



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 30, 2022

---

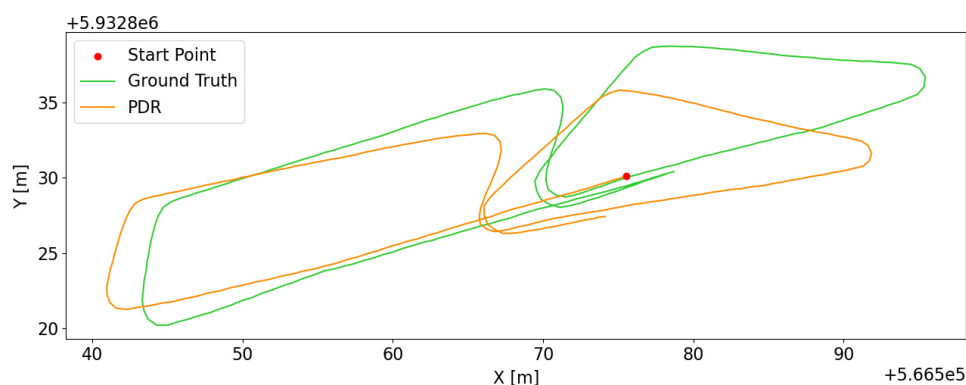
# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<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) . . . . .	5
2.2	Weigted Least-Square . . . . .	6
<b>3</b>	<b>Implementation</b>	<b>7</b>
3.1	Diskreter Kalman-Filter . . . . .	8
3.1.1	Ergebnisse . . . . .	9
3.2	Extended Kalman-Filter . . . . .	11
3.2.1	Ergebnisse . . . . .	11
3.3	Weighted-Least-Square . . . . .	11
3.3.1	Ergebnisse . . . . .	12
3.4	Vergleich der Ergebnisse . . . . .	13
<b>4</b>	<b>Fazit und Ausblick</b>	<b>14</b>
	<b>Bibliography</b>	<b>16</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. Deshalb 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 Referenzpunkten 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



**Figure 1.1** PDR Trajektorie und Ground Truth

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.

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.

Einige Herausforderungen, die während der Bearbeitung des Themas entstanden sind, war weniger das theoretische Verständnis, sondern viel mehr die Implementation in Python. Oft kommt es beim Programmieren zu Fehlermeldungen, die nicht immer so schnell zu lösen sind.

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.

Was sind die Anwendungsbereiche des Themas?

Was sind die bisherigen Herausforderungen?

## 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. Das Kalman-Filter liefert somit 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 (Becker, 2022).

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 (Balzer, 2013a).

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:

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

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 Kovarianzmatrix wird mit Hilfe des Varianzfortpflanzungsgesetzes (VFG) berechnet. Dabei wird sich die Frage gestellt welche Trajektorie des Fahrzeugs das Bewegungsmodell voraus sagt.

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

$$\Sigma(\bar{x}_{i+1}) = T_i \Sigma(\hat{x}_i) T_i^T + C_i \Sigma(w_i) C_i^T \quad (2.5)$$

In dem 2. Schritt der Innovation wird geschaut wie sehr die eigentlichen Beobachtungen von der zuvor berechneten Prädiktion abweichen, dafür wird die Differenz der Prädiktion zu den Beobachtungen berechnet und auch hier wird mit Hilfe des VFGs die Kovarianzmatrix gerechnet.

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

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

Dann folgt die Berechnung der Kalman Gain Matrix, welche die relative Gewichtung von Prädiktion und Beobachtungen anhand der jeweiligen Genauigkeiten beinhaltet. Die zugehörige Kovarianzmatrix des Updates wird auch wieder mit Hilfe des VFGs berechnet.

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

In dem vierten und letzten Schritt dem Update wird ein gewichtets Mittel aus der Prädiktion und der Innovation errechnet.

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

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

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

In den meisten Anwendungsbereichen sind weder Bewegungsmodell noch Beobachtungsmodell linear. Häufig schaut das System in eine Richtung und misst in die andere. Wenn man von nicht-linearen Funktionen spricht, werden damit häufig Winkelfunktionen gemeint, mit Sinus und Kosinus. Das Problem ist, dass nicht-lineare Funktionen zu keiner Normalverteilung führen und man dadurch keinen diskreten Kalman Filter verwendet kann. Eine Lösung des Problems ist die Linearisierung durch eine Approximation. Durch die Taylor Entwicklung kann so eine lineare Annäherung der nicht-linearen Funktionen geschehen. Nach Anwendung der Approximation erhält man einen erweiterten Kalman Filter den so genannten Extended Kalman Filter (Balzer, 2013b).

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:

$$A_{i+1} = \left. \frac{\partial a_{i+1}(x_{i+1})}{\partial x_{i+1}} \right|_{x_{i+1}=\bar{x}_{i+1}} \quad (2.13)$$

$$T_i = \left. \frac{\partial f_i^{i+1}(x_i, w_i)}{\partial x_i} \right|_{x_i=\hat{x}_i} \quad (2.14)$$

$$C_i = \left. \frac{\partial f_i^{i+1}(x_i, w_i)}{\partial w_i} \right|_{w_i} \quad (2.15)$$

Die Formeln für die Berechnung der vier Schritte: die Prädiktion, die Innovation, die Kalman Gain Matrix und das Update entsprechen den gleichen wie im Kapitel 2.1.1.

## 2.2 Weighted Least-Square

Die Least-Square Methode findet häufig Einsatz in der Ausgleichsrechnung. Die Idee ist es eine möglichst genau passende, parameterabhängige Kurve in einen Datensatz zu legen. Dabei werden die Parameter so gewählt, dass die Summe der quadratischen Abweichung der Kurve von den Beobachtungen minimiert wird (siehe Formel 2.16).

Vorteilhaft an dieser Methode ist die Erkennung von Ausreißern. Das Quadrieren der Verbesserungen (bzw. Fehlern) führt dazu, dass große Abweichungen große Werte erhalten womit sie einfacher sind aufzudecken. Dies können für den Ausgleich weniger gewichtet werden oder sogar komplett eliminiert. Anders als hier können bei der Weighted Least Square Methode die Varianzen für jede Beobachtung einzeln betrachtet werden. Je nachdem welche Beobachtungen genauer sind erhalten sie in der Gewichtsmatrix ein höheres Gewicht, womit sie mehr Einfluss in dem Ausgleich darstellen.

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

Das aufgestellte Weighted Least-Square Modell sieht wie folgt aus:

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

Die linke Gleichung repräsentiert das funktionale Modell, 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  $P$  eingeführt.



Um eine möglichst genau Kurve zu schätzen werden Parameter gesucht, welche mit folgender Formel berechnet werden können:

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

$A$  beschreibt wie beim Kalman Filter die Designmatrix. Die beinhaltet die partiellen Ableitungen der Beobachtungsgleichung nach den gesuchten Parametern.  $l$  beschreibt den Beobachtungsvektor und beinhaltet alle Messgrößen, welche beobachtet wurden.

### 3 Implementation

Detaillierte Ausführung der Arbeitsschritte einschließlich der Ergebnisse bei verschiedenen Implementierungsparametern.

In dieser Arbeiten werden für die 5G-Koordinaten die simulierte Datensätze 'nfg11' und 'nfg53' verwendet. Diese Datensätze sind identisch aufgebaut und beinhalteten neben dem Zeitstempel auch die X-, Y- und Z-Position der 5G-Koordinaten sowie ihre Genauigkeit. Der 'nfg'-Datensatz weist eine Messrate von einer Sekunde auf wohingegen der andere Datensatz ('nfg53') eine von fünf Sekunden ausweist. Doch auch die Genauigkeit der Datensätze unterscheiden sich. Bei dem 'nfg11'-Datensatz beträgt dieser einen Meter und bei dem anderen drei Meter. Somit kann gesagt werden, dass der 'nfg11'-Datensatz genauer bestimmt wurde als der andere.

Die PDR-Werte stammen aus zwei Datensätzen: 'nsl' für die Schrittlänge und 'nsh' für die Schrittrichtung. Die Genauigkeiten der beiden Datensätze betragen einmal 0.2 ('nsl') und 0.25 ('nsh'). Der zugehörige Zeitstempel liegt in der Datei 'EightStepTimes'. Wie man sieht liegen acht PDR-Werte zwischen zwei 5G-Koordinaten, womit der PDR-Datensatz deutlich mehr Beobachtungen ausweist als die 5G-Koordinaten.

Zu Kontrolle wird der Datensatz EightGroundTruth herangezogen, der die Wirklichkeit des Verlaufes der Trajektorie (Position?) beschreibt. Mit Hilfe dessen kann entschieden werden wie gut und richtig die unterschiedlichen Methoden funktionieren. Die Umsetzung der Methoden erfolgt mit Python.

### 3.1 Diskreter Kalman-Filter

Im ersten Schritt des Kalman Filters der Prädiktion wird eine Voraussage getroffen. Dies geschieht anhand des folgenden 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 steht.  $\Phi_0$  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 entweder 0 oder 1. Somit kann mit Hilfe der folgende Formel die Prädiktion  $\bar{x}_{i+1}$  berechnet werden.

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

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. Die im Beobachtungsvektor vorhandenen Größen sind neben dem 5G-Koordinaten  $X_{5G}$  und  $Y_{5G}$  auch die Differenz zwischen der aufsummierten Schrittlänge und -richtung  $\Delta x_i$  und  $\Delta y_i$ . Da beide Matrizen identisch aufgebaut sind, ist die  $A$  Matrix eine Einheitsmatrix, welche also folglich keinen Einfluss auf weitere Berechnungen hat.

$$d_{i+1} = \begin{pmatrix} X_{5G} \\ Y_{5G} \\ \Delta x_i \\ \Delta y_i \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \Delta x_{i+1} \\ \Delta y_{i+1} \end{pmatrix} \quad (3.6)$$

$\mathbf{l}_i \qquad \mathbf{A}_{i+1} \qquad \mathbf{\bar{x}}_{i+1}$

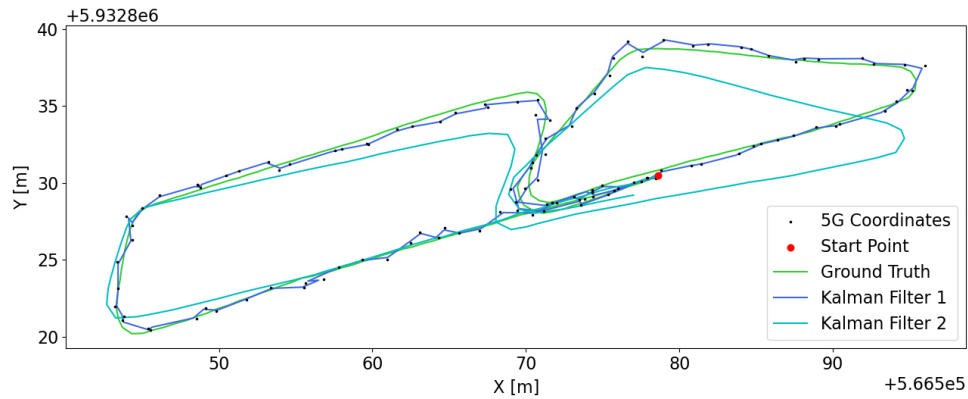
Die zugehörige Kovarianzmatrix der Beobachtungen wird wie folgt aufgestellt:

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

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  $\hat{x}_i$ .

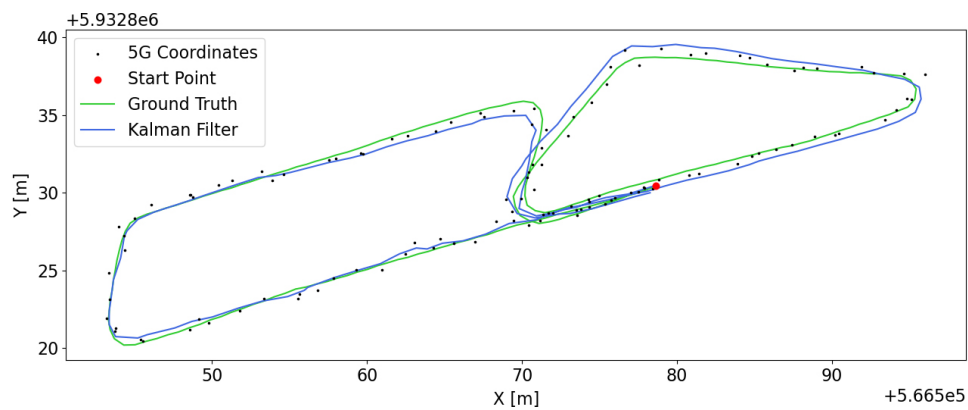
### 3.1.1 Ergebnisse

Das Ergebnis des Kalman Filters lässt sich steuern indem die Genauigkeiten der 5G Punkte bzw. der PDR Beobachtungen angepasst werden. Mit den Genauigkeiten wird die Kalman-Gain Matrix gebildet, welche im 4. Schritt des Kalman Filters die Beobachtungen gewichten. Das Ergebnis kann dementsprechend anders sein. In Abbildung 3.1 sind zwei Kalman Filter zu sehen, Kalman Filter 1 springt von 5G Koordinate zu 5G Koordinate, weil die Genauigkeit der 5G Koordinaten relativ zu der PDR Genauigkeit deutlich kleiner ist. Im Kalman Filter 2 ist dies umgekehrt, hier folgt der Kalman Filter fasst ausschließlich den PDR Werten. Beides sind Beispiele für einen schlecht eingestellten Kalman Filter. Die Genauigkeitswerte sollten so eingestellt werden, dass ein guter Kompromiss zwischen PDR und 5G Koordinaten gefunden wird und beide sich gut ausgleichen. In Abbildung 3.2 ist ein Beispiel mit einigermaßen gut



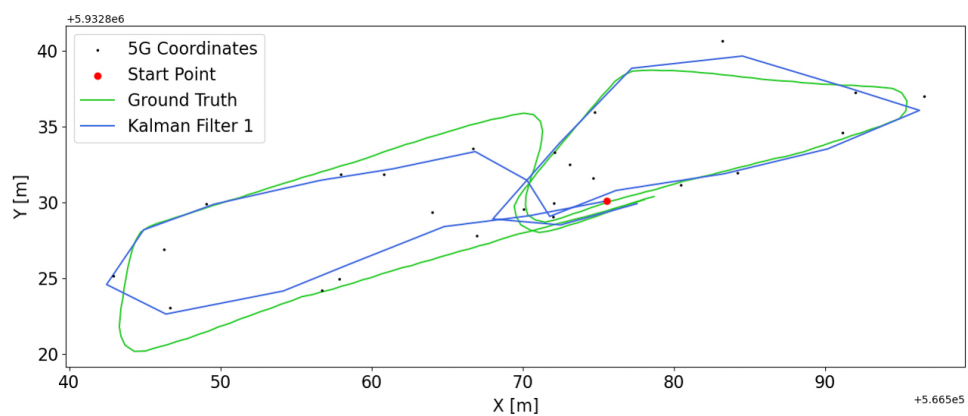
**Figure 3.1** Kalman Filter, 5G Koordinaten ('nfg11') und Ground Truth

austarierten Genauigkeitswerten. Das Ergebnis ist eine Trajektorie, welche nicht zu sehr auf die teils ungenauen 5G Koordinaten reagiert aber auch nicht auf Dauer von der Ground Truth wegdriftet. Abbildung 3.3 zeigt den gleichen Filter allerdings mit dem 5G Datensatz welcher



**Figure 3.2** Kalman Filter, 5G Koordinaten ('nfg11') und Ground Truth

wesentlich weniger Beobachtungen beinhaltet und ungenauer ist. Abbildung 3.3 zeigt die Tra-



**Figure 3.3** Kalman Filter, 5G Koordinaten ('nfg53') und Ground Truth

jektorie, welche sich durch den diskreten Kalman Filter ergibt, sowie die 5G Koordinaten und die Ground Truth. Hier ist deutlich zu sehen, dass der Filter nur so viele Punkte hat wie 5G Punkte in den Filter einfließen. Dadurch wirkt die Trajektorie sehr eckig und weist deutlich größere Abweichungen zur Ground Truth auf.

## 3.2 Extended Kalman-Filter

### 3.2.1 Ergebnisse

## 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. Je nachdem wie häufig eine neue 5G Beobachtung vorliegt wird entweder eine 5G Beobachtung für  $(x_i, y_i)$  herangezogen oder der vorangegangene interpolierte 5G Wert. Für die Schrittrichtung ist die Nordrichtung  $\Phi_0$  essentiell, welche für den Beispiel Datensatz bei 3.5 rad liegt.

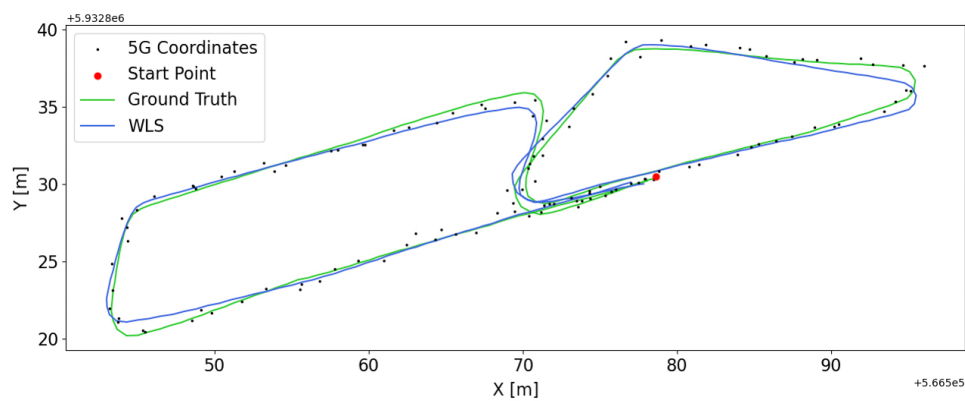
$$x_{i+1} = x_i + L_i \cdot \cos(\Phi_0 + R_i) \quad (3.8)$$

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

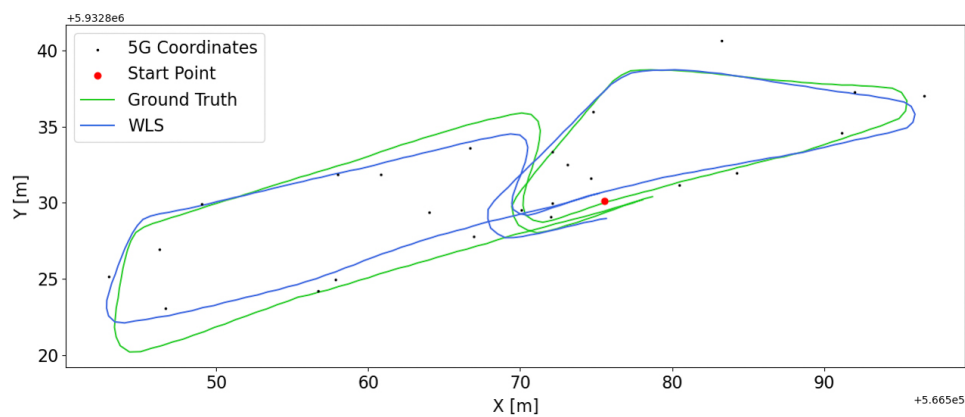
Dieses Interpolieren wird so häufig durchgeführt, bis eine neue 5G Beobachtung vorliegt. Als Ergebnis liegt nun für jede PDR Beobachtung eine 5G Beobachtung bzw. eine interpolierte 5G Beobachtung vor. Dies ist die Voraussetzung für das Anwenden der Least-Square-Methode. Anschließend können die Daten 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.

### 3.3.1 Ergebnisse

Die für die Weighted-Least-Square Methode verwendeten Genauigkeiten bilden auf der einen Seite die von den 5G Koordinaten (absolute) und auf der anderen Seite die für die PDR Werte (relativen). Das Ergebnis in Abbildung 3.4 zeigt den Datensatz 'nfg11', der weit aus mehr Beobachtungen hat als der 'nfg53' Datensatz. Hier wurden den PDR Beobachtungen mehr Gewicht gegeben womit sie eine sehr gute Genauigkeit aufweisen ( $\sigma_{PDR} = 0.2$ ). Im Gegensatz dazu wird den 5G Koordinaten weniger Einfluss in die Ausgleichung gegeben. Ihre Standardabweichung liegt deutlich höher als die von den PDR Beobachtungen ( $\sigma_{5G} = 3$ ). Das



**Figure 3.4** WLS, 5G Koordinaten ('nfg11') und Ground Truth



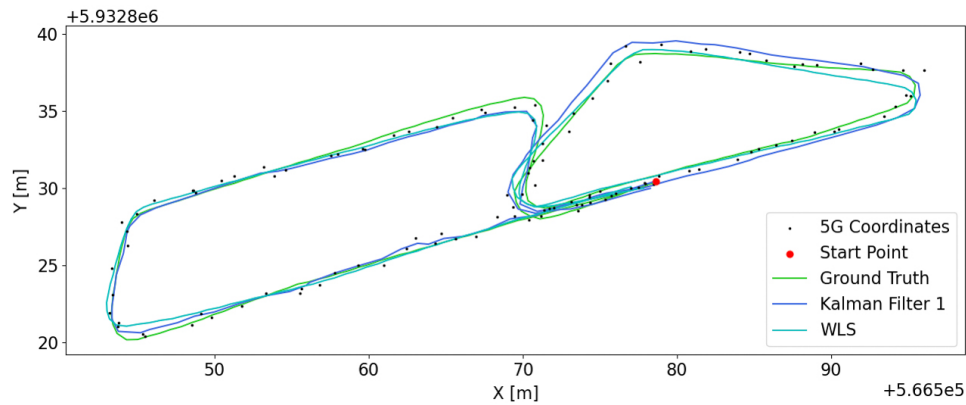
**Figure 3.5** WLS, 5G Koordinaten ('nfg53') und Ground Truth

Ergebnis zeigt, dass der Verlauf der WLS Ausgleichung der eigentlichen Wahrheit entspricht. Nur an einigen Kurvenbereichen gibt es einen kleinen Versatz. Das gleiche Muster ist auch in der Abbildung 3.5 zu sehen. Obwohl dieser Datensatz eine geringe Menge an Beobachtungen aufweist als der zuvor, kann hier auch beobachtet werden, dass der grobe Verlauf der

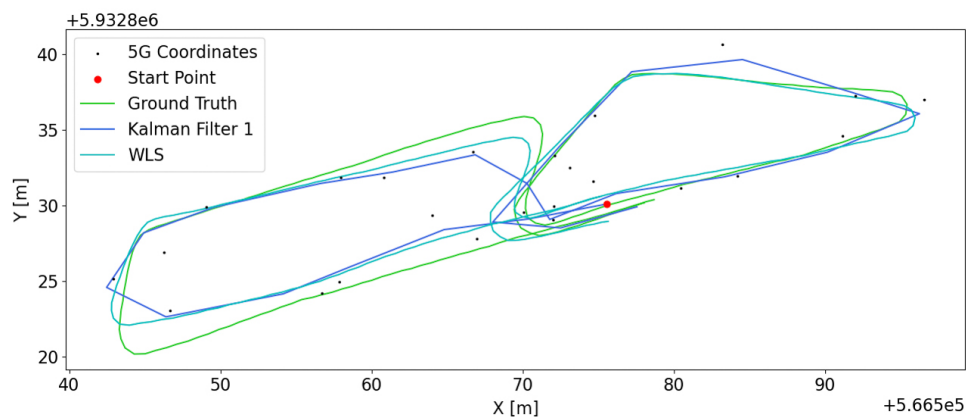
tatsächlichen Trajektorie entspricht. Die Einstellung der Standardabweichung sowohl für die 5G Koordinaten als auch für die PDR Beobachtungen sind unveränderlich. Sie entsprechen also den gleichen wie bereits in Abbildung 3.4 zu sehen ist.

### 3.4 Vergleich der Ergebnisse

Abbildung 3.6 und 3.7 zeigen den Vergleich von Weigthed-Least-Square und dem Kalman Filter mit der Ground Truth bei beiden 5G Datensätzen. Beim 'nfg11' Datensatz zeigen sich beide Trajektorien relativ gut, lediglich beim rechten Teil zeigt der Kalman Filter etwas größere Differenzen zur Ground Turth als WLS. Beim schlechteren 'nfg53' Datensatz werden die Unterschiede schon etwas größer.



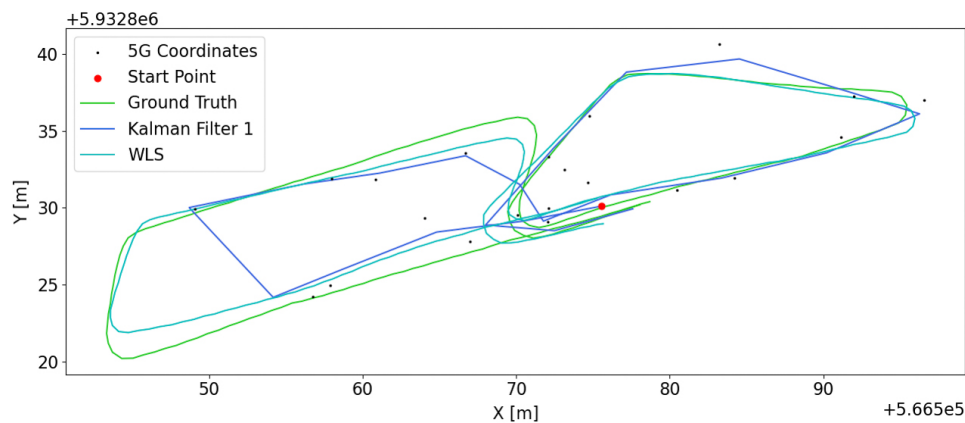
**Figure 3.6** Vergleich zwischen Kalman Filter und WLS ('nfg11')



**Figure 3.7** Vergleich zwischen Kalman Filter und WLS ('nfg53')

Vergleicht man den Kalman Filter mit der Weighted-Least-Square Methode so fällt auf, dass

der Kalman Filter insgesamt viel eckiger ausfällt als die vergleichsweise runde Trajektorie von WLS. Gerade beim 'nfg53' Datensatz wird dies deutlich. Dies liegt daran, dass der Kalman Filter nur bei jeder neuen 5G Beobachtung updatet und eine neue Position berechnet. Abbildung 3.8 verdeutlicht dies. Hier wurde eine größere Beobachtungslücke simuliert bei der die Ground Truth zwei Kurven zurücklegt. Während der Kalman Filter fast auf direktem Weg zur nächsten 5G Beobachtung abkürzt, simuliert die Least-Square Trajektorie die beiden Kurven einigermaßen gut. Hier schneidet die WLS Methode besser ab als der Kalman Filter.



**Figure 3.8** Vergleich zwischen Kalman Filter und WLS bei Beobachtungslücke

## 4 Fazit und Ausblick

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

Zusammenfassend kann vorab festgehalten werden, dass die Trajektorie der Weighted-Least-Square Methode besser abschneidet als die des diskreten Kalman Filters. Das zeigt sich auch noch mal an der Abbildung 3.8, in der eine Beobachtungslücke simuliert wurde. Dadurch dass die WLS Methode interpolierte 5G Werte mit in die Ausgleichung nimmt, kann die gesamte Trajektorie besser bestimmt werden als bei dem diskreten Kalman Filter.

Als zukünftige Forschungsperspektive könnte der Extended Kalman Filter noch mehr erweitert werden. Der Vektor der geschätzten Parameter  $\hat{x}_i$ , der eine Dimension von 4x1 aufweist könnte auf eine Dimension von 6x1 erweitert werden. Dafür kommen neben dem  $x_i$ ,  $y_i$ ,  $L_i$  und  $R_i$  noch die Fehler der Schrittlänge und -richtung ( $L_{error_i}, R_{error_i}$ ) hinzu. Es wird vermutet, dass



---

durch Berücksichtigungen der Ungenauigkeit für Schrittlänge und -richtung das Vorhersage noch besser funktionieren. Ob das der Wirklichkeit entspricht, kann dieser Ansatz in Zukunft überprüft werden.

## Literaturverzeichnis

- Balzer, P. (2013a). *Das kalman filter einfach erklärt [teil 1]*. <https://www.cbcity.de/das-kalman-filter-einfach-erklaert-teil-1>
- Balzer, P. (2013b). *Das kalman filter einfach erklärt [teil 2]*. <https://www.cbcity.de/das-kalman-filter-einfach-erklaert-teil-2>
- Becker, A. (2022). *About the kalman filter*. <https://www.kalmanfilter.net/default.aspx>
- 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.