Bachelorarbeit
BSC-064

Mechanik und Meerestechnik

Adaption und Vergleich von nichtlinearen Filtermethoden zur Selbstlokalisierung auf einem Feld mit dem humanoiden NAO-Robotiksystem

von
Lasse Peters

Betreuer: Dr. L. Dostal
Prof. Dr.-Ing. R. Seifried

Technische Universität Hamburg-Harburg (TUHH)
Institut für Mechanik und Meerestechnik
Prof. Dr.-Ing. R. Seifried

Hamburg, September 2017

Inhaltsverzeichnis

Kapitel 1

Einleitung

Robotiksysteme werden mehr und mehr integraler Bestandteil des alltäglichen Lebens und sind schon heute aus vielen Bereichen nicht mehr weg zu denken. Während die ersten Industrieroboter schon Ende der 1950er Jahre eingesetzt wurden, sind autonom agierende Robotiksysteme erst in jüngerer Vergangenheit in den Fokus von Forschung und Industrie gerückt, da diese Technologie erst durch leistungsfähige Sensorik, effiziente Recheneinheiten und mobile Energieversorgung ermöglicht wurde. So finden im Rahmen des RoboCup seit 1997 jährliche Wettbewerbe autonomer Systeme in verschiedenen Disziplinen statt [1]. Eine weitere populäre Anwendung stellt auch die erstmalige Durchführung der DARPA Grand Challenge im Jahr 2004 dar, in deren Kontext autonome Fahrzeuge für eine unbemannte Geländefahrt entwickelt wurden [2].

Zentraler Aspekt autonomer Systeme ist die eigenständige Interaktion mit der Umwelt. Eine wesentliche Voraussetzung für eine solche autonome Interaktion stellt die Verarbeitung der eingehenden Sensordaten dar, um daraus ein abstraktes Weltmodell zu generieren. Diese Abstraktion wird im Allgemeinen von digitalen Filtern übernommen, die die Informationen verschiedener Quellen kombinieren, um daraus eine höherwertige Information zu gewinnen.
Ein Beispiel eines solchen Filterproblems ist die sogenannte Selbstlokalisierung – das heißt die Bestimmung von Position und Orientierung eines Systems relativ zu seiner Umwelt. Eingangsgrößen solcher Filter sind vielfältig, unterschiedlich zuverlässig und reichen von GPS- über Beschleunigungs- bis hin zu Kameradaten. Populäre Methoden zur Lösung solcher Lokalisierungsprobleme basieren auf Filteralgorithmen wie dem Partikel-Filter oder verschiedenen Varianten des Kalman-Filters [3, 4, 5]. Ebene diese Algorithmen finden auch Verwendung in den zuvor genannten Beispielen [6, 7]. Im Rahmen dieser Arbeit werden mit dem Partikel-Filter (PF) sowie dem Unscented Kalman-Filter (UKF) zwei nichtlineare Filtermethoden zur Selbstlokalisierung in bekannter Umwelt verglichen.

1.1 Motivation

Die RoboCup Standard Platform League (SPL) ist eine Liga des internationalen Forschungswettbewerbs, bei dem alle Teams mit der gleichen Roboterplattform – zur Zeit dem humanoiden NAO – im autonomen Fußballspiel gegeneinander antreten. Hier stellt die zuverlässige Bestimmung von Position und Orientierung im zweidimensionalen Weltmodell – im Folgenden `2D-Pose` genannt – eine grundlegende Voraussetzung für den Erfolg im Wettbewerb dar. Jegliche Prozesse, die sich auf ein absolutes Referenzkoordinatensystem beziehen, hängen inhärent von der Genauigkeit der Schätzung der 2D-Pose ab. Davon betroffen ist vor allem der Austausch von Informationen zwischen den Robotern, wie z.B. die Kommunikation der Positionen von Hindernissen, Gegnern oder dem Ball. Ausgehend davon sind somit auch Fähigkeiten wie das Pass- und Stellungsspiel und damit die elementare Fähigkeit ein Tor zu erzielen bzw. zu verhindern auf eine gute und zuverlässige Selbstlokalisierung angewiesen.

1.2 Zielsetzung

Ziel dieser Arbeit ist der Vergleich zweier Lokalisierungsstrategien auf Basis des PF bzw. UKF zur echtzeitfähigen Anwendung auf dem NAO. Als Anwendungsbeispiel dient die Selbstlokalisierung eines humanoiden NAO-Robotiksystems auf einem Fußballfeld nach den Regeln der RoboCup SPL. Ausgangspunkt ist eine vorhandene Implementierung eines PF, wie sie vom RoboCup Team HULKs in der SPL eingesetzt wird. Zum Vergleich wird im Rahmen dieser Arbeit ein alternativer Algorithmus auf Basis des UKF entwickelt.
In dieser Arbeit wird erläutert, welche Rahmenbedingungen an einen Lokalisierungsalgorithmus im Kontext des hier gewählten Anwendungsbeispieles gestellt werden und welche Informationen der NAO zu Lokalisierungszwecken über seine Umwelt extrahieren kann. Es wird aufgezeigt, welche Anforderungen sich daraus für eine Filtermethode ergeben und beschrieben, wie diese bei der Realisierung des vorhandenen PF berücksichtigt wurden. Darauf aufbauend wird im Rahmen dieser Arbeit ein UKF adaptiert, dass an Stelle des PF im Software-Framework der HULKs implementiert wird. Die beiden Filtermethoden werden schließlich bezüglich ihrer Leistungsfähigkeit simulativ evaluiert, sowie Vor- und Nachteile beider Methoden diskutiert.

1.3 Gliederung der Arbeit

In Kapitel 2 werden zunächst die Grundlagen der hier verwendeten Filteralgorithmen im Hinblick auf Funktionsweise und theoretische Leistungsfähigkeit beleuchtet. Danach werden in Kapitel 3 Details der Problemstellung sowie die Rahmenbedingungen formuliert, unter denen die verwendeten Lokalisierungsalgorithmen operieren. In Kapitel 4 wird dann die algorithmische Realisierung des vorhandenen PF, sowie die der im Rahmen dieser Arbeit entwickelten Adaption eines UKF erörtert. Die Leistungsfähigkeit beider Filtermethoden wird in Kapitel 5 evaluiert und verglichen. Schließlich werden in Kapitel 6 die Ergebnisse dieser Arbeit zusammengefasst.

Kapitel 2

Theoretische Grundlagen

Im Fokus dieser Arbeit steht der Vergleich zweier nichtlinearer Filtermethoden im Kontext einer Lokalisierungsanwendung. Dazu wird ein PF mit einer hier entwickelten Methode verglichen, die im Kern das algorithmische Vorgehen des UKF nutzt. Bevor in Kapitel 4 die Adaption der genannten Algorithmen zur Anwendung auf das spezielle Lokalisierungsproblem beschrieben wird, werden in diesem Kapitel zunächst einige theoretische Grundlagen zur Funktionsweise und der theoretischen Leistungsfähigkeit der zugrundeliegenden Filtermethoden erläutert. Zu diesem Zweck werden zunächst die wesentlichen Notationskonventionen (Abschnitt 2.1) sowie das allgemeine Filterproblem (Abschnitt 2.2) vorgestellt. In Abschnitte 2.3 bis 2.5 werden darauf aufbauend die einzelnen Filtermethoden bezüglich ihrer Einschränkungen und Funktionsweise erläutert. Dazu wird zunächst das klassische Kalman-Filter (KF) in Abschnitt 2.3 beschrieben. Wenngleich das KF selbst im Rahmen des Anwendungsbeispiels nur in Teilen Anwendung findet, bieten die Ausführungen zum Aufbau dieses Filters eine geeignete Grundlage zum Verständnis des UKF. Im darauf folgenden Abschnitt 2.4 wird dann der algorithmische Aufbau des auf dem KF basierenden UKF beleuchtet. Dabei werden die vorangegangenen Ausführungen zum KF genutzt, um die Analogien zwischen diesen beiden Algorithmen aufzuzeigen und so das Verständnis des komplexeren UKF zu erleichtern. In Abschnitt 2.5 wird schließlich das PF beleuchtet.

Während im Folgenden die grundlegenden Funktionsweisen der Filteralgorithmen zum Verständnis der späteren Adaptionen erläutert werden, hat diese Arbeit nicht den Anspruch, die einzelnen Teilschritte der Algorithmen im Detail zu begründen. Ausführliche Erläuterungen zu mathematischen Hintergründen, sowie zur Herleitung der Algorithmen sind unter anderem in [5, 8] umfangreich dokumentiert.

2.1 Konventionen

Die im Rahmen dieser Arbeit verwendeten Notationen orientieren sich an den Konventionen aus [5]. Nachfolgend werden die wesentlichen, im weiteren Verlauf dieser Arbeit verwendeten Konventionen vorgestellt. Für ausführliche Erläuterungen zu grundlegenden Konventionen und Begriffserläuterungen, die dem allgemeinen Standard zugeordnet werden können, sei auf die einleitenden Kapitel von [5] verwiesen.

Zufallsvariablen Eine Zufallsvariable ist eine Abbildung, die jedem Ergebnis eines Zufallsversuches einen quantitativen Wert zuordnet. Die Ausprägung einer Zufallsvariablen gehorcht stochastischen Gesetzten. Zufallsvariablen werden zur Unterscheidung von Matrizen als große, aufrechte, lateinische Buchstaben notiert. Vektorwertige Zufallsvariablen werden fett gedruckt – Beispiel X bzw. $\mathbf{X}$.

Normalverteilung Die Verteilung einer Zufallsvariablen $\mathbf{X}$ im Sinne einer Normalverteilung mit Erwartungswert $\boldsymbol{x}$ und Kovarianz $\boldsymbol{P}$ wird als $\mathbf{X} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{P})$ notiert.

2.2 Das Filterproblem

Zunächst wird das Filterproblem, wie es von den nachfolgend behandelten Filteralgorithmen modelliert wird, definiert. Kern des allgemeinen Filterproblems ist die Schätzung des Zustandes $\boldsymbol{x} \in \mathbb{R}^{d_x}$ eines Systems. Allgemein sind dabei zeitkontinuierliche und zeitdiskrete Systeme denkbar. Auf Grund der Tatsache, dass das später betrachtete Anwendungsbeispiel eine digitale Implementierung behandelt, wird im Folgenden ausschließlich auf die zeitdiskrete Problematik weiter eingegangen.

Im Kontext der zeitdiskreten Problemformulierung wird die Folge der Zustände $\boldsymbol{x}_t$ als Markov-Kette betrachtet. Dabei indiziert $t \in \mathbb{N}$ die einzelnen Zeitschritte. Durch die Markoveigenschaft wird spezifiziert, dass der aktuelle Zustand genau so viele Informationen über künftige Systemzustände enthält, wie die Gesamtheit vergangener Systemzustände. Die mathematische Definition einer Markov-Kette ist durch

$$\begin{aligned} &p(\mathbf{X}_{t+1} = \boldsymbol{x}_{t+1} \mid \mathbf{X}_t = \boldsymbol{x}_t, \mathbf{X}_{t-1} = \boldsymbol{x}_{t-1}, \ldots, \mathbf{X}_1 = \boldsymbol{x}_0) \\ =&p(\mathbf{X}_{t+1} = \boldsymbol{x}_{t+1} \mid \mathbf{X}_t = \boldsymbol{x}_t) \end{aligned} \tag{2.1}$$

gegeben.

Im Rahmen des Filterproblems wird die initiale Wahrscheinlichkeitsverteilung des Zustandes $\mathbf{X}_0$ als gegeben vorausgesetzt. Für die Dynamik des d_x-dimensionalen Systemzustandes $\mathbf{X}$ wird das Zustandsraummodell

$$\mathbf{X}_t = \boldsymbol{f}_t\left(\mathbf{X}_{t-1}, \boldsymbol{u}_{t-1}, \mathbf{W}_{t-1}\right) \tag{2.2}$$

aufgestellt. Darin ergibt sich die Zufallvariable des Systemzustands $\mathbf{X}_t$ mit Hilfe der Zustandsübergangsfunktion $\boldsymbol{f}_t$ aus dem vorangegangenen Zustand $\mathbf{X}_{t-1}$. Der in das System zwischen Zeitpunkt $t-1$ und t eingebrachte Control-Input sei mit $\boldsymbol{u}_{t-1}$ bezeichnet und wird als bekannt vorausgesetzt. Des Weiteren wird das zufällige Rauschen der Systemdynamik mit $\mathbf{W}_{t-1}$ bezeichnet.

Die Wahrscheinlichkeit $p(\boldsymbol{x}_t|\boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1})$, dass die Zufallsvariable des Zustandes den Wert $\mathbf{X}_t = \boldsymbol{x}_t$ annimmt, hängt somit vom letzten Zustand $\boldsymbol{x}_{t-1}$, sowie dem Control-Input $\boldsymbol{u}_{t-1}$ ab. Zentrale Problematik der Zustandsschätzung ist die Tatsache, dass der Zustand $\boldsymbol{x}_t$ nicht direkt beobachtet werden kann. Stattdessen wird eine Messung $\boldsymbol{z}_t \in \mathbb{R}^{d_z}$ durchgeführt, deren Wahrscheinlichkeit selbst durch den aktuellen Zustand nach $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ bedingt ist [9]. Die Abbildung des Zustandes $\mathbf{X}_t$ auf die korrespondierende Messung $\mathbf{Z}_t$ erfolgt mit

$$\boldsymbol{Z}_t = \boldsymbol{h}_t\left(\mathbf{X}_t, \mathbf{V}_t\right) \tag{2.3}$$

durch Anwendung der Messfunktion $\boldsymbol{h}_t$. Darin bezeichnet $\mathbf{V}_t$ die Zufallsvariable des Messrauschens.

Während der tatsächliche Wert der Rauschkomponenten $\mathbf{W}$ und $\mathbf{V}$ unbekannt ist, wird vorausgesetzt, dass die Wahrscheinlichkeitsdichtefunktion dieser Zufallsvariablen bekannt ist. Darüber hinaus wird gefordert, dass die Rauschkomponenten voneinander unabhängig und zeitlich unkorreliert sind. Abschließend wird für die einzelnen Messungen $\mathbf{Z}_t$ vorausgesetzt, dass diese gegeben $\mathbf{X}_t$ bedingt unabhängig sind. Damit wird insbesondere gefordert, dass die Wahrscheinlichkeitsverteilung der Messung $\mathbf{Z}_t$ nur vom Zustand $\mathbf{X}_t$ nach $p(\mathbf{Z}_t|\mathbf{X}_t)$ abhängt.

Die Aufgabe des Filters besteht also zusammengefasst in der Schätzung der zeitlichen Entwicklung des Systemzustandes $\mathbf{X}_t$ durch Beobachtung der Größe $\mathbf{Z}_t$ (Gl. (2.3)) bei gegebenen Verteilungsfunktionen der Rauschkomponenten, sowie bekannter Systemdynamik (Gl. (2.2)).

2.3 Kalman-Filter

Das `Kalman-Filter` (KF) ist ein Algorithmus zur Behandlung linearer Filterprobleme. Das Verfahren wurde 1960 von Rudolf E. Kálmán veröffentlicht [10].

Wesentliche Einschränkung des Kalman-Filters ist die Tatsache, dass es nur auf lineare Systeme angewendet werden kann. Die notwendigen Einschränkungen des Filterproblems aus Abschnitt 2.2, die eine Anwendung des KF erlauben, werden in Abschnitt 2.3.1 erläutert. Anschließend wird in Abschnitt 2.3.2 der algorithmische Aufbau des KF erklärt.

2.3.1 Einschränkung der allgemeinen Problemformulierung

Als lineares Filter erfordert die Anwendung des KF eine Einschränkung des in Abschnitt 2.2 aufgeführten Filterproblems. Diese Einschränkungen seien wie folgt spezifiziert:

1. Die Systemdynamik $\boldsymbol{f}_t$ sowie die Messfunktion $\boldsymbol{h}_t$ sind linear.
2. Der Zustand $\mathbf{X}_t$ ist eine Gaußsche Zufallsvariable (GZV). Es gilt daher $\mathbf{X} \sim \mathcal{N}(\boldsymbol{x}_t, \boldsymbol{P}_t)$.
3. Alle auftretenden Fehler sind normalverteilt und verschwinden im statistischen Mittel. Für die GZV von System- und Messrauschen wird entsprechend $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_t)$ bzw. $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ gefordert.

Für lineare Systeme mit normalverteiltem Rauschen in System- und Messdynamik erlaubt das KF eine erwartungstreue Schätzung des Systemzustandes. Unter diesen Voraussetzungen besitzt die vom Filter errechnete Zustandsschätzung minimalen quadratischen Fehler. Das KF kann daher unter den genannten Voraussetzungen als optimales Filter bezeichnet werden.

Auf Grund der geforderten Linearität für Systemdynamik $\boldsymbol{f}_t$ und Messdynamik $\boldsymbol{h}_t$, können diese Abbildungen als lineare Matrixgleichungen formuliert werden. Eine lineare Formulierung der Systemdynamik ist daher mit

$$\mathbf{X}_t = \boldsymbol{F}_{t-1}\mathbf{X}_{t-1} + \boldsymbol{B}_{t-1}\boldsymbol{u}_{t-1} + \mathbf{W}_{t-1} \tag{2.4}$$

gegeben. Darin beschreibt die Systemmatrix $\boldsymbol{F}_{t-1} \in \mathbb{R}^{d_x \times d_x}$ den direkten Einfluss des letzten Zustandes $\mathbf{X}_{t-1}$ auf den nächsten Zustand $\mathbf{X}_t$. Die Dynamik des `Control-Inputs` $\boldsymbol{u}_{t-1} \in \mathbb{R}^{d_u}$ wird durch die Matrix $\boldsymbol{B}_{t-1} \in \mathbb{R}^{d_x \times d_u}$ beschrieben.

Analog dazu kann auch die Messdynamik in linearer Form als

$$\mathbf{Z}_t = \boldsymbol{H}_t \mathbf{X}_t + \mathbf{V}_t \tag{2.5}$$

formuliert werden. Darin beschreibt die Matrix $\boldsymbol{H}_t \in \mathbb{R}^{d_z \times d_x}$ die Abbildung des Zustandes in den Messraum.

2.3.2 KF-Algorithmus

Für eine digitale Implementierung ist eine zeitdiskrete Formulierung des Filters erforderlich. Entsprechend wird nachfolgend der Algorithmus des zeidiskreten Kalman-Filters erläutert. Wie auch die weiteren hier behandelten Algorithmen, ist das KF als Prädiktor-Korrektor-Verfahren formuliert. Nachfolgend seien Prädiktions- und Korrekturschritt separat beschrieben.
Zur Anwendung des Algorithmus wird jeweils ein Prädiktionsschritt zwischen Zeitpunkt $t-1$ und t ausgeführt. Darauf aufbauend folgt optional ein Korrekturschritt, falls eine Messung $\boldsymbol{z}_t$ für den entsprechenden Zeitpunkt vorhanden ist. Berechnungsergebnis des Algorithmus ist eine Zustandsschätzung $\mathbf{X}_t \sim \mathcal{N}(\boldsymbol{x}_t, \boldsymbol{P}_t)$ als GZV zu den fortschreitenden Zeitpunkten t.

Prädiktion

Im Schritt der Prädiktion wird der Zustand des Systems aus der letzten Zustandsschätzung durch

$$\bar{\boldsymbol{x}}_t = \boldsymbol{F}_{t-1}\boldsymbol{x}_{t-1} + \boldsymbol{B}_{t-1}\boldsymbol{u}_{t-1}, \tag{2.6}$$

$$\bar{\boldsymbol{P}}_t = \boldsymbol{F}_{t-1}\boldsymbol{P}_{t-1}\boldsymbol{F}_{t-1}^T + \boldsymbol{Q}_{t-1} \tag{2.7}$$

prädiziert. Prädizierte Größen, namentlich der prädizierte Zustandsmittelwert $\bar{\boldsymbol{x}}_t$ sowie die prädizierte Zustandskovarianz $\bar{\boldsymbol{P}}_t$, werden jeweils mit einem Querbalken gekennzeichnet. Diese so gekennzeichneten Größen werden im Folgenden auch als a-priori-Zustand bezeichnet, um auf den Umstand hinzuweisen, dass es sich dabei um die prädizierte Schätzung vor Integration der Messung handelt.

Die hier beschriebene Prädiktion entspricht der Formulierung aus Gl. (2.4) zur Transformation der Zustandszufallsvariablen. Es ist leicht ersichtlich, dass die Unsicherheit des Zustandes, repräsentiert durch dessen Kovarianz $\boldsymbol{P}$, im Prädiktionsschritt stets größer wird, da die positiv definite Kovarianzmatrix $\boldsymbol{Q}$ addiert wird.

Korrektur

Im Rahmen des Korrekturschrittes wird die Messung in die Zustandsschätzung einbezogen und somit die Prädiktion wie folgt korrigiert:

$$\boldsymbol{y}_t = \boldsymbol{z}_t - \boldsymbol{H}_t \bar{\boldsymbol{x}}_t, \tag{2.8}$$

$$\boldsymbol{S}_t = \boldsymbol{H}_t \bar{\boldsymbol{P}}_t \boldsymbol{H}_k^T + \boldsymbol{R}_t. \tag{2.9}$$

Die Messung $\mathbf{Z}_t \sim \mathcal{N}(\boldsymbol{z}_t, \boldsymbol{R}_t)$ ist als GZV durch Mittelwert $\boldsymbol{z}_t$ und Kovarianz $\boldsymbol{R}_t$ vollständig beschrieben. In Gl. (2.8) wird das Residuum $\boldsymbol{y}_t$ als Differenz der Messung $\boldsymbol{z}_t$ und der prädizierten Messung $\bar{\boldsymbol{z}}_t = \boldsymbol{H}_t \bar{\boldsymbol{x}}_t$ berechnet. Die dazugehörige Kovarianz $\boldsymbol{S}_t$ ergibt sich dann aus der in den Messraum transformierten Zustandskovarianz, sowie der Kovarianz der Messung $\boldsymbol{R}_t$.

Das Residuum $\mathbf{Y}_t \sim \mathcal{N}(\boldsymbol{y}_t, \boldsymbol{S}_t)$ – auch Innovation genannt – beschreibt also die Abweichung der in diesem Zustand getätigten Messung $\boldsymbol{z}_t$ von der in diesem Zustand erwarteten Messung $\bar{\boldsymbol{z}}_t$. Durch die Residualkovarianz $\boldsymbol{S}_t$ wird beschrieben, wie groß die Sicherheit dieser Abweichung ist.
Aus der Innovation wird die Kalman-Matrix (auch Kalman-Gain genannt)

$$\boldsymbol{K}_t = \bar{\boldsymbol{P}}_t \boldsymbol{H}_t^T \boldsymbol{S}_t^{-1} \tag{2.10}$$

berechnet. Sie beschreibt, wie die gemessene Abweichung in die Korrektur des Zustandes eingeht. Die Kalman-Matrix bildet vom Messraum in den Zustandsraum ab und stellt dabei eine dem Verhältnis der Kovarianzen entsprechende Verstärkung ein. Aus Gl. (2.10) wird ersichtlich, dass das Maß der Verstärkung von dem Verhältnis der Unsicherheiten von Zustand und Residuum abhängt. Ist die Unsicherheit des Zustandes klein im Verhältnis zur Residualkovarianz, wird der Zustand nur in kleinem Maße korrigiert.

Abschließend wird der Zustand mit Hilfe des Residuums $\mathbf{Y}_t \sim \mathcal{N}(\boldsymbol{y}_t, \boldsymbol{S}_t)$ unter Berücksichtigung des Kalman-Gains $\boldsymbol{K}_t$ zu

$$\boldsymbol{x}_t = \bar{\boldsymbol{x}}_t + \boldsymbol{K}_t \boldsymbol{y}_t, \tag{2.11}$$

$$\boldsymbol{P}_t = \bar{\boldsymbol{P}}_t - \boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^T \tag{2.12}$$

korrigiert. Aus Gl. (2.12) ist ersichtlich, dass die Unsicherheit des Zustandes im Schritt der Korrektur stets kleiner wird, da von der Zustandskovarianz $\boldsymbol{P}_t$ der positiv definite Ausdruck $\boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^T$ subtrahiert wird.

Die vorangegangenen Erläuterungen zu theoretischen Grundlagen des KF basieren auf [5, 10] und können dort in eingehenderem Detail nachvollzogen werden.

2.4 Unscented Kalman-Filter

Das „Unscented Kalman-Filter“ (UKF) ist ein nichtlineares Filter, welches auf Basis des klassischen Kalman-Filters (KF) von Julier und Uhlmann entwickelt wurde [11].
Das UKF überträgt die algorithmische Idee des KF auf nichtlineare Prozesse. Analog zum KF wird der Zustand des Systems dabei als GZV approximiert. Während die direkte Erweiterung des KF für nichtlineare Probleme - das Extended Kalman-Filter (EKF) – eine analytische Linearisierung der Zustandübergangsfunktion $\boldsymbol{f}_t$ und Messfunktion $\boldsymbol{h}_t$ erfordert, kommt das UKF ohne eine solche direkte Linearisierung aus.

Zentrale Innovation des UKF ist die Anwendung der sogenannten Unscented Transformation (UT). Diese in Abschnitt 2.4.2 beschriebene Transformation für Zufallsvariablen vermeidet die Linearisierung der Abbildungsvorschrift, indem stattdessen eine analytisch bestimmte repräsentative Stichprobe der Ausgangsverteilung durch Anwendung der vollständigen Nichtlinearitäten transformiert wird. Die so transformierte Stichprobe wird anschließend durch eine GZV approximiert.

Nachfolgend wird zunächst das allgemeine Filterproblem aus Abschnitt 2.2 für die Anwendung des UKF spezifiziert. Anschließend wird das Kernelement des UKF – die UT – beschrieben. Darauf aufbauend wird dann der UKF-Algorithmus in Analogie zu dem des KF erläutert.

2.4.1 Einschränkung der Allgemeinen Problemformulierung

Wird das UKF auf Filterprobleme angewendet, die den Anforderungen des KF aus Abschnitt 2.3.1 genügen, kommen beide Berechnungsvorschriften zum gleichen Ergebnis. Die Schätzgüte des UKF entspricht dann also der des KF. Darüber hinaus erlaubt das UKF jedoch die Behandlung weitaus komplexerer Filterprobleme, da es nicht auf Linearität der Problemstellung angewiesen ist.

Für das UKF kann die in Abschnitt 2.2 spezifizierte Problemformulierung mit der einzigen Einschränkung übernommen werden, dass die Rauschkomponenten analog zur Problemformulierung des KF als $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_t)$ bzw. $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ GVZ mit Mittelwert Null gegeben sind.

Für die Zustandsübergangsfunktion $\boldsymbol{f}_t$ und die Messfunktion $\boldsymbol{h}_t$ wird im Gegensatz zum KF keine Linearität gefordert. Beide Abbildungen werden durch eine korrespondierende UT approximiert. Es sei an dieser Stelle darauf hingewiesen, dass die Approximationsgüte nichtlinearer Abbildungen durch die UT

im Allgemeinen beschränkt ist. Auch wenn das UKF auf nichtlineare Probleme angewendet werden kann, ist eine gute Zustandsschätzung für stark nichtlineare Systeme nicht unbedingt gewährleistet. Die Anwendbarkeit des Filters wird in der Realität vor allem dadurch bestimmt, ob eine Approximation der Zustandszufallsvariablen als GZV sinnvoll ist. Die Güte einer solchen Approximation ist insbesondere dann schlecht, wenn die Verteilungsfunktion des Zustandes durch z.B. nichtlineare Transformation markante Multimodalität aufweist.

2.4.2 Unscented Transformation

Die Verwendung der UT stellt die wesentliche Innovation des UKF gegenüber dem KF dar. Sie ermöglicht die Approximation einer nichtlinearen Transformation einer GZV, ohne eine explizite analytische Linearisierung der Abbildungsvorschrift durch Berechnung der Jacobi-Matrix zu fordern. Zentrales Element der UT ist die Idee, statt einer Approximation der Abbildungsvorschrift, eine Approximation der GZV durch Bildung einer repräsentativen Stichprobe zu berechnen, um auf diese die vollständige Nichtlinearität anzuwenden.

Beispielhaft wird die Transformation einer GZV $\mathbf{X} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{P})$ durch eine nichtlineare Funktion $\boldsymbol{f}$ betrachtet. Ein solcher Transformationsvorgang ist exemplarisch in Abb. 2.1 visualisiert.
Zur Approximation einer d_x-dimensionalen Zufallsvariable $\mathbf{X}$ mit Erwartungswert $\boldsymbol{x} \in \mathbb{R}^{d_x}$ und Kovarianz $\boldsymbol{P} \in \mathbb{R}^{d_x \times d_x}$ werden $2d_x + 1$ gewichtete Auswertungen $\boldsymbol{\chi}_i$ der Verteilung – die sogenannten Sigma-Punkte – nach folgendem Schema gewählt

$$\begin{aligned} \boldsymbol{\chi}_0 &= \boldsymbol{x}, & W_0 &= \kappa/(d_x + \kappa), \\ \boldsymbol{\chi}_i &= \boldsymbol{x} + \left(\sqrt{(d_x + \kappa)\boldsymbol{P}}\right)_i, & W_i &= 1/2(d_x + \kappa), \\ \boldsymbol{\chi}_{i+d_x} &= \boldsymbol{x} - \left(\sqrt{(d_x + \kappa)\boldsymbol{P}}\right)_i, & W_{i+d_x} &= 1/2(d_x + \kappa). \end{aligned} \tag{2.13}$$

Dabei gilt $i = 1, \ldots, d_x$, $\kappa \in \mathbb{R}$ ist ein Skalierungsfaktor zur Wahl der Streuung der Sigma-Punkte, $\left(\sqrt{(d_x + \kappa)\boldsymbol{P}}\right)_i$ stellt die i-te Zeile der Cholesky-Zerlegung der Matrix $(d_x + \kappa)\boldsymbol{P}$ dar und W_i ist das Gewicht des i-ten Sigma-Punktes $\boldsymbol{\chi_i}$.

Die Sigma-Punkte werden in der Sigma-Punkt-Matrix $\boldsymbol{\chi}$ zusammengefasst. Der vektorwertige Sigma-Punkt $\boldsymbol{\chi}_i$ bildet dabei die i-te Spalte der Matrix

$$\boldsymbol{\chi} = \begin{bmatrix} \boldsymbol{\chi}_0 & \cdots & \boldsymbol{\chi}_i & \cdots & \boldsymbol{\chi}_{2d_x} \end{bmatrix}. \tag{2.14}$$

Die „Unscented Transformation“ der Zufallsvariable $\mathbf{X}$ mit der nichtlinearen Abbildung $\boldsymbol{f}$ erfolgt dann durch die folgenden Schritte:

1. Bestimmung der Sigma-Punkt-Matrix $\boldsymbol{\chi}$ nach Gl. (2.13) und (2.14).
2. Propagation der einzelnen Sigma-Punkte durch die Abbildung $\boldsymbol{f}$
$$\boldsymbol{\gamma}_i = \boldsymbol{f}\left(\boldsymbol{\chi}_i\right). \tag{2.15}$$
3. Extraktion der transformierten GZV $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{y}, \boldsymbol{P}_{\mathrm{y}})$ aus den transformierten Sigma-Punkten $\boldsymbol{\gamma}_i$
$$\boldsymbol{y} = \sum_{i=0}^{2d_x} W_i \boldsymbol{\gamma}_i, \tag{2.16}$$
$$\boldsymbol{P}_{\mathrm{y}} = \sum_{i=0}^{2d_x} W_i \left(\boldsymbol{\gamma}_i - \boldsymbol{y}\right)\left(\boldsymbol{\gamma}_i - \boldsymbol{y}\right)^T. \tag{2.17}$$

Die UT ermöglicht eine fehlerfreie Approximation der Abbildungsvorschrift $\boldsymbol{f}$ bis zur dritten Ordnung für beliebige Nichtlinearitäten. Insbesondere erlaubt die UT im Gegensatz zur Linearisierung der Abbildung als Taylor-Polynom auch eine Approximation nicht differenzierbarer Funktionen. Detaillierte Ausführungen zur Approximationsgüte, sowie zu mathematischen Hintergründen und weiteren Varianten der UT finden sich in [12, 13].

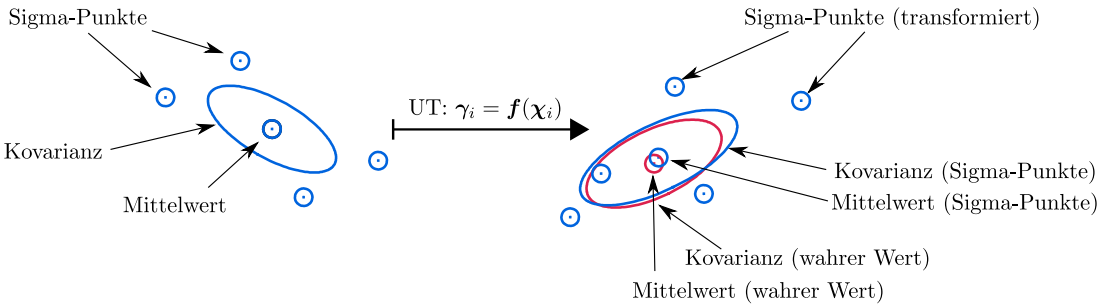

Abbildung 2.1: Schematische Darstellung einer Unscented Transformation für eine nichtlineare Abbildung $\boldsymbol{f}$ nach [8].

2.4.3 UKF-Algorithmus

Das algorithmische Vorgehen des UKF ergibt sich aus dem des KF (vgl. Abschnitt 2.3.2), indem der Zustandsübergang $\boldsymbol{f}$, sowie die Messfunktion $\boldsymbol{h}$ in Form ihrer korrespondierenden UT (vgl. Abschnitt 2.4.2) formuliert werden.

Nachfolgend wird daher der Algorithmus der UKF auf Basis der Ausführungen zum KF-Algorithmus aus Abschnitt 2.3.2 erläutert.

Erweiterung der Zustandsvariablen

Zur Formulierung des allgemeinen UKF-Algorithmus wird der erweiterte L-Dimensionale Zustand $\mathbf{X}_t^{\mathrm{a}} \sim \mathcal{N}(\boldsymbol{x}_t^{\mathrm{a}}, \boldsymbol{P}_t^{\mathrm{a}})$ definiert. Dieser setzt sich aus der d_x-dimensionalen Zustandsschätzung $\mathbf{X}_t$, dem Prozessrauschen $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_t)$, sowie dem Messrauschen $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ für die Zeitschritte $t \in \mathbb{N}_0$ wie folgt zusammen

$$\boldsymbol{x}_t^{\mathrm{a}} = E\left[\mathbf{X}_t^{\mathrm{a}}\right] = \left[\boldsymbol{x}_t^T, \boldsymbol{w}_t^T, \boldsymbol{v}_t^T\right]^T \qquad = \left[\boldsymbol{x}_t^T, \mathbf{0}, \mathbf{0}\right]^T, \tag{2.18}$$

$$\boldsymbol{P}_t^{\mathrm{a}} = \mathrm{Cov}\left[\mathbf{X}_t^{\mathrm{a}}\right] = E\left[\left(\mathbf{X}_t^{\mathrm{a}} - \boldsymbol{x}_t^{\mathrm{a}}\right)\left(\mathbf{X}_t^{\mathrm{a}} - \boldsymbol{x}_t^{\mathrm{a}}\right)^T\right] \qquad = \begin{bmatrix} \boldsymbol{P}_t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_t \end{bmatrix}. \tag{2.19}$$

Für den erweiterten Zustand $\mathbf{X}_t^{\mathrm{a}}$ werden die Sigma-Punkte nach dem in Abschnitt 2.4.2 beschriebenen Schema berechnet. Entsprechend der Aufteilung des erweiterten Zustandes aus Gl. (2.18) erfolgt die Benennung der Sigma-Punkt-Komponenten nach Gl. (2.21) als

$$\begin{aligned} \boldsymbol{\chi}_t^{\mathrm{a}} &= \left[\boldsymbol{x}_t^{\mathrm{a}}, \quad \boldsymbol{x}_t^{\mathrm{a}} \pm \sqrt{(L+\kappa)\,\boldsymbol{P}_t^{\mathrm{a}}}\right] \\ &= \left[\left(\boldsymbol{\chi}_t^{\mathrm{x}}\right)^T, \left(\boldsymbol{\chi}_t^{\mathrm{w}}\right)^T, \left(\boldsymbol{\chi}_t^{\mathrm{v}}\right)^T\right]^T. \end{aligned} \tag{2.20}$$

Prädiktion

Die Prädiktion des nächsten Zustandes erfolgt durch Transformation der Zustandsschätzung $\mathbf{X}_{t-1}$ mit der Systemdynamik $\boldsymbol{f}_t$ aus Gl. (2.2). Die Transformation wird dabei als UT nach dem Vorgehen aus Abschnitt 2.4.2 durchgeführt. Der prädizierte Zustand $\bar{\mathbf{X}}_t \sim \mathcal{N}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t)$ berechnet sich damit zu

$$\bar{\boldsymbol{\chi}}_t^{\mathrm{x}} = \boldsymbol{f}_t\left(\boldsymbol{\chi}_{t-1}^{\mathrm{x}}, \boldsymbol{u}_{t-1}, \boldsymbol{\chi}_{t-1}^{\mathrm{w}}\right), \tag{2.21}$$

$$\bar{\boldsymbol{x}}_t = \sum_{i=0}^{2L} W_i^{\mathrm{m}} \bar{\boldsymbol{\chi}}_{i,t}^{\mathrm{x}}, \tag{2.22}$$

$$\bar{\boldsymbol{P}}_t = \sum_{i=0}^{2L} W_i^{\mathrm{c}} \left(\bar{\boldsymbol{\chi}}_{i,t}^{\mathrm{x}} - \bar{\boldsymbol{x}}_t\right)\left(\bar{\boldsymbol{\chi}}_{i,t}^{\mathrm{x}} - \bar{\boldsymbol{x}}_t\right)^T. \tag{2.23}$$

Es ist wichtig zu bemerken, dass die Zustandsübergangsfunktion $\boldsymbol{f}_t$ weiterhin nach $\mathbb{R}^{d_x}$ abbildet, auch wenn die Zustandsvariable zur Anwendung der UT erweitert wurde. Die Gewichte W_i^{m} bzw. W_i^{c} sind der Berechnungsvorschrift der UT aus Abschnitt 2.4.2 zu entnehmen.

Korrektur

Aus dem a-priori-Zustand $\bar{\mathbf{X}}_t \sim \mathcal{N}\left(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t\right)$ wird zunächst die korrespondierende Messung $\bar{\mathbf{Z}}_t \sim \mathcal{N}\left(\bar{\boldsymbol{z}}_t, \bar{\boldsymbol{P}}_{\mathrm{z},t}\right)$ durch Anwendung der Messfunktion $\boldsymbol{h}_t$ prädiziert. Auch diese Abbildungsvorschrift wird als UT

$$\bar{\boldsymbol{\zeta}}_t = \boldsymbol{h}_t\left(\boldsymbol{\chi}_{t-1}^{\mathrm{x}}, \boldsymbol{\chi}_{t-1}^{\mathrm{v}}\right), \tag{2.24}$$

$$\bar{\boldsymbol{z}}_t = \sum_{i=0}^{2L} W_i^m \bar{\boldsymbol{\zeta}}_{i,t}, \tag{2.25}$$

$$\bar{\boldsymbol{P}}_{\mathrm{z,t}} = \sum_{i=0}^{2L} W_i^{\mathrm{c}} \left(\bar{\boldsymbol{\zeta}}_{i,t} - \bar{\boldsymbol{z}}_t\right)\left(\bar{\boldsymbol{\zeta}}_{i,t} - \bar{\boldsymbol{z}}_t\right)^T. \tag{2.26}$$

formuliert.

Wird im Vergleich das KF betrachtet entspricht die Kovarianz der prädizierten Messung $\bar{\boldsymbol{P}}_{\mathrm{z},t}$ der Residualkovarianz $\boldsymbol{S}_t$. Zur Bestimmung der Kalman-Matrix wird zunächst die Kreuzkorellationsmatrix

$$\bar{\boldsymbol{P}}_{\mathrm{xz},t} = \sum_{i=0}^{2L} W_i^{\mathrm{c}} \left(\bar{\boldsymbol{\chi}}_{i,k} - \bar{\boldsymbol{x}}_t\right)\left(\bar{\boldsymbol{\zeta}}_{i,k} - \bar{\boldsymbol{z}}_t\right)^T \tag{2.27}$$

zwischen prädiziertem Zustand $\bar{\mathbf{X}}_t$ und prädizierter Messung $\bar{\mathbf{Z}}_t$ bestimmt. In Analogie zum KF entspricht die Kreuzkorrelationsmatrix $\bar{\boldsymbol{P}}_{\mathrm{xz},t}$ dem Ausdruck $\bar{\boldsymbol{P}}_t \boldsymbol{H}_t$.

Damit folgt schließlich für die Kalman-Matrix

$$\boldsymbol{K}_t = \bar{\boldsymbol{P}}_{\mathrm{xz},t} \bar{\boldsymbol{P}}_{\mathrm{z},t}^{-1}. \tag{2.28}$$

Schlussendlich folgt die Berechnung des korrigierten a-posteriori Zustandes durch Integration der Messung $\boldsymbol{Z}_t$ analog zum Vorgehen des KF (vgl. Gl. (2.11) und (2.12)) mit

$$\boldsymbol{x}_t = \bar{\boldsymbol{x}}_t + \boldsymbol{K}_t\left(\boldsymbol{z}_t - \bar{\boldsymbol{z}}_t\right), \tag{2.29}$$

$$\boldsymbol{P}_t = \bar{\boldsymbol{P}}_t - \boldsymbol{K}_t \bar{\boldsymbol{P}}_{\mathrm{z},t} \boldsymbol{K}_t^T. \tag{2.30}$$

Die vorangegangenen Erläuterungen zu theoretischen Grundlagen des UKF basieren auf [5, 8, 13, 14] und können dort in eingehenderem Detail nachvollzogen werden.

2.5 Partikel-Filter

Das PF – auch Sequenzielle Monte-Carlo-Methode genannt – ist ein nichtdeterministischer Algorithmus zur Schätzung von Zuständen nahezu beliebiger Markov-Ketten.
Kernelement des PF ist die Approximation der Zustandszufallsvariablen $\mathbf{X}$. Diese erfolgt im Kontrast zum UKF durch die Wahl einer umfangreichen und insbesondere stochastischen Stichprobe (Sample) – den sogenannten Partikeln. Im Gegensatz zum UKF verzichtet das PF auf die Approximation der a-posteriori-Wahrscheinlichkeit durch eine analytische Verteilungsfunktion. Stattdessen wird die a-posteriori-Verteilung durch das vollständig propagierte Sample repräsentiert.
Die einzelnen Partikel können als Hypothesen möglicher Zustände des Systems betrachtet werden. Die Anzahl der Partikel zur Repräsentation der Wahrscheinlichkeitsverteilung kann in der Theorie beliebig groß gewählt werden. Auf diese Weise ist die Approximationsgüte der Wahrscheinlichkeitsverteilung durch die Wahl des Stichprobenumfangs skalierbar. Für ausreichend umfangreiches Sampling kann das PF daher auf das in Abschnitt 2.2 beschriebene Filterproblem angewendet werden, ohne weitere Einschränkungen des Problems fordern zu müssen. Für hinreichend große Partikelanzahlen ermöglicht das PF die Zustandsschätzung von Systemen mit nahezu beliebiger Nichtlinearität in Zustandsübergangsfunktion $\boldsymbol{f}_t$ und Messfunktion $\boldsymbol{h}_t$. Insbesondere ermöglicht das Partikel-Filter auch die Schätzung von Zuständen mit multimodaler Verteilung.

2.5.1 Einschränkung der allgemeinen Problemformulierung

Das Partikel-Filter kann auf das in Abschnitt 2.2 spezifizierte Filterproblem ohne weitere Einschränkung der Problemformulierung angewendet werden. Insbesondere sind für die Rauschkomponenten $\mathbf{W}_t$ und $\mathbf{V}_t$ beliebige Wahrscheinlichkeitsdichtefunktionen zulässig.

2.5.2 PF-Algorithmus

Der Algorithmus des Partikel-Filters lässt sich analog zum Aufbau des UKF-Algorithmus in Prädiktion und Korrektur unterteilen. Wie einleitend beschrieben, wird aus der Wahrscheinlichkeitsdichte $p(\boldsymbol{x}_t)$ des Systemzustandes $\boldsymbol{x}_t$ zum Zeitpunkt t eine Vielzahl von Partikeln als Stichprobe des Zustandsraumes zufällig gezogen. Dabei sei die Anzahl der Partikel mit $M \in \mathbb{N}$ bezeichnet. Diese wird in der Regel hoch gewählt (meist $M \geq 100$), damit ein ausreichend dichtes Sampling

der Zustandsdichtefunktion möglich ist. Die genaue Wahl von M hängt von der konkreten Problemstellung ab und ist unter Umständen durch Laufzeitanforderungen nach oben beschränkt.
Die Partikel $\boldsymbol{x}_t^{[m]}$ mit Laufindex $m \in \{1, \ldots, M\}$ der Stichprobe, die zum Zeitpunkt t aus der Zustandsdichtefunktion gezogen wurde, werden – ähnlich den Sigma-Punkten des UKF – zu einer Matrix $\boldsymbol{\chi}_t = \begin{bmatrix} \boldsymbol{x}_t^{[1]} & \cdots & \boldsymbol{x}_t^{[m]} & \cdots & \boldsymbol{x}_t^{[M]} \end{bmatrix}$ zusammengefasst. Basierend auf diesen Partikeln wird der PF-Algorithmus wie im Folgenden beschrieben konstruiert.

Initialisierung

Die Verteilungsfunktion des initialen Systemzustandes ist entsprechend der Problemformulierung in Abschnitt 2.2 durch $p(\boldsymbol{x}_0)$ gegeben. In einem initialen Schritt werden zunächst M zufällige Stichproben entsprechend der Wahrscheinlichkeitsdichtefunktion $p(\boldsymbol{x}_0)$ aus der Anfangsverteilung gezogen.

Für $m = 1, \ldots, M$ wähle

$$\boldsymbol{x}_0^{[m]} \sim p(\boldsymbol{x}_0), \tag{2.31}$$

$$\boldsymbol{\chi}_0 = \begin{bmatrix} \boldsymbol{x}_0^{[1]} & \cdots & \boldsymbol{x}_0^{[M]} \end{bmatrix}. \tag{2.32}$$

Prädiktion

Die Prädiktion des nächsten Zustandes erfolgt durch Anwendung der nichtdeterministischen Systemdynamik aus Gl. (2.2) auf jedes einzelne Partikel. Auf diese Weise wird für jedes Partikel $\boldsymbol{x}_{t-1}^{[m]}$ aus $\boldsymbol{\chi}_{t-1}$ ein Partikel $\bar{\boldsymbol{x}}_t^{[m]}$ in $\bar{\boldsymbol{\chi}}_t$ als Stichprobe der Verteilung $p(\bar{\boldsymbol{x}}_t | \boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1})$ zufällig gezogen.

Für $m = 1, \ldots, M$ wähle

$$\bar{\boldsymbol{x}}_t^{[m]} \sim p(\bar{\boldsymbol{x}}_t | \boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1}). \tag{2.33}$$

Für viele Anwendungen bietet sich die Formulierung dieses Schrittes als

$$\bar{\boldsymbol{x}}_t^{[m]} = \boldsymbol{f}_t \left(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}, \boldsymbol{w}_{t-1}^{[m]} \right) \text{ mit } \boldsymbol{w}_{t-1}^{[m]} \sim p(\boldsymbol{w}_{t-1}) \tag{2.34}$$

an. Dabei wird die Rauschkomponente $\boldsymbol{w}_{t-1}$ entsprechend ihrer als bekannt vorausgesetzten Verteilungsfunktion $p(\boldsymbol{w}_{t-1}^{[m]})$ für jedes Partikel zufällig gewählt. Die Menge prädizierter Partikel bildet die Matrix $\bar{\boldsymbol{\chi}}_t$ analog dem Vorgehen aus Gl. (2.32). Abbildung 2.2 zeigt beispielhaft eine nichtlineare Transformation einer Partikel-Population.

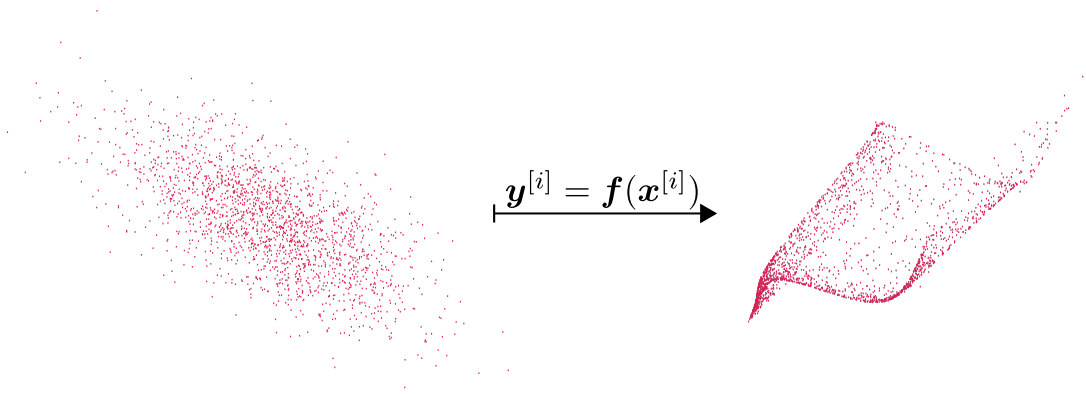

Abbildung 2.2: Schematische Darstellung einer nichtlinearen Transformation durch Anwendung einer Abbildung $\boldsymbol{f}$ auf jedes Partikel $\boldsymbol{x}^{[i]}$ nach [8].

Korrektur

Analog zum Algorithmus des KF und UKF wird in diesem Schritt die Information der Messung in die Zustandsschätzung einbezogen, um die vorangegangene Prädiktion zu korrigieren. Die Korrektur des prädizierten Zustandes erfolgt in zwei Schritten:

Bewertung der Partikel. Den einzelnen Partikeln $\bar{\boldsymbol{x}}_t^{[m]}$ aus $\bar{\boldsymbol{\chi}}_t$ wird zunächst anhand der Messung ein sogenanntes Importance-Weight $\mathrm{w}_t^{[m]}$ zugeordnet.

Für $m = 1, \ldots, M$ erzeuge

$$\mathrm{w}_t^{[m]} = p(\boldsymbol{z}_t | \bar{\boldsymbol{x}}_t^{[m]}). \tag{2.35}$$

Erzeuge Gewichtsvektor

$$\mathbf{w}_t = \begin{bmatrix} \mathrm{w}_t^{[1]} & \cdots & \mathrm{w}_t^{[M]} \end{bmatrix}^T. \tag{2.36}$$

Das Importance-Weight $\mathrm{w}_t^{[m]}$ entspricht der Wahrscheinlichkeit der Messung $\boldsymbol{z}_t$ im durch das m-te Partikel repräsentierten Zustand $\bar{\boldsymbol{x}}_t^{[m]}$.

Importance Sampling. Im zweiten Schritt wird aus der prädizierten Stichprobe $\bar{\boldsymbol{\chi}}_t$ das korrigierte Sample $\boldsymbol{\chi}_t$ gebildet.

Für $m = 1, \ldots, M$:

Ziehe zufälliges $i \in \{1, \ldots, M\}$ mit Wahrscheinlichkeit $\propto \mathrm{w}_t^{[i]}$. (2.37)

Füge $\bar{\boldsymbol{x}}_t^{[i]}$ der Matrix $\boldsymbol{\chi}_t$ als m-te Spalte $\boldsymbol{x}_t^{[m]}$ hinzu. (2.38)

Dazu wird aus $\bar{\boldsymbol{\chi}}_t$ ein Satz M zufällig gezogener Partikel gewählt (vgl. Gl. (2.37)), wobei die Auswahlwahrscheinlichkeit proportional zum korrespondierenden Importance-Weight $w_t^{[t]}$ ausfällt. Die Wahl der einzelnen Stichproben erfolgt unabhängig voneinander, also als Ziehung mit Zurücklegen. Es ist damit ausdrücklich möglich, dass ein Partikel aus $\bar{\boldsymbol{\chi}}_t$ mehrfach in die korrigierte Stichprobe $\boldsymbol{\chi}_t$ aufgenommen wird.

Die vorangegangenen Erläuterungen zu theoretischen Grundlagen des PF basieren auf [5, 9] und werden dort in eingehenderem Detail erläutert.

Kapitel 3

Problemstellung und Rahmenbedingungen

Die in den Abschnitten 2.4 und 2.5 beschriebenen Algorithmen werden im Rahmen dieser Arbeit auf die Problemstellung der „Selbstlokalisierung auf einem Feld mit dem humanoiden NAO-Robotiksystem“ angewendet. Zu diesem Zweck wird in Abschnitt 3.1 zunächst das Problem genauer formuliert. Darauf aufbauend wird in Abschnitt 3.2 diese allgemeine Problemformulierung für das Anwendungsbeispiel „Selbstlokalisierung im SPL-Kontext“ spezifiziert.

3.1 Allgemeine Problemformulierung

Das Problem der „Selbstlokalisierung auf dem Fußballfeld mit dem humanoiden NAO-Robotiksystem“ sei nachfolgen in seinen zentralen Aspekten beschrieben:

Bekannte Umwelt Die Selbstlokalisierung soll auf einem Fußballfeld stattfinden. Die wesentlichen Merkmale dieses Feldes sind im Voraus definiert, sodass ein internes Modell der Umwelt dem Lokalisierungsalgorithmus zur Verfügung steht.

Sensoren Das NAO-Robotiksystem verfügt über eine Reihe von Sensoren, die verwendet werden können, um Informationen über die Umwelt und den aktuellen Zustand des Systems zu gewinnen. Zu diesen Sensoren gehören unter anderem zwei Kameras im Kopf des Roboters, ein dreiachsiges Gyroskop zur Messung von Winkelgeschwindigkeiten, sowie ein dreiachsiger Beschleunigungssensor. Eine genauere Beschreibung der Sensoren erfolgt in Abschnitt 3.1.1.

Der Roboter als Mehrkörpersystem Der NAO ist ein humanoider Roboter mit 25 Freiheitsgraden. Als solcher ist er als Mehrkörpersystem zu betrachten und unterliegt daher einer komplexen Dynamik. Das System bewegt sich durch Generierung eines zweibeinigen Ganges durch den Raum. Diese Art der Fortbewegung erfordert eine dynamische Verlagerung des Schwerpunktes zwischen den beiden Füßen, wodurch insbesondere der Oberkörper eine oszillierende Bewegung ausführt.

Bewegung durch den Raum Das Robotiksystem wird auf dem Fußballfeld eingesetzt und bewegt sich ab diesem Zeitpunkt selbständig durch die Umwelt.

Zeitliche Persistenz von Wissen Die Selbstlokalisierung hat nicht den Anspruch, Position und Orientierung aus einer einzigen Situation zu bestimmen. Vielmehr können Informationen im Zeitverlauf akkumuliert werden, um daraus langfristig eine Schätzung der Lokalisierung zu berechnen.

Unter diesen Bedingungen umfasst das Filterproblem die Schätzung von Position und Orientierung des Roboters bezüglich eines inertialen Koordinatensystems.

3.1.1 Das NAO-Robotiksystem

Der NAO ist eine humanoide Robotik-Plattform der Firma „SoftBank Robotics“. Die Eigenschaften der Hardware bestimmen die Beschaffenheit des Problems maßgeblich. Vor diesem Hintergrund seien im Folgenden die wesentlichen, für die Problemformulierung relevanten Merkmale des Robotik-Systems zusammengetragen. Die nachfolgenden Informationen sind der Dokumentation des Herstellers [15] entnommen und können dort in ausführlicher Fassung eingesehen werden.

Körperbau und Freiheitsgrade Der humanoide NAO-Roboter ist 573mm groß und wiegt 5,18kg. Das System besitzt 25 Freiheitsgrade, die mit Motoren ausgestattet sind, um die einzelnen Körperteile zu bewegen. Eine schematische Übersicht des Systemaufbaus ist in Abb. 3.1 dargestellt.

Gelenkwinkelsensoren Zur Bestimmung der Gelenkwinkel, sind die Aktuatoren mit magnetischen Drehgebern ausgestattet. Diese Sensoren erlauben die Bestimmung der jeweils vorherrschenden Gelenkwinkel mit einem absoluten Fehler von $0.1\,^{\circ}$.

Kameras Zur Wahrnehmung der Umgebung ist das System mit zwei 2D-Kameras ausgestattet. Die Kameras bieten eine Auflösung von 640×480 Pixeln bei einer Bildrate von 30 fps. Die Kameras sind dabei wie in Abb. 3.1 dargestellt im Kopf des Reboters verbaut. Insbesondere überschneiden sich die beiden Sichtbereiche der Kameras im Bereich relevanter Entfernungen nur wenig, sodass die Kameradaten keine unmittelbare Entfernungsinformation zu den einzelnen Pixeln liefern können. Auch sei darauf hingewiesen, dass der horizontale Öffnungswinkel mit $60.97\,^\circ$ nur eine sehr eingeschränkte Wahrnehmung der Umgebung erlaubt.

Trägheitsreferenzsystem Im Torso des NAOs ist ein sogenanntes Trägheitsreferenzsystem verbaut. Dieses besteht aus einem Gyroskop zur Messung der Winkelgeschwindigkeiten bis $8.72\,\mathrm{rad\,s^{-1}}$ um alle drei räumlichen Achsen. Überdies ist die Einheit mit einem dreiachsigen Accelerometer bestückt, welches die Aufnahme von Beschleunigungen bis $19.6\,\mathrm{m\,s^{-2}}$ erlaubt. Das Trägheitsreferenzsystem operiert mit einer Abtastrate von 100 Hz.

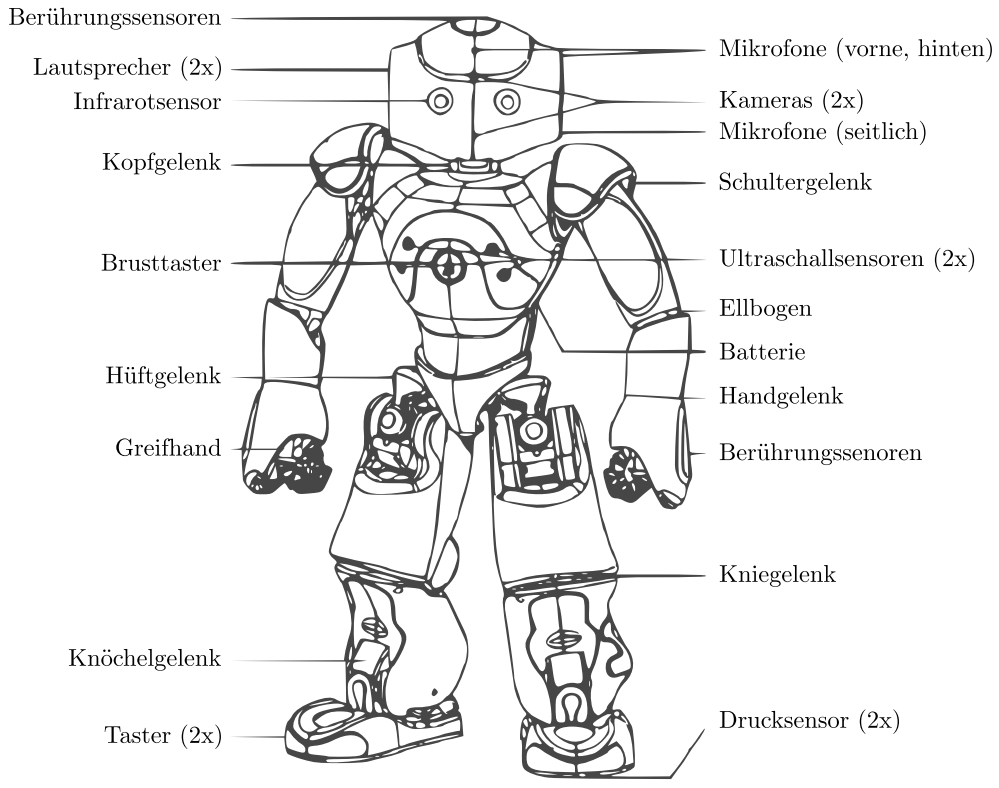

Abbildung 3.1: Hardware-Übersicht des NAO-Robotiksystems nach [15].

3.2 Anwendungsbeispiel – Selbstlokalisierung im SPL-Kontext

In Abschnitt 3.1 ist eine Formulierung der allgemeinen Problematik „Selbstlokalisierung auf dem Feld mit dem humanoiden NAO-Robotiksystem“ erfolgt. Ein Anwendungsbeispiel, in dem eben dieses formulierte Problem gelöst werden muss, ist die „RoboCup Standard Platform League“ (SPL). Die SPL ist ein Wettbewerb, in dem NAO-Roboter in Gruppen von fünf Spielern zum Fußballspiel gegeneinander antreten. Dieser Wettbewerb definiert klare Regeln für Umwelt und Spielverlauf und stellt so ein wohldefiniertes Anwendungsbeispiel für das im Rahmen dieser Arbeit behandelte Lokalisierungsproblem dar [16]. Die Selbstlokalisierung eines NAO-Roboters auf einem Fußballfeld nach dem Regelwerk der SPL soll daher als Anwendungsbeispiel dienen. Ausgangspunkt ist dazu das Software-Framework des RoboCup SPL Teams HULKs. Die im späteren Verlauf dieser Arbeit vorgestellten Lokalisierungsstrategien bauen auf diesem Framework auf und sind daher insbesondere auf die Eingangsgrößen beschränkt, die im Rahmen dieser Software zur Verfügung stehen.

In den nachfolgenden Unterabschnitten werden die einzelnen Teilpunkte der Problemformulierung aus Abschnitt 3.1 für das hier vorliegende Anwendungsbeispiel genauer spezifiziert. Dazu werden in Abschnitt 3.2.1 zunächst die wesentlichen Eigenschaften der Umgebung – dem Fußballfeld – erläutert. Nachdem in Abschnitt 3.2.2 die hier verwendeten Koordinatensysteme definiert werden, werden die Rahmenbedingungen beleuchtet, die sich aus dem vorhandenen Software-Framework der HULKs (Abschnitt 3.2.3) ergeben. Darauf aufbauend werden die Eingangsgrößen erläutert, die den hier entwickelten Filtermethoden zum Zweck der Selbstlokalisierung zur Verfügung stehen (Abschnitt 3.2.4). In Abschnitt 3.2.5 wird beschrieben, welche Informationen ein Lokalisierungsalgorithmus aus dem Regelwerk des SPL extrahieren kann. Abschließend werden zentrale, sich aus den Rahmenbedingungen ergebende Herausforderungen abgeleitet, denen eine Lokalisierungsstrategie im Kontext des Anwendungsbeispiels ausgesetzt ist.

3.2.1 Die Umgebung – Das Fußballfeld

Die Umgebung, in der sich der Roboter bewegt, spielt eine fundamentale Rolle für seine Fähigkeit den eigenen Standort bestimmen zu können. Im Rahmen der SPL bewegt sich der Roboter auf weitestgehend bekanntem Terrain. Durch das Regelwerk werden viele Eigenschaften dieses Terrains – dem Spielfeld – eindeutig definiert, sodass dieses Wissen in das interne Modell der Umwelt integriert werden kann. Dieses Modell beinhaltet die Position aller relevanten Feldmarkierungen in einem globalen, inertialen Koordinatensystem.

Das Regelwerk der Saison 2017 [16] schreibt vor, dass das Fußballspiel auf einem grünen Kunstrasenfeld stattfindet, in welches Feldmarkierungen als weiße Rasenelemente eingelassen sind. Die genauen Maße sind in Abb. 3.2 dargestellt und können Tab. 3.1 entnommen werden.

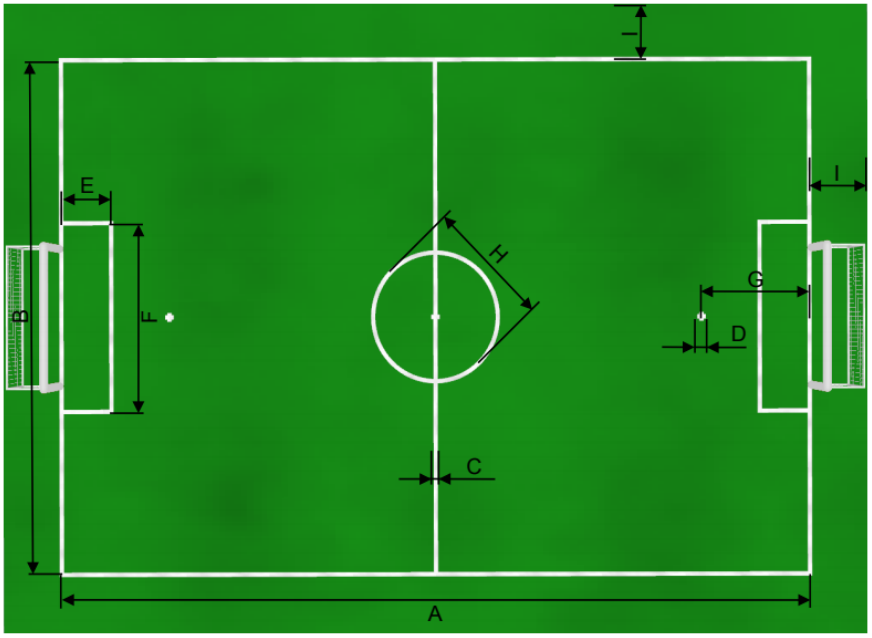

Abbildung 3.2: Schematischer Aufbau eines Fußballfeldes nach dem Regelwerk der RobCup SPL [16].

Tabelle 3.1: Maße eines Fußballfeldes nach dem Regelwerk der RobCup SPL [16].

ID	Bezeichnung	Länge in [mm]
A	Feldlänge	9000
B	Feldbreite	6000
C	Linienbreite	50
D	Strafstoßkreuzgröße	100
E	Strafraumlänge	600
F	Strafraumbreite	2200
G	Strafstoßkreuzdistanz	1300
H	Mittelkreisdurchmesser	1500
I	Seitenstreifenbreite	700

3.2.2 Koordinatensysteme

Im Kontext der weiteren Ausführungen finden die folgenden drei Koordinatensysteme Anwendung:

Torsokoordinatensystem

SoftBank Robotics definiert in der technischen Dokumentation des NAOs ein körperfestes Koordinatensystem [15]. Dieses Koordinatensystem wird im Folgenden als `Torsokoordinatensystem` $\mathrm{T} : \{\mathrm{O}^{\mathrm{t}}, \boldsymbol{e}_x^t, \boldsymbol{e}_y^t, \boldsymbol{e}_z^t\}$ bezeichnet und dient der Beschreibung von relativen Positionen im dreidimensionalen Raum. Die Lage des `Torsokoordinatensystems` ist in Abbildung Abb. 3.3 dargestellt.
Angaben in `Torsokoordinaten` werden nachfolgend mit einem tiefgestellten T indiziert.

2D-Roboterkoordinatensystem

Zur Definition des zweidimensionalen `Roboterkoordinatensystems` wird angenommen, dass der Erdbeschleunigungsvektor orthogonal auf der Ebene des Untergrunds steht und dass der Standfuß des Roboters zum Zeitpunkt der Gültigkeit dieses Koordinatensystems Kontakt zum Untergrund hat.
Unter diesen Annahmen wird das `Roboterkoordinatensystem` $\mathrm{R}' : \{\mathrm{O}', \boldsymbol{e}_{x'}, \boldsymbol{e}_{y'}\}$ aus dem `Torsokoordinatensystem` $\mathrm{T} : \{\mathrm{O}^{\mathrm{t}}, \boldsymbol{e}_x^t, \boldsymbol{e}_y^t, \boldsymbol{e}_z^t\}$ durch Projektion auf die Untergrundebene abgeleitet. Die Lage des `Roboterkoordinatensystems` ist in Abbildung Abb. 3.3 dargestellt. Das Koordinatensystem wird im Folgenden verwendet, um Positionsangaben relativ zum Roboter in einem zweidimensionalen Weltmodell anzugeben.
Größen in `Roboterkoordinaten` werden nachfolgend mit einem tiefgestellten R' indiziert.

2D-Feldkoordinatensystem

Der Ursprung des zweidimensionalen `Feldkoordinatensystems` O wird im Feldmittelpunkt definiert. Die Lage dieses Koordinatensystems ist in Abb. 3.4 dargestellt. Die x-Achse ist dabei zum gegnerischen Tor ausgerichtet. Das `Feldkoordinatensystem` $\mathrm{F} : \{\mathrm{O}, \boldsymbol{e}_x, \boldsymbol{e}_y\}$ dient als inertiale Referenz, in dem auch das Berechnungsergebnis des Selbstlokalisierungsalgorithmus angegeben wird.
Größen in `Feldkoordinaten` werden nachfolgend mit einem tiefgestellten F indiziert.

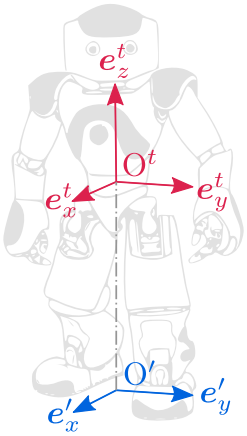

Abbildung 3.3: Lage des `Torsokoordinatensystems` T : $\{O^t, \boldsymbol{e}_x^t, \boldsymbol{e}_y^t, \boldsymbol{e}_z^t\}$ sowie des `Roboterkoordinatensystems` R′ : $\{O', \boldsymbol{e}_{x'}, \boldsymbol{e}_{y'}\}$ nach [15].

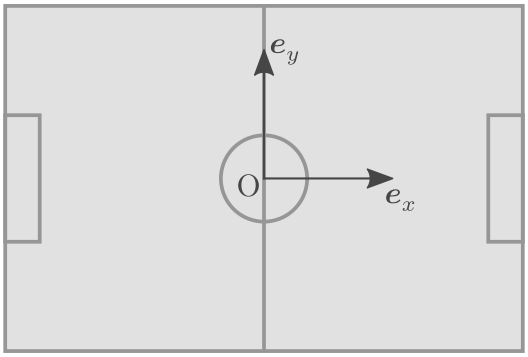

Abbildung 3.4: Lage des `Feldkoordinatensystems` F : $\{O, \boldsymbol{e}_x, \boldsymbol{e}_y\}$.

Posen und Koordinatentransformationen

Die relative Lage zweier Koordinatensysteme zueinander ist durch einen Verschiebungsvektor $\boldsymbol{r}_{\mathrm{OO'},\mathrm{K}}$ sowie eine Rotationsmatrix $\boldsymbol{S}_{KK'}$ eindeutig definiert. Unter Angabe dieser beiden Komponenten kann eine Koordinatentransformation zwischen den Koordinatensystemen $\mathrm{K} : \{\mathrm{O}, \boldsymbol{e}_\mathrm{x}, \boldsymbol{e}_\mathrm{y}, \boldsymbol{e}_\mathrm{z}\}$ und $\mathrm{K}' : \{\mathrm{O}', \boldsymbol{e}_{x'}, \boldsymbol{e}_{y'}, \boldsymbol{e}_{z'}\}$ wie folgt durchgeführt werden

$$\boldsymbol{r}_{\mathrm{OP},\mathrm{K}} = \boldsymbol{r}_{\mathrm{OO'},\mathrm{K}} + \boldsymbol{S}_{KK'}\boldsymbol{r}_{\mathrm{O'P},\mathrm{K'}}. \tag{3.1}$$

Das Tupel $\boldsymbol{\Theta}_{\mathrm{KK'},\mathrm{K}} := (\boldsymbol{r}_{\mathrm{OO'},\mathrm{K}}, \boldsymbol{S}_{KK'})$ aus relativer Verschiebung und Rotation zweier Koordinatensysteme zueinander wird nachfolgend als Pose von K′ in K bezeichnet. Diese Definition der `Pose` gelte auch für zweidimensionale Koordinatensysteme. In diesem Fall wird die Orientierung zweier Koordinatensysteme zueinander allerdings vollständig durch die Angabe eines Winkels $\alpha_{\mathrm{KK'}} \in (-\pi, \pi]$ beschrieben. Eine solche zweidimensionale Pose ist beispielhaft Abb. 3.5 dargestellt und wird im Folgenden als `2D-Pose` bezeichnet.

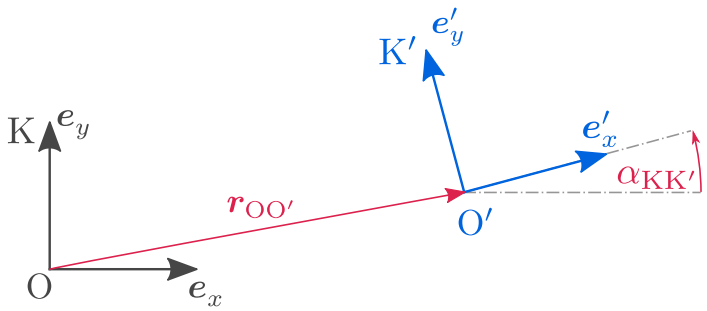

Abbildung 3.5: Definition einer `2D-Pose` $\boldsymbol{\Theta}_{\mathrm{KK'}} := (\boldsymbol{r}_{\mathrm{OO'}}, \alpha_{\mathrm{KK'}})$.

3.2.3 Das Software-Framework der HULKs

Das RoboCup SPL Team HULKs ist ein studentisches Projekt, welches an der Technischen Universität Hamburg (TUHH) angesiedelt ist. Ziel des Projekts ist die Entwicklung einer Software für das NAO-Robotiksystem zur Partizipation in der SPL. Im Rahmen dieses Projekts wird eine Codebasis in C++ entwickelt, mit der das Team unter anderem beim RoboCup 2016 in Leipzig angetreten ist. Im weiteren Verlauf dieser Arbeit soll diese Software als Grundlage für die Beschreibung einer Selbstlokalisierungsanwendung dienen.

Nachfolgend werden wesentliche, für diese Arbeit relevante Konzepte der Codebasis erläutert.

Cycle Die Verarbeitung der eingehenden Informationen erfolgt in einem regelmäßigen, streng getakteten Berechnungszyklus, dem sogenannten `Cycle`. Innerhalb eines `Cycles` müssen alle Algorithmen zur Verarbeitung der ihnen zugeordneten Eingangsdaten ein Berechnungsergebnis liefern, um die Echtzeitfähigkeit des Systems zu gewährleisten. Für den Teil der Software, in dem auch die Filteralgorithmen ausgeführt werden, beträgt die Taktfrequenz 60Hz.

Vorwärtskinematik Die `Vorwärtskinematik` berechnet die dreidimensionale Position und Orientierung (`3D-Pose`) eines Endeffektors in `Torsokoordinaten` auf Basis der beteiligten Gelenkwinkel. So wird z.B. die `3D-Pose` des Fußes in `Torsokoordinaten` mit Hilfe der Gelenkwinkel des zugehörigen Beines berechnet.

WalkingEngine Die `WalkingEngine` berechnet in jedem `Cycle` die Gelenkwinkel zur Generierung des humanoiden Laufens. Sie gibt überdies bekannt, welcher der beiden Füße zurzeit als `Standfuß` betrachtet wird.

FieldInfo Die `FieldInfo` ist das interne Modell der Umwelt des Roboters. Sie beinhaltet die Position aller Feldlinien und Torpfosten (vgl. Abschnitt 3.2.1) in `Feldkoordinaten`.

3.2.4 Eingangsgrößen

Im Datenflussmodell des hier verwendeten Software-Frameworks sind einem Lokalisierungsalgorithmus eine Reihe von Prozessschritten vorgelagert, die aus den rohen Sensordaten abstrakte Messungen gewinnen.

Zur internen Kommunikation dieser Messungen werden jeweils Minimal-Repräsentationen gewählt, die nur noch die wesentlichen Informationen dieser Messung enthalten. So wird z.B. eine Linienmessung nicht durch alle ihr zugeordneten Punkte repräsentiert, sondern lediglich durch ihren Anfangs- und Endpunkt. Nachfolgend werden die Messungen beschrieben, die einem Lokalisierungsalgorithmus im Software-Framework der HULKs zur Verfügung stehen.

Bildverarbeitung

Einen wesentlichen Anteil der Eingangsgrößen stellt die Bildverarbeitung zur Verfügung. Die sogenannte `Vision-Pipeline` bezeichnet die Gesamtheit der aufeinander folgenden Bildverarbeitungsschritte. Sie extrahiert aus den Bilddaten der Kameras markante Merkmale – sogenannte `Features` – die dann in abstrahierter Form als Ausgangsgröße zur Verfügung gestellt werden. Während der detaillierte Prozess der Bildverarbeitung hier nicht thematisiert werden soll, seien

nachfolgend jene Ausgangsgrößen der `Vision-Pipeline` beschrieben, die für die Selbstlokalisierung relevant sind:

Linien-Messungen Die in Abschnitt 3.2.1 beschriebenen Feldlinien werden von der `Vision-Pipeline` gradientenbasiert im Bild erkannt. Da Linien von anderen Objekten, wie Robotern, Schiedsrichterbeinen oder Torpfosten verdeckt sein können, wird eine Feldlinie häufig in Form mehrerer Segmente wahrgenommen. Zur internen Kommunikation einer Linienmessung wird die Linie durch ihren Anfangs- und Endpunkt repräsentiert.

Tor-Messungen Auf Basis einer Torpfostenerkennung können die beiden sich auf dem Feld befindenden Tore erkannt werden. Die Wahrnehmung eines Tores wird aus paarweise erkannten Torpfosten kombiniert. Zur internen Kommunikation einer Tormessung wird das Tor durch die Position des unteren Endes der beiden Torpfosten repräsentiert.

Projektion Wie in Abschnitt 3.1.1 beschrieben, ist das NAO-Robotiksystem ausschließlich mit 2D-Kameras ausgestattet. Jegliche Informationen über die Entfernung eines Bildpunktes kann daher nicht direkt aus dem Bild gewonnen werden, sondern wird indirekt aus einer Projektion bestimmt. Dazu wird mit Hilfe der Vorwärtskinematik die dreidimensionale Pose der Kamera in den Koordinaten des Standfußes bestimmt. Unter der Annahme dass der Standfuß den Boden berührt, kann bei Kenntnis der Neigung des Roboters die Kameramatrix in Koordinaten des Untergrundes transformiert werden. Damit sind Orientierung und Höhe der Kamera relativ zur Untergrundebene bekannt.
Für Feldlinien und das untere Ende der Torpfosten kann angenommen werden, dass sich diese in einer Ebene mit dem Untergrund befinden. Mit Hilfe der zuvor berechneten Pose der Kamera werden Messungen von Linien und Torpfosten daher in das `Roboterkoordinatensystem` überführt, indem sie auf die Ebene des Untergrundes projiziert werden. In [17] wird diese Projektionsmethode im Detail beleuchtet.

Odometrie

Die relative Bewegung des Roboters zum Untergrund wird von der sogenannten `Odometrie` berechnet. Diese Verschiebung – `Odometrie-Offset` genannt – wird durch eine `2D-Pose` $\boldsymbol{\Theta}_{\mathrm{R'R''},\mathrm{R'}} = (\boldsymbol{r}_{\mathrm{O'O''},\mathrm{R'}}, \alpha_{\mathrm{R'R''}})$ repräsentiert. Sie beschreibt die Änderung der Roboter-Pose relativ zur Pose des letzten `Cycles` und ist in Abb. 3.6 visualisiert.

Die Berechnung des `Odometrie-Offsets` erfolgt getrennt für den translatorischen Anteil $\boldsymbol{r}_{\mathrm{R'R''},\mathrm{R'}}$, sowie den rotatorischen Anteil $\alpha_{\mathrm{R'R''}}$. Nachfolgend sei der Ursprung der beiden Odometrieanteile genauer beschrieben.

Translatorischer Odometrie-Offset Die `WalkingEngine` initiiert durch Berechnung geeigneter Gelenkwinkel die selbständige Bewegung des Roboters über das Feld. Aus diesen Gelenkwinkeln wird der translatorisch zurückgelegte Weg $\boldsymbol{r}_{\mathrm{R'R''},\mathrm{R'}}$ berechnet. Dazu wird mit Hilfe der `Vorwärtskinematik` die Verschiebung des Torsos relativ zum `Standfuß` innerhalb eines `Cycles` gemessen. Unter der Annahme, dass der `Standfuß` keine Relativgeschwindigkeit zum Untergrund aufweist, ergibt sich der translatorische `Odometrie-Offset` durch Projektion der dreidimensionalen Torsoverschiebung auf die Ebene des Untergrunds.

Rotatorischer Odometrie-Offset Die sogenannte `IMU-Sensorfusion` bestimmt die dreidimensionale Orientierung des Torsos im Raum. Dazu werden die Messungen des dreiachsigen Gyroskops, sowie der Beschleunigungssensoren herangezogen. Die aufgenommenen Winkelgeschwindigkeiten werden zu einer Winkeländerung innerhalb des letzten `Cycles` integriert. Zur Kompensation des langfristigen Drifts auf Grund der Integration wird die daraus gewonnene Schätzung mit Hilfe des durch die Beschleunigungssensoren identifizierten Erdbeschleunigungsvektors korrigiert. Der Algorithmus dieser Sensorfusion orientiert sich an [18] und kann dort im Detail nachvollzogen werden.
Das Ergebnis der `IMU-Sensorfusion` liefert die Orientierung als Eulerwinkel in Roll-Nick-Gier-Darstellung. Der rotatorische Anteil $\boldsymbol{S}_{R'R''}$ des `Odometrie-Offsets` entspricht der Differenz des Gierwinkels zum Zustand des letzten `Cycles`.

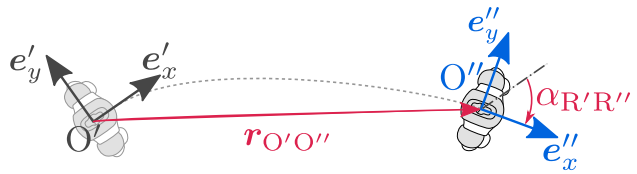

Abbildung 3.6: Schematisch Visualisierung der Odometrie – Behandlung der Eigenbewegung als `2D-Pose` $\boldsymbol{\Theta}_{\mathrm{R'R''}} := (\boldsymbol{r}_{\mathrm{R'R''}}, \alpha_{\mathrm{R'R''}})$.

3.2.5 Externes Wissen: Anfangs- und Randbedingungen

Während eines Fußballspiels stehen neben den Messungen der Sensoren noch weitere Informationen zu Verfügung. So lassen sich aus dem Regelwerk der SPL zusätzliche Informationen ableiten, die in den Berechnungsprozess mit einfließen können [16]. Während das Regelwerk hier nicht im Detail erläutert werden soll, sei nachfolgend die Qualität dieses „externen Wissens“ skizziert.

Anfangsbedingungen Wird der Roboter am Rand des Feldes eingesetzt, so ist sichergestellt, dass er sich zu diesem Zeitpunkt stets in der eigenen Hälfte befindet. Dieser Fall tritt zu Beginn jeder Halbzeit auf, wenn die Roboter zum Spiel auflaufen. Darüber hinaus werden sie bei Wiedereintritt nach einer Zeitstrafe auf Höhe des eigenen Strafstoßpunktes eingesetzt.
Mit Hilfe dieser Informationen kann die Menge der möglichen Anfangsposen auf ein Teilgebiet des Feldes eingegrenzt werden. Ungeachtet dessen erzeugt dieses Wissen immer mehrere mögliche Anfangspositionen.

Randbedingungen Während des Spielverlaufs sind nicht alle Positionen auf dem Feld zulässig. So ist unter anderem sichergestellt, dass sich zum Zeitpunkt des Anstoßes keine Roboter in der gegnerischen Hälfte aufhalten. Bei Regelverstoß wird der Roboter vom Schiedsrichter manuell in der eigenen Hälfte platziert. Darüber hinaus werden Roboter, die das Spielfeld verlassen, vorübergehend aus dem Spiel genommen und nach Ablauf einer Zeitstrafe am Rand des Spielfeldes eingesetzt.

Informationen über Spielzustandsänderungen, wie der Beginn einer Halbzeit oder das Einsetzen des Roboters am Rand, werden dem Roboter über das Netzwerk mitgeteilt. Intern stehen diese Informationen als sogenannter `GameState` zur Verfügung.

3.2.6 Besonderheiten des Anwendungsbeispiels

Aus der Kombination der Eigenschaften der Umwelt (vgl. Abschnitt 3.2.1), der verfügbaren Eingangsgrößen (vgl. Abschnitt 3.2.4), sowie dem Regelwerk ergeben sich einige Besonderheiten für das Lokalisierungsproblem. Nachfolgend werden diesbezüglich die wesentlichen Aspekte beleuchtet.

Symmetrie der Umgebung Wie aus der Beschreibung der Umgebung (vgl. Abschnitt 3.2.1) hervorgeht, ist das Fußballfeld symmetrisch. Da die verfügbaren

Eingangsgrößen ausschließlich auf `Features` des Feldes in Form von Feldmarkierungen und Toren beschränkt sind, kann daraus zunächst keine eindeutige Zuordnung zu einer 2D-Pose erfolgen. Selbst bei vollständiger Wahrnehmung dieser `Features` kann jede Messung mindestens zwei Posen auf dem Feld zugeordnet werden, da zum Feldmittelpunkt punktsymmetrische Posen paarweise die gleichen Messungen erzeugen.

Mehrdeutigkeit von Messungen Die Wahrnehmung der Feldfeatures, wie sie in Abschnitt 3.2.4 beschrieben wurde, erlaubt keine Unterscheidung von `Features` des selben Typs. So kann z.B. ein in `Roboterkoordinaten` wahrgenommenes Liniensegment allen Feldlinien des internen Weltmodells (vgl. Abschnitt 3.2.1) zugeordnet werden, die eine ausreichende Länge aufweisen. Diese Messungen sind daher im Allgemeinen mehrdeutig und die dazugehörige Wahrscheinlichkeitsverteilung strukturell stark multimodal ausgeprägt. Abbildung 3.7 veranschaulicht diese Problematik.

Multimodale Anfangsbedingungen Die in Abschnitt 3.2.5 beschriebenen Anfangsbedingungen zur Spezifikation möglicher Startzustände des Roboters sind im Allgemeinen nicht eindeutig. Insbesondere sind die Gebiete möglicher Anfangspositionen nicht notwendigerweise zusammenhängend, sodass die initiale Wahrscheinlichkeitsverteilung der Zustandsschätzung multimodal ausfällt.

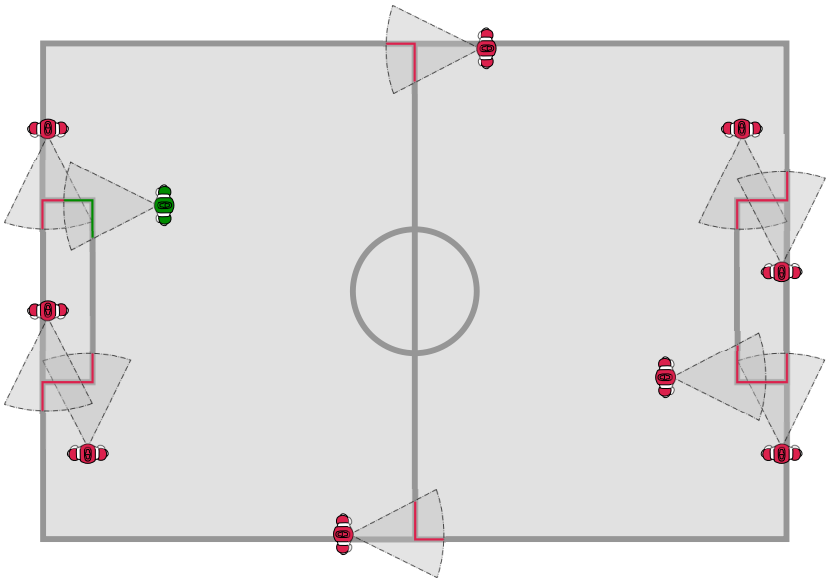

Abbildung 3.7: Mehrdeutigkeit von Messungen – Wahre Pose des Roboters mit dazugehöriger Wahrnehmung (grün) sowie weitere Zustände, die eine identische Wahrnehmung erzeugen (magenta).

Kapitel 4

Algorithmische Lösung

In den vorangegangenen Ausführungen wurden die Rahmenbedingungen für die Implementierung eines Algorithmus zur Selbstlokalisierung auf dem Fußballfeld für das NAO-Robotiksystem spezifiziert. In den nun folgenden Abschnitten wird die konkrete Realisierung zweier solcher Algorithmen – namentlich PF und UKF – auf dem NAO thematisiert. Dabei wird insbesondere auf die Anpassungen und Erweiterungen der allgemeinen Methoden aus Kapitel 2 eingegangen, die notwendig sind, um eine Anwendung im Kontext des hier betrachteten Lokalisierungsproblems zu ermöglichen.

Nach dem in Abschnitt 4.1 ein geeigneter Zustandsraum gewählt wird, wird in Abschnitt 4.2 die Modellierung der Systemdynamik beschrieben. Darauf aufbauend wird in Abschnitt 4.3 die Implementierung des vorhandenen PF erläutert, wie er zum Zeitpunkt dieser Arbeit in der Software der HULKs eingesetzt wird. Anschließend wird in Abschnitt 4.4 ein alternativer Algorithmus auf Basis des UKF entwickelt.

4.1 Wahl des Zustandsraums

Im Rahmen der hier betrachteten Lokalisierungsproblematik wird die `2D-Pose` $\boldsymbol{\Theta}_{\mathrm{FR}',\mathrm{F}}$ in `Feldkoordinaten` als zu schätzender Zustand gewählt. Die Zufallsvariable des Zustandes $\mathbf{X}$ beschreibt daher die Abbildung dieser `2D-Pose` auf den um die Orientierung erweiterten Positionsvektor

$$\begin{aligned} \mathbf{X} &: \mathbb{R}^2 \times (\mathbb{R}^2 \times \mathbb{R}^2) \to \mathbb{R}^3, \\ \boldsymbol{\Theta}_{\mathrm{FR}',\mathrm{F}} = (\boldsymbol{r}_{\mathrm{FR}',\mathrm{F}}(x,y), \boldsymbol{S}_{FR'}(\alpha)) &\mapsto \boldsymbol{x} = \begin{bmatrix} x & y & \alpha \end{bmatrix}^T. \end{aligned} \tag{4.1}$$

4.2 Modellierung der Systemdynamik

Die Änderung des Zustandes wird vornehmlich durch die selbsttätige Bewegung des Roboters initiiert. Wie in Abschnitt 3.2.4 beschrieben, steht die Änderung der Pose als bekannte Eingangsgröße in Form des `Odometrie-Offsets` zur Verfügung. In der Zustandsraumdarstellung ist es sinnvoll, eine solche aufgeprägte Zustandsänderung durch den Control-Input $\boldsymbol{u}$ zu modellieren. Der Zustandsübergang wird daher als

$$\mathbf{X}_t = \mathbf{X}_{t-1} + \boldsymbol{B}_{t-1}\boldsymbol{u}_{t-1} + \mathbf{W}_{t-1}, \tag{4.2}$$

$$\begin{bmatrix} x_{i,t} \\ y_{i,t} \\ \alpha_{i,t} \end{bmatrix} = \begin{bmatrix} x_{i,t-1} \\ y_{i,t-1} \\ \alpha_{i,t-1} \end{bmatrix} + \begin{bmatrix} \cos\alpha_{i,t-1} & -\sin\alpha_{i,t-1} & 0 \\ \sin\alpha_{i,t-1} & \cos\alpha_{i,t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{x',t-1} \\ u_{y',t-1} \\ u_{\alpha',t-1} \end{bmatrix} + \begin{bmatrix} w_{x',t-1} \\ w_{y',t-1} \\ w_{\alpha',t-1} \end{bmatrix}. \tag{4.3}$$

modelliert. Der Zustand $\mathbf{X}_{t-1}$ repräsentiert die `2D-Pose` des Roboters in `Feldkoordinaten` zum Zeitpunkt $t-1$. Das `Odometrie-Offset` $\boldsymbol{u}_{t-1} = [u_{x',t-1}, u_{y',t-1}, u_{\alpha',t-1}]^T$ repräsentiert die relative Änderung der `2D-Pose` zwischen Zeitpunkt $t-1$ und t. Das Zustandsübergangsmodell entspricht damit einer Koordinatentransformation der relativen Verschiebung $\boldsymbol{u}_t$ in `Feldkoordinaten`. Es ist wichtig zu bemerken, dass die Matrix $\boldsymbol{B}_{t-1}$ zur Transformation des `Odometrie-Offsets` $\boldsymbol{u}_{t-1}$ selbst eine Funktion des letzten Zustandes $\mathbf{X}_{t-1}$ ist. Insbesondere ist der Zustandsübergang damit nichtlinear. Die Rauschkomponente $\mathbf{W}_{t-1}$ wird als Prozessrauschen additiv in das System eingebracht.

Prozessrauschen Durch das Prozessrauschen $\mathbf{W}$ werden Unsicherheiten im Prädiktionsschritt des Filters berücksichtigt. Die Modellierung erfolgt als GZV mit $\mathbf{W} \sim \mathcal{N}(0, \boldsymbol{Q})$. Die Kovarianz des Rauschens

$$\boldsymbol{Q} = \boldsymbol{Q}_\mathrm{f} + \boldsymbol{Q}_\mathrm{o} \tag{4.4}$$

wird durch zwei Komponenten – `Filter-Prozessrauschen` $\boldsymbol{Q}_\mathrm{f}$ sowie `Odometrie-Prozessrauschen` $\boldsymbol{Q}_\mathrm{o}$ – modelliert. Diese Komponenten berücksichtigen unterschiedliche Quellen von Unsicherheiten und sind wie nachfolgend beschrieben.

Unsicherheiten durch Rundungsfehler Die Berechnungen des Algorithmus können nur mit endlicher Genauigkeit durchgeführt werden. Durch Rundungsfehler unterliegt das Ergebnis der Prädiktion selbst bei perfektem Modell somit stets einer gewissen Unsicherheit. Die Unsicherheiten die durch Rundungsfehler in das Modell eingehen, finden als

`Filter-Prozessrauschen` $\mathbf{V}_\mathrm{f} \sim \mathcal{N}(0, \boldsymbol{Q}_{fp})$ Berücksichtigung. Die Varianz $\boldsymbol{Q}_\mathrm{f}$ wird dabei als konstant angenommen.

Unsicherheiten im Control-Input Im Rahmen der Modellierung der Prädiktion wird angenommen, dass die Änderung der Pose durch das `Odometrie-Offset` als Control-Input erwirkt wird. Es ist daher sinnvoll, einen Teil des Rauschens in Abhängigkeit dieser Größe zu formulieren. Motiviert durch die Ausführungen in [19] wird eine zum Control-Input proportionale Rauschkomponente – das `Odometrie-Prozessrauschen` $\mathbf{V}_{op} \sim \mathcal{N}(0, \boldsymbol{Q}_{op})$ – eingebracht. Es wird angenommen, dass Varianzen pro überwundener Distanz

$$\boldsymbol{\sigma}_{\Delta op} = \begin{bmatrix} \sigma_{x'} & \sigma_{y'} & \sigma_{\alpha'} \end{bmatrix}^T \tag{4.5}$$

in x- und y-Richtung, sowie für die Rotation um die eigene Achse in `Roboterkoordinaten` konstant sind. Damit ergibt sich für die Berechnung der Varianz des `Odometrie-Prozessrauschens`:

$$\boldsymbol{u} = \begin{bmatrix} u_{x'} & u_{y'} & u_{\alpha'} \end{bmatrix}^T, \tag{4.6}$$

$$\boldsymbol{\sigma}_{op,R'} = \begin{bmatrix} |u_{x'}|\,\sigma_{x'} & |u_{y'}|\,\sigma_{y'} & |u_{\alpha'}|\,\sigma_{\alpha'} \end{bmatrix}^T, \tag{4.7}$$

$$\boldsymbol{\sigma}_{op,F} = \boldsymbol{S}_{FR'}\boldsymbol{\sigma}_{op,R'} = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_\alpha \end{bmatrix}^T, \tag{4.8}$$

$$\boldsymbol{Q}_{op} = \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_\alpha \end{bmatrix}. \tag{4.9}$$

Darin wird die streckenspezifische Varianz $\boldsymbol{\sigma}_{\Delta op}$ komponentenweise proportional zum Control-Input $\boldsymbol{u}$ skaliert und anschließend mit Hilfe des Zustandsmittelwertes mit $\boldsymbol{S}_{FR'}$ in `Feldkoordinaten` transformiert.

4.3 Adaption eines PF

Im Rahmen der folgenden Ausführungen wird die Funktionsweise eines Lokalisierungsalgorithmus beschrieben, wie er zum Zeitpunkt dieser Arbeit in der Software der HULKs eingesetzt wird. Dieser Algorithmus ist eine Adaption des in Abschnitt 2.5 beschriebenen PF.

4.3.1 Allgemeine algorithmische Struktur

Der Aufbau des Algorithmus orientiert sich in großen Teilen an dem des PF aus Abschnitt 2.5.2 und ergänzt diese Methode in wesentlichen Punkten, um

eine Anwendung vor dem Hintergrund der Rahmenbedingung (vgl. Kapitel 3) zu ermöglichen. Nachfolgend werden die einzelnen Schritte des Algorithmus übersichtsweise beschrieben. In Unterabschnitt 4.3.2 wird dann die Realisierung der einzelnen Teilschritte genauer erläutert.

1. **Initialisierung** Auf Basis des `GameStates` wird die Gesamtheit aller Partikel entsprechend der bekannten Ausgangsverteilung initialisiert.
2. **Prädiktion** Der nächste Zustand wird mit Hilfe des in Abschnitt 2.5.2 beschriebenen PF-Prädiktionsschrittes prädiziert.
3. **Zuordnung der Importance-Weights** Zur Korrektur der prädizierten Zustandsschätzung werden in diesem Schritt die Linienmessungen einbezogen. Dazu wird jedem Partikel jeweils ein Importance-Weight zugeordnet.
4. **Extraktion der resulitierenden `2D-Pose`** Eine eindeutige `2D-Pose` wird aus der Gesamtheit der `Positions-Partikel` zur Weitergabe an die nachgelagerten Algorithmen und Entscheidungsprozesse extrahiert.
5. **Resampling** Auf Basis der berechneten Importance-Weights wird das sogenannte Resampling durchgeführt. Der verwendete Sampling-Algorithmus umfasst wesentliche Adaptionen zur Verbesserung der Prozessdynamik und berücksichtigt gleichzeitig die Randbedingungen aus Abschnitt 3.2.5.

4.3.2 Implementierung

Nachfolgend wird die Realisierung der obigen Teilschritte im Detail beschrieben.

Das Positions-Partikel

Das `Positions-Partikel` implementiert die Funktionalitäten des Partikels aus Abschnitt 2.5.2. Als Zustandsgröße hält das i-te Partikel daher seine `2D-Pose` $\boldsymbol{\chi}_t^{[i]}$ in `Feldkoordinaten` (vgl. Abschnitt 4.1), sowie das ihm zugeordnete Importance-Weight $\mathrm{w}_t^{[i]}$.
Darüber hinaus wird das Partikel mit der sogenannten `Cluster-ID` $c^{[i]}$ um eine weitere Zustandsgröße ergänzt. Die `Cluster-ID` kennzeichnet die Zugehörigkeit des `Positions-Partikels` zu einer bestimmten Untergruppe der Partikel-Population. Für Partikel der gleichen `Cluster-ID` wird angenommen, dass sie der selben Mode der Zustandsverteilung zugeordnet werden können. Die `Cluster-ID` dient damit der Behandlung von eventuellen Multimodalitäten, wie sie für das Filterproblem zu erwarten sind (vgl. Abschnitt 3.2.6). Die Zuordnung und algorithmische Verwertung dieser `Cluster-ID` wird im Verlauf der nachfolgenden Ausführungen genauer erläutert.

Initialisierung

In diesem Schritt werden die `Positions-Partikel` basierend auf den Anfangsbedingungen aus Abschnitt 3.2.5 stochastisch im Bereich möglicher Anfangsposen initialisiert. Eine solche Initialisierung wird bei jeder Spielzustandsänderung des `GameStates` initiiert, die Informationen über mögliche Anfangspositionen des Roboters enthält.

Wird durch Anfangsbedingung der Spielzustandsänderung beispielsweise spezifiziert, dass der Roboter an der Seitenlinie in der eigenen Hälfte mit Ausrichtung zum Feld eingesetzt wird, so wird die Gesamtheit der `Positions-Partikel` stochastisch von einer entsprechenden Gleichverteilung wie in Abb. 4.1 dargestellt abgeleitet. Die Importance-Weights der einzelnen Partikel werden jeweils zu $\mathbf{w}_0^{[i]} = M^{-1}$ gewählt, da zunächst davon ausgegangen wird, dass alle Zustandshypothesen die gleiche Wahrscheinlichkeit aufweisen.
Des Weiteren wird die `Cluster-ID` $c^{[i]}$ abhängig von der y-Koordinate der `2D-Pose` der einzelnen Partikel $\boldsymbol{\chi}_t^{[i]}$ in `Feldkoordinaten` zu

$$\boldsymbol{\chi}_t^{[i]} = \begin{bmatrix} x_t^{[i]} \\ y_t^{[i]} \\ \alpha_t^{[i]} \end{bmatrix}, \qquad c^{[i]} = \begin{cases} 0, & \text{für } y_t^{[i]} \geq 0 \\ 1, & \text{für } y_t^{[i]} < 0 \end{cases} \tag{4.10}$$

gewählt. Auf diese Weise erhalten jeweils die Partikel, die an der gleichen Seitenlinie initialisiert wurden, die gleiche `Cluster-ID` $c^{[i]}$.

Prädiktion

Die Prädiktion des Zustandsübergangs der einzelnen Partikel $\boldsymbol{\chi}_i^{[m]}$ erfolgt nach dem Vorgehen aus Gl. (2.34). Dazu wird das Zustandsübergangsmodell aus Gl. (4.3) auf die einzelnen Partikel wie folgt angewendet:

$$\bar{\boldsymbol{\chi}}_t^{[i]} = \boldsymbol{\chi}_{t-1}^{[i]} + \boldsymbol{B}_{t-1}^{[i]} \boldsymbol{u}_{t-1} + \boldsymbol{w}_{t-1}^{[i]}, \tag{4.11}$$

$$\begin{bmatrix} \bar{x}_t^{[i]} \\ \bar{y}_t^{[i]} \\ \bar{\alpha}_t^{[i]} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{[i]} \\ y_{t-1}^{[i]} \\ \alpha_{t-1}^{[i]} \end{bmatrix} + \begin{bmatrix} \cos\alpha_{t-1}^{[i]} & -\sin\alpha_{t-1}^{[i]} & 0 \\ \sin\alpha_{t-1}^{[i]} & \cos\alpha_{t-1}^{[i]} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{x',t-1} \\ u_{y',t-1} \\ u_{\alpha',t-1} \end{bmatrix} + \begin{bmatrix} w_{x',t-1}^{[i]} \\ w_{y',t-1}^{[i]} \\ w_{\alpha',t-1}^{[i]} \end{bmatrix}. \tag{4.12}$$

Dabei wird für das $i-te$ Partikel $\boldsymbol{\chi}_{i,t-1}$ zum Zeitpunkt $t-1$ jeweils eine individuelle Stichprobe der Rauschkomponente $\boldsymbol{w}_{t-1}^{[i]} \sim p(\boldsymbol{w}_{t-1})$ erzeugt.

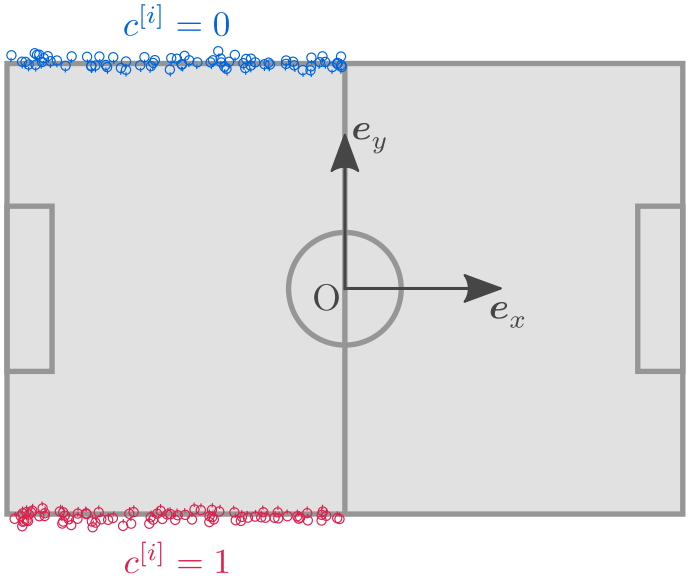

Abbildung 4.1: Initialisierung der Partikel für beliebige Startpositionen am Feldrand der eigenen Hälfte.

Zuordnung der Importance-Weights

Die Zuordnung der Importance-Weights zu den einzelnen Partikeln stellt das zentrale Element zur Integration der Messungen in die Zustandsschätzung dar. Als Messung stehen dem Roboter wahrgenommene Linien in `Roboterkoordinaten` zur Verfügung.
Wie in Abschnitt 2.5 beschrieben, wird durch das Importance-Weight $w_t^{[i]}$ die Wahrscheinlichkeit der Messungen für das jeweilige Partikel repräsentiert. Umgekehrt kann das zugeordnete Importance-Weight auch als Maß der Wahrscheinlichkeit betrachtet werden, dass die Zustandshypothese des einzelnen Partikels dem tatsächlichen Zustand des Systems entspricht.

Zunächst ist wichtig zu bemerken, dass Linienwahrnehmungen nicht nur durch gerade Feldlinien, sondern auch durch den Mittelkreis erzeugt werden. Der Mittelkreis wird daher im internen Modell der Umwelt als Polygon – das heißt in Form mehrerer gerader Feldlinien – modelliert. Auf diese Weise wird eine Behandlung der im Mittelkreis detektierten Linien äquivalent zu allen anderen Feldlinien erlaubt. Basierend auf dieser Betrachtung wird die Berechnung des Importance-Weights $w_t^{[i]}$ für ein `Positions-Partikel` mit $\bar{\boldsymbol{\chi}}_t^{[i]} = \begin{bmatrix} \bar{x}_t^{[i]} & \bar{y}_t^{[i]} & \bar{\alpha}_t^{[i]} \end{bmatrix}^T$ aus Linienmessungen $\boldsymbol{L}_{\mathrm{R}'}^{[k]}$ in `Roboterkoordinaten` wie nachfolgend erläutert konstruiert.

1. Transformation der Messungen in Weltkoordinaten Zunächst wird die in `Roboterkoordinaten` vorliegende Wahrnehmung mit Hilfe der `2D-Pose` $\bar{\boldsymbol{\chi}}_t^{[i]}$ des Partikels in `Feldkoordinaten` transformiert. Für einen in `Roboterkoordinaten` gegebenen Punkt $\boldsymbol{r}_{\mathrm{O'P,R'}} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ ist diese Transformationsvorschrift durch

$$\boldsymbol{r}_{\mathrm{OP,F}} = \boldsymbol{r}_{\mathrm{OO',F}} + \boldsymbol{S}_{\mathrm{FR'}} \boldsymbol{r}_{\mathrm{O'P,R'}}, \tag{4.13}$$

$$\boldsymbol{r}_{\mathrm{OP,F}} = \begin{bmatrix} \bar{x}_t^{[i]} \\ \bar{y}_t^{[i]} \end{bmatrix} + \begin{bmatrix} \cos\bar{\alpha}_t^{[i]} & -\sin\bar{\alpha}_t^{[i]} \\ \sin\bar{\alpha}_t^{[i]} & \cos\bar{\alpha}_t^{[i]} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \tag{4.14}$$

gegeben. Die Transformation eines gemessen Liniensegments $\boldsymbol{L}_{\mathrm{R'}}^{[k]} = \begin{bmatrix} \boldsymbol{p}_{\mathrm{1,R'}}^{[k]} & \boldsymbol{p}_{\mathrm{2,R'}}^{[k]} \end{bmatrix}$ erfolgt durch Anwendung dieser Transformationsvorschrift auf beide Endpunkte $\boldsymbol{p}_{\mathrm{1,R'}}^{[k]}, \boldsymbol{p}_{\mathrm{2,R'}}^{[k]}$ des Segments.

2. Ableitung der Importance-Weights Für eine gute Zustandshypothese ist zu erwarten, dass die in `Feldkoordinaten` transformierte Wahrnehmung in unmittelbarer Nähe einer korrespondierenden Feldmarkierung des internen Weltmodells liegt. Diese Erwartung kann in Abb. 4.2 nachvollzogen werden. Qualitativ wird hier ersichtlich, dass eine Zuordnung der Wahrnehmung zu einer `Feldmarkierung` des internen Modells für Partikel a mit geringerem Fehler möglich ist als für Partikel b.

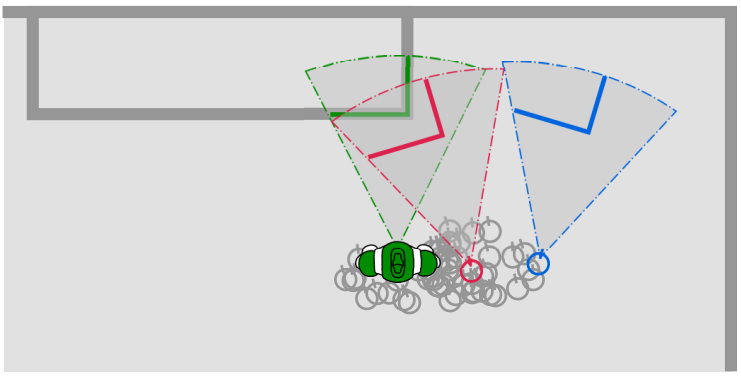

Abbildung 4.2: In `Feldkoordinaten` transformierte Linienwahrnehmungen für zwei Partikel a (magenta) und b (blau) sowie die wahre Position des Roboters (grün).

Für eine in `Feldkoordinaten` transformierte Linienmessung $\boldsymbol{L}_{\mathrm{F}}^{[k]} = \begin{bmatrix} \boldsymbol{p}_{\mathrm{l1,F}}^{[k]} & \boldsymbol{p}_{\mathrm{l2,F}}^{[k]} \end{bmatrix}$ wird die Berechnung des Zuordnungsfehlers $e^{[k,j]}$

zu einer gegeben Feldlinie $\boldsymbol{F}_{\mathrm{F}}^{[j]}$ des internen Modells mit

$$\boldsymbol{p}_{\mathrm{lm,F}}^{[k]} = \left(\boldsymbol{p}_{\mathrm{l1,F}}^{[k]} + \boldsymbol{p}_{\mathrm{l2,F}}^{[k]}\right)/2, \tag{4.15}$$

$$e_{\mathrm{dist}} = dist(\boldsymbol{p}_{\mathrm{lm,F}}, \boldsymbol{F}_{\mathrm{F}}), \tag{4.16}$$

$$e_{\alpha} = angleDiff(\boldsymbol{L}_{\mathrm{F}}, \boldsymbol{F}_{\mathrm{F}}) \tag{4.17}$$

wie folgt durchgeführt

$$e^{[k,j]} = \begin{cases} \left(e_{\mathrm{dist}}/d_{\max} + e_{\alpha}/\alpha_{\max}\right)/2, & \text{für } e_{\mathrm{dist}} \leq d_{\max} \wedge e_{\alpha} \leq \alpha_{\max} \\ 1, & \text{sonst.} \end{cases} \tag{4.18}$$

Darin bezeichnet $dist(\boldsymbol{p}, \boldsymbol{L})$ die Funktion zur Berechnung des orthogonalen Abstandes zwischen einem Punkt $\boldsymbol{p}$ und einem Liniensegment $\boldsymbol{L}$. Die Funktion $angleDiff(\boldsymbol{L}_1, \boldsymbol{L}_2)$ berechnet den Winkel zwischen zwei Liniensegmenten $\boldsymbol{L}_1$ und $\boldsymbol{L}_2$. Die Größen $d_{\max}$ und $\alpha_{\max}$ sind Parameter zur Konfiguration der maximal zulässigen Abweichung zwischen Messung $\boldsymbol{L}_{\mathrm{F}}^{[k]}$ und Feldlinie $\boldsymbol{F}_{\mathrm{F}}^{[j]}$. Für $e_{\mathrm{dist}} > d_{\max}$ bzw. $e_{\alpha} > \alpha_{\max}$ ist eine Zuordnung der Linienkombination j, k nicht sinnvoll, sodass der Zuordnungsfehler $e^{[k,j]}$ auf den Maximalwert gesetzt wird.

Für die Linienmessung $\boldsymbol{L}_{\mathrm{F}}^{[k]}$ wird der Fehler $e^{[k,j]}$ für jede Feldlinie $\boldsymbol{F}_{\mathrm{F}}^{[j]} \in \{\boldsymbol{F}_{\mathrm{F}}^{[1]}, \ldots, \boldsymbol{F}_{\mathrm{F}}^{[N]}\}$ des internen Weltmodells berechnet. Der Gewichtsfaktor $\mathrm{w}^{[i,k]}$ für diese Linienmessung wird dann als

$$\mathrm{w}^{[i,k]} = 1{,}1 - e_{\min}^{[i,k]}, \tag{4.19}$$

$$\text{mit } e_{\min}^{[i,k]} = \min_{k \in [N]} e^{[k,j]} \tag{4.20}$$

$$\tag{4.21}$$

bestimmt. Das Importance-Weight

$$\mathrm{w}_t^{[i]} = \prod_{j=1}^{N_{z_t}} \mathrm{w}_t^{[i,k]} \tag{4.22}$$

des i-ten `Positions-Partikels` ergibt sich schließlich durch Multiplikation der aus den N_{z_t} Einzelmessungen abgeleiteten Gewichtsfaktoren $\mathrm{w}^{[i,k]}$. Es ist wichtig zu bemerken, dass für die einzelnen Gewichtsfaktoren $\mathrm{w}^{[i,k]} \geq 0{,}1$ garantiert ist und diese damit insbesondere nicht verschwindend klein werden können. Auf diese Weise wird sichergestellt, dass eventuelle Fehlwahrnehmungen nicht den Einfluss weiterer Messungen unterdrücken.

Berücksichtigung der Randbedingungen Abschließend werden die in Abschnitt 3.2.5 formulierten Randbedingungen in die Zustandsschätzung einbezogen. Zu diesem Zweck wird das Gewicht aller `Positions-Partikel` auf Null gesetzt, die sich auf einer durch das Regelwerk ausgeschlossenen Position befinden. Dies betrifft z.B. alle Zustandshypothesen, die sich nicht auf dem Fußballfeld befinden.

Extraktion der resultierenden 2D-Pose

Das Berechnungsergebnis des Lokalisierungsalgorithmus soll eine eindeutige `2D-Pose` sein. Eine solche `2D-Pose` muss aus der Gesamtheit der `Positions-Partikel` extrahiert werden.

Ein einfacher und weit verbreiteter Ansatz zur Extraktion einer Zustandsschätzung aus den Partikeln $\bar{\boldsymbol{\chi}}_t$ ist die Berechnung der extrahierten Zustandsschätzung als Mittelwert der einzelnen Zustandshypothesen. Eine solche Strategie ist allerdings nur sinnvoll, wenn die Verteilung der Partikel-Population annähernd durch eine Normalverteilung beschrieben werden kann. Für die hier betrachtete Problemstellung sind im Allgemeinen stark multimodal ausgeprägte Verteilungen der Zustandshypothesen möglich (vgl. Abschnitt 3.2.6). Dies wird besonders deutlich, wenn das Beispiel der Initialisierung entlang der Seitenränder des Feldes aus Abschnitt 4.3.2 betrachtet wird. Hier wird schnell ersichtlich, dass eine reine Mittelwertbildung über die Gesamtheit aller Partikel zu einer vollständig falschen Zustandsschätzung führen würde.
Für Problemstellungen, bei denen eine multimodale Verteilung der Partikel zu erwarten ist, müssen die einzelnen Moden getrennt voneinander betrachtet werden, um eine sinnvolle Zustandsschätzung zu extrahieren. Dazu ist es allerdings notwendig, einzelne Partikel-Schwärme – sogenannte `Cluster` – zu identifizieren. Für eine solche Clusteranalyse existiert eine Vielzahl von Algorithmen, die eine Lösung dieser Problematik ermöglichen [20].
Für die hier betrachtete Problemstellung ist weder die Anzahl der `Cluster` im Voraus bekannt, noch ist für das Clustering umfassender Rechenaufwand wünschenswert, da dadurch die Echtzeitfähigkeit des Lokalisierungsalgorithmus gefährdet würde. Aus diesen Gründen wird die Zugehörigkeit zu einzelnen `Clustern` mit Hilfe der `Cluster-ID` indirekt modelliert, indem Partikel, die sich zum Zeitpunkt der Initialisierung im gleichen Cluster befinden, mit der gleichen `Cluster-ID` markiert werden (vgl. Abschnitt 4.3.2). Im Rahmen des Resamplings wird die `Cluster-ID` übernommen, sodass die Zuordnung der einzelnen `Positions-Partikel` zu einer Mode erhalten bleibt.
Diese Strategie trifft die Annahme, dass die anfänglich initialisierten Cluster auch im weiteren Verlauf des Filterprozesses zusammenhängend bleiben. Kommt es im Zeitverlauf zur Divergenz eines Clusters, schlägt diese Methode fehl. Wenngleich sich im Rahmen der praktischen Anwendung der Methode gezeigt hat, dass diese Art der indirekten Clusteranalyse in den meisten Fällen ausreichend gut funktioniert, stellt sie durchaus eine Schwachstelle des Algorithmus dar, die im Zweifelsfall zu gravierender Fehllokalisierung führen kann.

Unter der Prämisse, dass die `Cluster-ID` die Zugehörigkeit eines `Positions-Partikel` zu den einzelnen Moden korrekt wiedergibt, wird eine `2D-Pose` aus der Partikel-Population wie folgt extrahiert:

1. **Berechnung der Gewichtssummen pro Cluster** Für das k-te der N_c Cluster wird die Summe der zugehörigen `Positions-Partikel` mit $c^{[i]} = k$ zu

$$\mathrm{w}_{\mathrm{c},t}^{[k]} = \sum_{i=1}^{M} \mathrm{w}_t^{[i]} I(k, c^{[i]}) \tag{4.23}$$

berechnet. Darin bezeichnet I die Indikatorfunktion

$$I(a,b) = \begin{cases} 1, & \text{falls } a = b, \\ 0, & \text{falls } a \neq b. \end{cases} \tag{4.24}$$

2. **Auswahl des besten Clusters** Auf Basis der Gewichtssummen $\mathrm{w}_{\mathrm{c},t}^{[k]}$ wird der Index c_t^{best} des besten Clusters zu

$$c_t^{\mathrm{best}} = \underset{k \in [N_c - 1]_0}{\arg\max}\, \mathrm{w}_{\mathrm{c},t}^{[k]} \tag{4.25}$$

ermittelt.

3. **Mittelwertbildung aus bestem Cluster** Die `2D-Pose` $\boldsymbol{x}_t^{\mathrm{best}}$ wird schließlich als Mittelwert der Zustandshypothesen des besten Clusters zu

$$\boldsymbol{x}_t^{\mathrm{best}} = \frac{1}{\mathrm{w}_{\mathrm{c},t}^{[c_t^{\mathrm{best}}]}} \sum_{i=1}^{M} \bar{\boldsymbol{\chi}}_t^{[i]} \mathrm{w}_t^{[i]} I(c_t^{\mathrm{best}}, c^{[i]}) \tag{4.26}$$

berechnet. Darin werden die einzelnen Zustandshypothesen $\bar{\boldsymbol{\chi}}_t^{[i]}$ mit dem dazugehörigen Importance-Weight $\mathrm{w}_t^{[i]}$ gewichtet.

Resampling

Im Schritt des Resamplings wird wie in Abschnitt 2.5.2 beschrieben eine neue Stichprobe zur Repräsentation des Zustandes berechnet. Diese Stichprobe wird auf Basis der zuvor berechneten Importance-Weights generiert.

Die Resampling-Strategie aus Abschnitt 2.5.2 ist aus mehreren Gründen für die hier betrachtete Anwendung ungeeignet. Nachfolgend werden die wesentlichen Probleme dieser Strategie erläutert.

1. **Berechnungskomplexität** Für eine echtzeitfähige Anwendung des Algorithmus auf dem NAO ist eine niedrige Brechungskomplexität wünschenswert. Eine direkte Implementierung der Strategie würde für einen Stichprobenumfang M in $\mathcal{O}(M \log M)$ Zeitkomplexität resultieren [5]. Da die Partikelanzahl M i.d.R. große Werte annimmt, hat die Laufzeit des Resampling-Algorithmus wesentlichen Einfluss auf die Gesamtlaufzeit des PF. Eine Resampling-Strategie mit geringerer Zeitkomplexität ist daher wünschenswert.

2. **Stichprobenvarianz** Auf Grund der Laufzeitbeschränkungen zur echtzeitfähigen Anwendung auf dem NAO ist der Stichprobenumfang M stark begrenzt. Da die Stichprobe zufällig gewählt wird, kann es insbesondere für niedrige Partikelanzahlen M bei ungeeigneter Verteilung der Partikel zu hohen Varianzen zwischen den einzelnen Stichproben kommen. Die in Abschnitt 2.5.2 vorgestellte Resampling-Strategie begünstigt eine hohe Stichprobenvarianz [5, 21]. Diese Problematik wird besonders deutlich, wenn eine Situation betrachtet wird, in der allen Partikeln identische Importance-Weights zugeordnet werden. Wird für eine solche Gewichtung der einzelnen Partikel die Resampling-Strategie aus Abschnitt 2.5.2 angewendet, gehen mit hoher Wahrscheinlichkeit einige Partikel verloren, sodass sich die Partikel-Population bei wiederholtem Resampling langfristig im Bereich einer einzelnen Hypothese ansammelt. Ein solches Verhalten ist unerwünscht und durch ein geeignetes Resampling zu vermeiden.

In [5, 21, 22] werden verschiedene Resampling-Algorithmen vorgestellt, die die oben genannten Probleme berücksichtigen. Im Rahmen der Implementierung der HULKs wird eine Variante des sogenannten „Stochastic Universal Resampling“ (SUS) eingesetzt. Diese Methode – auch „low variance sampling“ genannt – wird vornehmlich im Bereich genetischer Algorithmen angewendet und erlaubt ein Resampling nach den Anforderungen des PF mit kleiner Stichprobenvarianz in $\mathcal{O}(M)$ [22].

Die Struktur des verwendeten SUS ist in Algorithmus 1 skizziert. Die algorithmische Idee ist überdies in Abb. 4.3 visualisiert. Jedes Segment repräsentiert darin ein Partikel $\bar{\boldsymbol{\chi}}_t^{[i]}$. Die Breite des jeweiligen Segments korrespondiert mit dem Importance-Weight $\mathrm{w}_t^{[i]}$ des dazugehörigen Partikels.
Das gesamte Sampling kommt mit der Wahl einer einzigen Zufallszahl $r \in [0, M^{-1}]$ aus. Ausgehend von diesem Startwert werden M Partikel in äquidistantem Gewichtsabstand gewählt. Die Auswahlwahrscheinlichkeit eines einzelnen Partikels entspricht damit weiterhin dem zugeordneten Importance-Weight, während gleichzeitig eine übermäßige Dominanz einzelner Partikel vermieden wird. Wird für diese Resampling-Strategie erneut die Situation gleichmäßig verteilter Importance-Weights betrachtet, ist leicht ersichtlich, dass

die Menge der Partikel unverändert vom Resampling-Algorithmus zurückgegeben wird.

Unter Einsatz von Algorithmus 1 wird schließlich das Resampling durchgeführt. Die `Cluster-ID` der einzelnen `Positions-Partikel` wird unverändert übernommen und die Importance-Weights werden jeweils auf $\mathrm{w}_{t+1}^{[i]} = M^{-1}$ zurückgesetzt.

Algorithmus 1 Stochastic Universal Sampling (SUS) nach [5].

procedure SUS($\bar{\boldsymbol{\chi}}_t, \mathbf{w}_t$) ▷ Sigma-Punkt-Matrix $\bar{\boldsymbol{\chi}}_t$, Gewichtsvektor $\mathbf{w}_t$
  $\boldsymbol{\chi}_t \leftarrow \emptyset$
  $r \leftarrow rand(0; M^{-1})$ ▷ Zufälliger Startpunkt zwischen 0 und M^{-1}
  $p \leftarrow M^{-1}$ ▷ Abstand der äquidistanten Zeiger
  $\mathbf{w}_t \leftarrow \mathbf{w}_t / \left\|\mathbf{w}_t\right\|_1$
  $\mathrm{w}_{sum} = 0$
  for $i = 1, \ldots, M$ **do**
    $\mathrm{w}_{sum} \leftarrow \mathrm{w}_{sum} + \mathrm{w}_t^{[i]}$
    while ($r < \mathrm{w}_{sum}$ **and** $size(\bar{\boldsymbol{\chi}}_t)$) **do**
      $\boldsymbol{\chi}_t \leftarrow \begin{bmatrix} \boldsymbol{\chi}_t & \bar{\boldsymbol{\chi}}_t^{[i]} \end{bmatrix}$
      $r \leftarrow r + p$
    end while
  end for
  return $\boldsymbol{\chi}_t$
end procedure

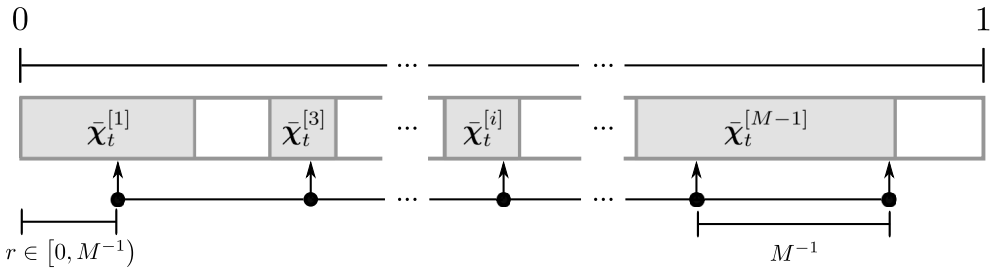

Abbildung 4.3: Stochastic Universal Sampling nach [5].

4.4 Adaption eines UKF

Das UKF in seiner allgemeinen Form kann nicht unmittelbar verwendet werden, um das hier behandelte Selbstlokalisierungsproblem zu lösen. Vielmehr sind einige Adaptionen notwendig, um die einzelnen in Abschnitt 2.4.3 beschriebenen algorithmischen Schritte im Kontext der Selbstlokalisierung auf dem Fußballfeld anwenden zu können. Im Rahmen dieser Arbeit wird daher eine Adaption eines UKF entwickelt, deren methodisches Vorgehen nachfolgend erläutert wird. Die Entwicklung dieses Algorithmus basiert auf den gleichen Eingangsgrößen wie das zuvor beschriebene PF. Ziel der Entwicklung ist eine robuste, echtzeitfähige Filtermethode für die Anwendung auf dem NAO. Der hier entwickelte Algorithmus wird im Rahmen dieser Arbeit an Stelle des vorhandenen PF im Software-Framework der HULKs als C++-Implementierung realisiert.

4.4.1 Multihypothesenstrategie

Die Ausführungen in Abschnitt 3.2.5 machen deutlich, dass das Wissen über Anfangs- und Randbedingungen eine Einschränkung der Menge der möglichen Anfangspositionen des Roboters erlaubt. In Abschnitt 3.2.6 wird allerdings auch darauf hingewiesen, dass für die Verteilungsfunktion des Zustandes aus verschiedenen Gründen Multimodalität zu erwarten ist. Wie in den Ausführungen zu theoretischen Grundlagen des UKF (vgl. Abschnitt 2.4) erläutert, approximiert das UKF die Wahrscheinlichkeitsdichtefunktion des Zustandes durch eine unimodale Normalverteilung. Eine multimodale Zustandsverteilung kann als gaußsche Mischverteilung, also als Summe mehrerer Normalverteilungen modelliert werden [5].
Strategien zur Behandlung dieser multimodalen Problemstellung werden im RoboCup schon seit längerer Zeit verfolgt [23] und werden auch im Kontext anderer Filterprobleme angewendet [24].

Im Rahmen dieser Arbeit wird eine Multihypothesenstrategie auf Basis des UKF entwickelt, die die einzelnen Moden als separate Hypothesen betrachtet und diese jeweils durch eine UKF-Instanz verfolgt. Die Hypothesen werden dann im zeitlichen Verlauf unter Berücksichtigung mehrerer Messungen gegeneinander ausgeschlossen, um schlussendlich eine eindeutige `Positionshypothese` zu erhalten. Darüber hinaus erlaubt diese Art der Behandlung eventuell widersprüchliche Messungen durch Erzeugung neuer Hypothesen zu berücksichtigen. Die vorgestellte UKF-Adaption übernimmt die grundlegende Idee der Verwendung mehrerer Hypothesen aus etablierten Strategien [23, 24] und entwickelt eine robuste Methode zur Auswahl und Reduktion der Hypothesen.

4.4.2 Allgemeine algorithmische Struktur

Der Ablauf der Filtermethode folgt den nachfolgend erwähnten Teilschritten. Eine detaillierte Beschreibung der einzelnen Schritte erfolgt in Abschnitt 4.4.6.

1. **Initialisierung** Auf Basis des `GameStates` wird eine initiale Menge von Hypothesen erzeugt, die das Gebiet möglicher Anfangsposen abdecken.

2. **Prädiktion** Die Änderung der `2D-Pose` der einzelnen Hypothesen wird aus der Eigenbewegung des Roboters prädiziert. Zu diesem Zweck wird ein UKF-Prädiktionsschritt für jede Hypothese durchgeführt.

3. **Korrektur** Die Zustände der einzelnen Hypothesen werden durch Integration von Linienmessungen korrigiert. Dazu werden aus den einzelnen Linienwahrnehmungen korrespondierende Posen abgeleitet, die die jeweilige Messung erklären. Mit Hilfe dieser abgeleiteten Posen wird ein UKF-Korrekturschritt für jede Hypothese durchgeführt.

4. **Hypothesen-Reduktion** Hypothesen, deren Zustandsschätzungen hinreichend nah beieinander liegen, werden zu einer einzelnen Hypothese zusammengefasst.

5. **Hypothesen-Generierung** Auf Basis von Tormessungen, aus denen eine vollständige `2D-Pose` erzeugt werden kann, werden unter bestimmten Kriterien neue Hypothesen erzeugt.

6. **Hypothesen-Evaluation** Die einzelnen Hypothesen werden anhand der Messungen bewertetet und hinreichend schlechte Hypothesen gelöscht. Abschließend wird auf Basis dieser Bewertungen die beste Hypothese ausgewählt.

4.4.3 Modellierung des Messrauschens

Unsicherheiten im Prozess des Messvorganges werden durch die Rauschkomponenten $\mathbf{V} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$ berücksichtigt. Ein verbreiteter Ansatz zur Rauschmodellierung ist die Annahme *additiven Rauschens* [8, 19]. Eine solche Modellierung wird daher mit

$$\bar{\mathbf{Z}}_t = \boldsymbol{h}_t\left(\bar{\mathbf{X}}_t\right) + \mathbf{V}_t \tag{4.27}$$

auch für das hier vorliegende Problem gewählt.

4.4.4 Adaption der Unscented Transformation

Im Zuge der Implementierung des in diesem Abschnitt beschriebenen UKF-Algorithmus hat sich gezeigt, dass die in Abschnitt 2.4.2 vorgestellte UT für eine numerisch stabile Implementierung auf dem NAO nicht geeignet ist. Nachfolgend wird diese Problematik diskutiert, um anschließend eine angepasste UT zur Lösung dieser Problematik vorzustellen.

Problem Die UT in ihrer Formulierung aus [11] wie sie in Abschnitt 2.4.2 beschrieben ist, lässt durch die Wahl des Parameters $\kappa \in \mathbb{R}$ explizit negative Gewichte W_i zu. Gleichzeitig wird zur Berechnung der Sigmapunkte die Matrixwurzel der Kovarianzmatrix benötigt. Zur effizienten Berechnung der Matrixwurzel soll im Rahmen dieser Arbeit die Cholesky-Zerlegung verwendet werden. Grundlegende Voraussetzung dafür ist allerdings, dass die Kovarianzmatrix zum Zeitpunkt der Zerlegung symmetrisch positiv definit ist.

Die Kovarianz der Sigma-Punkte wird – wie in Gl. (2.17) beschrieben – als eine gewichtete Summe dyadischer Produkte berechnet, die ihrerseits jeweils mindestens positiv-semidefinit sind. Gleichzeitig kann für die Kovarianzmatrizen $\boldsymbol{R}_t$ und $\boldsymbol{Q}_t$ von Mess- und Prozessrauschen gefordert werden, dass diese stets positiv definit sein müssen. Entsprechend wird positive Definitheit der resultierenden Kovarianzmatrix garantiert, solange kein Gewicht W_i der Berechnung negativ ist [12].

Lösung In [12] werden Adaptionen der UT vorgeschlagen, die diese Problematik umgehen. Nach [13] ist für jede Wahl von Sigmapunkten, deren Erwartungswert und Kovarianz mit der untransformierten Verteilung übereinstimmen, eine fehlerfreie Transformation dieser Größen bis zur zweiten Ordnung sichergestellt. Insbesondere ist die Wahl negativer Gewichte W_i ausschließlich dadurch motiviert, die Verteilung der Sigmapunkte auf eine kleinere Umgebung zu beschränken und damit Effekte der Transformation bezüglich höherer statistischer Momente besser abbilden zu können [13]. Zudem weist [8] darauf hin, dass die Skalierung der Sigmapunkte vordergründig nach Kriterien numerischer Stabilität gewählt werden sollten, solange das Problem keine gesonderten Anforderung an die Approximationsgüte höherer statistischer Momente stellt.

Für den hier vorliegenden Anwendungsfall wird angenommen, das eine Approximation der Transformationsvorschriften bis zur zweiten Ordnung ausreichend ist. Vor diesem Hintergrund kann die UT in einer Weise angepasst werden, die die Verwendung negativer Gewichte vermeidet und gleichzeitig die Anzahl der benötigten Rechenoperationen reduziert, ohne dabei die oben genannten Bedingungen zu verletzen.

Für eine n-dimensionale Zufallsvariable werden die $2n+1$ Sigmapunkte wie folgt gewählt:

Für $i = 1, \ldots, n$

$$\begin{aligned} \boldsymbol{\chi}_0 &= \boldsymbol{x}, \\ \boldsymbol{\chi}_i &= \boldsymbol{x} + \left(\sqrt{\boldsymbol{P}}\right)_i, \\ \boldsymbol{\chi}_{i+n} &= \boldsymbol{x} - \left(\sqrt{\boldsymbol{P}}\right)_i. \end{aligned} \tag{4.28}$$

Bei Verwendung dieser Berechnungsvorschrift für die Wahl der Sigmapunkte muss die Berechnung von Mittelwert und Kovarianz wie folgt angepasst werden

$$\boldsymbol{x} = \frac{1}{2n+1} \sum_{i=0}^{2n} \boldsymbol{\chi}_i, \tag{4.29}$$

$$\boldsymbol{P} = \frac{1}{2} \sum_{i=0}^{2n} \left(\boldsymbol{\chi}_i - \boldsymbol{x}\right) \left(\boldsymbol{\chi}_i - \boldsymbol{x}\right)^T. \tag{4.30}$$

4.4.5 Sensormodelle

Die Korrektur der Zustandsschätzung mit einer Messungen $\boldsymbol{z}$ erfordert die Kenntnis des zugehörigen Sensormodells $\boldsymbol{h}$ (vgl. Abschnitt 2.4). Dieses Sensormodell beschreibt, wie die Messung $\boldsymbol{z}$ vom Zustand $\boldsymbol{x}$ abhängt. Nachfolgend werden die einzelnen hier verwendeten Sensormodelle vorgestellt. Die Beschreibung der Verwendung dieser Sensormodelle erfolgt in Abschnitt 4.4.6.

1D-Posen-Sensor

Der `1D-Posen-Sensor` modelliert eine Messung, die ausschließlich Informationen über eine der beiden Positionskoordinaten (x oder y), sowie die Orientierung α des Roboters enthält. Die Abbildung eines Zustandes $\boldsymbol{x}$ auf die dazugehörige Messung $\bar{\boldsymbol{z}}_{\text{Pose1D}}$ erfolgt durch die lineare Abbildung

$$\begin{aligned} \boldsymbol{h}_{\text{Pose1D}} &: \mathbb{R}^3 \to \mathbb{R}^2, \\ \boldsymbol{x} = \begin{bmatrix} x & y & \alpha \end{bmatrix}^T &\mapsto \boldsymbol{H}_{\text{Pose1D}} \boldsymbol{x} = \bar{\boldsymbol{z}}_{\text{Pose1D}}, \end{aligned} \tag{4.31}$$

wobei gilt

$$\boldsymbol{H}_{\text{Pose1D}} = \begin{bmatrix} a & (1-a) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ mit } \begin{cases} a = 1, & \text{für Messungen mit x-Koordinate,} \\ a = 0, & \text{für Messungen mit y-Koordinate.} \end{cases}$$

2D-Posen-Sensor

Der `1D-Posen-Sensor` modelliert die Messung einer vollständigen `2D-Pose`. Die Abbildung $\boldsymbol{h}_{\text{Pose2D}}$ vom Zustandsraum in den Messraum ist trivialerweise die Identität

$$\begin{aligned} \boldsymbol{h}_{\text{Pose1D}} : \mathbb{R}^3 &\to \mathbb{R}^3, \\ \boldsymbol{x} &\mapsto \boldsymbol{H}_{\text{Pose1D}}\boldsymbol{x} = \bar{\boldsymbol{z}}_{\text{Pose2D}}, \\ \boldsymbol{H}_{\text{Pose2D}} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{4.32}$$

1D-Positions-Sensor

Der `1D-Positions-Sensor` modelliert die Messung $\boldsymbol{z}_{\text{Position1D}} = \boldsymbol{r}_{\text{O}'\text{P,R}'}$ eines Punktes P in `Roboterkoordinaten`, dessen Position in `Feldkoordinaten` $\boldsymbol{r}_{\text{OP,F}} = \begin{bmatrix} x_{\text{OP,F}} & y_{\text{OP,F}} \end{bmatrix}^T$ bekannt ist. Für eine solche Messung ist die nichtlineare Messfunktion

$$\begin{aligned} \boldsymbol{h}_{\text{Position1D}} &: \mathbb{R}^3 \to \mathbb{R}^2, \\ \boldsymbol{x} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} &\mapsto \boldsymbol{h}_{\text{Pose1D}}\left(\boldsymbol{x}; \boldsymbol{r}_{\text{OP,F}}\right) = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} (x_{\text{fm,F}} - x) \\ (y_{\text{fm,F}} - y) \end{bmatrix} \end{aligned} \tag{4.33}$$

gegeben.

4.4.6 Implementierung

Nachfolgend wird die Realisierung der Teilschritte des in Abschnitt 4.4.2 skizzierten Algorithmus im Detail erläutert.

Die Positionshypothese

Die einzelnen Moden der Zustandsverteilung werden wie einleitend beschrieben als separate Hypothesen möglicher Zustände betrachtet. Für diese Hypothesen erfolgt jeweils eine individuelle Zustandsschätzung. Die Implementierung einer solchen Hypothese wird nachfolgend als `Positionshypothese` bezeichnet. Zentrales Element der `Positionshypothese` ist ein UKF zur Schätzung der `2D-Pose` dieser Hypothese. Entsprechend hält die `Positionshypothese` Mittelwert $\boldsymbol{x}_t$ und Kovarianz $\boldsymbol{P}_t$ zur Beschreibung ihres Zustand $\mathbf{X}_t$ als Zustandsvariablen. Darüber hinaus beinhaltet die `Positionshypothese` den sogenannten `Evaluationsfehler`.

Dieser gibt Auskunft über die Qualität der Hypothese. Die Realisierung des zugehörigen UKF, sowie die Berechnung des `Evaluationsfehlers` werden im Verlauf der weiteren Ausführungen genauer erläutert.

Initialisierung

In einem ersten Schritt werden zunächst die Anfangsbedingungen berücksichtigt, die wie in Abschnitt 3.2.5 aus dem `GameState` abgeleitet werden können. Die Berücksichtigung dieser Anfangsbedingungen erfolgt, indem eine geeignete Menge von `Positionshypothesen` im Bereich möglicher Anfangspositionen erzeugt wird. Auf diese Weise wird die durch die Anfangsbedingung gegebene initiale Verteilung des Zustandes durch mehrere Normalverteilungen approximiert.
Wird durch Anfangsbedingung beispielsweise spezifiziert, dass der Roboter an der Seitenlinie in der eigenen Hälfte mit Ausrichtung zum Feld eingesetzt wird, so werden je vier äquidistant verteilte `Positionshypothesen` mit ausreichend großer Varianz an den Seitenlinien der eigenen Hälfte initialisiert. Eine solche Initialisierung ist in Abb. 4.4 dargestellt.
Die Initialisierung für andere Anfangsbedingungen erfolgt analog und wird hier nicht weiter erläutert. Es sei allerdings darauf hingewiesen, dass der Wahl einer ausreichend großen Varianz eine tragende Rolle zukommt, um die Unsicherheit der Anfangsposition abzubilden. Insbesondere kann auch bei diskreten Anfangspositionen nur eine endlich genaue Platzierung des Roboters erwartet werden.

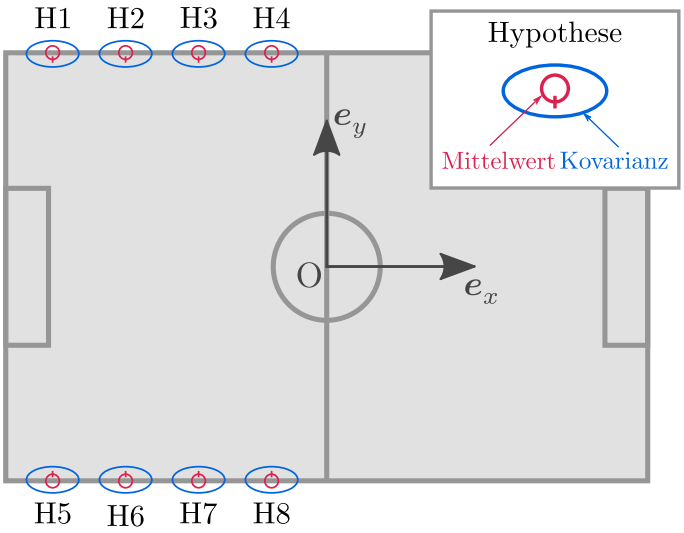

Abbildung 4.4: Initialisierung der `Positionshypothesen` für beliebige Startpositionen am Feldrand der eigenen Hälfte.

Prädiktion

Im Schritt der Prädiktion werden die `2D-Posen` der einzelnen `Positionshypothesen` für den nächsten Zeitschritt mit Hilfe der Systemdynamik aus Abschnitt 4.2 prädiziert. Zu diesem Zweck wird für jede Hypothese ein UKF-Prädiktionsschritt wie in Abschnitt 2.4 beschrieben durchgeführt. Da sich aus der hier gewählten Modellierung des Rauschens (vgl. Abschnitt 4.4.3), sowie der Adaption der UT (vgl. Abschnitt 4.4.4) einige Besonderheiten ergeben, sei die Prädiktion für eine einzelne `Positionshypothese` nachfolgend noch einmal übersichtsweise spezifiziert.

1. **Generierung der Sigma-Punkte** Die Modellierung des Prozessrauschens als additives gaußsches Rauschen (vgl. Abschnitt 4.4.3) erlaubt es, im Gegensatz zum Vorgehen des UKF aus Abschnitt 2.4.3 auf eine Erweiterung des Zustandes um die Rauschkomponenten zu verzichten [8, 19]. Für den dreidimensionalen Zustandsvektor $\boldsymbol{x}_t$ müssen daher nur sieben Sigma-Punkte nach dem angepassten Schema aus Gl. (4.28) erzeugt werden.

2. **Transformation** Zur Berechnung des prädizierten Zustandes $\bar{\mathbf{X}}_t \sim \mathcal{N}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t)$ werden zunächst die Sigma-Punkte $\boldsymbol{\chi}_{t-1}$ mit Hilfe der Systemdynamik aus Abschnitt 4.2 zu

$$\bar{\boldsymbol{\chi}}_{i,t} = \boldsymbol{\chi}_{i,t-1} + \boldsymbol{B}_{i,t-1}\boldsymbol{u}_{t-1} \tag{4.34}$$

transformiert (vgl. Gl. (4.34)). Es ist wichtig zu bemerken, dass für jeden Sigma-Punkt ein individuelles $\boldsymbol{B}_{i,t-1}$ zu wählen ist, da die Matrix selbst eine Funktion des Zustandes ist (vgl. Abschnitt 4.2). Anschließend wird der prädizierte Zustand $\bar{\mathbf{X}}_t \sim \mathcal{N}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{P}}_t)$ als

$$\bar{\boldsymbol{x}}_t = \frac{1}{7}\sum_{i=0}^{6} \bar{\boldsymbol{\chi}}_{i,t}, \tag{4.35}$$

$$\bar{\boldsymbol{P}}_t = \frac{1}{2}\sum_{i=0}^{6} \left(\bar{\boldsymbol{\chi}}_{i,t} - \bar{\boldsymbol{x}}_t\right)\left(\bar{\boldsymbol{\chi}}_{i,t} - \bar{\boldsymbol{x}}_t\right)^T + \boldsymbol{Q}_{t-1} \tag{4.36}$$

berechnet. Die Gewichtung der Sigma-Punkte entspricht der Adaption aus Abschnitt 4.4.4. Das Prozessrauschen $\mathbf{W}_{t-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_{t-1})$ wird wie in Abschnitt 4.4.3 beschrieben als additives Rauschen modelliert. Für additives Rauschen ist eine Behandlung der Rauschkomponente analog zum Vorgehen des KF (vgl. Abschnitt 2.3.2) zulässig [8]. Aus diesem Grund wird das Prozessrauschen bei der Transformation der Sigma-Punkte nicht unmittelbar berücksichtigt, sondern fließt durch nachträgliche Addition der Kovarianzmatrix $\boldsymbol{Q}_{t-1}$ in Gl. (4.36) ein.

Korrektur

Im Korrekturschritt des Algorithmus werden die Zustandsschätzungen der einzelnen `Positionshypothesen` mit Hilfe der wahrgenommenen Liniensegmente angepasst. Ziel dieses Schrittes ist die Durchführung eines UKF-Korrekturschrittes nach dem Vorgehen aus Abschnitt 2.4.3. Um dies zu ermöglichen, sind allerdings einige Vorverarbeitungsschritte notwendig.

Betrachtet wird die Wahrnehmung einer Menge von n Liniensegmenten $\{\boldsymbol{L}_{1,\mathrm{R}'}, \ldots, \boldsymbol{L}_{n,\mathrm{R}'}\}$ in `Roboterkoordinaten`. Für jede `Positionshypothese` werden die nachfolgend erläuterten Schritte durchgeführt.

Assoziation Um aus einem wahrgenommenen Liniensegment $\boldsymbol{L}_{k,\mathrm{R}'} \in \{\boldsymbol{L}_{1,\mathrm{R}'}, \ldots, \boldsymbol{L}_{n,\mathrm{R}'}\}$ Informationen über die `2D-Pose` des Roboters extrahieren zu können, muss diese zunächst mit einer zugehörigen Feldlinie $\boldsymbol{F}_{j,\mathrm{F}} \in \{\boldsymbol{F}_{1,\mathrm{F}}, \ldots, \boldsymbol{F}_{m,\mathrm{F}}\}$ der bekannten Umwelt (vgl. Abschnitt 3.2.1) assoziiert werden. Diese Assoziation orientiert sich an der Strategie des PF aus Abschnitt 4.3.2. Erneut sei bemerkt, dass der Mittelkreis im internen Weltmodell des Feldes – analog zum Vorgehen des PF – als Polygon mehrerer gerader Feldlinien modelliert wird. Die Detektion von Liniensegmenten im Mittelkreis bedarf daher zunächst keiner gesonderten Behandlung.
Im Gegensatz zum adaptierten PF werden die einzelnen Linienwahrnehmungen jedoch nicht getrennt voneinander betrachtet, sondern im Gesamtkontext assoziiert. Zu diesem Zweck werden alle wahrgenommenen Liniensegmente $\{\boldsymbol{L}_{1,\mathrm{R}'}, \ldots, \boldsymbol{L}_{n,\mathrm{R}'}\}$ für jeden Sigma-Punkt $\boldsymbol{\chi}_i$ des aktuellen Zustandes der Hypothese in `Feldkoordinaten` zu $\{\boldsymbol{L}^{[i]}_{1,\mathrm{F}}, \ldots, \boldsymbol{L}^{[i]}_{n,\mathrm{F}}\}$ transformiert. Für jede dieser transformierten Mengen wird der einzelnen Linie $\boldsymbol{L}^{[i]}_{k,\mathrm{F}} \in \{\boldsymbol{L}^{[i]}_{1,\mathrm{F}}, \ldots, \boldsymbol{L}^{[i]}_{n,\mathrm{F}}\}$ jeweils diejenige Feldlinie zugeordnet, die nach Gl. (4.15) bis 4.18 den kleinsten Assoziationsfehler $e^{[i,k]}_{\min}$ aufweist.

Die Assoziationsfehler $e^{[i,k]}_{\min}$ aller Liniensegmente werden für den i-ten Sigma-Punkt zum Gesamtfehler

$$e^{[i]} = \sum_{k=1}^{n} e^{[i,k]}_{\min} \tag{4.37}$$

akkumuliert.
So ergibt sich für jeden Sigma-Punkt eine individuelle Assoziation der Linienmessungen mit den Feldlinien des internen Modells. Für die `Positionshypothese` wird schlussendlich die Assoziation desjenigen Sigma-Punktes gewählt, für den sich der kleinste Gesamtfehler $e^{[i]}$ einstellt.

Ergebnis des Assoziationsschrittes ist somit eine Zuordnung der Linienmessungen zu genau einer Feldlinie $\boldsymbol{L}_{k,\mathrm{R}'} \rightarrow \boldsymbol{F}_{j,\mathrm{F}}$. Insbesondere ist es zulässig, dass mehrere

Linienmessungen der gleichen Feldlinie des internen Modells zugeordnet werden, da die Wahrnehmung einer Feldlinie in Form mehrerer Segmente denkbar ist.

Ableitung korrespondierender Messungen In diesem Schritt werden die assoziierten Linienmessungen zu korrespondierenden Messungen $\boldsymbol{z}$ übersetzt, für die ein Sensormodell zur Modellierung der Messung vorliegt. Zur Ableitung dieser Messungen werden Linien, die im vorangegangenen Schritt mit einem Segment des Mittelkreispolygons assoziiert wurden, separat behandelt. Die Weiterverarbeitung der assoziierten Linien $\boldsymbol{L}_{k,\mathrm{R}'} \rightarrow \boldsymbol{F}_{j,\mathrm{F}}$ erfolgt daher nach folgender Fallunterscheidung:

1. Feldlinien Eine Linienmessung $\boldsymbol{L}_{k,\mathrm{R}'}$ liefert Informationen über Abstand und Orientierung des Roboters zu dieser Linie. Durch die Assoziation $\boldsymbol{L}_{k,\mathrm{R}'} \rightarrow \boldsymbol{F}_{j,\mathrm{F}}$ mit einer der geraden Feldlinien kann diese Information in das `Feldkoordinatensystem` übertragen werden. Es ist dabei wichtig zu bemerken, dass aus der Assoziation nicht hervorgeht, von welcher Seite die Feldlinie beobachtet wurde. Aus dem assoziierten Liniensegment $\boldsymbol{L}_{k,\mathrm{R}'}$ ergeben sich daher zwei korrespondierende `1D-Posen` $\boldsymbol{z}'_{\mathrm{Pose1D}}$ und $\boldsymbol{z}''_{\mathrm{Pose1D}}$. Aus diesen wird diejenige `1D-Pose` $\boldsymbol{z}_{\mathrm{Pose1D},k}$ gewählt, deren Orientierung besser mit der Hypothese übereinstimmt.

Aus der Zuordnung $\boldsymbol{L}_{k,\mathrm{R}'} \rightarrow \boldsymbol{F}_{j,\mathrm{F}}$ wird diejenige Pose $\boldsymbol{z}_{\mathrm{Pose1D},k}$ abgeleitet, welche die Wahrnehmung der Feldlinie $\boldsymbol{F}_{j,\mathrm{F}}$ als $\boldsymbol{L}_{k,\mathrm{R}'}$ erklärt. Beispielhaft sei dieser Vorgang anhand einer einzelnen Linienmessung erläutert.
Die geraden Feldlinien des Fußballfeldes verlaufen stets entlang einer der beiden Koordinatenachsen. Je nach Alignierung werden daher nachfolgend Linien entlang der x- bzw. y-Achse als x- bzw. y-Feldlinien unterschieden. Betrachtet wird eine Linienmessung, die nach dem Vorgehen der vorangegangenen Ausführungen mit einer y-Feldlinie assoziiert wurde. Eine solche Situation ist in Abb. 4.5 dargestellt. Abstand a und Winkel φ des Roboters zur wahrgenommenen Linie sind unmittelbar bekannt. Weiterhin ist die x-Koordinate x_{fl} der assoziierten y-Feldlinie aus dem internen Modell der Umwelt – der `FieldInfo` – gegeben. Die Linienmessung kann dann durch jede Pose erklärt werden, deren Abstand und Winkel zur assoziierten y-Feldlinie dem Tupel (a, φ) entspricht. Da die Linienmessung ausschließlich eine Positionsbestimmung orthogonal zur assoziierten Feldlinie erlaubt, kann aus einer y-Feldlinie nicht die Position entlang der y-Achse bestimmt werden. Aus der Linienmessung wird daher keine vollständige `2D-Pose` abgeleitet, sondern im Fall der y-Feldlinie die um die y-Koordinate reduzierte `1D-Pose` $\boldsymbol{z}_{\mathrm{Pose1D}} = \begin{bmatrix} x_{\mathrm{Pose1D}} & \alpha_{\mathrm{Pose1D}} \end{bmatrix}^T$.

Ein analoges Vorgehen ergibt sich für Linien die mit einer x-Feldlinie mit y-Koordinate y_{fl} assoziiert wurden. Die allgemeine Berechnungsvorschrift zur Ableitung der `1D-Posen` aus einer Linienmessung ist durch

$$\boldsymbol{z}'_{\text{Pose1D}} = \begin{bmatrix} r_{\text{fl}} \\ \alpha_{\text{fl}} \end{bmatrix} - \begin{bmatrix} a \\ \varphi \end{bmatrix}, \tag{4.38}$$

$$\boldsymbol{z}''_{\text{Pose1D}} = \begin{bmatrix} r_{\text{fl}} \\ \alpha_{\text{fl}} \end{bmatrix} - \begin{bmatrix} -a \\ \varphi + \pi \end{bmatrix}, \tag{4.39}$$

$$\text{jeweils mit} \begin{cases} r_{\text{fl}} = x_{\text{fl}}, \alpha_{\text{fl}} = 0 & \text{für y-Feldlinien,} \\ r_{\text{fl}} = y_{\text{fl}}, \alpha_{\text{fl}} = \frac{\pi}{2} & \text{für x-Feldlinien} \end{cases}$$

gegeben. Es wird schließlich diejenige Pose $\boldsymbol{z}_{\text{Pose1D},k}$ aus $\{\boldsymbol{z}'_{\text{Pose1D}}, \boldsymbol{z}''_{\text{Pose1D}}\}$ gewählt, deren Orientierungsabweichung zur Hypothese geringer ausfällt.

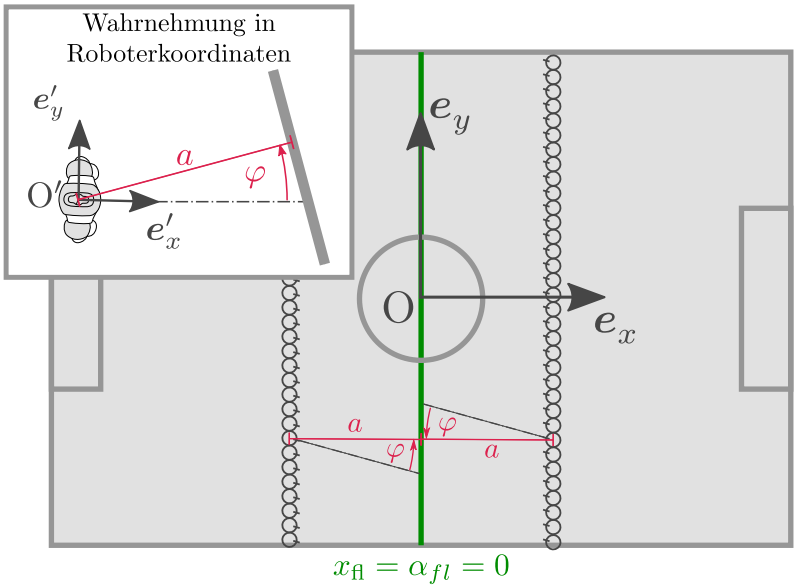

Abbildung 4.5: Berechnung der korrespondieren `1D-Posen` für eine Linienwahrnehmung bei Assoziation mit der Mittellinie (grün).

2. Mittelkreissegmente Liniensegmente, die mit dem Mittelkreispolygon des internen Modells assoziiert wurden, werden genutzt, um dessen Mittelpunkt $\boldsymbol{r}_{\text{O'C,R'}}$ zu approximieren. Diese Approximation orientiert sich an der grundlegenden Idee einer gradientenbasierten Hough-Transformation [25] und ist in Abb. 4.6 schematisch dargestellt.
Für jedes Liniensegment werden je zwei Punkte durch orthogonales Abtragen des bekannten Mittelkreisradius von der Linienmitte erzeugt. Die so erzeugte Punktmenge wird mittels einer rudimentären Clusteranalyse auf eine räumliche Häufung von Punkten untersucht. Wird auf diese

Weise ein ausreichend markantes Cluster entdeckt, erfolgt die Approximation des Kreismittelpunktes $\boldsymbol{r}_{\mathrm{O'C,R'}}$ durch Berechnung des Mittelwertes der diesem Cluster zugeordneten Punkte. Die Mittelkreisposition entspricht dann einer Messung $\boldsymbol{z}_{\mathrm{Position1D}} = \boldsymbol{r}_{\mathrm{O'C,R'}}$ nach dem Sensormodell des `1D-Position-Sensor` (vgl. Abschnitt 4.4.5).

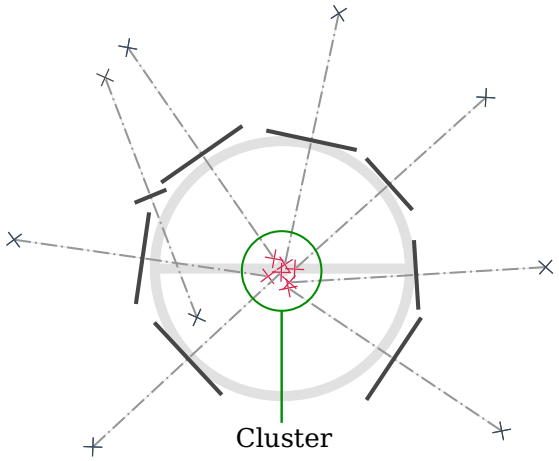

Abbildung 4.6: Mittelkreiserkennung auf Basis von Liniensegmenten in Anlehnung an eine gradientenbasierte Hough-Transformation.

UKF-Korrekturschritt Die abgeleiteten Messungen werden mit Hilfe der zugehörigen Sensormodelle (vgl. Abschnitt 4.4.5) durch einen Korrekturschritt in die Zustandsschätzung der Hypothese einbezogen.

1. **1D-Posen-Messungen** Die Messfunktion $\boldsymbol{h}_{\mathrm{Pose1D}}$ des `1D-Posen-Sensors` ist linear. Für eine lineare Messfunktion liefert die UT das gleiche Ergebnis wie die direkte lineare Transformation von Zustandsmittelwert und Kovarianz. Da die Berechnungskomplexität des klassischen Kalman-Korrekturschrittes geringer ausfällt als die des UKF, wird der Korrekturschritt nach dem Vorgehen des KF in Abschnitt 2.3.2 durchgeführt.
   Das Berechnungsvorgehen zur Korrektur der Zustandsschätzung mit einer Messung $\boldsymbol{z}_{\mathrm{Pose1D},k}$ folgt somit vollständig Gl. (2.6) bis (2.12) mit eingesetzter Messfunktion $\boldsymbol{h}_{\mathrm{Pose1D}}$.

2. **1D-Positions-Sensor** Das Sensormodell des `1D-Positions-Sensors` beinhaltet eine nichtlineare Messfunktion $\boldsymbol{h}_{\mathrm{Position1D}}$. Die Korrektur der Zustandsschätzung mit einer so modellierten Messung erfordert daher die Ver-

wendung der UT. Der Korrekturschritt folgt damit zu

$$\bar{\boldsymbol{\zeta}}_t = \boldsymbol{h}_{\text{Position1D}}\left(\bar{\boldsymbol{\chi}}_t\right), \tag{4.40}$$

$$\bar{\boldsymbol{z}}_t = \frac{1}{7}\sum_{i=0}^{6} \bar{\boldsymbol{\zeta}}_{i,t}, \tag{4.41}$$

$$\bar{\boldsymbol{P}}_{\text{z},t} = \frac{1}{2}\sum_{i=0}^{6} \left(\bar{\boldsymbol{\zeta}}_{i,t} - \bar{\boldsymbol{z}}_t\right)\left(\bar{\boldsymbol{\zeta}}_{i,t} - \bar{\boldsymbol{z}}_t\right)^T + \boldsymbol{R}_t, \tag{4.42}$$

$$\bar{\boldsymbol{P}}_{\text{xz},t} = \frac{1}{2}\sum_{i=0}^{6} \left(\bar{\boldsymbol{\chi}}_{i,t} - \bar{\boldsymbol{x}}_t\right)\left(\bar{\boldsymbol{\zeta}}_{i,t} - \bar{\boldsymbol{z}}_t\right)^T, \tag{4.43}$$

$$\boldsymbol{K}_t = \bar{\boldsymbol{P}}_{\text{xz},t}\bar{\boldsymbol{P}}_{\text{z},t}^{-1}, \tag{4.44}$$

$$\boldsymbol{x}_t = \bar{\boldsymbol{x}}_t + \boldsymbol{K}_t\left(\boldsymbol{z}_{\text{Position1D}} - \bar{\boldsymbol{z}}_t\right), \tag{4.45}$$

$$\boldsymbol{P}_t = \bar{\boldsymbol{P}}_t - \boldsymbol{K}_t\bar{\boldsymbol{P}}_{\text{z},t}\boldsymbol{K}_t^T. \tag{4.46}$$

Die Modellierung des Messrauschens $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t)$ als additives Rauschen erlaubt darin die Adaption der UT nach Abschnitt 4.4.4 sowie die nachträgliche Addition der Rauschkomponente in Gl. (4.42) ohne Erweiterung des Zustandsraumes.

Hypothesen-Reduktion

Durch den vorangegangenen Korrekturschritt kann es dazu kommen, das mehrere `Positionshypothesen` gegen die gleiche `2D-Pose` konvergieren. Für nahezu identische Hypothesen ist eine separate Betrachtung nicht mehr sinnvoll, da dies insbesondere einen erhöhten Rechenaufwand bedeutet. Es muss also zugelassen werden, dass hinreichend ähnliche Hypothesen zu einer einzelnen zusammengefasst werden.
Hypothesen die hinreichend kleine Winkeldifferenzen, sowie euklidische Abstände zueinander aufweisen, werden zu einer Menge ähnlicher Hypothesen zusammengefasst. Aus dieser Menge wird die `Positionshypothese` mit dem kleinsten Evaluationsfehler ausgewählt. Die `2D-Pose` dieser `Positionshypothese` wird mit den Zuständen der verbleibenden Hypothesen in ihrer Menge korrigiert. Dazu wird jeweils ein KF-Korrekturschritt nach Gl. (2.6) bis Gl. (2.12) durchgeführt. Als Sensormodell dient hier der lineare `2D-Posen-Sensor` mit Messfunktion $\boldsymbol{h}_{\text{Pose2D}}$, wie sie in Abschnitt 4.4.5 spezifiziert wurde.

Hypothesen-Generierung

Die Wahrnehmung eines Tores besteht wie in Abschnitt 3.2.4 aus den Positionen der beiden Torpfosten in `Roboterkoordinaten`. Kann das wahrgenommene Tor einer der beiden Tore aus dem internen Weltmodell zugeordnet werden, so gibt es eine eindeutige `2D-Pose`, die diese Wahrnehmung erklärt.
Auf Grund dieser Tatsache bietet es sich an, diese Informationen zur Erzeugung neuer Hypothesen zu nutzen. Nachfolgend wird zunächst beschrieben, wie aus einer Tormessung die korrespondierende `2D-Pose` abgeleitet wird. Anschließend wird der Vorgang der Hypothesen-Generierung genauer erläutert.

Ableitung einer 2D-Pose aus einer Tormessung Die Ableitung einer `2D-Pose` aus einer Tormessung erfolgt in den folgenden zwei Schritten:

1. **Ableitung einer Posenmessung** Das Tor wird zunächst einem beliebigen der beiden Tore des internen Weltmodells zugeordnet. Unter der Annahme, dass das Tor zu keinem Zeitpunkt von der Rückseite wahrgenommen wird, ist bekannt, welcher der beiden Torpfosten der linke bzw. der rechte ist. Die korrespondierende `2D-Pose` $(\boldsymbol{r}_{\mathrm{OO'},\mathrm{F}}, \boldsymbol{S}_{\mathrm{FR'}})$ wird damit aus der Tormessung wie folgt abgeleitet

$$\boldsymbol{r}_{\mathrm{RL,R'}} = \boldsymbol{r}_{\mathrm{O'L,R'}} - \boldsymbol{r}_{\mathrm{O'R,R'}} = \begin{bmatrix} x_{\mathrm{RL,R'}} \\ y_{\mathrm{RL,R'}} \end{bmatrix}, \tag{4.47}$$

$$\boldsymbol{r}_{\mathrm{O'T,R'}} = \frac{\boldsymbol{r}_{\mathrm{OL,R'}} + \boldsymbol{r}_{\mathrm{OR,R'}}}{2} = \begin{bmatrix} x_{\mathrm{O'T,R'}} \\ y_{\mathrm{O'T,R'}} \end{bmatrix}, \tag{4.48}$$

$$\alpha_{\mathrm{FR'}} = -\arctan \frac{x_{\mathrm{RL,R'}}}{y_{\mathrm{RL,R'}}}, \tag{4.49}$$

$$\boldsymbol{S}_{FR'} = \begin{bmatrix} \cos\alpha_{\mathrm{FR'}} & -\sin\alpha_{\mathrm{FR'}} \\ \sin\alpha_{\mathrm{FR'}} & \cos\alpha_{\mathrm{FR'}} \end{bmatrix}, \tag{4.50}$$

$$\boldsymbol{r}_{\mathrm{OO'},\mathrm{F}} = \boldsymbol{r}_{\mathrm{OT},\mathrm{F}} + \boldsymbol{S}_{\mathrm{FR'}}(-\boldsymbol{r}_{\mathrm{O'T,R'}}). \tag{4.51}$$

   Dabei bezeichnen $\boldsymbol{r}_{\mathrm{O'L,R'}}$ bzw. $\boldsymbol{r}_{\mathrm{O'R,R'}}$ die Positionen des linken bzw. rechten Torpfostens in `Roboterkoordinaten` und gehen aus der Tormessung hervor. Der Positionsvektor der Tormitte in `Feldkoordinaten` wird mit $\boldsymbol{r}_{\mathrm{OT},\mathrm{F}}$ bezeichnet und ist aus der `FieldInfo` bekannt. Aus den berechneten Größen ergibt sich eine `2D-Pose` $(\boldsymbol{r}_{\mathrm{OO'},\mathrm{F}}, \alpha_{\mathrm{FR'}})$.

2. **Optionale Punktspiegelung** Zu der zuvor berechneten `2D-Pose` $(\boldsymbol{r}_{\mathrm{OO'},\mathrm{F}}, \alpha_{\mathrm{FR'}})$ wird die punktsymmetrische Spiegelung $(-\boldsymbol{r}_{\mathrm{OO'},\mathrm{F}}, \alpha_{\mathrm{FR'}} + \pi)$ erzeugt. Schließlich wird diejenige der beiden `2D-Posen` ausgewählt, die näher an der Zustandsschätzung der `Positionshypothese` mit dem kleinsten `Evaluationsfehler` liegt.

Erzeugung neuer Hypothesen Mit Hilfe der zuvor abgeleiteten `2D-Pose` wird eine neue `Positionshypothese` initialisiert. Dabei wird zusätzlich darauf geachtet, die Gesamtanzahl an Hypothesen und damit die Laufzeit des Algorithmus beschränkt zu halten. Sobald durch die Hypothesen-Generierung eine definierte Maximalanzahl an Hypothesen überschritten wird, wird stattdessen diejenige mit dem größten Evaluationsfehler verworfen.
Unterscheidet sich die so erzeugte `Positionshypothese` hinreichend von den bereits vorhandenen, so wird sie langfristig separat verfolgt. Liegt sie hinreichend nahe an einer bereits existierenden, werden diese Hypothesen im Rahmen der nächsten Hypothesen-Reduktion zusammengeführt. Diese Zusammenführung zweier Hypothesen verhält sich äquivalent zu einer direkten Korrektur der Zustandsschätzung mit der Tormessung. Auf diese Weise werden die Informationen aus der Tormessung in jedem Fall genutzt.

Hypothesen-Evaluation

Im Rahmen der Hypothesen-Evaluation werden die Hypothesen hinsichtlich ihrer Qualität bewertet, um schließlich die beste `Positionshypothese` als Zustandsschätzung der `2D-Pose` des Roboters zu verwenden. Nachfolgend wird zunächst die Konstruktion einer Evaluationsheuristik erläutert, um abschließend die darauf basierende Auswertung der Hypothesen im Detail zu beleuchten.

Konstruktion einer Evaluationsheuristik Die Evaluationsheuristik dient der Bewertung der Qualität einer `Positionshypothese`. Es kann erwartet werden, dass mit steigender Güte der Hypothese die zugehörigen Assoziationsfehler der Linien nach Gl. (4.37) im Allgemeinen kleiner werden. Entsprechend wird die Heuristik zur Bewertung der `Positionshypothese` auf Basis dieser Assoziationsfehler konstruiert.

Wie bereits beschrieben ist es möglich, dass die Assoziation einer Linienmessung mit einer Feldlinie fehlschlägt. Aus diesem Grund wird in der Evaluationsheuristik neben den Assoziationsfehlern zusätzlich der Anteil der assoziierbaren Linien berücksichtigt. Der ungefilterte Fehler

$$\bar{e}_{\mathrm{Hypo},t} = \kappa_{Err} \left(\frac{1}{m} \sum_{i=1}^{m} e_{\min}^{[0,j]} \right) + (1 - \kappa_{Err})(1 - \frac{m}{n}) \tag{4.52}$$

für n Linienmessungen, von denen m mit einer Feldlinie assoziiert werden konnten, ergibt sich daher als gewichtete Summe dieser beiden Komponenten. Darin bezeichnet $\kappa_{Err} \in [0, 1]$ den Anteil des mittleren Assoziationsfehlers und $e_{\min}^{[0,j]}$ den Assoziationsfehler der j-ten Linienmessung auf Basis des nullten Sigma-Punktes – dem Zustandsmittelwert der Hypothese. Die Berechnung der Assoziationsfehler $e_{\min}^{[0,j]}$ erfolgt entsprechend der vorangegangenen Ausführungen um Gl. (4.37).

Darüber hinaus muss davon ausgegangen werden, dass es zu vereinzelten Fehlwahrnehmungen von Linien kommen kann. Eine solche Fehlwahrnehmung entsteht beispielsweise, falls Liniensegmente in anderen Objekten auf dem Feld wahrgenommen werden, die nicht Teil des internen Umweltmodells sind. Um diesbezüglich die Fehlerdynamik zu verbessern wird der `Evaluationsfehler`

$$e_{\text{Hypo},t} = (1 - \gamma_{\text{err}})e_{\text{Hypo},t-1} + \gamma_{\text{err}}\bar{e}_{\text{Hypo},t} \tag{4.53}$$

als Tiefpassfilterung des Ergebnisses aus Gl. (4.52) berechnet. Die Dämpfung des Tiefpassfilters erster Ordnung wird mit $\gamma_{\text{err}} \in [0, 1]$ eingestellt. Auf diese Weise wächst der gefilterte `Evaluationsfehler` $e_{\text{Hypo},t}$ erst maßgeblich, wenn die Assoziation über mehrere `Cycle` hinweg einem großen Fehler unterliegt. Für kleine Werte von γ_{err} müssen Hypothesen umfassend evaluiert werden, damit der `Evaluationsfehler` maßgeblich wächst. Auf diese Weise wird der Einfluss einzelner Fehlwahrnehmungen gedämpft.

Evaluation der Hypothesen Zunächst wird die Validität der einzelnen `Positionshypothesen` auf Basis der externen Randbedingungen evaluiert. So können nach Abschnitt 3.2.5 beispielsweise Hypothesen, deren `2D-Pose` sich nicht auf dem Spielfeld befindet, als ungültig betrachtet werden. Diese als ungültig klassifizierten Hypothesen werden verworfen. Mit Hilfe der in Gl. (4.53) beschriebenen Evaluationsheuristik werden im nächsten Teilschritt alle verbleibenden `Positionshypothesen` bezüglich ihrer Qualität bewertet und die Hypothese mit dem kleinsten `Evaluationsfehler` zur Publikation ausgewählt. Abschließend werden alle `Positionshypothesen`, die nach

$$e^{[l]}_{\text{Hypo},t} > \kappa_{drop} e_{best} \wedge e^{[l]}_{\text{Hypo},t} > \delta_{\min} \tag{4.54}$$

ausreichend große `Evaluationsfehler` und insbesondere wesentlich größere als die der besten Hypothese aufweisen, verworfen. Darin beschreibt $e^{[l]}_{\text{Hypo},t}$, den `Evaluationsfehler` der l-ten `Positionshypothese`, e_{best} den `Evaluationsfehler` der besten Hypothese. Die Größen $\kappa_{\text{drop}} > 1$ und $\delta_{\min} > 0$ sind Schwellwerte zur relativen bzw. absoluten Klassifizierung. Die Kombination aus absoluter und relativer Bewertung ermöglicht eine Konfiguration der Hypothesen-Evaluation, die schlechtere als die beste Hypothesen erhält, falls sie selbst einen verschwindend kleinen Fehler aufweisen.

Kapitel 5

Evaluation

Im Rahmen der nachfolgenden Ausführungen werden die in Kapitel 4 vorgestellten Algorithmen hinsichtlich ihrer Leistungsfähigkeit zur Selbstlokalisierung auf dem Fußballfeld simulativ evaluiert. Untersucht werden dazu sowohl die Fehler der durch PF und UKF geschätzten `2D-Pose` des Roboters, als auch die Laufzeit der beiden Algorithmen. Dazu werden verschiedene Szenarien, sowohl mit unimodalen als auch mutlimodalen Anfangsbedingungen untersucht.

5.1 Die Simulationsumgebung

Zum Zweck der simulativen Evaluation der Algorithmen wird die Simulationsumgebung SimRobot eingesetzt. SimRobot ist ein quelloffenes Projekt der Universität Bremen in Zusammenarbeit mit dem Deutschen Forschungszentrum für Künstliche Intelligenz. Die Umgebung erlaubt die Simulation physischer, sensorbestückter Robotiksysteme in dreidimensionaler Umwelt [26].
In dieser Simulationsumgebung wird ein NAO (vgl. Abschnitt 3.1.1) auf einem Fußballfeld der Spezifikationen aus Abschnitt 3.2.1 simuliert. Auf dem simulierten Roboter kommen dabei die gleichen Algorithmen zum Einsatz wie auf dem realen System. Simuliert wird daher neben der vollständigen `Vision-Pipeline` auch die Dynamik des Roboters als Mehrkörpersystem in Folge der zweibeinigen Fortbewegung.

5.2 Verifikation des UKF

Zur Verifikation der im Rahmen dieser Arbeit implementierten UKF-Methode werden die einzelnen algorithmischen Schritte auf Basis sogenannter Unit-Tests überprüft. Jede dieser Testfunktionen definiert dabei im Voraus alle relevanten Eingangsgrößen, sowie das zu erwartende Berechnungsergebnis der zu testenden Funktion. Auf diese Weise wird z.B. die Korrektur des Zustandes mit den verschiedenen Sensormodellen (vgl. Abschnitt 4.4.6) getestet. Für jeden dieser Tests wird eine dem Modell entsprechende Messung vorgegeben und die Konvergenz des korrigierten Zustandes gegen die zu erwartende `2D-Pose` überprüft.

5.3 Schätzgüte und Filterdynamik

Aufgabe der Algorithmen ist die präzise Schätzung der `2D-Pose` des Roboters. Als Hauptkriterium zur Bewertung der Leistungsfähigkeit wird daher zunächst der Fehler der Zustandsschätzungen durch PF und UKF untersucht. Beide Algorithmen operieren dabei auf Basis identischer Eingangsdaten. Als wahrer Zustand – auch Ground-Truth genannt – wird die `2D-Pose` des Roboters aus der Simulationsumgebung aufgezeichnet. Der Fehler der Zustandsschätzung gegenüber der wahren `2D-Pose` des Roboters wird in einen euklidischen Positionsfehler e_{p} sowie einen Orientierungsfehler e_{α} unterteilt.

Zur Untersuchung dieser Fehlerkomponenten werden in Abschnitt 5.3.1 zunächst Szenarien mit eindeutiger (unimodaler) Anfangsbedingung evaluiert. Der Fokus dieser Untersuchungen liegt dabei auf der Genauigkeit, mit der die zeitliche Entwicklung des wahren Zustandes geschätzt werden kann.
In Abschnitt 5.3.2 wird dann ein Szenario untersucht, bei dem die Anfangsbedingung multimodal ausfällt. Fokus dieser Untersuchung liegt weniger auf der Quantität des Schätzfehlers, sondern vielmehr auf der Fähigkeit der Algorithmen, überhaupt zum wahren Zustand zu konvergieren.

5.3.1 Unimodale Anfangsverteilungen

Die beiden Algorithmen werden zu Beginn mit einer Normalverteilung um den wahren Zustand initialisiert. Der simulierte Roboter startet jeweils an Punkt A und läuft auf Basis der Ground-Truth-Daten eine definierte Strecke zum Ziel B ab. Es werden zwei Szenarien im Bereich unterschiedlicher Feldmarkierungen analysiert.

Feldüberquerung in Strafraumnähe

Im Rahmen dieses Szenarios überquert der Roboter das Feld im Bereich des Strafraums. Ein Versuchsdurchlauf ist exemplarisch in Abb. 5.1 dargestellt. Die zeitliche Entwicklung von Positions- und Orientierungsfehler für diesen Versuchsdurchlauf ist in Abb. 5.2 abgebildet. Es ist ersichtlich, dass die Schätzungen beider Algorithmen dem wahren Zustand im Allgemeinen mit kleinem Fehler folgen können. Tabelle 5.1 zeigt die mittleren Positions- und Orientierungsfehler für UKF und PF.

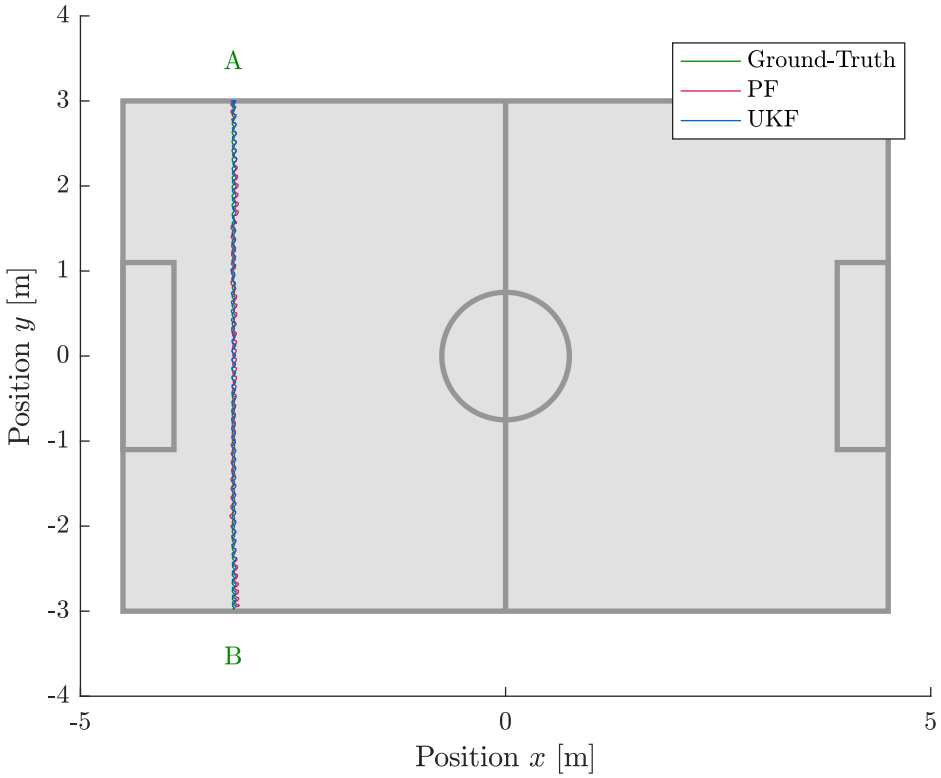

Abbildung 5.1: Wahrer Positionsverlauf und Positionsschätzungen der Filtermethoden für eine Feldüberquerung in Strafraumnähe mit unimodaler Anfangsbedingung.

Zunächst ist festzustellen, dass sowohl Positionsfehler als auch Orientierungsfehler von PF und UKF periodische Anteile enthalten. Die Frequenz dieser Fehler entspricht dabei der doppelten Schrittfrequenz des Roboters. Auch ist ersichtlich, dass diese Fehlerkomponenten im Stand des Roboters (Ende des Experiments) nicht auftreten.
Die periodischen Anteile des Fehlers sind daher auf die Tatsache zurückzuführen,

dass die Modellierung der Systemdynamik (vgl. Abschnitt 4.2) gewisse Anteile der Torsobewegung nicht berücksichtigt, da die `Odometrie` der Oberkörperbewegung nur bedingt folgen kann.

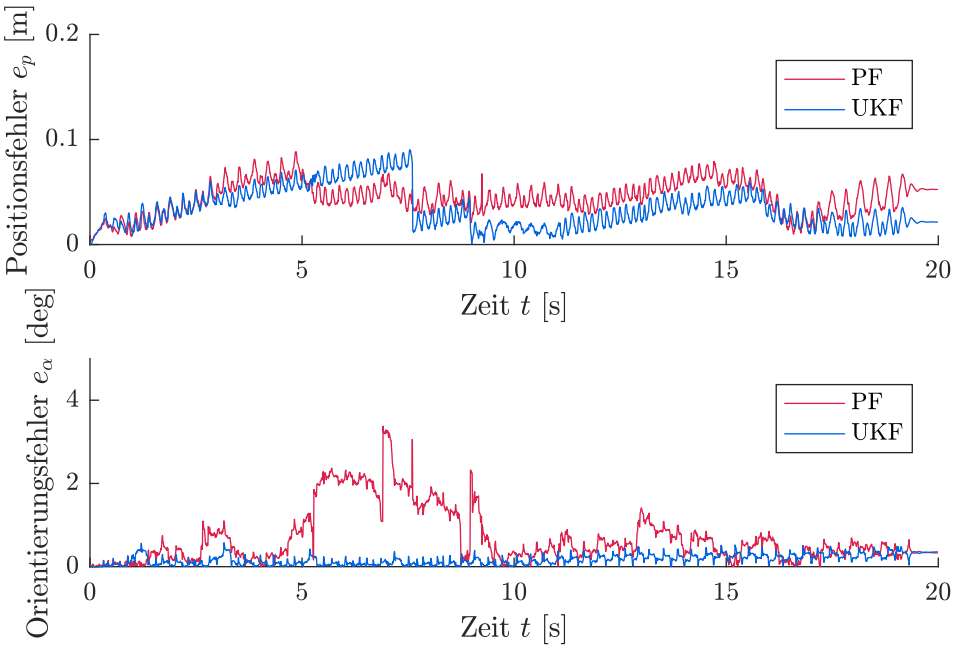

Abbildung 5.2: Positions- und Orientierungsfehler für den Versuch in Abb. 5.1.

Strukturell ist ersichtlich, dass PF und UKF eine ähnlich gute Schätzung der `2D-Pose` des Roboters abgeben. So wird in Tab. 5.1 deutlich, dass der mittlere Positionsfehler des UKF nur knapp unterhalb dem des PF liegt. Für die Schätzung der Orientierung zeigt sich allerdings eine klare Überlegenheit des UKF. Besonders im Bereich des Tores (bei $t = 5\,\mathrm{s}$ bis $t = 10\,\mathrm{s}$) fällt der Fehler der PF-Schätzung deutlich höher aus. Die Ursache für diese Abweichung kann Abb. 5.3 entnommen werden. Darin sind die Linienmessungen (schwarz) anhand der Ground-Truth-Pose in `Feldkoordinaten` transformiert dargestellt. Ein wahrgenommener Torpfosten ist rot markiert. Die weißen Linien entsprechen denjenigen Linien, die Teil des internen Modells der Umwelt sind. Hierin wird deutlich, dass es im hinteren Bereich des Tores zum Teil zu erheblichen Fehlwahrnehmungen kommt, die die Zustandsschätzung stören.

Zusammenfassend kann festgestellt werden, dass das UKF in diesem Szenario zwar dominiert, jedoch die Ergebnisse des PF in einem Präzisionsbereich liegen, der für die üblichen Anwendungen auf dem Fußballfeld ausreichend ist.

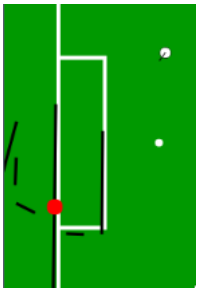

Abbildung 5.3: Exemplarische Wahrnehmung des Strafraums mit wahrer Position der Feldlinien (weiß) und wahrgenommen Liniensegmenten (schwarz).

Tabelle 5.1: Mittlere Fehler für den Versuch in Abb. 5.1.

	Positionsfehler [m]	Orientierungsfehler [deg]
PF	0,0447	0,6946
UKF	0,0342	0,1693

Feldüberquerung durch den Mittelkreis

Das Feld wird erneut von Punkt A zu Punkt B überquert, wobei im Rahmen dieses Versuches ein Umweg über die Feldmitte gewählt wird. Abbildung 5.4 zeigt exemplarisch einen Durchlauf eines solchen Versuches. Der zeitliche Verlauf der Fehler von PF und UKF ist in Abb. 5.5 dargestellt. Erneut wird ersichtlich, dass die Abweichungen der Zustandsschätzung beider Filter nur marginal ausfallen. Die Werte mittlerer Fehler sind in Tab. 5.2 dargestellt.

Die Ergebnisse fallen dabei vergleichbar zu denen des zuvor betrachteten Szenarios aus. So kann erneut beobachtet werden, dass die Orientierungsschätzung des UKF in den meisten Fällen mit einem kleineren Fehler behaftet ist, während die Positionsschätzung beider Filter vergleichbar gut ausfällt. Insbesondere wird ersichtlich, dass die Positionsschätzung beider Algorithmen im Bereich des Mittelkreises (im Zeitbereich um $t = 20\,\mathrm{s}$) sehr präzise ausfällt.

Die in Tab. 5.2 dargestellten mittleren Positions- und Orientierungsfehler zeigen ein ähnliches Ergebnis wie für das Szenario im Bereich des Strafraums.

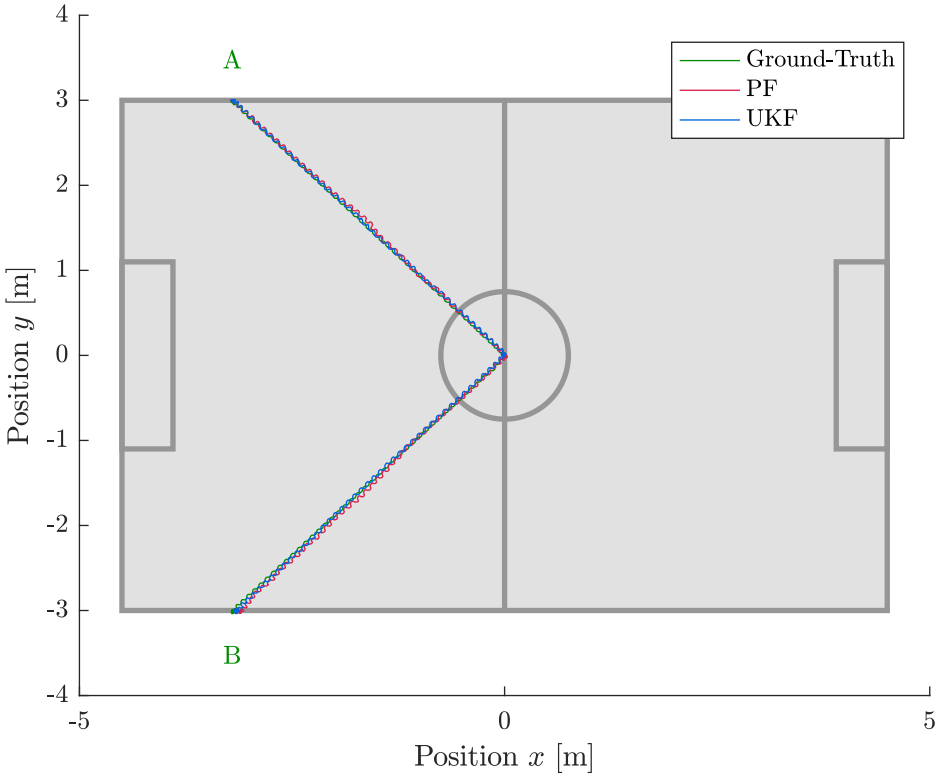

Abbildung 5.4: Wahrer Positionsverlauf und Positionsschätzungen der Filtermethoden für eine Feldüberquerung durch den Mittelkreis mit unimodaler Anfangsbedingung.

Tabelle 5.2: Mittlere Fehler für den Versuch in Abb. 5.4.

	Positionsfehler [m]	Orientierungsfehler [deg]
PF	0,0499	0,6501
UKF	0,0366	0,2806

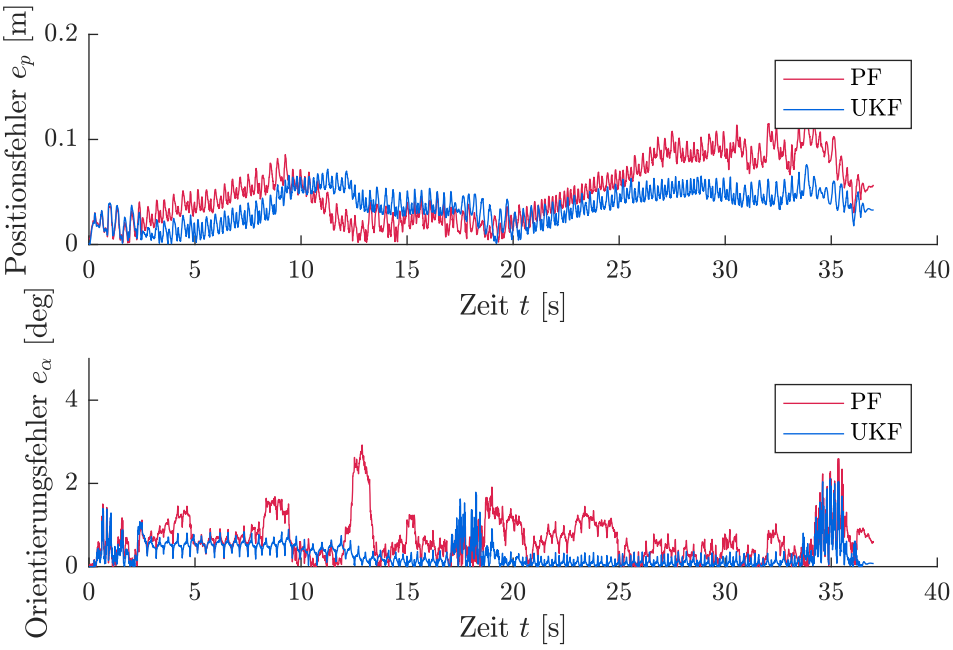

Abbildung 5.5: Positions- und Orientierungsfehler für den Versuch in Abb. 5.4.

5.3.2 Multimodale Anfangsverteilungen

Der Roboter überquert erneut das Feld von einem variierenden Punkt A über eine definierte Rute zu Punkt B. Im hier vorliegenden Szenario werden statt einer eindeutigen Anfangspose beliebige Posen am Feldrand der eigenen Hälfte mit Ausrichtung zur Feldinnenseite zugelassen. Entsprechend der Beschreibungen in Kapitel 4 wird die initiale Stichprobe des PF aus dieser Gleichverteilung generiert, während das UKF die Ausgangsverteilung durch acht Hypothesen approximiert.

Abbildung 5.6 zeigt einen Versuchsdurchlauf entsprechend des zuvor beschriebenen Szenarios. Darin sind neben den resultierenden Positionsschätzungen von PF und UKF auch die Pfade der einzelnen UKF-Hypothesen H1 bis H8 abgebildet. Das UKF wählt jeweils eine dieser Hypothesen auf Basis der Hypothesen-Evaluation zur Publikation aus. Wird eine Hypothese ausgeschlossen, bricht der entsprechende Pfad ab. Abbildung 5.7 zeigt die zeitliche Entwicklung von Positions- und Orientierungsfehler für diesen Versuch.

Zunächst ist ersichtlich, dass das PF sehr schnell zu einer guten Schätzung der `2D-Pose` gelangt. Das Cluster in der oberen Feldhälfte wird innerhalb von fünf Sekunden ausgewählt und die Schätzung konvergiert gegen die wahre Pose. Für das UKF dauert dieser Vorgang bedeutend länger. Zunächst wird Hypothese H8 als beste Schätzung ausgewählt. Nach kurzer Zeit widersprechen die Messungen den Hypothesen im unteren Teil des Feldes ausreichend, um diese ausschließen zu

können. Bei $t = 15\,\mathrm{s}$ Sekunden bleiben damit nur Hypothesen im oberen Bereich des Feldes erhalten und es wird zunächst Hypothese H4 als beste Schätzung ausgewählt. Schließlich kann auch diese `Positionshypothese` ausgeschlossen werden, sodass nach zwanzig Sekunden Hypothese H1 in unmittelbarer Nähe des wahren Zustandes ausgewählt wird. Ab diesem Zeitpunkt tritt der Mittelkreis in das Sichtfeld des Roboters, sodass sowohl PF und UKF gegen den wahren Zustand konvergieren. Dabei wird ersichtlich, dass Hypothese H1 und H2 gegen die gleiche Schätzung konvergieren und schließlich im Rahmen der Hypothesen-Reduktion (vgl. Abschnitt 4.4) zu einer einzelnen zusammengefasst werden.

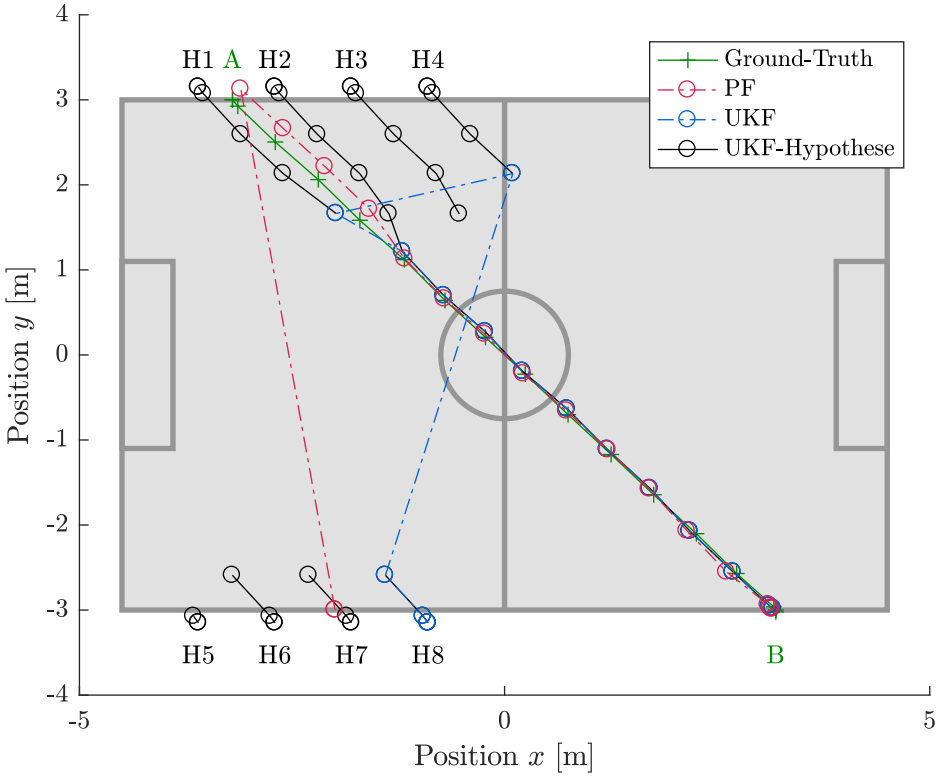

Abbildung 5.6: Positionsschätzungen für eine Feldüberquerung mit multimodaler Anfangsbedingung bei gelungener Konvergenz des PF.

Abbildung 5.7 zeigt, dass der Fehler des PF deutlich schneller kleiner wird, als der des UKF. So benötigt das UKF hier ca. 15 Sekunden länger, um eine ähnlich gute Schätzung wie das PF liefern zu können. Ab dem Zeitpunkt, ab dem durch das UKF die richtige Hypothese ausgewählt wurde, wird der Fehler der Schätzung schnell klein und es stellt sich ein ähnliches Verhalten der Fehler wie in Abschnitt 5.3.1 für beide Filter ein.

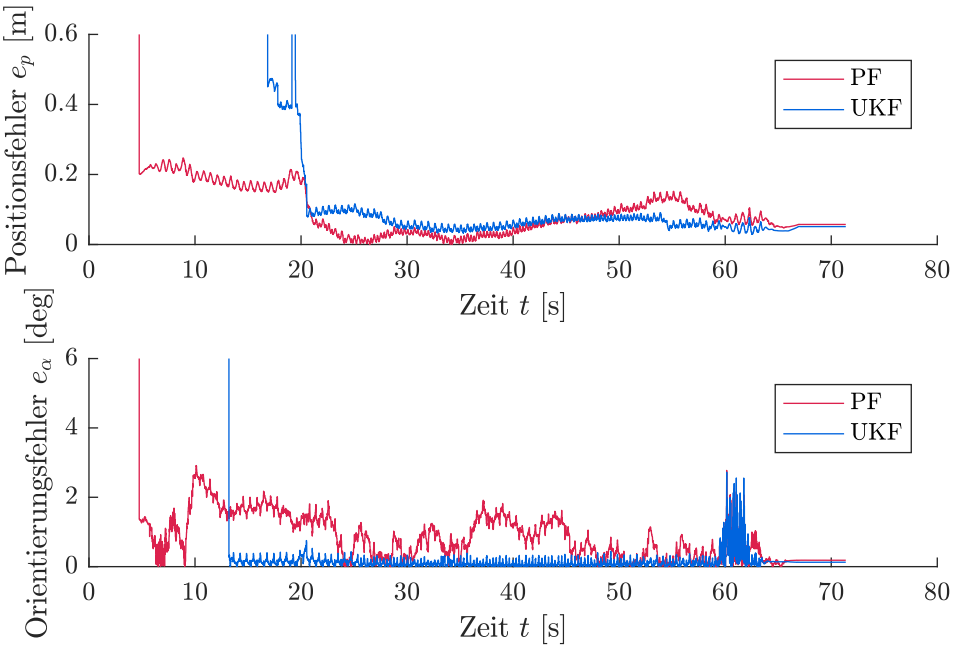

Abbildung 5.7: Positions- und Orientierungsfehler für den Versuch in Abb. 5.6.

Ein weiterer Fall des hier betrachteten Szenarios ist in Abb. 5.8 dargestellt. Zunächst ist zu beobachten, dass sich für das UKF eine vergleichbare Dynamik der Schätzung einstellt wie im zuvor betrachteten Fall. Nach kurzer Zeit werden alle Hypothesen bis auf H3 und H6 ausgeschlossen und die Hypothese in unmittelbarer Nähe des wahren Zustandes ausgewählt.
Die Zustandsschätzung des PF hingegen konvergiert nicht gegen den wahren Zustand. Dies ist dadurch zu erklären, dass nach einem Resampling kein Partikel im oberen Bereich des Feldes zurückgeblieben ist. Das Cluster in der Nähe des wahren Zustandes wird damit eliminiert und die Wahrnehmung aller weiteren Feldfeatures können nicht richtig zugeordnet werden. Eine Korrektur der Schätzung schlägt daher fehl. Das UKF hingegen verfolgt eine vergleichbare Hypothese zunächst auch, kann diese jedoch nach kurzer Zeit ausschließen. Insbesondere wird die Hypothese in der Nähe des wahren Zustandes nicht verworfen, sodass eine Konvergenz der Schätzung gewährleistet ist.

Strukturell ist zu beobachten, dass die Dynamik des PF dafür sorgt, dass einzelne Cluster schnell ausgeschlossen werden. Während dieses Verhalten für Fälle wie in Abb. 5.6 einen Vorteil darstellen, birgt diese schnelle Dynamik gleichzeitig die Gefahr, dass fälschlicherweise ein relevantes Cluster eliminiert wird und die Schätzung divergiert (vgl. Abb. 5.8). Diese Problematik des Verlusts relevanter Cluster wird in [5] diskutiert. Darin werden verschiedene Strategien erläutert, um das PF an Multi-Hypothesen-Problemstellungen anzupassen und so eine ausrei-

chende Beobachtung der einzelnen Moden sicherzustellen. Diese Strategien sind allerdings vornehmlich mit der Erhöhung der Partikelanzahl verbunden, um damit die Wahrscheinlichkeit eines kompletten Verlusts einzelner Hypothesen zu reduzieren. Eine Erhöhung der Partikelanzahl ist vor dem Hintergrund der damit verbundenen Laufzeiterhöhung jedoch nicht erstrebenswert.

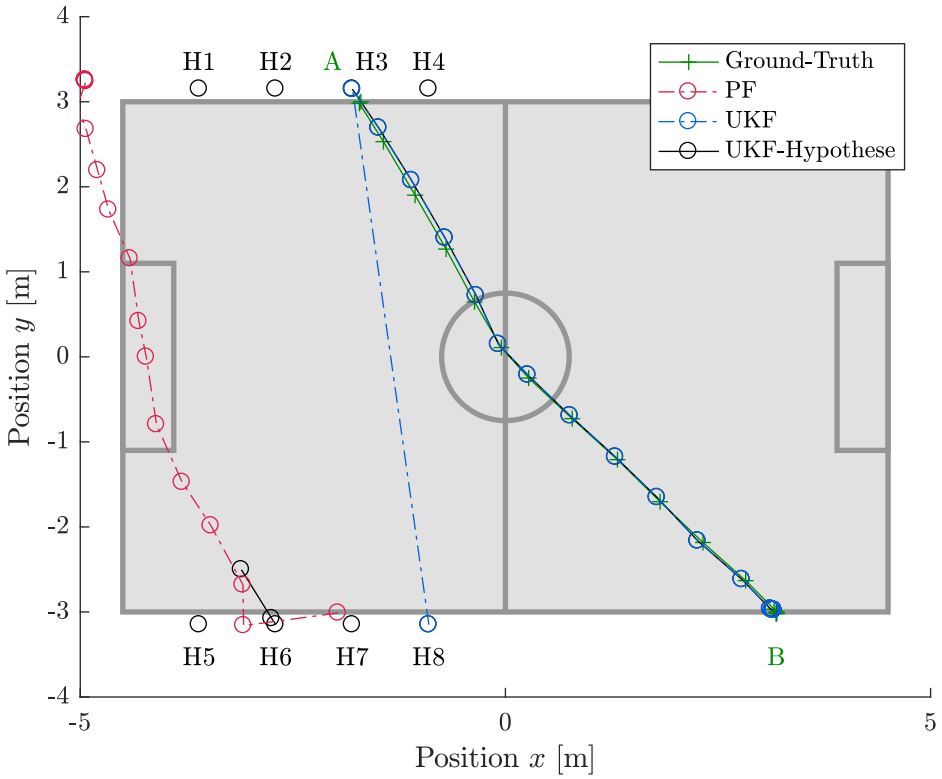

Abbildung 5.8: Positionsschätzungen für eine Feldüberquerung mit multimodaler Anfangsbedingung bei Divergenz des PF.

Für die Dynamik des UKF ist zu erkennen, dass die einzelnen Hypothesen länger verfolgt werden. Dieses Verhalten wird durch die Struktur der Evaluationsheuristik aus Gl. (4.53) erwirkt. Durch den Parameter γ_{err} kann die Dämpfung des `Evaluationsfehlers` darin explizit eingestellt werden. Auf diese Weise wird implizit festgelegt, wie ausgiebig die Hypothesen evaluiert werden müssen, bevor sie ausgeschlossen werden können. Eine starke Dämpfung der Evaluation sorgt allerdings gleichzeitig dafür, dass es länger dauern kann, bis die richtige `Positionshypothese` ausgewählt wird. Der hier entwickelte UKF-Algorithmus stellt damit einen zusätzlichen Freiheitsgrad zur Einstellung der Filterdynamik bereit, mit dem ein problemspezifischer Kompromiss zwischen „Robustheit gegen Fehlwahrnehmungen“ und „Konvergenzgeschwindigeit der Hypothesenauswahl“ gewählt werden kann.

5.4 Laufzeit

Wie in Abschnitt 3.1.1 beschrieben, verfügt das NAO-Robotiksystem nur über sehr limitierte Hardwareressourcen. Um eine echtzeitfähige Anwendung der Algorithmen zu erlauben, kommt der Effizienz der Filtermethoden daher eine besondere Bedeutung zu. Aus diesem Grund wird als weiteres Kriterium zur Bewertung der Leistungsfähigkeit die Laufzeit im Simulator evaluiert.

Abb. 5.9 zeigt die Entwicklung der Laufzeiten von PF und UKF für den Versuch aus Abb. 5.6 im Verlauf der Zeit, sowie die statistische Verteilung als Histogramm. In Tab. 5.3 sind zentrale Kennzahlen dieser Verteilung festgehalten. Hier ist erkennbar, dass die Laufzeit des UKF im Allgemeinen niedriger ausfällt und insbesondere eine wesentlich geringere Varianz aufweist.
Für die Echtzeitfähigkeit eines Algorithmus ist vor allem die Maximallaufzeit von zentraler Bedeutung, da diese in jedem Berechnungszyklus für dessen Ausführung reserviert werden muss. In Tab. 5.3 ist klar erkennbar, dass das UKF in diesem Punkt eindeutig dominiert. Eine Analyse des Laufzeithistogramms aus Abb. 5.9 ergibt, dass die Laufzeit des PF in mehr als 27% der Zeit oberhalb der Maximallaufzeit des UKF liegt.

Zusammenfassend ist festzustellen, dass das UKF in allen Aspekten der laufzeitbasierten Bewertung eindeutig besser abschneidet. Die niedrige und stabile Laufzeit des Algorithmus bieten hervorragende Voraussetzung für die Anwendung im Kontext echtzeitrelevanter Problemstellungen.

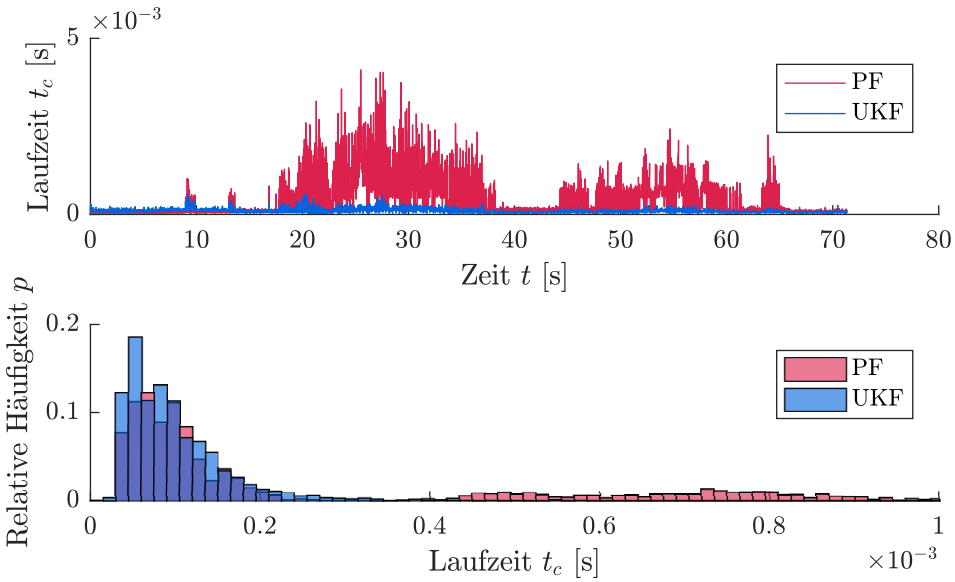

Abbildung 5.9: Laufzeitanalyse der Filteralgorithmen für den Versuch in Abb. 5.6.

Tabelle 5.3: Statistische Kennwerte der Laufzeitanalyse in Abb. 5.9.

	Mittelwert [ms]	Varianz [ns]	Maximum [ms]
PF	0,4186	316,48	4,1000
UKF	0,0969	3,4463	0,5824

5.5 Fazit

Die Ausführungen zu den theoretischen Grundlagen des PF und UKF (vgl. Kapitel 2) machen deutlich, dass das PF im Allgemeinen auf eine breitere Menge von Filterproblemen anwendbar ist und zunächst eine bessere Zustandsschätzung verspricht. Die Evaluation der beiden Methoden zeigt allerdings, dass das UKF für die hier vorliegende Problemstellung in vielen Fällen bessere Ergebnisse erzielt.

Die Ergebnisse aus Abschnitt 5.3 machen deutlich, dass bei eindeutiger Anfangsbedingung die Positionsschätzung beider Algorithmen vergleichbar gut ausfällt, das UKF jedoch eine deutlich bessere Schätzung der Orientierung erlaubt (vgl. Abschnitt 5.3.1). Für multimodale Anfangsbedingungen hat sich gezeigt, dass das PF in einigen Fällen schneller zu einer guten Zustandsschätzung gelangt. Dies ist allerdings auf einen schnelleren Ausschluss einzelner Moden zurückzuführen und birgt die Gefahr der fehlerhaften Eliminierung relevanter Cluster. Im Zweifelsfall kommt es damit zur Divergenz der Zustandsschätzung.

Das UKF hingegen benötigt in vielen Fällen mehr Berechnungszyklen, um eine geeignete Hypothese zur Zustandsschätzung auszuwählen, ist dafür allerdings robust gegen den fehlerhaften Ausschluss relevanter Hypothesen. Die Laufzeituntersuchungen in Abschnitt 5.4 zeigen überdies, dass die Berechnungskomplexität des UKF deutlich geringer ausfällt als die des PF.

Auf Basis dieser Evaluationsergebnisse kann daher zusammengefasst werden, dass das adaptierte UKF für den hier vorliegenden Anwendungsfall der geeignetere Lokalisierungsalgorithmus ist. Dies ist auf die Tatsache zurückzuführen, dass viele der theoretischen Vorteile des PF auf Grund der beschränkten Rechenkapazität nicht genutzt werden können. Insbesondere die vergleichsweise niedrige Anzahl an Partikeln führt bei der Schätzung des Zustandes zu Problemen. Gleichzeitig ist eine Erhöhung der Partikelanzahl vor dem Hintergrund der ohnehin schon höheren Berechnungskomplexität nicht praktikabel.
Es sei darauf hingewiesen, dass für die meisten Anwendungsfälle eine langsamere Auswahl der richtigen Hypothese akzeptiert werden kann, falls dadurch langfristig eine robustere Schätzung ermöglicht wird.

Für Anwendungen im Kontext echtzeitkritischer Problemstellungen mit beschränkten Rechenkapazitäten kann eine UKF-Methode – wie die hier entwickelte – empfohlen werden. Neben den quantitativen Vorteilen, die aus der vorangegangenen Evaluation ersichtlich werden, bietet das UKF weitere Vorteile, die insbesondere im Kontext autonomer Anwendungen interessant sind. So ist die Varianz als Maß für die Sicherheit der Positionsschätzung direkter Teil des Berechnungsergebnisses des UKF. Informationen über die Sicherheit des aktuellen Zustandes sind für weitere Entscheidungsprozesse äußerst wertvoll und können im Berechnungsvorgehen nachgelagerter Algorithmen verwertet werden.

Kapitel 6

Zusammenfassung

Diese Arbeit vergleicht zwei nichtlineare Filtermethoden zur Selbstlokalisierung eines humanoiden NAO-Robotiksystems in bekannter Umwelt. Ausgangslage ist eine vorhandene Implementierung eines adaptierten Partikel-Filters (PF). Zum Vergleich wird im Rahmen dieser Arbeit eine nichtlineare Filtermethode auf Basis des Unscented Kalman-Filters (UKF) entwickelt.
Zu diesem Zweck werden zunächst die relevanten Grundlagen zur Funktionsweise und der theoretischen Leistungsfähigkeit der zugrundeliegenden Filtermethoden beleuchtet. Im weiteren Verlauf werden die Rahmenbedingungen für einen Lokalisierungsalgorithmus erläutert, die sich aus den Eigenschaften des Roboters und der Umwelt ergeben. Dazu wird als Anwendungsbeispiel die Selbstlokalisierung im Kontext der RoboCup Standard Platform League (SPL) gewählt. Auf Basis des Software-Frameworks des RoboCup SPL Teams HULKs wird erläutert, welche Informationen der NAO über sich und seine Umwelt zu Lokalisierungszwecken extrahieren kann. Darauf aufbauend wird gezeigt, dass auf Grund der erläuterten Rahmenbedingungen für die Wahrscheinlichkeitsverteilung der Roboterposition Multimodalität zu erwarten ist. Die Arbeit stellt anschließend die vorhandene Implementierung des PF vor und erläutert die darin getroffenen Adaptionen, die eine Anwendung des allgemeinen PF-Algorithmus unter den erläuterten Rahmenbedingungen ermöglichen. Auf Basis des UKF wird dann ein alternativer Lokalisierungsalgorithmus entwickelt. Es werden fundamentale Adaptionen des UKF erläutert, die die Anwendung des klassischen Algorithmus auch im Kontext multimodaler Zustandsverteilungen erlauben. Zu diesem Zweck wird eine Multi-Hypothesen-Strategie entworfen, die eine effiziente Beobachtung relevanter Moden der Verteilung ermöglicht. Zentrales Element dieser Methode ist eine Heuristik, auf Basis derer die Hypothesen bewertet und langfristig gegeneinander ausgeschlossen werden. Eine abschließende Evaluation der beiden Filteralgorithmen zeigt, dass die entwickelte UKF-Strategie – trotz der theoretischen Überlegenheit des PF – für das hier vorliegende Problem eine robustere Zustandsschätzung bei

gleichzeitig kleinerem Fehler sowie geringerer und stabilerer Laufzeit ermöglicht. Insbesondere die hier entworfene Bewertungsheuristik sorgt für eine robustere, wenngleich etwas langsamere Dynamik der Multi-Hypothesen-Strategie bezüglich Auswahl und Ausschluss der Hypothesen.

Zusammenfassend kann bilanziert werden, dass das PF im Kontext dieser Anwendung unterliegt. Die im Rahmen dieser Arbeit entwickelte UKF-Methode stellt für diese und vergleichbare Anwendungen mit echtzeitkritischem, multimodalem Charakter bei eingeschränkter Rechenkapazität eine geeignete Filterstrategie dar.

Literaturverzeichnis

[1] Hiroaki Kitano: *RoboCup-97: Robot Soccer World Cup I.* Lecture Notes in Computer Science. Springer, 1998.

[2] Guna Seetharaman, Arun Lakhotia und Erik Philip Blasch: *Unmanned Vehicles Come of Age: The DARPA Grand Challenge.* Computer, 39(12), 2006.

[3] Sebastian Thrun, Fox Dieter, Wolfram Burgard und Frank Dellaert: *Robust Monte Carlo Localization for mobile Robots.* Artificial intelligence, 128(1-2):99–141, 2001.

[4] Jun S. Liu und Rong Chen: *Sequential Monte Carlo Methods for dynamic Systems.* Journal of the American statistical association, 93(443):1032–1044, 1998.

[5] Sebastian Thrun, Wolfram Burgard und Dieter Fox: *Probabilistic Robotics.* The MIT Press, 2005.

[6] Sebastian Thrun *et al.*: *Stanley: The Robot that won the DARPA Grand Challenge.* Journal of Field Robotics, 23(9):661–692, 2006.

[7] Tim Laue und Thomas Röfer: *Particle-Filter-Based State Estimation in a competitive and uncertain Environment.* In: *Proceedings of the 6th International Workshop on Embedded Systems.*, 2007.

[8] Rudolph Van Der Merwe: *Sigma-Point Kalman Filters for probabilistic Inference in dynamic State-Space Models.* 2004.

[9] Sebastian Thrun: *Particle Filters in Robotics.* In: *Proceedings of the 18th conference on Uncertainty in Artificial Intelligence*, Seiten 511–518. Morgan Kaufmann Publishers Inc., 2002.

[10] Rudolf Emil Kálmán *et al.*: *A new Approach to linear Filtering and Prediction Problems.* Journal of basic Engineering, 82(1):35–45, 1960.

[11] Simon J. Julier und Jeffrey K. Uhlmann: *A new extension of the Kalman Filter to nonlinear Systems.* In: *Proceedings of the 1995 American Control Conference*, Seite 1628–1632. Orlando, FL, 1995.

[12] Simon J. Julier: *The scaled Unscented Transformation.* In: *Proceedings of the American Control Conference*, Band 6, Seiten 4555–4559. IEEE, 2002.

[13] Simon J. Julier und Jeffrey K. Uhlmann: *Unscented Filtering and nonlinear Estimation.* Proceedings of the IEEE, 92(3):401–422, 2004.

[14] Eric A. Wan und Rudolph Van Der Merwe: *The Unscented Kalman Filter for nonlinear Estimation.* In: *Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Seiten 153–158. IEEE, 2000.

[15] SoftBank Robotics: *NAO - Technical Overview.* `http://doc.aldebaran.com/2-1/family/robots/index_robots.html`. Aufgerufen am 25. August 2017.

[16] RoboCup Technical Committee: *RoboCup Standard Platform League (NAO) rule book.* `http://www.tzi.de/spl/pub/Website/Downloads/Rules2017.pdf`, 2017. Aufgerufen am 26. August 2017.

[17] António J. R. Neves, Daniel A. Martins und Armando J. Pinho: *Obtaining the Distance Map for perspective Vision Systems.* In: *Proceedings of the EC-COMAS Thematic Conference on Computational Vision and Medical Image Processing*, 2009.

[18] Roberto G. Valenti, Ivan Dryanovski und Jizhong Xiao: *Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs.* Sensors, 15(8):19302–19330, 2015.

[19] S. Kolås, B. A. Foss und T. S. Schei: *Noise modeling concepts in nonlinear state estimation.* Journal of Process Control, 19(7):1111–1125, 2009.

[20] Anil K. Jain: *Data clustering: 50 years beyond K-means.* Pattern recognition letters, 31(8):651–666, 2010.

[21] Jeroen D. Hol, Thomas B. Schön und Fredrik Gustafsson: *On Resampling Algorithms for Particle Filters.* In: *Proceedings of the Nonlinear Statistical Signal Processing Workshop*, Seiten 79–82. IEEE, 2006.

[22] Ngai Ming Kwok, Gu Fang und Weizhen Zhou: *Evolutionary Particle Filter: Re-sampling from the Genetic Algorithm Perspective.* In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems.*, Seiten 2935–2940. IEEE, 2005.

[23] Gregor Jochmann, Sören Kerner, Stefan Tasse und Oliver Urbann: *Efficient Multi-Hypotheses Unscented Kalman Filtering for Robust Localization.* In: *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, Seiten 222–233. 2012.

[24] Daniel Alspach und Harold Sorenson: *Nonlinear Bayesian Estimation using Gaussian sum approximations.* IEEE transactions on automatic control, 17(4):439–448, 1972.

[25] Dimitrios Ioannou, Walter Huda und Andrew F. Laine: *Circle recognition through a 2D Hough transform and radius histogramming.* Image and vision computing, 17(1):15–26, 1999.

[26] Tim Laue und Thomas Röfer: *Simrobot - Development and Applications.* In: *Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2008.

Anhang

A.1 Inhalt Archiv

Im Archiv ist ein Verzeichnis **BSC_064_Peters/** abgelegt. Dieses enthält in der obersten Dateistruktur die Einträge

- **BSC_064_Peters.pdf**: das pdf-File zur Arbeit BSC-064.
- **Daten/**: ein Verzeichnis mit den für diese Arbeit relevanten Daten, Hilfsprogrammen, Skripts und Simulationsumgebungen.
- **Latex/**: ein Verzeichnis mit den *.tex-Dateien des in Latex verfassten Berichts zur Arbeit BSC-064 sowie alle dazugehörigen Grafiken (falls vorhanden auch als *.svg Dateien).

Erklärung

Ich, Lasse Peters (Student des Maschinenbaus an der Technischen Universität Hamburg-Harburg, Matrikelnummer 21486931), versichere, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

Unterschrift	Datum