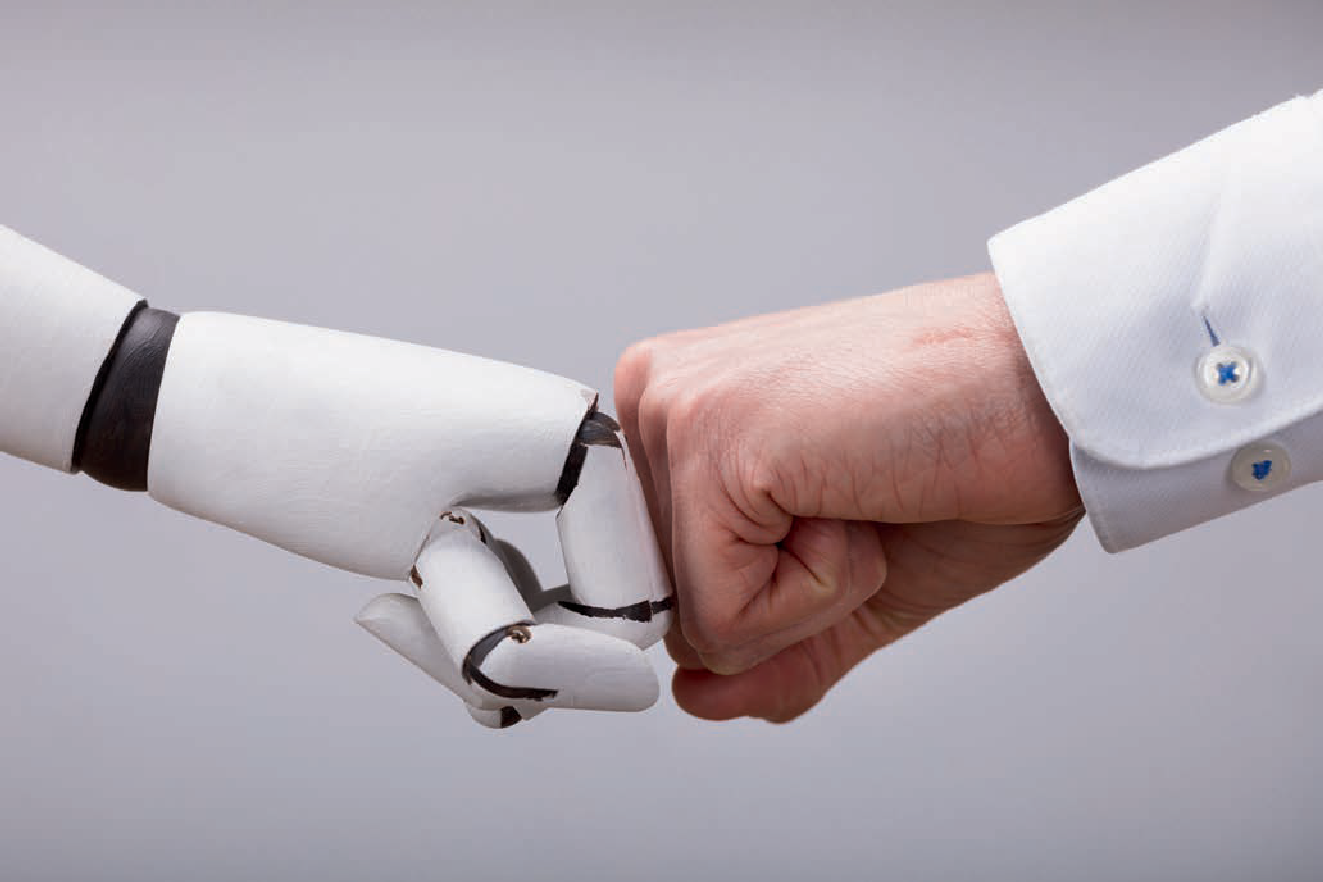
KURSBUCH



## Fallstudie: Lokalisierung, Bewegungs­planung und Sensordatenfusion

DLMDSEAAD02



Lernziele

##### Einleitung 9



Der Kurs „Fallstudie: Lokalisierung, Bewegungsplanung und Sensordatenfusion“ vermittelt die grundlegenden Konzepte und Methoden zur Lokalisierung, Bewegungsplanung und Sensordatenfusion für die mobile Robotik und selbstfahrende Autos. Mobile Roboter und autonome Fahrzeuge sind darauf angewiesen, ihre Umgebung wahrzunehmen und auf deren dynamische Veränderungen zu reagieren.

Der erste Teil des Kurses befasst sich mit der Darstellung von Bewegung und Navigation auf der Grundlage der Wegmessung (Odometrie), die aufgrund unsicherer Daten fehlerbehaftet sein kann. Eine mögliche Lösung bieten Lokalisierungsmethoden, die Odometrie und ergänzende Informationen wie z. B. ein GPS-Signal verwenden, um die Schätzung der Position des autonomen Fahrzeugs innerhalb eines Referenzrahmens zu verbessern. Auf diese Weise kann sich das Fahrzeug in Zielrichtung bewegen.

Die Probleme bei der Erkennung dynamischer Umgebungsveränderungen werden im letzten Teil des Kurses behandelt. Dort werden die Methoden der Sensordatenfusion vorgestellt. Dank der Verschmelzung mehrerer Datenquellen können Informationen extrahiert werden, z. B. ein sich näherndes Objekt oder eine veränderte Situation. Das autonome Fahrzeug muss in der Lage sein, das Objekt zu verfolgen und auf seine Bewegung zu reagieren, um Schäden und eine Gefährdung von Menschen zu vermeiden. Wie man die beste Bahnkurve (Trajektorie) bestimmt, wird ebenfalls im letzten Teil des Kurses behandelt.

Der Kurs gibt einen Überblick über die wichtigsten Methoden der Lokalisierung, Bewegungsplanung und Sensordatenfusion. Die Studierenden müssen die Konzepte und Methoden auf Fallstudien mit einem selbstfahrenden Fahrzeug anwenden, und zwar in zwei Hauptszenarien: „auf der Straße“ und in einer Fertigungsstätte.

[www.iubh.de](http://www.iubh.de/)



# Lektion 1

## Bewegung und Odometrie

#### LERNZIELE

Nach Abschluss dieser Lektion haben Sie gelernt, …

... was die grundlegenden physikalischen Prinzipien der Bewegung und die Beziehungen zwischen den Begriffen der Kinematik sind.

... welche verschiedenen Bewegungsmodelle es gibt, die für bewegliche Systeme verwendet werden können.

... wie Odometrie in der Navigation funktioniert und welche Arten möglicher Fehler auftreten können.

... was die Begriffe „Freiheitsgrad“ und „Mobilitätsgrad“ bedeuten.

... wie sich holonome und nichtholonome Bewegungen unterscheiden.

DL-E-DLMDSEAAD02-U01

1. Bewegung und Odometrie

### Einleitung

In diesem Abschnitt werden die Grundlagen der Bewegung und der Odometrie vorgestellt. Die Beziehungen zwischen Position, Geschwindigkeit und Beschleunigung (die physikalischen Prinzipien der Bewegung) sind vorgegeben. Sie sind für fast jedes automatisierte System unerlässlich und ermöglichen es uns, verschiedene Bewegungsmodelle abzuleiten, die zur Positionierung und Navigation verwendet werden können. Wir zeigen das Konzept der Starrkörperbewegung und die Beschreibungen der physikalischen Systeme mit ihren Freiheits- und Mobilitätsgraden, bis wir zur Definition der holonomen Bewegung kommen. Schließlich leiten wir einige mögliche Fehlerquellen ab, welche die Berechnungen verzerren können, und vergleichen die Auswirkungen von linearen Bewegungs- und Kursfehlern.

### Grundprinzipien

Es ist unerlässlich, die Grundlagen der Bewegungsphysik zu kennen und zu verstehen, wenn man die verschiedenen Methoden der Bewegung und Odometrie von automatisierten Systemen wie mobilen Robotern und automatisierten Fahrzeugen untersuchen möchte. Die Bewegung beruht auf den Begriffen Weg (Strecke), Zeit, Geschwindigkeit und Beschleunigung, sowie möglicherweise höheren Ableitungen des Wegs wie dem Ruck. Betrachten wir das folgende Beispiel: Ein mobiler Roboter bewegt sich mit der konstanten Geschwindigkeit v = 1 m/s während einer Zeitspanne von t = 10 s. Aus der grundlegenden Physik wissen wir, dass sich die in dieser Zeit zurückgelegte Wegstrecke s durch Multiplikation der Geschwindigkeit mit der Zeit wie folgt berechnen lässt:

s = v · t

(1.1)

In diesem Fall ergibt sich eine Wegstrecke von s = 10 m. Natürlich gilt diese Gleichung nur unter der Annahme einer konstanten Geschwindigkeit. Wenn wir zwei Größen kennen, können wir die dritte mit dieser Gleichung bestimmen. Wenn wir also eine zurückgelegte Strecke mit einem Lineal oder einem Maßband messen und die Dauer dieser Bewegung kennen, kann die Geschwindigkeit leicht durch Umstellen der Gleichung berechnet werden:

v = s

t

(1.2)

Dieses Konzept wird in der folgenden Abbildung veranschaulicht.



Wir berechnen nun die Differenz ∆si zwischen der Position xi+1 und xi wie folgt:

∆si = xi+ 1 – xi

(1.3)

Wenn die Zeit ti an jeder Position xi gemessen wird, ist die Zeitdifferenz ∆ti = ti+1 –ti die für die Bewegung von der Position xi nach xi+1 benötigte Zeitdauer. Die Geschwindigkeit im Streckenabschnitt si ist nun die Änderung der Wegstrecke ∆si während der Zeitspanne ∆ti:

v = ∆si i ∆ti

(1.4)

Die Annahme, die Geschwindigkeit in einem Streckenabschnitt si sei konstant, entspricht in den meisten Fällen nicht der Realität. Ein mobiles System muss beschleunigen, um vom Stillstand zu einer konstanten Geschwindigkeit zu gelangen, bzw. abbremsen, um von einer konstanten Geschwindigkeit zum Stillstand zu kommen. Die Beschleunigung ist die Änderung der Geschwindigkeit während einer bestimmten Zeitspanne und ist daher wie folgt definiert:

a = ∆vi,

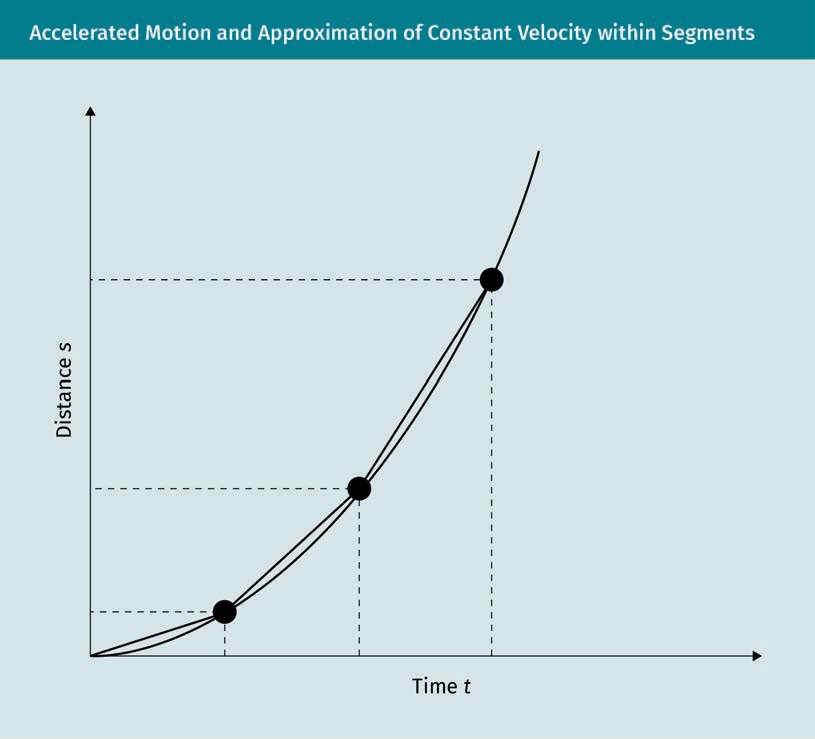
mit der Einheit m .

s2

i ∆ti

(1.5)

Betrachten wir nun als Beispiel die Beschleunigung eines sich bewegenden Objekts anhand des folgenden Diagramms, in dem die Wegstrecke als Funktion der Zeit aufgetragen ist.



Die durch jeweils zwei Punkte verbundenen Geraden zeigen die Änderung der Wegstrecke, wenn die Geschwindigkeit innerhalb der Abschnitte als konstant angenommen wird. Natürlich ist dies nur eine grobe Annäherung an die tatsächliche Beschleunigung. Was aber würde passieren, wenn wir kleinere Abschnitte wählten und sie gegen Null gehen ließen? In diesem Fall wäre die Momentangeschwindigkeit das Ergebnis der zeitlichen Ableitung der Wegstrecke:

v t = ds t

dt

(1.6)

Folglich kann die Beschleunigung anhand der zeitlichen Ableitung der Geschwindigkeit berechnet werden:

a t = dv t

dt

(1.7)

Damit ist sie auch die zweite Ableitung der Wegstrecke:

a t =

d2s t

dt2

(1.8)

Andererseits können wir die Geschwindigkeit bei konstanter Beschleunigung durch Integration über die Zeit berechnen:

v t = ∫a dt = a∫dt = at

(1.9)

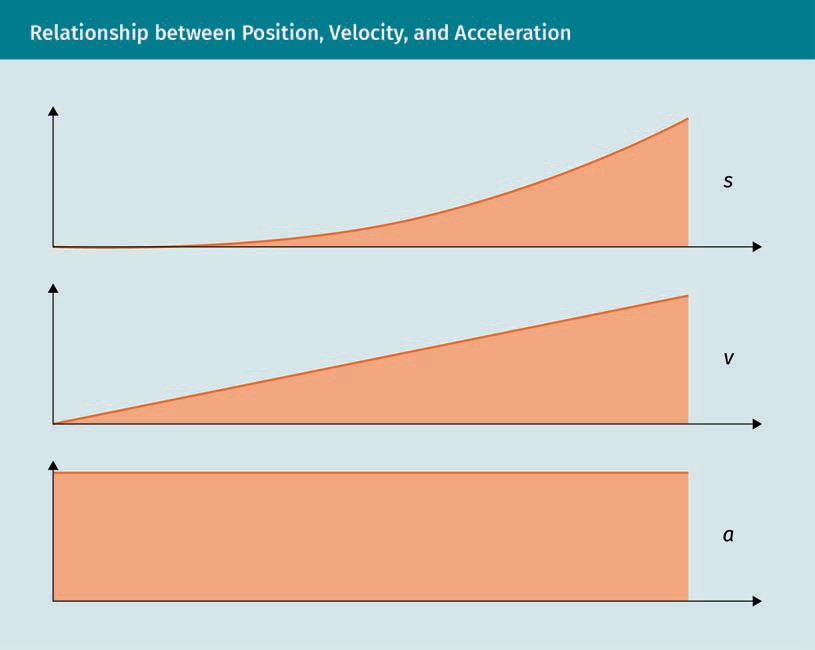
Die Wegstrecke ist das Ergebnis einer doppelten Integration:

s t = ∫v t dt = ∫ ∫a dt dt = ∫at dt = at2

2

(1.10)

Die folgende Abbildung veranschaulicht die Größen s(t) und v(t) , die sich aus der Integration der konstanten Beschleunigung ergeben.



Die Annahme einer konstanten Beschleunigung entspricht zwar oft nicht der Realität, liefert aber in den meisten Fällen einen geeigneten Näherungswert. Wenn Bewegungen mit hochdynamischem Verhalten berücksichtigt werden müssen, kann man auch Ableitungen höherer Ordnungen der Wegstrecke, wie etwa den Ruck, einbeziehen (Eager et al., 2016). So wie die Geschwindigkeit und die Beschleunigung die erste bzw. zweite Ableitung der Strecke sind, stellt der Ruck die dritte Ableitung dar:

j t =

d3s t

dt3

mit der Einheit m

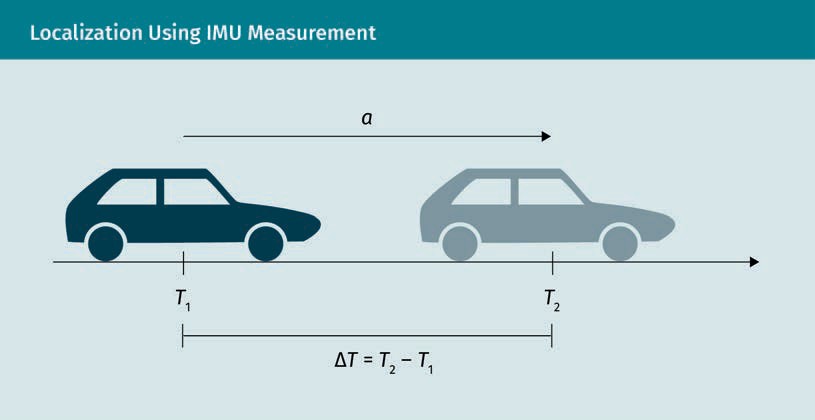
s3

### Bewegungsmodelle

(1.11)

Im Allgemeinen sind zwei verschiedene Bewegungsmodelle für automatisierte Systeme relevant: das odometrie- und das geschwindigkeitsbasierte Modell. Ersteres kann verwendet werden, wenn Radsensoren zur Verfügung stehen, und Letzteres, wenn diese nicht vorhanden sind. Nehmen wir an, unser Ego-System ist nicht mit Radsensoren ausgestattet, sondern mit einer inertialen Messeinheit (IMU von Engl. *inertial measurement unit*), die uns Informationen über die Beschleunigungen und die Winkelgeschwindigkeiten im dreidimensionalen euklidischen Raum liefert.

Wie würde man die aktuelle Position des Ego-Systems ermitteln, wenn man nur den Startpunkt kennt? Mit den Messungen der IMU können wir durch doppelte Integration der Beschleunigungen in jede Richtung den räumlichen Translationsvektor s = [sx sy sz]T sowie durch einfache Integration der Winkelgeschwindigkeit um jede Richtung eine Orientierungsänderung δ = [φ θ ψ]T berechnen. Dieser Prozess wird als Koppelnavigation oder Koppelung (Engl. *dead reckoning*) bezeichnet, wenn wir die Ego-Bewegung auf eine Dimension reduzieren und Orientierungsänderungen vernachlässigen. Dies kann wie in der folgenden Abbildung dargestellt werden.



Wenn zum Zeitpunkt T1 unsere Position x1 ist, und unsere IMU eine Beschleunigung a zwischen T1und T2 misst, können wir unsere Position bei T2 wie folgt bestimmen:

Dead reckoning

Der englische Begriff *dead reckoning* ist abgeleitet von *deduced reckoning*.

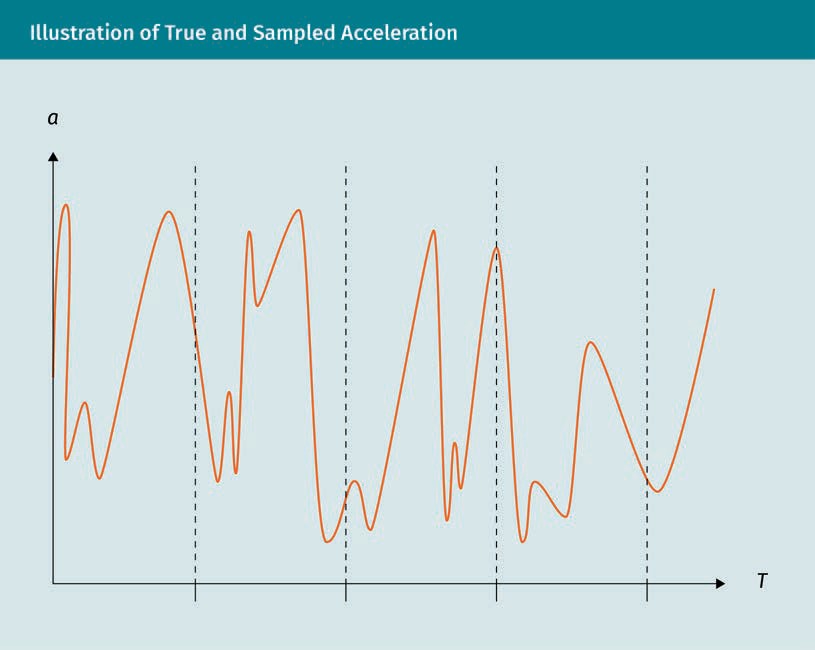
a 2

s2 = s1 + 2 ∆T

(1.12)

Messungen sind immer mit Unsicherheiten behaftet. In diesem Fall wird die gemessene Beschleunigung a durch zufällige Ruckbewegungen gestört, die zu Unterschieden in der Beschleunigung führen. Einer Messung liegt also eher eine Wahrscheinlichkeitsverteilung als ein bestimmter Wert zugrunde. Offensichtlich weist die auf Beschleunigungsmessungen basierende Ego-Positionsschätzung eine schwer zu identifizierende Abdrift auf, die den Einsatz einer IMU als einzigem Sensor unmöglich macht. Angenommen, wir messen eine Beschleunigung von 0,5 m/s² mit einer Unsicherheit von ±0,05 m/s². Wenn wir im schlimmsten Fall stets 0,45 m/ s² statt 0,5 m/s² messen, verlieren wir schnell unsere wirkliche Position. Bei einer Ausgangsposition von 0 und einem Messintervall von 0,1 s hätten wir nach 10 Sekunden eine Wegstrecke von 22,5 m statt 25 m geschätzt, was einer Abweichung von 2,5 m entspricht. Ohne externe Positionskorrektur wie z. B. über GPS ist eine Abweichung von der tatsächlichen Position unvermeidlich. Beschleunigungen können nur gemessen werden, wenn sich die Geschwindigkeit ändert. Aus diesem Grund funktioniert die Koppelnavigation nur, wenn das Ego-System seine Bewegung aus dem völligen Stillstand heraus beginnt oder wenn die Anfangsgeschwindigkeit gegeben ist. Bei einer konstanten Geschwindigkeit von 2 m/s beträgt die Beschleunigung 0 m/s², wie bei jeder anderen konstanten Geschwindigkeit auch.

Die Beschleunigung zwischen zwei Messungen wird meist als konstant angenommen. Daher ist es sehr wichtig, sehr hohe Messfrequenzen zu haben, um die dadurch verursachten Fehler so klein wie möglich zu halten. Die folgende Abbildung zeigt dieses Problem.



Die orangefarbene Kurve zeigt die tatsächliche Beschleunigung über die Zeit, und die gestrichelten Linien stellen die Messungen dar. In diesem Beispiel ist besonders im zweiten Abschnitt zu erkennen, dass die gemessene Beschleunigung von der tatsächlich in diesem Zeitraum aufgetretenen Beschleunigung abweicht. Daher muss die Abtastfrequenz so klein wie möglich sein, um die Beschleunigungsschwankungen genau zu erfassen.

Erweitert man die Bewegung auf den dreidimensionalen Raum, so ergibt sich die neue Position s2 = [sx2 sy2 sz2]T aus der vorherigen Position s1 = [sx1 sy1 sz1]T folgendermaßen:

sx2

sx1

1 0 0

cosθ 0 sinθ

cosψ −sinψ 0 ax

sy2 =

sy1 + ∫∫ 0 cosφ −sinφ ·

0 1 0

· sinψ cosψ 0 · ay dt2

sz2

sz1

0 sinφ cosφ

−sinθ 0 cosθ

0 0 1 az

(1.13)

Dabei sind die drei Matrizen sind die Rotationsmatrizen um die drei Achsen sind. Der Beschleunigungsvektor a = [ax ay az]T wird also gemäß den definierten Rotationsmatrizen gedreht, und der endgültige Translationsvektor wird durch eine doppelte Integration der gedrehten Beschleunigungen berechnet. Die gemessene Beschleunigung bezieht sich auf das Koordinatensystem, das mit dem bewegten Objekt verknüpft ist und dessen Ursprung im Zentrum des Objekts liegt. Da wir den Translationsvektor in Weltkoordinaten berechnen wollen, muss die Beschleunigung entsprechend gedreht werden, damit sie gültig ist.

Wenn ein System mit Radsensoren ausgestattet ist, kann die Positionsbestimmung mittels Odometrie erfolgen. Die Odometrie ist eine Methode zur Schätzung von Position und Orientierung anhand von Informationen aus dem Antriebssystem des Systems. Meistens werden die Räder oder Eigenschaften des Motors, wie z. B. die Motordrehzahl, für die Messung einer zurückgelegten Strecke verwendet. Allerdings können diese Eigenschaften durch eine hohe Nichtlinearität verfälscht werden (Ben-Ari & Mondada, 2017). Radsensoren zählen die Anzahl der Radumdrehungen zwischen zwei Messpunkten, aus denen mit Hilfe bekannter Radeigenschaften eine Wegstrecke berechnet wird. Diese Beziehung lässt sich sehr leicht herleiten. Der Umfang eines Rads mit dem Durchmesser d = 1 m wird bekanntlich durch U = π·d berechnet. Eine Radumdrehung ergibt also eine zurückgelegte Strecke von s = U = π·d. Also ergeben n Umdrehungen eine Wegstrecke von s = n·π·d. Natürlich erlaubt diese Annahme nur eine relative Positionsbestimmung in einer Richtung. Durch die Einbeziehung des Lenkwinkels der einzelnen Räder kann das Bewegungsmodell auf den zweidimensionalen Raum erweitert werden. Die Zuverlässigkeit dieses Verfahrens hängt von mehreren Fehlerquellen ab, z. B. von Fehlern in der Radgeometrie, dem Material, Luftdruckstößen auf der Straße und dem Fahrzeuggewicht. Störende Extremfälle bei diesem Ansatz sind z. B. Radschlupf aufgrund von Ausbrüchen, was zu einer Bewegungsmessung ohne echte Bewegung führt, oder das kurze Abheben eines Fahrzeugs, bei dem die Räder in der Luft durchdrehen.

### Navigation durch Odometrie

Die Bestimmung der Position eines bewegten Systems relativ zu einem gegebenen Ausgangspunkt in einer Dimension (entlang einer Linie) ist eine recht einfache Aufgabe. Natürlich hat dieses Szenario in der Praxis kaum eine Bedeutung. Wenn davon ausgegangen werden kann, dass das Gelände, in dem das System sich bewegt, flach ist und keine signifikanten Höhenänderungen aufweist, kann die Berechnung der Bewegungsparameter in zwei Dimensionen ausreichend sein. Bei der Aufzeichnung von Rad- oder Motorinformationen für eine Bestimmung der zurückgelegten Wegstrecke nur über Odometrie ist die Einbeziehung der Bewegung in z-Richtung nicht möglich, weil die Messungen diese nicht erfassen können. Daher müssten Messungen von inertialen Messeinheiten herangezogen werden.

Angenommen, wir wollen die Position eines sich bewegenden Systems in zwei Dimensionen mit Hilfe von Odometrie abschätzen. Die räumliche Lage (Pose) des Systems muss in jedem Zeitintervall berechnet werden. Sie wird durch das Tripel (x,y,θ) dargestellt. Hierbei sind x und y die x- bzw. y-Position und θ ist der Peilwinkel (der Kurs bzw. die Richtung, in die das System zeigt). Wenn eine Bewegung von einer Ausgangspose (x0, y0, θ0) in einer geraden Linie erfolgt, können die neuen Positionskomponenten wie folgt berechnet werden:

Rotationsmatrizen Reellwertige orthogonale Matrizen mit einer Determinante von 1. Eine Multiplikation mit einem Vektor kann als Drehung im Euklidischen Raum interpretiert werden.

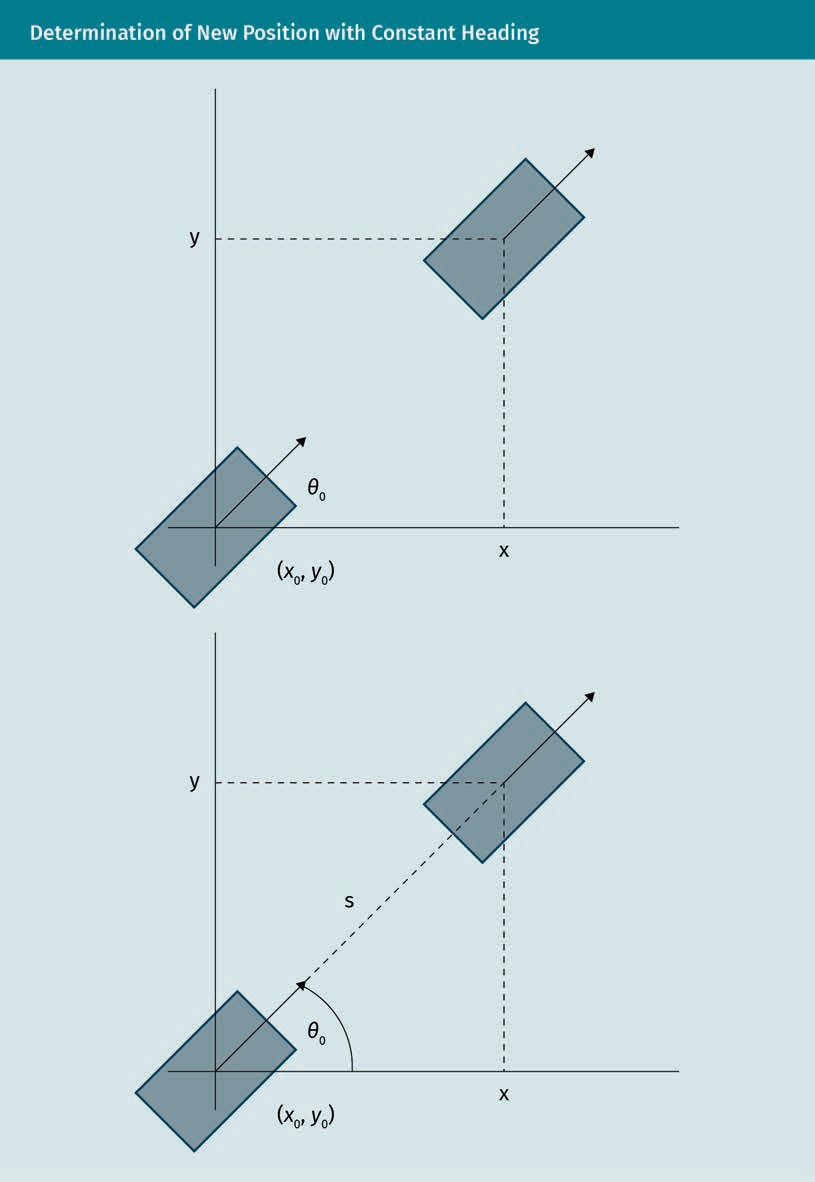
x = x0 + s · cosθ0

(1.14)

y = y0 + s · sinθ0

(1.15)

Diese Gleichungen lassen sich mit Hilfe einer einfachen geometrischen Überlegung herleiten, die in der folgenden Abbildung dargestellt ist.



Dies wird als Starrkörperbewegung bezeichnet und setzt voraus, dass sich der Körper des Systems unter den einwirkenden Kräften nicht verformt, was die Berechnungen vereinfacht. Wichtig für ein physisches System ist der durch die zugrundeliegenden Systemparameter definierte Konfigurationsraum. Im Falle der Bewegung eines starren Körpers in zwei Dimensionen ist der Konfigurationsraum Q das direkte Produkt aus einem zweidimensionalen euklidischen Vektor und der zweidimensionalen Rotationsmatrix, der sogenannten speziellen orthogonalen Gruppe:

Direktes Produkt Das direkte Produkt ist eine Verallgemeinerung des kartesischen Produkts für Mengen, Gruppen und andere algebraische Strukturen.

Q = (2 × SO 2 = x × cosθ −sinθ

y sinθ cosθ

(1.16)

Der euklidische Vektor erfasst die Positionskoordinaten des Systemursprungs, während die Rotationsmatrix den Kurs (Peilwinkel der Bewegungsrichtung) des Systems angibt. Der Konfigurationsraum des zweidimensionalen starren Körpers wird durch drei Parameter definiert. Man sagt, er hat drei Freiheitsgrade. Obwohl der Konfigurationsraum durch das direkte Produkt (² × SO(2)) deﬁniert ist, ist die homogene Transformationsmatrix, die sowohl Kopf als auch Position umfasst, das halbdirekte Produkt (² >< SO(2). Dies bedeutet, dass der euklidische Vektor auf die spezielle orthogonale Gruppe wirkt. Dies ist die Definition der speziellen Euklidischen Gruppe oder der Gruppe der Bewegungen starrer Körper, die durch die folgende Matrix dargestellt werden kann:

T = (2 >< SO 2 =

cosθ −sinθ x sinθ cosθ y 0 0 1

(1.17)

Jede Starrkörperbewegung kann durch eine solche Matrix dargestellt werden, und außerdem ist jede mögliche Pose im Konfigurationsraum ein Element dieser Gruppe. Darüber hinaus ergibt jede Gruppenoperation, also das Matrixprodukt zweier solcher Matrizen, ebenfalls eine Matrix:

R1 t1 0 1

R2 t2

· =

0 1

R1 · R2 R1 · t2 + t1 0 1

(1.18)

Hierbei sind Rk und tk die Rotationsmatrix bzw. der Translationsvektor.

Kehren wir zu unserem Beispiel der Bewegung auf einer geraden Linie zurück. Unter Verwendung homogener Transformationsmatrizen kann dies wie folgt formuliert werden:

cosθ0 −sinθ0 x0

1 0 s

cosθ0 −sinθ0 s · cosθ0 + x0

sinθ0 cosθ0 y0 · 0 1 0 = sinθ0 cosθ0 s · sinθ0 + y0

0 0 1

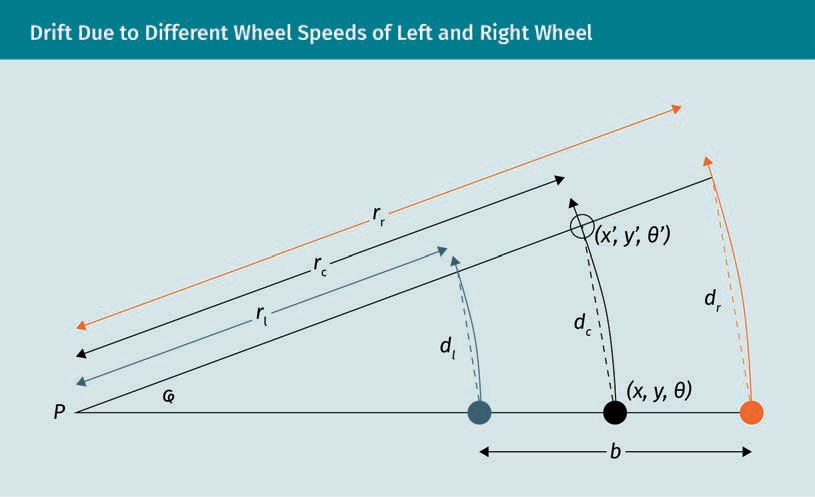
0 0 1

0 0 1

(1.19)

Dies ergibt die neue Pose des Systems, die der vorherigen Herleitung mittels geometrischer Beziehungen entspricht.

Bisher haben wir nur geradlinige Bewegungen ohne Kurswechsel berücksichtigt. Bewegt sich nun zum Beispiel das rechte Rad eines sich bewegenden Systems etwas schneller als das linke, driftet das System nach links, weil das rechte Rad mehr Umdrehungen gemacht und daher eine längere Strecke zurückgelegt hat. Die folgende Abbildung veranschaulicht eine solche Situation.



Hier zeigen die blauen, orangefarbenen und schwarzen Punkte die aktuelle Position des linken Rads, des rechten Rads bzw. der Mitte des sich bewegenden Systems. Die Mitte halbiert die Grundlinie b (die Strecke zwischen dem linken und dem rechten Rad). Die von den beiden Rädern und dem Mittelpunkt zurückgelegten Strecken werden durch dl*,* dr und dc dargestellt. Durch Messung der Anzahl der Umdrehungen pro Sekunde jedes Rads und Einbeziehung des Radius R (also dem halben Durchmesser) ergeben sich nach t Sekunden die zurückgelegten Strecken:

di = 2πRωit, mit i = l, r

(1.20)

Hierbei sind ωl und ωr die Winkelgeschwindigkeiten des linken bzw. rechten Rads. Wie lässt sich mit dieser Gleichung für jedes Rad die neue Lage des Systems nach t Sekunden bestimmen? Offensichtlich bewegen sich die Räder sowie der Mittelpunkt auf einem Bogen mit den Radien rl, rr und rc um einem momentanen Drehpunkt P. Die Länge eines Bogens mit dem Winkel φ rad lässt sich berechnen, indem man das Bogenmaß des Winkels mit dem Umfang des vorgestellten Kreises d = φ·r in Beziehung setzt. Die zurückgelegten Wegstrecken des linken Rads, des rechten Rads und des Systemmittelpunkts hängen also mit den Winkeln und Radien der Bögen wie folgt zusammen:

dl = φ · rl

(1.21)

dr = φ · rr

(1.22)

dc = φ · rc

(1.23)

Nun müssen wir den Winkel der Bögen berechnen. Zu diesem Zweck nehmen wir zwei der drei Gleichungen und lösen das folgende lineare Gleichungssystem:

dr = φ · rr

(1.24)

dl = φ · rl

(1.25)

Die Subtraktion von Gleichung 1.25 und 1.24 liefert:

dr − dl = φ · rr − φ · rl = φ · rr − rl

φ = dr − dl = dr − dl

(1.26)

rr − rl b

(1.27)

Wir wissen, dass der Mittelpunkt des Systems auf halbem Weg zwischen den Rädern liegt, also:

rr − rl .

=

r

c 2

Daraus können wir nun die vom Mittelpunkt des Systems zurückgelegte Strecke bestimmen:

dc = φ · rc

= φ · rr − rl

2

(1.28)

= φ · dr − dl

(1.29)

2 φ φ

(1.30)

= dr − dl

2

(1.31)

Wenn die Länge des Bogens dc klein ist, kann man ihn durch seine Tangente annähern, die senkrecht zum Radius verläuft. Das bedeutet, dass wir die räumlichen Translationen (Verschiebungen) dx und dy bei einer Bewegung entlang der x-Achse wie folgt berechnen können

dx = dc · cosφ

(1.32)

dy = dc · sinφ

(1.33)

Die neue räumliche Lage des Systems kann nun durch das Tripel (dc·cosφ, dc·sinφ, θ+φ) ausgedrückt werden. Diese Berechnung gilt aufgrund der Annäherung der Bögen durch ihre Tangenten und der Annahme konstanter Radgeschwindigkeiten nur für kurze Strecken. Daher müssen die Messschritte häufig durchgeführt werden, um zuverlässige Ergebnisse zu erhalten.

### Holonome und nichtholonome Bewegung

Bislang gingen wir davon aus, dass sich unser System ohne Einschränkungen in jede Richtung bewegen kann. Meistens sind Bewegungen jedoch durch den Mobilitätsgrad (engl. *DOM = degree of mobility*) δm eingeschränkt. Dieser entspricht der Anzahl der möglichen Freiheitsgrade (engl. *DOF = degrees of freedom*) δf (Ben-Ari & Mondada, 2017). Ein sich in der euklidischen Ebene bewegendes System hat höchstens drei Freiheitsgrade – die Position in x- und y-Richtung sowie den Kurs.

Das Konzept des Mobilitätsgrades kann anhand einiger Beispiele erläutert werden. Eine Straßenbahn kann sich nur entlang ihrer Gleise vorwärtsbewegen und hat somit nur einen einzigen Freiheitsgrad. Dieser Freiheitsgrad ist für die Lokomotive direkt zugänglich, so dass der Mobilitätsgrad einer Straßenbahn δm = 1 ist. Ein Roboter mit Differentialantrieb (ausgestattet mit separaten Motoren für das rechte und das linke Rad) hat drei Freiheitsgrade, da seine Pose aus der Position in der Ebene und dem Kurs besteht. Wenn jedes Rad die gleiche Geschwindigkeit hat, kann sich der Roboter vorwärts und rückwärts bewegen, und wenn er unterschiedliche Geschwindigkeiten in verschiedene Richtungen hat, kann er sich an Ort und Stelle drehen. Eine direkte Seitwärtsbewegung ist jedoch nicht möglich, da die Räder nicht zur Seite ausgerichtet werden können. Daher beträgt der Mobilitätsgrad eines Roboters mit Differentialantrieb δm = 2, was kleiner als sein Freiheitsgrad ist.

Im Vergleich dazu verfügt ein Auto nur über einen Mobilitätsgrad. Während der Motor des Fahrzeugs direkt eine Vorwärts- und Rückwärtsbewegung erlaubt, hat das Lenkrad keinen direkten Zugriff auf einen Freiheitsgrad und kann nur eine Orientierungsänderung um einen Freiheitsgrad vornehmen. Ein Auto kann sich nicht direkt seitwärts bewegen und sich auch nicht auf der Stelle drehen.

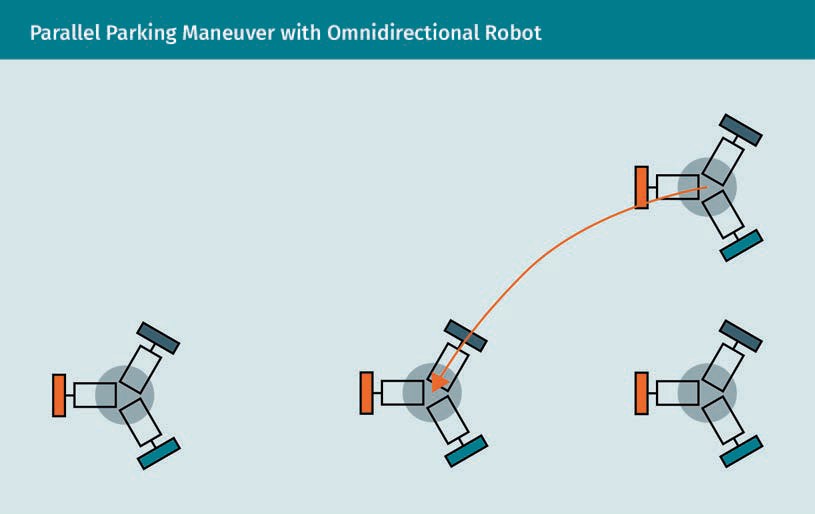
Ein Beispiel für ein System mit drei Mobilitätsgraden ist ein omnidirektionaler Roboter, der auch in der Lage ist, sich direkt seitlich zu bewegen. Dies gelingt zum Beispiel durch die Verwendung von Mecanum-Rädern (Ilon-Rädern). Dabei handelt es sich um normale Räder, auf deren Felgen (Umfang) kleine Rollen angebracht sind, die eine Seitwärtsbewegung ermöglichen (Tătar et al., 2013). Die maximale Anzahl von Freiheitsgraden eines Systems im dreidimensionalen euklidischen Raum beträgt sechs (drei Positions- und drei Orientierungsparameter). Die maximale Anzahl der Mobilitätsgrade beträgt also ebenfalls sechs.

Die Beziehung zwischen den Freiheitsgraden und den Mobilitätsgraden eines Systems ist die Grundlage für die Definition der holonomen Bewegung. Die Bewegung eines Systems wird als holonom bezeichnet, wenn Anzahl der Mobilitätsgrade = Anzahl der Freiheitsgrade. Sie gilt als nichtholonom, wenn Anzahl der Mobilitätsgrade < Anzahl der Freiheitsgrade. Systeme mit holonomer Bewegung können komplizierte Aufgaben sehr leicht und mit geringem Aufwand erledigen, auch bei eingeschränktem Konfigurationsraum des jeweiligen Systems. Im Allgemeinen gilt, dass je kleiner das Verhältnis von δm und δf ist, desto komplizierter werden die Manöver, die zur Erfüllung der Aufgaben erforderlich sind. Angenommen, ein System soll ein paralleles Einparkmanöver durchführen. δ Ein omnidirektionaler Roboter = 1 braucht für diese Aufgabe nur eine seitliche Bewegung, wie die folgende Abbildung zeigt.

m

f

δ



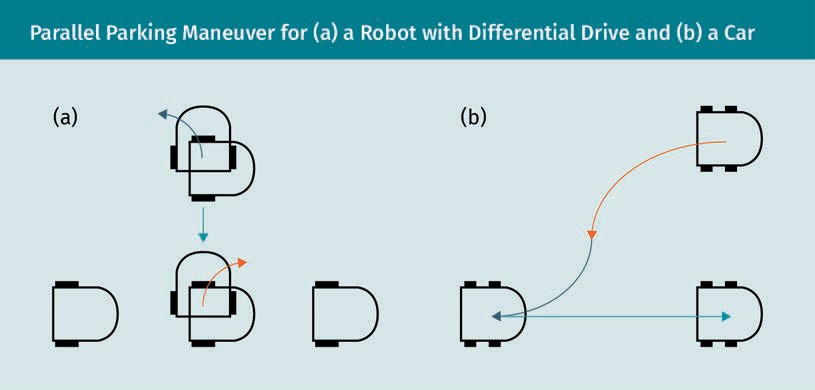
Ein Roboter mit Differentialantrieb δm = 2

hingegen braucht drei einfache und separate

δf 3

Bewegungen: Linksdrehung, Rückwärtsbewegung und Rechtsdrehung. Autofahrende wissen, δm = 1, dass paralleles Einparken viel komplizierter ist. Die beiden δf 3

Beispiele sind in der folgenden Abbildung dargestellt.



Es liegt auf der Hand, dass Systeme mit holonomer Bewegung viel flexibler sind, so dass Aufgaben viel leichter erledigt werden können. Ein hoher Wert des Quotienten aus δm und δf sollte also stets das Ziel sein.

### Fehler

Wie zuvor erwähnt, ist es aufgrund verschiedener Fehlerquellen nicht empfehlenswert, sich bei der Navigation ausschließlich auf die Odometrie zu verlassen. Diese Fehler können von dem sich bewegenden System selbst, von den Sensoren oder von der Umgebung herrühren. Natürlich haben auch Sensoren immer eine Messunsicherheit, die berücksichtigt werden muss. Ein Beispiel für Fehler, die vom fahrenden System selbst ausgehen, ist die Radgeometrie. Räder können unrund sein und einen Abrieb aufweisen, der den Durchmesser des Rades und damit den zur Berechnung der zurückgelegten Strecke verwendeten Umfang verfälscht. Außerdem können Fehler durch eine falsche Messung des Raddurchmessers entstehen. Auch das Material und der Luftdruck der Räder spielen eine wichtige Rolle für die Genauigkeit der Odometrie bei der Navigation. Es können auch Fehler in der Geometrie des Fahrgestells auftreten, z. B. eine fehlerhafte Messung der Radabstände oder ein Fehler in Bezug auf das Gewicht des Systems. Bei einer unausgewogenen Gewichtsverteilung werden z. B. einige Räder stärker belastet, wodurch sie sich verformen können. Ein wichtiger Umgebungsfehler kann zudem durch unebene Bodenbeschaffenheit entstehen sowie durch ein dadurch verursachtes Rutschen. Außerdem kann die Bewegung durch Wind verzerrt werden.

Allgemein lässt sich sagen, dass Fehler im Kurs (Richtung) eines sich bewegenden Systems viel schwerwiegender sind als Fehler in der Längsbewegung (Ben-Ari & Mondada, 2017). Nehmen wir an, dass sich unser System über eine Strecke von 10 m mit einer Unsicherheit von p Prozent bewegt. Die verursachte Abweichung in der Wegstrecke kann dann wie folgt berechnet werden:

∆x ≤ ± 10 m · p = ± p m

100 10

(1.34)

Die tatsächlich zurückgelegte Strecke liegt also zwischen 10 − p m und 10 + p m. Die Abweichung von p Prozent im Kurs des Systems in Grad kann folgendermaßen berechnet werden:

10 10

∆x = 360° · p = 3,6p°

100

(1.35)

Hieraus ergibt sich eine seitliche Abweichung von

∆y ≤ ± 10 · sin 3,6p° m

(1.36)

Ein Vergleich der Auswirkungen von Längs- und Kursfehlern für verschiedene Werte von p ist in der folgenden Tabelle dargestellt.

|  |  |  |
| --- | --- | --- |
| Auswirkung von Fehlern bei der linearen Bewegung und beim Kurs | | |
| p% | Δx [m] | Δy [m] |
| 1 | 0,1 | 0,63 |
| 2 | 0,2 | 1,25 |
| 5 | 0,5 | 3,09 |
| 10 | 1,0 | 5,88 |
| 20 | 2,0 | 9,51 |

Es liegt auf der Hand, dass Fehler beim Kurs viel größere Auswirkungen haben. Die Odometriefehler addieren sich und verringern die Zuverlässigkeit der Positionierung kontinuierlich. Es ist daher unvermeidlich, die räumliche Lage des Systems regelmäßig mit Hilfe einer absoluten Positionierungsmethode wie GPS zu aktualisieren, um neue Ausgangspositionen festzulegen, auf die sich die Messungen beziehen.

Zusammenfassung

Unter der Annahme einer konstanten Geschwindigkeit über einen bestimmten Zeitraum kann die zurückgelegte Strecke eines sich bewegenden Objekts durch Multiplikation der Geschwindigkeit mit der Zeitspanne zwischen Start und Stopp ermittelt werden. Die Geschwindigkeit für eine bestimmte zurückgelegte Strecke wiederum wird berechnet, indem die Strecke durch die gemessene Zeit dividiert wird. Die Geschwindigkeit ist die erste Ableitung der Position (Wegstrecke), was gezeigt werden kann, indem man die Zeitdifferenzen gegen Null gehen lässt. Die erste Ableitung der Geschwindigkeit, also die zweite Ableitung der Position, ist die Beschleunigung, welche die Änderung der Geschwindigkeit über die Zeit darstellt.

In der Praxis werden hauptsächlich zwei Bewegungsmodelle verwendet: das geschwindigkeitsbasierte und das odometriebasierte. Geschwindigkeitsbasierte Bewegungsmodelle arbeiten mit inertialen Messeinheiten (IMUs, Trägheitsmessgeräten) und berechnen die zurückgelegte Wegstrecke durch Integration der gemessenen Beschleunigungen und Winkelgeschwindigkeiten in jedem Freiheitsgrad. Das Verfahren zur Bestimmung der aktuellen Position relativ zu einer Ausgangsposition durch kontinuierliche Aktualisierung der Berechnungen wird als Koppelnavigation oder Koppelung (engl.: *dead reckoning*) bezeichnet. Odometriebasierte Bewegungsmodelle messen die zurückgelegte Strecke eines sich bewegenden Objekts mit Hilfe von Radsensoren (Raddrehgebern) oder Motordaten. Radsensoren messen die Radgeschwindigkeiten, indem sie die Anzahl der Umdrehungen zählen. Anhand des bekannten Raddurchmessers und ‑umfangs lässt sich dann die zurückgelegte Strecke berechnen. Bei beiden Bewegungsmodellen ist eine Verfälschung (Abdriften) nahezu unvermeidlich, so dass die absolute Position mit geeigneten Methoden wie GPS-Messungen aktualisiert werden muss. Fehler bei der Nutzung von Radsensordaten können z. B. durch falsche Messungen der Räder oder des Fahrwerks entstehen. Darüber hinaus können auch Abrieb, Temperatur, Luftdruck und Bodenbeschaffenheit zu Fehlern führen.

Die Anzahl der Freiheitsgrade eines Systems, die mit Aktuatoren direkt genutzt werden können, wird als Mobilitätsgrad bezeichnet. Wenn der Mobilitätsgrad gleich dem Freiheitsgrad ist, spricht man von einer holonomen Bewegung des Systems. Ist er kleiner, so ist die Bewegung nichtholonom.



# Lektion 2

## Lokale Navigation

#### LERNZIELE

Nach Abschluss dieser Lektion werden Sie gelernt haben, …

… was die Unterschiede zwischen globaler und lokaler Navigation sind.

… welche Grundlagen der Wegfindung und dabei zu lösende Teilprobleme es gibt.

… wie Hindernisse bei der Navigation umfahren werden können.

DL-E-DLMDSEAAD02-U02

1. Lokale Navigation

### Einleitung

Die Navigation ist eine der Hauptaufgaben der Funktion automatisierter, sich bewegender Systeme wie mobilen Robotern und automatisierten Fahrzeugen. Wenn sichergestellt werden kann, dass die Umgebung, durch die das System navigiert, völlig statisch ist, können globale Navigationstechniken verwendet werden, bei denen der Weg zwischen einem anfänglichen Start- und Endpunkt im Voraus berechnet wird. In der Praxis gilt diese Annahme selten, so dass andere Ansätze in Betracht gezogen werden müssen. Mit Hilfe der lokalen Navigation wird während des Betriebs ein optimaler Weg gefunden. Daher müssen Teilpfade berechnet werden, in die aktuelle Sensordaten einbezogen werden müssen. So erhält das System immer die neuesten Informationen über Hindernisse in der Umgebung, die es zu vermeiden gilt. Das dynamische Verhalten von Hindernissen stellt ein schwieriges Problem dar und erfordert reaktive Ansätze zur Vermeidung von Kollisionen.

In dieser Lektion werden Unterschiede zwischen den möglichen Navigationstechniken vorgestellt, wobei das Hauptaugenmerk auf die globale und die lokale Navigation gerichtet wird. Außerdem werden die Hauptbestandteile und die daraus ableitbaren Teilprobleme eines Navigationsalgorithmus untersucht. Anschließend werden das Wegfindungsproblem und die Hindernisvermeidung erörtert, wobei einige häufig verwendete Methoden für beide Aufgaben vorgestellt werden.

### Grundlegende Konzepte

Die Aufgabe der Navigation ist für jedes sich automatisch bewegende System unerlässlich und kann in drei verschiedene Kategorien eingeteilt werden: globale Navigation, lokale Navigation und persönliche Navigation (Patle et al., 2019). Globale Navigationstechniken können ein vorher festgelegtes Ziel anfahren und die Position von Hindernissen in der Umgebung bezüglich einer Referenzachse bestimmen. Die Informationen über die jeweilige Umgebung, die Position der Hindernisse und die Position des Ziels werden dabei im Voraus benötigt. Im Gegensatz dazu werden bei lokalen Navigationstechniken u. a. die dynamischen Umgebungsbedingungen und die Positionsbeziehungen ermittelt. Vorabinformationen, wie sie für die globale Navigation benötigt werden, sind für die lokale Navigation nicht erforderlich, weil die Informationen während des Prozesses gesammelt werden. Bei der persönlichen Navigation werden Umgebungselemente relativ zueinander auf der Grundlage ihrer Position behandelt.

Allgemein lässt sich sagen, dass die globale Navigation eine vollständig bekannte Umgebung voraussetzt, die sich während des Betriebs nicht ändern darf, während die lokale Navigation in unbekannten oder nur teilweise bekannten Umgebungen mit dynamischem Verhalten funktioniert. Globale und lokale Navigationsstrategien werden oft kombiniert, um ein bestimmtes Ziel zu erreichen. Während es bei der ersteren um die Wegfindung innerhalb eines zuvor generierten Umweltmodells geht, dreht es sich bei der letzteren um die Navigation in der unmittelbaren Umgebung, z. B. um das Fahren in eine bestimmte Richtung oder um das Verfolgen eines Korridors. Ein Hauptziel der lokalen Navigation ist die Vermeidung von Hinderniskollisionen.

Das Navigationsproblem für automatisierte mobile Systeme kann wie folgt umrissen werden: Ausgehend von der Startposition pStart des Systems, einer Endposition pEnde und einer Karte der Umgebung soll die beste Route von pStart zu pEnde berechnet werden. Dabei sollen bestimmte Optimalitätskriterien berücksichtigt werden, und das System soll sich dann entlang der besten Route zur Endposition bewegen. Dieses Problem besteht aus vielen Teilproblemen, die für sich genommen schon sehr komplex sind. Ein erster Versuch zur Lösung des Navigationsproblems bestünde darin, einen statischen Pfad von pStart nach pEnde zu planen und dann genau die zuvor berechneten Bewegungen auszuführen. Bei diesem Ansatz kann jedoch schon die Ausgangsposition des Systems nicht bestimmt werden. Darüber hinaus muss ein berechneter Pfad die physischen Abmessungen des Systems und sein kinematisches Modell berücksichtigen, um eine korrekte Bewegung zu gewährleisten und eine Überlastung des Systems zu vermeiden. In jedem Zeitintervall muss die aktuelle Position des Systems mit dem zuvor berechneten Pfad verglichen sowie der Geschwindigkeitsvektor bestimmt werden, damit auf mögliche Interaktionen mit anderen Hindernissen reagiert werden kann. Während der gesamten Bewegung des Systems muss die Betriebssicherheit gewährleistet sein.

###### Der Navigationsalgorithmus

Ein Navigationsalgorithmus muss das Umgebungsmodell, die Bestimmung der Position des Systems und das Bewegungsverhalten des Systems berücksichtigen (Schäfer, 2003).

Wahl des Umgebungsmodells

Im Allgemeinen können zwei verschiedene Modelle zur Simulation der Umgebung verwendet werden: ein statisches Modell, bei dem das automatisch fahrende System das einzige sich bewegende Objekt ist, und ein dynamisches Modell, bei dem die Umgebung sehr dynamisch ist, also weitere sich bewegende Objekte enthält. Natürlich vereinfacht das statische Modell die Navigationsaufgabe, es ist aber sehr eingeschränkt, weil es keine anderen sich bewegenden Objekte erfasst. Es kann als eine Momentaufnahme der Umgebung beschrieben werden. Dynamische Modelle erfassen kontinuierlich die gesamte gegenwärtige Dynamik. Dies ist zwar rechnerisch aufwendiger, aber aufgrund der Einbeziehung neuer Informationen bei jedem Zeitschritt auch besser.

Bestimmung der Position des Systems

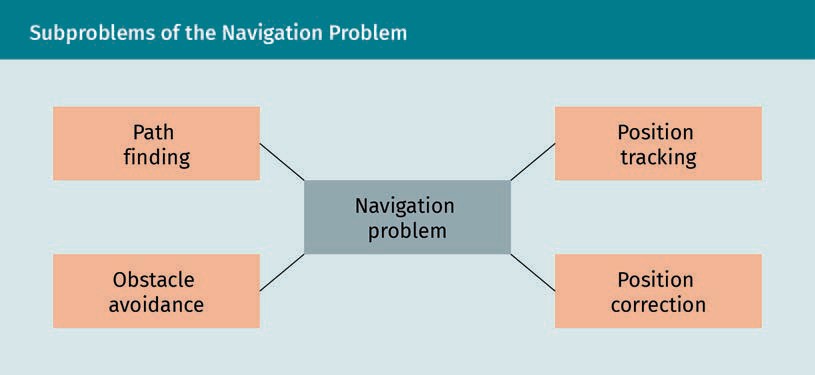
Im Allgemeinen gibt es zwei verschiedene Ansätze zur Positionsbestimmung, nämlich solche, die auf Orientierungspunkten basieren, und solche, die Sensordaten verarbeiten. Durch die Einbeziehung von Orientierungspunkten in der Umgebung kann die Position des Systems jederzeit bestimmt werden. Bei sensorbasierten Ansätzen kann die Position anhand der gesammelten Sensormessungen und einer vorgegebenen Karte bestimmt werden.

Bewegungsverhalten des Systems

Das Bewegungsverhalten kann in planerische und reaktive Ansätze unterteilt werden. Bei den ersteren geht es innerhalb einer bestimmten Umgebung um die Wegfindung von einer Start- zu einer Zielposition. Bei den letzteren hingegen wird der lokal beste Pfad anhand von Optimalitätskriterien kontinuierlich aktualisiert. Reaktive Ansätze benötigen zusätzlich zur vorgegebenen Karte Sensordaten.

###### Teilprobleme des Navigationsproblems

Unter Berücksichtigung eines dynamischen Umgebungsmodells und einer sensorbasierten Positionsbestimmung kann das Navigationsproblem in vier Teilprobleme zerlegt werden, wie die folgende Abbildung zeigt (Schäfer, 2003).



Wegfindung von einer Start- zu einer Endposition

Mittels Wegfindung wird der optimale Pfad zwischen einem Start- und einem Endpunkt innerhalb der lokalen Karte berechnet.

Positionsbestimmung

Die aktuelle Position des Systems in der Welt wird anhand von Sensor- und Odometriedaten sowie der Karte berechnet.

Hindernisvermeidung

Bei der Hindernisvermeidung geht es darum, den geplanten Pfad zum lokalen Ziel zurückzulegen sowie Kollisionen mit Hindernissen und Teilen der Umgebung zu vermeiden.

Problem der Positionskorrektur

Die berechnete Position muss mit Hilfe einer korrigierten Positionsbestimmung synchronisiert werden.

In den nächsten beiden Abschnitten werden zwei Aspekte der lokalen Navigation im Detail behandelt. Wir werden verschiedene Ansätze für die Wegfindung bewerten und das Problem der Vermeidung von Hinderniskollisionen untersuchen.

Lokale Navigation

### Wegfindung

Bereits die Aufgabe, einen geeigneten Pfad für das automatisierte Bewegungssystem zu finden, ist schwierig. Tatsächlich ist das Finden eines Pfades in einer Umgebung mit sich bewegenden Objekten mit konstanter Geschwindigkeit bereits ein NP-schweres Problem (Schäfer, 2003). Einen Überblick über die Komplexität von Berechnungen, einschließlich der Definition von NP-schweren Problemen, liefert einschlägige Literatur (Cook, 2006). Man könnte sagen, dass dieses Problem so schwierig ist, dass es von einem normalen Computer in einem annehmbaren Zeitraum nicht gelöst werden kann. Aufgrund dieses hohen Schwierigkeitsgrades müssen Heuristiken verwendet werden.

Betrachten wir eine zweidimensionale Karte der lokalen Umgebung unseres Systems, die gemeinhin als Vogelperspektive bezeichnet wird. Eine mögliche Darstellung der lokalen Umgebung ist eine Gitternetzkarte. Dabei wird die Umgebung in ein diskretes Gitter unterteilt (Rasterung), und jeder einzelnen Gitterzelle wird die Wahrscheinlichkeit zugewiesen, dass es darin ein Objekt gibt (Elfes, 1989). Eine andere Darstellung ist eine merkmalsbasierte Karte, z. B. eine Abschnittskarte, bei der jedes Kartenobjekt als Abschnitt durch die Weltkoordinaten seines Anfangs- und Endpunkts dargestellt wird. Solche merkmalsbasierten Karten werden im Allgemeinen als Listen gespeichert, wobei die Abschnitte als 4-Tupel parametrisiert sind. Ein Pfad in Cfree eines beweglichen Systems von einer Anfangskonﬁguration qstart zu einer Endkonﬁguration qend ist eine stetige Abbildung τ: [0,1] → Cfree mit τ(0) = qstart und r(1) = qend (Schäfer, 2003).

Ein Pfad gilt als regulär, wenn er unter Berücksichtigung der geometrischen Dimensionen des Systems befahrbar ist, und es möglich ist, das Ziel zu erreichen. Das Problem der Wegfindung besteht darin, einen regulären Pfad von einer Anfangskonﬁguration qstart zu einer Endkonﬁguration qend zu finden, also eine Abbildung τ: [0,1] → Cfree mit τ(0) = qstart und τ(1) = qend. Dynamische Hindernisse werden nicht bei der Wegfindung, sondern bei den reaktiven Algorithmen zur Hindernisvermeidung berücksichtigt. Zunächst wird der Pfad optimal geplant und später festgestellt, ob er aufgrund von Hindernissen geändert werden muss. Die Grundlage des gesamten Verfahrens ist die Bestimmung der Anfangs- und Endposition. Um die Unsicherheit der Position zu modellieren, müssen sowohl die Systemkonfiguration als auch der Positionsfehler berücksichtigt werden. Bei probabilistischen Methoden wird eine Kovarianz verwendet, um die Unsicherheit zu berücksichtigen, während bei geometrischen Ansätzen ein geometrischer Fehlerbereich verwendet wird (Schäfer, 2003). Die Ausgangsposition wird in der Regel bei der Erstellung des Umgebungsmodells bestimmt, wo sie als Nullpunkt der Karte angezeigt wird. Die folgenden Ausgangspunkte für die nächsten Zeitschritte können relativ zu diesem Nullpunkt bestimmt werden.

Für die Generierung eines Pfads von einer Ausgangs- zu einer Endkonfiguration werden Optimalitätskriterien benötigt, welche die Güte eines Pfads definieren. Unterschiedliche Kriterien führen zu unterschiedlichen Pfaden. Ein mögliches Kriterium ist die Suche nach einem Pfad mit minimaler Länge zwischen Anfangs- und Endpunkt. Eine weitere Möglichkeit für einen optimalen Pfad ist die Minimierung der Reisezeit. Dabei muss der Pfad mit der geringsten Reisezeit nicht automatisch der kürzeste sein. Häufig wird angestrebt, die Anzahl der Drehungen auf dem Pfad so gering wie möglich zu halten, weil Drehungen eine der Hauptfehlerquellen bei der Bewegung sind. Darüber hinaus kann der Pfad mit maximalem Abstand zu allen Hindernissen in der Umgebung geplant werden, um das Risiko einer Hinderniskollision so klein wie möglich zu halten.

4-Tupel

n-Tupel sind geordnete Folgen von n Elementen. Ein 4-Tupel ist also eine geordnete Folge von vier Elementen.

Für bestimmte Probleme können auch andere Kriterien berücksichtigt werden, was natürlich die Komplexität erhöht. Da die minimale Weglänge das einfachste Kriterium ist und in einfachen Szenarien auch am häufigsten verwendet wird, wollen wir sie hier betrachten. Außerdem gehen wir davon aus, dass unser System über holonome Bewegungen verfügt und sich daher in jedem Freiheitsgrad frei bewegen kann. Natürlich ist diese Annahme in der Praxis kaum zutreffend. Dennoch eignet sie sich gut für die Darstellung verschiedener Ansätze zur Wegfindung. Üblicherweise werden drei klassische Algorithmen zur Wegfindung verwendet (Latombe, 2012).

###### Roadmaps (Straßenkarten)

Mit Roadmap-Ansätzen wird ein Graph der möglichen Wege (Pfade) berechnet. Ein Graph G ist ein geordnetes Paar (V, E) mit einer Menge von Eckpunkten V und einer Menge von Kanten E. Jeder Punkt v ∊ V ist ein Punkt der euklidischen Ebene (2 , und jedes e ∊ E ist ein 3-Tupel e = (v1, v2, c) mit v1, v2 ∊ V sowie eine Kurve c ⸦ (2 mit den Kanten v1, v2 und der Länge l(c). Ein Weg (Pfad) ist eine Teilmenge eines solchen Graphen W⸦ G und eine Folge von w Kurven (c1, ..., cw). Für jedes ci = (vi,1, vi,2) gilt vi,2 = vi+1,1*,* was bedeutet, dass der Endpunkt der einen Kurve der Anfangspunkt der folgenden ist. Im Hinblick auf das Optimalitätskriterium der minimalen Weglänge kann die Weglänge wie folgt dargestellt werden:

w

l W = ∑l ci

i = 1

(2.1)

Eine Straßenkarte beschreibt ein Netz von eindimensionalen, durch einen Graphen dargestellten Kurven, der zweidimensionale Positionen im freien Raum Cfree umfasst. Auf diese Weise wird die Wegfindung auf die Suche nach einem Pfad auf der so erzeugten Straßenkarte zurückgeführt. Es gibt verschiedene Ansätze, von denen die folgenden die gängigsten sind.

Die randomisierte Straßenkarte erzeugt zufällige zweidimensionale Punkte im freien Raum Cfree, die in einem bestimmten Radius miteinander verbunden werden, wenn der verbundene Streckenabschnitt kein Hindernis kreuzt (Amato & Wu, 1996). Die Delaunay-Roadmap folgt demselben Ansatz der zufälligen Punktbildung, allerdings werden hier die Punkte mit einem Delaunay-Graphen verbunden (Klein, 2006). Ein weiterer Ansatz ist die Voronoi-Roadmap, bei der ein Voronoi-Diagramm für die Darstellung der Umgebung verwendet wird (Bhattacharya & Gavrilova, 2008). Die gitterbasierte Straßenkarte modelliert die Umgebung anhand eines Gitternetzes, wobei jede Kante mit der benachbarten Kante verbunden ist. Darüber hinaus kann eine Sichtbarkeitsgraph-Straßenkarte verwendet werden, welche die Sichtbarkeit der Hindernisse zeigt und aus der Anfangs- und Endkonfiguration, den Eckpunkten der Hindernisse und ihren Kanten besteht (Lulu & Elnagar, 2005).

Es ist möglich, ein Fehlergerüst für einen gültigen, aber nicht notwendigerweise regulären Pfad zu definieren. Dieser ist abhängig vom Radius des Kreises, der durch die längste räumliche Ausdehnung des beweglichen Systems rsystem und einen Sicherheitsabstand dsafety definiert wird. Für jeden Abschnitt wi eines sicheren und regulären Pfades Ws gilt, dass der Abstand zu jedem vorhandenen Hindernis dmin = rsystem + dsafety beträgt.

###### Zellzerlegung

Zellzerlegungsansätze sind ein weiteres beliebtes Verfahren zur Wegfindung. Bei dieser Methode wird die Karte in Zellen zerlegt, so dass sie einen Verbindungsgraphen ergeben. Dieser wird dann für die Wegfindung genutzt (Kloetzer et al., 2015; Gonzalez et al., 2017). Bei einer Zellzerlegung wird der freie Raum Cfree in eine Menge von n Regionen (Zellen) unterteilt. Der Verbindungsgraph Gcon ist ein geordnetes Paar (Vcon, Econ), wobei die Eckpunkte Vcon und die Kanten Econ einzelne Zellen bzw. direkte benachbarte Zellen darstellen. Eine fehlende Kante zwischen zwei Eckpunkten bedeutet, dass sie nicht benachbart sind. Die Zerlegung einer Karte in Zellen kann entweder durch exakte Zerlegung oder näherungsweise erfolgen. Während der Verbindungsgraph bei einer exakten Zerlegung die gesamte Karte erfasst, stellt der Verbindungsgraph einer näherungsweisen Zerlegung nur eine Teilmenge davon dar. Gängige Zerlegungsmethoden sind die sichtbarkeitsbasierte Zellzerlegung, die trapezförmige Zerlegung und die Gitterzerlegung (Latombe, 2012). Nach der Zellzerlegung muss der Verbindungsgraph bestimmt werden, bei dem der Mittelpunkt jeder Zelle als Eckpunkt betrachtet wird und benachbarte Zellen durch Abschnitte verbunden sind. Der so entstandene Graph kann nun für die Berechnung des Pfads von einem Anfangs- zu einem Endpunkt genutzt werden.

###### Potenzialfelder

Ein dritter Ansatz zum Auffinden eines Pfads funktioniert über ein Potenzialfeld. Hier wird von einem künstlichen Potenzialfeld U im Konfigurationsraum ausgegangen. Dies wirkt sich auf das bewegliche System aus. Ziel ist es, vom Punkt des höchsten Potenzials, dem Anfangspunkt, zum Punkt des kleinsten Potenzials, dem Endpunkt, zu gelangen, wobei ein Wechsel von der Konfiguration q1 zur Konfiguration q2 nur dann erfolgt, wenn U(q2) ≤ U(q1) gilt. Es werden zwei verschiedene Potentiale unterschieden: ein abstoßendes und ein anziehendes Potential. Das abstoßende Potenzial Urep drückt das System von den Hindernissen Bi weg. Daher werden solche abstoßenden Potenziale um Hindernisse herum platziert, um das System daran zu hindern, bestimmte Konfigurationen anzunehmen. Außerdem wirken abstoßende Potenziale nur in unmittelbarer Nähe der Hindernisse und sollten bei größerem Abstand keine Auswirkungen haben. Im Gegensatz dazu zieht das Anziehungspotential Uatt das System in Richtung des Endpunkts. Je größer der Abstand zum Ziel, desto geringer ist das Anziehungspotenzial. Die Potenzialfunktion U: C → ( bewertet dann die Systemkonfigurationen über die Summe der abstoßenden und anziehenden Potenziale. Für eine Konfiguration q gilt also die folgende Gleichung:

Potenzialfelder

Das für die Wegfindung genutzte Potenzialfeld ist an das Konzept der elektrischen Ladung angelehnt.

U q = Uatt q + ∑m Urep q

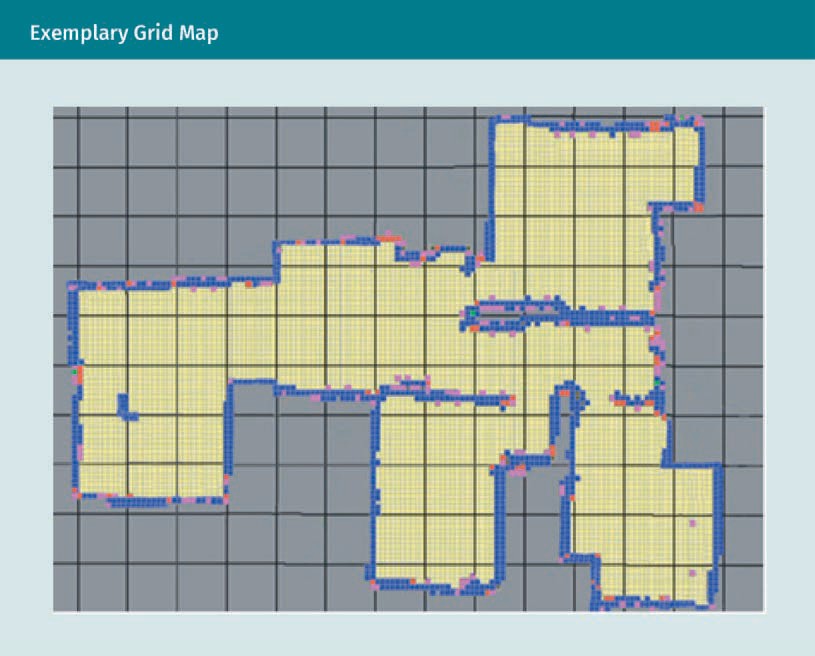
i = 1 i

(2.2)

Das Problem der Potenzialfeldmethode besteht darin, eine geeignete Bewertungsfunktion für den Konfigurationsraum zu finden. Da bei solchen Funktionen normalerweise lokale Minima zu berechnen sind, müssen Methoden entwickelt werden, um diese Einschränkung aufzulösen (Latombe, 2012).

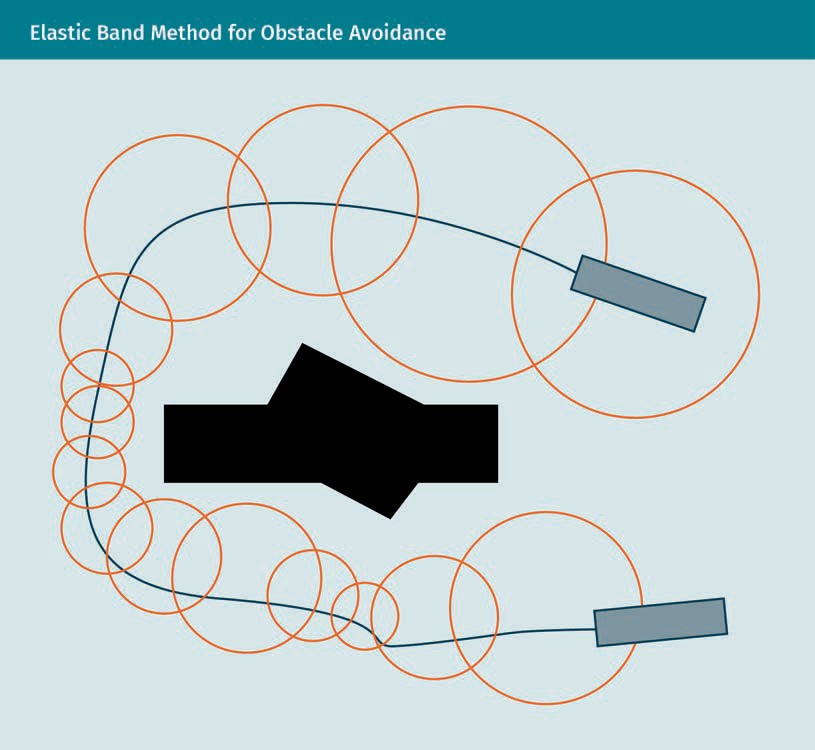
### Hindernisvermeidung

Jeder Navigationsalgorithmus muss Kollisionen mit Hindernissen vermeiden. Übliche Lösungsansätze für dieses Problem sehen die Berücksichtigung aktueller Sensorinformationen vor, damit das sich bewegende System entsprechend reagieren kann. Aufgrund von Ausweichmanövern kann es vorkommen, dass der Endpunkt nicht auf dem geplanten Pfad erreicht wird. Deshalb müssen Navigationsalgorithmen mit dynamischer Hindernisvermeidung die Fähigkeit zur Neuplanung des Pfads besitzen. Die folgenden Methoden zur Hindernisvermeidung sind die gängigsten. Das künstliche Potenzialfeld kann neben der Pfadplanung auch auf lokale Sensordaten angewandt werden, um das sich bewegende System mit abstoßenden Potenzialen abzulenken. Dies kann auch zur Hindernisvermeidung genutzt werden. Ein weiterer Ansatz ist das Vektorfeldhistogramm (Borenstein & Koren, 1991), bei dem aus aktuellen Sensordaten ein Belegungsraster erstellt wird, um daraus Hindernisinformationen zu extrahieren. In der folgenden Abbildung ist ein beispielhaftes Belegungsraster eines Roboters dargestellt.



Dabei wird die Umgebung des bewegten Systems in Sektoren eingeteilt, für welche jeweils die Hindernisdichte berechnet wird. Daraufhin wird ein freier Sektor ausgewählt und dazu passend werden der jeweilige Kurs und die Geschwindigkeit des Systems berechnet. Beim Gummiband-Ansatz wird ein kollisionsfreier Pfad durch eine Menge zusammenhängender Blasen (Kreise) dargestellt. Diese zeigen alle möglichen Konfigurationen p des sich bewegenden Systems, die von der Konfiguration q ohne Kollision erreicht werden können (Quinlan & Khatib, 1993). Die Radien der Blasen stellen den Mindestabstand zum nächsten Hindernis dar. Ähnlich wie beim Potentialfeldansatz können sie zusätzlich durch abstoßende und anziehende Potentiale modifiziert werden.

Eine zusätzliche Bedingung ist, dass der Anfang einer neuen Blase stets innerhalb der vorherigen Blase liegen muss. Die Gummiband-Methode zur Hindernisvermeidung wird in abstrakter Form dargestellt.



Eine weitere gängige Methode zur Hindernisvermeidung ist das dynamische Fenster, ein lokaler Suchraum innerhalb des Raums aller möglichen Translations- und Rotationsgeschwindigkeiten v bzw. ω (Fox et al., 1997; Ogren & Leonard, 2005). Mit Hilfe des aktuellen Geschwindigkeitsvektors (va, ωa) und des Beschleunigungsvektors (ν˙, ω˙) im Zeitintervall ∆t ist das dynamische Fenster folgendermaßen definiert:

Vd = ν, ω

ν ∈ νa − ∆t · ν˙, νa + ∆t · ν˙ , ω ∈ ωa − ∆t · ω˙ , ωa + ∆t · ω˙

(2.3)

Geometrisch beschreibt das dynamische Fenster ein Rechteck um den aktuellen Geschwindigkeitsvektor und stellt die Grenze der erreichbaren Geschwindigkeiten dar, also mögliche Bahnkurven, welche durch diejenigen im aktuellen Zeitintervall definiert sind. Eine Optimierungsfunktion bewertet die Menge der zulässigen Geschwindigkeiten unter Berücksichtigung der Strecke zum Endpunkt, der Orientierung am Endpunkt und der maximal zulässigen Geschwindigkeiten.

Daraus ermittelt sie den Geschwindigkeitsvektor mit der höchsten Bewertung. Wie bereits erwähnt, verringern reaktive Ansätze zur Hindernisvermeidung die Stabilität von Navigationsalgorithmen, was bedeutet, dass ein vorher festgelegtes Ziel möglicherweise nicht wie geplant erreicht wird. Aus diesem Grund müssen geeignete Gegenmaßnahmen betrachtet werden.

Zusammenfassung

Die Navigation ist ein wesentlicher Bestandteil von automatisierten mobilen Systemen. Generell kann zwischen globaler und lokaler Navigation unterschieden werden. Während die Umgebungsmerkmale im Voraus bekannt sein müssen, ist die lokale Navigation nicht auf diese Annahme beschränkt, sondern berechnet die Darstellung der Umgebung während des Betriebs. Die globale Navigation ermöglicht eine zuvor definierte Navigation aufgrund der bekannten Umgebung, was bei der lokalen Navigation nicht möglich ist. Da die Annahme einer zuvor bekannten statischen Umgebung in der Praxis fast nie zutrifft, wird in den meisten Fällen, in denen ein dynamisches Verhalten vorliegt, die lokale Navigation genutzt. Daher wird der Pfad von einem Start- zu einem Endpunkt in mehrere zu durchlaufende Teilpfade unterteilt.

Ein Navigationsalgorithmus muss drei verschiedene Aufgaben berücksichtigen: die Wahl des Umgebungsmodells, die Bestimmung der Systemposition und das Bewegungsverhalten des Systems. Die für die Darstellung der Umgebung verwendeten Modelle können in statische und dynamische Modelle unterteilt werden. Während erstere davon ausgehen, dass das Ego-System das einzige sich bewegende Objekt in der Umgebung ist, beziehen letztere andere mögliche dynamische Elemente mit ein. Die Position eines sich bewegenden Objekts in einer Umgebung kann über Ansätze mit Orientierungspunkten oder mit Sensoren bestimmt werden. Erstere stützen sich nur auf bekannte Orientierungspunkte, anhand derer die Position des Systems jederzeit bestimmt werden kann. Letztere nutzen Sensoren, um Umgebungsdaten zu sammeln. Für die Positionsbestimmung werden diese Daten dann mit einer Karte verglichen. Das Bewegungsverhalten des Systems kann generell durch zwei Ansätze unterschieden werden: Beim planungsbasierten Verhalten wird vor dem Betrieb der komplette Pfad von einem Start- zu einem Endpunkt bestimmt, während beim reaktiven Verhalten der lokal beste Pfad des Systems anhand bestimmter Optimalitätskriterien kontinuierlich aktualisiert wird.

Das Navigationsproblem kann in vier Teilprobleme unterteilt werden: Wegfindung von einer Start- zu einer Endposition, Positionsverfolgung, Hindernisvermeidung und das Problem der Positionskorrektur. Wegfindung bedeutet die Berechnung des optimalen Pfads zwischen einem Start- und einem Endpunkt innerhalb der lokalen Karte. Bei der Positionsbestimmung wird die aktuelle Position des Systems in der Welt anhand von Sensordaten, Odometrie und einer Karte berechnet. Hindernisvermeidung bedeutet, den geplanten Pfad zum lokalen Ziel zurückzulegen und dabei Kollisionen mit Hindernissen und Umgebungselementen zu vermeiden. Bei der Positionskorrektur wird die berechnete Position mit den korrigierten Positionsschätzungen synchronisiert.



# Lektion 3

## Lokalisierung

#### LERNZIELE

Nach Abschluss dieser Lektion haben Sie gelernt, …

… welche grundlegenden Konzepte der Lokalisierung es für automatische Bewegungssysteme gibt.

… wie die Triangulation zur Lokalisierung eingesetzt werden kann.

… wie GPS funktioniert und wie eine Position durch den Empfang von Satellitensignalen bestimmt  
werden kann.

… was die Grundlagen der probabilistischen Lokalisierungstechniken sind.

DL-E-DLMDSEAAD

02-U03

1. Lokalisierung

### Einleitung

Ohne Lokalisierung ist jedes automatisierte System in seiner Umgebung verloren. Die Position muss während des Betriebs jederzeit geschätzt werden können, was auf unterschiedliche Weise geschehen kann. Im Optimalfall existiert eine hochpräzise Karte, und dem System wird seine genaue Ausgangsposition mitgeteilt, allerdings ist dieses Szenario in der Regel nicht erreichbar. Deshalb müssen Techniken genutzt werden, die eine Lokalisierung anhand von Sensordaten ermöglichen. Mit der Methode der Triangulation kann die Position eines Objekts durch Messung der Winkel zu verschiedenen Orientierungspunkten in der Umgebung bestimmt werden. GPS-Empfänger arbeiten nach einem ähnlichen Prinzip. In einem Verfahren namens Trilateration werden anstelle von Winkeln Wegstrecken zur Positionsbestimmung verwendet. Die Lokalisierung kann auch probabilistisch erfolgen. Dazu gibt es verschiedene Methoden, die jeweils Vor- und Nachteile haben.

In dieser Lektion geben wir einen Überblick über die bestehenden Möglichkeiten der Lokalisierung und zeigen das Funktionsprinzip anhand einfacher Beispiele. Besonderes Augenmerk wird auf das Thema probabilistische Lokalisierung gelegt, weil diese heute in nahezu jeder Anwendung der Robotik und des autonomen Fahrens eingesetzt wird.

### Grundlegende Konzepte

Ein mobiles automatisiertes System, wie z. B. ein mobiler Roboter oder ein automatisiertes Fahrzeug, nimmt die Umgebung, in der es sich bewegt, wahr. Die vielleicht wichtigste Teilaufgabe hierbei ist die Lokalisierung. Sie bildet die Grundlage für alle anderen Aufgaben der beteiligten Software-Programme, wie Bewegungsplanung und Bewegungssteuerung, und dient als Referenz für diese. Das Ziel der Lokalisierung ist es, den aktuellen Zustand des Systems xk in einem globalen Koordinatensystem zu jedem Zeitschritt tk abzuschätzen. Eine Abschätzung der aktuellen Position innerhalb des Koordinatensystems reicht noch nicht aus, sondern auch die aktuelle Orientierung gehört zum Zustand des Ego-Systems. Bei der Lokalisierung in der zweidimensionalen euklidischen Ebene (2, kann der Zustand durch einen dreidimensionalen Vektor xk = (x y θ)T dargestellt werden, wobei x und y die Positionen in x- bzw. y-Richtung sind und die Orientierung durch die Variable θ erfasst wird. Ein bewegtes System in der euklidischen Ebene hat also drei Freiheitsgrade. Die Erweiterung auf die Lokalisierung im dreidimensionalen euklidischen Raum (3 führt zu drei weiteren Freiheitsgraden des Zustands. Trivialerweise müssen die Position in z-Richtung und zwei weitere Drehungen mit einbezogen werden, was den Zustandsvektor xk = (x y z φ θ ψ)T ergibt.

Im Allgemeinen gibt es zwei Hauptklassen der Lokalisierung: lokale und globale Selbstlokalisierung. Erstere erfordert die Kenntnis des Anfangszustands des Ego-Systems, um die Folgezustände in den nächsten Zeitschritten einzuschätzen. Bei Letzterer besteht die Aufgabe darin, den Zustand nur durch das Sammeln von Sensordaten einzuschätzen. Ein Extremfall der globalen Selbstlokalisierung ist das Problem des entführten Roboters, bei dem das System während des Lokalisierungsvorgangs entführt und an eine neue Position gebracht wird. Es gibt viele Sensoren, die für die Lokalisierung genutzt werden können. Dabei werden propriozeptive oder exterozeptive Sensoren unterschieden.

Propriozeptive Sensoren sammeln Informationen aus dem Ego-System selbst. Radsensoren messen zum Beispiel die Anzahl der Umdrehungen der Räder und leiten dann aus bekannten Raddurchmessern den zurückgelegten Weg ab. Zur Ermittlung des aktuellen Lenkwinkels werden in der Regel Drehgeber verwendet. Trägheitsmessgeräte (IMUs) bestehen aus Beschleunigungs- und Kreiselmessern, die alle sechs Freiheitsgrade abdecken und zur Berechnung der Bewegung in jede Richtung und der Änderung der Orientierung innerhalb eines Zeitschritts verwendet werden können. Ein weiterer propriozeptiver Sensor, der absolute Informationen über den aktuellen Kurs des Ego-Fahrzeugs liefert, ist der Kompass.

Im Gegensatz dazu sammeln exterozeptive Sensoren Informationen aus der Umgebung, durch die sich das Ego-System bewegt. Kameras können zur Abschätzung von Bewegungen verwendet werden, indem die Position von Merkmalen in aufeinanderfolgenden Bildern verglichen wird. Darüber hinaus eignen sie sich hervorragend für die Objekterkennung, was auch für die Bewegungsschätzung hilfreich sein kann. Bei RGBD-Kameras kommt die Tiefenerkennung hinzu, die eine direkte Abstandsmessung ermöglicht. Damit können Hindernisse erkannt oder der Abstand zu Begrenzungen in der Umgebung, wie z. B. Wänden, gemessen werden. Die Nutzung von LIDAR ermöglicht eine präzise Vermessung der Umgebung mit hoher räumlicher Auflösung, die für die Objekt- und Bewegungserkennung geeignet ist. Daher werden die in zwei aufeinanderfolgenden Scans erfassten Objekte verglichen, um Informationen über die Bewegung des Ego-Systems zu erhalten. Mit RADAR-Geräten *(Radio Detection and Ranging)* lassen sich Objekte in großer Entfernung erkennen. Das globale Navigationssystem (GPS) und die globalen Navigationssatellitensysteme sind im Allgemeinen exterozeptive Sensoren, auch wenn sie scheinbar Informationen aus dem Ego-System selbst sammeln. Sie kommunizieren nämlich mit einem Netz von Satelliten und erhalten ihre Informationen aus der Umgebung. So wird eine absolute Position auf der Erde erhalten, die direkt als Referenz genutzt werden kann.

Häufig werden für Lokalisierungssysteme Karten verwendet, was den Navigationsvorgang aufgrund der bekannten Umgebung vereinfacht. Intern können Karten metrisch oder topologisch dargestellt werden. Metrische Karten sind in zweidimensionalen Koordinatensystemen gezeichnet und werden am häufigsten von Menschen benutzt. Jedes Objekt wird mit seinen genauen Koordinaten positioniert. Diese Art der Darstellung ist zwar sehr übersichtlich und nützlich, hat aber den Nachteil, dass sie aufgrund von Unsicherheiten bei den genauen Positionskoordinaten anfällig für Störungen ist. Im Gegensatz dazu beschreiben topologische Karten die Beziehungen zwischen relevanten Umgebungselementen anhand eines Graphen, z. B. werden häufig die Abstände zwischen Objekten gespeichert. Während die Eckpunkte des Graphen die spezifischen Elemente oder Objekte der Umgebung darstellen, repräsentieren die Kanten die Wege zwischen ihnen. Die folgende Abbildung veranschaulicht die beiden unterschiedlichen Kartendarstellungen anhand eines Beispielszenarios.

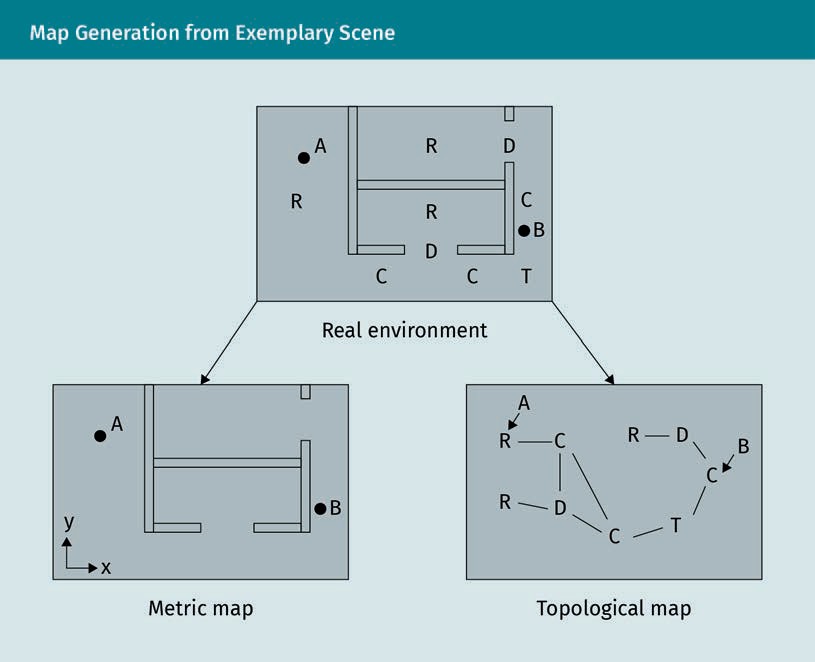
Propriozeptiv  
Die Vorsilbe „proprio“ kommt vom lateinischen Begriff *proprius* und bedeutet „eigen“.

Exterozeptiv

Die Vorsilbe   
„extero“ kommt vom lateinischen Wort *exter* und bedeutet „außen“ oder „extern“.

Merkmal

Ein Merkmal ist eine Information über den Inhalt von Daten wie z. B. Bildern.



Metrische Karten ermöglichen eine optimale Pfadberechnung und liefern sehr genaue Lokalisierungsergebnisse (Elasmar, 2012). Die Erstellung und Pflege metrischer Karten ist dagegen im Vergleich zur topologischen Darstellung schwierig. In der folgenden Tabelle werden die wichtigsten Eigenschaften dieser beiden Arten der Kartendarstellung verglichen.

|  |  |  |
| --- | --- | --- |
| Vergleich von metrischer und topologischer Kartendarstellung | | |
|  | Metrisch | Topologisch |
| Auflösung | Hoch | Niedrig |
| Berechnungszeit | Hoch | Niedrig |
| Speicherauslastung | Hoch | Niedrig |
| Wartung (Aktualisierungen) | Hoch | Niedrig |
| Skalierbarkeit | Niedrig | Hoch |

|  |  |  |
| --- | --- | --- |
|  | Metrisch | Topologisch |
| Empfindlichkeit gegenüber Rauschen | Höher | Niedriger |
| Größe der Karte | Größer | Kleiner |

Der Hauptvorteil einer metrischen Karte ist ihre hohe Auflösung. Die niedrige Auflösung stellt zugleich den größten Nachteil der topologischen Karte dar. Es gibt verschiedene Arten von Lokalisierungsmethoden, die in der Praxis eingesetzt werden, und die jeweils Vorteile in unterschiedlichen Anwendungen haben. Ein einfach umzusetzender Ansatz ist die Koppelnavigation *(Dead Reckoning)*. Bei dieser Methode werden durch interne Sensordaten kleine Positionsänderungen des Ego-Systems gemessen und zur Berechnung der aktuellen Position aufsummiert. Der Fehler wächst aufgrund des Sensorrauschens durch die kontinuierliche Summierung nahezu unbegrenzt an, was ein großer Nachteil dieser Technik ist. Aus diesem Grund wird die Koppelnavigation nur selten als eigenständiges Verfahren eingesetzt, sondern eher mit Navigationshilfen wie GPS kombiniert, um zusätzliche genaue Positionsinformationen zu erhalten.

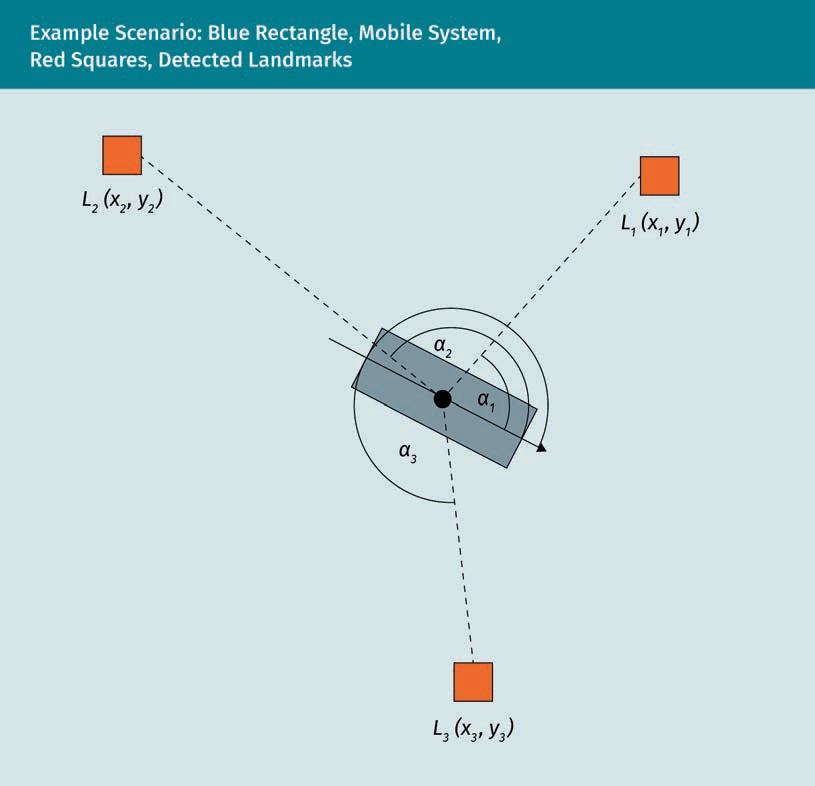
Ein weiterer gängiger Ansatz ist die Lokalisierung anhand von Orientierungspunkten. Es gibt natürliche Orientierungspunkte (tatsächliche Umgebungselemente), wie Türen und Wandkanten, und künstliche, wie z. B. Ultraschallsensoren oder Lichtquellen. Das einzige wichtige Kriterium für einen Orientierungspunkt ist, dass seine genaue Position im globalen Koordinatensystem verzeichnet und unabhängig von Entfernung und Perspektive gleichermaßen sichtbar ist.

Eine dritte Kategorie von Lokalisierungsmethoden sind probabilistische Methoden. Mithilfe von Ableitungen des Bayes-Verfahrens wird die Zustandsschätzung des Ego-Systems unter Einbeziehung neuer Sensordaten kontinuierlich aktualisiert. Probabilistische Methoden berücksichtigen Messunsicherheiten und liefern Zustandsschätzungen mit geringeren Unsicherheiten als bei der alleinigen Nutzung von Sensor-Rohdaten. Zwei wichtige Methoden sind die Markov-Lokalisierung und die Monte-Carlo-Lokalisierung (Fox et al., 1999a, 1999b; Dellaert et al., 1999).

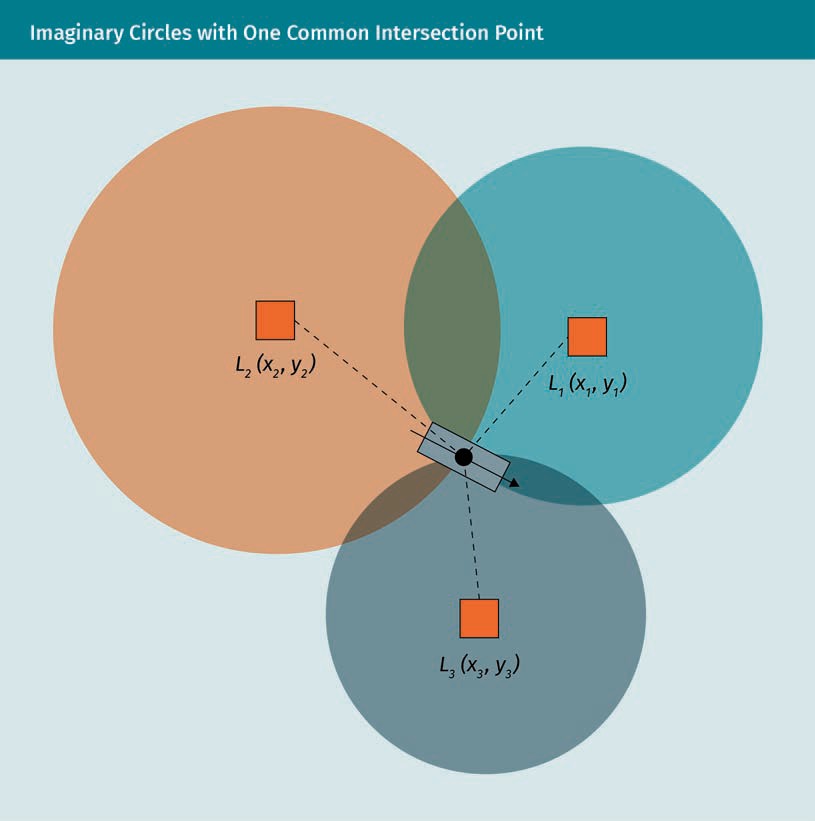
### Triangulation

Die Triangulation ist ein geometrisches Verfahren zur optischen Entfernungsmessung durch exakte Winkelmessungen von Referenzpunkten in Bezug auf das zu lokalisierende Objekt. Die Berechnung beruht lediglich auf trigonometrischen Funktionen und ist eine geeignete Methode zur Lokalisierung von automatisierten Bewegungssystemen. In der Regel werden bei dieser Methode drei Orientierungspunkte verwendet, doch ist auch die Einbeziehung einer größeren Anzahl möglich. Geeignete Orientierungspunkte können aus verschiedenen Sensordaten wie Kamerabildern und LIDAR-Scans *(Light Detection and Ranging)* extrahiert werden. Die richtige Wahl der Orientierungspunkte ist für das Funktionieren dieser Lokalisierungsmethode entscheidend.

Das folgende Beispiel zeigt, wie diese Methode funktioniert. Angenommen, wir haben drei geeignete Orientierungspunkte mit bekannten Positionen ausgemacht und können den genauen Peilwinkel zu jedem dieser Punkte bezüglich des Kurses unseres Ego-Systems messen. Dieses Szenario ist in der folgenden Abbildung dargestellt.



Gesucht wird der Schnittpunkt der drei imaginären Kreislinien, deren Mittelpunkte an den Orientierungspunkten liegen, und deren Radien die Abstände zwischen den Orientierungspunkten und der Position des Ego-Systems sind.



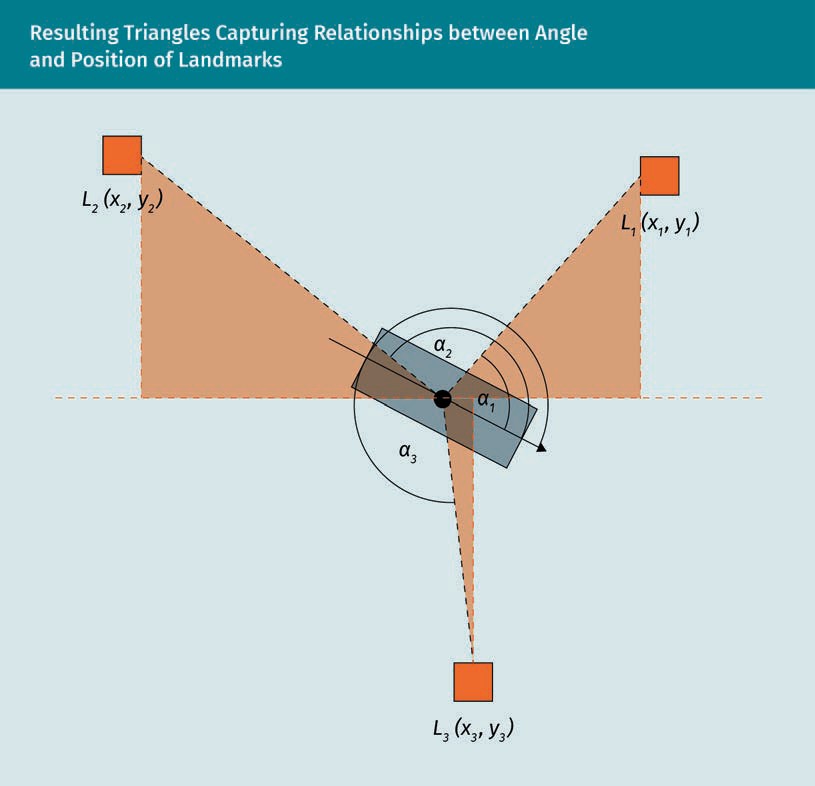
Der Zusammenhang zwischen den gemessenen Winkeln αi und der Position (xi,yi) des i-ten Orientierungspunkts kann mit Hilfe der Tangensfunktion folgendermaßen ausgedrückt werden:

tan α − θ = yi − yr

i xi − xr

(3.1)

Hierbei stehen (xr, yr) für die Position des Ego-Systems und θ für seinen Peilwinkel. Dies sind grundlegende Zusammenhänge bei Dreiecken, wie in der folgenden Abbildung dargestellt.



Hier wird auch deutlich, warum der Peilwinkel des Ego-Systems von dem zu den Orientierungspunkten gemessenen Winkeln abgezogen werden muss. Dieser Zusammenhang zwischen dem Winkel und der Position gilt nur für die Hauptachse des Koordinatensystems, auf die sich auch der Peilwinkel des Systems bezieht. Mit Hilfe der drei Tangensgleichungen in (3.1) können wir nun die Position des Systems berechnen und dabei gleichzeitig den algebraischen Fehler minimieren. Eine interessante Alternativlösung für dieses Triangulationsproblem, das auch als Drei-Punkte-Resektionsproblem bekannt ist, wurde in der Literatur von Pierlot und Van Droogenbroeck (2014) vorgestellt. Zunächst werden die folgenden modifizierten Koordinaten der Orientierungspunkte berechnet:

x′1 = x1 − x2

(3.2)

y′1 = y1 − y2

(3.3)

x′3 = x3 − x2

(3.4)

y′3 = y3 − y2

(3.5)

Anschließend werden die folgenden drei Kotangenswerte berechnet:

T12 = cot α2 − α1

(3.6)

T23 = cot α3 − α2

(3.7)

T31

= 1 − T12T23

T12 + T23

(3.8)

Dann werden die modifizierten Koordinaten der Kreismittelpunkte bestimmt:

x′12 = x′1 + T12y′1

(3.9)

y′12 = y′1 + T12x′1

(3.10)

x′23 = x′3 + T23y′3

(3.11)

y′23 = y′3 + T23x′3

(3.12)

x′31 = x′3 + x′1 + T31 y′3 − y′1

(3.13)

y′31 = y′3 + y′1 + T31 x′3 − x′1

(3.14)

Berechnung des Hilfsparameters k'31:

k′31 = x′1x′3 + y′1y′3 + T31 x′1y′3 − x′3y′1

(3.15)

Bestimmung des Nenners D und Ausgabe, falls D = 0:

D = x′12 − x′23 y′23 − y′31 − y′12 − y′23 x′23 − x′31

(3.16)

Hieraus kann nun die Position des Ego-Systems berechnet werden:

xR = x2

+ k′31 y′12 − y′23

D

yR = y2

+ k′31 x′23 − x′12

D

(3.17)

(3.18)

Offensichtlich gilt diese exakte Lösung nur dann, wenn die Unsicherheit bei den Winkelmessungen gleich Null ist. Das ist nahezu unmöglich und daher ist es notwendig, diese Winkelfehler in die Berechnungen der Position des Ego-Systems einzubeziehen. Nennen wir die tatsächlichen Winkel β = {β1, β2, β3}, im Gegensatz zu den gemessenen Winkeln α ={α1, α2, α3}.

Für den Zusammenhang zwischen dem tatsächlichen und dem gemessenen Winkel gilt:

αi = βi + ∆αi

(3.19)

Hierbei ist ∆αi der Fehler des gemessenen Winkels für den i-ten Orientierungspunkt. In der einfachsten Annahme, die in der Praxis fast immer ausreichend ist, können die Winkelfehler ∆α = (∆α1, ∆α2, ∆α3)T als Gaußsches Rauschen mit dem Mittelwert Null und der Kovarianzmatrix S = diag(σ1, σ2, σ3) dargestellt werden. In diesem Fall können wir nun eine Maximum-Likelihood-Schätzung (ML) zur Ermittlung der wahrscheinlichsten Position des Ego-Systems nutzen (Myung, 2003):

T 1 1

−1

β − α 2

x , y

= argmin

T

β − α S β − α = argmin ∑

3

i i

r r 2

2 i = 1 2

i

σ

(3.20)

Diese nichtlineare Minimierung der kleinsten Quadrate kann zum Beispiel mit dem Gauß-Newton-Algorithmus gelöst werden. Ausführliche Informationen zu diesem Thema liefert Deuﬂhard (2011).

Eine weitere mit der Triangulation verwandte Methode, die zur Lokalisierung herangezogen werden kann, ist die Trilateration. Im Gegensatz zu den Winkelmessungen beruht Triangulation auf Abstandsmessungen. Eine Erweiterung der Triangulation ist die Multilateration, bei der mehr als drei Referenzpunkte zur Positionsbestimmung verwendet werden.

### GPS

Globale Satellitennavigationssysteme (GNSS) sind Systeme zur Positionsbestimmung und Navigation, die auf dem Empfang von Signalen von Satelliten und Pseudosatelliten beruhen. GNSS ist ein Sammelbegriff für verschiedene Satellitensysteme wie NAVSTAR GPS aus den USA, GLONASS (Global Navigation Satellite System) aus der Russischen Föderation, Galileo aus der Europäischen Union und Beidou aus der Volksrepublik China (Petrovski, 2014). Das am weitesten verbreitete System ist NAVSTAR GPS, das im Folgenden als GPS bezeichnet werden soll. Die meisten bestehenden Navigationssysteme und Mobil­telefone sind mit GPS-Empfängern ausgestattet. Diese empfangen die von den Satelliten und Pseudosatelliten gesendeten Signale und erlauben so die Bestimmung der aktuellen Position. Im Allgemeinen bestehen alle GNSS aus drei Segmenten: dem Raumsegment, dem Kontrollsegment und dem Nutzersegment. Zu den Raumsegmenten gehören die Satelliten des Systems, die in einer speziellen Konstellation in einer Höhe von 20.000 bis 25.000 km unterwegs sind, um so die gewünschte Abdeckung zu gewährleisten. GPS zum Beispiel besteht aus 24 Satelliten, von denen jeder mehrere Signale für unterschiedliche Zwecke sendet. Manche Signale dienen zivilen Zwecken, andere sind dem Militär vorbehalten. Die GPS-Signale werden durch Frequenzmodulation erzeugt (El-Rabbany, 2002).

Das GPS-Kontrollsegment besteht aus einer weltweiten Infrastruktur von Ortungsstationen mit einer Hauptkontrollstation (MCS) in Colorado Springs in den Vereinigten Staaten. Ihr Zweck ist die Verfolgung aller Satelliten zur Bestimmung der Position, des Almanachs und anderer Daten. Um die Informationen aktuell zu halten, wird mindestens alle sechs Tage ein neuer Almanach übermittelt.

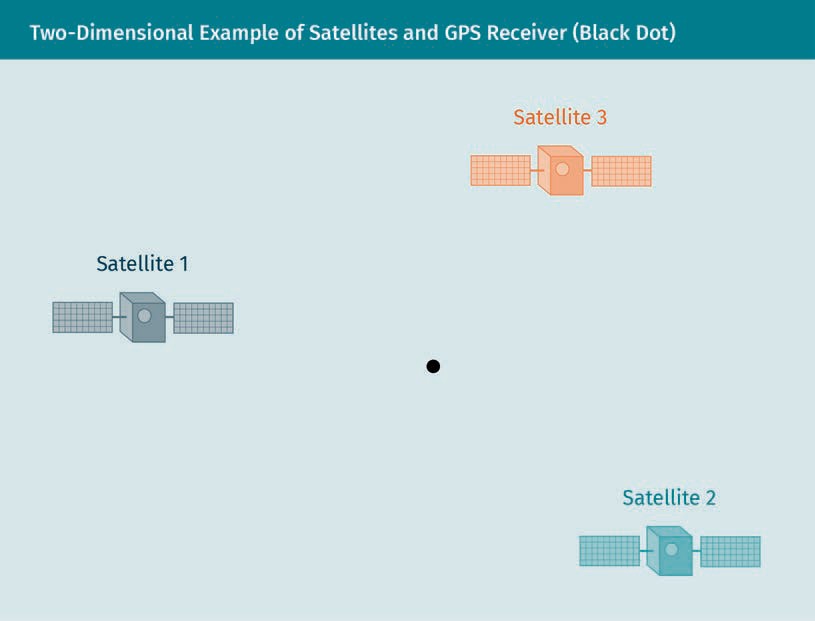
Alle bestehenden GPS-Empfänger und -Antennen bilden zusammen das Nutzersegment von GPS. Im Folgenden wird untersucht, wie die Position eines GPS-Empfängers anhand der Satelliteninformationen bestimmt werden kann. Häufig wird das zugrunde liegende Konzept fälschlicherweise als Triangulation bezeichnet. Es wird nämlich ein Prinzip namens Trilateration verwendet, bei dem nicht der Winkel, sondern die Entfernung gemessen wird. Vereinfachen wir das Satellitensystem auf die zweidimensionale euklidische Ebene und stellen uns vor, unser Empfänger würde zusammen mit drei Satelliten in einer horizontalen Ebene aufgestellt, wie in der folgenden Abbildung dargestellt.

Pseudoliten

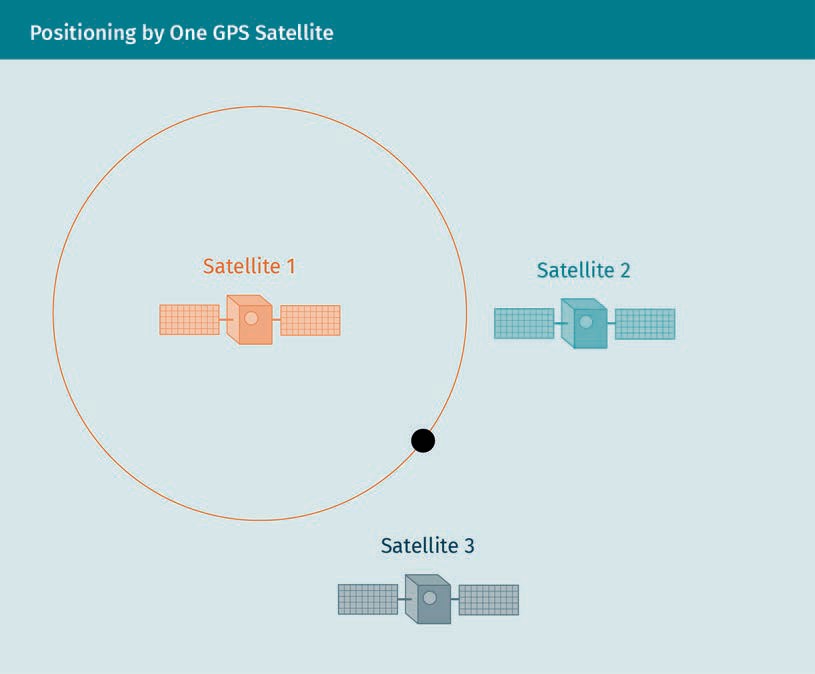
Ein Pseudolit ist ein Sender auf der Erde, der Signale sendet, die den von Satelliten ausgestrahlten Signalen ähneln.

Almanach

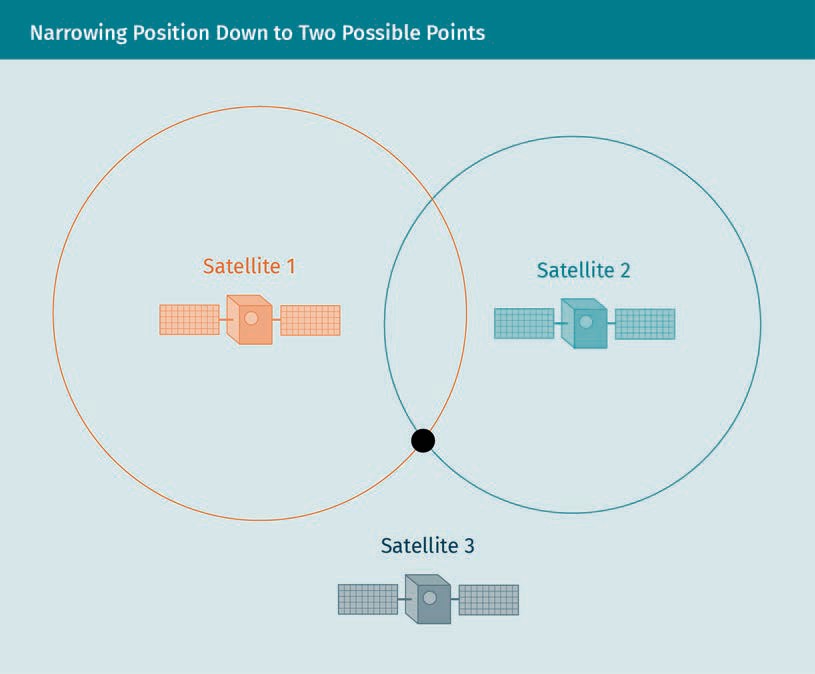
Der Almanach enthält Orbitalinformationen in geringer Auflösung und wird mit 50 Bit/s übertragen.



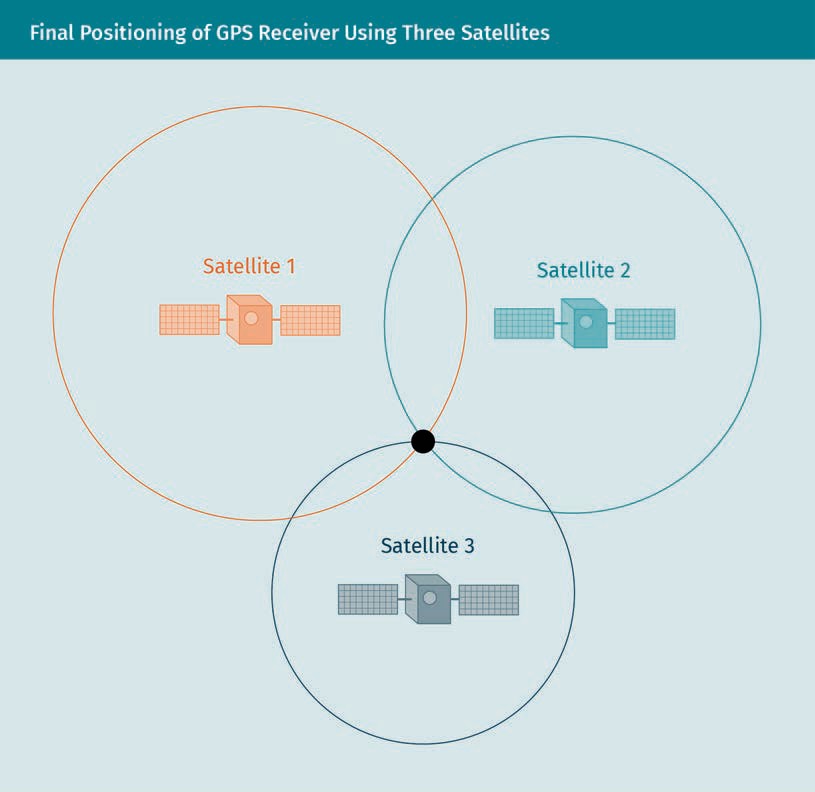
Jeder Satellit sendet ein Signal gleichmäßig in alle Richtungen aus, welches neben anderen Daten auch den Zeitpunkt der Aussendung enthält. Die Entfernung zu einem bestimmten Satelliten lässt sich ermitteln, indem die Differenz zwischen dem Zeitpunkt des Absendens und dem Zeitpunkt des Empfangs berechnet und dann mit der Lichtgeschwindigkeit multipliziert wird. Die berechnete Entfernung wird als Pseudo-Entfernung bezeichnet und ist aufgrund von Ungenauigkeiten der GPS-Empfängeruhr fehlerhaft. Vernachlässigen wir zunächst diese Ungenauigkeiten und nehmen an, dass unser GPS-Empfänger das Signal von Satellit 1 empfängt. Wir können die jeweilige Pseudo-Entfernung berechnen, wissen aber auch, dass das Signal in alle Richtungen gleichmäßig ausgestrahlt wird. Unsere Position könnte also überall auf einem Kreis mit dem Radius der berechneten Pseudo-Entfernung liegen, wie die folgende Abbildung zeigt.



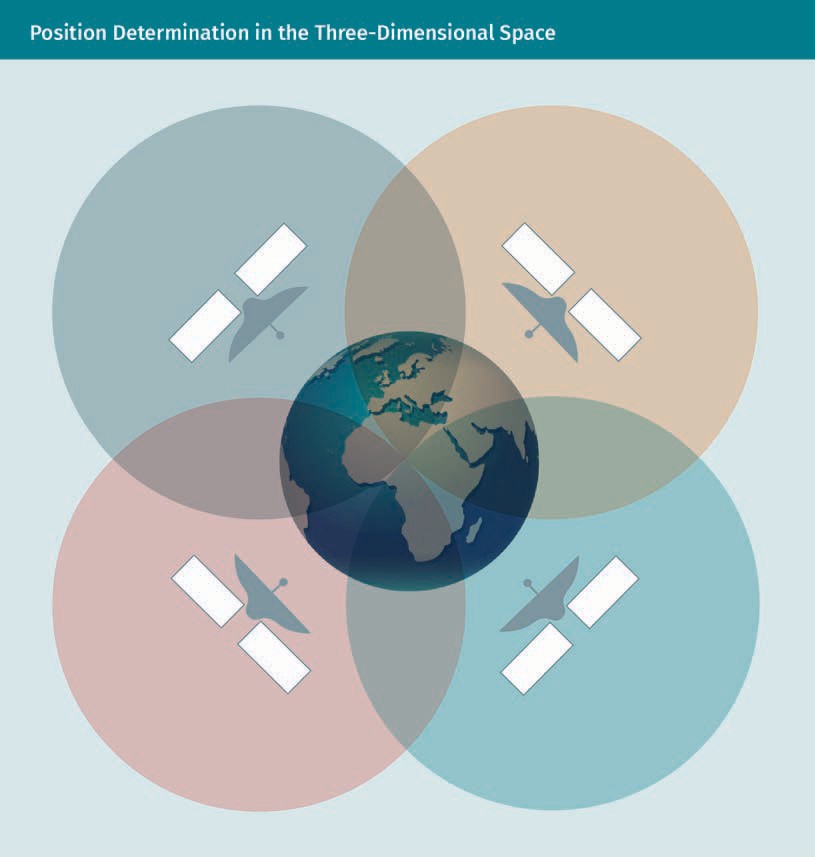
Wenn wir das Signal eines zweiten Satelliten einbeziehen, bildet die berechnete Pseudo-Entfernung einen neuen Kreis um den betreffenden Satelliten. Jetzt können wir die Position unseres GPS-Empfängers auf die beiden Schnittpunkte der zwei imaginären Kreise eingrenzen, wie in der folgenden Abbildung dargestellt.



Wenn wir schließlich ein drittes Signal berücksichtigen, können wir unsere Position als Schnittpunkt aller drei imaginären Kreise genau bestimmen. Die folgende Abbildung zeigt die endgültige Positionsbestimmung unseres GPS-Empfängers.



Die Eingrenzung des Positionierungsproblems auf zwei Dimensionen stellt das Prinzip vereinfacht dar. Es kann jedoch direkt auf den dreidimensionalen Raum ausgedehnt werden. Die von den Satelliten ausgestrahlten Signale sind Kugeln mit den Satellitenpositionen als Mittelpunkte und den berechneten Pseudo-Entfernungen als Radien. Anstelle des Schnittpunkts von drei Kreisen legt jetzt der Schnittpunkt von vier Kugeln die aktuelle Position eines GPS-Empfängers fest, wie in der folgenden Abbildung dargestellt.



Wie kann dieser Schnittpunkt berechnet werden? Wie bereits erwähnt, werden für eine genaue Bestimmung vier Satelliten benötigt. Daher müssen vier Navigationsgleichungen aufgestellt werden, die wie folgt formuliert werden können:

X − R 2 + Y − R 2 + Z − R 2 = P − c · ∆T 2

1 X 1 Y 1 Z 1 B

X − R 2 + Y − R 2 + Z − R 2 = P − c · ∆T 2

2 X 2 Y 2 Z 2 B

(3.21)

X − R 2 + Y − R 2 + Z − R 2 = P − c · ∆T 2

3 X 3 Y 3 Z 3 B

(3.22)

X − R 2 + Y − R 2 + Z − R 2 = P − c · ∆T 2

4 X 4 Y 4 Z 4 B

(3.23)

(3.24)

Die X-, Y- und Z-Positionen des GPS-Empfängers werden durch die Unbekannten RX, RY und RZ dargestellt. Pi ist die genaue Entfernung zwischen dem Empfänger und dem i-ten Satelliten, ∆TB steht für die unbekannte Abweichung der Uhr des Empfängers von den Uhren der GPS-Satelliten und c ist die Lichtgeschwindigkeit. Es wird angenommen, dass die Satellitenuhren viel genauer sind als die Uhr des Empfängers und dass die Uhren aller Satelliten synchron sind. Obwohl die Lösung der Gleichungen aufgrund ihrer Nichtlinearität nicht einfach ist, gibt es verschiedene Methoden, sie zu lösen, z. B. die Methode der kleinsten Quadrate, iterative Methoden wie der Gauss-Newton-Algorithmus, geschlossene Lösungen wie die Bancroft-Methode und Beobachtungsgewichtungsmodelle (Rahemi et al., 2014; Bancroft, 1985; Tay & Marais, 2013). Wir zeigen hier, wie das System nichtlinearer Gleichungen numerisch mit der Newton-Methode, auch bekannt als Newton-Raphson-Methode, gelöst werden kann (Ryaben'kii & Tsynkov, 2006). Im Allgemeinen handelt es sich um einen Lösungsalgorithmus, mit dem Funktionen näherungsweise gelöst werden. Wir müssen das Problem folgendermaßen umformulieren: F(RX, RY,RZ, ∆TB) = 0 für eine Funktion F. Die Funktion wählen wir wie folgt:

(RX, RY, RZ, ∆TB = f1 RX, RY, RZ, ∆TB , f2 RX, RY, RZ, ∆TB , f3 RX, RY, RZ, ∆TB , f4 RX, RY, RZ, ∆TB

(3.25)

Hierbei gilt:

f R , R ,R , ∆T =

X − R 2 + Y − R 2 + Z − R

2

1 X

1 Y

1 Z

+ c · ∆T − P

1. X Y Z B

B 1

(3.26)

f R , R

,R , ∆T =

+ c · ∆T − P

1. X Y Z B B 2

X − R 2 + Y − R 2 + Z − R

2

2 X

2 Y

2 Z

(3.27)

f R , R ,R , ∆T =

X − R 2 + Y − R 2 + Z − R

2

3 X

3 Y

3 Z

+ c · ∆T − P

1. X Y Z B

B 3

(3.28)

f R , R ,R , ∆T =

X − R 2 + Y − R 2 + Z − R

2

4 X

4 Y

4 Z

+ c · ∆T − P

1. X Y Z B

B 4

(3.29)

Wir haben unsere Funktion F nun so geschrieben, dass sie mit dem Newton-Verfahren gelöst werden kann. Durch Iterieren der folgenden Rekursion kann die Nullstelle von F mit einer Anfangsschätzung für die Lösung x0 = (RX0, RY0, RZ0, ∆TB0) approximiert werden:

)n + 1 = )n − J−1 )n ( )n

Dabei ist J die Jacobimatrix, die jede partielle Ableitung enthält. Die Berechnung von J ergibt sich wie folgt:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Nach Ableitung aller erforderlichen Terme können die Nullstellen der Funktion durch Iteration der Rekursion bis zu einer zuvor festgelegten Genauigkeit berechnet werden. Dies kann leicht mittels einer Schleife durchgeführt werden, wobei der rechenintensivste Teil die Berechnung der Inversen der Jacobimatrix ist. Anstatt die Umkehrung der Jakobimatrix zu berechnen, kann auch das folgende lineare Gleichungssystem für die Unbekannte xn+1 – xn gelöst werden:

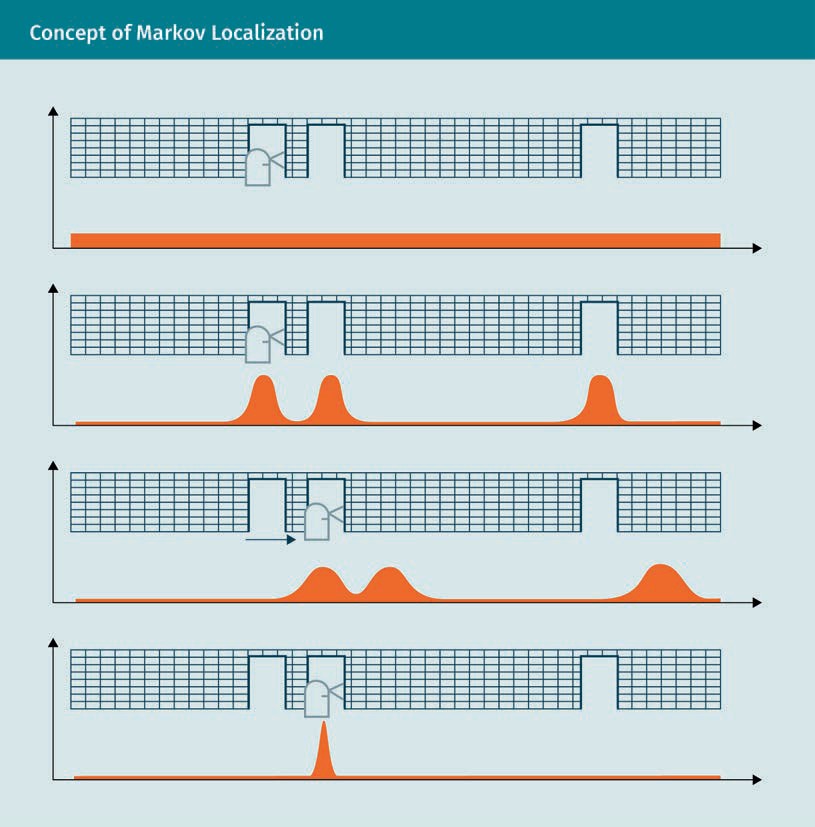
J)n )n + 1 −)n = − ( )n

(3.31)

### Probabilistische Lokalisierung

Probabilistische Lokalisierungsmethoden stehen seit Langem im Mittelpunkt der Forschung. Der Kerngedanke besteht darin, alle früheren Zustandsinformationen zur Schätzung des aktuellen Zustands zu verwenden. Zwei der am häufigsten verwendeten und in diesem Abschnitt vorgestellten Ansätze sind die Markov-Lokalisierung und die Partikel-Lokalisierung, die auch als Monte-Carlo-Lokalisierung bekannt ist. Bei beiden Methoden wird die Markov-Annahme getroffen. Dies bedeutet, dass die Umgebung, in der sich unser Ego-System befindet, als statisch betrachtet wird (Durrett, 2019). Mit dieser Annahme ist gewährleistet, dass der Standort des Systems der einzige Zustand in der Umgebung ist, der die Sensormessungen beeinflusst.

Was ist die Grundidee der Markov-Lokalisierung? Im Allgemeinen wird das Problem der Ego-Zustandsabschätzung mit Hilfe von Sensorinformationen betrachtet. Anstelle einer einzigen Hypothese über den aktuellen Standort des Systems propagiert der Markov-Lokalisierungsansatz eine Wahrscheinlichkeitsverteilung über den Raum aller Zustandshypothesen (Frontoni, 2012). Auf diese Weise werden zusätzlich zu den mittleren Schätzungen auch Unsicherheiten erfasst. Die Idee des Lokalisierungsansatzes ist in der folgenden Abbildung zu sehen, in der die Lokalisierung auf eine Dimension vereinfacht wird. Hier ist der blaue Roboter das zu lokalisierende Objekt, und die Überzeugungsverteilung unter jedem Umgebungsbild zeigt die Hypothesen des aktuellen Zustands. Die Abbildung zeigt vier aufeinanderfolgende Messschritte von oben nach unten.



In diesem Beispiel besteht der Zustand des Roboters nur aus der Position in einer Richtung. Wenn der Roboter in der Umgebung mit einer unbekannten Position platziert wird, ist die anfängliche Ungewissheitsdarstellung der Markov-Lokalisierungsmethode eine einheitliche Überzeugungsverteilung über alle möglichen Positionen, die ganz oben in der vorigen Abbildung dargestellt ist. Wenn die Sensoren des Roboters messen, dass sich die Position in der Nähe einer Tür befindet, wird die Wahrscheinlichkeitsverteilung geändert, indem die Wahrscheinlichkeit für die Position in der Nähe von Türen erhöht und für alle anderen Positionen verringert wird, wie im zweiten Diagramm der Abbildung dargestellt. Es gibt nun drei Haupthypothesen, d. h. die aktuelle Überzeugung ist multimodal. Wenn sich der Roboter bewegt, verschiebt sich auch die Überzeugungsverteilung, wie im dritten Diagramm dargestellt. Außerdem ist zu beobachten, dass die Unsicherheit der einzelnen Hypothesen zunimmt. So werden die Überzeugungsverteilungen durch die Bewegungsunsicherheit des Roboters breiter. Ohne zusätzliche Messungen wächst die Unsicherheit mit jedem Zeitschritt. Wenn der Roboter eine zweite Messung der Umgebung vornimmt und erneut seine Position neben einer Tür feststellt, wird die aktuelle Information in die vorherige Überzeugung einbezogen, wie das vierte Diagramm der Abbildung zeigt. Auf diese Weise reduziert der Markov-Lokalisierungsansatz seine Hypothesen auf eine einzige Zustandsschätzung.

Untersuchen wir nun die Markow-Lokalisierung auf eine mathematischere Weise. Wir wollen eine Lokalisierung in der euklidischen Ebene annehmen und unseren Zustand zum Zeitschritt t darstellen als |xt| = (x y θ)T*,* wobei x und y die Position in x- und y-Richtung darstellen, und θ die Orientierung. Xt (beachten Sie das große X) ist eine dreidimensionale Zufallsvariable. Anstatt den aktuellen Zustand genau zu kennen, behält das Ego-System eine Überzeugung Bel(Xt) über den gesamten Zustandsraum bei. Bel(Xt = xt) gibt die aktuell angenommene Wahrscheinlichkeit an, dass das System den Zustand x zum Zeitschritt t hat. Wir haben in der Einführung der Methode gesehen, dass die Überzeugung bei zwei Arten von Ereignissen aktualisiert werden kann. Entweder stehen neue (exterozeptive) Sensordaten zur Verfügung, die neue Informationen über die Umgebung liefern (im Folgenden mit ze bezeichnet), oder es treffen Messungen von propriozeptiven Sensoren ein (im Folgenden mit zp bezeichnet), z. B. Odometriedaten oder IMU-Messungen. Wir nehmen an, dass unser System einen Datenstrom d = (d0, d1, ..., dT) mit 0 ≤ t ≤ T empfängt (die Zeit t wird als Index der Messungen verwendet), der sowohl aus extero- als auch aus propriozeptiven Sensormessungen besteht. Die Messung mit dem Index T ist die letzte, die empfangen wurde. Ein Aspekt der Markov-Lokalisierung ist, dass die Schätzung der A-posteriori-Zustandsverteilung über alle verfügbaren Daten konditioniert werden muss (Frontoni, 2012). Daraus ergibt sich die folgende Gleichung:

P +T = ) d = P +T = ) d0, d1, …, dT

(3.33)

Der zweite Teil der Gleichung kann als „die Wahrscheinlichkeit von Xt = x unter der Bedingung der Daten d0, d1, ..., dT“ verstanden werden. Die bereits erwähnte Markov-Annahme impliziert, dass die gesamte Vergangenheit, also alle früheren Zustände, im aktuellen Zustand erfasst werden. Dies bedeutet, dass zusätzliches Wissen über frühere Daten keine Auswirkungen hat. Die Markov-Annahme kann in unserem Fall wie folgt formuliert werden:

P dt + 1, dt + 2,… +T = ), d0, d1, …, dT = P dt + 1, dt + 2,… +T = ) , 0 ≤ t ≤ T

(3.34)

In Worten lässt sich dies folgendermaßen ausdrücken: „Die Wahrscheinlichkeit für die Daten dt+1, dt+2, ... unter der Bedingung der vorherigen Daten d0, d1, ..., dT und dass XT = x gleich der Wahrscheinlichkeit für die Daten dt+1, dt+2, ... unter der Bedingung XT = x ist.“

Betrachten wir nun die Aktualisierung der Zustandsüberzeugung (engl. *belief*), wenn neue Sensorinformationen über die Umgebung verfügbar werden und somit d = (d0, d1, ..., zeT) gilt. Dies bedeutet, dass der Datenstrom alle zuvor gesammelten Messungen sowie die aktuelle Messung enthält. Die angenommene Wahrscheinlichkeit, der aktuelle Zustand sei XT = x, entspricht nun der Wahrscheinlichkeit, dass der aktuelle Zustand XT = x unter der Bedingung aller verfügbaren Messdaten d ist:

Bel +T = ) = P +T = ) d = P +T = ) d0, d1, …, dT − 1, zeT

(3.35)

Der Satz von Bayes

P B A = P A B P B

P A

(3.36)

(wobei A und B zwei Ereignisse bezeichnen) und die Markov-Annahme führen zur folgenden Darstellung der A-posteriori-Verteilung auf der rechten Seite der Gleichung:

Bel +T = ) =

P zeT d0, d1, …, dT − 1, +T = ) P +T = ) d0, d1, …, dT − 1

P z

eT

d , d , …, d

0 1 T − 1

=

P zeT +T = ) P +T = ) d0, d1, …, dT − 1

P z

eT 0 1 T − 1

d , d , …, d

(3.37)

Der Nenner P(zeT | d0, d1, ..., dT-1) hängt nicht vom Zustand x ab und kann daher durch einen konstanten Faktor für den Zähler αT ersetzt werden:

Bel +T = ) = αTP zeT +T = ) P +T = ) d0, d1, …, dT − 1

(3.38)

Die Wahrscheinlichkeitsverteilung P(XT = x|d0, d1, ..., dT-1) entspricht dem rechten Teil von Gleichung (3.3), ohne Berücksichtigung der letzten Messung. Wir können also die vorherige Überzeugung wie folgt darstellen:

Bel +T − 1 = ) = P +T = ) d0, d1, …, dT − 1

(3.39)

Einsetzen in Gleichung (3.6) ergibt:

Bel +T = ) = αTP zeT +T = ) Bel +T − 1 = )

(3.40)

Daher besteht die A-posteriori-Überzeugung nach Einbeziehung der Sensormessung zeT, die Informationen über die Umgebung liefert, aus dem Produkt der vorherigen Annahme, der Wahrscheinlichkeit für die aktuelle Messung (auch Wahrnehmungsmodell genannt) und einer Konstanten. Daraus ergibt sich der erste Teil der Markov-Lokalisierungsmethode.

Die Berechnung der A-posteriori-Überzeugung (wenn sich das Ego-System bewegt und neue Sensordaten sammelt, die Informationen über die Bewegung des Systems liefern), funktioniert auf andere Weise. Der Datenstrom kann in diesem Fall durch d = (d0, d1, ..., zpT) dargestellt werden. Wie zuvor wollen wir Folgendes berechnen:

Bel +T = ) = P +T = ) d

(3.41)

Wir wenden nun den Satz der Gesamtwahrscheinlichkeit auf den rechten Teil der Gleichung an (Zwillinger & Kokosa, 2000):

Bel +T = ) = ∫P +T = ) d, +T − 1 = )′ P +T − 1 = )′ d d)′

(3.42)

Einsetzen von d = (d0, d1, ..., zpT) in die Wahrscheinlichkeitsverteilungen des Integrals ergibt:

Bel +T = ) = ∫P +T = ) d0, d1, …, zpT, +T − 1 = )′ P +T − 1 = )′ d0, d1, …, zpT d)′

(3.43)

Die erste Wahrscheinlichkeitsfunktion kann unter Berücksichtigung der Markov-Annahme folgendermaßen vereinfacht werden:

Bel +T = ) = ∫P +T = ) zpT, +T − 1 = )′ P +T − 1 = )′ d0, d1, …, zpT d)′

(3.44)

Da die Messung zpT keine Informationen über den vorherigen Zustand XT–1 enthält, kann die Gleichung wie folgt vereinfacht werden:

Bel +T = ) = ∫P +T = ) zpT, +T − 1 = )′ P +T − 1 = )′ d0, d1, …dT − 1 d)′

(3.45)

Wie im vorherigen Fall kann die Wahrscheinlichkeitsverteilung P(XT – 1 = x’|d0, d1, ... dT – 1) nun durch Bel(XT – 1 = x’) dargestellt werden, und die Substitution in Gleichung (3.13) ergibt:

Bel +T = ) = ∫P +T = ) zpT, +T − 1 = )′ Bel +T − 1 = )′ d)′

(3.46)

Daher besteht die A-posteriori-Überzeugung nach Einbeziehung einer neuen Messung des Bewegungssensors zpT aus dem Integral des Produkts aus der vorherigen Überzeugung Bel(XT–1 = x’) und dem Bewegungsmodell P(XT = x|zpT, XT – 1 = x’) des Systems. Daraus ergibt sich der zweite Teil der Markov-Lokalisierungsmethode.

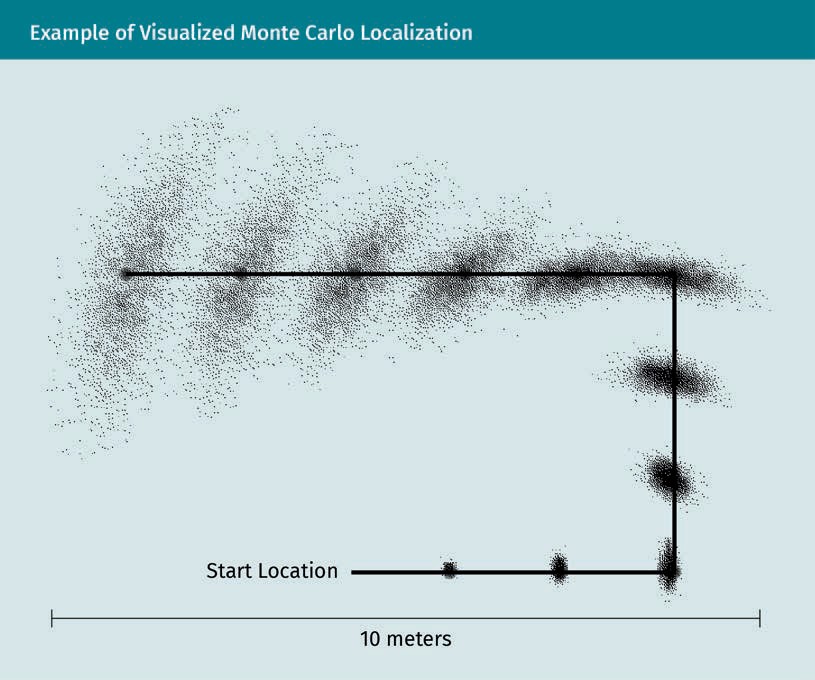
Die Gleichungen (3.8) und (3.14) bilden den Kern des Markov-Lokalisierungsalgorithmus. Die Initiierung des Algorithmus erfolgt meist auf zwei Arten. Wenn der Zustand des Systems völlig unbekannt ist, wird für die A-priori-Verteilung P(X0) eine Gleichverteilung angenommen. Im Falle eines ungefähr bekannten Ausgangszustands wird eine Gauß-Verteilung mit geringer Unsicherheit verwendet (Fox, 1999a). Im Folgenden ist ein Pseudocode für den Markov-Lokalisierungsalgorithmus dargestellt.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein großer Vorteil der Markov-Lokalisierung besteht darin, dass sie auch dann funktioniert, wenn keine anfänglichen Informationen über den Zustand des Systems vorhanden sind. Die anfängliche Schätzung ist das Hauptproblem bei alternativen Algorithmen, wie z. B. dem Kalman-Verfahren, die für das Lokalisierungsproblem genutzt werden können (Frontoni, 2012). Lokalisierungsfehler, bei denen Informationen über den aktuellen Zustand im Laufe der Zeit verloren gehen, lassen den Lokalisierungsalgorithmus nicht divergieren, wie es bei Kalman-Filtern der Fall ist. Ein weiterer großer Vorteil ist die Möglichkeit, mehrere Hypothesen über den Zustand zu propagieren. Wenn also mehrere Zustände wahrscheinlich sind, behält die Markov-Lokalisierungsmethode diese im Auge. Die Annahme einer statischen Umgebung ist dagegen eine sehr starke Einschränkung, die zu Problemen führt, wenn die Lokalisierung in dynamischen Umgebungen getestet wird, z. B. wenn andere sich bewegende Objekte, wie Personen, vorhanden sind. Ein weiteres Problem ist die am häufigsten verwendete topologische Darstellung des Zustandsraums, die zu einer geringen Auflösung der Zustände führt. Diesem Problem kann durch gitterbasierte Darstellungen begegnet werden, die jedoch sehr speicherintensiv sind.

Eine zweite gängige Lokalisierungsmethode ist die Monte-Carlo-Lokalisierung. Die Darstellung der Umgebung spielt für Lokalisierungsmethoden eine wichtige Rolle und kann als kontinuierlicher Zustandsraum betrachtet werden, weil der Zustand aus reellwertigen Zufallsvariablen besteht. Die Markov-Lokalisierung ist sehr speicherintensiv, wenn die A-posteriori-Dichten in einem kontinuierlichen Zustandsraum betrachtet werden. Deshalb ist sie ungeeignet, wenn eine hohe Genauigkeit erforderlich ist. Die Monte-Carlo-Lokalisierung, die auch als Partikelfilter bekannt ist, beschreibt die Dichte auf andere Weise. Sie nutzt eine Reihe von Stichproben, die aus ihr gezogen und als Partikel bezeichnet werden. Auf diese Weise wird der kontinuierliche Zustandsraum diskretisiert, um den Speicherbedarf zu verringern. Außerdem ist die Stichprobenentnahme aus einem solchen diskretisierten Zustandsraum einfacher als aus einem kontinuierlichen Raum (Frontoni, 2012). Wie die Markov-Lokalisierungsmethode ist ein Partikelfilter in der Lage, mehrere Hypothesen darzustellen, wenn mehrere verschiedene Zustände des Systems berücksichtigt werden müssen. Die Wahrscheinlichkeitsverteilung für einen bestimmten Zustand besteht aus einer Wolke von Partikeln, wobei eine hohe Konzentration eine hohe Wahrscheinlichkeit anzeigt. Die folgende Abbildung zeigt beispielhaft die Visualisierung einer Monte-Carlo-Lokalisierung.



Es ist deutlich zu erkennen, wie die Positionsunsicherheit mit der Zeit zunimmt. Am Startort sind alle Partikel auf einen Punkt konzentriert. Mit jedem weiteren Zeitschritt werden die Partikel über ein größeres Gebiet verteilt, was auf eine höhere Lokalisierungsunsicherheit hinweist. Die Möglichkeit, mehrere Hypothesen aufrechtzuerhalten, ist von großer Bedeutung, da oftmals der Ausgangszustand des Systems unbekannt ist. Dies ist auch im Falle von Lokalisierungsfehlern wichtig, bei denen der Zustand im Laufe der Zeit verloren geht, was dem Fall des unbekannten Anfangszustands ähnelt. Die Monte-Carlo-Lokalisierung ist eine geeignete Methode für die globale Lokalisierung, wenn die oben genannten Fälle vorliegen.

Im Gegensatz zur Markov-Lokalisierungsmethode besteht der Partikel-Algorithmus aus einem Aktualisierungsschritt, in den neue Messungen und eine Vorhersage nach einem bestimmten Bewegungsmodell einfließen. Er basiert auf rekursivem Filtern nach Bayes mit folgendem Übergangsmodell:

p )t + 1 )t

(3.47)

Es modelliert den Übergang von Zustand xt zu xt+1 sowie das Beobachtungsmodell

p zt )t

(3.48)

Das letztere gibt die Wahrscheinlichkeit für die Messung zt unter Berücksichtigung des Zustands xtan. Eine genaue Einführung in die Bayes-Statistik und die Bayes-Filter findet sich bei Chen (2003). Die Grundlage für das Bayes-Filtern ist der Satz von Bayes:

P B A = P A B P B

∑AP B A P A

(3.49)

Damit können die Gleichungen für die A-priori- und A-posteriori-Dichten erhalten werden. Die A-priori-Dichte ist wie folgt definiert:

p )t + 1 = ∫p )t + 1 )t p )t zt d)t

(3.50)

Die A-posteriori-Dichte zum Zeitschritt t + 1 kann wie folgt geschrieben werden:

p )t + 1 zt + 1 ∝ p zt + 1 )t + 1 ·p )t + 1

(3.51)

Dies entspricht dem Produkt aus der Wahrscheinlichkeit der Messung zt+1 und der A-priori-Dichte. Dieselben Gleichungen bilden auch die Grundlage des Kalman-Filters, welcher der optimale lineare Filter für Gaußsche Übergangs- und Beobachtungsmodelle ist. Die Monte-Carlo-Lokalisierung funktioniert für Fälle, in denen eine solche Annahme nicht zutrifft. Wie bereits erwähnt, wird die A-posteriori-Dichte zum Zeitpunkt t dargestellt durch eine Menge von N Partikeln

)i mit bestimmten zugehörigen Wahrscheinlichkeitsgewichten wi. Während jedes Partikel für einen

t t

Zustand steht, ist die Summe der Gewichte aller Partikel gleich eins, was eine Annäherung an die Dichte darstellt.

Wie bei der Markov-Lokalisierungstechnik wird eine anfängliche Schätzung der Partikelverteilung benötigt. In völlig unbekannten Umgebungen wird sie deshalb als Gleichverteilung gewählt. Anstatt die gesamte Gleichverteilung über den kontinuierlichen Zustandsraum zu erfassen, wird eine Anzahl von Partikeln aus der Verteilung gezogen. Dies führt zu Partikeln mit gleichen Gewichten und somit ergibt eine Stichprobe von N Partikeln für jedes Partikel ein Gewicht von

wi = 1

t N

(3.52)

Die diskretisierte A-posteriori-Dichte zum Zeitschritt t + 1 kann nun durch folgende empirische Schätzung dargestellt werden (Frontoni, 2012):

N

p )t + 1 zt + 1 = ∑wiδ )t + 1 − )i

t

i = 1

t + 1

(3.53)

Dabei ist

δ )t + 1 − )i

t + 1

(3.54)

eine auf dem i-ten Teilchen zentrierte Dirac-Delta-Funktion:

i

)

t + 1

(3.55)

Die Dirac-Delta-Funktion hat den Wert 1 bei

)t + 1 = )i

t + 1

(3.56)

und ist überall sonst gleich Null. Durch Aufsummieren der N gewichteten Dirac-Delta-Funktionen aller Partikel erhalten wir die diskretisierte A-posteriori-Dichte. Das Integral der A-priori-Dichte wird mittels Diskretisierung durch eine leichter zu berechnende Summe ersetzt, weil somit keine analytische Lösung der Gleichung notwendig ist. Wir können sie damit wie folgt schreiben:

N

p )t = ∑wip )t + 1 )i

t t

i = 1

(3.57)

Dies wird auch als Mischdichte bezeichnet (Frontoni, 2012). Die Berechnung der A-posteriori-Dichte zum Zeitpunkt t + 1 erfordert den berechneten A-priori-Wert p(xt+1). Dies wäre im stetigen Raum schwierig, ist nach Diskretisierung jedoch sehr einfach. Die Mischdichte kann direkt in Gleichung (3.51) eingesetzt werden, und wir erhalten:

N

p )t + 1 zt + 1 ∝ p zt + 1 )t + 1 ∑wip )t + 1 )i

t t

i = 1

(3.58)

Diese Summen können nun mit viel geringerem Speicher- und Rechenaufwand berechnet werden. In der Initialisierungsphase des Partikel-Algorithmus zum Zeitschritt t = 0 wird eine zufällige Stichprobenmenge von N Partikeln aus einer Gleichverteilung gezogen:

+0 = )i

0

(3.59)

Der erste Schritt ist die Vorhersage: Hierbei werden Stichproben aus dem Übergangsmodell gezogen:

p )t + 1 )i

t

(3.60)

Diese bilden eine neue Partikelmenge:

+′t + 1 = )′i

t + 1

(3.61)

Dies stellt die A-priori-Dichte p(xt+1) dar. Anschließend die Zustandsschätzung durch Berechnung der A-posteriori-Dichte mit neuen Messdaten zt+1 aktualisiert. Das Gewicht jedes Partikels muss anhand der Wahrscheinlichkeit der Messung des jeweiligen Partikels aktualisiert werden. Die geschieht wie folgt:

wi = p y

t + 1

t + 1

i

t + 1

)′

(3.62)

Das neue Gewicht des i-ten Partikels ist also genau diese Wahrscheinlichkeit. Wir haben nun eine gewichtete, noch nicht normalisierte Partikelmenge:

)′i , wi

t + 1 t + 1

(3.63)

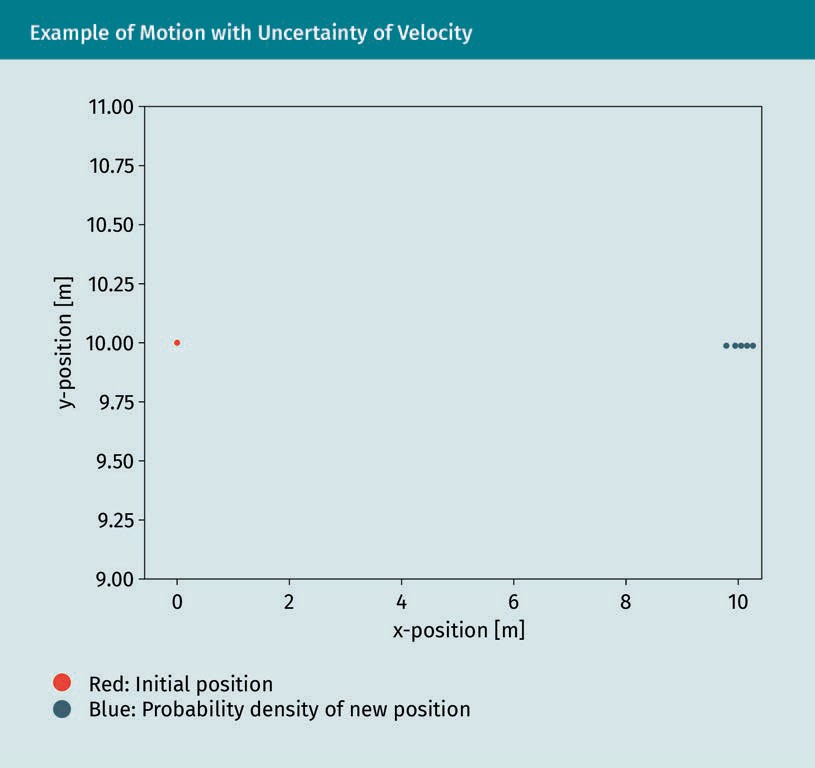
Daraus ziehen wir neue Stichproben und erhalten eine Menge Xt+1. Partikel mit einem größeren Gewicht werden mit größerer Wahrscheinlichkeit und damit häufiger gezogen als solche mit geringerem Gewicht. Anschließend wird die neue Partikelmenge so normiert, dass sie die A-posteriori-Dichte zum Zeitpunkt t + 1 darstellt.

Trotz der oben genannten Vorteile gibt es beim Partikelfilter auch Probleme. Wenn sich die Messwahrscheinlichkeit und die A-priori-Dichte nur in geringem Maße überschneiden, kann es sein, dass die Stichprobenziehung des A-priori-Werts zur Bildung des A-posteriori-Werts nicht genügend Partikel im Überlappungsbereich erzeugt. Aus diesem Grund kann es vorkommen, dass die A-posteriori-Dichte eine geringe Genauigkeit aufweist. Ein weiteres Problem ist die Darstellung des Beobachtungsmodells, das oft nicht vorliegt und deshalb aus den Sensormessungen geschätzt werden muss. Dennoch wird das Partikelverfahren bei Lokalisierungs- und Verfolgungsaufgaben häufig eingesetzt, wenn die Gaußsche Annahme für die zugrunde liegenden Übergangs- und Beobachtungsmodelle nicht zutrifft.

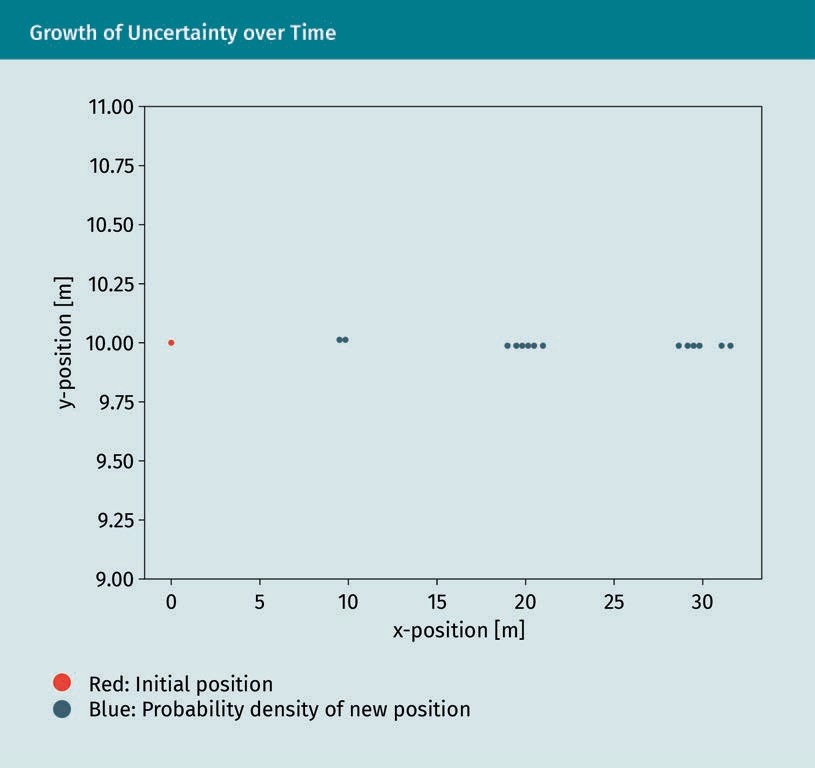
### Unsicherheit der Bewegung

Wenn ein System seinen Zustand, also seine Position und Orientierung, in seinem Zustandsraum ändert, ist die Bewegung immer mit Unsicherheit behaftet. Die Ursachen hierfür können in externe und interne unterteilt werden. Äußere Einflüsse können der unebene Boden sein, auf dem sich das System bewegt, oder der Wind, der es von seiner gewünschten Position wegtreibt. Interne Einflüsse, die also vom bewegten System selbst ausgehen, können Unterschiede bei den Rädern oder im Axialspiel sein. Diese verhindern, dass es sich in den gewünschten neuen Zustand bewegt. Außerdem kann die automatische Längs- und Quersteuerung nur mit einer gewissen Genauigkeit erfolgen. Daher sind der berechnete Lenkwinkel und die Geschwindigkeit ebenfalls mit Unsicherheiten behaftet.

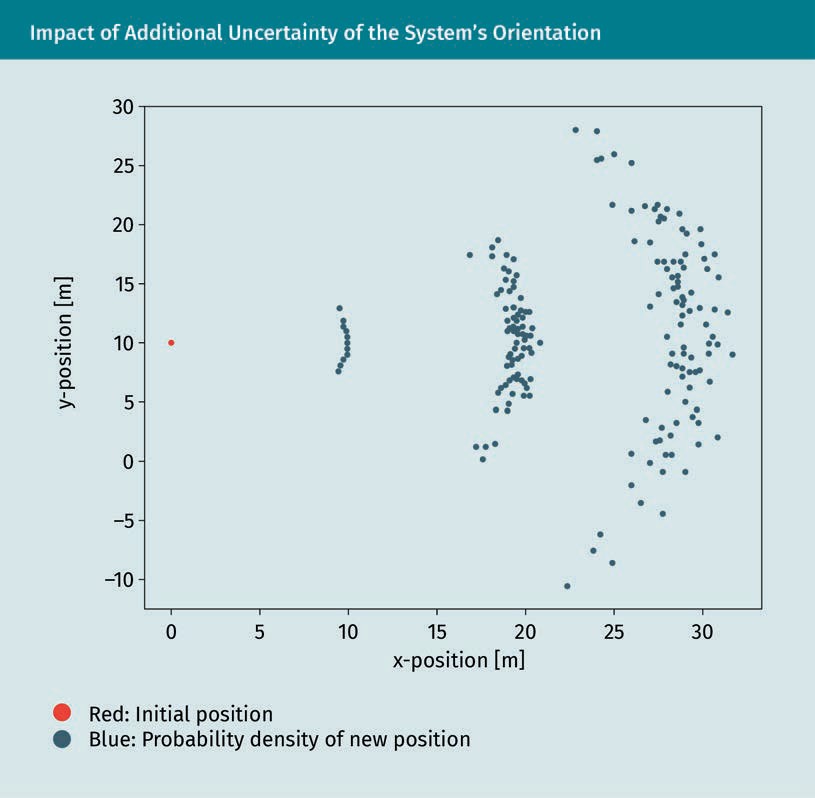
Untersuchen wir die Unsicherheit der Bewegung eines Systems im Detail. Angenommen, unser Ego-System bewege sich in der euklidischen Ebene und sein statischer Zustand im Zeitschritt t sei ein dreidimensionaler Vektor xs\_t = (x y θ)T mit den x- und y-Werten der Position sowie der Orientierung θ. Darüber hinaus verfolgen wir seinen kinematischen Zustand xd\_t = (v -)T , der sich aus der Längsgeschwindigkeit v und der Winkelgeschwindigkeit ω zusammensetzt. Zunächst wollen wir die Unsicherheit in der Orientierung ignorieren und davon ausgehen, dass der Lenkwinkel mit 100-prozentiger Genauigkeit gesteuert werden kann. Die mittels der Bewegung mit der Geschwindigkeit v von der Position (xt yt) berechnete neue Position (xt+1 yt+1) ist aufgrund der unsicheren Geschwindigkeit auch unsicher. Die Positionsänderung unter der Annahme einer Gaußschen Unsicherheit der Geschwindigkeit kann also wie in der folgenden Abbildung dargestellt werden.



Ohne Berücksichtigung neuer Messungen wächst die Unsicherheit im Laufe der Zeit unbegrenzt an, wie in der folgenden Abbildung zu sehen ist.



Es ist deutlich zu erkennen, dass die Positionsunsicherheit bereits nach drei Zeitschritten breit verteilt ist. Was passiert, wenn wir die Unsicherheit der Orientierung (also der Richtung, in die wir uns bewegen) zusätzlich berücksichtigen? Die folgende Abbildung zeigt beispielhaft die Wahrscheinlichkeitsdichten für die Position nach drei Zeitschritten.



Obwohl die Unsicherheit der Geschwindigkeit und der Orientierung als gaußverteilt angenommen wird, sehen wir, dass die Positionsunsicherheit nicht einer Gaußverteilung, sondern einer bananenförmigen Verteilung ähnelt. Dies ist zu berücksichtigen, wenn für die Lokalisierung probabilistische Methoden wie das Kalman-Verfahren und seine Ableitungen verwendet werden, die von Gaußschen Verteilungen ausgehen. Interessant hierbei ist, dass die Positionsunsicherheit zwar im euklidischen Raum nicht gaußförmig ist, sehr wohl aber gaußförmig ist, wenn ihr die spezielle euklidische Gruppe SE(2), die Liesche Gruppe der Starrkörperbewegungen, zugrunde gelegt wird (Long et al., 2013). Die Lokalisierung auf dieser Gruppendarstellung ist daher seit langem ein Forschungsschwerpunkt. Wer sich hierfür interessiert, sei auf die Literatur von Giefer et al. (2020) und Zhang et al. (2016) verwiesen.

Zusammenfassung

Die Lokalisierung ist eine der wichtigsten Methoden für automatisierte mobile Systeme und bildet die Grundlage für andere Aufgaben wie die Objektverfolgung oder die Bewegungsplanung. Eine zuverlässige und genaue Schätzung des aktuellen Standorts in der Welt ist unerlässlich. Die Verfügbarkeit einer Karte vereinfacht den Prozess. Es können verschiedene Kartendarstellungen verwendet werden, wie z. B. metrische und topologische Karten.

Triangulation ist eine geeignete Methode zur Lokalisierung eines Objekts anhand von Orientierungspunkten. Bei dieser Methode werden die Winkel zwischen dem Kurs des Objekts und mehreren Orientierungspunkten gemessen. Für die Lokalisierung in zwei Dimensionen sind mindestens drei Orientierungspunkte mit genau bekannten Positionen erforderlich. Eine verwandte Technik ist die Trilateration, bei der nicht die Winkel, sondern die Abstände zu den Orientierungspunkten gemessen werden. Diese Methode bildet die Grundlage für die Positionsbestimmung mit Hilfe von GPS. Ein GPS-Empfänger benötigt Signale von vier verschiedenen Satelliten, um die Positionierung im dreidimensionalen Raum zu ermöglichen. Geometrisch wird die Position des Objekts durch den Schnittpunkt von vier imaginären Kugeln bestimmt, deren Mittelpunkte auf den Positionen der Satelliten liegen und deren Radien die jeweiligen Pseudo-Entfernungen sind.

In den meisten automatisierten mobilen Systemen werden probabilistische Lokalisierungsmethoden eingesetzt. Der Zustand eines Objekts wird unter Verwendung aller vergangenen Informationen kontinuierlich abgeschätzt. Dies beruht auf der Markov-Annahme, wonach der aktuelle Zustand die Informationen aller vorherigen Zustände enthält. Die Markov- und die Monte-Carlo-Lokalisierung sind zwei häufig verwendete probabilistische Lokalisierungsmethoden.

Die Unsicherheit der Bewegung führt zu unsicheren Positionsschätzungen bewegter Objekte. Die Position kann nicht exakt bestimmt werden, sondern liegt auf einer Wahrscheinlichkeitsverteilung, die einer Bananenform ähnelt. Im euklidischen Raum ist die Wahrscheinlichkeitsverteilung nicht gaußförmig, jedoch sehr wohl in der entsprechenden Lieschen Gruppe. Wenn probabilistische Lokalisierungsmethoden auf der Basis von Gauß-Modellen verwendet werden, sollte dies bedacht werden.



# Lektion 4

## Sensordatenfusion

#### LERNZIELE

Nach Abschluss dieser Lektion haben Sie gelernt, …

... welches die wichtigsten Sensoren sind, die für die Lokalisierung und die Wahrnehmung der Umgebung genutzt werden.

... wie Daten von Sensoren verarbeitet werden können und welche Ebenen der Datenfusion es gibt.

... was die Grundlagen der Bayesschen Statistik sind und wie die Kalman- und die erweiterten Kalman-Filter abgeleitet werden.

... wie Objektverfolgungsalgorithmen funktionieren und wie die Verfolgung eines einzelnen Objekts auf die Verfolgung mehrerer Objekte ausgeweitet werden kann.

DL-E-DLMDSEAAD02-U04

4.Sensordatenfusion

### Einleitung

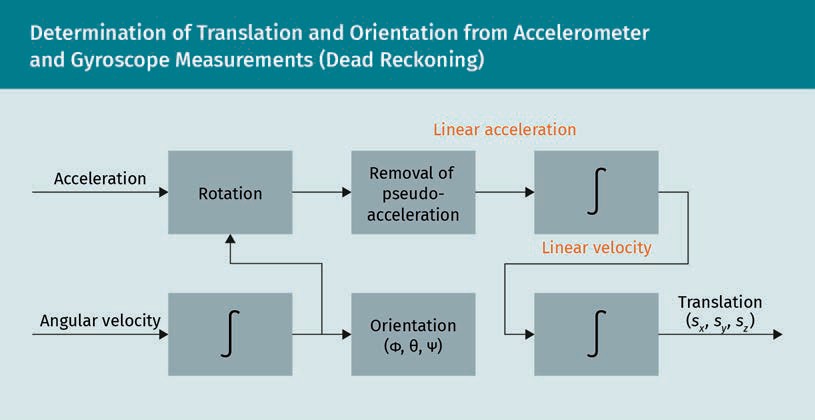
In diesem Abschnitt wird das Konzept der Sensordatenfusion vorgestellt, das sich in vielen realen Anwendungen wiederfindet. Zunächst werden dafür die wichtigsten Sensoren zusammengefasst, die in automatisierten Systemen zur Lokalisierung und Umgebungswahrnehmung eingesetzt werden. Anschließend wird gezeigt, wie Sensordaten für die weitere Verarbeitung aufbereitet werden können. In diesem Zusammenhang sollen Konzepte vorgestellt werden, mit denen Datenfusion auf verschiedenen Ebenen durchgeführt werden kann. Der Schwerpunkt dieser Lektion liegt auf der bayesschen Statistik und den Kalman-Filtern, welche die Grundlage für viele Fusionsalgorithmen bilden. Das erweiterte Kalman-Verfahren ist *de facto* der Stand der Technik für moderne Navigationssysteme und wird ausführlich behandelt. Zum Abschluss dieser Lektion werden Konzepte für die Objektverfolgung vorgestellt. Wir beginnen mit der Verfolgung eines einzelnen Objekts und untersuchen daraufhin das Szenario der Verfolgung mehrerer Objekte.

### Sensoren

Generell lassen sich Sensoren für automatisierte Systeme wie mobile Roboter oder automatisierte Fahrzeuge in extero- und propriozeptive Sensoren unterteilen. Exterozeptive Sensoren dienen zur Wahrnehmung der Umgebung, durch die das System manövriert, während propriozeptive Sensoren interne Fahrzeugzustandsparameter messen (Yurtsever et al., 2020).

Das globale Satellitennavigationssystem (GNSS) gilt als ein Sammelbegriff für verschiedene Satellitensysteme, die zur Lokalisierung und Positionsbestimmung auf der Erde und in der Luft eingesetzt werden. Die Grundfunktionalität beruht auf dem Empfang von Signalen von Navigationssatelliten und Pseudosatelliten, also terrestrischen Sendern, die das Verhalten eines Satelliten imitieren, um so die Lokalisierungsgenauigkeit zu erhöhen. Globale Satellitennavigationssysteme lassen sich in drei Segmente unterteilen, das Raum-, Kontroll- und Nutzersegment. Das Raumsegment besteht aus den Satelliten und ist für die Erzeugung der GNSS-Signale einschließlich der Informationen über das Datum, die Korrektur der Satellitenuhr, den Gesundheitszustand, die Ephemeriden und den Almanach verantwortlich. Das Kontrollsegment verfolgt die Satelliten und die Kommunikation zum Empfang und Hochladen von Informationen. Das Nutzersegment setzt sich aus allen GNSS-Empfängern zusammen. Diese bestimmen ihre aktuelle Position durch Abschätzen ihrer Entfernung zu den Satelliten. Da zur Berechnung der Größen externe Informationen genutzt werden, gehören GNSS-Empfänger zu den exterozeptiven Sensoren. Für detaillierte Informationen über die Funktionsweise von GNSS sei auf die Literatur von Rahemi et al. (2014) verwiesen.

Während GNSS zur Abschätzung der absoluten Position in Weltkoordinaten genutzt werden kann, lassen sich inertiale Messeinheiten (IMUs) zur Berechnung der relativen Position in Bezug auf einen bestimmten Startpunkt verwenden. Sie bestehen aus mehreren Beschleunigungsmessern und Drehratensensoren wie Gyroskopen und Magnetometern. Mit diesen können alle sechs kinematischen Freiheitsgrade abgedeckt werden. Die drei Translations- und drei Winkelgrößen lassen sich durch kontinuierliche Doppelintegration der Beschleunigungswerte und Einzelintegration der Winkelgeschwindigkeiten ermitteln. Die folgende Abbildung zeigt das Blockdiagramm für dieses Koppelnavigationssystem.



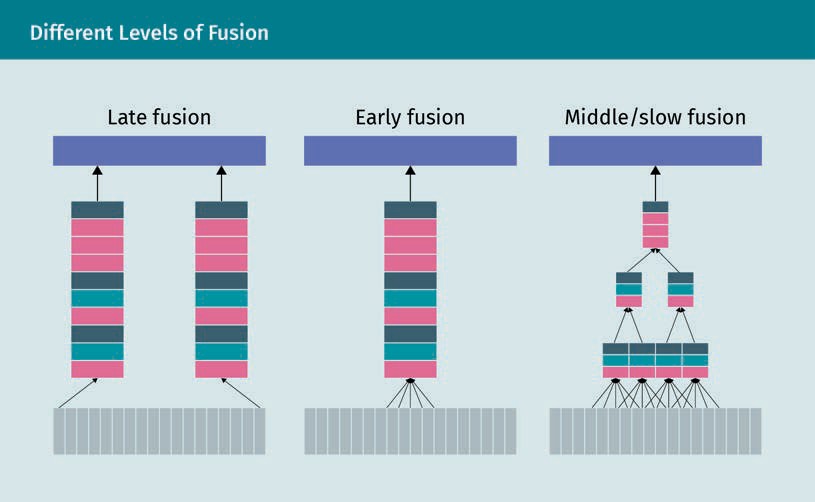
Im Vergleich zu den zur Lokalisierung eines Systems verwendeten GNSS und IMUs sind RADAR- (Radio Detection and Ranging) und LIDAR-Geräte (Light Detection and Ranging) zwei externe Sensoren, die zur Wahrnehmung des Überwachungsbereichs und zur Gewinnung von Umgebungsinformationen, einschließlich statischer und dynamischer Objekte, herangezogen werden können. RADAR-Geräte messen über Funk- und Mikrowellen Abstände und Geschwindigkeiten mit Hilfe der Dopplerfrequenz. LIDAR-Geräte hingegen nutzen Laserstrahlen für die Abstandsmessung, was zu genaueren Werten führt und aufgrund der geringeren Wellenlängen die Möglichkeit der Konturmessung eines Objekts bietet. Während das gepulste und das amplitudenmodulierte Dauerstrichradar (AMCW) für beide Sensorarten genutzt werden, können RADAR-Geräte auch mit frequenzmodulierten Dauerstrichwellen (FMCW) arbeiten.

Kameras liefern hochpräzise und zuverlässige Informationen über die Umgebung und eignen sich am besten für Objekterkennungsaufgaben, bei denen es auf die Farberkennung ankommt. RADAR-und LIDAR-Geräte können zum Beispiel nur dann etwas erkennen, wenn ein Tiefenunterschied in der Messrichtung vorhanden ist. Bei automatisierten Fahrzeugen könnte eine Fahrbahnbegrenzung aufgrund des Höhenunterschieds zwischen dem Bordstein und der Fahrbahn erkannt werden. Wenn jedoch nur eine Fahrbahnmarkierung vorhanden ist, können weder LIDAR- noch RADAR-Geräte diese erkennen. Kameras sind dagegen in der Lage, Farbunterschiede zu erkennen. Aufgrund der unterschiedlichen Vorteile der vielen vorhandenen Sensortypen wird in autonomen Fahrzeugen und mobilen Robotern meist eine Kombination aus mehreren Sensoren eingesetzt. Mit dieser Kombination können Unsicherheiten reduziert werden, um genauere Schätzungen der Eigen- und Zielpositionierung zu ermöglichen.

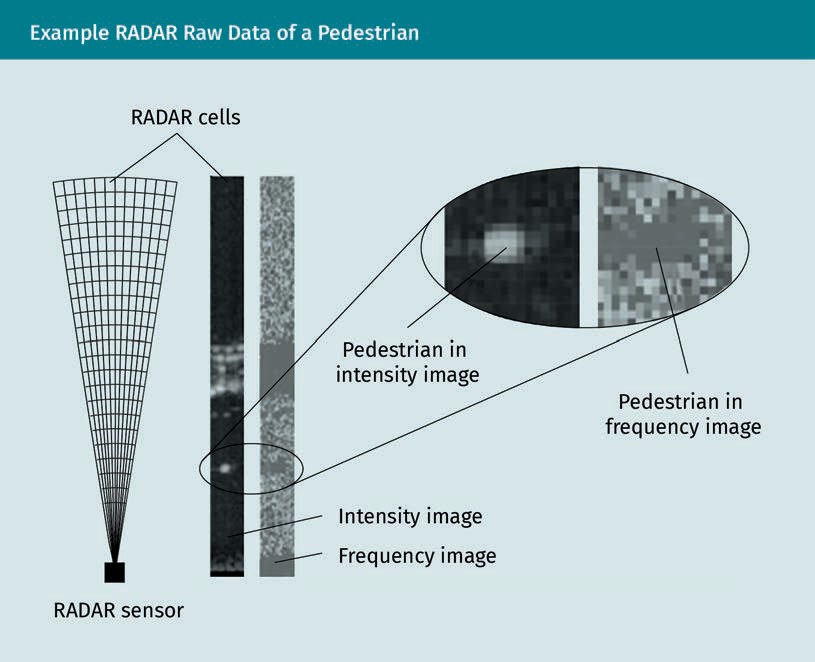
### Auswertung von Sensordaten

Im Allgemeinen kann die Sensordatenfusion mehrerer Sensorquellen auf verschiedenen Ebenen erfolgen, die sich grob in die frühe, mittlere (oder langsame) und späte Fusion unterteilen lässt (Peng et al., 2017). Die frühe Fusion erfolgt auf der Datenebene ohne Vorverarbeitung. Vor der Verarbeitung wird ein gemeinsamer Merkmalsraum erstellt. Mit dem Ansatz der späten Datenfusion werden mehrere generierte Hypothesen auf der Entscheidungsebene verschmolzen, was einen Auswertungsvorgang darstellt.

Zwischen diesen beiden Extremen kann Datenfusion auch auf jeder anderen Abstraktionsebene stattfinden, die oft als Merkmalsebenen bezeichnet werden. Die verschiedenen Stufen der Datenfusion sind in der folgenden Abbildung zu sehen.



In der Regel ist es notwendig, Sensordaten zu verarbeiten, um die darin enthaltenen Informationen zu gewinnen. Die folgende Abbildung ist ein Beispiel für Daten, die durch ein RADAR-System erhalten wurden.



Die linke Seite des Bildes zeigt, wie ein RADAR-Sensor den räumlichen Messbereich in Auflösungszellen unterteilt. Wir erhalten zwei verschiedene Bilder, das Intensitäts- und das Frequenzbild. Diese zeigen die reflektierte Intensität bzw. die Dopplerverschiebung im erfassten Signal für jede Zelle. Die Detailansicht zeigt den Teil, der den Fußgänger darstellt. Die Identifikation von Objekten in den Rohdaten ist keine triviale Aufgabe. Bei RADAR-Bildern werden häufig Objektlisten von Clustern erstellt, die Gebiete mit einer bestimmten Reflektion zeigen.

LIDAR-Scans messen die Ausdehnung r eines Hindernisses bezogen auf einen bestimmten Azimut α. Sie erzeugen Punktwolken, die zumindest die Position eines Punktes und die Intensität der Messung enthalten und zeigen, wie gut die das ausgesendete Signal reflektiert wurde. Die Position kann entweder in kartesischen Koordinaten oder in Polarkoordinaten dargestellt werden, wobei letztere das Messprinzip am besten wiedergeben. Somit ist eine Messung zpolar ein Vektor der Form zpolar = [α r]T. Durch folgende Umrechnung können daraus die kartesischen Koordinaten zCartesian

= [x y]T berechnet werden:

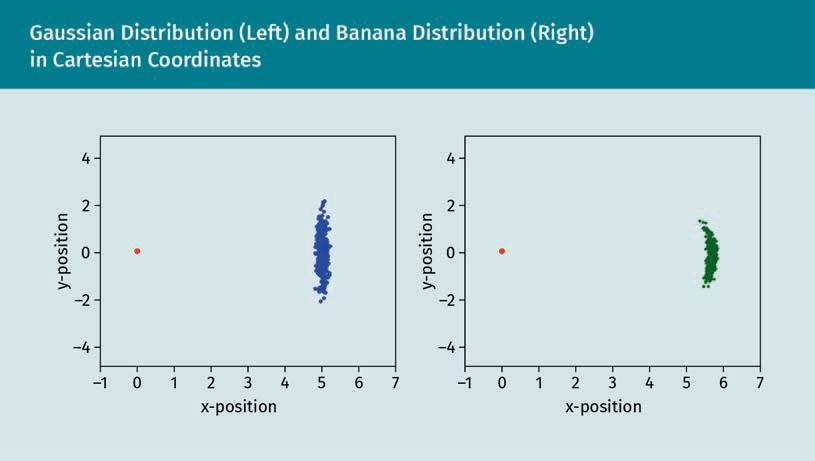
x = r · cos α

(4.1)

y = r · sin α

(4.2)

Auf den ersten Blick erscheinen beide Darstellungen gleichwertig, und diese Annahme trifft zu, wenn nur ein bestimmter Punkt betrachtet wird. Allerdings liegt jeder Messung eine Unsicherheit zugrunde. Wir können also nicht sagen, dass ein gemessener Punkt genau die richtige Position darstellt. Das Messprinzip eines LIDAR-Scanners (und auch für RADAR-Geräte) zeigt Unsicherheiten im Azimut und in der Reichweite, jedoch nicht in den x- und y-Werten. Wie sich dies auswirkt, ist in der folgenden Abbildung dargestellt.



Auf der linken Seite ist eine Wahrscheinlichkeitsverteilung der Position mit Unsicherheit in den kartesischen Koordinaten dargestellt. Rechts ist eine Wahrscheinlichkeitsverteilung mit Unsicherheit in Azimut und Entfernung zu sehen, die als Bananenverteilung bekannt ist (Long et al., 2013). Offensichtlich repräsentiert die Verteilung rechts die tatsächlichen Unsicherheitseigenschaften. Interessant ist, dass die Bananenverteilung auf der zugrundeliegenden Mannigfaltigkeit gaußförmig ist, jedoch nicht im euklidischen Raum – auch wenn es auf den ersten Blick nicht so aussieht. Diese Eigenschaft ist besonders interessant, wenn bestimmte Filter verwendet werden, die von Gaußschen Verteilungen ausgehen, wie z. B. der Kalman- oder der erweiterte Kalman-Filter. Da dieses komplexe Thema den Rahmen des vorliegenden Studienskripts sprengen würde, werden Interessierte auf die Literatur von Giefer et al. (2020) verwiesen.

### Kalman-Filter

Der Kalman-Filter ist ein Algorithmus, mit dessen Hilfe eine Reihe von verrauschten Beobachtungen (Messungen) zum Abschätzen unbekannter Variablen genutzt werden. Diese sind genauer als eine einzelne Messung. Das Verfahren basiert auf der Schätzung einer gemeinsamen Wahrscheinlichkeitsverteilung über alle Variablen für jeden Zeitrahmen. Der Kalman-Filter wird auch als lineare quadratische Schätzung (engl. LQE für *Linear Quadratic Estimation*) bezeichnet und ist vom Bayes-Filter abgeleitet.

Um diesen Algorithmus im Detail zu verstehen, müssen wir uns zunächst mit den theoretischen Grundlagen der bayesschen Statistik befassen. Der Satz von Bayes lautet:

P B A = P A B P B

P A

(4.3)

P(A) und P(B) sind die Wahrscheinlichkeiten der Messung A bzw. der Ausgabe B. Letztere wird als A-priori-Wahrscheinlichkeit bezeichnet, d. h. die Wahrscheinlichkeit, bevor die Messung A in Betracht gezogen wird. P(B|A) ist die A-posteriori-Wahrscheinlichkeit, d. h. die Wahrscheinlichkeit der Ausgabe B unter der Bedingung, dass die Messung A gemacht wurde. P(A|B) bezeichnet die Plausibilität (Likelihood), dass die Messung A durch B erzeugt wird. Die A-posteriori-Wahrscheinlichkeit und die Plausibilität sind bedingte Wahrscheinlichkeiten. Es gelten dann die folgenden Wahrscheinlichkeits-Rechenregeln:

P A, B = P A B P B = P B A P A

Gemeinsame Wahrscheinlichkeitsverteilung

Die gemeinsame Wahrscheinlichkeitsverteilung für mehrere Zufallsvariationen gibt die Wahrscheinlichkeit an, dass jede der Zufallsvariablen in eine bestimmte diskrete Menge von Werten fällt.

(4.4)

wobei P(A,B) die Wahrscheinlichkeit für A und B bezeichnet, sowie:

P B = ∑P A, B = ∑P B A P A

A A

(4.5)

Diese beiden Regeln sind als Produkt- bzw. Summenregel bekannt. Die erste Regel kann folgendermaßen verstanden werden: Die Wahrscheinlichkeit der Ereignisse A und B ist gleich der Wahrscheinlichkeit von A unter der Bedingung B, multipliziert mit der Wahrscheinlichkeit von B. Sie ist gleich wie die Wahrscheinlichkeit von B unter der Bedingung A, multipliziert mit der Wahrscheinlichkeit von A. Die Gleichung (4.5) bedeutet, dass die Summe aller möglichen Ergebnisse von A und dem Ereignis B gleich der Wahrscheinlichkeit von B ist. Mit Hilfe der Produktregel kann diese Gleichung so umgeformt werden, dass die Wahrscheinlichkeit von B der Summe aller möglichen Ergebnisse von A unter der Bedingung des Ergebnisses B , multipliziert mit der Wahrscheinlichkeit von A entspricht. Wenn zwei zufällige Ereignisse A und B unabhängig sind, dann gilt:

P A, B = P A P B

(4.6)

Dies bedeutet

P A B = P A

(4.7)

P B A = P B

(4.8)

Außerdem gilt für zwei voneinander unabhängige Zufallsereignisse A und B bedingt:

P A, B C = P A C P B C

(4.9)

Mit Hilfe dieser Regeln kann der Satz von Bayes wie folgt geschrieben werden:

P B A = P A B P B

∑AP B A P A

(4.10)

Der Bayes-Filter nutzt diesen Satz zur rekursiven Schätzung unbekannter Wahrscheinlichkeitsdichtefunktionen (WDF) über die Zeit unter Annahme bestimmter zugrunde liegender Prozessmodelle und unter Einbeziehung von Messungen. Dieser Vorgang wird auch als rekursiver Bayes-Schätzer bezeichnet und besteht aus Vorhersage und Korrektur. Bei diesem Modell wird davon ausgegangen, dass der tatsächliche Zustand x eine unbeobachtete Markow-Kette ist, d. h., dass der Zustand xk im Zeitschritt k nur vom vorherigen Zustand xk – 1 im Zeitschritt k – 1 abhängt. Das bedeutet auch, dass xk – 1 die Information aller vorherigen Zustände trägt und somit die folgende Gleichung gilt:

p )k )k − 1, )k − 2, …, )0 = p )k )k − 1

(4.11)

Im Vergleich dazu wird angenommen, dass die Messungen z ein verdecktes Markow-Modell (engl.: *Hidden Markov model, HMM*) darstellen, dessen Verhalten von der Markow-Kette x abhängt. Dies wird wie folgt dargestellt:

p zk )k, )k − 1, …, )0 = p zk )k

(4.12)

Die Wahrscheinlichkeitsverteilung aller Zustände des verdeckten Markow-Modells kann nun wie folgt ausgedrückt werden

k

p )0, …, )k, z1, …, zk = p )0 ∏ p zi )i p )i )i − 1

i = 1

(4.13)

Der Kalman-Filter lässt sich aus dem Bayes-Filter ableiten, wenn der Übergang vom Zustand xk – 1 nach xk als linear angenommen wird, und die Variablen als normalverteilt gelten. Ziel ist es, den Zustand xk anhand der Messungen bis zum aktuellen Zeitschritt zu schätzen. Die sich daraus ergebenden Gleichungen für die Vorhersage- und Korrekturschritte können probabilistisch wie folgt geschrieben werden:

Die Wahrscheinlichkeitsverteilung des vorhergesagten Zustands xk ist das Integral über das Produkt aus der Wahrscheinlichkeitsverteilung des Übergangs von xk – 1 nach xk und der Wahrscheinlichkeitsverteilung der A-priori-Wahrscheinlichkeit über alle möglichen vorherigen Zustände xk – 1.

Vorhersage:

p )k z1:k − 1 = ∫p )k )k − 1 p )k − 1 z1:k − 1 d)k − 1

(4.14)

Die Wahrscheinlichkeitsverteilung der Aktualisierung des vorhergesagten Zustands xk unter Berücksichtigung der Messungen bis zum aktuellen Zeitschritt ist das Produkt aus der Plausibilität (Likelihood) der Messung zk unter Berücksichtigung des vorhergesagten Zustands xk und der Wahrscheinlichkeitsverteilung des vorhergesagten Zustands, geteilt durch die Wahrscheinlichkeitsverteilung der Messung zk unter Berücksichtigung der vorherigen Messungen.

Aktualisierung:

p )k

z1:k

= p zk )k p )k z1: k − 1

p zk z1:k − 1

(4.15)

Der Nenner wird in der Regel ignoriert, da er in Bezug auf den Zustand x konstant ist. Daher kann die aktualisierte Gleichung wie folgt geschrieben werden:

p )k z1:k ∝ p zk )k p )k z1:k − 1

(4.16)

Wie können wir diese Gleichungen nun nutzen, um einen Zustand auf der Grundlage von Beobachtungen zu schätzen? Es wird angenommen, dass der tatsächliche Zustand xk aus xk–1 folgt, und zwar gemäß:

)k = Fk)k − 1 + Bkuk + wk

(4.17)

Fk steht für das Zustandsübergangsmodell, und uk bzw. Bk bezeichnen den Steuerungsvektor bzw. das Steuerungseingangsmodell. Das Prozessrauschen wird durch wk dargestellt, eine multivariate Gauß-Verteilung mit Nullmittelwert und Kovarianz Qk ist, sodass wk ~ N(0, Qk) gilt. Eine Messung zk des tatsächlichen Zustands xk wird modelliert durch:

zk = Hk)k + vk

(4.18)

Hierbei sind Hk das Mess- (oder Beobachtungs-)Modell und vk ~ N(0, Rk) das Beobachtungsrauschen mit Null-Mittelwert und der Messkovarianz Rk. Der Kalman-Filteralgorithmus propagiert nun das erste und zweite statistische Moment, den Mittelwert und die Kovarianz des Zustands. Die Vorhersage des Zustands xk|k – 1 und seiner Kovarianz Pk|k – 1 kann wie folgt ausgedrückt werden:

)k k − 1 = Fk)k − 1 k − 1 + Bkuk

(4.19)

bzw.

Pk k − 1 = F P

FT + Q

k k − 1 k − 1 k k

(4.20)

Der Korrektur- (oder Aktualisierungs-)Schritt besteht aus den folgenden Gleichungen Durch die Berechnung der Differenz zwischen Messung und vorhergesagter Messung kann das Innovationsresiduum berechnet werden:

yk = zk − Hk)k k − 1

(4.21)

Die Innovationskovarianz

Sk = H P

HT + R

k k k − 1 k k

(4.22)

Kalman-Verstärkung

Die Kalman-Verstär­kung bestimmt, wie stark sich die Differenz zwischen geschätzter und aktueller Messung auf die Zustandskorrektur auswirkt.

wird zur Bestimmung der optimalen Kalman-Verstärkung verwendet:

Kk = P

HTS−1

k k − 1 k k

(4.23)

Letztendlich sind die A-posteriori-Zustandsschätzung und -Kovarianz festgelegt durch:

)k k = )k k − 1 + Kkyk

(4.24)

bzw.

Pk k = I − KkHk Pk k − 1

(4.25)

I ist hierbei die Identitätsmatrix.

Anhand eines Beispiels soll gezeigt werden, wie die Vorhersage- und Aktualisierungsschritte funktionieren. Angenommen, wir wollen die Ego-Position und die Geschwindigkeit eines selbstfahrenden Fahrzeugs in einer Dimension schätzen, dann ist der Zustand xk im Zeitschritt k ein Zufallsvektor xk = [x vx]T. Nehmen wir an, wir könnten die Geschwindigkeit mit dem Tachometer messen, so erhalten wir Messungen in der Form zk = [vx]. Wir nehmen an, dass unsere Bewegung einem Modell mit konstanter Geschwindigkeit folgt:

)k = xk − 1 + / t · vx

(4.26)

vx,k = vx, k − 1

(4.27)

Dies kann durch folgende Zustandsübergangsmatrix dargestellt werden:

Fk =

1 / t

0 1

(4.28)

Die Kovarianzmatrix des Prozessrauschens Qk sollte mit Bedacht gewählt werden, um das gewünschte Verhalten zu erreichen. Diese teilt dem Filter mit, wie weit er von einem Schritt zum nächsten vom Übergangsmodell abweichen kann. Äußere Faktoren, wie z. B. Wind, können das Fahrzeug von der vorausberechneten Bewegung abbringen. Wenn wir mit konstanter Geschwindigkeit fahren, werden Geschwindigkeitsschwankungen ebenfalls durch die Varianz des Prozessrauschens erfasst. Nehmen wir (ohne Beleg für die Gültigkeit dieser Annahme) eine Prozessrauschkovarianz für Modelle mit konstanter Geschwindigkeit an (Bar-Shalom et al., 2004):

/ t3 3

/ t2 2

/ t2 2

/ t

Q =

(4.29)

Unsere Zustandsschätzung wird durch den Steuereingangsvektor uk dargestellt, der durch das Steuereingangsmodell Bk auf den Zustandsraum abgebildet wird. Für diese Schätzung können wir beliebige Steuereingaben wie Beschleunigung, Bremsen und Lenkung heranziehen. Für unser Beispiel werden wir keinen aktiven Eingang berücksichtigen, also den Steuereingangsvektor als Nullvektor annehmen.

Das Messmodell Hk ist eine Abbildung des Zustandsraums auf den Messraum. Das heißt, es wird ermittelt, welche Messung sich aus der aktuellen Zustandsschätzung ergeben würde. In unserem Fall messen wir die Geschwindigkeit direkt, ohne irgendeinen Faktor, und erhalten daher:

Hk = 0 1

(4.30)

Daraus ergibt sich die folgende Abbildung:

x

vx = 0 1 v

x

(4.31)

Schließlich muss der Messung eine Unsicherheit zugeordnet werden. Das Messrauschen Rk gibt an, wie sehr die Filter den einzelnen Parametern der Messung vertrauen, so dass kleine Werte eine hohe und große Werte eine niedrige Genauigkeit anzeigen. Wir setzen Rk auf:

Rk = 0,3

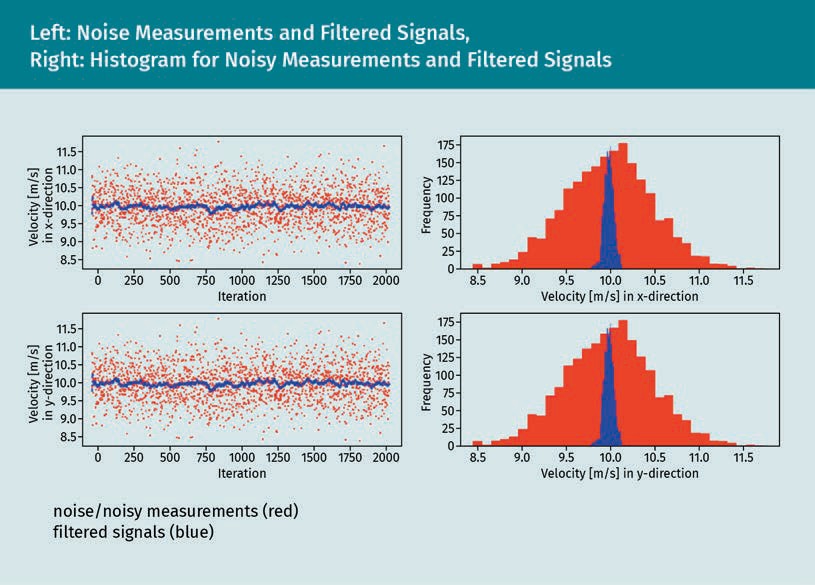
(4.32)

Nachdem wir nun alle Bestandteile des Kalman-Filters bestimmt haben, können wir den Algorithmus leicht implementieren und sehen, wie er funktioniert. Der folgende Code zeigt eine beispielhafte Python-Implementierung.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Da sowohl die Kalman-Verstärkung K als auch das Messmodell H Vektoren sind, muss das Kreuzprodukt K⊗ H() verwendet werden. Es wird durch die NumPy-Funktion „outer“ wiedergegeben. Wenn die Vorhersage- und Korrekturschritte in einer Schleife 2000 Mal mit konstanten Geschwindigkeitsmessungen von 10 m/s in jeder Richtung durchgeführt werden, ergeben sich die folgenden geschätzten Geschwindigkeiten des Objekts (blau) im Vergleich zu den Messungen (rot). In diesem Fall wurden die Messungen von einem Gaußschen Rauschen mit einer Standardabweichung von 0,5 m/s überlagert.



Die Histogramme auf der rechten Seite der Abbildung zeigen außerdem die Häufigkeitsverteilung der Geschwindigkeiten. Die Abbildung zeigt, dass die Gaußverteilungen der Geschwindigkeitsfehler durch den Filtervorgang schmaler werden. Dies führt zu einer genaueren Schätzung der tatsächlichen Geschwindigkeit des Objekts.

### Erweiterter Kalman-Filter

In der Realität sind die meisten Systeme nichtlinear, so dass der Kalman-Filter als optimaler linearer Schätzer für lineare Systeme bei diesen nicht gut funktioniert. Es gibt verschiedene Ansätze, um die Beschränkung der Linearität der Übergangs- und Messsysteme aufzuheben. Davon sind der erweiterte Kalman-Filter (EKF) und der Unscented Kalman-Filter (UKF) die beliebtesten. In diesem Abschnitt werden wir die theoretischen Aspekte des EKF im Detail untersuchen und die Verbesserungen gegenüber der Standardversion des Kalman-Filters aufzeigen.

Für die Nutzung des erweiterten Kalman-Filters brauchen die Übergangs- und Messmodelle nicht linear sein, sondern können jetzt differenzierbare Funktionen sein. Es wird angenommen, dass der wahre Zustand xk und die Messung zk diesen Modellen folgen:

)k = f )k − 1, uk + wk

(4.33)

zk = h )k + vk

(4.34)

Die Übergangs- und Messfunktionen f und h können für die Berechnung des vorhergesagten Zustands und der Messung verwendet werden, aber wie kann die Kovarianz ermittelt werden? Hier kommt zum Tragen, dass die entsprechenden Funktionen differenzierbar sein müssen. Wir müssen die Jacobi-Matrix der beiden Funktionen, also die partiellen Ableitungen nach jeder Variablen, berechnen und mit dem aktuellen vorhergesagten Zustand bewerten. Bei dieser Methode werden die nichtlinearen Übergangs- und Messfunktionen um die aktuelle Zustandsschätzung herum linearisiert. Die Gleichungen für die Vorhersage- und Aktualisierungsschritte entsprechen im Wesentlichen dem Standard-Kalman-Filter und können wie folgt formuliert werden:

Vorhersage:

)k k − 1 = f )k − 1 k − 1, uk

(4.35)

Pk k − 1 = F P

k

FT + Q

(4.36)

k − 1 k − 1

k

k

Hierbei gilt:

∂f

Fk =

∂) )k − 1 k − 1, uk

(4.37)

Aktualisierung:

yk = zk − h )k k − 1

(4.38)

k k − 1

k

k

Sk = H P

k

HT + R

Kk = P

HTS−1

(4.39)

(4.40)

k k − 1

k

k

)k k = )k k − 1 + Kkyk

(4.41)

Pk k = I − KkHk Pk k − 1

(4.42)

Hierbei gilt:

∂h

Hk =

∂) )k k − 1

(4.43)

Der erweiterte Kalman-Filter ist derzeit der De-facto-Standard für Navigationssysteme und GPS (Abdollahpouri et al., 2017). Er hat jedoch auch einige Nachteile, die zu Problemen führen können. Aufgrund der Linearisierung kann der Filter divergieren, wenn der Prozess falsch modelliert wird oder wenn die anfängliche Schätzung weit vom tatsächlichen Zustand entfernt ist. Außerdem wird die geschätzte Kovarianzmatrix im Vergleich zur wahren Kovarianz tendenziell unterschätzt. Aus diesem Grund wird häufig ein stabilisierendes Rauschen verwendet, um eine Konsistenz im statistischen Sinne zu erreichen (Huang et al., 2008).

Betrachten wir ein Beispiel für ein nichtlineares Übergangsmodell, das die Vorteile des EKF im Vergleich zum Standard-Kalman-Filter zeigt. Angenommen, wir bewegen uns auf einem Einheitskreis (einem Kreis mit dem Radius 1) und wollen unsere Position, den Winkel und die Winkelgeschwindigkeit schätzen. Unser Zustandsvektor kann als xk = [x y θ ω]T ausgedrückt werden. Nehmen wir an, dass wir unseren aktuellen Winkel in Bezug auf den Mittelpunkt des Kreises messen können, so haben unsere Messungen die Form zk = [θ]. Unsere Zustandsübergangsfunktion, mit der wir unseren vorhergesagten Zustand erhalten, wird dann wie folgt dargestellt:

f = cos θ + / t · ω sin θ + / t · ω θ + / t · ω ω T

k

(4.44)

Diese Übergangsfunktion besteht aus differenzierbaren trigonometrischen Funktionen. Für die EKF müssen wir nun die Jacobi-Matrix von f bestimmen. Dazu müssen wir alle partiellen Ableitungen in Bezug auf jede Zustandsvariable berechnen.

∂f1 = 0, ∂f1 = 0, ∂f1 = − sin θ + / t · ω , ∂f1 = − / t · sin θ + / t · ω

∂x ∂y ∂θ ∂ω

∂f2 = 0, ∂f2 = 0, ∂f2 = cos θ + / t · ω , ∂f2 = / t · cos θ + / t · ω

(4.45)

∂x ∂y ∂θ ∂ω

∂f3 = 0, ∂f3 = 0, ∂f3 = 1, ∂f3 = / t

(4.46)

∂x ∂y ∂θ ∂ω

∂f4 = 0, ∂f4 = 0, ∂f4 = 0, ∂f4 = 1

(4.47)

∂x ∂y ∂θ ∂ω

(4.48)

Die Jacobi-Matrix wird also wie folgt dargestellt:

∂f

∂) =

0 0 −sin θ + / t · ω −/ t · sin θ + / t · ω 0 0 cos θ + / t · ω / t · cos θ + / t · ω 0 0 1 / t

0 0 0 1

(4.49)

Die Messfunktion h bildet unseren Zustandsraum xk = [x y θ ω]T auf den Messraum zk = [θ] ab, was wie folgt ausgedrückt werden kann:

hk = θ

(4.50)

Die Jakobi-Matrix besteht aus allen partiellen Ableitungen in Bezug auf die Zustandsvariablen:

∂h = 0, ∂h = 0, ∂h = 1, ∂h = 0

∂x ∂y ∂θ ∂ω

(4.51)

Dies führt zu:

∂h = 0 0 1 0

∂)

(4.52)

Nach der Ableitung der erforderlichen Filterbestandteile können wir nun den EKF für dieses Beispiel wie folgt implementieren:

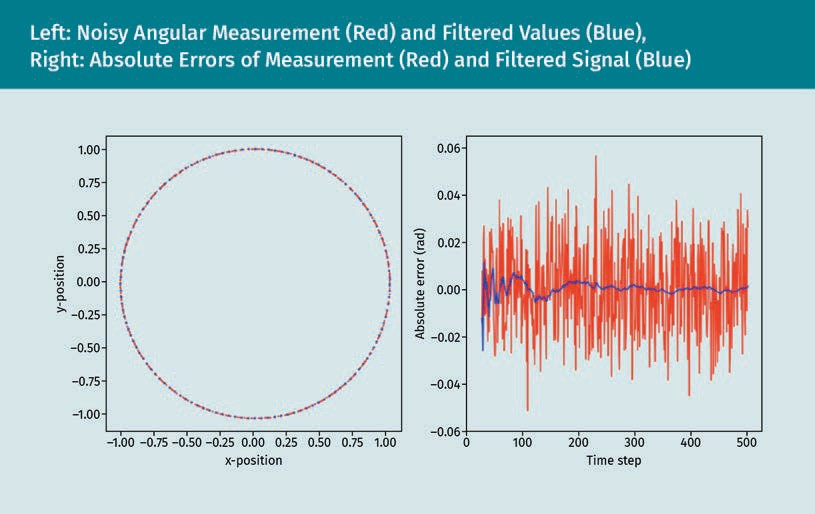
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

In diesem Fall gehen wir von einem sehr geringen Prozessrauschen aus, weil wir uns des Prozessmodells ziemlich sicher sind. Wir setzen die Standardabweichung des Messrauschens auf 0,05 rad und simulieren verrauschte Winkelmessungen, die eine Kreisbewegung für 500 Zeitschritte mit ∆t = 0,1 s darstellen. Die folgende Abbildung zeigt auf der linken Seite die Messungen (rot) und den geschätzten Zustand (blau), sowie auf der rechten Seite die absoluten Fehler der verrauschten Winkelmessungen (rot) und die geschätzten Winkel (blau) in Relation zu den tatsächlichen Winkeln.

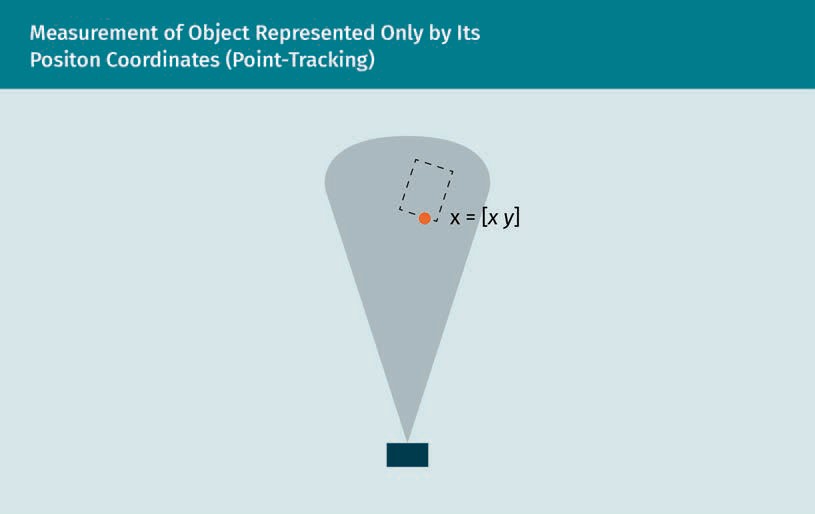


Es ist schön zu sehen, dass unser EKF sehr gut funktioniert und den Fehler in hohem Maße verringert. Dieses nichtlineare Übergangsmodell kann mit einem Standard-Kalman-Filter nicht erfasst werden. Es gibt aber andere Ansätze, wie z. B. den UKF, der auf einer deterministischen Methode namens *Unscented Transformation* („duftlose“ Transformation), basiert. Beim UKF wird ein minimaler Satz von Sigma-Punkten um den Mittelwert herum als Stichprobe genommen und zur Berechnung der neuen Mittelwerte und Kovarianzen propagiert. Diese Ableitung kann mit höheren nichtlinearen Modellen besser umgehen als der EKF (Julier & Uhlmann, 1997).

Oft hängt es von der Situation ab, welcher der Filter besser abschneidet. Daher sollten die Leistungen bei einem bestimmten Problem verglichen werden, um so den am besten geeigneten Schätzer zu ermitteln.

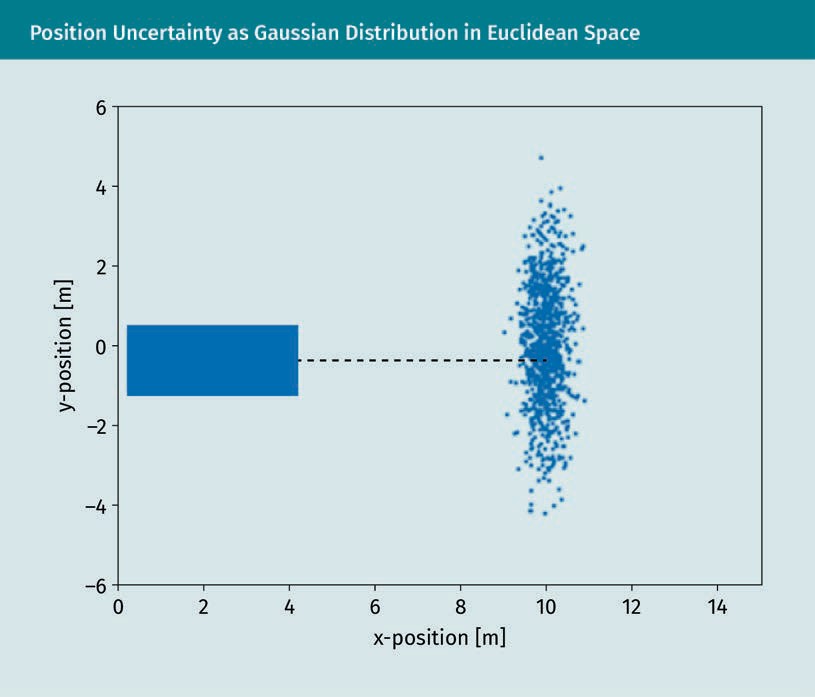
### Objektverfolgung

Die Objektverfolgung (engl.: *object tracking*) ist ein wesentlicher Bestandteil jedes autonomen Systems, das in dynamischen Umgebungen manövriert. Die Umgebungswahrnehmung und die Erstellung einer zuverlässigen und genauen Darstellung der Umgebung sind die Grundlage für jedes sichere Manövrieren. Während durch die Objekterkennung alle dynamischen Objekte innerhalb des Überwachungsbereichs des Systems erkannt werden, dient die Objektverfolgung der Abschätzung der Objektzustände im Laufe der Zeit durch Verringerung der Messunsicherheiten. Im Allgemeinen unterscheiden sich die für die Verfolgung genutzten Ansätze nicht von denen für die Lokalisierung. Daher werden auch hierfür am häufigsten Ableitungen des Bayes-Filters verwendet. Grundsätzlich können zwei verschiedene Verfolgungsszenarien unterschieden werden: der statische und der dynamische Ego-Zustand. Während im statischen Zustand das Ego-System zur Überwachung an einer festen Position verharrt, bewegt es sich im dynamischen Zustand selbst. Das letztere Szenario ist aufgrund der Unsicherheit des Ego-Status, zusätzlich zu den Unsicherheiten der zu verfolgenden Objekte, komplizierter. Wenn wir wissen, dass sich genau ein Objekt in unserem Überwachungsbereich befindet und dass jede erhaltene Messung zu diesem Objekt gehört, ist die Verfolgung trivial. In diesem Fall können wir direkt einen probabilistischen Filter wie den erweiterten Kalman-Filter einsetzen und den Zustand des Objekts schätzen. Dieses als Einzelobjektverfolgung bekannte Verfahren ist in der folgenden Abbildung dargestellt.



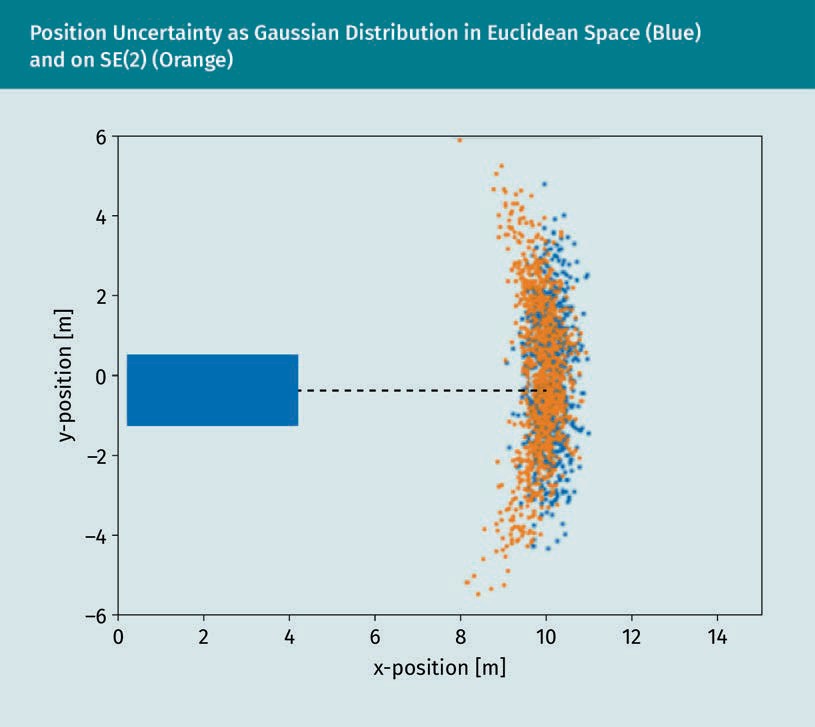
Traditionell werden Objekte als Punkte verfolgt. Ihre Zustände stellen Punkte im euklidischen Raum dar, z. B. im zweidimensionalen Raum (2, wenn wir in zwei Dimensionen verfolgen wollen, oder im dreidimensionalen Raum (3, wenn auch die Höhe von Interesse ist. Die Vernachlässigung der räumlichen Objektausdehnung kann jedoch zu Problemen führen. Stellen wir uns vor, unser autonomes Fahrzeug (blaues Rechteck) verfolgt ein Auto (gestricheltes Rechteck) nur als Punkt (roter Punkt) und berücksichtigt seine Länge nicht. Im schlimmsten Fall berechnet es eine falsche Bahnkurve und verursacht einen Zusammenstoß.

Gibt es dafür eine Lösung? LIDAR-Geräte zum Beispiel haben eine sehr hohe räumliche Auflösung und erzeugen meist eine Menge von Punkten pro Objekt. Mit dieser Punktmenge lassen sich nicht nur die Positionen, sondern auch die Größen von Objekten verfolgen. So kann der statische Teil des Zustands eines einzelnen Objekts durch den Zustandsraum x = [x y θ a b]T dargestellt werden, wobei θ, a und b für die Orientierung, Länge und Breite stehen. Dieses Verfahren wird als erweiterte Objektverfolgung bezeichnet und erfasst alle für den sicheren Betrieb erforderlichen Zustandsparameter. Der klassische Ansatz der erweiterten Objektverfolgung stellt den Zustand und seine Unsicherheiten im euklidischen Raum dar; diese Darstellung entspricht jedoch nicht der Realität.



Nehmen wir an, dass das blaue Rechteck ein Objekt ist, das wir verfolgen wollen. Die gestrichelte Linie stellt die Bewegung zur nächsten Position dar, und die blauen Punkte sind Stichproben aus der Wahrscheinlichkeitsverteilung der Position im euklidischen Raum. Offensichtlich erfasst diese Verteilung die tatsächlichen Unsicherheiten nicht, da eine Änderung der Objektausrichtung nicht dazu führen würde.

Sie liegt vielmehr auf einem Abschnitt eines imaginären Kreises, wie in der folgenden Abbildung (orange) dargestellt.



Die Unsicherheit weist eine Bananenverteilung im euklidischen Raum auf (Long et al., 2013). Wie können wir damit umgehen, wenn wir für unsere Verfolgungsfilter Gaußsche Verteilungen annehmen? Das Interessante an dieser Art von Verteilung ist, dass sie zwar im euklidischen Raum nicht gaußförmig ist, sehr wohl aber dann, wenn sie in einem geeigneten Zustandsraum betrachtet wird, der Bewegungsgruppe bzw. speziellen euklidischen Dimensionsgruppe SE(n). Sie gehört zur Matrix der Lie-Gruppen und besteht aus Starrkörperbewegungen im euklidischen Raum mit intrinsischen Zusammenhängen zwischen Position und Orientierung. Die Lie-Theorie ist ein sehr komplexes Thema und wird in diesem Kursbuch nicht weiter behandelt. Für Interessierte sei daher auf die Literatur von Isaev und Rubakov (2018) sowie Hilgert und Neeb (2012) verwiesen.

Kurz gesagt, in der Literatur wird der Zusammenhang zwischen Lie-Gruppen und Lie-Algebren genauer untersucht. Die spezielle euklidische Gruppe in zwei Dimensionen SE(2) besteht zum Beispiel aus allen Matrizen der Form:

cos θ −sin θ tx G = sin θ cos θ ty

0 0 1

(4.53)

Afﬁne   
Transformationen

Afﬁne Transformationen sind unter anderem Rotationen, Translationen, Skalierungen und Scherabbildungen im n-dimensionalen Raum.

Hierbei ist θ die Orientierung und tx bzw. ty sind die Translationen in x- bzw. y-Richtung. Das Interessante an Gruppen ist, dass eine Gruppenoperation, in diesem Fall das Matrixprodukt zweier Gruppenelemente, wiederum ein Element der Gruppe erzeugt. Es handelt sich also um eine geeignete Darstellung der Bewegung in zwei Dimensionen. Neben der speziellen euklidischen Gruppe spielt bei Tracking-Aufgaben auch die spezielle orthogonale Gruppe SO(n) in n-Dimensionen eine wichtige Rolle. Sie besteht aus allen Drehungen im n-dimensionalen euklidischen Raum. Dies kann sich als sinnvoll erweisen, wenn bekannt ist, dass sich ein Objekt auf einer Kreisbahn bewegt, ohne den Radius zu verändern. Kürzlich haben Giefer et al. (2020) eine andere Art von Lie-Gruppe vorgeschlagen: Die afﬁne Gruppe Aff(n), zu der alle afﬁnen Transformationen im n-dimensionalen Raum gehören. Somit kann sie für eine geeignete Darstellung des Zustandsraums bei der erweiterten Objektverfolgung herangezogen werden. Die bisherigen Ergebnisse zeigen, dass mit dieser Gruppe ein höheres Maß an Stabilität und Zuverlässigkeit bei geringeren Schätzfehlern erreicht wird.

Die Annahme, dass jede gewonnene Messung zu einem existierenden Objekt gehört, entspricht natürlich nicht der Realität. Sensoren können Störungsmessungen erzeugen, die nichts Reales abbilden. Der Begriff wird im Allgemeinen für unerwünschte Sensorreflexionen verwendet und bezeichnet meist das Phänomen der unerwünschten Echos in RADAR-Systemen. Bei solchen Störungen können wir nicht garantieren, dass eine Messung wirklich von einem vorhandenen Objekt erzeugt wird. Ein unsachgemäßer Umgang mit Störungen kann zu Geisterobjekten führen. Außerdem gibt es in den meisten Szenarien nicht nur ein Objekt. An diesem Punkt wird das Tracking, das dann Multi-Objekt-Tracking genannt wird, aufgrund von Datenzuordnungsproblemen besonders wichtig. Wir wissen nicht, welches Objekt welche Messung erzeugt hat. Wir können nur über den zeitlichen Verlauf abschätzen, welches Objekt mit hoher Wahrscheinlichkeit das richtige ist.

Wenn die Objekte weit voneinander entfernt sind, ist die Aufgabe natürlich einfacher als bei geringeren Abständen. Wenn in diesen Fällen für jedes neu auftretende Objekt ein eigener Kalman-Filter (oder ein Abkömmling davon) gebildet und jedes Objekt separat verfolgt wird, ist dies meist ein geeigneter Ansatz. Dies kann jedoch schnell an Grenzen stoßen, wenn die Objekte näherkommen und die Datenzuordnung wichtiger wird. Ein fortschrittlicherer Ansatz ist die Verwendung von zufälligen, endlichen Mengen *(random ﬁnite sets, RFS)*. Dabei handelt es sich um ungeordnete Mengen mit einer unbekannten Anzahl einzelner Objektzustände. Anstatt mehrere Einzelobjektzustände zu erfassen, wird also der komplette Zustand mehrerer Objekte in einer zufälligen endlichen Menge erfasst. Nehmen wir an, unser Einzelobjektzustand zum Zeitpunkt k besteht nur aus den x- und y-Positionen in der euklidischen Ebene, also xk = [x y]T. Eine zufällige endliche Menge dieses Einzelobjektzustands kann wie folgt dargestellt werden:

+k = )k, 1, )k, 2, …, )k, N k

(4.54)

Hierbei ist N(k) die Anzahl der Objekte im Zeitschritt k. Dies ist ein Element der Potenzmenge des Zustandsraums ℘(X) des Einzelobjekts – sie besteht also aus allen möglichen Teilmengen. Ein sehr einfaches Beispiel, um das Konzept der Potenzmengen zu verstehen, ist das folgende: Wenn der Zustandsraum unseres einzelnen Objekts die natürlichen Zahlen von 1 bis 3 enthält (tN1:3 ), dann ist die Potenzmenge davon:

℘ tN1:3 = ∅, 1 , 2 , 3 , 1,2 , 1,3 , 2,3 , 1,2, 3

(4.55)

∅ ist die leere Menge, die in jeder Menge enthalten ist. Der Zustand mehrerer Objekte kann theoretisch mit Hilfe des Multi-Objekt-Bayes-Filters verfolgt werden, doch ist dies rechnerisch schwer zu bewältigen. Stattdessen gibt es verschiedene Näherungsverfahren. Der Wahrscheinlichkeitshypothesen-Dichtefilter (PHD-Filter, engl.: *probability hypothesis density ﬁlter*) beispielsweise propagiert nur das erste statistische Moment des Multi-Objekt-Zustands, das als Intensität der Wahrscheinlichkeitshypothesendichte bekannt ist (Vo & Ma, 2006). Das Integral der Wahrscheinlichkeitshypothesendichte über ein bestimmtes Gebiet ergibt die geschätzte Anzahl der darin befindlichen Objekte. Ein anderer Ansatz ist die Annäherung an die A-posteriori-Wahrscheinlichkeit mit Hilfe von Multi-Bernoulli-Komponenten, was zum Multi-Target-Multi-Bernoulli-Filter (MeMBer) führt (Vo et al., 2009). Eine Multi-Bernoulli-Komponente kann wie folgt dargestellt werden:

N

πk =

ri Existenz

, pi

Wahrscheinlichkeit

Wahrscheinlichkeit

Dichte

i = 1

(4.56)

Es gibt verschiedene Variationen dieser beiden Filter, und es wurden viele weitere Näherungslösungen an den Mehr-Objekt-Bayes-Filter vorgeschlagen, was den Rahmen dieses Lehrbuchs jedoch sprengen würde. Alle diese Filter gehen von bestimmten Geburts- und Sterbemodellen aus. In diesem Sinne sind Geburten neue entstehende Objekte, und Sterbefälle sind Objekte, die aus dem Überwachungsbereich verschwinden. Mit Hilfe von Geburtenwahrscheinlichkeitsfunktionen kann die Wahrscheinlichkeit des Auftretens neuer Objekte in bestimmten Teilen der Umgebung angenommen werden. Messungen in einem Gebiet mit sehr hoher Geburtenwahrscheinlichkeit haben eine geringere Unsicherheit, dass es sich um echte Messungen handelt, als solche, die in Gebieten mit geringer Geburtenwahrscheinlichkeit durchgeführt werden. So kann dies einen Hinweis darauf geben, ob es sich bei einer Messung eher um Störgeräusche oder um ein tatsächlich vorhandenes Objekt handelt. Das Gegenteil kann durch Sterbemodelle erreicht werden. Gebiete mit hoher Sterbewahrscheinlichkeit bedeuten, dass Objekte, die plötzlich keine Messwerte mehr erzeugen, möglicherweise wirklich verschwunden sind und nicht nur vorübergehend verdeckt werden.

Zusammenfassung

Sensoren für automatisierte Systeme können in proprio- und exterozeptive Sensoren eingeteilt werden. Während erstere Informationen aus dem System selbst messen, werden letztere zur Erfassung von Informationen aus der Umgebung eingesetzt. Die wichtigsten propriozeptiven Sensoren sind Trägheitsmessgeräte (inertiale Messeinheiten), die Informationen über die Kinematik des Systems liefern, wie z. B. Beschleunigung und Winkelgeschwindigkeit, sowie Radsensoren (Raddrehgeber), welche die Geschwindigkeit der Räder durch Zählen der Umdrehungen messen.

GNSS, LIDAR- und RADAR-Geräte sowie Kameras sind die wichtigsten exterozeptiven Sensoren, die zur absoluten Lokalisierung und Umgebungswahrnehmung genutzt werden. Um die Sensordaten zu verarbeiten, müssen sie zunächst aufbereitet werden, d. h. die Rohdaten müssen in ein verarbeitbares Format umgewandelt werden.

Die rekursive Bayes-Schätzung, auch Bayes-Filter genannt, stützt sich auf die bayesschen Statistik. Dabei handelt es sich um einen probabilistischen Ansatz zur rekursiven Schätzung unbekannter Wahrscheinlichkeitsdichtefunktionen im Zeitverlauf unter Verwendung von Messungen und eines bestimmten Prozessmodells. Es besteht aus kontinuierlichen Vorhersage- und Korrekturschritten (Aktualisierung), um die Schätzung eines Zustands zu verbessern. Wenn sowohl die Mess- als auch die Übergangsmodelle als linear angenommen werden und die Zufallsvariablen gaußverteilt sind, wird der Bayes-Filter zum Kalman-Filter, der ein optimaler linearer Filter ist. In der Realität ist die Annahme von Linearität fast nie gegeben, so dass es verschiedene Ansätze zur Behandlung der Nichtlinearität gibt. Der erweiterte Kalman-Filter linearisiert die Daten über eine Schätzung des aktuellen Mittelwerts und der Kovarianz und ist *de facto* der Stand der Technik bei Navigationssystemen.

Die Verfolgung eines einzelnen Objekts unter der Annahme, dass keine Störungsmessungen vorliegen, ist eine triviale Aufgabe. Wenn mehrere Objekte verfolgt werden müssen, wird die Datenzuordnung zu einem Problem. Der Multi-Objekt-Bayes-Filter ist die Erweiterung des Bayes-Filters auf den Multi-Objekt-Zustands- und Messraum. Er beruht auf zufälligen endlichen Mengen (*random finite sets, RFS*). Da dieses Verfahren rechnerisch schwer zu bewältigen ist, wurden verschiedene Näherungsverfahren entwickelt, wie z. B. der Wahrscheinlichkeitshypothesen-Dichtefilter (PHD). Bei diesem besonderen Näherungsverfahren wird nur das erste statistische Moment weitergegeben. Darüber hinaus gibt es den Multi-Target-Multi-Bernoulli-Filter (MeMBer-Filter), der die A-posteriori-Verteilungen durch Multi-Bernoulli-Komponenten approximiert.



# Lektion 5

## Bewegungsplanung

#### LERNZIELE

Nach Abschluss dieser Lektion haben Sie gelernt, …

... wie Bewegungsplanung und Pfadplanung unterschieden werden können.

... wie die Pipeline der Bewegungsplanung funktioniert.

... welche gebräuchlichen Ansätze für die Pfadplanung es gibt.

... warum die Bewegungsvorhersage wichtig ist und welche Annahmen zur Vereinfachung getroffen werden können.

... wie die Bahnkurvenermittlung funktioniert und welche gängigen Ansätze es dafür gibt.

DL-E-DLMDSEAAD02-U05

1. Bewegungsplanung

### Einleitung

Die Planung von Bewegungen von einer Ausgangs- zu einer Zielposition in einer Umgebung mit statischen Hindernissen ist eine der größten Herausforderungen für autonome Fahrzeuge und Roboter. Die Planung kollisionsfreier Bahnkurven und das möglichst schnelle Erreichen des Ziels sind die wichtigsten Aspekte dieser Aufgabe. In diesem Abschnitt werden die drei Hauptbestandteile der Bewegungsplanung vorgestellt: Pfadplanung, Bewegungsvorhersage und Bahnkurvenermittlung. Unser Ziel ist es, einen Überblick über die bestehenden Ansätze zu geben und die zur Entwicklung eigener Planungsalgorithmen notwendigen Grundlagen zu vermitteln.

### Pfadplanung

Obwohl die Begriffe „Bewegungsplanung“ und „Pfadplanung“ häufig synonym verwendet werden, beschreiben sie in der Regel unterschiedliche Probleme. Das Problem der Pfadplanung besteht darin, den kürzesten Pfad von einer Ausgangsposition zur Zielposition zu finden. Andererseits beschreibt die Bewegungsplanung, wie das System von der Ausgangsposition zur Zielposition kommt. Die Aufgabe besteht also darin, eine Folge von Systemkonfigurationen zu finden, um von einem Startpunkt zu einem Endpunkt zu gelangen. Eine Konfiguration q ist eine räumliche Lage eines Systems, und die Menge aller möglichen Konfigurationen wird als der Konfigurationsraum Q des Systems bezeichnet. Zum besseren Verständnis dieses Unterschieds soll das folgende Szenario untersucht werden.

Ein Bild, das Text, Berg, Natur enthält.

Automatisch generierte Beschreibung

Die Pfadplanung eines Systems über ein Hochseil ist recht einfach, weil es sich im Grunde um eine direkte Verbindung der Ausgangs- und Zielposition handelt. Andererseits kann die Bewegungsplanung für ein solches Szenario äußerst komplex sein. Ein Labyrinth ist ein gutes Beispiel, bei dem die Pfadplanung komplex, aber die Bewegungsplanung einfacher ist. Die Berechnung eines Weges (Pfads) von der Start- zur Endposition kann rechenintensiv sein, während die Planung der Bewegung für einen berechneten Pfad recht einfach ist, sofern genügend Platz zum Rangieren vorhanden ist. Natürlich gibt es auch Fälle, in denen sich beide Aufgaben als sehr schwierig erweisen, z. B. bei einem größeren System, das sich auf einem schmalen Pfad zwischen vielen statischen Hindernissen bewegt. Zusammenfassend lässt sich sagen, dass die Bewegungsplanung beschreibt, wie eine Reihe von Bewegungen von einer Ausgangs- zu einer Zielpose geplant werden kann, ohne mit der Umgebung zu kollidieren. Dabei sind folgende Elemente vorgegeben:

* die Anfangskonﬁguration (Pose oder räumliche Lage) qinitial des Systems,
* die Zielkonﬁguration (Pose oder räumliche Lage) qgoal des Systems,
* eine geometrische Beschreibung des Systems A, und
* eine geometrische Beschreibung der Umgebung W, auch Welt oder Arbeitsraum genannt.

Bei der Pfadplanung hingegen wird nur versucht, einen kollisionsfreien Pfad von der Ausgangs- zur Zielpose zu finden. Mit dieser Definition kann die Pfadplanung als eine Teilaufgabe der Bewegungsplanung betrachtet werden. Es hat sich gezeigt, dass der Konfigurationsraum Q eines Systems bei der Betrachtung abstrakter Planungsprobleme nützlich ist (Siciliano & Khatib, 2016). Dies ergibt sich aus der Tatsache, dass ein bewegliches System mit komplexer Form und Struktur einen einzigen Punkt in Q darstellt, während die Dimension von Q durch die Anzahl der Freiheitsgrade des Systems bestimmt wird. Ein einzelnes bewegtes Partikel, also ein durch einen einzelnen Punkt in der euklidischen Ebene (2 dargestelltes System, hat einen Konfigurationsraum, der durch die Menge der zweidimensionalen Vektoren (tx, ty) definiert ist, wobei tx und ty Translationen in x- bzw. y-Richtung sind. Für ein autonomes, in (2 navigierendes Fahrzeug ist der Konfigurationsraum die zweidimensionale spezielle euklidische Gruppe SE(2). Als dreidimensionale Mannigfaltigkeit umfasst sie die euklidischen Vektoren der Form (tx, ty, θ), wobei θ die Orientierung des Systems ist (Solà et al., 2018).

Wir definieren die Hindernisregion des Arbeitsraums W als die geschlossene Menge Wobs ⸦ W und damit als den freien Raum von W als Wfree = W . Die Menge der Punkte, die vom System A in der Konfiguration q . Q besetzt sind, ist auch eine Teilmenge A(q) W. Die Hindernisregion Qobs des System-Konfigurationsraums kann nun wie folgt definiert werden: Qobs = {q 2 Q|A(q) ∩ Wobs ≠ ∅}. Qobs besteht also aus allen Konfigurationen von A, die zu einer Überlappung der von A und dem Arbeitsraum-Hindernisbereich Wobs besetzten Punkte führen. Daraus folgt, dass der freie Raum Qfree des Konfigurationsraums wie folgt definiert ist: Qfree = Q .

⸦

Mannigfaltigkeit

Eine Mannigfaltigkeit ist ein topologischer Raum, der lokal dem euklidischen Raum ähnelt, sich aber global unterscheiden kann.

Wir können nun die früheren Definitionen von Pfadplanung und Bewegungsplanung auf mathematisch präzise Weise beschreiben. Die Aufgabe der Pfadplanung ist die Berechnung einer Bahnkurve (eines Pfades) im Arbeitsraum W,σ : [0,1] → Wfree, mit 3(0) = qinitial und σ(1) = qgoal. Andererseits bedeutet Bewegungsplanung

die Aufgabe, eine (kontinuierliche) Bahnkurve (einen Pfad) im Konfigurationsraum τ : [0,1]

→ Qfree mit τ(0) = qinitial und τ(1) = qgoal zu berechnen. Der Grund dafür, dass diese beiden Begriffe häufig synonym verwendet werden, liegt auf der Hand: Bei beiden handelt es sich um eine Art der Pfadplanung, allerdings in unterschiedlichen Räumen.

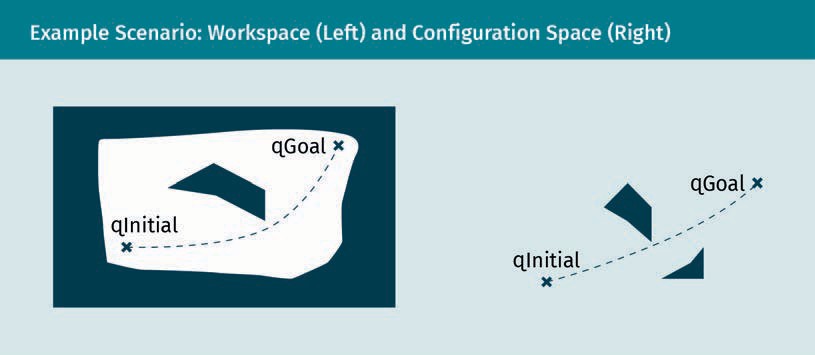
Die folgende Abbildung veranschaulicht die beiden unterschiedlichen Aufgaben.

Ein Bild, das Text, ClipArt, Schild enthält.

Automatisch generierte Beschreibung

Der linke Teil der Abbildung zeigt die Pfadplanung, dargestellt durch die grüne Linie, die von der Ausgangsposition zur endgültigen räumlichen Lage führt. Auf der rechten Seite ist das Ergebnis der Bewegungsplanung im Arbeitsraum zu sehen, das durch die verschiedenen Posen (Konfigurationen) des Klaviers dargestellt wird. Dieses Beispiel zeigt sehr gut, warum das Problem der Bewegungsplanung auch als das Problem des Klavier-Möbelpackers (engl.: *piano mover‘s problem*) bekannt ist. Da das bewegte Objekt in diesem Beispiel sechs Freiheitsgrade hat, wird die Menge der Konfigurationen durch einen (kontinuierlichen) Pfad im sechsdimensionalen Konfigurationsraum dargestellt.

Für die meisten Planungsalgorithmen muss der Konfigurationsraum diskretisiert werden. Die beiden gebräuchlichsten Ansätze für die Diskretisierung sind die kombinatorische Planung und die stichprobenbasierte Planung. Bei kombinatorischen Planungsmethoden werden mit Hilfe von Graphen die Konnektivität von Qfree erfasst und Lösungen gefunden. Das Ergebnis der kombinatorischen Planung ist eine Straßenkarte (*„road map“*), d. h. ein Graph in Qfree , dessen Eckpunkte und Kanten Konfigurationen in Qfree bzw. kollisionsfreie Pfade darstellen. Betrachten wir einige Vertreter dieser Methode anhand des folgenden Beispielszenarios etwas genauer.

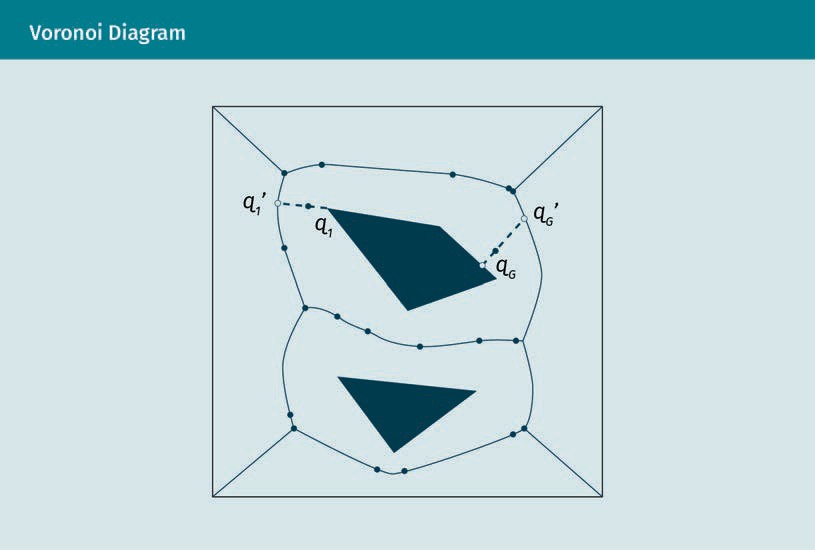


Unser System A, zum besseren Verständnis durch ein einzelnes Partikel dargestellt, bewegt sich im Arbeitsraum W = (2 von der Konfiguration qinitial nach qgoal, wie auf der linken Seite der Abbildung dargestellt. Auf der rechten Seite ist der resultierende Konfigurationsraum Q = (2 abstrakt dargestellt. Der Sichtbarkeitsgraph ist einer der frühesten kombinatorischen Planungsalgorithmen. Mit dieser Methode wird versucht, einen Pfad als polygonale Linie von qinitial nach qgoal durch die Eckpunkte von Qobs zu planen (Burgard et al., 2011). Das Ergebnis dieser Methode ist in der folgenden Abbildung dargestellt.

Ein Bild, das Text, Visitenkarte enthält.

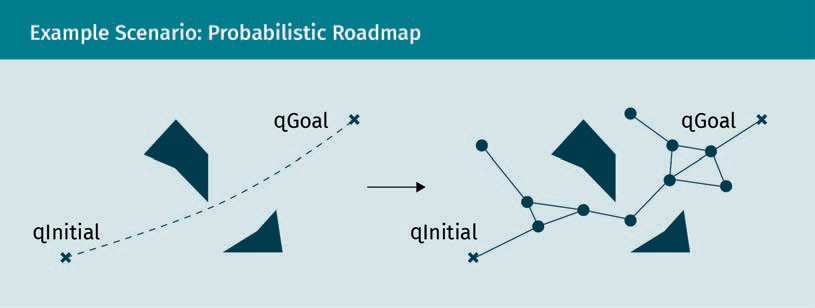
Automatisch generierte Beschreibung

Im ersten Schritt wird der Sichtbarkeitsgraph erstellt, dann wird der kürzeste Pfad mit einem anderen Algorithmus, z. B. dem Dijkstra-Algorithmus ermittelt (Sniedovich, 2006). Eine weitere kombinatorische Planungsmethode ist das (verallgemeinerte) Voronoi-Diagramm (Bhattacharya & Gavrilova, 2008). Voronoi-Diagramme bestehen aus den Punkten q E Q mit dem gleichen maximalen Abstand zu allen nächstgelegenen Hindernissen (Burgard et al., 2011). Dieser Ansatz eignet sich besonders für bewegte Systeme mit hoher Unsicherheit, weil der Abstand dabei maximiert wird. Doch genau dieser Vorteil ist gleichzeitig ein Nachteil, weil dabei suboptimale Wege gefördert werden. Ein beispielhaftes Voronoi-Diagramm ist in der folgenden Abbildung dargestellt. Hierbei stehen qI und qG für qinitial bzw. qgoal. qI’ und qG’ stellen die nächstgelegenen Punkte des Voronoi-Diagramms von qinitial und qgoal dar.



Die Zellzerlegung ist ein weiterer Ansatz der kombinatorischen Planung. Die Idee besteht darin, Qfree in eine Menge von n Bereichen zu zerlegen, die als Zellen bezeichnet werden. Dies führt zu einem Verbindungsgraphen, der zum Finden eines Pfades genutzt wird (Kloetzer et al., 2015; Gonzalez et al., 2017). Der Verbindungsgraph Gcon ist ein geordnetes Paar (Vcon, Econ). Die einzelnen Zellen sind die Eckpunkte Vcon , und die benachbarten Zellen sind die Kanten Econ. Wenn zwei Eckpunkte nicht durch eine Kante verbunden sind, sind sie nicht benachbart. Die Zellzerlegung kann exakt oder näherungsweise erfolgen. Bei einer exakten Zellzerlegung wird der gesamte Raum erfasst, während bei einem näherungsweisen Ansatz nur ein Teil davon erfasst wird. Gängige Zerlegungsmethoden sind die sichtbarkeitsbasierte Zellzerlegung, die trapezförmige Zerlegung und die Gitterzerlegung (Latombe, 2012). Nach der Zerlegung des Raums in Zellen muss der Verbindungsgraph bestimmt werden, wobei der Mittelpunkt jeder Zelle als Eckpunkt betrachtet wird und benachbarte Zellen durch Segmente verbunden werden.

Der Vorteil der kombinatorischen Planung ist, dass eine Lösung gefunden wird, wenn sie existiert, aber sobald der Konfigurationsraum hochdimensional wird, wird diese Methode unlösbar (Burgard et al., 2011). Bei der Stichproben-basierten Planung wird ein anderer Ansatz zur Diskretisierung des Konfigurationsraum genutzt: Mit Hilfe von Stichproben q und einem Kollisionserkennungsalgorithmus wird geprüft, ob q E Q ist. Zwei häufig verwendete Methoden sind die probabilistischen Straßenkarten *(road maps)* und schnell erkundende Zufallsbäume (*rapidly exploring random trees*, Kavraki et al., 1996; LaValle, 1998). Die Methode probabilistischer Straßenkarten besteht darin, Muster in Qfree als Eckpunkte zu deklarieren und dann nahegelegene Eckpunkte mithilfe eines lokalen Planers zu verbinden, der die Verbindung auf Kollisionen überprüft (Burgard et al., 2011). Mögliche Nachbarn können z. B. mit Hilfe eines bestimmten Radius oder einer Suche nach dem k-nächsten Nachbarn ermittelt werden. Dies ist jedoch keine triviale Aufgabe, weil der Konfigurationsraum in der Regel eine hochdimensionale Mannigfaltigkeit darstellt. Daher ist es wichtig, eine geeignete Metrik festzulegen. Es werden so lange Stichproben genommen, bis die Straßenkarte eine bestimmte Dichte erreicht. In der folgenden Abbildung ist eine beispielhafte probabilistische Straßenkarte dargestellt.



Probabilistische Straßenkarten lassen sich leicht auf hochdimensionale und komplexe Konﬁgurationsräume anpassen. Außerdem sind sie probabilistisch vollständig. Ein großer Nachteil ist, dass dieser Ansatz nicht für alle Arten von Umgebungen geeignet ist, z. B. für enge Passagen. Dies rührt daher, dass die Wahrscheinlichkeit, eine Probe aus einem engen Bereich zu ziehen, viel geringer ist als aus einem großen freien Raum.

Die Idee der schnell erkundenden Zufallsbäume besteht darin, den Konfigurationsraum zu erkunden, indem die anfängliche Konfiguration qinitial auf einen Baum erweitert wird, dessen Wurzel bei qinitial liegt. Für Interessierte sei daher auf die Literatur von LaValle (1998) verwiesen, die eine detaillierte Beschreibung dieses Algorithmus wiedergibt. Eine einmal erstellte Straßenkarte kann mit verschiedenen Paaren von qinitial und qgoal erneut verwendet werden. Zum Finden eines Verbindungspfads müssen die Knoten oder Zellen, die am nächsten bei qinitial und qgoal liegen, gemäß einer geeigneten Metrik für Q bestimmt werden.

Ein weiterer Ansatz zur Wegfindung ist das Potenzialfeld, das ein künstliches Potenzialfeld U im Konfigurationsraum des bewegten Systems simuliert. Die maximalen und minimalen Potentiale werden der Ausgangskonfiguration qinitial bzw. der Zielkonfiguration qgoal zugeordnet. Ein Wechsel von einer Konfiguration q1 zu einer Konfiguration q2 findet statt, wenn U(q1) ≤ U(q2) ist. Bei dieser Methode werden zwei verschiedene Arten von Potenzialen verwendet: das abstoßende Potenzial Urep, das das System von Hindernissen wegdrängt, und das anziehende Potenzial Uatt, das das System in Abhängigkeit vom Abstand zum Endpunkt in dessen Richtung zieht. Die Potenzialfunktion U: C → ( wertet die Systemkonfigurationen als Summe von abstoßenden und anziehenden Potenzialen aus und wird wie folgt deﬁniert:

Probabilistisch vollständig

Ein Planungsalgorithmus ist probabilistisch vollständig, wenn die Wahrscheinlichkeit, eine Lösung zu finden, mit zunehmendem Zeitaufwand gegen eins strebt.

m

U q = Uatt q + ∑Urepi q

i = 1

(5.1)

Oft ist es schwierig, eine geeignete Bewertungsfunktion für die meist hochdimensionalen Konfigurationsräume zu finden. Darüber hinaus können lokale Minima bei der Verwendung der Potenzialfeldmethode zu einem Problem werden. Mehrere Studien haben jedoch gezeigt, wie diese vermieden oder umgangen werden können (Park & Lee, 2003; Guerra et al., 2016).

### Bewegungsvorhersage

Bisher sind wir davon ausgegangen, dass die Welt, durch die sich unser System bewegt, statisch ist. Das bedeutet, dass wir die Pose jedes vorhandenen Hindernisses im Voraus kennen und dies in die Planung der Bewegung des Systems einbeziehen können. So kann ein einmal geplanter Pfad für zukünftige Situationen ohne Anpassung wiederverwendet werden (mit Ausnahme der Änderung der Ausgangs- und Zielkonfiguration des Systems). Es liegt auf der Hand, dass diese Annahme in der Realität nicht zutrifft. Dynamische Objekte sind in allen Arten von Umgebungen außerhalb des Labors allgegenwärtig und müssen bei jedem Planungsalgorithmus berücksichtigt werden.

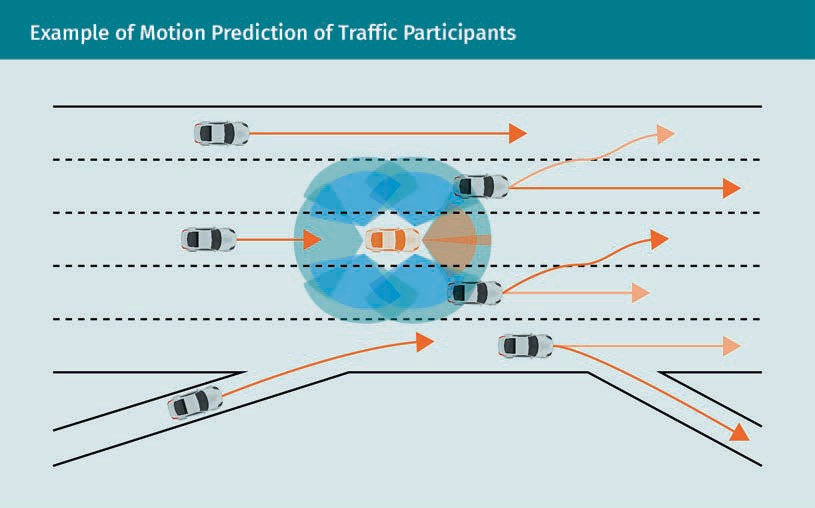
Überwachungsbereich

Der Überwachungsbereich eines Systems ist der Bereich, der von seinen Sensoren erfasst wird und somit überwacht werden kann.

Die Bewegungsvorhersage ist ein wesentlicher Bestandteil der Bewegungsplanung. Ziel ist es, die zukünftigen Zustände aller dynamischen Objekte im Überwachungsbereich des beweglichen Systems abzuschätzen. Der Zustand eines Objekts besteht zumindest aus seiner Position in der Welt, seinem Kurs und der Geschwindigkeit, kann aber auch weitere Parameter wie die Beschleunigung und die Winkelgeschwindigkeit der räumlichen Ausdehnung umfassen. Eine exakte Bewegungsvorhersage ist für eine sichere und kollisionsfreie Bewegungsplanung unerlässlich, weil sie die Berechnung zukünftiger Aktionen auf der Grundlage der berechneten Bewegungen der aktuellen dynamischen Objekte ermöglicht. Die Grundlage für eine zuverlässige und genaue Bewegungsvorhersage ist die Erkennung dieser Objekte mit Hilfe der verfügbaren Sensoren, die an den mobilen Systemen angebracht sind, wie Kameras, LIDAR- und RADAR-Geräte. Diese Erkennungen sind nur Rohmessungen und mit Unsicherheiten behaftet. Die Aufgabe der Objektverfolgung besteht darin, über bestimmte mathematische Bewegungsmodelle aus diesen Messungen die tatsächlichen Objektzustände im Laufe der Zeit zu schätzen. Je nach Art des dynamischen Objekts können die verwendeten Bewegungsmodelle unterschiedlich sein. So kann beispielsweise eine gehende Person andere Bewegungen ausführen als ein Fahrzeug oder ein Fahrrad. Die Kenntnis der Zustände der aktuellen dynamischen Objekte dient als Grundlage für die Vorhersage ihrer Bewegung. Die Kenntnis der vollständigen Geschichte der Objekte kann zusätzliche hilfreiche Informationen für eine zuverlässige Vorhersage liefern. Es liegt auf der Hand, dass Bewegungsvorhersagen abhängig von der Art des Objekts gemacht werden sollten. Die Fahrzeuge folgen in der Regel der befahrenen Fahrspur. Daher ist die Wahrscheinlichkeit, dass sie ihre aktuelle Bewegung fortsetzen, viel größer als die, dass sie diese Spur verlassen. Ein zuverlässiges und genaues Straßenmodell ist für die Bewegungsvorhersage von Fahrzeugen sehr wichtig. Dies kann kartenbasiert (im Voraus bekannt) oder sensorbasiert (während des Betriebs des beweglichen Systems) erfolgen. Die Einbeziehung von Blink- und Bremslichtern in die Vorhersagealgorithmen kann zusätzliche Informationen für zukünftige Bewegungen liefern (Deo & Trivedi, 2018b). Fußgänger:innen sind viel flexibler in ihrer Bewegung und können schnell die Richtung wechseln.

Die Bewegungsvorhersage ist sehr komplex – jedoch können zur Vereinfachung dieser Aufgabe bestimmte Annahmen getroffen werden. Diese Annahmen hängen von der Klasse des dynamischen Objekts ab. Im Allgemeinen können drei verschiedene Arten unterschieden werden: physikalische, manöverbasierte und interaktionsbasierte Annahmen. Physikalische Annahmen beruhen auf bestimmten physikalischen Randbedingungen für die Bewegungen der Objekte. Fahrzeuge können zum Beispiel ihre Orientierung nicht ohne Translation ändern. Bei Fußgängern gibt es diese Einschränkung nicht: ihr Mobilitätsgrad ist höher, so dass weniger physikalische Annahmen getroffen werden können. Trotz der Möglichkeit schneller Geschwindigkeits- und Orientierungsänderungen von laufenden Personen ist jedoch der Bereich der in kurzer Zeit erreichbaren Positionen sehr klein. Manöverbasierte Annahmen beruhen auf bestimmten Manövern (Aktionen), die von einer bestimmten Art von dynamischen Objekten ausgeführt werden. Die Bewegungen eines Fahrzeugs folgen im Allgemeinen einer endlichen Menge von Manövern – zum Beispiel bleiben die Fahrzeuge auf ihrer Fahrspur, bremsen unter bestimmten Bedingungen (z. B. bei Verkehrsampeln), und wechseln ihre Fahrspur, wenn sie dies durch ihre Blinker angezeigt wird.

In der Regel wird davon ausgegangen, dass Fußgänger:innen beim Gehen auf dem Gehweg bleiben und ihre Richtung entsprechend ihrer Kopfbewegungen ändern. In mehreren Studien wurden Kopfposition und Blickerfassung von Fußgängern und Fußgängerinnen zur Bewegungsvorhersage untersucht (Baxter et al., 2014; Karasev et al., 2016). Es kann davon ausgegangen werden, dass Straßen nur an Fußgängerüberwegen überquert werden. Allerdings kann die Bewegung von zu Fuß gehenden im Allgemeinen nicht mit hoher Wahrscheinlichkeit vorhergesagt werden. Daher ist es notwendig, mehrere Bewegungshypothesen im Auge zu behalten. Interaktionsbasierte Annahmen berücksichtigen jedes dynamische Objekt einzeln, dazu aber auch ihre Interaktionen. Ein Beispiel wäre der Fahrspurwechsel eines Fahrzeugs. Ein auf der angepeilten Fahrspur fahrendes zweites Fahrzeug wird höchstwahrscheinlich langsamer, um dem ersten Fahrzeug die Möglichkeit des Spurwechsels zu geben. Fußgänger:innen, insbesondere Kinder, betreten unaufmerksam gelegentlich die Straße, worauf die Fahrzeuge reagieren müssen. Die folgende Abbildung zeigt ein Beispiel für die Bewegungsvorhersage anderer Fahrzeuge um ein Ego-Fahrzeug.



Betrachten wir nun als einfachste Bewegungsvorhersage die nur auf dem aktuellen Zustand des Objekts basierende. Angenommen, Sie verfolgen ein Objekt mit dem Zustand x0 = [x, y, vx, vy] zum Zeitpunkt 0.

Ausgehend von einem Modell mit konstanter Geschwindigkeit ist der neue Zustand nach der Zeit t:

)k = x + t · vx, y + t · vy, vx, vy

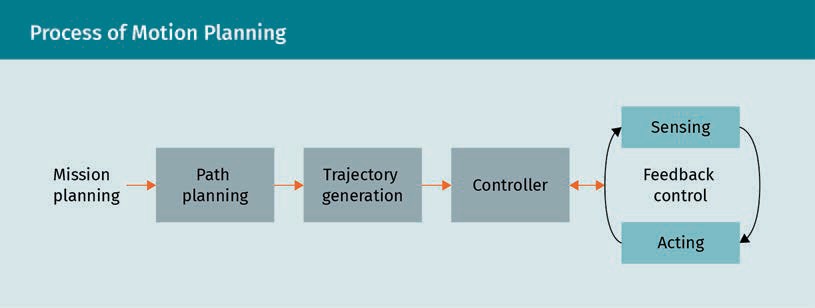
(5.2)

Natürlich reicht es in der Realität nicht aus, nur die Position und die Geschwindigkeit des Objekts zu schätzen, aber dieses einfache Beispiel verdeutlicht, wie eine grundlegende Bewegungsvorhersage durchgeführt werden kann. Ein größerer Vorhersagehorizont führt zu einer größeren Unsicherheit in der Vorhersage und damit zu einer wachsenden Anzahl von Hypothesen über zukünftige Zustände.

Die Bewegungsvorhersage dynamischer Objekte ausschließlich auf der Grundlage physikalischer Annahmen wäre naiv. Viele aktuelle Studien liefern einen Beitrag zum Einsatz neuronaler Netze beim Erlernen manöverbasierter Vorhersagen (Deo & Trivedi, 2018a, 2018b; Dai et al., 2019, 2020). Besonders interessant für diese Aufgabe sind Netze mit langem Kurzzeitgedächtnis (engl.: *long short-term memory, LSTM*). Diese rekurrenten neuronalen Netze verfügen über einen Rückkopplungsmechanismus und sind daher in der Lage, ganze Datenreihen zu verarbeiten (Hochreiter & Schmidhuber, 1997). Das Problem ist hierbei das gleiche wie bei anderen überwachten Lernmethoden: die enorme Menge an benötigten Trainingsdaten. Theoretisch können die Vorhersageunsicherheiten bei ausreichender Datenmenge auf ein Minimum reduziert werden. In den nächsten Jahren wird sich zeigen, wie weit wir damit kommen und welche Strategie sich letztlich durchsetzt.

### Bahnkurvenermittlung

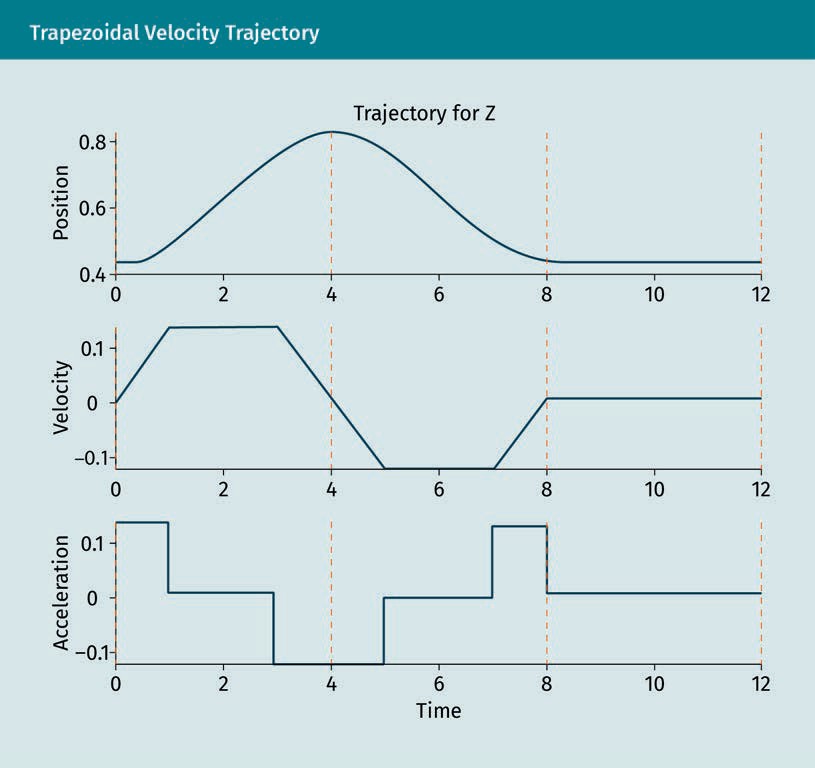
Bisher haben wir gelernt, wie wir einen Pfad innerhalb des Arbeitsraums oder des Konfigurationsraums planen, was als Pfad- bzw. Bewegungsplanung bezeichnet wird. Ein Pfad ist eine geometrische Beschreibung, der das System folgen sollte. Ein weiterer wichtiger Begriff ist die Bahnkurve (Trajektorie) und damit die Bahnkurvenermittlung. Eine Bahnkurve ist ein Pfad mit einem bestimmten Zeitplan. Die Werte der Systemgeschwindigkeit, Beschleunigung und höheren Ableitungen der Wegfunktion sind also an jedem Punkt deﬁniert. Während ein Pfad ein rein räumliches Konstrukt ist, hat eine Bahnkurve eine Zeitkomponente. Der Zweck einer Bahnkurve ist die Annäherung an den geplanten Pfad. Auf Grundlage der Interpolation von Wegpunkten und unter Verwendung zeitbasierter Steuersollwerte kann das System so von einer Ausgangs- zu einer Zielkonfiguration gelangen. Im Allgemeinen wird die Bahnkurve lokal generiert und benötigt daher nicht die gesamte globale Information über den Pfad. Teilabschnitte des Pfades werden durch eine bestimmte Klasse von Polynomfunktionen approximiert. Die Verbindung von zwei oder mehr Wegpunkten wird als Bahnkurvenermittlung bezeichnet. In der folgenden Abbildung ist der Vorgang der Bewegungsplanung schematisch dargestellt.



Die Eingabe der gesamten Bewegungsplanungs-Pipeline ist eine externe Aufgabe, die als Missions- oder Auftragsplanung bezeichnet wird. Sie stellt die Definition der Ausgangs- und Zielkonfiguration des Systems dar. Zunächst wird der geometrische Pfad für diese Aufgabe geplant, und anschließend werden die generierten Bahnkurven als Eingaben für die Systemsteuerung verwendet, wobei eine Rückkopplungssteuerung (Regelkreis) den gewünschten Zustand mit dem tatsächlichen Zustand vergleicht. Im Allgemeinen besteht das Ziel darin, gleichmäßige Bahnkurven mit bestimmten Geschwindigkeits- oder Beschleunigungsbegrenzungen zu erzeugen, sodass ein plötzliches Abbremsen und Beschleunigen verhindert wird.

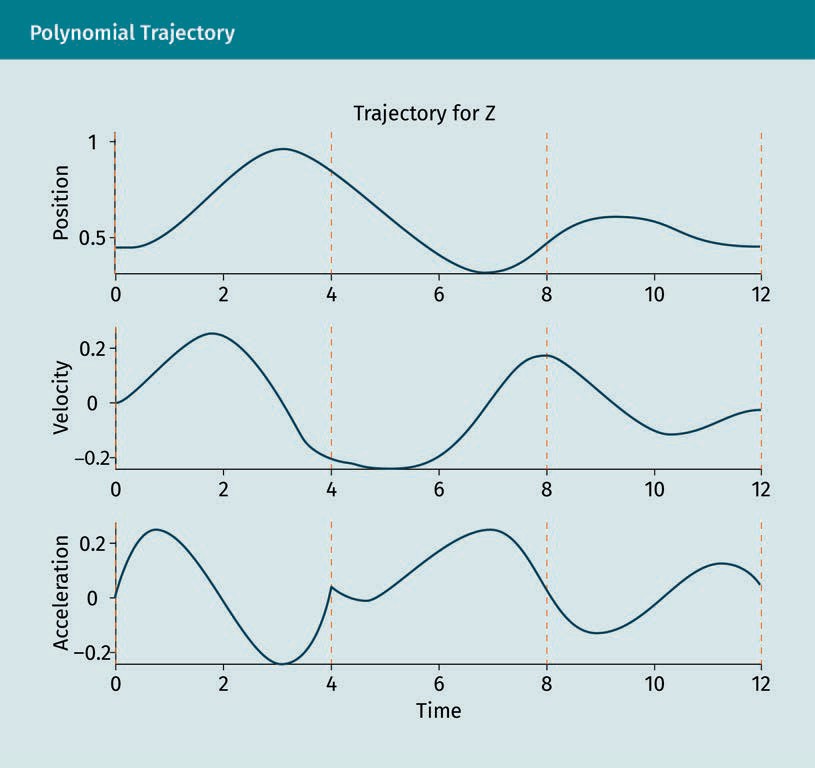
Dies ist besonders wichtig für autonome Fahrzeuge, bei denen der Fahrkomfort berücksichtigt werden muss.

Bahnkurven können auf viele verschiedene Arten erzeugt werden. Die gebräuchlichsten Ansätze sind trapezförmige Geschwindigkeits-Bahnkurven, polynomiale Bahnkurven und Polynomzug-Bahnkurven (engl.: *spline*). Für trapezförmige Geschwindigkeits-Bahnkurven werden Teile von Bahnen mit konstanter Beschleunigung, Nullbeschleunigung und konstanter Abbremsung verwendet. Die resultierende Geschwindigkeit hat aufgrund der Integration der zugrunde liegenden Beschleunigung die Form eines Trapezes. Folglich ähnelt ihr Positionsprofil einer S-Form. Die Grafen der drei Parameter sind in der folgenden Abbildung zu sehen.

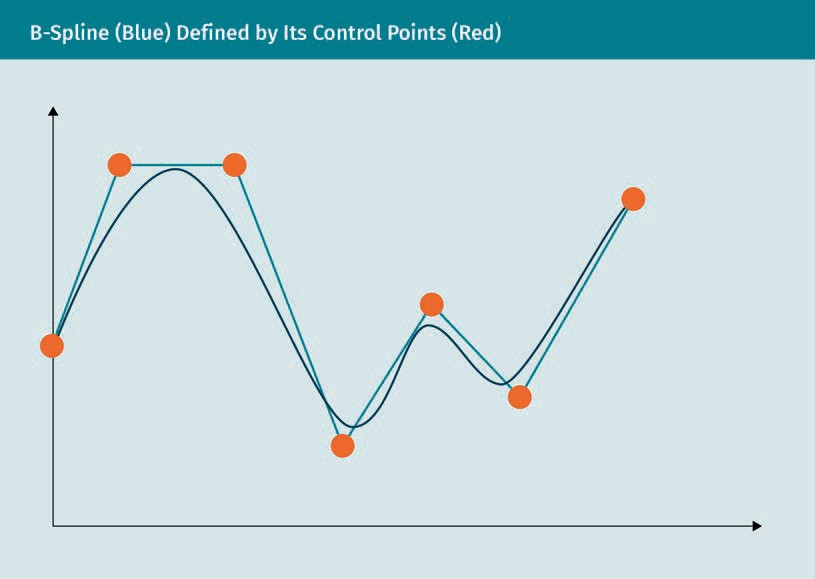


Ein anderer Ansatz ist die Interpolation von zwei Wegpunkten mit Hilfe von Polynomfunktionen unterschiedlicher Ordnung. Hierbei werden meist Polynome dritter (kubischer) oder fünfter (quintischer) Ordnung verwendet, die vier bzw. sechs Randbedingungen erfordern. Erstere benötigen die Position und die Geschwindigkeit am ersten und zweiten Wegpunkt, letztere verwenden auch die Beschleunigung an beiden Punkten.

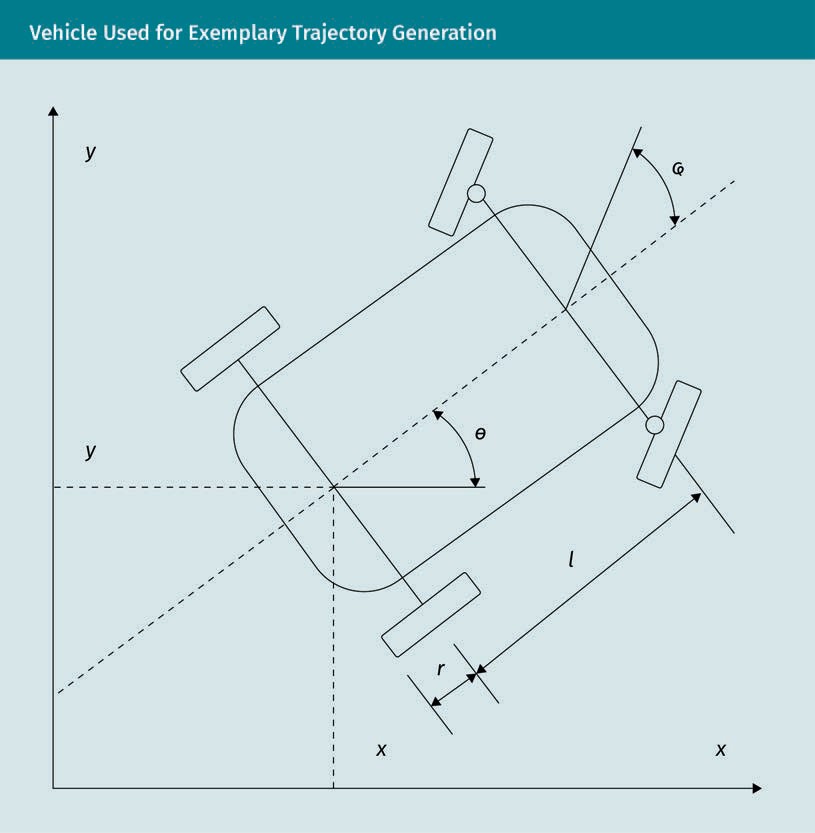
Die Beschleunigungsproﬁle sind im Vergleich zu den trapezförmigen Geschwindigkeits-Bahnkurven glatt, aber die Validierung und Feinabstimmung ist eine schwierigere Aufgabe. Die folgende Abbildung zeigt eine beispielhafte polynomiale Bahnkurve.



Darüber hinaus ist es üblich, interpolierende Bahnkurven mit Hilfe von Splines zu erzeugen. Im Gegensatz zu den für polynomiale Bahnkurven verwendeten Polynomen handelt es sich hierbei um Kombinationen räumlicher Polynome. Splines ermöglichen sehr viel komplexere Formen von Bahnkurven. Der erforderliche Zeitablauf wird erreicht, indem die erzeugten Splines mit gleichmäßiger Geschwindigkeit verfolgt werden. Unter den verschiedenen genutzten Spline-Arten sind Basis-Splines (B-Splines) am häufigsten. B-Splines legen Zwischenkontrollpunkte fest, deren konvexe Hüllen als Begrenzung für die Splines verwendet werden. Die folgende Abbildung zeigt ein Beispiel für einen B-Spline.



In der folgenden Abbildung sehen wir eine beispielhafte Bahnkurvenermittlung für ein Fahrzeug.



Die x- und y-Positionen des Mittelpunkts der Hinterachse sind mit x bzw. y bezeichnet. θ steht für die Orientierung des Roboters und φ für den Lenkwinkel. Der Abstand zwischen Vorder- und Hinterachse sowie der Raddurchmesser werden durch l bzw. r dargestellt. Das kinematische Modell des abgebildeten Fahrzeugs kann wie folgt beschrieben werden (Minh, 2014):

x˙ cos θ 0

y˙ sin θ 0

θ˙ tan φ r · v1 + 0 v2

=

l

φ˙ 0 1

(5.3)

Hier stehen v1 und v2 für die Winkelgeschwindigkeit des Hinterrads bzw. die Lenkgeschwindigkeit. v1 und θ können wie folgt berechnet werden:

v1 =

x˙2 + y˙2

r

θ = arctan

y˙

x˙

(5.4)

(5.5)

Die Winkelgeschwindigkeit der Fahrzeugkarosserie wird durch Ableitung der Gleichung für θ berechnet:

θ˙ = y¨x˙ − x¨y˙ 1 = y¨x˙ − x¨y˙ = tan φr ·v

x˙2 2 + 1 x˙2 + y˙2 l 1

y˙

x˙

(5.6)

Somit können θ und φ direkt anhand der Geschwindigkeiten und Beschleunigungen in x- und y-Richtung berechnet werden. Mit Hilfe der Flachheits-Methode erhalten wir für die Bahnkurvenermittlung die folgende Randbedingung (Levine, 2009; Minh, 2014):

∂2y tan φ

∂x2 = lcos3 θ

(5.7)

Folglich ergeben sich die Anfangs- und Endzustände von x und y zum Zeitpunkt t = 0 bzw. t = T wie folgt:

x 0 = x0

(5.8)

x T = xT

(5.9)

y 0 = y0 = tan θ0

t = 0



∂2y

∂x2

= tan φ0

lcos3 θ0

(5.10)

y T = yT = tan θT

t = T



∂2y

∂x2

= tan φT

lcos3 θT

(5.11)

Wir nehmen die folgenden Bedingungen an:

x˙ t ≥ ε > 0

und

ε = xT − x0 > 0 2T

Dann können wir x(t) folgendermaßen schreiben:

x t = T − t

x0 + t xT + xT − x0 · t t − T

T T 2T2

(5.12)

Nun wählen wir y(t) als Polynom fünfter Ordnung:

y t = y + tα · tan θ

+ t2 α2tan φ0 + t3b

+ t4b + t5b

0 1 0

2lcos3 θ0

1 2 3

(5.13)

Hierbei gilt:

b = b1, b2, b3 = A−1c

(5.14)

T3 T4 T5

A = 3T2 4T3 5T4 6T 12T2 20T3

y − y − Tα tan θ

— T2 α2tan φ0

(5.15)

T 0 1 0

2lcos3 θ0

c = α tan θ − α tan θ

— Tα2tan φ0

3 0 1 0

2lcos3 θ0

α2tan φT − α2tan φ0

2lcos3 θT 2lcos3 θ0

α = 2 xT − x0 − xT − x0

1

2T

(5.16)

α = xT − x0 2 T2

(5.17)

α = 2 xT − x0 + xT − x0

3

2T

(5.18)

(5.19)

Aus (5.5) und (5.11) erhalten wir nun:

2 α2tanφ0 2 3 4

2T α1 · tan θ0 + t 3 + 3t b1 + 4t b2 + 5t b3

θ = arctan 2lcos θ0

2T xT − x0 − T xT − x0 + 2t xT − x0

(5.20)

α2tanφ0 + 6t · b

+ 12t2b

+ 20t3b

2T2 2 3

3 1 2

3 lcos θ

φ = arctan 2lcos θ0

2T xT

* x0
* T xT
* x0

+ 2t xT

2

* x0

(5.21)

Dieses Ergebnis können wir nun zur Bestimmung der Winkelgeschwindigkeit in Gleichung (5.3) heranziehen. Durch Ableitung der Gleichungen (5.12) und (5.13) erhalten wir die Geschwindigkeiten und Beschleunigungen in x- und y-Richtung:

x˙ t = 2T xT − x0 − T xT − x0 + 2t xT − x0

2T2

(5.22)

x¨ t = xT − x0

T2

˙ α2tan φ0 2 3 4

(5.23)

y t = α1 · tan θ0 + t 3 + 3t b1 + 4t b2 + 5t b3

2lcos θ0

y¨ t = α2tan φ0 + 6tb 2lcos3 θ0

1

+ 12t2b2

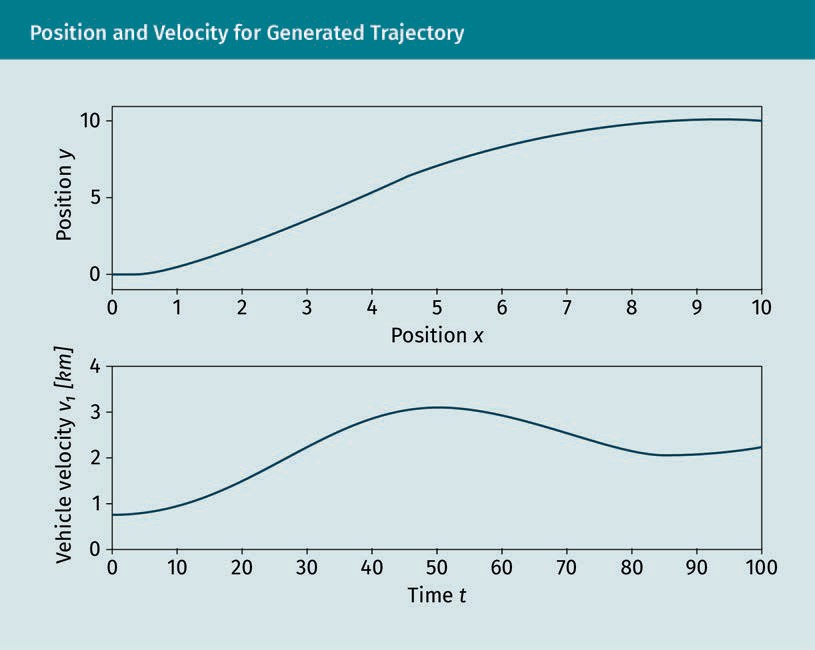
+ 20t3b3

(5.24)

(5.25)

Unter Verwendung der Gleichungen (5.22)–(5.24) für (5.4) und (5.6) können wir nun die Winkelgeschwindigkeiten v1 und θ˙ des Fahrzeugs berechnen. Eine detaillierte Herleitung der Gleichungen ist in der Literatur von Minh (2014) zu finden. Die hergeleiteten Gleichungen können nun zur Bahnkurvenermittlung von der Anfang- bis zur End-Pose genutzt werden.

In der folgenden Abbildung sind die Graphen der Position (oben) und der Geschwindigkeit (unten) für eine erzeugte Bahnkurve von x0 = [0 0 0 0] nach xT = [10 10 0 π/6] mit T = 100 dargestellt.



Die vorgestellte Berechnung ist lediglich ein möglicher Ansatz zur Bahnkurvenermittlung. In neueren und vielversprechenden Studien werden der Einsatz von evolutionären Algorithmen und Partikelschwarm-Optimierung für eine verbesserte Bahnkurvenermittlung untersucht (Sathiya & Chinnadurai, 2019; Zhou et al., 2019).

Zusammenfassung

Die Bewegungsplanung ist ein wesentlicher Bestandteil mobiler Systeme und besteht aus Missionsplanung, Pfadplanung, Bahnkurvenermittlung und Steuerung. Der Missionsplan ist eine externe Aufgabe und legt die Ausgangs- und Zielpunkte der Mission fest. Ziel der Pfadplanung ist es, einen geometrischen Pfad zu erstellen, wobei das Problem darin besteht, den kürzesten Pfad von einer Ausgangs- zur Zielposition zu finden. Für die Diskretisierung des Konfigurationsraums eines Systems, der für die meisten Planungsalgorithmen benötigt wird, gibt es verschiedene Ansätze. Diese lassen sich in kombinatorische und stichprobenbasierte Planung unterteilen. Bei kombinatorischen Planungsmethoden werden mit Hilfe von Graphen die Konnektivität des freien Konfigurationsraums erfasst und Lösungen darin gefunden, was zu einer Straßenkarte (Roadmap) führt. Die gängigsten kombinatorischen Planungsmethoden sind der Sichtbarkeitsgraph, das Voronoi-Diagramm und die Zellzerlegung.

Eine Umgebung, die nur aus statischen Hindernissen besteht, entspricht nicht der Realität. Dynamische Objekte wie andere Verkehrsteilnehmer, müssen bei der Bewegungsplanung berücksichtigt werden. Um diese recht komplexe Aufgabe zu vereinfachen, können verschiedene Annahmen getroffen werden, darunter physikbasierte, manöverbasierte und interaktionsbasierte Annahmen. Physikbasierte und manöverbasierte Annahmen stützen sich auf physikalische Beschränkungen der Bewegung bzw. auf üblicherweise ausgeführte Manöver einer bestimmten Art dynamischer Objekte. Interaktionsbasierte Annahmen berücksichtigen ebenfalls die Menge der üblichen Manöver, darüber hinaus aber auch die Interaktion mehrerer Objekte.

Eine Bahnkurve (Trajektorie) ist ein Pfad mit einem bestimmten Zeitplan. Die Werte der Systemgeschwindigkeit, Beschleunigung und höheren Ableitungen der Wegfunktion sind an jedem Punkt deﬁniert. Während ein Pfad ein rein räumliches Konstrukt ist, hat eine Bahnkurve eine Zeitkomponente. Die Idee der Bahnkurvenermittlung besteht darin, den geplanten Pfad durch die Interpolation von Wegpunkten anzunähern. Unter verschiedenen Ansätzen sind die trapezförmige Geschwindigkeits-Bahnkurve, die polynomiale Bahnkurve und die Polynomzug-Bahnkurve (engl.: *spline*) am gebräuchlichsten.



# Anhang 1

## Literaturverzeichnis

Literaturverzeichnis

Abdollahpouri, M., Takács, G., & Rohaľ-Ilkiv, B. (2017). Real-time moving horizon estima- tion for a vibrating active cantilever. *Mechanical Systems and Signal Processing, 86*, 1— 15. https://doi.org/10.1016/j.ymssp.2016.09.028

Aidan. (2018). *How GPS receivers work*. Core Electronics. https://core-electron- ics.com.au/tutorials/how-GPS-works.html

Amato, N. M., & Wu, Y. (1996). A randomized roadmap method for path and manipulation planning. *Proceedings of the 1996 IEEE International Conference on Robotics and Auto- mation, 1,* 113—120.

Andreas06. (2006). *Zusatzzeichen 1024-10—Personenkraftwagen frei, StVO 1992* [Addi- tional sign 1024-10—passenger cars free, Road Trafﬁc Regulations 1992]. *Wikimedia Com- mons*. https://en.wikipedia.org/wiki/File:Zusatzzeichen\_1024-10\_-\_Personenkraftwa- gen\_frei,\_StVO\_1992.svg

Bancroft, S. (1985). An algebraic solution of the GPS equations*. IEEE Transactions on Aerospace and Electronic Systems*, (1), 56—59. https://doi.org/10.1109/TAES.1985.310538

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: Theory algorithms and software*. Wiley.

Bartsch, A., Fitzek, F., & Rasshofer, R. H. (2012). Pedestrian recognition using automotive radar sensors. *Advances in Radio Science: ARS, 10*, 45—55.

Baxter, R. H., Leach, M. J., Mukherjee, S. S., & Robertson, N. M. (2014). An adaptive motion model for person tracking with instantaneous head-pose features. *IEEE Signal Process- ing Letters, 22*(5), 578—582.

Ben-Ari, M., & Mondada, F. (2017). *Elements of robotics*. Springer.

Bhattacharya, P., & Gavrilova, M. L. (2008). Roadmap-based path planning—Using the Voronoi diagram for a clearance-based shortest path. *IEEE Robotics & Automation Mag- azine, 15*(2), 58—66. https://doi.org/10.1109/MRA.2008.921540

Borenstein, J., & Koren, Y. (1991). The vector ﬁeld histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation, 7*(3), 278—288.

Burgard, W., Stachniss, C., Bennewitz, M., & Arras, K. (2011). *Introduction to mobile robot- ics: Robot motion planning*. Uni Freiburg. <http://ais.informatik.uni-freiburg.de/teach-> ing/ss11/robotics/slides/18-robot-motion-planning.pdf

Castro, S. (2019). *Trajectory planning for robot manipulators.* Mathworks. https:// medium.com/mathworks/trajectory-planning-for-robot-manipulators-522404efb6f0

List of References

Chen, Z. S. (2003). Bayesian ﬁltering: From Kalman ﬁlters to particle ﬁlters, and beyond. *Statistics: A Journal of Theoretical and Applied Statistics, 182*(1), 1—69. https://doi.org/ 10.1080/02331880309257

Cook, S. (2006). The P versus NP problem. In J. Carlson, A. Jaffe, & A. Wiles (Eds.), *The mil- lennium prize problems* (pp. 87—104). Clay Mathematics Institute.

Dai, S., Li, L., & Li, Z. (2019). Modeling vehicle interactions via modiﬁed LSTM models for trajectory prediction. *IEEE Access, 7*, 38287—38296. https://doi.org/10.1109/ ACCESS.2019.2907000

Dai, S., Li, Z., Li, L., Zheng, N., & Wang, S. (2020). A ﬂexible and explainable vehicle motion prediction and inference framework combining semi-supervised AOG and ST-LSTM. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/ TITS.2020.3016304

Dellaert, F., Fox, D., Burgard, W., & Thrun, S. (1999). Monte Carlo localization for mobile robots. *Proceedings of the 1999 IEEE International Conference on Robotics and Automa- tion, 2*, 1322—1328.

Deo, N., & Trivedi, M. M. (2018a). Convolutional social pooling for vehicle trajectory pre- diction. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1468—1476). IEEE. https://arxiv.org/pdf/1805.06771.pdf

Deo, N., & Trivedi, M. M. (2018b). Multi-modal trajectory prediction of surrounding vehi- cles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1179—1184). IEEE. https://doi.org/10.1109/IVS.2018.8500493

Deuﬂhard, P. (2011). *Newton methods for nonlinear problems*. Springer.

Durrett, R. (2019). *Probability: Theory and examples (Vol. 49)*. Cambridge University Press. https://services.math.duke.edu/~rtd/PTE/PTE5\_011119.pdf

Eager, D., Pendrill, A. M., & Reistad, N. (2016). Beyond velocity and acceleration: Jerk, snap and higher derivatives. *European Journal of Physics, 37*(6). https://doi.org/ 10.1088/0143-0807/37/6/065008

Elasmar, S. R. A. G. (2012). *Information-theoretic environment modeling for mobile robot localization* [Unpublished doctoral dissertation]. Universität Mannheim.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Com- puter, 22*(6), 46—57. https://doi.org/10.1109/2.30720

El-Rabbany, A. (2002). *Introduction to GPS: The global positioning system*. Artech House.

Fox, D. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artiﬁcal Intelligence Research, 11,* 391—427.

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoid- ance. *IEEE Robotics & Automation Magazine, 4*(1), 23—33. https://doi.org/ 10.1109/100.580977

Fox, D., Burgard, W., & Thrun, S. (1999a). Markov localization for mobile robots in dynamic environments. *Journal of artiﬁcial intelligence research, 11*, 391—427. https:// doi.org/10.1613/jair.616

Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999b). Monte Carlo localization: Efﬁcient position estimation for mobile robots. *Proceedings of the Sixteenth National Confer- ence on Artiﬁcial Intelligence and the Eleventh Innovative Applications of Artiﬁcial Intel- ligence Conference Innovative Applications of Artiﬁcial Intelligence,* 343—349.

Frontoni, E. (2012). *Vision based mobile robotics: Mobile robot localization using vision sensors and active probabilistic approaches.* Università Politecnica Delle Marche.

Giefer, L. A., Clemens, J., & Schill, K. (2020). Extended object tracking on the afﬁne group Aff (2). In *2020 23rd International Conference on Information Fusion (FUSION 2020)* (pp. 321—328). IEEE. https://doi.org/10.23919/FUSION45008.2020.9190566

Gonzalez, R., Kloetzer, M., & Mahulea, C. (2017). Comparative study of trajectories resulted from cell decomposition path planning approaches. In M. Kloetzer & L. Ferariu (Eds.), *Proceedings of the 2017 21st International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 49—54). IEEE. https://doi.org/10.1109/ICSTCC.2017.8107010

Guerra, M., Eﬁmov, D., Zheng, G., & Perruquetti, W. (2016). Avoiding local minima in the potential ﬁeld method using input-to-state stability. *Control Engineering Practice, 55*, 174—184. https://doi.org/10.1016/j.conengprac.2016.07.008

Hilgert, J., & Neeb, K. H. (2012). *Structure and geometry of Lie groups*. Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735—1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, G. P., Mourikis, A. I., & Roumeliotis, S. I. (2008). Analysis and improvement of the consistency of extended Kalman ﬁlter based SLAM. *Proceedings of the 2008 IEEE Inter- national Conference on Robotics and Automation*, 473—479. https://doi.org/10.1109/ ROBOT.2008.4543252

IIP Photo Archive. (2014). Taft point, Yosemite national park, California, US. *Wikimedia Commons.* https://commons.wikimedia.org/wiki/File:Taft\_Point,\_Yosem- ite\_National\_Park,\_California,\_U.S.jpg

Isaev, A. P., & Rubakov, V. A. (2018). Theory of groups and symmetries: Finite groups, Lie groups, and Lie algebras. *World Scientiﬁc, 60*(3), 275. https://doi.org/ 10.1080/00107514.2019.1663933

List of References

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman ﬁlter to nonlinear sys- tems. *Signal Processing, Sensor Fusion, and Target Recognition VI,* (3068), 182—193.

Karasev, V., Ayvaci, A., Heisele, B., & Soatto, S. (2016, May). Intent-aware long-term predic- tion of pedestrian motion. In *2016 IEEE International Conference on Robotics and Auto- mation (ICRA)* (pp. 2543—2549). IEEE.

Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional conﬁguration spaces. *IEEE transactions on Robotics and Automation, 12*(4), 566—580.

kevint3141. (2009). This is a photo of Nik Wallenda walking on a tight rope from Medieval Fare to Wonder Mountain at Canada's Wonderland. *Wikimedia Commons*. https:// commons.wikimedia.org/wiki/File:Wonderland\_Walker\_2.jpg

Klein, R. (2006). *Algorithmische geometrie: Grundlagen, methoden, anwendungen* [Algo- rithmic geometry: Fundamentals, methods, applications]. Springer.

Kloetzer, M., Mahulea, C., & Gonzalez, R. (2015). Optimizing cell decomposition path plan- ning for mobile robots using different metrics. In S. Caraman, M. Barbi, & R. Şolea (Eds.), *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 565—570). IEEE. https://doi.org/10.1109/ICSTCC.2015.7321353

Latombe, J.-C. (2012). *Robot motion planning*. Springer.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. In K.

R. Kozlowski (Ed.), *Robot Motion in Control 2009* (pp. 307—316). Springer.

Lee, T. J., Kim, C. H., & Cho, D. (2019). A monocular vision sensor-based efﬁcient SLAM method for indoor service robots. *IEEE Transactions on Industrial Electronics, 66*(1), 318

—328.

Levine, J. (2009). *Analysis and control of nonlinear systems: A ﬂatness-based approach*. Springer Science & Business Media.

Long, A. W., Wolfe, K. C., Mashner, M. J., & Chirikjian, G. S. (2013). The banana distribution is Gaussian: A localization study with exponential coordinates. *Robotics: Science and Systems VIII, 8*, 265—272.

Lulu, L., & Elnagar, A. (2005). A comparative study between visibility-based roadmap path planning algorithms. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3263—3268). IEEE. https://doi.org/10.1109/ IROS.2005.1545545

Meyer, J.-A., & Filliat, D. (2003). Map-based navigation in mobile robots—II. A review of map-learning and path-planning strategies. *Cognitive Systems Research, 4*(4), 283—317.

Minh, V. T. (2014). Trajectory generation for autonomous mobile robots. In Z. H. Kahn, A.

B. M. Shawkat, & Z. Riez (Eds.). *Computational intelligence for decision support in cyber- physical systems* (pp. 195—214). Springer.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology, 47*(1), 90—100. https://doi.org/10.1016/S0022-2496(02)00028-7

Ogren, P., & Leonard, N. E. (2005). A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics, 21*(2), 188—195. https://doi.org/10.1109/ TRO.2004.838008

Park, M. G., & Lee, M. C. (2003). A new technique to escape local minimum in artiﬁcial potential ﬁeld based path planning. *KSME International Journal, 17*(12), 1876—1885.

Patle, B. K., Ganesh Babu, L., Pandey, A., Parhi, D. R. K., & Jagadeesh, A. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology, 15*(4), 582—606. https://doi.org/10.1016/j.dt.2019.04.011

Peng, M., Wang, C., Chen, T., Liu, G., & Fu, X. (2017). Dual temporal scale convolutional neural network for micro-expression recognition. *Frontiers in Psychology, 8(*1745). https://doi.org/10.3389/fpsyg.2017.01745

Petrovski, I. G. (2014). *GPS, GLONASS, Galileo, and BeiDou for mobile devices: From instant to precise positioning*. Cambridge University Press.

Pierlot, V., & Van Droogenbroeck, M. (2014). A new three object triangulation algorithm for mobile robot positioning. *IEEE Transactions on Robotics, 30*(3), 566—577. https:// doi.org/10.1109/TRO.2013.2294061

Quinlan, S., & Khatib, O. (1993). Elastic bands: Connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 802— 807). IEEE. https://doi.org/10.1109/ROBOT.1993.291936

Rahemi, N., Mosavi, M. R., Abedi, A. A., & Mirzakuchaki, S. (2014). Accurate solution of nav- igation equations in GPS receivers for very high velocities using pseudorange measure- ments. *Advances in Aerospace Engineering*, *2014*, 1—8. https://doi.org/ 10.1155/2014/435891

Ryaben’kii, V. S., & Tsynkov, S. V. (2006). *A theoretical introduction to numerical analysis*. CRC Press.

Sathiya, V., & Chinnadurai, M. (2019). Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning. *Robotica, 37*(8), 1363—1382.

Schäfer, D. (2003). *Globale selbstlokalisation autonomer mobiler roboter—Ein schlüssel- problem der service-robotik* [Global self-localization of autonomous mobile robots—A key problem in service robotics] [Unpublished doctoral dissertation]. Universität Würz- burg.

List of References

Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics*. Springer.

Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connex- ion. *Control and Cybernetics, 35*(3), 599—620.

Solà Ortega, J., Deray, J., & Atchuthan, D. (2018). A micro Lie theory for state estimation in robotics. Cornell University. https://arxiv.org/abs/1812.01537

Tătar, M. O., Cirebea, C., & Mândru, D. (2013). Structures of the omnidirectional robots with swedish wheels. *Solid State Phenomena, 198*, 132—137.

Tay, S., & Marais, J. (2013). Weighting models for GPS Pseudorange observations for land transportation in urban canyons. In *6th European Workshop on GNSS Signals and Signal Processing.* HAL Archives-Ouvertes. https://hal.archives-ouvertes.fr/hal-00942180/ document

Vo, B.-N., & Ma, W.-K. (2006). The Gaussian mixture probability hypothesis density ﬁlter.

*IEEE Transactions on Signal Processing, 54*(11), 4091—4104.

Vo, B.-T., Vo, B.-N., & Cantoni, A. (2009). The cardinality balanced multi-target multi-Ber- noulli ﬁlter and its implementations. *IEEE Transactions on Signal Processing, 57*(2), 409

—423.

Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driv- ing: Common practices and emerging technologies. *IEEE Access*, *8*, 58443—58469. https://doi.org/10.1109/access.2020.2983149

Zhang, C., Taghvaei, A., & Mehta, P. G. (2016). Feedback particle ﬁlter on matrix Lie groups. In D. Abramovitch (Ed.), *2016 American Control Conference (ACC)* (pp. 2723—2728). IEEE.

Zhou, H., Wang, X., & Cui, N. (2019). A novel reentry trajectory generation method using improved particle swarm optimization. *IEEE Transactions on Vehicular Technology, 68*(4), 3212—3223.

Zwillinger, D., & Kokoska, S. (2000). *Standard probability and statistics tables and formu- lae*. CRC Press.