

# **Analyse der Fehlertoleranz eines Unscented Kalman Filters anhand einer Simulation zur Lokalisierung eines Roboters**

Raoul Bickmann 2217470  
raoul.bickmann@haw-hamburg.de  
HAW Hamburg

## **1 EINLEITUNG**

Kalman Filter Algorithmen und dessen Varianten werden für viele Anwendungen genutzt, um die Genauigkeit von Sensoren zu verbessern. Eine dieser Varianten ist der Unscented Kalman Filter Algorithmus (UKF). In diesem Paper wird dessen Fehlertoleranz in einer Beispielsimulation getestet. Hierzu werden verschiedene Arten von Fehlern in die Konfiguration des UKFs hinzugefügt und deren Auswirkungen anhand der Ergebnisunterschiede überprüft.

## **2 KALMAN FILTER UND UKF**

Der Kalman Filter wird angewendet, um trotz ungenauer Sensordaten, den Zustand eines Systems mit möglichst geringem Fehler bestimmen zu können. Diesen Zustand schätzt der Filter aus den Messdaten und einem Modell des Systems.

Dafür muss, je nach Anwendungsfall, das Verhalten des Systems möglichst exakt mathematisch ausgedrückt werden. Zusätzlich sind die Ungenauigkeit dieses Modells sowie die der Sensoren zu ermitteln. Mit Hilfe dieser Informationen kann der Kalman Filter anschließend eine präzisere Abschätzung des Systemzustandes zum gewünschten Zeitpunkt liefern, als dies lediglich die Vorhersage eines Modells, oder ausschließlich die Messdaten der Sensoren vermögen.

### **2.1 Unscented Kalman Filter**

Der Unscented Kalman Filter wurde von Simon J. Julier und Jeffrey K. Uhlmann als Weiterentwicklung des Kalman Filters für nichtlineare Problemstellungen vorgestellt [1]. Vor dieser Entwicklung existierte bereits der Extended Kalman Filter (EKF), der jedoch für viele Anwendungsmöglichkeiten als komplex zu implementieren oder ungeeignet gilt [1].

Für den UKF führen die Autoren die Unscented Transformation ein. Mit Hilfe dieser, kann der UKF eine bessere Annäherung an das Verhalten einer nicht-linearen Funktion erreichen. Dafür müssen sogenannte Sigma-

Punkte festgelegt werden, wobei es für deren Auswahl eine Reihe verschiedener Verfahren gibt [2]. Hierzu gehören beispielsweise die initial vorgestellte Vorgehensweise von Julier und Uhlmann [1] oder ein von Rudolph van der Merwe entwickeltes Verfahren [3].

## **3 SIMULATION**

Für die Analyse des Kalman Filters wurde eine bereits bestehende Simulation eines sich bewegenden Roboters verwendet. Hierbei misst der Roboter die Entfernung zu bekannten Orientierungspunkten. Aus diesen Messwerten und unter Nutzung eines UKFs kann er seine Position feststellen.

Diese Simulation wurde von Roger R Labbe Jr. in seinem Buch „Kalman and Bayesian Filters in Python“ [4] veröffentlicht. Das Buch ist als Begleitung für Labbes Bibliothek Filterpy [5] gedacht. Die Bibliothek implementiert verschiedene Varianten des Kalman Filter Algorithmus, darunter auch den UKF. Diese UKF-Implementierung findet ebenfalls Verwendung in der Simulation zur Lokalisation des Roboters [6].

### **3.1 Robotermodell der Simulation**

In der Simulation verwendet Labbe einen Roboter mit zwei Rädern, angeordnet wie die Räder eines Fahrrads. Für dieses Modell muss der Radabstand zwischen den beiden Rädern ausgewählt werden. Dieser wurde in der vorliegenden Simulation auf zunächst 0,5 Meter festgelegt.

### **3.2 Konfiguration des UKFs**

Für die Simulation konfiguriert der Autor den UKF mit den anwendungsspezifischen Funktionen und Variablen.

Hierzu gehört die Zustandstransferfunktion, die aus einem Zustand mit Hilfe des Control Inputs des Kalman Filters die Schätzung für den nächsten Zustand berechnet. Des Weiteren wird die Messfunktion implementiert, die zu einem gegebenen Zustand den zugehörigen Messwert zurückgibt.

Zusätzlich zu Zustandstransfer- und Messfunktion wird eine Reihe von Hilfsmethoden erstellt, die in diesen benutzt werden, um beispielsweise die Winkel zu vereinheitlichen oder Differenzen zu berechnen.

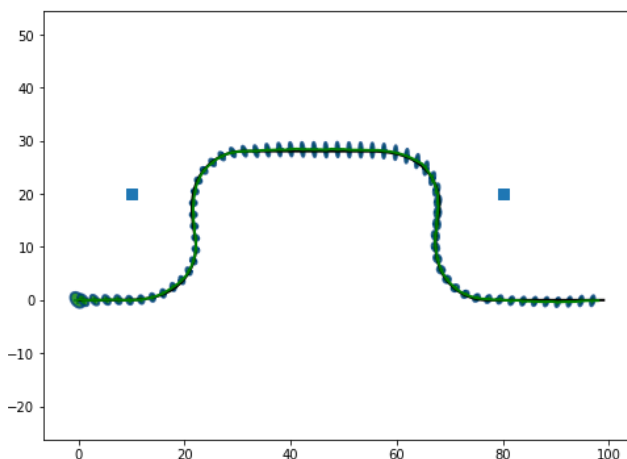
Für den UKF nötigen Sigma-Punkte werden ebenfalls mit einer Funktion von Filterpy generiert. Diese verwendet ein von Rudolph van der Merwe entwickeltes Verfahren [3].

Abschließend werden die Werte für das Messrauschen und das Prozessrauschen des UKFs konfiguriert. Diese können in der Simulation aus der Erzeugung der Simulationsdaten gefolgert werden, wogegen sie in der Realität aus den Richtwerten der Sensoren und durch Experimentieren bestimmt werden müssen.

Der vollständige Code der Simulation kann in Kapitel zehn des Buches [6] oder im Github-Repository zu diesem Paper eingesehen werden [7].

## 4 DURCHFÜHRUNG

Für die Analyse des so erstellten UKFs wurde eine Strecke festgelegt, die der Roboter bei jedem Simulationsdurchlauf zurücklegen sollte [Abbildung 1]. Hierbei gibt es zwei Orientierungspunkte zu denen der Roboter die Entfernung misst.



**Abbildung 1: Zurückgelegte Strecke des Roboters. In schwarz tatsächliche Strecke, Schätzung des UKFs in grün, Orientierungspunkte in blau**

Um die Fehlertoleranz des UKFs zu ermitteln, wurden 1000 Simulationen mit der Standardkonfiguration (Konfiguration 1) des UKFs durchgeführt. Anschließend gab es weitere vier Durchläufe, mit jeweils 1000 Simulationen, bei denen die Konfiguration des UKFs durch Fehler von den richtigen Werten der Simulation abweicht.

Die fehlerhaften Parameter wurden normalverteilt mit, je nach Konfiguration wechselnden, Standardabweichungen randomisiert.

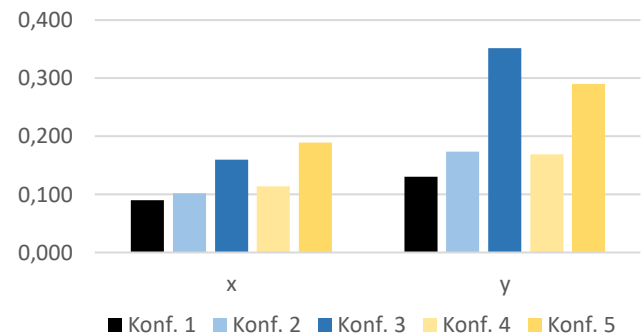
Beim ersten Durchlauf beträgt die Standardabweichung des Radabstands über die 1000 Simulationen  $SD = 0,05m$  (Konfiguration 2), beim zweiten Durchlauf  $SD = 0,125m$  (Konfiguration 3). Zwei weitere Durchläufe wurden durchgeführt, bei denen stattdessen die Position der Orientierungspunkte fehlerhaft ist. Diese haben eine

Standardabweichung in Richtung x und y von  $SD = 0,1m$  (Konfiguration 4) und  $SD = 0,25m$  (Konfiguration 5).

Anschließend konnte die Performance des UKFs in den jeweiligen Konfigurationen miteinander verglichen werden.

### 4.1 Methode

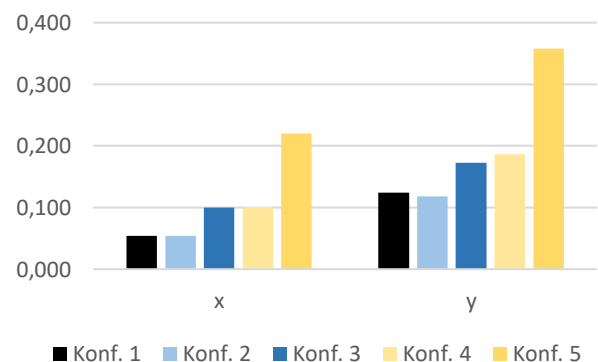
Für jede der Konfigurationen wurden 1000 Simulationen durchgeführt. Dabei wurde währenddessen die approximierte Position des UKFs mit der richtigen Position Zeitpunkt verglichen.



**Abbildung 2: Mittelwerte der durchschnittlichen Abweichung pro Simulation des UKFs über 1000 Simulationen in x,y-Richtung in Metern**

Daraus wurde die Differenz der approximierten zur simulierten Position errechnet. Als Folge dessen konnte für jeden Simulationsdurchlauf zum einen die durchschnittliche Abweichung über den Verlauf der Simulation, zum anderen die finale Abweichung am Ende der Simulation berechnet werden.

Anschließend wurden für die Simulationen jeder Konfiguration die Mittelwerte dieser beiden Abweichungen berechnet und anschließend miteinander verglichen.



**Abbildung 3: Mittelwerte der finalen Abweichungen des UKFs über 1000 Simulationen in x,y-Richtung in Metern**

In Abbildung 2 sind die Mittelwerte der durchschnittlichen Abweichungen dargestellt. In Abbildung 3 die Mittelwerte der finalen Abweichungen.

## 5 ERGEBNISSE

### 5.1 Analyse

Wie sich an Abbildung zwei erkennen lässt, sind sowohl bei der Konfiguration 2 sowie bei der Konfiguration 4 die Unterschiede zu den Abweichungen der Standardkonfiguration sehr gering. So ist die Differenz der durchschnittlichen Abweichungen zwischen der Konfiguration 1 und 2 circa 0,012m in x- und 0,044m in y-Richtung. Zwischen Konfigurationen 1 und 4 liegen die Differenzen bei 0,024m und 0,039m.

Bei den Konfigurationen 3 und 5 mit stärkeren Abweichungen zu den simulierten Werten, lassen sich größere Differenzen feststellen. Diese liegen bei Konfiguration 3 bei 0,070m in x- und 0,222m in y-Richtung und bei Konfiguration 5 bei 0,099m und 0,160m.

Die Mittelwerte der finalen Abweichungen zeigen ähnliche Ergebnisse. Hier sind ebenfalls bei Konfigurationen 2 (x: <0,000m y: -0,005m) und 4 (x: 0,047m, y: 0,063m) nur geringe Unterschiede zur Standardkonfiguration festzustellen. Bei den y-Werten von Konfiguration 2 bestand, aufgrund der Schwankungen des UKFs, im Schnitt sogar eine leicht geringere Abweichung am Ende der Simulation. Zusätzlich sind hier auch bei Konfiguration 3 (x: 0,046m y: 0,049m) die Werte sehr eng an den Werten der Standardkonfiguration.

Nur Konfiguration 5 (x: 0,166m y: 0,234m) zeigt zu diesem Zeitpunkt wieder deutlich größere Abweichungen.

### 5.2 Interpretation

Bei den Konfigurationen 2 und 4 ist zu erkennen, dass trotz der Fehler in der Konfiguration, die durchschnittliche Abweichung des Filters nur um bis zu vier Zentimeter ansteigt. Dies ist der Fall, obwohl die Konfigurationsfehler selbst bei beiden in der gleichen Größenordnung von mehreren Zentimetern liegen.

Bei Konfiguration 2 beträgt die Abweichung im Mittel 5cm, dies entspricht 10 Prozent des eigentlichen Radabstandes von 50 Zentimetern, und bei Konfiguration 4 sogar 10cm in x- sowie y-Richtung.

Messfehler dieser Höhe sollten bei der Konfiguration eines UKFs durch entsprechende Sorgfalt vermieden werden können.

Bei den Konfigurationen 3 und 5 sind die Abweichungen dagegen wesentlich angestiegen. Diese sind mindestens viermal so hoch, obwohl sich die Konfigurationsfehler im Mittel nur um das 2,5-fache gegenüber den vorherigen Konfigurationen erhöht haben.

Bei so deutlichen Ungenauigkeiten in der Konfiguration kann der UKF dies scheinbar nicht mehr effektiv ausgleichen.

Jedoch driften auch bei diesen verhältnismäßig großen Konfigurationsfehlern, die Werte des Kalman Filters nicht endgültig ab, sondern schwanken weiter in der Nähe der richtigen Werte. Dies lässt sich auch an den finalen Abweichungen in Abbildung 3 erkennen, die im gleichen Rahmen liegen, wie die durchschnittlichen Abweichungen.

Je nach Anwendungsfall muss dann evaluiert werden, ob die Konfigurationsfehler geringer gehalten werden können, oder ob die Abweichungen des UKFs akzeptabel sind.

Denkbar wären beispielsweise medizinische Anwendungsfälle, bei denen selbst die relativ geringen Abweichungen des UKFs, wie bei den Simulationen der ersten Konfigurationen, bereits zu fatale Folgen haben könnten.

In diesem Fall müsste die Konfiguration des UKFs präziser durchgeführt werden. Eventuell ist dann die Nutzung eines UKFs ohnehin nicht die korrekte Wahl, da dieser immer eine gewisse Ungenauigkeit hat.

Bei anderen Anwendungsfällen, wie beispielsweise der Navigation eines Roboters, ähnlich wie in dieser Simulation, ist es denkbar, dass auch Abweichungen von mehreren Zentimetern für die Orientierung in dem zu navigierenden Gebiet hinreichend sein können.

## 6 FAZIT

Abschließend lässt sich sagen, dass der Unscented Kalman Filter trotz Messfehlern bei der Konfiguration vergleichbar präzise Ergebnisse liefern kann, wie ein exakt konfigurierter Filter. Ob dies ausreichend ist, muss im Einzelfall für die jeweilige Anwendung beurteilt werden.

Bei weiteren Untersuchungen könnte die Analyse mit Simulationen aus anderen Anwendungsgebieten wiederholt werden, um die vorliegenden Ergebnisse zu validieren.

Auch denkbar wäre eine Wiederholung mit mehreren gleichzeitig abweichenden Faktoren, wie beispielsweise Fehler im Radabstand und in der Position der Orientierungspunkte im gleichen Simulationsdurchlauf. Hierdurch könnte zusätzlich evaluiert werden, ob mehrere Abweichungen an verschiedenen Stellen, die Ergebnisse stärker beeinflussen als Abweichungen an einzelnen Punkten.

## 7 QUELLEN

[1] Simon J. Julier, Jeffrey K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, (28.07.1997); <https://doi.org/10.1117/12.280797>

[2] H. M. T. Menegaz, J. Y. Ishihara, G. A. Borges and A. N. Vargas, "A Systematization of the Unscented Kalman Filter Theory," in IEEE Transactions on Automatic Control, vol. 60, no. 10, pp. 2583-2598, Oktober 2015. <https://doi.org/10.1109/TAC.2015.2404511>

[3] Rudolph van der Merwe: "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models", Dissertation, 04.2004

[4] Roger R Labbe Jr, "Kalman and Bayesian Filters in Python", <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>, 2015, zuletzt abgerufen: 12.12.2019

[5] Roger R Labbe Jr, Filterpy, <https://github.com/rlabbe/filterpy>, 2015, zuletzt abgerufen: 12.12.2019

[6] Roger R Labbe Jr, “Kalman and Bayesian Filters in Python”, <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python/blob/master/10-Unscented-Kalman-Filter.ipynb>, 2015, zuletzt abgerufen: 12.12.2019

[7] <https://github.com/RaoulBickmann/UKFAnalyse>