

Robotik und Künstliche Intelligenz

Prof. Dr.-Ing. J. Adamy

Projektseminar

REGELUNGSMETHODEN
UND ROBOTIK **rmr**



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Robotik und Künstliche Intelligenz

Vorlesung und Skript: Prof. Dr.-Ing. J. Adamy

Druck: citycopies
Holzstr. 5
64283 Darmstadt
www.citycopies.de

8. korrigierte Auflage

© Darmstadt 2010

Technische Universität Darmstadt
Institut für Automatisierungstechnik
Fachgebiet für Regelungsmethoden und Robotik
D-64283 Darmstadt, Landgraf-Georg-Straße 4
<http://www.rmr.tu-darmstadt.de>

Vorwort zur 1. Auflage

Das (1+3)-Projektseminar „Robotik und Künstliche Intelligenz“ besteht aus einem Vorlesungsteil und Projektarbeiten, die in Teams von zwei bis vier Studenten durchgeführt werden. Der Vorlesungsteil vermittelt das Grundlagenwissen über Industrieroboter und autonome mobile Roboter. Die Projektarbeiten behandeln forschungsrelevante Themen aus der Robotik oder dem Bereich der Künstlichen Intelligenz und werden jeweils von einem Assistenten betreut.

Die Note setzt sich zu 50 % aus der Bewertung der Projektarbeiten und zu 50 % aus einer Prüfung über die Vorlesungsinhalte zusammen.

Darmstadt, August 2000

Jürgen Adamy

INHALTSVERZEICHNIS

1	Roboter – Einführung und Historie	1
2	Mobile Roboter	7
2.1	Typen und Anwendungen	9
2.2	Sensoren	23
2.3	Umweltkarten und Kartenaufbau	37
2.4	Bahnplanung	49
3	Industrieroboter	65
3.1	Typen und Anwendungen	67
3.2	Geometrie und Kinematik	79
3.3	Dynamisches Modell	95
3.4	Regelung von Industrierobotern	105
	Literatur	119

KAPITEL

1

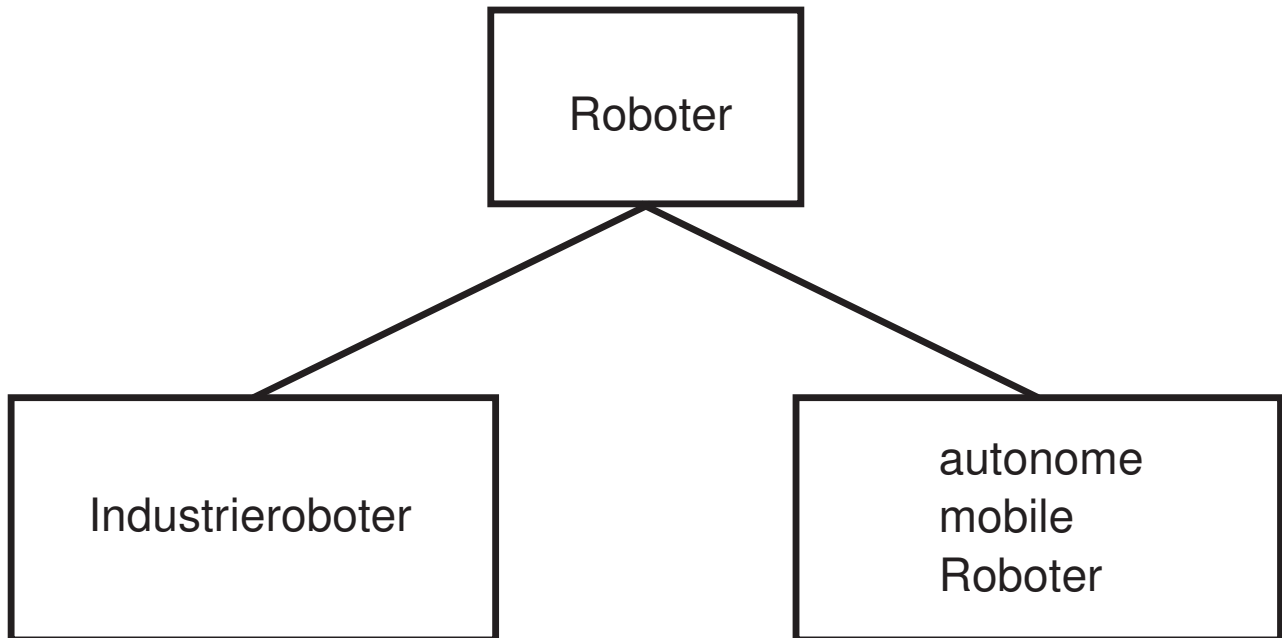
Roboter – Einführung und Historie

Der Begriff Roboter wurde 1921 von K. Čapek, einem tschechischen Schriftsteller, in dem Theaterstück R.U.R. (Rossum's Universal Robots) geprägt. In diesem Theaterstück sind Roboter künstliche Menschen, die den Menschen eintönige Fabrikarbeit abnehmen. Sie entwickeln jedoch ein Eigenleben und bekämpfen ihre Erbauer. Der Begriff Roboter leitet sich vom slawischen Wort für Fronarbeit, robota, ab.

Den ersten Industrieroboter bauten G.C. Devol und J. Engelberger 1958 in den USA. Eingesetzt wurde ein von ihnen entwickelter Industrieroboter 1962 bei General Motors in einer Produktionslinie. Die Neuerung gegenüber ferngesteuerten Manipulatoren oder automatischen Maschinen bestand in der frei programmierbaren Bewegungsabfolge des Systems. Die erste Anwendung erfolgte in Japan 1967 und in Deutschland ab 1970 in der Automobilindustrie. Ab 1980 wurden Industrieroboter in vielen Bereichen und in großer Zahl eingesetzt. Sie gehören heute zum Standard industrieller Automatisierungsmöglichkeiten.

Der erste autonome mobile Roboter, Shakey, wurde 1968 am Stanford Research Institute gebaut. Er verfügte über eine Kamera, konnte seine Umwelt modellieren und eine Bahnplanung durchführen. Shakey war ein landgestütztes System und fuhr auf ebenem Untergrund. Neben landgestützten autonomen mobilen Robotern werden auch autonome mobile Roboter für Unterwasseraufgaben sowie Aufgaben im Luft- und Weltraum entwickelt. Erste kommerzielle, militärische und forschungsorientierte Anwendungen gibt es heute bereits.

Grob lassen sich Roboter in zwei Klassen unterteilen:



Weitgefasst läßt sich der Begriff „Roboter“ wie folgt definieren:

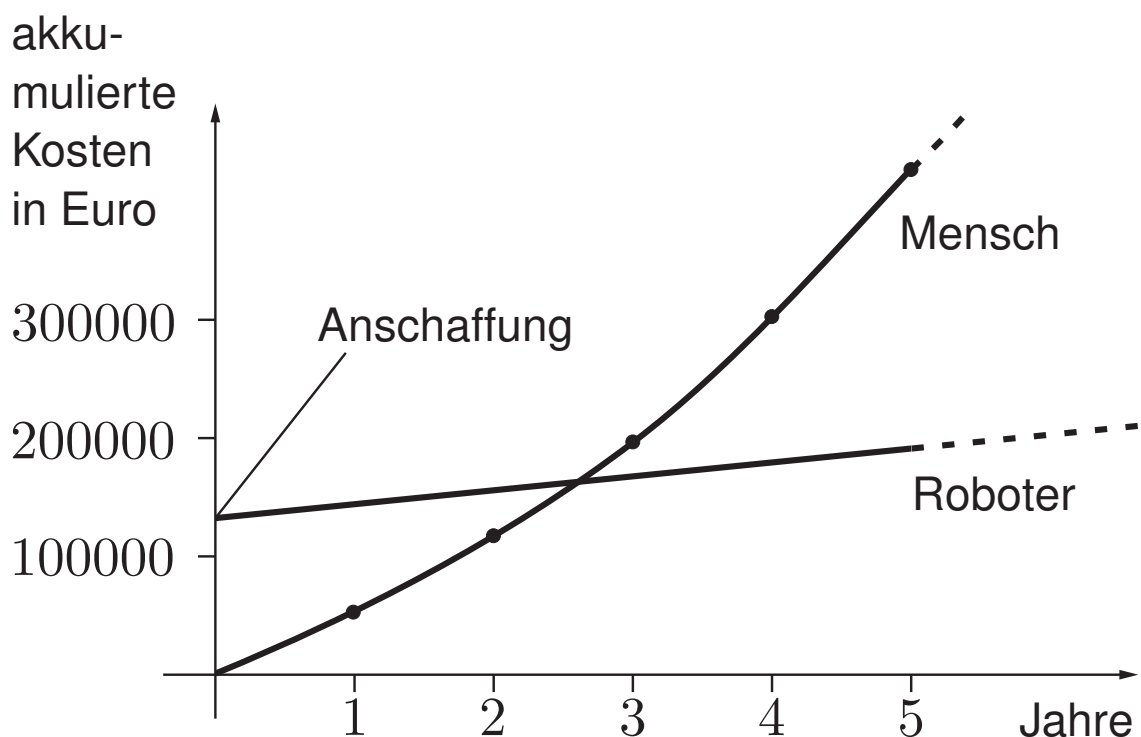
Definition

Ein Roboter ist eine Bewegungen ausführende Maschine, die Eigenschaften besitzt, die denen von Lebewesen nachgebildet sind.

Diese Definition deutet auch bereits den Einsatzzweck von Robotern an: Sie sollen Arbeiten übernehmen, die normalerweise von Menschen oder Tieren ausgeführt werden.

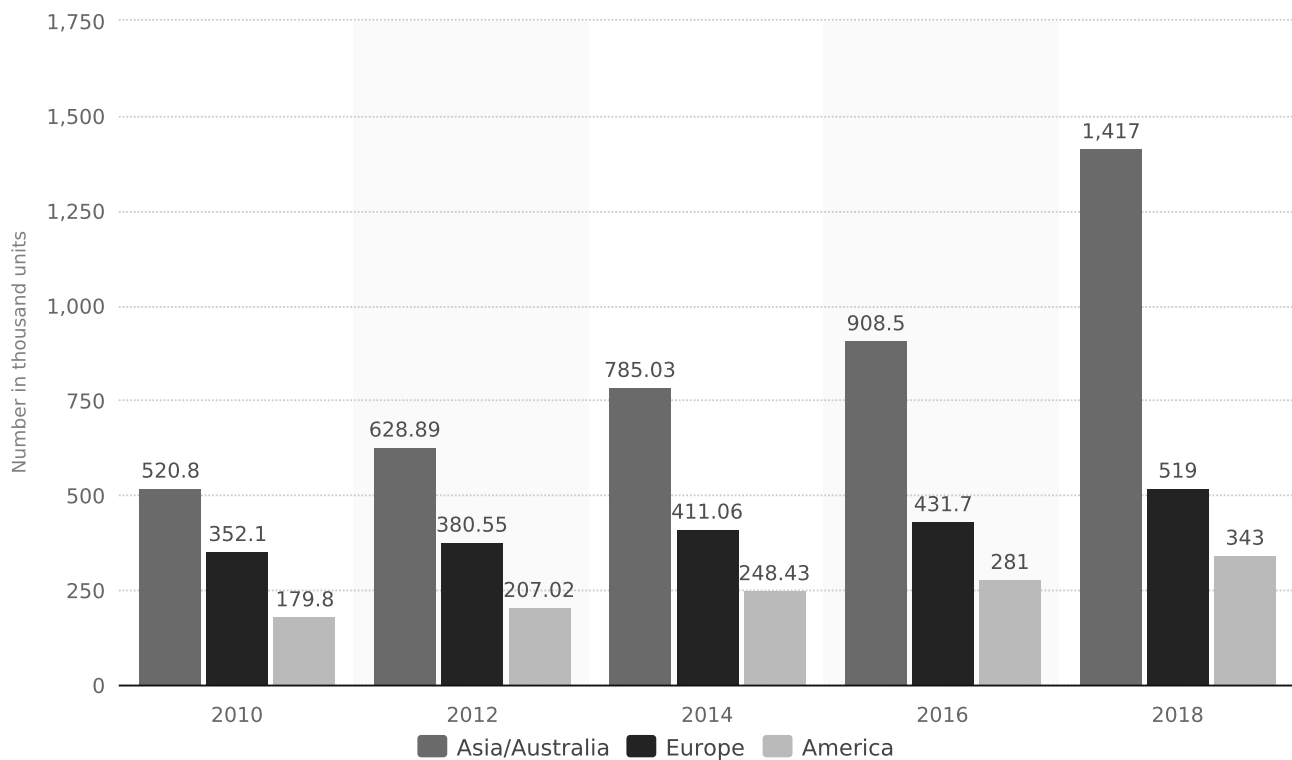
Für den Einsatz von Robotern sind vor allem zwei Gesichtspunkte maßgeblich:

1. Die zu verrichtende Aufgabe kann oder darf nicht von einem Menschen ausgeführt werden. Beispiele:
 - Exploration auf anderen Planeten
 - Inspektion oder Reparatur auf kontaminiertem Gelände
2. Der Nutzen/Kosten-Quotient ist hoch und der Roboter amortisiert sich innerhalb eines bestimmten Zeitraumes. Beispiele:
 - Einlegen des Reserverades in PKW. Die Kosten eines Roboters setzen sich aus den konstanten Anschaffungskosten und den sich jährlich akkumulierenden Wartungskosten zusammen. Die Lohnkosten eines diese Arbeit ausführenden Menschen akkumulieren sich stetig auf, wobei aufgrund von Lohnsteigerungen eine exponentielle Komponente enthalten ist.



Nachfolgendes Diagramm zeigt die Einsatzzahlen von Industrierobotern weltweit¹.

Projected operational stock of multipurpose industrial robots from 2010 to 2018, by region (in 1,000 units)



¹Quelle: <http://www.statista.com/statistics/281379/estimated-operational-stock-of-industrial-robots-by-region/> (Zugriff: März 2016)

KAPITEL

2

Mobile Roboter

ABSCHNITT 2.1

Typen und Anwendungen

Autonome mobile Roboter sollen sich in einer veränderlichen Umgebung ohne externe Steuerung – also autonom – frei bewegen. Dabei soll eine vorgegebene Aufgabe selbständig und umgebungsangepaßt durchgeführt werden. Diese Anforderungen lassen sich detaillieren. D. h. ein autonomer mobiler Roboter soll bzw. kann folgende Eigenschaften aufweisen:

- selbständige Umweltmodellierung (Option),
- Positionsbestimmung,
- selbständige Planung und Ausführung von Aktionen,
- Kommunikation mit der Umwelt,
- Reaktion auf ungeplante Situationen, z. B. Kollisionsvermeidung.

Prinzipiell sind

- landgestützte,
- seegestützte und
- luftgestützte

Roboter zu unterscheiden und entsprechend unterschiedlich ist dann auch ihre technische Auslegung und ihr Einsatzzweck. In der nachfolgenden Tabelle sind einige Anwendungsbeispiele aufgeführt.

landgestützte Systeme:

- Transport
- Überwachung z. B. von Räumen
- Reinigung von Flächen oder Rohren
- Museumsführer
- Exploration
- Spielzeug
- Rasenmäher
- Melkroboter
- Mähdrescher
- automatische Asphaltierung
- Golfcaddy
- Rohrinspektion
- ⋮

seegestützte Systeme:

- Kabelverlegung
- Exploration
- Bergung
- ⋮

luftgestützte Systeme:

- Marschflugkörper
- Drohnen
- ⋮

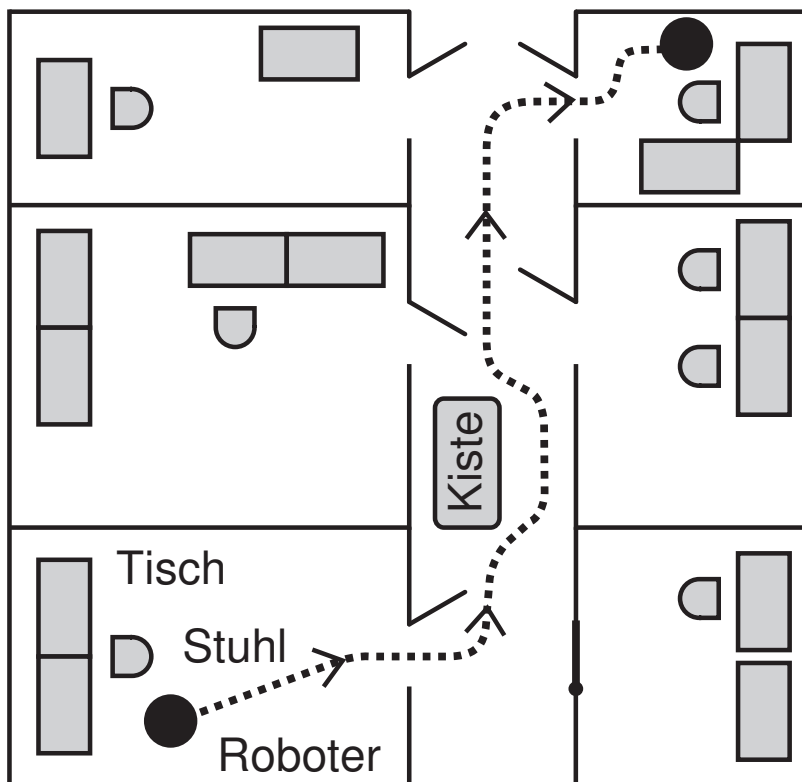
Die Anwendungsbandbreite autonomer mobiler Roboter ist also groß. Nachfolgend werden die wichtigsten landgestützten Systeme etwas genauer betrachtet.

Transportroboter, Überwachungsroboter und Museumsführer o. ä. haben die Aufgabe bestimmte Wege zurückzulegen.

Diese Roboter müssen daher über

- eine Karte des Operationsgebietes,
- eine Wegplanung,
- eine Hindernisvermeidung und
- eventuell über Kommunikationsfähigkeiten

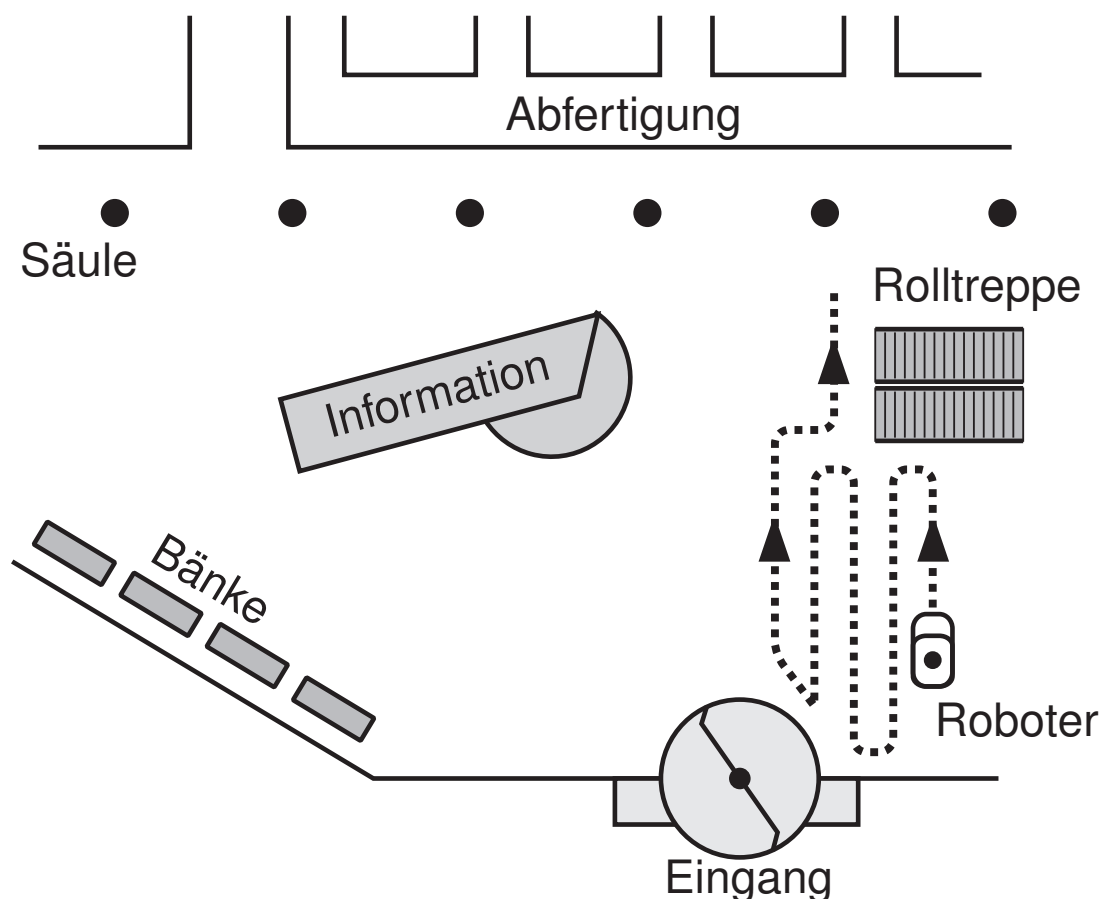
verfügen. Die Kommunikationsfähigkeiten werden z. B. benötigt, um Personen anzusprechen, die den Weg versperren. Nachfolgendes Bild zeigt ein Beispiel für eine Transportaufgabe.



Neben den obigen Autonomieforderungen muß der Roboter auch über entsprechende Sicherheitseinrichtungen verfügen, um Kollisionen mit Personen und Gegenständen zu vermeiden, um Glasscheiben und Treppen zu erkennen, etc.

Transportroboter werden heute z. B. in Krankenhäusern eingesetzt, um Mahlzeiten, Bettlaken, Medikamente etc. zu transportieren. Überwachungsroboter werden z. B. in der chemischen Industrie und in militärischen Anlagen eingesetzt.

Autonome Reinigungsroboter werden als Staubsauger und Schrubbmachines zur Bodenreinigung eingesetzt. Des Weiteren gibt es Roboter für die Fassadenreinigung von Hochhäusern, autonome Rasenmäher und Mähdrescher. Ihnen ist gemeinsam, dass ein vorgegebenes Terrain flächendeckend abgefahren werden soll. Die Wegplanung unterscheidet sich also wesentlich von der des Transportroboters. Nachstehendes Bild illustriert eine solche Flächenabdeckungsaufgabe.



Eingesetzt werden solche Reinigungsroboter heute in der Pariser Metro, in Supermärkten, auf Bahnhöfen in Japan und dem Haneda Flughafen. Diese Reinigungsroboter sind automatisierte Bodenreinigungsmaschinen mit bis zu 800 *kg* Gewicht. Kleinere Reinigungsroboter für Wohnräume mit geringeren Sensor- und Navigationsfähigkeiten finden sich ebenfalls.

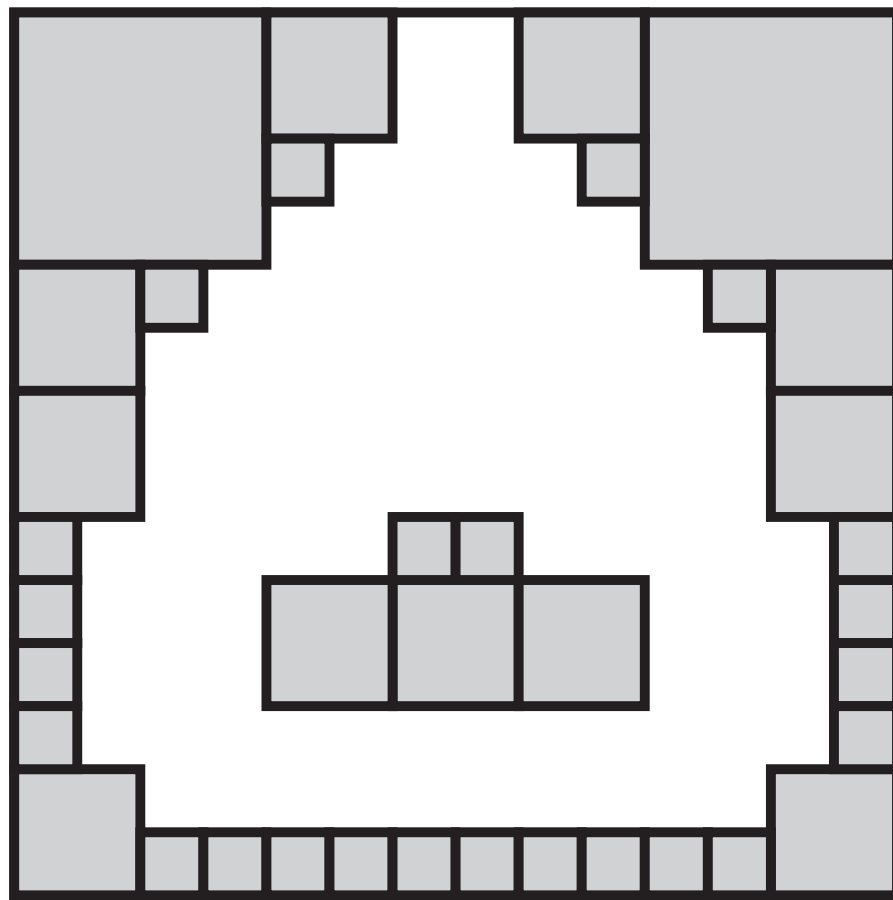
Um die oben beschriebenen Aufgaben abarbeiten zu können, verfügt ein autonomer mobiler Roboter über die folgenden Funktionsmodule.

1. Umweltmodell

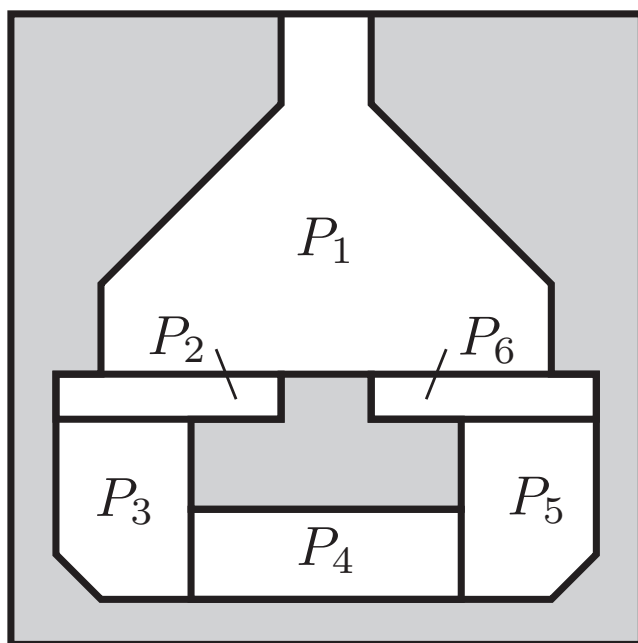
Um sich zielorientiert und kollisionsfrei bewegen zu können, benötigt ein Roboter ein Umweltmodell bzw. eine Umgebungskarte. Eine der einfachsten Umweltmodellierungen ermöglicht das Quadtree-Modell. Hier wird eine Fläche zuerst in vier Quadrate aufgeteilt. Befahrbare Quadrate und nicht befahrbare Quadrate werden unterschiedlich gekennzeichnet. Ist ein Quadrat aus befahrbaren und nicht befahrbaren Flächen zusammengesetzt, so wird es wieder in vier Quadrate unterteilt usw.

Eine weniger speicherintensive Modellierung erlauben polygonale Modelle. Befahrbare Flächen werden hierbei durch konvexe Polyeder bzw. Polygone P_i dargestellt. Neben den Polygonkoordinaten verfügt der Roboter auch über einen Verbindungsgraphen, der angibt, welche Polygone P_i miteinander verbunden sind. Nachfolgende Bilder illustrieren dies. Außer diesen Modellen gibt es eine Reihe weiterer Modellierungsmöglichkeiten.

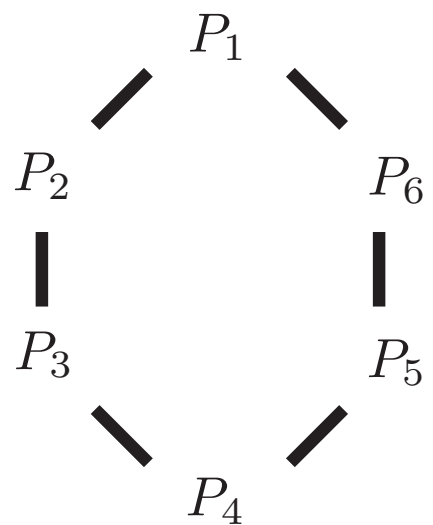
Das Umweltmodell wird entweder off-line eingegeben oder selbständig vom Roboter gelernt.



Quadtrees-Modell



Polygonales Modell



Topologisches Modell

2. Positionsbestimmung

Die Positionsbestimmung erfolgt mittels

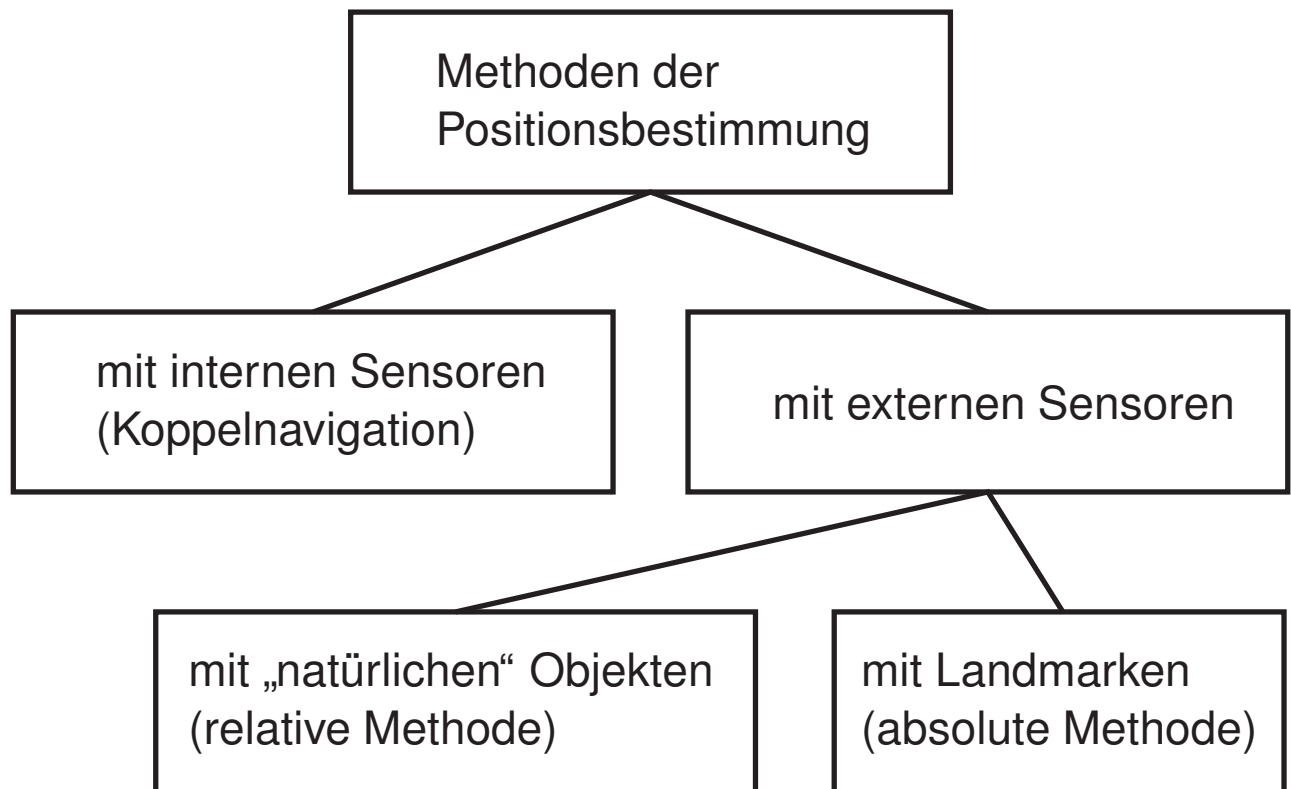
- interner Sensoren und bzw. oder
- externer Sensoren.

Interne Sensoren erlauben die Bestimmung der Position und der Orientierung des Roboters nur aus der Integration gemessener Beschleunigungen bzw. Geschwindigkeiten und eines kinematischen Robotermodells. Also ohne Kenntnis von Umweltdaten. Dieses Vorgehen bezeichnet man als Koppelnavigation. Gyroskope und Tachometer sind interne Sensoren.

Externe Sensoren, wie Kameras oder Entfernungsmesser, erkennen die Umgebung – genauer Teile von ihr – und schließen so auf die Roboterposition. Dies kann beispielsweise anhand besonders markanter Objekte wie Säulen, Tischen, Bäumen, Mittelstreifen von Straßen etc. geschehen. Im einfachsten Fall werden solche Objekte künstlich in der Umwelt gesetzt (Landmarken).

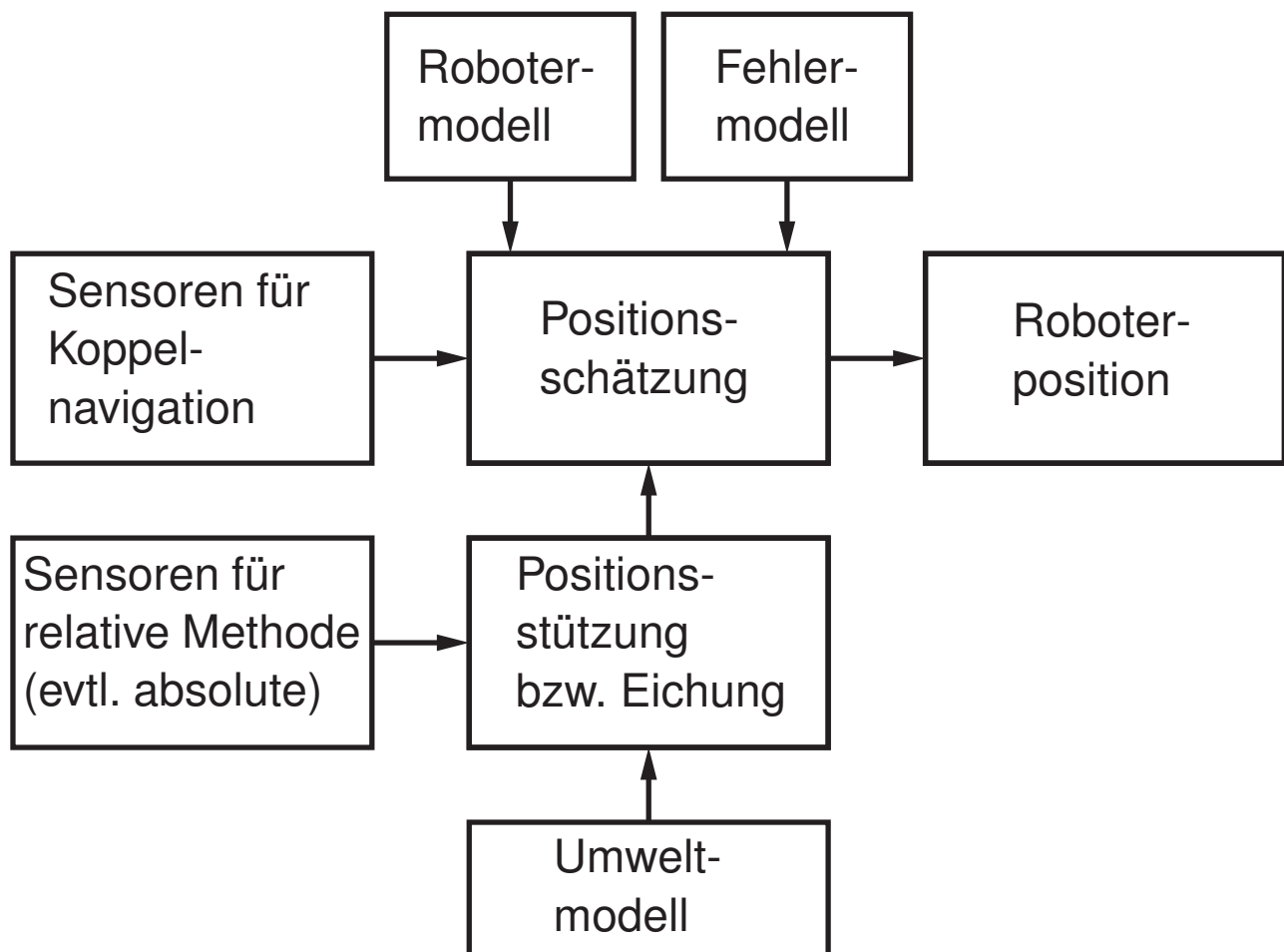
Alle drei Methoden weisen Nachteile auf. Die Koppelnavigation summiert Positionsfehler bei der Integration auf, was nach einer längeren Fahrtzeit zu einer sehr ungenauen Positionsbestimmung führt. Die Positionsbestimmung anhand künstlicher Landmarken ist dagegen sehr genau, aufgrund der Strukturierung der Roboterumwelt aber auch sehr aufwendig. Letzteres gilt auch für die Orientierung anhand natürlicher Umweltobjekte, die in ein entsprechend erstelltes Umweltmodell eingetragen werden.

Nachfolgendes Bild illustriert die drei Möglichkeiten der Positionsbestimmung.



Bei modernen Systemen kombiniert man die Koppelnavigation mit der relativen Methode. Letztere dient zum einen dazu, die von der Koppelnavigation geschätzte Lage zu stützen, und zum anderen die Koppelnavigation zu

bestimmten Zeitpunkten bzw. an bestimmten Orten zu eichen. Prinzipiell kann das auch anhand von Landmarken geschehen. Nachfolgendes Bild zeigt die Struktur einer solchen Positionsbestimmung.

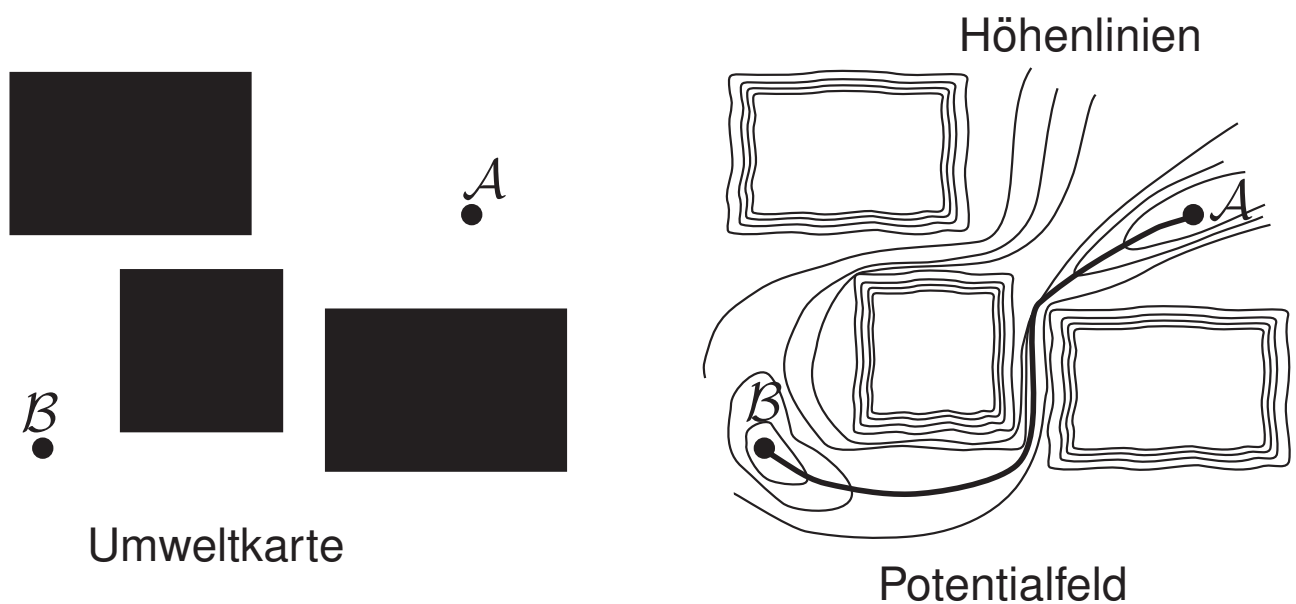


3. Bahnplanung

Bei der Bahnplanung ist, wie schon beschrieben, zwischen flächenabdeckenden Aufgaben und Aufgaben, bei denen der Roboter von \mathcal{A} nach \mathcal{B} fahren soll, zu unterscheiden. Beide Aufgaben sollen im Allgemeinen in möglichst kurzer Zeit gelöst werden.

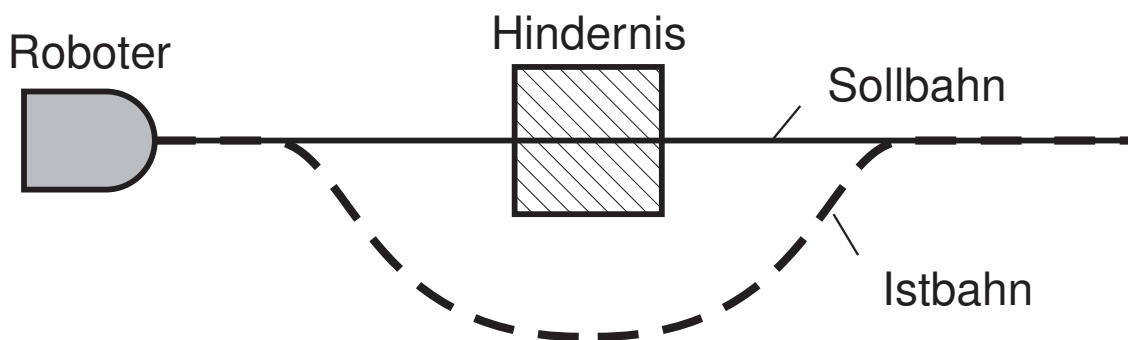
Für Fahraufgaben, die von \mathcal{A} nach \mathcal{B} führen, ist die Potentialfeldmethode eine der Standardmethoden. In Abschnitt 2.4 wird sie noch ausführlich behandelt werden. Ihr Prinzip ist das Folgende:

Es wird ein künstliches Potentialfeld erzeugt, bei dem der Startpunkt \mathcal{A} auf einem Berg und der Zielpunkt \mathcal{B} in einem Tal – genauer dem globalen Minimum des Potentialfeldes – liegt. Der Roboter ist dann sinnbildlich eine Kugel die vom Startberg \mathcal{A} in das Zieltal \mathcal{B} rollt. Nachfolgendes Bild illustriert das Prinzip.

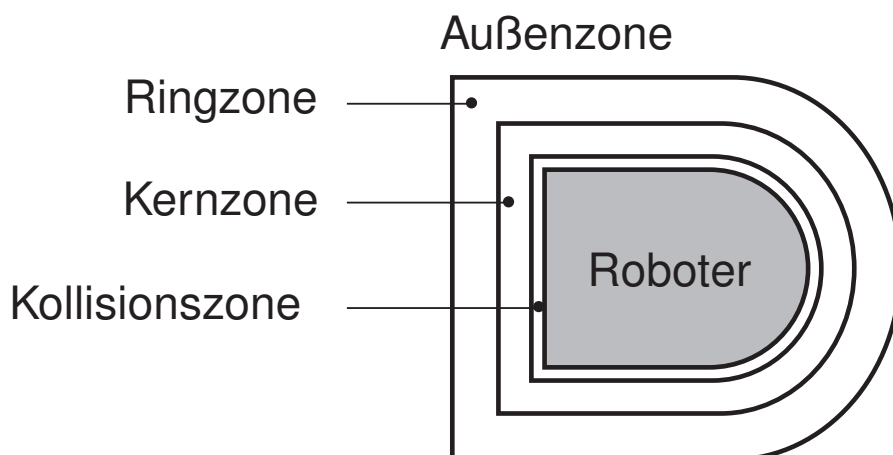


4. Hinderniserkennung und -vermeidung

Ein autonomer mobiler Roboter verfügt über ein System zur Erkennung von Hindernissen, die nicht in seinem Umweltmodell verzeichnet sind. Des Weiteren kann er diese Hindernisse umfahren und wieder auf seine vorgeplante Sollbahn zurückkehren. Das Bild illustriert dies.



Eine der möglichen Methoden¹, eine Hindernisvermeidung zu realisieren, ist die Folgende. Die direkte Umgebung des Roboters wird in vier Zonen unterteilt:

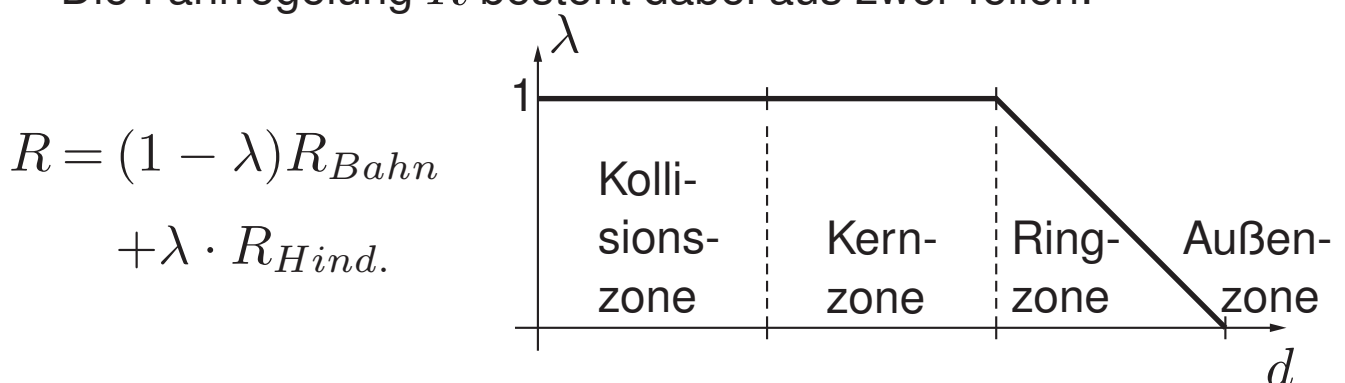


¹ von Puttkamer, E. et al.: Lokale Kollisionsvermeidung durch Ansteuerung der maximalen Hindernisdistanz für einen autonomen mobilen Roboter. 9. Fachgespräch des SFB 331 - Autonome mobile Systeme, München, 1993, S.227-238.

Mittels Ultraschallsensoren, die rundherum am Roboter installiert werden, wird der Abstand eines Hindernisses vom Roboter gemessen. Je nachdem in welcher Zone das Hindernis dann liegt, tritt folgende Situation auf:

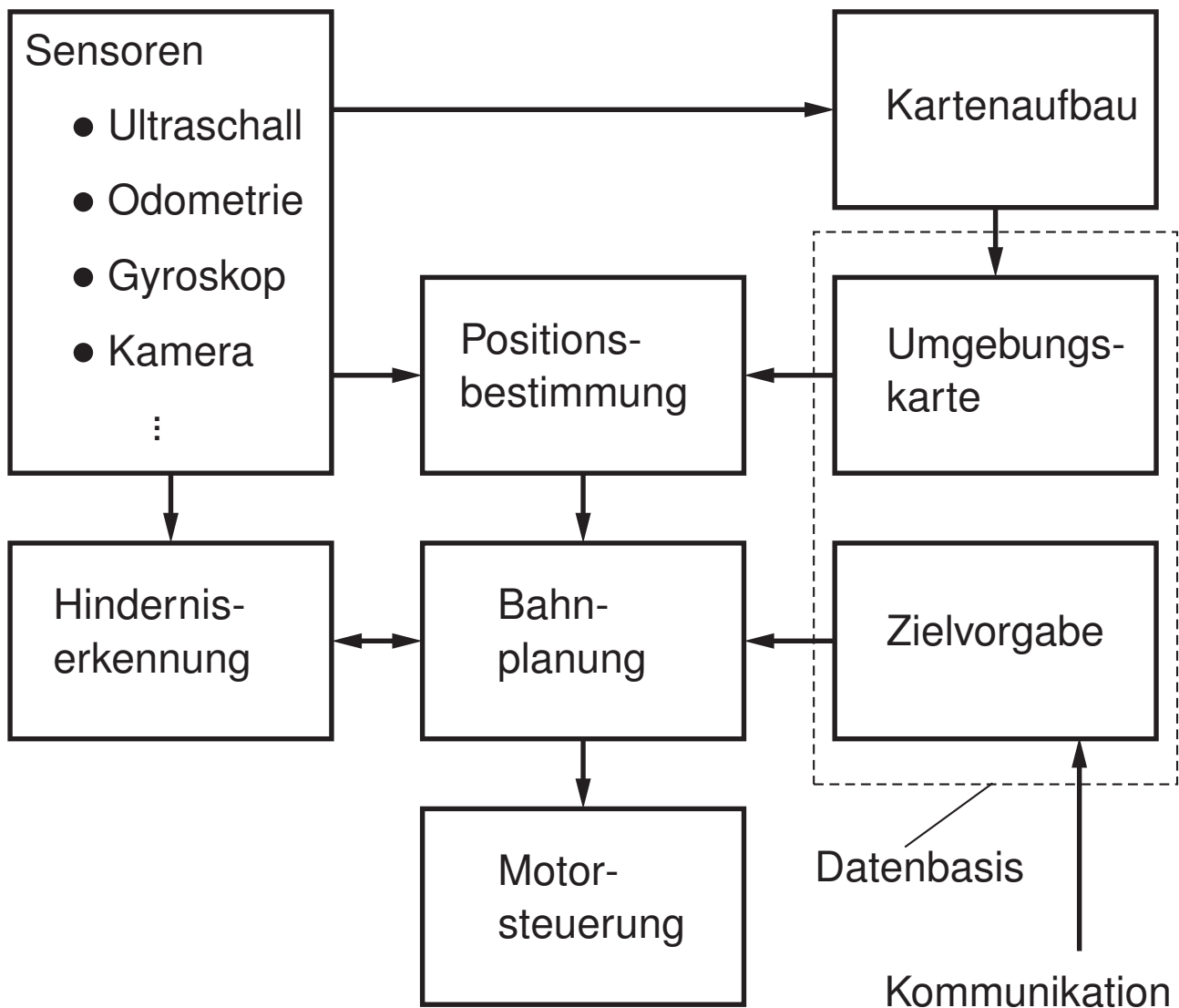
1. Hindernis in der Kollisionszone → Roboter hält an.
2. Hindernis in der Kernzone → Roboter wird über eine reine Hindernisvermeidungsstrategie geregelt.
3. Hindernis in der Ringzone → Roboter wird sowohl über die Hindernisvermeidungsstrategie als auch über die Bahnfolgeregelung geregelt.
4. Hindernis in der Außenzone oder kein Hindernis vorhanden → Roboter wird über die Bahnfolgeregelung geregelt.

Die Fahrregelung R besteht dabei aus zwei Teilen:



Dabei ist R_{Bahn} der Bahnregler und $R_{Hind.}$ die Hindernisvermeidungsstrategie bzw. der -regler. λ ist der Hindernisannteil der Regelung und wie dargestellt vom Abstand d abhängig. Stellgrößen sind Robotergeschwindigkeit und Roboterwinkelgeschwindigkeit.

Neben den beschriebenen Grundmodulen werden noch eine Motorsteuerung, ein Zielvorgabemodul und Sensoren wie Gyroskope, Kameras, Ultraschallsensoren, Odometer etc. benötigt. Das Bild zeigt den Gesamtaufbau und das Zusammenspiel der einzelnen Module.



In den nachfolgenden Abschnitten wird detaillierter auf die Sensoren, den Kartenaufbau und die Bahnplanung eingegangen.

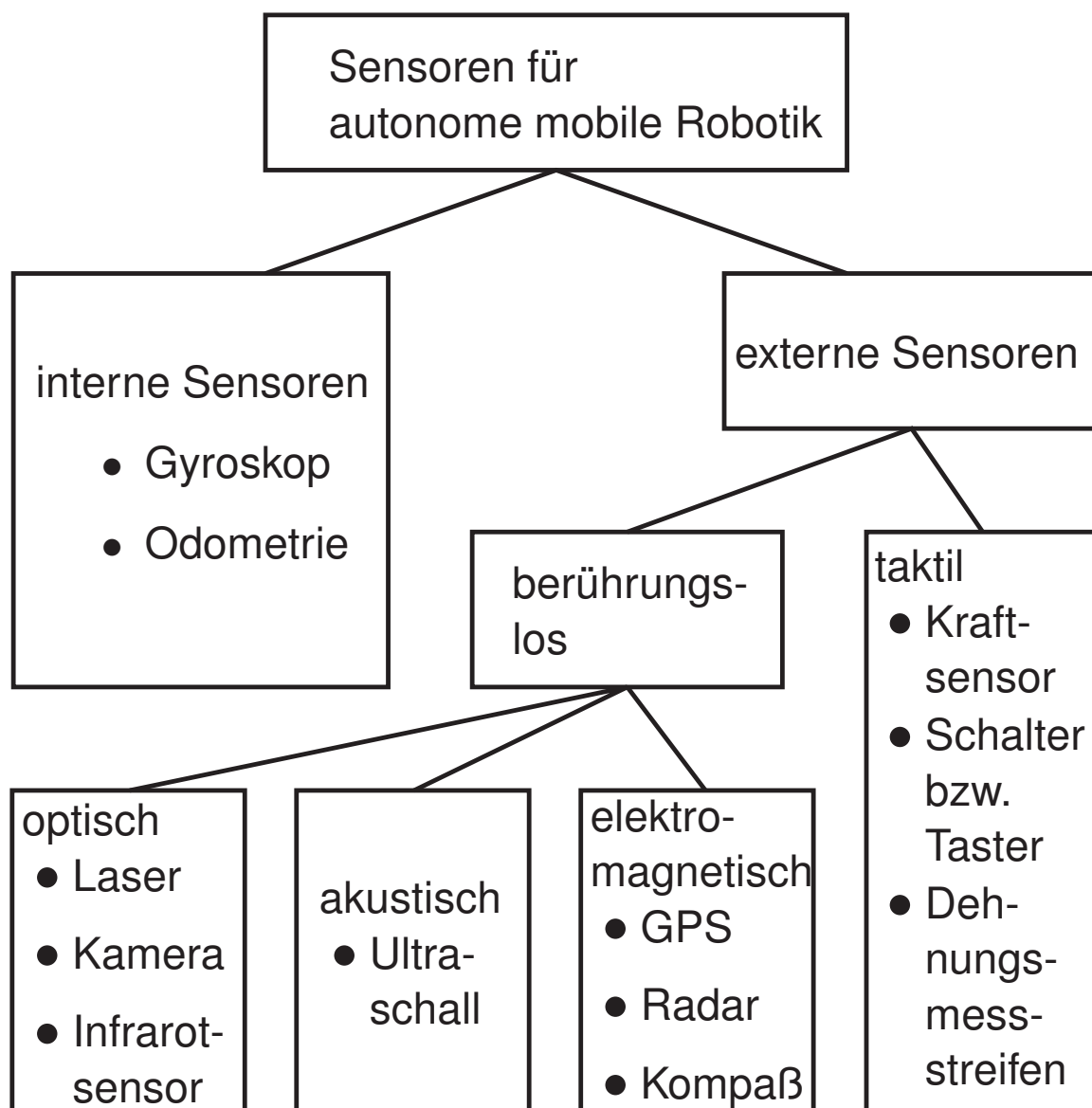
ABSCHNITT 2.2

Sensoren

Zentrale Bedeutung besitzt die Sensorik bei autonomen mobilen Robotern. Ohne Sensoren, die eine Orientierung in der Umwelt erlauben, ist keine autonome Navigation des Roboters möglich. Oft werden sogar verschiedene Sensoren zu

- Multisensorsystemen

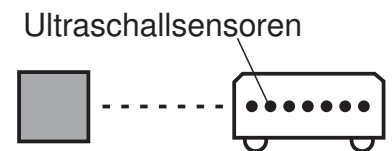
kombiniert, um ihre unterschiedlichen Möglichkeiten für die Positionsbestimmung und die Navigation zu nutzen. Eine Klassifikation kann wie folgt vorgenommen werden:



Mit den verschiedenen Sensortypen sollen im Wesentlichen fünf Aufgaben erledigt werden:

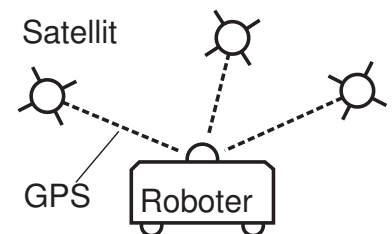
1. Entfernungsmessung

- Ultraschall
- Laser
- Stereokamera
- Radar



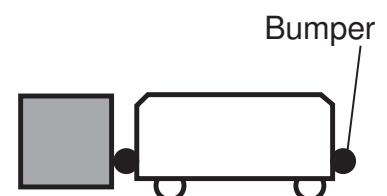
2. Positions- und Richtungsbestimmung

- Gyroskop
- Odometrie
- GPS
- Kompaß



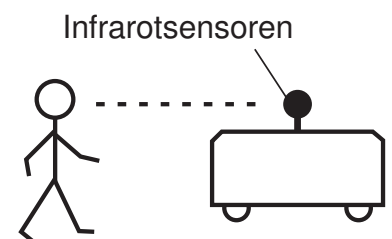
3. taktile Objektdetektion

- Kraftsensoren
- Schalter und Taster
- Dehnungsmessstreifen



4. berührungslose Objektdetektion

- Infrarotsensoren
- unter 1. genannte Sensoren

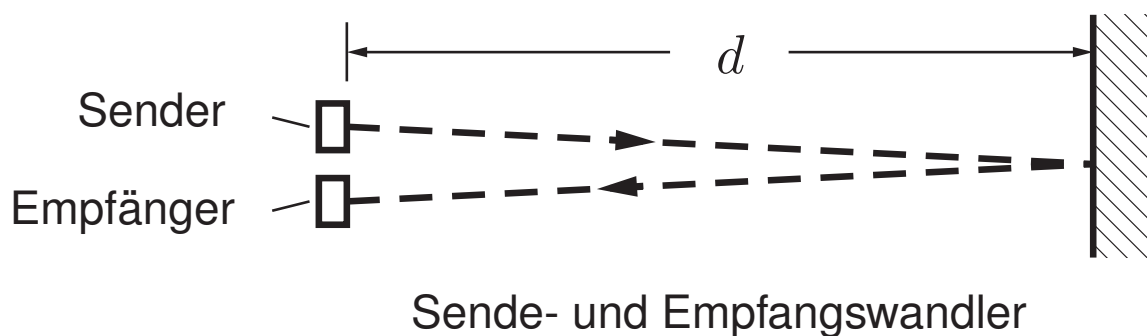


5. Objekt- und Situationserkennung

- Kamera

Mittels dieser Basismessdaten sind dann weitergehende Aufgaben lösbar, z. B. Umweltmodellierung, Positionsbestimmung, Hindernisumgehung, etc.

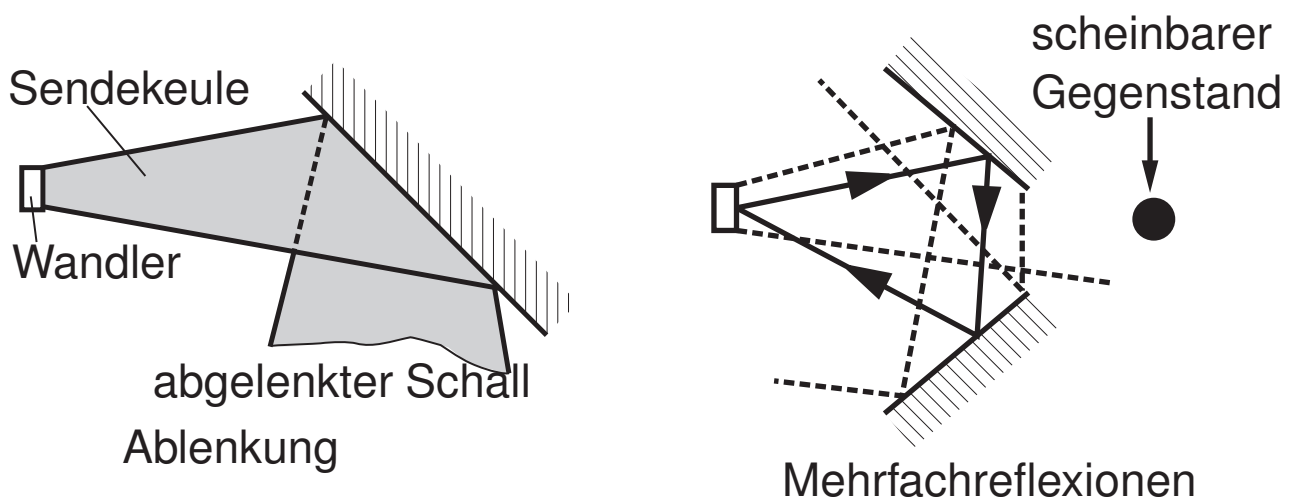
Ultraschall-Sensoren werden zur Entfernungsmessung verwendet. In der Regel ist der Roboter mit einer größeren Anzahl von Ultraschall-Sensoren bestückt – z.B. ringsum. So kann die nähere Umgebung relativ lückenlos erfaßt werden. Dabei sendet ein piezoelektrischer Wandler Ultraschallwellen aus. Diese werden an Gegenständen der Umgebung reflektiert und treffen auf einen Empfangswandler.



Für Abstand d , Laufzeit t und Schallgeschwindigkeit v_c gilt

$$d = \frac{v_c \cdot t}{2}.$$

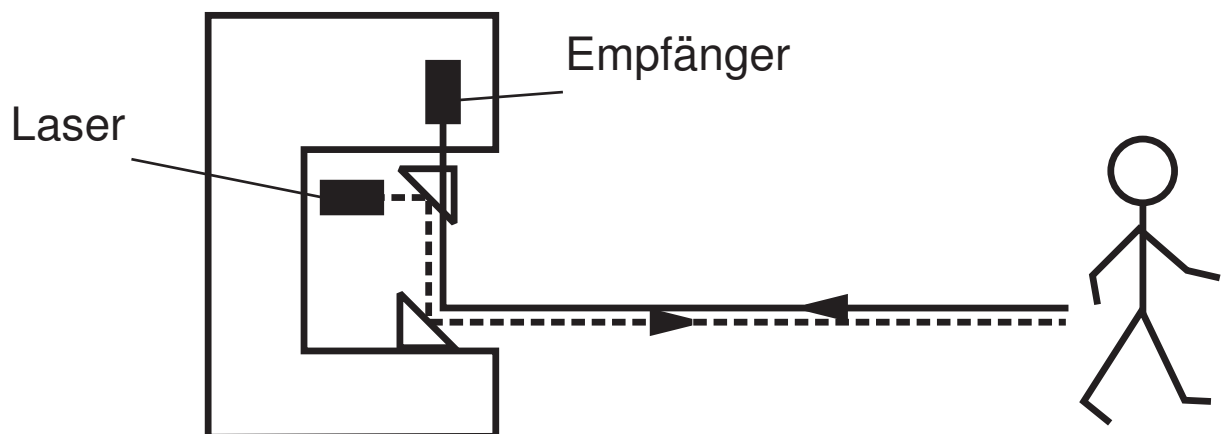
Die Entfernungsbestimmung ist aufgrund von Ablenkungen und Mehrfachreflexionen mit Schwierigkeiten behaftet. Das illustrieren nachfolgende Situationen:



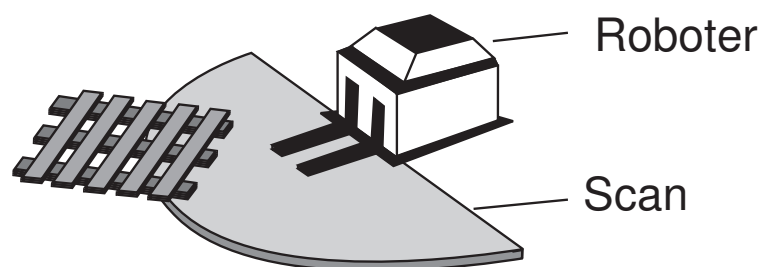
Laser-Scanner werden ebenfalls für die Entfernungsmessung verwendet. Dabei werden Laserimpulse gesendet, die, sobald sie auf ein Objekt treffen, reflektiert werden. Der reflektierte Laserimpuls wird von einem Empfänger registriert, und die Objektdistanz d kann aus seiner Laufzeit t zu

$$d = \frac{t \cdot c}{2}$$

bestimmt werden. Nachstehendes Bild zeigt das Messprinzip.



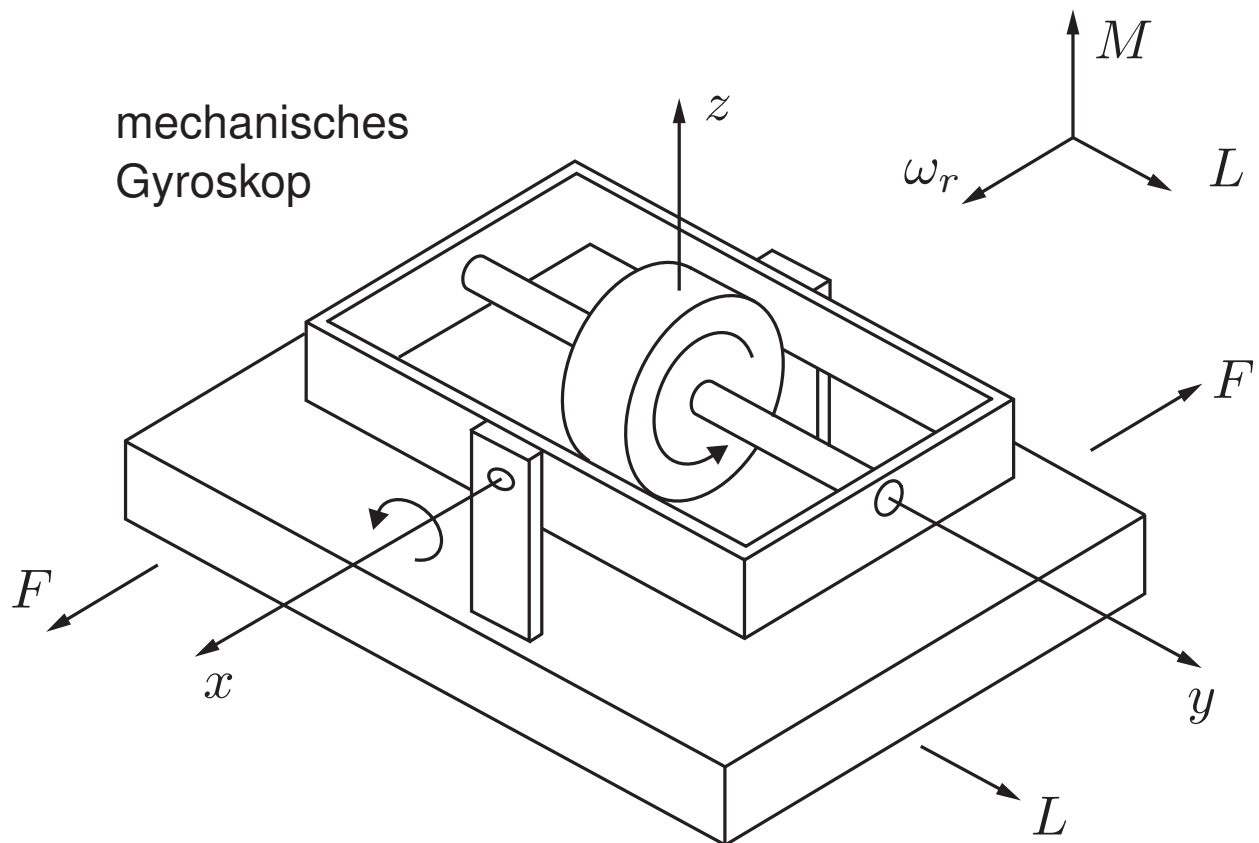
Mittels eines rotierenden Spiegels kann die Umgebung in einer bestimmten Höhe und einem Winkelbereich von $0 \dots 180^\circ$ gescannt werden – wie unten illustriert.



Gyroskope dienen zur Bestimmung von Beschleunigungen. Durch zweimalige Integration der gemessenen Beschleunigungen kann die Orientierung des Gyroskops bestimmt werden. Es gibt verschiedene Ausführungen von Gyroskopen:

- mechanische Gyroskope,
- Glasfaser-Gyroskope,
- piezoelektrische Gyroskope und,
- Ringlasergyroskope (RLG).

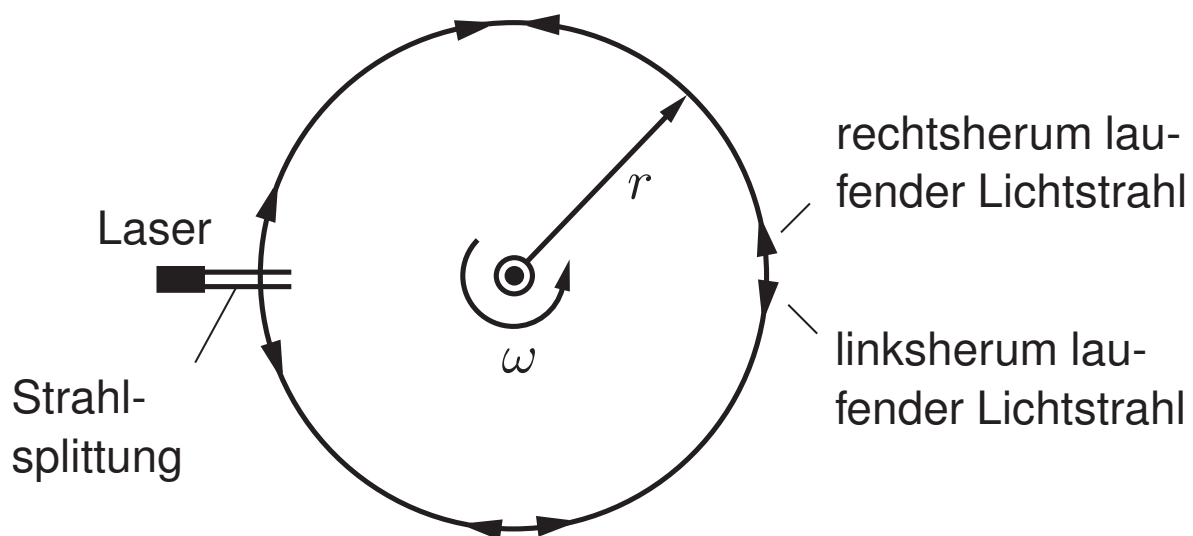
Ein mechanisches Gyroskop funktioniert nach folgendem Prinzip. Eine rotierende achssymmetrische Masse ist wie im Bild gezeigt in einem Käfig drehbar gelagert.



Der Kreisel versucht seine Lage, d. h. seinen Drehimpuls L , im kräftefreien Fall beizubehalten. Verändert man seine Position durch Kippen um die z -Achse mittels einer Kraft F bzw. eines Momentes M , so reagiert er mit einer Drehbewegung mit ω_r um die x -Achse. Diese Drehung kann man messen und so auf das beschleunigende Moment M rückschließen.

Gyroskope können auch auf Basis des Sagnac-Effekts konstruiert werden. FOG's (fibre optic gyroscopes) sind solche Gyroskope.

Der Sagnac-Effekt läßt sich wie folgt erklären: Man speist zwei kohärente Lichtstrahlen (z. B. Laserstrahlen) gegenläufig in eine geschlossene Lichtkreisbahn ein – wie unten gezeigt.



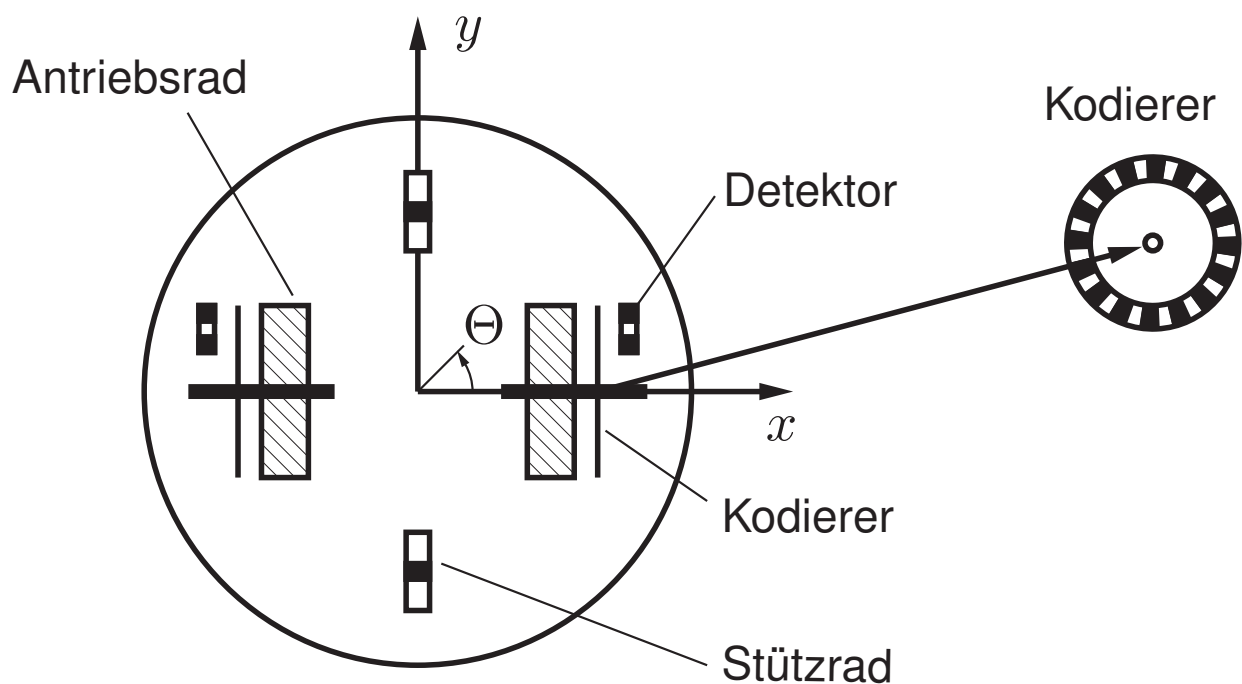
Die Lichtkreisbahn wird z. B. durch Glasfaserkabel realisiert, die zu n Kreisen aufgewickelt werden.

Versetzt man dann die gesamte Anordnung in Rotation mit ω , so ist der Weg für den einen Laserstrahl kürzer, für den anderen länger. Entsprechend ergibt sich eine Phasendifferenz $\Delta\phi$ zwischen beiden Lichtstrahlen nach Durchlaufen der Kreisbahn:

$$\Delta\phi = \frac{8\pi^2 r^2 n}{c\lambda} \cdot \omega.$$

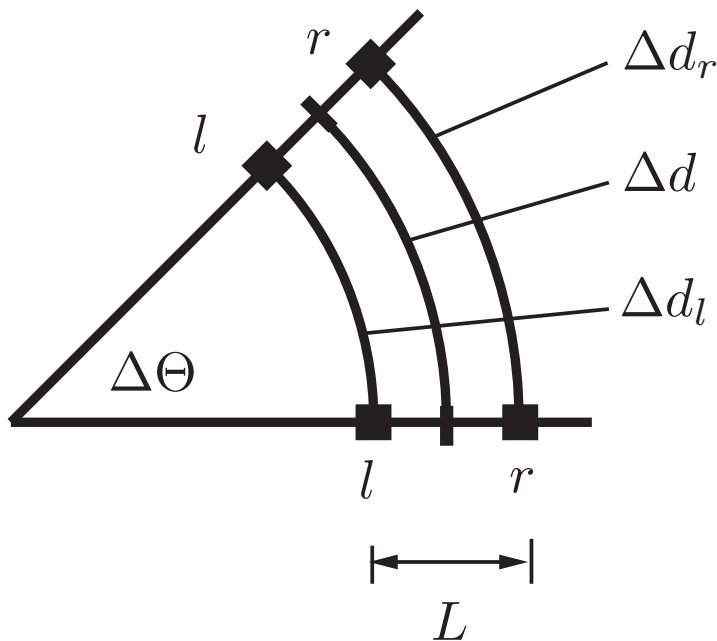
Dabei ist λ die Wellenlänge des eingespeisten Lichts. FOG's ermöglichen eine relativ einfache und genaue Messung von Winkelbeschleunigungen.

Odometrie, auch Koppelnavigation genannt, dient ebenfalls der Positionsbestimmung. Hier werden über inkrementale Kodiereinrichtungen die Geschwindigkeiten der Räder gemessen. Mit einem zweiten, um 90° phasenverschobenen Inkrementalkodierer läßt sich auch die Durchlaufrichtung bestimmen. Durch Integration der Geschwindigkeiten erhält man dann den zurückgelegten Weg jedes Rades. Daraus kann wiederum auf die Position des Roboters zurückgeschlossen werden. Nachfolgendes Bild illustriert die Messung am Beispiel eines über zwei Räder angetriebenen Roboters. Jedes Rad besitzt dabei einen eigenen Antrieb. Die Stützräder sind frei beweglich.



Die Position des Roboters ist durch die x - und y -Koordinaten gegeben. Seine Orientierung durch den Winkel Θ .

Fährt der Roboter, so misst man die Radgeschwindigkeiten v_l und v_r des linken und des rechten Rades. Dann bestimmt man die zwischen zwei Zeitpunkten t_{i-1} und t_i zurückgelegten Wege Δd_l und Δd_r des linken und des rechten Rades. Das folgende Bild illustriert dies. Die mittlere Positionsänderung ist Δd .



$$\Delta d = \frac{1}{2}(\Delta d_r + \Delta d_l)$$

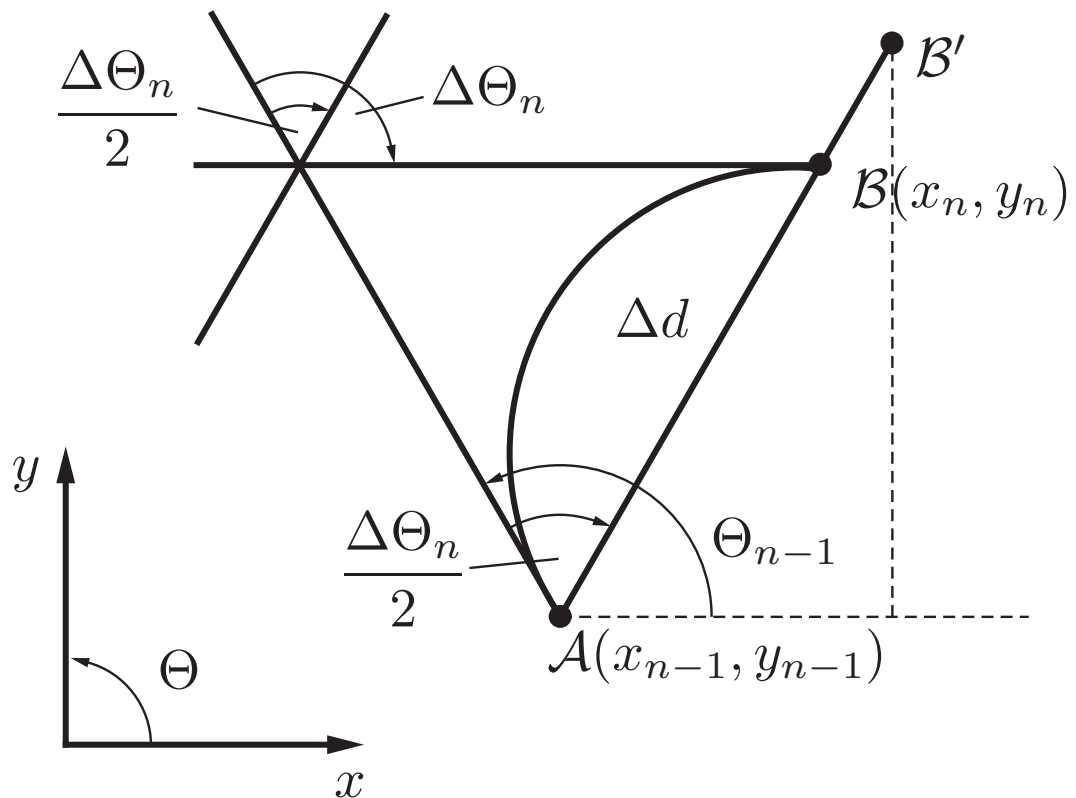
$$\Delta \Theta = \frac{\Delta d_r - \Delta d_l}{L}$$

Mittels der Inkremente Δd und $\Delta \Theta$ kann die Position (x, y) und die Orientierung Θ des Roboters bestimmt werden. Für den Winkel Θ gilt

$$\Theta_n = \Theta_{n-1} + \Delta \Theta_n.$$

Der Index n zählt dabei die Inkremente. Die Position (x, y) des Roboters bzw. die Positionsänderung $(\Delta x, \Delta y)$ kann im Gegensatz zur Orientierung nicht exakt aus den Inkrementen Δd und $\Delta \Theta$ bestimmt werden. Denn man kennt den Fahrweg während der inkrementellen Messung nicht.

Daher schätzt man die Position, indem man als Fahrweg einen Kreisbogen annimmt. Siehe Bild.



Der Roboter ist von \mathcal{A} nach \mathcal{B} gefahren. Als Näherung für seine Position \mathcal{B} nutzt man \mathcal{B}' , wobei für die Strecke $\overline{\mathcal{A}\mathcal{B}'} = \Delta d$ gilt. Dann gilt

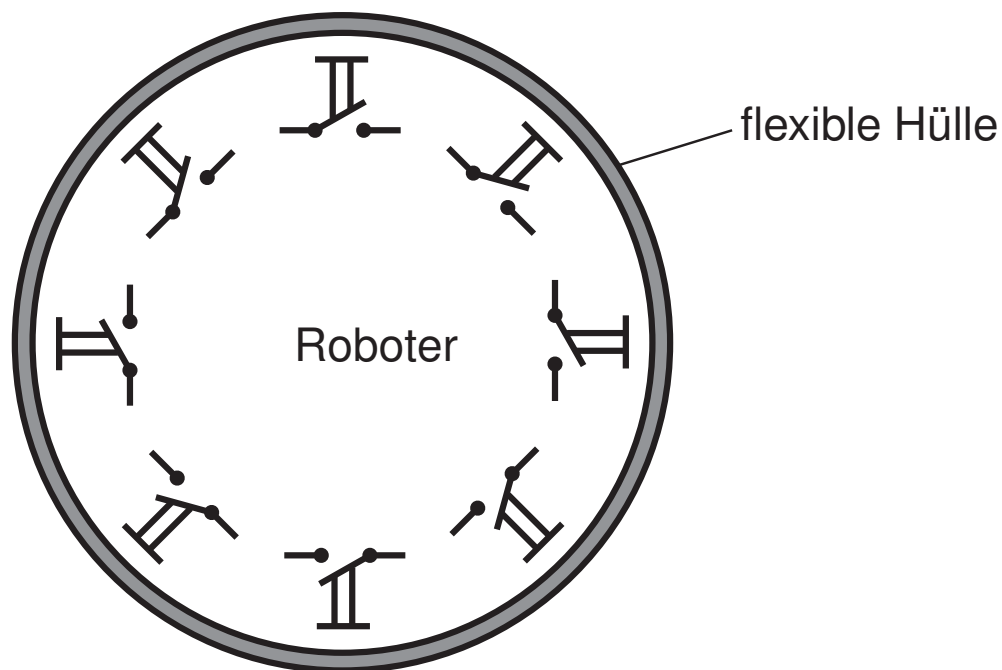
$$x_n \approx x_{n-1} + \Delta d \cos\left(\Theta_{n-1} + \frac{\Delta\Theta_n}{2}\right),$$

$$y_n \approx y_{n-1} + \Delta d \sin\left(\Theta_{n-1} + \frac{\Delta\Theta_n}{2}\right).$$

Man beachte: Im obigen Beispiel ist $\Delta\Theta$ negativ !

Kraftsensoren und Taster werden für die Kontakterkennung genutzt. Wird über diese Sensoren eine Kollision oder ein Kontakt des Roboters mit einem Objekt festgestellt, so führt dies entweder zum Stillstand des Roboters oder zu einem Rückzugsmanöver.

Im Allgemeinen werden mehrere Sensoren in Form eines Gürtels rund um den Roboter angeordnet:



Bumper obiger Art dienen also nicht der Kollisionsvermeidung, sondern sie bilden eine Sicherheitseinrichtung, die Schäden beim Fahren gegen ein Objekt vermeiden soll.

Die Kollisionsvermeidung selbst wird in der Regel mit Hilfe von Kameras, Ultraschallsensoren oder Laserscannern durchgeführt.

Jeder Sensor besitzt Vor- und Nachteile. Eine autonome Navigation auf Basis eines Sensortyps ist in den meisten Anwendungsfällen nicht möglich. Daher kombiniert man verschiedene Typen, wie z. B.

- Ultraschallsensoren,
- Gyroskope,
- Kameras,
- Odometer.

Die aufgenommenen Daten ergänzen sich und erlauben so eine möglichst sichere

- Positionsbestimmung und
- Hinderniserkennung.

Diesen Schritt bezeichnet man auch als Sensorfusion. Sind Position und eventuelle Hindernisse bestimmt, wird mittels eines Navigationsalgorithmus ein vorgegebenes Ziel angesteuert. Die Positionsbestimmung stützt sich dabei zum einen auf die aktuellen Sensordaten und zum anderen auf eine Umgebungskarte. Die Umgebungskarte kann dabei fest vorgegeben sein. Sie kann aber auch – wenn der Roboter geeignet ausgelegt ist – selbstlernend durch den Roboter aufgebaut werden.

ABSCHNITT 2.3

Umweltkarten und Kartenaufbau

Damit ein mobiler Roboter sich in einer vorgegebenen Umwelt autonom bewegen kann, benötigt er wie schon erwähnt eine Umweltkarte, auch Weltmodell genannt. Um zu einem solchen Weltmodell zu gelangen, existieren drei wesentliche Möglichkeiten:

1. Vorabmodellierung der Umgebung:

Hier wird off-line ein Modell der Umwelt ermittelt, d. h. eine Karte, und dieses als Datei im Rechner des Roboters abgelegt.

2. Selbständige Konstruktion:

Bei diesem Verfahren erkundet der Roboter mittels einer geeigneten Explorationsstrategie seine Umgebung selbst und zeichnet dabei on-line eine Karte auf Basis seiner Sensordaten.

3. Interaktive Erkundung (Teach-in):

Beim Teach-in wird der Roboter durch einen Menschen zu bestimmten Umgebungspunkten geführt, an denen er seine Umgebung mittels seiner Sensoren messtechnisch erfaßt und speichert.

Die „Selbständige Konstruktion“ ist das aufwendigste aber auch leistungsfähigste dieser Verfahren. Es ermöglicht u. a. eine adaptive Anpassung des Roboters an seine Umwelt, da er Veränderungen in ihr erkennen und durch eine entsprechende Änderung des Umweltmodells behandeln kann.

Als mögliche Formen von Weltmodellen sind drei Abstraktionsstufen zu unterscheiden.

1. Semantische Weltmodelle:

Bei semantischen Modellen werden die Umgebung, ihre Struktur und die in ihr enthaltenen Objekte verbal beschrieben.

Auf Basis semantischer Modelle erfolgt i. Allg. die Aufgabenplanung.

Beispiel:

„Raum 1 ist mit Raum 2 verbunden.“

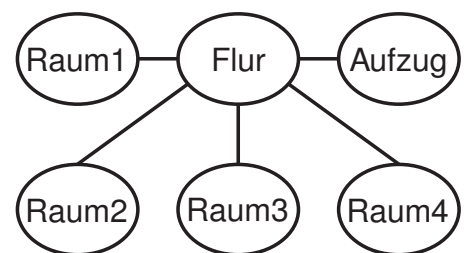
„In Raum 2 steht Paul's Tisch.“

2. Topologische Weltmodelle:

Topologische Modelle geben insbesondere die Raumstruktur wieder. Einzelne Details und Abmessungen enthalten sie nicht.

Sie dienen der globalen Wegeplanung.

Beispiel:

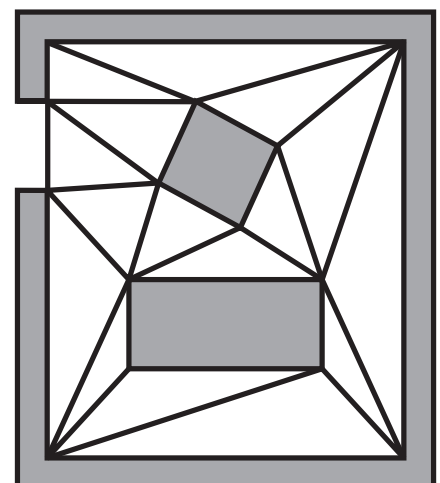


3. Geometrische Modelle:

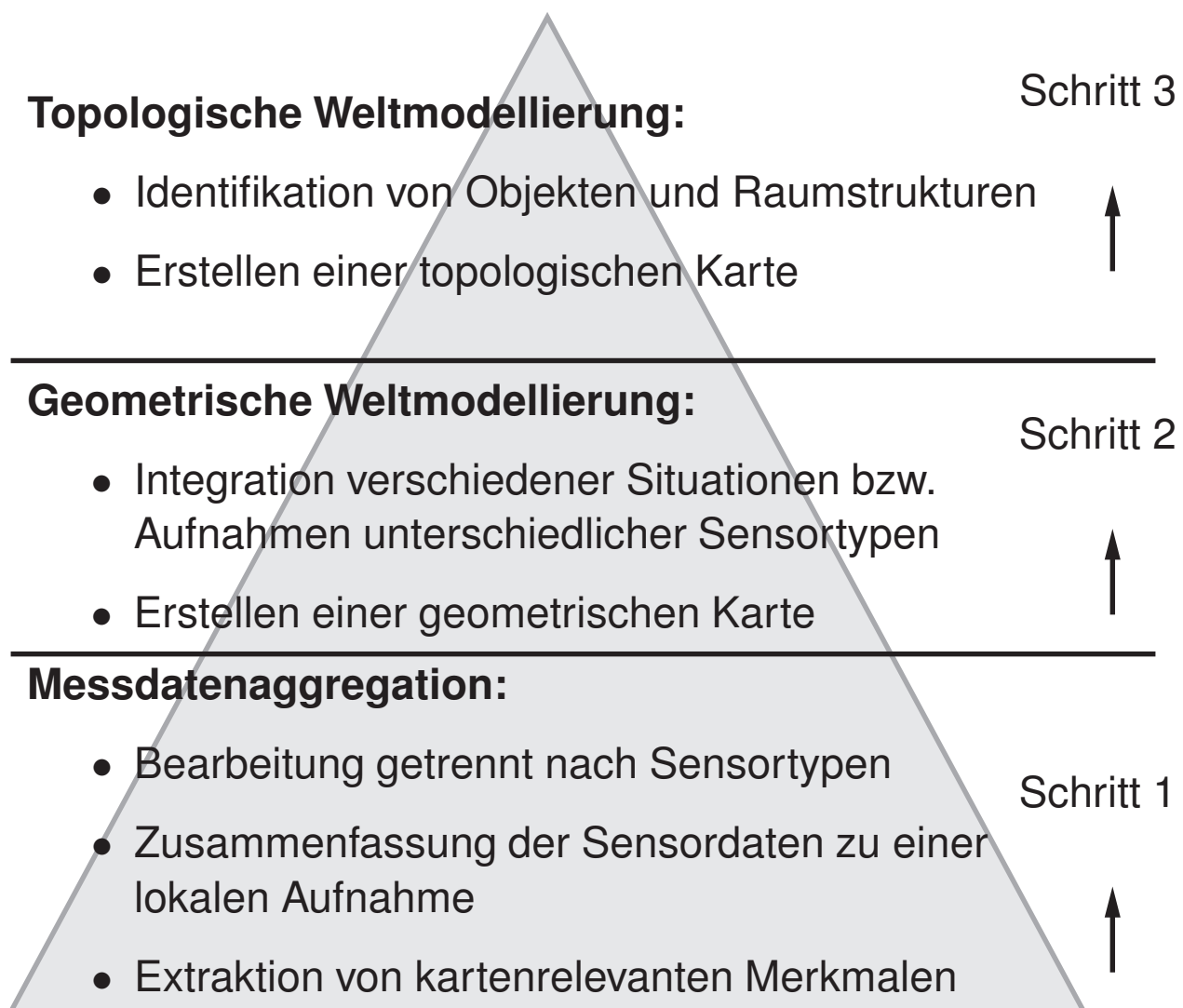
Geometrische Modelle enthalten in abstrahierter Form die konkreten Abmessungen der Umgebung und der in ihr enthaltenen Objekte. Z. B. als polygonales oder als Zellen-Modell.

Diese Modelle dienen der lokalen Wegeplanung.

Beispiel:



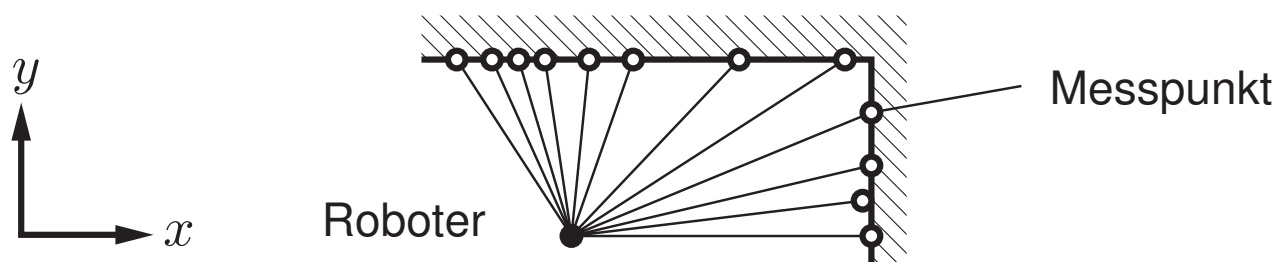
Die selbständige Konstruktion eines Weltmodells durch den Roboter erfolgt anhand von Sensoraufnahmen in einem dreistufigen Abstraktionsprozeß. In einem ersten Schritt werden die Messdaten der verschiedenen Sensoren gesammelt und zu einer lokalen Aufnahme zusammengefaßt. Aus dieser werden Merkmale, wie Kanten von Hindernissen etc. extrahiert. Im zweiten Schritt wird auf Basis dieser Merkmale und ihrer Abmessungen das geometrische Modell aufgebaut. Aus diesem wird wiederum im dritten Schritt das topologische Weltmodell erstellt. Nachfolgendes Bild illustriert dieses Schema.



Im Weiteren werden einige wichtige Elemente der obigen drei Schritte skizziert bzw. Realisierungsmöglichkeiten beschrieben. Das gesamte Gebiet, wie es die voranstehende Grafik zeigt, ist jedoch wesentlich komplexer und umfangreicher. Diesbezüglich wird auf die angegebene Literatur verwiesen.

Messdatenaggregation

Beim Messen von Entfernungen durch den Roboter – z. B. mittels Laser oder Ultraschall – erhält man eine Reihe von Messpunkten i , deren kartesische Koordinaten (x_i, y_i) man errechnen kann:



Diese Punkte können auch aus der Aggregation der Messdaten verschiedener Sensoren resultieren. Aus ihnen werden nun kartenrelevante Merkmale extrahiert – z. B. Linien. Hierfür gibt es eine Reihe von Verfahren. Eines der einfacheren, das

- Line Tracking,

soll hier betrachtet werden.

Beim Line Tracking wird während der Aufnahme der Messpunkte (x_i, y_i) eine

Gerade

$$ax + by = c$$

durch die Messpunkte gelegt. Dies geschieht bei jedem neuen Messpunkt

$$(x_i, y_i),$$

wobei dieser und die $n - 1$ vorherigen Messpunkte Basis der Berechnung sind.

Für die Koeffizienten gilt:

$$a = \sum_{i=1}^n x_i \sum_{i=1}^n y_i^2 - \sum_{i=1}^n y_i \sum_{i=1}^n x_i y_i,$$

$$b = \sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i,$$

$$c = \sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n x_i y_i \right)^2.$$

Dabei geht man davon aus, dass die Punkte (x_i, y_i) solange zu ein und derselben Geraden gehören bis der Fehler

$$\epsilon = \frac{c \cdot (n \cdot c - a \cdot \sum_{i=1}^n x_i - b \cdot \sum_{i=1}^n y_i)}{n \cdot (a^2 + b^2)}$$

einen vorgegebenen Wert überschreitet.

Geometrische Weltmodellierung

In diesem Schritt wird aus den abstrahierten Messdaten – z. B. Linien – eine geometrische Weltkarte aufgebaut. Dies geschieht oftmals, indem man lokale Aufnahmen verschiedener Sensortypen aus unterschiedlichen Positionen fusioniert. Aus ihnen wird dann ein globales Weltmodell aufgebaut bzw. die Daten werden in ein bestehendes Weltmodell integriert.

Eine Möglichkeit hierfür ist das

- Line Matching.

Dabei wird ein Weltmodell als Menge geometrischer Merkmale m_i aufgefaßt:

$$\mathcal{W} = \{m_1, m_2, \dots, m_n\}.$$

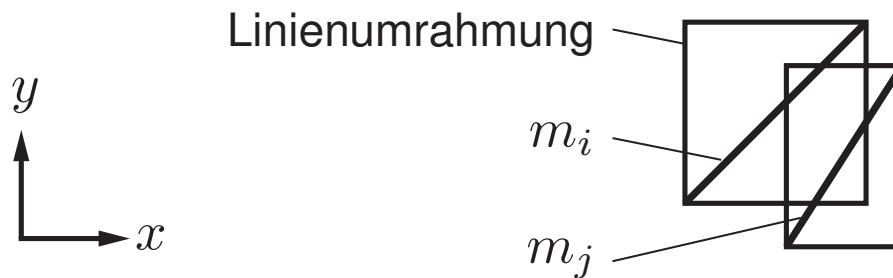
Die Merkmale sind z. B. Linien. Wird nun ein neues Merkmal m_j bestimmt, so stellt sich die Frage:

Läßt sich m_j mit einem $m_i \in \mathcal{W}$ zusammenfassen oder ist es ein neues Merkmal, das zu \mathcal{W} hinzuzufügen ist?

Diese Frage läßt sich mittels des Line Matching klären. Die Zusammenfassung des neuen Merkmals m_j mit einem Merkmal $m_i \in \mathcal{W}$ ist z. B. dann relevant, wenn zwei unterschiedliche Sensortypen dasselbe Objekt vermessen.

Eine mögliche Ausprägung des Line Matching ist die Folgende:

1. Gegeben sei eine neue Linie m_j . Gesucht werden nun in \mathcal{W} Linien m_i , die eine Linienumrahmung gemäß nachfolgendem Bild besitzen, die mit der Linienumrahmung von m_j überlappt.

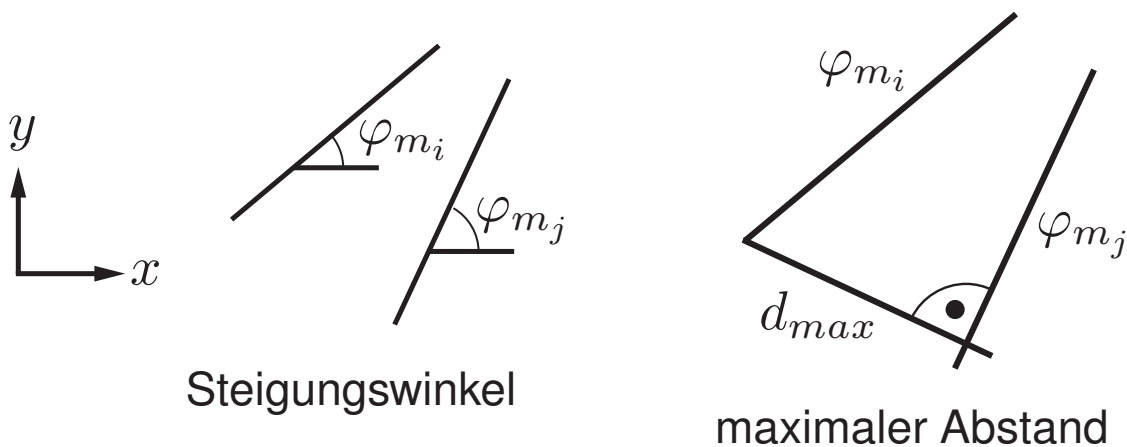


Denn nur wenn sich die Linienumrahmungen überlappen, ist es eventuell sinnvoll, sie zu fusionieren.

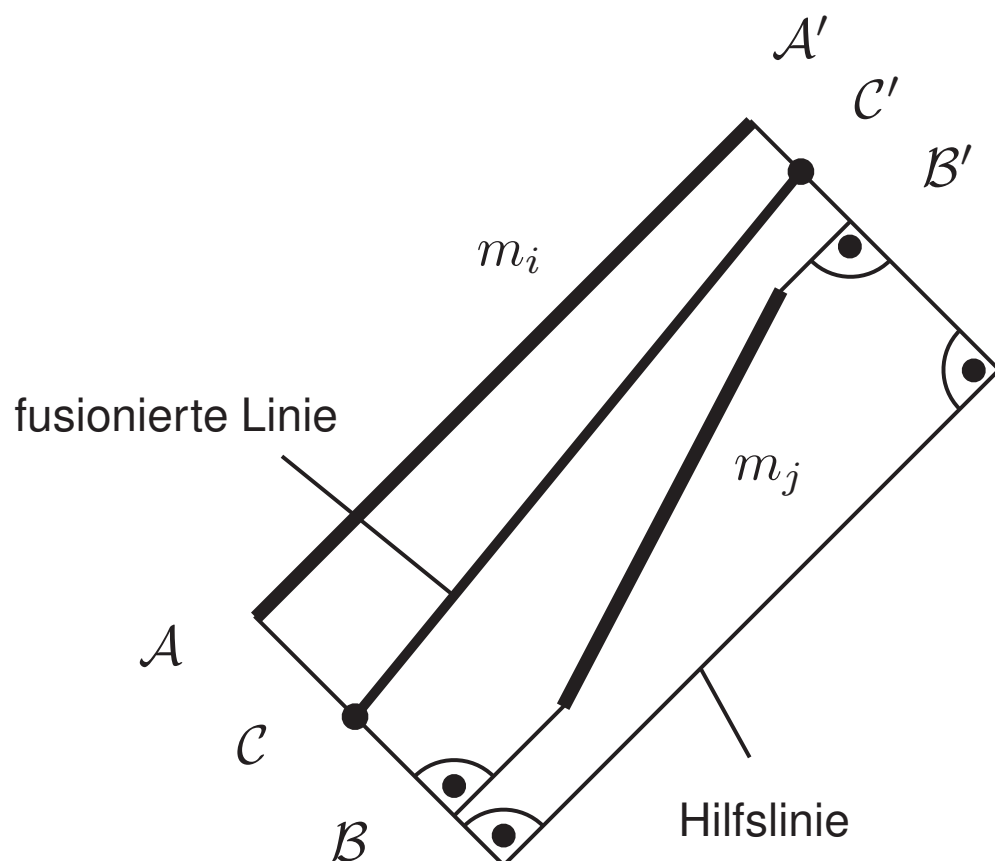
2. Kriterien für die Fusion sind Steigungswinkeldifferenz

$$\Delta\varphi = \varphi_{m_i} - \varphi_{m_j}$$

und maximaler Abstand d_{max} der Linien. Liegen sie in einem Toleranzintervall, so werden die Linien fusioniert.



3. Um die Linien zu fusionieren, wird eine Hilfslinie bestimmt, die parallel zur längeren der beiden Linien $\overline{AA'}$ verläuft und genauso lang ist wie diese:



Dann werden die Senkrechten zwischen der Hilfslinie und der längeren Linie bestimmt. Auf diese Senkrechten wird das Lot von jeweils einem der Eckpunkte der kürzeren Linie gefällt. So entstehen die Fußpunkte des Lots B bzw. B' . Nun werden die Strecken \overline{AB} bzw. $\overline{A'B'}$ halbiert: C und C' entstehen. Die Linie $\overline{CC'}$ bildet die fusionierte Linie.

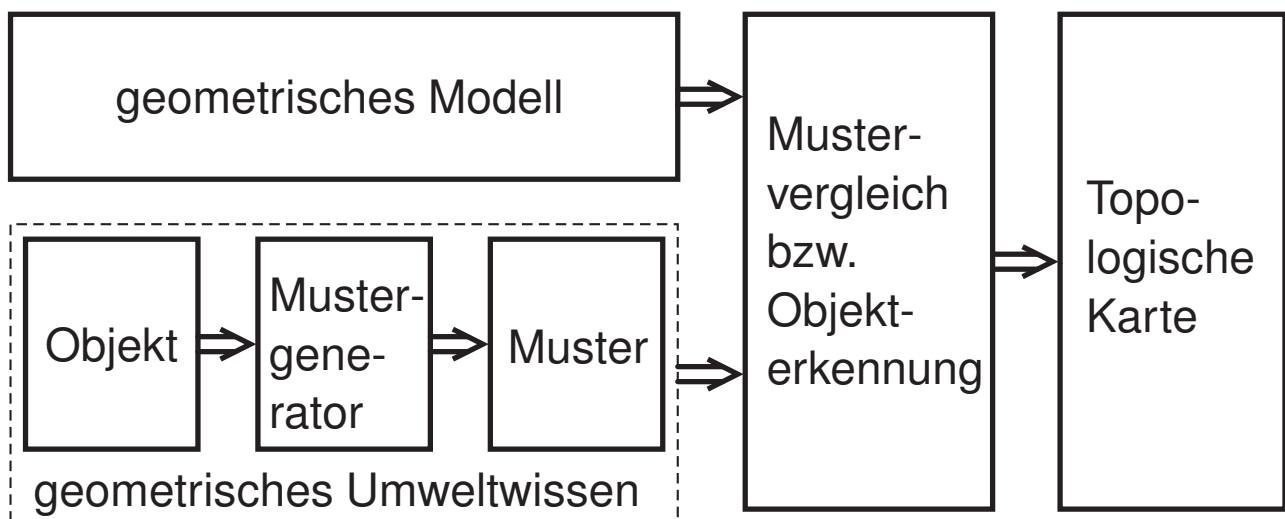
Mittels des Line Matching kann also sukzessiv ein Weltmodell $\mathcal{W} = \{m_1, \dots, m_n\}$ aufgebaut werden. Dieses kann jederzeit durch neu gemessene Linien m_j ergänzt werden.

Topologische Weltmodellierung

Bei der topologischen Weltmodellierung werden Objekte (wie Tische, Türen etc.) und Strukturen (Raumverbindungen, Sackgassen etc.) anhand des geometrischen Weltmodells identifiziert.

Hierbei werden die Geometriedaten einer lokal begrenzten Umgebung mit abgespeicherten Linienmodellen bzw. Mustern bestimmter Objekte (Wand, Tür, Stuhl etc.) verglichen und bei Übereinstimmung entsprechend identifiziert. Beispielsweise kann eine vier Meter lange gerade Linie nur eine Wand sein.

Nachfolgendes Bild illustriert diesen Aufbau der topologischen Karte:



Im Prinzip stellt die topologische Kartographie also eine Objekt- oder Mustererkennung dar.

Die globale Wegplanung erfolgt anhand der topologischen Karten, anhand derer Fahrfolgen wie

$$\begin{array}{ccccccc} \textit{Raum23} & \rightarrow & \textit{Flur22} & \rightarrow & \textit{Fahrstuhl2} & & \\ & & \rightarrow & \textit{Flur1} & \rightarrow & \textit{Raum12} & \end{array}$$

ablesbar sind, um von *Raum23* in den *Raum12* zu gelangen.

Die konkrete Wegplanung erfolgt allerdings immer anhand des geometrischen Modells. Hierfür sind verschiedene Verfahren entwickelt worden, von denen die bekanntesten im nachfolgenden Abschnitt beschrieben werden.

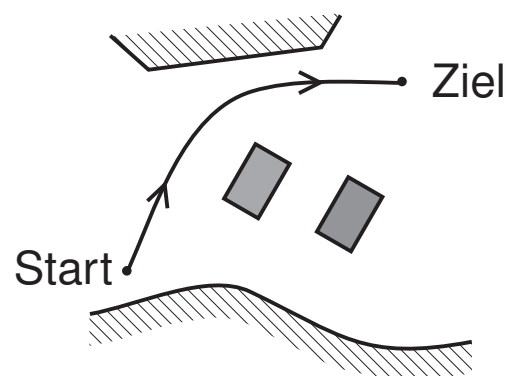
ABSCHNITT 2.4

Bahnplanung

Um von einem Ort zu einem vorgegebenen Zielort zu fahren, benötigt ein autonomer mobiler Roboter, wie schon erwähnt, ein Bahnplanungsmodell. Das gilt natürlich auch für den Fall der Flächenabdeckung.

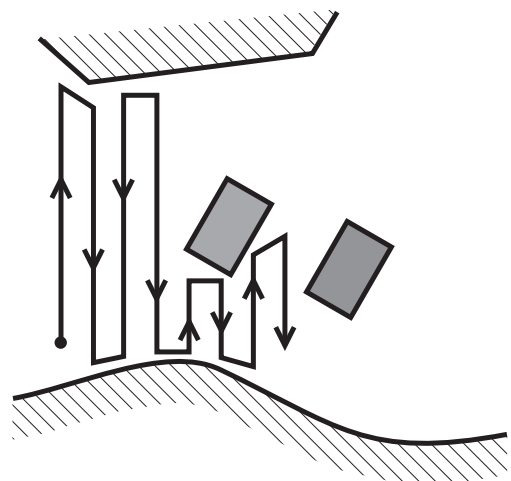
Im Folgenden werden zuerst die Bahnplanungsverfahren betrachtet, bei denen der Roboter einen Zielort anfahren soll. Behandelt werden

- Roadmap-Verfahren,
- Cell-decomposition-Verfahren und
- Potential-Verfahren.



Abschließend wird dann noch ein

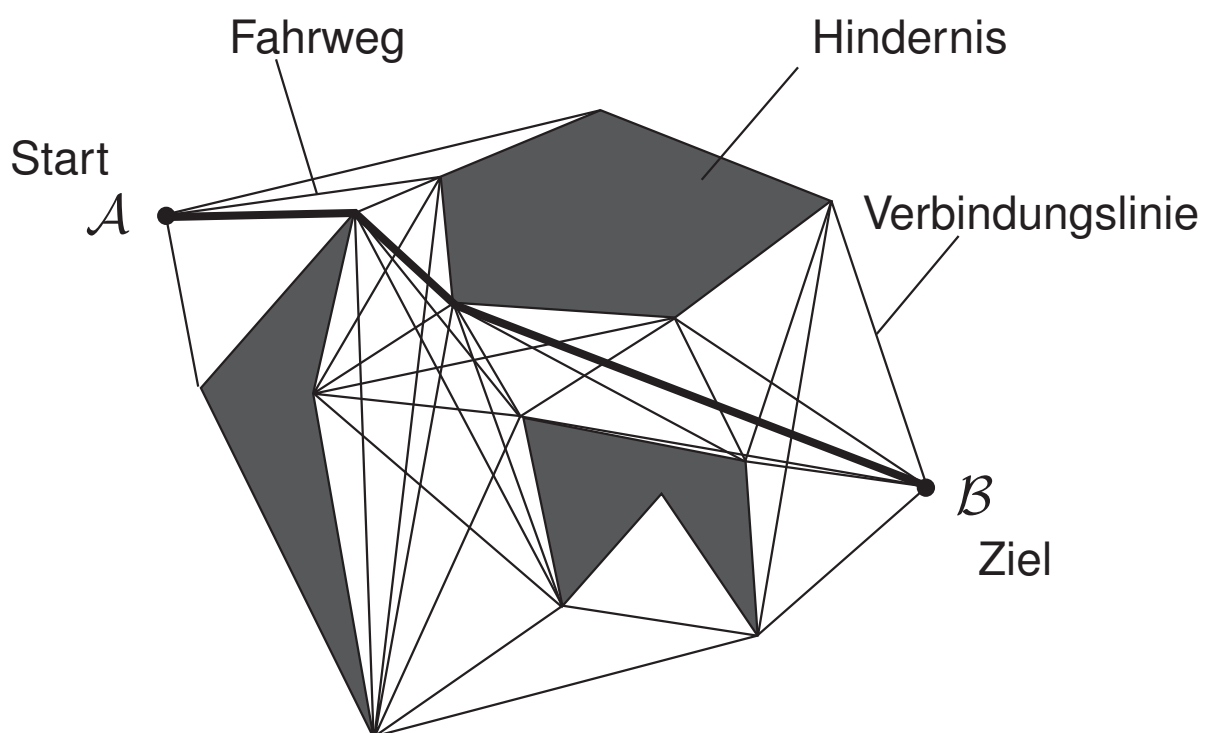
- Verfahren für Flächenabdeckungsaufgaben



beschrieben.

Zu den bekanntesten Roadmap-Verfahren gehören die „Visibility-Graph-Methode“, das „Voronoi-Diagramm-Verfahren“ und die „Freeway-Methode“. Roadmap-Verfahren basieren, wie der Name schon andeutet, auf einer Art Straßenkarte. Auf den Straßen dieser Karte kann sich der Roboter bewegen.

Die Visibility-Graph-Methode wird zuerst betrachtet. Anschaulich ist das Verfahren sehr einfach zu erklären: Man geht von einer Umgebung aus, in der alle Hindernisse durch Polygone beschrieben sind. Des Weiteren sind der Startpunkt \mathcal{A} und der Zielpunkt \mathcal{B} des Roboters bekannt. Man bestimmt nun alle Verbindungslinien zwischen den Eckpunkten der Polyeder untereinander und auch die mit den Punkten \mathcal{A} und \mathcal{B} , die nicht durch ein Hindernis laufen. Nachfolgendes Bild illustriert einen solchen sich ergebenden Graph, den „Visibility-Graph“. Er enthält alle möglichen Verbindungslinien.



Der Fahrweg des Roboters wird dann aus den Verbindungslinien ermittelt, indem man den kürzesten Weg von \mathcal{A} nach \mathcal{B} entlang dieser Linien bestimmt.

Bezeichnet man das Gebiet aller Hindernisse mit

\mathbf{H}

und den befahrbaren Raum mit

$$\mathbf{F} = \mathbb{R}^2 / \mathbf{H},$$

so lautet die Definition des Visibility-Graph G exakt:

Definition

Der nicht gerichtete Graph mit

- \mathcal{A} , \mathcal{B} und den Polyederecken von \mathbf{H} als Knoten und
 - allen Verbindungen der Knoten untereinander, die eine Kante von \mathbf{H} bilden oder vollständig in \mathbf{F} liegen,
- heißt „Visibility-Graph“.

Die einfachste Art den „Visibility-Graph“ zu ermitteln, ist die Berechnung aller Verbindungen und die nachfolgende Überprüfung, welche das Gebiet H der Hindernisse schneiden und welche nicht. Allerdings ist diese Methode auch die zeitaufwendigste.

Schnellere und effizientere Algorithmen finden sich z. B. bei Asano¹ et. al. oder Edelsbrunner².

¹Asano, T. et. al.: Visibility of Disjoint Polygons. Algorithmica 1(1), 1986, S. 49-63.

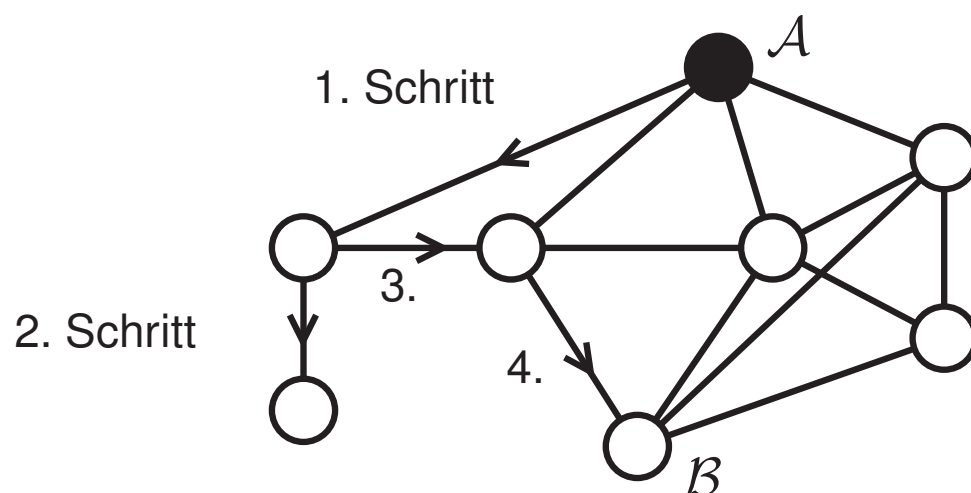
²Edelsbrunner, H.: Algorithms in Combinatorial Geometry. Monographs on Theoretical Computer Science. Springer-Verlag, 1987.

Für die Suche nach einem

- Weg zwischen \mathcal{A} und \mathcal{B} im „Visibility-Graph“

sind verschiedene Verfahren entwickelt worden. Insbesondere auch Verfahren zur Auffindung des kürzesten Weges.

Ein einfaches Verfahren zur Auffindung eines Weges von \mathcal{A} nach \mathcal{B} ist die Tiefensuche. Hier wird ausgehend von \mathcal{A} ein mit ihm verbundener Nachbarknoten bestimmt. Von diesem Knoten wird wieder ein Nachbarknoten gesucht usw. Jeder durchlaufene Knoten wird dabei sofort markiert und ein schon markierter Knoten wird nicht noch einmal durchlaufen. Es sind nun zwei Fälle möglich: Entweder man gelangt in den Knoten \mathcal{B} und hat einen Weg gefunden (wenn auch nicht zwingend den kürzesten), oder man gelangt in einen Knoten, von dem kein neuer Nachbarknoten erreichbar ist und der nicht \mathcal{B} ist. Im letzteren Fall geht man auf dem bisherigen Weg solange zurück bis man erstmals wieder einen neuen Nachbarknoten erreichen kann. Nachfolgendes Bild illustriert dies.



Das „Voronoi-Diagramm-Verfahren“ zählt ebenfalls zu den Roadmap-Verfahren. Auch dieses Verfahren basiert auf einer polygonalen Umgebungskarte. Das Voronoi-Diagramm ist ein Wegenetz, dessen Wege jeweils den maximalen Freiraum bzw. Abstand zu allen Hindernissen besitzen. Ersichtlich ist ein solches Netz günstig für die Bahnplanung, vermeidet es doch mit größtmöglicher Sicherheit Kollisionen.

Das Voronoi-Diagramm ist wie folgt definiert:

Definition

Es sei $\delta\mathbf{F}$ der Rand des befahrbaren Raumes \mathbf{F} , $\underline{x} \in \mathbf{F}$ und

$$clearance(\underline{x}) = \min_{\underline{p} \in \delta\mathbf{F}} \|\underline{x} - \underline{p}\|$$

der minimale Abstand zu einem Hindernis. Ferner sei

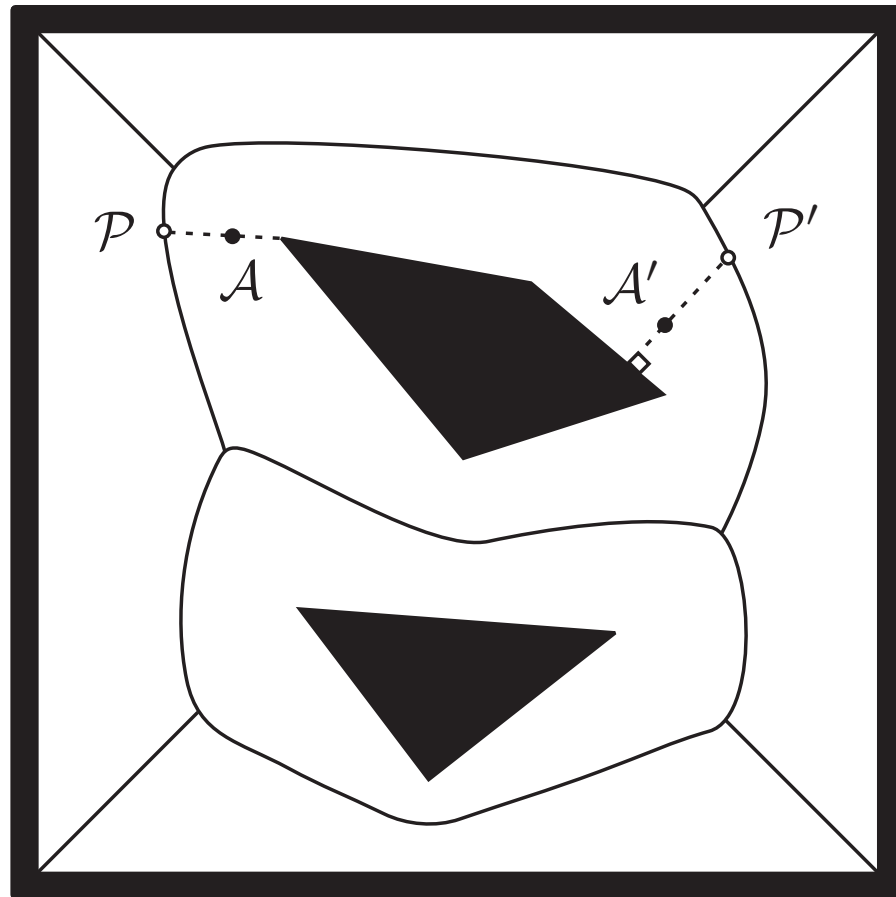
$$near(\underline{x}) = \{\underline{p} \in \delta\mathbf{F} \mid \|\underline{x} - \underline{p}\| = clearance(\underline{x})\}$$

die Menge der Punkte $\underline{p} \in \delta\mathbf{F}$ deren Abstand zu \underline{x} der $clearance(\underline{x})$ entspricht. Voronoi-Diagramm von \mathbf{F} heißt die Menge

$$voronoi(\mathbf{F}) = \{\underline{x} \in \mathbf{F} \mid card(near(\underline{x})) > 1\},$$

wobei $card(\mathbb{M})$ die Anzahl der Elemente einer Menge \mathbb{M} angibt.

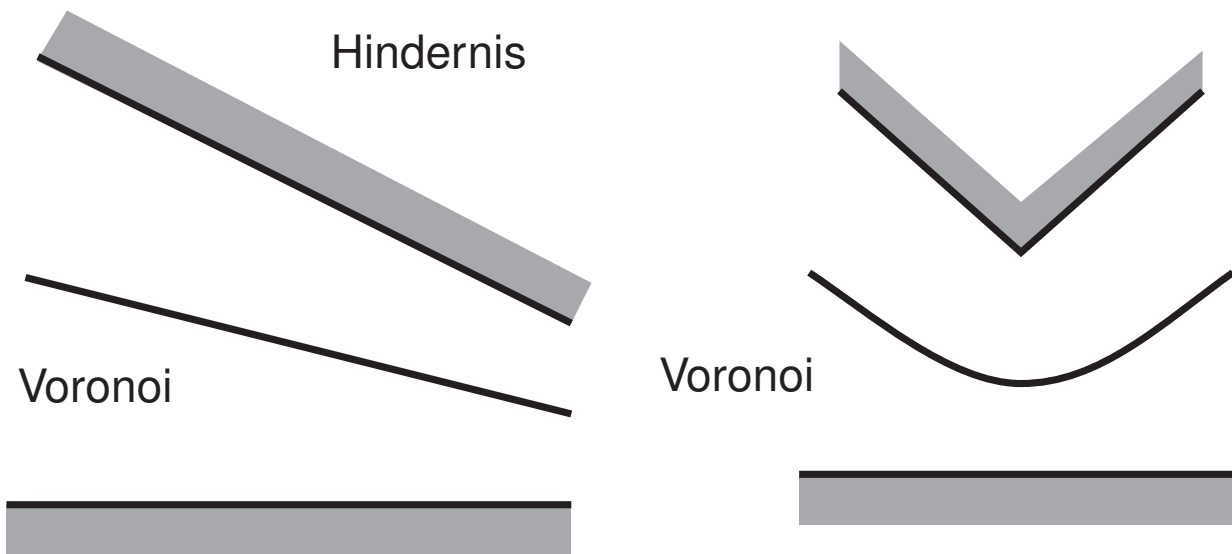
Nachfolgend ist ein Voronoi-Diagramm³ dargestellt.



Befindet sich der Roboter nicht auf dem Voronoi-Diagramm, sondern z. B. in \mathcal{A} oder \mathcal{A}' , so wird die kürzeste Entfernung zu $\delta\mathbf{F}$ ermittelt und diese Richtung verlängert, bis das Voronoi-Diagramm geschnitten wird. Im Beispiel oben ist das im Punkt \mathcal{P} bzw. \mathcal{P}' der Fall. Auf der Verbindung $\overline{\mathcal{A}\mathcal{P}}$ bzw. $\overline{\mathcal{A}'\mathcal{P}'}$ gelangt der Roboter dann auf das Wegenetz des Voronoi-Diagramms. Dieses Wegenetz garantiert den größtmöglichen Abstand des Roboters von dem Hindernis.

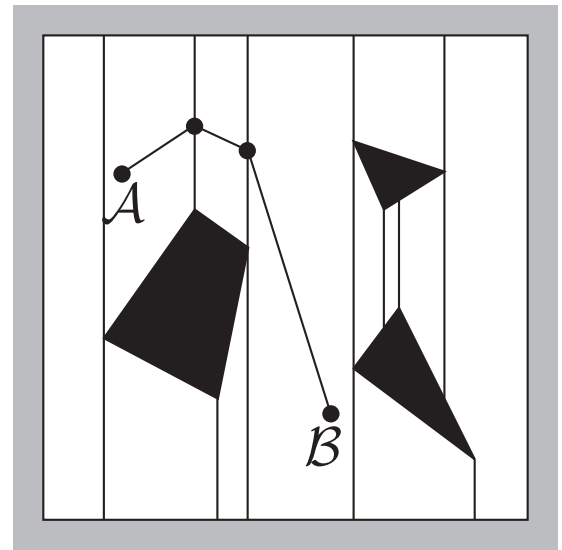
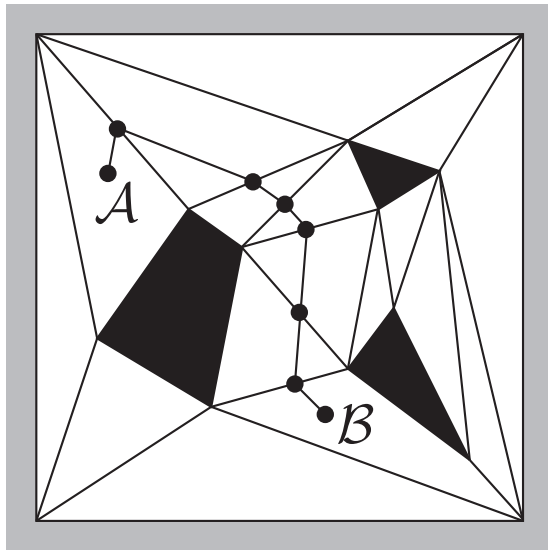
³ aus Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers, 1991.

Im betrachteten Fall polygonaler Hindernisse ist das Voronoi-Diagramm stückweise aus Geraden und Parabelteilen zusammengesetzt. Zwischen zwei durch Geraden beschriebenen Hindernissen ist das Voronoi-Diagramm ebenfalls durch ein Geradenstück gegeben. Liegen eine Ecke und eine Gerade als Hindernisumrandung gegenüber, so ergibt sich ein Parabelstück:



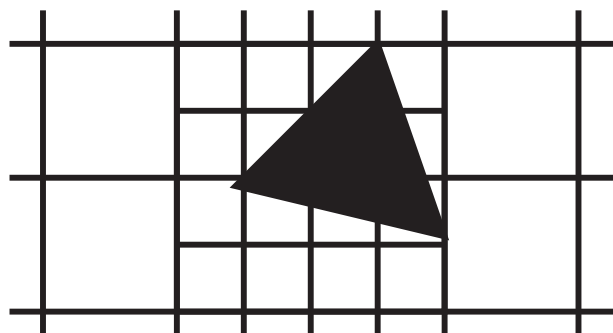
Auf obiger Basis aufbauend ist es möglich, das Voronoi-Diagramm zu konstruieren. Die Geradenstücke bzw. Parabelteile enden dabei jeweils an ihren Schnittpunkten.

Als nächstes Verfahren wird ein „Exact-cell-decomposition-Verfahren“ betrachtet. Hier wird der frei befahrbare Raum F in konvexe Polyeder (Zellen genannt) unterteilt. Diese Zellen überschneiden sich nicht und füllen F vollständig aus. Nachfolgende Bilder zeigen zwei mögliche Unterteilungen.



Als Weg zwischen Start A und Ziel B werden diese z. B. mit den Mittelpunkten der Zellenkanten verbunden und diese geeignet untereinander.

„Approximate-cell-decomposition-Verfahren“ bilden eine weitere Verfahrensklasse. Sie unterscheiden sich von „Exact-cell-decomposition-Verfahren“ dadurch, dass standardisierte Zellen, z. B. Quadrate, verwendet werden. Im Allgemeinen kann dann F nur noch näherungsweise beschrieben werden:



Eine weitere wichtige Gruppe von Bahnplanungsmethoden ist die der Potential-Verfahren⁴. Sie basieren auf einem Potentialgebirge, in dessen globalem Minimum sich das Ziel des Roboters befindet. Der Roboter wird als volumenloser Partikel betrachtet, der von seinem Startort \underline{x} in das Minimum des Potentials gezogen wird.

Dabei resultiert das Potential aus zwei Kräften,

- einer Kraft \underline{F}_{an} , die den Roboter in den Zielpunkt \underline{x}_B zieht, und
- einer Kraft \underline{F}_{ab} , die den Roboter von Hindernissen abstößt.

Das Potential

$$V_{\text{ges}}(\underline{x}) = V_{\text{an}}(\underline{x}) + V_{\text{ab}}(\underline{x})$$

ist gemäß diesen Kräften aus zwei Komponenten zusammengesetzt. Als Potential der anziehenden Kraft wird oftmals

$$V_{\text{an}}(\underline{x}) = \frac{1}{2} \zeta \|\underline{x} - \underline{x}_B\|^2$$

gewählt, wobei $\|\cdot\|$ der euklidische Abstand zwischen \underline{x} und \underline{x}_B ist. Der Wert ζ ist eine Skalierungskonstante.

Für das Potential der abstoßenden Kraft wählt man in der Regel

$$V_{\text{ab}}(\underline{x}) = \begin{cases} \frac{1}{2} \eta \cdot \left(\frac{1}{d(\underline{x})} - \frac{1}{d_0} \right)^2 & \text{für } d(\underline{x}) \leq d_0 \\ 0 & \text{für } d(\underline{x}) > d_0 \end{cases} .$$

⁴In der Literatur häufig fälschlicherweise als „Potentialfeld-Verfahren“ bezeichnet.

Dabei ist η eine Skalierungskonstante und $d(\underline{x})$ der minimale Abstand des Roboters von einem Hindernis:

$$d(\underline{x}) = \min_{\tilde{\underline{x}} \in \delta\mathbf{F}} \|\underline{x} - \tilde{\underline{x}}\|.$$

\mathbf{V}_{ab} ist identisch Null, wenn der Roboter einen größeren Abstand als d_0 von allen Hindernissen besitzt.

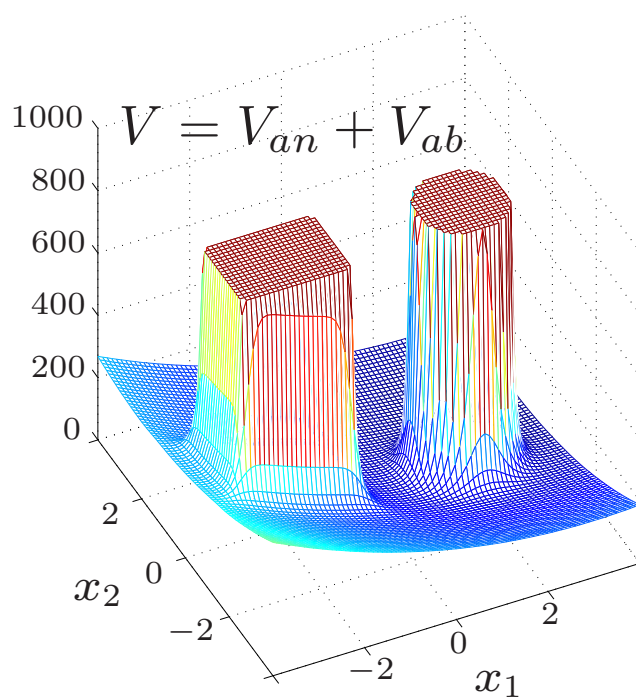
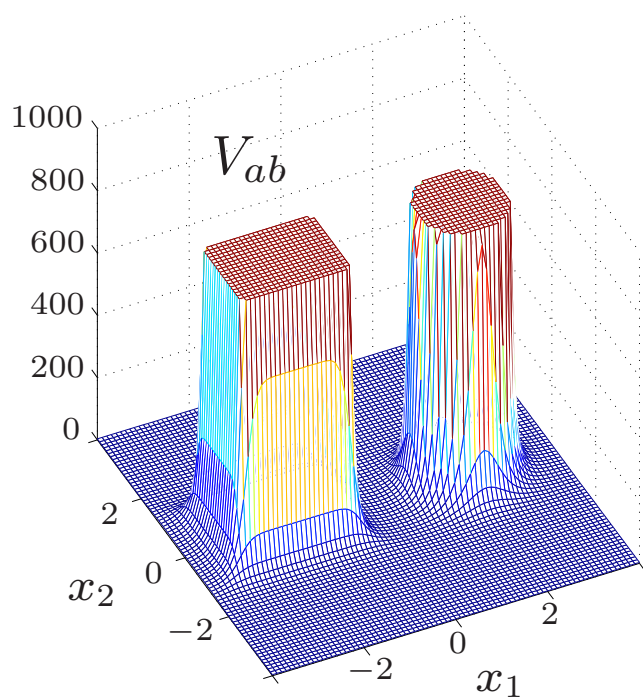
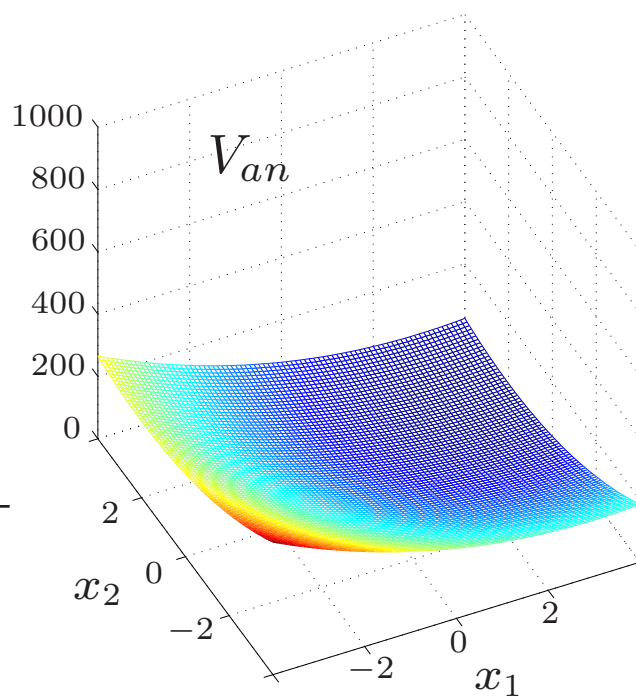
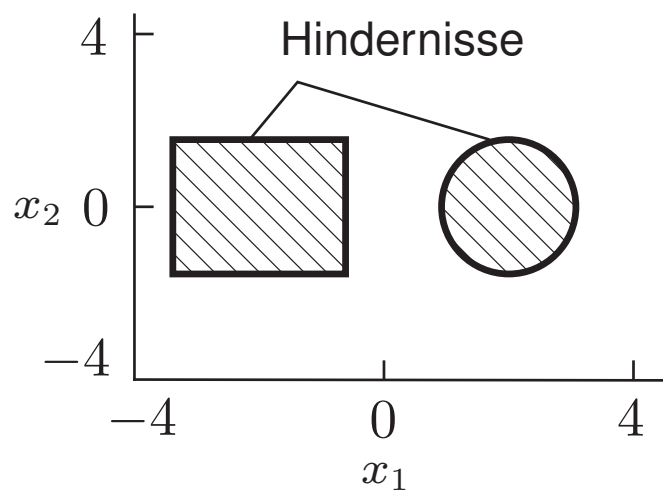
Das Kraftfeld setzt sich aus den Komponenten

$$\mathbf{F}_{\text{an}} = -\text{grad } \mathbf{V}_{\text{an}}(\underline{x}) = -\zeta(\underline{x} - \underline{x}_B),$$

$$\mathbf{F}_{\text{ab}} = -\text{grad } \mathbf{V}_{\text{ab}}(\underline{x})$$

$$= \begin{cases} \eta \left(\frac{1}{d(\underline{x})} - \frac{1}{d_0} \right) \frac{1}{d(\underline{x})^2} \text{grad } d(\underline{x}) & \text{für } d(\underline{x}) \leq d_0 \\ 0 & \text{für } d(\underline{x}) > d_0 \end{cases}.$$

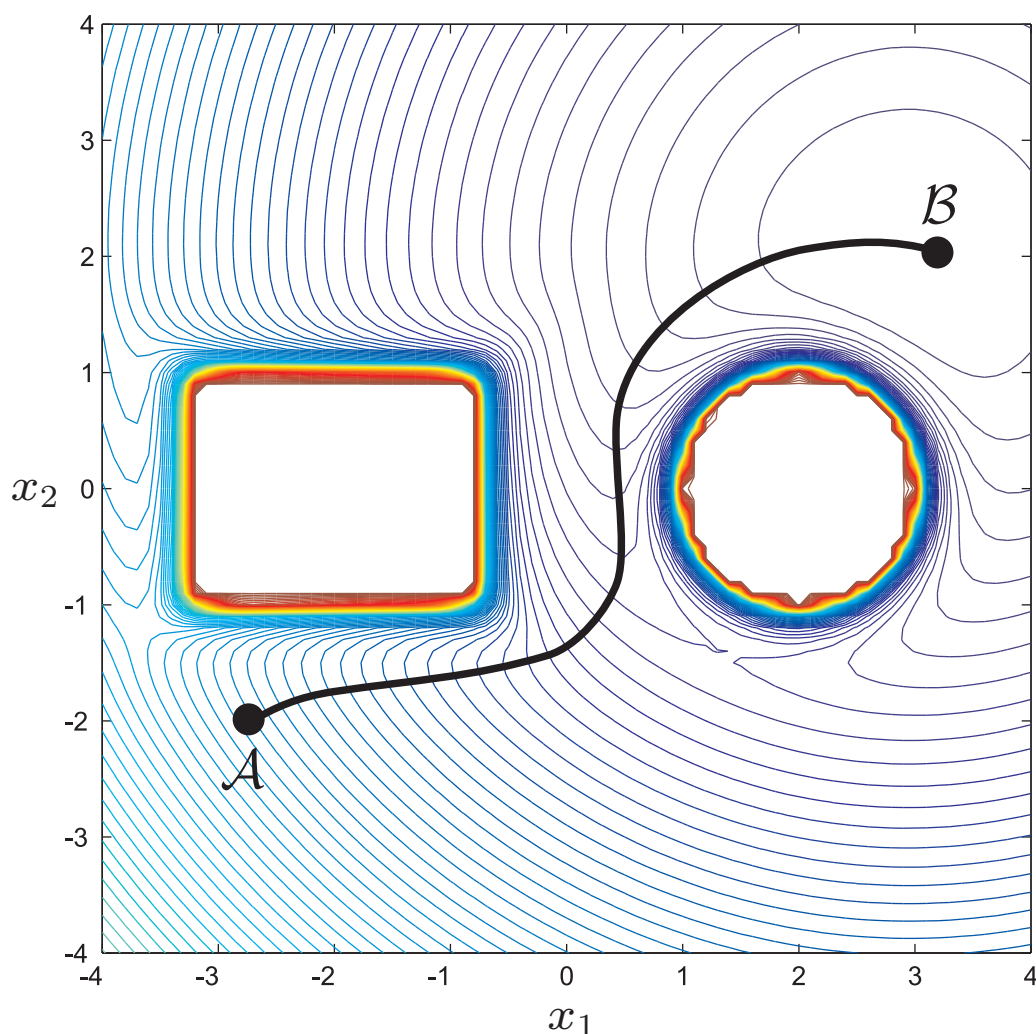
zusammen. Nachfolgende Bilder zeigen ein Beispiel für das Potential.



Eine der einfachsten Methoden mittels des Potentials eine Bahn zu planen, ist die Gradientensuche. Hierbei wird ausgehend vom Startpunkt \underline{x}_0 in kleinen Schritten entlang des Gradienten im Potentialgebirge ein Pfad gesucht:

$$\underline{x}_{i+1} = \underline{x}_i - h \cdot \text{grad } \mathbf{V}_{\text{ges}}(\underline{x}_i).$$

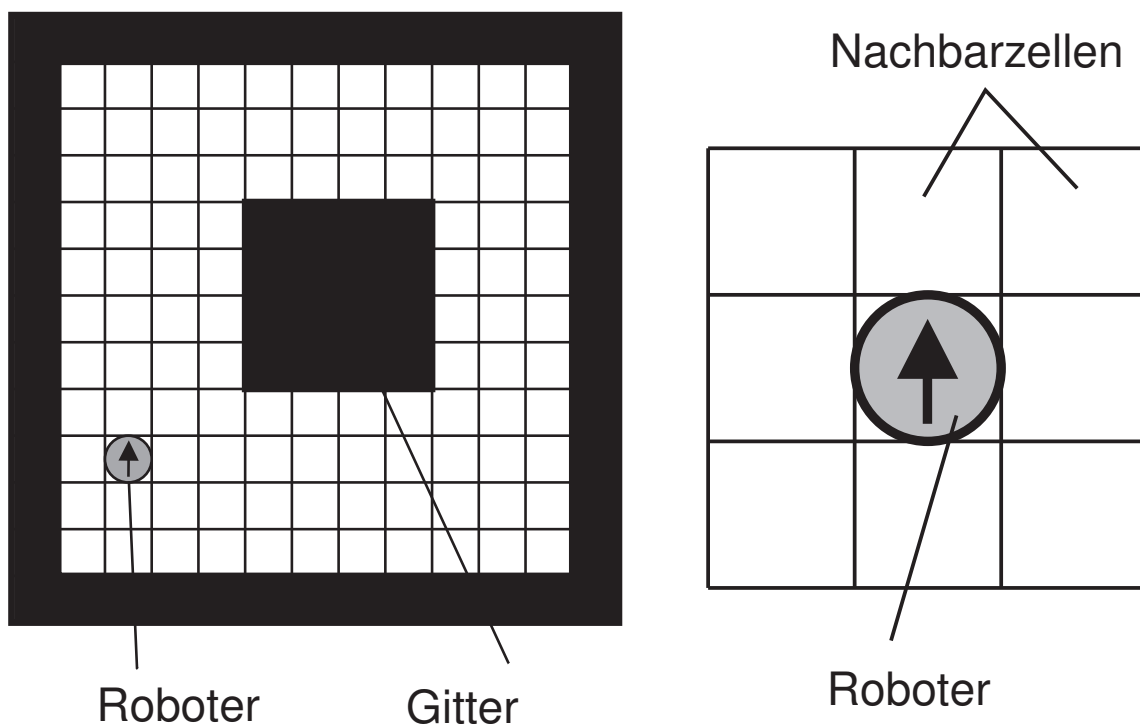
Dabei gibt h die Schrittlänge an. Nachfolgendes Bild zeigt ein Beispiel.



Nachteil dieses einfachen Verfahrens ist, dass es in lokalen Minima – sofern vorhanden – hängen bleiben kann.

Leistungsfähigere aber auch aufwendigere Verfahren finden sich in der angegebenen Literatur.

Als Beispiel für eine flächendeckende Bahnplanung wird der Algorithmus von Henrich und Graf⁵ betrachtet. Die Umgebung soll polygonal und durch ein rechteckiges Gitternetz überzogen sein, dessen Elemente die Grundfläche des Roboters besitzen. Der Roboter kann dann von seinem aktuellen Standpunkt aus acht Nachbarfelder erreichen. Nachfolgende Bilder illustrieren dies.



Dabei ist vorausgesetzt, dass der Roboter sich auf der Stelle drehen kann. Das ist tatsächlich bei den gängigen Modellen der Fall.

⁵Henrich, D. und Graf, R.: Flächenabdeckende Bahnplanung in vollständig oder nichtbekannten Umgebungen. 14.Fachgespräche AMS '98, Karlsruhe, 30.11.-01.12.'98. Reihe „Informatik Aktuell“, Springer Verlag, S.188-198.

Bei der Bahnplanung werden nicht befahrbare und schon befahrene Zellen als „kritische Zellen“ bezeichnet. Die Bahnplanung basiert auf den folgenden einfachen Regeln:

1. Fahre nur in Zellen, die nicht kritisch sind.
2. Fahre nach rechts.
3. Fahre geradeaus, wenn rechts nicht möglich ist.
4. Fahre nach links, wenn rechts und geradeaus nicht möglich ist.

Besteht keine Möglichkeit mehr in eine nicht kritische Nachbarzelle zu fahren, so wird mittels der Potentialmethode ein Weg gesucht, der den Roboter in eine Region mit nicht kritischen Zellen führt. Dort wird dann wieder obiger Algorithmus angewendet.

Ein zentrales Element des Verfahrens ist ein Speicher, in dem festgehalten wird, welche Zellen kritisch sind und welche nicht.

In nachfolgenden Bildern ist eine Beispielsituation gezeigt, die die Arbeitsweise des Algorithmus veranschaulicht.

Bild 1

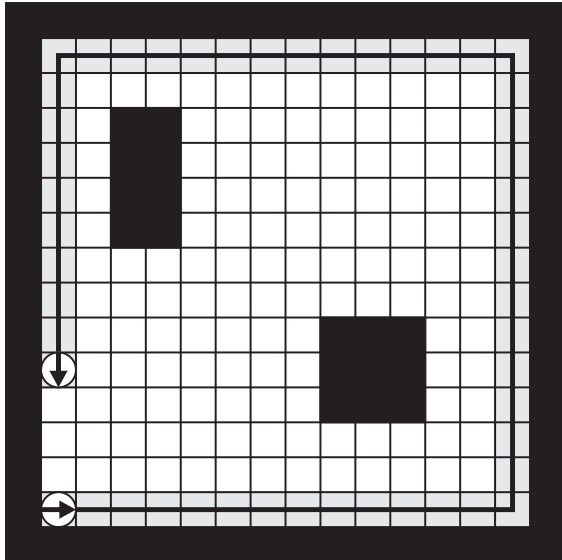


Bild 2

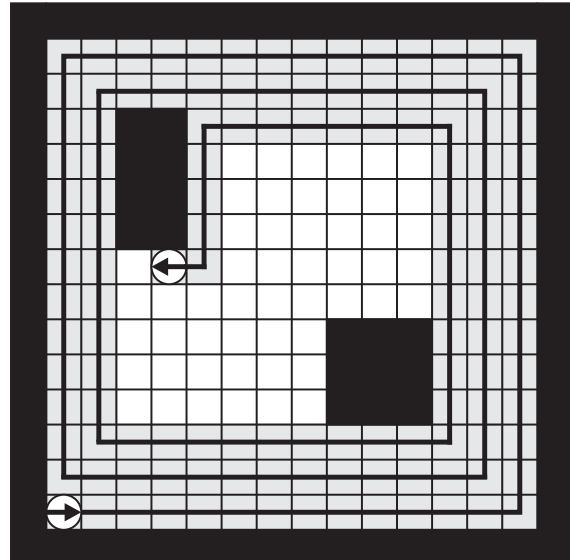


Bild 3

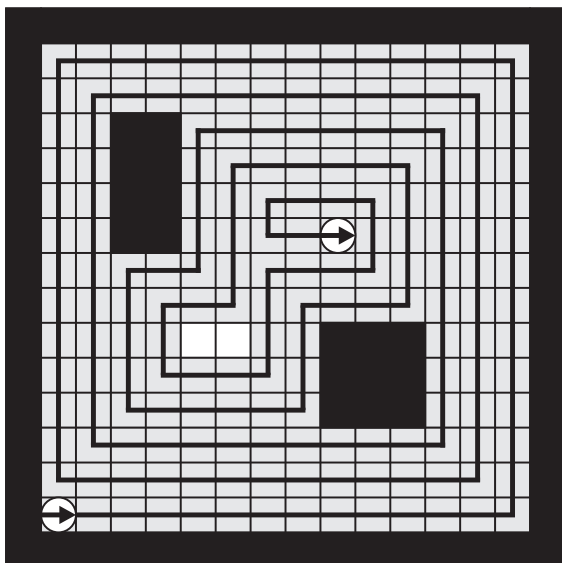
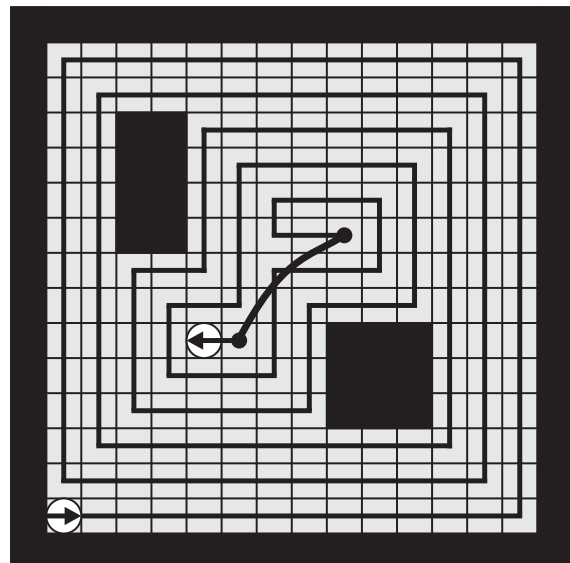


Bild 4



In Bild 3 erreicht der Roboter eine Position, in der alle Nachbarzellen kritisch sind. Daher wird im nächsten Schritt mittels des Potentialfeld-Verfahrens der noch nicht befahrene Bereich angesteuert. Dies zeigt Bild 4.

KAPITEL

3

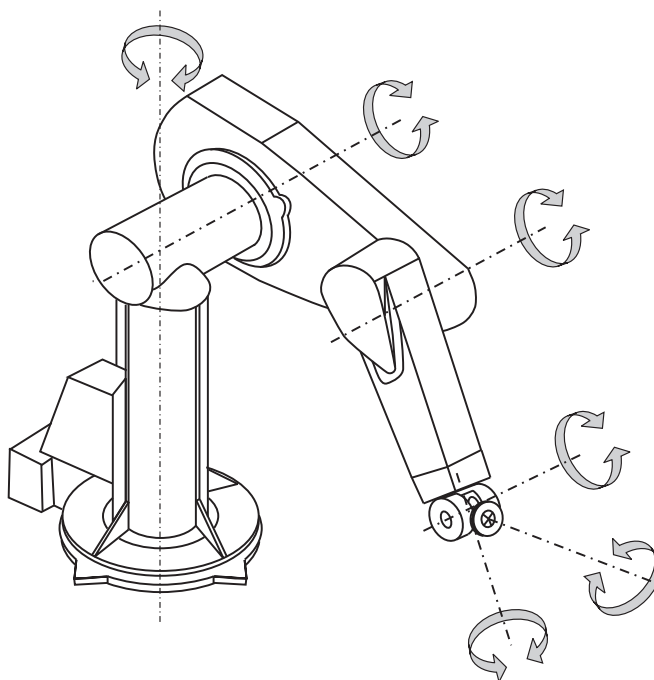
Industrieroboter

ABSCHNITT 3.1

Typen und Anwendungen

Industrieroboter sind mechanische autonome Handhabungsgeräte. Ihre Autonomie, d.h. ihre vom Menschen nicht in direkter Weise abhängende Aufgabenausführung, wird durch Computer erreicht. Sie steuern die Antriebe eines Roboters so, daß er eine bestimmte Aufgabe ohne Fremdeinfluß erledigt. Im Gegensatz hierzu bezeichnet man mechanische Handhabungsgeräte, die vom Menschen gesteuert oder ferngesteuert werden, als Manipulatoren.

In der Regel werden Industrieroboter in Produktionseinrichtungen eingesetzt, um deren Automationsgrad zu erhöhen. Sie bestehen im Allgemeinen aus einem Basiskörper, an den eine Kette von Gliedern angebracht ist. Die einzelnen Glieder sind dabei über Gelenke verbunden, die durch Antriebe bewegbar sind. Das Bild zeigt einen solchen Roboter: den PUMA 560 (Programmable Universal Machine for Assembly).



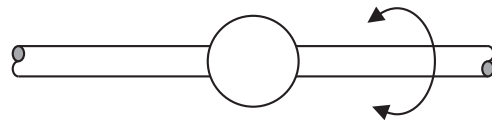
PUMA 560

Neben rotatorischen Gelenken, wie sie der PUMA 560 besitzt, gibt es auch Roboter mit translatorischen Gelenken. Nachfolgende Liste zeigt die vier möglichen Gelenktypen:

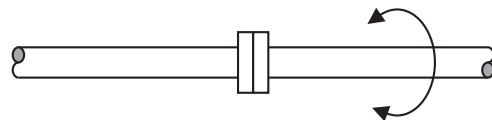
1. Translationsgelenk



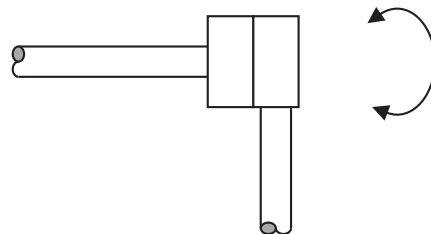
2. Rotationsgelenk



3. Torsionsgelenk



4. Revolvergelenk



Jede Bewegungsmöglichkeit bezeichnet man als Freiheitsgrad (der Bewegung). Ein sechsgelenkiger Roboter verfügt also im Allgemeinen über sechs Freiheitsgrade. Als Stellglieder sind

- Elektromotoren
- Hydraulikantriebe und
- pneumatische Antriebe

möglich.

Nachfolgende Tabelle vergleicht die Eigenschaften von elektrischen und hydraulischen Antrieben bezüglich ihrer Eignung als Stellantriebe von Robotern.

	Elektromotoren	Hydraulikantriebe
<u>Stellkraft</u>		
Gewicht des Antriebs	mittel	hoch bis sehr hoch
Kosten	mittel	hoch
Wartungsaufwand	niedrig	mittel
Verschmutzungspotential	sehr niedrig	hoch

Pneumatische Antriebe werden nur bei sehr kleinen Robotern eingesetzt, bei denen geringe Stellkräfte ausreichend für die zu bearbeitende Aufgabe sind. Ein Vorteil sind ihre niedrigen Kosten.

Elektromotoren haben sich als Standardantriebe für Industrieroboter etabliert. Gegenüber Hydraulikantrieben besitzen sie die in der Tabelle dargestellten Vorteile. Hydraulikantriebe besitzen als Vorteil im Wesentlichen nur eine höhere Stellkraft.

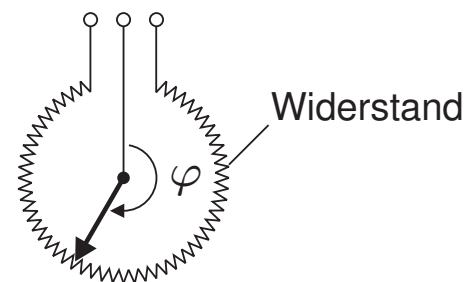
Als Sensoren werden bei Industrierobotern vor allem

- Positionssensoren und
- Geschwindigkeitssensoren

eingesetzt. Positionssensoren können sein:

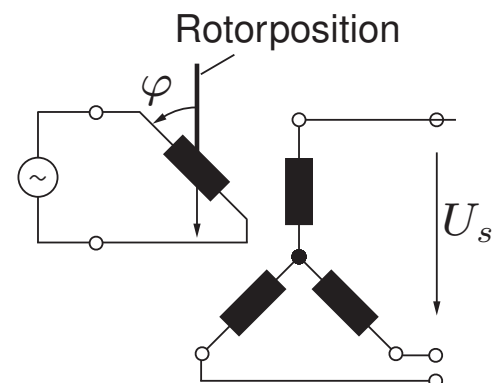
- **Potentiometer**

Am Greifer kann eine dem Drehwinkel φ proportionale Spannung abgegriffen werden.



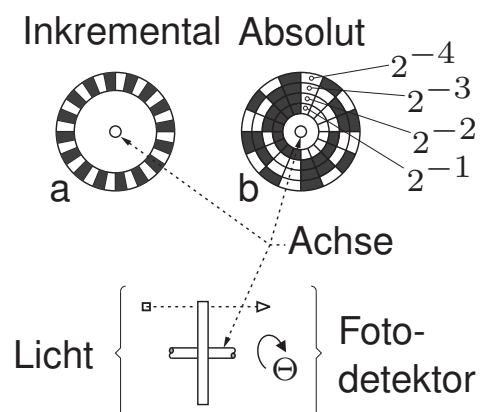
- **Drehmelder**

Drehmelder bestehen im einfachsten Fall aus einer Rotorwicklung und einigen Stabwicklungen. Sie arbeiten ähnlich wie Transformatoren. Die Statorspannung U_s ist abhängig vom Drehwinkel φ .



- **Kodiereinrichtungen**

Die häufigsten Kodiereinrichtungen sind optische. Inkrementale und absolute sind zu unterscheiden. Bei ihnen kann der Winkel direkt in Binärcode ermittelt werden.



Geschwindigkeitssensoren sind im Allgemeinen Tachogeneratoren (Gleichstromgeneratoren).

Die Einsatzbandbreite von Industrierobotern ist sehr groß. Aufgrund ihrer universellen Bewegungsmöglichkeiten und einer Reihe von an den Greifer montierbaren Werkzeugen lassen sich mit ihnen sehr verschiedene Arbeiten automatisieren.

Nachfolgende Tabelle¹ gibt einen Überblick über typische Anwendungsfelder.

	Schweißen	Schneiden	Bearbeiten	Auftragen	Handhaben	Montieren
Nahrungsmittel					x	
Holz			x		x	
Papier					x	x
Chemie					x	
Glas, Keramik			x	x		
Metallerzeugnisse	x	x	x		x	
Maschinenbau	x		x	x		
Elektroindustrie	x					x
Optik					x	
Fahrzeugbau	x				x	x

¹ Quelle = KUKA

Je nach Einsatzzweck, der in der folgenden Tabelle² noch etwas detaillierter ist als in der vorherigen, sind die Anforderungen an einen Industrieroboter und seine Regelung sehr unterschiedlich.

Anforderung / Einsatzbereich	Wieder- holge- nauigkeit	Pose- Genauig- keit	Geschw. der Bewegung	Bahn- wieder- holge- nauigkeit	absolute Bahn- genauig- keit	Bahn- geschw.	Program- mierung
Werkstück- handhabung	hoch	hoch	hoch	keine	keine	keine	Teach-In
Kommissionieren	hoch	hoch	hoch	keine	keine	keine	Aut. Gen. CAD/Offline
Plattieren	hoch	hoch	hoch	keine	keine	keine	Aut. Gen. CAD/Offline
Beladen	hoch	mittel	sehr hoch	gering	keine	keine	Teach-In
Entladen	mittel	hoch	sehr hoch	gering	keine	keine	Teach-In
Messen/Prüfen	hoch	hoch	hoch	hoch	hoch	gering	CAD/Offline
Montage	sehr hoch	sehr hoch	sehr hoch	sehr hoch	sehr hoch	gering	CAD/Offline
Bahnschweißen MIG/MAG	keine	keine	hoch	hoch	gering	hoch	Teach-In
Bahnschweißen WIG	hoch zünden	keine	hoch	sehr hoch	gering	hoch	Teach-In
Laserschneiden/ -schweißen	hoch	hoch	gering	sehr hoch	hoch	sehr hoch	CAD/Offline
Punktschweißen	hoch	hoch	sehr hoch	keine	keine	keine	Aut. Gen. CAD/Offline
Entgraten	keine	keine	hoch	sehr hoch	hoch	hoch	Teach-In
Schleifen/Polieren	keine	keine	hoch	sehr hoch	hoch	mittel	Teach-In
Klebe- und Dichtmittelauftrag	keine	keine	mittel	hoch	keine	sehr hoch	Teach-In
Profilextrusion	keine	keine	mittel	sehr hoch	hoch	hoch	Teach-In
Lackierung	keine	keine	mittel	gering	keine	sehr hoch	Play-Back
Inspektion/ Qualitätssicherung	hoch	hoch	hoch	hoch	sehr hoch	mittel	Teach-In

²Quelle = Reis Robotics

Die Sicherheit von Industrierobotern ist eine weitere wichtige Anforderung bei ihrem Einsatz. Prinzipiell kann es zu Gefährdungen von Menschen, die sich im Bewegungsbereich des Roboters aufhalten, kommen. Unkontrollierte oder unvorhergesehene Bewegungen des Roboters können dabei zu erheblichen Verletzungen führen. Auftreten können solche Situationen, wenn

- Menschen in der Nähe des Roboters arbeiten,
- Menschen den Bewegungsbereich des Roboters unwissentlich oder fahrlässig betreten,
- der Roboter gewartet wird,
- der Roboter programmiert wird.

Um obige Situationen zu verhindern, werden sowohl

- passive als auch
- aktive Sicherheitseinrichtungen

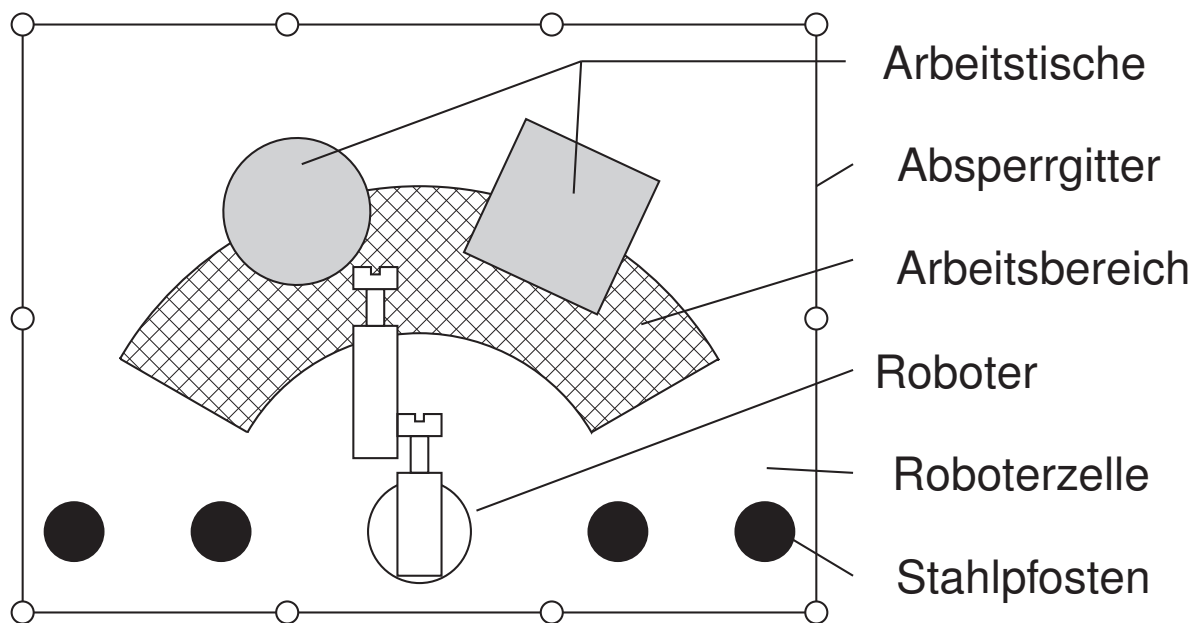
eingesetzt.

Zu den passiven Sicherheitseinrichtungen gehören

- Absperrgitter,
- Stahlpfosten, die den Roboterarm bei unkontrollierten Bewegungen zerstören,
- NOT-AUS-Schalter.

Installiert werden diese Einrichtungen im Bewegungsbereich des Roboters.

Dieser umfaßt das gesamte Gebiet, das durch den Roboterarm erreicht werden kann. Der Arbeitsbereich dagegen umfaßt nur den Bereich, in dem sich der Roboterarm im normalen Betrieb bewegt. Nachfolgende Skizze zeigt eine Roboterarbeitszelle und die beschriebene Einrichtungen.



Neben den passiven gibt es auch aktive Sicherheitsmaßnahmen. Hier wird über

- Sensoren das Eindringen in die Bereiche der Roboterzelle erkannt. Als Sensoren werden
 - Lichtvorhänge und
 - druckempfindliche Matten
- eingesetzt.

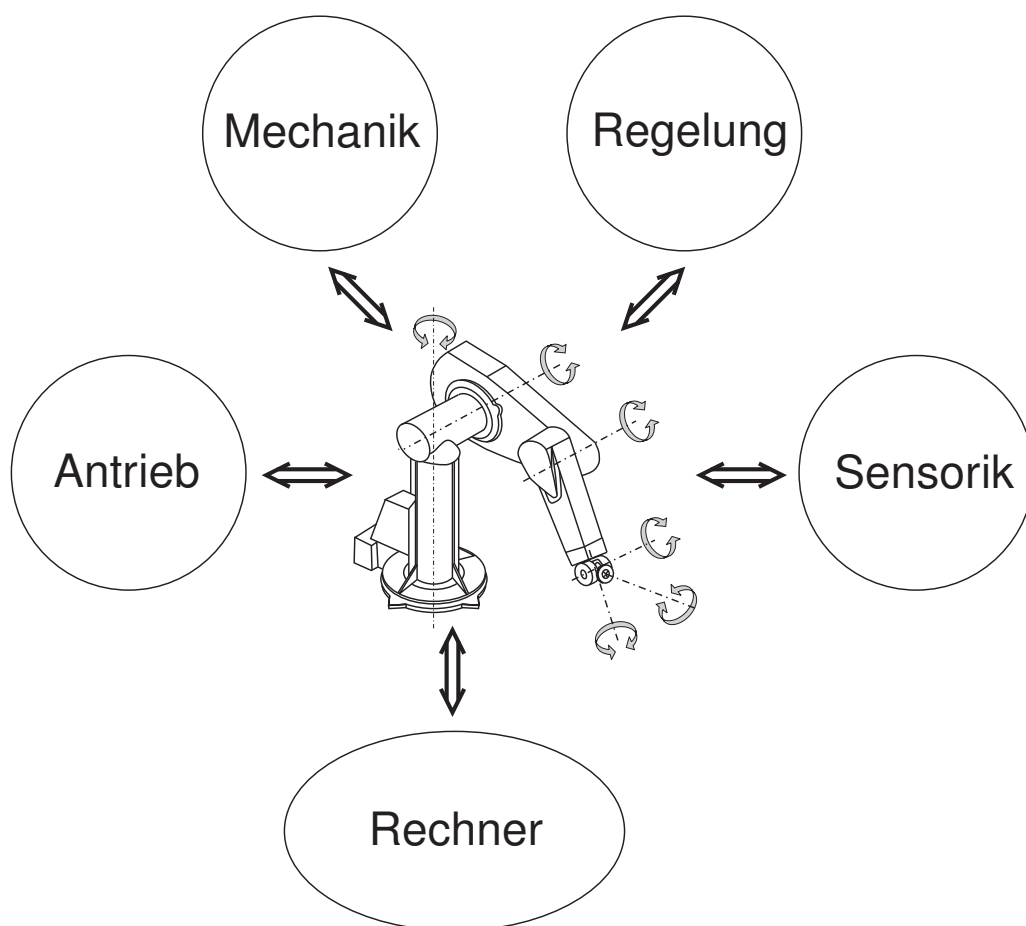
Man unterscheidet drei Sicherheitsstufen:

1. Eindringen in die äußere Umgebung der Roboterzelle,
2. Eindringen in das Innere der Roboterzelle,
3. Eindringen in den Arbeitsbereich oder die unmittelbare Nähe des Roboters.

Je nach Situation, die durch die Sensoren erkannt wird, sind dann verschiedene Reaktionen möglich:

1. Warnsignal,
2. Reduktion der Robotergeschwindigkeit auf eine sichere Stufe,
3. aktive Verhinderung einer Kollision durch den Roboter,
4. Durchführung von Arbeiten in entfernten Bereichen,
5. Abschaltung des Roboters.

Industrieroboter sind von ihrer Anwendungsseite betrachtet im Bereich der Automatisierungstechnik angesiedelt. Betrachtet man sie aus technischer, d.h. konstruktiver Sicht, so ist ihre Konstruktion eine interdisziplinäre Aufgabe. Fünf Bereiche fließen hier zusammen: Mechanik, Antriebstechnik, Meßtechnik, Regelungstechnik, Rechnertechnik (Hard- und Software). Nachfolgendes Bild illustriert dies.

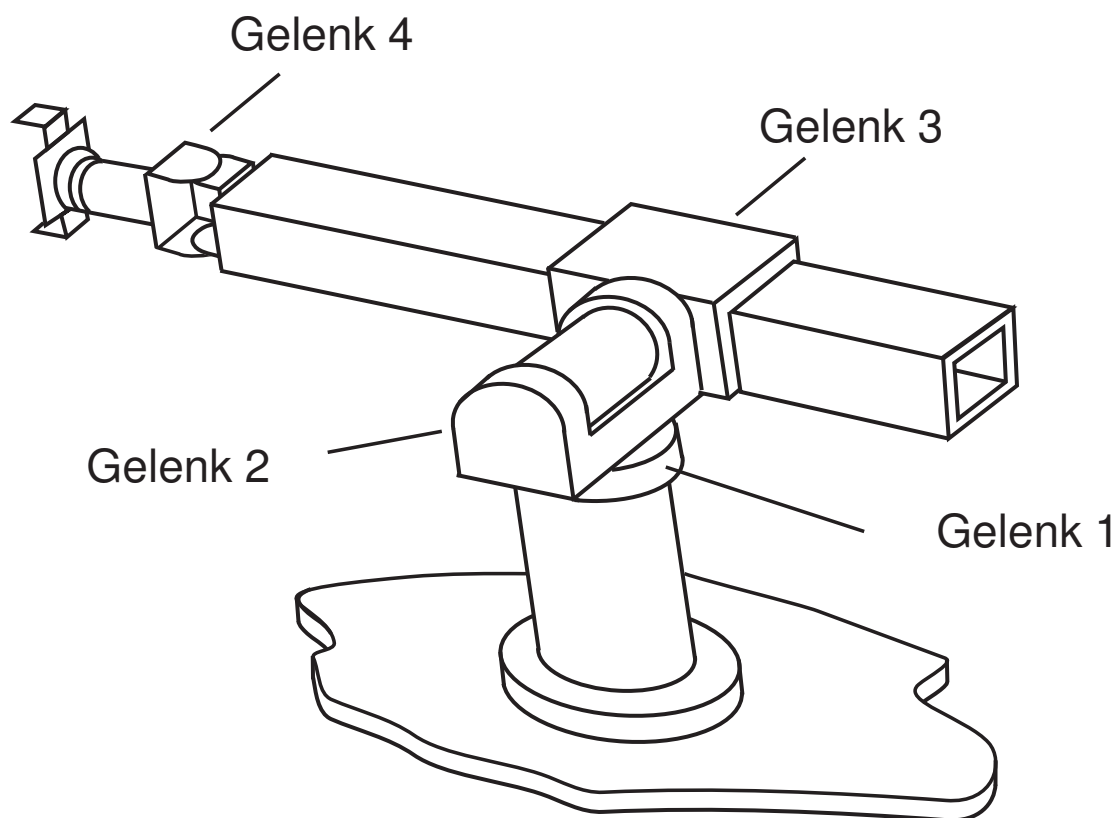


In den folgenden Abschnitten wird auf diese Themen eingegangen. Hierbei bilden die mechanischen Grundlagen und die Regelungstechnik die Hauptthemen.

ABSCHNITT 3.2

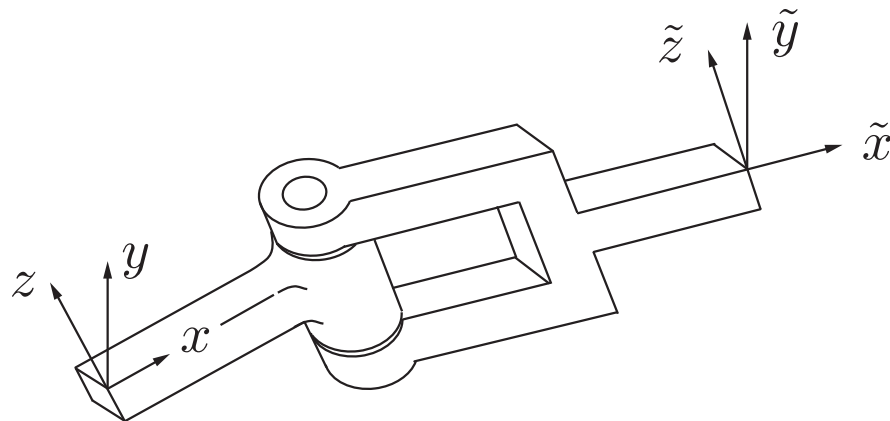
Geometrie und Kinematik

Im Weiteren wird die Kinematik von Industrierobotern beschrieben, d.h. die Lage und Bewegung von Roboter-elementen wird mittels der analytischen Geometrie bestimmt. Durch Bewegungen der Roboter-gelenke und -glieder kann der Roboter-greifer rotatorisch und translatorisch positioniert werden. Nachfolgendes Bild illustriert noch einmal diese Möglichkeiten anhand eines einfachen Industrieroboters mit drei rotatorischen und einem translatorischen Freiheitsgrad.

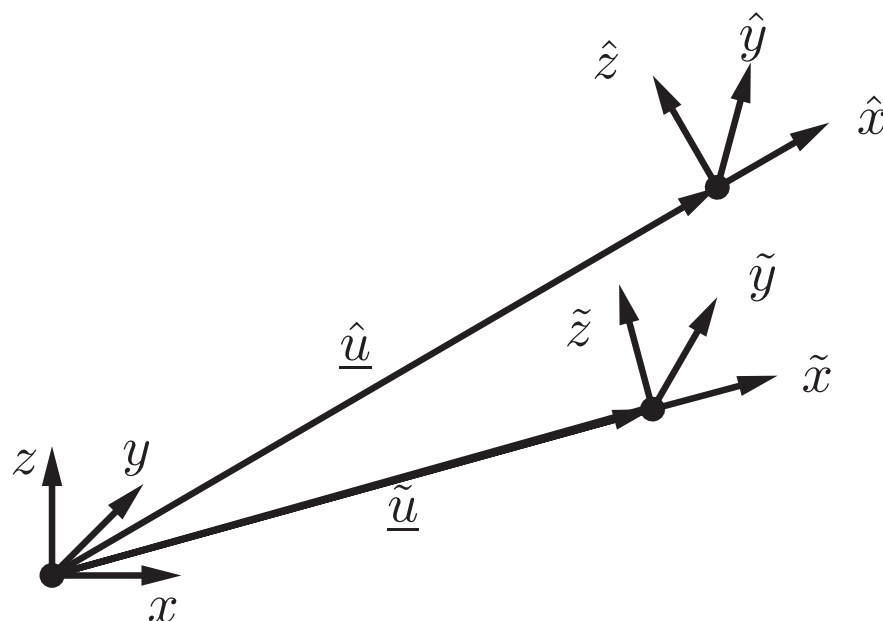


Stanford
Manipulator

Für die Positionierung des Robotergreifers ist es erforderlich zu wissen, wie die Rotationen und Translationen von Gelenken und Gliedern wirken. D.h. wie sich aus alten Gliedpositionen neue berechnen lassen. Um dies zu beschreiben, werden im Folgenden zuerst nur ein Gelenk und ein Glied betrachtet:



Durch eine Bewegung im Gelenk ändert sich die Position des Koordinatensystems $(\tilde{x}, \tilde{y}, \tilde{z})$, und es entsteht das neu orientierte Koordinatensystem $(\hat{x}, \hat{y}, \hat{z})$. Dabei sei es möglich, daß Translationen wie auch Rotationen auftreten. Nachfolgendes Bild illustriert die Koordinatensysteme:



Das neue Koordinatensystem $(\hat{x}, \hat{y}, \hat{z})$ kann durch lineare Transformationen aus dem alten System $(\tilde{x}, \tilde{y}, \tilde{z})$ berechnet werden. Zu diesem Zweck geht man wie folgt vor:

1. Man überführt den Ursprungsvektor $\underline{\tilde{u}}$ in den neuen Ursprungsvektor $\underline{\hat{u}}$. Dazu dreht, d.h. rotiert man $\underline{\tilde{u}}$ mittels einer Rotationsmatrix \underline{R} so, daß $\underline{\tilde{u}}$ in dieselbe Richtung wie $\underline{\hat{u}}$ weist. Addition eines Translationsvektors \underline{t} liefert dann $\underline{\hat{u}}$:

$$\underline{\hat{u}} = \underline{R} \underline{\tilde{u}} + \underline{t}. \quad (1)$$

2. Auch die Achsen des Koordinatensystems $(\tilde{x}, \tilde{y}, \tilde{z})$ werden durch die Rotation neu ausgerichtet. D.h. die Einheitsvektoren $\underline{\tilde{e}}_x, \underline{\tilde{e}}_y, \underline{\tilde{e}}_z$ haben im neuen System die Ausrichtungen $\underline{\hat{e}}_x, \underline{\hat{e}}_y, \underline{\hat{e}}_z$. Beschreiben läßt sich diese Rotation des Koordinatensystems durch die Gleichungen

$$\underline{\hat{e}}_x = \underline{R} \underline{\tilde{e}}_x, \quad \underline{\hat{e}}_y = \underline{R} \underline{\tilde{e}}_y, \quad \underline{\hat{e}}_z = \underline{R} \underline{\tilde{e}}_z. \quad (2)$$

Die vier Transformationsgleichungen (1), (2) kann man zu einer zusammenfassen:

$$\underbrace{\begin{bmatrix} \underline{\hat{e}}_x & \underline{\hat{e}}_y & \underline{\hat{e}}_z & \underline{\hat{u}} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\underline{\hat{K}}} = \underbrace{\begin{bmatrix} \underline{R} & \underline{t} \\ 0 & 1 \end{bmatrix}}_{\underline{T}} \cdot \underbrace{\begin{bmatrix} \underline{\tilde{e}}_x & \underline{\tilde{e}}_y & \underline{\tilde{e}}_z & \underline{\tilde{u}} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\underline{\tilde{K}}}.$$

Koordinatensystem Transformation Koordinatensystem

Es fällt auf, dass man durch obige Form der Transformationsgleichung vom \mathbb{R}^3 in den \mathbb{R}^4 gewechselt ist. Dies hat den Nachteil einer Dimensionserhöhung, aber auch den Vorteil, dass man nun Translation und Rotation durch eine Transformationsmatrix \underline{T} beschreiben kann. Die Vektoraddition der Translation entfällt also bei dieser Schreibweise.

Die vierdimensionalen Vektoren bzw. Koordinaten

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad (1)$$

dieser Darstellung bezeichnet man auch als homogene Koordinaten. Das vierte Element w beinhaltet dabei gegenüber Vektoren des \mathbb{R}^3 eine Zusatzinformation. Ist $w = 0$, so beschreibt der Vektor (1) eine Richtung. Ist $w = 1$, so beschreibt er einen Punkt.

Vektoren wie

$$\begin{bmatrix} \hat{e}_x \\ 0 \end{bmatrix} \quad \text{und} \quad \begin{bmatrix} \tilde{e}_z \\ 0 \end{bmatrix}$$

beschreiben also Richtungen im \mathbb{R}^3 ; Vektoren wie

$$\begin{bmatrix} \hat{u} \\ 1 \end{bmatrix} \quad \text{und} \quad \begin{bmatrix} \tilde{u} \\ 1 \end{bmatrix}$$

Punkte.

Translationen und Rotationen können, wie gesagt, durch eine Matrix mit der Struktur

$$\underline{T} = \begin{bmatrix} \underline{R} & \underline{t} \\ \underline{0} & 1 \end{bmatrix}$$

beschrieben werden. Hierbei leistet \underline{R} den rotatorischen, \underline{t} den translativen Teil.

Bei Translationen hat \underline{T} die Form

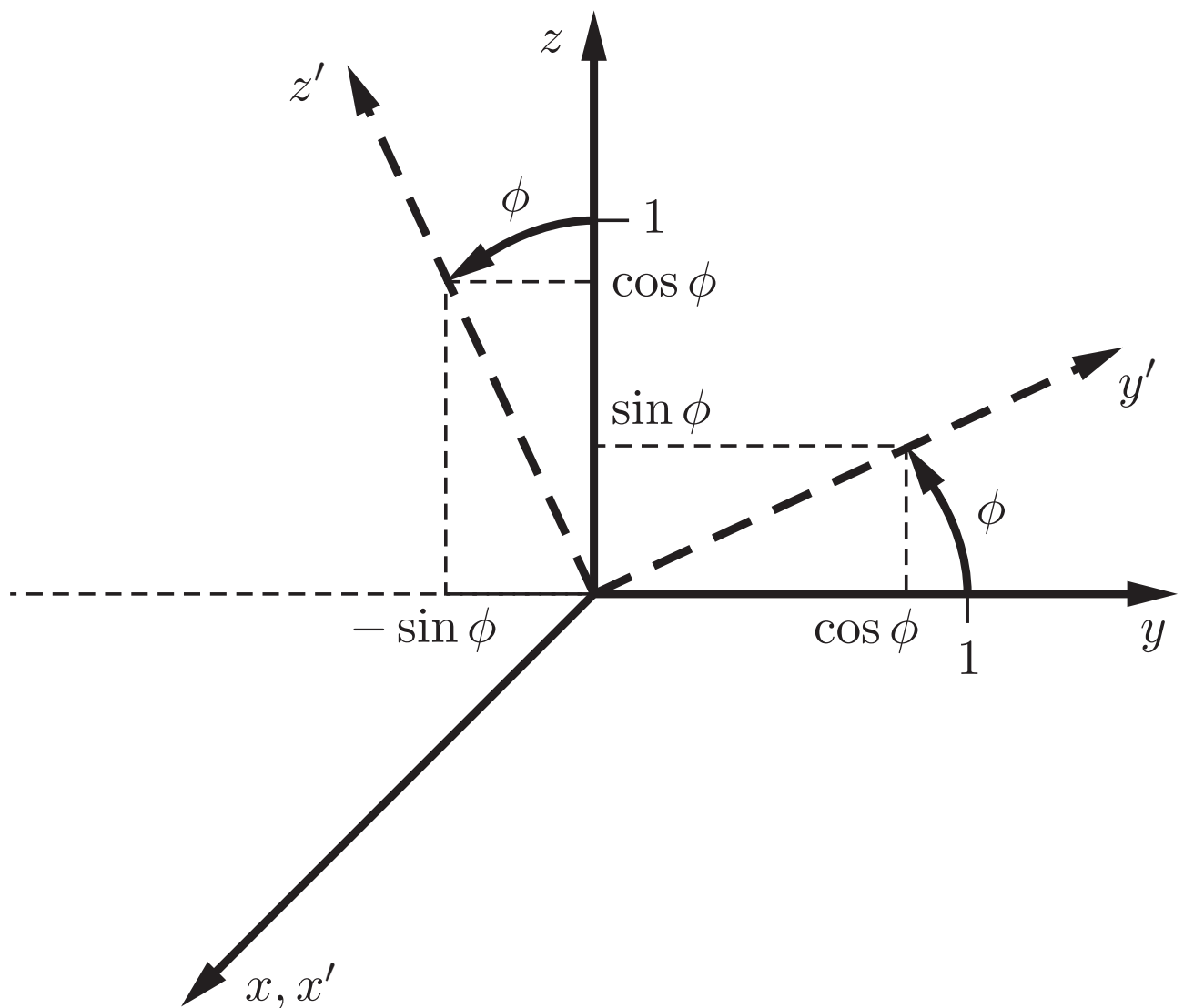
$$\underline{T}_{trans}(t_1, t_2, t_3) = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Bei Rotationen hat \underline{T} die allgemeine Form

$$\underline{T}_{rot} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Basisrotationen sind die Drehungen um einen Winkel ϕ um die x , y oder z -Achse.

Betrachtet man folgende Abbildung, so läßt sich die Rotationsmatrix $\underline{T}_{rot}(x, \phi)$, bei der eine Drehung mit Winkel ϕ um die x -Achse ausgeführt wird, direkt plausibilisieren.



Man erhält die Transformationsmatrix

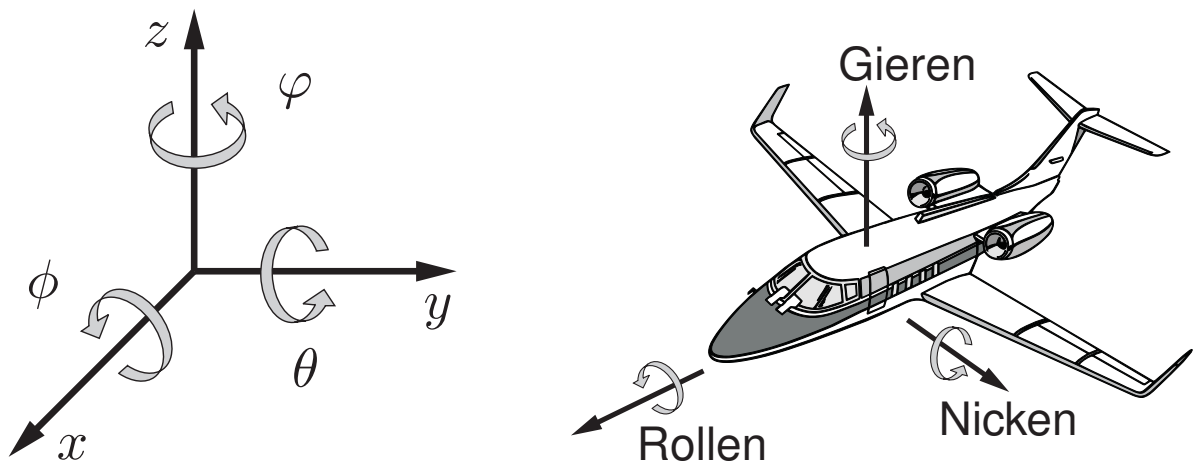
$$\underline{T}_{rot}(x, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Analog für Drehungen um die y - bzw. z -Achse mit den Drehwinkeln θ, φ

$$\underline{T}_{rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ,$$

$$\underline{T}_{rot}(z, \varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Drehungen um die Winkel ϕ , θ , φ bezeichnet man als Rollen, Nicken, Gieren. Diese Begriffsdefinition stammt aus der Luftfahrt. Sie wird aber auch bei Schiffen und Automobilen verwendet. Dazu nachfolgendes Bild.

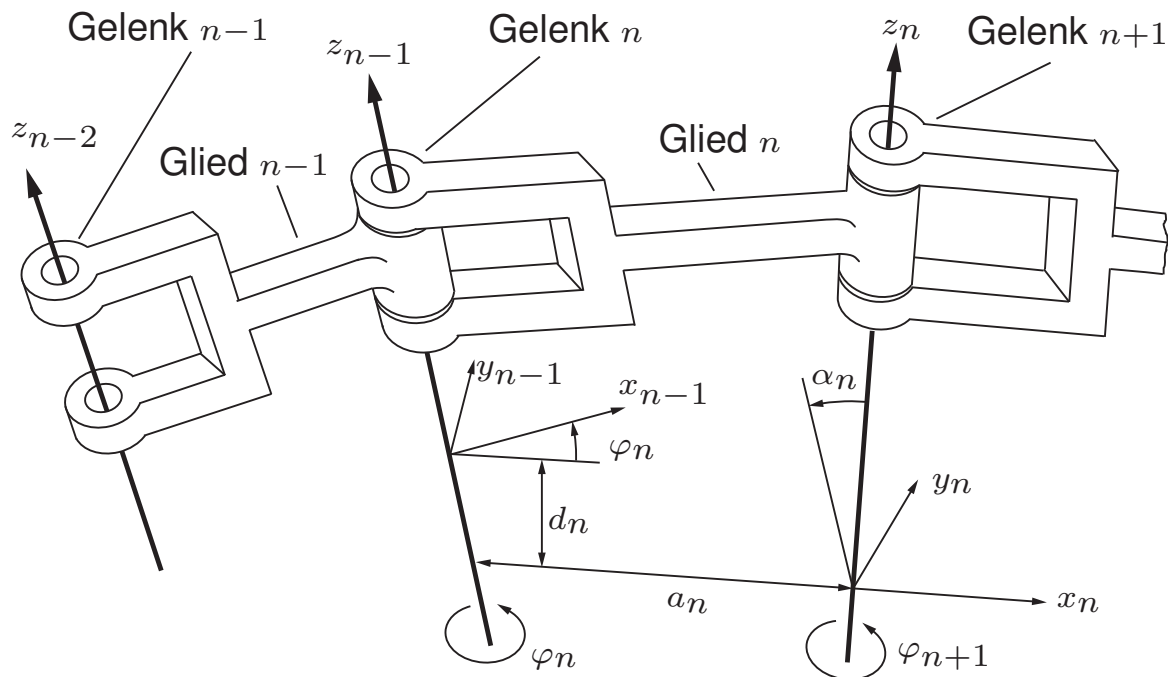


Die Rotation kann nun aus der Kombination, d.h. der Hintereinanderfolge der drei Basisrotationen um die Achsen x , y , z erfolgen. Man erhält als Transformationsmatrix:

$$\underline{T}_{rot} = \underline{T}_{rot}(z, \varphi) \cdot \underline{T}_{rot}(y, \theta) \cdot \underline{T}_{rot}(x, \phi)$$

$$= \begin{bmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta \sin \phi - \sin \varphi \cos \phi & \cos \varphi \sin \theta \cos \phi + \sin \varphi \sin \phi & 0 \\ \sin \varphi \cos \theta & \sin \varphi \sin \theta \sin \phi + \cos \varphi \cos \phi & \sin \varphi \sin \theta \cos \phi - \cos \varphi \sin \phi & 0 \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Mit den obigen geometrischen Betrachtungen ist das Rüstzeug vorhanden, um die Positionen von Robotergliedern bezüglich anderer Roboterglieder oder dem Basiskoordinatensystem des Roboters zu berechnen. Dazu betrachtet man folgenden Ausschnitt aus einer Robotergliedkette:



Die im Bild eingezeichneten Größen $\varphi_n, d_n, a_n, \alpha_n$ bezeichnet man als Denavit-Hartenberg-Parameter. Man kann nun mittels Rotationen und Translationen das Koordinatensystem des n -ten Gliedes auf das Koordinatensystem des $(n-1)$ -ten Gliedes beziehen. Also das Koordinatensystem (x_n, y_n, z_n) in $(x_{n-1}, y_{n-1}, z_{n-1})$ -Koordinaten beschreiben. Hierzu sind folgende Transformationen auszuführen:

1. Rotation mit φ_n um die z_{n-1} -Achse,
2. Translation um d_n entlang der z_{n-1} -Achse,
3. Translation um a_n entlang der x_n -Achse,
4. Rotation mit Torsionswinkel α_n um die x_n -Achse.

Die Winkel α_i werden ermittelt, indem man die z_{i-1} -Achse um die x_i -Achse dreht, bis Deckungsgleichheit mit der z_i -Achse hergestellt ist.

Dementsprechend ergibt sich für die homogene Transformationsmatrix \underline{A}_n , die die Position und Orientierung des Koordinatensystems des n -ten Gliedes bezüglich des Koordinatensystems des $(n - 1)$ -ten Gliedes beschreibt:

$$\underline{A}_n = \underline{T}_{rot}(z_{n-1}, \varphi_n) \underline{T}_{trans}(0, 0, d_n) \underline{T}_{trans}(a_n, 0, 0) \underline{T}_{rot}(x_n, \alpha_n).$$

Es gilt dann:

$$\underline{A}_n = \begin{bmatrix} \cos \varphi_n & -\sin \varphi_n \cos \alpha_n & \sin \varphi_n \sin \alpha_n & a_n \cos \varphi_n \\ \sin \varphi_n & \cos \varphi_n \cos \alpha_n & -\cos \varphi_n \sin \alpha_n & a_n \sin \varphi_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

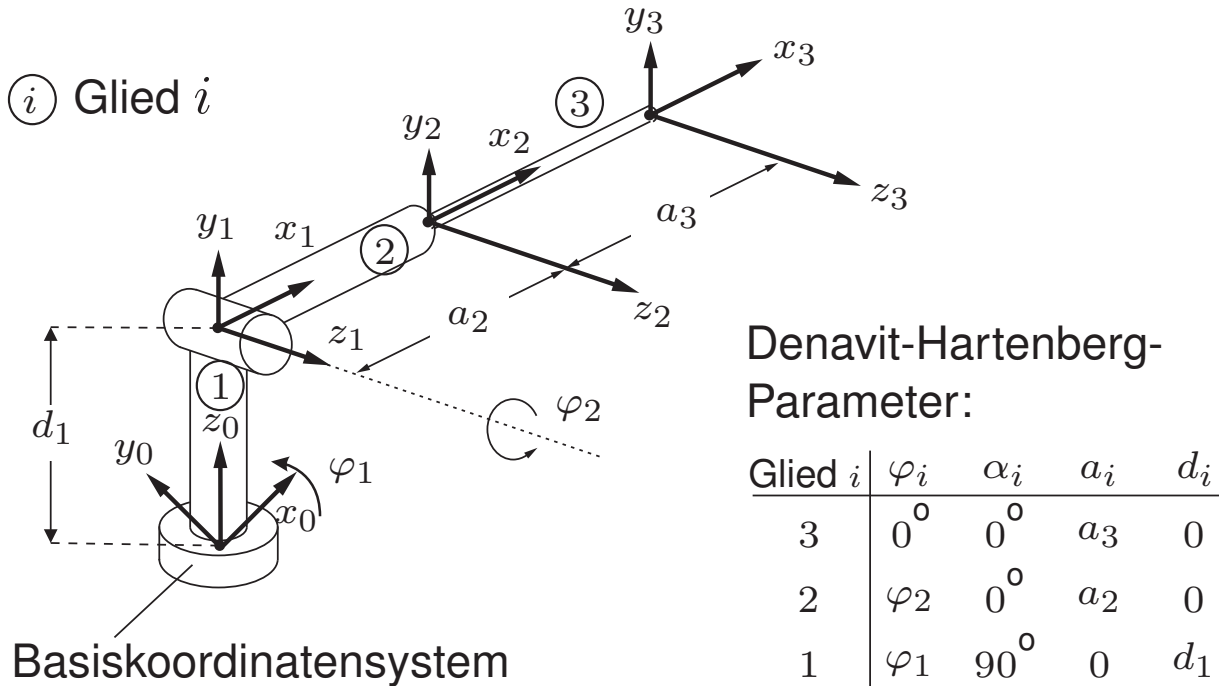
Möchte man nun ein Glied n – genauer sein Koordinatensystem – bezüglich des Basiskoordinatensystems des Roboters beschreiben, so muß man es zuerst mittels \underline{A}_n auf das Koordinatensystem des $(n - 1)$ -ten Gliedes beziehen. Dann mittels \underline{A}_{n-1} auf das des $(n - 2)$ -ten Gliedes und so weiter, bis man beim ersten Glied angelangt ist. Die entsprechende Koordinatendarstellung resultiert aus

$${}^0\underline{T}_n = \underline{A}_1 \cdot \underline{A}_2 \cdot \dots \cdot \underline{A}_n.$$

Entsprechend kann jedes Glied n bezüglich einem vor ihm liegenden Glied m beschrieben werden:

$${}^m\underline{T}_n = \underline{A}_{m+1} \cdot \underline{A}_{m+2} \cdot \dots \cdot \underline{A}_n.$$

Als einfaches Beispiel wird ein Roboter mit drei Freiheitsgraden betrachtet. Dabei ist das Koordinatensystem mit dem Index 0 das Basiskoordinatensystem.



Man erhält für die Matrizen A mit obigen Parametern:

$$\underline{A}_1 = \begin{bmatrix} \cos \varphi_1 & 0 & \sin \varphi_1 & 0 \\ \sin \varphi_1 & 0 & -\cos \varphi_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \underline{A}_3 = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\underline{A}_2 = \begin{bmatrix} \cos \varphi_2 & -\sin \varphi_2 & 0 & a_2 \cos \varphi_2 \\ \sin \varphi_2 & \cos \varphi_2 & 0 & a_2 \sin \varphi_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Mittels der durchgeführten Betrachtungen können nun die Ursprungskoordinaten und die Einheitsvektoren des Greiferkoordinatensystems bezüglich der vorangehenden Gelenkkoordinatensysteme bzw. des Basiskoordinatensystems beschrieben werden.

Für die Beschreibung der Greiferkoordinaten bezüglich des Koordinatensystems von Gelenk 2 gilt:

$${}^2T_3 = \underline{A}_3 = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Für die Beschreibung der Greiferkoordinaten bezüglich des Koordinatensystems von Gelenk 1 gilt:

$${}^1T_3 = \underline{A}_2 \cdot \underline{A}_3 = \begin{bmatrix} \cos \varphi_2 & -\sin \varphi_2 & 0 & (a_3 + a_2) \cos \varphi_2 \\ \sin \varphi_2 & \cos \varphi_2 & 0 & (a_3 + a_2) \sin \varphi_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Und schließlich erhält man für die Beschreibung der Greiferkoordinaten in Basiskoordinaten:

$$\begin{aligned} {}^0T_3 &= \underline{A}_1 \cdot \underline{A}_2 \cdot \underline{A}_3 \\ &= \begin{bmatrix} \cos \varphi_1 \cos \varphi_2 & -\cos \varphi_1 \sin \varphi_2 & \sin \varphi_1 & (a_3 + a_2) \cos \varphi_1 \cos \varphi_2 \\ \sin \varphi_1 \cos \varphi_2 & -\sin \varphi_1 \sin \varphi_2 & -\cos \varphi_1 & (a_3 + a_2) \sin \varphi_1 \cos \varphi_2 \\ \sin \varphi_2 & \cos \varphi_2 & 0 & (a_3 + a_2) \sin \varphi_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Die Matrizen \underline{A}_i sind – wenn jedes Gelenk einen Freiheitsgrad realisiert – entweder von einem Winkel φ_i oder einer Translationsstrecke d_i abhängig. Bezeichnet man φ_i und d_i allgemeiner mit q_i , so gilt

$$\underline{A}_i(q_i).$$

Die Position und Orientierung des Greifers, die durch das Koordinatensystem 0T_n gegeben sind, hängen also nichtlinear von den Gelenkwinkeln bzw. -translationen q_i ab:

$${}^0T_n(\underline{q}) = \underline{A}_1(q_1) \cdot \underline{A}_2(q_2) \cdot \dots \cdot \underline{A}_n(q_n). \quad (1)$$

Sind die Größen q_i bekannt, können Position und Orientierung des Greifers 0T_n berechnet werden. Dies nennt man

- Vorwärtstransformation.

Oftmals möchte man allerdings den umgekehrten Weg gehen und für eine vorgegebene Greiferlage I_n die erforderlichen Gelenkwinkel und -translationen q_i bestimmen. Dies nennt man

- Rückwärtstransformation.

Hierzu existieren zwei Lösungsmöglichkeiten für das nichtlineare Gleichungssystem (1):

- numerische iterative Lösung,
- analytische Lösung.

Im ersten Fall kann man z.B. das Newton-Verfahren verwenden.

Im zweiten Fall folgt man dem Vorschlag von R. P. C. Paul (siehe Literaturverzeichnis). Man multipliziert linksseitig mit der Inversen von $A_1(q_1)$:

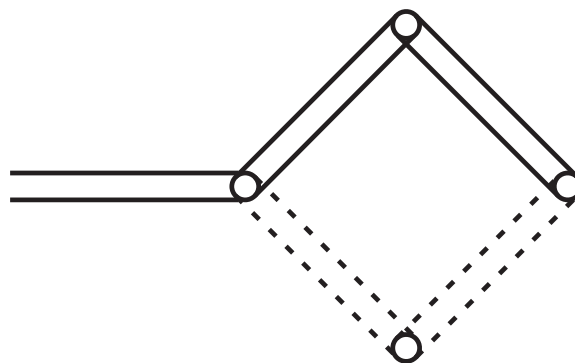
$$\underline{A}_1^{-1}(q_1) \cdot {}^0T_n = \underline{A}_2(q_2) \cdot \dots \cdot \underline{A}_n(q_n). \quad (1)$$

Die linke Seite hängt nur von q_1 ab. Man sucht nun auf der rechten ein nicht von q_2, \dots, q_n abhängiges Matrixelement. In der Regel existiert ein solches, und man kann q_1 auch explizit angeben als Lösung einer nichtlinearen Gleichung mit einer Unbekannten. Die linke Seite in Gl. (1) ist dann als gegeben voraussetzbar. Nun multipliziert man linksseitig mit $A_2^{-1}(q_2)$ und erhält:

$$\underline{A}_2^{-1}(q_2) \cdot \underline{A}_1^{-1}(q_1) \cdot {}^0T_n = \underline{A}_3(q_3) \cdot \dots \cdot \underline{A}_n(q_n).$$

Auch hier kann man wie im Fall von q_1 den Wert q_2 explizit errechnen. Dieses Vorgehen setzt man fort, bis alle Parameter q_i ermittelt sind.

Schwierigkeiten bei dieser Vorgehensweise können durch mehrere Lösungen auftreten. Dann sind die Gelenkstellungen des Roboters nicht eindeutig. Das Bild illustriert diesen Fall.



Die Matrix

$${}^0T_n = \begin{bmatrix} \underline{R} & \underline{t} \\ \underline{0} & 1 \end{bmatrix}$$

beschreibt, wie gesagt, die Lage und Position des Robotergreifers (bzw. -gelenks) in Basiskoordinaten. Dabei werden 4×4 Matricelemente benutzt für die Position \underline{t} des Greifers und seine Orientierung \underline{R} im Raum. Prinzipiell ist es auch ausreichend, neben den drei Koordinaten der Position \underline{t} noch drei Winkel für die Orientierung des Greifers zu nutzen – anstelle des Dreibeins \underline{R} .

Die drei Koordinaten von \underline{t} werden im Weiteren mit p_1, p_2, p_3 bezeichnet und die drei Winkel für die Orientierung des Greifers mit p_4, p_5, p_6 . Zusammengefaßt läßt sich dann der Zusammenhang zwischen Gelenkwinkeln bzw. -auslenkungen q_i mit den Basiskoordinaten des Greifers und seiner Orientierung in Basiskoordinaten als nichtlinearer vektorieller Zusammenhang

$$\boxed{\underline{p} = \underline{f}(\underline{q})}$$

bzw.

$$\underline{q} = \underline{f}^{-1}(\underline{p})$$

darstellen.

Die Kinematik eines Roboters ist somit berechnet. Im nächsten Abschnitt wird seine Dynamik betrachtet.

ABSCHNITT 3.3

Dynamisches Modell

Im vorherigen Abschnitt wurden ausschließlich die geometrischen Verhältnisse des Roboters betrachtet. Im Weiteren soll nun seine Dynamik berechnet werden. Also der Zusammenhang zwischen den Geschwindigkeiten von Robotersegmenten oder -punkten und den Antriebskräften bzw. -momenten. Kräfte und Momente bezeichnet man dabei verallgemeinert mit τ_ν .

Berechnet werden kann der Zusammenhang z.B. mit der Methode von Lagrange. Dabei berechnet man zuerst die kinetische Energie E_{kin} und die potentielle Energie E_{pot} des Roboters. Aus ihrer Differenz

$$\lambda = E_{kin} - E_{pot}$$

können über die Lagrange-Gleichung

$$\tau_\nu = \frac{d}{dt} \cdot \frac{\partial \lambda}{\partial \dot{q}_\nu} - \frac{\partial \lambda}{\partial q_\nu}$$

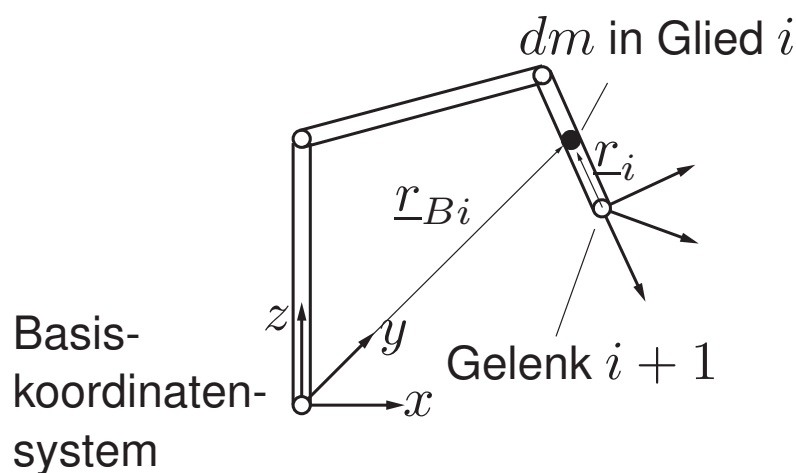
die Kräfte und Momente τ_ν aus den Längen und Winkel q_ν sowie ihren Ableitungen \dot{q}_ν bestimmt werden. Die Kräfte bzw. Momente τ_ν sind extern angreifende Kräfte bzw. Momente.

Um die Lagrange-Gleichung nun für die Berechnung der Roboterdynamik nutzen zu können, muß zuerst die kinetische Energie des Roboters ermittelt werden.

Zu diesem Zweck betrachtet man einen beliebigen Massenpunkt dm des Gliedes i mit der Position

$$\underline{r}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix},$$

die durch das im Gelenk $i + 1$ gelegene Koordinatensystem gegeben ist. Dazu nachfolgendes Bild.



Über die Transformation 0T_i kann dann die Position von dm in Basiskoordinaten \underline{r}_{Bi} beschrieben werden:

$$\underline{r}_{Bi} = {}^0T_i(q_i) \cdot \underline{r}_i.$$

Bewegt sich der Roboter, so ändern sich die Gelenkwinkel \underline{q}_ν und damit auch die Position \underline{r}_{Bi} des Punktes dm in Basiskoor-
dinaten. Der Vektor \underline{r}_i bleibt als Relativposition konstant. Somit
kann der Geschwindigkeitsvektor $\dot{\underline{r}}_{Bi}$ berechnet werden gemäß

$$\dot{\underline{r}}_{Bi} = \sum_{j=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_j} \dot{q}_j \cdot \underline{r}_i. \quad (1)$$

Die kinetische Energie des Gliedes i ergibt sich, indem man
über alle Massenpunkte dm des Gliedes i integriert:

$$E_{kin,i} = \frac{1}{2} \int_{\text{Glieder } i} |\dot{\underline{r}}_{Bi}|^2 dm.$$

Um dieses Integral zu berechnen, bedient man sich eines Tricks.
Und zwar schreibt man den Integranden in der Form

$$\begin{aligned} |\dot{\underline{r}}|^2 &= \dot{\underline{r}}^T \dot{\underline{r}} = \text{spur} \begin{bmatrix} \dot{x}^2 & \dot{x}\dot{y} & \dot{x}\dot{z} & 0 \\ \dot{x}\dot{y} & \dot{y}^2 & \dot{y}\dot{z} & 0 \\ \dot{z}\dot{x} & \dot{z}\dot{y} & \dot{z}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \text{spur} \left(\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 0 \end{bmatrix} (\dot{x} \ \dot{y} \ \dot{z} \ 0) \right) = \text{spur}(\dot{\underline{r}} \cdot \dot{\underline{r}}^T) \end{aligned} \quad (2)$$

Man beachte: $\underline{r}^T = (x, y, z, 1)$ und $\dot{\underline{r}}^T = (\dot{x}, \dot{y}, \dot{z}, 0)$. Die
Spur einer Matrix ist die Summe ihrer Diagonalelemente. Man
setzt nun Gl. (1) in Gl. (2) ein.

So ergibt sich

$$\begin{aligned}
 |\dot{\underline{r}}_{Bi}|^2 &= \text{spur} \left(\left[\sum_{j=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_j} \dot{q}_j \underline{r}_i \right] \left[\sum_{k=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_k} \dot{q}_k \underline{r}_i \right]^T \right) \\
 &= \text{spur} \left(\sum_{j=1}^i \sum_{k=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_j} \cdot \underline{r}_i \underline{r}_i^T \cdot \frac{\partial^0 \underline{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right).
 \end{aligned}$$

Setzt man obigen Term in

$$E_{kin,i} = \frac{1}{2} \int_{\text{Glied } i} |\dot{\underline{r}}_{Bi}|^2 dm.$$

ein, so ergibt sich

$$E_{kin,i} = \frac{1}{2} \text{spur} \left(\sum_{j=1}^i \sum_{k=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_j} \int_{\text{Glied } i} \underline{r}_i \underline{r}_i^T dm \frac{\partial^0 \underline{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right).$$

Der darin enthaltene Term

$$\int_{\text{Glied } i} \underline{r}_i \underline{r}_i^T dm$$

kann durch die Trägheitsmomente, die Deviationsmomente, die Masse und die Schwerpunktskoordinaten des Gliedes i beschrieben werden.

Es gilt nämlich

$$\underline{r}_i \underline{r}_i^T = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} [x_i, y_i, z_i, 1] = \begin{bmatrix} x_i^2 & x_i y_i & x_i z_i & x_i \\ y_i x_i & y_i^2 & y_i z_i & y_i \\ z_i x_i & z_i y_i & z_i^2 & z_i \\ x_i & y_i & z_i & 1 \end{bmatrix}$$

und die Integrale über einige der Elemente dieser Matrix sind direkt die Deviationsmomente, Masse und Schwerpunktskoordinaten

$$I_{xy}^i = \int_{\text{Glied } i} x_i y_i dm,$$

$$m_i = \int_{\text{Glied } i} dm \quad \text{und} \quad \bar{x}_i \cdot m_i = \int_{\text{Glied } i} x_i dm,$$

wobei \bar{x}_i die Schwerpunktskoordinate in x_i -Richtung angibt. Durch die Trägheitsmomente, z. B.

$$I_{xx}^i = \int_{\text{Glied } i} (y_i^2 + z_i^2) dm,$$

lassen sich die restlichen Terme folgendermaßen ausdrücken:

$$\begin{aligned} \int_{\text{Glied } i} x_i^2 dm &= -\frac{1}{2} \int_{\text{Glied } i} (y_i^2 + z_i^2) dm \\ &\quad + \frac{1}{2} \int_{\text{Glied } i} (x_i^2 + z_i^2) dm \\ &\quad + \frac{1}{2} \int_{\text{Glied } i} (x_i^2 + y_i^2) dm \\ &= \frac{1}{2} (-I_{xx}^i + I_{yy}^i + I_{zz}^i) \end{aligned}$$

Mit den entsprechenden Termen für y^2 und z^2 erhält man:

$$\underline{I}_i = \int_{\text{Glieder } i} \underline{r}_i \underline{r}_i^T dm$$

$$= \begin{bmatrix} \frac{-I_{xx}^i + I_{yy}^i + I_{zz}^i}{2} & I_{xy}^i & I_{xz}^i & \bar{x}_i \cdot m_i \\ I_{xy}^i & \frac{+I_{xx}^i - I_{yy}^i + I_{zz}^i}{2} & I_{yz}^i & \bar{y}_i \cdot m_i \\ I_{xz}^i & I_{yz}^i & \frac{+I_{xx}^i + I_{yy}^i - I_{zz}^i}{2} & \bar{z}_i \cdot m_i \\ \bar{x}_i \cdot m_i & \bar{y}_i \cdot m_i & \bar{z}_i \cdot m_i & m_i \end{bmatrix}.$$

Nun setzt man diese Trägheitsmatrix in die schon berechnete kinetische Energie

$$E_{kin,i} = \frac{1}{2} \text{spur} \left(\sum_{j=1}^i \sum_{k=1}^i \frac{\partial^0 \underline{T}_i}{\partial q_j} \int_{\text{Glieder } i} \underline{r}_i \underline{r}_i^T dm \frac{\partial^0 \underline{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right)$$

ein und erhält nach Summation über alle n Glieder

$$E_{kin} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{spur} \left(\frac{\partial^0 \underline{T}_i}{\partial q_j} \underline{I}_i \frac{\partial^0 \underline{T}_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k.$$

Berücksichtigt werden muß noch die kinetische Energie der Stellantriebe in den n Gelenken:

$$E_{kin,stell} = \sum_{i=1}^n \frac{1}{2} I_{stell,i} \cdot \dot{q}_i^2.$$

Dabei sind die Werte $I_{stell,i}$ die Trägheitsmomente des jeweiligen Stellantriebs.

So erhält man also für die gesamte kinetische Energie des Roboters

$$\begin{aligned} E_{kin,ges} &= E_{kin} + E_{kin,stell} \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{spur} \left(\frac{\partial^0 \underline{T}_i}{\partial q_j} \underline{I}_i \frac{\partial^0 \underline{T}_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k \\ &\quad + \frac{1}{2} \sum_{i=1}^n I_{stell,i} \cdot \dot{q}_i^2. \end{aligned}$$

Für die potentielle Energie ergibt sich mit den Schwerpunktsvektoren $\underline{\bar{r}}_{Bi}$ des i -ten Gliedes in Basiskoordinaten und dem Gravitationsvektor

$$\underline{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \\ 0 \end{bmatrix};$$

der ebenfalls in Basiskoordinaten angegeben wird:

$$E_{pot} = - \sum_{i=1}^n m_i \underline{g}^T \underline{\bar{r}}_{Bi} = - \sum_{i=1}^n m_i \underline{g}^T {}^0T_i(q) \underline{\bar{r}}_i. \quad (1)$$

Anmerkung: Das Vorzeichen in (1) ist so gewählt, dass die potentielle Energie in Richtung der Gravitationskraft abnimmt.

Nun kann man

$$\lambda = E_{kin} - E_{pot}$$

in die Lagrange-Gleichung

$$\tau_\nu = \frac{d}{dt} \cdot \frac{\partial \lambda}{\partial \dot{q}_\nu} - \frac{\partial \lambda}{\partial q_\nu}$$

einsetzen.

So erhält man nach einigen Umformungen

$$\begin{aligned}
 \tau_\nu = & \sum_{j=\nu}^n \sum_{k=1}^j \text{spur} \left[\frac{\partial^0 \underline{T}_j}{\partial q_k} \underline{I}_j \frac{\partial^0 \underline{T}_j^T}{\partial q_\nu} \right] \cdot \ddot{q}_k + I_{stell,\nu} \ddot{q}_\nu \\
 & + \sum_{j=\nu}^n \sum_{k=1}^j \sum_{l=1}^j \text{spur} \left(\frac{\partial^2 {}^0 \underline{T}_j}{\partial q_k \partial q_l} \underline{I}_j \frac{\partial^0 \underline{T}_j^T}{\partial q_k} \right) \dot{q}_k \dot{q}_l \\
 & - \sum_{j=\nu}^n m_j \underline{g}^T \frac{\partial^0 \underline{T}_j}{\partial q_\nu} \underline{\bar{r}}_j.
 \end{aligned} \tag{1}$$

Dabei enthält

- der erste Summand die Trägheitsterme,
- der dritte die Zentrifugal- und Coriolis-Kräfte und
- der letzte die Gravitationskräfte.

Vereinfacht läßt sich Gl. (1) auch in der Form

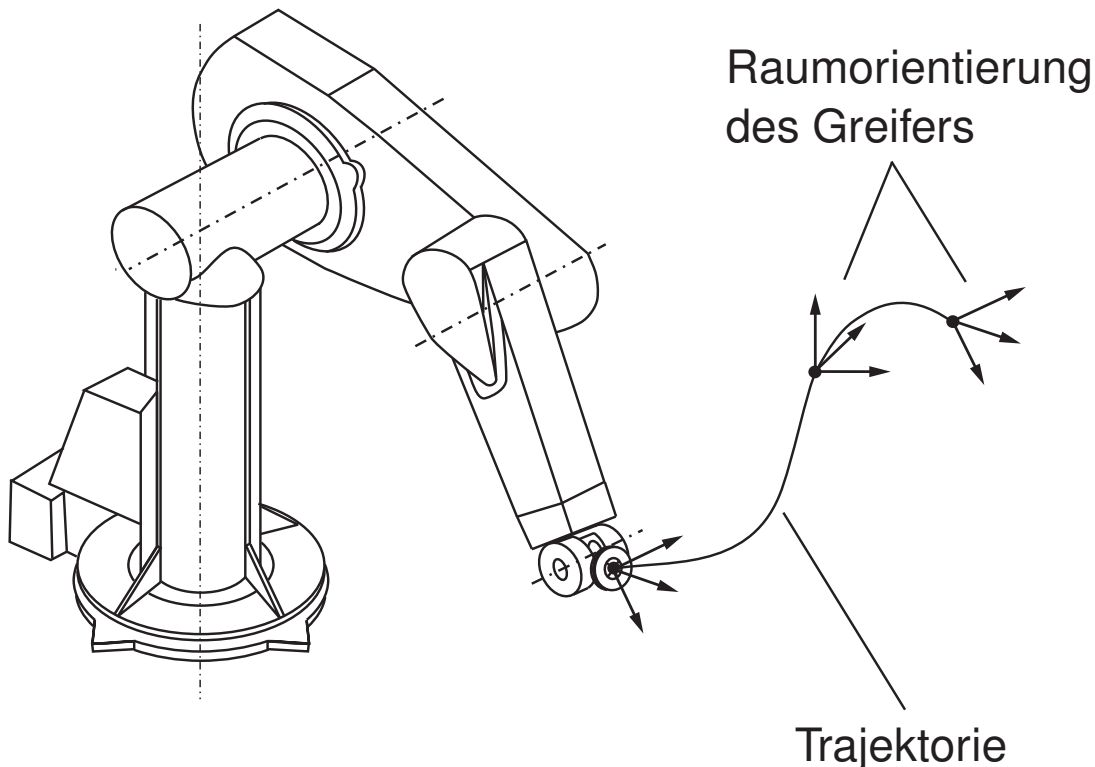
$$\boxed{\underline{H}(\underline{q}) \ddot{\underline{q}} = \underline{\tau} - \underline{h}(\underline{q}, \dot{\underline{q}})}$$

darstellen. Diese Gleichung beschreibt die Roboterdynamik.

ABSCHNITT 3.4

Regelung von Industrierobotern

Ziel einer Roboterregelung ist im Allgemeinen das möglichst exakte Abfahren einer vorgegebenen Trajektorie, entweder um entlang der Trajektorie einen Bearbeitungsvorgang vorzunehmen, oder aber den Robotergriffefer am Endpunkt der Trajektorie zu positionieren:



Man gibt also eine Trajektorie einschließlich Raumorientierung des Greifers

$$\underline{p}_{soll}(t)$$

vor und wird dann über eine Regelung die Kräfte τ_ν so einstellen, dass der Robotergriffefer diese Trajektorie möglichst exakt abfährt.

Für die Berechnung der für diese Aufgabe erforderlichen Kräfte τ_ν und Gelenkpositionen q_i werden dabei die Ergebnisse der beiden vorherigen Abschnitte genutzt.

Die Gelenkwinkel bzw. -auslenkungen q_i werden aus der kinematischen Beziehung

$$\underline{p} = \underline{f}(\underline{q}) \quad \text{bzw.} \quad \underline{q} = \underline{f}^{-1}(\underline{p}) \quad (1)$$

errechnet.

Die benötigten Kräfte und Momente τ_ν ergeben sich aus der Dynamikgleichung zu


$$\underline{\tau} = \underline{H}(\underline{q})\ddot{\underline{q}} + \underline{h}(\underline{q}, \dot{\underline{q}}). \quad (2)$$

Hat man also mittels der Kinematik-Gleichung (1) aus dem Sollverlauf $\underline{p}_{soll}(t)$ den Sollverlauf der Gelenkwinkel bzw. -auslenkungen $\underline{q}_{soll}(t)$ ermittelt, so kann über die Dynamik-Gleichung (2) der Kräfte- und Momentenvektor $\underline{\tau}_{soll}(t)$ ermittelt werden.

Dabei ist es erforderlich, $\dot{\underline{q}}$ und $\ddot{\underline{q}}$ zu bestimmen. Es gilt

$$\begin{aligned} \dot{\underline{p}} &= \frac{\partial \underline{f}(\underline{q})}{\partial \underline{q}} \cdot \dot{\underline{q}} = \underline{J}(\underline{q}) \cdot \dot{\underline{q}} \\ \ddot{\underline{p}} &= \dot{\underline{J}}(\underline{q}) \cdot \dot{\underline{q}} + \underline{J}(\underline{q}) \cdot \ddot{\underline{q}} \end{aligned}$$

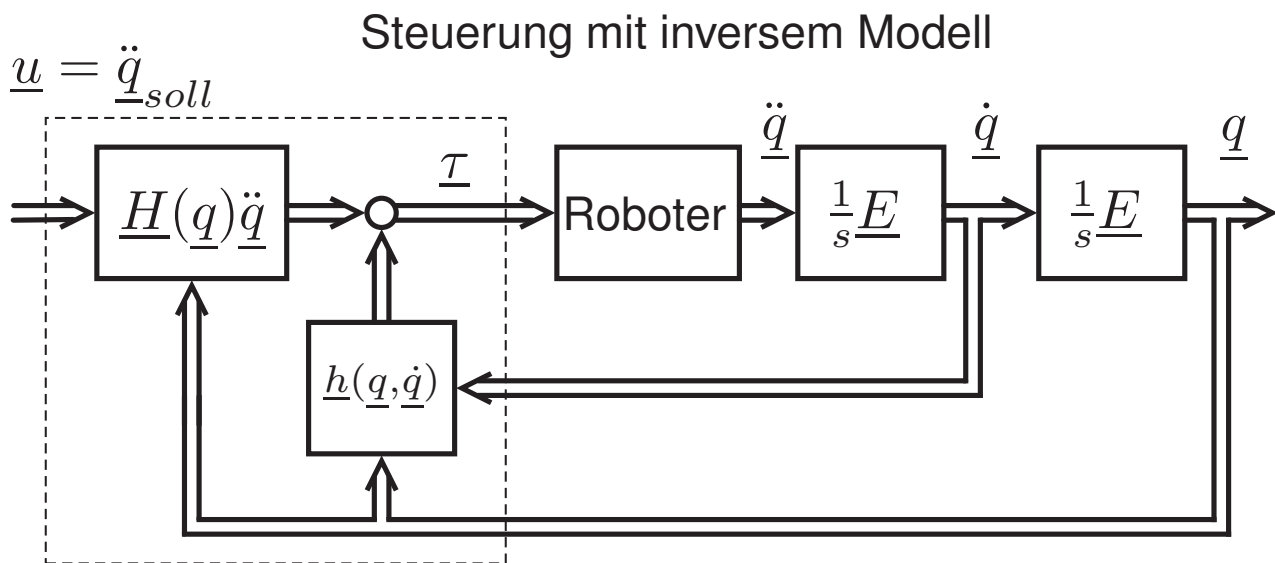
Jacobi-Matrix



und somit

$$\begin{aligned} \dot{\underline{q}} &= \underline{J}^{-1}(\underline{q}) \cdot \dot{\underline{p}} \\ \ddot{\underline{q}} &= \underline{J}^{-1}(\underline{q}) \cdot (\ddot{\underline{p}} - \dot{\underline{J}}(\underline{q}) \cdot \dot{\underline{q}}). \end{aligned}$$

Die nun einfachste Möglichkeit, den Greifer der vorgegebenen Trajektorie $\underline{p}_{soll}(t)$ folgen zu lassen, ist eine Steuerung. Hierzu schaltet man die errechneten Kräfte und Momentenverläufe $\tau_{soll}(t)$ auf die Roboterstellantriebe. Nachfolgendes Bild zeigt die entsprechende Struktur.

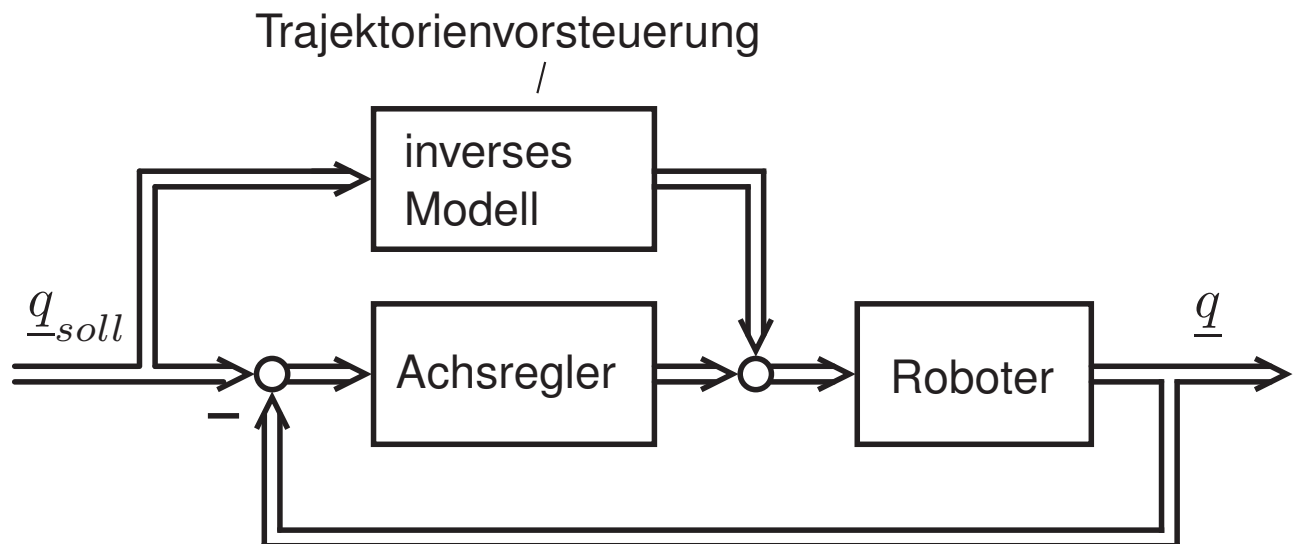


\underline{E} bezeichnet die Einheitsmatrix.

Die Gelenkstellungen \underline{q} und die Veränderungsgeschwindigkeiten $\underline{\dot{q}}$ werden im Allgemeinen messtechnisch erfaßt.

Eine Steuerung nach obiger Struktur hat jedoch – wie jede Steuerung – den Nachteil, dass Störungen nicht ausgeglichen werden. Des Weiteren ist die obige Steuerung für den Idealfall ausgelegt. Das heißt, sie wird nur die gewünschten Ergebnisse liefern, wenn das Modell den Roboter exakt beschreibt. Aufgrund der nicht modellierten Reibungen, Lose, Strukturschwingungen etc. ist das jedoch nicht der Fall.

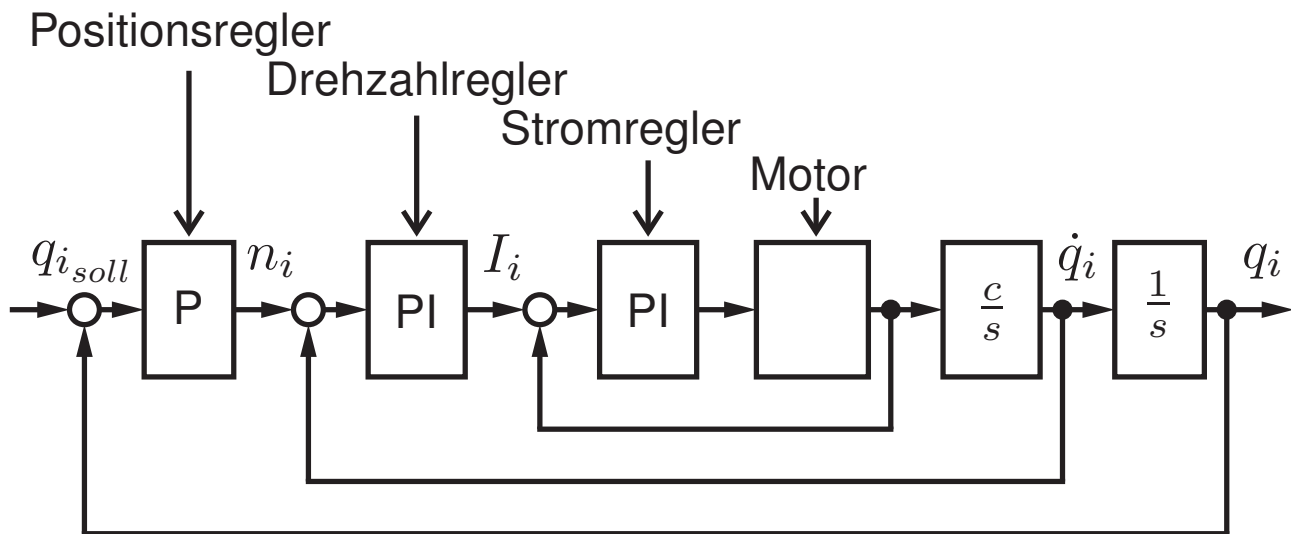
Aus diesem Grund verwendet man zusätzlich zur Steuerung mit dem inversen Modell eine überlagerte Achsregelung gemäß der abgebildeten Struktur.



Die Trajektorienvorsteuerung mittels des inversen Modells leistet bereits eine gute Bahnfolgung durch den Roboter. Die auf die Ausregelung von Störungen optimierten Achsregler eliminieren dann noch kleinere Abweichungen, die z.B. auch auf Modellungenauigkeiten basieren.

Die einzelnen Achsregler haben die Aufgabe, die einzelnen Gelenke auf die vorgegebenen Sollwerte $q_{soll,i}$ einzustellen. Entworfen werden sie im Allgemeinen als Einzelregler, die die jeweilige Achse positionieren. Die Verkopplung der einzelnen Achsen und damit auch der einzelnen Achsregelkreise wird also bei ihrem Entwurf nicht berücksichtigt.

Achsregler werden im Allgemeinen als Kaskadenregler ausgelegt. Dabei wird für den Stromregelkreis ein PI-Regler verwendet. Für den Drehzahlregler nutzt man einen PI-Regler und für den überlagerten Positionsregler einen P-Regler. Nachfolgendes Bild zeigt die Struktur des Regelkreises.

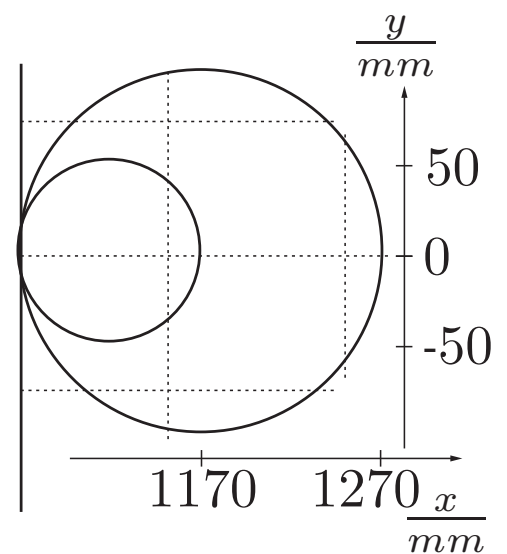
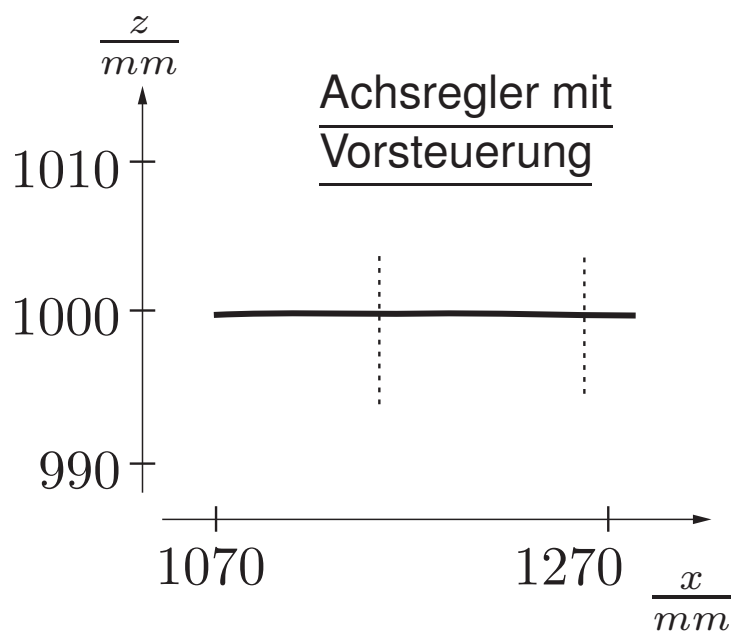
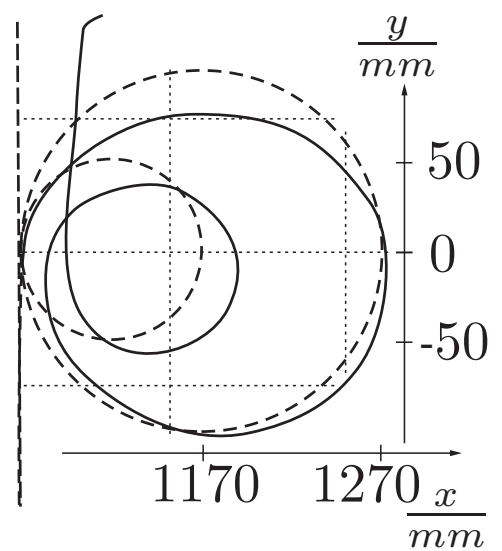
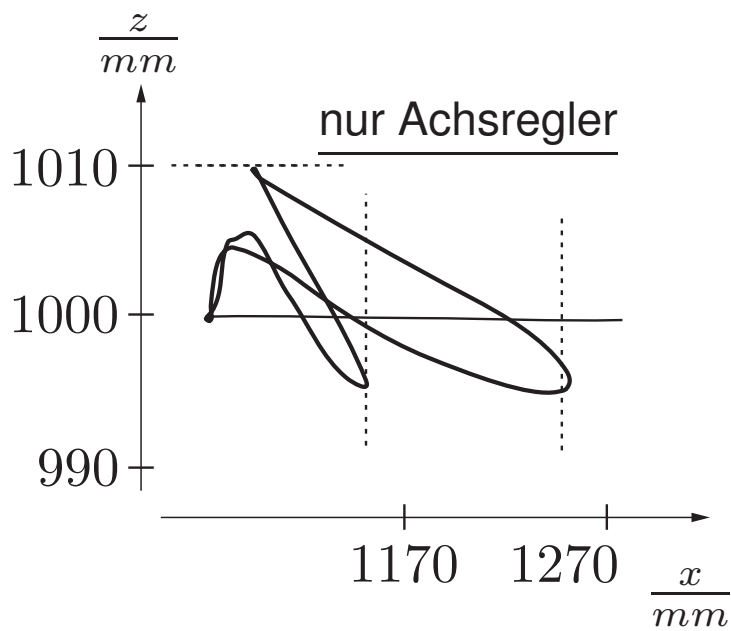
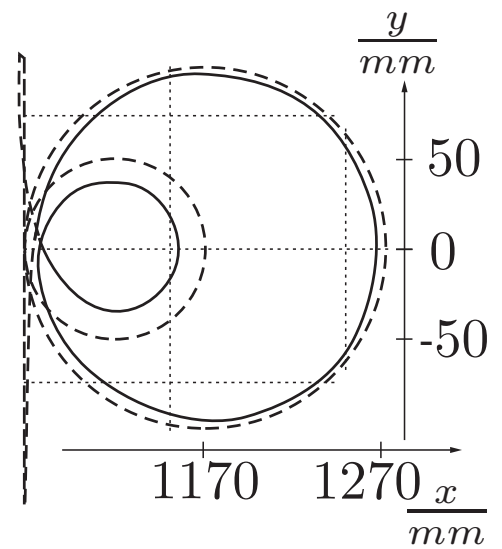
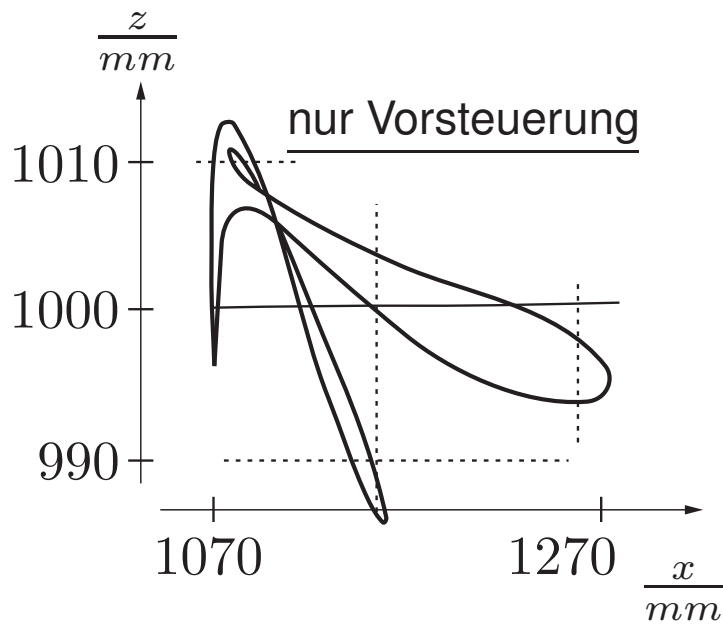


Neben dem schon beschriebenen Steuerungskonzept und der Regelung mit Vorsteuerung über das inverse Robotermodell, kann als eine dritte Möglichkeit auch die Regelung nur über die Achsregler erfolgen. Die letztere Variante wird aufgrund ihres einfachen Entwurfs und ihrer einfachen Realisierbarkeit oft in industriellen Roboteranwendungen eingesetzt.

Die nachfolgenden Bahnverläufe liefern einen Vergleich der drei Varianten:

- Steuerung mit inversem Modell,
- Achsregelung ohne Vorsteuerung,
- Achsregelung mit Vorsteuerung über inverses Modell.

Die Verläufe basieren auf Simulationen. Als Roboter wurde ein KUKA 160/15 angenommen.



Eine Überprüfung der durchgeführten Simulationen anhand eines praktischen Experiments an einem KUKA 160/15-Industrieroboter hat eine gute Übereinstimmung zwischen Simulation und Realität ergeben.

Als Fazit läßt sich festhalten, dass die mit Abstand besten Regelergebnisse mit der Variante

- Achsregler mit Vorsteuerung über das inverse Modell

erzielt werden. Nachteil dieser Variante ist ihr doch recht hoher Realisierungsaufwand, der für die Vorsteuerung erforderlich ist.

Die ungenaue Bahnfolgung im Fall der Achsregler ohne Vorsteuerung über das inverse Modell ist im Wesentlichen auf die große

- nichtlineare Verkopplung der einzelnen Achsen

zurückzuführen. Diese Verkopplung wird beim Entwurf der einzelnen Achsregler nicht berücksichtigt.

Für den Fall, dass zusätzlich zu den Achsreglern noch die Vorsteuerung benutzt wird, sind die Verkopplungen nicht von so großer Bedeutung. Denn hier müssen die Achsregler nur die (relativ kleine) Differenz ausregeln, die durch die Unterschiede zwischen Modell und realem Roboter entsteht.

Der zuletzt ausgeführte Gedanke legt es nahe, das inverse Modell zur

- Entkopplung und
- Linearisierung

des Roboters einzusetzen, und für diesen entkoppelten und linearisierten Roboter dann einen linearen Regler zu entwerfen.

Wie durch das inverse Modell

$$\underline{\tau} = \underline{\hat{H}}(\underline{q})\ddot{\underline{q}}_{soll} + \underline{\hat{h}}(\underline{q}, \dot{\underline{q}}) \quad (1)$$

eine lineare entkoppelte Regelstrecke entsteht, ist einfach zu zeigen.

Anmerkung: $\underline{\hat{H}}$ und $\underline{\hat{h}}$ wurden gegenüber dem realen Roboter mit einem „ $\hat{}$ “ versehen, um die Differenz zwischen Modell und Roboter zu kennzeichnen.

Man setzt (1) in die Dynamik des Roboters

$$\underline{H}(\underline{q})\ddot{\underline{q}} = \underline{\tau} - \underline{h}(\underline{q}, \dot{\underline{q}})$$

ein und erhält

$$\underline{H}(\underline{q})\ddot{\underline{q}} = \underline{\hat{H}}(\underline{q})\ddot{\underline{q}}_{soll} + \underline{\hat{h}}(\underline{q}, \dot{\underline{q}}) - \underline{h}(\underline{q}, \dot{\underline{q}})$$

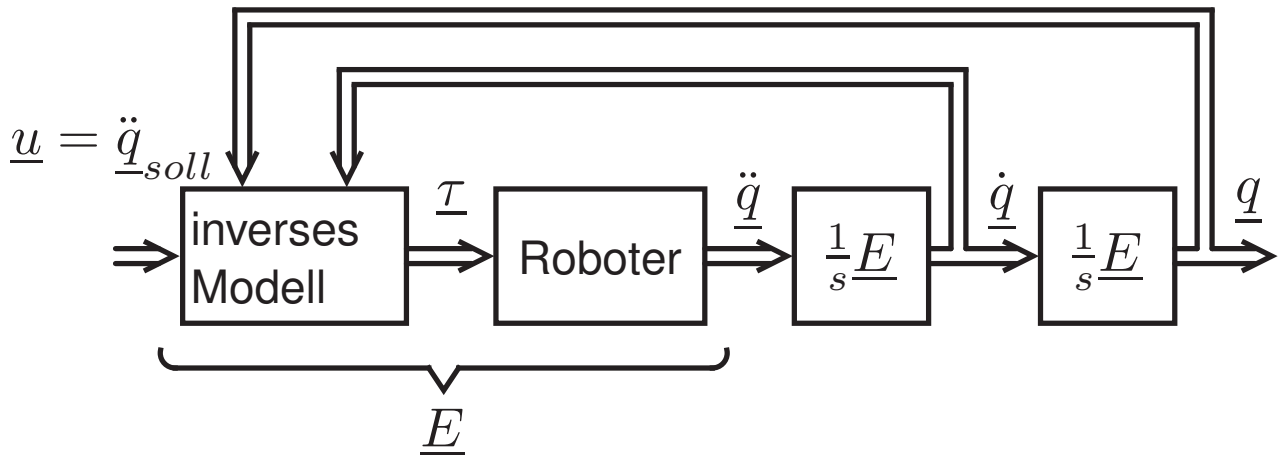
Wurde genau genug modelliert, so gilt

$$\underline{H}(\underline{q}) = \underline{\hat{H}}(\underline{q}) \quad \text{und} \quad \underline{\hat{h}}(\underline{q}, \dot{\underline{q}}) = \underline{h}(\underline{q}, \dot{\underline{q}})$$

und somit

$$\ddot{\underline{q}} = \ddot{\underline{q}}_{soll}.$$

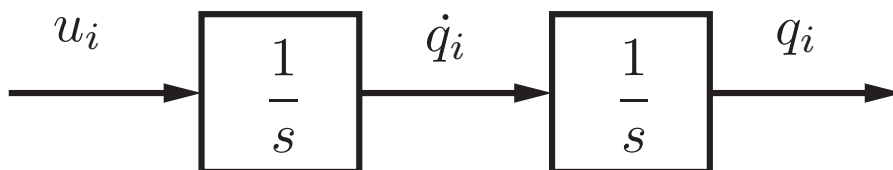
Da die Gelenkpositionen \underline{q} die Regelgrößen bilden, erhält man n lineare, entkoppelte Regelstrecken. Nachfolgendes Bild veranschaulicht dies:



\underline{E} ist die Einheitsmatrix. Man erhält daher als Regelstrecken n entkoppelte Doppelintegrierer

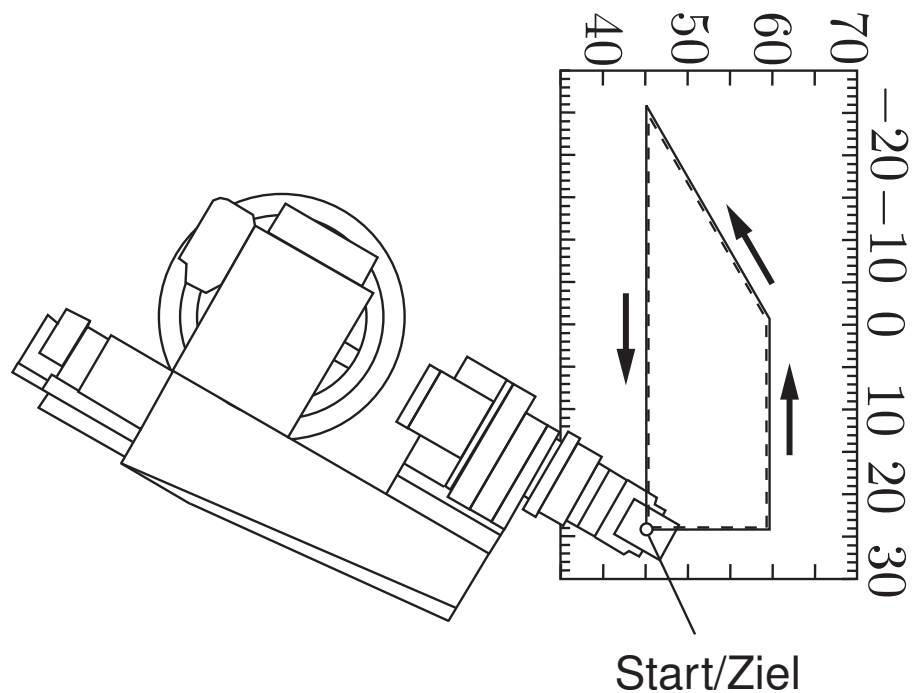
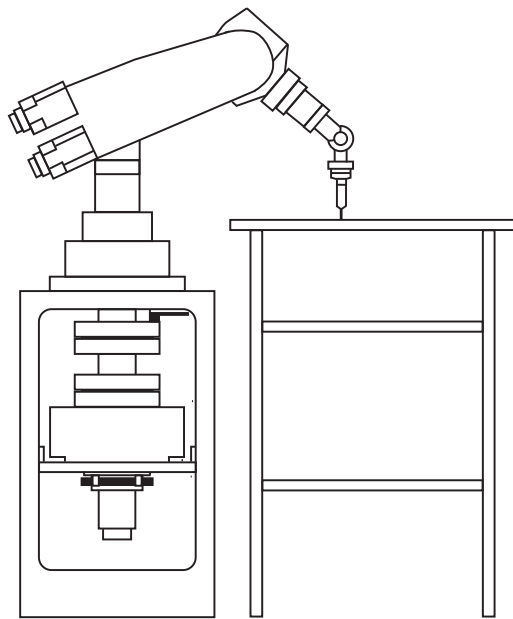
$$\ddot{q}_i = u_i$$

mit der Struktur:



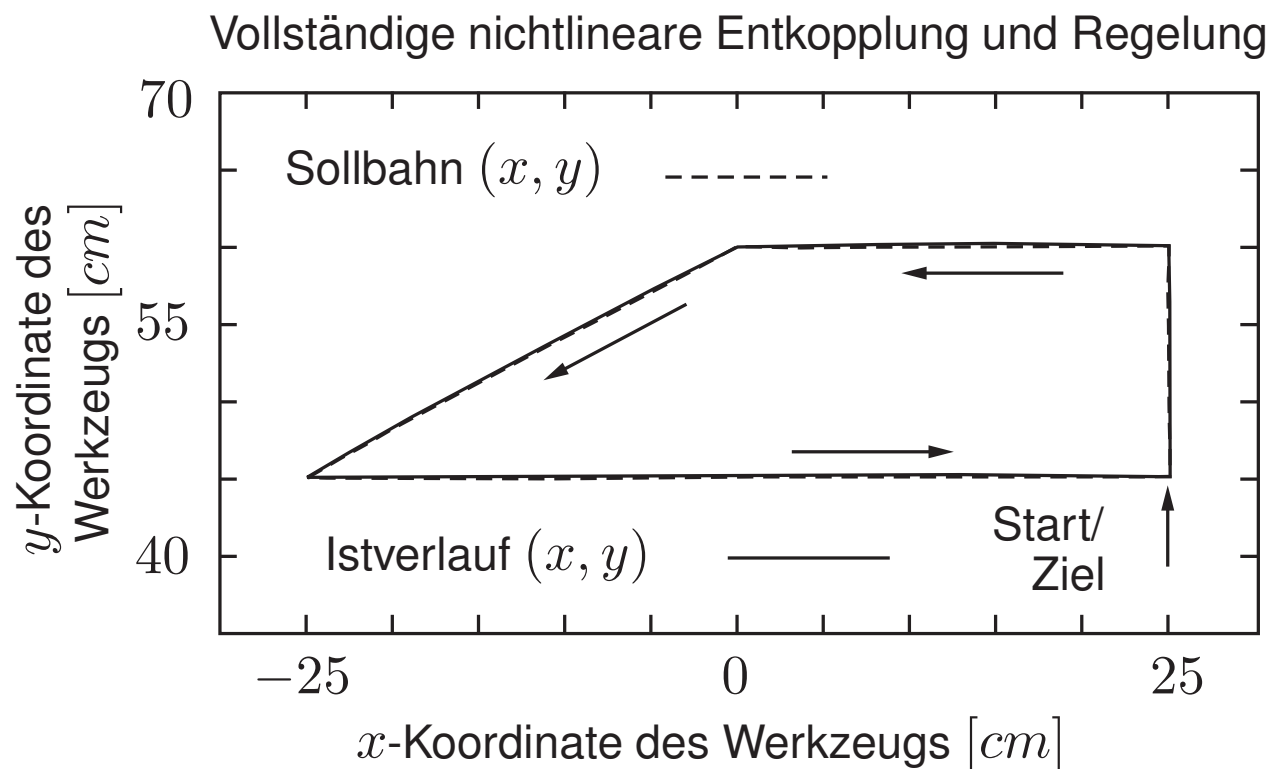
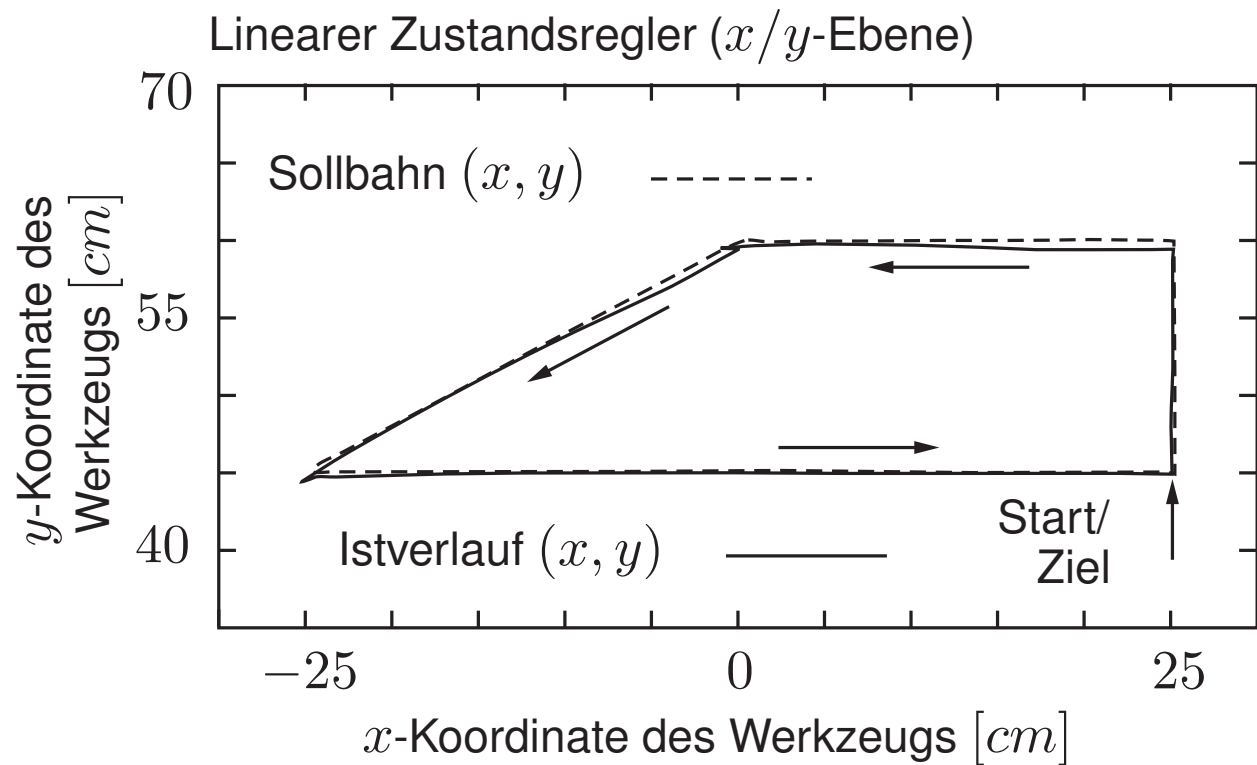
Da \dot{q}_i und q_i messbar sind, kann man nun über einen Zustandsregler die Dynamik des Roboterregelkreises beliebig (innerhalb der zur Verfügung stehenden Stellgrößenbeschränkungen) festlegen.

Die nachfolgenden Simulationen¹ illustrieren die hohe Güte der Regelung mit vollständiger Entkopplung und Linearisierung durch das inverse Modell im Vergleich mit einem linearen Zustandsregler ohne vollständige Entkopplung des Roboters. Die beiden nachfolgenden Bilder zeigen die abzufahrende Bahn:



¹ Bühler, Ch.: Regelung direktangetriebener Roboterachsen am Beispiel eines sechssach-sigen Industrieroboters. Dissertation, 1989.

Die Simulationsergebnisse sind nachfolgend gezeigt:



Man erkennt, dass der lineare Zustandsregler bis zu 5 mm Bahnabweichung aufweist. Für viele Anwendungen ist dieser Regelfehler zu groß. Bei der vollständigen Entkopplung und Regelung sind die Abweichungen deutlich geringer. Von allen betrachteten Roboterregelungen weist sie die höchste Regelgüte auf. Allerdings ist die Achsregelung mit Vorsteuerung durch das inverse Modell fast genauso gut.

Trotz der hohen Regelgüte der Regelung mit vollständiger Entkopplung wird sie in der Praxis nur selten verwendet. Das hat die folgenden Gründe:

1. Die in der Praxis bewährten und einfach einstellbaren Achsregelungen können nicht mehr direkt eingesetzt werden.
2. Modellierungsungenauigkeiten verhindern eine vollständige bzw. optimale Entkopplung.
3. Durch den hohen Online-Rechenaufwand bei der Auswertung des inversen Modells können die Stellbefehle nur verzögert auf die Gelenkantriebe geschaltet werden.

Die am häufigsten eingesetzten Regelungen sind die klassischen Achsregelungen und die Achsregelung mit Vorsteuerung über das inverse Modell.

Literaturhinweise

Literatur zur „Robotik“

- Gorinevsky, D. M.; Formalsky, A. M. und Schneider, A. Y.: *Force Control of Robotic Systems*. CRC Press, 1997
- Asada, H. und Slotine, J.-J. E.: *Robot Analysis and Control*. John Wiley & Sons, 1986
- Paul, R. P. C.: *Robot Manipulators*. Mathematics, Programming and Control, The MIT Press, 1986
- Martin, J. F. und Schoenwaldt, J. S.: *Robotic Sensors*. Encyclopedia of Physical Science and Technology, Vol. 12, Academic Press, 1987, S. 243-261
- Craig, J. J.: *Robot Motion Planning*. Encyclopedia of Applied Physics, Vol.16, VCH Publishers Inc., 1996, S. 505-517
- Vukobratović, M.: *Introduction to Robotics*. Springer, 1989
- Murray, R.-N.; Li, Z. und Sastry, S. S.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994
- Cox, J.-J. und Wilfong, G. T. (Hrsg.): *Autonomous Robot Vehicles*. Springer, 1990
- Latombe, J.-C.: *Robot Motion Planning*. Kluwer Academic Publishers, 1991
- Knieriemen, T.: *Autonome mobile Roboter*. BI, 1992
- Hoppen, P.: *Autonome mobile Roboter*. BI, 1995
- Schraft, R. D. und Volz, R. D.: *Serviceroboter*. Springer, 1995
- Nehmzow: *Mobile Robotics: A Practical Introduction*. Springer, 2000

