

Mohamed Oubbati

# Einführung in die Robotik

Institut für Neuroinformatik  
Universität Ulm  
Albert-Einstein-Allee 11, D-89069 Ulm

Stand: WS 2009/2010



# Inhaltsverzeichnis

<b>Contents</b>	<b>i</b>
<b>Preface</b>	<b>1</b>
<b>1 Einleitung</b>	<b>3</b>
1.1 Einsatzgebiete der Robotik . . . . .	4
1.1.1 Forschung . . . . .	4
1.1.2 Industrie . . . . .	4
1.1.3 Medizin . . . . .	5
1.1.4 Militär . . . . .	5
1.2 Was ist ein autonomer mobiler Roboter? . . . . .	6
1.2.1 Definition “Roboter” . . . . .	6
1.2.2 Definition “Autonom” . . . . .	6
1.2.3 Definition “Mobile” . . . . .	7
1.3 Klassifikation von Robotern . . . . .	7
1.3.1 Stationäre Roboter . . . . .	7
1.3.2 Mobile Roboter . . . . .	8
1.4 Ausstattung . . . . .	9
<b>2 Grundbestandteile eines Roboters</b>	<b>11</b>
2.1 Sensoren . . . . .	11
2.1.1 Interne/Externe Sensoren . . . . .	12
2.1.2 Aktive/Passive Sensoren . . . . .	12

2.1.3	Abstandssensoren . . . . .	13
2.2	Aktoren . . . . .	16
2.2.1	Räder . . . . .	16
2.2.2	Motoren . . . . .	17
2.2.3	Getriebe . . . . .	20
2.2.4	Antriebskonzepte . . . . .	21
2.3	Steuereinheit . . . . .	24
<b>3</b>	<b>Kinematik</b>	<b>25</b>
3.1	Geschwindigkeit . . . . .	25
3.1.1	Mittlere Geschwindigkeit . . . . .	26
3.1.2	Momentangeschwindigkeit . . . . .	26
3.2	Winkelgeschwindigkeit . . . . .	26
3.2.1	Mittlere Winkelgeschwindigkeit . . . . .	26
3.2.2	Momentane Winkelgeschwindigkeit . . . . .	26
3.3	Geschwindigkeitsverteilung bei einem Zwanglauf . . . . .	27
3.4	Geschwindigkeit des mobilen Roboters . . . . .	28
3.4.1	Differenzialantrieb . . . . .	28
3.4.2	Dreiradantrieb . . . . .	30
3.4.3	Omnidirektionaler Antrieb . . . . .	31
3.5	Kinematik des mobilen Roboters . . . . .	31
3.5.1	Vorwärtskinematik . . . . .	32
3.5.2	Rückwärtskinematik . . . . .	34
<b>4</b>	<b>Bewegungsregelung des Roboters</b>	<b>35</b>
4.1	Grundlagen der Regelungstechnik . . . . .	35
4.1.1	Unterschied Steuerung - Regelung . . . . .	35
4.1.2	Grundsätzlicher Aufbau von Regelkreisen . . . . .	38
4.1.3	Allgemeine Anforderungen an einen Regelkreis . . . . .	38
4.1.4	Lineare Standard-Regler (PID Regler) . . . . .	40

4.2	Geschwindigkeitsregelung (Dynamics Control) . . . . .	43
4.3	Positionsregelung (Kinematics Control) . . . . .	44
4.3.1	Problemdarstellung . . . . .	44
4.3.2	Holonom vs. Nicht-holonom . . . . .	45
4.3.3	Lösungen . . . . .	46
<b>5</b>	<b>Navigation</b>	<b>49</b>
5.1	Kartenerstellung . . . . .	49
5.2	Lokalisierung . . . . .	50
5.2.1	Dead-Reckoning . . . . .	50
5.2.2	Lokalisierung mittels Landmarken . . . . .	50
5.2.3	Probabilistische Lokalisierungsverfahren . . . . .	51
5.3	Bahnplanung . . . . .	51
5.3.1	Globale Bahnplanung . . . . .	51
5.3.2	Lokale Bahnplanung . . . . .	53
5.4	Steuerungsarchitekturen . . . . .	57
5.4.1	Deliberative Architektur . . . . .	58
5.4.2	Reaktive Architektur . . . . .	58
5.4.3	Hybride Architektur (Reaktive/Deliberative) . . . . .	59



# Vorwort

Robotik ist ein sehr interessantes Gebiet, in dem man nicht nur Programmierung lernt, sondern auch wie man mit realen Daten umgeht. In der Vorlesung wird fundiertes fachliches Grundwissen über die Robotersteuerung vermittelt. Die wichtigsten Vorlesungsthemen sind Sensoren, Aktoren, Kinematik, Regelungsprobleme in der Robotik, Einsatz von PID-Reglern in der Bewegungsregelung, und Grundlagen der Navigation. Praktische Übungen (auf realen Robotern) werden durchgeführt, um den Stoff der Vorlesung zu vertiefen.

Die Veranstaltung richtet sich an alle interessierten Studenten, die sich mit diesem spannenden Gebiet vertraut machen wollen.



# Kapitel 1

## Einleitung

Roboter stehen nicht mehr nur in der Industrie; Sie erobern jetzt das Wasser, die Luft, und das Wohnzimmer. Die Roboter werden im Prinzip alles tun können, was ein Mensch auch kann und noch vieles mehr. Einige Forschungsteams haben bereits Roboter entwickelt, die Gehen lernen wie Kinder. Mit jedem Schritt verbessern sie ihre Balance und das Gehen selbst. Mittlerweile können Roboter sogar selbstständig Objekte erkennen und Sprache verstehen. Vergleichbar mit dem Menschen sollen die zukünftige Roboter auch das “Denken” lernen.

Die interdisziplinäre Wissenschaft der Roboter ist die **Robotik**. Sie hat Beziehungen zu folgenden Disziplinen:

1. Informatik
2. Elektrotechnik (z.B. Hardware, Sensorik, Aktorik)
3. Maschinenbau (mechanischer Aufbau)
4. Mathematik
5. Künstliche Intelligenz (Bildererkennung, Spracherkennung, Lernen, Multiagentensysteme, Planung, ...)
6. Kognitionswissenschaften
7. Psychologie
8. Biologie (Mensch-, Tierverhalten)

Der wichtigste Aspekt in der Robotik ist die Sicherheit für die Menschen. Dazu schlug Issac Asimov die drei Gesetze der Robotik vor, die wie folgt lauten:

1. Ein Roboter darf keinen Menschen verletzen.
2. Ein Roboter muss den Anweisungen gehorchen, die ihm von Menschen gegeben werden, außer wenn diese dem ersten Gesetz widersprechen.
3. Ein Roboter muss seine eigene Existenz solange zu sichern versuchen, wie dies nicht dem ersten oder zweiten Gesetz widerspricht.

## 1.1 Einsatzgebiete der Robotik

### 1.1.1 Forschung

Forschungsroboter werden meist in Umgebungen eingesetzt, die für den Menschen zu gefährlich sind. Sie können Vulkane oder Abwasserrohre vordringen, und sie können sogar andere Planeten erkunden (Abbildung 1.1).



(a)



(b)

Abbildung 1.1: Forschungsroboter. (a) Mars-Rover (NASA). (b) Kanalroboter (Fraunhofer IAIS).

### 1.1.2 Industrie

Industrieroboter erledigen “stupide” Fließbandarbeit schneller und wesentlich genauer als ein Mensch. Außerdem brauchen sie fast keine Pausen, und können rund um die Uhr arbeiten.



Abbildung 1.2: Armroboter im Rohbau der Mercedes A-Klasse (KUKA GmbH).

### 1.1.3 Medizin

Der Vorteil von Robotern in der Medizin ist die Unabhängigkeit von mentalen Einflüssen und natürlich die konstante Arbeitsleistung. Neben der Präzision wird auch die Operationszeit verkürzt und damit die Belastung des Patienten verringert.



Abbildung 1.3: Roboter in der Medizin

### 1.1.4 Militär

Beim Militär werden bewaffnete Roboter eingesetzt. So ein "Soldat" hat keinen Hunger, kriegt keinen Sold, und vor allem kennt keine Angst. Auch um feindliches Gebiet auszuspionieren werden Roboter (Dronen) genutzt (Abb. 1.4).

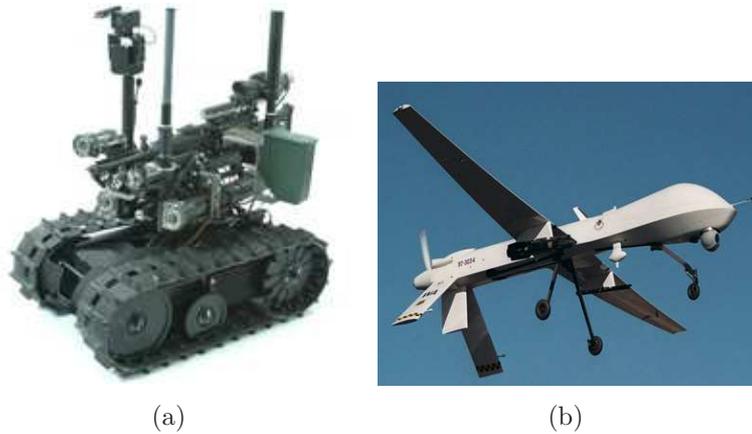


Abbildung 1.4: Militär-Roboter. (a) Assault TALON (U.S Army). (b) Drone: MQ-1 Predator UAV (U.S Airforce).

## 1.2 Was ist ein autonomer mobiler Roboter?

### 1.2.1 Definition “Roboter”

Der Begriff “*Roboter*” wurde von Karel Capek Anfang des 20. Jahrhunderts in der Science-Fiction-Literatur eingeführt. Sein Ursprung liegt im slawischen Wort robot, welches mit (Zwang-) Arbeit übersetzt werden kann.

**Definition der RIA (Robot Institute of America):** According to the Robot Institute of America (1979) a robot is: *a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.*

Ein Roboter ist ein programmierbares Mehrzweck-Handhabungsgerät für das Bewegen von Material, Werkstücken, Werkzeugen oder Spezialgeräten. Der frei programmierbare Bewegungsablauf macht ihn für verschiedenste Aufgaben einsetzbar.

### 1.2.2 Definition “Autonom”

“*Autonom*” hat seinen Ursprung in zwei griechischen Wörtern: “*autos*” ist griechisch für selbst und “*nomos*” für Gesetz. Es bedeutet also “eigengesetzlich machend”. Der Begriff “autonom” in der Robotik bedeutet, dass der Roboter sich ohne externe

Unterstützung in seiner Umgebung funktionieren kann, je nach Einsatzumgebung kann die Autonomie in verschiedene Grade unterteilt werden (Abb. 1.5).

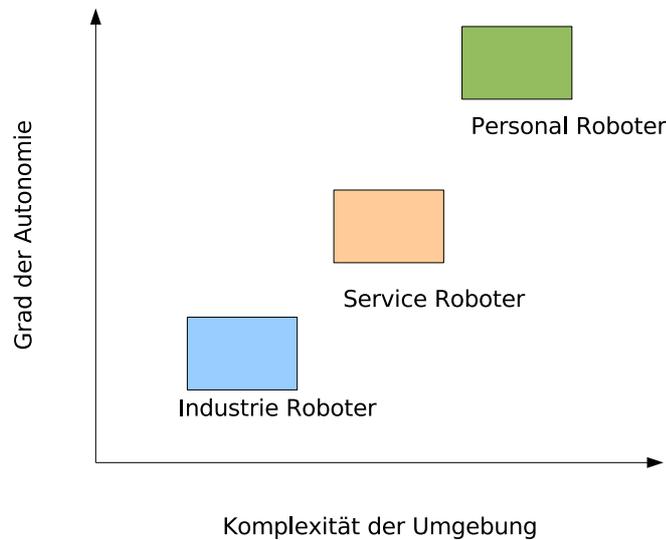


Abbildung 1.5: Autonomie-Grade.

### 1.2.3 Definition “Mobile”

Ein Roboter wird mobile genannt, wenn er in seiner Umgebung (Land, Wasser, Luft) bewegen kann. Er ist also nicht an einen festen Standort gebunden. Für die Fortbewegung gibt es eine Vielzahl von Bewegungsarten: Fahren, Laufen, Fliegen, Schwimmen usw.

## 1.3 Klassifikation von Robotern

Es gibt insgesamt zwei Hauptgruppen von Robotern, welche in unterschiedlichen Gebieten eingesetzt werden.

### 1.3.1 Stationäre Roboter

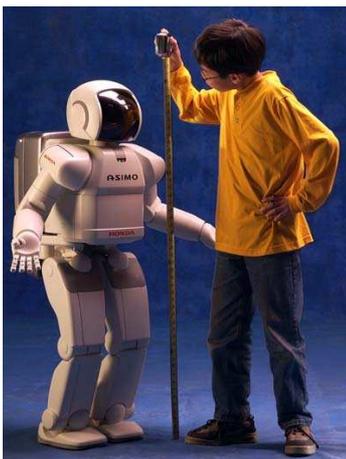
Wie der Name schon sagt, ein stationärer Roboter ist an einem festen Punkt gebunden, und kann nicht durch eigenen Antrieb seinen Standort wechseln (Abb. 1.6). So ein Roboter ist für einen eindeutigen und repetitiven Arbeitsablauf ausgelegt.



Abbildung 1.6: ABB Industrieroboter.

### 1.3.2 Mobile Roboter

Wie schon definiert, ein Roboter wird mobile genannt, wenn er Aktoren zur Veränderung seiner Position in der Umwelt besitzt. Die Hauptaufgabe eines mobilen Roboters ist das Navigieren in bekannten und unbekanntem Umgebungen.



(a)



(b)



(c)

Abbildung 1.7: Mobile Roboter. (a) Humanoid Roboter (ASIMO-Honda). (b) Der Roboter Stanley (Stanford University- USA). (c) Indoor Roboter (PeopleBot - Activemedia).

## 1.4 Ausstattung

Die bisher gezeigten Beispiele von existierenden Robotern zeigen, dass sie nach Aufgabe und Einsatzmöglichkeit völlig unterschiedlich ausgestattet sind. In einer indoor Umgebung kann ein Roboter sich einfach mit zwei Rädern bewegen. Der Mars-rover bewegt sich (Bild 1.1. (a)) mit 6 Rädern im Allradantrieb fort, während sich der ASIMO Roboter von Honda auf zwei Beinen geht (Bild 1.7. (a)). Um Tätigkeiten der Industrie zu erledigen, wird zumeist ein Industrieroboter (Bild 1.2) mit vielen Freiheitsgraden und einem Greifer ausgestattet, abhängig von der Arbeit, die er ausführen soll. Die Art und Anzahl der Sensoren sind auch stark abhängig von der Umgebung und der Aufgabe. Zumeist werden Sonar-, und Lasersensoren benutzt um Abstände zu messen. Eine Kamera kann sie zusätzlich ergänzen. (Bild 1.8)



Abbildung 1.8: Indoor Roboter.



# Kapitel 2

## Grundbestandteile eines Roboters

Ein Roboter besteht aus mehreren Teilsystemen: Sensoren, Aktoren, und die Steuereinheit (Abb. 2.1).

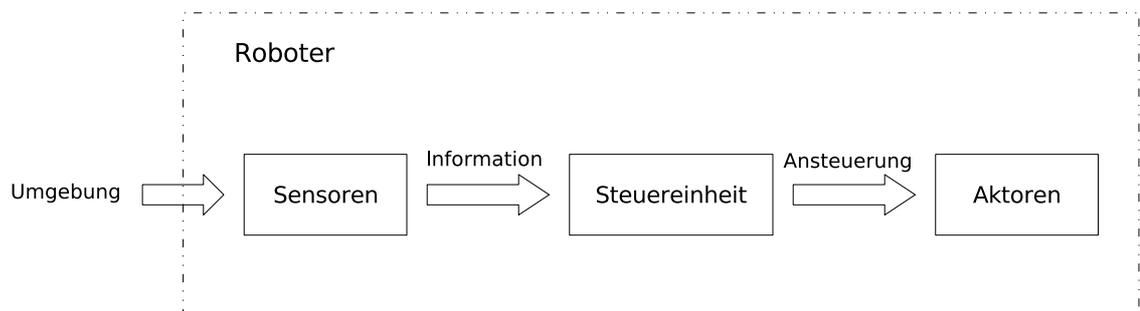


Abbildung 2.1: Roboter Blockschema

Die **Sensoren** empfangen Umgebungseinflüsse. Durch die empfangenen Daten entscheidet die **Steuereinheit** welche Aktion der Roboter, durch die **Aktoren**, ausführen soll.

### 2.1 Sensoren

Mit Hilfe von Sensoren kann ein Roboter Informationen aus seiner Umgebung aufnehmen. Ein Sensor empfängt ein physikalisches Signal und reagiert darauf mit einem elektrischen Ausgangssignal (Abb. 2.2).



Abbildung 2.2: Sensor

### 2.1.1 Interne/Externe Sensoren

Die Sensoren können klassifiziert werden, ob sie intern oder extern sind:

Interne Sensoren messen Eigenschaften “on board” des Roboters, zum Beispiel:

- Radenkoder (Position)
- BatterieLadezustand (Spannung)
- Temperatur

Externe Sensoren liefern dagegen Informationen aus der Umgebung:

- Ultraschall
- Bumper
- Kamera
- Laser
- Infrarot

### 2.1.2 Aktive/Passive Sensoren

Ein weiteres Unterscheidungskriterium ist die Einteilung in aktive und passive Sensortypen.

**Aktive Sensoren** wirken wie eine elektrische Spannungsquelle. Sie variieren das elektrische Signal bei einer Änderung der Messgröße (mechanische, thermische ...).

Ein **passiver Sensor** verändert seinen Widerstand, seine Kapazität oder seine Induktivität, und damit wandelt er die Erregungsenergie in ein elektrisches Signal um.

### 2.1.3 Abstandssensoren

Die Daten von Abstandssensoren eignen sich zur Hindernisvermeidung sowie zur Extraktion eines Modells der Umgebung, was zur Navigation genutzt werden kann.

**Bumper:** Bumper (Touch Sensor) werden durch den direkten Kontakt mit dem Hindernis aktiviert. Jedoch sind sie zur Kollisionsvermeidung nicht geeignet.

**Ultraschall:** Ultraschall wird hauptsächlich auf Basis des Piezoelektrisches Effekts erzeugt und empfangen. Die Frequenzen des Ultraschalls liegen oberhalb des Hörbereiches des Menschen (20 Khz bis Gigaherzbereich). Die Ultraschallsensoren funktionieren, indem sie kegelförmige Schallwellen abstrahlen. Wenn diese Wellen auf einen Gegenstand treffen, werden sie von ihm in Richtung des Senders reflektiert. Aus der Laufzeit  $t$  zwischen gesendetem und empfangenem Signal kann die Entfernung  $\Delta x$  zur Reflexionsfläche berechnet werden:

$$(2.1) \quad \Delta x = \frac{c' \cdot t}{2}$$

Die Schallgeschwindigkeit hängt von der Eigenschaft des Mediums und insbesondere von der Temperatur ab. Sie beträgt bei Raumtemperatur in Luft  $c' = 344m/s$ .

#### *Probleme*

1. Wegen der kegelförmigen Schallwellen, gibt das empfangene Signal Informationen über die Entfernung einer Reflexionsfläche, aber nicht über die Winkelrichtung, in der sie sich befindet (siehe Abb. 2.3)

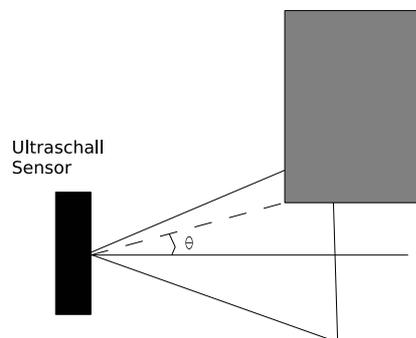


Abbildung 2.3: Winkel Probleme beim Ultraschall

2. Scheinechos: Scheinechos können aus verschiedenen Gründen auftreten:

a) eine Welle kann auf mehreren Objekten reflektiert werden, bevor sie als Echo vom Sensor empfangen wird. Es könnte vorkommen, dass Echos von weit entfernten Objekten während einer neuen Messung empfangen werden. So scheint ihre Entfernung viel kleiner, als sie tatsächlich ist.

b) bei gleichzeitigem Betrieb von mehreren Ultraschallsensoren kann ein Sensor das Echo einer Welle empfangen, die von einem anderen Sensor gesendet wurde (Crosstalk Problem). (siehe Abb. 2.4)

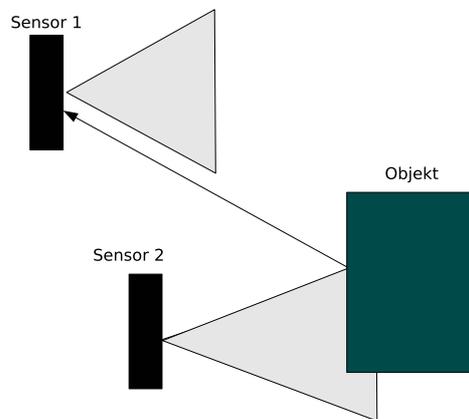


Abbildung 2.4: Crosstalk Problem

**Lasersensoren:** Laser oder “*light amplification by stimulated emission of radiation*” bietet eine sehr genaue Entfernungsmessung. Ein Lasersensor arbeitet nach dem time-of-flight Prinzip (TOF). Er sendet einen gepulsten Laserstrahl aus, der dann von Objekten reflektiert wird. Dann wird die Zeit  $t$  zwischen Aussendung und Empfang des Laserimpulses gemessen. Daraus lässt sich die Entfernung  $\Delta x$  mit Hilfe der Lichtgeschwindigkeit  $c$  wie folgt berechnen:

$$(2.2) \quad \Delta x = \frac{c \cdot t}{2}$$

Der Messbereich beträgt bis zu 80 Meter bei einer Auflösung von einem Zentimeter. Es wird eine Wellenlänge, die im nicht sichtbaren Bereich liegt. Zum Beispiel bei den Lasermesssystemen von SICK (Bild 2.5) wird bei einer Wellenlänge von 905 nm gemessen. Das Lasersensor von SICK sendet den Laserimpuls, und dabei wird mit einer Winkelauflösung von 0,5 gemessen. Daraus ergeben sich in einem Abstand von 10 Metern 3,13 cm breite Bereiche, die in keine Messung eingehen.



Abbildung 2.5: Lasersensor von Sick

Ein weiterer Vorteil sind die kurzen Taktzeiten, welche auch bei hohen Geschwindigkeiten noch möglich sind. Allerdings trotz all dieser Vorteile, besitzen Lasersensoren auch Nachteile. Der Hauptnachteil ist es, dass sie im Vergleich mit anderen Sensoren sehr teurer sind.

**Licht-/Infrarotsensor:** Ein Lichtsensor arbeitet mit Hilfe von Fotowiderständen (Diode- Transistor) um die Lichtstärke aufzunehmen. Eine veränderte Lichtstärke resultiert dabei in einem anderen "Widerstand", damit er diese Erregungsenergie in ein elektrisches Signal umwandelt. Der Infrarotsensor funktioniert nach dem gleichen Prinzip wie der Lichtsensor, allerdings wird mit ihm das Infrarot Licht (leicht unterhalb des sichtbaren Lichtes) gemessen. Er sendet kodierte Infrarotsignale aus und misst die Echostärke dieser Signale. Infrarotsensoren werden für den Nahbereich eingesetzt.

**Radencoder:** Um die Geschwindigkeit des Roboters regeln zu können, muss der Roboter die Geschwindigkeit der Räder messen können. Dies kann durch einen Radencoder berechnet werden. Der Radencoder wertet ein kodierte Rad aus, wobei Markierungen über Licht-Reflexion beobachtet werden (Abb. 2.6). Da die Anzahl von Impulsen pro Umdrehung fest ist, die resultierende Pulsfolge ist direkt proportional zur Rotation.

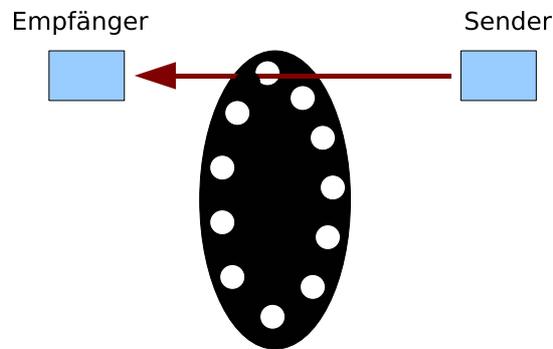


Abbildung 2.6: Modell Radencoder

## 2.2 Aktoren

Aktoren sind die Teile, mit denen der Roboter physikalische Aktionen durchführen kann. Ein mobiler Roboter besitzt insbesondere Aktoren (Z. b. Räder, Beine, oder Ketten) zur Veränderung seiner Position in seiner Umgebung. Mit Rädern kann ein Roboter auf glattem Untergrund höhere Geschwindigkeiten erreichen. Beine, und Ketten ermöglichen eine sehr flexible Fortbewegung auf eine Gelände.

### 2.2.1 Räder

Es gibt verschiedene Möglichkeiten den mobilen Roboter anzutreiben, zum Beispiel mit Rädern unterschiedlicher Größe und Anzahl, oder auch mit Kettenantrieb. Abbildung (2.7) zeigt ein Rad, das ein Radius ( $r$ ) und eine Winkelgeschwindigkeit ( $\Omega$ ) hat.

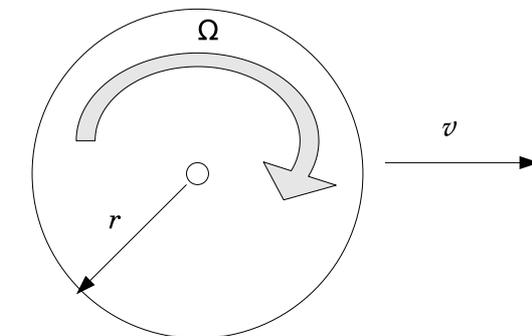


Abbildung 2.7: Rad

### 2.2.2 Motoren

Ein **Elektromotor** ist ein elektrisches Gerät, das mit Hilfe von magnetischen Feldern elektrische Energie in mechanische Energie umsetzt. Dabei wird die Kraftwirkung zwischen Elektromagneten zur Erzeugung einer Drehbewegung ausgenutzt. Auf dem Roboter werden häufig Getriebemotoren (DC motors) eingesetzt (Abb. 2.8). Zuerst schließt man die Spannungsquelle an, und je Höhe die Spannung, desto schneller dreht sich der Motor (Abb. 2.9).



Abbildung 2.8: DC Motor (Maxon)

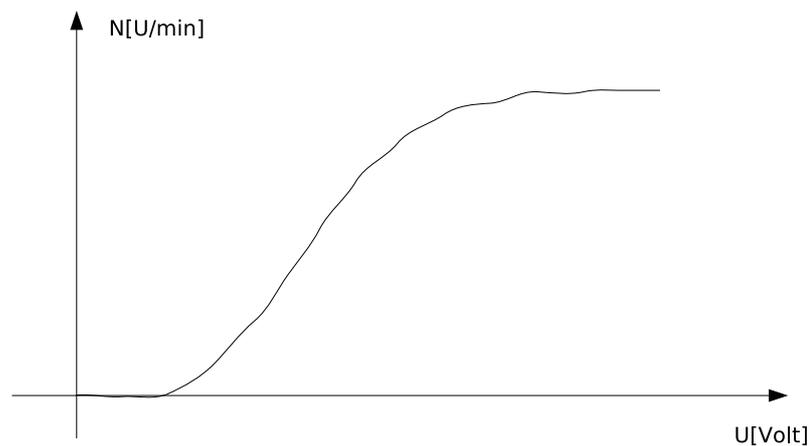


Abbildung 2.9: Drehzahl VS. Motorspannung

**Pulsweitenmodulation PWM:** Um die Geschwindigkeit des Motors zu regulieren, muss man Die Pulsweitenmodulation (engl. Pulse Width Modulation, abgekürzt PWM) nutzen. Bei der PWM wird die Ein- und Ausschaltzeit eines

Rechtecksignals bei fester Grundfrequenz variiert. Je länger die Einschaltdauer, desto schneller dreht sich der Motor. Das heißt, je länger die Einschaltdauer, desto höher der Mittelwert der Spannung (Abb. 2.10).

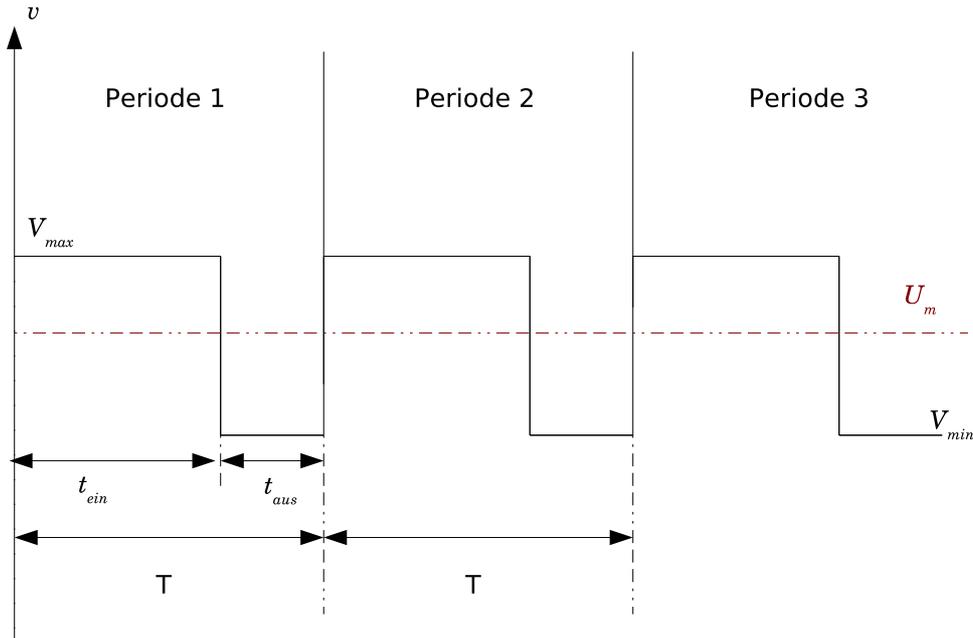


Abbildung 2.10: Pulsweitenmodulation

So wird der Mittelwert  $U_m$  berechnet:

$$(2.3) \quad U_m = \frac{1}{T} \int_0^T v(\tau) d\tau$$

$$(2.4) \quad U_m = \frac{1}{T} \int_0^{t_{ein}} V_{max} d\tau + \frac{1}{T} \int_{t_{ein}}^T V_{min} d\tau$$

$$(2.5) \quad U_m = \frac{t_{ein}}{T} V_{max} + \frac{T - t_{ein}}{T} V_{min}$$

$$(2.6) \quad U_m = \frac{t_{ein}}{t_{ein} + t_{aus}} V_{max} + \left(1 - \frac{t_{ein}}{t_{ein} + t_{aus}}\right) V_{min}$$

Das Verhältnis  $\frac{t_{ein}}{t_{ein} + t_{aus}}$  bezeichnet man als Tastverhältnis ( $D$ ) (engl. Duty Cycle):

$$(2.7) \quad D = \frac{t_{ein}}{t_{ein} + t_{aus}}$$

Die Gleichung (2.6) wird dann :

$$(2.8) \quad U_m = DV_{max} + (1 - D)V_{min}$$

$V_{min}$  ist dabei normalerweise  $0V$ , und  $V_{max}$  ist die Betriebsspannung  $V_{CC}$ . Deshalb kann man vereinfacht schreiben:

$$(2.9) \quad U_m = DV_{CC}$$

**Motorsteuerung mit H-Brücke:** Ein Motor kann nicht direkt durch einen Mikroprozessor angetrieben werden, da dieser nicht genügend Strom liefert. Stattdessen muß eine Verbindungsschaltung aufgebaut werden, so dass die Energie für den Motor von einer anderen Spannungsquelle kommt und der Mikroprozessor nur die Steuersignale sendet. Man verwendet am meisten die sogenannte H-Brücke. Diese Schaltung hat ihre Name von ihrer Form, die wie ein "H" aussieht, und besteht aus Bipolartransistoren (Abb. 2.11). Die blaue Linie zeigt den Stromweg für die eine Drehrichtung, die rote Linie den für die andere Drehrichtung. Dazu passend die logischen Signale "0" für einen gesperrten und "1" für einen durchgeschalteten Transistor. Zur Steuerung der Motorgeschwindigkeit werden die Transistoren, durch PWMs Signale, in verschiedenen Abständen geöffnet und geschlossen.

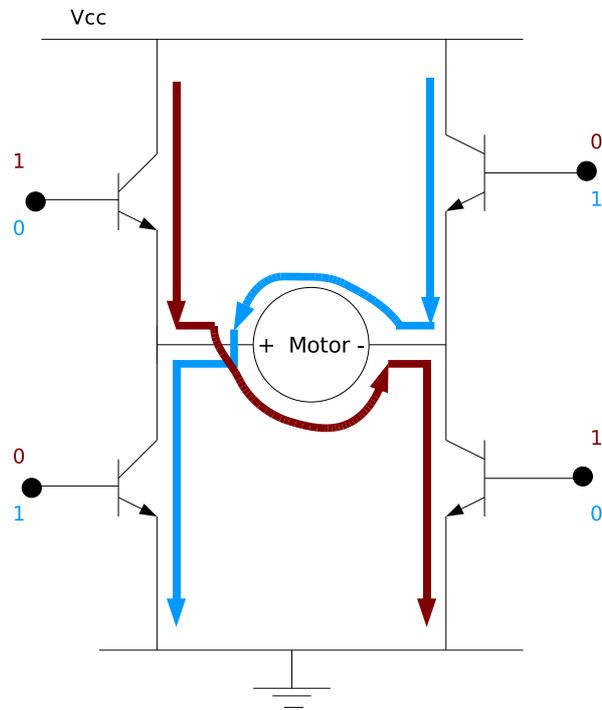


Abbildung 2.11: Schema einer H-Brücke

### 2.2.3 Getriebe

Um die gewünschten Fahrleistungen des Roboters zu erreichen ist es notwendig den Motor mit Getriebe zu koppeln. Getriebe sind Verbindungen, die zur Übertragung von Kraft dienen. Beim mobilen Roboter werden Getriebe mit Hilfe von Zahnrädern eingesetzt, um die Drehzahl und das Drehmoment an die Anforderungen anzupassen (Bild 2.12).

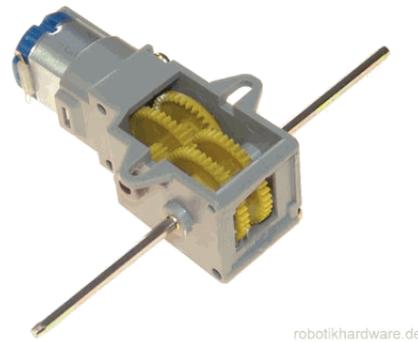


Abbildung 2.12: Motor mit Getriebe

### 2.2.4 Antriebskonzepte

Für stabile Fahrzeuge braucht man mindestens drei Räder. Die Häufigste Radanordnungen sind:

- Differentialantrieb** Dieses besteht aus zwei angetriebene Räder auf einer Achse mit einem oder zwei passiv mitlaufenden Stützrädern, die sich frei drehen können (siehe Abb. 2.13). Die beiden Antriebsräder liegen auf derselben Achse und werden getrennt angetrieben. Durch die Geschwindigkeit der beiden angetriebenen Räder hat der Roboter die Möglichkeit vorwärts und rückwärts zu fahren, auf der Stelle zu drehen und sich auf einer Kreisbahn zu bewegen.

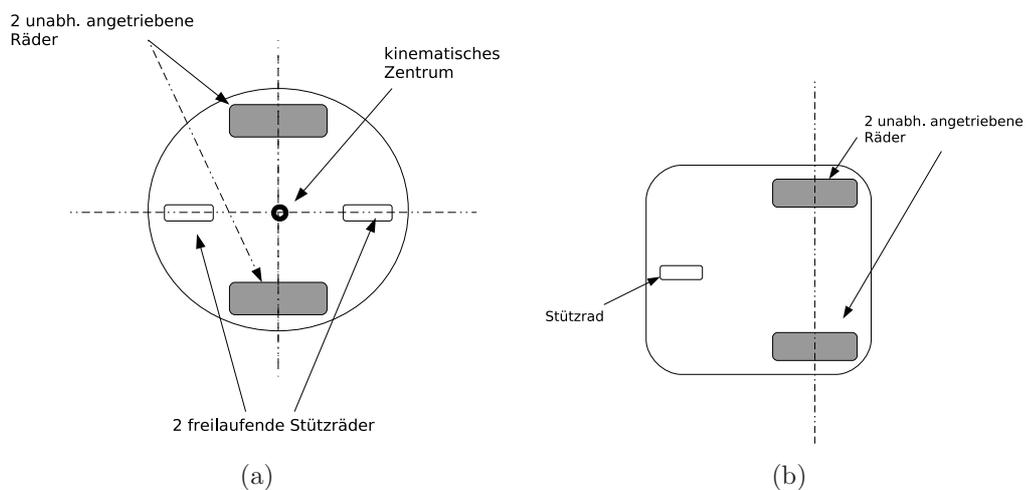


Abbildung 2.13: Schema Differentialantrieb

- 3. Dreiradantrieb:** Der mobile Roboter ist mit 2 freilaufenden Rädern und 1 angetriebenem und gelenktem Rad ausgestattet (Abb. 2.14). Es kann aber auch sein, dass ein Motor die beiden Hinterräder antreibt, und ein zweiter Motor, das Vorderrad steuert.

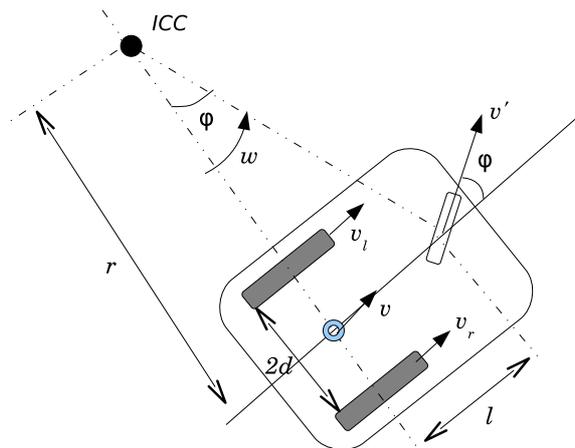


Abbildung 2.14: Dreiradantrieb

- 2. Kettenantrieb:** Hier wird der Roboter nicht von Rädern, sondern, wie ein Panzer, von zwei Ketten angetrieben. Das Prinzip ist analog zum Differentialantrieb.

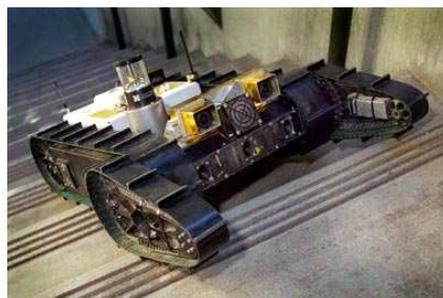


Abbildung 2.15: Roboter mit Kettenantrieb (US. Force)

- 4. Ackermannlenkung:** Der mobile Roboter hat eine Lenkung wie die von Autos bekannte Lenkung. Sie ist analog zum Dreiradantrieb, nur dass hier zwei synchron lenkbare Vorderräder vorhanden sind: zwei angetriebene (Hinter) Räder und zwei parallel geführte Lenkräder (Hinterradantrieb) (Abb. 2.16 (a)) oder zwei angetriebene und gelenkte parallel geführte Räder, dazu zwei passiv mitlaufende Hinterräder (Vorderradantrieb) (Abb. 2.16 (b)).

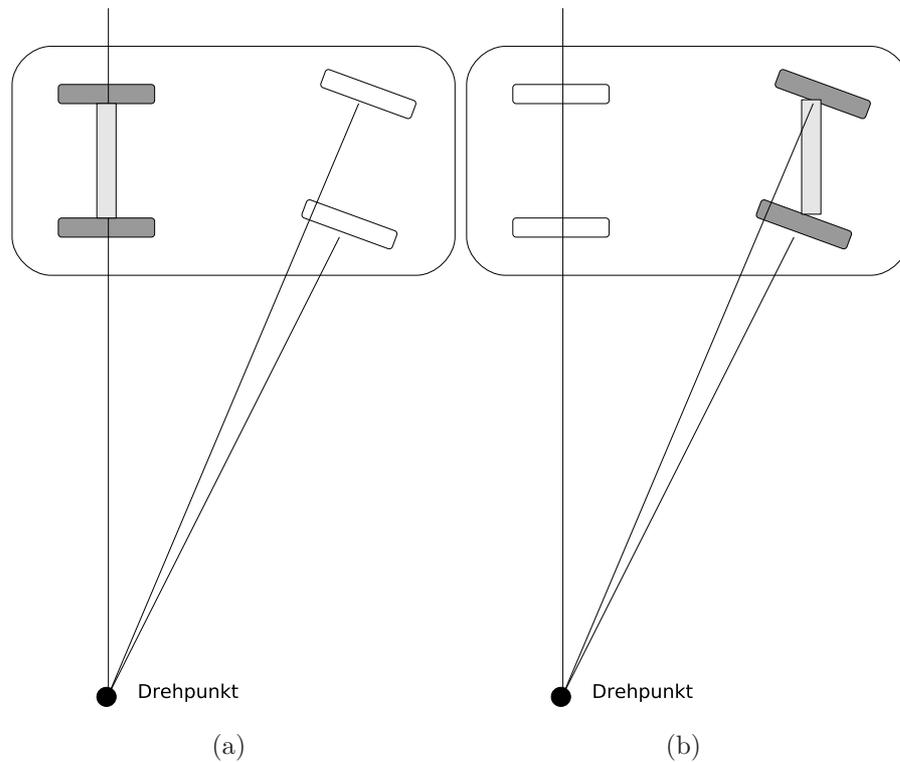


Abbildung 2.16: (a) Hinterradtrieb. (b) Vorderradtrieb

**5. Synchrodriebe:** Drei oder mehr angetriebene Räder werden synchron gelenkt (Abb. 2.17).

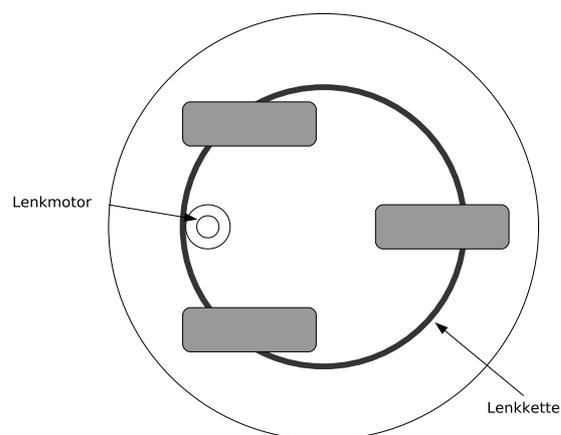


Abbildung 2.17: Synchrodriebe Radantrieb

**6. Omnidirectional:** Unter Verwendung von drei (manchmal auch 4) angetriebenen sogenannten Allseitenrädern (auch Omni-Wheels, Mecanum-Räder) (Abb.

2.18) ist der Roboter in der Lage, sich in alle Richtungen zu bewegen ohne sich vorher drehen zu müssen.

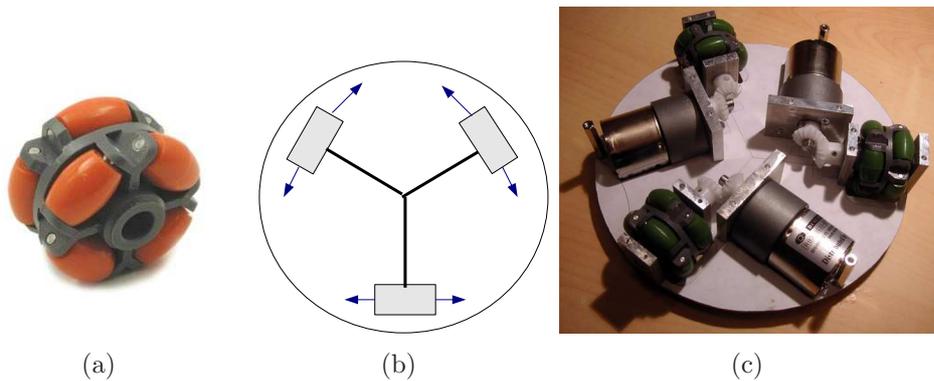


Abbildung 2.18: (a) Allseitenrad (Omniwheel). (b) Anordnung von 3 Omniwheels. (c) Hardware Omnidrive Chassis.

## 2.3 Steuereinheit

Die Steuerungseinheit ist das Gehirn jedes Roboters. Dort werden die Informationen der Sensoren verarbeitet und die Aktoren angesteuert.

Die Steuerungseinheit übernimmt folgende Aufgaben:

- Realisierung des Gesamtablaufs dem eingegebenen Programm.
- Auswertung von Sensordaten.
- Steuerung der Bewegungen (z.B. Punkt-zu-Punkt-Steuerung, Bahnsteuerung).

# Kapitel 3

## Kinematik

Die **Kinematik** beschäftigt sich mit der Geometrie und den zeitabhängigen Aspekten der Bewegung, ohne die Kräfte, die die Bewegung verursacht.

### 3.1 Geschwindigkeit

Gegeben sei ein Zwangslauf  $\xi$ . Weiters seien  $X_0$ ,  $X_1$  zwei benachbarte Positionen von einem Punkt  $X$  auf  $\xi$  zu den Zeitpunkten  $t_0 < t_1$  (Abb. 3.1).

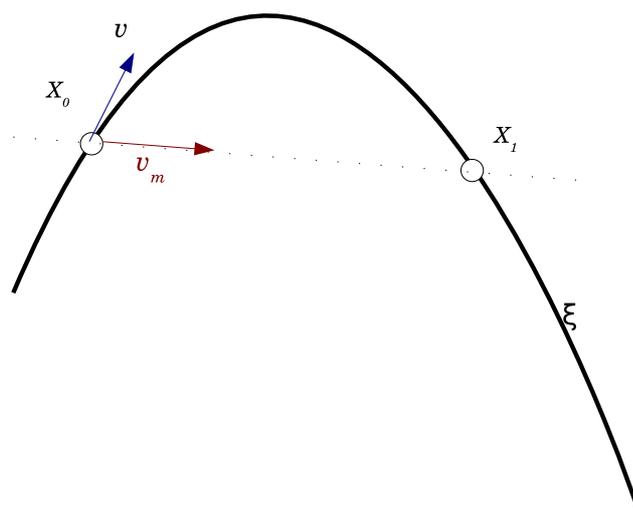


Abbildung 3.1: Definition des mittleren Geschwindigkeitsvektors  $v_m$  und des Momentangeschwindigkeitsvektors  $v$

### 3.1.1 Mittlere Geschwindigkeit

Die skalare Größe

$$(3.1) \quad v_m = \frac{\Delta s}{\Delta t}$$

heißt *mittlere Geschwindigkeit* von  $X$  im Intervall  $[t_0, t_1]$ . Hierbei bezeichnet  $\Delta s$  die Länge des Kurvenbogens zwischen  $X_0$  und  $X_1$  auf  $\xi$ , und  $\Delta t = t_1 - t_0$ .

### 3.1.2 Momentangeschwindigkeit

Die skalare Größe

$$(3.2) \quad v = \lim_{\Delta t \rightarrow 0} v_m = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t}$$

heißt *Momentangeschwindigkeit* des Punktes  $X$  zum Zeitpunkt  $t_0$ .

## 3.2 Winkelgeschwindigkeit

Die obige Begriffe lassen sich auch auf Winkelgrößen übertragen. Es sei  $\rho(t)$  eine kontinuierliche Drehung um einem Punkt  $P$  mit dem Drehwinkel  $\varphi(t)$ . Sei weiters  $\Delta\varphi = \varphi(t_1) - \varphi(t_0)$  und  $\Delta t$ , die wieder dieselbe Bedeutung wie in Definition 1 haben (Abb. 3.2).

### 3.2.1 Mittlere Winkelgeschwindigkeit

Die skalare Größe

$$(3.3) \quad w_m = \frac{\Delta\varphi}{\Delta t}$$

heißt *mittlere Winkelgeschwindigkeit* der Drehung  $\rho$  im Intervall  $[t_0, t_1]$ .

### 3.2.2 Momentane Winkelgeschwindigkeit

Die Größe

$$(3.4) \quad w = \lim_{\Delta t \rightarrow 0} w_m = \lim_{\Delta t \rightarrow 0} \frac{\Delta\varphi}{\Delta t}$$

heißt *momentane Winkelgeschwindigkeit* der Drehung  $\rho$  zum Zeitpunkt  $t_0$ .

Die mittlere Winkelgeschwindigkeit und die Winkelgeschwindigkeit sind unabhängig vom betrachteten Punkt des bewegten Systems, und das im Gegensatz zur mittleren Geschwindigkeit und zur Momentangeschwindigkeit.

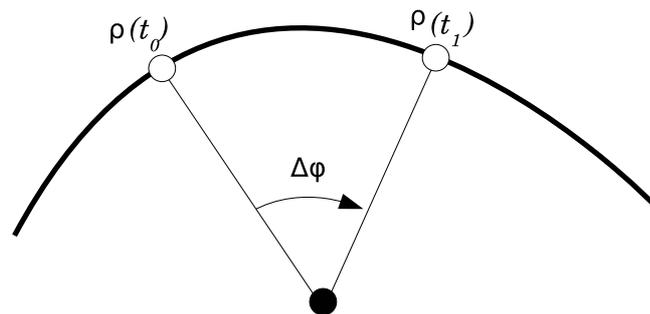


Abbildung 3.2: Definition der mittleren Winkelgeschwindigkeit  $w_m$  und der momentanen Winkelgeschwindigkeit  $w$ .

### 3.3 Geschwindigkeitsverteilung bei einem Zwangslauf

Sei der Punkt  $P$  das Zentrum eines Zwangslauf  $\Gamma$ , und  $\varphi$  die Winkel jeder Drehung. Weiters bezeichne  $r$  den Radius des Drehkreises eines Punkt  $X$  (Abb. 3.3). Dann kann die mittlere Geschwindigkeit  $v$  von  $X$  im Intervall  $[t_0, t_1]$  durch

$$(3.5) \quad v_m \simeq \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t}$$

berechnet werden.

Die momentangeschwindigkeit  $v$  zum Zeitpunkt  $t_0$  wird dann:

$$(3.6) \quad v = \lim_{\Delta t \rightarrow 0} \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t}$$

Es ist bekannt dass  $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$ . Daher kann die rechte Seite der Gleichung (3.6) wie folgt umgeschrieben werden:

$$(3.7) \quad \lim_{\Delta t \rightarrow 0} \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t} = 1 \cdot \lim_{\Delta t \rightarrow 0} \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t} = \left( \lim_{\Delta t \rightarrow 0} \frac{\frac{\Delta\varphi}{2}}{\sin \frac{\Delta\varphi}{2}} \right) \left( \lim_{\Delta t \rightarrow 0} \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t} \right)$$

$$(3.8) \quad v = \lim_{\Delta t \rightarrow 0} \left( \frac{\frac{\Delta\varphi}{2}}{\sin \frac{\Delta\varphi}{2}} \frac{2r \sin \frac{\Delta\varphi}{2}}{\Delta t} \right) = \lim_{\Delta t \rightarrow 0} \frac{\Delta\varphi}{\Delta t} r$$

$$(3.9) \quad v = w \cdot r$$

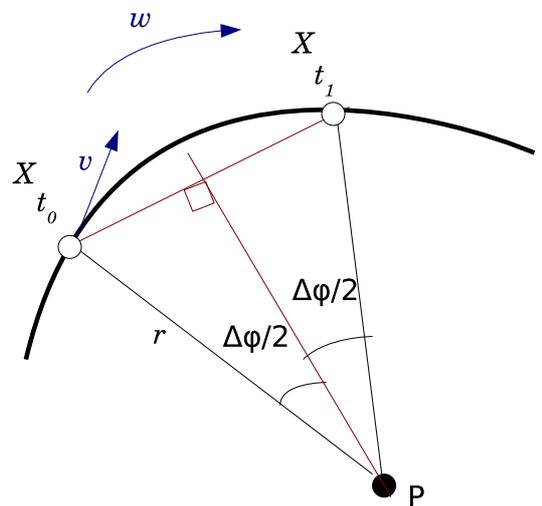


Abbildung 3.3: Geschwindigkeitsverteilung bei einem Zwangslauf

## 3.4 Geschwindigkeit des mobilen Roboters

### 3.4.1 Differenzialantrieb

Der Roboter verfügt über einen Differenzialantrieb. Die beiden Antriebsräder liegen auf derselben Achse mit einem Abstand  $2d$  und werden getrennt angetrieben, und das hintere Stützrad ist nicht angetrieben (Abb. 3.4). Die Geschwindigkeiten des rechten bzw. linken Rades sind  $v_r$  bzw.  $v_l$ , und die Geschwindigkeit des Roboters ist  $v$ . Unter der Annahme, dass  $v_r$  und  $v_l$  konstant sind und  $v_r \neq v_l$  ist, bewegt sich der Roboter auf einer Kreisbahn mit konstanter Rotationsgeschwindigkeit  $w$ . Der Punkt **ICC** (*Instantaneous Center of Curvature*) beschreibt das Zentrum der Drehung und  $r$  den Radius der Kreisbahn, die der Roboter abfährt.

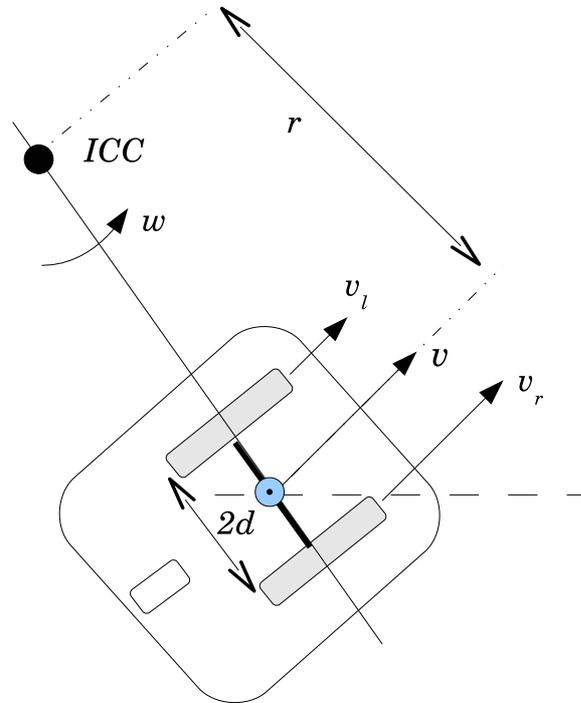


Abbildung 3.4: Die Geschwindigkeit des Roboters. Der Roboter bewegt sich bei konstanten Radgeschwindigkeiten auf einer Kreisbahn mit dem Radius  $r$  um ein Drehzentrum  $ICC$ .

Die Bewegung des Roboters wird durch die Geschwindigkeit der beiden angetriebenen Räder bestimmt. Durch die Gleichung (3.9) erhalten wir die Geschwindigkeit  $v_r$  und  $v_l$ :

$$(3.10) \quad v_l = (r - d) \cdot w$$

$$(3.11) \quad v_r = (r + d) \cdot w$$

Mit Hilfe der Gleichungen (3.9), (3.10), und (3.11) können die Geschwindigkeit und die Winkelgeschwindigkeit folgendermaßen bestimmt werden:

$$(3.12) \quad v = \frac{v_l + v_r}{2}$$

$$(3.13) \quad w = \frac{v_r - v_l}{2d}$$

Die Distanz  $r$  wird dann:

$$(3.14) \quad r = d \frac{v_l + v_r}{v_r - v_l}$$

- Für  $v_l = v_r$  ergibt sich  $r = \infty$  und  $w = 0$ , der Roboter fährt geradeaus.
- Für  $v_l = -v_r$  ergibt sich  $r = 0$  und  $v = 0$ , der Roboter dreht sich auf der Stelle.

### 3.4.2 Dreiradantrieb

Der mobile Roboter ist mit 2 freilaufenden Räder und 1 angetriebenen und gelenkten Rad ausgestattet. Es kann aber auch sein, dass ein Motor die beiden Hinterräder antreibt, und ein zweiter Motor das Vorderrad steuert (Abb. 3.5).

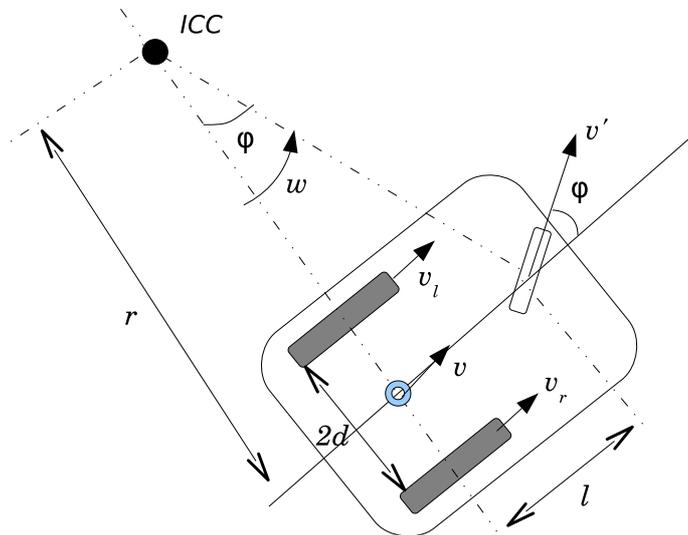


Abbildung 3.5: Dreiradantrieb

Aus der Abbildung (3.5) haben wir:

$$(3.15) \quad \tan \varphi = \frac{l}{r} \Leftrightarrow r = \frac{l}{\tan \varphi}$$

und

$$(3.16) \quad v = v' \cos \varphi$$

Durch die Gleichung (3.9) erhalten wir die Geschwindigkeit  $v_r$  und  $v_l$ :

$$(3.17) \quad v_r = (r + d) \cdot w = \frac{r + d}{r} v$$

$$(3.18) \quad v_l = (r - d) \cdot w = \frac{r - d}{r} v$$

### 3.4.3 Omnidirektionaler Antrieb

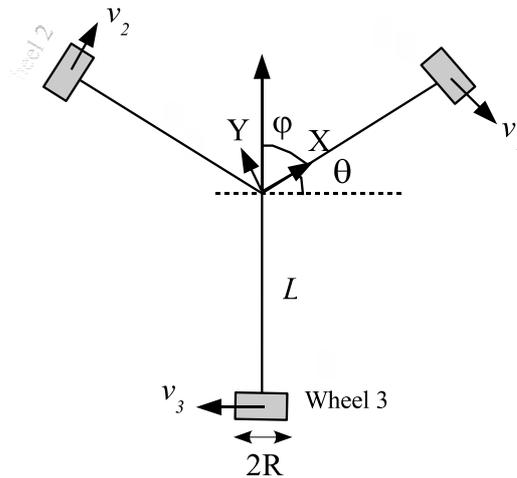


Abbildung 3.6: Omnidirektionaler Antrieb

Nachfolgend die Formeln zur Ansteuerung der Geschwindigkeiten  $w_i$  der Motoren.

$$(3.19) \quad \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} \sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3} - \theta) & -\cos(\frac{\pi}{3} - \theta) & L \\ \sin(\frac{\pi}{3} + \theta) & -\cos(\frac{\pi}{3} + \theta) & L \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

Dabei bezeichnen  $\dot{x}$  und  $\dot{y}$  jeweils die  $x$  und  $y$ -Komponente des gewünschten Geschwindigkeitsvektors.  $\dot{\theta}$  bezeichnet die Drehgeschwindigkeit, wobei man im Normalfall während des Fahrens keine zusätzliche Drehbewegung ausführen sollte, so dass  $\dot{\theta} = 0$  gesetzt werden kann.  $L$  ist der Abstand der Räder vom Mittelpunkt des Roboters.

## 3.5 Kinematik des mobilen Roboters

Abbildung (3.7) beschreibt die Position und die Orientierung  $(x, y, \Theta)$  (engl. pose) eines Roboters mit einem Differentialantrieb bezüglich des Weltkoordinatensystems  $\Sigma$ . Dabei liegt  $(x, y)$  in der Mitte der Antriebsräder.

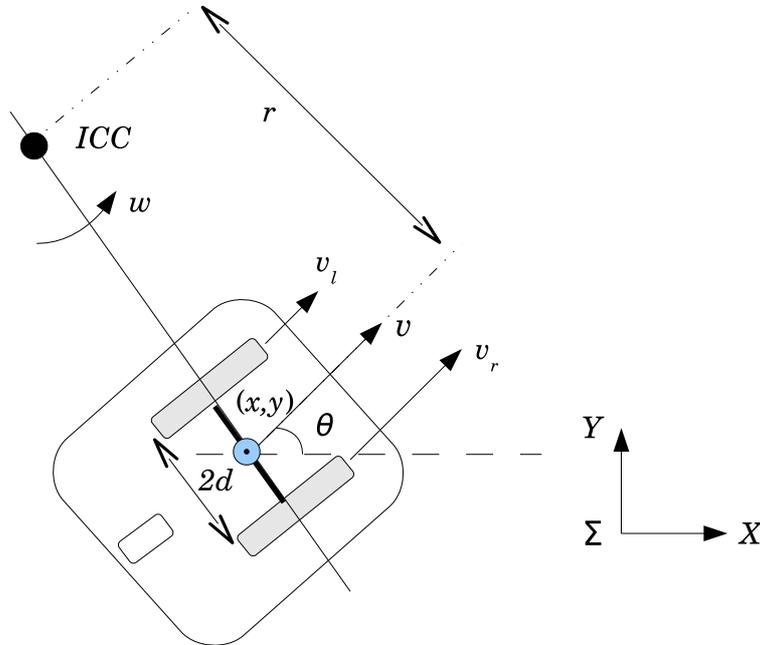


Abbildung 3.7: Kinematik des Roboters.

### 3.5.1 Vorwärtskinematik

*“If I do this, what will happen?”*

In der Vorwärtskinematik berechnet man die Position des Roboters bei gegebenen Geschwindigkeit  $v$  und Winkelgeschwindigkeit  $w$ .

Sei die initiale Position des Roboters  $P(t=0) = (0, 0, 0)$ . Der Roboter bewegt sich im Intervall  $\delta t = [0, t]$  mit einer Geschwindigkeit  $v$  und einer Winkelgeschwindigkeit  $w$ , so dass die neue Position  $P(t) = (x(t), y(t), \theta(t))$  gilt.

Mit Hilfe der Abbildung (3.7) können die Geschwindigkeiten  $(v_x(t), v_y(t))$  für den Zeitraum  $\delta t$  folgendermaßen bestimmt werden:

$$(3.20) \quad v_x(t) = v(t) \cos(\theta)$$

$$(3.21) \quad v_y(t) = v(t) \sin(\theta)$$

die neue Position  $P(t) = (x(t), y(t), \theta(t))$  wird dann:

$$(3.22) \quad x(t) = \int_0^t v_x(\tau) d\tau = \int_0^t v(\tau) \cos(\theta(\tau)) d\tau$$

$$(3.23) \quad y(t) = \int_0^t v_y(\tau) d\tau = \int_0^t v(\tau) \sin(\theta(\tau)) d\tau$$

$$(3.24) \quad \theta(t) = \int_0^t w(\tau) d\tau$$

Mit der Annahme, dass  $v(t)$  in  $\delta t = [0, t]$  konstant ist, und  $v_r \neq v_l$ , und mit der Hilfe von (3.13) folgt:

$$(3.25) \quad \theta(t) = \int_0^t \frac{v_r - v_l}{2d} d\tau = \frac{v_r - v_l}{2d} \int_0^t d\tau$$

$$(3.26) \quad \theta(t) = \frac{v_r - v_l}{2d} t$$

Von (3.26) und (3.12) können  $x(t)$  und  $y(t)$  berechnet werden:

$$(3.27) \quad x(t) = \int_0^t v(\tau) \cos(\theta(\tau)) d\tau = \frac{v_l + v_r}{2} \int_0^t \cos\left(\frac{v_r - v_l}{2d} \tau\right) d\tau$$

$$(3.28) \quad x(t) = \frac{v_l + v_r}{v_r - v_l} d \sin\left(\frac{v_r - v_l}{2d} t\right)$$

und

$$(3.29) \quad y(t) = \int_0^t v(\tau) \sin(\theta(\tau)) d\tau = \frac{v_l + v_r}{2} \int_0^t \sin\left(\frac{v_r - v_l}{2d} \tau\right) d\tau$$

$$(3.30) \quad y(t) = \frac{v_l + v_r}{2} \int_0^t \sin\left(\frac{v_r - v_l}{2d} \tau\right) d\tau$$

$$(3.31) \quad y(t) = \frac{v_l + v_r}{v_r - v_l} d \left(1 - \cos\left(\frac{v_r - v_l}{2d} t\right)\right)$$

wenn  $v_r = v_l$ , folgt:

$$(3.32) \quad x(t) = \frac{v+v}{2} \int_0^t \cos\left(\frac{v-v}{2d}\tau\right) d\tau = v\delta t$$

$$(3.33) \quad y(t) = 0$$

$$(3.34) \quad \theta(t) = 0$$

Die Position und die Orientierung  $\langle x, y, \Theta \rangle$  bezüglich des Weltkoordinatensystems  $\Sigma$  ist:

$$(3.35) \quad x = \begin{cases} \frac{v_l + v_r}{v_r - v_l} d \sin\left(\frac{v_r - v_l}{2d} t\right) & \text{wenn } v_r \neq v_l \\ v \cos(\theta_0) t & \text{wenn } v_r = v_l \text{ und } \theta(0) = \theta_0 \end{cases}$$

$$(3.36) \quad y = \begin{cases} \frac{v_l + v_r}{v_r - v_l} d \left(1 - \cos\left(\frac{v_r - v_l}{2d} t\right)\right) & \text{wenn } v_r \neq v_l \\ v \sin(\theta_0) t & \text{wenn } v_r = v_l \text{ und } \theta(0) = \theta_0 \end{cases}$$

$$(3.37) \quad \theta(t) = \theta_0 + \frac{v_r - v_l}{2d} t$$

### 3.5.2 Rückwärtskinematik

“If I want this to happen, what should I do?”

Mit Hilfe der Vorwärtskinematik wurde bei gegebener Geschwindigkeit die Position des Roboters ermittelt. Sehr viel häufiger stellt sich die Frage, *welche Werte die Geschwindigkeiten  $v$  and  $w$  annehmen müssen, damit der Roboter eine bestimmte Pose (Position + Orientierung) erreichen kann*. Die Aufgabe der Rückwärtskinematik (auch **inverse Kinematik** genannt) besteht nun genau in der Antwort dieser Frage. Im Gegensatz zur Vorwärtskinematik existiert für die Rückwärtskinematik kein allgemeines Verfahren, mit dem ein mobiler Roboter explizite Lösungsformeln für die Geschwindigkeiten haben kann.

# Kapitel 4

## Bewegungsregelung des Roboters

Um mobile Roboter in ihrem Umfeld einsetzen zu können, sind eine genaue und robuste **Regelung** notwendig.

### 4.1 Grundlagen der Regelungstechnik

Der Begriff “Regelung” beschreibt Tätigkeiten, bei denen eine physikalische Größe Beeinflußt, so dass eine gewünschte Betriebsart eingestellt wird oder erhalten bleibt. Die Regelungstechnik ist in vielen technischen und natürlichen Systemen benutzt. Einige Beispiele sind im folgenden aufgeföhrt:

- Raumtemperaturre
- Körpertemperature
- Drehzahlregelung eines Motors

#### 4.1.1 Unterschied Steuerung - Regelung

Im alltäglichen Sprachgebrauch werden die Begriffe Steuerung und Regelung nicht immer sauber unterschieden, und manchmal widersprechen der technischen Definition des Steuern und Regeln. Zum Beispiel beim “Steuern” eines Auto (Abbildung 4.1) wird ständig die augenblickliche Fahrtrichtung mit der vom Fahrbahnverlauf Richtung vergleicht. Bei Abweichungen wird der Fahrer den Kurs entsprechend korrigieren. Dieser Vorgang wiederholt sich kontinuierlich, daher wird in der Regelungstechnik von einem Regelkreis gesprochen.

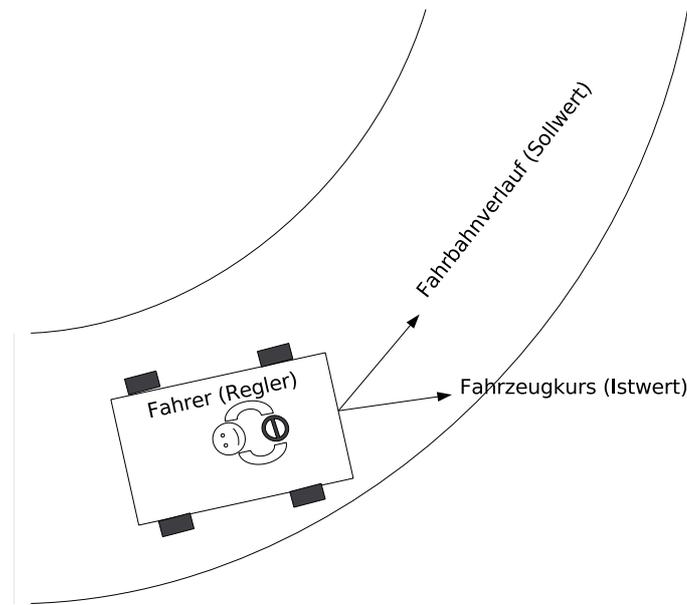


Abbildung 4.1: Autofahren: “Steuern” oder “Regeln”?

Ein anderes Beispiel ist der “Lautstärkeregel” beim Radiogerät. Die Lautstärke ändert sich bei gleicher Einstellung, egal ob das empfangene Signal stark oder schwach ist. Es wird keinen Vergleich zwischen die jetzige und die gewünschte Stärke stattfinden, also kein Regelkreis.

Bei diesen bekannten Beispielen aus dem Alltagsleben treten bereits die ersten Unterschiede zwischen Steuerung und Regelung deutlich hervor.

### Steuerung (Open Loop Control)

Die Steuerung (Abbildung 4.2) wirkt auf das Eingangssignal und beeinflusst damit das Ausgangssignal. Dieses Vorgehen erzeugt einen rein vorwärts gerichteten Signalfluss, und damit hat es den Vorteil, dass es die Stabilität des Systems nicht gefährdet. Es hat aber entscheidende Nachteile. Die Steuerung hat keine Kenntnisse, ob das Ausgangssignal die gewünschte Wert hat. Da die meisten technischen Systeme unterliegen externer Störeinflüssen, kann es bei einer Steuerung zu Abweichungen zwischen den gewünschten und aktuellen Ausgangsgrößen kommen. Deshalb, diese Strategie kann nur in sehr einfachen Fällen angewendet werden. Instabile Systeme können überhaupt nicht mit einer Steuerung kontrolliert werden.

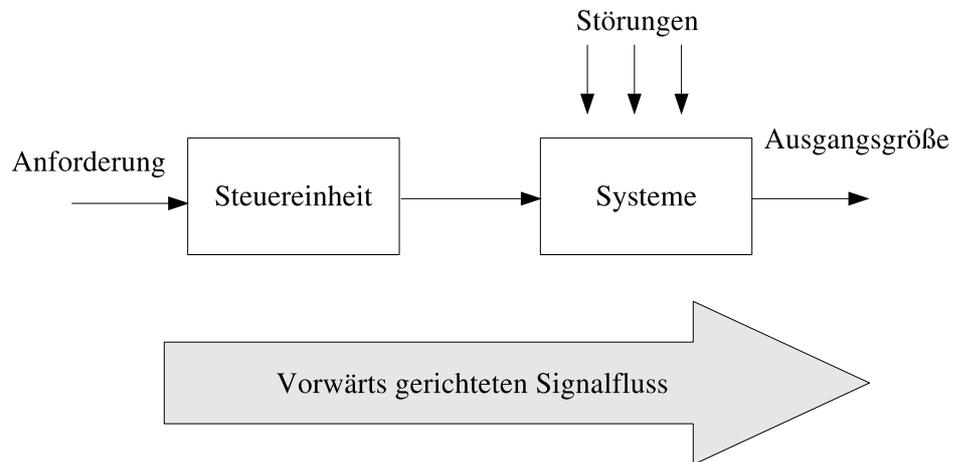


Abbildung 4.2: Steuerung

### Regelung (Closed Loop Control)

Die Regelung wirkt auch auf das Eingangssignal um das Ausgangssignal zu beeinflussen, aber in diesem Vorgehen "weiß" man, ob das Ausgangssignal die gewünschte Wert hat. Im Gegensatz zur Steuerung, in der Regelung entsteht ein Kreis, in dem Signale laufen (*Regelschleife*) (Abbildung 4.3). Es wird dann ein Vergleich mit den gewünschten Sollwerten stattfinden, und somit kann eine bessere Unterdrückung der Störeinflüsse auf das System gegeben werden. Im Unterschied zur Steuerung, eine Regelung kann instabile Systeme stabilisieren und damit technisch nutzbar machen.

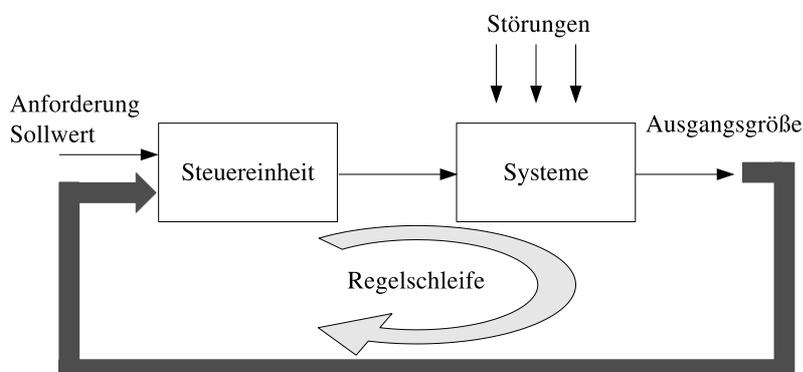


Abbildung 4.3: Regelung

### 4.1.2 Grundsätzlicher Aufbau von Regelkreisen

Die Regelung ist eine systematische Beeinflussung einer Eingangsgröße eines zu regelnden Systems, damit die Ausgangsgröße, die als *Regelgröße* bezeichnet, des Systems, den Wunschwert möglichst schnell und genau erreicht. Der ‐Wunschwert‐ der *Regelgröße* wird in der Praxis als *Sollwert* oder auch als *Führungsgröße* bezeichnet. Bei einem Regelprozess werden die *Regelgröße* und der *Sollwert* ständig gemessen und verglichen. In Abhängigkeit des Ergebnisses dieses Vergleichs, ergibt der Regler eine so genannte *Stellgröße* als Eingangsgröße, um die *Regelgröße* an den *Sollwert* heranzuführen und dort zu halten (Bild 3).

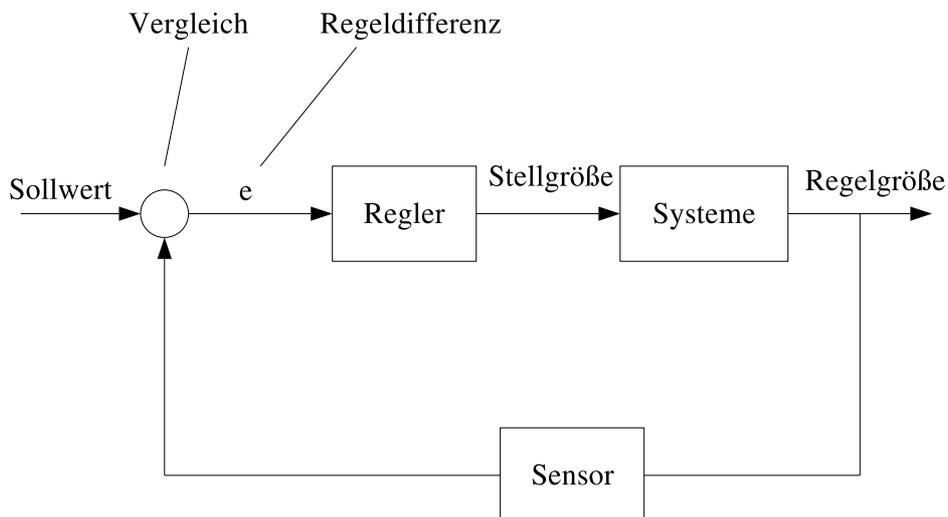


Abbildung 4.4: Blockschaltbild der Regelung

### 4.1.3 Allgemeine Anforderungen an einen Regelkreis

Grundaufgabe einer jeden Regelung ist es, dass der geschlossene Regelkreis genau und robust ist. Diese Anforderung können quantifiziert werden. Es wird daher Kenngrößen für die Sprungantwort des Regelkreises festgelegt.

Kenngrößen, die durch das Führungsverhalten bei Anregung mit Sprungfunktion erzielt werden, sind (Abbildung 4.5 ):

**Anregelzeit (rise time)  $T_{an}$**  gibt den Zeitpunkt an, zu dem erstmals der Sollwert erreicht wird.

**Ausregelzeit (transient time)  $T_{aus}$**  gibt den Zeitpunkt an, ab dem die Regelab-

weichung kleiner als eine vorgegebene Schranke  $\pm\varepsilon$ .

**Maximale Überschwingweite (overshoot)  $e_{max}$**  gibt die maximale Regelabweichung an, die nach dem erstmaligen Erreichen des Sollwertes (100%) auftritt.

**Bleibende Regeldifferenz (steady-state error)** ist die Abweichung der Sollwert von der Istwert für  $t \rightarrow \infty$

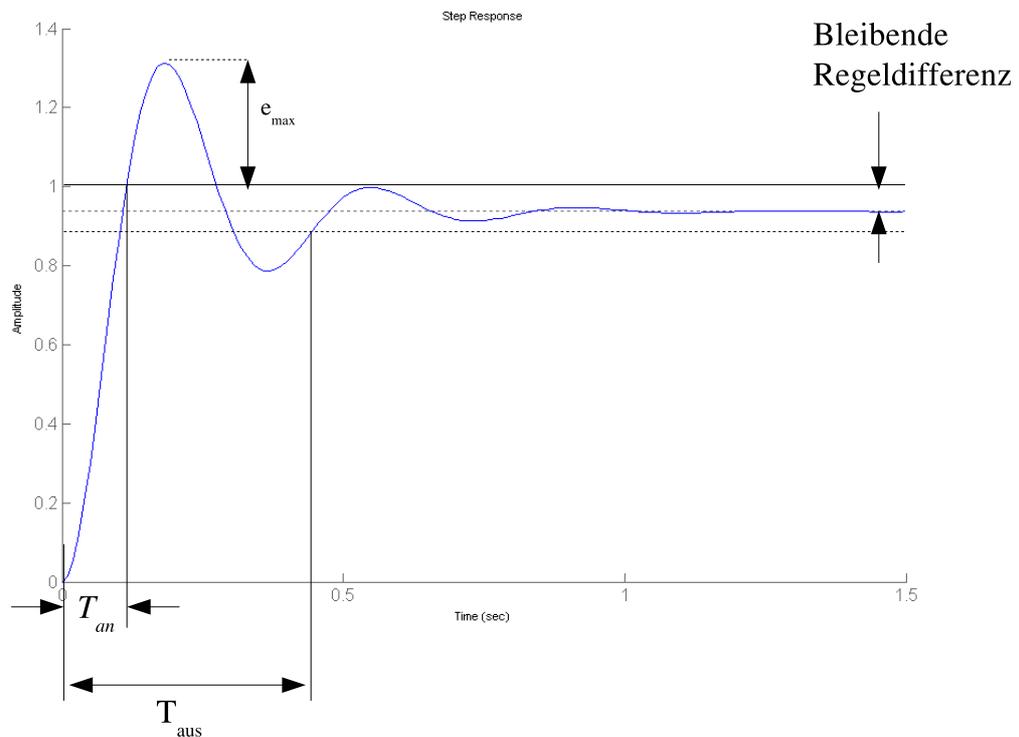


Abbildung 4.5: Sprungantwort des Regelkreises

An eine Regelung werden vier Anforderungen gestellt:

- Kleine bleibende Regelabweichung.
- geringes Überschwingen.
- Kleine Anregelzeit.
- Kleine Ausregelzeit.

#### 4.1.4 Lineare Standard-Regler (PID Regler)

PID-Regler besteht aus folgenden Anteilen:

- Proportionalen Anteil (P-Regler)
- Integralen Anteil (I-Regler)
- Differentialen Anteil (D-Regler)

##### **P-Regler**

*P*-Regler, auch *P*-Glied genannt, ist das einfachste Element in einer Regelungskreis. Er bestimmt die Stellgröße  $u(t)$  durch Multiplikation der Regelabweichung  $e(t)$  mit einem Faktor ( $K_p$ ):

$$(4.1) \quad u(t) = K_p e(t)$$

##### **Vorteile:**

- einfach im Aufbau und damit relative preiswert.
- das Verhalten wird nur durch die Einstellung des Parameters  $K_p$ .

**Nachteil:** bleibende Regeldifferenz. Diese wird umso größer, je kleiner die Kreisverstärkung ( $K_p$ ) ist, und Eine Erhöhung des Faktors  $K_p$  wird zur Erhöhung der Überschwingweite führen.

##### **I-Regler**

Beim *I*-Regler (integrierender Regler) ist die Stellgröße  $u(t)$ , abgesehen vom Anfangswert, proportional zum Zeitintegral der Regelabweichung  $e(t)$ :

$$(4.2) \quad u(t) = \frac{1}{K_i} \int_0^t e(\tau) d\tau$$

Die Aufgabe des *I*-Reglers is es, die Regeldifferenz nicht zuzulassen.

**Vorteil:** keine bleibende Regeldifferenz.

**Nachteil:** *I*-Regler greifen langsamer ein als *P*-Regler.

### **D-Regler**

Der D-Regler (differentialer Regler) bestimmt den Stellwert aus der zeitlichen Ableitung der Regelabweichung:

$$(4.3) \quad u(t) = K_d \frac{d}{dt} e(t)$$

Der *D*-Regler wird also aus der Änderung der Regeldifferenz reagieren. Je schneller sich die Regeldifferenz verändert, desto größer ist der Stellwert. Wenn die Regeldifferenz konstant bleibt, wird der *D*-Regler diese nicht ausgleichen.

**Vorteil:** Beim Auftreten einer Regeldifferenz greift der *D*-Regler sofort proportional zur Änderungsgeschwindigkeit ein. Der Vorteil ist es, das Überschwingen zu minimieren.

### **Nachteile:**

- reagiert nicht auf eine konstante Regeldifferenz.
- verstärkt Messrauschen.

### **PI-Regler**

Die Kombination eines *I*-Reglers mit einem *P*-Regler führt zum *PI*-Regler. Der *PI*-Regler vereint die Wirkungsweisen von *P*- und *I*-Reglern in sich. Der *P*-Reglerteil versucht eine auftretende Regeldifferenz schnell abzufangen, und die *I*-Reglerteil beseitigt anschließend die restliche Regeldifferenz.

Das dynamische Verhalten des *PI*-Reglers kann durch die folgende Gleichung beschrieben werden:

$$(4.4) \quad u(t) = K_p e(t) + \frac{1}{K_i} \int_0^t e(\tau) d\tau$$

### **Vorteile:**

- anfangs schneller als der *I*-Regler.
- präziser als der *P*-Regler (keine bleibende Regeldifferenz)

**Nachteil:** Einstellung von zwei Reglerparametern (Zeit Aufwand beim Tuning)

**PD-Regler**

Die Parallelschaltung eines  $D$ -Reglers und eines  $P$ -Reglers führt zum  $PD$ -Regler. Er ist durch die folgende Gleichung beschrieben:

$$(4.5) \quad u(t) = K_p e(t) + K_d \frac{d}{dt} e(t)$$

**Vorteil:** sehr schneller Regler, und größere Regeldifferenz werden vermieden.

**Nachteil:** bleibende Regeldifferenz tritt auf.

**PID-Regler**

Der PID-Regler vereint die Verhaltensweisen der  $P$ -,  $I$ -, und  $D$ -Regler. Folgende Gleichung beschreibt das dynamische Verhalten eines  $PID$ -Reglers:

$$(4.6) \quad u(t) = K_p e(t) + \frac{1}{K_i} \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

**Vorteile:**

- reagiert schnell
- läßt keine bleibende Regeldifferenz zu
- erlaubt im Verlauf der Regelung keine großen Regelabweichungen

**Nachteil:** Einstellung eines PID-Regler erfordert einen großen Zeitaufwand

Wir fassen die Merkmale der elementaren PID-Regler in der folgenden Tabelle zusammen.

Regler	Anregelzeit	Überschwinger	Ausregelzeit	Regelabweichung
$K_p$	sinkt	wächst	geringe Änderung	sinkt
$K_I$	sinkt	wächst	wächst	0
$K_D$	geringe Änderung	sinkt	sinkt	geringe Änderung

## 4.2 Geschwindigkeitsregelung (Dynamics Control)

Um ein sauberes Fahren zu ermöglichen, muss die Drehgeschwindigkeit der Motoren geregelt werden. Dabei werden die Motoren nicht kontinuierlich mit Strom versorgt, sondern in kurzen Impulsen (PWMs). Je nach Verhältnis zwischen Impuls und Pause dreht sich der Motor schneller oder langsamer (siehe Section 2.2). Die Geschwindigkeitsregelung erfolgt mit Hilfe eines PID-Reglers (Abb. 4.6). Der Radenkoder gibt dem PID-Regler Informationen über die Aktuelle Geschwindigkeit des Rads. Durch die Stellgröße, hier die Frequenz und das Tastverhältnis der PWM, wird die Regelgröße also die Rad-Geschwindigkeit verändert (geregelt).

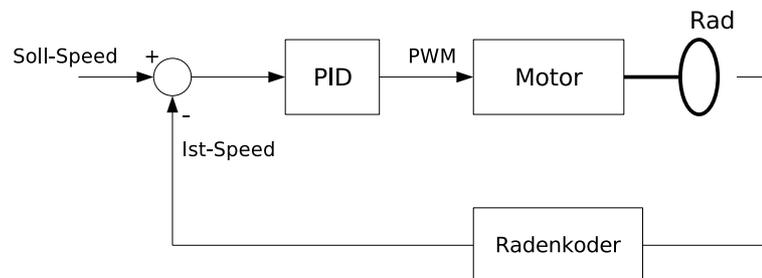


Abbildung 4.6: Motor-Regelung mit PID-Regler

Um die Geschwindigkeit des Roboters zu regeln, müssen wir eine Transformationfunktion zwischen die lineare- und Winkelgeschwindigkeit ( $v$ ,  $w$ ) berechnen. Sei  $w_r$ ,  $w_l$ ,  $v_r$ ,  $v_l$  die Rotations- und Translationsgeschwindigkeiten der beiden Räder (rechts und links). Dann durch die Gleichung (3.9) und die Abbildung (4.7) wird erreicht

$$(4.7) \quad v_r = w_r \cdot R$$

und

$$(4.8) \quad v_l = w_l \cdot R$$

Dann mit der Hilfe von (3.12), (3.13), (4.7), und (4.8) kann die Transformationsfunktion  $T$  zwischen  $(v, w)$  und  $(w_r, w_l)$  durch

$$(4.9) \quad \begin{pmatrix} w_r \\ w_l \end{pmatrix} = \begin{pmatrix} \frac{1}{R} & \frac{d}{R} \\ \frac{1}{R} & \frac{-d}{R} \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}$$

berechnet werden.

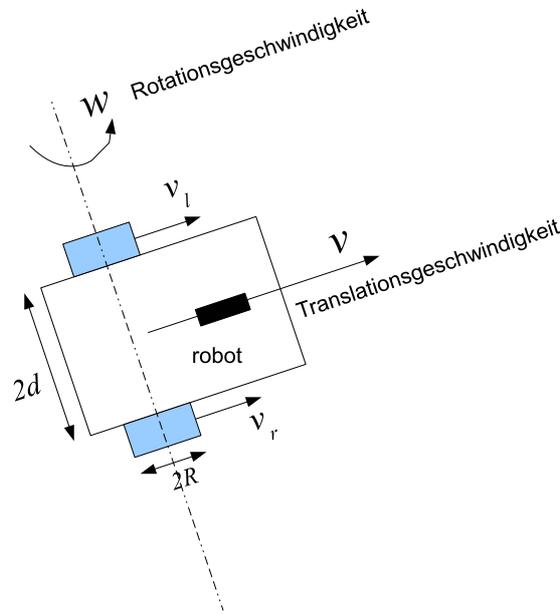


Abbildung 4.7: Geschwindigkeiten eines mobilen Roboters mit Differentialantrieb.

Die Abbildung (4.8) zeigt die detaillierte Geschwindigkeitsregelung eines mobilen Roboters mit Differentialantrieb.

## 4.3 Positionsregelung (Kinematics Control)

### 4.3.1 Problemdarstellung

Das Problem ist es, geeignete Translationsgeschwindigkeiten und Rotationsgeschwindigkeit  $(v, w)$  zu finden, sodass der Roboter eine bestimmte Position erreichen kann ( $\Delta l \rightarrow 0$ ) (Abb. 4.9). Für den allgemeinen Fall ist eine Lösung schwierig und nicht eindeutig zu finden. Es liegt darin, dass sich Einschränkungen in den Bewegungsmöglichkeiten eines mobilen Roboters ergeben können. Eine wichtige Einschränkung ist es, ob der Roboter sich *holonom* oder *nicht-holonom* bewegt.

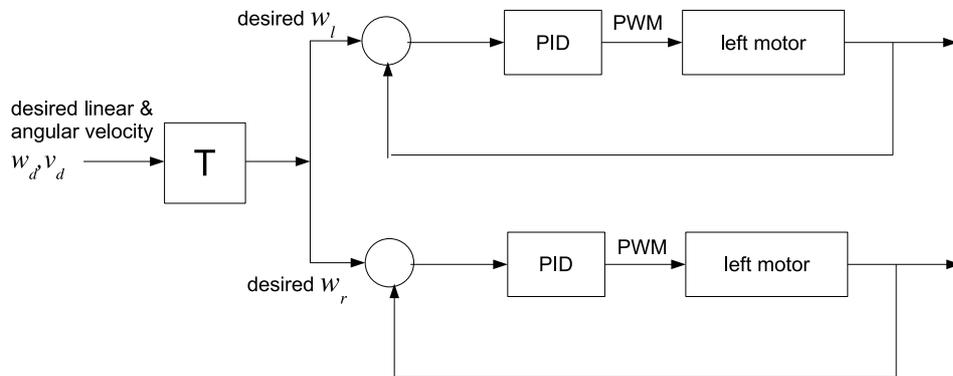


Abbildung 4.8: Geschwindigkeitsregelung mit PID-Regler für ein mobiler Roboter mit Differentialantrieb.

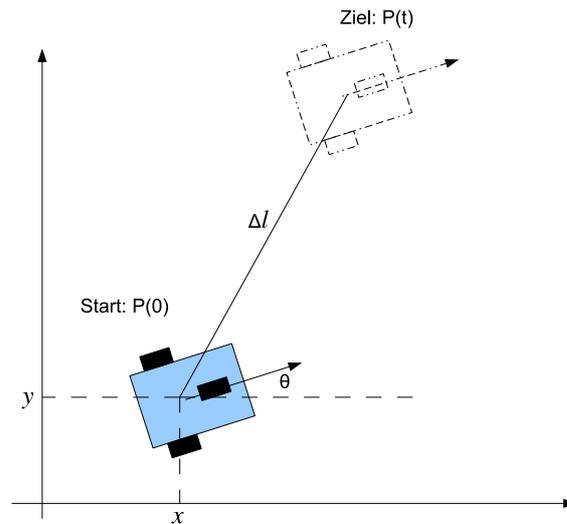


Abbildung 4.9: Problemstellung der Positionsregelung

### 4.3.2 Holonom vs. Nicht-holonom

Die Eigenschaft, ob sich ein Roboter holonom oder nicht-holonom bewegen kann, hat großen Einfluss auf die Positionierung von mobilen Robotern. Ein holonomer Roboter, z.B. mit Mecanum-Rädern, kann ohne Rotationsbewegung jeden Punkt im 2D Raum erreichen. Das bedeutet in der Praxis, dass so ein Roboter als Punkt für die Bewegungsregelung betrachtet werden kann. Bei einem nicht holonomen Roboter

muss berücksichtigt werden, dass nicht jede Richtung direkt steuerbar ist. Ein nicht holonomes Beispiel ist ein mobiler Roboter mit Differentialantrieb. Er kann sich seitwärts bewegen, aber nicht auf direktem Weg: *zuerst Rotation dann Translation*. Dieses Problem ist beim Auto-Einparken bemerkbar, in dem man vorwärts und rückwärts fahren muss um seitwärts einparken zu können. Beim Differentialantrieb ist die Unabhängigkeit der Koordinaten  $x, y, \theta$  (Abb. 4.10) durch folgende Bedingung eingeschränkt:

$$(4.10) \quad \dot{x} \cos \theta - \dot{y} \sin \theta = 0$$

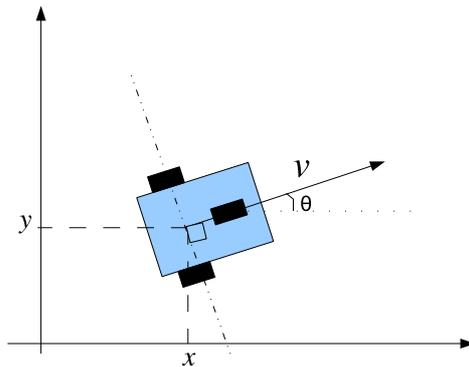


Abbildung 4.10: Non holonomic constraint beim Roboter mit Differentialantrieb. Die Einschränkung  $\dot{x} \cos \theta - \dot{y} \sin \theta = 0$  bedeutet in der Praxis, dass die Räder keine Bewegung seitwärts zulassen sondern nur in die momentane Fahrtrichtung (pure Translation) oder eine pure Rotation wenn  $v = 0$ .

### 4.3.3 Lösungen

Für ein Roboter mit Differentialantrieb ist eine Lösung schwierig und nicht eindeutig zu finden. Es liegt darin, dass keine direkte Regelung von  $\Delta l$  in der Kinematik-Gleichung (4.9) möglich ist (Nicht-holonome Einschränkung). Um dieses Problem zu vermeiden, definiert man eine zweite Variable  $\Delta \varphi$  (Abb. 4.12). Das Ziel jetzt ist es dann, geeignete  $(v, w)$  zu finden, sodass  $\Delta l \rightarrow 0$  und  $\Delta \varphi \rightarrow 0$ . Dies kann mit Hilfe von zwei PID-Regler erreicht werden.

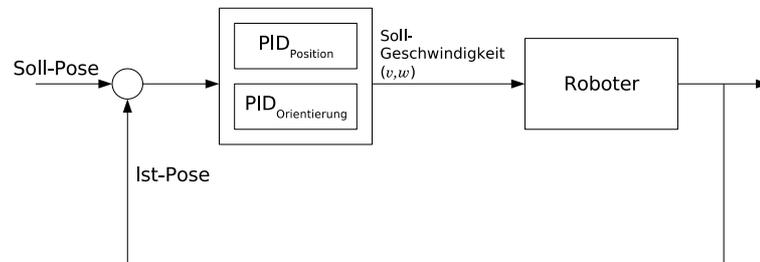


Abbildung 4.11: Positionsregelung mit PID-Regler

**Lösung 1 (pure Translation und pure Rotation)** : Häufiger werden  $\Delta l$  und  $\Delta\varphi$  getrennt und nicht gleichzeitig geregelt, d.h. sich drehen richtung Ziel und dann geradeaus fahren. In dieser Lösung werden die Bewegungen durch puren Translationen und Rotationen geplant (rolling and no slipping condition!).

**Lösung 2** : Hier werden  $\Delta l$  und  $\Delta\varphi$  getrennt, aber gleichzeitig, geregelt. Dazu nehmen wir ein “pseudo-Ziel”, sodass  $\Delta l$ , indirekt, durch die Parameters  $\Delta\varphi$  und  $\Delta\lambda$  geregelt werden kann (Abb. 4.12). Mit der Zeit werden  $\Delta\varphi \rightarrow 0$ , pseudo-Ziel  $\rightarrow P_{Ziel}$ ,  $\Delta\lambda \rightarrow 0$ , und damit  $\Delta l \rightarrow 0$ .

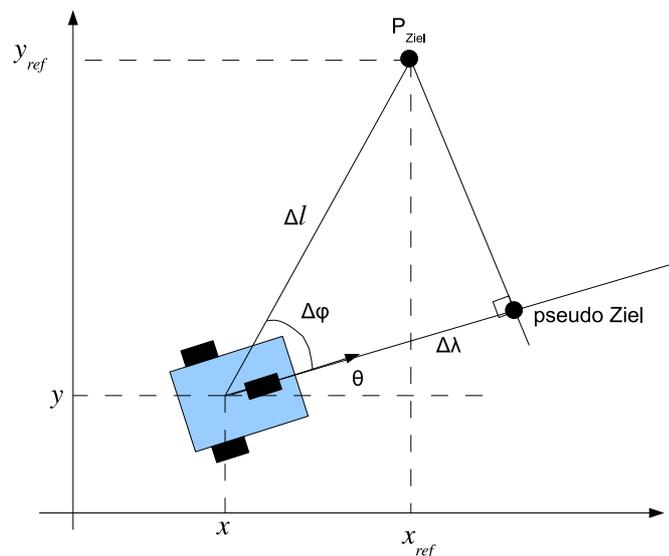


Abbildung 4.12: Problemstellung der Positionsregelung

Folgende Gleichungen gelten

$$(4.11) \quad \Delta\varphi = \tan^{-1} \left( \frac{y_{ref} - y}{x_{ref} - x} \right) - \theta$$

$$(4.12) \quad \Delta\lambda = \Delta l \cos \Delta\varphi$$

$$(4.13) \quad \Leftrightarrow \Delta\lambda = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2} \cos \Delta\varphi$$

die Idee jetzt ist es, wenn  $\Delta\varphi \rightarrow 0$  und  $\Delta\lambda \rightarrow 0$ , dann  $\Delta l \rightarrow 0$ . Es kann aber nicht Garantiert werden, dass dass der Roboter eine Stabile Bewegung führt. Das ist erwartet, weil die zwei Regler total Unabhängig von einander funktionieren.

Abbildung (4.13) zeigt die gesamte (Geschwindigkeitsregelung+Positionsregelung) Bewegungsregelung des Roboters.

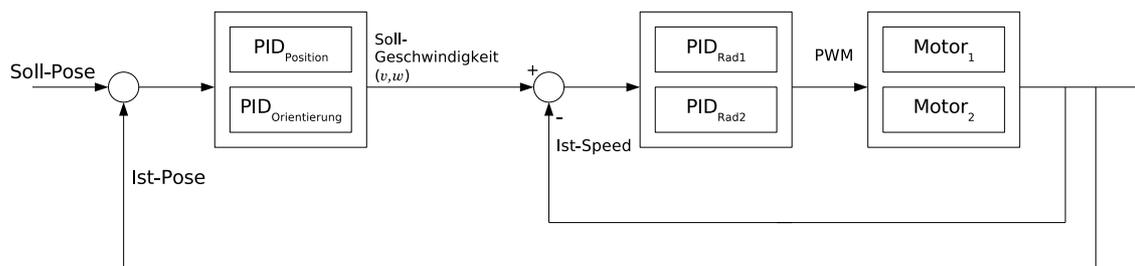


Abbildung 4.13: Die gesamte Bewegungsregelung eines mobilen Roboters mit 2 Rädern

# Kapitel 5

## Navigation

Ein mobiler Roboter sollte fähig sein, einen Auftrag in der Art: "Fahre zu einem Zielort", ausführen zu können. Damit ein Roboter diese Aufgaben erfüllen kann, muß er in der Lage sein zu navigieren. Er soll sich also effizient und kollisionsfrei in seiner Umgebung bewegen. Navigation ist eine der wichtigsten Aufgaben eines mobilen Roboters. Die folgenden drei Grundfragen der Navigation muß ein autonomer mobiler Roboter bei der Erfüllung seiner Aufgabe beantworten können: Wo bin ich? wohin gehe ich? wie gelange ich dorthin? Deshalb lässt die Navigation sich in drei große Gebiete unterteilen:

- . **Kartenerstellung:** Auswertung der erfassten Sensordaten der Umgebung mit dem Ziel, ein Umgebungsmodell zu schaffen.
- . **Lokalisierung:** Ermitteln der aktuellen Position des Roboters in dem Umgebungsmodell.
- . **Bahnplanung:** Berechnung der Fahrtroute vom aktuellen Punkt zum Zielort.

Die ersten beiden Punkte sind in der Literatur unter dem Begriff **SLAM** (simultaneous localizing and mapping) behandelt, und sie sind nicht Teil dieser Vorlesung.

### 5.1 Kartenerstellung

Ein Roboter benötigt eine Karte aus den selben Gründen wie ein Mensch:

- Die Selbstlokalisierung. Viele Verfahren zur Bestimmung der eigenen Position verwenden Umgebungskarten.

- Wichtige Informationen “Orte” können eingetragen werden, um systematische Aktionen im Zusammenhang mit diesen Orten durchzuführen, z.B. Treppen sollten von einem Wheeled-Robot vermeiden werden.
- Die Bahnplanung. Ist die Umgebung bekannt kann eine effiziente Wegeplanung zum Ziel durchgeführt werden.

Der Roboter dient zuerst als “Datensammler”. Er soll beliebig fahren, um die Umgebung zu erkunden. Dabei ist zu beachten, dass genügend Daten aufgenommen werden sollten. Dann werden Odometriedaten und Daten von anderen Sensoren mittels mathematischer Methoden zusammengeführt, um eine Karte zu erstellen. Es existieren eine Reihe von Verfahren, z.B. “Robotic Mapping: A Survey” von Sebastian Thrun (2002).

## 5.2 Lokalisierung

Die Lokalisierung bezieht sich auf die Frage “Wo bin ich?”. Ohne die Beantwortung dieser Frage kann die Bahnplanung und die sich anschließende Fahrt nicht durchgeführt werden. Sie behandelt also das Problem, heraus zu finden, wo sich der Roboter relativ zu seiner Umgebung befindet. Im Folgenden werden einige wichtige Konzepte herausgestellt.

### 5.2.1 Dead-Reckoning

Die internen Sensoren (z.B. Odometrie) messen kleine Positionsveränderungen des Roboters, die zur Bestimmung der Position aufsummiert werden. Da alle Sensoren fehlerbehaftet sind, wächst der Fehler der bestimmten Position unbeschränkt mit der Zeit.

### 5.2.2 Lokalisierung mittels Landmarken

Bei der Lokalisierung mit Landmarken werden natürliche (z.B. Säulen, Türrahmen, Wändecken, Möbel,...) oder künstliche Landmarken (z.B. Ampeln, Lichtquellen, Ultraschallquellen,...) in der Umgebung verwendet, deren Position im globalen Weltmodell bekannt ist. Dabei ist es wichtig, dass die Landmarken unabhängig von Blickwinkel und Entfernung erkannt werden können.

### 5.2.3 Probabilistische Lokalisierungsverfahren

Probabilistische Lokalisierungsverfahren benutzen die Bayes-Regel, um den Zustand des Weltmodells mittels Sensordaten zu aktualisieren. Insbesondere, wird für jede mögliche Position die Wahrscheinlichkeit, dass der Roboter sich dort befindet, berechnet. Markov-Lokalisierung und Monte-Carlo-Lokalisierung sind Methoden, die zur Probabilistische Lokalisierungsverfahren gehören.

## 5.3 Bahnplanung

Mit Hilfe der Bahnplanung kann eine kollisionsfreie Trajektorie von einer Start- zu einer Zielkonfiguration errechnet werden. Vor der Planung soll bekannt sein, was zu tun ist (z.B. “Finde einen Weg vom Start zum Ziel”, oder “Finde einen kürzesten Weg vom Start zum Ziel”, oder auch “Finde einen Weg vom Start zum Ziel ohne die Umgebung komplett zu kennen”).

### 5.3.1 Globale Bahnplanung

Die globale Bahnplanung hat die Aufgabe, das Ziel zu verfolgen und dabei bekannte lokale Sackgassen und Hindernisse zu vermeiden. Hierzu benötigt die globale Navigation zuerst ein globales Modell der Umgebung. Dieses Modell ist eine Umgebungskarte, in der alle statischen Hindernisse verzeichnet sind.

**Problemstellung:** Sei einen Konfigurationsraum  $C$ , der alle möglichen Konfigurationen  $q$  enthält.  $C$  enthält einen freien Konfigurationsraum  $C_{free}$  und einen blockierten Konfigurationsraum  $C_{obstacle}$ . In diesem Raum wird der Roboter als ein Punkt modelliert, der bestimmte Konfigurationen  $q_r$  haben kann, sodass die Planung unabhängig von der tatsächlichen Robotergeometrie durchgeführt werden kann. Das Problem zum Lösen ist dann, einen freien Pfad zu finden, also eine Folge von Konfigurationen  $q$ , die die Startkonfiguration mit der Zielkonfiguration verbinden (Abb. 5.1). In dem globalen Fall ist  $C_{free}$  und  $C_{obstacle}$  von Beginn an durch die Umgebungskarte komplett bekannt.

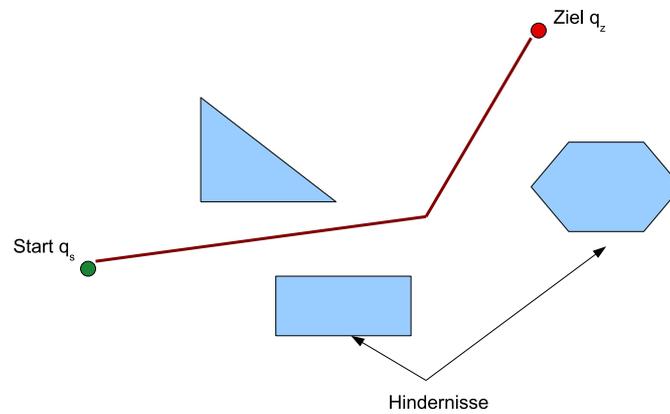


Abbildung 5.1: Problemstellung: Finde einen freien Pfad, also eine Folge von Konfigurationen  $q_i \in C_{free}$ , die die Startkonfiguration  $q_s$  mit der Zielkonfiguration  $q_z$  verbinden!

**Sichtbarkeitsgraph Methode (visibility graphs):** Zuerst werden der Startpunkt, das Ziel, und die Ecken der Hindernisse mit Knoten identifiziert. dann werden die Knoten mit Kanten verbunden, wenn sie sich gegenseitig sehen. Sehen bedeutet, dass eine Strecke zwischen zwei Knoten kein anderes Hindernis schneidet. Die gefundene Pfadsegmente werden mit der Start- und Zielposition verbunden. Danach kann der kürzesten Pfad im Graphen gesucht werden (Abb. 5.2).

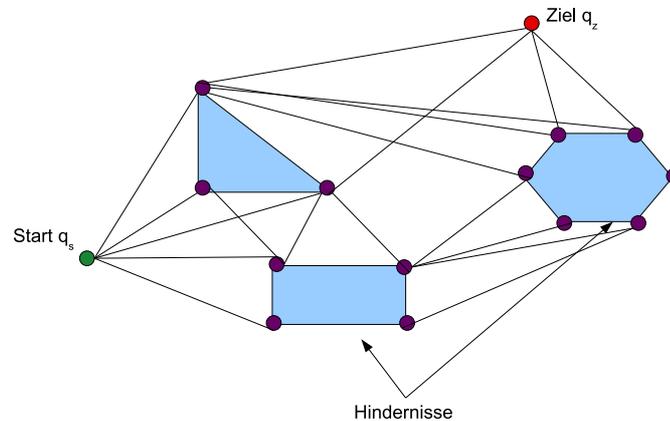


Abbildung 5.2: Sichtbarkeitsgraph. Startpunkt, Zielpunkt, und die Eckpunkte der Hindernisse werden mit Knoten identifiziert. Die Knoten werden mit Kanten verbunden, wenn sie sich gegenseitig sehen. Das Algorithm blickt dann von Knoten zu Knoten und baut so die Roadmap für den Weg vom Start zum Ziel auf.

### 5.3.2 Lokale Bahnplanung

Lokale Bahnplanung hingegen ist eine Navigation in der nahen Umgebung des Roboters. Sie benutzen nur die aktuellen Sensorinformationen, um einen kollisionsfreien Pfad zum Ziel zu finden. Hierbei wird ein lokales Navigationsziel angestrebt. Dies kann z.B. Fahren in eine Zielrichtung, das Folgen eines Korridors, oder das Durchfahren einer Tür sein. Eine der Hauptaufgaben der lokalen Navigation ist die reaktive Kollisionsvermeidung.

#### “Bug” Methoden:

Bei Verwendung der “Bug” Methoden fährt der Roboter auf ein vorgegebenes Ziel zu. Trifft er auf ein Hindernis, folgt er so lange dessen Kontur bis das Hindernis umgefahren wird.

**Beispiel:** A Bug algorithm “Bug1” for the scenarion of Abb. (5.3).

#### Bug1:Repeat

1. Head toward the goal.
2. If the goal is attained then stop.
3. If contact is made with an obstacle then follow the obstacle boundary (toward the left) until heading toward the goal is possible again.



**Potentialfeldern:** Im Bereich der Robotik wurden Potentialfelder erstmals von Oussama Khatib (AI Lab, Stanford University, 1986) zur Realisierung einer Echtzeit-Hindernisvermeidung sowohl für mobile Roboter als auch für Roboterarme eingesetzt. Die Grundidee ist es, dass der Roboter als Punkt im Konfigurationsraum betrachtet wird, der unter Einfluss von künstlichen Kräften vom Zielpunkt und den Hindernissen steht. Der Zielpunkt besitzt ein anziehendes Potential, und allen anderen Objekten jeweils repulsive Felder wirken (Abb. 5.5). Durch die Berechnung des Gradienten ist es möglich, die Richtung des Vektorfeldes und somit eine Bewegungsrichtung zu bestimmen, in der der Roboter kollisionsfrei zum Ziel gelangen kann.

$$(5.1) \quad U(q) = U_{tar}(q) + U_{obst}(q)$$

where  $U$ ,  $U_{tar}$ ,  $U_{obst}$  denote artificial potential field, the attractive potential from the target, and the repulsive potential from obstacles, respectively. And  $q$  denotes the robot pose. The most commonly used attractive potential has the form:

$$(5.2) \quad U_{tar}(q) = \xi \lambda^n(q, q_{tar})$$

where  $\xi$  is a positive scaling factor,  $\lambda(q, q_{tar}) = \|q_{tar} - q\|$  is the distance between the robot pose  $q$  and the target  $q_{tar}$ , and  $n = 1$  or  $n = 2$ .

The commonly used repulsive potential function has the form:

$$(5.3) \quad U_{obst}(q) = \begin{cases} \eta \left( \frac{1}{\lambda(q, q_{obst})} - \frac{1}{\lambda_{th}} \right), & \lambda(q, q_{obst}) \leq \lambda_{th} \\ 0, & \lambda(q, q_{obst}) > \lambda_{th} \end{cases}$$

where  $\eta$  is a positive scaling factor,  $\lambda(q, q_{obst})$  is the distance between the robot pose  $q$  and the obstacle position  $q_{obst}$ . The parameter  $\lambda_{th}$  denotes a threshold distance, beyond no repulsive force will be taken into consideration by the robot.

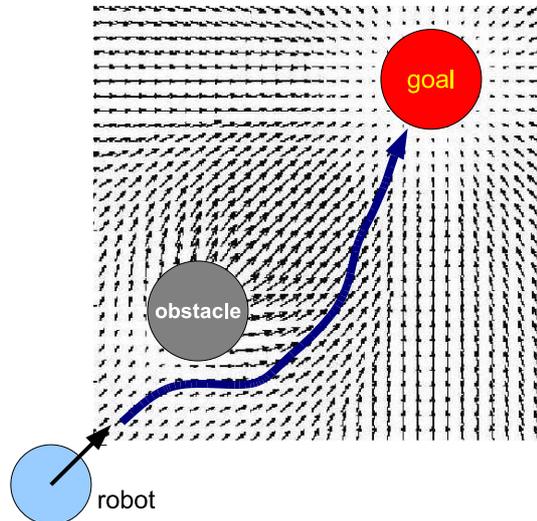


Abbildung 5.5: Goal acquisition and obstacle avoidance using artificial potential fields

**Virtual Force Field:** Virtual Force Field Methode ist eine Verbesserung der Potentialfelder Methode, die zur Navigation verwendet wird. Ergänzt wird sie durch das Certainty Grid Konzept, das für die Positionsbestimmung der Hindernisse zuständig ist. Anhand dieser berechnet die Potentialfelder-Methode die abstoßende Kraft. In Abbildung (5.6) sieht man eine Beispielsituation. Der Roboter und das Hindernis befindet sich in einem Grid. Die einzelnen Kräfte des Hindernisses werden zu der resultierenden abstoßenden Kraft  $F_o$  aufsummiert. Das Ziel (Target) übt eine anziehende Kraft  $F_t$  auf den Roboter aus. Diese zwei Kräfte ergeben im Kräftegleichgewicht die resultierende Kraft  $F_r$ , in dessen Richtung der Roboter dann fährt. Das Certainty Grid wurde ursprünglich zur Vermeidung von Problemen durch Sensorungenauigkeit verwendet. Der Wert einer Zelle wird inkrementiert, wenn die Sensoren ein Hindernis erkennen. Wird ein Hindernis nur einmal erkannt, ist es auf fehlerhafte Sensoren zurückzuführen und wird bei der Hindernisvermeidung nicht weiter als Hindernis beachtet. Ist der Wert einer Zelle größer als 1, befindet sich dort ein Hindernis. Wenn kein Hindernis vorhanden ist, hat die Zelle den Wert 0.

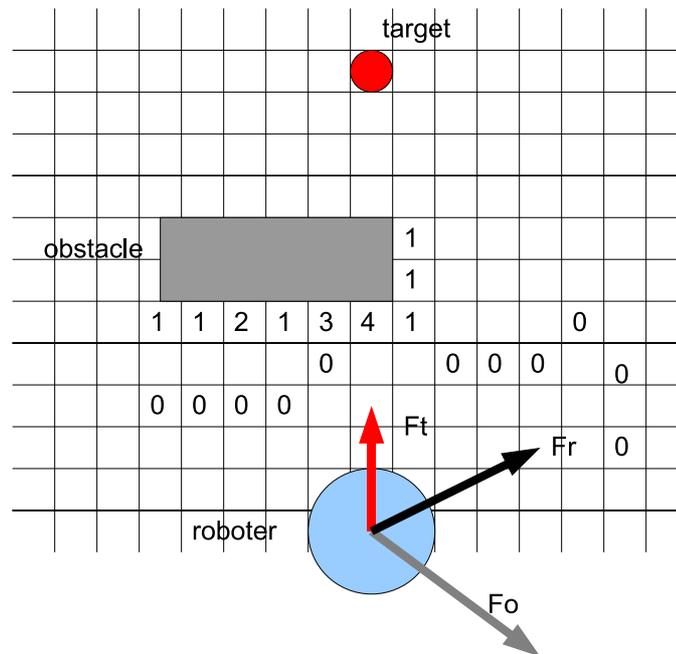


Abbildung 5.6: Grundidee der VFF-Methode

## 5.4 Steuerungsarchitekturen

Eine Steuerungsarchitektur ist die Struktur der Steuerung eines Systems. Für ein Roboter stellt eine Steuerungsarchitektur ein Softwaresystem sowie die Spezifikationen zur Konstruktion des Roboters zur Verfügung. Zu den wichtigsten Architekturen gehören “Deliberativ”, “Reaktiv” sowie die Kombination “Hybrid”. Sie setzen drei abstrakten primitiven Funktionen **sense**, **plan** und **act** in Beziehung.

Unter **sense** versteht man, alles was mit der Wahrnehmung der Umgebung zu tun hat.

In den Bereich **plan** werden die Daten mit schon vorhandenen Informationen verknüpft, und daraus einen Plan erstellt, der z.B die Fortbewegung des Roboters bestimmt.

Die dritte Funktion ist **act**. Hierunter versteht man die Handlung, wodurch sich der Roboter bewegt.

### 5.4.1 Deliberative Architektur

Die deliberative Architektur (eng. deliberate) (Abb. 5.7) ist die älteste Steuerungsarchitektur für mobile Roboter. Dabei werden die Einheiten des Roboters hierarchisch angeordnet. Aus den Sensorinformationen wird ein Welt-Modell erstellt. Aus diesem Modell können jederzeit wieder Informationen, wie Entfernungen von Objekten, abgerufen werden. Die zweite Funktion “plan” ist dafür zuständig, Lösungswege für Aufgaben zu finden. Diese Ebene ist von der Qualität des Welt-Modells total abhängig, und von der wirklichen Welt isoliert. Der fertige Plan wird danach zur Ausführung in der Hierarchie nach unten an die Steuerungseinheiten aus dem Bereich “act” geleitet.

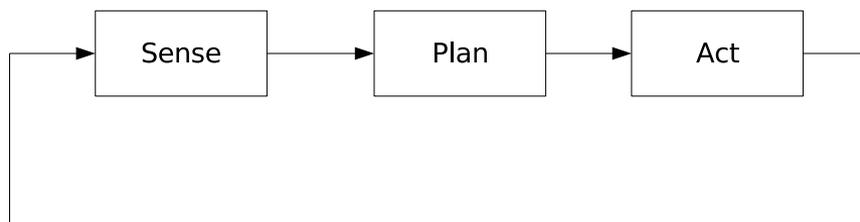


Abbildung 5.7: Deliberative Architektur: Sense-Plan-Act (SPA) Architektur hat drei Teil-Komponenten: Die Wahrnehmung, Planung und Handlung.

**Vorteil:** Auf der Grundlage des Weltmodells können optimale Lösungen gefunden werden.

**Nachteile:**

- Alle Module sind voneinander abhängig (wenn ein Module ausfällt, das gesamte System fällt aus).
- Veränderungen der Umgebung erfordern Anpassung des Weltmodells, Die viel Rechen-Aufwand benötigt. Das System ist langsam.

### 5.4.2 Reaktive Architektur

Hier reagiert der Roboter direkt auf seine Wahrnehmung. Die Konsequenz ist ein kompletter Verzicht auf das Welt-Modell aus dem Bereich “sense”, und auf die Planungseinheit. Ein Vorbild dazu ist der Mensch. Beim Gehen z.B muss er nicht jede

Muskelbewegung einzeln berechnen.

Die Struktur wird somit wesentlich einfacher (Abb. 5.8). Anstatt erst ein Welt-Modell zu erstellen, arbeiten die Sensoren mit den Aktoren direkt zusammen.

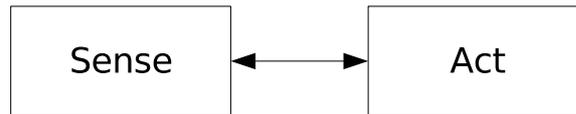


Abbildung 5.8: Reaktive Architektur: Der Roboter handelt direkt als Reaktion auf seine Wahrnehmung.

**Vorteil:** Die Rechenschritte laufen direkt bei den Sensoren und Aktoren ab. Dadurch kann sich der Roboter neuen Situationen schnell anpassen.

**Nachteil:** Da die Rechenschritte vor Ort realisiert werden, kann der Roboter kein längerfristiges Ziel verfolgen.

### 5.4.3 Hybride Architektur (Reaktive/Deliberative)

Eine Kombination aus reaktivem und deliberativem Ansatz stellt die Hybride Architektur dar (Abb. 5.9). Dabei werden die kurzfristigen Handeln und Planen getrennt. Dadurch können die einzelnen Vorteile beider Architekturen genutzt werden. Reaktive Handlungen werden z.B für die Hindernissevermeidung eingesetzt. Soll der Roboter zusätzlich ein langfristiges Ziel verfolgen, kommt die deliberative Steuerung und Planung ins Spiel.

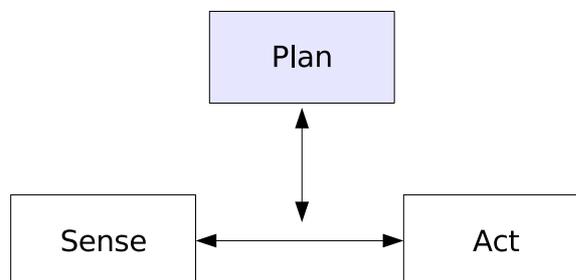


Abbildung 5.9: Hybride Architektur: Der Roboter verfolgt ein längerfristiges Ziel, hat aber auch untergeordnete Prozesse, die reaktiv sind.

**Vorteil:** Durch die reaktive Handlungsweise reagiert der Roboter schnell, kann aber auch längerfristige Ziele verfolgen.

**Nachteil:** Wie alle Hybrid-Lösungen kann es leicht zu Konflikten kommen, z.B. dass Nachteile beider Lösungen sich vereinen.