



Universität
Bremen

Fachbereich 3 - Mathematik und Informatik

Eine Maus nur mit einer inertialen Messeinheit

Bachelorarbeit

Name:	Merlin Kannengießer
Matrikelnummer:	6034915
Datum:	16.07.2026
1. Gutachter:	Prof. Dr.-Ing. Udo Frese
2. Gutachterin:	Dr.-Ing. Nina Wenig

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen und Stand der Forschung	2
2.1	Bewegungserfassung bei einer optischen Maus	2
2.2	Inertiale Messeinheit	2
2.2.1	Accelerometer	2
2.2.2	Gyroskop	3
2.3	Von Beschleunigung zu Position	3
2.4	Bias, Rauschen und Drift	4
2.4.1	Bias und Biasinstabilität	4
2.4.2	Rauschen und Drift	4
2.5	Verwandte Arbeiten	4
3	Fragestellung und Untersuchungsdesign	6
4	Voruntersuchung mit der MPU6050	7
4.1	Versuchsaufbau und Datenerfassung	7
4.2	Ableitung einer Referenzbeschleunigung	7
4.3	Statische Ausrichtung des Accelerometers	11
4.4	Integration und Ruheerkennung	11
4.5	Zwischenergebnis	13
5	Entwicklung der IMU-Maus	15
5.1	Hardwareauswahl	15
5.2	Schaltung und mechanischer Aufbau	15
5.3	Firmware	18
5.3.1	Startkalibrierung	18
5.3.2	Hauptschleife und Zeitmessung	18
5.3.3	Drehung des Beschleunigungsvektors	19
5.3.4	IIR-Tiefpass	19
5.3.5	Integration und USB-HID-Ausgabe	19
5.4	Zusammenfassung des Algorithmus	20
6	User Study	21
6.1	Aufgabe und Anwendung	21
6.2	Messgrößen	21
6.3	Ablauf	21
6.4	Teilnehmende	23
7	Ergebnisse	25
7.1	Bearbeitungszeit	25
7.2	Overshoots	25

7.3	Fragebogen	26
7.4	Freitextfeedback	27
8	Diskussion	28
8.1	Beantwortung der Forschungsfragen	28
8.2	Warum schnelle Bewegungen besser funktionieren	28
8.3	Einfluss der Implementierung	29
8.4	Mögliche Verbesserungen	29
9	Zusammenfassung	30
10	Quellen- und Literaturverzeichnis	31
11	Nutzung KI-basierter Anwendungen	32

1 Einleitung

Die Computermouse ist trotz neuerer Eingabemethoden wie Touchpads und Touchscreens weiterhin eines der wichtigsten Eingabegeräte für die Bedienung von Computern. Herkömmliche Mäuse erfassen ihre Bewegung heute in der Regel mit einem optischen Sensor, welcher die Oberfläche unter der Maus betrachtet und aus den Bildern eine relative Verschiebung berechnet [vgl. 4, S. 2]. Die Nutzung eines solchen optischen Sensors ist günstig und auf geeigneten Oberflächen wie einem Mauspad sehr präzise. Die Oberfläche muss jedoch genügend optisch erkennbare Struktur liefern. Auf transparenten, stark spiegelnden oder sehr gleichförmigen Flächen kann die Erfassung beeinträchtigt sein. Die Bedienung der Maus auf Glas ist damit beispielsweise praktisch nicht möglich. Ein Mauspad liefert in solchen Fällen Abhilfe, ist aber ein zusätzliches Hilfsmittel, auf welches die Maus oft angewiesen ist.

Eine inertielle Messeinheit, im Folgenden IMU genannt, hat diese Einschränkung nicht: Sie betrachtet die Unterlage nicht, sondern misst die Beschleunigung und Drehrate des Geräts. Könnte aus diesen Größen die Bewegung einer Maus zuverlässig rekonstruiert werden, wäre die Maussteuerung unabhängig von den optischen Eigenschaften der Oberfläche. Eine solche Maus könnte auf vielen Unterlagen eingesetzt werden, auf denen ein optischer Sensor Schwierigkeiten hat. Gleichzeitig ist die Umsetzung nicht trivial, weil die Position nicht direkt gemessen wird, sondern aus den rausch- und fehlerbehafteten Beschleunigungswerten durch zweifache Integration berechnet wird.

Ziel dieser Arbeit ist die praktische Untersuchung, ob sich eine Computermouse mit einer günstigen IMU steuern lässt und wie zuverlässig ein entsprechender Prototyp im Vergleich zu einer herkömmlichen optischen Maus funktioniert. Dazu wurde zunächst mit einer MPU6050-IMU durch gleichzeitige Bewegung mit einer Maus untersucht, wie gut sich aufgezeichnete Mausbewegungen aus IMU-Daten rekonstruieren lassen. Auf Grundlage dieser Voruntersuchung entstand anschließend ein eigenständiger Mausprototyp mit einer LSM6DS3-IMU und einem Arduino Pro Micro. Abschließend wurde der Prototyp in einer User Study mit einer baugleichen optischen Vergleichsmouse getestet.

Die Arbeit untersucht damit nicht, ob die Positionsbestimmung unter idealen Bedingungen prinzipiell möglich ist, sondern ob ein kompakter und kostengünstiger Aufbau mit einfacher Echtzeitverarbeitung für eine alltagstaugliche Maus ausreicht. Im Rahmen einer User-Study erfolgt eine Bewertung anhand technischer Messungen bei einer Zielerfassungsaufgabe sowie die subjektive Einschätzung der Teilnehmenden.

2 Grundlagen und Stand der Forschung

2.1 Bewegungserfassung bei einer optischen Maus

Eine optische Maus misst die relative Bewegung eines Sensors gegenüber der Unterlage. Vereinfacht arbeitet der Sensor wie eine schnell abtastende Kamera. Durch den Vergleich aufeinanderfolgender Aufnahmen wird eine Verschiebung in zwei Achsen geschätzt [vgl. 4, S. 2]. Das Ergebnis sind relative Änderungen der Position, welche an den Computer zur Bewegung des Mauscursors gesendet werden. Dabei ist keine doppelte zeitliche Integration notwendig, wodurch Messfehler in der Regel lokal begrenzt bleiben und sich nicht wie bei einer inertialen Positionsschätzung akkumulieren.

Die gemessene Bewegung hängt nicht nur von der Translation des Mausgehäuses ab. Beim Bedienen wird die Maus zusätzlich um Handgelenk und Arm gedreht. Kim et al. zeigen für optische Mäuse, dass die Position des Sensors im Gehäuse deshalb beeinflusst, wie sich Drehungen der Maus auf die Mauscursorbewegung auswirken. In ihrer Studie war eine zentrale Sensorposition im Mittel der beste Kompromiss. Ein weiter vorne platzierter Sensor führte bei einer Drehbewegung zu deutlich größerer horizontaler Zeigerverschiebung als ein weiter hinten platzierter Sensor [vgl. 4, S. 1]. Die Sensortechnologie dieser Arbeit ist eine andere, die Beobachtung bleibt jedoch relevant: Auch eine IMU misst an ihrem konkreten Einbauort, sodass Rotationen je nach Abstand zum momentanen Drehzentrum unterschiedliche lineare Beschleunigungen hervorrufen können.

2.2 Inertiale Messeinheit

Eine IMU kombiniert mehrere inertielle Sensoren in einem Bauteil. Die hier verwendeten Module MPU6050 und LSM6DS3 enthalten jeweils ein dreiaxsiges Accelerometer und ein dreiaxsiges Gyroskop. Damit stehen pro Messzeitpunkt drei Beschleunigungskomponenten und drei Drehratenkomponenten zur Verfügung [vgl. 3, S. 7, vgl. 6, S. 1].

2.2.1 Accelerometer

Nach der Terminologie des IEEE ist ein lineares Accelerometer ein inertialer Sensor, der entlang seiner Eingangsachsen die translatorische Beschleunigung abzüglich des Gravitationsanteils erfasst [vgl. 2, S. 22]. Unter idealen Bedingungen erscheinen bei einem ruhenden Sensor auf einer horizontalen Fläche etwa $1g$ auf der vertikalen z-Achse, während die beiden horizontalen x- und y-Achsen $0g$ liefern. Das Datenblatt der LSM6DS3 beschreibt diesen Ruhezustand und bezeichnet Abweichungen als Zero-g-Offset [vgl. 6, S. 28].

Für eine Maus sind vor allem die horizontalen Komponenten relevant. Das Problem besteht darin, sehr kleine vom Nutzer induzierte Beschleunigungen von Offset, Rauschen und

Anteilen der Erdbeschleunigung zu trennen. Bereits eine geringe Verkipfung des Sensors projiziert einen Teil von g auf die x- oder y-Achse. Dieser Anteil kann wesentlich größer sein als die Beschleunigung einer langsamen Mausbewegung.

2.2.2 Gyroskop

Ein Gyroskop ist nach IEEE ein inertialer Sensor, der eine Winkelrotation relativ zum Inertialraum um seine Eingangsachsen misst [vgl. 2, S. 19]. Moderne Gyroskope liefern in der Regel die Drehrate. Durch zeitliche Integration kann daraus eine Winkeländerung berechnet werden. In dieser Arbeit wird die Drehrate um die z-Achse genutzt, weil eine Maus bei normaler Bedienung auf der Tischfläche häufig um diese Achse gedreht wird.

Die Drehrate wird verwendet, um den gemessenen Beschleunigungsvektor so um die z-Achse zu drehen, dass die Beschleunigung bestmöglich der Translation auf der Unterlage in Weltkoordinaten entspricht. Dies führt zwar zu einem anderen Verhalten als bei einer optischen Maus, ist jedoch notwendig, um sicherzustellen, dass sich die Beschleunigung nach Ende einer Mausbewegung zu null aufsummiert.

2.3 Von Beschleunigung zu Position

Aus einer bekannten Beschleunigung $\mathbf{a}(t)$ ergeben sich Geschwindigkeit und Position durch Integration:

$$\mathbf{v}(t) = \mathbf{v}(t_0) + \int_{t_0}^t \mathbf{a}(\tau) d\tau, \quad (2.1)$$

$$\mathbf{s}(t) = \mathbf{s}(t_0) + \int_{t_0}^t \mathbf{v}(\tau) d\tau. \quad (2.2)$$

Für diskrete Messwerte wurde in dieser Arbeit eine einfache schrittweise Integration verwendet:

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{a}_k \Delta t_k, \quad (2.3)$$

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \mathbf{v}_k \Delta t_k. \quad (2.4)$$

Dass ein Beschleunigungswert zweimal integriert werden muss, um eine Position zu erhalten, ist auch die Grundlage des von Lee et al. vorgeschlagenen Systems für eine unterlagenunabhängige Maus [vgl. 5, S. 8].

2.4 Bias, Rauschen und Drift

2.4.1 Bias und Biasinstabilität

Der IEEE-Standard definiert den Bias als den über einen festgelegten Zeitraum gemittelten Sensorausgang, der nicht mit der tatsächlichen Eingangsbewegung korreliert. Beim Accelerometer wird er als Beschleunigung, beim Gyroskop üblicherweise als Winkelrate angegeben. Der Standard beschreibt außerdem die Biasinstabilität, also eine zufällige zeitliche Veränderung dieses Offsets [vgl. 2, S. 13].

Ein konstanter Beschleunigungsbias ε führt nach zweimaliger Integration zu einem Positionsfehler, der quadratisch mit der Zeit wächst [vgl. 7, S. 15]:

$$\mathbf{e}_s(t) = \varepsilon \cdot \frac{t^2}{2}. \quad (2.5)$$

Zudem können Flickerrauschen und Temperaturänderungen den Bias über die Zeit wandern lassen [vgl. 7, S. 15 f.]. Eine einmalige Kalibrierung zu Beginn kann daher nur den zu diesem Zeitpunkt beobachteten Offset kompensieren.

2.4.2 Rauschen und Drift

Zufälliges Messrauschen hat idealerweise einen Mittelwert von null. Durch die Integration heben sich die einzelnen Fehler dennoch nicht vollständig auf, da sich die Fehler mit der Zeit ändern können. Hinzu kommen Orientierungsfehler: Bleibt die Maus nicht mit konstanter Orientierung bezüglich der x- und y-Achse auf der Oberfläche liegen, gelangt ein Anteil der Erdbeschleunigung in die horizontalen Achsen. Bei einer Maus reichen schon kleine Winkelabweichungen aus, um die eigentliche Bewegung zu überdecken.

Im Folgenden bezeichnet Drift die zunehmende Abweichung zwischen der aus der IMU berechneten Position und der tatsächlichen Mausposition. Drift entsteht aus mehreren Faktoren wie Offset, Biasänderung, Rauschen, unvollständiger Gravitationskompensation, Integrationsfehlern und gegebenenfalls falscher Ruheerkennung.

2.5 Verwandte Arbeiten

Lee et al. entwarfen ein zweidimensionales MEMS-System (mikro-elektromechanisches System) für eine Maus ohne zusätzliche Unterlage. Die Architektur misst Beschleunigungen in zwei Achsen, filtert sie und berechnet durch Integration Bildschirmkoordinaten [vgl. 5, S. 1]. Zur Filterung hoher Frequenzen, die nicht durch Bedienung des Nutzers entstehen, wird dort ein FIR-Tiefpassfilter mit einer Frequenz von ungefähr 100 Hz vorgeschlagen [vgl. 5, S. 7]. Der Schwerpunkt liegt auf der praktischen Sensordatenverarbeitung und dem Mikroprozessorentwurf. Die Arbeit geht jedoch nicht auf die Auswirkung von Bias und Drift auf die Nutzbarkeit der Maus ein und erwähnt die Begriffe nur am Rande. Ein Vergleich mit einer optischen Maus wird ebenfalls nicht durchgeführt.

Barbosa Tarazona et al. entwickelten eine 5-DOF-Eingabe, bei der ein Gerät in der Luft bewegt und über USB als Maus verwendet wird. Zur Schätzung eines Neigungswinkels wurde ein modifizierter Kalman-Filter verwendet, um die Sensordaten von Accelerometer und Gyroskop zu fusionieren [vgl. 1, S. 1 f.]. Zur Kompensation des Gyroskop-Offsets berechnen sie beim Gerätestart den Durchschnitt der ersten 5000 Messwerte und bestimmen daraus einen Offset, der anschließend fortlaufend abgezogen wird [vgl. 1, S. 3]. Eine ähnliche Kalibrierung beim Gerätestart wird auch in dieser Arbeit verwendet. Das Prinzip des dort entworfenen Gerätes unterscheidet sich jedoch, weil dort vor allem Neigung und Rotation eines frei gehaltenen Geräts ausgewertet werden, während hier eine zweidimensionale Wegbewegung auf einer Oberfläche betrachtet wird.

Die Sensorfusion mit einem Kalman-Filter kann in so einem Fall sinnvoll sein, da sowohl Accelerometer als auch Gyroskop relevante Messwerte für die Orientierungsbestimmung liefern. Für eine zweidimensionale Position wie in dieser Arbeit ist der Nutzen eines Kalman-Filters jedoch deutlich eingeschränkt, da das Gyroskop abgesehen von der Rotation während der Bewegung kaum hilfreiche Informationen liefert. Ein eindimensionaler Kalman-Filter kann zwar Rauschen reduzieren, jedoch ist dies auch mit einem Tiefpassfilter erreichbar.

3 Fragestellung und Untersuchungsdesign

Aus der Motivation und den beschriebenen Fehlermechanismen ergeben sich zwei Forschungsfragen:

- F1:** Ist es möglich, eine Maus nur mit einer inertialen Messeinheit zu bauen, die praktisch nutzbar und präzise genug ist?
- F2:** Wie unterscheidet sich die Zielgenauigkeit und die subjektive Bewertung einer solchen IMU-Maus im Vergleich zu einer herkömmlichen Maus?

Die Untersuchung besteht aus zwei aufeinander aufbauenden Phasen. In der ersten Phase werden die Daten einer MPU6050 gemeinsam mit den tatsächlichen Cursorpositionen einer optischen Maus aufgezeichnet. Die Cursorgeschwindigkeit und -position werden aus der IMU-Beschleunigung berechnet und mit den tatsächlichen Mausdaten verglichen. Durch Zurücksetzung der Geschwindigkeit in Ruhephasen wird versucht, die berechnete IMU-Position an die tatsächliche Mausposition anzugleichen.

In der zweiten Phase wird ein eigenständiger Prototyp entwickelt. Er verarbeitet die Sensordaten auf einem Arduino Pro Micro und sendet relative USB-HID-Mausbewegungen. Die Platine des Prototyps wird im Gehäuse einer Logitech-G400s-Maus verbaut. Die Funktionsfähigkeit wird anschließend mit einer optischen Logitech MX518 verglichen, welche dieselbe Gehäuseform wie die Logitech G400s hat. Durch die gleiche Form der Maus sollen ergonomische Unterschiede zwischen den beiden Geräten reduziert werden.

Die Auswertung ist deskriptiv. Die Stichprobe ist klein und lässt sich nicht auf die Gesamtbevölkerung übertragen. Entscheidend ist deshalb die gemeinsame Betrachtung technischer Beobachtungen, die Auswertung der Messergebnisse und qualitativer Rückmeldungen.

4 Voruntersuchung mit der MPU6050

4.1 Versuchsaufbau und Datenerfassung

Für die Voruntersuchung wurde eine MPU6050 auf einem Breadboard mit einem Arduino Mega verbunden. Die MPU6050 kombiniert ein dreiachsiges Accelerometer und ein dreiachsiges Gyroskop, digitalisiert beide Sensorgruppen mit 16 Bit und unterstützt für die Accelerometerbereiche unter anderem $\pm 2g$, $\pm 4g$, $\pm 8g$ und $\pm 16g$ [vgl. 3, S. 7, 10]. Abbildung 4.1 zeigt die Testschaltung.

Ein Python-Skript erfasste gleichzeitig die Cursorkoordinaten des Computers und die über eine serielle Verbindung übertragenen IMU-Messwerte. Während einer Aufnahme wurde das IMU-Modul an die optische Maus gehalten, sodass beide Geräte näherungsweise dieselbe Bewegung ausführten. Der Aufbau ist in Abbildung 4.2 dargestellt.

Die gespeicherten Daten umfassten Zeitstempel, Mausposition, Accelerometerwerte und Gyroskopwerte. Die Rohdaten dienten anschließend als Eingabe eines Jupyter-Notebooks.

4.2 Ableitung einer Referenzbeschleunigung

Die optische Maus liefert eine Position in Bildschirmpixeln. Um sie mit den IMU-Werten vergleichen zu können, wurden aus aufeinanderfolgenden Positionen zunächst Geschwindigkeit und anschließend Beschleunigung durch Differenzenbildung berechnet:

$$\mathbf{v}_k^{\text{Maus}} \approx \frac{\mathbf{p}_k^{\text{Maus}} - \mathbf{p}_{k-1}^{\text{Maus}}}{\Delta t_k}, \quad (4.1)$$

$$\mathbf{a}_k^{\text{Maus}} \approx \frac{\mathbf{v}_k^{\text{Maus}} - \mathbf{v}_{k-1}^{\text{Maus}}}{\Delta t_k}. \quad (4.2)$$

Die so erhaltenen Größen sind keine physikalische Beschleunigung in m/s^2 . Physikalische Größen sind für den Vergleich jedoch nicht nötig, da die Werte nur grob mit der Accelerometerbeschleunigung und berechneten Geschwindigkeit verglichen werden und die genauen Werte hierfür keine hohe Relevanz haben. Abbildung 4.4 zeigt die abgeleiteten Größen.

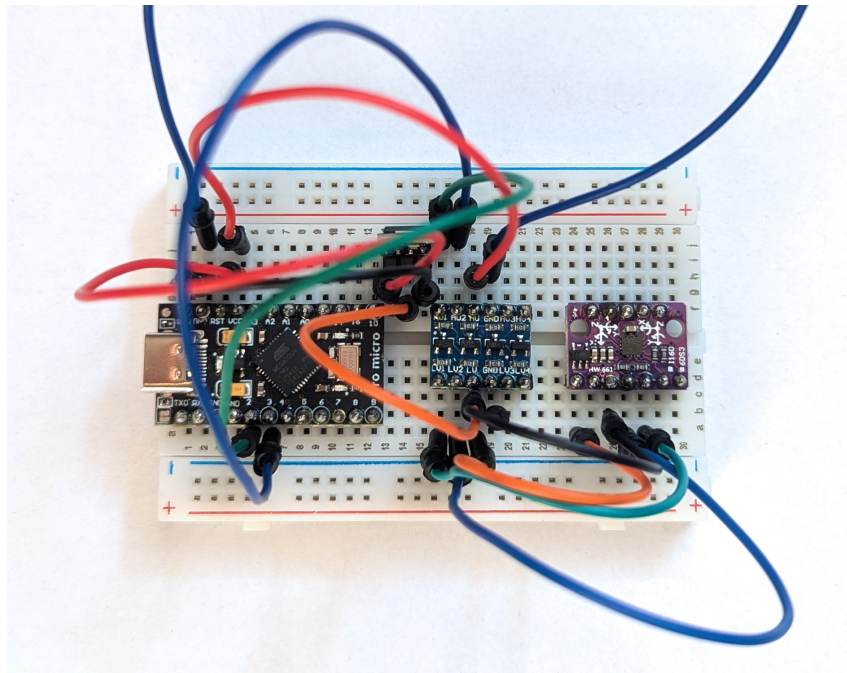


Abbildung 4.1: Testschaltung mit MPU6050, Breadboard und Arduino Mega.

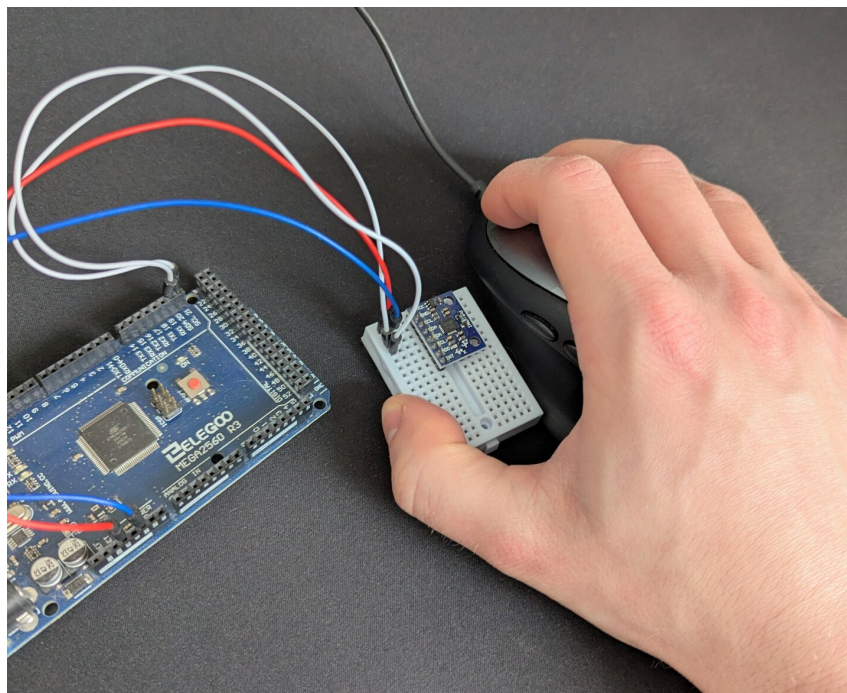


Abbildung 4.2: Gemeinsame Bewegung der optischen Maus und der MPU6050 während der Datenaufnahme.

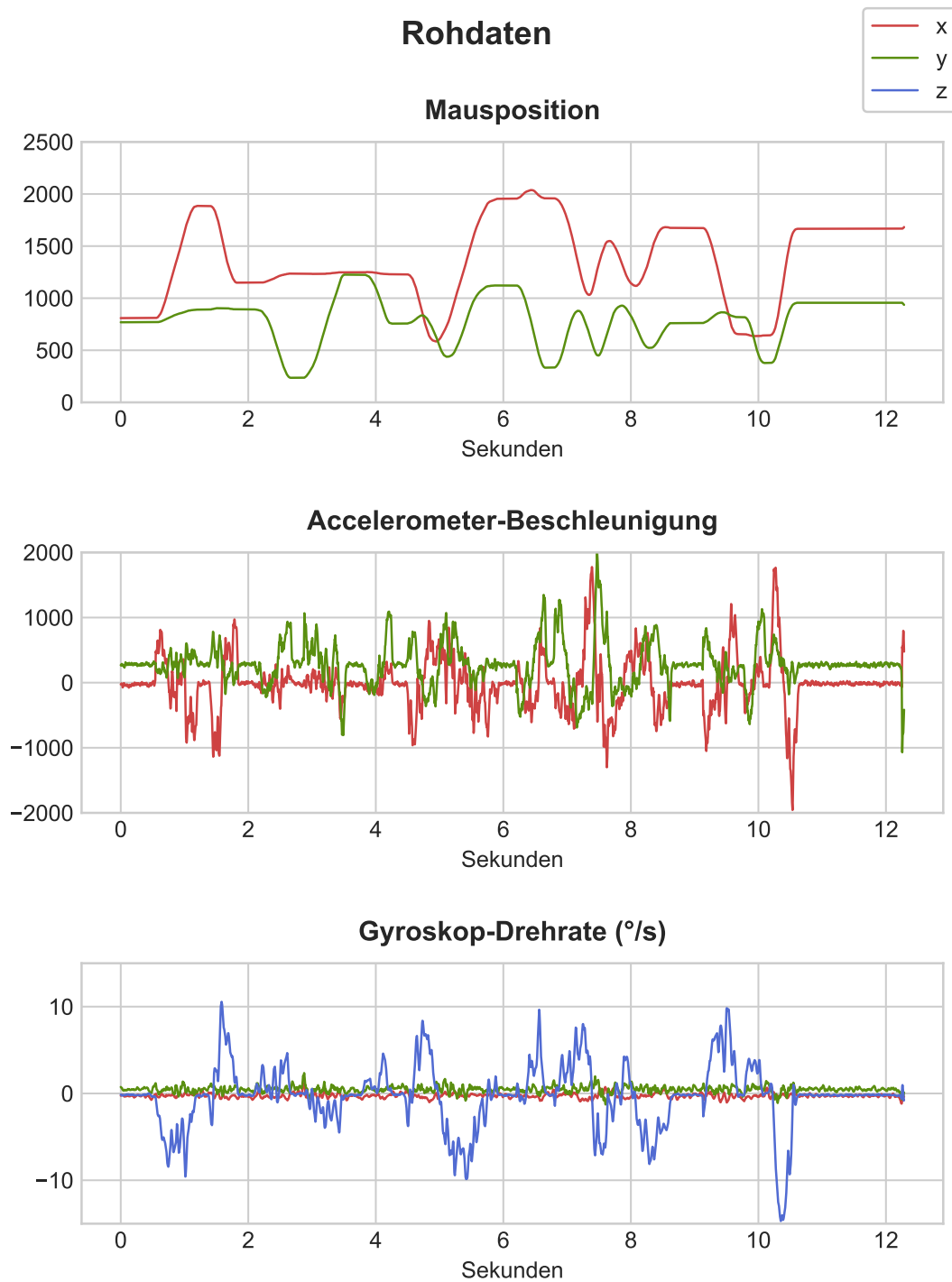


Abbildung 4.3: Gemeinsam aufgezeichnete Rohdaten: Cursorposition, Beschleunigung und Drehrate.

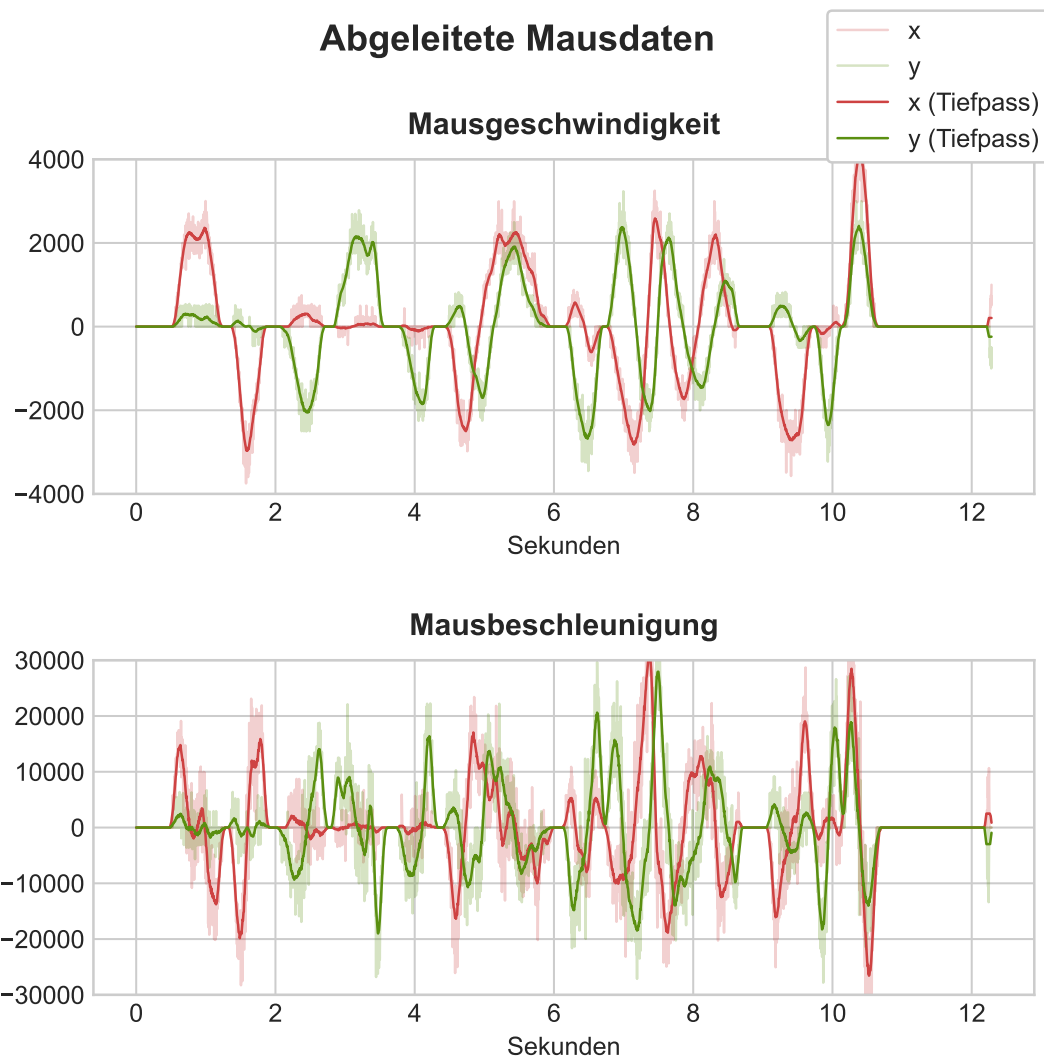


Abbildung 4.4: Aus der aufgezeichneten Cursorposition abgeleitete Mausgeschwindigkeit und Mausbeschleunigung.

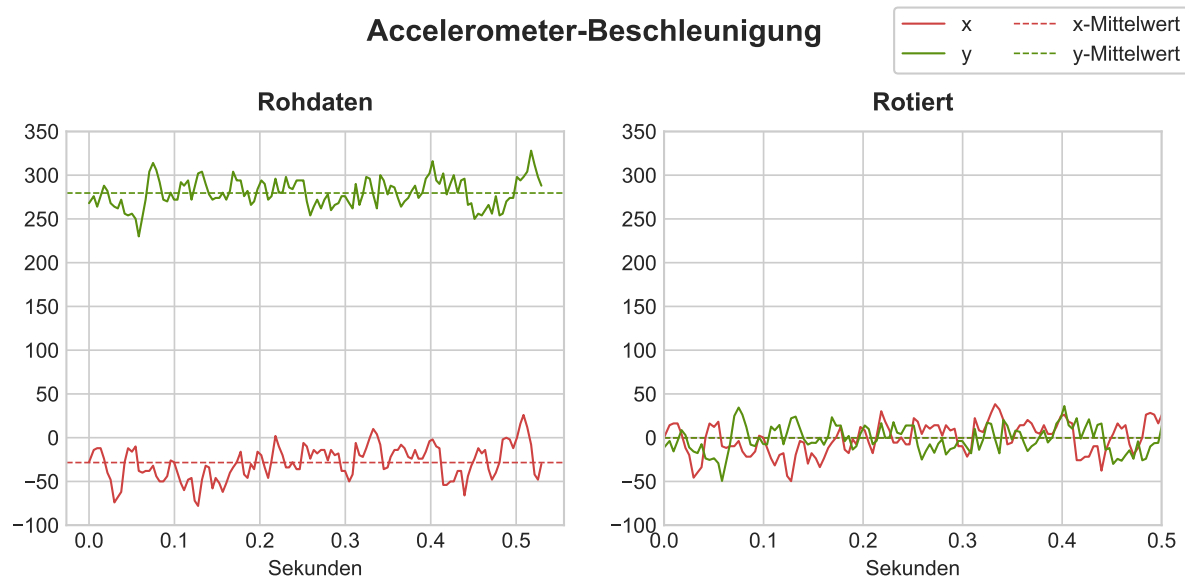


Abbildung 4.5: Accelerometerdaten nach der statischen Rotation des anfänglichen Gravitationsvektors auf die z-Achse.

4.3 Statische Ausrichtung des Accelerometers

Zu Beginn der Aufnahme lag die IMU für einige Samples (ca. 0,5 Sekunden) ruhig. In diesem Abschnitt wurden x- und y-Winkel so gewählt, dass eine Drehung der Beschleunigungsvektoren um die Rotationsmatrizen R_y und R_x eine Beschleunigung von näherungsweise null auf der x- und y-Achse ergeben, sodass die Gravitation nur auf die z-Achse wirkt:

$$\mathbf{a}_k^{\text{ausgerichtet}} = \mathbf{R}_y \mathbf{R}_x \mathbf{a}_k^{\text{roh}}. \quad (4.3)$$

Das Ergebnis ist in Abbildung 4.5 dargestellt. Die Korrektur berücksichtigt nur die anfängliche Lage. Eine spätere Verkipfung kann weiterhin Gravitationsanteile in die horizontalen Achsen einbringen.

4.4 Integration und Ruheerkennung

Nach Achsenausrichtung und Skalierung wurden die horizontalen Beschleunigungen mit Gleichung (2.4) integriert. Die Beschleunigung der IMU deckt sich in dem Vergleich gut mit der berechneten Mausbeschleunigung. Auch die daraus berechnete Geschwindigkeit ähnelte sich. Gleichzeitig verschob sich ihr Nullpunkt der IMU-Geschwindigkeit mit zunehmender Dauer. Abbildung 4.6 zeigt diesen Effekt.

Als Gegenmaßnahme wurde eine einfache Ruheerkennung ergänzt. Unterschritt der Betrag der Beschleunigung einen Schwellwert und blieb für eine festgelegte Mindestzeit darunter, wurde die Beschleunigung auf null gesetzt. Zusätzlich wurde die Geschwindigkeit beim erkannten Stillstand zurückgesetzt. Dieser Ansatz verhindert, dass ein kleiner Restoffset während einer echten Pause unbegrenzt weiterintegriert wird.

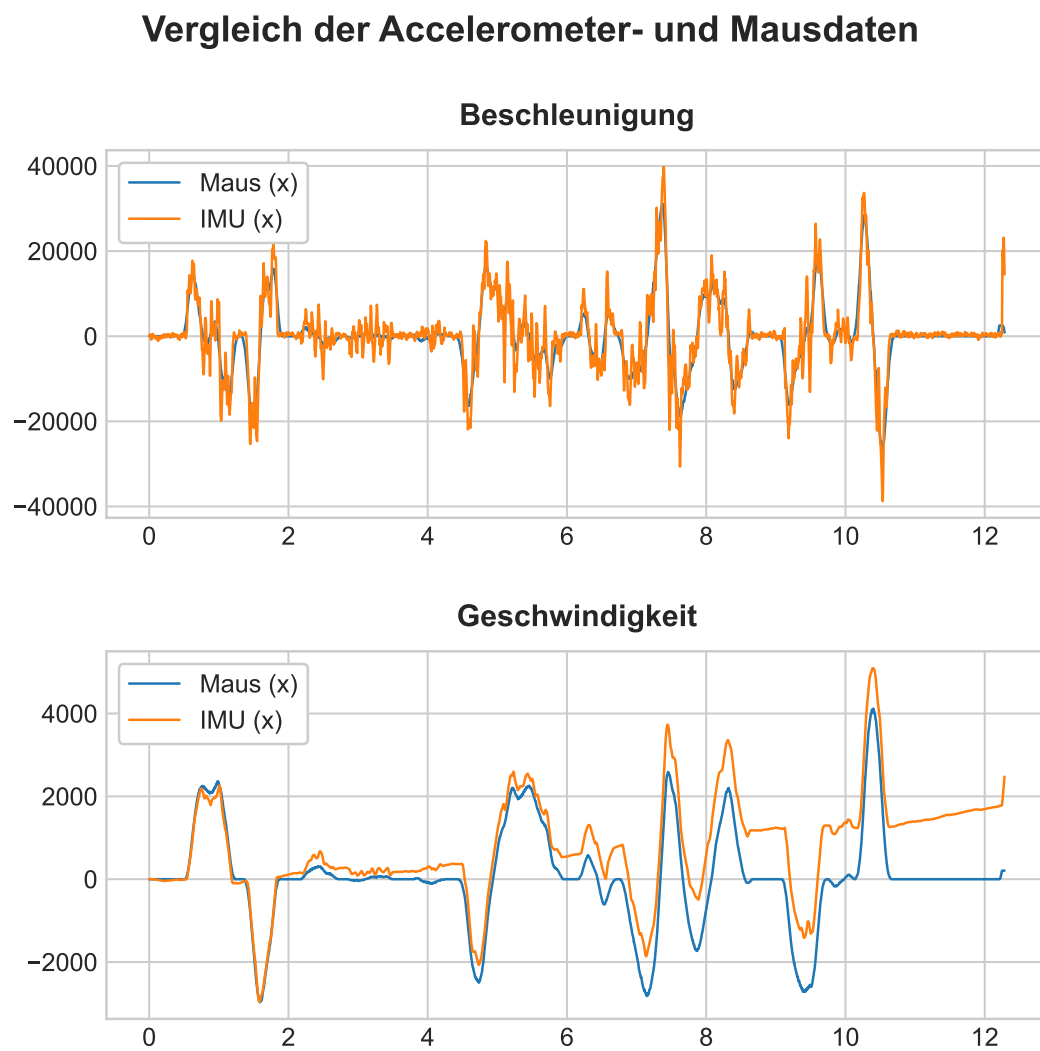


Abbildung 4.6: Qualitativer Vergleich von IMU- und Mausdaten vor einer Ruhekorrektur. Die Verläufe ähneln sich abschnittsweise, die integrierte IMU-Geschwindigkeit driftet jedoch.

Abbildung 4.7 zeigt, dass die so korrigierte Geschwindigkeit der aus der optischen Maus abgeleiteten Geschwindigkeit deutlich ähnlicher verläuft. Die Position driftet trotzdem weiter, insbesondere während längerer Bewegungsphasen. In diesen Phasen steht keine sichere Nullbedingung zur Verfügung. Jeder verbleibende Beschleunigungsfehler wirkt bis zum nächsten Reset auf Geschwindigkeit und Position.

4.5 Zwischenergebnis

Einerseits enthalten die IMU-Daten klar erkennbare Informationen über die Mausbewegung. Vor allem kurze und deutliche Beschleunigungsimpulse lassen sich in der Referenz wiederfinden. Andererseits reicht eine anfängliche Kalibrierung mit anschließender doppelter Integration nicht aus, um eine langfristig stabile Position zu erhalten. Die Ruheerkennung begrenzt den Fehler während Pausen, kann ihn aber während einer fortgesetzten Bewegung nicht korrigieren.

Damit war eine perfekte Positionsrekonstruktion bereits vor dem Bau des Prototyps nicht zu erwarten. Die Ähnlichkeit der kurzfristigen Geschwindigkeitsverläufe rechtfertigte jedoch die weitere Untersuchung, ob eine relative Cursorsteuerung mit häufigen Resets und kurzer Integrationsdauer dennoch nutzbar sein könnte.

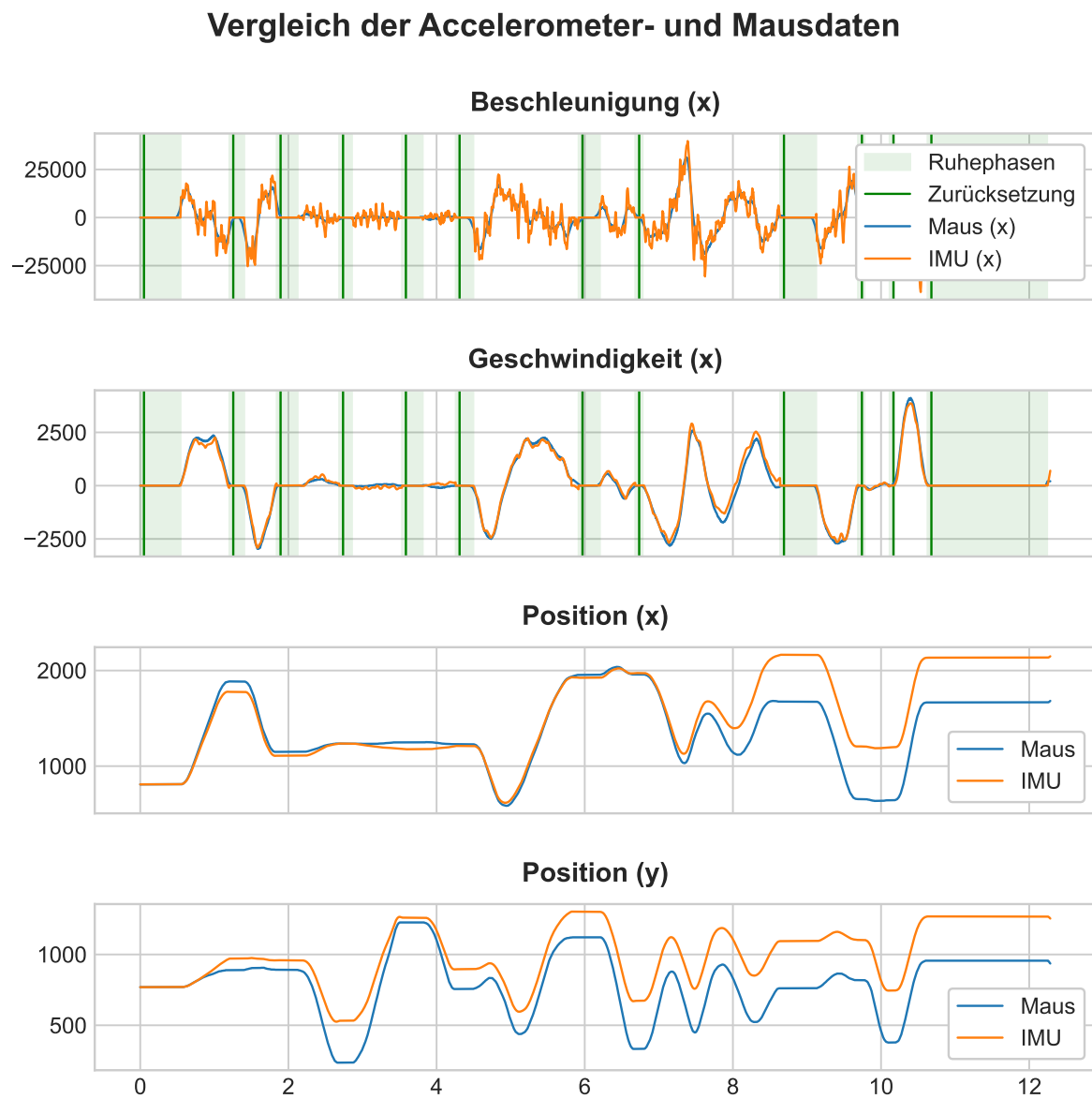


Abbildung 4.7: Vergleich von Beschleunigung, Geschwindigkeit und Position nach Ergänzung der Ruheerkennung und des Geschwindigkeitsresets.

5 Entwicklung der IMU-Maus

5.1 Hardwareauswahl

Der Prototyp wurde in das Gehäuse einer Logitech G400s eingebaut. Als optische Vergleichsmaus diente eine Logitech MX518 mit derselben Gehäuseform. Dadurch blieb die Handauflage vergleichbar, während sich die Art der Bewegungserfassung unterschied.

Als Sensor wurde eine LSM6DS3 gewählt. Das Bauteil enthält ein dreiachsiges Accelerometer und ein dreiachsiges Gyroskop, unterstützt I²C und SPI und arbeitet mit einer Versorgungsspannung von 1,71 V bis 3,6 V [vgl. 6, S. 1]. Das Datenblatt nennt im Hochleistungsmodus bei einem Messbereich von $\pm 2g$ eine typische Beschleunigungsrauschdichte von $90 \mu g/\sqrt{\text{Hz}}$ [vgl. 6, S. 20 f.]. Für die MPU6050 werden unter den dort angegebenen Bedingungen $400 \mu g/\sqrt{\text{Hz}}$ genannt [vgl. 3, S. 13]. Die Messbedingungen der beiden Datenblätter sind nicht vollständig identisch, sodass dies kein kontrollierter Direktvergleich ist. Für die Auswahl war der deutlich niedrigere Datenblattwert der LSM6DS3 dennoch ein sinnvolles Indiz, weil die Positionsschätzung hauptsächlich von den Accelerometerdaten abhängt.

Als Mikrocontroller wurde ein Arduino Pro Micro eingesetzt. Der Controller kann sich gegenüber dem Computer direkt als USB-HID-Maus anmelden. Damit ist keine zusätzliche Treibersoftware erforderlich. Die LSM6DS3 wurde über I²C angebunden. Da der verwendete Pro Micro mit 5 V arbeitet, kamen ein 3,3 V-Spannungsregler des im Aufbau verwendeten Typs AMS1117 sowie ein Pegelwandler für die Signalleitungen zum Einsatz.

5.2 Schaltung und mechanischer Aufbau

Abbildung 5.1 zeigt den Schaltplan. Neben Versorgung, Masse und den beiden I²C-Leitungen wird auch eine rote Status-LED mit dem Arduino Pro Micro verbunden, welche hier jedoch nicht weiter betrachtet wird.

Die Komponenten wurden auf einer Lochrasterplatine verlötet. Abbildung 5.2a zeigt Vorder- und Rückseite der Platine.

Die IMU befindet sich konstruktionsbedingt nicht exakt in der Mitte des Gehäuses, sondern eher im unteren rechten Bereich. Die Untersuchung von Kim et al. zeigt, dass sich eine nicht-zentrierte Sensorposition bei der Drehung auf die Cursorbewegung auswirken kann [vgl. 4, S. 1]. Bei der IMU können durch die Positionierung des Bauteils zusätzliche Beschleunigungskräfte auftreten. Der Einfluss wurde nicht separat gemessen und ist deshalb als mögliche Fehlerquelle zu berücksichtigen. Allerdings führt ein weiter hinten platzierter Sensor zu einer geringeren Auswirkung auf die Cursorposition, was in diesem Fall sogar von Vorteil sein könnte [vgl. 4, S. 1].

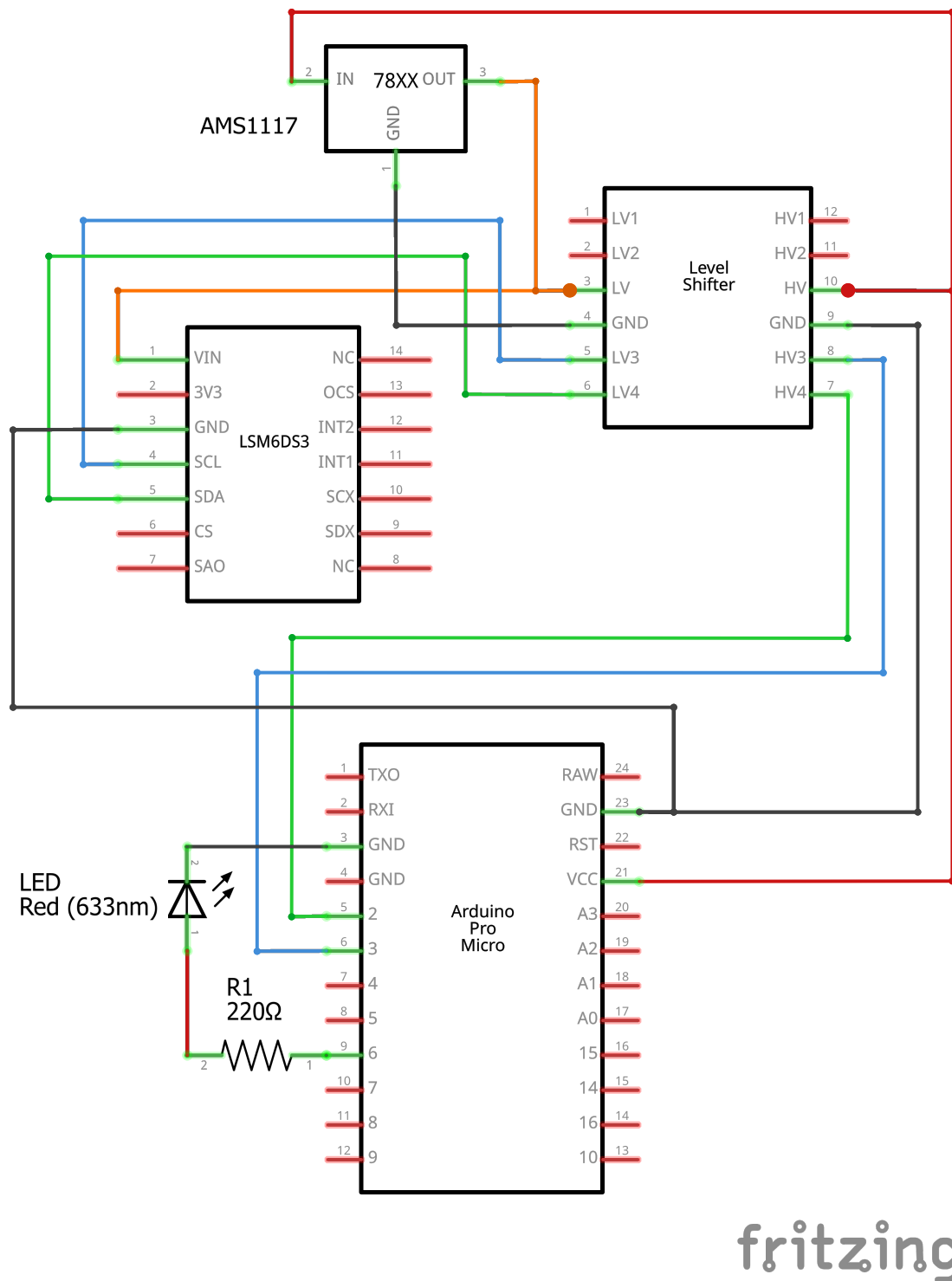
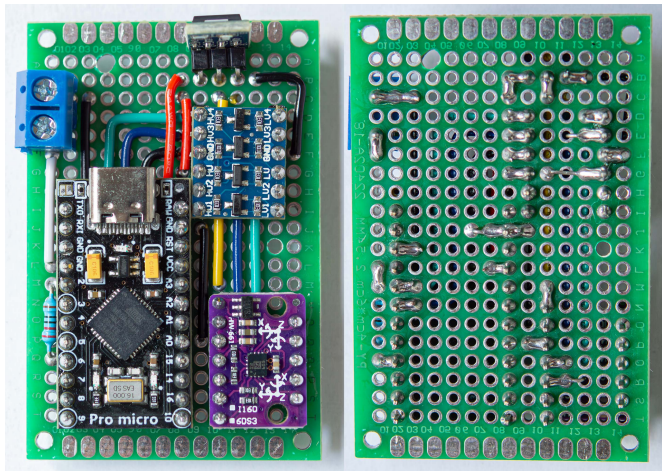
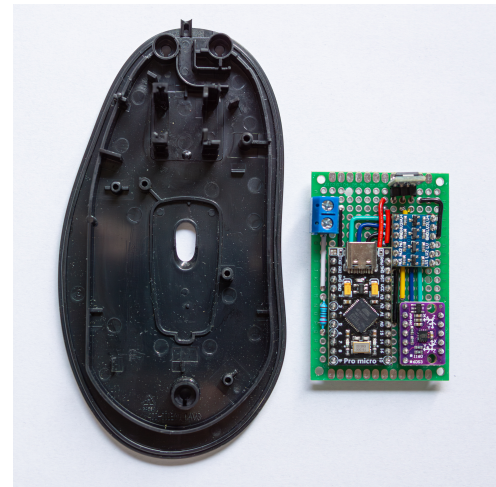


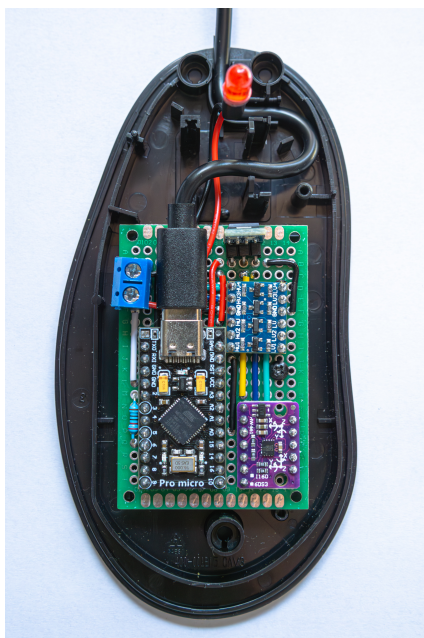
Abbildung 5.1: Schaltplan der IMU-Maus mit Arduino Pro Micro, LSM6DS3, Spannungsregler und Pegelwandler.



(a) Lochrasterplatte mit den verlöteten Komponenten.



(b) Platine neben dem geöffneten Mausgehäuse.

Abbildung 5.2: Elektronischer Aufbau vor dem Einbau in das Gehäuse.

(a) Im Gehäuse verbaute Platine.



(b) IMU-Maus und optische Vergleichsmaus.

Abbildung 5.3: Fertigstellung und Vergleich der beiden Eingabegeräte.

5.3 Firmware

5.3.1 Startkalibrierung

Nach dem Einschalten liest die Firmware $N = 500$ Samples des Accelerometers und des Gyroskops ein. Aus den Messwerten werden Mittelwerte gebildet:

$$\mathbf{b}_a = \frac{1}{N} \sum_{i=1}^N \mathbf{a}_i, \quad (5.1)$$

$$\mathbf{b}_\omega = \frac{1}{N} \sum_{i=1}^N \boldsymbol{\omega}_i. \quad (5.2)$$

Diese Werte werden im Programm als Bias gespeichert und von späteren Messungen abgezogen. Terminologisch ist insbesondere $\mathbf{b}_a$ ein Kalibrier- oder Offsetvektor: In der statischen Einbaulage enthält die z-Komponente neben dem Sensorbias auch den konstanten Gravitationsanteil. Zusätzlich wird während der Kalibrierung die größte beobachtete Länge des Beschleunigungsvektors als Ruhegrenze gespeichert. Diese wird so interpretiert, dass jede Vektorlänge der Beschleunigung, welche kleiner oder gleich der Ruhegrenze ist, als Ruhe interpretiert wird.

Die Methode entspricht grundsätzlich der in der verwandten 5-DOF-Maus eingesetzten Startmittelung, dort allerdings mit 5000 Samples [vgl. 1, S. 3]. Sie ist einfach und benötigt keine externe Referenz. Ihre Schwäche besteht darin, dass spätere Biasänderungen nicht direkt korrigiert werden.

5.3.2 Hauptschleife und Zeitmessung

In jedem Durchlauf der Hauptschleife werden die aktuellen Sensorwerte und die seit dem vorherigen Durchlauf vergangene Zeit Δt_k gelesen.

Anschließend prüft die Firmware, ob die Maus angehoben wurde oder ruht. Als angehoben gilt sie, wenn die rohe z-Beschleunigung unter $0,95g$ fällt oder die biasbereinigte z-Komponente größer als $0,1g$ ist. Ruhe wird erkannt, wenn die aktuelle Länge des Beschleunigungsvektors mindestens 50 ms lang kleiner oder gleich dem während der Startkalibrierung beobachteten Maximalwert bleibt.

Wird einer der Zustände erkannt, setzt die Firmware Beschleunigung, Geschwindigkeit und z-Rotation zurück und beendet den aktuellen Durchlauf ohne Cursorbewegung. Das Zurücksetzen verhindert eine Integration während erkannter Pausen. Gleichzeitig kann eine zu großzügige Ruhegrenze sehr langsame Bewegungen fälschlich unterdrücken.

5.3.3 Drehung des Beschleunigungsvektors

Liegt weder Ruhe noch ein Anheben vor, wird durch Integration analog zu Geschwindigkeit und Position zunächst der Drehwinkel um die z-Achse aktualisiert:

$$\psi_k = \psi_{k-1} + \omega_{z,k} \Delta t_k. \quad (5.3)$$

Der horizontale Beschleunigungsvektor wird anschließend entsprechend gedreht. In allgemeiner Form gilt, abhängig von der in der Firmware verwendeten Vorzeichenrichtung:

$$\begin{pmatrix} a_{x,k}^W \\ a_{y,k}^W \end{pmatrix} = \begin{pmatrix} \cos \psi_k & -\sin \psi_k \\ \sin \psi_k & \cos \psi_k \end{pmatrix} \begin{pmatrix} a_{x,k}^M \\ a_{y,k}^M \end{pmatrix}. \quad (5.4)$$

Die Indizes M und W bezeichnen das von der Maus und das für die Cursorbewegung verwendete Koordinatensystem. Die Berechnung kompensiert Drehungen auf der Tischfläche.

5.3.4 IIR-Tiefpass

Zur Unterdrückung hoher Frequenzen wie sie beispielsweise durch Rauschen entstehen wird auf die Beschleunigung ein einfacher Tiefpass angewendet. Die allgemeine rekursive Form lautet:

$$y_k = \alpha \cdot y_{k-1} + (1 - \alpha) \cdot x_k, \quad 0 \leq \alpha < 1. \quad (5.5)$$

Im Prototyp wurde $\alpha = 0,8$ gewählt:

$$\mathbf{a}_k^{\text{LP}} = 0,8 \mathbf{a}_{k-1}^{\text{LP}} + 0,2 \mathbf{a}_k. \quad (5.6)$$

Ein größerer Wert von α glättet stärker, erhöht aber auch die Verzögerung. Die wirksame Grenzfrequenz hängt zusätzlich von der Abtastrate ab. Lee et al. verwenden für ihren Systementwurf ebenfalls eine Tiefpassfilterung, allerdings als FIR-Filter [vgl. 5, S. 7]. Der hier eingesetzte IIR-Filter benötigt weniger Speicher und Rechenoperationen, hat aber keine scharfe Frequenzbegrenzung.

5.3.5 Integration und USB-HID-Ausgabe

Aus der gefilterten Beschleunigung werden Geschwindigkeit und Position mit Gleichung (2.4) berechnet. Gesendet wird nicht die absolute Position, sondern die Differenz zwischen aktueller und vorheriger berechneter Position. Die interne Position bleibt als Gleitkommazahl erhalten.

Diese zusätzliche Positionsakkumulation wird benötigt, da die Mausbewegungen über USB als ganzzahlige Werte übermittelt werden. Falls die aktuelle Geschwindigkeit als Fließkommazahl jedoch kleiner als 1 ist, würde als Ganzzahl eine Positionsänderung

von 0 übertragen werden. Dadurch würden sehr langsame Bewegungen schlicht nicht funktionieren.

5.4 Zusammenfassung des Algorithmus

Der vollständige Ablauf lässt sich kompakt in acht Schritten zusammenfassen:

1. Startmittelung von 500 Accelerometer- und Gyroskopsamples sowie Bestimmung der Ruhegrenze.
2. Einlesen von Sensorwerten und Δt .
3. Erkennung von Anheben oder Ruhe; gegebenenfalls Reset und Abbruch des Zyklus.
4. Integration der z-Drehrate zum Winkel ψ .
5. Rotation der horizontalen Beschleunigung in das Cursor-Koordinatensystem.
6. IIR-Tiefpass mit $\alpha = 0,8$.
7. Integration zu Geschwindigkeit und interner Position.
8. Ausgabe der Positionsdivergenz als USB-HID-Mausbewegung.

6 User Study

6.1 Aufgabe und Anwendung

Für die vergleichende Untersuchung wurde eine Anwendung entwickelt, die nacheinander Zielkreise auf dem Bildschirm anzeigt. Die teilnehmende Person bewegt den Mauszeiger in den aktuellen Kreis und muss ihn dort eine Sekunde halten. Verlässt der Zeiger den Kreis vor Ablauf der Haltezeit, wird der Timer für dieses Ziel zurückgesetzt.

Die Zielpositionen werden in normierten Bildschirmkoordinaten zwischen $(0, 0)$ oben links und $(1, 1)$ unten rechts angegeben. Alle Teilnehmenden verwendeten denselben Bildschirm. Dadurch waren Pixelauflösung, physische Bildschirmgröße und die konkrete Anordnung der Ziele für beide Mäuse und alle teilnehmenden Personen gleich. Abbildung 6.1 zeigt die Zielpositionen, Abbildung 6.2 die Benutzeroberfläche.

6.2 Messgrößen

Für jeden Zielkreis wurden drei Leistungsmaße erfasst:

- Bearbeitungszeit** Zeit vom Beginn der Bewegung bis zum erfolgreichen Abschluss des Zielkreises einschließlich der geforderten Haltephase.
- Overshoot-Anzahl** Anzahl der Fälle, in denen der Cursor den Zielkreis bereits erreicht hatte, ihn aber vor Abschluss wieder verließ.
- Overshoot-Distanz** Maximale Entfernung, die der Cursor nach einem solchen Verlassen wieder vom Zielkreis erreichte. Die Auswertung wird in Pixeln dargestellt.

Die Bearbeitungszeit erfasst sowohl grobe Zielannäherung als auch die Feinkorrektur und die Stabilität innerhalb des Kreises. Die Overshoot-Anzahl zeigt, wie oft eine Korrektur notwendig wurde. Die Distanz ergänzt diese Häufigkeit um die Größe der Abweichung.

6.3 Ablauf

Jede Maus wurde in vier vollständigen Durchläufen verwendet. Die ersten beiden Durchläufe dienten der Eingewöhnung und gingen nicht in die Messwertauswertung ein. Die letzten beiden Durchläufe wurden gewertet. Zu Beginn erfasste die Anwendung demografische und nutzungsbezogene Angaben. Nach dem Test der IMU-Maus beantworteten die Teilnehmenden einen kurzen Fragebogen und konnten Freitext angeben.

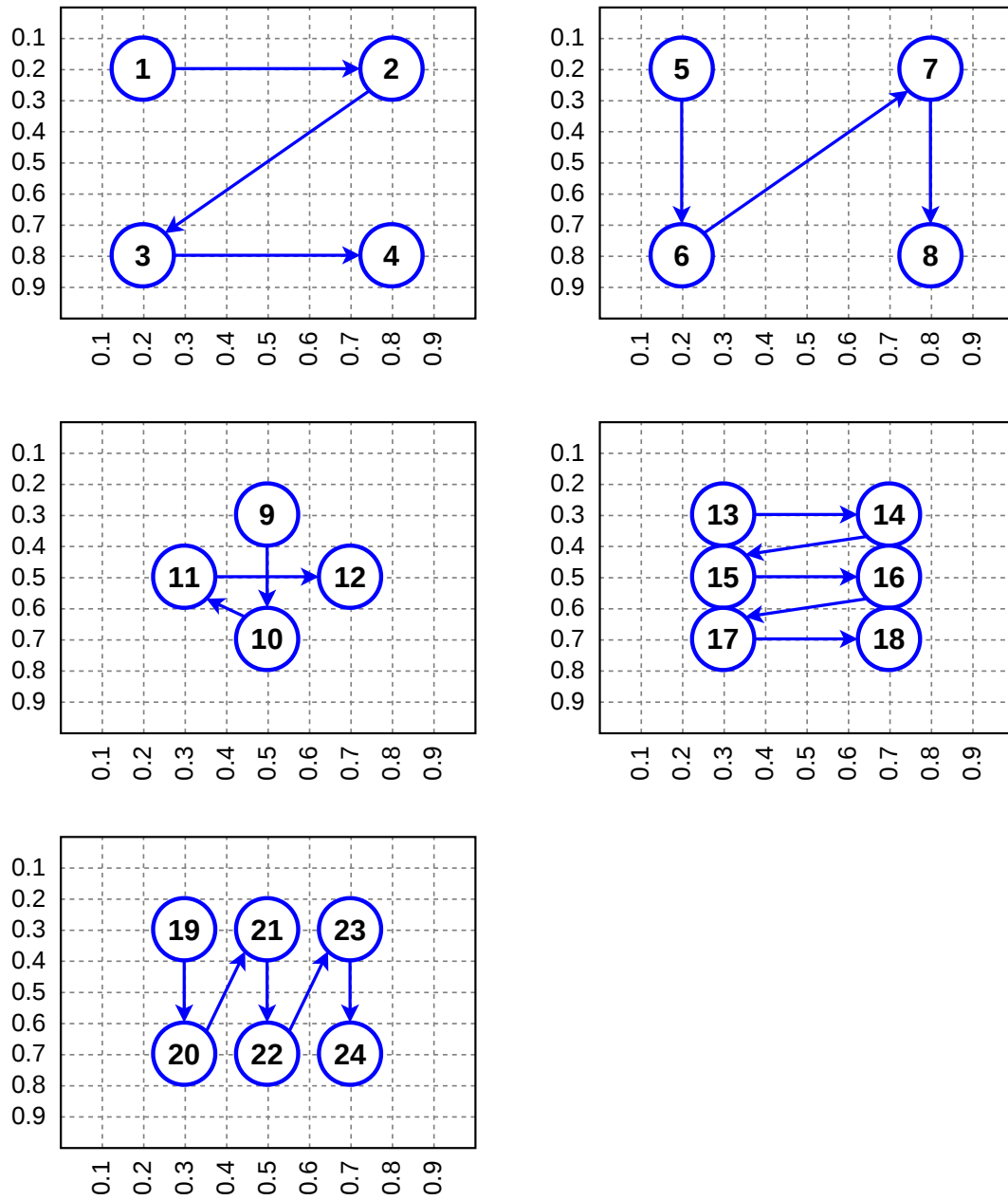


Abbildung 6.1: Koordinaten der Hitcircles.

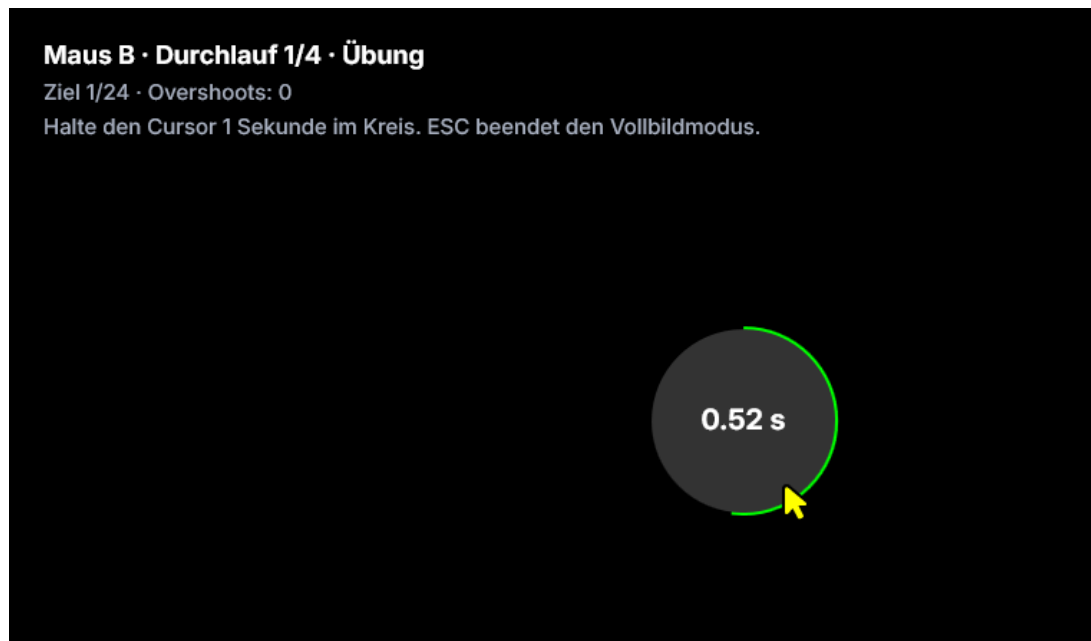


Abbildung 6.2: Screenshot der Testanwendung.

Der Fragebogen war an den System Usability Scale angelehnt, verwendete jedoch nur fünf an die konkrete Eingabemethode angepasste Aussagen. Daher wurde kein standardisierter SUS-Gesamtwert berechnet. Die Anwendung speicherte Werte auf einer fünfstufigen Skala. In der folgenden Auswertung stehen höhere Werte für stärkere Zustimmung zur jeweiligen Aussage.

6.4 Teilnehmende

An der Studie nahmen sechs Personen teil. Das Alter lag zwischen 22 und 60 Jahren, der Mittelwert betrug 32 Jahre und der Median 27 Jahre. Alle Personen gaben männliches Geschlecht, tägliche Mausnutzung und eine Tätigkeit in der Softwareentwicklung an. Eine Person gab an, linkshändisch zu sein, aber eine Computermouse mit der rechten Hand zu bedienen. Die restlichen Personen waren rechtshändisch. Fünf Personen nutzten Computer mehr als sechs Stunden täglich, eine Person drei bis sechs Stunden. Keine Person berichtete eine relevante Einschränkung. Tabelle 6.1 fasst die Angaben zusammen.

Tabelle 6.1: Demografische und nutzungsbezogene Angaben der Teilnehmenden.

ID	Alter	Geschlecht	Beruf	Hand	Maus-Hand	Maus	Computer/Tag	Einschr.
T1	28	männlich	SWE	rechts	rechts	täglich	mehr als 6 h	nein
T2	33	männlich	SWE	rechts	rechts	täglich	mehr als 6 h	nein
T3	22	männlich	SWE	rechts	rechts	täglich	mehr als 6 h	nein
T4	60	männlich	SWE	links	rechts	täglich	mehr als 6 h	nein
T5	23	männlich	SWE	rechts	rechts	täglich	3 bis 6 h	nein
T6	26	männlich	SWE	rechts	rechts	täglich	mehr als 6 h	nein

Die Stichprobe ist damit in Bezug auf Beruf, Geschlecht, Maushand und Nutzungserfahrung

sehr homogen. Das erleichtert zwar die Vergleichbarkeit innerhalb der kleinen Gruppe, begrenzt aber die Übertragbarkeit der Ergebnisse auf andere Personengruppen.

7 Ergebnisse

7.1 Bearbeitungszeit

Abbildung 7.1 vergleicht die Bearbeitungszeiten pro Zielkreis, wobei einige extreme Ausreißer der Übersicht halber abgeschnitten wurden. Nach der bereitgestellten Auswertung liegt die Verteilung der IMU-Maus deutlich ungünstiger als die der optischen Maus. Die Zielerfassung benötigte häufiger lange Korrekturphasen. Besonders bei langsamen Nachführbewegungen konnte der Cursor stehen bleiben, verzögert reagieren oder sich entgegen der beabsichtigten Richtung bewegen. Dadurch verlängerte sich nicht nur die Anfahrt, sondern auch das stabile Halten im Zielkreis.

Es wurden Boxplot-Darstellungen gewählt, welche die Längen der Antennen auf maximal das 1,5-Fache des Interquartilsabstands beschränken und Ausreißer als Kreise darstellen.

7.2 Overshoots

Die maximalen Overshoot-Distanzen sind in Abbildung 7.2 dargestellt. Auch hier zeigt die IMU-Maus die ungünstigere Verteilung. Ein einmal verfehltes Ziel führte häufiger zu einer größeren Entfernung vom Kreis, bevor eine erneute Annäherung gelang. Das passt zur Beobachtung, dass die Maus bei kurzen deutlichen Impulsen besser reagiert als bei einer langsamen Feinkorrektur.

Für die Overshoot-Anzahl liegen zusätzlich arithmetische Mittelwerte und Mediane vor. Tabelle 7.1 zeigt die Ergebnisse.

Der Mittelwert ist bei der IMU-Maus etwa 3,8-mal so hoch. Beide Mediane sind null. Das bedeutet, dass in mindestens der Hälfte der jeweils zugrunde liegenden Beobachtungen kein Overshoot auftrat. Außerdem lässt sich daraus schließen, dass es in der Regel möglich

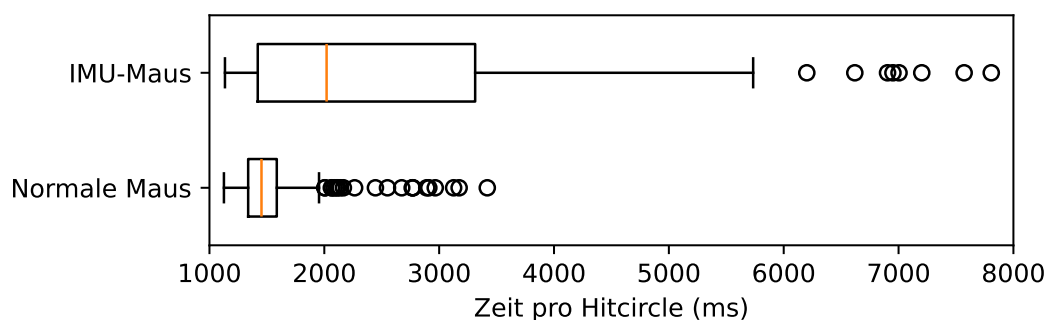


Abbildung 7.1: Boxplots der Bearbeitungszeiten pro Zielkreis für die optische Maus und die IMU-Maus (einige Ausreißer abgeschnitten).

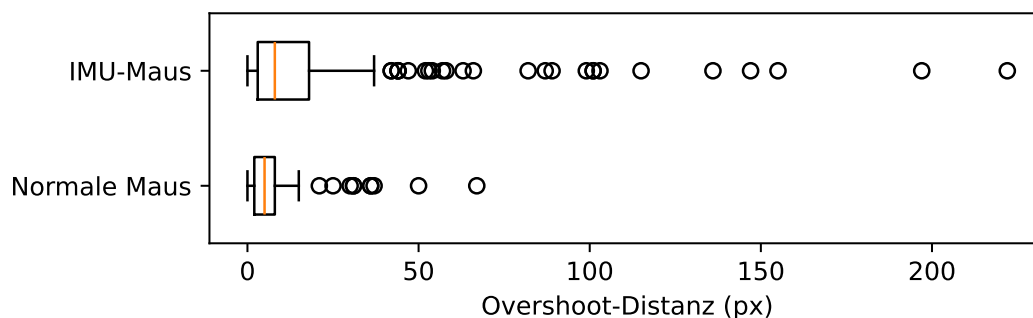


Abbildung 7.2: Boxplots der maximalen Overshoot-Distanzen für die optische Maus und die IMU-Maus.

Tabelle 7.1: Deskriptive Kennwerte der Overshoot-Anzahl.

Maus	Arithmetisches Mittel	Median
Optische Maus	0,2292	0
IMU-Maus	0,8750	0

war, die IMU-Maus ruhig zu halten, da der Mauscursor die Kreise sonst öfter verlassen hätte. Der Mittelwert zeigt gleichzeitig, dass die verbleibenden problematischen Fälle bei der IMU-Maus deutlich häufiger oder stärker gebündelt waren.

7.3 Fragebogen

Die fünf Aussagen und die individuellen Bewertungen sind in Tabelle 7.2 zusammengefasst. Die Kurzbezeichnungen stehen für folgende Aussagen:

Alltag	Ich würde diese Eingabemethode im Alltag nutzen.
Verständlich	Die Steuerung war leicht verständlich.
Präzise	Die Steuerung fühlte sich präzise an.
Anstrengend	Die Steuerung war anstrengend.
Lernbar	Ich kann mir vorstellen, dass die meisten Leute den Umgang mit der Maus schnell lernen.

Die Alltagstauglichkeit erhielt mit einem Mittelwert von 1,17 die niedrigste Zustimmung. Die Aussage zur Anstrengung erhielt mit 3,83 die höchste Zustimmung. Die Steuerung wurde im Mittel mehr verständlich als präzise bewertet. Die Ergebnisse sprechen deshalb nicht für ein grundsätzlich unverständliches Interaktionskonzept. Sie zeigen vielmehr, dass die technische Umsetzung die beabsichtigte Bewegung nicht zuverlässig genug abbildete und dadurch hohe Korrekturanforderungen erzeugte.

Tabelle 7.2: Fragebogenwerte auf der vierstufigen Zustimmungsskala. Höhere Werte bedeuten stärkere Zustimmung.

ID	Alltag	Verständlich	Präzise	Anstrengend	Lernbar
T1	1	1	1	4	2
T2	1	4	3	4	3
T3	1	4	2	4	1
T4	1	4	1	4	1
T5	2	2	3	4	2
T6	1	2	2	3	2
Mittelwert	1,17	2,83	2,00	3,83	1,83
Median	1	3	2	4	2

7.4 Freitextfeedback

Vier Personen gaben Freitext an:

T2: Schnelle Mausbewegungen funktionieren recht zuverlässig, sobald man sich einmal an die Steuerung gewöhnt hat. Kleine Bewegungen aber sind mühsam. Probleme scheint es auch zu geben, wenn man die Maus anhebt.

T3: Die Maus war bei schnellen Bewegungen deutlich genauer.

T4: Um die Maus präzise zu steuern, verlasse ich mich darauf, dass der mit der Hand zurückgelegte Weg skaliert dem Weg der Maus entspricht - das ist bei dieser Maus nicht der Fall. Es kam häufig vor, dass die Maus trotz Bewegung an der Stelle verharrte oder sich sogar in die entgegengesetzte Richtung bewegte. Bei schneller Bewegung in Richtung Ziel konnte ich die Maus gut führen, langsames Nachführen bei Verfehlen des Ziels war dann wieder schwierig. Im Alltag werden eher kleinere Schaltflächen und eine Stelle an einem Wort (Textcursor) angesteuert - ich kann mir nicht vorstellen, dass diese Methode dort Vorteile haben kann. Dennoch denke ich, dass mehr Übung mit der neuen Maus meine Genauigkeit verbessern kann.

T5: Die Eingabemethode funktioniert besser, wenn die Maus schneller bewegt wird.

Die Rückmeldungen wiederholen konsistent drei Beobachtungen: Schnelle Bewegungen funktionieren vergleichsweise gut, langsame Korrekturen sind schwierig, und das Anheben oder erneute Aufsetzen kann zu unerwartetem Verhalten führen. Diese qualitativen Aussagen stimmen mit der technischen Voruntersuchung und den Overshoot-Ergebnissen überein.

8 Diskussion

8.1 Beantwortung der Forschungsfragen

Die erste Forschungsfrage ist für die untersuchte Hardware und Implementierung zu verneinen. Aus den IMU-Daten ließ sich zwar eine Cursorbewegung erzeugen, die kurzfristig erkennbare Gemeinsamkeiten mit der tatsächlichen Mausbewegung aufwies. Die Genauigkeit und Stabilität reichten jedoch nicht für eine allgemein nutzbare Computermouse. Insbesondere langsame, kleine und längere Bewegungen wurden unzuverlässig abgebildet.

Auch die zweite Forschungsfrage ergibt ein klares deskriptives Bild. Die optische Vergleichsmouse war bei der Zielerfassungsaufgabe schneller und präziser. Die IMU-Mouse verursachte im Mittel deutlich mehr Overshoots und größere Korrekturbewegungen. Die Fragebogenwerte zeigen geringe Bereitschaft zur Alltagsnutzung und eine hohe empfundene Anstrengung. Das Freitextfeedback bestätigt diese Tendenz.

Die Aussage ist bewusst auf den realisierten Prototyp begrenzt. Aus der Studie folgt nicht, dass jede denkbare inertielle Mouse ungeeignet ist. Sie zeigt aber, dass eine kostengünstige MEMS-IMU mit einmaliger Startkalibrierung, einfacher Filterung und doppelter Integration keine direkte, gleichwertige Ersetzung eines optischen Sensors darstellt.

8.2 Warum schnelle Bewegungen besser funktionieren

Die wiederholt beobachtete Stärke bei schnellen ruckartigen Bewegungen lässt sich durch das Verhältnis zwischen Nutzsignal und Fehler erklären. Ein kurzer kräftiger Impuls erzeugt eine Beschleunigung, die deutlich über Rauschen und Restoffset liegt. Außerdem ist die Integrationsdauer kurz. Nach Gleichung (2.5) wächst der durch einen konstanten Bias verursachte Positionsfehler quadratisch mit der Zeit. Wird ein Ziel mit einem kurzen Impuls angenähert und unmittelbar danach Ruhe erkannt, begrenzt der Reset die Fehlerakkumulation.

Bei langsamen Bewegungen ist die Situation umgekehrt. Die tatsächliche Beschleunigung liegt näher an Rauschen und Schwellwerten. Die Ruherkennung kann eine beabsichtigte Bewegung als Stillstand klassifizieren und unterdrücken. Wird die Bewegung erkannt, dominiert ein kleiner Offset wegen der längeren Integrationsdauer zunehmend die berechnete Geschwindigkeit. Das erklärt sowohl das Verharren trotz Handbewegung als auch Bewegungen in die falsche Richtung.

Lange Bewegungen sind zusätzlich problematisch, weil während der aktiven Phase kein sicherer Referenzpunkt vorhanden ist. Ein Geschwindigkeitsreset wäre dort falsch, ein ausbleibender Reset lässt den Fehler weiter anwachsen. Genau dieses Grundproblem zeigt die Voruntersuchung: Der Verlauf der Geschwindigkeit kann abschnittsweise plausibel sein, während die Position trotzdem deutlich wegdriftet.

8.3 Einfluss der Implementierung

Mehrere Designentscheidungen können die Ergebnisse verschlechtert haben:

1. **Nur einmalige Kalibrierung:** Der zu Beginn bestimmte Offset wird konstant gehalten. Biasinstabilität und temperaturabhängige Änderungen werden nicht nachgeführt, obwohl beide Effekte bei MEMS-Sensoren relevant sind [vgl. 2, S. 13, vgl. 7, S. 16].
2. **Begrenzte Lagekompensation:** Die Firmware integriert nur die Drehrate um die z-Achse. Kleine Kippbewegungen um x und y können Gravitationsanteile in die horizontalen Messungen projizieren. Wegen der Größe von g reichen geringe Winkelabweichungen für erhebliche Scheingeschwindigkeiten.
3. **Einfache Ruheerkennung:** Der Schwellwert wird aus einer kurzen Startphase abgeleitet und verwendet keine genauere Bewertung. Er kann deshalb langsame Bewegungen unterdrücken und nach einem Anheben zu Fehlklassifikationen führen.
4. **Einfache Filterung:** Der IIR-Tiefpass reduziert hochfrequentes Rauschen, kann aber keinen zeitlich veränderlichen Bias entfernen. Eine stärkere Glättung würde zudem die Reaktion verzögern.
5. **Außermittige Sensorposition:** Die IMU sitzt nicht im Zentrum. Rotationen der Maus erzeugen daher zusätzliche lineare Beschleunigung am Messort. Die genaue Wirkung hängt vom momentanen Drehzentrum ab.

Diese Punkte zeigen, dass die konkrete Implementierung nicht als technisch optimal gelten kann. Gleichzeitig beheben sie nicht das Grundproblem einer absoluten Positionsbestimmung ohne externe Referenz. Auch ein aufwendigerer Filter benötigt Beobachtungen oder Randbedingungen, mit denen Fehlerzustände korrigiert werden können.

8.4 Mögliche Verbesserungen

Eine weiterentwickelte Version könnte die Startkalibrierung durch eine kontinuierliche Aktualisierung des Bias in sicher erkannten Ruhephasen ergänzen. Dies wurde bereits getestet, hat sich jedoch auch als schwierig herausgestellt, da die Ruhephasen deutlich kürzer als die Kalibrierungszeit sein können und der Nutzer die Maus während der Ruhephasen berühren kann, was zu einer ungenauen Ermittlung des Bias führt.

Anstatt nur einen ermittelten Bias abzuziehen, könnte man mit dem Accelerometer und Gyroskop die Orientierung des Sensors bestimmen, um die Beschleunigungsvektoren auf allen Achsen zu rotieren. Hier könnte ein Kalman-Filter oder Komplementärfilter möglicherweise sinnvoll sein.

Eine zentrale Sensorposition könnte getestet und mit anderen Positionen verglichen werden.

9 Zusammenfassung

In dieser Arbeit wurde untersucht, ob eine Computermouse mit einer inertialen Messeinheit anstelle eines optischen Bewegungssensors realisiert werden kann. Die Motivation war eine von optischen Oberflächeneigenschaften unabhängige Bewegungserfassung. Zunächst wurden eine MPU6050 und eine optische Maus gemeinsam bewegt und aufgezeichnet. Die Verarbeitung zeigte, dass Beschleunigungsereignisse und kurzfristige Geschwindigkeitsverläufe grundsätzlich miteinander zusammenhängen. Trotz Achsenausrichtung, Filterung und Reset bei erkannter Ruhe driftete die aus der IMU berechnete Position jedoch während Bewegungsphasen.

Darauf aufbauend wurde ein eigenständiger Prototyp mit LSM6DS3 und Arduino Pro Micro in ein Mausgehäuse eingebaut. Die Firmware kalibriert den Sensor beim Start, erkennt Ruhe und Anheben, kompensiert Drehungen um die z-Achse, filtert die Beschleunigung, integriert zu Geschwindigkeit und Position und sendet USB-HID-Mausbewegungen.

Die User Study mit sechs Teilnehmenden zeigte deutliche Nachteile gegenüber einer optischen Maus gleicher Form. Die IMU-Maus benötigte deutlich längere Zielkorrekturen, erzeugte im Mittel etwa 3,8-mal so viele Overshoots und wurde als anstrengend sowie nicht alltagstauglich bewertet. Schnelle kurze Bewegungen funktionierten vergleichsweise gut. Langsame, kleine und lange Bewegungen waren besonders ungenau. Dieser Befund stimmt mit der erwarteten Fehlerakkumulation bei doppelter Integration überein.

Für die vorliegende Implementierung lautet das Ergebnis deshalb, dass eine reine IMU keine geeignete Alternative zu einem optischen Maussensor ist. Anpassungen bei Bias-Berechnung, Lagebestimmung, Ruheerkennung, Sensorposition und numerischer Verarbeitung könnten die Bedienung verbessern. Ohne eine absolute Referenz bleibt der langfristige Drift jedoch die zentrale Grenze.

10 Quellen- und Literaturverzeichnis

- [1] D. I. Barbosa Tarazona, D. A. Diaz Gómez und S. Arévalo González. „Design and Implementation of a Computer Mouse Using a 5-DOF Inertial Measurement Unit and a Sensor Fusion Algorithm“. In: *2012 Workshop on Engineering Applications*. 2012, 1 bis 6. DOI: 10.1109/WEA.2012.6220076.
- [2] IEEE. *IEEE Standard for Inertial Sensor Terminology*. IEEE Std 528-2019. 2019, 1 bis 35. DOI: 10.1109/IEEESTD.2019.8863799.
- [3] InvenSense Inc. *MPU-6000 and MPU-6050 Product Specification, Revision 3.4*. Document Number PS-MPU-6000A-00. 2013. URL: https://www.cdiweb.com/datasheets/invensense/mpu-6050_datasheet_v3%204.pdf.
- [4] S. Kim, B. Lee, T. van Gemert und A. Oulasvirta. „Optimal Sensor Position for a Computer Mouse“. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1 bis 13. DOI: 10.1145/3313831.3376735. URL: <https://doi.org/10.1145/3313831.3376735>.
- [5] S. Lee, G.-J. Nam, J. Chae, H. Kim und A. J. Drake. „Two-Dimensional Position Detection System with MEMS Accelerometers, Readout Circuitry, and Microprocessor for Padless Mouse Applications“. In: *IEEE Transactions on Very Large Scale Integration Systems* 13.10 (2005), 1167 bis 1178. DOI: 10.1109/TVLSI.2005.859473.
- [6] STMicroelectronics. *LSM6DS3: iNEMO Inertial Module, Always-on 3D Accelerometer and 3D Gyroscope*. Datasheet, DocID026899 Rev 10. 2017. URL: https://content.arduino.cc/assets/st_imu_lsm6ds3_datasheet.pdf.
- [7] O. J. Woodman. *An Introduction to Inertial Navigation*. Techn. Ber. UCAM-CL-TR-696. University of Cambridge, Computer Laboratory, 2007. URL: <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>.

11 Nutzung KI-basierter Anwendungen

	KI-basiertes Hilfsmittel	Einsatzform	Betroffene Teile der Arbeit	Beschreibung der Eingabe	Bemerkung
1	ChatGPT	Literaturüberblick zu ähnlichen Themen und Projekten	Literaturrecherche	Ich baue eine Maus mit einer inertialen Messeinheit statt einem optischen Sensor. Recherchiere, ob sowas schonmal gemacht wurde und ob es Literatur zu ähnlichen Themen gibt.	Ich konnte auf die gefundene Literatur aufbauen und sie diente als guter Einstieg für die weitere Recherche.
2	ChatGPT	Vergleich elektronischer Komponenten	Auswahl der inertialen Messeinheit sowie des Spannungsreglers für meinen Prototypen	Ich möchte eine Maus mit einer inertialen Messeinheit bauen. Welche Module sind dafür geeignet? Ich brauche einen Spannungsregler, welcher 5V zu 3,3V umwandelt. Vergleiche die verfügbaren Module.	Hat mir geholfen, Leistung bzw. technische Daten und Preis von den Komponenten zu vergleichen und zu entscheiden, welche für meinen Prototypen geeignet sind.
3	ChatGPT	Erstellen einer Webanwendung	Webanwendung für die User-Study zum Vergleich der normalen und der IMU-Maus durch Erfassung von Messdaten	Ich möchte eine Webanwendung bauen, mit welcher ausgewertet werden soll, wie genau die Nutzer den Mauscursor bewegen können. Das ist Teil einer User-Study, in welcher die Genauigkeit von zwei Mäusen verglichen wird. [...]	Ich habe das Grundgerüst der Anwendung schon erstellt, welches ausgebaut werden sollte. Der Prompt war durch viel Kontext sehr lang. Da dieser Teil der BA recht uninteressant war und ich etwas Zeit sparen musste, hat ChatGPT einen großen Teil übernommen.
4	ChatGPT	Erstellen einer LaTeX-Vorlage	Form der Bachelorarbeit	Ich schreibe eine Bachelorarbeit. Erstelle mit eine LaTeX-Vorlage mit folgender Gliederung: [...]	Hat mir geholfen, schnell eine Vorlage zu erstellen und mich auf den Inhalt zu konzentrieren.

Eigenständigkeits- und Einverständniserklärung zur Überprüfung mit Plagiatsoftware sowie die Erklärung zur Veröffentlichung bei Bachelor- und Masterarbeiten

Declarations of Authorship and Consent for Checking with Plagiarism Software and the Declaration of Publication for Bachelor's and Master's Thesis

Studierenden-Angaben / Student Information:

Matrikelnr. / Student ID	6034915
Nachname / Surname	Kannengießer
Vorname / First Name	Merlin

Titel der Arbeit / Title of Thesis

Eine Maus nur mit einer inertialen Messeinheit

(Englischer Titel: A Mouse Using Only an Inertial Measurement Unit)

A) Eigenständigkeitserklärung / Declaration of Authorship

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge. Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht.

I hereby affirm that I have written the present work independently and have used no sources or aids other than those indicated. All parts of my work that have been taken from other works, either verbatim or in terms of meaning, have been marked as such, indicating the source. The same applies to drawings, sketches, pictorial representations and sources from the Internet, including AI-based applications or tools. The work has not yet been submitted in the same or a similar form as a final examination paper.

- ☒ Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI basierte Anwendungen" dokumentiert.

I have used AI-based applications and/or tools and documented them in the appendix "Use of AI-based applications".

B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten

Declaration regarding the publication of bachelor's or master's thesis

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten. Archiviert werden:

Two years after graduation, the thesis is offered to the archive of the University of Bremen for permanent archiving. The following are archived:

- 1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Masterarbeiten
Master's theses with a local or regional focus, as well as per subject and academic year 10 % of all Master's thesis
- 2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.
Bachelor's thesis for the first and last bachelor's degrees per subject and year.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I agree that my thesis may be viewed by third parties in the university archive for academic purposes.

☐ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I agree that my thesis may be viewed by third parties for academic purposes in the university archive after 30 years in accordance with para. 2 BremArchivG).

☐ Ich bin **nicht** damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.
I do not consent to my thesis being made available in the university archive for third parties to view for academic purposes.

C) Einverständniserklärung zur elektronischen Überprüfung der Arbeit auf Plagiate Declaration of consent for electronic checking of the work for plagiarism

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatsvorwürfe untersucht werden.

Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatssoftware.

Submitted papers can be checked for plagiarism using qualified software in accordance with § 18 of the General Section of the Bachelor's or Master's Degree Examination Regulations of the University of Bremen. For the purpose of checking for plagiarism, the upload to the server is done using the plagiarism software currently used by the University of Bremen.

- ☐ Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

I agree that the work I have submitted and written will be stored permanently on the external server of the plagiarism software currently used by the University of Bremen, in a library belonging to the institution accessed only by the University of Bremen), for the above-mentioned purpose.

- ☒ Ich bin **nicht** damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

I do not consent to the work I submitted and wrote being permanently stored on the external server of the plagiarism software currently used by the University of Bremen, in a library belonging to the institution accessed only by the University of Bremen), for the above-mentioned purpose.

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig. Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden. Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Consent to the permanent storage of the text is voluntary. Consent can be withdrawn at any time by making a declaration to this effect to the University of Bremen, with effect for the future. Further information on the checking of written work using plagiarism software can be found in the data protection and usage concept. This can be found on the University of Bremen website.

Mit meiner Unterschrift versichere ich, dass ich die obenstehenden Erklärungen gelesen und verstanden habe und bestätige die Richtigkeit der gemachten Angaben.

With my signature, I confirm that I have read and understood the above explanations and confirm the accuracy of the information provided.

16.07.2026



Datum / Date

Unterschrift als Scan/ Signature