Rotation um eine Achse angeben, heißen *Eulerwinkel*.

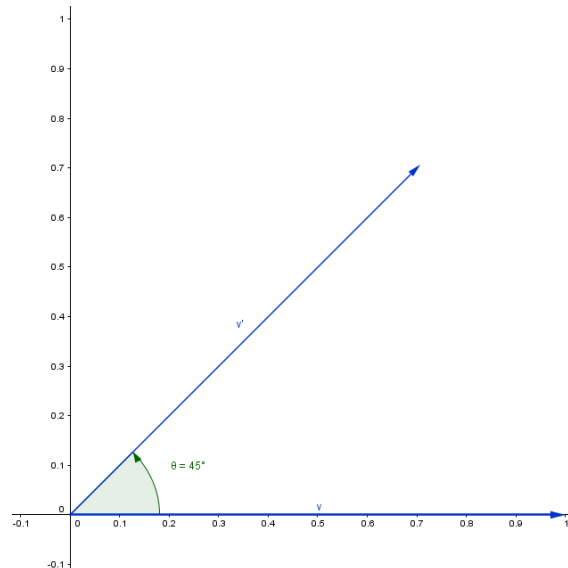


Abbildung 11 - Rotation im zweidimensionalen Raum

In Abbildung 11 wird der Einheitsvektor v um 45° gedreht. Dafür muss v mit R multipliziert werden:

$$v' = R \cdot v = \begin{pmatrix} \cos(45^\circ) & -\sin(45^\circ) \\ \sin(45^\circ) & \cos(45^\circ) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad (3-2)$$

Im dreidimensionalen Raum sind Rotationen um die x, y und der z -Achse möglich. Eine Rotation wird durch drei Winkelangaben (ein Winkel für jede Achse) beschrieben. Anders ausgedrückt: Eine Rotation wird als Produkt der Achsenrotationen beschrieben:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} = R_x R_y R_z \quad (3-3)$$

Mit α : Rotation um z_n

β : Rotation um y_n

γ : Rotation um x_n

Die Reihenfolge der Rotationen ist nicht beliebig. In (3-3) wird zunächst die Rotation um die Achsen z_n , dann um y_n und schließlich um x_n durchgeführt. Rotationen, die sich auf das Referenzsystem beziehen, heißen *absolute Rotationen*.

Bei *relativen Rotationen* wird (3-3) von ‚links nach rechts‘ gelesen. Dabei beziehen sich die Rotationen nicht mehr auf dem n-frame, sondern auf die Achsen des b-frame:

- Rotation um x_b um γ (hier noch $x_b = x_n$)
- Rotation um y_b um β
- Rotation um z_b um α

3.1.2 Quaternionen

Quaternionen wurden 1843 von Sir William Rowan Hamilton erdacht, als er nach einer Multiplikation von Tripeln suchte, für die die mathematischen Gesetze weiterhin gelten. Sie werden durch einen reellen und drei imaginären Teilen dargestellt:

$$q = r + ix + jy + kz \quad (3-4)$$

Dabei gilt für die imaginären Zahlen i, j und z :

$$i^2 + j^2 + k^2 = ijk = -1 \quad (3-5)$$

Die definierten Rechenregeln für Quaternionen, sind der Anlage zu entnehmen.

3.1.2.1 Quaternionen und Rotationen

Quaternionen können zur Beschreibung von Rotationen genutzt werden (Abbildung 12). Die drei imaginären Teile definieren den Einheitsvektor um den rotiert wird. Der reelle Teil gibt an, wie stark rotiert wird.

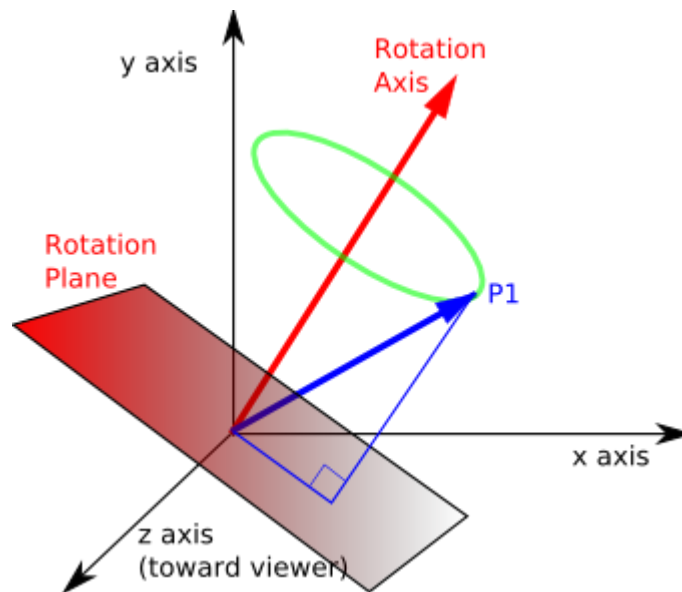


Abbildung 12 - Quaternionen und Rotation¹⁷

3.1.3 Vergleich von Quaternionen und Rotationsmatrizen

Es gibt zwei wesentliche Punkte, die für den Einsatz von Quaternionen zur Lagebestimmung sprechen.

Gimbal Lock

Der *Gimbal Lock*, auch *kardanische Blockade*, ist in Abbildung 13 zu sehen:



Abbildung 13 - Gimbal Lock¹⁸

Beim Gimbal Lock liegen zwei (oder drei) Rotationsachsen übereinander und blockieren somit eine Rotationsrichtung. In der rechten Abbildung kann mathematisch nicht mehr das

¹⁷ [26]

¹⁸ [21]

Gieren des Flugzeugs beschrieben werden. Das führt zu dem Verlust eines Bewegungsgrads (DOF).

Bei Quaternionen kann der Gimbal Lock nicht auftreten, da im Gedankenmodell, wie in Kapitel 3.1.2.1 erwähnt, die imaginären Teile jede beliebige Rotationsachse definieren können.

Vorteile bei Kalkulationen

Eine Rotation wird bei mit einem Quaternion anhand von vier Elementen beschrieben. Rotationsmatrizen benötigen dafür neun Elemente. Um Rotationen zu verknüpfen werden bei Quaternionen 16, bei Rotationsmatrizen 27 Multiplikationen benötigt. Beide Punkte führen beim Einsatz von Quaternionen zu einer schnelleren Berechnung mit geringerem Speicherbedarf als bei Rotationsmatrizen.

Außerdem bringt die geringere Anzahl der Multiplikationen auch Vorteile bei der Genauigkeit. Sie verringert das Fortpflanzen von Fehlern, die wegen der endlichen Rechengenauigkeit von Prozessoren zustande kommen. Diese Fehler bewirken den Verlust der Orthonormalbasis des Koordinatensystems.

Ein Nachteil ist, dass die Anschaulichkeit bei Quaternionen sehr gering ist. Sie werden stets genormt dargestellt und Rotationsachse und -winkel sind nicht erkennbar,

4 Filter

Da die Sensordaten stark verrauscht sind, müssen diese vor der Verarbeitung gefiltert werden. Bei verrauschten Daten bietet sich ein Tiefpass bzw. ein Filter mit Tiefpasscharakteristik an. Außerdem wird bei der Bestimmung der Orientierung oft eine Sensorfusion genutzt. Dabei unterstützen die Beschleunigungsmesser die Gyroskope. Das bietet sich an, da mit dem Gravitationsvektor ein fester Bezug für Roll und Pitch zur Verfügung steht.

Neben dem klassischen Tiefpassfilter kommt vor allem im Amateurbereich (Modellbau, Roboterbau) der Komplementärfilter vor, da dieser recht einfach zu implementieren und zu parametrisieren ist.

Eine etwas sophisticatedere Methode bieten die Bayes-Filter. Es sind statistische Filter, welche anhand vorhandener Sensordaten einen Erwartungswert ausgeben, welcher dann wieder anhand von Messdaten, respektive der Kovarianzen korrigiert wird.

Ein linearer Bayes-Filter ist der Standard Kalman-Filter (KF). Als state of the art im Navigationsbereich, hat sich der Extended Kalman Filter (EKF) durchgesetzt, welcher auch bei nichtlinearen Systemmodellen eingesetzt werden kann.

Der KF und EKF können bei Daten mit gaußischem Rauschen eingesetzt werden. Bei dieser Arbeit wird ein gaußisches Rauschen für die Sensordaten angenommen.

4.1 Kalman Filter

Im Allgemeinen sagt der Kalman Filter Zustände voraus (*Prediction, Schätzung*) und gleicht diese dann mit den Zuständen, welche aus Messwerten ermittelt worden sind, ab (*Update, Correction*).

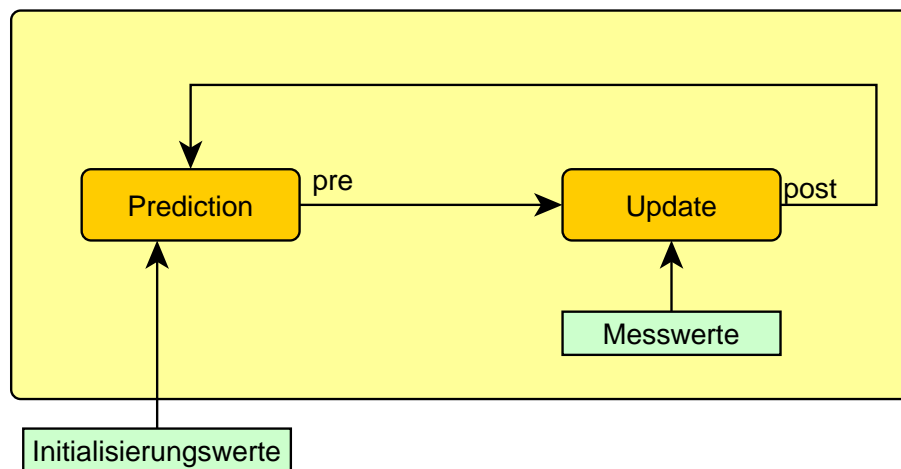


Abbildung 14 - Kalman Filter, Ablauf allgemein

4.1.1 Einführung

In diesem Kapitel wird die grundlegende Funktionsweise eines Kalman Filters an einem Beispiel beschrieben.

Es soll die Drehrate um die x-Achse eines Objekt integriert und somit der Winkel (Roll) bestimmt werden. Die Drehrate beträgt $1 \text{ }^\circ/\text{s}$ für 45s. Über den Messwerten des Gyroskops liegt ein Rauschen mit einer Standardabweichung $\sigma = 0,02$:

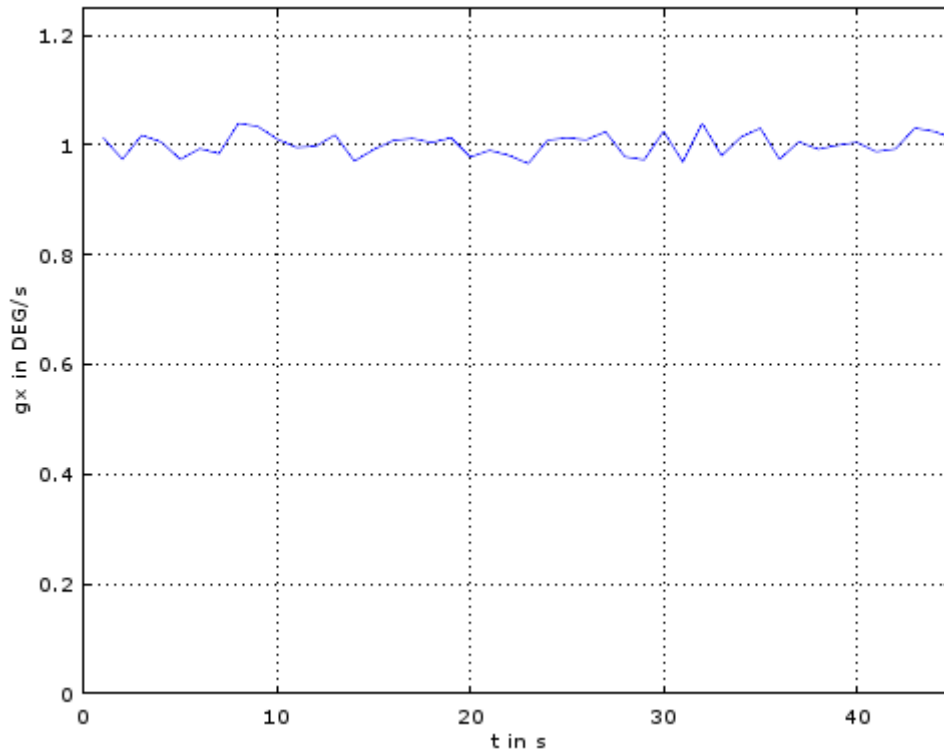


Abbildung 15 - Verrauschte Werte vom Gyroskop; Standardabweichung Rauschen $\sigma=0.02$

Die Integration der Werte führt zu falschen Winkelangaben:

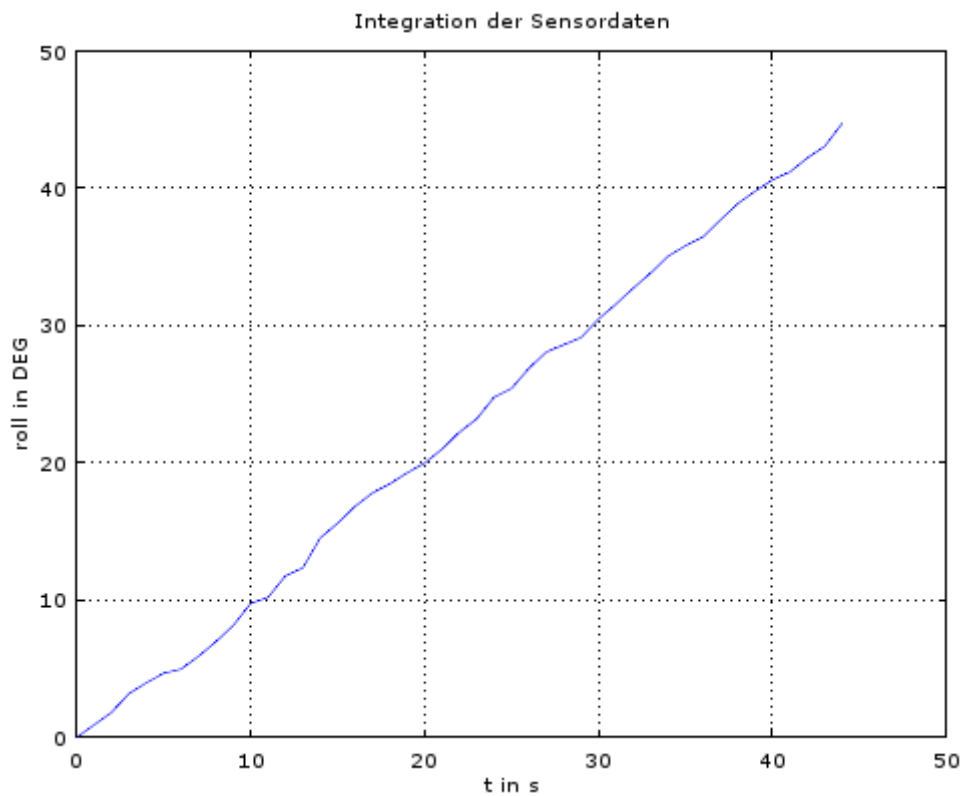


Abbildung 16 - Integration der verrauschten Sensordaten

4.1.2 Varianz und Mittelwert

Ein Merkmal vom Kalman-Filter ist, dass er nicht mit Absolutwerten rechnet sondern mit Mittelwerten μ und Varianzen σ^2 . Die Varianz kann so verstanden werden, als dass sie die Unsicherheit bezüglich des Mittelwertes wiedergibt. Dabei gilt: je größer die Varianz, desto größer die Unsicherheit.

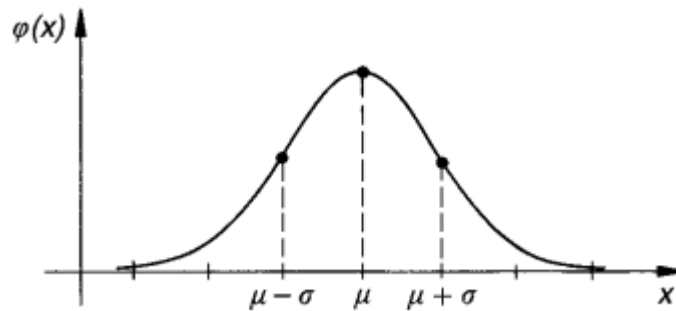


Abbildung 17 - Verteilungsdichtefunktion mit Mittelwert und Varianzen¹⁹

In Abbildung 17 ist die Verteilungsdichtefunktion $\phi(x)$ einer Gaußschen Normalverteilung dargestellt. Die Standardabweichung σ kann an den Wendepunkten der Funktion bestimmt werden. Hier ist auch noch einmal zu erkennen, je größer die Varianz, also je mehr sich die Wendepunkte vom Mittelwert entfernen, desto größer ist die Unsicherheit bei der Annahme des Mittelwertes (entsprechend: Zustand beim Prediction-Schritt des Kalman-Filters).²⁰

Nach der Initialisierung, also beim ersten Iterationsschritt, wird der vermutete Wert (0°) mit einer sehr geringen Wahrscheinlichkeit angenommen (Abbildung 18).

¹⁹ [22] S.292

²⁰ Bei der Parametrisierung des Kalman-Filters sind die angegebenen Varianzen (Q und R-Matrix, s. Kapitel 4.2.1.3) maßgeblich für dessen richtigen Funktion.

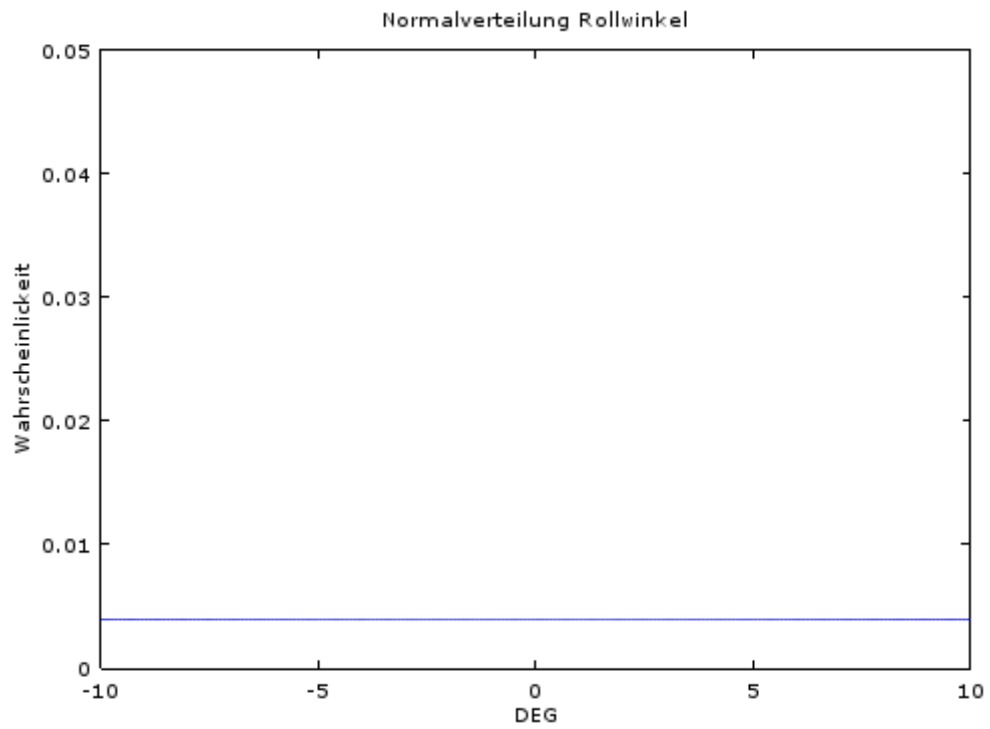


Abbildung 18 - Normalverteilung DEG nach Initialisierung

Mit jeder Iteration nimmt die Varianz ab und der folgende Wert kann mit höherer Wahrscheinlichkeit bestimmt werden (folgt in Kapitel 4.1.3).

4.1.3 Prozessschritte ‚Prediction‘ und ‚Update‘

Der Kalman-Filter führt bei jedem Iterationsschritt eine Vorhersage (Prediction) und eine Korrektur (Update, Correction) durch.

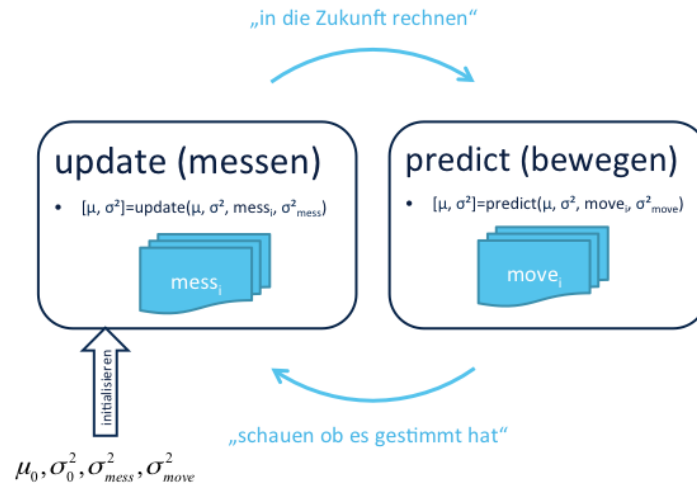


Abbildung 19 - Ablauf 1D - Kalman-Filter²¹

Prediction

Beim Prediction-Schritt wird der folgende Messwert geschätzt. Dies geschieht anhand der Normalverteilung die aus dem Update-Schritt resultierte. Die Varianz erhöht sich, da die Wahrscheinlichkeit sinkt beim Berechnen eines zukünftigen Wertes. In diesem Schritt spielt die Systemmodellierung eine Rolle. Im vorliegenden Beispiel werden die Mittelwerte und Varianzen addiert.

$$\mu_{k+1} = \mu_k + \mu \tag{4-1}$$

$$\sigma_{k+1}^2 = \sigma_k^2 + \sigma^2 \tag{4-2}$$

²¹ [27]

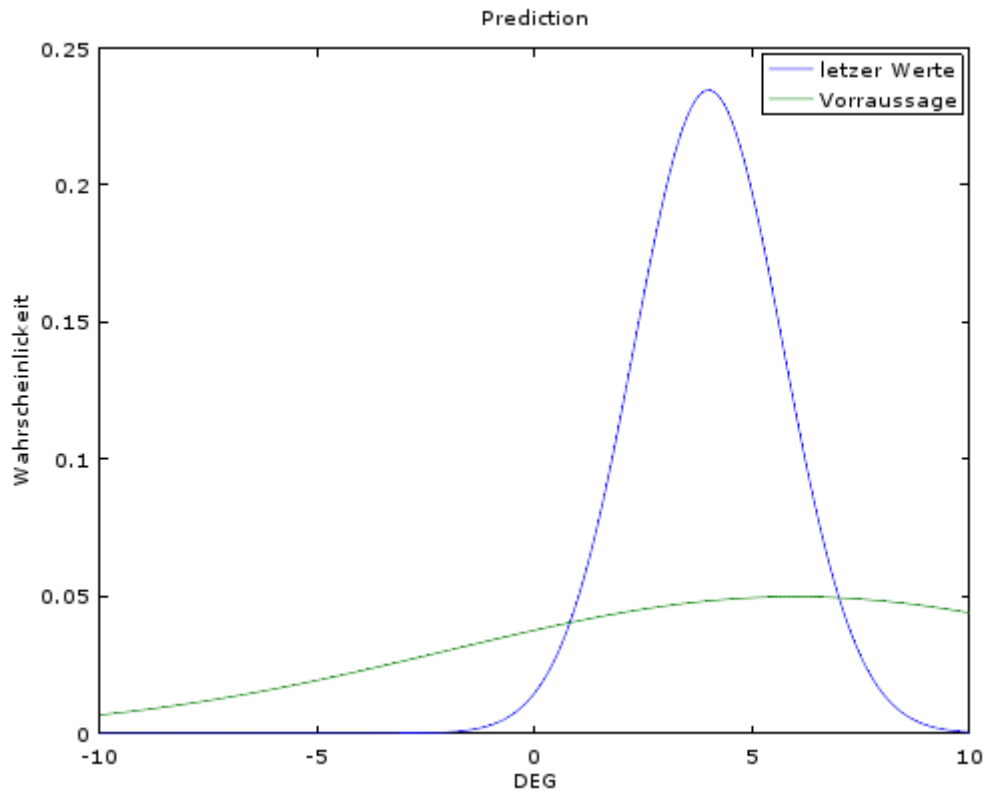


Abbildung 20 - Normalverteilung nach dem Prediction-Schritt

Der Prediction-Schritt wird immer mit einer festen Varianz durchgeführt. Diese wird vor dem Einsatz des Kalman-Filters festgelegt.

Update

Beim Update werden die Normalverteilung des vorhergesagten Werts und des aktuell gemessenen Werts addiert. Da beide Werte mit unterschiedlichen Varianzen angenommen werden, werden diese mit Hilfe der Bayes'schen Regel addiert.²² Dabei erfolgt eine Gewichtung anhand der Varianzen. Der resultierende Mittelwert, befindet sich näher am Mittelwert mit der geringeren Varianz.

Bayes'sche Regel:

$$\sigma_{k+1}^2 = \frac{1}{\left(\frac{1}{\sigma_{1,k}^2} + \frac{1}{\sigma_{2,k}^2}\right)} \quad (4-3)$$

$$\mu_{k+1} = \frac{\sigma_{2,k}^2 \cdot \mu_{1,k} + \sigma_{1,k}^2 \cdot \mu_{2,k}}{\sigma_{1,k}^2 + \sigma_{2,k}^2} \quad (4-4)$$

²² [27]

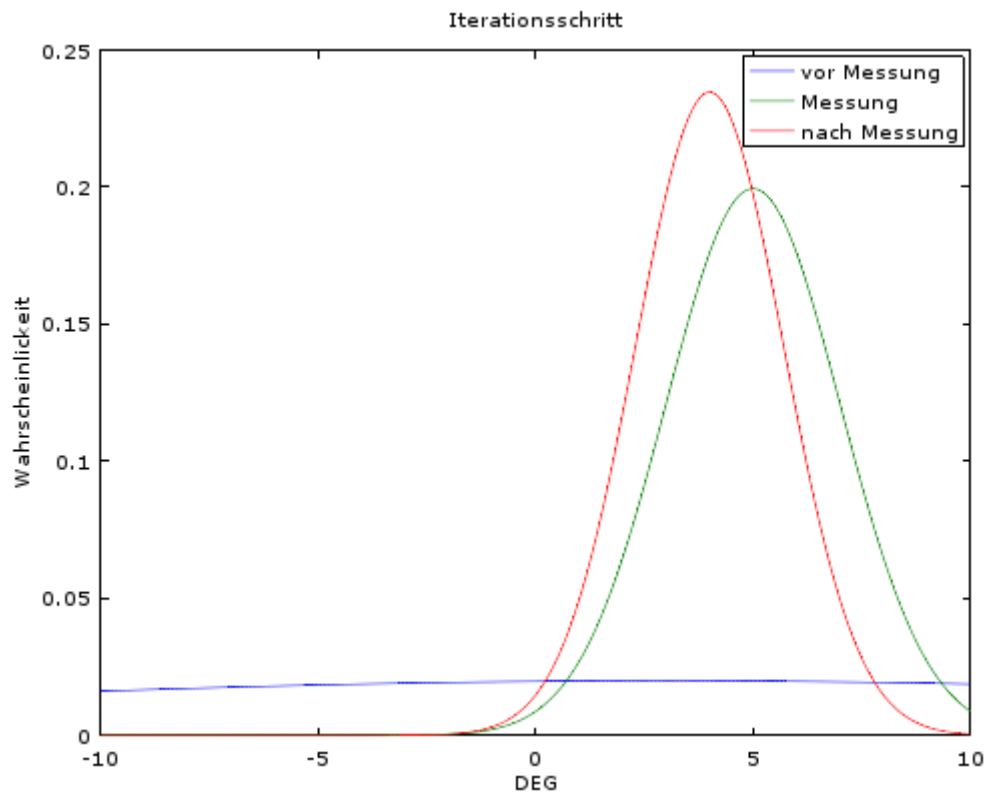


Abbildung 21 - Darstellung der Normalverteilungen während einer Iteration

Außerdem verringert sich die Varianz. Die resultierende Normalverteilung ist nun der Ausgangszustand für den Prediction-Schritt der nächsten Iteration.

Beispiel Anwendung

Das Beispiel wurde mit zwei unterschiedlichen Rauschvarianzen simuliert:

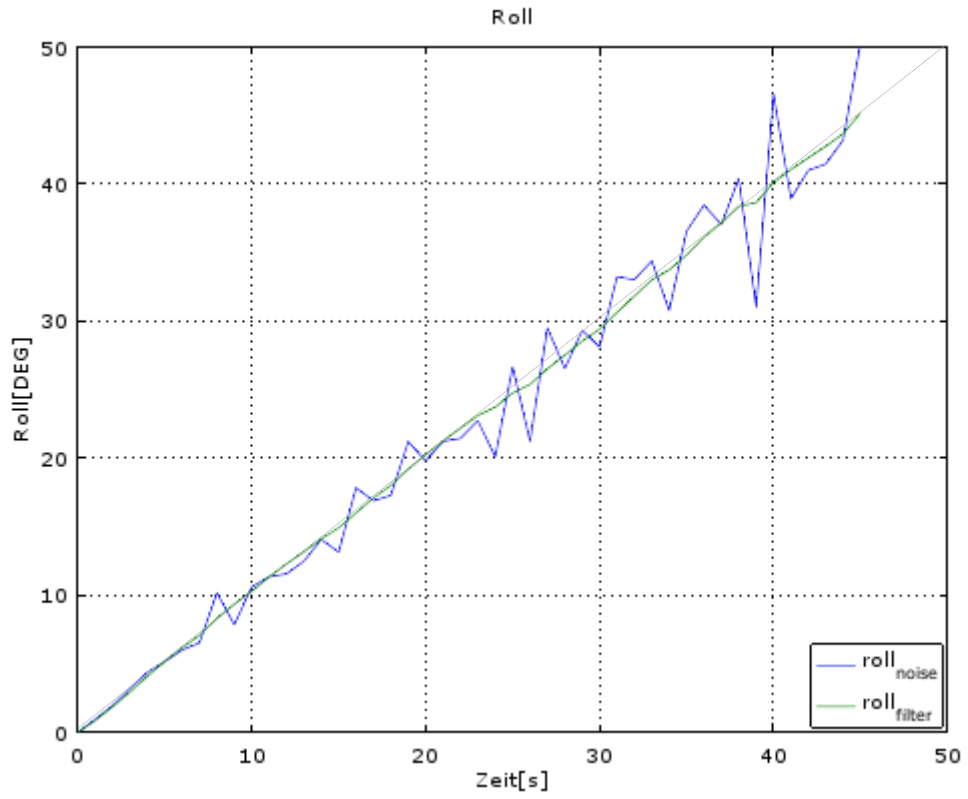


Abbildung 22 - 1D-KF, Sensordaten mit Rauschvarianz = 0.1, Systemvarianz = 0.1

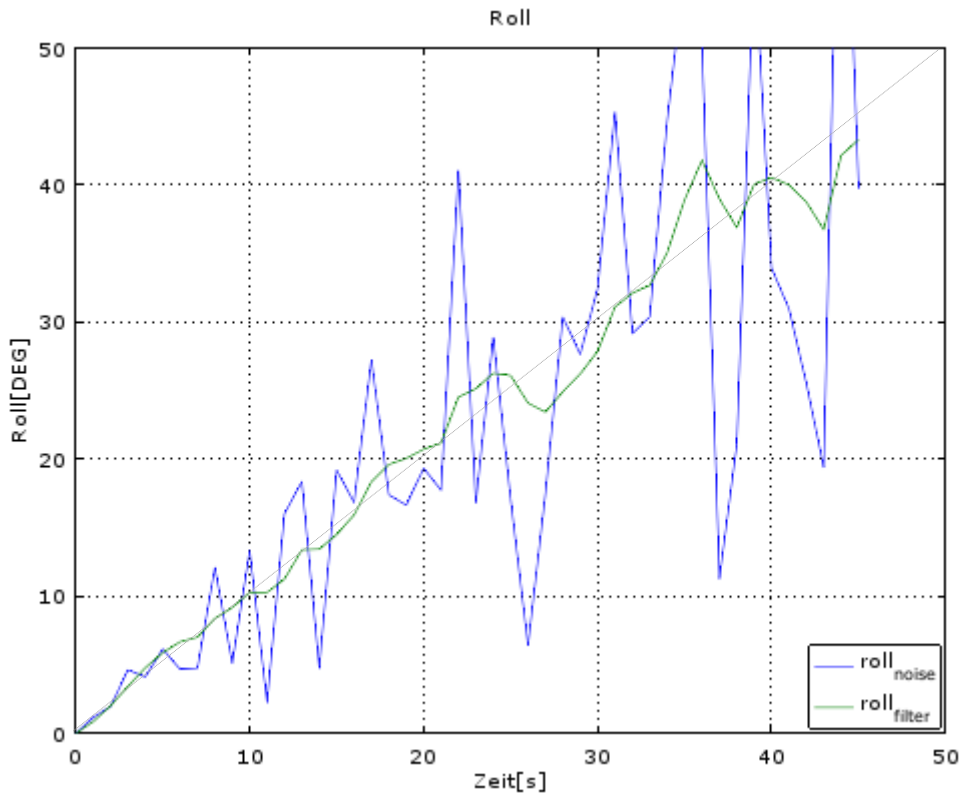


Abbildung 23 - 1D-KF, Sensordaten mit Rauschvarianz = 0.4, Systemvarianz = 0.1, Messvarianz = 5

Die blauen Graphen, sind die Integration der Sensordaten ohne Filterung. Die grünen Graphen stellen den Ausgang des Kalman-Filters dar.

In beiden Bildern ist an den blauen Graphen sehr schön zu erkennen, dass sich bei der Integration von verrauschten Daten die Fehler fortpflanzen, also mit integriert werden und der Absolutfehler sich erhöht.

In Abbildung 22 haben die Sensordaten eine Rauschvarianz von 0.1. Die Systemvarianz wurde mit 0.1 gewählt, deutlich geringer als die Messvarianz von 5. Der Kalman-Filter ‚vertraut‘ somit, den vor raus bestimmten Werten (Prediction) eher als den gemessenen.

Bei der Simulation in Abbildung 23 wurde die Rauschvarianz erhöht auf 0.4 und die Systemvarianz bei 0.1 belassen. Der Filterausgang ist deutlich verrauschter als in Abbildung 22.

Um zu zeigen, wie sehr die Parametrisierung der Varianzen des Kalman-Filters, dessen Funktion beeinflussen, wurde beim folgenden Beispiel die Systemvarianz mit 0.01 gewählt:

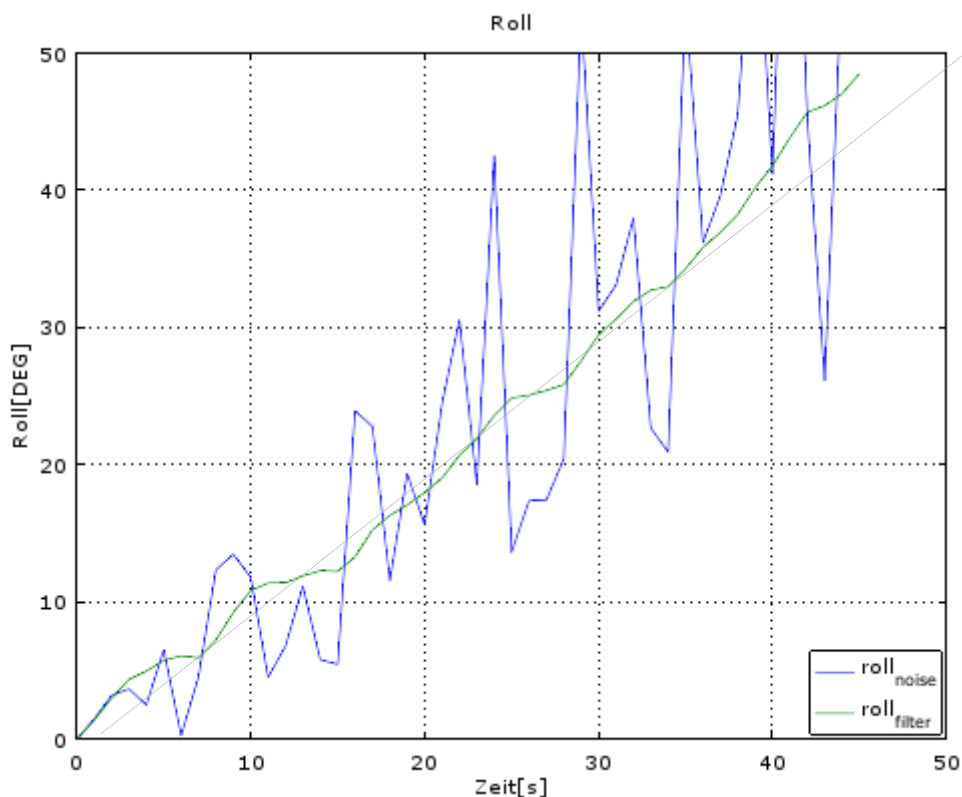


Abbildung 24 - 1D-KF, Sensordaten mit Rauschvarianz = 0.4, Systemvarianz = 0.01, Messvarianz = 5

Der Filterausgang ist deutlich glatter als in Abbildung 23. Da die Systemvarianz geringer gewählt wurde, folgt er den vor raus bestimmten Werten nicht mehr so stark.

Es liegt jetzt nahe, bei stark verrauschten Daten die Systemvarianz besonders klein und die Rauschvarianz besonders hoch zu stellen. Hier ein Beispiel mit einer Systemvarianz von $1e-5$ und einer Messvarianz von 1000:

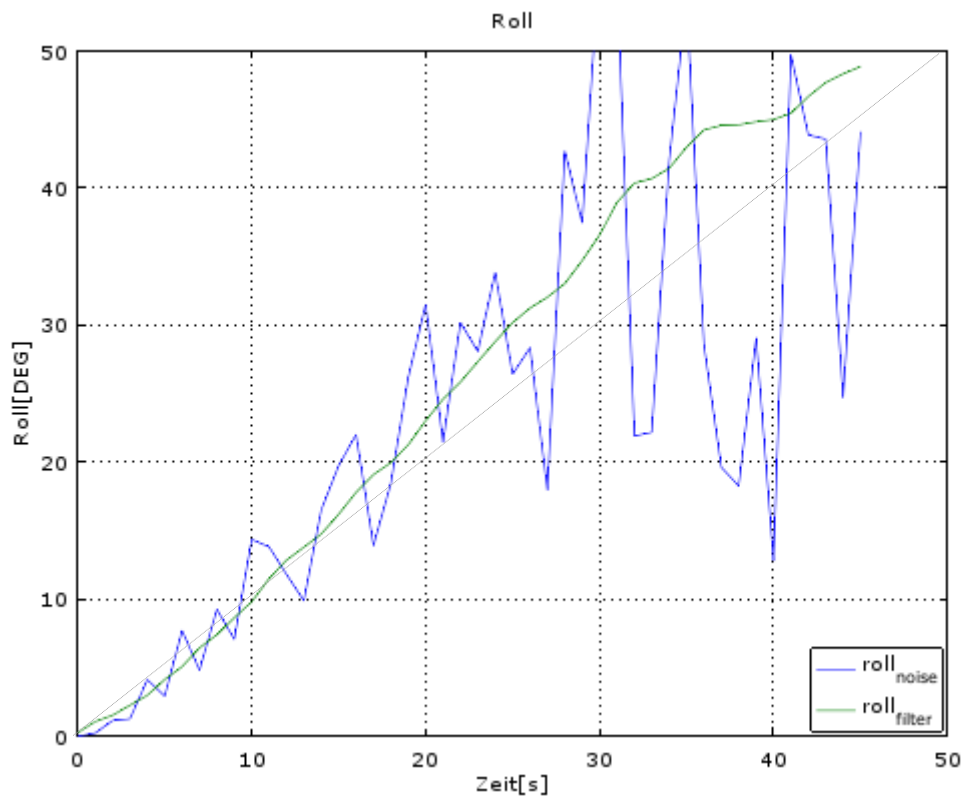


Abbildung 25 - 1D-KF mit Rauschvarianz = 0.4, Systemvarianz = $1e-5$, Messvarianz = 1000

Der Ausgang des Kalman-Filters ist zwar glatt, verläuft sich jedoch und grenzt nicht mehr an den gewünschten Graphen, wie in den anderen Abbildungen. Die Werte sind somit nicht mehr für eine Weiterverarbeitung zu gebrauchen.

4.2 Mehrdimensionaler Kalman Filter

Der Kalman-Filter soll am Ausgang die aktuelle Lage des MEMS-Chips ausgeben. Als Eingabedaten dient die Lageänderung, berechnet anhand der Gyroskope, und als Messeingaben die Lage berechnet anhand der Beschleunigungsmesser.

4.2.1 Kalman-Filter Gleichungen

Wie in der Einführung erwähnt, besteht ein Kalman-Filter im Wesentlichen aus zwei Schritten, Prediction und Update (s. Kapitel 4.1.3). Ein großer Nachteil vom Kalman-Filter ist, dass System- und Messfunktionen linear sein müssen. Aus diesem Grund wurde für das Projekt die Formulierung des Extended Kalman Filter (EKF) benutzt. Beim Kalman-Filter werden der Schätzzustand und der neue Zustand mit *priori* und *posteriori* unterschieden (auch *pre* und *post*). A-priori sind Werte nach der Prediction und werden mit einem hochgestellten Minus gekennzeichnet: x_k^- und P_k^- . A-posteriori werden die Werte nach dem Update bezeichnet und mit einem hochgestellten Plus gekennzeichnet: x_k^+ und P_k^+ .

4.2.1.1 Prediction

$$x_k^- = Ax_{k-1}^+ + Bu_k + w_k \quad (4-5)$$

Der Zustand aus der vorangegangenen Iteration des Kalman-Filters wird mit der Systemmatrix A multipliziert. Mit Hilfe der Systemmatrix (auch Dynamikmatrix, Transitionsmatrix) wird das betrachtete System modelliert.

Der Vektor u enthält, falls vorhanden, die aktuellen Eingabegrößen an das System, zum Beispiel die Steuereingabe bei einem Fortbewegungsmittel.

Die Eingangsmatrix B beschreibt den Zusammenhang zwischen Eingabegrößen und Zustandsvektor. Ein simples Beispiel: die Lenkradstellung bei einem Kraftfahrzeug wird in die Winkelstellung der Vorderreifen umgerechnet.

Die Variable w ist zufällig und steht für das Rauschen des Systems. Beim Kalman-Filter wird das Rauschen als weißes, gaußsches Rauschen vorausgesetzt. Es enthält alle Frequenzen, ist nicht korreliert und hat ein konstantes Leistungsspektrum.

$$P_k^- = AP_{k-1}^+ A^T + Q \quad (4-6)$$

Die Schätzfehlerkovarianzmatrix P beschreibt die Unsicherheit der Werte nach der Zustandsschätzung (Prediction) in dem sich die Werte ihrer Matrixelemente erhöhen.

4.2.1.2 Update

Im Update-Prozess muss zunächst der Kalman-Gain K bestimmt werden. Er legt fest inwiefern die Mess- oder Schätzwerte übernommen werden:

$$K_k = \frac{P_k^- H}{H P_k^- H^T + R} \quad (4-7)$$

Die Messmatrix H transferiert die Messwerte in den Zustandsraum. Die Messkovarianzmatrix R gibt die Kovarianz der Messung wieder. Sie wird, zusammen mit der Systemkovarianzmatrix Q bei der Initialisierung des Filters festgelegt und ändert sich während der Filterung nicht.

Bevor auf den Kalman-Gain K eingegangen wird, soll noch einmal die Bestimmung des neuen Zustands gezeigt werden:

$$x_k^+ = x_k^- + K_k (z_k - H x_k^-) \quad (4-8)$$

Wobei z der Messvektor, also die Messwerte beinhaltet.

In Gleichung (4-8) kann man die Funktion des Kalman-Gain K erkennen. Der Kalman-Gain kann Werte zwischen 0 und 1 annehmen. Wenn in Gleichung (4-7) die Rauschmatrix R im Grenzfall $R = 0$ ist (die Messwerte sind als sicher anzunehmen), dann geht der Kalman-Gain K gegen 1. In Gleichung (4-8) fließt die Messung maximal gewichtet in den post-Zustand x_k^+ ein. Umgekehrt, geht der Kalman-Gain K bei einem sehr großen Nenner gegen 0. Somit werden die Messwerte in Gleichung (4-8) ignoriert und der priori-Zustand wird zum posteriori-Zustand.

Der Kalman-Filter passt nun mit Gleichung (4-9) die Schätzfehlerkovarianz an:

$$P_k^+ = (I - K_k H) P_k^- \quad (4-9)$$

Auch hier hat der Kalman-Gain K eine ähnliche Funktion wie im vorherigen Schritt. Ist $K = 0$, dann wird die priori-Schätzfehlerkovarianz übernommen. Die Unsicherheit des Filters bleibt gleich. Ist $K = 1$, wird also den Messwerten vertraut, sinkt die Schätzfehlerkovarianz.

4.2.1.3 Systemkovarianzmatrix Q und Messkovarianzmatrix R

Die Messkovarianzmatrix R gibt die Rauschkovarianz der Messungen wieder. Um sie zu bestimmen, können die ungefilterten Werte aufgenommen werden und anhand von mathematisch-numerischen Analysesoftware (z.B. MATLAB oder Octave) die Kovarianz bestimmt werden. Dadurch erhält man einen Richtwert für R .

Die Systemkovarianzmatrix Q gibt an, wie sehr der Systemmodellierung vertraut werden kann. Sie ist nicht deterministisch ermittelbar. Bei adaptiven Kalman-Filtern wird zur Be-

stimmung der ‚kleinste mittlere quadratische Schätzungsfehler‘ (englisch: minimum mean square estimation error) ermittelt.

$$MMSE = \min E[(z_k - x_k^+)] \quad (4-10)$$

Unter Berücksichtigung des MMSE wird die Systemkovarianzmatrix Q variiert, bis der Fehler ein Minimum erreicht hat.

Für die Elemente des Zustandsvektors x wurde die Orientierung als Quaternionen gewählt. Dies führte zu dem Problem, dass die Rauschkovarianzmatrix R nicht direkt aus den Messwerten ermittelt werden konnte.

4.3 Extended Kalman Filter

Ein Nachteil beim Standard Kalman Filter ist, dass er nur für lineare Systeme eingesetzt werden kann. Um den Kalman Filter auch für nicht lineare Systeme einzusetzen, wurde der Extended Kalman Filter (EKF) erfunden, basierend auf den Arbeiten im Ames Research Center.²³

Beim Extended Kalman Filter wird die Systemmatrix A durch die Systemfunktion $f(x, u)$ ersetzt. Die Vektoren x und u stehen auch hier für den Zustands- und Eingabevektor. Außerdem wird der Zustandsvektor x mit der Funktion $h(x, u)$ in den Messraum transferiert.

Der Extended Kalman Filter wird im aktuellen Arbeitspunkt linearisiert. Dies passiert mit Hilfe von Jakobi-Matrizen (s. Gleichung (5-2))

Ein Vergleich der Gleichungen vom Standard Kalman Filter und Extended Kalman Filter ist in Tabelle 1 zu sehen:

	Kalman Filter	Extended Kalman Filter
Prediction	$x_k^- = Ax_{k-1}^+ + Bu_k + w_k$	$x_k^- = f(x_{k-1}^+, u_k)$
	$P_k^- = AP_{k-1}^+ A^T + Q$	$P_k^- = J_{A,k} P_{k-1}^+ J_{A,k}^T + Q$
Update	$K_k = \frac{P_k^- H}{HP_k^- H^T + R}$	$K_k = \frac{P_k^- J_{H,k}^T}{J_{H,k} P_k^- J_{H,k}^T + R}$
	$x_k^+ = x_k^- + K_k(z_k - Hx_k^-)$	$x_k^+ = x_k^- + K_k(z_k - h(x_k^-))$
	$P_k^+ = (I - K_k H)P_k^-$	$P_k^+ = (I - K_k J_{H,k})P_k^-$

Tabelle 1 - Übersicht von Standard und Extended Kalman Filter

²³ [24]

Die durch die Jakobi-Matrizen ersetzten Matrizen H und A haben nun ebenfalls einen Iterationsindex, da sie bei jeder Iteration neu berechnet werden müssen. Dies führt zu einer höheren Prozessorauslastung. Die Berechnungen steigen mindestens quadratisch mit jedem weiteren Element im Zustandsvektor an.

5 Testumgebung und Umsetzung

In diesem Kapitel werden die benutzte Hardware und Software beschrieben. In Kapitel 5.4 wird auf die Umsetzung eingegangen.

5.1 Hardware

5.1.1 MEMS-Chip

Wie der Titel dieser Thesis aussagt, handelt es sich bei den MEMS-Chips um die, die unter der Kategorie ‚low-cost‘ fallen. Es stellt sich die Frage, wann dieses Attribut zutrifft.

Im Sinne der Wörter ‚low-cost‘ sollte der MEMS-Chip günstig sein. Um das zu bestimmen, wurde gezielt nach MEMS-Chip gesucht, welche im Amateur- und Semiprofessionellen technischen Bereichen weit verbreitet sind, da diesen Bereichen häufig nur geringe finanzielle Mittel zu Verfügung stehen. Entsprechen ist viel praktisches Know-How für diese MEMS-Chips vorhanden, welche zu ersten Eindrücken führen können.

In die engere Auswahl kamen der LSM9DS0 von ST, der Bosch BMX055 und der MPU9250 bzw. der MPU6050 von Invensense.

In Tabelle 2 sind die Daten der MEMS-Chips aufgelistet:

	Preis (Breakout)	höchste Auflösung, Gyroskope	höchste Auflösung, Beschleunigungsmesser	
ST LSM9DS0	25,95\$	$\pm 245^\circ/\text{s}$	$\pm 2\text{g}$	kA
Bosch BMX055	29.95\$	$\pm 125^\circ/\text{s}$	$\pm 2\text{g}$	(A): $150 \mu\text{g}/\sqrt{\text{Hz}}$ (G): $0.014^\circ/\text{s}/\sqrt{\text{Hz}}$
IS MPU6050	5\$ - 10\$	$\pm 250^\circ/\text{s}$	$\pm 2\text{g}$	(A) ²⁴ : $400 \mu\text{g}/\sqrt{\text{Hz}}$ (G) ²⁵ : $0.005^\circ/\text{s}/\sqrt{\text{Hz}}$
IS MPU9250	12\$-20\$	$\pm 250^\circ/\text{s}$	$\pm 2\text{g}$	(A): $300 \mu\text{g}/\sqrt{\text{Hz}}$ (G): $0.01^\circ/\text{s}/\sqrt{\text{Hz}}$

Tabelle 2 - MEMS-Chips Datenvergleich

Ein Vergleich der Trefferergebnisse in einer Online-Suchmaschine ergab, dass die Trefferergebnisse des MPU6050 mindestens vier Mal häufiger waren, als die der restlichen MEMS-Chips. Diese Bachelorthesis bezieht sich somit auf den MPU6050.

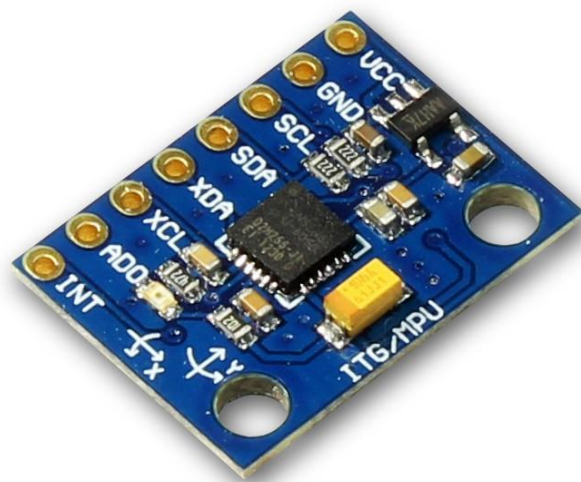


Abbildung 26 - MPU6050 auf einem GY-521 breakout Board

5.1.2 Mikroprozessor

Die Berechnungen und Kommunikation zwischen MEMS-Chip und Computer sollten auf einer Plattform erfolgen, die wenig Einarbeitungszeit bedarf. Dafür bieten sich die Arduino Boards an. Installation von Treiber und IDE erfolgt beim verwendeten Betriebssystem Microsoft Windows unkompliziert.

²⁴ @ 10Hz

²⁵ @ 10Hz

Der Prozessor musste imstande sein die Filterberechnungen in Echtzeit durchzuführen und gleichzeitig an einen PC/Laptop zu übertragen. Ein Vergleich der Arduino Produkte ist in [1] zu sehen. Die Wahl fiel auf den Arduino Due mit einem CPU-Takt von 84MHz. Außerdem stehen 512kB-Flash-Speicher zu Verfügung, welcher genügend Speicherplatz für den Programmiercode bietet.

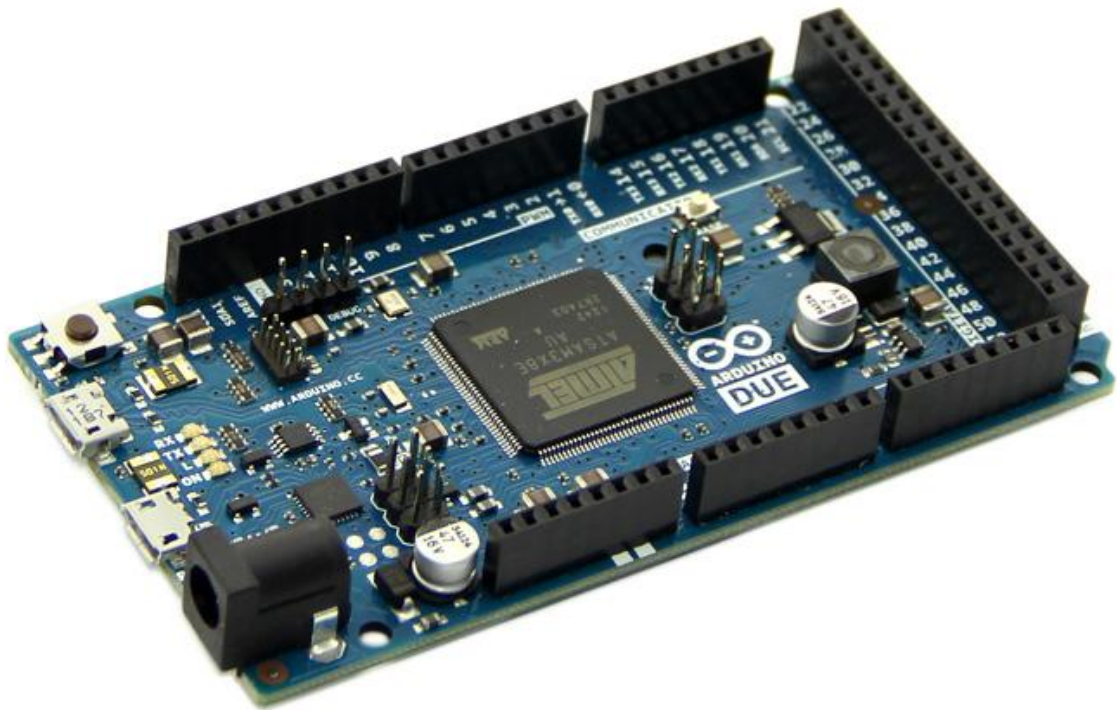


Abbildung 27 - Arduino Due

5.1.3 Bluetooth

Um eine gewisse Bewegungsfreiheit beim Umgang mit dem MEMS-Chip zu haben, wurde von der HAW Hamburg ein Bluetooth-Modul zu Verfügung gestellt. Das SparkFun Bluetooth Modem - BlueSMiRF Gold mit dem RN-41-Modul. Es kann mit 3,3V betrieben werden. Daher wird kein TTL-Konverter benötigt. Die Reichweite kann, laut Datenblatt, bis zu 100m betragen.

Als Empfänger wurde der CSL-USB nano Bluetooth-Adapter V4.0 eingesetzt. Er ist Kompatibel mit dem Microsoft Windows Betriebssystem ab Windows XP.

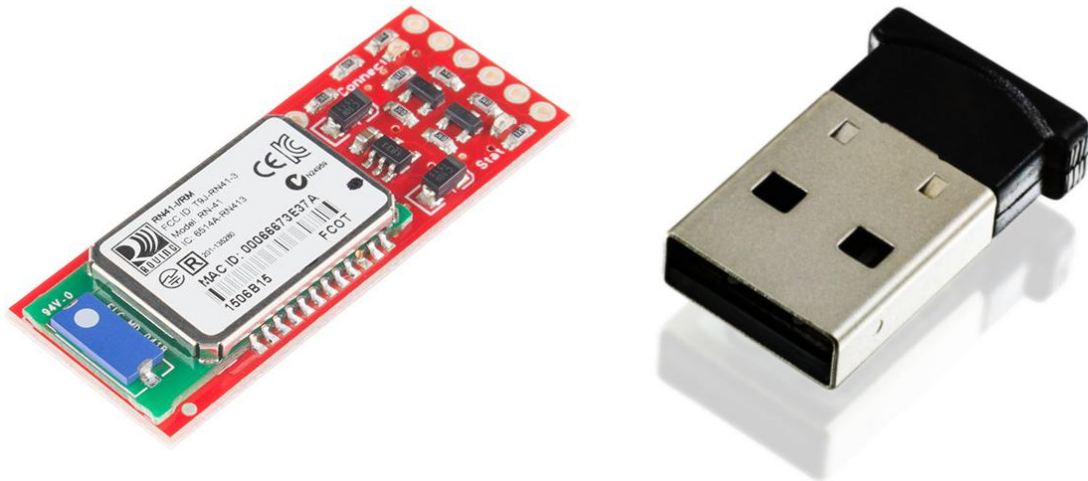


Abbildung 28 - SparkFun Bluetooth Modem - BlueSMiRF Gold und CSL-USB nano Bluetooth Adapter

5.1.4 Befestigung des Chips

Um den Chip in jede Lage bringen zu können, wurde eine Apparatur benötigt, welche dreidimensional ausgerichtet werden kann. Dafür wurde mit der Hilfe von Lego®-Technik eine kardanische Aufhängung nachempfunden. Der MPU6050 ist direkt, auf der inneren Lagerung liegend, daran fixiert (Abbildung 29)

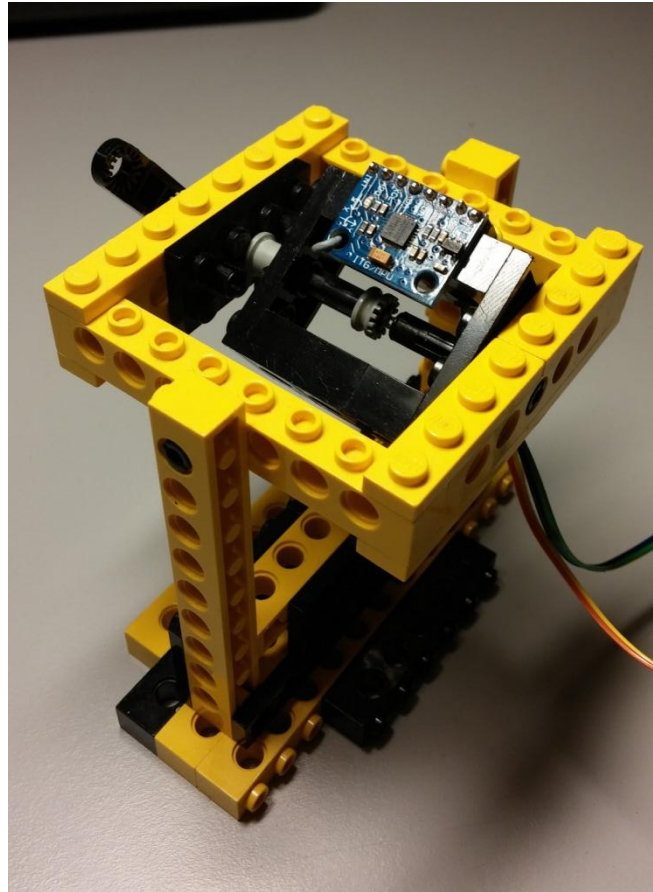


Abbildung 29 - Befestigung des MPU6050

5.2 Software

5.2.1 Arduino IDE, Visual Studio

Die Programmierung erfolgte ursprünglich auf der Arduino IDE 1.6.0. Die Arduino IDE war jedoch zu rudimentär um den Anspruch zu genügen, die Übersicht über mehrere Dateien zu behalten und diese miteinander zu verknüpfen.²⁶

Als Alternative wurden das Atmel Studio 7 von Atmel Corporation und das Visual Studio 2015 Community von Microsoft in Betracht gezogen. Beide können das Arduino Plugin von Visual Micro nutzen, welches die Programmierung der Arduino-Boards erlaubt.

Die Wahl fiel auf Visual Studio 2015 Community. Die Arduino-Plattform werden im Wesentlichen mit einer Sprache programmiert, die sehr stark auf C/C++ angelehnt ist. Visual Studio 2015 bietet darüber hinaus die Möglichkeit Windows-Programme mit C# zu programmieren (s. Kapitel 5.2.2).

²⁶ persönliche Meinung des Authors

5.2.2 Empfang, Verarbeitung und Speicherung

Ein Problem stellte die Wahl der Software dar, welche die Daten vom Arduino empfangen sollte, per Bluetooth und Kabelverbindung und diese möglichst grafisch darstellen sowie speichern kann.

Nach intensiver Suche, wurde festgestellt, dass es keine freie Software, bzw. Software im kostengünstigen Bereich, für diese Anforderungen gab. Die preisgünstigen Student- oder Education-Versionen sind meist nicht für Forschungszwecke zugelassen.

Aus diesem Grund, wurde eine Software speziell für die Bachelorarbeit entwickelt. Als Sprache bot sich die objektorientierte Programmiersprache C# mit der Visual Studio IDE an. Die Windows-Fenster konnten direkt in der IDE grafisch konstruiert werden.

5.2.2.1 Kommunikation

Die Software musste in der Lage sein, die Daten per Bluetooth und bei direkter Kabelverbindung empfangen zu können. Dabei war die Wahl der IDE und Programmiersprache nicht unwesentlich. Die Kommunikation per C# und Visual Studio mit dem Bluetooth-Receiver konnte wie eine standardmäßige serielle Verbindung behandelt werden, während z.B. für Java spezielle Bibliotheken runtergeladen werden und mit dem Bluetooth-Protokoll der genutzten Windows Versionen abgestimmt werden müssen.

Um den Einsatz auf verschiedenen PCs und Laptops zu ermöglichen, bietet die Software die Möglichkeit den genutzten seriellen COM-Port manuell zu wählen.

5.2.2.2 Darstellung der Daten

Ein wichtiger Punkt war die grafische Darstellung der Daten in Echtzeit. Die grafische Darstellung in Form von Graphen, führt zu einem intuitiven handlichen Umgang mit dem MEMS-Chip und bietet außerdem eine einfache und schnelle Möglichkeit die angezeigten Daten auf Plausibilität zu prüfen.

Für die grafische Darstellung wurde die Klassenbibliothek ZedGraph²⁷ benutzt. Sie stellt eine konfigurierbare Fensterfläche bereit. Die Daten werden als Listen gespeichert und anschließend geplottet.

²⁷ [29]

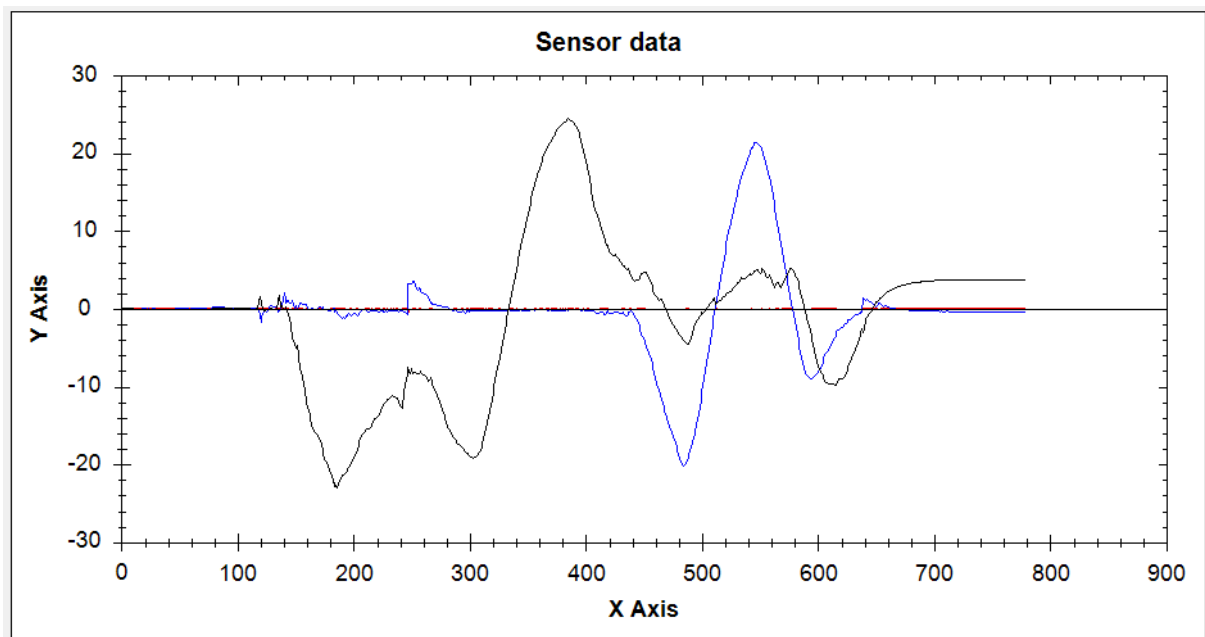


Abbildung 30 - ZedGraph

Zusätzlich zur grafischen Darstellung werden die Daten auch numerisch auf der Fensteroberfläche wiedergegeben.

5.2.2.3 *Sicherung der Daten*

Zur Sicherung der Daten wurde das verbreitete CSV-Format (Comma Separated Values) benutzt. Unter anderem unterstützt Microsoft Office dieses Format. Die Microsoft Office 2007 - Version wurde zur Weiterverarbeitung der empfangenen Daten genutzt. Die Daten können somit auch in Programmen wie Matlab, Octave oder LabView dargestellt und analysiert werden.

Das Loggen der Daten erfolgt nach Aktivierung durch ein Button-Element. Nach Aufnahme der gewünschten Daten, können diese in eine gewünschte Datei gespeichert werden.

5.3 Diagramm

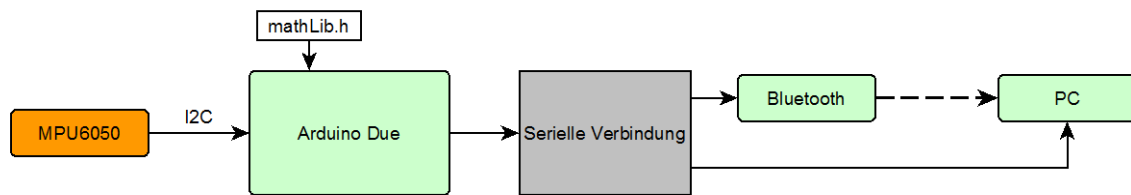


Abbildung 31 - Diagramm, Aufbau

Die Daten vom Mikroprozessor können sowohl per Kabelverbindung, als auch per Bluetooth an den PC gesendet werden.

5.4 Umsetzung

Der Arduino bietet keine Möglichkeit, mit frei erhältlicher Software, den Programmcode in Einzelschritten zu debuggen. Für die Entwicklung des Kalman-Filters, den Algorithmus und die benötigten mathematischen Funktionen (Quaternionen, Matrizen) wurde dies allerdings als notwendig erachtet.

Daher sind die Berechnungen, welche auf dem Mikroprozessor stattfinden sollen, zunächst mit C# im Visual Studio entwickelt worden. Die Daten der Beschleunigungsmesser und Gyroskope wurden unbearbeitet mit Hilfe des Arduino Due an den Computer geschickt.

5.4.1 Algorithmus und Kalman Filter

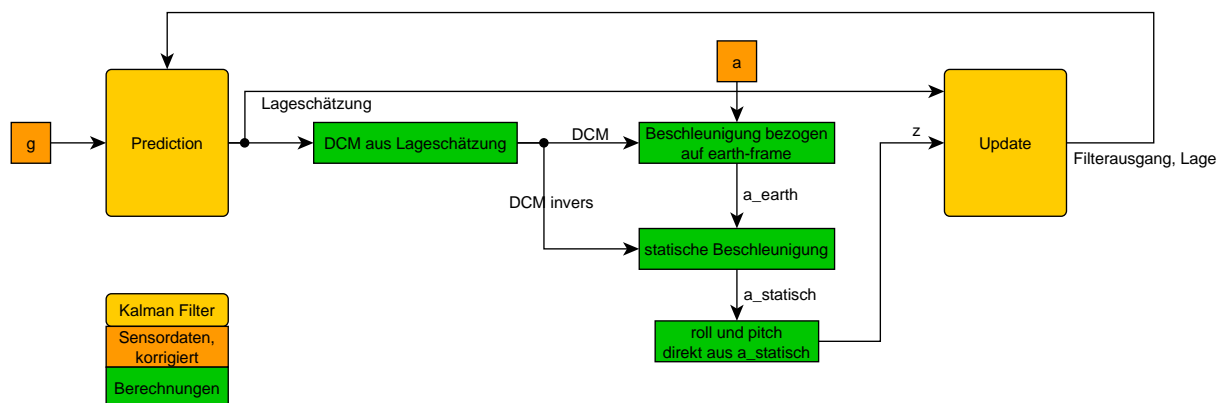


Abbildung 32 - Algorithmus und Kalman Filter

Der Prediction- und der Update-Schritt sind aufgespalten worden. Die Gyroskop-Werte g werden in eine Rotation umgerechnet $(q_\Delta)^{28}$ und als Eingabewert u in den Kalman-Filter gegeben. Die Systemfunktion

$$f(x, u) = q_{r,k}^- = x \cdot u = q_{r,k-1}^+ \cdot q_\Delta \quad (5-1)$$

schätzt die neue Orientierung²⁹ q_r . Die Multiplikation ist eine Quaternionenmultiplikation (s. Anhang). Die Jakobi-Matrix J_F kann im Prediction-Block als Funktion von u codiert werden.

$$J_F|_{x,u} = \begin{bmatrix} \frac{\partial f_1}{\partial q_{rot}} & \dots & \frac{\partial f_1}{\partial q_z} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial q_{rot}} & \dots & \frac{\partial f_4}{\partial q_z} \end{bmatrix} = \begin{bmatrix} q_{\Delta,rot} & -q_{\Delta,x} & -q_{\Delta,y} & -q_{\Delta,z} \\ q_{\Delta,x} & q_{\Delta,rot} & q_{\Delta,z} & -q_{\Delta,y} \\ q_{\Delta,y} & -q_{\Delta,z} & q_{\Delta,rot} & q_{\Delta,x} \\ q_{\Delta,z} & q_{\Delta,y} & -q_{\Delta,x} & q_{\Delta,rot} \end{bmatrix} \quad (5-2)$$

Anhand der Lageschätzung $q_{r,k}^-$ wird die DCM-Matrix berechnet (Anhang, DCM). Sie wird benötigt um die Beschleunigung a_b , welche körperbezogen ist, in das navigation-frame zu transferieren.

$$a_{n,k} = DCM_k \cdot a_k - G \quad (5-3)$$

Im navigation-frame wird der Gravitationsvektor, also die statische Beschleunigung, abgezogen. Möchte man eine Positionsbestimmung durchführen, würde die Integration von a_n zu der Geschwindigkeit v_n und ihr Integral zur Position p_n .

Um bei einem sich bewegenden Objekt die Lage über die Beschleunigungsmesser zu bestimmen, muss die dynamische Beschleunigung abgezogen werden.

Mit der inversen DCM-Matrix wird a_n , also die dynamische Beschleunigung wieder in das body-frame gerechnet und von a abgezogen.

$$a_{stat,k} = a_k - DCM_k^{-1} a_{n,k} \quad (5-4)$$

Mit folgenden Formeln³⁰ kann nun über die Beschleunigung a_{stat} der Roll- und Pitch-Winkel bestimmt werden:

$$roll_{stat,k} = \arctan\left(\frac{a_{stat,y}}{a_{stat,z}}\right) \quad (5-5)$$

²⁸ Das Formelzeichen q steht für die Darstellung als Quaternion

²⁹ Für den Orientierungsvektor wird in der Literatur häufig das Formelzeichen r benutzt.

³⁰ [30]

$$pitch_{stat,k} = \arctan\left(-\frac{a_{stat,x}}{\sqrt{a_{stat,y}^2 + a_{stat,z}^2}}\right) \quad (5-6)$$

Diese Lageangaben werden als Quaternion $z_k = q_{z,k}$ an den Update-Schritt gegeben. Der Yaw-Winkel wird mit $yaw = 0$ hardgecodet. Zur Bestimmung des Yaw bedarf es weitere Bezugspunkte. Bei inertialen Navigationssystemen wird bei der Initialisierung der geographische Norden bestimmt. Das ist mit den Gyroskopen des MPU6050 nicht möglich (s. Kapitel 6.1.1).

Im Update-Block wird, wie in Kapitel 4.1.3 beschrieben, der Kalman-Gain berechnet und die korrigierte Lage ausgegeben.

Die benötigten mathematischen Funktionen für Matrizen- und Quaternionenmathematik wurden in einer Bibliothek *mathLib.h* programmiert und dem Programm hinzugefügt.

5.4.1.1 Filterung auf Computer

Da die Vorbereitungen zum Kalman-Filter in Visual Studio gemacht worden sind, bot es sich an eine Projektversion zu haben, bei der die Filterung tatsächlich in der Software stattfindet anhand der Rohdaten vom MPU6050. Dies hat den Vorteil, dass Parameter während der Laufzeit geändert werden können und die Auswirkung direkt an den Graphen erkennbar ist.

Dafür wurde die Oberfläche um ein paar Elemente erweitert (Abbildung 33)

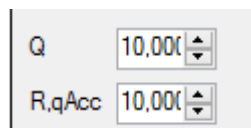


Abbildung 33 - Bedienelemente für Filterparameter

Dabei handelt es sich um die Diagonalen der Matrizen Q und R .

6 Tests

Um zu bestimmen welche Tests sinnvoll und möglich sind, um die Lagebestimmung zu testen, musste zunächst überlegt werden, welche Bewegungen der MEMS-Chip macht und welche Störungen es dieser gibt.

Bewegung:

- Ruhelage
- Bewegung im 3D-Raum
- Rotation
- Beschleunigung

Außerdem die Kombination aller Punkte.

Störungen, Anomalien:

- Temperaturänderung
- Höhenänderung (Luftdruck)
- Vibration
- Stoß

Es können hauptsächlich Tests in Ruhelage durchgeführt werden. Dazu gehören das Ermitteln der Offsets, der Einfluss der Temperatur und die Lagebestimmung. Das Verhalten bei Bewegung ist mit den vorhandenen Mittel nur schwer nachvollziehbar, da der Testapparat nur per Hand geführt werden kann und somit keine definierte Bewegung möglich ist, auf die sich bezogen werden kann.

6.1 Offsets

Als ersten Schritt wird der MEMS-Chips kalibriert. Das bedeutet, es wird eine Lage definiert und anhand dieser, die Offsets der Beschleunigungsmessern und Gyroskopen ermittelt.

6.1.1 Offset Gyroskope

Die Gyroskope messen die Drehrate des Chips. Um die Offsets zu ermitteln, liegt es somit nahe, den Chip in Ruhelage zu bringen und die gemessenen Werte zu mitteln.

Einflüsse auf den Offset

Bei der Nutzung eines Gyroskops auf der Erde, muss die Erdrotation beachtet werden. Diese wird im WGS84 mit $\omega = 72,92115 \cdot 10^{-11} \text{ RAD}/\text{s}$ ($= 4,178 \cdot 10^{-3} \text{ }^\circ/\text{s}$) definiert.

Die höchste Genauigkeit der Gyroskope im MPU6050 wird mit der Skalierung $\pm 250 \text{ }^\circ/\text{s}$ erreicht.³¹ Bei einer Bitanzahl von 16, beträgt die Genauigkeit $\frac{250 \text{ }^\circ/\text{s}}{2^{15}} = 7.629 \cdot 10^{-3} \frac{\text{ }^\circ/\text{s}}{\text{LSB}}$. Die Erdrotation kann somit nicht vom MPU6050 erfasst werden und muss bei der Bestimmung des Offsets nicht berücksichtigt werden³². In inertialen Navigationssystemen, z.B. in Flugkörpern, wird mit Hilfe der Gyroskope der geographische Nordpol bestimmt, indem das inertielle Navigationssystem lange genug ruhig gehalten wird um die Erdrotation zu messen.

Einen weiteren Einfluss auf den Offset, hat die Temperatur. Sie beeinflusst das Volumen der Messkörper, zum Beispiel der Piezoelemente (Kapitel 2.5), und führt dazu, dass sich die Offsets ändern. Dazu die Messung in Abbildung 34 und Abbildung 35.

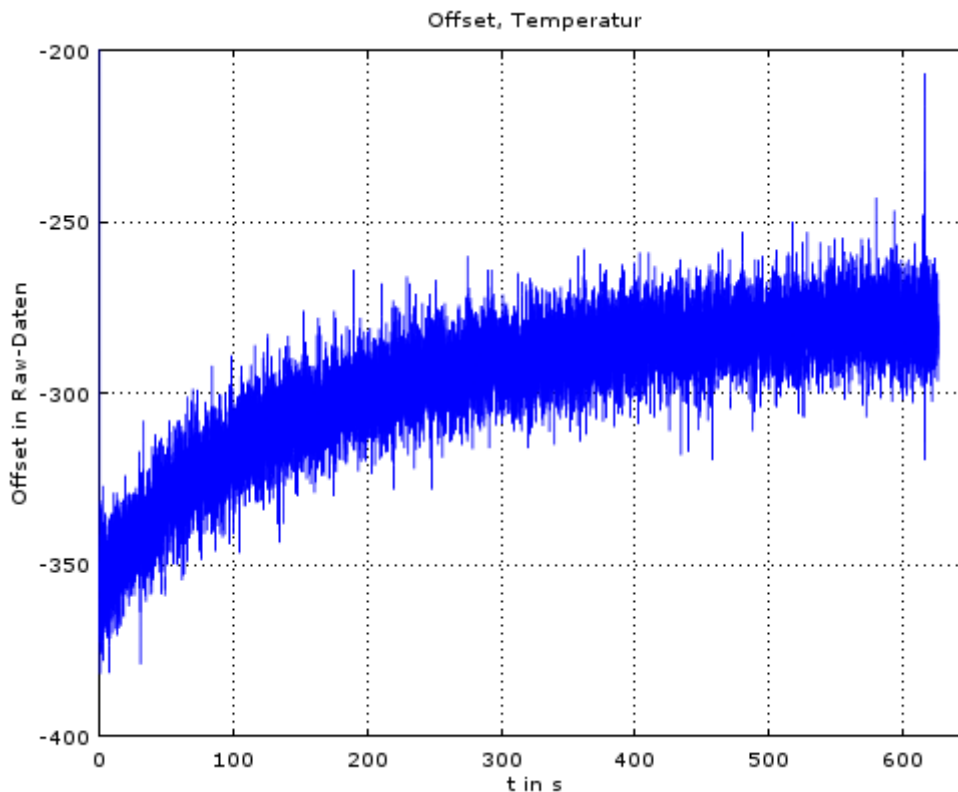


Abbildung 34 - Messung des Offsets vom Gyroskop, während Abkühlung

³¹ [18]

³² Die Erdrotation kann mit dem MPU6050 jedoch nachgewiesen werden [20]

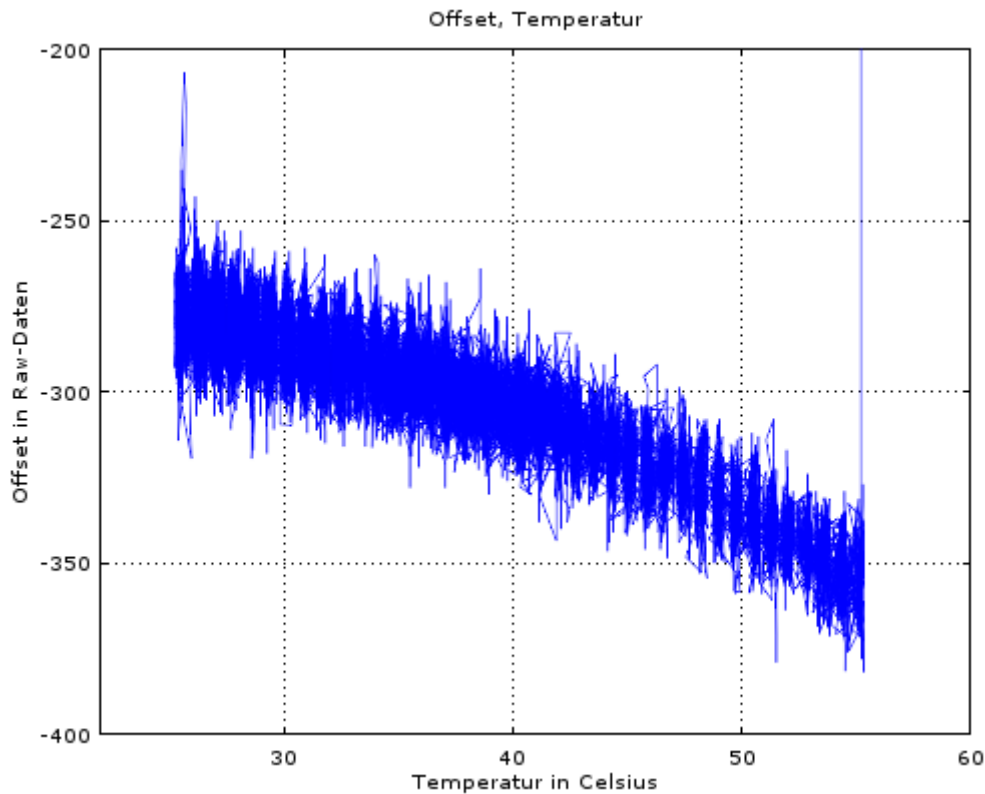


Abbildung 35 - Gyroskop-Offset über Temperatur

Mit Hilfe eines Heißluftgerätes wurde der MPU6050 auf eine Temperatur von etwa 55° Celsius erhitzt. Während des Abkühlvorgangs, bis auf etwa 23° Celsius, wurde der Ausgang des Gyroskops zur x-Achse gemessen. In Abbildung 34 erkennt man, dass die Werte bei 55° Celsius etwa -350 ($-350\text{LSB} \cdot 7.629 \cdot 10^{-3} \frac{\text{°/s}}{\text{LSB}} = -2,67 \text{°/s}$) liegen und bis auf -275 ($-275\text{LSB} \cdot 7.629 \cdot 10^{-3} \frac{\text{°/s}}{\text{LSB}} = -2,1 \text{°/s}$) steigen. Bei einem Temperaturunterschied von 32° Celsius ergibt sich eine Differenz von $0,57 \text{°/s}$ (ca. $0,018 \frac{\text{°/s}}{\text{K}}$)

In Abbildung 35 wurden noch einmal die Raw-Werte in Abhängigkeit zur Temperatur dargestellt.³³ Die Raw-Werte sind bis etwa 42° Celsius linear abhängig von der Temperatur. Danach steigt die Änderungsrate negativ an. Eine Korrigierung der Temperatur ist bis etwa 45° Celsius möglich. In einer Umgebung mit höheren Temperaturen, muss das Verhalten gegenüber der Temperatur genauer bestimmt werden.

³³ Wegen den verrauschten Daten, nicht sauber dargestellt. Die Tendenz ist jedoch erkennbar.

6.1.2 Offset Beschleunigungsmesser

Die Bestimmung der Offsets der Beschleunigungsmesser erweist sich als nicht trivial. Mit der Festlegung der Offsets wird auch das Referenzsystem definiert. In diesem Fall die Kalibrierungslage als Referenzsystem (n-frame).

Für ein Referenzsystem gilt, dass die Achsen zueinander orthogonal sind. Bei falsch bestimmten Offsets wird das Referenzsystem verzerrt.

Zur Bestimmung der Offsets muss die Lage des Chips bekannt sein. Es bietet sich an, den Chip so auszurichten, dass die z-Achse, exakt zum Erdmittelpunkt zeigt (also 9.81m/s^2 misst) und die x- und y-Achsen orthogonal zu ihr ausgerichtet sind und somit keine Beschleunigung messen.

Die Ausrichtung des Chips erfolgte mit Hilfe einer Kamerawasserwaage (Abbildung 36). Die Beschleunigungsmesser haben bei einer Skalierung von $\pm 2g$ und einer Bitzahl von 16 eine Genauigkeit von $\frac{2 \cdot 9.81\text{m/s}^2}{2^{15}} = 598,75 \cdot 10^{-6} \frac{\text{m/s}^2}{\text{LSB}}$. Da die Ausrichtung des Chips per Hand ungenau ist, muss davon ausgegangen werden, dass die Offsets nicht optimal gemessen werden können.

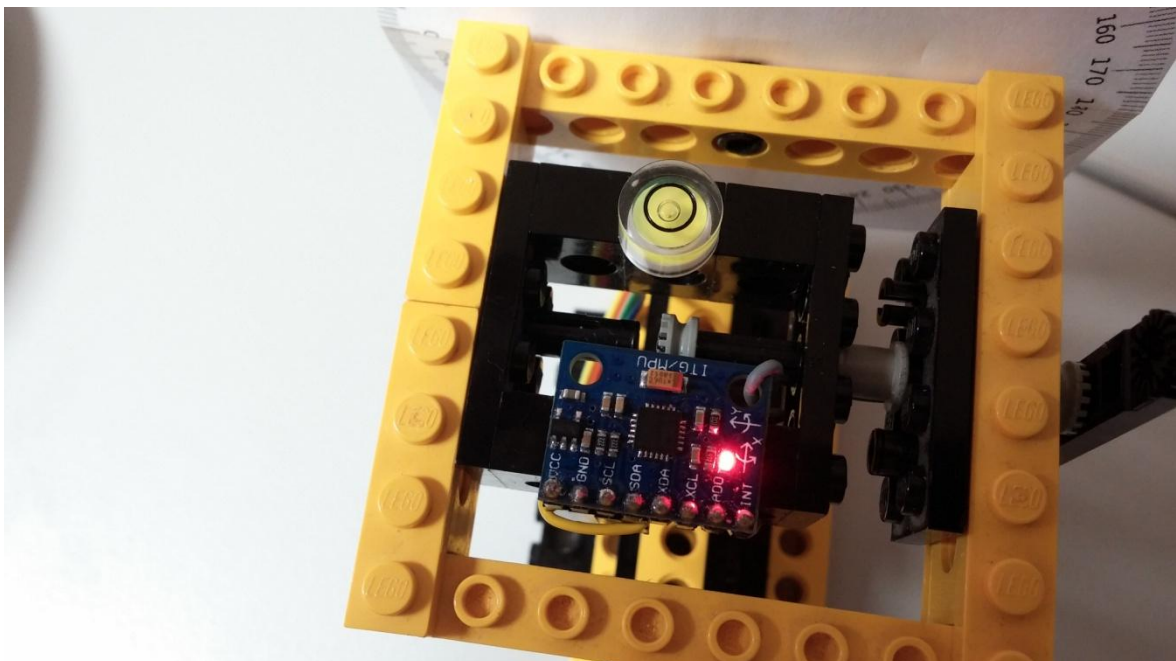


Abbildung 36 - Kamerawasserwaage zum Leveln des MPU6050

Die oben erwähnte Methode lässt sich noch etwas erweitern. Um bessere Ergebnisse zu bekommen, werden die Offsets an verschiedenen Punkten der Kennlinie gemessen. Dafür

bieten sich an die Ausrichtung einer Achse Richtung Erdmittelpunkt, die Ausrichtung exakt senkrecht zum Erdmittelpunkt und die Ausrichtung entgegen des Erdmittelpunkts. Es kann davon ausgegangen werden, dass die Kennlinie nicht linear ist.

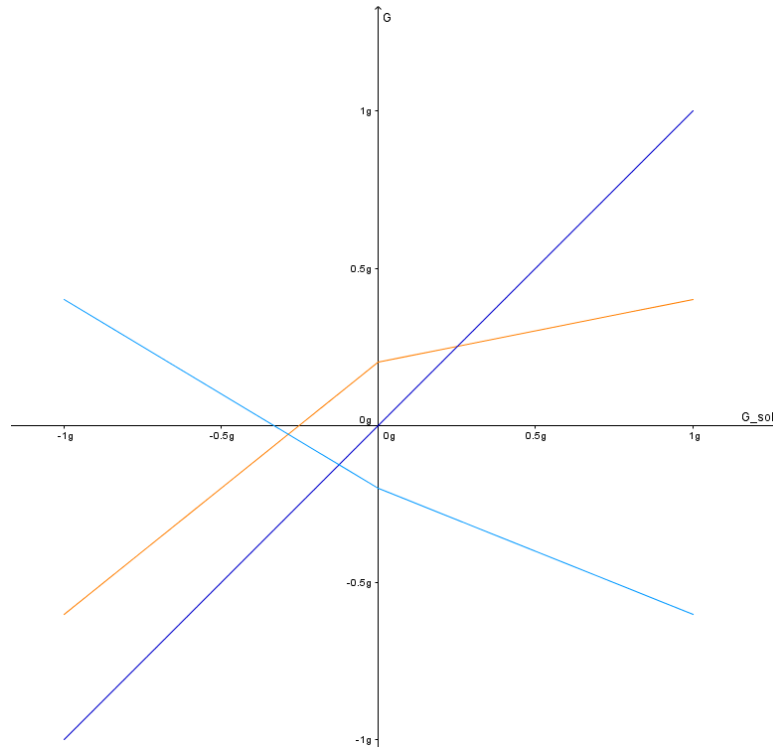


Abbildung 37 - genauere Offsetkorrektur

Anhand dieser Messpunkte ist es möglich die Korrigierung des Offsets im positiven und negativen Bereich anzupassen. Abbildung 37 soll dies noch einmal verdeutlichen. Der blaue Graph gibt die korrekte Beschleunigung in einem Winkel zur Erdgravitation von -180° bis 0° . Der orange Graph zeigt die gemessenen Werte, bzw. wird aus den drei Punkten $g(-180^\circ)$, $g(90^\circ)$ und $g(0^\circ)$ ermittelt (in der Abbildung die Sollwerte auf die x-Achse eingetragen). Der türkise Graph, ist die Differenz der beiden. Die Messwerte können nun mit der Differenz korrigiert werden. Je mehr Punkte bestimmt werden können, desto genauer kann der Offset eliminiert werden.

Einfluss Temperatur

Auf Abbildung 38 und Abbildung 39 wurde der Einfluss der Temperatur auf den Offset von Beschleunigungsmesser zur x-Achse.

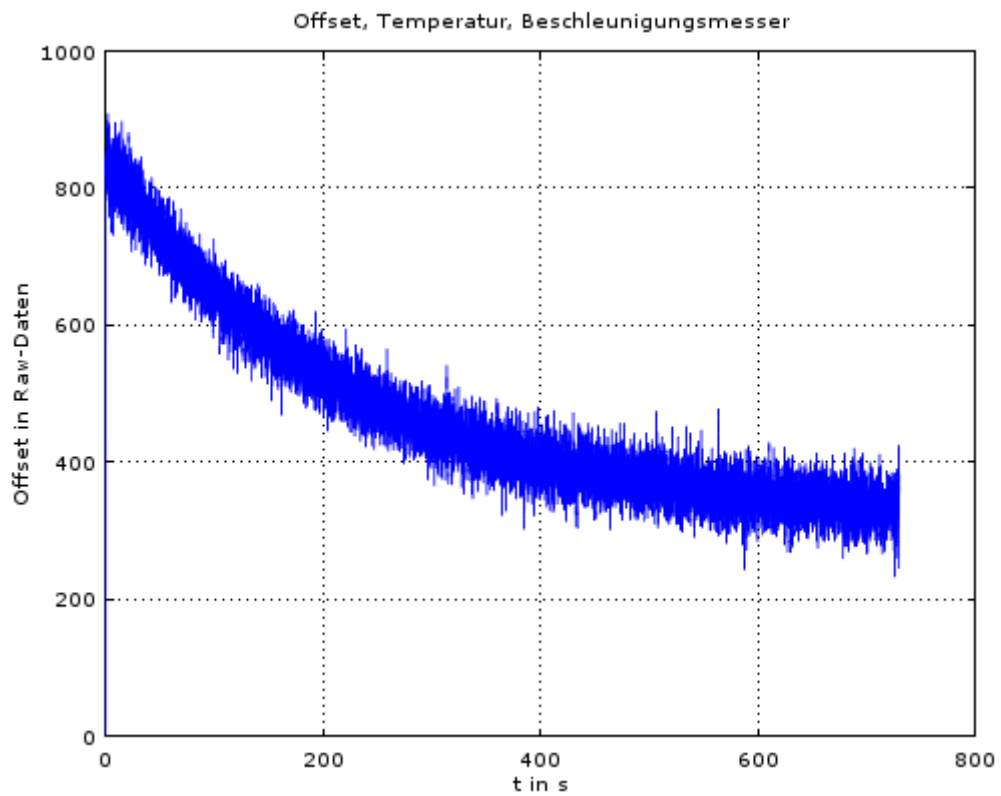


Abbildung 38 - Messung des Offsets vom Beschleunigungsmesser, während Abkühlung

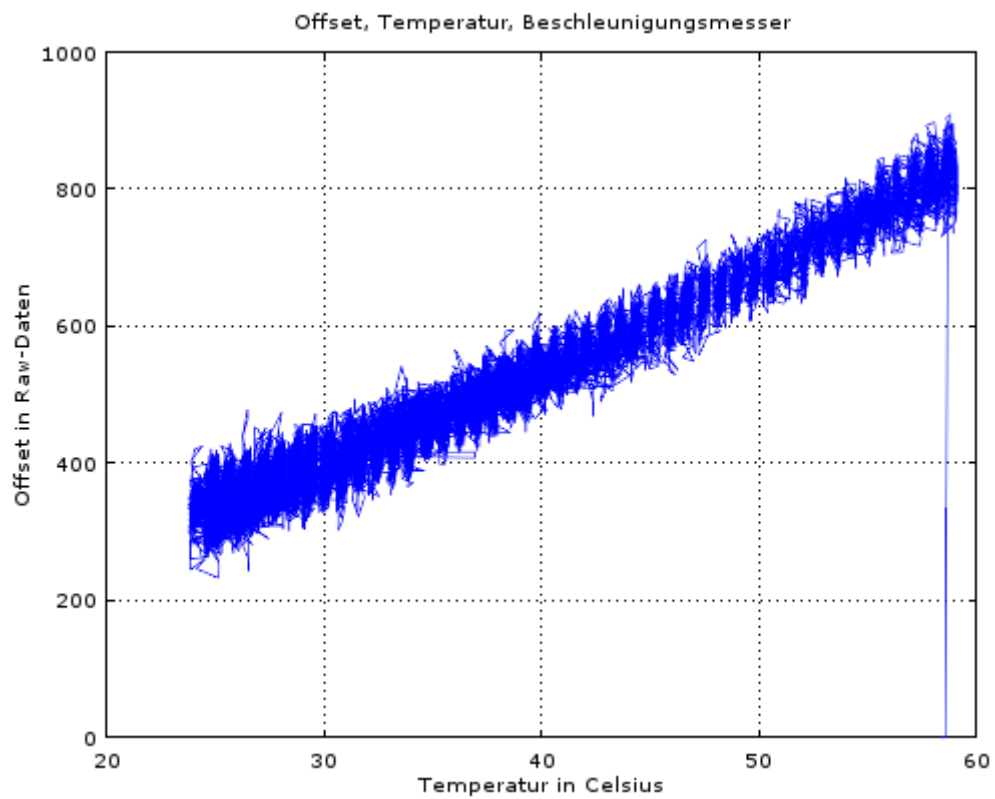


Abbildung 39 - Beschleunigungsmesser-Offset über Temperatur

Der MPU6050 wurde mit einem Heißluftgerät bis auf etwa 58° Celsius erhitzt. Der Ausgangswerte des Beschleunigungsmessers zur x-Achse wurden aufgenommen. Die x-Achse wurde senkrecht zum Gravitationsvektor ausgerichtet.

Bei einer Temperatur von 58° Celsius hatte der Offset einen Wert von etwa 850LSB ($850\text{LSB} \cdot 598,75 \cdot 10^{-6} \frac{\text{m/s}^2}{\text{LSB}} = 0,510\text{m/s}^2$) und sank bei 24° Celsius auf etwa 350 ($350\text{LSB} \cdot 598,75 \cdot 10^{-6} \frac{\text{m/s}^2}{\text{LSB}} = 0,210\text{m/s}^2$). Das entspricht ca $8,824 \frac{\text{m/s}^2}{\text{K}}$.

Auch hier ist die Abhängigkeit der Ausgangswerte vom Beschleunigungsmesser nicht ganz linear. Kann aber, wie beim Gyroskop, bis ca. 40°-45° Celsius als linear angenommen werden und korrigiert werden.

6.2 Lagebestimmung

Rollwinkel von 30° wird per Hand eingestellt:

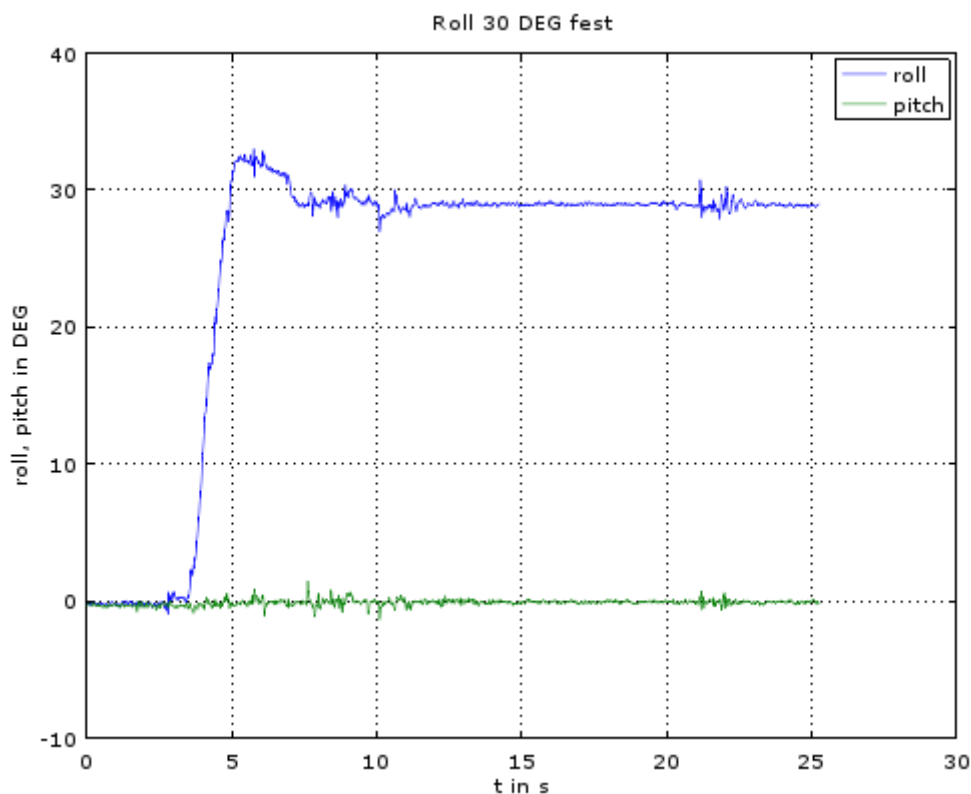


Abbildung 40 - Roll 30°, fest

Der Chip wird per Hand auf 30° gedreht. Der angezeigte Wert liegt bei Roll 28,93. Der relative Fehler liegt bei -3,57%.

Der Versuch wurde noch zweimal wiederholt:

	Angezeigt	Relativer Fehler
Versuch 1	■ Roll 28,93	-3,57%
Versuch 2	■ Roll 29,41	-1,97%
Versuch 3	■ Roll 29,13	-2,9%

Tabelle 3 - Ergebnisse, Rollwinkel 30°, fest

Der Fehler ist bei jedem Versuch negativ gewesen. Dies könnte an falschen Offsets liegen. Aus diesem Grund wurde der Chip noch einmal kalibriert und der Versuch noch einmal durchgeführt:

	Angezeigt	Relativer Fehler
Versuch 1	■ Roll 29,63	-1,2%
Versuch 2	■ Roll 28,92	-3,6%
Versuch 3	■ Roll 29,43	-1,9%

Tabelle 4 - Ergebnisse , Rollwinkel 30°, fest, neukalibriert

Auch dieses mal war der Fehler in derselben Größenordnung und ebenfalls negativ.

Das könnte zum einen am Testaufbau liegen. Zwar wurden Lego-Bausteine schon 1964 mit einer Genauigkeit von einem Tausendstel Millimeter hergestellt³⁴, dies bedeutet allerdings nicht, dass genutzte Konstruktion stabil genug ist.

Ein weiterer Grund ist, dass die Offsets der Beschleunigungsmesser über der Arbeitskennlinie nicht konstant sind (s. Kapitel Abbildung 376.1.2).

Rollwinkel von 45° wurde per Hand eingestellt:

³⁴ [28]

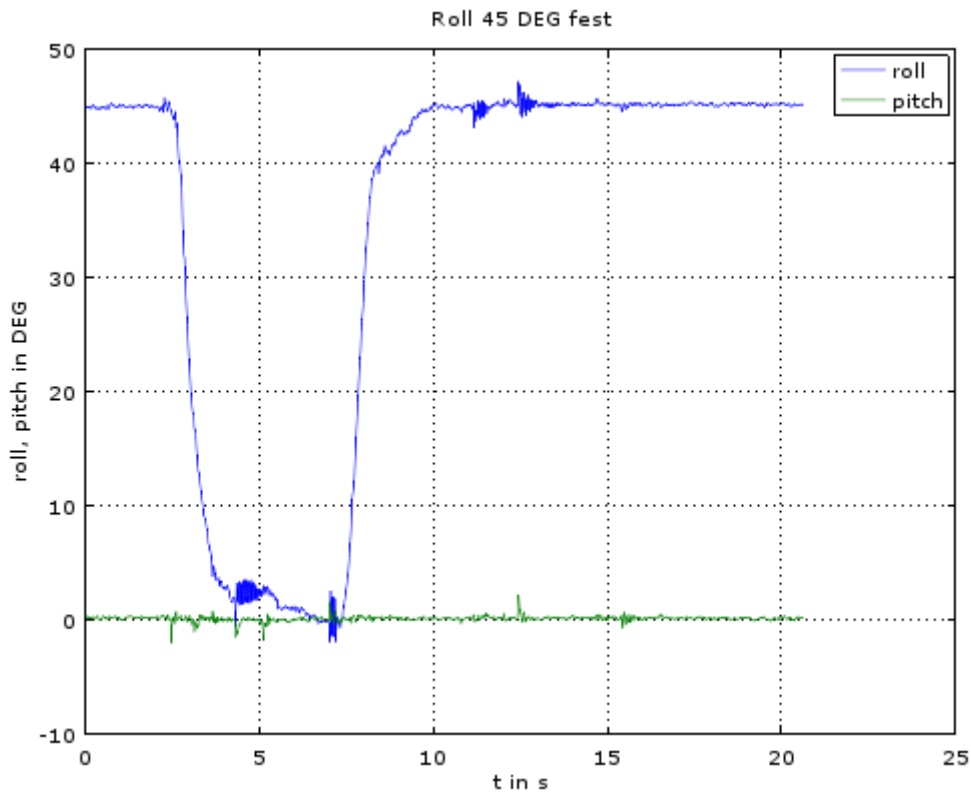


Abbildung 41 - Roll 45° fest

Ab etwa 7,5s wurde der Chip auf einen Rollwinkel von 45° gedreht. Die numerische Anzeige auf der Oberfläche zeigte für den Rollwinkel etwas weniger als 45° an ■ Roll 44.72. Der relative Fehler liegt somit bei etwa -0,6%. Es muss berücksichtigt werden, dass der Winkel per Hand nicht genau eingestellt werden kann. Auch hier wurden noch drei weitere Versuche gemacht um den Fehler zu rekonstruieren.

	Angezeigt	Relativer Fehler
Versuch 1	■ Roll 44,93	-0,16%
Versuch 2	■ Roll 45,12	0,27%
Versuch 3	■ Roll 45,14	0,31%

Tabelle 5 - Ergebnisse , Rollwinkel 45°, fest

Der Fehler bei 45° fällt geringer als bei 30° aus und ist im gleichen Maße positiv und negativ. Hier kann die Testumgebung und das manuelle Einstellen des Winkels eine Rolle spielen. Auch spielt, wie im letzten Versuch erwähnt, die Bestimmung der Offsets eine große Rolle.

7 Fazit und Ausblick

Das der MPU6050 im Amateurbereich weit verbreitet ist, lässt sich wohl mit dem einfachen (preisgünstigen) Zugang zu den Sensordaten erklären. Die Bestimmung der Lage (stationär) ist mit geringen, relativen Fehler verbunden. Er wird vor allem im Modellbau eingesetzt bei Quadrocopter, Modelautos oder in der Robotik.

Auch findet er Verwendung in einem HeadTracking-Projekt³⁵. Dies wurde vom Author, unabhängig von dieser Arbeit, ausprobiert. Die Ergebnisse dieser Arbeit spiegeln die Erfahrung mit dem Headtracker wieder: es war nicht möglich eine Orientierung dauerhaft zu halten. Dies wurde vor allem schon bei kleinen Bewegungen deutlich. Dynamische Beschleunigungen haben einen großen Einfluss auf die Lagebestimmung, da diese vorher ermittelt und abgezogen werden müssen.

Um genauere Untersuchungen durchführen zu können, sind geeignete Testmittel unabdingbar. Eine Redundanz der Hardware könnte aber auch nützlich sein. So ist es mit dem eingesetzten Bluetooth-Modul, laut Datenblatt, möglich bis zu 100m zu senden. Damit ergibt sich die Möglichkeit einen Versuchsaufbau von einem Hochhaus in die Tiefe fallen zu lassen. Bei einem Gehäuse mit niedrigem Luftwiderstand könnte so, die gleichmäßige Beschleunigung auf den Chip untersucht werden. Diese Idee ist daran angelehnt, dass man bei Erdbohrungen Module in das Bohrloch fallen lassen kann und anhand der Daten des Chips Tiefe und Geradlinigkeit der Bohrung feststellt.

Eine Verbesserung der Leistung, wäre die redundante Nutzung von weiteren Chips. Das führt zu höherer Genauigkeit aber auch zu einer höheren Prozessor- und Speicheranforderung. Eine höhere Genauigkeit fordert auch die Beachtung der Erdrotation und Corioliskraft.

Ein weiterer Punkt zu Verbesserung der Daten, ist der Kalman-Filter. In dieser Arbeit wurden die Lage, dargestellt als Quaternion, als Zustandsvektor genutzt. Häufig wird der Systemfehler in den Zustandsvektor übernommen (sog. indirektes Modellieren, Complementary Kalman Filter). Dadurch verringert sich der Modellierungsfehler. Außerdem kann der Fehler oft linear beschrieben werden, wodurch der Einsatz eines nicht-linearen Kalman-Filters entfällt.

³⁵ [31]

8 Quellen

- [1]. **Arduino.** www.arduino.cc. [Online] 16. 01 2016.
<https://www.arduino.cc/en/Products/Compare>.
- [2]. **Wagner, Jörg F. und Trierenberg, Andor.** *The History of Theoretical, Material and Computational Mechanics - Mathematics Meets Mechanics and Engineering.* s.l. : Springer Berlin Heidelberg, 2014. 978-3-642-39904-6.
- [3]. **InvenSense.** *MPU-6000 and MPU-6050 Register Map and Description.* Revision 4.0, Sunnyvale, Californien, USA : s.n., 03. 09 2012.
- [4]. **Jones, Steven E. und Thiruvathukal, George K.** *Codename Revolution: The Nintendo Wii platform.* Cambridge, Massachusetts : The MIT Press, 2012.
- [5]. **Boeker, P., et al.** Methodik und Technik der Online-Geruchsmessung. *Gefahrstoffe - Reinhaltung der Luft.* Juli/August 2003, Nr. 7/8, S. 284.
- [6]. **Woodman, Oliver J.** *An introduction to inertial navigation.* Cambridge, Cambridge, United Kingdom : s.n., August 2007.
- [7]. **Titterton, David und John, Weston L.** *Strapdown Inertial Navigation Technology.* United Kingdom, Herts : 2004: The Institution of Electrical Engineers, 2005.
- [8]. inertialnavigations.blogspot.de. [Online] 2012. [Zitat vom: 24. 10 2015.]
<http://inertialnavigations.blogspot.de/>.
- [9]. **Kanth, Krishna, Kumar, Sandeep und Kumar, Ravi.** FPGA Based Real time systems for position tracking. *International Journal of Engineering Trends and Technology.* April 2013, Volume4Issue4, S. 857.
- [10]. www.de.wikipedia.org. [Online] [Zitat vom: 25. Oktober 2015.]
<https://de.wikipedia.org/wiki/Kreiselinstrument>.
- [11]. <http://hyperphysics.phy-astr.gsu.edu>. [Online] [Zitat vom: 29. 10 2015.]
<http://hyperphysics.phy-astr.gsu.edu/hbase/gyr.html>.
-

- [12]. www.mpoweruk.com. [Online] [Zitat vom: 29. 10 2015.]
<http://www.mpoweruk.com/figs/gyros.htm>.
- [13]. *Laser gyroscope will guide passenger jets*. **Owen, Robert**. Januar 1982, Popular Science, S. 34.
- [14]. How It Works - The Vibrating Gyro- (Science And Stuff). [Online] 24. 03 2012.
[Zitat vom: 31. 10 2015.] <https://www.youtube.com/watch?v=zwe6LEYF0j8>.
- [15]. **Graham, Brian Barkley**. *Using an Accelerometer Sensor to Measure Human Hand Motion*. Massachusetts, USA : s.n., 11. Mai 2000.
- [16]. **Schmidt, Wolf-Dieter**. *Sensorschaltungstechnik*. 3. s.l. : Vogel Buchverlag, 2007.
- [17]. **Eckert, Peter**. www.sascha.uni-saarland.de. [Online] 2006. [Zitat vom: 02. 11 2015.]
<http://www.sascha.uni-saarland.de/sascha2006/de/sensoren/beschleunigungssensor.html>.
- [18]. **InvenSense**. *MPU-6000/Mpu6050 Product Specification*. Sunnyvale, Californian, U.S.A. : s.n., 08 19, 2013.
- [19]. **National Imagery And Mapping Agency**. *World Geodetic System 1984*. St.Louis, Missouri, U.S.A : s.n., 3. Januar 2000.
- [20]. St Kwan's Home for the Terminally ADD. [Online] [Zitat vom: 13. 01 2016.]
<http://stkwans.blogspot.de/2012/11/detection-of-rotation-of-earth.html>.
- [21]. **MathsPoetry, User:.** wikipedia.org. [Online] 14. Februar 2009. [Zitat vom: 14. 01 2016.] https://en.wikipedia.org/wiki/Gimbal_lock#/media/File:No_gimbal_lock.png.
- [22]. **Papula, Lothar**. *Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler*. Wiesbaden : Vieweg Verlag, 2006.
- [23]. **McElhoe, Bruce A**. *An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus*. s.l. : Aerospace and Electronic Systems, 1966.
- [24]. **Gerald L. Smith, Stanley F. Schmidt, Leonard A. McGee**. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. Californien, Moffett Field : Ames Research Center, 1962.
-

<Quellen

[25]. www.en.wikipedia.org. [Online] [Zitat vom: 18. 01 2016.]

https://en.wikipedia.org/wiki/Piezoelectric_accelerometer#/media/File:PiezoAccelTheory.gif.

[26]. www.euclideanspace.com. [Online] [Zitat vom: 18. 01 2016.]

<http://www.euclideanspace.com/math/algebra/realNormedAlgebra/quaternions/geometric/axisAngle/axisAngle1.png>.

[27]. **Balzer, Paul**. www.cbcity.de. [Online] [Zitat vom: 16. 11 2015.]

<http://www.cbcity.de/das-kalman-filter-einfach-erklaert-teil-1>.

[28]. **Spiegel Magazin**. www.spiegel.de. [Online] 16. 12 1964. [Zitat vom: 19. 01 2016.]

<http://magazin.spiegel.de/EpubDelivery/spiegel/pdf/46176691>.

[29]. zedgraph.sourceforge.net/. [Online] [Zitat vom: 05. 12 2015.]

<http://zedgraph.sourceforge.net/>.

Anhang

Weiterer Anhang auf beigefügter CD.

Quaternionen

$$q = q_0 + iq_1 + jq_2 + kq_3$$

Addition

$$x + y = (x_0 + y_0) + i(x_1 + y_1) + j(x_2 + y_2) + k(x_3 + y_3)$$

Multiplikation

$$\begin{aligned} x \cdot y &= (x_0y_0 - x_1y_1 - x_2y_2 - x_3y_3) + i(x_0y_1 + x_1y_0 + x_2y_3 - x_3y_2) \\ &\quad + j(x_0y_2 - x_1y_3 + x_2y_0 + x_3y_1) + k(x_0y_3 + x_1y_2 - x_2y_1 + x_3y_0) \end{aligned}$$

Konjugation

$$\bar{x} = x_0 - ix_1 - jx_2 - kx_3$$

Betrag

$$|x| = \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}$$

Quaternionen zu Winkel

$$\alpha = \arctan\left(\frac{q_0q_1 + q_2q_3}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right)$$

$$\beta = \arcsin\left(\frac{2(q_1q_3 - q_0q_2)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right)$$

$$\gamma = \arctan\left(\frac{2(q_0q_3 + q_1q_2)}{q_1^2 - q_2^2 - q_3^2 + q_0^2}\right)$$

Quaternion aus Winkel

$$c_\alpha = \cos\left(\frac{\alpha}{2}\right) \qquad s_\alpha = \sin\left(\frac{\alpha}{2}\right)$$

$$c_\beta = \cos\left(\frac{\beta}{2}\right) \qquad s_\beta = \sin\left(\frac{\beta}{2}\right)$$

$$c_\gamma = \cos\left(\frac{\gamma}{2}\right) \qquad s_\gamma = \sin\left(\frac{\gamma}{2}\right)$$

$$q_0 = c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma$$

$$q_1 = s_\alpha c_\beta c_\gamma - c_\alpha s_\beta c_\gamma$$

$$q_2 = c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma$$

$$q_3 = -s_\alpha s_\beta c_\gamma + c_\alpha c_\beta s_\gamma$$

Direct Cosine Matrix (DCM) aus Quaternionen

$$DCM = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_0^2 & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 - q_3 q_0) & -q_1^2 + q_2^2 - q_3^2 + q_0^2 & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & -q_1^2 - q_2^2 + q_3^2 + q_0^2 \end{bmatrix}$$

Eigenständigkeitserklärung

Papadopoulos, Dimitrios

Matrikelnr.:1979815

Hiermit erkläre ich, dass ich die vorliegende Arbeit bzw. Leistung eigenständig, ohne fremde Hilfe und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe. Alle sinngemäß und wörtlich übernommenen Textstellen aus der Literatur bzw. dem Internet habe ich als solche kenntlich gemacht.

Hamburg, den 20.01.2016

Unterschrift