

# Positionsschätzung eines Bahnrennrades auf der Basis von Inertialsensordaten und Umgebungswissen

Bachelorarbeit

Lukas Post

Matrikelnummer: 4000271

4. Januar 2018



Fachbereich Mathematik / Informatik  
Studiengang Informatik

1. Gutachter: Dr. Tim Laue
2. Gutachter: Prof. Dr. Rolf Drechsler

# Erklärung

Hiermit versichere ich, die Bachelorarbeit oder den von mir zu verantwortenden Teil einer Gruppenarbeit, ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Bremen, den 4. Januar 2018

.....

(Lukas Post)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ziele und Aufbau der Arbeit . . . . .	2
1.3	Verwandte Arbeiten . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Bilineare Interpolation . . . . .	4
2.2	Unscented Kalman Filter . . . . .	5
2.3	Rotationen und Mannigfaltigkeiten . . . . .	7
2.4	Drehmatrizen und das Matrix Exponential . . . . .	9
<b>3</b>	<b>Umsetzung</b>	<b>11</b>
3.1	Die verwendete IMU . . . . .	11
3.2	Die Höhenkarte . . . . .	13
3.3	Der Zustand . . . . .	14
3.4	Das Dynamikmodell . . . . .	16
3.5	Die Messmodelle . . . . .	17
3.5.1	Höhenkorrektur . . . . .	18
3.5.2	Geschwindigkeitskorrektur . . . . .	19
<b>4</b>	<b>Evaluation</b>	<b>20</b>
4.1	Abdriftverhalten . . . . .	21
4.1.1	Ohne Korrektur . . . . .	21
4.1.2	Einschub: Systematische Fehler . . . . .	22
4.1.3	Mit Höhenkorrektur . . . . .	24

---

4.1.4	Mit Geschwindigkeitskorrektur . . . . .	25
4.1.5	Mit Höhen- und Geschwindigkeitskorrektur . . . . .	26
4.1.6	Diskussion . . . . .	26
4.2	Präzision . . . . .	27
4.2.1	Experiment . . . . .	28
4.2.2	Diskussion . . . . .	30
<b>5</b>	<b>Fazit und Ausblick</b>	<b>31</b>
	<b>Literaturverzeichnis</b>	<b>33</b>

# Kapitel 1

## Einleitung

Diese Arbeit beschäftigt sich mit der Korrektur einer Positionsmessung, die aus den Messungen eines Inertialsensors (IMU) berechnet wurde. In diesem speziellen Fall handelt es sich um die Position eines Bahnrennrades auf einer Rennbahn. Die Positionskorrektur soll durch die Integration einer Karte der Rennbahn sowie durch Berücksichtigung der dynamischen Eigenschaften eines Fahrrads erreicht werden.

Eine Schwierigkeit ist, dass die korrigierte Positionsschätzung geschehen muss, nachdem die Daten bereits aufgezeichnet wurden. Das bedeutet, dass keine Informationen über systematische Fehler vorliegen und diese dementsprechend auch nicht sofort korrigiert werden können.

Diese Arbeit kann als Vorarbeit zu dem Projekt ZaVI (Zustandsschätzung allein durch Vorwissen und Inertialsensorik) der AG Multisensorische Interaktive Systeme an der Uni Bremen angesehen werden. In diesem Projekt soll untersucht werden, in wie weit es möglich ist, nur mit Vorwissen und ohne weitere Sensormessungen, den Fehler, der durch aufakkumulierte Inertialsensormessungen entsteht, aus einer Schätzung zu entfernen. In ZaVI soll auch der Ansatz der Posenschätzung eines Bahnradfahrers weiter untersucht werden.

### 1.1 Motivation

Ein großes Problem von Inertialen Navigationssystemen (INS) ist, dass diese durch das starke Rauschen der Beschleunigungs- und Drehratensensoren sehr schnell abdriften, und die Messung dadurch unbrauchbar wird. Dieses Problem wird im Normalfall behandelt, indem weitere absolute Positionsmessungen vorgenommen werden, die dann mit den relativen Positionsmessungen des INS fusioniert werden, um so eine gute korrigierte Schätzung der Position zu bekommen.

Die Frage, die sich an dieser Stelle stellt, ist, ob diese Messungen überhaupt notwendig sind, wenn man sich auf einem festen Untergrund bewegt. Hat man genug Informationen über den Untergrund und ist dieser an verschiedenen Stellen verschieden hoch, kann man also die ungenaue Positionsmessung auf den Untergrund korrigieren, um dabei den wahrscheinlichsten

Zustand des Fahrrads zu schätzen.

Durch das ständige Korrigieren des Fahrrads zurück auf die Rennbahn sollte es möglich sein, ein zu starkes Abdriften der Positionsmessung auf einen Bereich des Untergrunds einzugrenzen, der von ähnlicher Höhe ist wie die wahre Position.

Zusätzlich zu der Korrektur der Höhe wird im Normalfall nicht bedacht, dass der Körper, an dem der Inertialsensor befestigt ist, Eigenschaften hat, mit denen eine Messung korrigiert werden kann. So kann ein Satellit nicht schlagartig seine Geschwindigkeit ändern oder ein Auto seine Richtung.

In einem Anwendungsszenario wie dem Bahnradfahren lassen sich solche Ansätze sehr gut evaluieren. Der Untergrund ist an vielen Stellen sehr steil. Durch diese Höhenunterschiede kann eingeschränkt werden, in welchem Bereich sich das Fahrrad befindet.

Wenn die Ansätze, das Abdriften einer Positionsmessung zu verhindern, Erfolg zeigen sollten, können diese in Trackingsystemen eingesetzt werden, um bei Hochfrequenzmessungen eine noch bessere Schätzung zu erreichen.

## 1.2 Ziele und Aufbau der Arbeit

Das Ziel dieser Arbeit ist es, ohne absolute Messungen das Wegdriften der Positionsmessung eines Bahnfahrrads zu verhindern. Stattdessen sollen die Beschaffenheit der Umgebung und die fahrphysikalischen Eigenschaften eines Fahrrads eine Schätzung auf der Strecke halten und so die Grundlage dafür legen, sehr genaue Positionsmessungen aus IMU-Daten zu berechnen, ohne dass diese mit weiteren Messungen korrigiert werden müssen.

Zuerst werden in Kapitel 2 die dafür benötigten mathematischen Grundlagen erläutert. Es folgt in Kapitel 3 die Umsetzung des Verfahrens. Hier wird erklärt, wie die Messungen der IMU durch Annahmen über die Dynamik und Umwelt korrigiert werden. In Kapitel 4 werden die in Kapitel 3 beschriebenen Ansätze getestet, und in Kapitel 5 werden die Ergebnisse erläutert.

## 1.3 Verwandte Arbeiten

Li et al.[6] verwenden eine interpolierte Karte, um GPS Höhenmessungen zu korrigieren. Dabei ist aufgefallen, dass, obwohl die Korrektur von einer potentiell falschen Position abhängt, die Schätzung immer besser war als die alleinige Messung vom GPS. Diese Messungen wurden dann verwendet, um die Positionsmessung des GPS zu verbessern. Eine Art dieser Korrektur wird auch in dieser Arbeit verwendet. Der einzige Unterschied ist, dass die ursprüngliche Positionshypothese nicht aus einer Messung eines Sensors stammt, sondern der Schätzung aus dem vorherigen Zeitschritt entspricht.

Mandel und Laue[7] verwenden eine virtuelle Messung der Entfernung eines Rollstuhls zu der Mitte einer Straße, um Positionen zu korrigieren. Diese Messung ist immer 0, was dazu führt,

---

dass die Positionshypothesen immer auf der Straße bleiben. Da für diese Arbeit keine externen Positionsmessungen zur Verfügung stehen, werden diese virtuellen Messungen verwendet, um Annahmen über den Zustand zu modellieren.

Klein et al.[5] korrigieren mit Annahmen über das Fahrverhalten und der Umgebung eines Fahrzeugs dessen Position. Hier wird angenommen, dass der Geschwindigkeitsvektor eines Autos nach vorne zeigen muss. Das bedeutet, dass ein Auto nicht seitwärts oder auf und ab fahren kann. Diese Annahme kann direkt auf Fahrräder übertragen werden. Andere Annahmen, wie, dass Autos kaum gekippt sein können oder die Höhe sich nur sehr langsam verändert, können nicht auf Fahrräder angewandt werden.

## Kapitel 2

# Grundlagen

### 2.1 Bilineare Interpolation

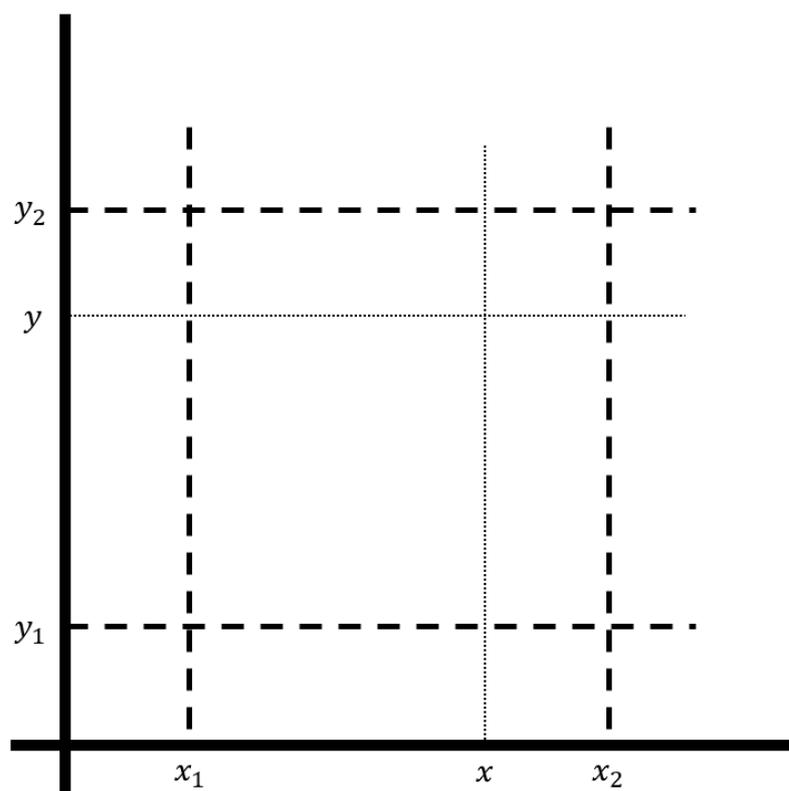


Abbildung 2.1 Eine grafische Darstellung der bilinearen Interpolation

Da in Kapitel 3.2 das 3D-Modell der Rennbahn diskret in 10 cm Schritten abgetastet wird, muss entschieden werden, was passiert, wenn eine Höhe gefunden werden soll, die nicht genau in der Mitte eines abgetasteten Punktes liegt. Würde man eine Nearest-Neighbor-Interpolation durchführen, endet man mit einer Strecke, die an vielen Stellen extreme Ver-

änderungen aufweist und an anderen Stellen gar keine.

Um die gleichmäßige Oberfläche der Rennbahn aus den abgetasteten Punkten wieder herzustellen, muss bilinear über die Daten interpoliert werden. Der Unterschied zwischen linearer und bilinearer Interpolation ist, dass bei der bilinearen Interpolation nicht nur über eine Variable interpoliert wird, sondern über zwei. In diesem Fall sind dies die X- und Y-Koordinaten der Rennstrecke.

Soll nun die Höhe eines Punktes gefunden werden, der zwischen mehreren Kacheln liegt, muss zuerst die Höhe der vier anliegenden Kacheln herausgefunden werden. Diese lassen sich in dem Beispiel dieser Arbeit direkt aus der Höhenkarte auslesen und sind von hier an mit  $h_1(x_1, y_1), h_2(x_2, y_1), h_3(x_1, y_2)$  und  $h_4(x_2, y_2)$  bezeichnet. In Abb. 2.1 ist sichtbar, wie diese Punkte zueinander liegen.

Um nun zwischen diesen Punkten zu interpolieren, damit Höhen zwischen den Kacheln ermittelt werden können, wird die Formel der bilinearen Interpolation benötigt. Diese lautet:

$$h(x, y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} h_1(x_1, y_1) & h_3(x_1, y_2) \\ h_2(x_2, y_1) & h_4(x_2, y_2) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \quad (2.1)$$

Wenn man über die Oberfläche der Rennbahn bilinear interpoliert, kommt eine sehr gleichmäßige Oberfläche heraus. Diese hat nur noch minimale Kanten, die durch Nichtlinearitäten an manchen Stellen in der Strecke entstehen. Diese sind aber so klein und so flach, dass sie bei einem 10 cm Raster nicht weiter auffallen.

## 2.2 Unscented Kalman Filter

Das Unscented Kalman Filter (UKF)[3] ist eine Weiterentwicklung des Kalman Filters[4] für nichtlineare Modelle. Beide werden in der Zustandsschätzung eingesetzt, um aus verrauschten Messungen, bei denen das Rauschen normalverteilt ist, eine möglichst genaue Schätzung eines Zustands  $\mu$  zu liefern.

Der Kalman Filter basiert auf dem Bayes Filter. Dieser ist für Echtzeitmessungen interessant, da für ideale Schätzungen nicht alle Daten verwendet werden, sondern der aktuelle Zustand nur aus dem letzten Zustand und einer Zustandsübergangsmessung berechnet wird.

Neben dem Zustand werden auch die Varianzen und Kovarianzen des Zustands in einer Kovarianzmatrix  $\Sigma$  geschätzt. Das ist notwendig, um ungenaue Messungen nicht so stark in das Ergebnis einfließen zu lassen wie genaue.

Der Kalman Filter verwendet für die Zustandsschätzung zwei Funktionen:

**Predict** In dieser Funktion wird vorausberechnet, wie der Zustand aussehen müsste, nachdem ein Zeitschritt vergangen ist. Dabei ist das Ergebnis nur von der Dynamik des

Zustands und eventuell einer Zustandsübergangsmessung abhängig. Anschaulich bedeutet dies, wenn ein Zustand eine Position enthält, die sich mit einer bestimmten Geschwindigkeit bewegt, bewegt sich die Position entsprechend der Geschwindigkeit und der Länge des Zeitschritts weiter.

**Update** Mit dieser Funktion wird der Zustand mit einer Messung korrigiert. Hier wird die Messung mit dem verglichen, was dem Zustand nach hätte gemessen werden müssen, um die optimale Schätzung aus beiden Werten zu berechnen. Die Berechnung geschieht auf Basis der Varianzen des Zustands und der Messung.

Die Prediction muss in jedem Zeitschritt ausgeführt werden. Es ist aber nicht notwendig, dass überhaupt oder nur eine Updatefunktion aufgerufen wird. Wenn man beispielsweise eine hochfrequente Zustandsübergangsmessung aber nur eine niederfrequente Update-Messung hat, wird Update nur in den Zeitschritten aufgerufen, in denen eine neue Messung verfügbar ist. Andererseits kann Update auch mehrere Male mit verschiedenartigen Messungen aufgerufen werden.

Für die Berechnung des neuen Zustands in Predict und die Berechnung einer Messung aus dem Zustand in Update werden Modelle benötigt, die diese Berechnungen durchführen. Im ursprünglichen Kalman Filter handelt es sich bei diesen Modellen um Linearkombinationen aus dem Zustand und den Messungen. Das ist bei nichtlinearen Messungen aber nicht mehr möglich. Deshalb wurde beim UKF das Prinzip der Sigmapunkte in Kombination mit Dynamik- und Messmodellen eingeführt. Das Grundprinzip ist, dass man in einer Dimension drei verschiedene Zustände an die Modelle gibt und das arithmetische Mittel der Ergebnisse als eigentliches Ergebnis benutzt. Die Sigmapunkte werden folgendermaßen berechnet:

$$\mathcal{X} = \begin{pmatrix} \mu_t & \mu_t + \sqrt{\Sigma_t} & \mu_t - \sqrt{\Sigma_t} \end{pmatrix} \quad (2.2)$$

Wobei  $\mu$  der momentane Zustand und  $\sqrt{\Sigma}$  die Cholesky-Zerlegung der Kovarianzmatrix ist.

Die Dynamik- und Messmodelle werden verwendet, um die Sigmapunkte zu propagieren und daraus einen neuen Mittelwert zu berechnen. Das Dynamikmodell  $g$  bekommt so einen Zustand vom letzten Zeitschritt als Sigmapunkt und eine Zustandsübergangsmessung übergeben und gibt als Ergebnis den Zustand zurück, den das System am Anfang eines Zeitschritts innehaben müsste. Das Messmodell  $h$  bekommt einen Zustand als Sigmapunkt übergeben und berechnet daraus, was hätte gemessen werden müssen, wenn dieser Zustand korrekt wäre.

Der Algorithmus des UKF ist nach [1] folgendermaßen definiert:

Annahmen:

$$\begin{aligned} \mu_{t-1} &\in \mathbb{R}^n, \Sigma_{t-1} \in \mathbb{R}^{n \times n}, u_t \in \mathbb{R}^p, z_t \in \mathbb{R}^m \\ g : \mathbb{R}^n \times \mathbb{R}^p &\rightarrow \mathbb{R}^n, X_t = g(X_{t-1}, u_t) + \mathcal{N}(0, \Sigma_{\epsilon t}) \\ h : \mathbb{R}^n &\rightarrow \mathbb{R}^m, z_t = h(X_t) + \mathcal{N}(0, \Sigma_{\delta t}) \end{aligned} \quad (2.3)$$

Predict:

$$\begin{aligned}
\mathcal{X}_{t-1} &= \begin{pmatrix} \mu_{t-1} & \mu_{t-1} + \sqrt{\Sigma_{t-1}} & \mu_{t-1} - \sqrt{\Sigma_{t-1}} \end{pmatrix} \\
\bar{\mathcal{X}}_t^* &= g(\mathcal{X}_{t-1}, u_t) \\
\bar{\mu}_t &= \frac{1}{2n+1} \sum_{i=0}^{2n} \bar{\mathcal{X}}_t^{*[i]} \\
\bar{\Sigma}_t &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + \Sigma_{ct}
\end{aligned} \tag{2.4}$$

Update:

$$\begin{aligned}
\bar{\mathcal{X}}_t &= \begin{pmatrix} \bar{\mu}_t & \bar{\mu}_t + \sqrt{\bar{\Sigma}_t} & \bar{\mu}_t - \sqrt{\bar{\Sigma}_t} \end{pmatrix} \\
\bar{\mathcal{Z}}_t &= h(\bar{\mathcal{X}}_t) \\
\hat{z}_t &= \frac{1}{2n+1} \sum_{i=0}^{2n} \bar{\mathcal{Z}}_t^{[i]} \\
\Sigma_t^z &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + \Sigma_{\delta t} \\
\bar{\Sigma}_t^{x,z} &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T \\
K_t &= \bar{\Sigma}_t^{x,z} (\bar{\Sigma}_t^z)^{-1} \\
\mu_t &= \bar{\mu}_t + K_t(z_t - \hat{z}_t) \\
\Sigma_t &= \bar{\Sigma}_t - K_t(\bar{\Sigma}_t^{x,z})^T
\end{aligned} \tag{2.5}$$

## 2.3 Rotationen und Mannigfaltigkeiten

Es gibt viele Arten, Rotationen in 3D darzustellen. Der rechnerisch einfachste Weg ist eine  $3 \times 3$ -Matrix, mit der man direkt Koordinatentransformationen durchführen kann. Das Problem ist, dass nicht einfach alle neun Parameter in den Zustand des Kalman Filters übernommen werden können. Damit wäre  $Q^T Q = I$  nicht mehr gewährleistet. Es muss also eine Darstellung von Rotationen gefunden werden, die keine zu berücksichtigende Struktur hat und trotzdem alle Rotationen darstellen kann.

Eine solche Darstellung ist das sogenannte Matrix Exponential. Nach Eulers Rotationstheorem ist es im dreidimensionalen Raum möglich, jede Rotation als eine Drehung um eine Achse darzustellen. Dabei muss eine Achse als dreidimensionaler Vektor und ein Winkel als Skalar gegeben sein.

Das Matrix Exponential stellt den Winkel der Drehung als die Länge des Achsenvektors dar. Man dreht also um einen Vektor  $v$  mit dem Winkel  $|v|$ . Da in dieser Darstellung jeder Vektor als valide Rotation angesehen wird, kann diese Darstellung im Kalman Filter verwendet werden.

Das einzige Problem ist, dass es im Matrix Exponential sogenannte Singularitäten gibt. Singularitäten sind Rotationen, an denen für kleine Änderungen der Rotation sehr große Parameteränderungen vorgenommen werden müssen.

Um dieses Problem zu umgehen, wurden Mannigfaltigkeiten als Zustände eingeführt [1][2]. So kann man im Zustand eine Rotationsdarstellung ohne Singularitäten wählen, muss aber einen Operator  $\boxplus$  definieren, der auf einen Zustand eine Änderung in Matrix Exponential Darstellung heraufrechnet und einen Operator  $\boxminus$ , der die Differenz zweier Zustände als Vektor zurückgibt.

Der Ansatz der Mannigfaltigkeiten funktioniert, da sich der Kalman Filter auf Veränderungen des Zustands stützt. Die Veränderungen werden also in einem Format dargestellt, das der Kalman Filter verarbeiten kann. Der Zustand enthält eine Darstellung, die frei von Singularitäten ist. Im Normalfall ist dies eine  $3 \times 3$ -Matrix.

Es gilt für den Zustand  $\mu \in \mathcal{S} \subset \mathbb{R}^S$ . Die beiden Operatoren  $\boxplus$  und  $\boxminus$  sind nach Hertzberg[2] folgendermaßen definiert:

$$\boxplus_{\mathcal{S}} : \mathcal{S} \times \mathbb{R}^n \rightarrow \mathcal{S} \quad (2.6)$$

$$\boxminus_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^n \quad (2.7)$$

Mit diesen Definitionen ist es möglich, den Kalman Filter neu zu definieren, um Rotationen in drei Dimensionen darzustellen. Der Zustand kann dabei jede Form haben. Es müssen zusätzlich zum UKF nur die Operatoren  $\boxplus$  und  $\boxminus$  definiert werden. Die Neudefinition des Kalman Filters auf  $\boxplus$ -Mannigfaltigkeiten ist nach Frese[1]:

Annahmen:

$$\begin{aligned} \mu_{t-1} &\in \mathcal{S}, \Sigma_{t-1} \in \mathbb{R}^{n \times n}, u_t \in \mathbb{R}^p, z_t \in \mathbb{R}^m \\ g : \mathcal{S} \times \mathbb{R}^p &\rightarrow \mathcal{S}, X_t = g(X_{t-1}, u_t) \boxplus \mathcal{N}(0, \Sigma_{ct}) \\ h : \mathcal{S} &\rightarrow \mathbb{R}^m, z_t = h(X_t) + \mathcal{N}(0, \Sigma_{\delta t}) \end{aligned} \quad (2.8)$$

Predict:

$$\begin{aligned} \mathcal{X}_{t-1} &= \left( \mu_{t-1} \quad \mu_{t-1} \boxplus \sqrt{\Sigma_{t-1}} \quad \mu_{t-1} \boxplus -\sqrt{\Sigma_{t-1}} \right) \\ \bar{\mathcal{X}}_t^* &= g(\mathcal{X}_{t-1}, u_t) \\ \bar{\mu}_t &= \text{MEANOFSIGMAPPOINTS}(\bar{\mathcal{X}}_t^{*[i]}) \\ \bar{\Sigma}_t &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{X}}_t^{*[i]} \boxminus \bar{\mu}_t)(\bar{\mathcal{X}}_t^{*[i]} \boxminus \bar{\mu}_t)^T + \Sigma_{ct} \end{aligned} \quad (2.9)$$

Update:

$$\begin{aligned}
\bar{\mathcal{X}}_t &= \left( \bar{\mu}_t \quad \bar{\mu}_t \boxplus \sqrt{\bar{\Sigma}_t} \quad \bar{\mu}_t \boxplus -\sqrt{\bar{\Sigma}_t} \right) \\
\mathcal{Z}_t &= h(\bar{\mathcal{X}}_t) \\
\hat{z}_t &= \frac{1}{2n+1} \sum_{i=0}^{2n} \bar{\mathcal{Z}}_t^{[i]} \\
\Sigma_t^z &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + \Sigma_{\delta t} \\
\bar{\Sigma}_t^{x,z} &= \frac{1}{2} \sum_{i=0}^{2n} (\bar{\mathcal{X}}_t^{[i]} \boxplus \bar{\mu}_t)(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T \\
K_t &= \bar{\Sigma}_t^{x,z} (\bar{\Sigma}_t^z)^{-1} \\
\mu_t &= \bar{\mu}_t \boxplus K_t(z_t - \hat{z}_t) \\
\Sigma_t &= \bar{\Sigma}_t - K_t (\bar{\Sigma}_t^{x,z})^T
\end{aligned} \tag{2.10}$$

In der Prediction des UKF auf  $\boxplus$ -Mannigfaltigkeiten wird die Funktion `MEANOFSIGMAPOINTS` eingeführt. Diese erwartet eine Menge aus Zuständen und gibt das Mittel dieser Zustandsmenge zurück. Ohne  $\boxplus$ -Mannigfaltigkeiten wird das durch den Durchschnitt berechnet. Dies ist nun aber nicht mehr möglich, da die Struktur der Mannigfaltigkeit nicht bekannt ist. Der Algorithmus, um das Mittel aus Mannigfaltigkeiten zu berechnen, ist wie folgt:

$$\text{Input: } \mathcal{Y}^{[i]}, i = 0, \dots, 2n \tag{2.11}$$

$$\text{Result: } \mu' \tag{2.12}$$

$$\mu'_0 = \mathcal{Y}^{[0]} \tag{2.13}$$

$$\mu'_{k+1} = \mu'_k \boxplus \frac{1}{2n+1} \sum_{i=0}^{2n} \mathcal{Y}^{[i]} \boxplus \mu'_k \tag{2.14}$$

$$\mu' = \lim_{k \rightarrow \infty} \mu'_k \tag{2.15}$$

## 2.4 Drehmatrizen und das Matrix Exponential

Um Rotationsmatrizen mit Rotationen in Form des Matrix Exponentials zu verrechnen, werden Methoden benötigt, um diese Formen ineinander umzuwandeln. Ein Weg, das Matrixexponential in eine Drehmatrix umzuwandeln, ist die sogenannte Rodriguez-Formel. Diese ist nach Frese[1] folgendermaßen definiert:

$$\text{Rot} \left( \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \alpha \right) = \begin{pmatrix} (1-c)x^2 + c & (1-c)xy - sz & (1-c)xz + sy \\ (1-c)xy + sz & (1-c)y^2 + c & (1-c)yz - sx \\ (1-c)xz - sy & (1-c)yz + sx & (1-c)z^2 + c \end{pmatrix} \quad (2.16)$$

$$\text{mit } c = \cos \alpha, s = \sin \alpha \quad (2.17)$$

Um Matrizen in das Matrixexponential umzuwandeln, wird eine Umkehrfunktion benötigt. Diese lautet folgendermaßen (vgl. [8]):

$$a\text{Rot}(Q) = \frac{\alpha}{2 \sin(\alpha)} \begin{pmatrix} Q_{2,1} - Q_{1,2} \\ Q_{0,2} - Q_{2,0} \\ Q_{1,0} - Q_{0,1} \end{pmatrix}, \alpha = \cos^{-1} \left( \frac{Q_{0,0} + Q_{1,1} + Q_{2,2} - 1}{2} \right) \quad (2.18)$$

$$a\text{Rot}(I) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.19)$$

## Kapitel 3

# Umsetzung

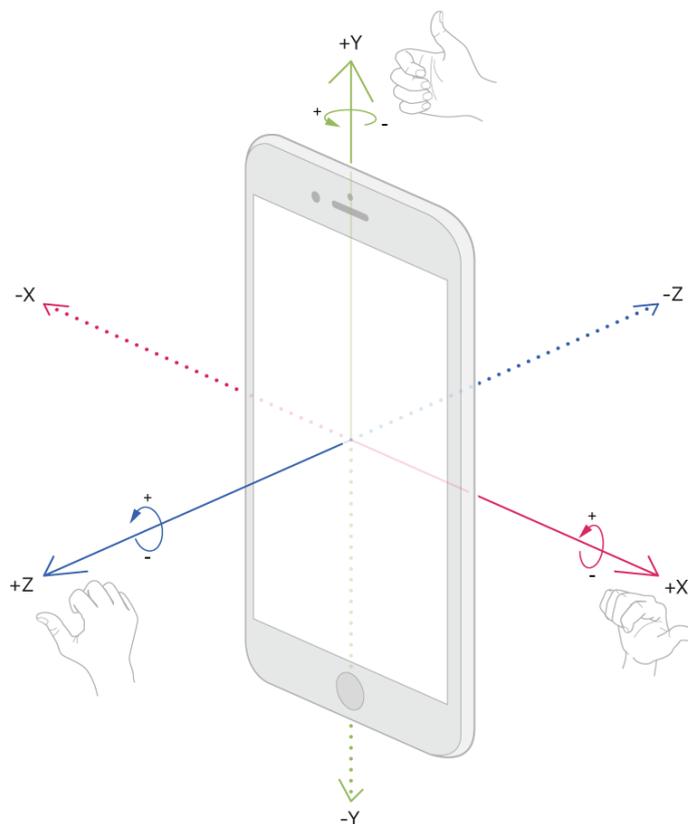
Die Methodik zur Korrektur bezieht sich an vielen Stellen direkt auf den Anwendungsfall, mit dem die Evaluation in Kapitel 4 durchgeführt wurde. Viele der Parameter sind direkt auf den Anwendungsfall zugeschnitten. So kann ein Modell der Rennbahn nicht allgemeingültig für alle Rennbahnen auf der Welt verwendet werden. Allerdings muss die Methodik so einfach gehalten werden, dass diese über die Testläufe hinaus verwendet werden kann und nicht nur speziell für einen Fall gilt.

Die in diesem Kapitel erklärte Methodik basiert nur auf drei Testläufen, die vorher aufgezeichnet wurden. Es können deshalb keine Methoden getestet werden, die das Aufnehmen der Daten verbessert hätten. Das bedeutet, dass diese hier auch nicht erklärt werden können.

### 3.1 Die verwendete IMU

Eine Inertiale Messeinheit (engl. Inertial Measurement Unit, IMU) ist eine Kombination aus einem Beschleunigungssensor (Accelerometer) und einem Drehratensensor (Gyrometer). Diese Sensoren messen in einem Inertialkoordinatensystem Beschleunigungen entlang der Achsen und Drehgeschwindigkeiten um die Achsen herum. Zum Aufnehmen der Daten wurde ein iPhone 6[11] benutzt. Dieses Smartphone hat eine InvenSense MP67B 6-Achsen IMU verbaut[12]. Zu dieser IMU existiert leider kein Datenblatt, da es sich um eine Sonderanfertigung für Apple handelt. Diese Sonderanfertigung basiert allerdings auf einer anderen 6-Achsen IMU von InvenSense, bei der nur Veränderungen an der Hülle und dem Chipdesign vorgenommen wurden[15]. Es kann also angenommen werden, dass die Eigenschaften, die die Messungen betreffen mit dem Standardmodell InvenSense MPU-6500[13] übereinstimmen.

Die IMU gibt die Daten immer in Form eines Vektors in Abhängigkeit eines eigenen Koordinatensystems zurück. Nach Abb. 3.1 sind also Beschleunigungen des Smartphones, welche entlang der Längsachse nach vorne gerichtet sind, in dem Y-Teil des Vektors wieder zu finden. Dasselbe gilt für die Gyrometermessungen, die Geschwindigkeiten um die Achsen herum beinhalten.



**Abbildung 3.1** Das Koordinatensystem der IMU[10]

Durch diese Messungen ist es möglich, die Position der IMU durch Aufintegrieren zu messen. Man dreht das IMU-Koordinatensystem in einem Weltkoordinatensystem um die aufintegrierten Gyrometermessungen und verschiebt das IMU-Koordinatensystem um die gemessenen Accelerometermessungen.

Um die Position eines Fahrrads zu messen, muss die IMU fest am Fahrrad befestigt werden. Dies muss so geschehen, dass, wenn das Fahrrad gerade und nicht gekippt auf dem Boden steht, ein Vektor des IMU-Koordinatensystems genau nach oben und ein Vektor genau nach vorne gerichtet ist. Wenn das nicht gegeben ist, muss, bevor die IMU-Daten aufgezeichnet werden, die Rotation von der IMU zum Fahrrad kalibriert werden. Die in der Evaluation dieser Arbeit verwendete IMU wurde so am Lenker befestigt, dass die Z-Achse der IMU genau nach oben und die Y-Achse nach vorne zeigt.

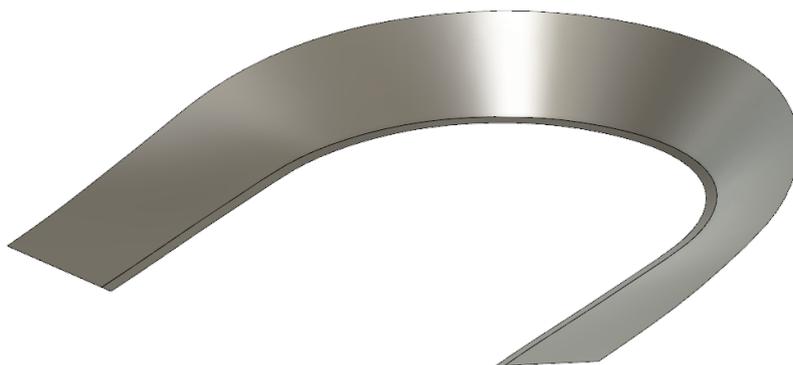
Das Problem mit diesem Ansatz ist, dass die IMUs nicht perfekt sind. Durch technische Limitierungen rauschen die Messwerte um den wahren Wert. In der Zustandsschätzung kann man die Stärke des Rauschens verwenden, die im Datenblatt des Sensors angegeben ist. Für die MPU-6500 IMU lautet die Standardabweichung des Rauschens:

$$\sigma_{acc} = 3 mg = 0.02942 m/s^2 \quad (3.1)$$

$$\sigma_{gyro} = 0.1 rad/s \quad (3.2)$$

## 3.2 Die Höhenkarte

Zur Korrektur der Position soll in dieser Arbeit auch die Höhe des Untergrunds verwendet werden. Um diese zu benutzen wird ein 3D-Modell der Strecke benötigt, auf der sich das Fahrrad bewegt. Bei der Strecke handelt es sich um die mobile Rennbahn der Sixdays Bremen[14]. Leider existieren von dieser Strecke weder öffentlich zugängliche Höhendaten, noch sind Höhendaten in den offiziellen Regeln weiter spezifiziert[16]. Deshalb wurde direkt auf die Originalpläne zurückgegriffen und ein 3D-Modell (Abb. 3.2) der Rennbahn angefertigt.

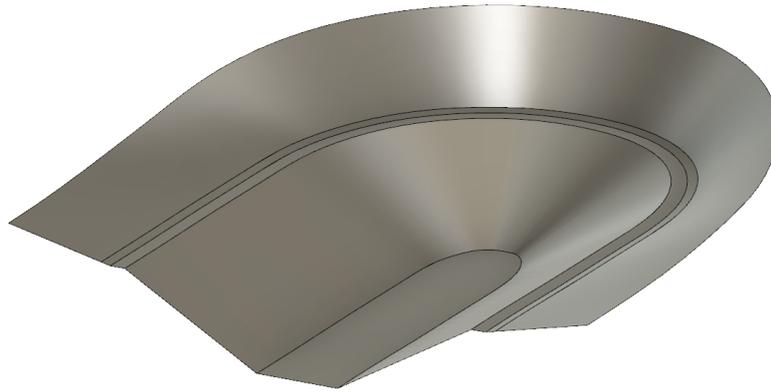


**Abbildung 3.2** Das 3D-Modell der Rennbahn

Es wurde nur die Hälfte der Bahn modelliert, da es sich bei der anderen Hälfte um eine um 180 Grad gedrehte Version der ersten Hälfte handelt.

Dieses 3D-Modell wurde dann in Schritten von 10 Zentimetern abgetastet und als eine 2D-Matrix abgespeichert. So kann mit Hilfe von bilinearer Interpolation eine sehr genaue Schätzung darüber abgegeben werden, auf welcher Höhe sich das Fahrrad gerade befinden müsste.

Neben eines genauen 3D-Modells wurde auch noch eine veränderte Version der Rennbahn modelliert. Bei dieser Version wurde der Mittelteil der Strecke um 5 m abgesenkt und durch einen flach ansteigenden Abhang mit der Strecke verbunden (Abb. 3.3).



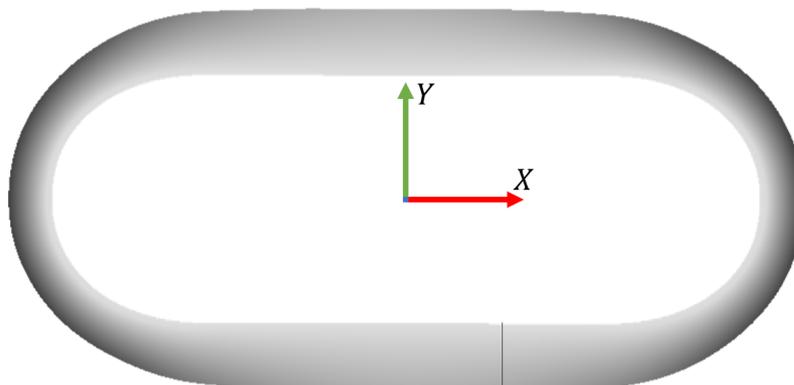
**Abbildung 3.3** Ein modifiziertes 3D-Modell der Rennbahn

Es hat sich in den Tests herausgestellt, dass die Positionsschätzung, wenn diese sehr nah am Innenraum ist, in den Innenraum hineindriftet. Das liegt daran, dass die in Kapitel 3.5 definierten Messmodelle nicht die befahrbare Strecke beschränken. Um diesen Drift zu verhindern, wird ausgenutzt, dass die Höhenkarte zur Korrektur der Höhe eingesetzt wird. Wenn die Höhe aus der Höhenkarte zu stark von der geschätzten Höhe abweicht, wird die Position in eine Richtung korrigiert, in der die Höhe wieder stimmt.

Deshalb wird für die Positionskorrektur das modifizierte 3D-Modell verwendet.

### 3.3 Der Zustand

In dem Zustand des Kalman Filters steht alles, was vom Kalman Filter geschätzt werden soll. Dazu gehören die Position des Fahrrads, die Geschwindigkeit und die Rotation als Drehmatrix. Alle diese Werte sind in einem Weltkoordinatensystem (Abb. 3.4) angegeben.



**Abbildung 3.4** Das Weltkoordinatensystem (Start/Ziel unten)

Der verwendete Zustand ist folgendermaßen aufgebaut:

$$X_t = \begin{pmatrix} Q_{I \rightarrow W_t} \\ p_t \\ v_t \end{pmatrix} \quad (3.3)$$

$Q_{I \rightarrow W}$  ist eine orthonormale  $3 \times 3$  Drehmatrix, die Koordinaten vom IMU-Koordinatensystem in das Weltkoordinatensystem transformiert.  $p$  ist die Position (Meter) und  $v$  die Geschwindigkeit (Meter/Sekunde) des Fahrrads, beide in Form eines 3-dimensionalen Vektors. Die geschätzte Position liegt im Ursprung des IMU-Koordinatensystems.

Zum Start des Filters muss der Zustand initialisiert werden. Es wird angenommen, dass das Fahrrad aufrecht im Innenbereich auf der Höhe der Startlinie steht und in die korrekte Fahrtrichtung ausgerichtet ist. Ist all das gegeben, kann der Zustand folgendermaßen initialisiert werden:

$$Q_{I \rightarrow W_0} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

$$p_0 = \begin{pmatrix} 9 \\ -10.85 \\ -0.135 \end{pmatrix} + Q_{I \rightarrow W_0} * \vec{v}_{offset} \quad (3.5)$$

$$v_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.6)$$

Da im IMU-Koordinatensystem die Y-Achse nach vorne zeigt, das Fahrrad aber im Weltkoordinatensystem in Richtung der X-Achse startet, darf die Rotation nicht mit einer Einheitsmatrix initialisiert werden. Anstelle dessen wird eine Einheitsmatrix um 90 Grad um die Z-Achse gedreht und das Ergebnis als Initialisierung verwendet.

Es ist ungefähr bekannt, von welcher Position aus das Fahrrad startet. Deshalb wird der X- und Y-Anteil vorerst damit und der Z-Anteil mit dem Wert initialisiert, der in der Höhenkarte (Kapitel 3.2) für diese Position angegeben ist. Zum Schluss wird diese Position noch um einen Offset  $\vec{v}_{offset}$  verschoben, so dass diese im Ursprung des IMU-Koordinatensystems liegt. Idealerweise sollte die IMU genau über dem Kontaktpunkt liegen, damit nur noch ein Höhenoffset auf die Position addiert werden muss. Um sicher zu gehen, dass die Position auch korrekt ist, wenn das Fahrrad gedreht startet, wird der Offset mit der Rotation in das Weltkoordinatensystem transformiert.

Die Geschwindigkeit wird mit einem Nullvektor initialisiert, weil das Fahrrad noch nicht fährt.

Die Kovarianzmatrix wird folgendermaßen initialisiert:

$$\Sigma_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{pmatrix} \quad (3.7)$$

Die Varianz der Rotation und dem X- und Y-Anteil der Position ist dabei recht groß, da die exakte Pose nicht ganz bekannt ist. Es ist aber bekannt, dass das Fahrrad sich auf einer bestimmten Höhe befindet und gerade nicht fährt. Daher sind diese Varianzen kleiner.

Nach Kapitel 2.3 wird dieser Zustand als Mannigfaltigkeit angesehen. Er kann also jede Form haben, aber es müssen die Operatoren  $\boxplus$  und  $\boxminus$  definiert werden. Das ist notwendig, damit dieser Zustand vom Kalman Filter verarbeitet werden kann.

Der  $\boxplus$ -Operator rechnet eine Änderung  $\delta$  auf einen Zustand  $\mathcal{X}$  herauf. Das Ergebnis lautet für diesen Zustand:

$$\mathcal{X} \boxplus \delta = \begin{pmatrix} Q * Rot(\delta_Q) \\ p + \delta_p \\ v + \delta_v \end{pmatrix} \quad (3.8)$$

Der  $\boxminus$ -Operator berechnet die Änderung zwischen zwei Zuständen. Diese berechnet sich folgendermaßen:

$$\mathcal{X}_2 \boxminus \mathcal{X}_1 = \begin{pmatrix} aRot(Q_1^{-1} * Q_2) \\ p_2 - p_1 \\ v_2 - v_1 \end{pmatrix} \quad (3.9)$$

### 3.4 Das Dynamikmodell

Das Dynamikmodell berechnet den Zustand, den das System nach einem Zeitschritt innehaben muss. Dazu wird der Zustand aus dem vorherigen Zeitschritt und eine Zustandsübergangsmessung verwendet, die in diesem Fall die neuen Daten vom Accelerometer  $a_t$  und vom Gyrometer  $\omega_t$  sind. Allgemein findet eine Integration statt. Die Gyromessungen werden über die Zeitschrittlänge  $\Delta t$  integriert und fließen dann in die Rotation ein. Etwas Ähnliches geschieht auch mit den Accelerometermessungen. Diese werden integriert und fließen so in die Geschwindigkeit und zusammen mit der Geschwindigkeit in die Positionsschätzung ein.

Die Rotation für den Zeitschritt  $t$  berechnet sich folgendermaßen:

$$Q_t = Q_{t-1} * Rot(\omega_t * \Delta t) \quad (3.10)$$

In der Berechnung der neuen Position und Geschwindigkeit muss beachtet werden, dass in den Accelerometerdaten noch die Erdbeschleunigung von  $g \approx 9.81 \frac{m}{s^2}$  vorhanden ist. Diese muss entfernt werden, bevor die Beschleunigung integriert wird, da es sonst zu Fehlern kommt.

Die neue Geschwindigkeit und Position errechnet sich folgendermaßen:

$$p_t = p_{t-1} + v_{t-1} * \Delta t + (Q_t a_t - (0, 0, g)^T) \frac{\Delta t^2}{2} \quad (3.11)$$

$$v_t = v_{t-1} + (Q_t a_t - (0, 0, g)^T) * \Delta t \quad (3.12)$$

Da im Kalman Filter neben dem Zustand auch die Kovarianzen des Zustands mitgeschätzt werden, müssen auch die Varianzen der Zustandsübergangsmessung in die Kovarianzmatrix einfließen. Die Standardabweichungen der Zustandsübergangsmessungen in Formel 3.1 und 3.2 wurden aus dem Datenblatt des Sensors entnommen und in einer Kovarianzmatrix  $\Sigma_{ct}$  zusammengefasst, so dass die Schätzungen ungenauer werden.

$$\Sigma_{ct} = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.02942 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02942 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02942 \end{pmatrix} \quad (3.13)$$

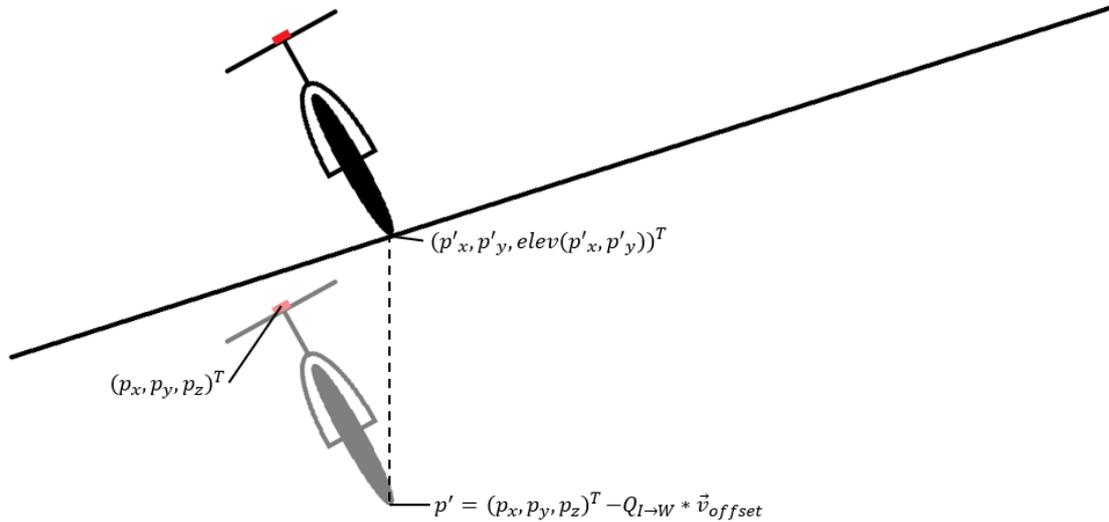
### 3.5 Die Messmodelle

Um den Zustand zu korrigieren werden keine absoluten Messungen benutzt. Anstelle dessen werden Annahmen über den Zustand getroffen, die für den momentanen Zustand zutreffen sollen.

Eine dieser Annahmen ist, dass sich das Fahrrad, während es fährt, die ganze Zeit auf der Strecke befinden muss. Es kann also weder unterhalb noch oberhalb der Strecke sein.

Die andere getroffene Annahme ist, dass ein Fahrrad weder seitwärts noch auf und ab fahren kann. Die einzige Richtung, in die sich ein Fahrrad bewegt, muss die Richtung sein, in die der Lenker geschwenkt ist.

### 3.5.1 Höhenkorrektur



**Abbildung 3.5** Überspitzte Darstellung der Höhenkorrektur

Die Korrektur der Höhe wird in jedem Zeitschritt durchgeführt. Sie basiert auf der Annahme, dass sich die Kontaktfläche des Fahrrads immer genau auf der Strecke befinden muss.

Zu diesem Zweck werden die Koordinaten von zwei Punkten berechnet. Der erste Punkt ist der Kontaktpunkt des Reifens. Der zweite befindet sich auf der Höhe der Strecke an den X- und Y-Koordinaten des ersten Punktes. In der Messung wird nun angenommen, dass diese beiden Punkte einen Abstand von 0 haben müssen.

Das Messmodell für die Höhenkorrektur sieht folgendermaßen aus:

$$h_1 = p' - \begin{pmatrix} p'_x \\ p'_y \\ elev(p'_x, p'_y) \end{pmatrix}, p' = p - Q * \vec{v}_{offset} \quad (3.14)$$

Die Funktion  $elev(x, y)$  ist ein Lookup auf die Höhenkarte, die in Kapitel 3.2 modelliert wurde. Durch bilineare Interpolation wird aus den Kacheln der Höhenkarte die korrekte Höhe an der übergebenen Position berechnet und zurückgegeben.

Da die Kovarianzmatrix  $\Sigma_{\delta t}$  dieser Messung nicht genau bekannt ist, muss sie geschätzt werden. Diese Varianz kann nicht als allgemeingültige Lösung angesehen werden, weil sie sehr stark von Ungenauigkeiten in der Rennbahn abhängt. Experimentell konnte folgende Kovarianzmatrix ermittelt werden:

$$\Sigma_{\delta t_1} = (0.504495^2) \quad (3.15)$$

### 3.5.2 Geschwindigkeitskorrektur

Die Geschwindigkeitskorrektur besteht aus zwei Messungen. Beide beziehen sich auf den in das IMU-Koordinatensystem transformierten Geschwindigkeitsvektor. Das Fahrrad fährt in Richtung der Y-Achse der IMU. Es sollten deshalb nur geringe Geschwindigkeiten in X- und Z-Richtung gemessen werden, die im Mittel 0 sind, da Fahrräder weder seitwärts noch auf und ab fahren können.

In den beiden Messmodellen wird der X- und Y-Anteil des ins IMU-Koordinatensystem gedrehten Geschwindigkeitsvektors zurückgegeben.

$$h_2 = (Q^{-1} * v)_x \quad (3.16)$$

$$h_3 = (Q^{-1} * v)_z \quad (3.17)$$

Die Kovarianzmatrizen der Modelle sind:

$$\Sigma_{\delta t_2} = (2^2) \quad (3.18)$$

$$\Sigma_{\delta t_3} = (1^2) \quad (3.19)$$

Dabei ist die Standardabweichung von  $\Sigma_{\delta t_2}$  höher, da durch Rollbewegungen um die Y-Achse höhere Geschwindigkeiten auftreten können.

## Kapitel 4

# Evaluation

Es liegen keine Ground truth Positionsdaten zu den IMU-Daten vor, so dass direkt der Positionierungsaspekt des Kalman Filters getestet werden könnte. Deshalb werden die Messmodelle zuerst darauf getestet, wie schnell die Schätzung von der Bahn wegdriftet. Sollte die Schätzung ständig auf der Bahn bleiben, wird darüber hinaus noch die Genauigkeit der Schätzung getestet.

Die ungefähre Position ist dabei aus einem Video extrahiert worden, das von den Testläufen aufgenommen worden ist.

Zur Evaluation ist dafür ein Programm geschrieben worden, in dem der Kalman Filter implementiert worden ist, und man in der grafischen Oberfläche die momentane Schätzung sehen kann. Neben der eingezeichneten Schätzung ist es auch möglich, ein Video des Testlaufs oder die IMU-Daten als Plot einblenden zu lassen.

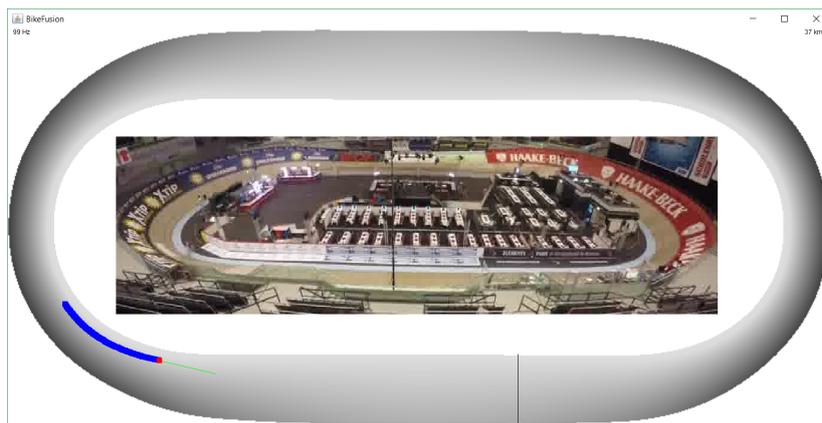


Abbildung 4.1 Die grafische Oberfläche des Programms

Ein großes Problem mit den Daten ist, dass weder eine Kalibrierung der Anfangsrotation durchgeführt wurde, noch Informationen darüber vorliegen, wie valide die Daten sind. Es existieren große Lücken in den Daten. Daher können nicht alle benutzt werden.

Bei allen drei Testläufen gab es Momente, bei denen der Timestamp sehr große Sprünge

gemacht hat. Außerdem gab es Ausschläge in den IMU-Daten, die sich nicht im Video widerspiegeln.

Die Schätzung wurde also nur mit ununterbrochenen Daten durchgeführt. Dies hat dazu geführt, dass jeweils nur ungefähr sieben Runden evaluiert werden konnten.

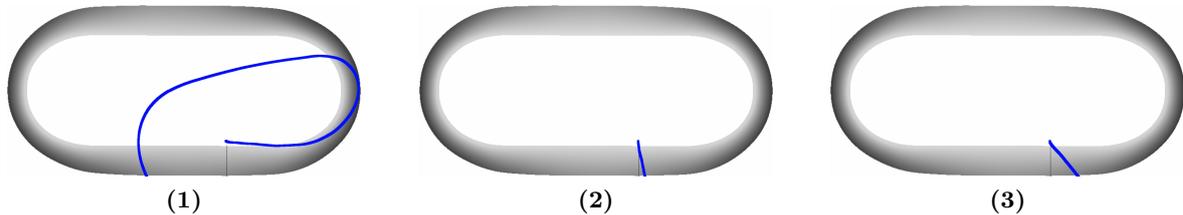
## 4.1 Abdriftverhalten

Bei diesem Experiment soll getestet werden, wie gut die einzelnen Modelle den IMU-Drift verhindern können oder sogar die Schätzung auf der Strecke halten.

Dazu wurden Tests durchgeführt, wobei alle Testläufe mit verschiedenen Modellkonfigurationen gestartet wurden. Als Benchmark werden alle Modelle, die zur Korrektur verwendet werden können, ausgestellt, um festzustellen, welches Ergebnis eine einfache Integrierung ergeben hätte.

Zum Schluss wurden dann noch alle Korrekturmodelle eingeschaltet, um ein möglichst gutes Ergebnis zu erhalten.

### 4.1.1 Ohne Korrektur



**Abbildung 4.2** Testdatensatz (1), (2) und (3) ohne Korrektur. (1) Es wird in die korrekte Richtung gestartet und nach einer Kurve driftet die Schätzung nach unten weg. (2+3) Es wird in die falsche Richtung gestartet und die Schätzungen driften sofort ab.

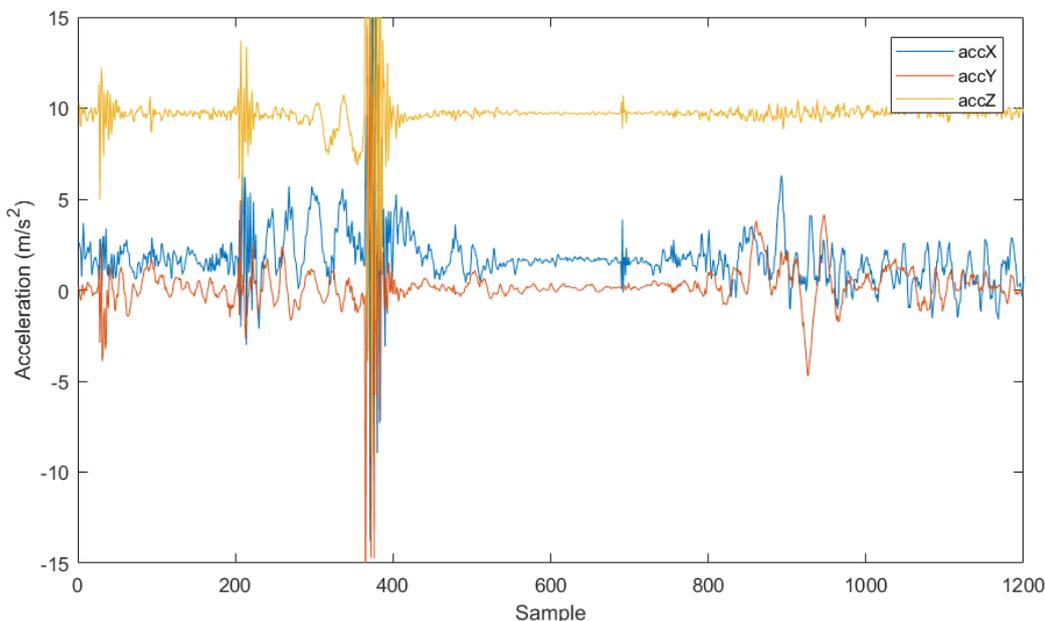
Bei diesem Experiment wurde die Position nur durch aufintegrieren der Gyro- und Accelerometerdaten errechnet. Ohne jedwede Korrektur muss eine Messung früher oder später abdriften. In Abb. 4.2.1 sieht man sehr gut, wie die Schätzung von der Strecke abkommt und danach durch die Mitte die Strecke verlässt. Interessant ist aber, dass die gefahrene Kurve schon Ähnlichkeit zur tatsächlichen Strecke hat.

Was in Abb. 4.2.2 und Abb. 4.2.3 passiert, ist auf den ersten Blick unklar. In den Accelerometerdaten von Testlauf 2 findet sich das Problem. Obwohl sich das Fahrrad kaum seitlich bewegt hat, ist die Beschleunigung in Richtung der X-Achse sehr hoch. Das lässt sich durch eine zu schiefe Montage der IMU am Fahrrad erklären. Sollte die Halterung um die Y-Achse herum verdreht worden sein, wirkt die Gravitation nicht nur auf die Z-Achse, sondern auch auf die X-Achse. Diese Beschleunigung wird aber wegen der initialen Rotation nicht herausge-

rechnet, sondern beschleunigt die Schätzung aus der Strecke hinaus. Tatsächlich gibt es auch am Anfang von Testlauf 3 ein Problem. Es gibt einen großen Ausschlag in den Z-Gyro Daten, der wegen der hohen initialen Varianz die Schätzung sehr stark im Uhrzeigersinn dreht. Die einzige Erklärung hierfür ist, dass die Rotation um die Z-Achse am Anfang falsch war, und diese Drehung notwendig war, um den Lenker wieder nach vorne zeigen zu lassen.

#### 4.1.2 Einschub: Systematische Fehler

Wie in Kapitel 4.1.1 beschrieben, liegen in den letzten beiden Testläufen offensichtlich Fehler vor. Es handelt sich bei beiden um einen sogenannten systematischen Fehler. Also einen Fehler, der sich gleichermaßen auf alle Messungen auswirkt. Wenn man herausfindet, was bei den Messungen schiefgegangen ist, ist es möglich den systematischen Fehler herauszurechnen.



**Abbildung 4.3** Der Anfang der Accelerometerdaten des zweiten Testlaufs. Es ist deutlich zu sehen, dass die Beschleunigung in X-Richtung größer ist als in Y-Richtung

Die IMU wurde durch eine Halterung am Fahrrad befestigt, die es unmöglich macht, dass die IMU um die X- oder Z-Achse verdreht ist. Es ist allerdings sehr gut möglich, dass die IMU während des Testlaufs seitlich gekippt war. Das macht den Daten (Abb. 4.3) nach auch Sinn. Wenn das Fahrrad aufrecht steht und nicht bewegt wird, was sehr kurz vor dem Start so ist, sollte auf die X-Achse keine Beschleunigung ausgeübt werden. Falls doch Beschleunigung gemessen werden sollte, muss die X-Achse etwas nach oben oder unten zeigen.

In diesem Fall wirkt auf die X-Achse eine positive Beschleunigung, also muss die X-Achse leicht nach oben zeigen.

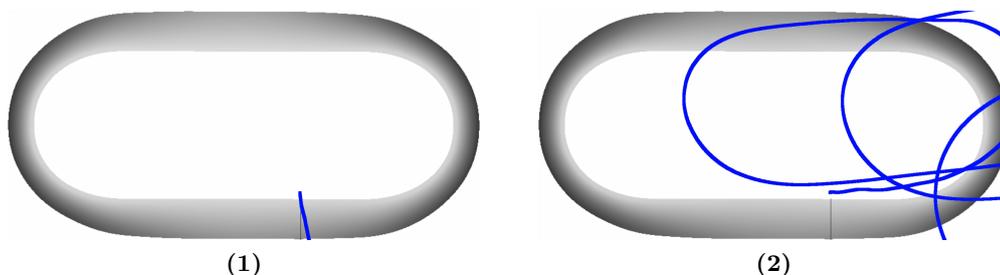
Der systematische Fehler lässt sich entfernen, indem man die Gyro- und Accelerometermessungen mit einer einfachen Drehmatrix multipliziert, die die Messungen mit einem bestimm-

ten Winkel um die Y-Achse dreht.

$$acc' = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix} acc \quad (4.1)$$

$$gyro' = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix} gyro \quad (4.2)$$

In dem Fall des zweiten Testlaufs handelt es sich bei dem Winkel um ungefähr 10 Grad. Führt man nun den Testlauf unter den Bedingungen von Kapitel 4.1.1 ein weiteres Mal durch und korrigiert den systematischen Fehler, sieht das Ergebnis folgendermaßen aus:

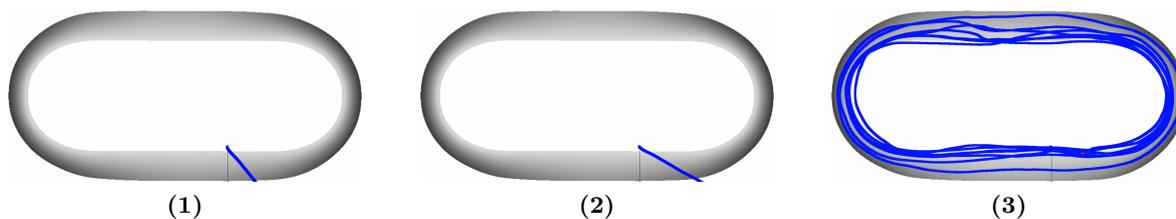


**Abbildung 4.4** (1) Der Ursprüngliche Pfad des Testlaufs (2) Der Pfad des Testlaufs mit korrigiertem systematischen Fehler

Die Schätzung startet nun in die richtige Richtung. Sie driftet zwar noch ab, aber das ist ohne Korrektur der Schätzung zu erwarten. Da die neue Schätzung mehr den Erwartungen entspricht, wird diese von nun an für diese Evaluation verwendet.

In Testlauf 3 gibt es keine Anzeichen, dass die IMU verdreht war. Wahrscheinlicher ist, dass der Zustand falsch initialisiert worden ist und durch die hohen Varianzen sehr schnell abgedriftet ist. Leider ist nicht bekannt, wie der initiale Zustand für diesen Testlauf aussehen muss. Daher kann der Fehler nicht genau identifiziert werden und daher auch nicht entfernt werden.

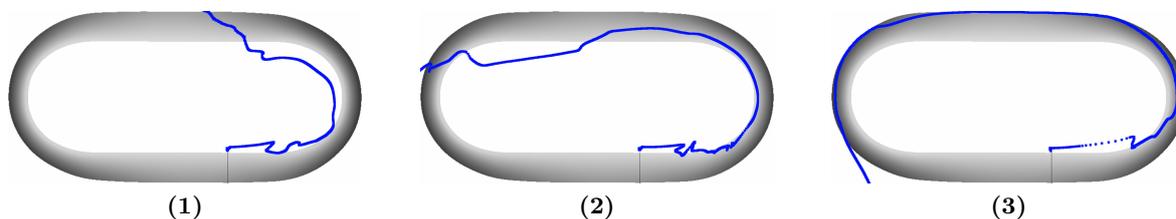
Auf den ersten Blick sieht es so aus, als wenn die Schätzung zu weit nach rechts gedreht ist, weil sie nach rechts gestartet ist. Die initiale Rotation wird also um ungefähr 20 Grad nach links gedreht und getestet, wie sich die Schätzung verhält.



**Abbildung 4.5** Testdatensatz 3 (1) mit normalem Initialzustand ohne Korrekturen. (2) mit gedrehtem Initialzustand ohne Korrekturen. (3) mit gedrehtem Initialzustand inklusive Höhen- und Geschwindigkeitskorrektur.

Ohne Korrekturen startet der Testlauf mit gedrehtem Initialzustand immer noch nach unten aus der Bahn hinaus, was keine sehr große Verbesserung ist. Schaltet man aber die Höhen- und Geschwindigkeitskorrektur ein, driftet die Schätzung nicht mehr von der Strecke ab. Es kann aber nicht erklärt werden, warum die Schätzung immer noch nach unten aus der Bahn steuert. Aus diesem Grund wird für die weiteren Experimente nicht die korrigierte Version verwendet, sondern die Originalversion. Es zeigt sich aber, dass es möglich ist, durch eine Anpassung des Zustands ein Wegdriften der Schätzung von der Strecke zu verhindern.

#### 4.1.3 Mit Höhenkorrektur



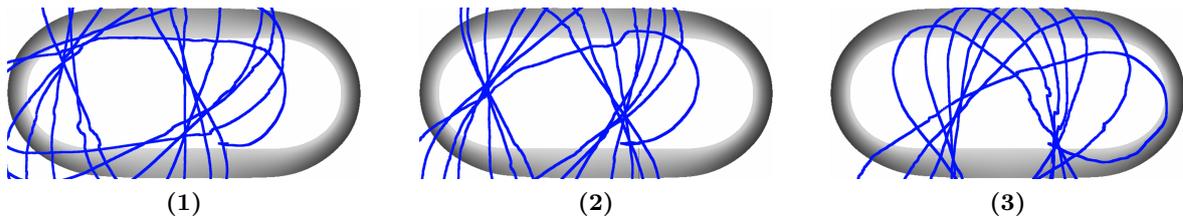
**Abbildung 4.6** Testdatensatz (1), (2) und (3) mit Höhenkorrektur. (1) Die Schätzung bleibt sehr lange im Innenbereich und überkorrigiert aus der Bahn heraus. (2) Die Schätzung startet in die richtige Richtung und korrigiert sofort auf die Strecke. Auf der Geraden schwankt die Schätzung in den Innenbereich und verlässt über die Hochkurve die Strecke. (3) Die Schätzung bewegt sich am Anfang zu weit nach außen und wird von IMU-Messungen aus der Bahn gedrängt.

Die Höhenkorrektur alleine ist nicht zufriedenstellend. Es kann zu keinem Zeitpunkt eine stabile Schätzung erreicht werden. Im Vergleich zu den Ergebnissen ohne Korrektur gibt es eine leichte Verbesserung, was die Zeit auf der Strecke angeht. Aber die Schätzungen sind instabil, und im dritten Testlauf sieht es so aus, als wenn die Schätzung nur am äußeren Rand entlangschleift. Dieses Schleifen lässt sich durch den großen Höhenunterschied zwischen der äußeren Bahn und dem Untergrund erklären.

Es ist aber zu erkennen, dass, solange die Unsicherheit nicht zu hoch ist, ein Abdriften nach außen unterdrückt werden kann. Durch die Kombination mit der angepassten Höhenkarte geht die Schätzung auch nicht mehr durch die Mitte der Rennbahn.

Es lässt sich sagen, dass diese Methode allein nicht besonders gut geeignet ist, um das Abdriften zu verhindern. Die Messung ist sehr unstet und bleibt nur minimal länger auf der Strecke als die unkorrigierte Messung.

#### 4.1.4 Mit Geschwindigkeitskorrektur



**Abbildung 4.7** Testdatensatz (1), (2) und (3) mit Geschwindigkeitskorrektur. Keiner der drei Schätzungen driftet ab. Aber wegen der nicht korrigierbaren Ausrichtung drehen sich die Schätzungen um die Bahn herum.

Die Korrektur des Geschwindigkeitsvektors liefert sehr überraschende Ergebnisse. Es sieht aus, als wenn nur durch Beachtung der Dynamik des Fahrrads der Drift fast komplett entfernt wurde. Nachdem die Testdatensätze durchgelaufen sind, sind alle Schätzungen noch in der Nähe der Rennbahn, obwohl weder Umgebungsdaten noch absolute Positionsmessungen in die Schätzung eingeflossen sind.

In Abb. 4.7 fallen zwei Sachbestände auf:

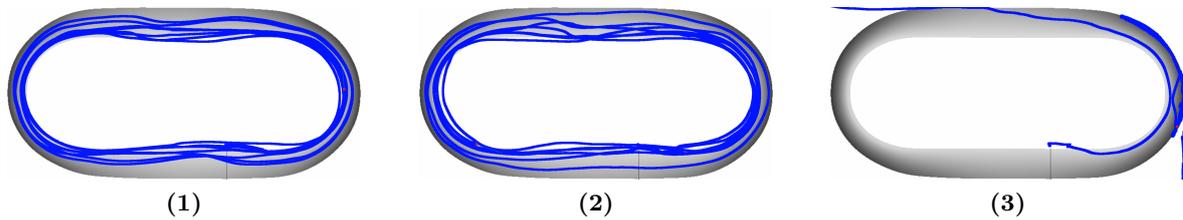
##### 1. Rotation der Schätzung

Nach jeder Runde driftet die Ausrichtung der Schätzung etwas gegen den Uhrzeigersinn. Dieser Effekt tritt durch den Z-Gyro, also die Drehratenmessung um die Z-Achse, auf, indem in jeder Hochkurve ein etwas zu starker Rotationsoffset aufintegriert wird. Durch diesen kleinen Drift sieht der Verlauf der Schätzung aus, als wenn er mit einem Spirographen gezeichnet worden wäre.

##### 2. Form des Schätzungsverlaufs

Sieht man sich die einzelnen Runden des Verlaufs an, haben alle Runden annähernd die Form und Größe der Rennstrecke. Das bedeutet, dass obwohl die Schätzung um die Z-Achse gedreht ist, die Geschwindigkeitskorrektur nicht nur dafür sorgt, dass die Schätzung nicht abdriftet, sondern auch, dass durch diese Methode eine Positionsbestimmung möglich ist, wenn der Ausrichtungsdrift eliminiert werden kann.

#### 4.1.5 Mit Höhen- und Geschwindigkeitskorrektur



**Abbildung 4.8** Testdatensatz (1), (2) und (3) mit Höhen- und Geschwindigkeitskorrektur. (1+2) Zwei sehr gute Schätzungen, welche nicht abdriften, aber auf den Geraden weit nach innen gedrückt werden. (3) Die Schätzung verlässt die Strecke wie in Abb. 4.6.3. Korrigiert dann im falschen Moment auf die Strecke zurück und bleibt am rechten Rand hängen.

Die Kombination aus den beiden Korrekturen führt in den Testläufen 1 und 2 dazu, dass die Schätzung permanent auf der Strecke bleibt. Dabei ist eine Abweichung der Schätzung im Mittelbereich der Geraden zu beobachten, in der die Schätzung in den Innenbereich der Bahn gezogen wird.

Der dritte Testlauf ist fehlgeschlagen. Diese Entwicklung ist auf eine falsche Zustandsinitialisierung zurückzuführen, die bereits in Kapitel 4.1.1 beobachtet wurde. Die Schätzung scheint sich am rechten Rand der Strecke festzufahren und verlässt diese dabei kurz. Als es so aussieht, als wenn der Zustand korrigiert worden wäre, verlässt die Schätzung jedoch über den oberen Rand die Rennbahn.

#### 4.1.6 Diskussion

Obwohl durch die Korrekturen in zwei von drei Testläufen gute Ergebnisse erzielt wurden, werfen diese Ergebnisse doch einige Fragen auf.

##### **Warum hat die Höhenkorrektur so schlechte Ergebnisse geliefert?**

Die Ergebnisse des Tests der Höhenkorrektur in Kapitel 4.1.3 haben zwar in Kombination mit der Geschwindigkeitskorrektur gute Ergebnisse geliefert, aber allein waren die Ergebnisse unstet und haben schnell die Bahn verlassen. Es kann natürlich sein, dass der Ansatz, eine ungenaue Positionsschätzung auf diese Art zu korrigieren, an sich nicht gut funktioniert. Daneben ist es aber auch sehr gut möglich, dass das 3D-Modell nicht genau genug ist. Einerseits ist das Modell nicht direkt von den Architekten, sondern eine Eigenanfertigung, die auf Plänen aus dem Jahr 1956 basiert. Andererseits ist die Bremer Radrennbahn eine mobile Bahn, welche seit 1965 jedes Jahr auf und wieder ab gebaut wird. Es kann also nicht darauf vertraut werden, dass die Originalpläne und damit auch das 3D-Modell immer noch mit der Rennbahn übereinstimmen.

Eine Idee um die Höhenkorrektur zu verbessern, wäre die Rennbahn mit einem Laserscanner zu vermessen und anhand von diesen Daten eine Höhenkarte anzufertigen.

### **Wie wichtig ist die Rotationskalibrierung der IMU?**

Die IMU-Daten wurden aufgezeichnet, bevor diese Arbeit begonnen hat. Es wurden keine Vorkehrungen getroffen, die der Präzision der Schätzung zuträglich gewesen wären. Im Normalfall müsste, nachdem die IMU fest am Fahrrad befestigt wurde, das Fahrrad auf eine gerade Fläche gestellt werden und dafür gesorgt werden, dass die Z-Achse genau nach oben zeigt. Dann muss eine Drehmatrix berechnet werden, die das IMU-Koordinatensystem in das Fahrradkoordinatensystem transformiert. Das Fahrradkoordinatensystem sollte jetzt um keine Achse gedreht sein. Mit dieser Drehmatrix können Fehler, die durch die Befestigung der IMU entstehen sofort und präzise korrigiert werden.

Obwohl dieses Vorgehen beim Aufnehmen der IMU-Daten nicht beachtet wurde, sind trotzdem Ergebnisse herausgekommen, bei denen der Drift korrigiert werden konnte. Es ist immer besser eine Rotationskalibrierung durchzuführen, aber der Kalman Filter kommt mit kleinen Ungenauigkeiten in der Rotation klar.

### **Ist mit diesen Ergebnissen schon eine genaue Positionierung möglich?**

In Abb. 4.8 ist zu sehen, dass auf den Geraden die Schätzung sehr stark in den Innenbereich gezogen wird. Das liegt daran, dass sich auf der Geraden die Höhe kaum verändert, und sich die Schätzung daher zu stark auf die IMU verlässt. In Kapitel 4.1.4 hat sich herausgestellt, dass durch die Integration die Ausrichtung des Fahrrads gegen den Uhrzeigersinn driftet. Durch die Kombination dieser beiden Sachverhalte driftet die Schätzung auf den Geraden in den Innenbereich.

Obwohl dieser Drift die Schätzung nicht aus der Bahn befördert, führt er doch dazu, dass die laterale Positionsschätzung unbrauchbar wird. Durch das Fehlen von Ground truth Positionsdaten wird die Bestimmung der Positionsgenauigkeit noch erschwert. Eine Möglichkeit, die Genauigkeit zu bestimmen, ist, die Zeitpunkte, in denen das Fahrrad auf dem Video die Ziellinie passiert mit den Zeitpunkten, in denen das Fahrrad im Programm die Ziellinie passiert, zu vergleichen. Diese Art der Auswertung wird in Kapitel 4.2 durchgeführt.

## **4.2 Präzision**

Die Präzision der Positionsschätzung ist nicht einfach über die Ground truth zu machen, weil keine Ground truth Daten vorhanden sind. Der einzige Weg, eine Position zu einem bestimmten Zeitpunkt zu bestimmen, ist über die Videos, die von den Testläufen aufgenommen wurden.

Da es sehr aufwändig ist, für jeden Zeitpunkt die Position eines Rades aus dem Video zu extrahieren, wird sich auf die Momente beschränkt, in denen das Fahrrad über die Ziellinie fährt. Es wurden die Zeiten im Video notiert, in denen das Fahrrad die Ziellinie passiert und aus dem Programm die Timestamps in Millisekunden ausgegeben, in denen die Schätzung die Ziellinie passiert. Die Millisekunden wurden in Minuten und Sekunden umgerechnet und so ein Offset gebildet, wie weit die Schätzung und das Video auseinanderliegen.

Problematisch ist, dass der Beginn des Videos und der Daten nicht synchronisiert werden können, da in den Videos nicht ersichtlich ist, in welchem Moment die Messung begonnen hat. Deshalb werden nicht die absoluten Offsets verwendet, um eine Annahme darüber aufzustellen, wie genau die Schätzung ist, sondern die Verteilung der Offsets untereinander.

Auf diese Art kann nicht die absolute Positionsgenauigkeit ermittelt werden, da die Schätzung einen festen Fehler haben könnte, also zum Beispiel zu jeder Zeit 5m hinter oder vor der Schätzung sein könnte.

Diese Art von Fehler kann aber entfernt werden, wie es in Kapitel 4.1.2 bereits getan wurde. Es wird also nicht evaluiert, wie genau die Positionsschätzung ist, sondern, wie genau sie werden könnte, wenn dieser Fehler entfernt wird.

Es wird also kein genauer Wert gesucht, sondern eine Grenze, wie genau die Positionsschätzung mit dieser Methode und diesen Vorbedingungen werden kann.

### 4.2.1 Experiment

Da die ersten beiden Testläufe nicht abgedriftet sind, können nur diese beiden auf ihre Genauigkeit untersucht werden. Die Zeiten, bei denen das Fahrrad in jeder Runde die Ziellinie überquert hat, lauten für beide Testläufe folgendermaßen:

	<b>Programm</b>	<b>Video</b>	<b>Offset</b>
1	0:29.908	0:30.481	0:00.573
2	0:46.252	0:46.800	0:00.548
3	1:02.378	1:02.562	0:00.184
4	1:17.890	1:18.200	0:00.310
5	1:32.899	1:33.160	0:00.261
6	1:46.495	1:46.321	-0:00.174
7	1:59.602	1:59.641	0:00.039

**Tabelle 4.1** Zeiten des Testdatensatz 1

	<b>Programm</b>	<b>Video</b>	<b>Offset</b>
1	0:29.180	0:34.280	0:05.100
2	0:45.370	0:50.560	0:05.190
3	1:02.004	1:07.040	0:05.036
4	1:18.060	1:23.121	0:05.061
5	1:33.240	1:38.601	0:05.361
6	1:48.010	1:52.921	0:04.911
7	2:01.439	2:06.200	0:04.761

**Tabelle 4.2** Zeiten des Testdatensatz 2

In Tab. 4.1 ist zu erkennen, dass die Zeiten von Programm und Video beim ersten Testlauf sehr nah aneinander liegen. Beim zweiten Testlauf existiert ein Versatz von ungefähr 5 Sekunden, was daran liegt, dass das Video früher gestartet wurde als die Aufzeichnung der IMU-Daten.

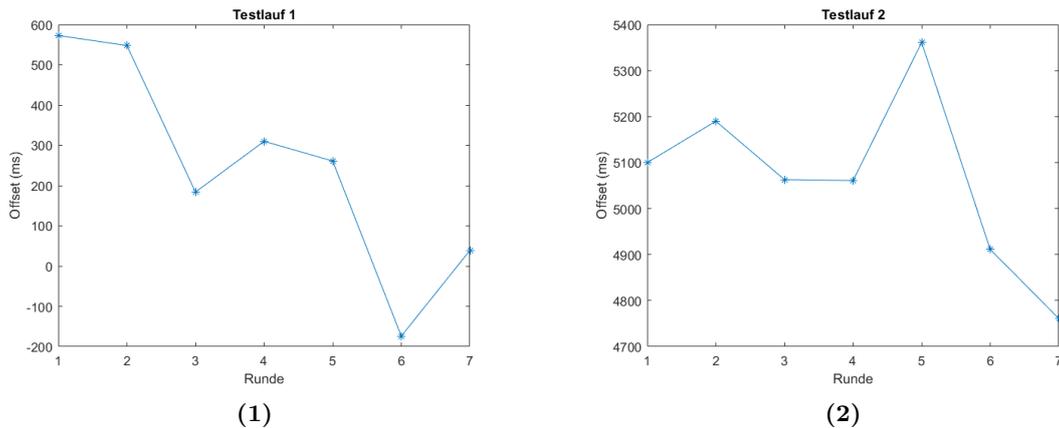


Abbildung 4.9 Die Offsets der beiden Testdatensätze

Um die minimale Genauigkeitsgrenze zu berechnen müssen zwei Annahmen getroffen werden.

1. Es wird angenommen, dass der Versatz von Video und Programm das Mittel zwischen dem größten und kleinsten berechneten Versatz ist. Ansonsten könnten die Abweichungen in eine Richtung größer sein.
2. Der mittlere Offset ist nur eine Zeit. Um eine minimale Positionsgenauigkeit zu ermitteln, wird eine Geschwindigkeit benötigt. Experimentell wurde eine durchschnittliche Geschwindigkeit von  $40 \frac{km}{h}$  beobachtet. Es wird also angenommen, dass sich das Fahrrad beim Überqueren der Ziellinie mit einer Geschwindigkeit von  $40 \frac{km}{h}$  bewegt hat.

Da eine Grenze ermittelt werden soll, wird nun die minimale Abweichung des mittleren Offsets zu den Extremwerten für beide Testdatensätze bestimmt.

$$\begin{aligned}
 & (\text{Offset}_{max} - \text{Offset}_{min})/2 \\
 \text{Testdatensatz 1: } & (573 \text{ ms} - (-174 \text{ ms}))/2 & = 373.5 \text{ ms} & (4.3)
 \end{aligned}$$

$$\text{Testdatensatz 2: } (5361 \text{ ms} - 4761 \text{ ms})/2 = 300 \text{ ms} \quad (4.4)$$

Die größte Abweichung bei idealen Verhältnissen ist 373.5 ms. Bei einer Geschwindigkeit von  $40 \frac{km}{h}$  ist die maximale Positionsabweichung:

$$373.5 \text{ ms} * 40 \frac{km}{h} = 4.15 \text{ m} \quad (4.5)$$

Die maximale Abweichung der Position bei idealen Verhältnissen ist 4.15 m.

### 4.2.2 Diskussion

Diese Art der Präzisionsbestimmung ist sehr ungenau. Die Annahme dass der Offset zwischen Video und Programm ideal ist, ist für ein wahres Ergebnis nicht anwendbar. Da aber das Ergebnis von 4.15 m unter idealen Bedingungen entstanden ist, kann angenommen werden, dass eine Präzision von unter 4.15 m unter diesen Umständen nicht möglich ist. Dabei müssen aber die Erläuterungen aus Kapitel 4.1.6 bedacht werden, da sich die Präzision unter besseren Bedingungen noch erhöhen kann.

Ferner wurde die Genauigkeit nur an 7 Punkten berechnet. Um eine wirklich genaue Positionsgenauigkeit herauszufinden, werden auf jeden Fall für den gesamten Testlauf Ground truth Daten benötigt. Diese können dann mit der Schätzung verglichen werden. Dazu ist es notwendig, dass diese Ground truth Daten mit den IMU-Daten, auf denen die Schätzung basiert, synchronisiert werden können. Es müssen also beide Datensätze zeitlich in Relation gesetzt werden können.

Wenn dieses Experiment noch einmal mit einer korrekt synchronisierten Ground truth durchgeführt wird, muss außerdem evaluiert werden, wie die Positionsgenauigkeit von der Position auf der Rennbahn abhängt. In Kapitel 4.1.6 wurde erklärt, wie die Schätzung auf den Geraden in den Innenbereich driftet. Da dies in den Kurven nicht passiert, ist anzunehmen, dass die Positionsgenauigkeit in den Kurven höher ist als auf den Geraden. Dieses Experiment ist nur an einer Position am Ende einer Geraden durchgeführt worden. Es ist also wahrscheinlich, dass unter Idealbedingungen die maximale Abweichung in den Kurven niedriger als 4.15 m ist.

## Kapitel 5

# Fazit und Ausblick

In dieser Arbeit wurde herausgefunden, dass es möglich ist, nur mit einem Inertialsensor und Umgebungswissen, den Drift einer Positionsschätzung auf einer Radrennbahn zu entfernen. Um dieser Ergebnisse zu erzielen, wurden verschiedene Methoden evaluiert. Alleine die in Kapitel 3.5.2 beschriebene Geschwindigkeitskorrektur hat dafür gesorgt, dass die Positionsschätzung eines im Kreis fahrenden Fahrrads zwar nicht auf der Strecke bleibt, aber die Schätzung in unmittelbarer Nähe der Strecke geblieben ist. Durch die Kombination dieses Ergebnisses mit der Höhenkorrektur aus Kapitel 3.5.1 wurde es sogar geschafft, in zwei von drei Testfällen das Ergebnis auf der Strecke in der Nähe der wahren Position zu halten.

Ein wesentlicher Aspekt, der hervorsteicht ist, dass ein annehmbares Ergebnis erreicht wurde, obwohl bei den IMU-Daten und bei der Höhenkarte Präzision nur eine Nebenrolle gespielt hat. Bei der Aufnahme der IMU-Daten wurde auf eine präzise Montage und eine Rotationskalibrierung der IMU verzichtet. Dieses Problem spiegelt sich zwar im Testdatensatz 2 wieder, konnte aber durch eine systematische Analyse der potenziellen Fehlerquellen herausgerechnet werden. Bei einer Präzisionsanwendung muss darauf geachtet werden, dass die IMU möglichst gerade montiert und danach eingemessen wird, so dass ein Rotationsfehler direkt entfernt werden kann. Während dieses Einmessens sollte neben der Rotation auch die Gravitation gemessen werden, die, je nachdem an welchem Ort man sich befindet, unterschiedlich ist.

Auf die potenziell falsche Höhenkarte wurde bereits in Kapitel 4.1.6 eingegangen. Der in dieser Arbeit vorgestellte Ansatz hat trotzdem funktioniert, da für die Entfernung des Drifts hauptsächlich die Geschwindigkeitskorrektur verantwortlich war. Die Höhenkorrektur wurde dann nur zum Entfernen des Rotationsfehlers benötigt, den die Geschwindigkeitskorrektur nach jeder Kurve aufakkumuliert hat.

Die Verwendung eines Kalman Filters legt nahe, dass ein System, das auf dem in dieser Arbeit vorgestellten Ansatz basiert, auch in einem Echtzeitszenario eingesetzt werden könnte. So kann beim Training live die Telemetrie überwacht werden, ohne dass teure Messsysteme zum Einsatz kommen müssen. Auch als einfacher Ersatz eines Tachometers ist dieser Ansatz

gut verwendbar. Da die Geschwindigkeit des Fahrrads parallel mitgeschätzt wird und dabei durch die nur einfache Integration einen niedrigeren Fehler erfährt als die Position, kann diese direkt aus dem Zustand entnommen werden und ist dabei ausreichend genau.

# Literaturverzeichnis

- [1] FRESE, Udo ; SCHRÖDER, Lutz: *Theorie der Sensorfusion*. 2016. – Vorlesungsskript an der Universität Bremen
- [2] HERTZBERG, Christoph ; WAGNER, René ; FRESE, Udo ; SCHRÖDER, Lutz: Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. In: *Information Fusion* 14 (2013), Nr. 1, S. 57 – 77
- [3] JULIER, Simon J. ; UHLMANN, Jeffrey K.: A new extension of the Kalman filter to nonlinear systems. In: *Proceedings of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997
- [4] KÁLMÁN, Rudolph E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME–Journal of Basic Engineering* 82 (1960), Nr. Series D, S. 35–45
- [5] KLEIN, Itzik ; FILIN, Sagi ; TOLEDO, Tomer: Pseudo-Measurements as Aiding to INS. 57 (2010), 03, S. 25–34
- [6] LI, Jing ; TAYLOR, George ; KIDNER, David B.: Accuracy and reliability of map-matched GPS coordinates: the dependence on terrain model resolution and interpolation algorithm. In: *Computers & Geosciences* 31 (2005), Nr. 2, S. 241 – 251
- [7] MANDEL, Christian ; LAUE, Tim: Particle filter-based position estimation in road networks using digital elevation models. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, S. 5744–5749
- [8] STOLPMANN, Andreas: Innenraumfußgängerfolgung mit Inertialsensoren und Gebäudeplänen. (2016). – Masterarbeit
- [9] APPLE INC.: *Getting Raw Accelerometer Events*. [https://developer.apple.com/documentation/coremotion/getting\\_raw\\_accelerometer\\_events](https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events), 2017. – Nur online: Abgerufen am 07.12.2017
- [10] APPLE INC.: *Getting Raw Gyroscope Events*. [https://developer.apple.com/documentation/coremotion/getting\\_raw\\_gyroscope\\_events](https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events), 2017. – Nur online: Abgerufen am 07.12.2017

- 
- [11] APPLE INC.: *iPhone 6 - Technische Daten*. <https://support.apple.com/kb/sp705>, 2017.  
– Nur online: Abgerufen am 07.12.2017
- [12] IFIXIT: *iPhone 6 Teardown*. <https://de.ifixit.com/Teardown/iPhone+6+Teardown/29213>, 2014. – Nur online: Abgerufen am 07.12.2017
- [13] INVENSENSE: *MPU-6500*. <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6500-Datasheet2.pdf>, 2017. – Nur online: Abgerufen am 07.12.2017
- [14] *SIXDAYS BREMEN*. <http://sixdaysbremen.de/>, 2017. – Nur online: Abgerufen am 07.12.2017
- [15] SYSTEM PLUS CONSULTING: *InvenSense MP67B*. <http://www.systemplus.fr/reverse-costing-reports/invensense-mp67b-2/>, 2017. – Nur online: Abgerufen am 07.12.2017
- [16] *UCI CYCLING REGULATIONS: PART 3 TRACK RACES*. [http://www.uci.ch/mm/Document/News/Rulesandregulation/16/26/54/3-PIS-20160604-E\\_English.PDF](http://www.uci.ch/mm/Document/News/Rulesandregulation/16/26/54/3-PIS-20160604-E_English.PDF), 2016.  
– Nur online: Abgerufen am 11.12.2017