Inhaltsverzeichnis

Abbildungsverzeichnis

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. Weshalb 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 sogenannte *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. Ohne einen Startpunkt als Referenz ist eine absolute Positionierung nicht möglich. In dieser Arbeit werden absolute 5G-Koordinaten als bekannte Referenzpunkte verwendet. Diese neue 5G-Technologie zur Standortbestimmung bietet eine Vielzahl von Mobilfunk-basierten und hybriden Ortungsdiensten an. Die sowohl absolute als auch relative Positionierungen wiedergeben [1, 8]. Die zu erwartende Genauigkeit von 5G soll sich laut Frauenhofer [9] in den nächsten Jahren stetig verbessern.

Nachteile eines PDR-System ist, dass sie durch akkumulierte Messfehlern einen Drift aufweisen (siehe Abbildung 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

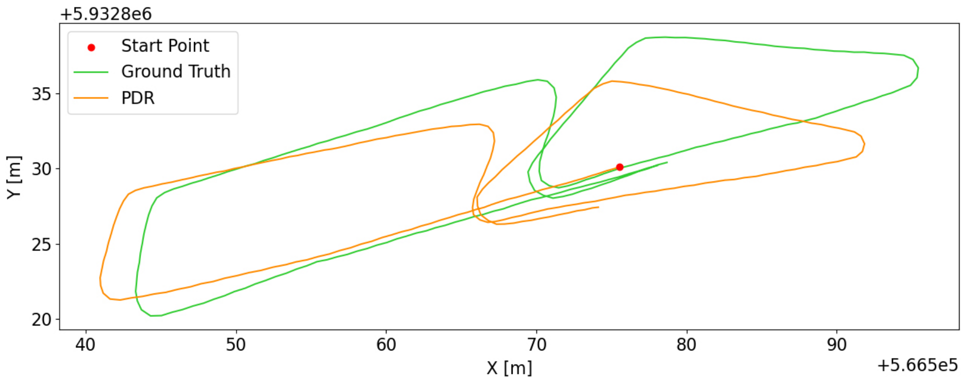

Abbildung 1: PDR Trajektorie und Ground Truth

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 mithilfe regelmäßiger Abgleichung von Referenzpunkten (in diesem Fall absolute 5G-Koordinaten) korrigiert werden. Die Kombination von den relativ ungenauen mittels 5G beobachteten Koordinaten und dem genaueren aber Drift aufweisenden PDR Messungen soll zusammen beide Nachteile verringern und so zu einem guten Ergebnis führen.

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 näher 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 des dynamischen Kalman-Filters und der Least-Square Methode unterscheiden? Auch andere Arbeiten beschäftigen sich mit diesem Thema. Das Paper [2] 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 mithilfe der gewichteten kleinsten Quadrate (engl.: weighted least square) kombiniert.

Einige Herausforderungen, die während der Bearbeitung des Themas entstanden sind, waren 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.

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 ihren 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. 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 [3].
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 auseinandersetzen. 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 [5, 6].
Das generelle Beobachtungsmodell sieht wie folgt aus:

$$\boldsymbol{l}_{i+1} = \boldsymbol{A}_{i+1}\boldsymbol{x}_{i+1} + \boldsymbol{e}_{i+1}, \tag{1}$$

mit $\boldsymbol{l}$ = Beobachtungen/Messungen, $\boldsymbol{A}$ = Designmatrix, $\boldsymbol{x}$ = unbekannte Parameter und $\boldsymbol{e}$ = 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:

$$\boldsymbol{x}_{i+1} = \boldsymbol{T}_i \boldsymbol{x}_i + \boldsymbol{C}_i \boldsymbol{w}_i, \tag{2}$$

mit $\boldsymbol{T}$ = Transitionsmatrix (prädiziert Bewegung von einem Zeitpunkt zum nächsten), $\boldsymbol{w}$ = Störgröße (Unsicherheit im Bewegungsmodell: Rauschen), $\boldsymbol{C}$ = 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 durch die folgende Formel ersetzt:

$$\boldsymbol{x}_{i+1} = \boldsymbol{T}_i \boldsymbol{x}_i. \tag{3}$$

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 mithilfe des Varianzfortpflanzungsgesetzes (VFG) berechnet. Dabei wird sich die Frage gestellt, welche Trajektorie des Fahrzeugs das Bewegungsmodell voraussagt.

$$\bar{\boldsymbol{x}}_{i+1} = \boldsymbol{T}_i + \hat{\boldsymbol{x}}_i \tag{4}$$

$$\boldsymbol{\Sigma}(\bar{\boldsymbol{x}}_{i+1}) = \boldsymbol{T}_i \boldsymbol{\Sigma}(\hat{\boldsymbol{x}}_i) \boldsymbol{T}_i^T + \boldsymbol{C}_i \boldsymbol{\Sigma}(\boldsymbol{w}_i) \boldsymbol{C}_i^T \tag{5}$$

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 mithilfe des VFGs die Kovarianzmatrix gerechnet.

$$\boldsymbol{d}_{i+1} = \boldsymbol{l}_{i+1} - \boldsymbol{A}_{i+1} \bar{\boldsymbol{x}}_{i+1} \tag{6}$$

$$\boldsymbol{\Sigma}(\boldsymbol{d}_{i+1}) = \boldsymbol{\Sigma}(\boldsymbol{l}_{i+1}) + \boldsymbol{A}_{i+1} \boldsymbol{\Sigma}(\bar{\boldsymbol{x}}_{i+1}) \boldsymbol{A}_{i+1}^T \tag{7}$$

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

$$\boldsymbol{K}_{i+1} = \boldsymbol{\Sigma}(\bar{\boldsymbol{x}}_{i+1}) \boldsymbol{A}_{i+1}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{d}_{i+1}) \tag{8}$$

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

$$\hat{\boldsymbol{x}}_{i+1} = \bar{\boldsymbol{x}}_{i+1} + \boldsymbol{K}_{i+1} \boldsymbol{d}_{i+1} \tag{9}$$

$$\boldsymbol{\Sigma}(\hat{\boldsymbol{x}}_{i+1}) = [\boldsymbol{I} - \boldsymbol{K}_{i+1} \boldsymbol{A}_{i+1}] \boldsymbol{\Sigma}(\bar{\boldsymbol{x}}_{i+1}) \tag{10}$$

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 verwenden 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 sogenannten Extended Kalman Filter [4, 6].

Neues Bewegungsmodell:

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}_i^{i+1}(\boldsymbol{x}_i, \boldsymbol{w}_i) \tag{11}$$

$$\boldsymbol{l}_{i+1} = \boldsymbol{a}_{i+1}(\boldsymbol{x}_{i+1}) + \boldsymbol{e}_{i+1} \tag{12}$$

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

$$\boldsymbol{A}_{i+1} = \left.\frac{\partial \boldsymbol{a}_{i+1}(\boldsymbol{x}_{i+1})}{\partial \boldsymbol{x}_{i+1}}\right|_{x_{i+1}=\bar{x}_{i+1}} \tag{13}$$

$$\boldsymbol{T}_i = \left.\frac{\partial \boldsymbol{f}_i^{i+1}(\boldsymbol{x}_i, \boldsymbol{w}_i)}{\partial \boldsymbol{x}_i}\right|_{x_i=\hat{x}_i} \tag{14}$$

$$\boldsymbol{C}_i = \left.\frac{\partial \boldsymbol{f}_i^{i+1}(\boldsymbol{x}_i, \boldsymbol{w}_i)}{\partial \boldsymbol{w}_i}\right|_{w_i} \tag{15}$$

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 Weigted Least-Square

Die Least-Square Methode findet häufig Einsatz in der Ausgleichungsrechnung. 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 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.

$$|\boldsymbol{v}|^2 = \Sigma_{i=1}^{n} v_i^2 \rightarrow min \tag{16}$$

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

$$\boldsymbol{l} + \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x} \text{ mit } \boldsymbol{\Sigma}(\boldsymbol{l}) = \sigma_0^2 \boldsymbol{P}^{-1} \tag{17}$$

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 $\boldsymbol{P}$ eingeführt.

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

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{A}^T \boldsymbol{P} \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{P} \boldsymbol{l} \tag{18}$$

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

3 Implementation

In diesem Kapitel Implementation wird die detaillierte Ausführung der Arbeitsschritte einschließlich der Ergebnisse bei verschiedenen Implementierungsparametern beschrieben. Die für diese Arbeiten verwendet simulierten Datensätze 'nfg11' und 'nfg53' beinhalten absolute 5G-Koordinaten. 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. Mithilfe 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 mithilfe des folgenden Bewegungsmodells:

$$\begin{pmatrix} x_i + \Delta x_i \\ y_i + \Delta y_i \\ \Delta x_i \\ \Delta y_i \end{pmatrix} \tag{19}$$

In diesem wird auf den vorangegangenen Punkt die folgenden Δx_i bzw. Δy_i Werten aufaddiert. Die Δ-Werte wurden bereits im Vorlauf aus den PDR Werten berechnet. Die Umrechnung erfolgte durch folgende Formeln:

$$\Delta x_i = L_i \cdot \cos(\Phi_0 + R_i) \tag{20}$$

$$\Delta y_i = L_i \cdot \sin(\Phi_0 + R_i), \tag{21}$$

wobei L für die Schrittlänge und R für die Schrittrichtung steht. Φ_0 ist dabei die Startrichtung, welche für diesen Datensatz auf 3.5 [radiant] festgelegt worden ist. Im Kalman Filter funktioniert das Aufaddieren mittels einer Transitionsmatrix $\boldsymbol{T}$, welche die partiellen Ableitungen des Bewegungsmodells nach den geschätzten Parametern beinhaltet. In diesem Fall sind diese partiellen Ableitungen entweder 0 oder 1. Somit kann mithilfe der folgenden Formel die Prädiktion $\bar{\boldsymbol{x}}_{i+1}$ berechnet werden.

$$\bar{\boldsymbol{x}}_{i+1} = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{T}_i} \cdot \underbrace{\begin{pmatrix} x_i \\ y_i \\ \Delta x_i \\ \Delta y_i \end{pmatrix}}_{\hat{\boldsymbol{x}}_i} = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \Delta x_{i+1} \\ \Delta y_{i+1} \end{pmatrix} \tag{22}$$

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

$$\boldsymbol{\Sigma}(\bar{\boldsymbol{x}}_{i+1}) = \begin{pmatrix} \sigma^2_{x_i} & 0 & 0 & 0 \\ 0 & \sigma^2_{y_i} & 0 & 0 \\ 0 & 0 & \sigma^2_{\Delta x_i} & 0 \\ 0 & 0 & 0 & \sigma^2_{\Delta y_i} \end{pmatrix} \tag{23}$$

Im zweiten Schritt, der Innovation, wird die Differenz $\boldsymbol{d}_{i+1}$ aus den Beobachtungen und der Prädiktion berechnet. Die im Beobachtungsvektor vorhanden Größen sind neben den 5G-Koordinaten X_{5G} und Y_{5G} auch die Differenz zwischen der aufsummierten Schrittlänge und -richtung Δx_i und Δy_i. Da beide Matrizen identisch aufgebaut sind, ist die $\boldsymbol{A}$ Matrix eine Einheitsmatrix, welche folglich keinen Einfluss auf weitere Berechnungen hat.

$$\boldsymbol{d}_{i+1} = \underbrace{\begin{pmatrix} X_{5G} \\ Y_{5G} \\ \Delta x_i \\ \Delta y_i \end{pmatrix}}_{\boldsymbol{l}_i} - \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{A}_{i+1}} \cdot \underbrace{\begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \Delta x_{i+1} \\ \Delta y_{i+1} \end{pmatrix}}_{\bar{\boldsymbol{x}}_{i+1}} \tag{24}$$

Die zugehörige Kovarianzmatrix der Beobachtungen wird wie folgt aufgestellt:

$$\boldsymbol{\Sigma}(\boldsymbol{l}_{i+1}) = \begin{pmatrix} \sigma^2_{X_{5G}} & 0 & 0 & 0 \\ 0 & \sigma^2_{Y_{5G}} & 0 & 0 \\ 0 & 0 & \sigma^2_{\Delta x_i} & 0 \\ 0 & 0 & 0 & \sigma^2_{\Delta y_i} \end{pmatrix} \tag{25}$$

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 $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$.

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 2 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 fast ausschließlich den PDR Werten. Beides sind Beispiele für einen schlecht eingestellten

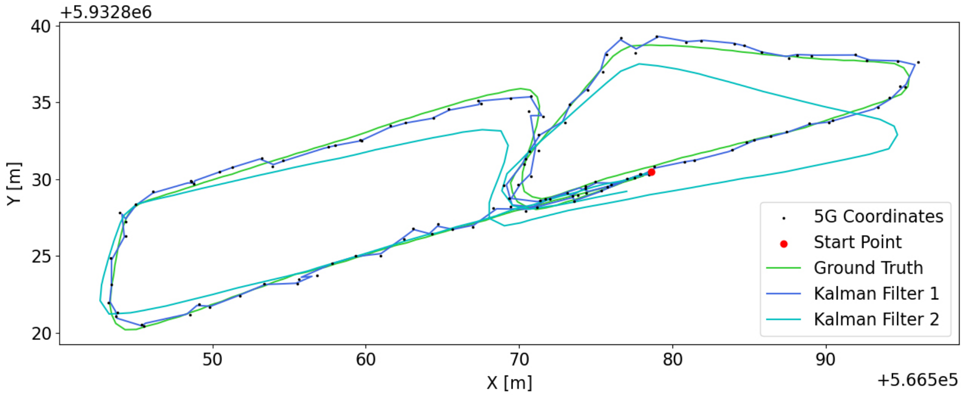

Abbildung 2: Kalman Filter, 5G-Koordinaten ('nfg11') und Ground Truth

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 ist ein Beispiel mit einigermaßen gut 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 4 zeigt den gleichen Filter allerdings mit dem 5G Daten-

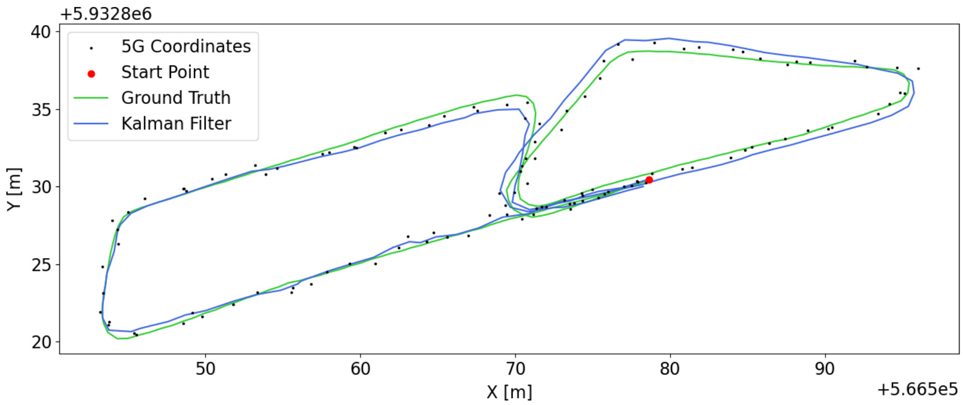

Abbildung 3: Kalman Filter, 5G-Koordinaten ('nfg11') und Ground Truth

satz welcher wesentlich weniger Beobachtungen beinhaltet und ungenauer ist. Abbildung 4 zeigt die Trajektorie, 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 weißt deutlich größere Abweichungen zur Ground Truth auf.

Das Ergebnis der Trajektorie und somit auch die Präzision der Trajektorie hängt maßgeblich von den Genauigkeitsparametern ein, welche in der Innovation mit in den Algorithmus einfließen.

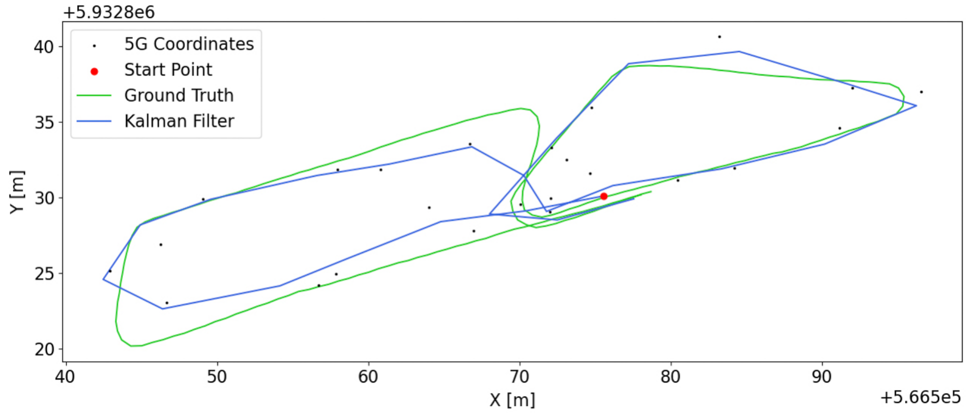

Abbildung 4: Kalman Filter, 5G-Koordinaten ('nfg53') und Ground Truth

Abbildung 5 zeigt verschiedene Trajektorien, bei denen jeweils die Varianz von $\Delta_{x,y}$ angepasst wurde. Die Varianz der 5G-Koordinaten (0.0001) ist in allen Trajektorien gleich geblieben. Es zeigt sich folgender Zusammenhang: Je größer die Varianz von $\Delta_{x,y}$ wird, desto näher orientiert sich die Trajektorie an den 5G-Koordinaten, was mit den Nachteilen einer springenden Trajektorie mit einhergeht. Dies liegt daran, dass mit steigender Varianz $\Delta_{x,y}$ den PDR Beobachtungen zunehmend weniger vertraut wird und die 5G-Koordinaten wichtiger werden zur Bildung der Trajektorie. Andersherum zeigt sich, dass bei kleineren Varianzen sich die Trajektorie zunehmend von der Ground Truth entfernt und sich einer reinen PDR Trajektorie (siehe Abb. 1) annähert. Auch hier gilt nun, dass der Algorithmus den PDR Werten im Vergleich zu den 5G Werten zu sehr vertraut, da die Varianzen sehr niedrig sind.

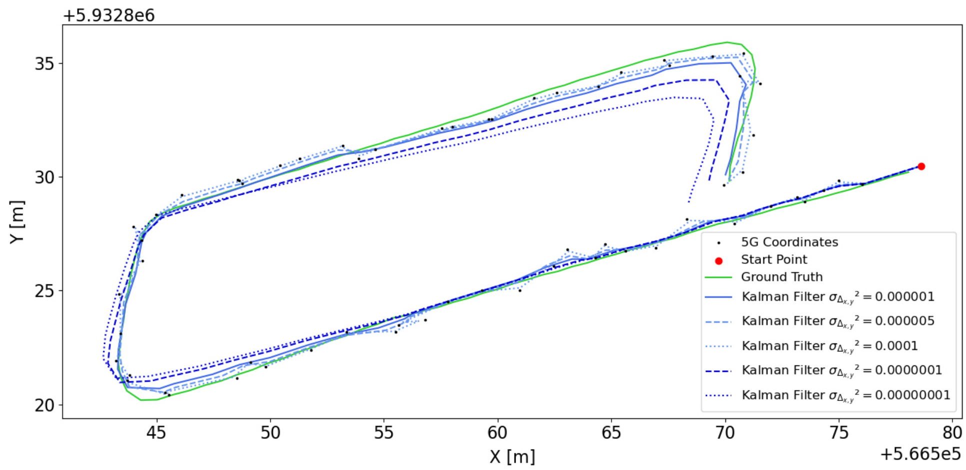

Abbildung 5: Kalman Filter Vergleich $\sigma_{\Delta_{x,y}}{}^2$

Abbildung 6 zeigt einen ähnlichen Zusammenhang, nur dass hier nun die PDR Varianzen (0.000001) unverändert sind und nur die Varianzen von den 5G-Koordinaten verändert worden. Das Ergebnis ist ähnlich, erhöht man die Varianzen von den 5G-Koordinaten driftet die Trajektorie wie die reinen PDR Beobachtungen. Senken wir die Varianzen, dann springt die Trajektorie von 5G-Koordinate zu 5G-Koordinate. Es lässt sich also feststellen, dass die Varianzen gut austariert werden müssen, um ein zufriedenstellendes Ergebnis zu erhalten, welches die beiden Beobachtungsverfahren gut kombiniert und sich nicht zu sehr auf ein Beobachtungsverfahren orientiert.

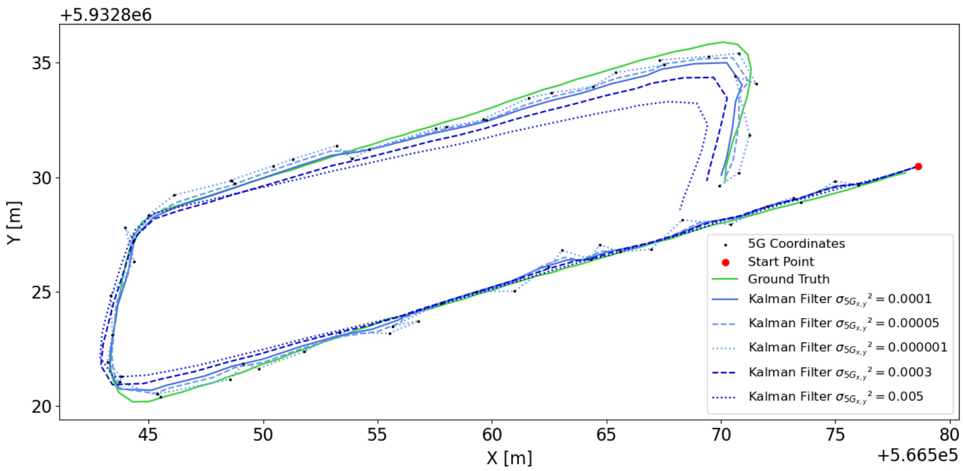

Abbildung 6: Kalman Filter Vergleich $\sigma_{5G_{x,y}}{}^2$

3.2 Extended Kalman-Filter

3.2.1 Ergebnisse

Die Implementation des Extended Kalman Filters konnte leider nicht beendet werden, da Fehler im Code nicht gefunden werden konnten. Die Ergebnisse werden daher im weiteren Verlauf nicht diskutiert und berücksichtigt.

3.3 Weighted-Least-Square

Für die Implementation 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, Δx und Δy 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 Φ_0 essenziell, welche für den Beispieldatensatz bei 3.5 rad liegt.

$$x_{i+1} = x_i + L_i \cdot \cos(\Phi_0 + R_i) \tag{26}$$

$$y_{i+1} = y_i + L_i \cdot \sin(\Phi_0 + R_i) \tag{27}$$

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 18). In der Gewichtsmatrix $\boldsymbol{P}$ 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 7 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 (σ_{PDR} = 0.2). Im Gegensatz dazu wird den 5G-Koordinaten weniger Einfluss in die Ausgleichung gegeben. Ihre Standardabweichung liegt deutlicher höher als die von den PDR Beobachtungen (σ_{5G} = 3).

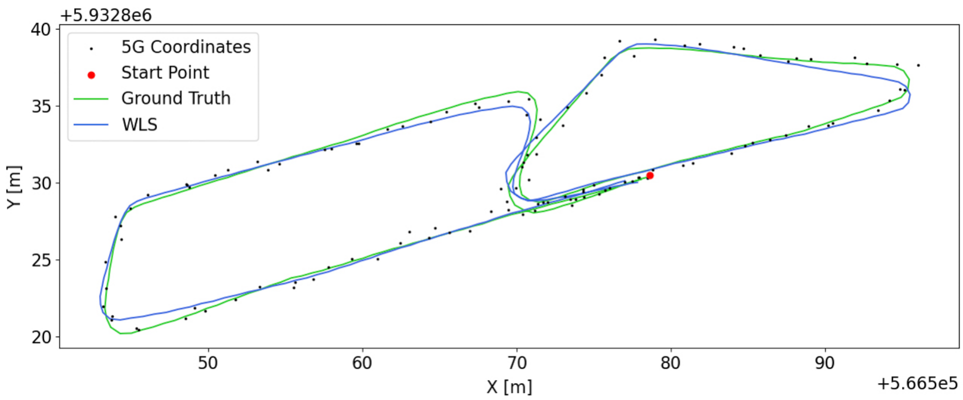

Abbildung 7: WLS, 5G-Koordinaten ('nfg11') und Ground Truth

Das 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 8 zu sehen. Obwohl dieser Datensatz eine geringe Menge an Beobachtungen aufweist als

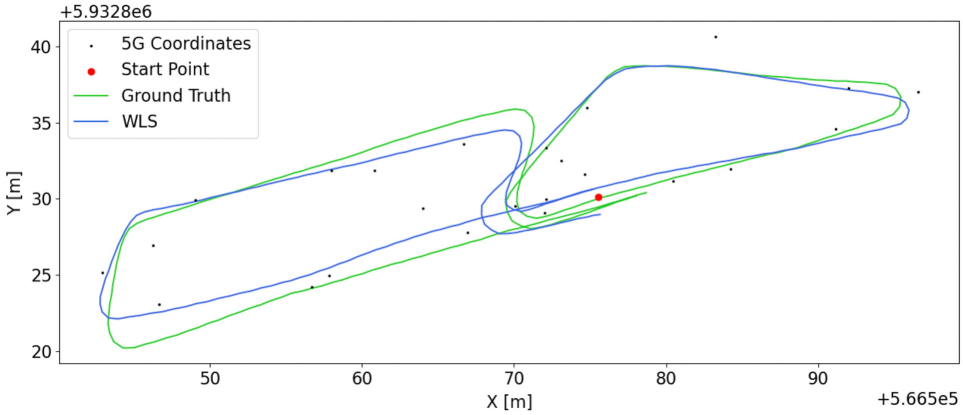

Abbildung 8: WLS, 5G-Koordinaten ('nfg53') und Ground Truth

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 7 zu sehen ist.

Auch bei der Least Square Methode ist das Ergebnis abhängig von den Genauigkeitsparametern welche steuern, inwiefern die PDR Daten und die 5G-Koordinaten gewichtet werden. Je nach Gewichtung kann entweder den 5G-Koordinaten mehr vertraut werden oder den PDR Beobachtungen. Die Genauigkeitsparameter sind in dem Algorithmus als s_1 und s_2 implementiert. s_1 steht für die absolute Genauigkeit, also die Genauigkeit der 5G-Koordinaten und s_2 steht für die relative Genauigkeit, also die der PDR Beobachtungen.

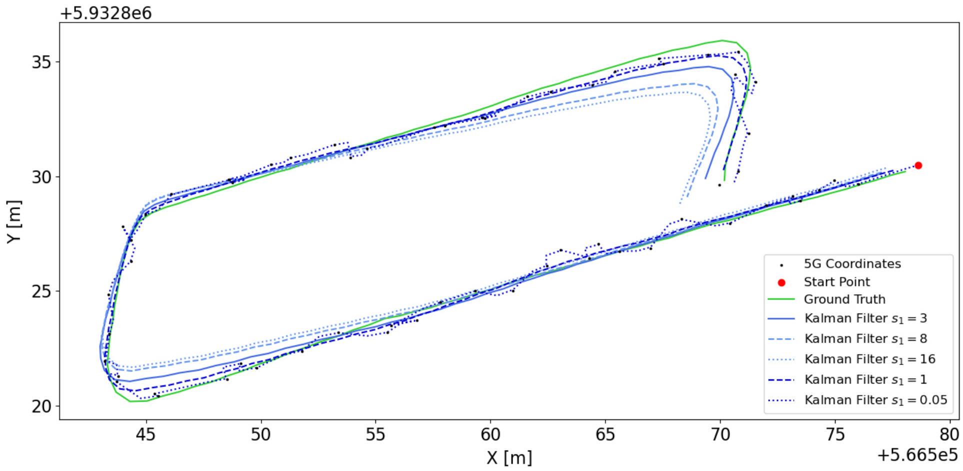

Abbildung 9: Least Square s_1

Die Abbildung 9 zeigt die Auswirkungen, wenn der s_2 Wert (0.2) konstant bleibt und der s_1 Wert verändert wird. Es zeigt sich, dass je größer s_1 gewählt wird, desto weniger wird den absoluten (5G-Koordinaten) vertraut und desto mehr nähert sich die Trajektorie der reinen PDR Trajektorie an (siehe Abbildung 1). Umgekehrt gilt, je kleiner s_1 ist, desto mehr wird den absoluten Werten vertraut und die Trajektorie nähert sich den 5G-Koordinaten an. Beide Extrembeispiele sind jedoch mit Nachteilen im Ergebnis verbunden: entweder driftet die Trajektorie von der Ground Truth weg oder die Trajektorie beinhaltet alle Ungenauigkeiten der 5G-Koordinaten.

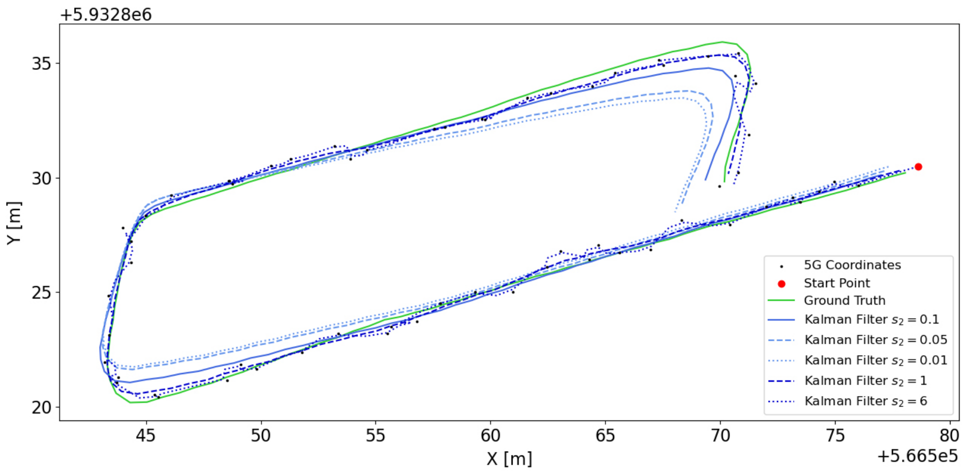

Abbildung 10: Least Square s_2

Die Abbildung 10 zeigt die Auswirkungen, wenn der s_1-Wert (= 3) konstant bleibt und der s_2 Wert verändert wird. Es zeigt sich, dass je größer s_2 gewählt wird, desto weniger wird den relativen (PDR Beobachtungen) vertraut und desto mehr nähert sich die Trajektorie den 5G-Koordinaten an. Umgekehrt gilt, je kleiner s_2, desto mehr wird den relativen Werten vertraut und die Trajektorie nähert sich der reinen PDR Trajektorie (siehe Abbildung 1) an. Dies führt ebenfalls wieder dazu, dass die Trajektorie entweder von der Ground Truth weg driftet oder alle Ungenauigkeiten der 5G-Koordinaten beinhaltet.

3.4 Vergleich der Ergebnisse

Abbildung 11 und 12 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 Truth als WLS. Beim schlechteren 'nfg53' Datensatz werden die Unterschiede schon etwas größer. 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

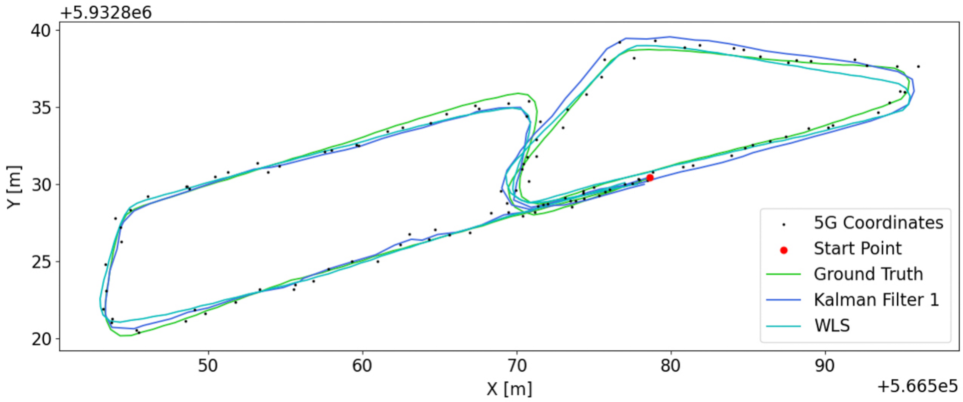

Abbildung 11: Vergleich zwischen Kalman Filter und WLS ('nfg11')

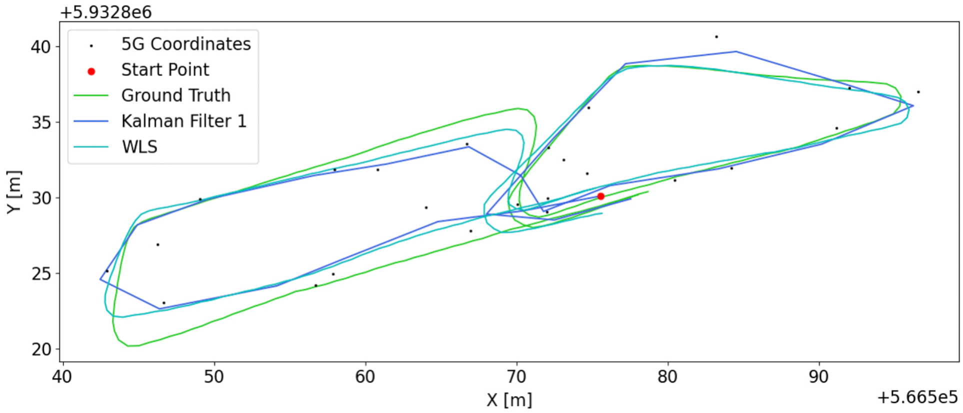

Abbildung 12: Vergleich zwischen Kalman Filter und WLS ('nfg53')

jeder neuen 5G Beobachtung updatet und eine neue Position berechnet. Abbildung 13 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.

4 Fazit und Ausblick

In diesem Bericht wurden verschiedene Möglichkeiten gezeigt, wie 5G-Koordinaten mit Schrittlängen und Schrittrichtungsbeobachtungen eines PDRs kombiniert werden können, um eine Trajektorie zu berechnen, welche möglichst der Ground Truth nahekommt. Einsatzzweck könnten Indoorumgebungen sein, bei der eine genaue Lokalisation notwendig ist. Wir haben dabei drei Methoden untersucht: der Diskrete Kalman Filter, der Extended Kalman Filter und die Weighted-Least-Square

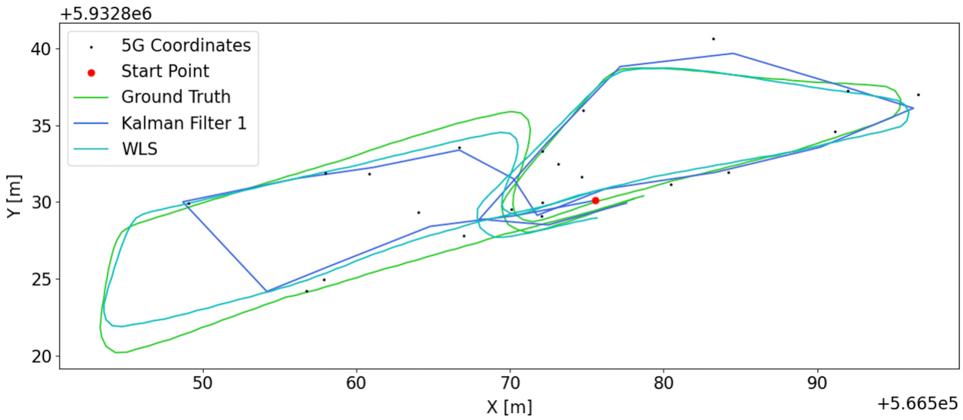

Abbildung 13: Vergleich zwischen Kalman Filter und WLS bei Beobachtungslücke

Methode. Der Extended Kalmen Filter konnte leider nicht komplett implementiert werden, weshalb keine Extended Kalman Filter Ergebnisse produziert werden konnten.

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 13, 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.

Ausblickend sollte die Fehlersuche im Code des Extended Kalman Filters fortgesetzt werden, um auch hier Ergebnisse zu produzieren und den Extended Kalman Filter mit in den Vergleich aufnehmen zu können. Als zukünftige Forschungsperspektive könnte der Extended Kalman Filter noch mehr erweitert werden. Der Vektor der geschätzten Parameter $\hat{\boldsymbol{x}}_i$, 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 die Vorhersage noch besser funktionieren. Ob das der Wirklichkeit entspricht, kann dieser Ansatz in Zukunft überprüft werden.

Des Weiteren sollten die Filter auch mit anderen Datensätzen getestet werden, um zu vermeiden, dass die getroffenen Einstellungen die Filter zu sehr an die beiden vorhandenen Datensätze anpasst und nicht universell einsetzbar sind. Ein weiterer Ansatz könnte sein, die Methoden echtzeitfähig zu machen. In der Theorie ist dies möglich, allerdings würde dies eine Umstrukturierung des Quellcodes erfordern.

Literatur

[1] L. Ojeda and J. Borenstein, “Personal dead-reckoning system for gps-denied environments,” pp. 1–6, 2007.

[2] C. Kjellson, M. Larsson, K. Astrom, and M. Oskarsson, “Accurate indoor positioning based on learned absolute and relative models,” pp. 1–8, 2021.

[3] A. Becker. About the kalman filter. [Online]. Available: https://www.kalmanfilter.net/default.aspx

[4] P. Balzer. Das kalman filter einfach erklärt [teil 2]. [Online]. Available: https://www.cbcity.de/das-kalman-filter-einfach-erklaert-teil-2

[5] ——. Das kalman filter einfach erklärt [teil 1]. [Online]. Available: https://www.cbcity.de/das-kalman-filter-einfach-erklaert-teil-1

[6] A. Ahmadi. Kalmanfilter-vehicle-gnss-ins. [Online]. Available: https://github.com/alirezaahmadi/KalmanFilter-Vehicle-GNSS-INS

[7] T. S. community. scipy.optimize.least_squares. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

[8] S. Lu and M. Eckstein. 5g zur besseren standorterkennung nutzen. [Online]. Available: https://www.elektronikpraxis.vogel.de/5g-zur-besseren-standorterkennung-nutzen-a-843345/

[9] K. Loidl. Voraussichtliche verbesserung der lokalisierungsgenauigkeit von 5g. [Online]. Available: https://www.iis.fraunhofer.de/de/ff/lv/lok/5g/accuracy.html

Erklärung des Python Codes

Eingangsdaten

EightGroundTruth.csv --> Ground Truth

EightStepTimes.csv --> Schrittzeiten

nfg11.csv --> 5G Koordinaten hohe Frequenz hohe Genauigkeit

nfg53.csv --> 5G Koordinaten niedrige Frequenz niedrige Genauigkeit

nsh.csv --> Schrittlängen

nsl.csv --> Schrittrichtungen

Kalman Filter

MyKalmanFilterGNSSODO.py

main()

Ruft data_import() auf um die Daten einzulesen.

Es werden die Nullwerte festgelegt.

Ruft calc_delta() auf um Delta Werte zu berechnen aus Schrittlängen und Schrittrichtungen.

Ruft initial_covariance() auf um die Kovarianzmatrix der Parameter zu beginn zu definieren.

Ruft Kalman_Filter() auf den Kalman Filter durchlaufen zu lassen.

Ruft die Plot Funktion auf.

data_import()

Importiert Daten aus den .csv Dateien. Änderungen an den importierten Daten müssen hier vorgenommen werden.

Es werden die Daten zurückgegeben.

calc_delta()

Berechnet.

Es werden die Delta Werte zurückgegeben.

Kalman_Filter()

Läuft iterativ durch die Beobachtungsdaten (for Schleife).

Genauigkeit von Delta x und Delta y abhängig von Anzahl der Schritte (while Schleife).

Ruft prediction() auf.

Ruft correction() auf.

Es wird das Endergebnis zurückgegben.

prediction()

In dieser Funktion findet der Prediktionsschritt des Kalman Filters statt

Zurückgegeben wird der prädizierte x-Vektor und die dazugehörige Kovarianzmatrix.

correction()

In dieser Funktion wird die Innovation durchgeführt, die Kalman Gain Matrix erstellt und das Update berechnet.

Es wird der korrigierte x-Vektor zurückgegeben und die dazugehörige Kovarianzmatrix.

plot()

Funktion zum plotten der Ergebnisse

Least Square

ls5G.py

main()

Lädt Daten aus .csv Dateien ein.

Genauigkeitsparameter s1 und s2 werden definiert.

Funktion ls() wird aufgerufen

ls()

Läuft iterativ durch die Beobachtungen (for Schleife)

5G Beobachtungen werden für jeden Schritt simuliert durch addieren der Schrittlänge + Schrittrichtung auf die letzten 5G Koordinaten (while Schleife).

Ruft in jedem Schritt die WLS() Funktion auf.

Gibt die gesamte optimierte Trajektorie zurück.

WLS()

In dieser Funktion findet das berechnen der Trajektorie statt.

WLS gibt die optimierte Trajektorie für den einzelnen Schritt wieder aus.

plot()

Diese Funktion dient zum plotten der Ergebnisse

Extended Kalman Filter

MyExtendedKalmanFilterGNSSODO.py

main()

Ruft data_import() auf.

Definiert Startwerte.

Ruft calc_delta() auf.

Ruft initial_covariance() auf um die Kovarianzmatrix der Parameter zu beginn zu definieren.

Ruft Kalman_Filter().

data_import()

Importiert Daten aus den .csv Dateien. Änderungen an den importierten Daten müssen hier vorgenommen werden.

Es werden die Daten zurückgegeben.

calc_delta()

Berechnet.

Es werden die Delta Werte zurückgegeben.

Kalman_Filter():

Läuft iterativ durch die Beobachtungsdaten (for Schleife).

Schritte und Richtungen bis zur nächsten 5G Koordinate werden gesammelt (while Schleiife).

prediction() wird aufgerufen.

correction() wird aufgerufen.

Endergebnis wird zurückgegeben.

prediction()

Transitionsmatrix wird ensprechend der Anzahl der eingehenden Beobachtungen gebildet (for Schleife)

Zurückgegeben wird der prädizierte x-Vektor und die dazugehörige Kovarianzmatrix.

correction()

In dieser Funktion wird die Innovation durchgeführt, die Kalman Gain Matrix erstellt und das Update berechnet.

Funktion wird angepasst im Gegensatz zum diskreten Kalman Filter (siehe oben).

Die Kovarianzmatrix der Beobachtungen wird dynamisch, je nach Anzahl der Schritte zum nächsten 5G Punkt, angepasst (for Schleife).

Es wird das der korrigierte x-Vektor zurückgegeben und die dazugehörige Kovarianzmatrix.