



INSTITUT FÜR INFORMATIK VII
ROBOTIK UND TELEMATIK

Bachelorarbeit

Attitude determination mittels einer inertialen Messeinheit (IMU)

Janis Nicolas Kaltenthaler

29. August 2019

Erstgutachter: Prof. Dr. Andreas Nüchter

Betreuer: Dr. Dorit Borrmann, MSc. Helge Lauterbach

Zusammenfassung

In der heutigen Zeit hat autonome Fortbewegung eine hohe Relevanz. Diese wird mit großer Wahrscheinlichkeit in Zukunft noch deutlich steigen. Unterschiedlichste mobile Roboter und viele weitere Fahrzeuge nutzen diese Technologie bereits. Für jedes sich autonom fortbewegende System ist die Information über seine aktuelle Position, Geschwindigkeit und Orientierung für die Berechnung seiner nachfolgenden Bewegungen essentiell.

Es existieren verschiedene Möglichkeiten, um diese benötigten Informationen zu erhalten. Die Position eines bewegten Systems kann zum Beispiel durch Laserscanner mithilfe von Lichttechnologien oder durch ein Global Navigation Satellite System (GNSS) mithilfe von Funktechnologien bestimmt werden. Diese Technologien versagen jedoch in speziellen Situationen, zum Beispiel bei nicht ausreichendem Satelliten-Empfang oder in einem nicht zur Laserscanner-Ortung geeigneten Umfeld. In solchen Situationen ist es notwendig, auf eine Technologie zurückzugreifen, welche Positionsänderungen und Orientierungen auch in diesen berechnen kann. Eine vielversprechende Methode ist die Verwendung von Beschleunigungs-, Drehraten- und Magnetfeldinformationen, um die benötigten Daten zu erhalten. Diese Informationen sind in den meisten Situationen verfügbar. Hierbei macht es keinen Unterschied, abgesehen von der Magnetfeldmessung, ob die Werte inner- oder außerhalb von Gebäuden, unter oder über Wasser oder auch im Weltall erfasst werden.

Sensoren, welche diese Werte erfassen, existieren bereits und können im Miniaturformat hergestellt werden. Das ermöglicht eine platzsparende Implementierung in unterschiedlichsten Systemen.

Die vorliegende Arbeit behandelt unter anderem die Methode der Bewegungsbestimmung, basierend auf Beschleunigungs-, Drehraten- und Magnetfeldinformationen erfassenden Sensoren. Verschiedene Problematiken im Bezug auf die Nutzung dieser Sensorik werden ebenfalls untersucht.

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während meiner Bachelorarbeit unterstützt und motiviert haben:

- bei Prof. Dr. Andreas Nüchter für die Ermöglichung dieser Abschlussarbeit am Lehrstuhl für Robotik und Telematik
- bei Dr. Christian Herrmann für die Einweisung in den KUKA-Roboter und für Hilfestellungen bei dessen Bedienung
- bei MSc. Sven Jörissen für die Unterstützung bei der Entwicklung der CAD-Modelle der Halterungen und deren 3D-Druck

Einen besonderen Dank möchte ich an meine Betreuer Dr. Dorit Borrmann und MSc. Helge Lauterbach richten, welche mir mit Rat und Tat jederzeit zur Seite standen, Fragen unterschiedlichster Art beantworteten und durch die offenen Gespräche, wertvollen Tipps und Hinweise maßgeblich zum Erfolg dieser Arbeit beigetragen haben.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	1
1.2	Aufbau der Arbeit	1
2	Grundlagen	3
2.1	Inertial Measurement Unit (IMU)	3
2.2	Roboter Operating System (ROS)	5
2.2.1	Grundgedanken	5
2.2.2	Aufbau	6
2.3	Madgwick-Filter	7
2.3.1	Grundgedanken	7
2.3.2	Mathematische Grundlagen	8
3	Systemaufbau, Hard- und Softwarekomponenten	13
3.1	Verwendete IMUs	13
3.2	Messwerterfassung	14
3.2.1	Raspberry Pi	14
3.2.2	Energieversorgung	14
3.2.3	Halterung	17
3.3	Simulationsumgebung	18
3.3.1	KUKA-Roboter	18
3.3.2	Volksbot-Roboter	18
3.4	Algorithmik	20
3.4.1	Allgemeiner Aufbau	20
3.4.2	Bestimmung Gravitation	21
3.4.3	Bestimmung Orientierung	21
3.4.4	Abziehen Gravitation	22
3.4.5	Bestimmung Offsetwerte	22
3.4.6	Bestimmung relative Geschwindigkeit und zurückgelegte Strecke	23
4	Messreihen und Optimierungen	25
4.1	Untersuchungen	25
4.1.1	Erkenntnisse Madgwick-Filter	25

4.1.2	Erkenntnisse Beschleunigungssensor	29
4.2	Messtrajektorien	43
4.2.1	KUKA-Roboter	43
4.2.2	Volksbot-Roboter	47
5	Fazit	57
5.1	Ausblick	58

Kapitel 1

Einleitung

1.1 Aufgabenstellung

Die Aufgabenstellung im Rahmen dieser Bachelorarbeit war es, Messwerte von miniaturisierten Beschleunigungs- und Drehratensensoren zu untersuchen, um aus diesen Werten Informationen über Geschwindigkeiten und zurückgelegte Strecken verschiedener Bewegungen zu errechnen. Zusätzlich soll eine Betrachtung der Güte dieser Resultate im Vergleich zu den tatsächlichen Trajektorien erfolgen.

Zu Testzwecken standen hierfür drei Sensoren zweier unterschiedlicher Hersteller und zwei Robotersysteme für die Simulation verschiedener Trajektorien zur Verfügung.

Herausforderungen waren unter anderem die präzise Bestimmung der absoluten Orientierung der Sensoren zur Gravitation. Weiterhin war bekannt, dass bei dieser Art von Sensoren mit größeren Messungenauigkeiten zu rechnen ist.

1.2 Aufbau der Arbeit

In Kapitel 2 wird grundlegend auf die Messtechnik der verwendeten Sensoren, auf das verwendete Roboter-Betriebssystem und auf einen zur Orientierungsbestimmung genutzten Filter-Algorithmus eingegangen.

Kapitel 3 stellt die verwendeten Hardwarekomponenten, das Prinzip der Messwernerfassung und die für Testzwecke genutzte Simulationsumgebung vor. Zudem beschreibt es die im Rahmen dieser Arbeit entwickelte Algorithmik.

Die durchgeführten Messreihen der Sensorik und Algorithmik, deren Auswertung und mögliche Optimierungen werden in Kapitel 4 vorgestellt.

In Kapitel 5 wird ein abschließendes Fazit und ein Ausblick der erlangten Erkenntnisse gegeben.

Kapitel 2

Grundlagen

2.1 Inertial Measurement Unit (IMU)

Als Inertial Measurement Unit (IMU) bezeichnet man die Kombination mehrere Inertialsensoren. Ein Intertialsensor kann dabei z.B. ein Beschleunigungssensor, ein Drehratensensor oder ein Magnetfeldsensor sein. Die IMUs, welche in dieser Arbeit genutzt werden, stellen eine Kombination der genannten Sensoren dar. Dabei besitzen sie für jede Sensorgattung drei orthogonal aufeinander stehende Sensoren, um für alle drei Raumrichtungen translatorische und rotationale Bewegungen, bzw. die Magnetfeldstärke aufzeichnen zu können [29, S. 4][6].

Im Folgenden soll das Funktionsprinzip der drei Sensorgattungen kurz erläutert werden:

Beschleunigungssensor

Der Beschleunigungssensor misst in alle drei Raumrichtungen die auf ihn wirkende Beschleunigung. Das zweite Newtonsche Gesetz $\mathbf{F} = m \cdot \mathbf{a}$ bildet dabei die Grundlage der Beschleunigungsmessung. Es wird die auf eine Probemasse wirkende Trägheitskraft gemessen.

Realisiert ist dies in einem Microelectromechanical System (MEMS), einem miniaturisierten Beschleunigungssensor. Die Probemasse stellt zusammen mit einer Feder ein Feder-Masse-System dar, in welchem beide Komponenten aus Silizium gefertigt sind. Um die Auslenkung der Masse zu bestimmen, wird die Kapazitätsänderung zwischen der Probemasse und einer festen Bezugselektrode bei einer auftretenden Beschleunigung gemessen. Mit dem Wissen über die Auslenkung, die Federkonstante und die Masse wird nun die auf die Probemasse wirkende Kraft und daraus die Beschleunigung berechnet.

Der Sensor kann dabei nicht zwischen statischer und dynamischer Beschleunigung unterscheiden. Auf der Erde muss daher die Gravitation von den Messwerten abgezogen werden, um die dynamische Beschleunigung zu erhalten [29, S. 4ff.][3].

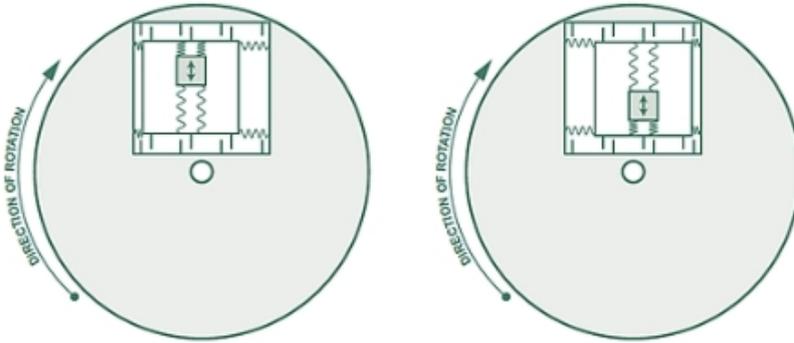


Abbildung 2.1: Funktionsweise Drehratensensor (Quelle: [8])

Drehratensensor

Der Drehratensensor misst die Rotationsgeschwindigkeit um alle drei Raumrichtungen. Auch dieser Sensor ist in einem MEMS realisiert. Das Messprinzip beruht dabei auf dem Corioliseffekt, wobei die Coriolisbeschleunigung gemessen wird.

Für jede der drei Rotationsrichtungen existiert ein gefederter Rahmen, welcher sich in Richtung der Rotation bewegen kann. Innerhalb der Rahmen ist eine Probemasse befestigt, welche orthogonal zur Rotationsbewegung schwingt. In Abbildung 2.1 sind dieser Sensoraufbau und das Messprinzip dargestellt.

Bewegt sich die Probemasse in Richtung Außenseite der Rotation, erfährt diese eine Corioliskraft entgegengesetzt der Rotationsrichtung. Bewegt sie sich zum Mittelpunkt der Rotation hin, erfährt sie eine Corioliskraft in Rotationsrichtung. Diese Kräfte bewegen die Rahmen in Richtung, bzw. entgegen der Richtung der Rotationsbewegung. Diese Bewegung wird über kapazitive Messelemente aufgezeichnet und daraus die Rotationsgeschwindigkeit berechnet [29, S. 9ff.][8].

Magnetfeldsensor

Der Magnetfeldsensor misst in allen drei Raumrichtungen die magnetische Flussdichte, beziehungsweise das magnetische Feld der Erde. Die Himmelsrichtung wird über die relative Orientierung des Sensors zum Magnetfeld bestimmt.

Der Großteil der als MEMS realisierten Magnetometer nutzt den Hall-Effekt zur Bestimmung der magnetischen Flussdichte. Dazu lässt man elektrischen Strom durch eine leitende Metallplatte fließen. Die Elektronen fließen dabei auf direktem Weg durch die Platte. Wird die Platte von einem magnetischen Feld durchkreuzt, werden die Elektronen, abhängig von der Stärke und der Richtung des Magnetfeldes, auf eine Seite der Platte abgelenkt. Aufgrund des resultierenden Elektronenüberschusses auf einer der Plattenseiten, wird eine Spannung gemessen. Diese ist direkt abhängig von der Stärke des Magnetfeldes [9][7].

2.2 Roboter Operating System (ROS)

2.2.1 Grundgedanken

Bei dem Roboter Operating System (ROS) handelt es sich um ein flexibles allgemeingültiges Framework für verschiedenste autonome Systeme. Es stellt unter anderem folgende Dienste zur Verfügung:

- Hardwareabstraktion
- Gerätetreiber
- Bibliotheken
- Visualisierungen
- Nachrichtenvermittlung
- Paketverwaltung

Ziel der im Jahre 2007 unter dem Namen *Switchyard* begonnenen Entwicklungsarbeiten ist die Vereinfachung komplexer und stabiler Aufgabenstellungen an verschiedene Robotersysteme.

Die Entwicklung einer Universal-Roboter-Software durch eine einzelne Person, bzw. eine Institution wäre mühevoll. Die Open-Source Software ROS wird seither von einer Gemeinschaft an Wissenschaftlern, Laboratorien und Institutionen entwickelt. Jeder kann sein Fachwissen in einem bestimmten Bereich einbringen.

Die fünf Grundgedanken, welchen die Design-Philosophie zugrunde liegt, sind:

Peer-to-Peer

Dadurch wird der Einsatz mehrerer Computer im selben Roboter-System ermöglicht. Zusätzlich können Daten zur rechenintensiven Verarbeitung an externe Computer übertragen werden.

Werkzeugbasiert

Zugunsten einer geringen Komplexität ist ROS im Mikrokerneldesign umgesetzt.

Mehrsprachig

ROS verfügt über Anbindungen für die Programmiersprachen C++, Python und Lisp. Zudem existieren unter anderem Module für JAVA, Haskell und Lua.

Schlank

Die Entwicklung von Algorithmen erfolgt unabhängig von konkreter Hardware als Bibliothek. Dies erleichtert die Portierung auf andere Hardwareplattformen und die Wiederverwendbarkeit der Algorithmen.

Open Source

ROS steht unter der BSD-Lizenz und kann daher sowohl für kommerzielle als auch für nicht kommerzielle Projekte ohne Einschränkungen zum Einsatz kommen.

[4][21][18][19][20]

2.2.2 Aufbau

Die Hauptaufgabe eines Roboter-Betriebssystem ist die gleichzeitige Abarbeitung verschiedener Aufträge. Die Informationen können hierbei je nach Anforderung synchron oder asynchron ausgetauscht werden.

Im Folgenden werden die einzelnen Bauteile des ROS-Systems vorgestellt:

Knoten

Ein Knoten, auch Node genannt, ist ein Prozess, welcher Berechnungen ausführt. Er kann sich hierbei auf einen Sensor, einen Motor, ein Verarbeitungsalgorithmus oder einen Überwachungsalgorithmus beziehen. Mehrere Knoten können parallel ablaufen und erklären sich jeweils dem Master.

Master

Der Master ist der Deklarations- und Speicherservice der einzelnen Knoten und ermöglicht es ihnen, sich untereinander zu kennen. Er hat folgende Aufgaben:

- Speicherung der Informationen, welche Knoten welche Themen publizieren, bzw. abonnieren
- Verwaltung der zur Verfügung gestellten Services
- Bereitstellung von Diensten für das Auffinden der Kommunikationspartner zur Laufzeit

Topics

ROS ermöglicht einen asynchronen Informationsaustausch durch Topics und synchronen Informationsaustausch durch Services. Ein Topic ist eine Informationsbewegung, basierend auf einem Subscribe/Publish System. Knoten senden und empfangen die Topic-Messages.

Messages

Eine Message besteht aus typisierten Einträgen, welche verschiedene Basisdatentypen, Felder oder Verschachtelungen sein können.

Bags

Mithilfe von Bags können sämtliche Nachrichten des ROS-Systems für eine spätere Verwendung oder Auswertung abgespeichert werden.

Die Möglichkeit, verschiedene Rechenprozesse auf einzelne Knoten aufteilen zu können, vereinfacht die Softwareentwicklung für komplexe autonome Systeme und ihrer Modellierung [4][23].

”Das ROS ist ein komfortables Framework, dessen Design-Philosophie speziell auf die Entwicklung wiederverwendbarer Software für autonome Systeme zugeschnitten ist.”[4]

2.3 Madgwick-Filter

In dieser Arbeit wird der von Sebastian O.H. Madgwick entwickelte Filteralgorithmus [30] verwendet und wird im Folgenden als Madgwick-Filter bezeichnet.

Zunächst soll ein Einblick in die grundlegende Funktionsweise des Filters vermittelt werden.

2.3.1 Grundgedanken

Für die Bestimmung räumlicher Bewegungen mittels IMU, bzw. Magnetic, Angular Rate, Gravity (MARG) Sensor, ist präzises Wissen über die aktuelle Orientierung notwendig. Sowohl für die Bestimmung der Richtung der Bewegung, als auch für die Abrechnung der Gravitation aus den Beschleunigungsmesswerten wird die exakte Orientierung des jeweiligen Sensors benötigt. Wird eine IMU zur Datenerfassung verwendet, existieren zwei Möglichkeiten zur Bestimmung der Orientierung. Verfügt man über einen MARG-Sensor, hat man zusätzlich einen Magnetfeldsensor.

Gyroskop

Das Gyroskop misst die Winkelgeschwindigkeiten der drei zueinander orthogonal ausgerichteten Achsen. Es ist nicht möglich, die absolute Orientierung mit einem Gyroskop zu bestimmen. Ist jedoch die Ausgangsorientierung bekannt, kann über die zeitliche Integration der einzelnen Winkelgeschwindigkeiten die absolute Orientierung bestimmt werden. Ansonsten erhält man eine relative rotationale Bewegung.

Messwerte jedes Gyroskops sind zu einem gewissen Grad fehlerbehaftet. Diese Messabweichungen akkumulieren bei der Integration und können mit der Zeit erhebliche Abweichungen verursachen.

Beschleunigungs- und Magnetfeldsensor

Mithilfe eines Beschleunigungs-, bzw. Magnetfeldsensors wird das Gravitations-, bzw. Magnetfeld der Erde gemessen.

Über die Messdaten dieser beiden Sensoren wird die absolute Orientierung bestimmt. Sobald jedoch zusätzlich zur Gravitation durch eine Bewegung eine Beschleunigung auftritt, verfälscht dies den Gravitationsvektor in Betrag und Richtung. Ebenso kann die Magnetfeldmessung durch andere Magnetfelder gestört werden. Durch diese und weitere Störeinflüsse kann die Bestimmung der Orientierung signifikant beeinflusst werden.

Die Aufgabe des Madgwick-Filters ist es, die Messungen von Gyroskop, Beschleunigungs- und Magnetfeldsensor optimal zu fusionieren, um eine möglichst genaue Schätzung der Orientierung zu erhalten. Optimal bedeutet in diesem Sinn die eben erwähnten Schwächen der einzelnen Sensoren zu kompensieren [30, S. 3].

Das Resultat ist in Bezug auf die Genauigkeit einem System mit Kalman-Filterung überlegen [30, S. 1].

2.3.2 Mathematische Grundlagen

In diesem Abschnitt wird grundlegend die Fusion der Sensordaten aus Gyroskop, Beschleunigungs- und Magnetfeldsensor erläutert. Der Zahlenbereich der Quaternionen stellt mitunter die mathematische Grundlage des Madgwick-Filters dar. Da in Kapitel 4 jedoch größtenteils mit Euler-Winkeln gearbeitet wird, werden beide Begriffe kurz eingeführt.

Quaternionen

Eine Quaternion ist eine vierdimensionale Zahl, ähnlich den komplexen Zahlen. Für die folgenden Berechnungen sind die Quaternionen vor allem nützlich, weil sich mit ihnen ein dreidimensionaler euklidischer Raum darstellen lässt. Darüber hinaus können Rotationen im dreidimensionalen Raum mit Quaternionen beschrieben werden.

Um die Berechnung von Quaternion-Rotationen zu vereinfachen, wird die Quaternion, welche die Rotation beschreibt, zunächst normalisiert.

Nun seien A und B dreidimensionale Koordinatensysteme, die gegeneinander um den Koordinatenursprung gedreht sind. Die entsprechende Drehachse, welche sich während der Rotation nicht ändert, sei durch ${}^A\hat{\mathbf{r}}$ mit den Komponenten r_x , r_y und r_z beschrieben. Um nun B aus A zu erhalten, muss A um ${}^A\hat{\mathbf{r}}$ mit dem Drehwinkel θ rotiert werden.

Die normalisierte Quaternion, welche die genannte Rotation beschreibt, ist wie folgt gegeben:

$${}^A_B\hat{\mathbf{q}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} & -r_x \sin \frac{\theta}{2} & -r_y \sin \frac{\theta}{2} & -r_z \sin \frac{\theta}{2} \end{bmatrix}$$

Der in A definierte Vektor ${}^A\mathbf{v}$ lässt sich mit ${}^A_B\hat{\mathbf{q}}$ zu ${}^B\mathbf{v}$ nach B rotieren. In diesem Fall fügt man dem Vektor ${}^A\mathbf{v}$ an erster Stelle eine 0 hinzu. Es gilt nun $\dim({}^A\mathbf{v}) = \dim({}^B\mathbf{v}) = 4$. Das Symbol \otimes bezeichnet hier das Quaternionenprodukt gemäß der Hamilton-Regeln und ${}^A_B\hat{\mathbf{q}}^*$ die Konjugation von ${}^A_B\hat{\mathbf{q}}$.

$${}^B\mathbf{v} = {}^A_B\hat{\mathbf{q}} \otimes {}^A\mathbf{v} \otimes {}^A_B\hat{\mathbf{q}}^* \quad (2.1)$$

[30, S. 4 ff.][13][26]

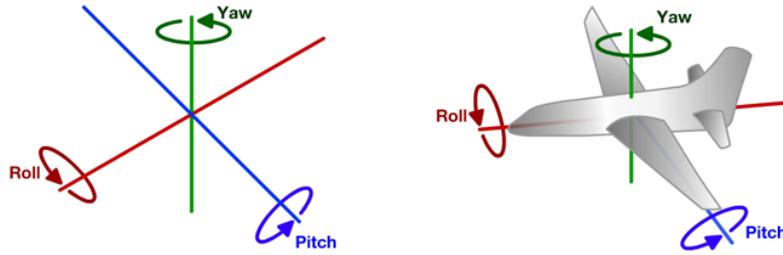


Abbildung 2.2: Definition der Euler-Winkel *Roll*, *Pitch* und *Yaw* (Quelle: [12])

Euler-Winkel

Orientierungsangaben in Euler-Winkeln sind im Vergleich zu Quaternionen besser vorstellbar. Mathematische Berechnungen, basierend auf Euler-Winkel, sind zudem weniger kompliziert.

Wie mit Quaternionen können Rotationen auch mit den drei Euler-Winkeln ψ , θ und ϕ beschrieben werden. Einheitsvektoren des Koordinatensystems B sind gegeben durch \hat{x}_B , \hat{y}_B und \hat{z}_B . Rotationen werden sequentiell durchgeführt. Es wird mit ψ um \hat{z}_B , mit θ um \hat{y}_B und zuletzt mit ϕ um \hat{x}_B rotiert.

In Abbildung 2.2 sind die Bedeutungen der geläufigen Begriffe *Pitch*, *Roll* und *Yaw* dargestellt. Dabei bezeichnet *Pitch* den Winkel θ , *Roll* den Winkel ϕ und *Yaw* den Winkel ψ . Diese Euler-Winkel repräsentieren dieselbe Rotation wie ${}^A_B\hat{q}$. Die Winkel lassen sich wie folgt aus der Quaternion berechnen:

$$\begin{aligned}\psi &= \arctan2(2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1) \\ \theta &= -\sin^{-1}(2q_2q_4 + 2q_1q_3) \\ \phi &= \arctan2(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1)\end{aligned}$$

Bei der Nutzung von Euler-Winkeln muss beachtet werden, dass es zu einem sogenannten Gimbal-Lock kommen kann, da die Lage einer Drehachse von den Drehungen der anderen beiden Drehachsen abhängig ist.

Ein Gimbal-Lock tritt bei $\theta = 90$ deg auf. In diesem Fall kann die Orientierung nicht mehr eindeutig durch Euler-Winkel angegeben werden. Eine Rotation mit ϕ oder ψ ist hierbei identisch [26][25].

Im Folgenden wird die Berechnung der Orientierung über Gyroskop, Beschleunigungs- und Magnetfeldsensor erläutert, anschließend die Fusion dieser Ergebnisse.

Hierbei sind S das Koordinatensystem des Sensors und E das Koordinatensystem der Erde.

Orientierung über Gyroskop

Das Gyroskop liefert mit ω_x , ω_y und ω_z die Winkelgeschwindigkeiten um die einzelnen Achsen von S . Der Vektor ${}^S\boldsymbol{\omega}_t$ mit $\dim({}^S\boldsymbol{\omega}_t) = 4$ nimmt diese Werte und 0 an erster Stelle für den Zeitpunkt t auf. Die Quaternion ${}^S_E\hat{\mathbf{q}}_{\omega,t}$ stellt die zeitliche Änderung der Orientierung des Erd-Koordinatensystems relativ zum Sensor-Koordinatensystem dar, ${}^S_E\hat{\mathbf{q}}_{\omega,t}$ die entsprechende Orientierung zum jeweiligen Zeitpunkt.

Mit der Definition der Quaternionen Differentiation aus [14] folgt:

$$\begin{aligned} {}^S\boldsymbol{\omega}_t &= \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix} \\ {}^S_E\dot{\mathbf{q}}_{\omega,t} &= \frac{1}{2} {}^S_E\hat{\mathbf{q}}_{\omega,t-1} \otimes {}^S\boldsymbol{\omega}_t \end{aligned}$$

Ist das Abtastintervall Δt des Sensors und die Anfangsorientierung gegeben, wird daraus die Orientierung des Erd-Koordinatensystems relativ zum Sensor-Koordinatensystem zum Zeitpunkt t berechnet [30, S. 6]:

$${}^S_E\mathbf{q}_{\omega,t} = {}^S_E\hat{\mathbf{q}}_{\omega,t-1} + {}^S_E\dot{\mathbf{q}}_{\omega,t}\Delta t$$

Orientierung über Beschleunigungs- und Magnetfeldsensor

Beschleunigungs- bzw. Magnetfeldsensoren liefern die Größenordnung und Richtung eines Gravitations- bzw. Magnetfeldes. Die gemessene Richtung des jeweiligen Feldes ist mit der normierten Quaternion ${}^S\hat{\mathbf{s}}$ im Sensor-Koordinatensystem S gegeben. Zusätzlich wird eine Bezugsorientierung des jeweiligen Feldes im Erd-Koordinatensystem mit ${}^E\hat{\mathbf{d}}$ definiert. Im Folgenden beschreibt ${}^S_E\hat{\mathbf{q}}$ die relative Orientierung zwischen Sensor und eben genannter Bezugsorientierung. Gleichung 2.2 beschreibt die in Gleichung 2.1 definierte Rotation von der Bezugsorientierung in das Sensor-Koordinatensystem.

$${}^S\hat{\mathbf{s}} = {}^S_E\hat{\mathbf{q}}^* \otimes {}^E\hat{\mathbf{d}} \otimes {}^S_E\hat{\mathbf{q}} \quad (2.2)$$

Ziel ist es, die Unbekannte ${}^S_E\hat{\mathbf{q}}$ zu bestimmen. Dafür werden das Minimierungsproblem in Gleichung 2.4 und die zugehörige Funktion in Gleichung 2.3 definiert. Die Komponenten der einzelnen Vektoren sind in den Gleichungen 2.5 bis 2.7 definiert.

$$\mathbf{f}({}^S_E\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) = {}^S_E\hat{\mathbf{q}}^* \otimes {}^E\hat{\mathbf{d}} \otimes {}^S_E\hat{\mathbf{q}} - {}^S\hat{\mathbf{s}} \quad (2.3)$$

$$\min_{{}^S_E\hat{\mathbf{q}} \in \mathbb{R}^4} \mathbf{f}({}^S_E\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) \quad (2.4)$$

$${}^S_E\hat{\mathbf{q}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \quad (2.5)$$

$${}^E\hat{\mathbf{d}} = \begin{bmatrix} 0 & d_x & d_y & d_z \end{bmatrix} \quad (2.6)$$

$${}^S\hat{\mathbf{s}} = \begin{bmatrix} 0 & s_x & s_y & s_z \end{bmatrix} \quad (2.7)$$

Um eine Lösung für die Gleichung 2.4 zu erhalten, wird das Gradientenverfahren gemäß [30, S. 7f.] angewendet.

Im Folgenden wird dieser allgemeine Lösungsansatz auf die Gravitations- und Magnetfeldmessung angewendet.

Gravitationsfeld

Um eine Lösung für die relative Orientierung des Sensors zur Erde zu finden, muss die Bezugsorientierung ${}^E\hat{\mathbf{d}}$ wie in Gleichung 2.8 definiert werden. Es wird dabei angenommen, dass dieser Vektor nur die Erdgravitation als Komponente beinhaltet.

$${}^E\hat{\mathbf{g}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Die Beschleunigungsmesswerte ${}^S\hat{\mathbf{s}}$ sind wie folgt gegeben und normiert.

$${}^S\hat{\mathbf{a}} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix}$$

Man erhält die Funktion $\mathbf{f}_g({}^S\hat{\mathbf{q}}, {}^S\hat{\mathbf{a}})$ gemäß Gleichung 2.3, welche nach dem Gradientenverfahren gelöst werden kann.

Magnetfeld

Das Magnetfeld der Erde weist je nach Standort eine gewisse Inklination auf. Es wird angenommen, dass die Bezugsorientierung ${}^E\hat{\mathbf{d}}$ Komponenten in der vertikalen und in einer horizontalen Achse gemäß Gleichung 2.9 aufweist.

$${}^E\hat{\mathbf{b}} = \begin{bmatrix} 0 & b_x & 0 & b_z \end{bmatrix} \quad (2.9)$$

Die Magnetfeldmesswerte ${}^S\hat{\mathbf{s}}$ sind wie folgt gegeben und normiert.

$${}^S\hat{\mathbf{m}} = \begin{bmatrix} 0 & m_x & m_y & m_z \end{bmatrix}$$

Man erhält die Funktion $\mathbf{f}_b({}^S\hat{\mathbf{q}}, {}^E\hat{\mathbf{b}}, {}^S\hat{\mathbf{m}})$ gemäß Gleichung 2.3, welche nach dem Gradientenverfahren gelöst werden kann.

Weder für $\mathbf{f}_g({}^S\hat{\mathbf{q}}, {}^S\hat{\mathbf{a}})$ noch für $\mathbf{f}_b({}^S\hat{\mathbf{q}}, {}^E\hat{\mathbf{b}}, {}^S\hat{\mathbf{m}})$ existiert eine eindeutige Lösung für ${}^S\hat{\mathbf{q}}$, da für beide Gleichungssysteme unendlich viele Lösungen existieren. Kombiniert man die beiden Funktionen in Gleichung 2.10, kann eine eindeutige Lösung für ${}^S\hat{\mathbf{q}}$ mithilfe des Gradientenverfahrens gefunden werden.

$$\mathbf{f}_{g,b}({}^S\hat{\mathbf{q}}, {}^S\hat{\mathbf{a}}, {}^E\hat{\mathbf{b}}, {}^S\hat{\mathbf{m}}) = \begin{bmatrix} \mathbf{f}_g({}^S\hat{\mathbf{q}}, {}^S\hat{\mathbf{a}}) \\ \mathbf{f}_b({}^S\hat{\mathbf{q}}, {}^E\hat{\mathbf{b}}, {}^S\hat{\mathbf{m}}) \end{bmatrix} \quad (2.10)$$

Als Ergebnis erhält man die Orientierung aus Beschleunigungs- und Magnetfeldmesswerten als ${}^S\mathbf{q}_{\nabla,t}$ für jeden Zeitpunkt t .

Fusion der Orientierungen

Wie in 2.3.1 bereits beschrieben, besitzt jeder Sensor seine eigenen Schwächen und Stärken. Um eine möglichst genaue Schätzung der aktuellen Orientierung ${}^S_E \mathbf{q}_{est,t}$ zu erhalten, können die beiden Orientierungsangaben wie folgt fusioniert werden:

$${}^S_E \mathbf{q}_{est,t} = \gamma_t {}^S_E \mathbf{q}_{\nabla,t} + (1 - \gamma_t) {}^S_E \mathbf{q}_{\omega,t} \text{ für } 0 \leq \gamma_t \leq 1$$

Die Schätzung der Orientierung kann weiterhin verbessert werden, indem man, wie in [30, S. 10f.] gezeigt, den optimalen Wert für γ_t bestimmt.

Kapitel 3

Systemaufbau, Hard- und Softwarekomponenten

3.1 Verwendete IMUs

Für diese Arbeit standen drei IMUs zur Verfügung. Zwei baugleiche Sensoren des Herstellers *Phidgets*, welche als Phidgets IMU 1 und 2 bezeichnet werden sollen. Für die Messungen und Untersuchungen wurde größtenteils die Phidgets IMU 1 verwendet.

Zusätzlich stand eine IMU der Marke *Xsens* zur Verfügung.

Beide IMU Modelle besitzen einen Beschleunigungs-, Drehraten- und Magnetfeldsensor mit jeweils drei Achsen, welche in einem MEMS realisiert sind. Die Funktionsweise dieser Sensorik wurde bereits in Abschnitt 2.1 erläutert.

Phidgets IMU

Die verwendeten Phidgets IMUs tragen die Modellbezeichnung *PhidgetSpatial Precision 3/3/3 High Resolution 1044_0* und arbeiten mit einer Datenaufzeichnungsrate von 250Hz . Sie sind preisgünstige Modelle und können für etwa \$140 [11] erworben werden.

Xsens IMU

Die verwendete Xsens IMU trägt die Modellbezeichnung *MTi-G-700* und arbeitet mit einer Datenaufzeichnungsrate von 100Hz . Sie ist ein höherpreisiges Modell und kann für etwa 3800€ [27] erworben werden.

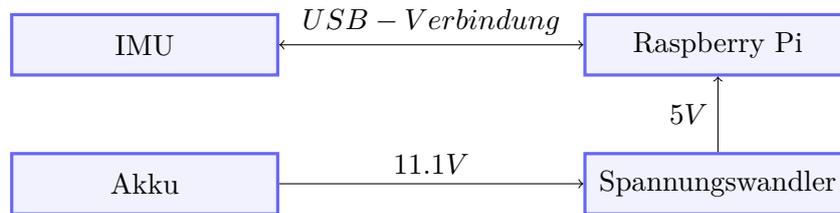


Abbildung 3.1: Blockschaltbild des Messaufbaus

3.2 Messwarterfassung

Um die IMUs für Testzwecke unabhängig von einer externen Stromversorgung und einer externen Rechereinheit betreiben zu können, wurde der in Abbildung 3.1 dargestellte Messaufbau genutzt.

3.2.1 Raspberry Pi

Als Recheneinheit kommt ein *Raspberry Pi 3 Model B+* zum Einsatz. Es besitzt einen 1.4GHz 64-bit Quad-Core Prozessor und verfügt über 1GB LPDDR2 SDRAM. Mithilfe des Dualband WLANs kann eine drahtlose Verbindung von einem externen Rechner zum Messaufbau hergestellt werden [15]. Des Weiteren kommt der geringe Stromverbrauch von maximal 7W der Akkulaufzeit des Messaufbaus zugute [17].

Ein vorgefertigtes und kompatibles Softwareimage kann von [16] bezogen werden. Das Image basiert auf Ubuntu 16.04 und besitzt die vorinstallierte ROS Version *Kinetic*.

Die benötigten Treiber für ROS zur Ansteuerung der in Abschnitt 3.1 genannten IMUs sind unter [28] und [10] zu finden.

Um den Madgwick-Filter unter ROS nutzen zu können, findet man unter [22] ein vorgefertigtes ROS-Package, welches den Filter implementiert.

3.2.2 Energieversorgung

Wie in Abbildung 3.1 dargestellt, besteht die Energieversorgung aus einem Akku und einem Spannungswandler, welcher die für das Raspberry Pi nötige Spannung von 5V erzeugt.

Es kommt ein Lithium-Polymer-Akku des Herstellers *Turnigy* zum Einsatz. Dieser erzeugt eine Spannung von 11.1V und verfügt über eine Kapazität von 1000mAh, mit welcher es möglich ist, den Messaufbau mindestens 30min mit Energie zu versorgen.

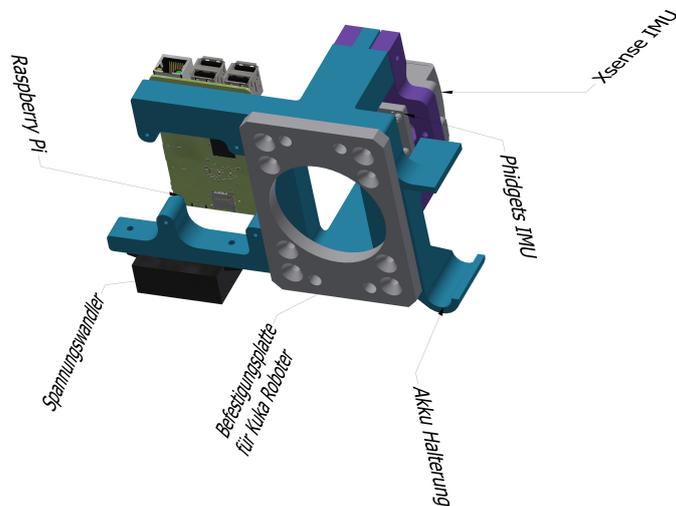


Abbildung 3.2: Übersicht Plattform Unterseite

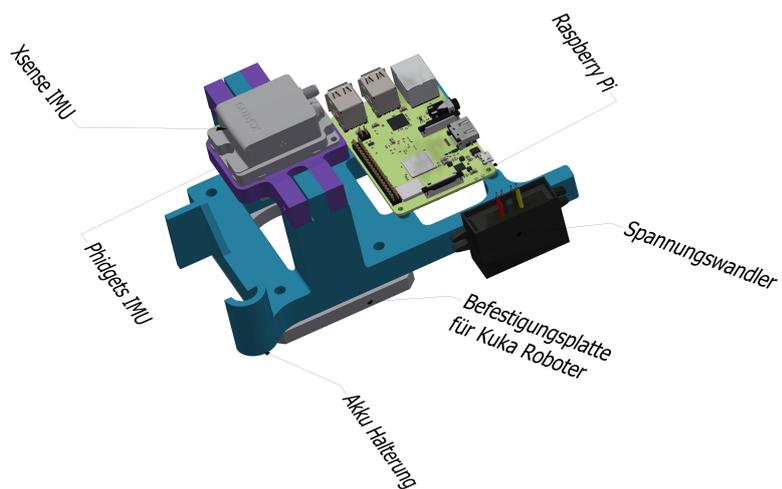


Abbildung 3.3: Übersicht Plattform Oberseite

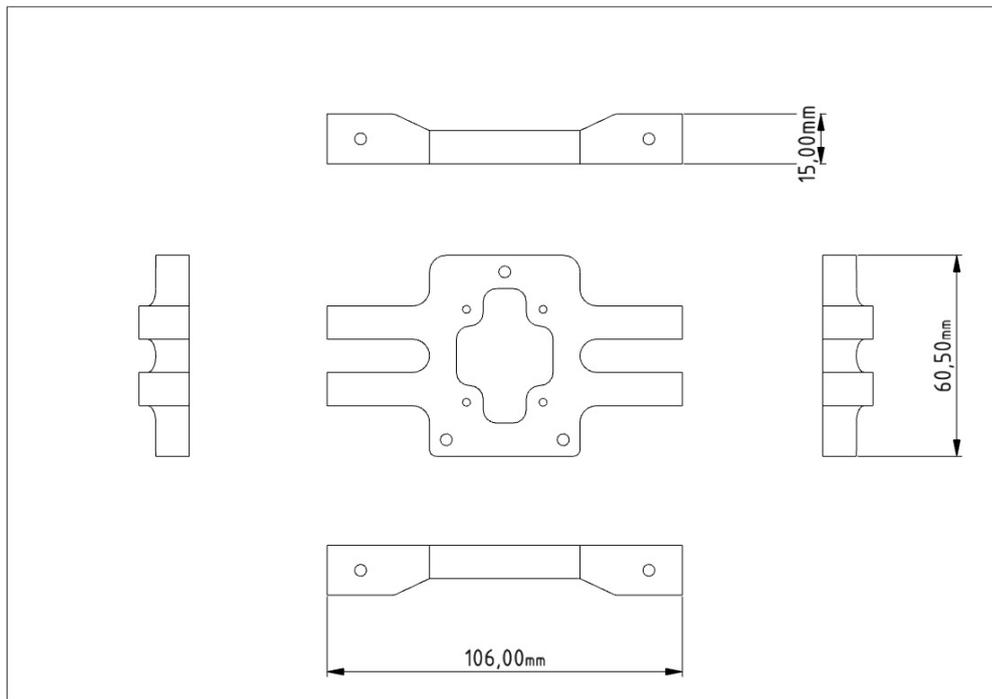


Abbildung 3.4: Konstruktionszeichnung des lilafarbenen Bauteils im Maßstab 2 : 1

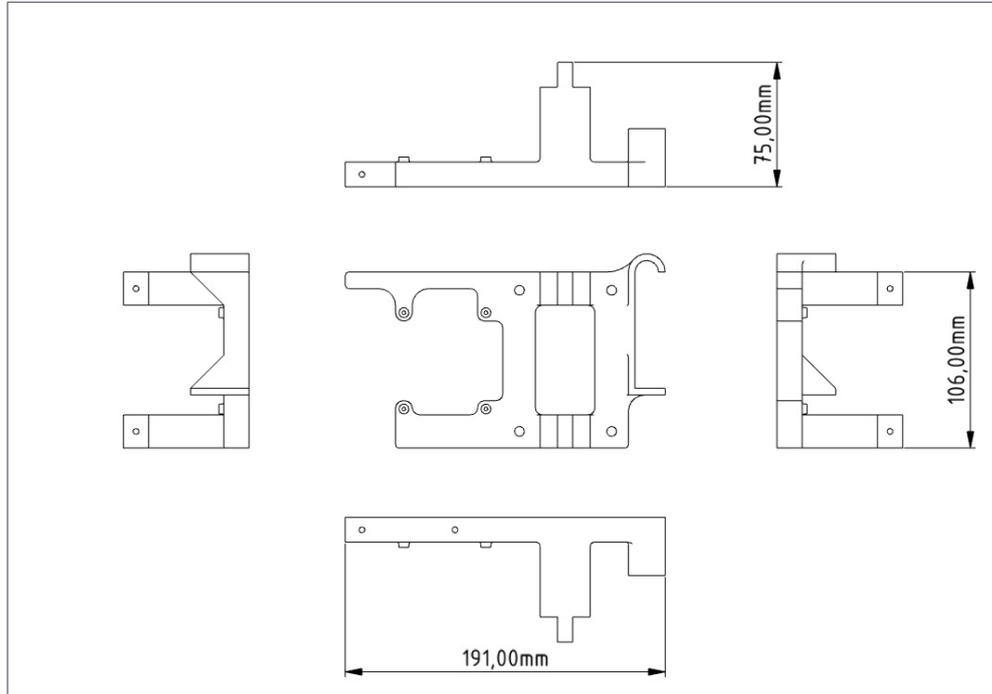


Abbildung 3.5: Konstruktionszeichnung des blauen Bauteils im Maßstab 1 : 1

3.2.3 Halterung

Um die zuvor genannten Bauteile sinnvoll auf einer Plattform zusammenfassen zu können, wurde die in den Abbildungen 3.2 und 3.3 sichtbare Konstruktion mithilfe einer 3D-Computer-Aided Design (CAD)-Software entworfen. Verwendet wurde dafür das von dem Unternehmen *Autodesk* entwickelte Programm *Inventor*.

Da die Bauteile mit einem 3D-Drucker gedruckt werden sollten, wurde die Konstruktion in zwei Bauteile aufgeteilt, um den Druck zu erleichtern und unnötige Stützstrukturen während des Druckes zu verhindern.

IMU-Halterung (Lilafarbenes Bauteil)

In der Konstruktionszeichnung 3.4 ist das lilafarbene Bauteil aus den Abbildungen 3.2 und 3.3 zu sehen. An diesem können sowohl die Phidgets IMU, als auch die Xsens IMU befestigt werden. Dabei hat man die Wahl, ob man beide IMUs gleichzeitig oder einzeln anbringen möchte.

Beim Entwurf wurde darauf geachtet, dass beide IMUs etwa dieselbe Position auf der Konstruktion einnehmen, damit die Messergebnisse besser vergleichbar sind.

Das Bauteil kann passgenau an dem zweiten Bauteil befestigt werden.

Basis-Halterung (Blaues Bauteil)

In der Konstruktionszeichnung 3.5 ist das blaue Bauteil aus den Abbildungen 3.2 und 3.3 zu sehen. Es nimmt das Raspberry Pi, den Spannungswandler und einen Akku auf. An seiner Unterseite besitzt es Bohrungen, um die gesamte Konstruktion an dem in Abschnitt 3.3.1 beschriebenen KUKA-Roboter befestigen zu können.

Um die Auswertung der Trajektoriedaten zu vereinfachen, wurde darauf geachtet, dass die beiden IMUs etwa auf der Verlängerungsachse des Handgelenks des Roboters liegen.



Abbildung 3.6: KUKA-Roboter (Quelle: [1])

3.3 Simulationsumgebung

Für die Ausübung unterschiedlicher Bewegungsmuster auf die in Abschnitt 3.1 vorgestellten IMUs standen die folgenden Roboter-Systeme zur Verfügung.

3.3.1 KUKA-Roboter

Um möglichst exakte Bewegungen zu simulieren, wurde ein 6-Achs-Gelenkarmroboter verwendet. Der in Abbildung 3.6 dargestellte Industrieroboter des Herstellers *KUKA* mit der Modellbezeichnung *KR16* verfügt über 6 Rotationsachsen, mit welchen es möglich ist, die Orientierung des am Greifer befestigten Messaufbaus während einer Bewegung konstant zu halten, bzw. vordefinierte Orientierungsänderungen exakt auszuführen.

3.3.2 Volksbot-Roboter

Um realistische Beschleunigungswerte während einer Bewegung zu untersuchen, wurde der in Abbildung 3.7 dargestellte dreirädrige Roboter vom Typ *Volksbot* verwendet. Die Motoren der beiden Vorderräder können gesondert angesteuert werden und werden von einem EPOS2 Motor Controller geregelt.

Ein nach vorne gerichteter Laserscanner vom Typ SICK LMS100 ermöglicht die Erfassung von horizontalen Entfernungsdaten.



Abbildung 3.7: Volksbot-Roboter

Die beiden folgenden Möglichkeiten zur Bestimmung der Position des Roboters stehen zur Verfügung:

Positionsbestimmung über Odometrie

Odometrie ist ein Verfahren zur Berechnung der Position eines Fahrzeugs, das sich mithilfe von Rädern fortbewegt (hier der Roboter). Ist der Umfang beider antreibender Vorderräder und deren Winkelgeschwindigkeit bekannt, lässt sich daraus die zurückgelegte Strecke nach einer bestimmten Zeit und so auch die neue Position des Roboters berechnen.

Der Nachteil bei ausschließlicher Verwendung der Odometrie zur Positionsbestimmung ist jedoch die hohe Ungenauigkeit durch kleinste äußere Einflüsse. So können die Berechnungen durch ungleichmäßig aufgepumpte Reifen, Schlupf aufgrund zu großer Beschleunigungen, Unebenheiten des Bodens oder ungleiche Belastung der Reifen verfälscht werden. Diese kleinen Fehler akkumulieren und sorgen letztendlich für eine falsche Positionsangabe des Roboters.

Positionsbestimmung über AMCL

Mithilfe von GMapping, eines Algorithmus zur Kartierung von Gebäuden mittels Laserscannern [5], wurde die Umgebung kartiert, in welcher sich der Roboter für die Durchführung der Tests befand.

Über die erstellte Umgebungskarte kann sich der Roboter innerhalb dieser Karte lokalisieren und daraus die aktuelle Position errechnen. Für die Lokalisierung wird der Adaptive Monte Carlo Localization (AMCL) Algorithmus verwendet [2].

3.4 Algorithmik

Die entwickelten Algorithmen wurden in der Programmiersprache *C++* geschrieben und als ROS-Knoten, siehe Abschnitt 2.2.2, implementiert und ausgeführt.

3.4.1 Allgemeiner Aufbau

Der Grundgedanke des im Rahmen der Arbeit erstellten Programms ist ein Zustandsautomat. Ein Algorithmus prüft ständig, ob sich das Gerät im **Stillstand** befindet oder eine **Bewegung** ausgeübt wird.

Stillstand

Der Algorithmus hält kontinuierlich die Maxima und Minima der Messwerte aller drei Beschleunigungssensoren aufrecht. Die Gravitation wurde bereits abgezogen, wie in 3.4.4 beschrieben. Für jede Messung wird anschließend geprüft, ob ein bestimmter Grenzwert überschritten wurde:

$$\begin{aligned} |a_{x,max} - a_{x,min}| &> \epsilon_1 \\ |a_{y,max} - a_{y,min}| &> \epsilon_1 \\ |a_{z,max} - a_{z,min}| &> \epsilon_1 \end{aligned}$$

Trifft **eine** dieser Bedingungen zu, so interpretiert der Algorithmus dies als Beginn einer Bewegung. Mithilfe von ϵ_1 lässt sich die Empfindlichkeit der Erkennung von Bewegungen festlegen und dem jeweiligen Anwendungszweck anpassen. Solange keine Bewegung erkannt wird, werden sämtliche n Beschleunigungsmesswerte abgespeichert, um diese später für die Kalibrierung zu verwenden.

Bewegung

Sobald eine Bewegung erkannt wird, wechselt der Algorithmus in den Zustand Bewegung. Der Algorithmus prüft kontinuierlich, ob ein Stillstand des Gerätes vorliegt. Dazu werden stets die letzten n Beschleunigungsmesswerte gesichert. Für jeden übergebenen Datensatz werden unter den n letzten Werten für jede Sensor-Achse das Minimum und das Maximum gesucht. Anschließend wird geprüft, ob alle n Werte innerhalb eines Toleranzbandes liegen:

$$\begin{aligned} |a_{x,max} - a_{x,min}| &< \epsilon_2 \\ |a_{y,max} - a_{y,min}| &< \epsilon_2 \\ |a_{z,max} - a_{z,min}| &< \epsilon_2 \end{aligned}$$

Treffen **alle** diese Bedingungen zu, befinden sich alle n Werte innerhalb des durch ϵ_2 definierten Toleranzbandes. Der Algorithmus interpretiert dies als Ende der Bewegungsphase und wechselt

wieder zurück in den Zustand **Stillstand**. Über die Anzahl der n letzten Beschleunigungsmesswerte wird die Dauer bis zur Stillstanderkennung festgelegt. Solange kein Stillstand erkannt wird, werden sämtliche Daten abgespeichert, um diese später für die Berechnung der Geschwindigkeit und der zurückgelegten Strecke zu verwenden.

3.4.2 Bestimmung Gravitation

Bei jedem Wechsel vom Zustand **Stillstand** in den Zustand **Bewegung** wird die Gravitation aus der in Abschnitt 3.4.1 beschriebenen Datenaufzeichnung berechnet. Die Menge der während der Stillstandphase aufgezeichneten Datensätze sei N und t_n die jeweiligen Abtastzeitpunkte.

$$\begin{aligned}\bar{a}_x &= \frac{1}{N} \cdot \sum_{n=0}^N a_x(t_n) \\ \bar{a}_y &= \frac{1}{N} \cdot \sum_{n=0}^N a_y(t_n) \\ \bar{a}_z &= \frac{1}{N} \cdot \sum_{n=0}^N a_z(t_n)\end{aligned}$$

Um Fehlmessungen herauszumitteln zu können, wird die Gravitation nur bestimmt, falls die vorangegangene Stillstandphase mindestens eine Sekunde gedauert hat.

Der Betrag des Gravitationsvektors berechnet sich wie folgt:

$$\begin{aligned}\vec{g} &= \begin{pmatrix} \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{pmatrix} \\ |\vec{g}| &= \sqrt{\bar{a}_x^2 + \bar{a}_y^2 + \bar{a}_z^2}\end{aligned}$$

Dieser Wert wird anschließend in eine Datei gespeichert und für die folgenden Berechnungen verwendet.

3.4.3 Bestimmung Orientierung

Bei jedem Wechsel vom Zustand **Stillstand** in den Zustand **Bewegung** werden die durch den Madgwick-Filter bestimmten Orientierungswerte $Pitch(0)$, $Roll(0)$ und $Yaw(0)$ der IMU als Anfangsorientierung gespeichert. Während der Bewegungsphase wird die aktuelle Orientierung lediglich aus den Messwerten des Gyroskops bestimmt. Die mit der Abtastrate Δt zu den Abtastzeitpunkten t_n gemessenen Winkelgeschwindigkeiten $\omega_x(t_n)$, $\omega_y(t_n)$ und $\omega_z(t_n)$ werden mit

der Rotationsmatrix \mathbf{R} in das Erd-Koordinatensystem rotiert und anschließend integriert:

$$\mathbf{R} = \begin{bmatrix} 0 & \cos(\text{Roll}(t_n)) \cdot \cos(\text{Pitch}(t_n)) & -\sin(\text{Roll}(t_n)) \cdot \cos(\text{Pitch}(t_n)) \\ \cos(\text{Pitch}(t_n)) & \sin(\text{Roll}(t_n)) \cdot \sin(\text{Pitch}(t_n)) & \cos(\text{Roll}(t_n)) \cdot \sin(\text{Pitch}(t_n)) \\ 0 & \sin(\text{Roll}(t_n)) & \cos(\text{Roll}(t_n)) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\delta \text{Pitch}_{rel}(t_n)}{\delta t} \\ \frac{\delta \text{Roll}_{rel}(t_n)}{\delta t} \\ \frac{\delta \text{Yaw}_{rel}(t_n)}{\delta t} \end{bmatrix} = \frac{1}{\cos(\text{Pitch}(t_n))} \cdot \mathbf{R} \cdot \begin{bmatrix} \omega_x(t_n) \\ \omega_y(t_n) \\ \omega_z(t_n) \end{bmatrix}$$

$$\begin{bmatrix} \text{Pitch}_{rel}(t_n) \\ \text{Roll}_{rel}(t_n) \\ \text{Yaw}_{rel}(t_n) \end{bmatrix} = \begin{bmatrix} \text{Pitch}_{rel}(t_{n-1}) \\ \text{Roll}_{rel}(t_{n-1}) \\ \text{Yaw}_{rel}(t_{n-1}) \end{bmatrix} + \begin{bmatrix} \frac{\delta \text{Pitch}_{rel}}{\delta t} \\ \frac{\delta \text{Roll}_{rel}}{\delta t} \\ \frac{\delta \text{Yaw}_{rel}}{\delta t} \end{bmatrix} \cdot \Delta t$$

Mit $\text{Pitch}_{rel}(t_n)$, $\text{Roll}_{rel}(t_n)$ und $\text{Yaw}_{rel}(t_n)$ ist die relative Orientierung seit Beginn der Bewegungs-Phase zum Abtastzeitpunkt t_n gegeben. Die absolute Orientierung erhält man mit der Anfangsorientierung wie folgt:

$$\begin{aligned} \text{Pitch}(t_n) &= \text{Pitch}(0) + \text{Pitch}_{rel}(t_n) \\ \text{Roll}(t_n) &= \text{Roll}(0) + \text{Roll}_{rel}(t_n) \\ \text{Yaw}(t_n) &= \text{Yaw}(0) + \text{Yaw}_{rel}(t_n) \end{aligned}$$

3.4.4 Abziehen Gravitation

Da die Gravitation \vec{g} stets auf den Sensor einwirkt, muss diese abgezogen werden, um die relativen Beschleunigungen $a_{x,rel}(t_n)$, $a_{y,rel}(t_n)$ und $a_{z,rel}(t_n)$ des Sensors zu erhalten. Dafür benötigt man zu jedem Zeitpunkt die Orientierung dieses zum Gravitationsvektor. Während der Bewegungsphase werden die in Abschnitt 3.4.3 berechneten Winkel $\text{Pitch}(t_n)$ und $\text{Roll}(t_n)$ genutzt, anderenfalls die Orientierungswerte des Madgwick-Filters, um den Einfluss der in Abschnitt 3.4.2 bestimmten Gravitation von den Beschleunigungsmesswerten $a_x(t_n)$, $a_y(t_n)$ und $a_z(t_n)$ abzuziehen:

$$\begin{aligned} a_{x,rel}(t_n) &= a_x(t_n) + \sin(\text{Pitch}(t_n)) \cdot |\vec{g}| \\ a_{y,rel}(t_n) &= a_y(t_n) - \sin(\text{Roll}(t_n)) \cdot \cos(\text{Pitch}(t_n)) \cdot |\vec{g}| \\ a_{z,rel}(t_n) &= a_z(t_n) - \cos(\text{Roll}(t_n)) \cdot \cos(\text{Pitch}(t_n)) \cdot |\vec{g}| \end{aligned}$$

Mit $a_{x,rel}(t_n)$, $a_{y,rel}(t_n)$ und $a_{z,rel}(t_n)$ erhält man die auf den Sensor bezogenen relativen Beschleunigungswerte für jeden Abtastzeitpunkt t_n der Bewegungs-, bzw. Stillstandphase.

3.4.5 Bestimmung Offsetwerte

Bei jedem Wechsel vom Zustand **Stillstand** in den Zustand **Bewegung** wird für jede Sensor-Achse zusätzlich ein Offsetwert aus der in Abschnitt 3.4.1 beschriebenen Datenaufzeichnung

berechnet, um Störeinflüsse später korrigieren zu können:

$$\begin{aligned} offset_x &= \frac{1}{N} \cdot \sum_{n=0}^N a_{x,rel}(t_n) \\ offset_y &= \frac{1}{N} \cdot \sum_{n=0}^N a_{y,rel}(t_n) \\ offset_z &= \frac{1}{N} \cdot \sum_{n=0}^N a_{z,rel}(t_n) \end{aligned}$$

3.4.6 Bestimmung relative Geschwindigkeit und zurückgelegte Strecke

Bei jedem Wechsel vom Zustand **Bewegung** in den Zustand **Stillstand** wird die Geschwindigkeit und die zurückgelegte Strecke aus der in 3.4.1 beschriebenen Datenaufzeichnung berechnet:

$$\begin{aligned} v_{x,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N (a_{x,rel}(t_n) - offset_x) \\ v_{y,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N (a_{y,rel}(t_n) - offset_y) \\ v_{z,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N (a_{z,rel}(t_n) - offset_z) \end{aligned}$$

Mit $v_{x,rel}(t_N)$, $v_{y,rel}(t_N)$ und $v_{z,rel}(t_N)$ erhält man die aktuelle Geschwindigkeit in die jeweilige Achsrichtung nach N Datensätzen, bzw. zum Zeitpunkt $t_N = N \cdot \Delta t$.

Hiermit werden die zurückgelegten Strecken $s_{x,rel}(t_N)$, $s_{y,rel}(t_N)$ und $s_{z,rel}(t_N)$ berechnet:

$$\begin{aligned} s_{x,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N v_{x,rel}(t_n) \\ s_{y,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N v_{y,rel}(t_n) \\ s_{z,rel}(t_N) &= \Delta t \cdot \sum_{n=0}^N v_{z,rel}(t_n) \end{aligned}$$

Kapitel 4

Messreihen und Optimierungen

4.1 Untersuchungen

In den folgenden Abschnitten werden verschiedene Untersuchungen mit den zur Verfügung stehenden IMUs beschrieben und deren Auswirkungen auf die Messgenauigkeit ausgewertet.

4.1.1 Erkenntnisse Madgwick-Filter

Wie in Abschnitt 2.3.1 erläutert, ist für die Bestimmung räumlicher Bewegungen über Beschleunigungsmesswerte das exakte Wissen über die aktuelle Orientierung des aufzeichnenden Sensors nötig.

Der Madgwick-Filter liefert präzise Orientierungswerte, falls sich der Sensor im Stillstand befindet.

In Gleichung 2.10 ist ersichtlich, dass die Beschleunigungswerte direkt in die Berechnung der Orientierung eingehen. Der Filter nimmt die Parallelität des Beschleunigungsvektors und des Gravitationsvektors an. In Situationen, in denen dies nicht der Fall ist, liefert der Filter verfälschte Orientierungswerte. Dies soll im Folgenden für *Pitch* und *Roll* veranschaulicht werden.

Mithilfe des KUKA-Roboters wurden lineare Bewegungen auf die Phidgets IMU 1 ausgeübt. Abbildung 4.1 zeigt die durch den Madgwick-Filter berechneten Orientierungswerte, wenn eine dynamische Beschleunigung in Richtung der Z-Achse des Sensor-Koordinatensystems gemessen wurde. Abbildung 4.2 zeigt die Werte für eine Beschleunigung in Richtung der X-Achse des Sensors. Der KUKA-Roboter befand sich jeweils für $t < 0.5s$ und $t > 2s$ im Stillstand. Zum Zeitpunkt $t = 1.25s$ war die Geschwindigkeit maximal.

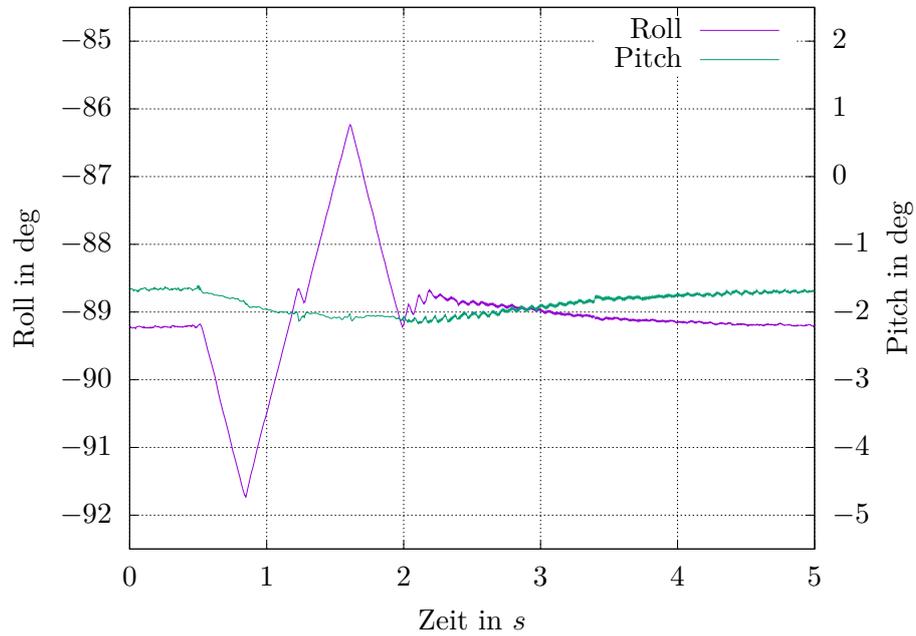


Abbildung 4.1: Roll und Pitch des Madgwick-Filters während einer Beschleunigung in Z-Richtung der Phidgets IMU 1

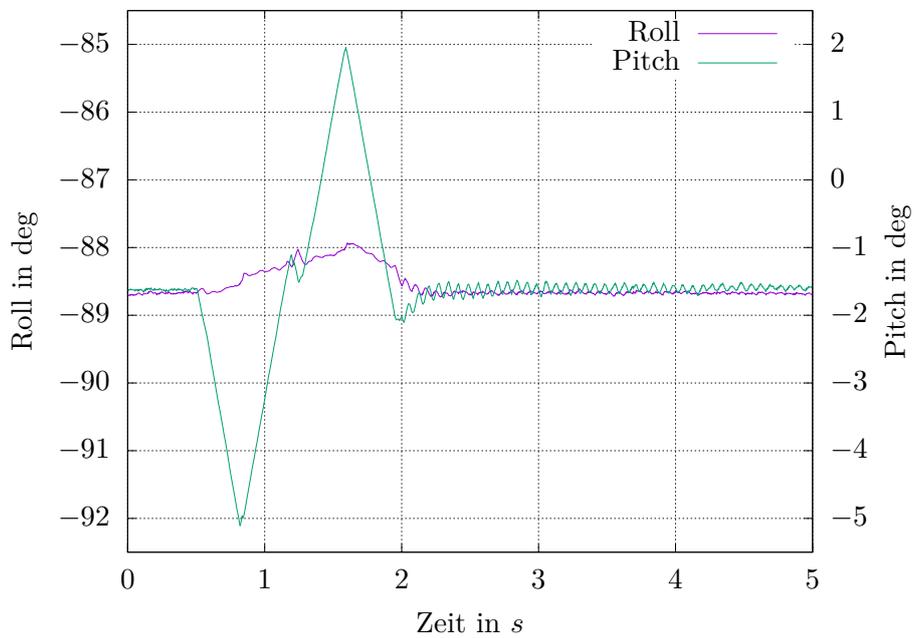


Abbildung 4.2: Roll und Pitch des Madgwick-Filters während einer Beschleunigung in X-Richtung der Phidgets IMU 1

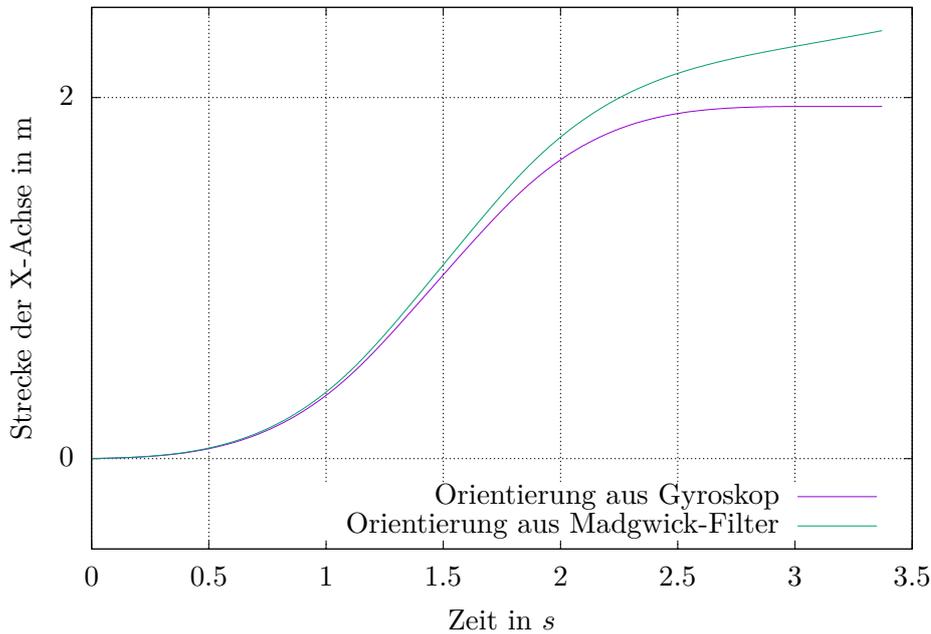


Abbildung 4.3: Vergleich der zurückgelegten Strecke bei einer Halbkreisbewegung mit maximaler Geschwindigkeit unter Verwendung der Orientierungswerte des Madgwick-Filters, bzw. des Gyroskops

Die absolute Orientierung des Sensors wurde während der Bewegung nicht geändert. In beiden Abbildungen ist jedoch ersichtlich, dass der Madgwick-Filter eine Änderung der Orientierung berechnet, wenn eine dynamische Beschleunigung gemessen wird.

In Abbildung 4.1 ist zudem ersichtlich, dass die Orientierungswerte des Madgwick-Filters in diesem Fall erst etwa 2.5s nach Bewegungsstopp wieder mit der tatsächlichen Orientierung übereinstimmen.

Die Orientierungswerte *Pitch* und *Roll* sind, wie in Abschnitt 3.4.4 zu sehen, essentiell für die Bestimmung der relativen Beschleunigung des Sensors. Sind diese Werte fehlerbehaftet, führt dies zu Ungenauigkeiten bei der in Abschnitt 3.4.6 erläuterten Berechnung der relativen Geschwindigkeit und der zurückgelegten Strecke.

Optimierung der Orientierungswerte

Verwendet man zur Berechnung der aktuellen Orientierung lediglich die Messwerte des Gyroskops, wie bereits in Abschnitt 3.4.3 erklärt, erhält man exaktere Orientierungswerte.

Zum Vergleich wurden mit dem KUKA-Roboter horizontale Halbkreisbewegungen gefahren. Sämtliche Orientierungswerte des Sensors wurden dabei konstant gehalten. Die X-Achse des Beschleunigungssensors war so ausgerichtet, dass dessen zweifach integrierte Messwerte dem Durchmesser des Halbkreises von 2m entsprechen sollten.

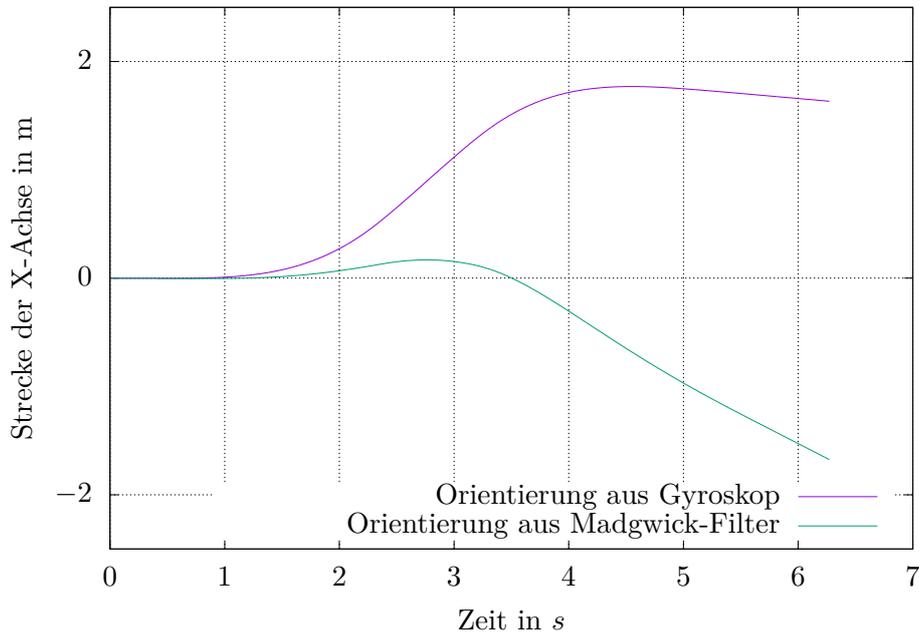


Abbildung 4.4: Vergleich der zurückgelegten Strecke bei einer Halbkreisbewegung mit halber Maximalgeschwindigkeit unter Verwendung der Orientierungswerte des Madgwick-Filters, bzw. des Gyroskops

Die Abbildungen 4.3 und 4.4 stellen die berechneten Strecken bei der Verwendung der Orientierungswerte des Gyroskops, bzw. bei der Verwendung der Werte des Madgwick-Filters dar.

Fährt der KUKA-Roboter die Kreisbewegung mit maximaler Geschwindigkeit ab, erreicht die berechnete Strecke bei Nutzung der Gyroskopwerte annähernd den tatsächlichen Durchmesser von $2m$. Unter Verwendung der Werte des Madgwick-Filters wird das Ergebnis schlechter und die Werte Driften nach Beendigung der Bewegung zum Zeitpunkt $t = 3s$ ab.

Fährt der KUKA-Roboter die Bewegung mit halber Geschwindigkeit, sinkt die Genauigkeit der berechneten Strecke bei Verwendung der Gyroskopwerte. Im Vergleich dazu sind die Werte der berechneten Strecke unter Verwendung des Madgwick-Filters nahezu unbrauchbar und driften schon vor der Beendigung der Bewegung zum Zeitpunkt $t = 6s$ stark ab.

Die vorangegangenen Erkenntnisse zeigen, dass bei der Abrechnung der Gravitation während einer Bewegung auf die Orientierungswerte des Gyroskops zurückgegriffen werden sollte. Die Orientierungswerte des Madgwick-Filters sind dafür ungeeignet, da diese von dynamischen Beschleunigungen stark beeinflusst werden.

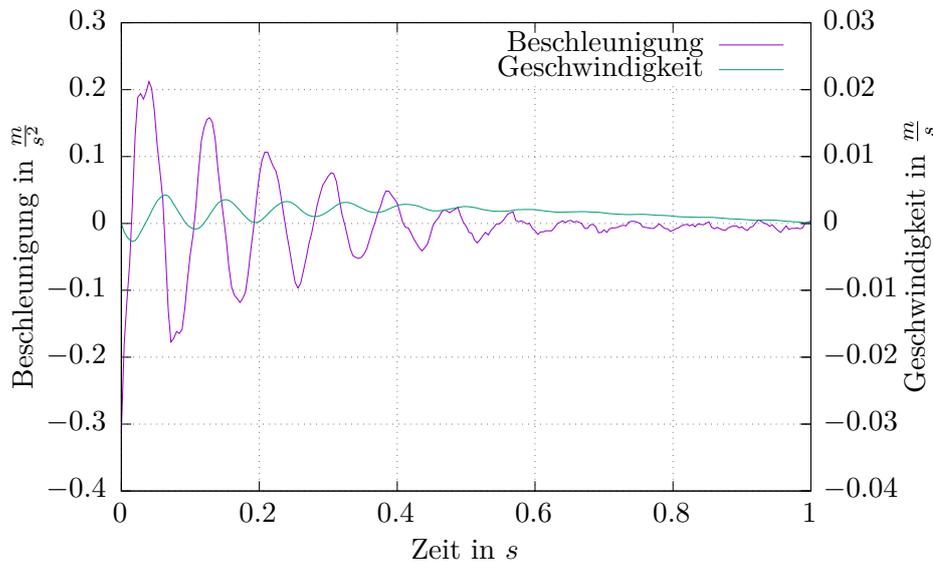


Abbildung 4.5: Schwingung der Beschleunigungsmesswerte nach Bewegungsstopp mit daraus resultierender Geschwindigkeit

4.1.2 Erkenntnisse Beschleunigungssensor

Schwingung der Beschleunigungsmesswerte

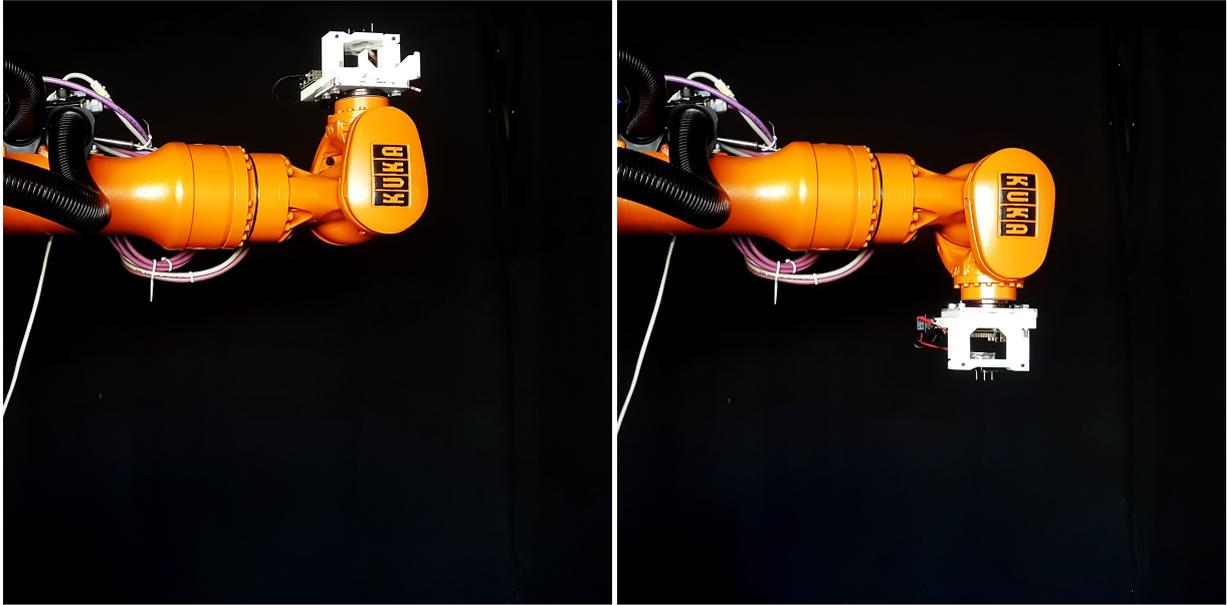
Die verwendeten IMUs basieren, wie in Abschnitt 2.1 erläutert, auf einem Feder-Masse-System. Die Masse wird aufgrund einer sich ändernden Beschleunigung aus der Ruhelage gebracht. Ändert sich die einwirkende Beschleunigung nicht mehr, schwingt sie noch kurze Zeit nach. Dieses Verhalten soll nachgewiesen und ausgewertet werden.

In Abbildung 4.5 ist die gemessene Beschleunigung der Phidgets IMU 1 und die daraus berechnete Geschwindigkeit dargestellt. Zum Zeitpunkt $t = 0$ ändert sich in diesem Fall die wirkende Beschleunigung nicht mehr. Bis etwa zum Zeitpunkt $t = 0.6s$ wird jedoch noch eine abklingende periodische Beschleunigung gemessen. Betrachtet man die berechnete Geschwindigkeit, so zeigt sich, dass diese periodische Schwingung einen vernachlässigbaren Einfluss darstellt, da diese sich durch eine zeitliche Integration aufhebt.

Einfluss äußerer Störgrößen auf die Beschleunigungsmesswerte

Anlass für die folgenden Untersuchungen war die Beobachtung, dass die verwendete Phidgets IMU 1 zeitweise deutlich unterschiedliche Gravitationen aufzeichnete, sobald diese am KUKA-Roboter befestigt war.

In dieser Versuchsreihe wurde die Erd-Gravitation mit den drei Sensor-Achsen der am KUKA-Roboter befestigten Phidgets IMU 1, wie in Abbildung 4.6 veranschaulicht, jeweils in positiver



(a) Sensor oberhalb des Handgelenks

(b) Sensor unterhalb des Handgelenks

Abbildung 4.6: Unterschiedliche Ausrichtungen des KUKA-Roboters

und negativer Richtung gemessen. Dazu wurde jede Achse des Beschleunigungssensors parallel ($\uparrow\uparrow$), bzw. antiparallel ($\uparrow\downarrow$) zum Erd-Gravitationsvektor ausgerichtet, um die volle Gravitation mit nur einer Achsrichtung messen zu können.

Während der Sensor sich 30 Minuten im Stillstand befand, wurden die unbearbeiteten Beschleunigungswerte a_x , a_y und a_z aufgezeichnet.

Da es nicht möglich ist, die Achsen ohne Abweichung exakt parallel zum Erd-Gravitationsvektor auszurichten, erfahren auch die horizontal ausgerichteten Sensor-Achsen eine Beschleunigung. Um dennoch vergleichbare Ergebnisse zu erhalten, wurde der Betrag des dreidimensionalen Beschleunigungsvektors berechnet:

$$|\vec{g}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Der Fehler, welcher dadurch in die berechnete Gravitation einging, ist vernachlässigbar gering, da die zwei orthogonal zum Erd-Gravitationsvektor stehenden Achsen nur eine durchschnittliche Beschleunigung von $0.0065 \frac{m}{s^2}$ aufgezeichnet haben.

Für jeden Abtastzeitpunkt des Sensors wurde eine Gravitation $|\vec{g}|$ bestimmt. Um diese Werte vergleichen zu können, wurde der Median der Gravitationsmesswerte bestimmt und dieser auf vier signifikante Dezimalstellen gerundet: Der Betrag der gemessenen Gravitation $|\vec{g}|$ sollte theoretisch ohne Störeinflüsse und Skalierungsfehler unabhängig von der Orientierung des Sensors sein. Die Ergebnisse zeigen jedoch eine deutliche Abhängigkeit von der Orientierung.

Aus den Werten von Tabelle 4.1 erhalten wir den folgenden Mittelwert der Gravitation mit

Tabelle 4.1: Gravitationsmesswerte der einzelnen Achsen

Ausrichtung der IMU	$\vec{g} \uparrow \uparrow \vec{x}$	$\vec{g} \uparrow \downarrow \vec{x}$	$\vec{g} \uparrow \uparrow \vec{y}$	$\vec{g} \uparrow \downarrow \vec{y}$	$\vec{g} \uparrow \uparrow \vec{z}$	$\vec{g} \uparrow \downarrow \vec{z}$
Gravitation $ \vec{g} $ in $\frac{m}{s^2}$	9.8510	9.8054	9.8196	9.8361	10.1377	9.5225

entsprechender Standardabweichung:

$$g_{Erde,1} = (9.82872 \pm 0.19515) \frac{m}{s^2} \quad (4.1)$$

Dieser Wert kommt nahe an den Literaturwert von $g = 9.81 \frac{m}{s^2}$ [24]. Man kann davon ausgehen, dass ein von außen wirkender Störfaktor auf den Sensor wirkt und kein produktionsbedingtes Problem seitens des Sensors vorliegt.

Theoretisch müsste jede Sensor-Achse ohne Störeinflüsse exakt g messen.

Wir betrachten nun die Mittelwerte der Messwerte pro Achse, welche wie folgt berechnet wurden:

$$\begin{aligned} g_x &= \frac{1}{2} \cdot (g_{\vec{g} \uparrow \uparrow \vec{x}} + g_{\vec{g} \uparrow \downarrow \vec{x}}) = \mathbf{9.8282} \frac{m}{s^2} \\ g_y &= \frac{1}{2} \cdot (g_{\vec{g} \uparrow \uparrow \vec{y}} + g_{\vec{g} \uparrow \downarrow \vec{y}}) = \mathbf{9.82785} \frac{m}{s^2} \\ g_z &= \frac{1}{2} \cdot (g_{\vec{g} \uparrow \uparrow \vec{z}} + g_{\vec{g} \uparrow \downarrow \vec{z}}) = \mathbf{9.8301} \frac{m}{s^2} \end{aligned}$$

Aus diesen Werten erhalten wir dieselbe Gravitation wie in Gleichung 4.1 mit geringerer Standardabweichung:

$$g_{Erde,2} = (9.82872 \pm 0.001211) \frac{m}{s^2} \quad (4.2)$$

Da der Fehler der Gravitation in Gleichung 4.2 deutlich geringer ist als der Fehler der Gravitation in Gleichung 4.1, kann man davon ausgehen, dass der KUKA-Roboter diesen Störeinfluss systematisch ausübt. Dies soll im Folgenden genauer untersucht werden.

Wie in Tabelle 4.1 zu sehen, zeichnen die positive und die negative Z-Achse deutlich unterschiedliche Werte auf. Bei der Ausrichtung der positiven Z-Achse parallel zum Erd-Gravitationsvektor befand sich das Gelenk des KUKA Roboters, wie in Abbildung 4.6a zu sehen, unterhalb der IMU. Bei der Ausrichtung der negativen Z-Achse befand sich das Gelenk, dargestellt in Abbildung 4.6b, oberhalb der IMU.

Man kann somit davon ausgehen, dass die Masse des Roboter-Gelenks die Beschleunigungsmessung der IMU stört. Die Elektromotoren des Roboters können als Störquelle ausgeschlossen werden, da diese während der Messdatenaufzeichnung abgeschaltet waren.

Um den beobachteten Störeffekt des KUKA-Roboters auf die Beschleunigungsmessung zu verdeutlichen, wurde eine spezielle Messreihe aufgezeichnet. Diese zeigt die gemessene Gravitation für unterschiedliche Orientierungen. Zur Verifizierung der Ergebnisse wurde das Experiment mit der Phidgets IMU 2 und der Xsens IMU wiederholt.

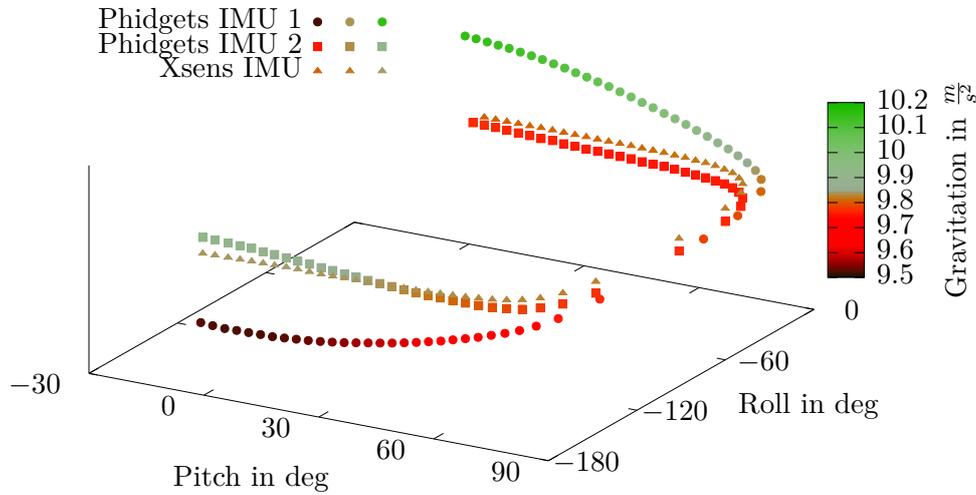


Abbildung 4.7: Orientierungsabhängige Gravitationsmessung mit verschiedenen IMUs

Wie in Abbildung 4.7 zu sehen ist, zeichnen die beiden baugleichen IMUs deutlich unterschiedliche Gravitationen auf. Dabei ist es unerklärlich, weshalb die Messwerte der IMUs solch große Abweichungen zueinander aufweisen. Dennoch kann man bei allen Sensoren den Einfluss des Roboters erkennen. Mit steigendem *Pitch* weichen die Gravitationsmessungen zunehmend vom Literaturwert der Erdgravitation ab.

Im Vergleich zu den beiden Phidgets IMUs sind die Messwerte der Xsens IMU weniger orientierungsabhängig.

Für einen direkten Vergleich sind in Tabelle 4.2 die Messwerte aus Abbildung 4.7 im Mittel und mit zugehöriger Standardabweichung aufgelistet:

Tabelle 4.2: Gravitationsmesswerte unterschiedlicher IMUs

IMU	Phidgets IMU 1	Phidgets IMU 2	Xsens IMU
Gravitation in $\frac{m}{s^2}$	(9.8056 ± 0.2247)	(9.7987 ± 0.0529)	(9.8226 ± 0.0123)

Für die Werte *Pitch* ≈ 0 deg und *Roll* ≈ -180 deg befand sich das Handgelenk des Roboters unterhalb der IMU und für die Werte *Pitch* ≈ 0 deg und *Roll* ≈ 0 deg oberhalb der IMU.

Der Störeinfluss erreicht sein Maximum, wenn sich die IMU auf der Verbindungslinie zwischen Gravitationsmittelpunkt der Erde und dem Handgelenk des Roboters befindet.

Sehr interessant ist dabei auch, dass die baugleichen IMUs gegensätzlich durch den Roboter beeinflusst werden. Auch dafür gibt es zum jetzigen Kenntnisstand keine plausible Erklärung.

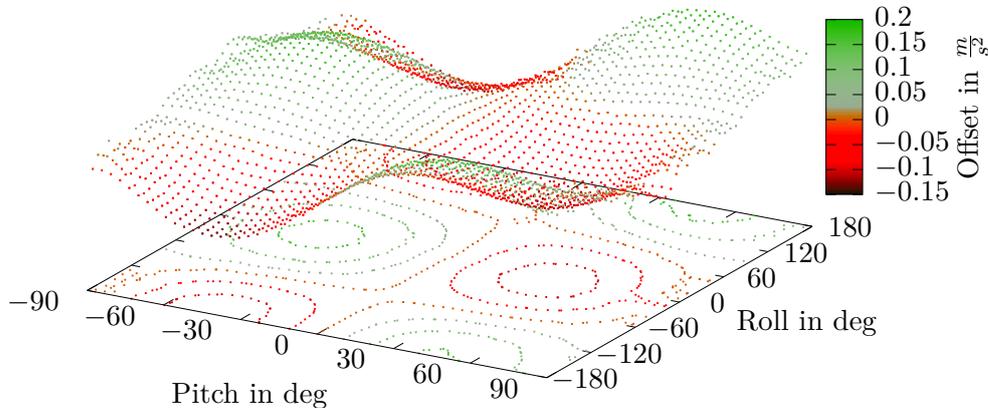


Abbildung 4.8: Offsetwerte der X-Sensor-Achse

Korrektur der Beschleunigungsmesswerte

Der vorherige Abschnitt zeigt auf, in welchen Größenordnungen die Beschleunigungsmesswerte durch äußere Störfaktoren beeinflusst werden können. Wie in Abbildung 4.7 dargestellt, kann dieselbe Störquelle unterschiedliche Effekte auf selbst baugleiche Sensoren ausüben.

Da das Ziel dieser Arbeit unter anderem die Berechnung der zurückgelegten Strecke ist, müssen die Beschleunigungswerte möglichst fehlerfrei bestimmt werden, da ansonsten der Fehler zweifach zeitlich integriert wird und dadurch die Ergebnisse unbrauchbar werden.

Wie bereits dargelegt, verfälscht der KUKA-Roboter die Messwerte systematisch, weswegen es möglich ist, den Fehler abhängig von der Orientierung zu bestimmen und zu korrigieren. Es soll nun eine Strategie entwickelt werden, um die eben genannten Störgrößen zu eliminieren.

In den Abbildungen 4.8, 4.9 und 4.10 sind die absoluten Fehler der Gravitationsmessung jeder Sensor-Achse der Phidgets IMU 1 in Abhängigkeit von der Orientierung dargestellt. Dabei ist unter Offset die Abweichung der gemessenen Beschleunigung im Stillstand von der tatsächlichen Erdbeschleunigung zu verstehen. Da die Fehler wie abgebildet orientierungsabhängig sind, müsste jede Beschleunigungsmessung mit einem spezifischen Offsetwert korrigiert werden.

Eine Möglichkeit zur Korrektur wäre, die Speicherung sämtlicher Offset-Werte für jede Orientierung und die ständige Ergänzung dieser während des Programmablaufs. Da der Suchaufwand für die gespeicherten Offset-Werte hoch wäre und die Leistungsfähigkeit des Raspberry Pi's an seine Grenzen bringen könnte, ist dieser Lösungsansatz nicht praktikabel.

Eine effizientere Möglichkeit ist die Annäherung einer dreidimensionalen Funktion an die Offset-Messwerte. Damit können ohne großen Rechen- und Suchaufwand die entsprechenden Offset-Werte für jede Orientierung und jede Achse berechnet werden.

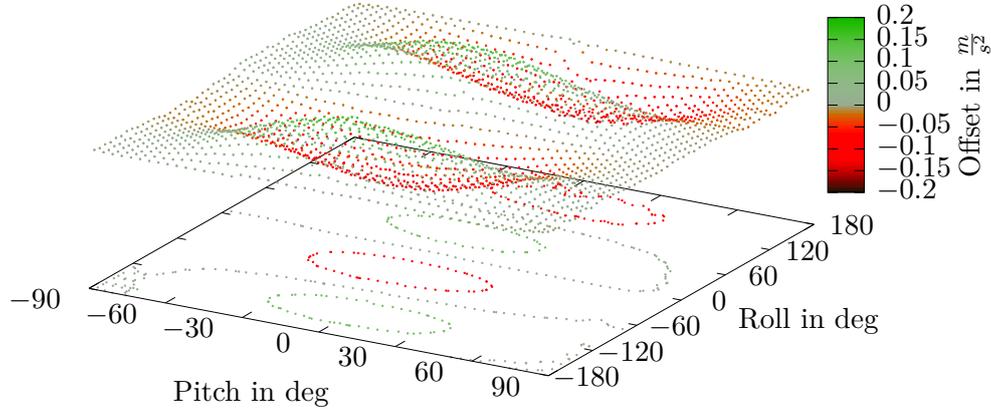


Abbildung 4.9: Offsetwerte der Y-Sensor-Achse

Die folgenden drei Funktionen nähern die Offset-Werte für jede Sensor-Achse der Phidgets IMU 1 an:

$$\text{offset}_x(\text{Pitch}, \text{Roll}) = \alpha_x \cdot \sin(2 \cdot \text{Pitch}) \cdot \cos(\text{Roll}) + \frac{\text{Pitch}}{\beta_x} + \frac{\text{Roll}}{\gamma_x} \quad (4.3)$$

$$\text{offset}_y(\text{Pitch}, \text{Roll}) = \alpha_y \cdot \sin(2 \cdot \text{Roll}) \cdot \cos(\text{Pitch}) + \frac{\text{Pitch}}{\beta_y} + \frac{\text{Roll}}{\gamma_y} \quad (4.4)$$

$$\text{offset}_z(\text{Pitch}, \text{Roll}) = \alpha_z \cdot |\cos(\text{Roll})| \cdot \cos(\text{Pitch}) + \frac{\text{Pitch}}{\beta_z} + \frac{\text{Roll}}{\gamma_z} \quad (4.5)$$

Die Konstanten α , β und γ können dabei so angepasst werden, dass der Fehler minimal wird. Um die Genauigkeit der Approximation zu erhöhen, werden die Funktionen pro Sensor-Achsen in acht Quadranten, aufgelistet in Tabelle 4.3, eingeteilt.

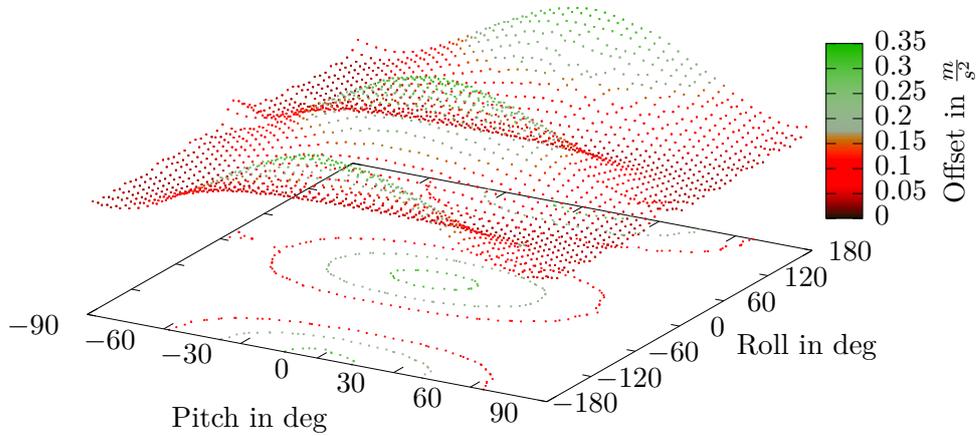


Abbildung 4.10: Offsetwerte der Z-Sensor-Achse

Tabelle 4.3: Definition der Quadranten

Quadrant \ Werte	$Pitch\ x_1$ in deg	$Roll\ x_2$ in deg
1	$0 \leq x_1 < 90$	$-180 \leq x_2 < -90$
2	$0 \leq x_1 < 90$	$90 \leq x_2 < 180$
3	$0 \leq x_1 < 90$	$-90 \leq x_2 < 0$
4	$-90 \leq x_1 < 0$	$-90 \leq x_2 < 0$
5	$-90 \leq x_1 < 0$	$-180 \leq x_2 < -90$
6	$-90 \leq x_1 < 0$	$90 \leq x_2 < 180$
7	$-90 \leq x_1 < 0$	$0 \leq x_2 < 90$
8	$0 \leq x_1 < 90$	$0 \leq x_2 < 90$

Für die Bestimmung der jeweiligen Konstanten wurde für jeden Quadranten die in den Abbildungen 4.8 bis 4.10 dargestellten Offsetwerte mit jeweils unterschiedlicher Orientierung bestimmt. Von den Beschleunigungswerten der jeweils vier Sekunden andauernden Stillstand-Phasen wurde entsprechend der in Abschnitt 3.4.4 erläuterten Methode die Gravitation abgerechnet und die Werte gemittelt.

Die Konstanten wurden so gewählt, dass die in Gleichung 4.6 definierte quadratische Fehlerfunktion minimal wird. Dabei wurden zuerst die Konstante α mit $\beta \rightarrow \infty$ und $\gamma \rightarrow \infty$, danach β mit $\gamma \rightarrow \infty$ und anschließend γ bestimmt.

Im folgenden ist N die Anzahl der gemessenen Offsetwerte je Quadrant und pro Sensor-Achse.

Die statische Beschleunigung nach Abzug der Gravitation ist durch $a_{rel,n}$ mit den zugehörigen Orientierungswerten $Pitch_n$ und $Roll_n$ gegeben.

$$F = \sum_{n=0}^N (offset(Pitch_n, Roll_n) - a_{rel,n})^2 \quad (4.6)$$

Die Tabellen 4.4 bis 4.6 listen die Konstanten für die Funktionen 4.3 bis 4.5 auf, für welche die Fehlerfunktion 4.6 minimal wird.

Tabelle 4.4: Konstanten der X-Sensor-Achse

Quadrant \ Konstanten	α_x in $\frac{m}{s^2}$	β_x in deg	γ_x in deg
1	-0,156	7890	47700
2	-0,165	4470	-21800
3	-0,131	7690	20000
4	-0,165	-5410	17400
5	-0,132	-7230	42000
6	-0,122	-5050	-24530
7	-0,167	-5670	-34600
8	-0,118	5500	28400

Tabelle 4.5: Konstanten der Y-Sensor-Achse

Quadrant \ Konstanten	α_y in $\frac{m}{s^2}$	β_y in deg	γ_y in deg
1	0,136	-5660	-65900
2	0,146	-17000	-1290000
3	0,118	2940	11700
4	0,146	-23000	25100
5	0,128	3440	-26200
6	0,109	-3310	-31840
7	0,134	8480	11800
8	0,109	-3270	208000

Tabelle 4.6: Konstanten der Z-Sensor-Achse

Quadrant \ Konstanten	α_z in $\frac{m}{s^2}$	β_z in deg	γ_z in deg
1	0,285	-4310	-45100
2	0,293	-6030	56800
3	0,271	-2300	41600
4	0,301	4680	8550
5	0,28	2280	-23100
6	0,269	2210	22800
7	0,296	4650	-8640
8	0,269	-2280	-86100

Um eine Bewertung über die Güte der angenäherten Offsetfunktionen 4.3, 4.4 und 4.5 zu erhalten, wird im Folgenden die mittlere absolute Abweichung der berechneten Offsetwerte zu den gemessenen Offsetwerten gemäß der Gleichung 4.7 berechnet. N ist die Anzahl der gemessenen Offsetwerte pro Sensor-Achse der Phidgets IMU 1.

$$d = \frac{1}{N} \sum_{n=0}^N |offset(Pitch_n, Roll_n) - a_{rel,n}| \quad (4.7)$$

Für die berechneten Konstanten aus den Tabellen 4.4 bis 4.6 ergeben sich die folgenden mittleren Abweichungen:

$$d_x = \mathbf{0.79872876} \frac{cm}{s^2}$$

$$d_y = \mathbf{0.87150554} \frac{cm}{s^2}$$

$$d_z = \mathbf{2.02239646} \frac{cm}{s^2}$$

Nun wird der Mittelwert der Konstanten α pro Sensor-Achse mit der zugehöriger Standardabweichung berechnet:

$$\bar{\alpha}_x = (-0.1445 \pm 0.0208) \frac{m}{s^2}$$

$$\bar{\alpha}_y = (0.12825 \pm 0.01497) \frac{m}{s^2}$$

$$\bar{\alpha}_z = (0.283 \pm 0.013) \frac{m}{s^2}$$

Da die Konstanten α näherungsweise die Maxima bzw. Minima der jeweiligen Offsetfunktionen angeben, beschreiben diese auch direkt den maximalen Fehler der Beschleunigungsmessung. Der maximale Fehler bezeichnet hier die maximale Abweichung der Beschleunigungsmessung von der tatsächlichen Gravitation, wenn keine dynamische Beschleunigung auf den Sensor einwirkt.

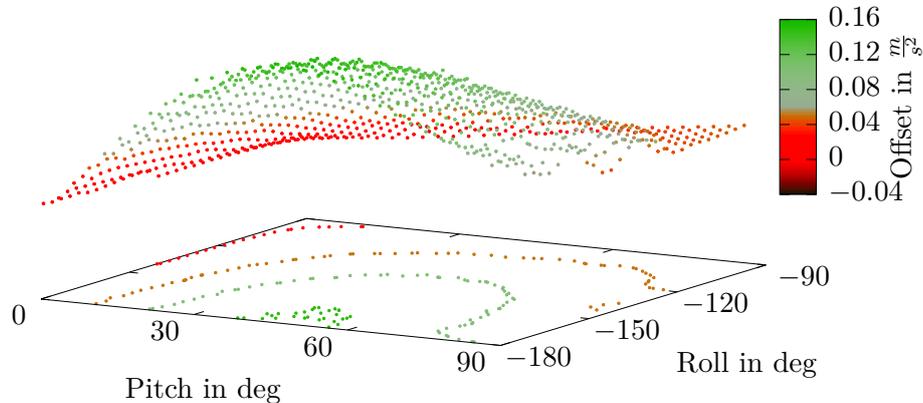


Abbildung 4.11: Offsetwerte der Phidgets IMU 1

Die Konstanten α beschreiben ebenfalls den Betrag der Störgröße, welche jeweils auf die drei Sensor-Achsen einwirkt.

Wie in Abschnitt 4.1.2 bereits veranschaulicht, erzeugt der KUKA-Roboter vermutlich diese Störgröße.

Wenn man den KUKA-Roboter als einzige konstante Störgröße annimmt und sämtliche möglichen Messungenauigkeiten außer Acht lässt, können die Standardabweichungen der drei Konstanten auf einen Skalierungsfehler der Sensor-Achsen zurückgeführt werden.

Die folgende Messreihe stellt dar, dass die gemessenen Offsetwerte sowohl vom Hersteller der IMU als auch produktionsbedingt deutliche Unterschiede aufweisen.

Zum Vergleich wurden die Offsetwerte der jeweiligen X-Sensor-Achse des 1. Quadranten zusätzlich mit der Phidgets IMU 2 und der Xsens IMU bestimmt und in den Abbildungen 4.11 bis 4.13 dargestellt. Es ist ersichtlich, dass jede IMU unterschiedlich beeinflusst wird. Dabei werden die beiden baugleichen Phidgets IMU gegensätzlich von der Störgröße beeinflusst, wobei die Phidgets IMU 1 insgesamt deutlich stärker beeinflusst wird, was auch schon in Abbildung 4.7 gezeigt wurde.

Die Offsetwerte der Xsens IMU sind dagegen wenig bis gar nicht anfällig für die Störgröße, obwohl dieser Sensor nach dem gleichen Messprinzip arbeitet wie dem der Phidgets IMUs.

Da die Sensorik der Xsens IMU im Gegensatz zu den Phidgets IMUs von einem Metallgehäuse umgeben ist, liegt die Vermutung nahe, dass dadurch Störeinflüsse abgeschirmt werden.

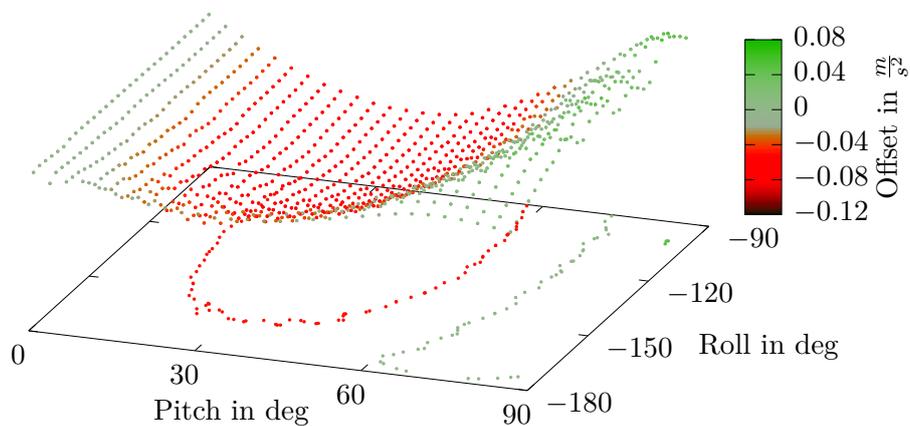


Abbildung 4.12: Offsetwerte der Phidgets IMU 2

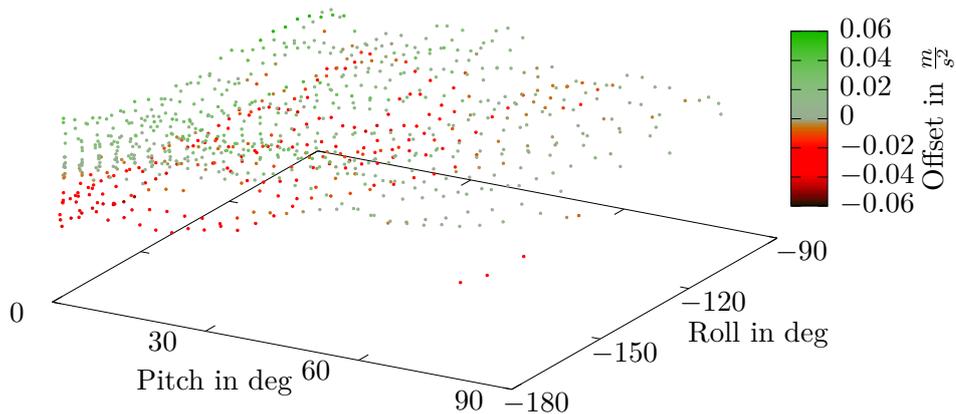


Abbildung 4.13: Offsetwerte der Xsens IMU

Test der Methoden zur Offsetbestimmung

Dieser Abschnitt, soll darlegen ob und in welchem Maße sich die Nutzung der in Abschnitt 4.1.2 entwickelten Methode positiv auf die Genauigkeit der Beschleunigungsmessung auswirkt.

Der KUKA-Roboter hat dafür eine Halbkreisbewegung auf den Sensor, in diesem Fall die Phidgets IMU 1, ausgeübt. Start- und Endposition der Bewegung sind in der Abbildung 4.6a, bzw. 4.6b dargestellt. Der Sensor befand sich vor und nach der Bewegung im Stillstand. Während der Bewegung wurde die Orientierung so gehalten, dass die X-Achse der IMU stets tangential zur Kreisbahn ausgerichtet war. Die nach den Methoden in Abschnitt 3.4.6 berechnete Strecke $s_{x,rel}$ sollte daher im Optimalfall, nach Abschluss der Bewegung, dem halben Umfang der Kreisbewegung von $0.75m$ entsprechen.

In den Abbildungen 4.14 bis 4.16 sind die durch die Phidgets IMU 1 aufgezeichneten Streckenprofile dargestellt. Derselbe Bewegungsablauf wurde mit 30%, 50% und 75% der maximal möglichen Geschwindigkeit des KUKA-Roboters wiederholt.

Für einen Vergleich wurden die Beschleunigungsmesswerte einerseits mit nach Abschnitt 3.4.5 berechneten konstanten Offset-Werten korrigiert, andererseits mit den gemäß Abschnitt 4.1.2 berechneten orientierungsabhängigen Offset-Werten korrigiert.

Um die Orientierung des Sensors während der Bewegung zu erhalten, wurde neben den *Pitch* und *Roll* Werten des Madgwick-Filters, auch auf die Werte der in Abschnitt 3.4.3 vorgestellten Methode zurückgegriffen. Die jeweiligen Orientierungswerte wurden dabei sowohl für die Berücksichtigung der Gravitation als auch für die Bestimmung der Offsetwerte genutzt.

Vergleicht man die Abbildungen 4.14 bis 4.16, fällt auf, dass die Ergebnisse abhängig von der gefahrenen Geschwindigkeit deutlich unterschiedlich ausfallen. Tendenziell driften die Werte der gemessenen Strecke bei Verwendung der Orientierungswerte des Madgwick-Filters mehr, als bei Verwendung der Gyroskopwerte, was vor allem in den Abbildungen 4.14 und 4.15 sichtbar ist. In den Abbildungen 4.15 und 4.16 fällt auf, dass unter Verwendung der berechneten Offsetwerte, die berechnete Strecke der tatsächliche gefahrenen Strecke näher kommt, wohingegen in Abbildung 4.14 bessere Ergebnisse durch die Verwendung eines konstanten Offset erzielt werden.

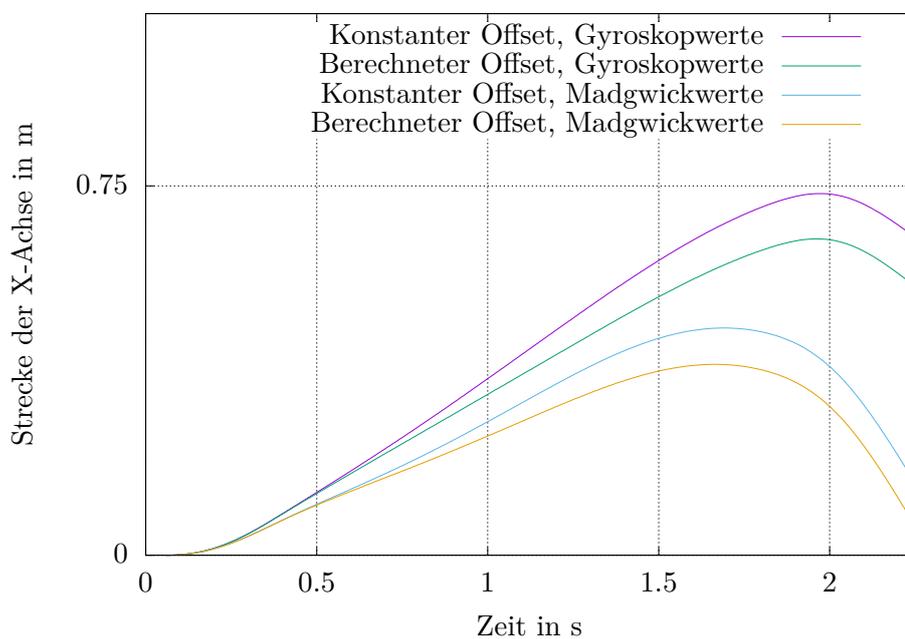


Abbildung 4.14: Streckenprofil der Halbkreisbewegung mit 30% Geschwindigkeit

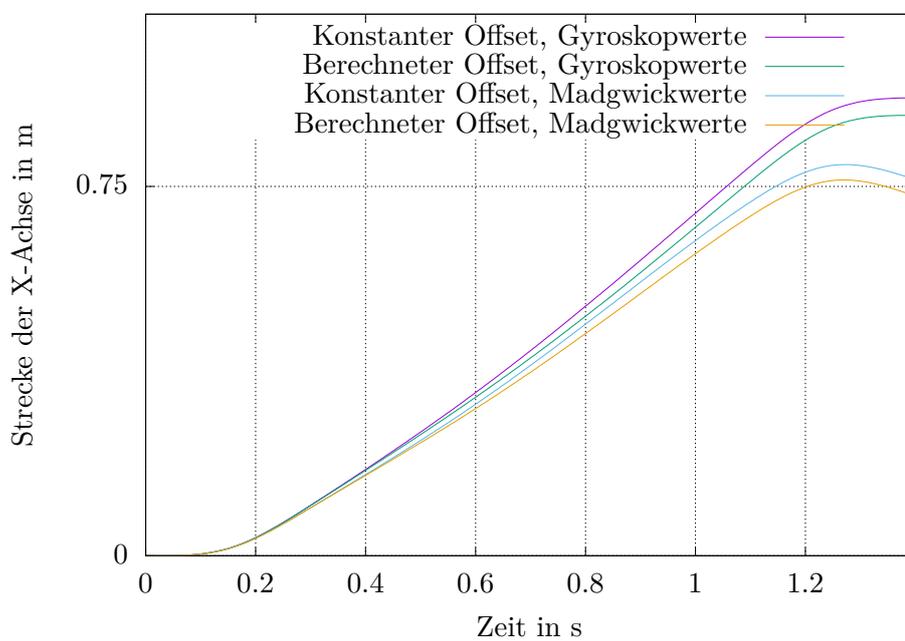


Abbildung 4.15: Streckenprofil der Halbkreisbewegung mit 50% Geschwindigkeit

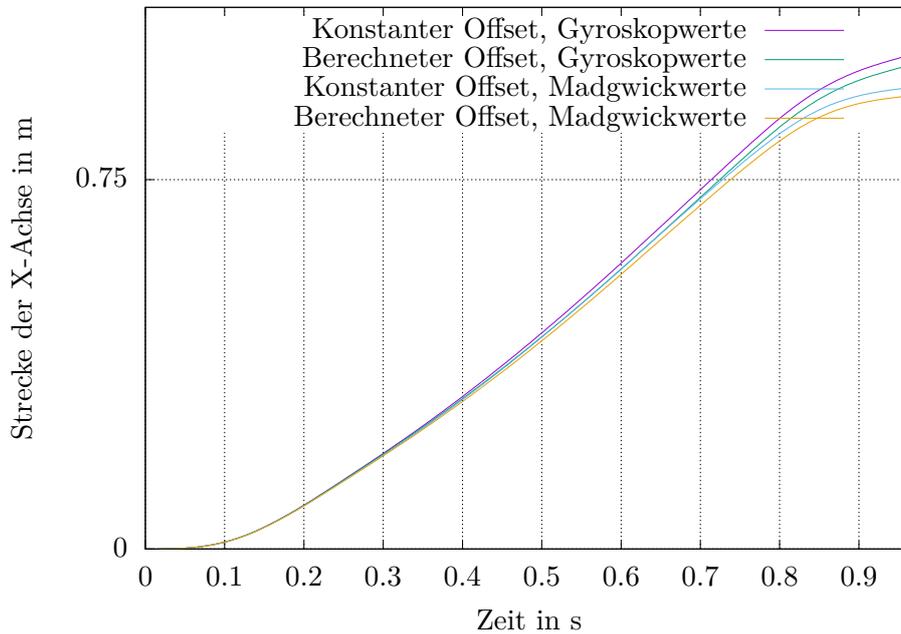


Abbildung 4.16: Streckenprofil der Halbkreisbewegung mit 75% Geschwindigkeit

Zusammenfassung

Zusammenfassend ist nun ersichtlich, dass man eine etwaige Störgröße, welche auf einen Beschleunigungssensor wirkt, bestimmen und eliminieren kann. Da jedoch selbst identische Sensoren unterschiedlich auf Störgrößen reagieren, kann man weder die gezeigten Funktionen noch die entsprechenden Konstanten auf andere Sensoren übertragen. Jeder Sensor muss gesondert untersucht werden, um passende Approximationsfunktionen zu finden. Zudem gelten die oben genannten Funktionen lediglich für eine spezifische konstante Störquelle, in diesem Fall für den KUKA-Roboter.

4.2 Messtrajektorien

Um die erreichte Genauigkeit der vorangegangenen Optimierungsversuche zu testen und bewerten, wurden mit dem KUKA-Roboter lineare und mit dem Volksbot-Roboter realistische Trajektorien gefahren.

4.2.1 KUKA-Roboter

Für die folgenden Messreihen wurde die Phidgets IMU 1, bzw. die Xsens IMU, am KUKA-Roboter befestigt. Dieser bewegte die Sensoren im Sensor-Koordinatensystem gleichzeitig und linear um $0.4m$ in die drei Achsrichtungen. Die akkumulierte Strecke jeder Sensor-Achse sollte dementsprechend im Optimalfall $0.4m$ entsprechen. Vor Beginn und nach Ende der Bewegung befanden sich die Sensoren im Stillstand und die Orientierung wurde während der ausgeübten Bewegung konstant gehalten. Derselbe Bewegungsablauf wurden mit 30% und 100% der maximal möglichen Geschwindigkeit des KUKA-Roboters wiederholt.

Für die Bestimmung der Geschwindigkeit und der zurückgelegten Strecke wurden die Algorithmen aus Abschnitt 3.4 genutzt.

Die Abbildungen 4.17 bis 4.22 stellen das Geschwindigkeits- und Streckenprofil der beiden verwendeten IMUs für die entsprechende Achse und Geschwindigkeit dar.

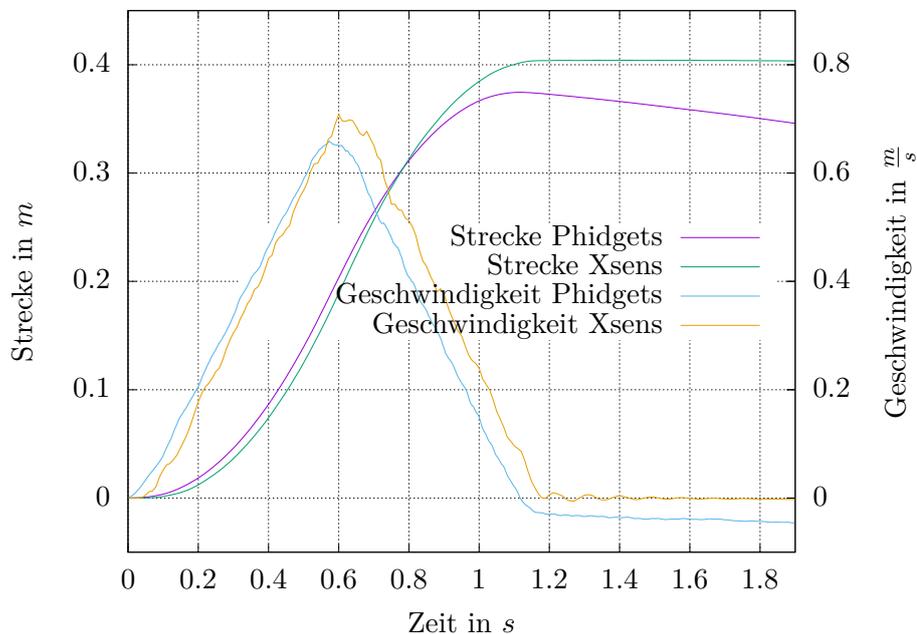


Abbildung 4.17: Geschwindigkeits- und Streckenprofil der X-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich

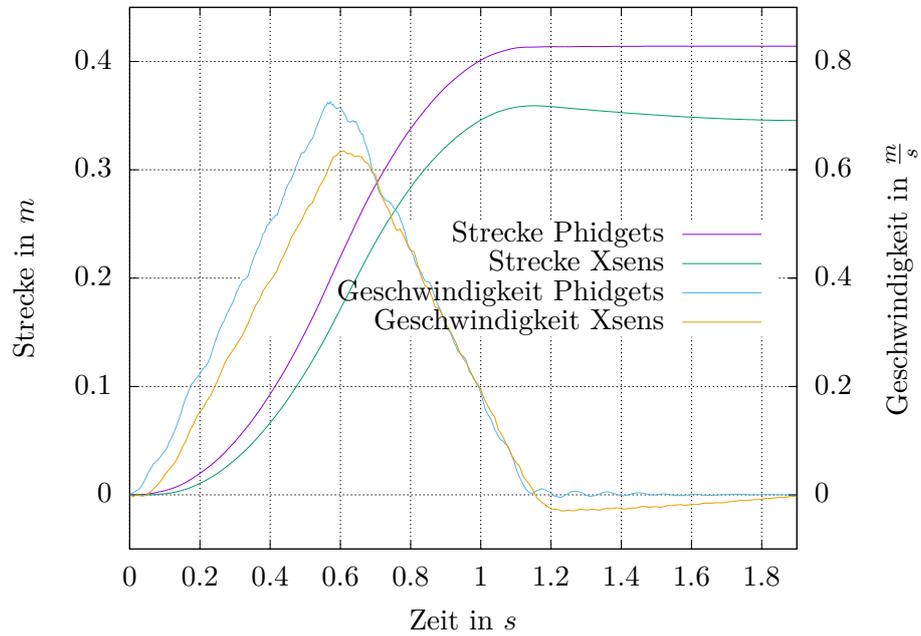


Abbildung 4.18: Geschwindigkeits- und Streckenprofil der Y-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich

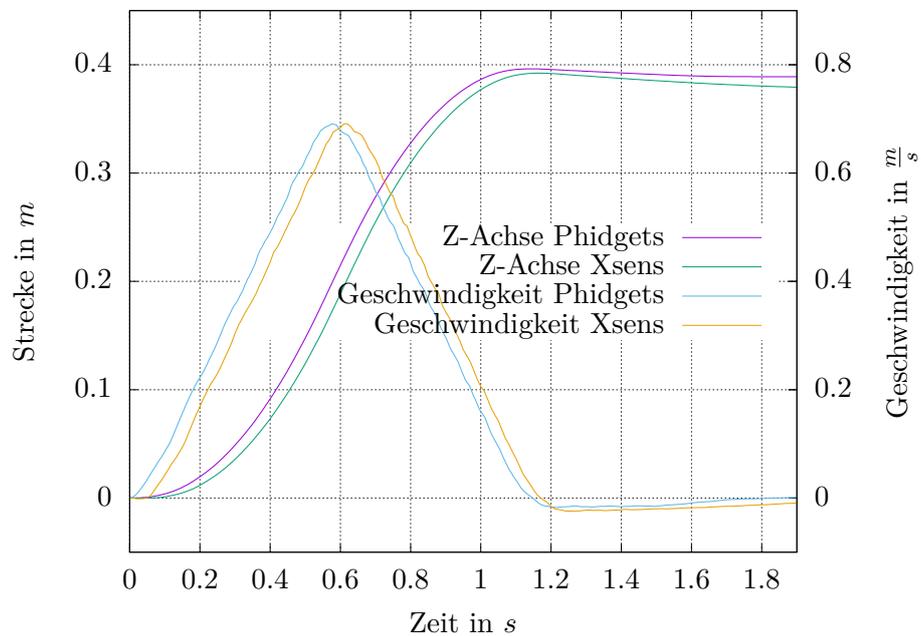


Abbildung 4.19: Geschwindigkeits- und Streckenprofil der Z-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich

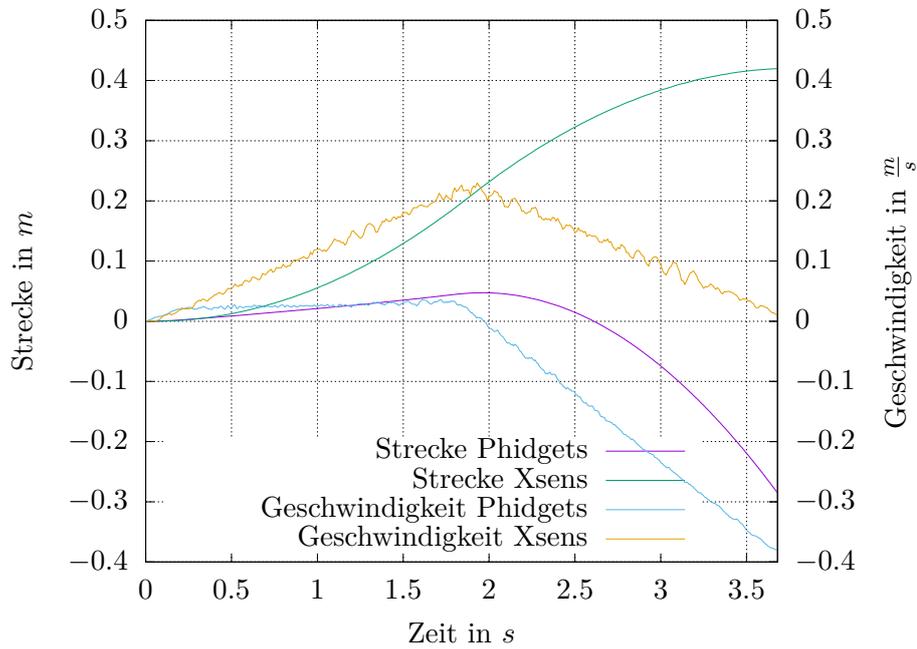


Abbildung 4.20: Geschwindigkeits- und Streckenprofil der X-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich

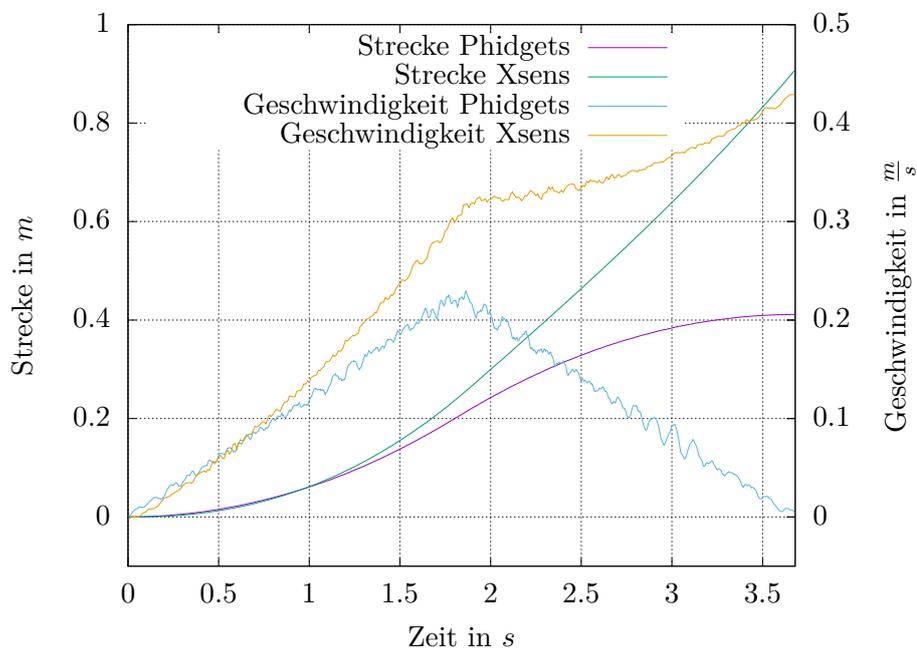


Abbildung 4.21: Geschwindigkeits- und Streckenprofil der Y-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich

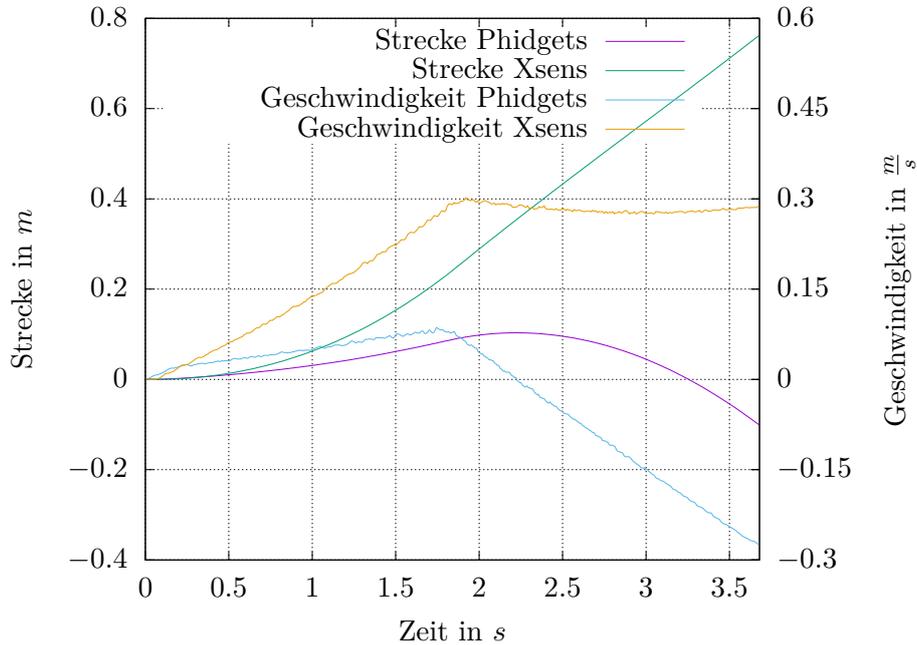


Abbildung 4.22: Geschwindigkeits- und Streckenprofil der Z-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich

Zusammenfassung

Vergleicht man die Ergebnisse fällt auf, dass bei der höheren Geschwindigkeit um einiges exaktere Ergebnisse erzielt werden. Sämtliche Achsen erreichen hierbei annähernd die gefahrene Strecke von $0.4m$ und der Drift der Geschwindigkeitswerte nach Bewegungsstopp für $t > 1.2s$ hält sich in Grenzen, bzw. ist teilweise gar nicht vorhanden.

Die Ergebnisse für die geringe Geschwindigkeit sind nahezu unbrauchbar, da die durch die IMUs aufgezeichneten Geschwindigkeiten stark driften. Lediglich in Abbildung 4.21 erreicht die Phidgets IMU 1 und in Abbildung 4.20 die Xsens IMU die tatsächlich gefahrene Strecke.

4.2.2 Volksbot-Roboter

Für die folgenden Messreihen wurden die Phidgets IMU 1, bzw. die Xsens IMU mittig auf der Verbindungslinie der beiden Vorderreifen des Volksbot-Roboters befestigt. Dabei wurde darauf geachtet, dass die X-Achse der jeweiligen IMU möglichst exakt in Fahrtrichtung ausgerichtet war, um nur mit dieser die Beschleunigungen der Bewegung des Volksbot-Roboters aufzuzeichnen.

Da sich der Roboter auf einer horizontalen Fläche bewegt, sind dementsprechend die Orientierungswerte *Pitch* und *Roll* konstant. Zu Beginn jeder Bewegungsphase wurden die Orientierungswerte des Madgwick-Filters mit $Pitch(0)$ und $Roll(0)$ gespeichert und während der Bewegungsphase konstant gehalten.

Yaw gibt in diesem Fall die Fahrtrichtung des Roboters an. Dieser Wert wurde gemäß Abschnitt 3.4.3 berechnet und sei $Yaw(t)$. Die Orientierung des Roboters Yaw_{Robot} wurde zu Beginn der Bewegungsphase als Anfangsorientierung $Yaw(0)$ gesetzt.

$Offset_x$ wurde berechnet, wie in Abschnitt 4.1.2 beschrieben ist, und blieb während der Bewegung konstant.

Die Geschwindigkeit $v_{x,IMU}$ wurde gemäß Abschnitt 3.4.6 bestimmt und die zurückgelegte Strecke im Roboter-Koordinatensystem wie folgt berechnet:

$$\begin{aligned} s_{x,Robot}(t) &= s_{x,Robot}(t-1) + v_{x,IMU}(t) \cdot \cos(Yaw(t)) \cdot \Delta t \\ s_{y,Robot}(t) &= s_{y,Robot}(t-1) + v_{x,IMU}(t) \cdot \sin(Yaw(t)) \cdot \Delta t \end{aligned}$$

Positionsdaten zum Vergleich erhält man direkt vom Volksbot-Roboter. Diese basieren wie in Abschnitt 3.3.2 beschrieben auf Odometrie, bzw. auf Laserscanner-Ortung.

In den folgenden Messungen fallen diese Positionsdaten teilweise unterschiedlich aus. Die Bewegungen des Roboters wurden pro IMU durchgeführt, da es nicht möglich war beide IMU gleichzeitig zu montieren. Durch Ungenauigkeiten des Roboter-Antriebs haben trotz identischen Bewegungsprofils die gefahrenen Trajektorien nicht exakt übereingestimmt.

Langsame geradlinige Bewegung

In dieser Messung wurde eine möglichst geradlinige Bewegung gefahren. Dabei wurde langsam beschleunigt und anschließend langsam abgebremst. Auf einen Vergleich der Daten mit den errechneten Positionsdaten der Laserscanner-Ortung wurde verzichtet, da diese zu fehlerhaft waren.

Die Abbildungen 4.23 und 4.24 zeigen die Positionsdiagramme der durch die Phidgets IMU 1, bzw. Xsens IMU, aufgenommenen Daten.

Im Vergleich fällt auf, dass die Xsens IMU der Fahrtrichtung des Roboters etwas besser folgt als die Phidgets IMU 1.

Der in Abbildung 4.24 sichtbare „Haken“ am Ende der Bewegung lässt sich mithilfe der durch die IMUs aufgezeichneten Geschwindigkeiten in Abbildung 4.25 erklären. Ab etwa $t = 5s$ zeichnen beide IMUs eine negative Geschwindigkeit auf. Die Ursache hierfür ist nicht bekannt, da sich der Volksbot-Roboter zu keinem Zeitpunkt rückwärts bewegt hat. Eine Vermutung könnte sein, dass die „ruckeligen“ Bewegungen des Roboters zu diesen Abweichungen führen.

Im Positionsdiagramm 4.23 der Phidgets IMU 1 ist dieser plötzliche Richtungswechsel auch vorhanden, jedoch aufgrund der Überlagerungen der Datenpunkt nur schwer zu erkennen.

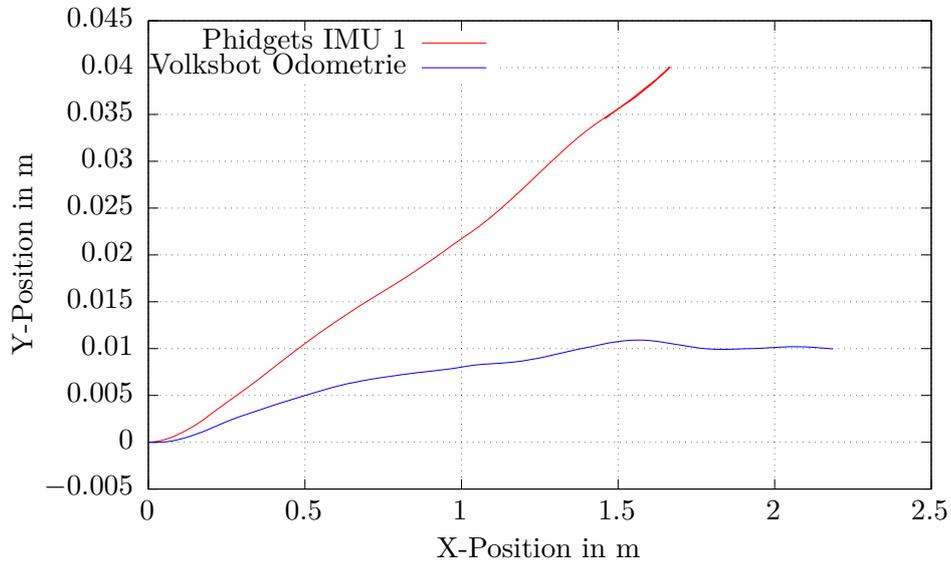


Abbildung 4.23: Durch Phidgets IMU 1 aufgezeichnete langsame geradlinige Bewegung

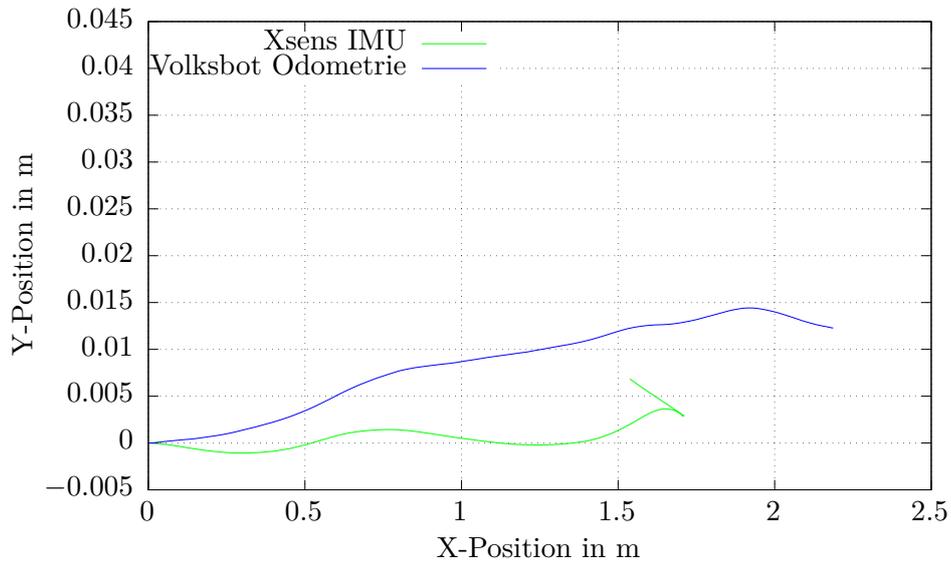


Abbildung 4.24: Durch Xsens IMU aufgezeichnete langsame geradlinige Bewegung

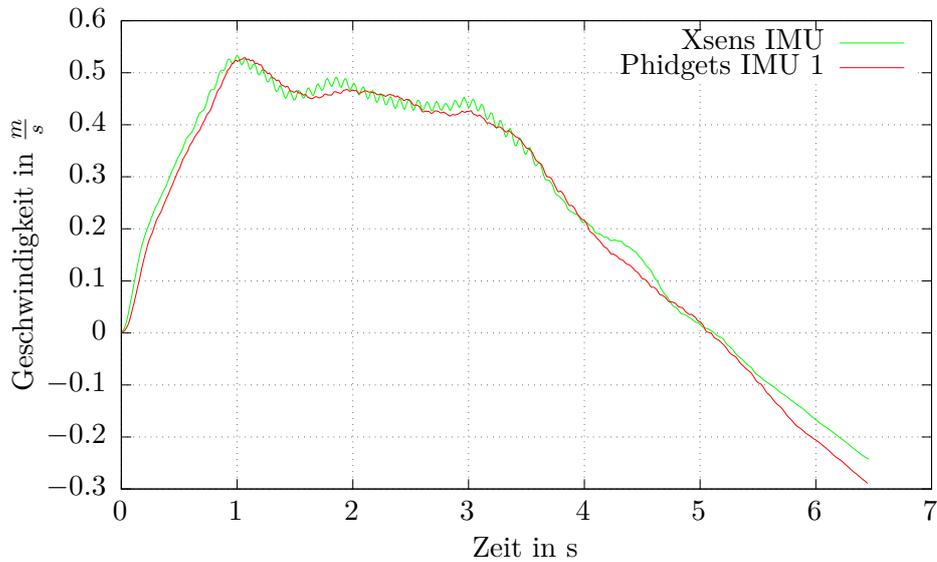


Abbildung 4.25: Geschwindigkeitsprofil der langsamen geradlinigen Bewegungen

Schnelle geradlinige Bewegung

In dieser Messung wurde eine möglichst geradlinige Bewegung gefahren. Dabei wurde stark beschleunigt und anschließend stark abgebremst. Auf einen Vergleich der Daten mit den errechneten Positionsdaten der Laserscanner-Ortung wurde auch verzichtet, da diese zu fehlerbehaftet waren.

Wie in Abschnitt 4.2.2 bereits gezeigt, ist auch in den Abbildungen 4.26 und 4.27 ersichtlich, dass die Xsens IMU der Fahrtrichtung des Roboters besser folgt.

Das Geschwindigkeitsprofil in Abbildung 4.28 zeigt, dass beide IMUs ab ca. $t = 5s$ eine negative Geschwindigkeit aufgezeichnet hatten, womit der plötzliche Richtungswechsel in den Abbildungen 4.26 und 4.27 erklärt werden kann.

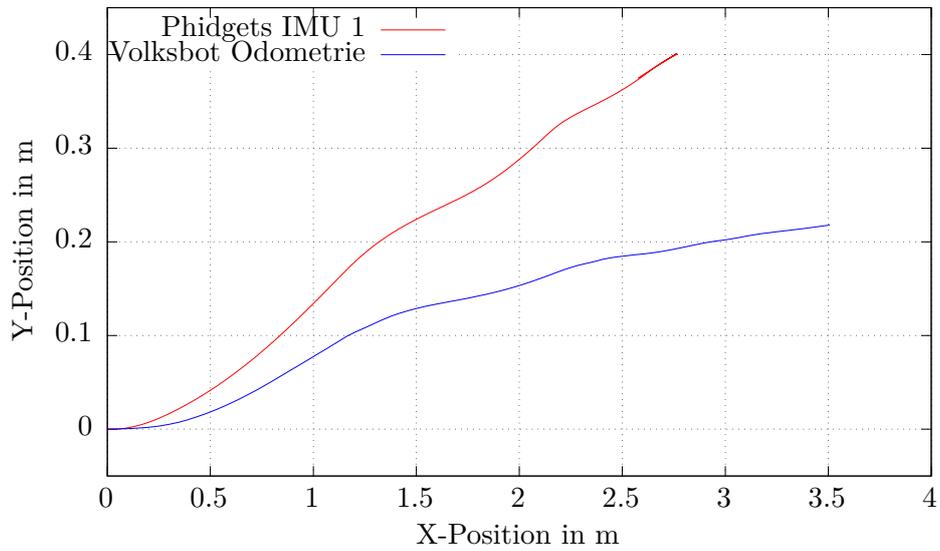


Abbildung 4.26: Durch Phidgets IMU 1 aufgezeichnete schnelle geradlinige Bewegung

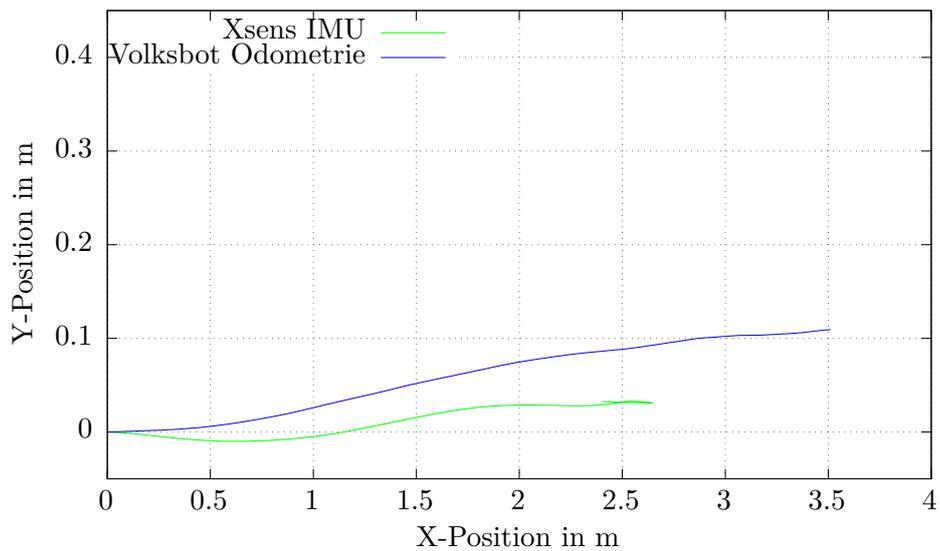


Abbildung 4.27: Durch Xsens IMU aufgezeichnete schnelle geradlinige Bewegung

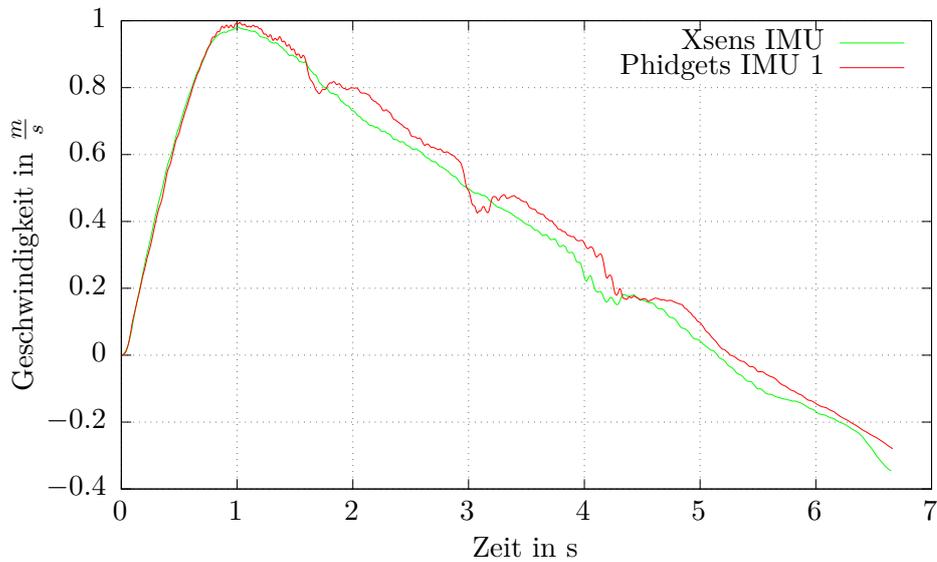


Abbildung 4.28: Geschwindigkeitsprofil der schnellen geradlinigen Bewegungen

Kurvenförmige Bewegung

In dieser Messung wurde eine kurvenförmige Bewegung gefahren. Auch hier wurde auf einen Vergleich der Daten mit den errechneten Positionsdaten der Laserscanner-Ortung verzichtet, da diese aufgrund großer Abweichungen keinen sinnvollen Vergleich zuließen.

Bei dieser Form der Bewegung kann die Phidgets IMU 1, wie in Abbildung 4.29 zu sehen ist, der Fahrtrichtung des Roboters besser folgen als die Xsens IMU, dargestellt in Schaubild 4.30.

In beiden Positionsdiagrammen ist der aus Abschnitt 4.2.2 bekannte „Haken“ zu sehen. Dieser kann auch wieder mit dem folgenden Geschwindigkeitsprofil erklärt werden. Beide IMUs zeichnen auch hier zum Ende der Bewegung unerklärliche negative Geschwindigkeiten auf, wie in Abbildung 4.31 dargestellt.

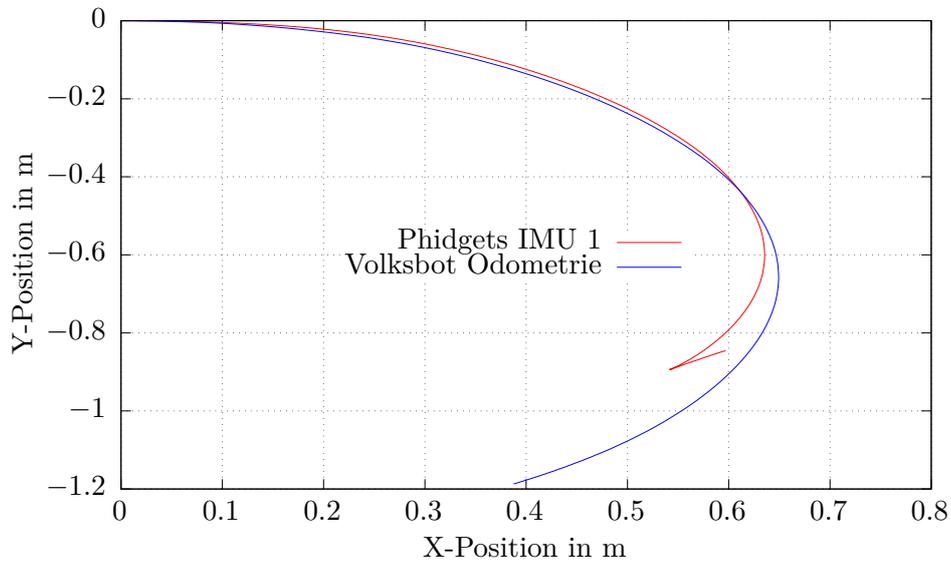


Abbildung 4.29: Durch Phidgets IMU 1 aufgezeichnete kurvenförmige Bewegung

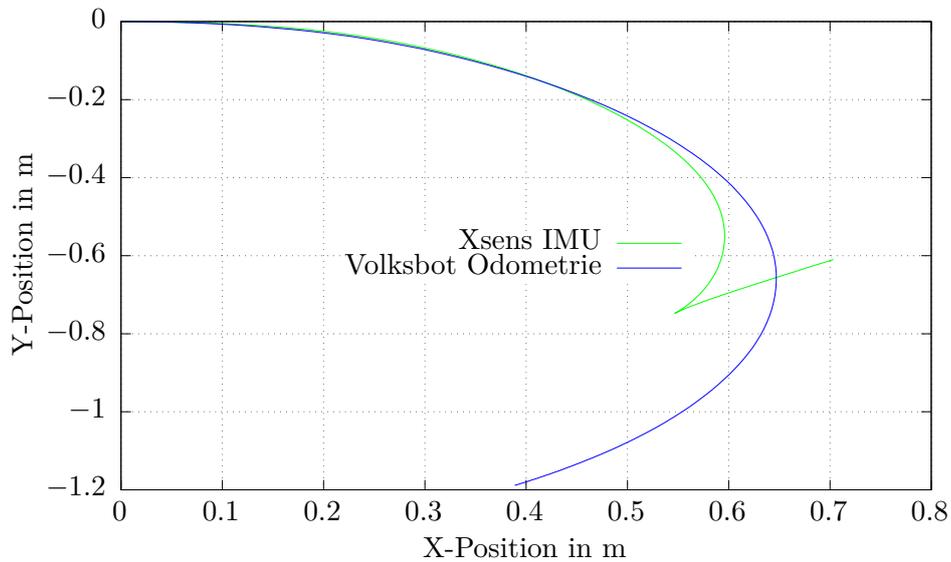


Abbildung 4.30: Durch Xsens IMU aufgezeichnete kurvenförmige Bewegung

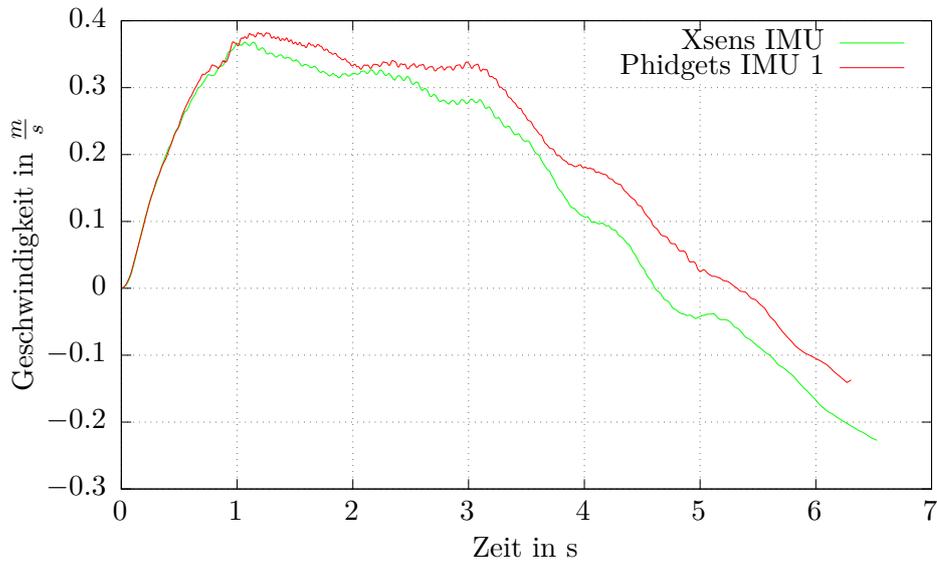


Abbildung 4.31: Geschwindigkeitsprofil der kurvenförmige Bewegungen

Kreisförmige Bewegung

In dieser Messung wurde eine kreisförmige Bewegung gefahren. Um die Odometrie-Positionsdaten verifizieren zu können, wurden die durch den AMCL-Algorithmus gewonnenen Positionsdaten zusätzlich in den folgenden Abbildungen dargestellt. Beim Vergleich der Positionsdiagramme 4.32 und 4.33 fällt direkt auf, dass die Positionsdaten, welche aus den Messwerten der Phidgets IMU 1 gewonnen wurden, um ein Vielfaches genauer sind als jene, welche von der Xsens IMU stammen.

In Schaubild 4.33 ist zudem der aus den vorherigen Messungen bekannte „Haken“ sichtbar, welcher mit dem Geschwindigkeitsprofil in Abbildung 4.34 erklärt werden kann. Die Positionsdaten der Phidgets IMU 1 folgten nicht nur besser der Fahrtrichtung des Roboters, sondern auch die berechnete Geschwindigkeit wurde exakter als jene der Xsens IMU gemessen.

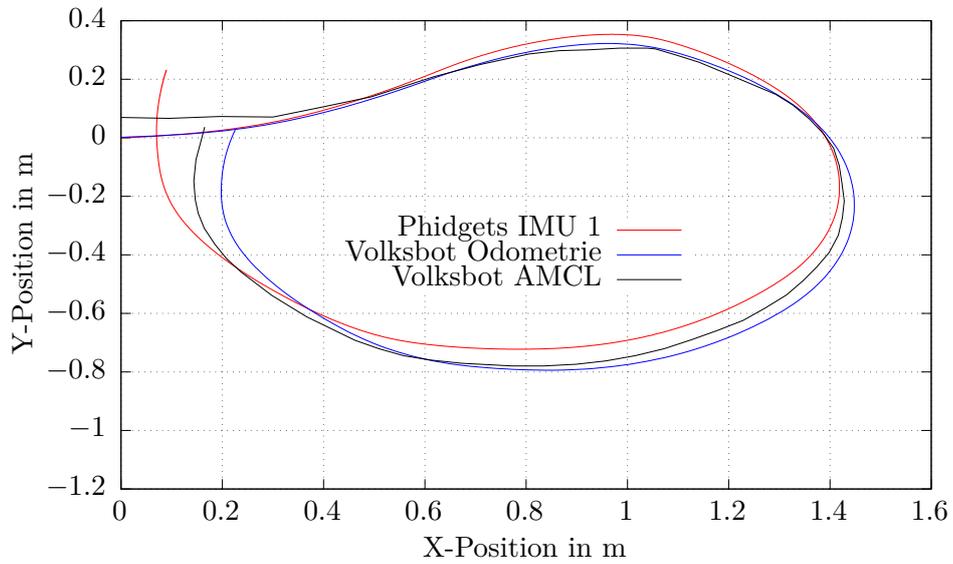


Abbildung 4.32: Durch Phidgets IMU 1 aufgezeichnete kreisförmige Bewegung

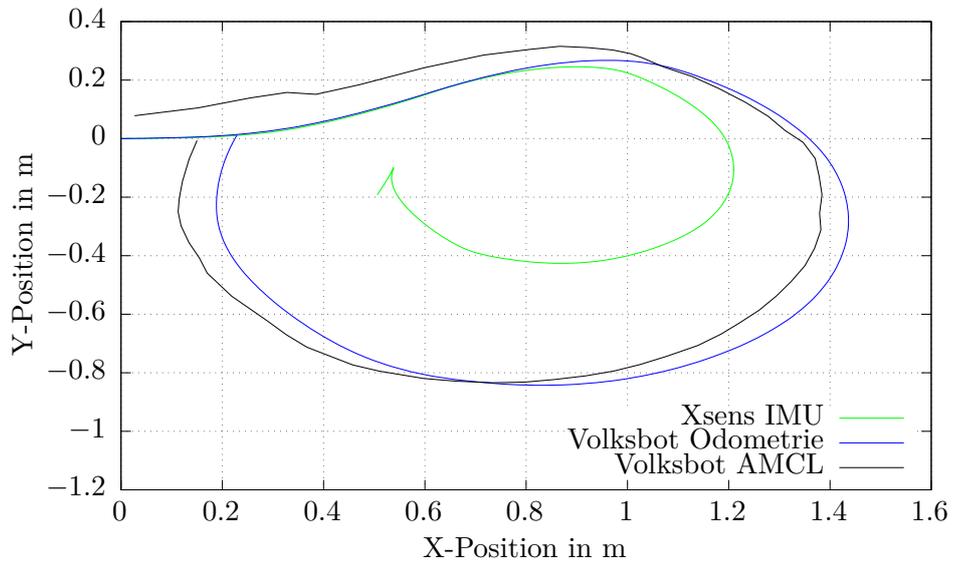


Abbildung 4.33: Durch Xsens IMU aufgezeichnete kreisförmige Bewegung

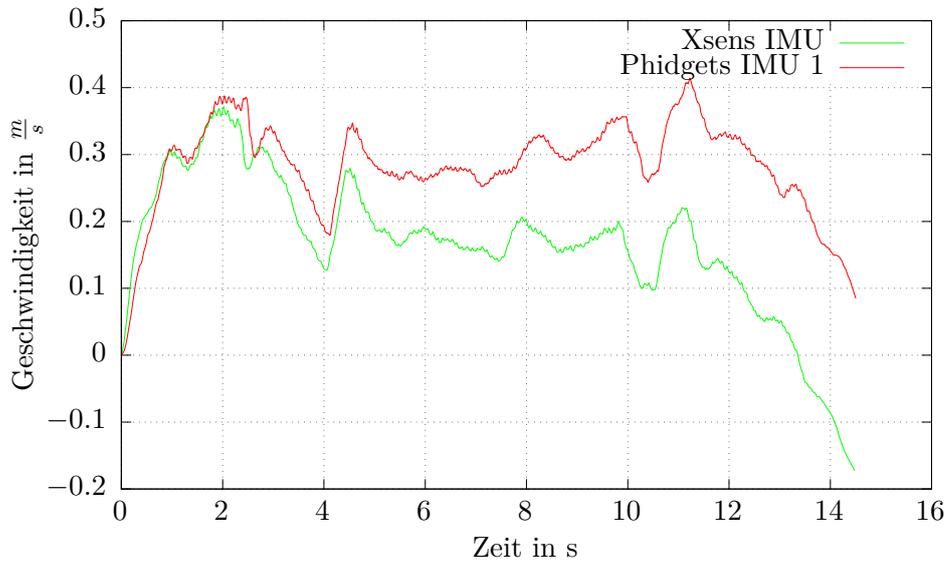


Abbildung 4.34: Geschwindigkeitsprofil der kreisförmigen Bewegungen

Zusammenfassung

Über alle Messungen hinweg kann gesagt werden, dass bei geradlinigen Bewegungen die Berechnung des *Yaw*-Winkels durch die Xsens IMU exakter ist. Diese kann der Fahrtrichtung des Roboters besser folgen als die Phidgets IMU 1. Im Gegensatz dazu erzielt die Phidgets IMU 1 bei kurven- und kreisförmigen Bewegungen genauere Orientierungswerte. Dies kann vor allem bei Vergleich der Abbildungen 4.33 und 4.32 beobachtet werden.

Zum Ende der Bewegungsphasen sollten die gemessenen Geschwindigkeiten im Optimalfall wieder $0 \frac{m}{s}$ erreichen. Abgesehen von der langsamen geradlinigen Bewegung kommen die Ergebnisse der Phidgets IMU 1 diesem Wert näher als die Ergebnisse der Xsens IMU.

Da die Elektromotoren des Roboters nur etwa $5cm$ von den IMUs entfernt waren, könnten diese einen zusätzlichen Fehler auf die Messungen ausgeübt haben.

Kapitel 5

Fazit

Ziel dieser wissenschaftlichen Arbeit war es, Sensormesswerte von verschiedenen IMUs auf Verlässlichkeit zu prüfen, um daraus resultierende Beschleunigungen, Geschwindigkeiten und zurückgelegte Strecken ableiten zu können.

Dabei wurde auf die Relevanz von exakten Orientierungsangaben eingegangen, welche unter anderem für die Berücksichtigung statischer Beschleunigungen, wie die Schwerkraft essentiell sind. Es wurde gezeigt, dass die Orientierungswerte des Madgwick-Filters während linearen Bewegungen nicht verlässlich genug sind und anstelle dieser eher auf Methoden zurückgegriffen werden sollte, welche ohne die direkte Verwendung von Beschleunigungsmesswerten auskommen, wie in Abschnitt 3.4.3 erläutert wurde.

Darüber hinaus wurde gezeigt, in welchem Maße äußere Störquellen Einfluss auf die Beschleunigungsmessung haben können. Dafür wurde in Abschnitt 4.1.2 der störende Einfluss eines KUKA-Roboters genauer untersucht mit dem überraschenden Ergebnis, dass die Beschleunigungssensoren zweier IMUs des Herstellers *Phidgets* gravierend beeinflusst werden. Ein nach demselben Messprinzip aufgebauter Beschleunigungssensor der Marke *Xsens* wurde entgegen aller Erwartungen von derselben Störquelle nicht beeinflusst. Im gleichen Abschnitt wurde auch eine Strategie entwickelt, um diesen Störeinfluss auf die Messwerte einer *Phidgets* IMU zu korrigieren.

Schlussendlich wurde die Genauigkeit der Messung von zurückgelegten Strecken und entsprechend aufgezeichneten Geschwindigkeiten untersucht, wofür zwei unterschiedliche Robotersysteme zur Verfügung standen. Die Ergebnisse fielen sehr unterschiedlich aus und waren abhängig von der auf die Sensoren ausgeübte Geschwindigkeit und der Methode der Orientierungs- und Offsetwertbestimmung.

Abschließend kann festgehalten werden, dass die im Rahmen der Bachelorarbeit gewonnenen Ergebnisse keine eindeutige Beurteilung über die Qualität der Messwerterfassung beider Sensorhersteller zulassen. Abhängig von eben genannten Faktoren lieferten einerseits die IMUs von *Phidgets*, andererseits die IMU von *Xsens* bessere Ergebnisse.

Die Untersuchungen zeigen auch, mit welcher Problematik bei Systemen zur Bestimmung zurückgelegter Strecken umgegangen werden muss, wenn lediglich auf Beschleunigungs-, Drehraten- und Magnetfeldsensoren zurückgegriffen wird.

5.1 Ausblick

Die vorgestellten Möglichkeiten zur Bewegungsbestimmung ohne externe Signale, wie zum Beispiel Licht oder Funk haben durchaus Potential Trajektoriedaten aufzunehmen. Insbesondere bei zeitlich kurzen, stark beschleunigten Bewegungen liefern diese Methoden gute Ergebnisse. Um jedoch verlässliche Ergebnisse zu erhalten, sollte weiter untersucht werden, warum die verschiedenen IMUs in unterschiedlichen Situationen nicht vorhersehbare und auch nicht zwangsläufig reproduzierbare Ergebnisse lieferten.

Glossar

AHRS Attitude and Heading Reference System 58

AMCL Adaptive Monte Carlo Localization 19, 53, 58

CAD Computer-Aided Design v, 17, 58

GNSS Global Navigation Satellite System iii, 58

IMU Inertial Measurement Unit vii, 3, 7, 13, 14, 17, 18, 21, 25, 26, 29, 31–34, 37, 38, 40, 43–49, 51, 53, 55, 57, 58, 63, 64

MARG Magnetic, Angular Rate, Gravity 7, 58

MEMS Microelectromechanical System 3, 4, 13, 58

ROS Roboter Operating System vii, 5, 6, 14, 20, 58

Literaturverzeichnis

- [1] *6Achse-Roboter Kuka KR16-2*. https://www.egsautomatisierung.de/s/cc_images/teaserbox_2455102697.png?t=1423730535. – Eingesehen am 19.08.2019
- [2] *Amcl*. <http://wiki.ros.org/amcl>. – Eingesehen am 19.08.2019
- [3] *Beschleunigungssensor*. <https://de.wikipedia.org/wiki/Beschleunigungssensor>. – Eingesehen am 29.07.2019
- [4] *Einführung in das Robot Operating System*. <https://www.heise.de/developer/artikel/Einfuehrung-in-das-Robot-Operating-System-3273655.html?seite=all>. – Eingesehen am 16.07.2019
- [5] *GMapping*. <http://wiki.ros.org/gmapping>. – Eingesehen am 19.08.2019
- [6] *Inertiale Messeinheit*. https://de.wikipedia.org/wiki/Inertiale_Messeinheit. – Eingesehen am 28.07.2019
- [7] *MEMS Accelerometer Gyroscope Magnetometer and Arduino*. <https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>. – Eingesehen am 12.08.2019
- [8] *MEMS-Gyroskop für genaue Drehratenmessungen in rauen Umgebungen bei hohen Temperaturen*. <https://www.channel-e.de/designcorner/artikel/article/mems-gyroskop-fuer-genaue-drehratenmessungen-in-rauen-umgebungen-bei-hohen-temperaturen.html>. – Eingesehen am 29.07.2019
- [9] *MEMS magnetic field sensor*. https://en.wikipedia.org/wiki/MEMS_magnetic_field_sensor. – Eingesehen am 29.07.2019
- [10] *Phidgets Treiber*. <http://wiki.ros.org/phidgets>. – Eingesehen am 19.08.2019
- [11] *PhidgetSpatial Precision 3/3/3 High Resolution*. <https://www.phidgets.com/?tier=3&catid=10&pcid=8&prodid=1038>. – Eingesehen am 19.08.2019
- [12] *Pitch, Roll, and Yaw*. <https://www.touringmachine.com/Articles/aircraft/6/>. – Eingesehen am 13.08.2019
- [13] *Quaternion*. <https://de.wikipedia.org/wiki/Quaternion>. – Eingesehen am 22.07.2019

-
- [14] *Quaternion differentiation.* <https://fgiesen.wordpress.com/2012/08/24/quaternion-differentiation/>. – Eingesehen am 24.07.2019
- [15] *Raspberry Pi 3 Model B+.* <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. – Eingesehen am 19.08.2019
- [16] *Raspberry Pi Images.* <https://downloads.ubiquityrobotics.com/pi.html>. – Eingesehen am 19.08.2019
- [17] *RASPBERRY PI – MODELLVERGLEICH.* <https://www.datenreise.de/raspberry-pi-unterschiede-zwischen-den-modellen/>. – Eingesehen am 19.08.2019
- [18] *Robot Operating System.* <http://wiki.ros.org/de>. – Eingesehen am 16.07.2019
- [19] *Robot Operating System.* <http://www.ros.org/about-ros/>. – Eingesehen am 16.07.2019
- [20] *Robot Operating System.* <http://www.ros.org/history/>. – Eingesehen am 16.07.2019
- [21] *Roboter Operating System.* https://de.wikipedia.org/wiki/Robot_Operating_System. – Eingesehen am 16.07.2019
- [22] *ROS-Package Madgwick-Filter.* http://wiki.ros.org/imu_filter_madgwick. – Eingesehen am 25.08.2019
- [23] *ROS – Robot Operating System.* <https://www.generationrobots.com/blog/de/ros-robot-operating-system/>. – Eingesehen am 16.07.2019
- [24] *Schwerefeld.* <https://de.wikipedia.org/wiki/Schwerefeld>. – Eingesehen am 08.08.2019
- [25] *Understanding Euler Angles.* <http://www.chrobotics.com/library/understanding-euler-angles>. – Eingesehen am 23.07.2019
- [26] *Understanding Quaternions.* <http://www.chrobotics.com/library/understanding-quaternions>. – Eingesehen am 23.07.2019
- [27] *Xsens MTi-G-700.* <https://shop.xsens.com/shop/mti-g-710/mti-g-710-gnssins-2a8g4>. – Eingesehen am 19.08.2019
- [28] *Xsens Treiber.* http://wiki.ros.org/xsens_driver. – Eingesehen am 19.08.2019
- [29] LESJAK, Roman: *Die Rolle einer inertialen Messeinheit in der Anwendung Moving-Base Gravimetry.* September 2010. – Eingesehen am 28.07.2019
- [30] MADGWICK, Sebastian O.: *An efficient orientation filter for inertial and inertial/magnetic sensor arrays.* Proceedings of the 5th Symposium on Intelligent Autonomous Vehicles, April 2010. – Eingesehen am 22.07.2019

Abbildungsverzeichnis

2.1	Funktionsweise Drehratensensor (Quelle: [8])	4
2.2	Definition der Euler-Winkel <i>Roll</i> , <i>Pitch</i> und <i>Yaw</i> (Quelle: [12])	9
3.1	Blockschaltbild des Messaufbaus	14
3.2	Übersicht Plattform Unterseite	15
3.3	Übersicht Plattform Oberseite	15
3.4	Konstruktionszeichnung des lilafarbenen Bauteils im Maßstab 2 : 1	16
3.5	Konstruktionszeichnung des blauen Bauteils im Maßstab 1 : 1	16
3.6	KUKA-Roboter (Quelle: [1])	18
3.7	Volksbot-Roboter	19
4.1	Roll und Pitch des Madgwick-Filters während einer Beschleunigung in Z-Richtung der Phidgets IMU 1	26
4.2	Roll und Pitch des Madgwick-Filters während einer Beschleunigung in X-Richtung der Phidgets IMU 1	26
4.3	Vergleich der zurückgelegten Strecke bei einer Halbkreisbewegung mit maximaler Geschwindigkeit unter Verwendung der Orientierungswerte des Madgwick-Filters, bzw. des Gyroskops	27
4.4	Vergleich der zurückgelegten Strecke bei einer Halbkreisbewegung mit halber Maximalgeschwindigkeit unter Verwendung der Orientierungswerte des Madgwick-Filters, bzw. des Gyroskops	28
4.5	Schwingung der Beschleunigungsmesswerte nach Bewegungsstopp mit daraus resultierender Geschwindigkeit	29
4.6	Unterschiedliche Ausrichtungen des KUKA-Roboters	30
4.7	Orientierungsabhängige Gravitationsmessung mit verschiedenen IMUs	32
4.8	Offsetwerte der X-Sensor-Achse	33
4.9	Offsetwerte der Y-Sensor-Achse	34
4.10	Offsetwerte der Z-Sensor-Achse	35
4.11	Offsetwerte der Phidgets IMU 1	38
4.12	Offsetwerte der Phidgets IMU 2	39
4.13	Offsetwerte der Xsens IMU	39
4.14	Streckenprofil der Halbkreisbewegung mit 30% Geschwindigkeit	41
4.15	Streckenprofil der Halbkreisbewegung mit 50% Geschwindigkeit	41

4.16	Streckenprofil der Halbkreisbewegung mit 75% Geschwindigkeit	42
4.17	Geschwindigkeits- und Streckenprofil der X-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich	43
4.18	Geschwindigkeits- und Streckenprofil der Y-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich	44
4.19	Geschwindigkeits- und Streckenprofil der Z-Achse bei 100% Geschwindigkeit beider IMUs im Vergleich	44
4.20	Geschwindigkeits- und Streckenprofil der X-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich	45
4.21	Geschwindigkeits- und Streckenprofil der Y-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich	45
4.22	Geschwindigkeits- und Streckenprofil der Z-Achse bei 30% Geschwindigkeit beider IMUs im Vergleich	46
4.23	Durch Phidgets IMU 1 aufgezeichnete langsame geradlinige Bewegung	48
4.24	Durch Xsens IMU aufgezeichnete langsame geradlinige Bewegung	48
4.25	Geschwindigkeitsprofil der langsamen geradlinigen Bewegungen	49
4.26	Durch Phidgets IMU 1 aufgezeichnete schnelle geradlinige Bewegung	50
4.27	Durch Xsens IMU aufgezeichnete schnelle geradlinige Bewegung	50
4.28	Geschwindigkeitsprofil der schnellen geradlinigen Bewegungen	51
4.29	Durch Phidgets IMU 1 aufgezeichnete kurvenförmige Bewegung	52
4.30	Durch Xsens IMU aufgezeichnete kurvenförmige Bewegung	52
4.31	Geschwindigkeitsprofil der kurvenförmigen Bewegungen	53
4.32	Durch Phidgets IMU 1 aufgezeichnete kreisförmige Bewegung	54
4.33	Durch Xsens IMU aufgezeichnete kreisförmige Bewegung	54
4.34	Geschwindigkeitsprofil der kreisförmigen Bewegungen	55

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Würzburg, 29. August 2019