



HafenCity Universität Hamburg  
Universität für Baukunst und Metropolenentwicklung

**Location Based Service**

**Optimieren einer Trajektorie mit absoluten 5G Koordinaten**

Studiengang  
**Geodesy and Geoinformatics (GIT)**

Matrikelnummer:  
**6059167 und 6056745**

Professor:  
**M.Sc. Hossein Shoushtari**

Gruppe 3:  
**Sumit Kaur und Simeon Zeyse**

July 21, 2022

---

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Methodologie</b>	<b>3</b>
2.1	Das Kalman Filter . . . . .	3
2.1.1	Diskreter Kalman-Filter (lineares Modell) . . . . .	4
2.1.2	Extended Kalman-Filter (nicht- lineares Modell) . . . . .	6
2.1.3	Weitere Quellen: . . . . .	6
2.2	Weigted Least-Square . . . . .	6
<b>3</b>	<b>Implementation</b>	<b>7</b>
3.1	Diskreter Kalman-Filter . . . . .	7
3.2	Extended Kalman-Filter . . . . .	9
3.3	Weighted-Least-Square . . . . .	9
3.4	Vergleich der Ergebnisse . . . . .	10
<b>4</b>	<b>Fazit und Ausblick</b>	<b>10</b>
	<b>Bibliography</b>	<b>11</b>

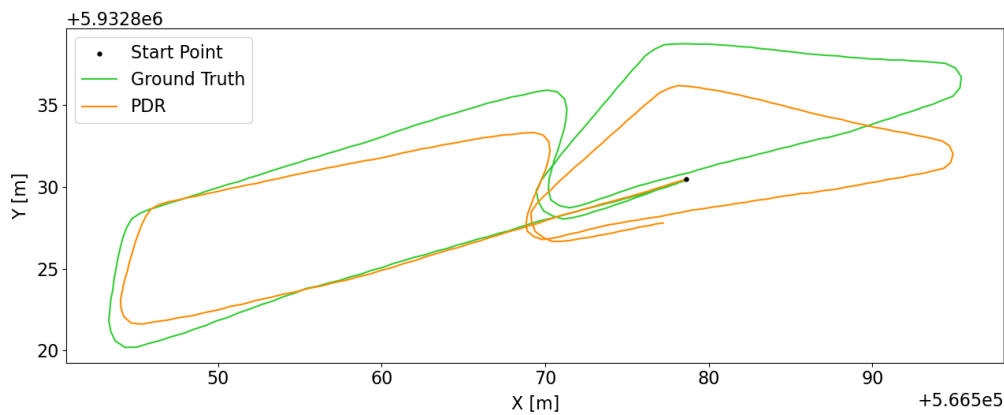
# 1 Einleitung

Vor allem in unübersichtlichen und unbekannten Gebäuden kann die Verwendung von Indoor Navigation nützlich sein, um die Positionen von Menschen oder Objekten zu verfolgen. Das Problem bei der Navigation im Gebäude ist der schlechte bis fehlende GNSS-Empfang. We-  
shalb das sonst übliche Verfahren, mittels Satellitennavigation im Gebäude meistens keine  
zufriedenstellende Positionslösung bietet und somit meistens keine zuverlässige Möglichkeit  
bei der Indoor Navigation darstellt. Da Inertiales Navigationssysteme (INS) ohne GNSS-Signale  
auskommen, können sie hervorragend in der Indoor-Navigation eingesetzt werden. Dafür wird  
ein einfacher Bewegungssensor benötigt, welcher sich beispielsweise in einem Ortungssystem  
für gehende Personen befindet, das so genannte *Personal Dead Reckoning (PDR)*-System.  
Das PDR-System verwendet eine Inertialmesseinheit (IMU). Diese wird am Schuh des Nutzers  
befestigt und liefert Informationen wie die Schrittlänge und Schrittrichtung. Diese Informationen  
können dabei helfen den Standpunkt des Nutzers relativ zu einem bekannten Referenzpunk-  
ten in Echtzeit zu ermitteln (Ojeda & Borenstein, 2007). In dieser Arbeit werden sogenannte  
5G-Koordinaten als bekannte Referenzpunkte verwendet.

Nachteile eines PDR-System ist das sie durch akkumulierte Messfehlern einen Drift aufweisen  
(siehe Abbildung 1.1). Wie man sieht, kann die PDR Trajektorie der Ground Truth am Anfang  
für etwa 10 - 15 Meter relativ genau folgen. Anschließend driftet die PDR Trajektorie allerdings  
zunehmend nach rechts ab.

Dies liegt daran, dass sie der Fehler der Schrittrichtung immer akkumuliert und so auch bei  
einem geringen Fehler mit der Zeit ein erheblicher Drift entsteht. Insgesamt lässt sich aber  
feststellen, dass die PDR Trajektorie für kurze Distanzen sehr gut geeignet ist. Dieser Drift  
kann jedoch mit Hilfe regelmäßiger Abgleichung von Referenzpunkten (in diesem Fall absolute  
5G-Koordinaten) korrigiert werden.

Durch diese Abgleichung entsteht eine optimierte Trajektorie, welche dann sowohl aus 5G-  
Koordinaten und PDR-Messwerten besteht. So nutzt man die 5G-Koordinaten zur absoluten  
Positionsbestimmung, während die PDR-Messungen durch ihre höhere Messrate dafür sorgen,  
dass der Verlauf der Trajektorie der Ground Truth nähert kommt. Dafür sind in der Geodäsie  
Optimierungsansätze wie das Kalman-Filter oder die Methode der kleinsten Quadrate (engl.:  
Least-Square) häufig im Einsatz.



**Figure 1.1** PDR Trajektorie und Ground Truth

Diese Arbeit beschäftigt sich daher mit der Frage wie sich die Ergebnisse einer Ausgleichung von Personal Dead Reckoning (PDR) mittels dem dynamischen Kalman-Filter und der Least-Square Methode unterscheiden?

Auch andere Arbeiten beschäftigen sich mit solchen Thema. Das Paper Kjellson et al. (2021) z.B. beschreibt Deep-Learning-Methoden sowohl für die Schätzung von absoluten Positionen als auch für die Durchführung einer Koppelnavigation für Fußgänger (PDR). Beiden Ansätze wurden zur Optimierung der Trajektorie mit Hilfe der gewichteten kleinsten Quadrate (engl.: weighted least square) kombiniert.

Was ist das Problem?

Warum ist es wichtig?

Was sind die Anwendungsbereiche des Themas?

Was sind die bisherigen Herausforderungen?

Was ist der Schwerpunkt dieser Arbeit?

## 1.1 Aufbau der Arbeit

In diese Arbeit ist aufgeteilt in drei Oberkapitel: Methodologie, Implementation und Fazit und Ausblick. In dem Kapitel Methodologie werden die generellen Konzepte des diskreten Kalman-

Filters, des extended Kalman-Filter und der des weighted Least-Square Methode dargestellt. Bei der Implementation wird die detaillierte Ausführung der Arbeitsschritte samt ihrer Ergebnissen vorgestellt. Damit werden die Ergebnisse der unterschiedlichen Verfahren miteinander verglichen. Zum Schluss werden die Ergebnisse zusammengefasst und kritisch analysiert. Als Ausblick wird/werden zukünftige Forschungsperspektiven dargelegt.

## 2 Methodologie

In diesem Kapitel werden die theoretischen Methoden und allgemeinen Konzepten des Kalman Filters und der Least Square-Methode beschrieben.

### 2.1 Das Kalman Filter

Während die meisten Systeme mit zahlreichen Sensoren ausgestattet sind führen diese mit Hilfe von Messungen eine Schätzung der unbekannten Parameter aus. Dabei kann es herausfordernd sein eine genaue und präzise Schätzung dieser Unbekannten unter Berücksichtigungen ihrer Unsicherheiten durchzuführen. Dafür wird nicht selten der Kalman-Filter zur Hilfe genommen. Er findet dort Anwendung, wo bestimmte Sensoren nicht funktionieren oder sogar ausfallen und man dennoch die Systemgrößen schätzen möchte. Somit liefert das Kalman-Filter eine Vorhersage des zukünftigen Systemzustands auf der Grundlage vergangener Schätzungen.

Der Filter ist nach Rudolf E. Kálmán (19. Mai 1930 - 2. Juli 2016) benannt wurden. Im Jahr 1960 veröffentlichte Kálmán seine berühmte Arbeit, in der er eine rekursive Lösung für das lineare Filterproblem mit diskreten Daten beschrieb (Quelle: <https://www.kalmanfilter.net/default.aspx>).

Bei dem Kalman Filter unterscheidet man zwei Arten von Filter: der statische und der dynamische Kalman Filter. Diese Arbeit wird sich mit dem dynamischen Kalman Filter auseinander setzen. Dieser setzt sich aus einem Beobachtungsmodell und einem Bewegungsmodell zusammen. Bei dem Beobachtungsmodell handelt es sich um Beobachtungen und ihre Unsicherheiten. Bei dem Bewegungsmodell wird auch von einer Prädiktion (dynamisches Modell) und ihren Unsicherheiten gesprochen. Aus ihr erfolgt dann die Schätzung des Zustandes.

Wenn sowohl Beobachtungsmodell als auch Bewegungsmodell linear sind wird der diskrete Kalman Filter verwendet. Wenn beide Modelle nicht-linear sind findet der Extended Kalman Filter Anwendung.

### 2.1.1 Diskreter Kalman-Filter (lineares Modell)

Bei dem diskreten Kalman-Filter ist die Idee die Beobachtungen zu bestimmten (diskreten) Zeitpunkten mit einem Bewegungsmodell zu kombinieren. Das Bewegungsmodell prädiziert dann den Zustandsvektor ausgehend von der Schätzung des vorherigen Schrittes. Generell werden vier Schritte benötigt: die Prädiktion, die Innovation, die Kalman-Gain-Matrix und das Update.

Das generelle Beobachtungsmodell sieht wie folgt aus:

$$l_{i+1} = A_{i+1}x_{i+1} + e_{i+1}, \quad (2.1)$$

mit  $l$  = Beobachtungen/Messungen,  $A$  = Designmatrix,  $x$  = unbekannte Parameter und  $e$  = Residuen (Beobachtungsrauschen). Die Designmatrix beinhaltet die partiellen Ableitungen der Beobachtungsgleichungen nach den Parametern. Bei den Residuen handelt es sich um die negierten Verbesserungen. Sie sind normalverteilt mit einem Erwartungswert von 0.

Das hier verwendete Bewegungsmodell sieht wie folgt aus:

$$x_{i+1} = T_i x_i + C_i w_i, \quad (2.2)$$

mit  $T$  = Transitionsmatrix (prädiziert Bewegung von einem Zeitpunkt zum nächsten),  $w$  = Störgröße (Unsicherheit im Bewegungsmodell: Rauschen),  $C$  = Störgrößenmatrix (Auswirkung dieser Unsicherheit auf die Prädiktion des Zustandes). Da wir annehmen, dass die Unsicherheiten im Bewegungsmodell ausreichend von den Varianzen der geschätzten Parameter abgedeckt werden, wird die Formel 2.2 durch die folgende Formeln ersetzt:

$$x_{i+1} = T_i x_i. \quad (2.3)$$

1. Prädiktion: Welche Trajektorie des Fahrzeugs sagt das Bewegungsmodell voraus?

In der Prädiktion wird der aktuelle Schritt mit den Daten des vorherigen Schrittes und der Transitionsmatrix berechnet. Die Transitionsmatrix beinhaltet die partiellen Ableitungen der prädizierten Größen nach den Schätzungen des vorigen Schrittes. Die zugehörige Kovarianz-

matrix wird mit Hilfe des Varianzfortpflanzungsgesetzes berechnet.

$$\bar{\mathbf{x}}_{i+1} = \mathbf{T}_i + \hat{\mathbf{x}}_i \quad (2.4)$$

$$\Sigma(\bar{\mathbf{x}}_{i+1}) = \mathbf{T}_i \Sigma(\hat{\mathbf{x}}_i) \mathbf{T}_i^T + \mathbf{C}_i \Sigma(\mathbf{w}_i) \mathbf{C}_i^T \quad (2.5)$$

2. Innovation: Welche Trajektorie des Fahrzeugs sagt das Bewegungsmodell voraus? Was behaupten die Beobachtungen? Wie sehr weichen die Beobachtungen von der Prädiktion ab? => Innovation

In dem 2. Schritt der Innovation wird die Differenz der Prädiktion zu den Beobachtungen berechnet und auch hier wird mit Hilfe des VFGs die Kovarianzmatrix gerechnet

$X_{5G}$  und  $Y_{5G}$ ,  $\Delta x_i$  und  $\Delta y_i$  = Differenz zwischen der aufsummierten schrittänge, und schrittrichtung.

$$\mathbf{d}_{i+1} = \mathbf{l}_{i+1} - \mathbf{A}_{i+1} \bar{\mathbf{x}}_{i+1} \quad (2.6)$$

$$\Sigma(\mathbf{d}_{i+1}) = \Sigma(\mathbf{l}_{i+1}) + \mathbf{A}_{i+1} \Sigma(\bar{\mathbf{x}}_{i+1}) \mathbf{A}_{i+1}^T \quad (2.7)$$

Dann folgt die Berechnung der Kalman Gain Matrix, die in dem Update die Innovation gewichtet. Die zugehörige Kov.matrix des Updates wird auch wieder mit dem VFG berechnet.

3. Gain Matrix (K-Matrix): Relative Gewichtung von Prädiktion und Beobachtungen anhand der jeweiligen Genauigkeiten

Kalman Gain Matrix setzt sich aus den Kov.Matrizen der prädiktion und der innovation zusammen. Je genauer die Kov matrix der innovation ist desto mehr gewicht erhalten die

$$\mathbf{K}_{i+1} = \Sigma(\bar{\mathbf{x}}_{i+1}) \mathbf{A}_{i+1}^T \Sigma^{-1}(\mathbf{d}_{i+1}) \quad (2.8)$$

4. Update:

Gewichtetes Mittel aus Prädiktion und Innovation

$$\hat{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_{i+1} + \mathbf{K}_{i+1} \mathbf{d}_{i+1} \quad (2.9)$$

$$\Sigma(\hat{\mathbf{x}}_{i+1}) = [\mathbf{I} - \mathbf{K}_{i+1} \mathbf{A}_{i+1}] \Sigma(\bar{\mathbf{x}}_{i+1}) \quad (2.10)$$

Geschichte des Kalman-Filters:

Erdunfen von Rudolf E. Kalman (Transcations of the ASME-Journal of Basic Engineering, 82 (Series D): 35-45.Copyright by ASME)

Gut geeignet, um die Bahnen von Raketen zu berechnen (der Apollo Mondmission)

Dynamisches Modell: Trajektorie der Mondrakete Beobachtungen: Space sextant, inertial navigator (Weltraumsextant, Trägheitsnavigator)

### 2.1.2 Extended Kalman-Filter (nicht- lineares Modell)

Weder Bewegungsmodell noch das Beobachtungsmodell ist linear, deshalb wird das extended Kalman Filter gebraucht.

Neues Bewegungsmodell:

$$\mathbf{x}_{i+1} = \mathbf{f}_i^{i+1}(\mathbf{x}_i, \mathbf{w}_i) \quad (2.11)$$

$$\mathbf{l}_{i+1} = \mathbf{a}_{i+1}(\mathbf{x}_{i+1}) + \mathbf{e}_{i+1} \quad (2.12)$$

Bei nicht-linearen Zusammenhängen werden die Matrizen A, T und C durch Linearisierung (partielle Ableitungen) der nicht-linearen Funktionen f und a bestimmt:

$$\mathbf{T}_i = \left. \frac{\partial \mathbf{f}_i^{i+1}(\mathbf{x}_i, \mathbf{w}_i)}{\partial \mathbf{x}_i} \right|_{\mathbf{x}_i = \hat{\mathbf{x}}_i} \quad (2.13)$$

$$\mathbf{C}_i = \left. \frac{\partial \mathbf{f}_i^{i+1}(\mathbf{x}_i, \mathbf{w}_i)}{\partial \mathbf{w}_i} \right|_{\mathbf{w}_i} \quad (2.14)$$

$$\mathbf{A}_{i+1} = \left. \frac{\partial \mathbf{a}_{i+1}(\mathbf{x}_{i+1})}{\partial \mathbf{x}_{i+1}} \right|_{\mathbf{x}_{i+1} = \bar{\mathbf{x}}_{i+1}} \quad (2.15)$$

### 2.1.3 Weitere Quellen:

<https://www.cbcity.de/das-kalman-filter-einfach-erklart-teil-2>

## 2.2 Weighted Least-Square

Weighted Least-Square basiert auf der Grundlage der Methode der kleinsten Quadrate. Dabei werden die Parameter so gewählt, dass die Quadratsumme der Verbesserungen möglichst



klein ist.

$$|\mathbf{v}|^2 = \sum_{i=1}^n v_i^2 \rightarrow \min \quad (2.16)$$

$$\mathbf{l} + \mathbf{v} = \mathbf{A}\mathbf{x} \text{ mit } \Sigma(\mathbf{l}) = \sigma_0^2 \mathbf{P}^{-1} \quad (2.17)$$

Wobei die linke Gleichung das funktionale Modell repräsentiert, in dem der funktionale Zusammenhang zwischen Beobachtungen und Unbekannten steckt. Die rechte Gleichung nennt sich auch das stochastische Modell und dient der Beschreibung der Messunsicherheit bzw. zur Einführung von Gewichten für die Beobachtungen. Dafür wird die Gewichtsmatrix  $\mathbf{P}$  eingeführt.

Schätzung der unbekannten Parameter:

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \mathbf{l} \quad (2.18)$$

## 3 Implementation

Detaillierte Ausführung der Arbeitsschritte einschließlich der Ergebnisse bei verschiedenen Implementierungsparametern.

Informationen über Datensatz, Ground Truth, Implementation in Python

### 3.1 Diskreter Kalman-Filter

Im ersten Schritt des Kalman Filters der Prädiktion wird eine Voraussage getroffen. Dies geschieht anhand des Bewegungsmodells:

$$\begin{pmatrix} x_i + \Delta x_i \\ y_i + \Delta y_i \\ \Delta x_i \\ \Delta y_i \end{pmatrix} \quad (3.1)$$

In diesem wird auf den vorangegangenen Punkt die folgenden  $\Delta x_i$  bzw.  $\Delta y_i$  Werten aufaddiert. Die  $\Delta$ -Werte wurden bereits im Vorlauf aus den PDR Werten berechnet. Die Umrechnung erfolgte durch folgende Formeln:

$$\Delta x_i = L_i \cdot \cos(\Phi_0 + R_i) \quad (3.2)$$

$$\Delta y_i = L_i \cdot \sin(\Phi_0 + R_i) \quad (3.3)$$

wobei  $L$  für die Schrittlänge und  $R$  für die Schrittrichtung gilt.  $\Phi_0$  ist dabei die Startrichtung, welche für diesen Datensatz auf 3.5 [radian] festgelegt worden ist. Im Kalman Filter funktioniert das aufaddieren mittels einer Transitionsmatrix  $T$ , welche die partiellen Ableitungen des Bewegungsmodell nach den geschätzten Parametern beinhaltet. In diesem Fall sind diese partiellen Ableitungen allerdings alle trivial und sind entweder 0 oder 1. So ergibt sich folgende Formel zur Berechnung der Prädiktion  $\bar{x}_{i+1}$ .

$$\bar{x}_{i+1} = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{T_i} \cdot \underbrace{\begin{pmatrix} x_i \\ y_i \\ \Delta x_i \\ \Delta y_i \end{pmatrix}}_{\hat{x}_i} = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \Delta x_{i+1} \\ \Delta y_{i+1} \end{pmatrix} \quad (3.4)$$

Dieser Prädiktionsschritt wird so lange durchgeführt, bis der Zeitpunkt erreicht worden ist, an dem eine neue 5G Beobachtung vorliegt. Neben der Voraussage  $\bar{x}_{i+1}$  wird auch die dazu gehörige Kovarianzmatrix aufgestellt:

$$\Sigma(\bar{x}_{i+1}) = \begin{pmatrix} \sigma_{x_i}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y_i}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\Delta x_i}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\Delta y_i}^2 \end{pmatrix} \quad (3.5)$$

Im zweiten Schritt, der Innovation, wird die Differenz  $d_{i+1}$  aus den Beobachtungen und der Prädiktion berechnet, da beide identisch aufgebaut sind, muss die Prädiktion nicht umgerechnet werden, aus diesem Grund ist die  $A$  Matrix eine Einheitsmatrix, welche also folglich keinen

Einfluss hat.

$$\mathbf{d}_{i+1} = \begin{pmatrix} X_{5G} \\ Y_{5G} \\ \Delta x_i \\ \Delta y_i \\ \mathbf{l}_i \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \mathbf{A}_{i+1} \end{pmatrix} \cdot \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \Delta x_{i+1} \\ \Delta y_{i+1} \\ \bar{\mathbf{x}}_{i+1} \end{pmatrix} \quad (3.6)$$

Die zugehörige Kovarianzmatrix der Beobachtungen wird wie folgt aufgestellt:

$$\Sigma(\mathbf{l}_{i+1}) = \begin{pmatrix} \sigma_{X_{5G}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{Y_{5G}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\Delta x_i}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\Delta y_i}^2 \end{pmatrix} \quad (3.7)$$

Die Ergebnisse der Zwischenschritte werden dann in die Formeln, die bereits im Kapitel 2.1.1 (3. Gain-Matrix und 4. Update) aufgelistet sind eingesetzt, um am Ende ein Update zu erhalten. Dieses Update wird dann im nächsten Schritt das neue  $\hat{x}_i$ .

## 3.2 Extended Kalman-Filter

## 3.3 Weighted-Least-Square

Für die Implentation der Weighted-Least-Square Methode werden in einem ersten Schritt die 5G Beobachtungen anhand der PDR Beobachtungen Interpoliert  $(x_i, y_i)$ , indem aus der Schrittlänge  $L$  und der Schrittrichtung  $R$ , Delta  $x$  und Delta  $y$  berechnet werden und diese auf die 5G Koordinaten aufaddiert werden. Für die Schrittrichtung ist die Nordrichtung  $\Phi_0$  essentiell, welche für den Beispiel Datensatz ei 3.5 rad liegt.

$$x_{i+1} = x_i + L_i \cdot \cos(\Phi_0 + R_i) \quad (3.8)$$

$$y_{i+1} = y_i + L_i \cdot \sin(\Phi_0 + R_i) \quad (3.9)$$

Dieses Interpolieren wird so häufig durchgeführt, bis eine neue 5G Beobachtung vorliegt. Als Ergebnis liegt nun für jede PDR Beobachtung eine 5G Beobachtung bzw. eine Interpolierte 5G

Beobachtung vor. Dies ist die Voraussetzung für das Anwenden der Least-Square-Methode. Anschließend können die Daten in die Schätzung der unbekannten Parameter eingehen (siehe Formel 2.18). In der Gewichtsmatrix  $P$  findet die Gewichtung der Beobachtungen statt, wobei absolute (5G Koordinaten) und relative Beobachtungen (PDR Werte) unterschiedliche Standardabweichungen und somit Gewichte erhalten. So wird gesteuert, welche Beobachtungen mehr Einfluss bei dem Ausgleich haben sollen und bei welchen die quadratische Verbesserung höher sein darf. Die Ausgleichung findet bei jedem dazukommenden 5G Punkt neu statt. Genauigkeit:

s1 (absolut, 5G): 3 s2 (relativ, PDR): 0.2

### 3.4 Vergleich der Ergebnisse

## 4 Fazit und Ausblick

Eine Erörterung (oder "Bewertung") der Ergebnisse, Zusammenfassung und zukünftige Forschungsperspektiven

## Literaturverzeichnis

Kjellson, C., Larsson, M., Astrom, K., & Oskarsson, M. (2021). Accurate indoor positioning based on learned absolute and relative models, 1–8.

Ojeda, L., & Borenstein, J. (2007). Personal dead-reckoning system for gps-denied environments, 1–6.