



Karlsruher Institut für Technologie

ZUSAMMENFASSUNG

# Biologisch Motivierte Robotersysteme

Sophie v. SCHMETTOW  
SoSe 15

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>6</b>
1.1 Bionik . . . . .	6
1.2 Klassische Robotik . . . . .	6
1.3 Bionik + Robotik . . . . .	7
1.4 Motivation . . . . .	7
1.5 Begriffserklärung . . . . .	7
1.6 Systemanforderungen BioBots . . . . .	7
1.7 Anwendungsbeispiel . . . . .	9
1.8 Anwendungsfelder . . . . .	9
<b>2 Biomechanik</b>	<b>10</b>
2.1 Leichtbau (Knochen/Skelett) . . . . .	10
2.1.1 Biologische Grundprinzipien und typische Merkmale . . . . .	10
2.1.2 Materialauswahl und Reduzierung . . . . .	11
2.1.3 Konstruktionsprinzipien . . . . .	11
2.1.4 Technische Materialien – Werkstoffe im Roboterbau . . . . .	12
2.1.5 Verhältnis: Kraft zu Gewicht . . . . .	12
2.2 Kinematische Struktur (Morphologie) . . . . .	13
2.2.1 Gelenke . . . . .	13
2.2.2 Multisegmentsysteme . . . . .	13
2.2.3 Laufmaschine (Invertebrate, Vertebrate) . . . . .	13
2.2.4 Lokomotionsarten . . . . .	15
2.2.5 Arbeitsraumvergrößerung durch Anstellwinkel . . . . .	16
2.3 Energie–Effizienz . . . . .	16
2.3.1 Reduktion des Strömungswiderstandes . . . . .	16
2.3.2 Energieeffizienz beim Laufen . . . . .	16
2.3.3 Beispiele für Energiebilanzen in der Natur . . . . .	18
<b>3 Muskeln</b>	<b>18</b>
3.1 Der biologische Muskel – das Wirkprinzip . . . . .	18
3.1.1 Muskelaufbau und unterschiedliche Muskelformen . . . . .	18
3.1.2 Prinzipielle Funktionsweise . . . . .	19
3.1.3 Muskel als Effektor des Nervensystems . . . . .	20
3.2 Klassische Antriebe in der Robotik . . . . .	21
3.3 Künstliche Muskelsysteme . . . . .	22
3.3.1 Mechanischer Muskel . . . . .	22
3.3.2 Chemische Muskeln (SMP, LAP) . . . . .	22
3.3.3 Elektroaktive Polymere (EAP) . . . . .	22
3.3.4 Piezoelektrische Antriebe . . . . .	22
3.4 Fluidische Muskeln . . . . .	22
3.4.1 Spinnenprinzip . . . . .	22
3.4.2 Weitere Muskeln . . . . .	22
3.4.3 McKibben-Muskel . . . . .	23
3.5 McKibben Prinzip . . . . .	23
3.5.1 Aufbau und Funktionsweise am Beispiel FESTO-MAS . . . . .	23
3.5.2 Verschiedene Muskelmodelle – Simplizität vs. Präzision . . . . .	24
3.6 Antagonistische Regelung . . . . .	26
3.6.1 Antagonistisches Gelenk . . . . .	26
3.6.2 Zusammenspiel zweier Fluidischer Muskeln . . . . .	28
3.6.3 Regelungsbeispiel für ein flexibles Gelenk . . . . .	29
3.7 Anwendungsbeispiele . . . . .	29
3.7.1 Vergleich biologischer vs. fluidischer Muskel . . . . .	31

## Inhaltsverzeichnis

<b>4 Sensoren</b>	<b>32</b>
4.1 Sinneseindrücke und Sensoren – Übersicht und Klassifikation . . . . .	32
4.1.1 Klassifikation von Sinneseindrücken . . . . .	32
4.1.2 Rezeptoren und Reizverarbeitung . . . . .	32
4.1.3 Sensorik in der Technik . . . . .	32
4.2 Sensorsysteme in Biologie und Technik – Eine vergleichende Einführung . . . . .	35
4.2.1 Hören – Von der Fledermaus zum Ultraschallsensor . . . . .	35
4.2.2 Gelenkwinkel – Wie ist die Extremität orientiert? . . . . .	37
4.2.3 Gravitation bzw. Beschleunigung – Innenohr gegen Inertialsystem . . . . .	38
4.2.4 Temperatur – Wo ist es warm? . . . . .	39
4.2.5 Tasten – Schreibtischplatte oder Kaktus... . . . . .	40
4.2.6 Riechen und Geschmack – Chemie der Stoffe . . . . .	42
4.2.7 Sehen – Facetten, Kameras und der optische Fluss . . . . .	44
<b>5 Bewegungssteuerung</b>	<b>46</b>
5.1 Neuronale Kontrolle in der Natur . . . . .	46
5.1.1 Struktur vs. Parametrisierung . . . . .	46
5.1.2 Zentraler Mustergenerator vs. Reflexe . . . . .	47
5.1.3 Das Neuron: Funktionale Approximation, KNN . . . . .	48
5.2 Neuronale Steuerung der Stabheuschrecke . . . . .	50
5.2.1 Warum Stabheuschrecke als Vorbild? . . . . .	50
5.2.2 Laufmuster und Cruse-Regeln . . . . .	50
5.3 Oszillatoren als Grundbausteine für natürliche Bewegungssteuerung . . . . .	51
5.3.1 Rhythmusgeneratoren, Erweiterung des Neuronen Modells . . . . .	51
5.3.2 Oszillatormodelle: Leaky Integrator, Matsuoka . . . . .	53
5.4 Anwendungsbeispiele für Oszillatoren . . . . .	55
5.4.1 Neunauge, Salamander – Schwimmen und Laufen, Tekken – Vierbeiniges Laufen . . . . .	55
5.4.2 Zusammenfassung & Diskussion . . . . .	55
<b>6 Mehrbeinige Lokomotion</b>	<b>55</b>
6.1 Lokomotionsarten . . . . .	56
6.2 Stabilität beim Laufen . . . . .	56
6.2.1 Statische Stabilität . . . . .	56
6.2.2 Dynamische Stabilität . . . . .	57
6.2.3 Stabilitätskriterien . . . . .	57
6.3 Laufmuster . . . . .	58
6.3.1 Unterscheidungskriterien für Laufmuster . . . . .	59
6.3.2 Sechsbeinige, vierbeinige, zweibeinige Laufmuster . . . . .	61
6.4 Mehrbeiniges Laufen . . . . .	62
6.4.1 Zentrale Muster Generatoren (CPG) . . . . .	62
6.4.2 Verhaltens- und regelbasierte Steuerung . . . . .	62
6.5 Zweibeiniges Laufen . . . . .	63
6.5.1 Biomechanische Ansätze menschliches Laufen zu beschreiben . . . . .	63
6.5.2 Technische Umsetzung dieser Prinzipien auf reale Systeme . . . . .	67
<b>7 Rechnerarchitektur</b>	<b>70</b>
7.1 Neuronen, Synapsen und Nervenfasern . . . . .	70
7.1.1 Das Neuron als Basisrecheneinheit . . . . .	70
7.2 Hierarchien der Informationsverarbeitung im Gehirn . . . . .	70
7.2.1 Das Zentrale Nervensystem . . . . .	70
7.2.2 Aufteilung der Rindenfeldes . . . . .	70
7.2.3 Aufteilung der Großhirnhälften . . . . .	71
7.2.4 Stabheuschrecke – Abstraktion des Steuerungskonzeptes . . . . .	71

## Inhaltsverzeichnis

7.3	Rechnerarchitektur für biologisch motivierte Roboter . . . . .	72
7.3.1	Anforderungen an die Rechnerarchitektur . . . . .	72
7.3.2	Systemkomponenten – die einzelnen Bausteine . . . . .	72
7.3.3	Kopplung – Bus-Systeme . . . . .	75
7.4	Beispielarchitekturen . . . . .	76
7.4.1	Rechnerarchitektur von LAURON IVc . . . . .	76
7.4.2	Rechnerarchitektur von LAURON V . . . . .	77
7.4.3	Rechnerarchitektur von KAIRO 3 . . . . .	77
7.5	Gegenüberstellung Rechnerarchitektur in Natur und Technik . . . . .	78
<b>8</b>	<b>Verhalten</b>	<b>78</b>
8.1	Verhalten im Tierreich: Neurologisch, Psychologisch, Ethologisch . . . . .	79
8.1.1	Neurologisch – Vektorfeldaddition im Frosch-Rückenmark . . . . .	79
8.1.2	Psychologisch – Verhaltens-, Gestalt- und Kognitive Psychologie . . . . .	79
8.1.3	Ethologisch – Reflexe, Taxe und feste Muster . . . . .	79
8.2	Verhaltensbasierte Robotik – Übersicht und Einführung . . . . .	79
8.2.1	Motivation – Warum verhaltensbasierte Architekturen . . . . .	80
8.2.2	Roboter Steuerungs-Spektrum – Von reaktiv bis deliberativ . . . . .	80
8.2.3	Die Anfänge . . . . .	80
8.3	Beispiele für verhaltensbasierte Architekturen . . . . .	81
8.3.1	Anforderung an die Architektur / Bewertungsschema . . . . .	81
8.3.2	Subsumption (MIT) . . . . .	81
8.3.3	Verhaltensnetzwerke/ Verhaltenssteuerung . . . . .	82
8.3.4	Verhaltensnetzwerke (FZI) . . . . .	85
<b>9</b>	<b>Lernende Ansätze</b>	<b>90</b>
9.1	Motivation und Einführung in lernende Systeme . . . . .	90
9.1.1	Gradientenabstieg . . . . .	91
9.1.2	Reinforcement Learning – Zuckerbrot und Peitsche in der Robotik . . . . .	91
9.2	Verschiedene Möglichkeiten der Wissensrepräsentation . . . . .	93
9.2.1	Cerebellar Model Articulation Controller (CMAC) . . . . .	93
9.2.2	Radial Basis Functions (RBF) . . . . .	93
9.3	Algorithmen zum Online-Lernen . . . . .	93
9.3.1	Adaptive Heuristic Critic (AHC) mit Stochastic Real Value Exploration (SRV) . . . . .	93
9.3.2	Zustandsraum Randwertkontrolle mit Boundary Controllern (BC) . . . . .	95
9.4	Anwendung . . . . .	96
9.4.1	Zweibeiniges Laufen mit Hilfe von CMAC und SRV . . . . .	96
9.4.2	Online-Lernen auf der vierbeinigen Laufmaschine BISAM . . . . .	97
9.4.3	Diskussion . . . . .	97
<b>10</b>	<b>Humanoide Service- und Assistenzroboter</b>	<b>100</b>
10.1	Übersicht aktueller, humanoider Robotersysteme . . . . .	100
10.1.1	Fragestellungen bei der Entwicklung Humanoider . . . . .	100
10.1.2	Beispiele humanoider Roboter . . . . .	100
10.1.3	Uncanny Valley . . . . .	101
10.2	ARMAR der humanoide Roboter aus Karlsruhe . . . . .	101
10.2.1	Zielsetzung des ARMAR Projekts . . . . .	101
10.2.2	ARMAR-III . . . . .	102
10.3	Programmieren humanoider Roboter . . . . .	103
10.3.1	Einführung . . . . .	103
10.3.2	Programmierung von Manipulationsaufgaben . . . . .	103
10.4	Perzeption für humanoide Roboter . . . . .	105
10.4.1	Detektion und Modellierung menschlicher Operator und Umwelt . . . . .	105
10.4.2	Akustische Szenen-Analyse und Stereo Vision . . . . .	105

## *Inhaltsverzeichnis*

10.4.3 Modeling of Scenes and Objects . . . . .	106
10.4.4 Colored Object Recognition . . . . .	106
10.5 Anwendungen und Einsatz . . . . .	106
<b>11 Mutation</b>	<b>107</b>
11.1 Vererbung und Evolution in der Biologie . . . . .	107
11.1.1 Die Mendelschen Gesetze . . . . .	107
11.1.2 Zellteilung, Vererbung und Mutation . . . . .	108
11.2 Genetische Algorithmen – „Evolution im Rechner“ . . . . .	109
11.2.1 Genetische Algorithmen – Motivation und Nomenklatur . . . . .	109
11.2.2 Grundalgorithmus . . . . .	110
11.2.3 Generierung von Nachkommen . . . . .	110
11.2.4 Selektion . . . . .	111
11.2.5 Anwendungen . . . . .	113
11.3 Künstliche Ontogenese: Entwicklung von Agenten durch künstliche Zellen . . . . .	115
11.4 Selbstkonfigurierende Systeme – Mutierende Roboter . . . . .	116
11.4.1 Kurzer Überblick (AIST, Japan) – Selbstkonfiguration = Evolution? Mutation? . . . . .	116
11.4.2 Reales 2D System . . . . .	116
11.4.3 3D Selbstkonfigurierendes Robotersystem . . . . .	116
<b>12 Anwendungen</b>	<b>117</b>

## Disclaimer

Dieses Dokument wurde im Rahmen des Master-Studiums für das Modul „Autonome Robotik“ erstellt. Es stellt eine Zusammenfassung dar und dient zur Vorbereitung auf die mündliche Prüfung. Neben den Materialien der Vorlesung fließen auch weitere Quellen ein, um den behandelten Stoff auszuarbeiten. Auf den Verweis von Quellen wird verzichtet, da die Erstellung keinerlei wissenschaftlichen Zweck verfolgt und nur für den privaten Gebrauch bestimmt ist.

## 1 Einleitung

### 1.1 Bionik

*Viel Forschungs- und Entwicklungsarbeit kann man sich sparen, wenn man die Problemlösungen bei biologischen Systemen studiert.*

- Biologisch inspirierte Forschung
- Systematisches Lernen von der Natur
- Bionik ist die Umsetzung von biologischen Problemlösungen in die Technik
- Anwendungsbereiche: Konstruktionen, Sensorik, Bewegung, Algorithmen...

Beispiele:

- In der Luftfahrt: Inspiration Vogelflug
  - Gezielte Luftverwirbelung am Ende der Tragfläche
  - Treibstoffverbrauch um ca. 3 – 5% reduziert
- Im Alltag: Inspiration Lotusblume
  - Schmutzabweisende Oberflächen durch Nanostrukturen (z.B. Fassadenfarben)
- Im Alltag: Inspiration Klettfrüchte
  - Textilien Klettverschluss basiert auf Mikrohaken

### 1.2 Klassische Robotik

Vier Robotergenerationen:

1. Generation: (programmierbare Manipulatoren, 1960 – 1975)
  - geringe Rechenleistung, nur feste Haltepunkte (Punkt-zu-Punkt-Programmierung), kaum sensorielle Fähigkeiten (Pick-and-Place-Aktionen)
2. Generation: (adaptive Roboter, 1976 – 1982)
  - mehr Sensoren (z.B. Kameras), Anpassung an Umwelt, eigene Programmiersprache (z.B. VAL), geringe Roboter-Intelligenz (adaptive Aufgabendurchführung)
3. Generation: (autonome Roboter, ab 1982)
  - hohe Rechenleistung (Multiprozessorsysteme), Aufgabenorientierte Programmierung, Forderung nach (maschineller) Autonomie
4. Generation: (humanoide AI-Roboter)
  - Flexibilität bzgl. Umwelt und Aufgabe, Lernfähigkeit, Selbstreflexion, Emotion

Anwendungsfelder: Industrie-Roboter, Service-Roboter, Personal-Roboter

### 1.3 Bionik + Robotik

#### Systematisches Lernen von der Natur mit Anwendung in der Robotik

- Biologisch motivierte Roboter
- Umsetzung von biologischen Problemlösungen in der Robotik
- Konstruktion von biologisch inspirierten Robotern, Bewegungssteuerung, Evolutionäre Algorithmen, etc.

### 1.4 Motivation

Die Roboter-Akzeptanz hängt primär von dessen Natürlichkeit ab: Begeisterung für natürliche, biologisch motivierte Systeme.

Vorteile biologisch motivierter Roboter:

- leicht, robust, anpassungs- und lernfähig, energetisch optimiert, fehlertolerant, flexibel, flüssige Bewegungen (hardwareschonend)
- Integration in menschliches Umfeld
- Keine Roboter-Infrastruktur notwendig

### 1.5 Begriffserklärung

- Biologisch motivierte Robotersysteme sollen primär durch ihre Lokomotionsformen definiert werden: **Schwimmen, Kriechen, Laufen, Hüpfen, Fliegen**
- Für Realisierung der Lokomotion werden Prinzipien aus der Natur übertragen: **Antrieb, Konstruktion, Umwelterfassung, Steuerung**
- Die **direkte Abbildung** aus der Natur ist **nicht notwendig** und meistens auch nicht sinnvoll/machbar

### 1.6 Systemanforderungen BioBots

- Autonomie und Mobilität
- Energieautarkie
- Hohe Flexibilität
- Robustheit gegenüber unterschiedlichen Störungen
- Umfangreiche perzeptive Komponenten zur Erfassung interner Zustände und der Umwelt
- Adaptive Verhaltenssteuerung
- Erweiterbarkeit (perzeptive Komponenten, Skills)

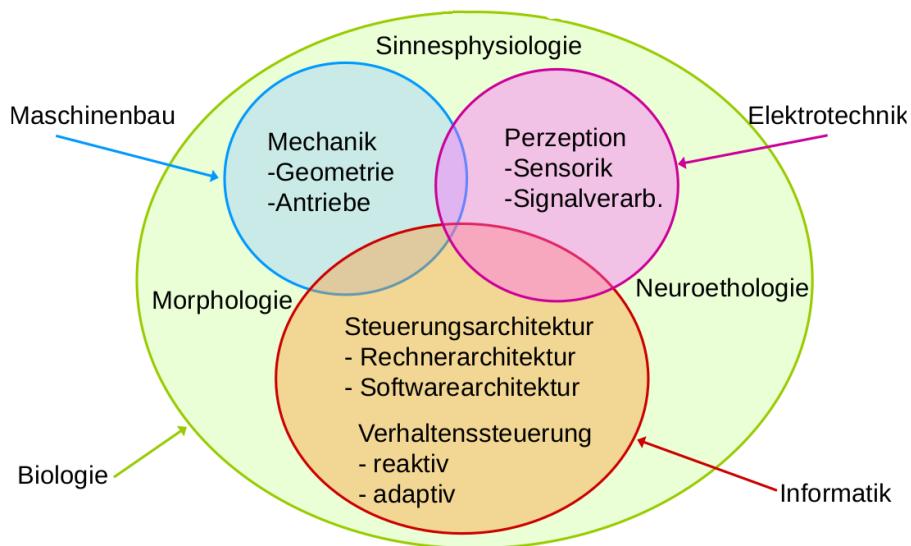


Abbildung 1: Interdisziplinärer Lösungsansatz

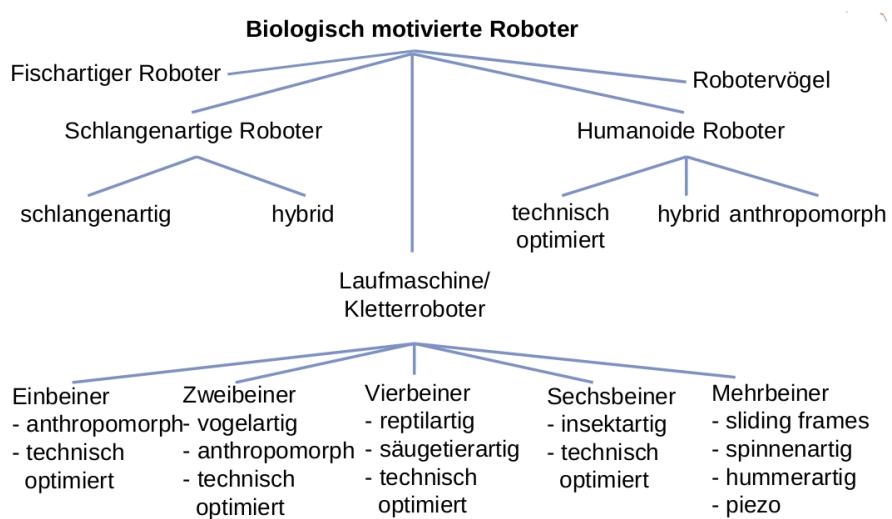


Abbildung 2: Systematische Einordnung

## 1.7 Anwendungsbeispiel

- Applikation
  - Leitzustand
  - Benutzerinteraktion
  - Missionsplanung
- Navigation
  - Lokalisation
  - Umweltmodell
  - Pfadplanung
- Lokomotion
  - Internes Modell
  - Lokale Adaption
  - Gangarten
  - Reflexe
- Basissteuerung
  - Gelenke
  - Beine/Körper
  - Motorprimitive
- Hardware
  - Mechanik
  - Aktuatoren
  - Sensorik
  - Elektronik

## 1.8 Anwendungsfelder

- Wartungs- und Inspektionsaufgaben
- Erkundung von rauem Gelände (Vulkane, Meeresboden, Planeten)
- Untersuchung von Meeresböden, Pipelines
- Forst- und Landwirtschaft
- Katastrophenschutz
- Prothetik
- Rehabilitation
- Assistenzaufgaben
- Entertainment

## 2 Biomechanik

Anforderung an die Mechanik – Was hat die Natur an Lösungen zu bieten

### 2.1 Leichtbau (Knochen/Skelett)

*Biologische Materialien werden für einen bestimmten Zweck und möglichst massearm hergestellt*

#### 2.1.1 Biologische Grundprinzipien und typische Merkmale

10 Grundprinzipien natürlicher Konstruktion:

1. Hohe Funktionsintegration statt Funktionstrennung (z.B. zahlreiche Funktionen menschlicher Haut)
2. Ganzheitliche Optimierung
3. Multifunktionalität statt Monofunktionalität
4. Minimaler Einsatz von Energie (Ökologische Nische muss durch bestmöglichen Energieeinsatz ausgefüllt werden)
5. Nutzung von Fremdenergie (z.B. Nutzung von Sonnenlicht durch Poikilotherme)
6. Zeitliche Limitierung statt Langlebigkeit (z.B. Eintagsfliegen; ständig nachwachsende Hautschichten)
7. Hohe Vernetzung / Wechselwirkung
8. Veränderungen in kleinen Schritten (z.B. Wachstum bei Kindern; inkrementelle Änderung von Generation zu Generation → ganz anders in der Robotik)
9. Feinabstimmung gegenüber der Umwelt
10. Totales Recycling (z.B. Häutung von Reptilien)

Typische Merkmale biologischer Materialien:

1. Materialschichtung während des Entstehens (z.B. Chitinschädel bei Insekten: zwischen Elementen elastisch/weniger Schichten, außen hart/mehr Schichten)
2. Streng funktionell (genaue Abstimmung auf Umweltanforderungen)
3. Funktionelle Kompartimente
4. Häufig Ultraleicht (Verhältnis Kraft/Gewicht)
5. Sandwich-Bauweise
6. Selbstreparatur (z.B. nachwachsende Beine der Stabheuschrecke; wäre tolles Feature für LAU-RON)
7. Terminierte Lebensdauer

### 2.1.2 Materialauswahl und Reduzierung

- Technische Möglichkeiten zur Gewichtsoptimierung
  - Menge (Volumen) des eingesetzten Materials minimieren:
    - \* Materialdicken reduzieren
    - \* Aussparungen einfügen (Materialmenge solange reduzieren, dass Funktion gerade noch erhalten wird; z.B. Spaichen beim Fahrrad)
  - Gewichte (Dichte) des eingesetzten Materials reduzieren
    - \* Austausch des verwendeten Materials (z.B. Aluminium statt Stahl, Karbon statt Aluminium, ...)
- Stabilität und Belastbarkeit muss erhalten bleiben!

### 2.1.3 Konstruktionsprinzipien

- Hohe Stabilität bei geringem Gewicht durch Netzwerk aus Knochenlammellen
- Hohlbauweise (Rohr mechanisch viel stabiler als ein Volleisenstab)
- Extrem stabil (ein menschlicher Oberschenkelknochen kann ca. 1t Längsbelastung aushalten)

#### Innenskelett:

- CFE – Rohre als Knochen
  - Kohlenstofffaser–Gewebe–Epoxy Verbundstoff
  - Leicht maschinell zu fertigen, aber schlecht nachbearbeitbar
  - Hohe Biege- und Torsionsfestigkeit
  - Sehr leicht ( $1,5g/cm^3$ )

#### Exoskelett:

- Chitin
  - Exoskelett ist Stützstruktur für Organismen mit stabiler äußerer Hülle
  - Stickstoffhaltiges Polysaccharid
  - Primärer Baustoff der Insekten
  - Unterschiedliche Einlagerungen führen zu unterschiedlichen Härtegraden
  - Schichtbauweise mit unterschiedlichen mechanischen Eigenschaften
- Glasfaser Verbundstoffe
  - Verbundwerkstoff aus Gewebematten (Glasfaser, Kohlefaser) und Harz (Epoxyd, Aramid)
  - Freie Formen möglich
  - Dicke lässt sich an kritischen Stellen erhöhen, an unkritischen senken
  - Relativ leicht ( $2,52g/cm^3$ )
  - Genutzt z.B. zur Gewichtsoptimierung bei Formel1; aber: viel Handarbeit/teuer

### 2.1.4 Technische Materialien – Werkstoffe im Roboterbau

Begriffe und Definitionen:

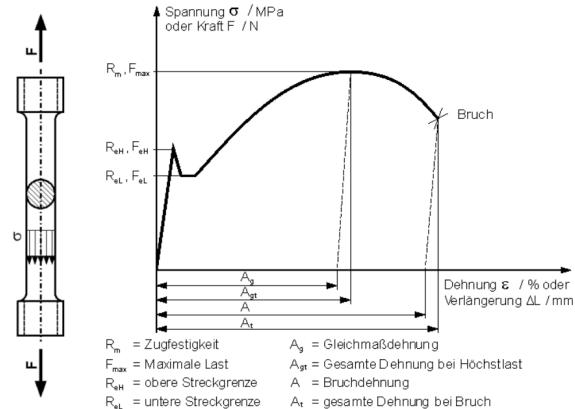
- Dichte  $\rho = \frac{m}{V} = \frac{\text{Masse}}{\text{Volumen}}$
- Zugfestigkeit  $R = \frac{F_{MAX}}{A} = \frac{\text{Kraft}_{MAX}}{\text{Flaeche}}$
- mechanische Spannung  $\sigma = \frac{F}{A} = \frac{\text{Kraft}}{\text{Flaeche}}$
- Dehnung  $\epsilon = \frac{\Delta l}{l_0} = \frac{\text{Laengenaenderung}}{\text{urspruengliche Laenge}} \text{ relative Längenänderung}$
- Elastizitätsmodul/E-Modul  $E = \frac{\sigma}{\epsilon} = \frac{\text{Spannung}}{\text{Dehnung}} = const$   
Materialkennwert beschreibt Zusammenhang zwischen Spannung und Dehnung
- Bruchfestigkeit: mechanische Spannung, die unter gleichmäßiger Steigerung der Belastung bei einem Bauteil zum Bruch führt

Unterschiedliche Anwendungen benötigen unterschiedliche Werkstoffe. Häufig ist auch ein Materialmix notwendig.

#### Werkstoffprüfung – Zugversuch

##### Werkstoffe im Roboterbau

	Dichte g/cm <sup>3</sup>	Zugfestigkeit N/mm <sup>2</sup>	E-Modul kN/mm <sup>2</sup>	Bruchfestigkeit %
GFK Glasf. Verst. Kunst.	2,52	300	73	2,8
CFE Carbonfibre Epoxy	1,5	350	50	1,5
POM Polyoxynethylen Kun.	1.42 ... 1.6	60 ... 140	2,5 ... 12	6 ... 15
AlCuMg Aluminiumlegierung	2,8	390	71,5	13
34CrNiMo6 Edelstahl	~ 8	1300	216	10



### 2.1.5 Verhältnis: Kraft zu Gewicht

- Klassischer Industrie-Roboter (KUKA KR 1000 TITAN von 2007):
  - Traglast 1000kg
  - Gewicht 4950kg
  - Kraft/Gewicht 0.2
- Mariusz Pudzianowski („stärkster Mensch der Welt“)
  - Traglast 415kg
  - Gewicht 138kg
  - Kraft/Gewicht ca. 3.0 (nur kurzfristig!)
- Ameise (hier: *Formica polyctena* - Waldameise)
  - Traglast 350mg
  - Gewicht 9 mg
  - Kraft/Gewicht 40
- Hauspferd
  - Leistung 0.4 – 1.2 PS (kurzfristige Spitzenleistung bis zu 20PS)
  - Gewicht 200 – 800kg
  - Kraft/Gewicht 0.16

## 2.2 Kinematische Struktur (Morphologie)

### 2.2.1 Gelenke

#### In der Natur

- Arten:
  - Kugelgelenk
  - Sattelgelenk
  - Scharniergelenk
- Eigenschaften:
  - Selbstschmierend (Knorpel, Gelenkflüssigkeit)
  - Gekapselt (Kapsel, Muskelgewebe, Haut, Chitopanzer)
  - Optimierte Geometrie für die Anwendung (z.B. menschliches Knie: geringer Reibwert aber hoher Wirkungsgrad; aber Kugelgelenke in Robotik schwer anzutreiben)

**Kugelgelenk – Technische Approximation:** Kugelgelenk der Stabheuschrecke wird durch drei Rotationsgelenke approximiert

**Gelenkkapselung:** Klassische mechanische Kapselung durch Simmer- oder Wellendichtringe meist zu schwer; Gummimembrane nur begrenzt haltbar, daher regelmäßige Wartung erforderlich

### 2.2.2 Multisegmentsysteme

Die **Wirbelsäule** als Beispiel für ein biologisches Multisegmentssystem ist ein flexibles System mit vielen gekoppelten Einzelementen. Ein Beispiel für einen Multisegmentarm ist der **Festo Rüssel**, ein bionischer Handling Assistent mit 13 pneumatischen Aktuatoren, 11 Freiheitsgraden, 1.8kg Gewicht und 0.5kg Traglast. Er enthält Elastizitäten und durch den Luftdruck zusätzliche Nicht-Linearitäten.

### 2.2.3 Laufmaschine (Invertebrate, Vertebrate)

Warum die Stabheuschrecke als Vorbild?

- 6-Beiner
- Langsames Insekt
- Einfacher kinematischer Aufbau und trotzdem sehr flexibel
- Weitreichende biologische Untersuchungen vorhanden

Abbildung 3 zeigt ein mechanisches Modell der Stabheuschrecke.

#### Sechsbeinige Laufmaschine – Optimierung

- Berechnung der Dynamik  

$$M \ddot{q} + h = J_R^T t + J_T^T f$$
- Optimierung nach folgenden Kriterien:
  - Minimales Biegepotential
  - Verspannungsfreiheit Beine
  - Minimale Gelenkzwangsmomente
  - Optimale Aufstandskraft-Richtungen
  - Minimale Antriebsleistung

⇒ Optimal sind 40% Biegepotential und 60% Verspannung. Das entspricht genau der Stabheuschrecke!

#### Vorteil eines langen Zentralkörpers

Je nach Beinanordnung verändert sich die Stabilitätsreserve (vgl. Abbildung 4).

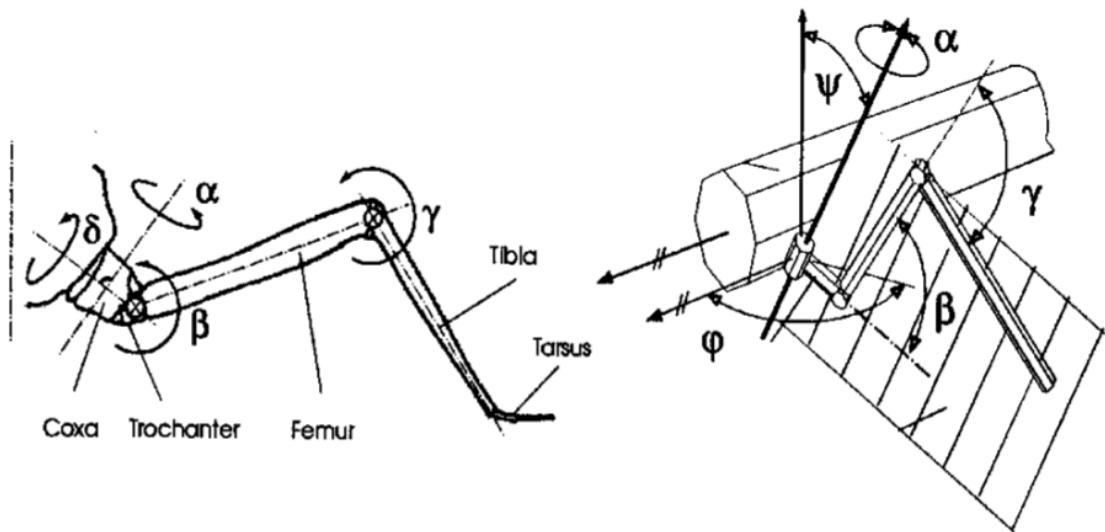
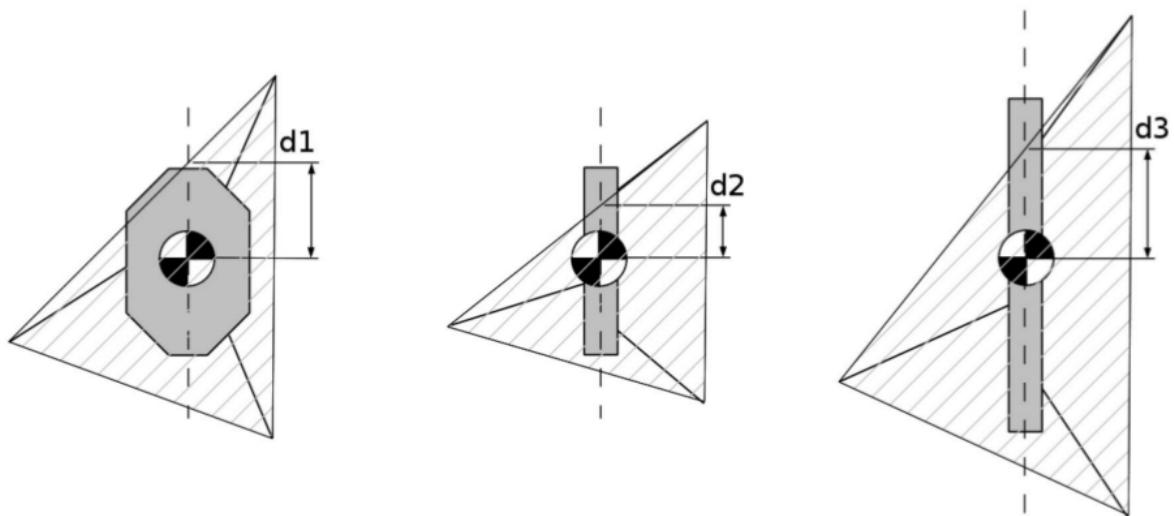


Abbildung 3: Stabheuschrecke – Mechanisches Modell



(a) Stabilitätsreserve bei lateraler Ausdehnung des Körpers

(b) Stabilitätsreserve bei kurzem Körper

(c) Stabilitätsreserve bei langgestrecktem Körper

Abbildung 4

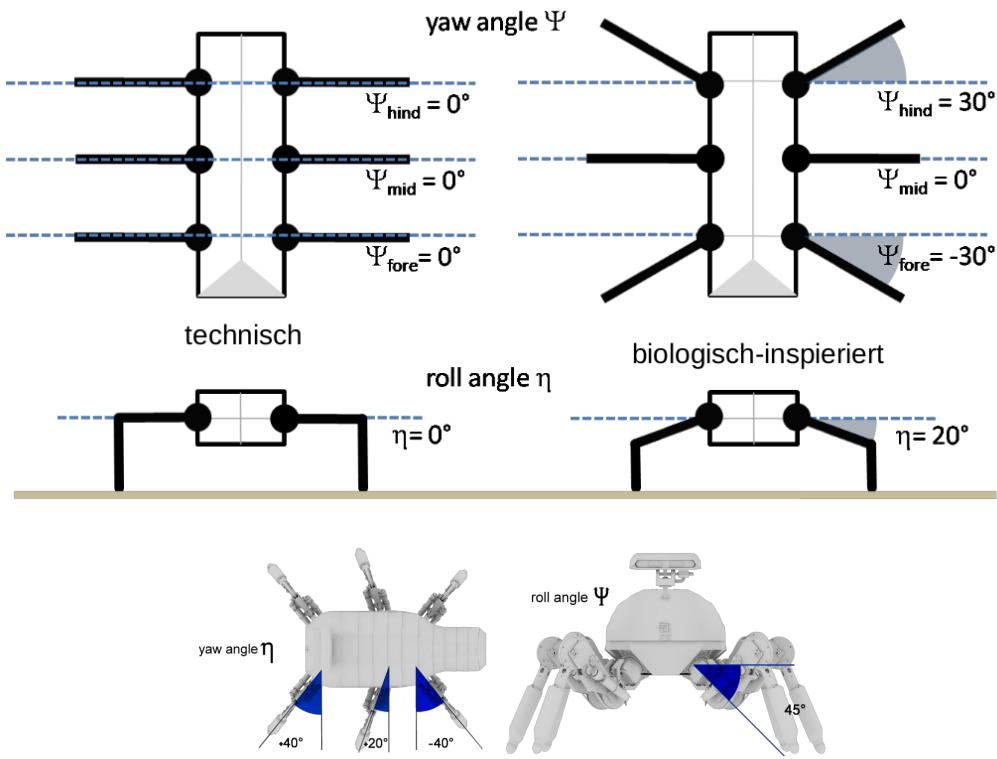


Abbildung 5: Vergleich der Befestigungswinkel im Beinansatzpunkt für technische und biologisch-inspirierte Roboter, hier LAURON

#### 2.2.4 Lokomotionsarten

- Zweibeiniges Laufen: Menschen, Vögel
- Vierbeiniges Laufen: Sehr viele unterschiedliche Morphologien/Kinematiken vierbeiniger Läufer
  - Reptilienartiges Laufen: Schildkröten, Eidechsen, Geckos (Tier trägt viel von seiner Gewichtskraft)
  - Amphibienartiges Laufen: Salamander, Frösche
  - Säugetierartiges Laufen:
    - \* Zehengänger: z.B. Hund, Katze; Stand ohne Energieverbrauch bei komplett arretierten Beinen
    - \* Spitzengänger: z.B. Pferde, Elefanten
    - \* Sohlengänger: z.B. Bären, Kängurus; niedrigerer Hebelarm aber immer noch angespanntes System im Stehen; führt zu hohem Energieverbrauch, aber dafür hoch agil/dynamisch; für Elefanten wäre z.B. Sohlengang nicht möglich
- Anordnung der Beine am Zentralkörper ist eine wichtige Designentscheidung
- Sechsbeiniges Laufen: Stabheuschrecke, Ameisen, Kakerlaken, Gottesanbeterinnen
- Achtbeiniges Laufen: Spinnen, Skorpione, Krebse
- Mehrbeiniges Laufen: Hundertfüßler, Tausendfüßler

Bei Vertebraten (Wirbeltieren) hat die Wirbelsäule eine Einfluss beim Auf- und Abfußen.

### 2.2.5 Arbeitsraumvergrößerung durch Anstellwinkel

1. Analyse des Arbeitsraums für einen Laufroboter
  - DLR-Crawler – Sechsbeiniger Laufroboter
  - Finger der DLR-Hand II als Beine
  - Kinematisch eingeschränkter Arbeitsraum
  - Achsensymmetrische Beinanordnung
  - **Arbeitsraumoptimierung durch Anstellwinkel der Beine**
2. Durch den Anstellwinkel  $\alpha = 30^\circ$  vergrößert sich der nutzbare Arbeitsbereich
  - +60% nutzbarer Arbeitsbereich
  - Größere Flexibilität (Schritthöhe, Schrittweite)
  - Ähnlichkeit zum Anstellwinkel der Stabheuschrecke
  - Komplexere mechanische Anbringung

## 2.3 Energie–Effizienz

*Billig bedeutet in der Biologie immer „mit geringem Energieaufwand erreichbar“*

### 2.3.1 Reduktion des Strömungswiderstandes

$$\text{Strömungswiderstandscoefficient } c_w = \frac{F_w}{q \cdot A} = \frac{\text{Widerstandskraft}}{\text{Staudruck der Anstroemung} \cdot \text{Referenzflächeninhalt}}$$

- Strömungswiderstand und Form am Beispiel Pinguin
  - er kann bis zu einer Geschwindigkeit von 7m/s schwimmen, das entspricht 100m/s in der Luft
  - er taucht täglich bis zu 100km weit und 400m tief, mit einer Magenfüllung Plankton pro Tag (ca. 1500 bis 2000km mit 1l Benzin)
  - ermöglicht durch einen  $c_w$ -Wert von 0.07 und strömungsgünstigen Flossen
- Strömungsoptimierte Fahrzeugkarosserie: Bionic Car von Mercedes Benz
  - 140PS, 4.3 Liter/100km,  $c_w$ -Wert 0.19
  - Pate für die Entwicklung war der Kofferfisch, der trotz seiner kantigen Form hervorragende Strömungseigenschaften hat
- Strömungswiderstand und Oberfläche
  - Haifischhaut und Schmetterlingschuppen bilden Streichlinien
  - Durch die Wasser- bzw. Luftverwirbelung entsteht ein sehr strömungsgünstiges Profil
  - Der Energieverbrauch von Verkehrsflugzeugen ließe sich damit um 4% reduzieren

### 2.3.2 Energieeffizienz beim Laufen

**Specific Resistance** beschreibt die Effizienz der Lokomotion:

$$\varepsilon = \frac{P_{lok}(t)}{G \cdot v(t)} \quad (1)$$

mit Leistung der Fortbewegung  $P_{lok}(t)$ , Laufgeschwindigkeit  $v(t)$  und Gewichtskraft  $G$ . Sie ist dimensionslos, unabhängig von der Art des Fahrzeugs oder des Roboters und gut vergleichbar. Die *Metabolic Cost of Transport* ist formelmäßig ähnlich, was einen Vergleich zur Natur ermöglicht.

In der Robotik gebräuchliche Form (integriert über die Zeit):  $\varepsilon = \frac{E}{G \cdot s}$ .

Effizienzsteigerung durch Gewichtsreduktion und/oder Geschwindigkeitssteigerung

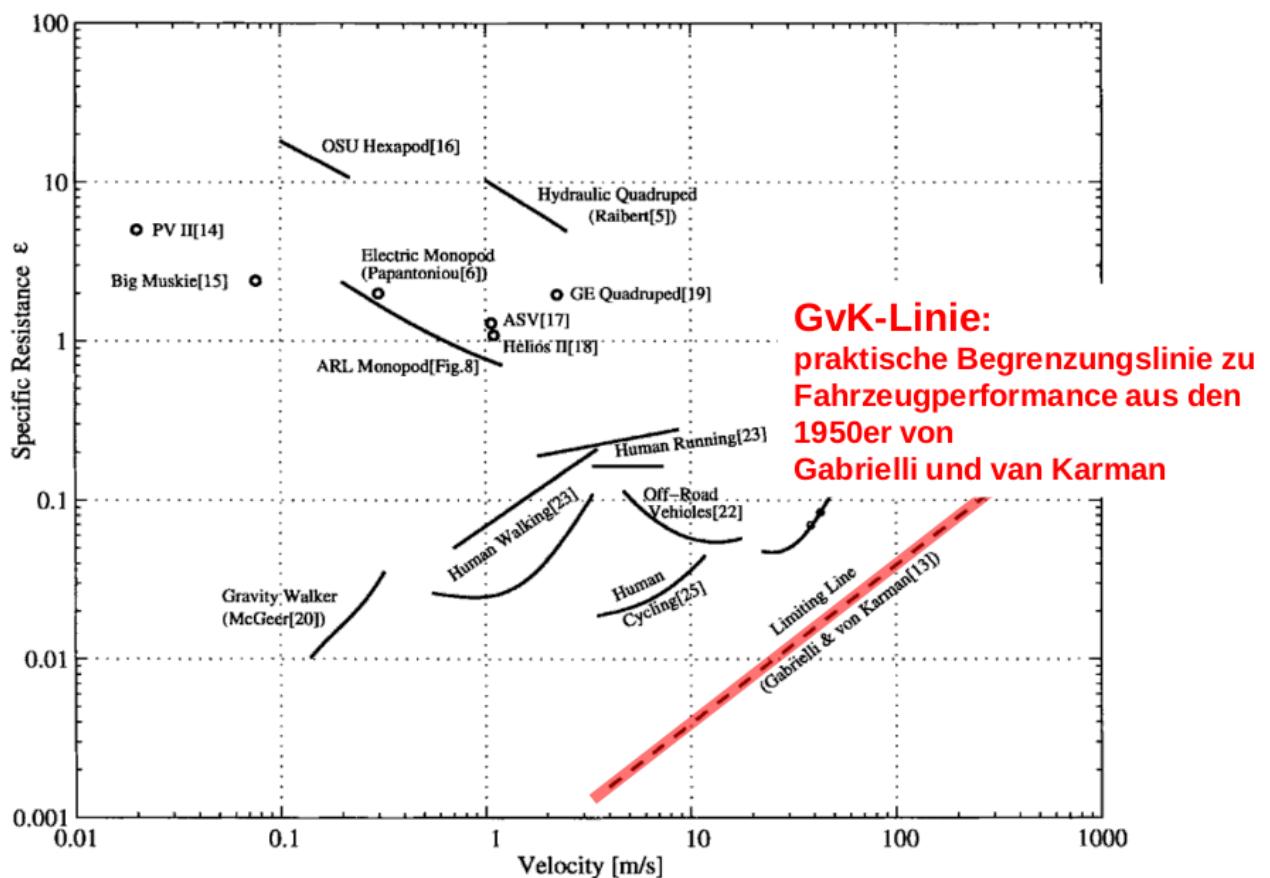


Abbildung 6: Specific Resistance: Für Hund, Vogel, etc. wurde der  $CO_2$ -Gehalt und somit indirekt der Energieverbrauch gemessen. Kein System konnte sich unterhalb der G&K-Linie bewegen.

### 2.3.3 Beispiele für Energiebilanzen in der Natur

- Ein **Gepard** läuft beim Erlegen seiner Beute anaerob. Nach dem schlagen ist er so erschöpft, dass er 50% seiner Beute an Aasfresser verliert.
- Das Fell von **Eisbären** absorbiert Sonnenlicht und hält damit einen hohen Anteil der Sonnenenergie als Wärme zurück.
- Ein biologische **Muskel** wächst nur dort wo er benötigt wird; damit wird sowohl der Energieverbrauch beim Wachsen als auch bei der Bewegung optimiert.

## 3 Muskeln

*Muskeln - Natürliche Antagonisten*

### 3.1 Der biologische Muskel – das Wirkprinzip

#### 3.1.1 Muskelaufbau und unterschiedliche Muskelformen

- Der Muskel ist nur ein Teil des Bewegungsapparates (Sehnen, Muskelfaszien, Sehnenscheiden, Gleitbeutel, Sesambeine und Sehnenknorpel).
- Muskeln können nur ziehen, deshalb Gegenspieler (antagonistisches Prinzip).
- Viele verschiedene Muskeln für spezielle Bereiche (der Mensch hat ca. 300 Muskeln).

**Muskelgewebe:**

- Herzmuskulatur (quergestreifte Muskulatur): das Herz ist ein Sonderfall
- Skelettmuskulatur (quergestreifte Muskulatur): für aktive Körperbewegungen; Trizeps, Bizeps
- Glatte Muskulatur: vor allem Eingeweidemuskulatur, z.B. Darm; keine willkürliche Kontrolle

**Muskelformen (Skelettmuskulatur):** siehe Abbildung 7

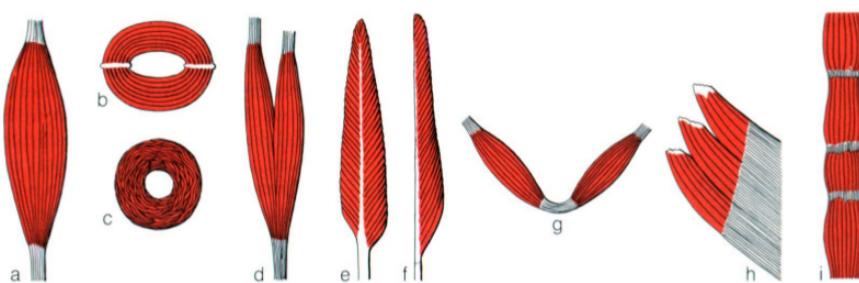


Abbildung 7: Formen der Skelettmuskulatur: a. Spindelförmig, b. Ringförmig, glatt, c. Ringförmig, gefiedert, d. Zweiköpfig, e. Doppelt gefiedert, f. Einfach gefiedert, g. Zweibauchig, h. Platter Muskel, i. Mehrbauchiger Muskel

**Muskelaufbau (Skelettmuskulatur):** siehe Abbildungen 8 und 9

- Muskelfasern verlaufen fast parallel in Längsrichtung
- Fiederung für Blutversorgung und Raum bei Dickenzunahme
- Volumen bleibt während der Kontraktion konstant
- Dickenzunahme durch Vergrößerung des Faserwinkels

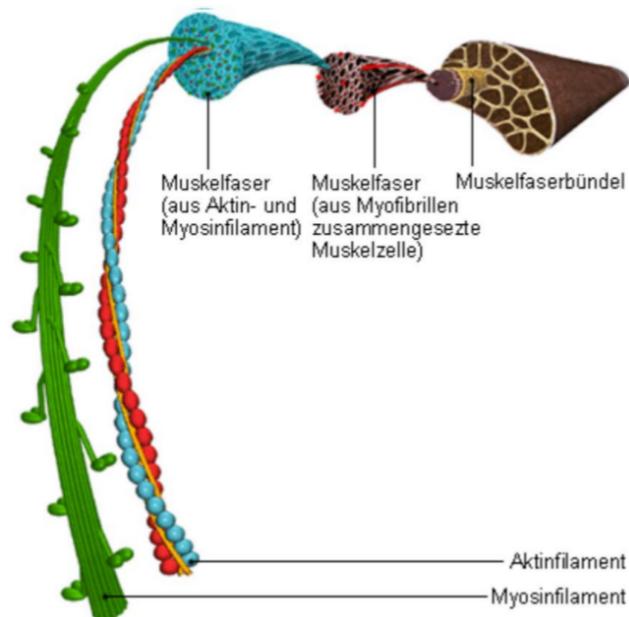


Abbildung 8: Muskelaufbau des Skelettmuskels

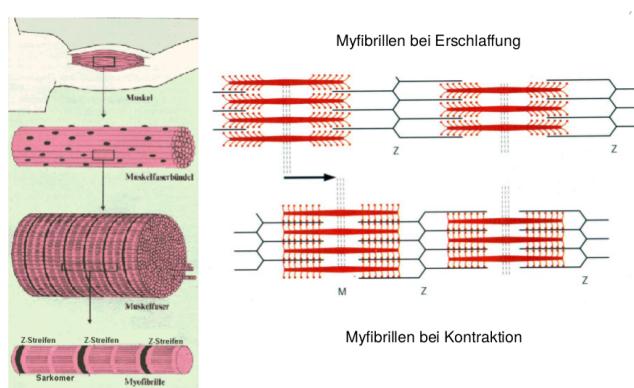


Abbildung 9: Muskelaufbau des Skelettmuskels

### 3.1.2 Prinzipielle Funktionsweise

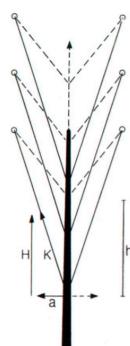


Abbildung 10: Funktionsweise des Skelettmuskels

Abbildung 10 stellt die Funktionsweise dar, hierbei gilt:

- Gedehnter Zustand (schwarz)
- Kontrahiert Zustand (gestrichelt)
- Hubhöhe  $h$ , Zugkraft einer Muskelfaser  $K$

- Zerlegung von  $K$  in Komponenten  $H$  und  $a$ : Je größer Fiederungswinkel, desto geringer die erreichte Kraft
- Kompensation in der Biologie durch:
  - Großer Fiederungswinkel
  - Mehr Muskelfasern sind beteiligt
- Hubhöhe immer größer als Faserverkürzung
- Faustregel: Je größer der Querschnitt, desto größer die Kraft

#### Geschwindigkeit - Last Beziehung (Hillsche Beziehung)



Abbildung 11: Hillsche Beziehung

- geringe Last → Verkürzungsgeschwindigkeit maximal
- mit zunehmender Last → nimmt Geschwindigkeit ab
- ab gewisser Last → Muskel nicht verkürzbar (isometrische Verhältnisse)

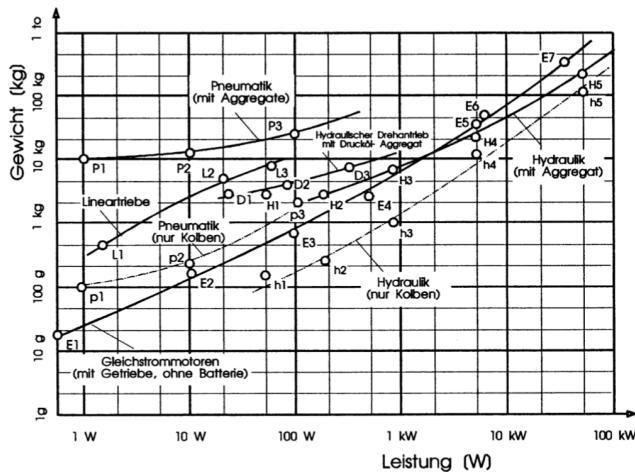
#### 3.1.3 Muskel als Effektor des Nervensystems

- Muskeln werden von einem oder mehreren Nerven versorgt, die **motorische und sensible** Fasern enthalten.
- Muskeln besitzen **Rezeptoren**, die über den jeweiligen Muskelzustand informieren:
  - Dehnungsrezeptoren (Muskelspindeln)
  - Spannungsrezeptoren (Sehnenorgane)
- Auch im Ruhezustand laufen ständig Nervenimpulse in den Muskel. Sie bewirken eine Grundspannung (**Tonus**) des Muskels.
- Verstärkte Nervenimpulse verursachen eine Verkürzung (isotonische Kontraktion) oder Spannungszunahme (isometrische Kontraktion).
- Bewegungen erfordern **synergistisches und antagonistisches** Zusammenwirken von Muskeln
- Synergisten: Muskeln mit gleicher Wirkrichtung
- Antagonisten: Muskeln mit gegensinniger Wirkrichtung
- Durch die Kombination beider Wirkprinzipien: Bremsende, geziigerte, harmonische und genau dosierte Bewegung möglich
- Einfacher Aufbau von einachsigen Gelenken, aber kompliziertes Zusammenwirken bei komplexeren Gelenken (z.B. Kugelgelenke)!

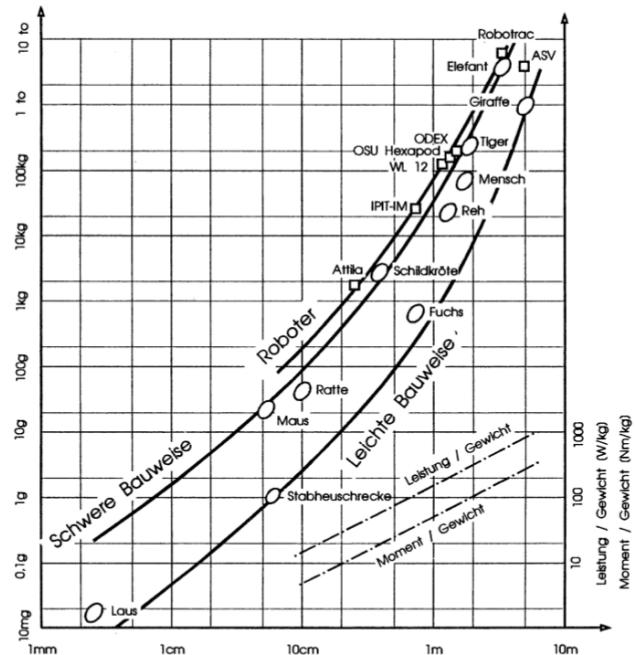
### 3.2 Klassische Antriebe in der Robotik

- **Pneumatische Antriebe:** Einsatz häufig bei Explosionsgefährdung oder Reinraumerfordernissen
- **Hydraulische Antriebe:** ab gewissem Gewicht besser, aber durch Linearachse Rotationsbewegungen schwierig
- **Elektrische Antriebe:** am weitesten verbreitet da günstig, hochpräzise Getriebeübersetzung

Leistungsgewichtsvergleich unterschiedlicher Antriebe in Natur und Technik:



(a) Technik



(b) Natur

Abbildung 12

### 3.3 Künstliche Muskelsysteme

*Von drehender Mechanik zu selbstverformenden Polymeren*

#### 3.3.1 Mechanischer Muskel

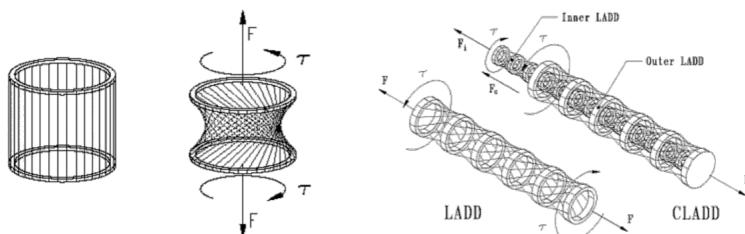


Abbildung 13: Typen mechanischer Muskeln

- Linear to Angle Displacement Device (LADD)
- Concentric LADD (CLADD)

#### 3.3.2 Chemische Muskeln (SMP, LAP)

- Light-Activated Polymers (LAP)
- Shape Memory Polymers (SMP)

#### 3.3.3 Elektroaktive Polymere (EAP)

- Electronic EAP: Ferroelectric, Dielectric, Electrostrictive
  - Ionic EAP: Ionic Polymer Gels, Ionometric Polymer-Metal Composites, Conductive Polymers
  - Anwendungen: Greifer, Lobster
- eher Grundlagenforschung, man kann keine großen Kräfte aufbringen, aber: kann sogar Schwimmbewegungen erzeugen; eher für sehr feine Bewegungen geeignet

#### 3.3.4 Piezoelektrische Antriebe

### 3.4 Fluidische Muskeln

*Fluidik ist die Wissenschaft, die sich mit flüssiger oder gasförmiger Materie befasst.*

#### 3.4.1 Spinnenprinzip

z.B. „Black Spider“: pneumatische mehrbeinige Laufmaschine

#### 3.4.2 Weitere Muskeln

z.B. Pleated Pneumatic Artificial Muscle (siehe Abbildung 14)

- Steifigkeit künstlich vorgegeben (Feder-Masse-System, Federkonstante zur Laufzeit veränderbar durch Aufblasen)
- Vorteile:
  - Elastizität, fließende Bewegungen, Steifigkeit einstellbar
  - Energie im System speicherbar

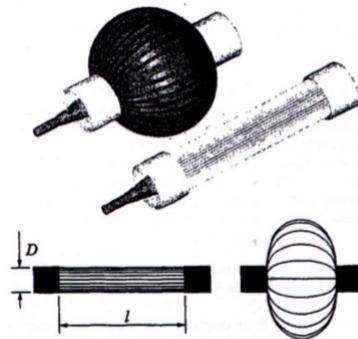


Abbildung 14

- schnelle Bewegung, passive Dämpfung gegenüber Stößen
- spannend für Laufbewegungen

- Nachteile:

- sehr komplex, pro Gelenk mindestens zwei Antriebe, alle müssen mit Zu- und Ableitungen/Ventilen versorgt werden
- Kompressor mit Druckluft nötig, schwierig unterzubringen

### 3.4.3 McKibben-Muskel

z.B. Shadow-Muskel (schnell konfektionierbar), FESTO-Muskel (Industrienorm)

## 3.5 McKibben Prinzip

### 3.5.1 Aufbau und Funktionsweise am Beispiel FESTO-MAS

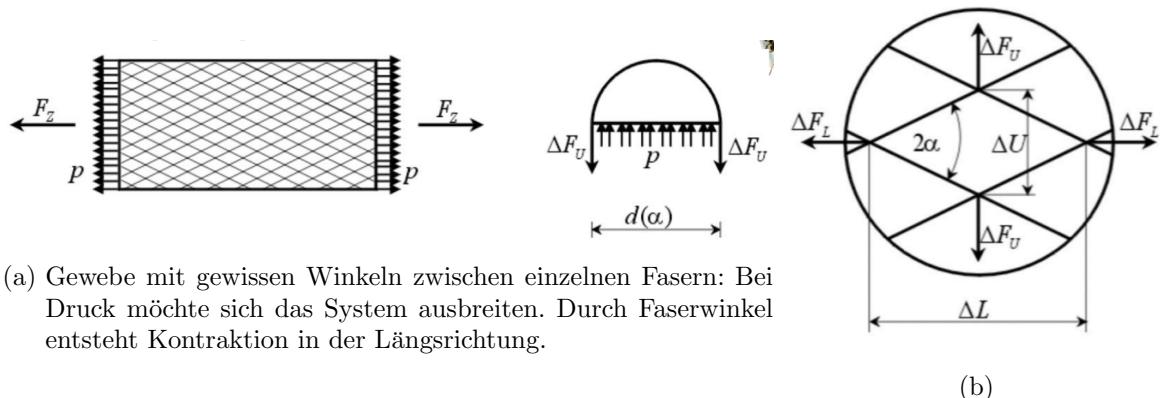


Abbildung 15: McKibben-Prinzip

Der Innendruck im Muskel bewirkt Längskontraktion, bzw. eine Zugkraft in axialer Richtung, da sich die Länge der einzelnen Gewebefasern nicht verändern kann. Dieses Prinzip funktioniert nur für einen Grundfaserwinkel von weniger als ca. 55°.

Abbildung 16 zeigt die Kennkurve für den FESTO Fluidic-Muscle unter verschiedenen Drücken. Auf der x-Achse ist die Längenänderung  $h$  in % abgetragen, die y-Achse zeigt die Kraft  $F$ . Es entstehen verschiedene Kurven durch unterschiedliche Kräfte je nach Längenänderung.

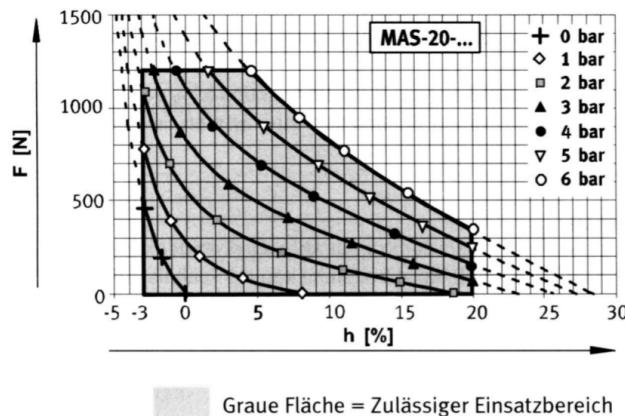


Abbildung 16: Zusammenhang zwischen Kontraktion, Druck und Kraft des Muskels

### 3.5.2 Verschiedene Muskelmodelle – Simplizität vs. Präzision

**Statisches Muskelmodell:** Das Modell basiert auf der Annahme, dass sich der Muskel wie ein idealer Zylinder verhält (vgl. Abbildung 17)!

Kraft hängt von Kontraktion und Druck ab:

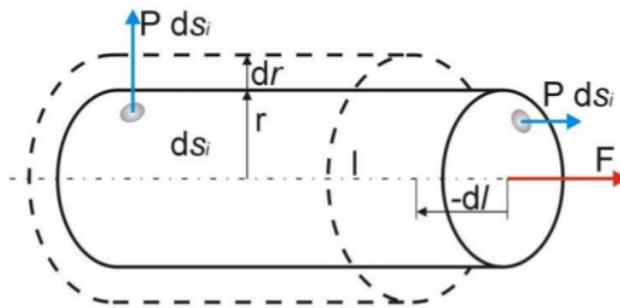


Abbildung 17

$$F(\kappa, P) = (\pi \cdot r_0^2) \cdot P \cdot [a \cdot (1 - \kappa)^2 - b]$$

mit

$$\kappa = \frac{(l_0 - l)}{l_0}, 0 < \kappa \leq \kappa_{max}$$

$$a = \frac{3}{\tan^2(\alpha_0)}, b = \frac{1}{\sin^2(\alpha_0)}$$

mit  $\alpha_0$  = Ausgangsfaserwinkel,  $\kappa$  = Kontraktion,  $P$  = Druck

$$F_{max} = (\pi \cdot r_0^2) \cdot P \cdot [a - b] \text{ mit } \kappa = 0$$

$$\kappa_{max} = 1 - \sqrt{\frac{b}{a}} \text{ mit } F = 0$$

Der Faktor  $\kappa$  ist die relative Längenänderung, für 0 Kontraktion wird Term maximal, für Kraft = 0 maximale Kontraktion in Abhängigkeit vom Faserwinkel.

Erweiterung des Modells durch Einführung eines Parameters  $\varepsilon$ :

- Berücksichtigung der Einschnürung an den Enden des Muskels

- Modifizierung der Kurve für den Fall eines geringen Drucks
- Bestimmung von  $c_1$  und  $c_2$  durch Least-Square-Schätzer: Vergleich des Modells mit dem realen System

$$F(\kappa, P) = (\pi \cdot r_0^2) \cdot P \cdot [a \cdot (1 - \varepsilon \cdot \kappa)^2 - b] \text{ mit } \varepsilon = c_1 \cdot e^{-P} + c_2$$

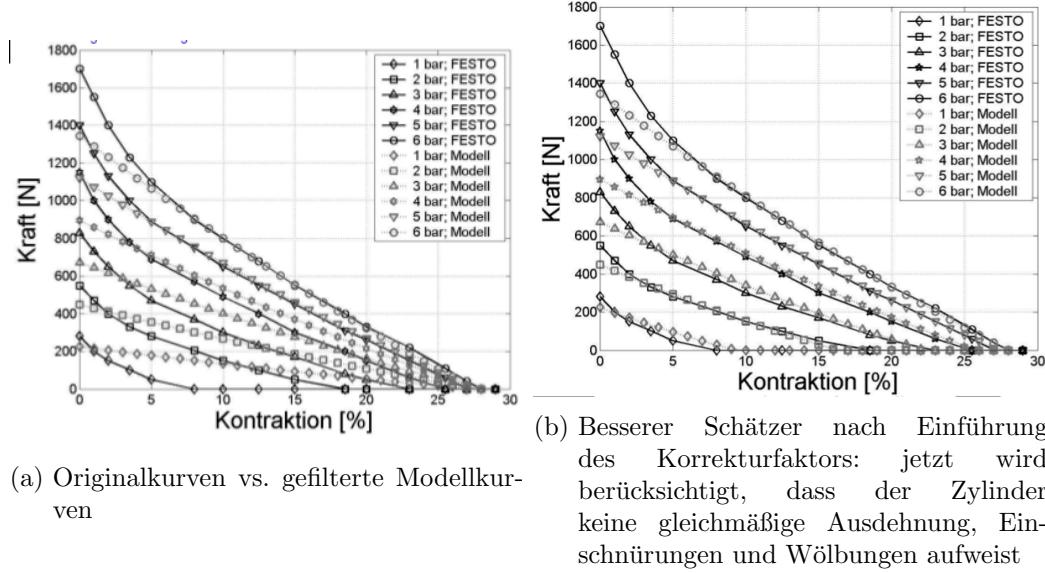


Abbildung 18: Statisches Muskel-Modell

**Dynamisches Muskel-Modell:** Kraftgleichung:

$$F_{Mus} = F_{Feder}(P, \kappa) + F_{Daempfer}(P, \dot{\kappa})$$

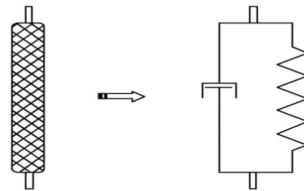


Abbildung 19: Mechanisches Ersatzschaltbild

### Quick-Release Test Umgebung

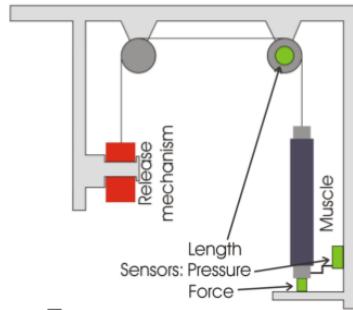


Abbildung 20: Test-Umgebung

- Erfassung der dynamischen Eigenschaften des Muskels
- Aufzeichnung aller relevanten Muskelkenngrößen
- Vergleich des Verhaltens mit biologischem Muskel: Validierung des Muskelmodells z.B. durch Vergleich zwischen aufgezeichneten Daten und Matlab-Simulation
- $P_{mech} = F \cdot v$
- Validierung des Muskelmodells durch realen mechanischen Aufbau: Bis auf die Anfangsbereiche sehr gute Annäherung an Simulation, daher nutzbar für die Regelung

### Kraft-Geschwindigkeits-Zusammenhang

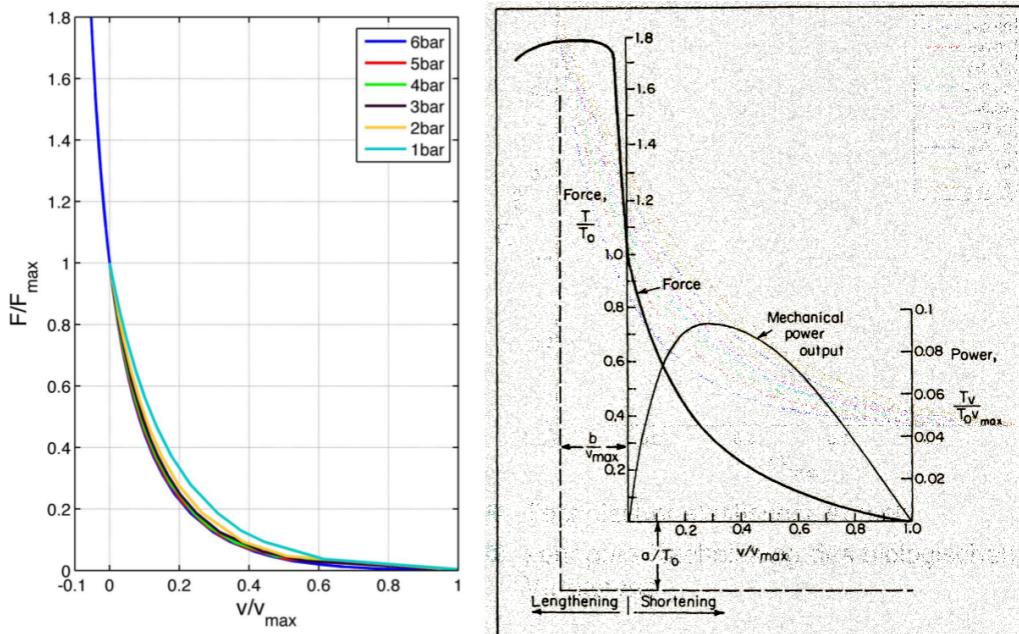


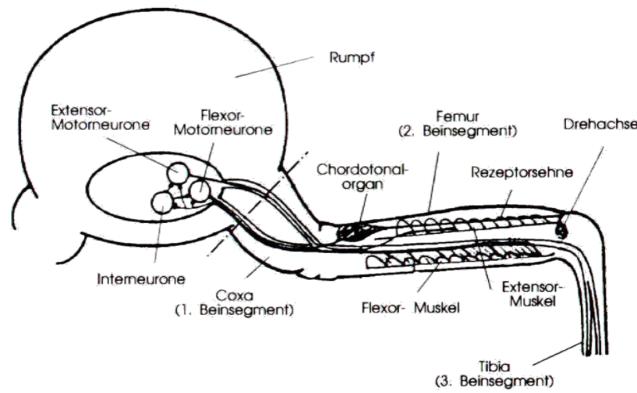
Abbildung 21: Kraft-Geschwindigkeits-Zusammenhang: vgl. Hiller'sches Prinzip, genau die gleiche Kurve in technischem System implementiert, biologisches Verhalten schön nachgebildet

## 3.6 Antagonistische Regelung

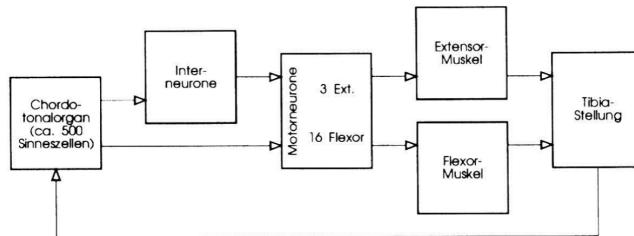
### 3.6.1 Antagonistisches Gelenk

Gelenkregelung Stabheuschrecke:

### 3 Muskeln



(a)



(b) Interneurone zur Verstärkung zur Regelung der Tibia-Stellung

Abbildung 22: Statisches Muskel-Modell

**Kräfteverhältnis in einem Gelenk** (vgl. Abbildung 23):

$(F_A - F_B) \cdot r = M_S$  mit

- $F_{A(B)}$ : Muskelkraft in Muskel A(B)
- $r$ : Radius der Gelenktrommel
- $M_S$  Störmoment, bzw. äußeres Moment

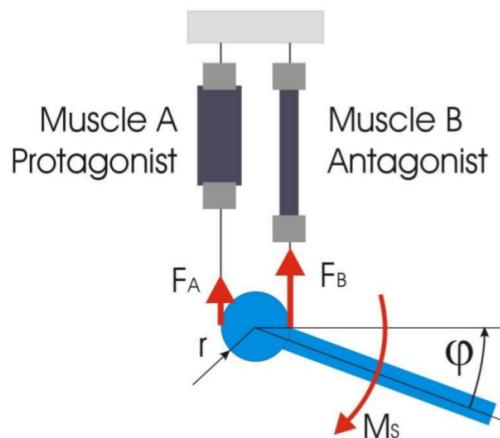


Abbildung 23: Technische Umsetzung als klassisches Rotationsgelenk über Seil/Draht, zwei Muskeln zusammen schalten, in Abhängigkeit von Hebelarm (Durchmesser der Scheibe) und  $F_A, F_B$ , Regelung der Steifigkeit durch Zug auf Seil

**Modellbildung – Aufbau eines Gelenks** (vgl. Abbildung 24): gut mathematisch modellierbar  
Gelenk-Bewegungsgleichung:  $J\ddot{\varphi} = -rF_A(\kappa_A, \dot{\kappa}_A, p_A) + rF_B(\kappa_B, \dot{\kappa}_B, p_B) - \frac{1}{2}\cos(\varphi - \varphi_0)F_g + lF_s$

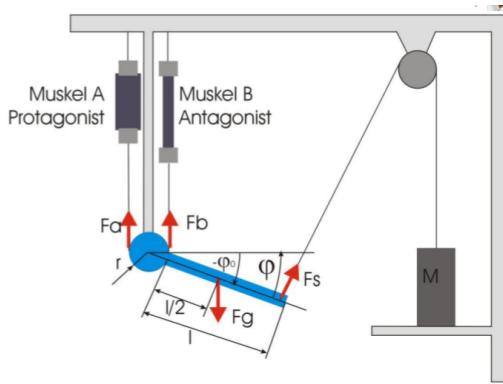


Abbildung 24

**Modellbildung – Festlegung der Zustandsgrößen** (vgl. Abbildung 25):

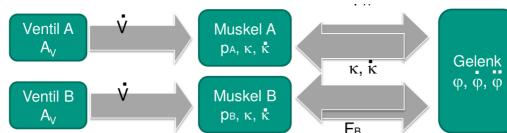


Abbildung 25

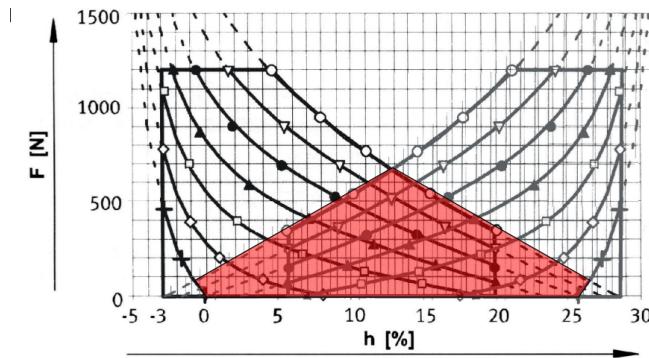
- Zustandsgrößen  $\underline{x}$ 
  - Druck Muskel A
  - Druck Muskel B
  - Gelenkwinkel  $\varphi$
  - Winkelgeschwindigkeit  $\dot{\varphi}$
  - Winkelbeschleunigung  $\ddot{\varphi}$
- Eingangsgrößen  $\underline{u}$ 
  - Öffnungsfläche  $A_V$  von Ventil A
  - Öffnungsfläche  $A_V$  von Ventil B
- nichtlineare DGL (kann zur Regelung benutzt werden):
 
$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t)$$

$$\underline{y} = \underline{g}(\underline{x}, \underline{u}, t)$$

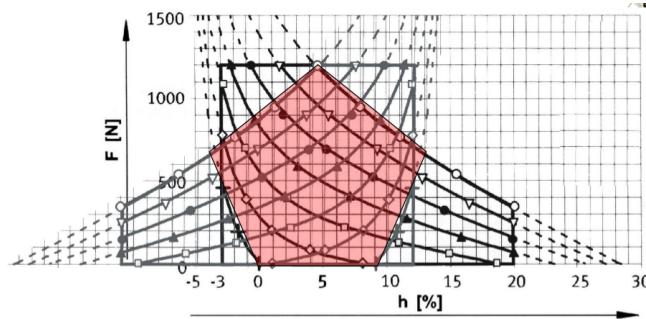
### 3.6.2 Zusammenspiel zweier Fluidischer Muskeln

**Pneumatischer Muskel – Robotergelenke** (vgl. Abbildung 27):

- Einstellbare Steifigkeit
- Flexibles Gelenk durch passive Nachgiebigkeit
- Schonung und Reduktion der Mechanik
- Inhärente Gelenkanschläge
- Gelenkeigenschaften über Motorgrößen einstellbar (mit Elektromotoren undenkbar)



- (a) In neutraler Gelenkposition stark kontrahiert; je nach gewünschter Längenänderung Veränderung der maximalen möglichen Kraft; Schnittstelle: Neutrale Gelenkposition, bereits starke Kontraktion (etwa 12-13%)



- (b) In neutraler Gelenkposition wenig kontrahiert; dichtere Anordnung; höhere Kräfte möglich aber kleinerer Gelenkwinkelbereich, daher viele Möglichkeiten, System einzustellen

Abbildung 26

### 3.6.3 Regelungsbeispiel für ein flexibles Gelenk

Zu Abbildung 28c:

- Regelung eines künstlichen Muskels nach McKibben-Prinzip: hochkompliziert, trotz vieler Vorteile hochgradig nicht-linear auch wenn Grundtechniken nicht so schwer
- Vorgabe der Sollposition und der Gelenksteifigkeit
- Berücksichtigung des am Gelenk angreifenden Störmomentes
- Berücksichtigung der maximal möglichen Kraft in allen möglichen Gelenkstellungen

## 3.7 Anwendungsbeispiele

*There is no business like show business*

### Motivation – Muskel als Antrieb von Laufmaschinen:

- schnelle und präzise langsame Bewegungsabläufe möglich
- zusammen mit Sehnen sowohl Aktuator als auch Sensor
- Nutzung der passiven Dämpfung als Energiespeicher
- Anpassung an unebenes Gelände gut möglich (passive Nachgiebigkeit)

### 3 Muskeln

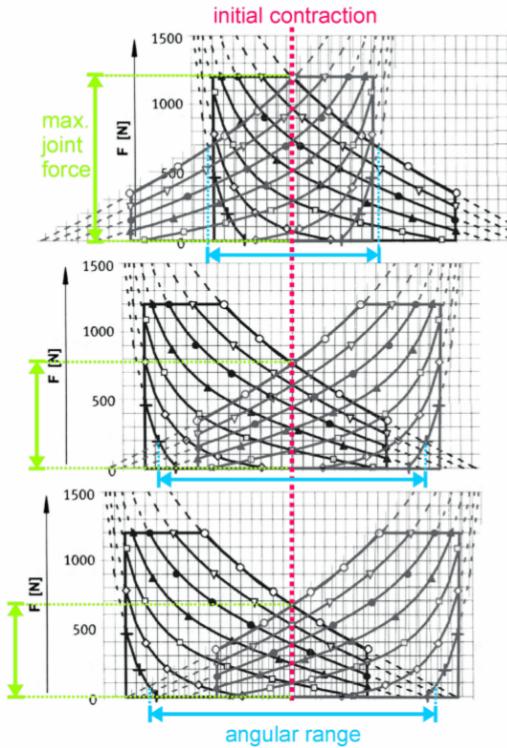
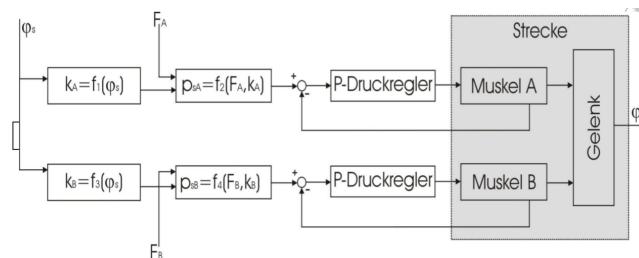
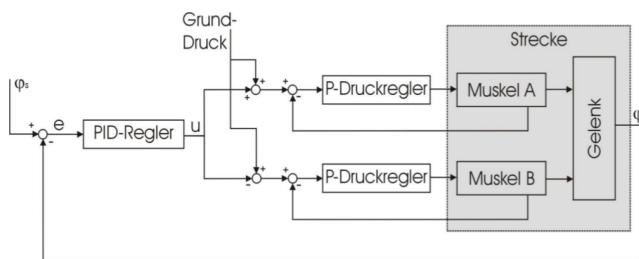


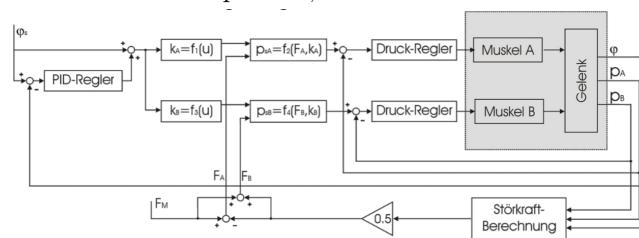
Abbildung 27



(a) Steuerung eines Gelenkes: Nur möglich wenn keine Störungen → Regelung



(b) Einfache PID-P-Regler: soll Regelfaktor des Sollgelenkwinkels anpasssen; funktioniert nicht wirklich



(c) Regelung mit Muskelmodell

Abbildung 28

- nur Zugkräfte, daher immer als antagonistisches Paar eingesetzt
- Eigenschaften, die auch für Laufmaschinenantriebe wünschenswert sind

**AirBug:** sechsbeiniger, durch fluidische Muskeln angetriebener Roboter.

Mechanische Parameter:

- Gewicht 32 kg (14kg Zentralkörper, 6kg Ventile, 2 kg pro Bein)
- Zentralkörper (mm):  $850 \times 600 \times 250$
- Aufstandsfläche (mm):  $1800$  (width)  $\times 1500$  (length)
- Bein-Länge (mm):  $100 \times 472 \times 560$
- Muskel-Längen (mm):  $\alpha : 160, \beta : 240, \gamma : 260$
- Winkelbereich:  $\alpha : 55^\circ; \beta : 90^\circ; \gamma : 75^\circ$  (sehr ähnlich wie bei der Stabheuschrecke)
- Max. Drehmoment:  $\alpha : 26.4Nm, \beta : 48.6Nm, \gamma : 31.3Nm$
- Winkelgeschwindigkeit:  $60^\circ/s$
- Max. Geschwindigkeit:  $10cm/s$

**PANTER:** Pneumatic Actuated dyNamical sTable quadrupEd Robot

Adaptives Elastisches Laufen mit künstlichen Muskeln durch den Entwurf eines Algorithmus zur elastischen Beinlängen-Schwingungsregelung:

- Adaptives Einstellen der Eigenfrequenz über Systemgrößen
  - Kontrolliertes Anregen mit Resonanzfrequenz
- Optimale energieeffiziente Bewegung
- Unterschiedliche Lauffrequenzen durch Modifikation von Systemgrößen

### FESTO-Muskel

#### 3.7.1 Vergleich biologischer vs. fluidischer Muskel

	biologisch	fluidisch
Kontraktion	50 – 70%	max. 35%
Kraft/cm <sup>2</sup>	20 – 60N	bis 700N
Leistung/kg	40 – 250W/kg	0,5 – 2kW/kg
Wirkungsgrad	20 – 30%	22 – 50%
Kontraktionsgeschwindigkeit	bis 8m/s	bis 2m/s
Bandweite	20-100Hz (Zuckungen)	5Hz
Regelung	gut bis sehr gut	befriedigend bis gut
Betrieb in Wasser	ja	ja
Temperaturbereich	0 – 40°C	-30 bis +80°C
Widerstandsfähigkeit	sehr gut	befriedigend bis gut
Regeneration	ja	nein
Energiequelle	chemisch	pneumatisch
Umweltschmutz	produziert CO <sub>2</sub>	je nach Herstellung der Druckluft
Skalierbarkeit	$\mu m - m$	mm – m
Linearbetrieb	ja	ja

Tabelle 1: Biologischer vs. fluidischer Muskel

## 4 Sensoren

*Vieles, was Ingenieure bei technischen Sensoren noch für utopisch halten, hat die Natur mit größter Raffinesse schon realisiert*

### 4.1 Sinneseindrücke und Sensoren – Übersicht und Klassifikation

#### 4.1.1 Klassifikation von Sinneseindrücken

Sinne & Rezeptortyp	Adäquater Reiz	Sinnesorgane beim Menschen/ ⇒ (Bsp. aus dem Tierreich)
<b>Mechanische Sinne</b> Mechanorezeptoren		
Tastsinn	Zug, Druck	Tastkörperchen
Strömungssinn	Strömungen in Wasser & Gasen	⇒ (z.B. Seitenlinienorgan bei Amphibien und Fischen)
Drehsinn	Winkelbeschleunigung	Bogengänge im Innenohr
Lagesinn	Erdbeschleunigung = Schwerkraft	3 Bogengänge im Innenohr ⇒ (z.B. Borstenfeld bei Ameise)
Hörsinn	Schallwellen (= Druckwellen) 20 – 20.000 Hz beim Menschen Ultraschall (über 20 kHz)	Schnecke im Innenohr ⇒ (z.B. Hörorgan bei Fledermaus)
<b>Chemische Sinne</b> Chemorezeptoren		
Geruchssinn	Chemische Stoffe in Gasphase	Riechschleimhaut der Nase
Geschmackssinn	Gelöster Stoff	Geschmacksknospen der Zunge
<b>Elektrischer Sinn</b> Elektrorezeptoren	Elektrisches Feld	⇒ (bei elektrischen Fischen)
<b>Magnetischer Sinn</b> Magnetorezeptoren	Magnetfeld	⇒ (bei Zugvögeln, Bienen)
<b>Lichtsinn</b> Photorezeptoren	Elektromagnetische Wellen „sichtbares Licht“ beim Menschen 380-780 nm, UV (unter 380 nm) Polarisiertes Licht	Augen ⇒ (z.B. Facettenauge der Insekten)

Tabelle 2: Übersicht über Sinne, Reize und Sinnesorgane

#### 4.1.2 Rezeptoren und Reizverarbeitung

Siehe Abbildungen 29 bis 31.

#### 4.1.3 Sensorik in der Technik

Der Begriff des Sensors (lateinisch „sensus“ der Sinn) leitet sich ab von den menschlichen Sinnen.

- Auge ⇒ Optik, Kameras
- Nase ⇒ Gassensor
- Tastsinn ⇒ Druck, Temperatur
- Geschmack ⇒ chemische Sensoren

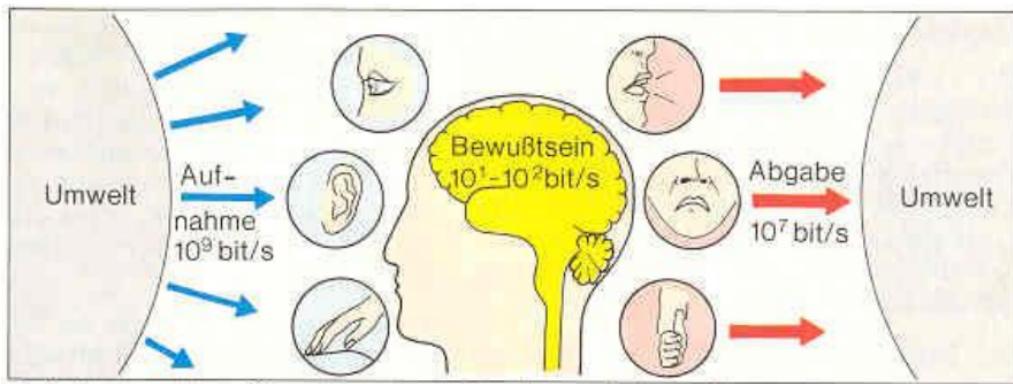


Abbildung 29: Aufnahme, Bewusstmachung und Abgabe von Information

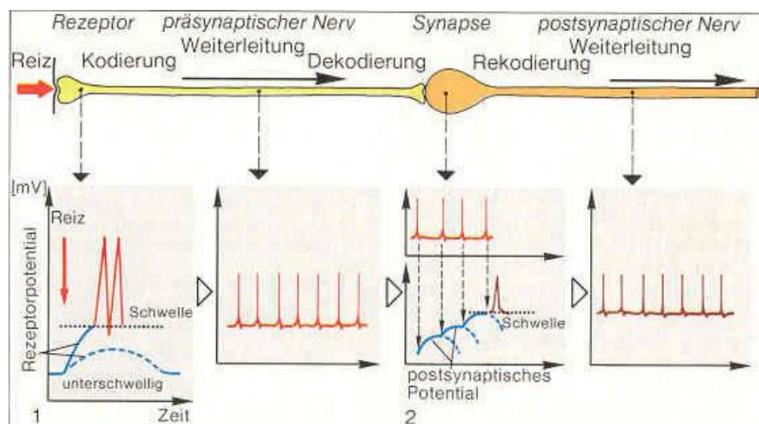


Abbildung 30: Reizverarbeitung

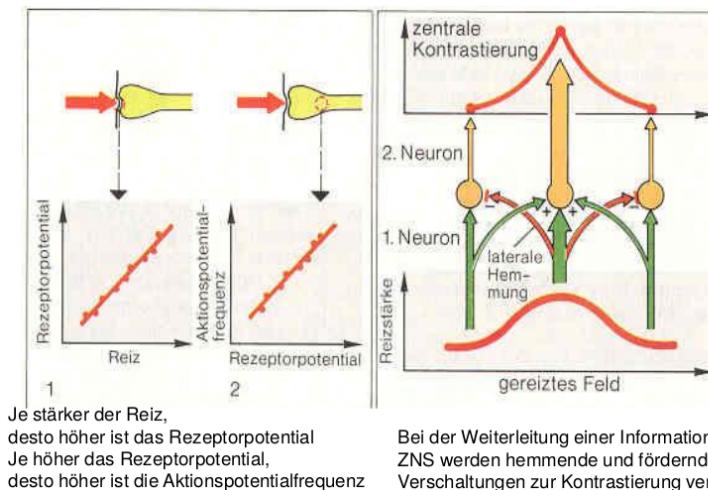


Abbildung 31: Beziehung/Kontrastierung zwischen Reiz, Rezeptor und Aktionspotential

IEC-Definition für technische Sensoren: „Ein Sensor ist das primäre Element in einer Messkette, das eine variable Eingangsgröße in ein geeignetes Messsignal umsetzt.“

### Sensoren in der Robotik:

Siehe Abbildungen 32 bis 34.

### Sensoren und Perzeption:

Siehe Abbildung 35.

## 4 Sensoren

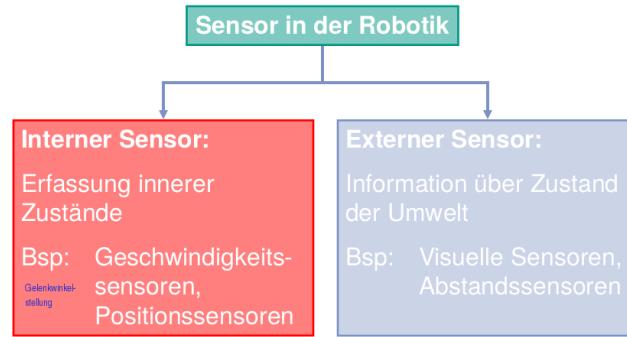


Abbildung 32: Klassifikation von Sensoren nach ihrer Funktionsweise

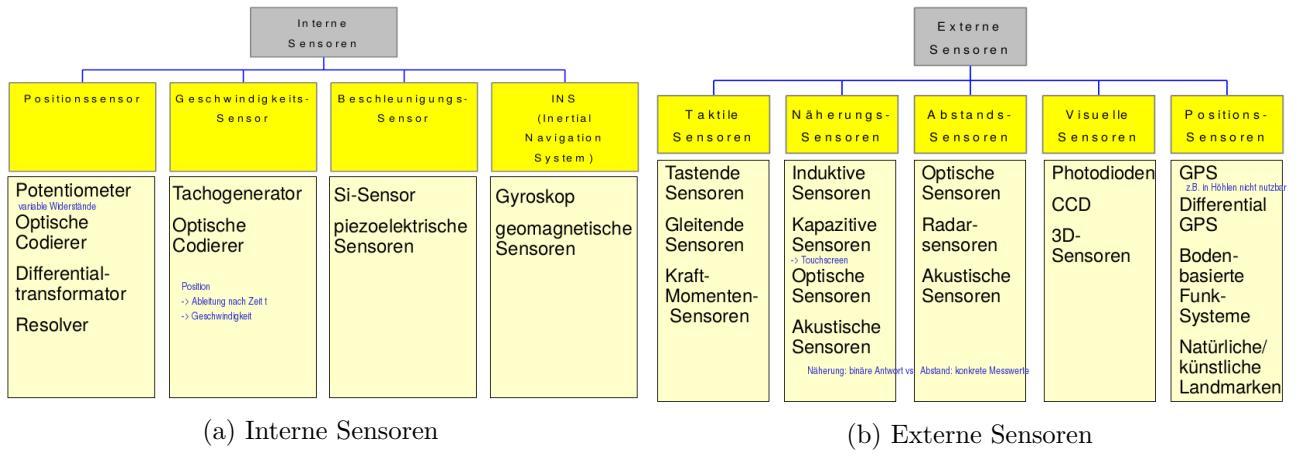


Abbildung 33

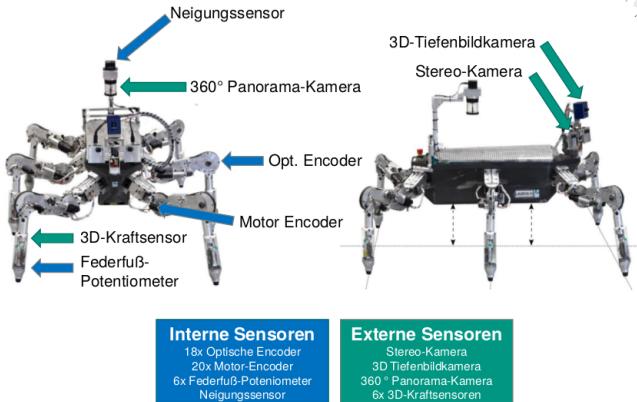


Abbildung 34: Beispiel Interne vs. Externe Sensoren

### Anforderungen an technische Sensorik:

- Genauigkeit
- Präzision
- Betriebsbereich
- Antwortgeschwindigkeit
- Kalibrierung
- Zuverlässigkeit
- Kosten
- einfache Installation

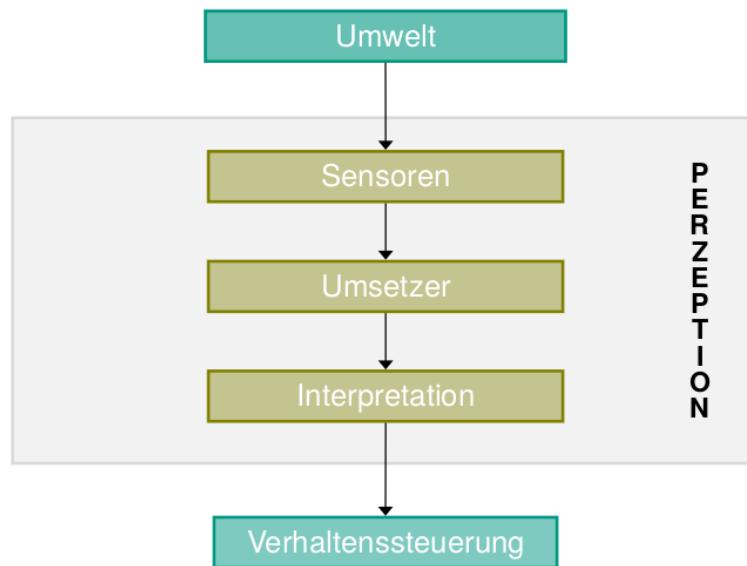


Abbildung 35

- Linearität
- Auflösung
- Reproduzierbarkeit
- Querempfindlichkeit
- Alterung
- Ansprechverhalten

⇒ Ausgehend von der Aufgabenstellung und des Ortes der Integration wird der Sensor ausgewählt; jedoch können meistens nur 2-3 Eigenschaften auf einmal erfüllt werden da viele sich gegenseitig ausschließen; man muss auf den angestrebten Anwendungsfall optimieren

#### Bionik in der Sensorik:

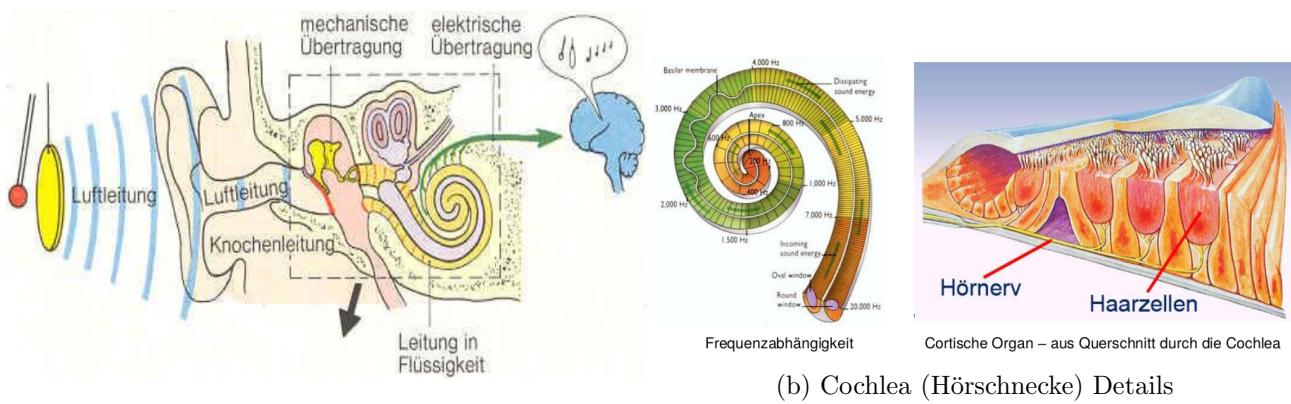
- Biologische Sensoren sind...
  - ... hoch spezialisiert auf ihren Anwendungsbereich,
  - ... hoch miniaturisiert & parallel,
  - ... in großen „Stückzahlen“ in Sensorfeldern angeordnet.
- Schwerpunkte für eine technische Umsetzung:
  - Sensorprinzip
  - Datenauswertung
- Kopieren der Natur an dieser Stelle (noch) unmöglich! (Parallelität, Datenrate wäre zu hoch für Verarbeitung)

## 4.2 Sensorsysteme in Biologie und Technik – Eine vergleichende Einführung

### 4.2.1 Hören – Von der Fledermaus zum Ultraschallsensor

**Menschliches Gehör:** hat sehr guten Frequenzbereich, aber spezialisiert:

- Ohrmuschel: unterstützt räumliche Ortung
- Mittelohr: Impedanzwandlung
- Innenohr: Umwandlung von Schallsignalen in Neuronenimpulse; med. Konstruktion von Knöchelchen



(a) Schallaufnahme und -weiterleitung

(b) Cochlea (Hörschnecke) Details

Abbildung 36

### Ultraschall – Die Fledermaus als Vorbild:

- Hoch entwickeltes Echoorientierungssystem
- Kehlkopf erzeugt Ultraschalltöne im Bereich von 20 - 200 kHz
- Echos werden von den Ohren empfangen und ausgewertet
- Fähigkeit sich bei absoluter Dunkelheit zu orientieren:
  - Distanz, Richtung, Form
  - Größe, Struktur und
  - Eigenbewegung der reflektierenden Ziele
- Zum Teil auch Klassifikation der Beute

### Sonar:

- Echoorientierung durch Serien kurzer Klicklauten (bis zu 130 kHz)
- können verschiedene Fische, Pflanzen und Gegenstände unterscheiden (auch Minen unter dem Sand)
- Echoorientierung nicht angeboren

### Ultraschall – Technische Umsetzung

- Einsatz:
  - Abstandsmessung
  - Hinderniserkennung (Nahbereich)
  - Kartographieren der Umgebung (Fernbereich)
- Eigenschaften:
  - akustische Wellen im Frequenzbereich oberhalb 20 kHz
  - Ausbreitung ist nur in Materie möglich
  - beim Auftreffen einer Schallwelle auf eine Grenzfläche zwischen zwei Medien teilt sich die Welle in eine reflektierten und transmittierten Anteil auf
  - Ausbreitungsgeschwindigkeit von Schallwellen in Luft (20°C): 343,8 m/s
  - Distanzbestimmung über Laufzeitmessung eines Schallsignals
- Nachteile:
  - Schallgeschwindigkeit ist abhängig von Temperatur, Luftfeuchtigkeit und Druck
  - Reflexion und Streuung an verschiedenen Oberflächen kann Messergebnisse verfälschen
  - es existiert eine kürzeste und längste messbare Entfernung

#### 4.2.2 Gelenkwinkel – Wie ist die Extremität orientiert?

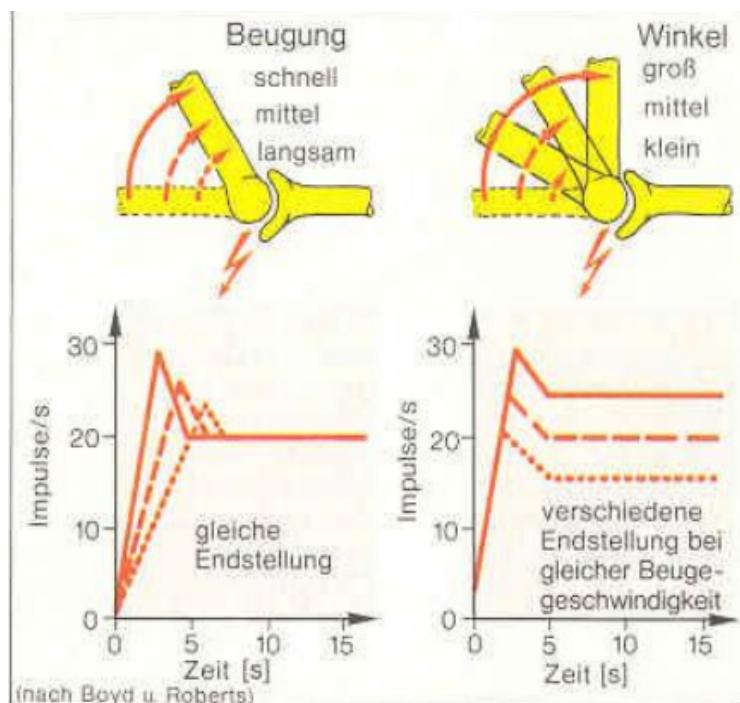


Abbildung 37: Geschwindigkeit/Winkel der Gelenkbeugung beim Menschen

#### Gelenkwinkelsensoren

1. Messprinzipien (Ohmisch, Induktiv, Magnetisch, Photoelektrisch )
2. Interpolation
3. Absolute Positionsmessung (Anschlag, Indexkanal, Potentiometer, Graycodierung, Quasiabsolutcode)

#### Interpolation von Zwischenwerten:

- Erhöhung der Auflösung
- Aus analogem Signal (egal bei welchem Sensor) können Zwischenstufen interpoliert werden

#### Absolute Positionsmessung

- Relative optische Encoder:
  - Nur relative Ticks
  - Absoluten Winkel nur durch Anschlag, Indexkanal
  - Pro: einfacher & robuster Sensor, kostengünstig
  - Contra: Komplexere Reset-Routine, nur einmalige Referenzierung
- Absolute optische Encoder:
  - Absolute Position in jedem Schritt
  - Spezielle Codierung: z.B. Graycode
  - Pro: Kein extra Reset notwendig, kein Encoder-Drift möglich
  - Contra: Viele optische Spuren notwendig, komplexere Sensorauswertung, teurer & empfindlicher
- Gelenkwinkelerfassung mit Quasiabsolutwertgeber:

- Problematik: Absolute Gelenkposition notwendig; Bewegung zu einem Referenzpunkt nicht möglich; Gray codierte Sensoren schwer integrierbar und teuer
- Vorteil: Nur 2 optische Spuren; Inkrementalcode + einfacher Absolutcode; Leicht integrierbar (Größe skalierbar, Runder / Längscode, Reflex / Gabellichtschranke)
- Nachteil: Bewegen des Gelenks um Absolutcode auszulesen; Software zur Positionsberichtigung
- Positionscode durch zwei Spuren abschnittsweise eindeutig

#### 4.2.3 Gravitation bzw. Beschleunigung – Innenohr gegen Inertialsystem

Das Gleichgewichts-(Vestibular-)organ des Menschen enthält Bogengänge (siehe Abbildung 38): drei Kreisel im Ohr, welche jeweils um  $90^\circ$  gegeneinander verschoben sind. So ist Neigung in allen drei Raumrichtungen messbar. Ein weiteres Beispiel aus der Natur sind Borstenfelder zur Messung des Schwersinns bei der Ameise. Die Borstenverformung ist messbar, so können Insekten Gravitation kompensieren. **Inertialsysteme (Kreiselsysteme)**

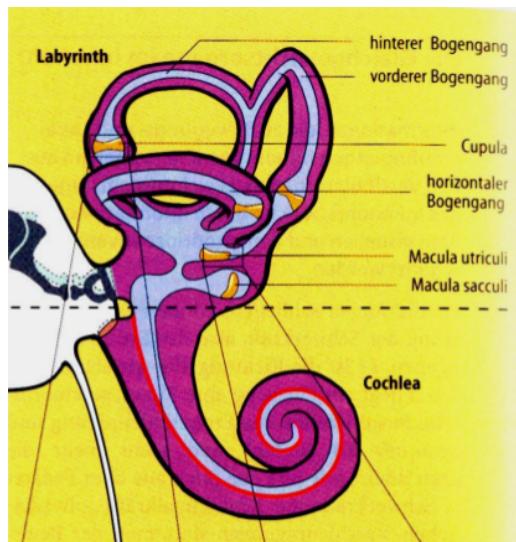


Abbildung 38: Gleichgewichts-(Vestibular-)organ des Menschen

- Realisierbar mit:
  - Mechanischen Kreiseln
  - 3 Beschleunigungssensoren, 3 Piezo- oder Laser-Kreisel
- Messwerte:
  - Beschleunigung in 3 Raumachsen
  - Winkelgeschwindigkeit um 3 Raumachsen
- Messergebnis:
  - Bestimmung der Gravitationsrichtung
  - Bestimmung der relativen Beschleunigungen und Drehungen relativ zum Raumkoordinatensystem
  - Trajektorienberechnung
  - Genauigkeit hängt vom verwendeten Kreiselsystem ab (Laserkreisel: 1cm Abweichung pro Stunde)

Abbildung 39 zeigt ein Inertialsystem. Es enthält drei Rotationssensoren, drei Beschleunigungssensoren sowie Magnetkompass. Die Messung (und somit auch ihr Fehler) wird hierbei zweimal aufintegriert, was bei der Positions berechnung nicht zu vernachlässigen ist.

## 4 Sensoren

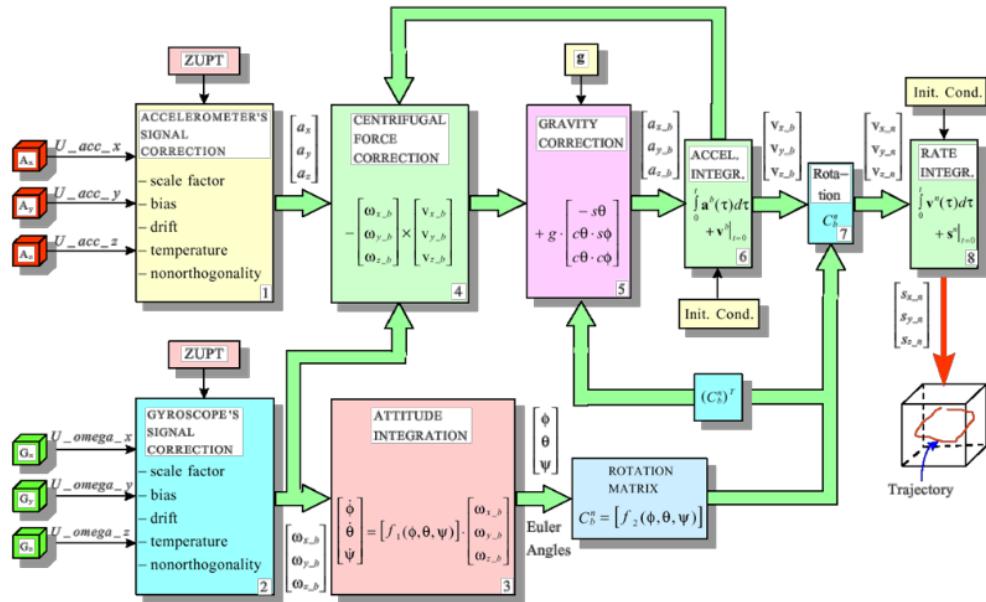


Abbildung 39: Inertialsystem

### 4.2.4 Temperatur – Wo ist es warm?

Temperaturmessung beim Menschen (vgl. Abbildung 40)

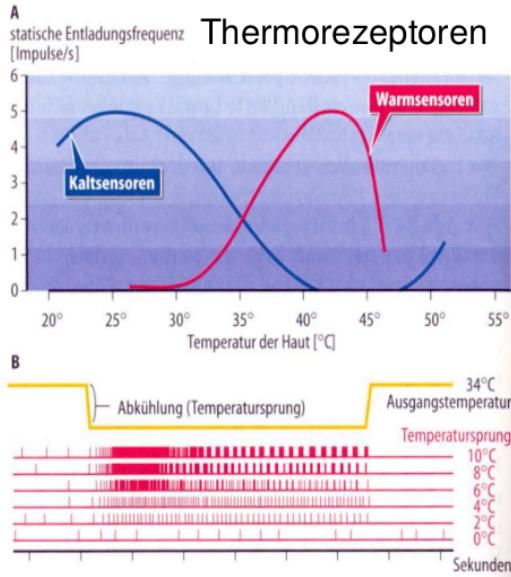


Abbildung 40: Thermorezeptoren

- Die Haut hat spezifische Kalt- und Warmpunkte
- Pro cm<sup>2</sup> Haut 1-4 Kaltpunkte und 0,4 Warmpunkte
- Kalt- und Warmsinn haben unterschiedliche statische und dynamische Empfindlichkeiten
- Empfindlichkeit steigt mit der Geschwindigkeit von Temperaturänderungen; sehr starke Gewöhnung an Signale

### Der Feuerkäfer

- wird vom Feuer angezogen und legt Eier in verbrannte Bäume welche nach Brand „wehrlos“ sind; kann beim Anflug heiße von kühlen Stellen unterscheiden

- Besitzt **photomechanische Rezeptoren**
  - winzige Sinnesborsten auf dem Chitinpanzer
  - Auftreffen von Wärmestrahlung bewirkt Bewegung von Kugelchen durch physikalische Resonanz-Effekt  $\Rightarrow$  Nervenimpuls
  - Sinnesorgane reagieren über viele km auf kleinste Temperaturunterschiede

## Temperaturmessung in der Technik

- Thermoresistive Sensoren: Materialien, die ihren Widerstand in Abhängigkeit von Temperatur verändern oder sich verformen; Einsatz vor allem bei sicherheitskritischen Schaltkreisen (z.B. Feuermelder)
  - Metalle
    - \* Temperaturabhängigkeit des Widerstands zur Messung der Temperatur
    - \* bereichsweise linear
    - \* häufig wird Platin als Messwiderstandsmaterial verwendet
    - \* Platin wird im Bereich -200°C bis +500°C eingesetzt
    - \* als Drähte oder als Metallschichtwiderstände in Dünnschichttechnik ausgeführt
    - \* Messstrom nimmt Einfluss auf die Genauigkeit
  - Keramikwerkstoffe
    - \* Nichtlinear
    - \* PTC-Widerstände (Kaltleiter, dotierter polykristalliner Titanatkeramik)
    - \* NTC-Widerstände (Heißleiter, polykristalline Mischoxidkeramik)
- Thermoelektrische Sensoren:
  - zwei verschiedene Metalle in innigen Kontakt
  - zwischen Enden kann eine Spannung (Thermospannung) gemessen werden
  - Thermospannung ist proportional zur Temperatur des Kontaktes

### 4.2.5 Tasten – Schreibtischplatte oder Kaktus...

#### Menschliche Haut:

Die menschliche Haut weist spezialisierte Tastrezeptoren auf. Die Hauptrezeptoren beim Menschen (in Abbildung 41 dargestellt) sind

1. Berührung
2. Druck
3. Vibration
4. Berührung
5. Druck
6. Dehnung

#### Fühler von Insekten:

- Erfassung der Umgebung mittels Fühlern (können einen torusartigen Bereich abdecken; häufig dort auch Geruchssinn)
- bei LAURON: wenn kein Bodenkontakt wird erst gesucht
- Anpassung der Laufbewegung anhand der Fühler; Beine folgen Fühlern

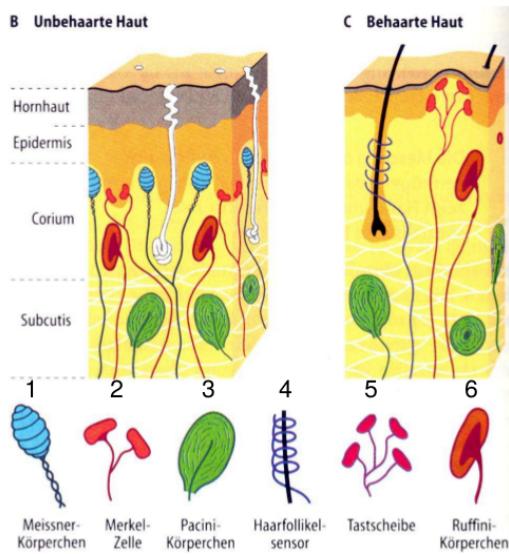


Abbildung 41: Tastende Sensorik

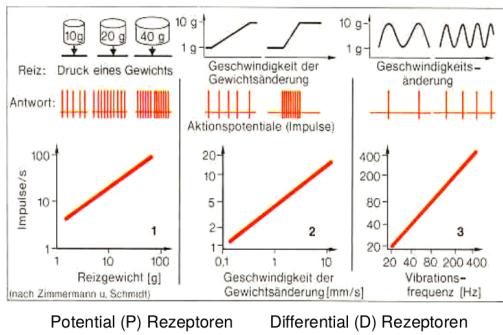


Abbildung 42: Reaktion der Hautrezeptoren für Druck (1), Berührungen (2) und Vibration (3)

- Zyklische Fühlerbewegung synchron mit dem Laufmuster

#### Künstlicher Schnurrbart:

- Schnurrbart bei Mäusen zum Ertasten der Umgebung
- Technische Umsetzung: Feines Haar (menschliches Haar) auf beweglichem Mikrofon

#### Dehnmessstreifen als Kraftsensoren

- Dehnmessstreifen verformt sich bei Krafteinwirkung  
→ Veränderung des ohmschen Widerstands
- Zusammenhang:  $\frac{\Delta R}{R} = \kappa \cdot \varepsilon$  mit
  - $\varepsilon = \frac{\Delta l}{l}$  Längenänderung
  - $\kappa$  Dehnungsempfindlichkeitskonstante
- bei Lauron: 3D Kraftvektor als Messwerte, in Z- sowie X- und Y-Richtung
  - hochpräzise, empfindlich, aufwändige Auswerteelektronik, anfällige Kontaktierung

**Taktile Sensoren:** Kraftmessung über die Nutzung physikalischer Effekte von Materialien. Man unterscheidet folgende Klassen taktiler Sensoren:

- Piezoresistiv
- Piezoelektrisch

- Kapazitiv
- Induktiv
- Optisch (Reflektion, Transduktion, Polarisation)

**Touchscreens** können mittels zahlreicher, unterschiedlicher Technologien umgesetzt werden:

- Resistive Touchscreens
  - Zwei leitfähige, transparente Schichten
  - Gleichspannung an jede Schicht anlegen
  - An Berührpunkt ist Spannung in Schichten gleich
  - Über Spannungsteiler kann Berührpunkt ermittelt werden
- Kapazitive Touchscreens
  - Beschichtete Folie mit transparentem Metalloxid
  - Über Wechselspannung entsteht elektrisches Feld
  - Bei Berührung fließt kleiner Entladestrom
  - Dominierende Technologie: Smartphones (Apple, Samsung, HTC,...)
- Induktion, Akustische Oberflächenwelle, Infrarote Gitter, Hybride

#### 4.2.6 Riechen und Geschmack – Chemie der Stoffe

Das menschliche Riechorgan ist in Abbildung 43 dargestellt.

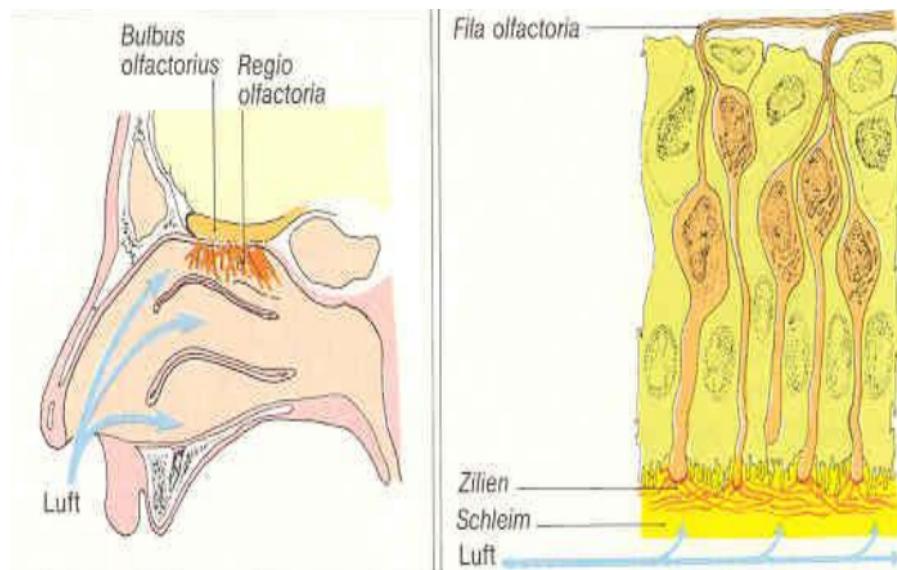


Abbildung 43: Nasenhöhle mit Riechorgan/ Riechepithel: Mehrere Dutzend bis mehrere Tausend Rezeptortypen

**Feuchtigkeitssensoren** (vgl. Abbildung 44):

- Messprinzip erfolgt über Kapazitätsmessung des Dielektrikums
- Schaltung ist aufwendig, wegen benötigtem Wechselspannung

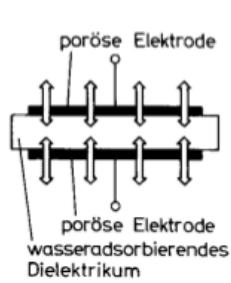


Abb. 26.47 Schematischer Aufbau eines kapazitiven Feuchtesensors

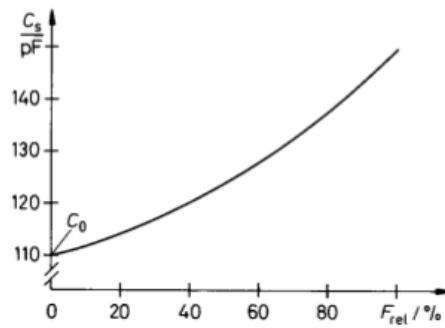


Abb. 26.48 Abhängigkeit der Sensorkapazität von der relativen Feuchte  
Beispiel: Nr. 232269190001 von Valvo

$$\frac{C_s}{C_0} = 1 + 0,4 \left( \frac{F_{\text{rel}}}{100\%} \right)^{1,4}$$

Abbildung 44: Prinzip Feuchtigkeitssensoren

### Gassensor-Arrays:

- KArlsruher MIkroNAse – KAMINA (siehe Abbildung 45a):
  - Entwicklung eines speziellen Gassensor-Mikroarrays
  - Chip, monolithische Fertigung
  - Schnelles Probeentnahmemodul sowie spezielle Aufbereitung
  - Lernalgorithmus, Aufzeichnung von Feature-Vektoren für verschiedene Gerüche → Matching der charakteristischen Kurven
  - Software zur Online-Analyse
  - Optimierung der Baugröße
- Spezielle Gassensor-Mikroarrays als „electronic noses“ zur Weinunterscheidung (siehe Abbildung 45b):
  - fruity, floral, herbaceous, vegetative, spicy, smoky

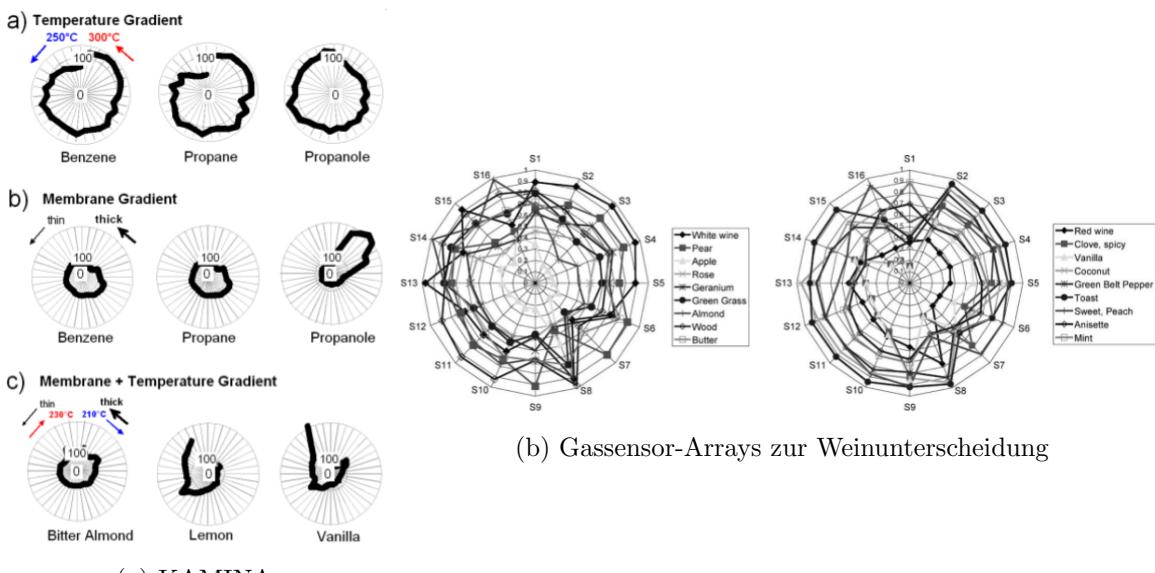
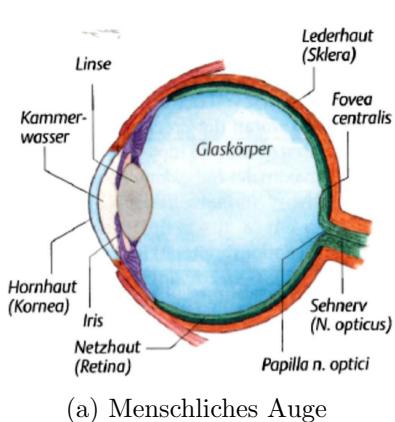
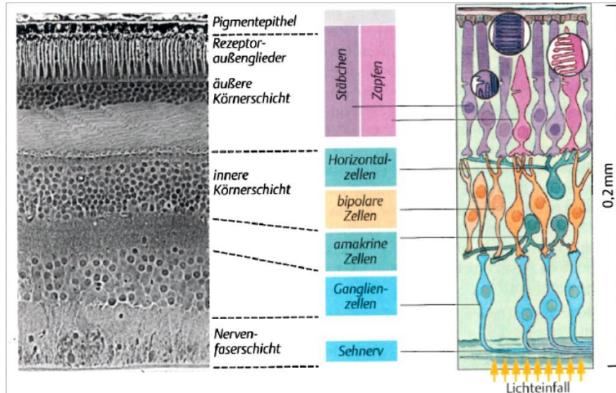


Abbildung 45

#### 4.2.7 Sehen – Facetten, Kamerae und der optische Fluss



(a) Menschliches Auge



(b) Gassensor-Arrays zur Weinunterscheidung

Abbildung 46: Menschlicher Sehsinn

Die **Netzhaut** enthält zwei Rezeptortypen:

- Stäbchen: Hell – Dunkel empfindlich (gestrichelte Linie)
- Zapfen: 3 Arten mit unterschiedlichen Empfindlichkeiten (vgl. Abbildung 47)

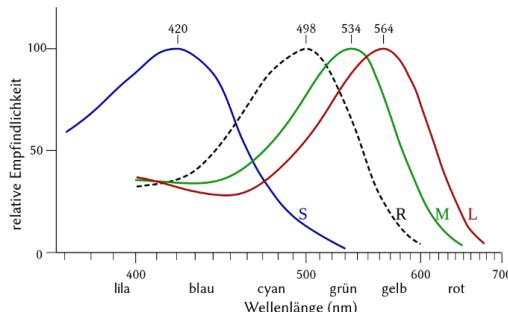


Abbildung 47: Trichromatisches Farbsehen

**Facettenauge bei Insekten:** Die Navigation von Insekten erfordert „**Sensorfusion**“:

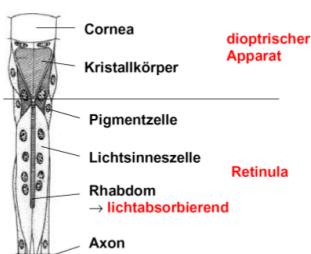


Abbildung 48: Facettenauge

- Biene orientiert sich am Erdmagnetfeld
- Berücksichtigung von Sonnenstand und Sonnenweg
- Polarisation des Himmellichtes
- Geschwindigkeit der Bilder des linken + rechten Auges
- ermöglicht Flug mit geringstem Energieverbrauch und kürzestem Weg

**Technische Facettenaugen** können folgendermaßen realisiert werden:

- Kameraarrays mit unterschiedlichen opt. Eigenschaften:
  - Unterschiedliche Spektralfilter, Brennweiten, Fokus
  - Multi-Kamera Fusion, verbesserte Bild- Segmentierung, größere Tiefenschärfe, 3D-Rekonstruktion
- Kameraarrays mit gleichen optischen Eigenschaften:
  - Einsatz von Mikrolinsen zum Zusammensetzen von Bildern
  - Prinzip der Lichtfeld-Fotografie
  - Schärfe und Fokus nachträglich veränderbar
  - weniger Platzbedarf
  - 3D-Rekonstruktion möglich

Ein Beispiel für die Verwendung **insektenartiger Odometrie** (Methode zur Lageschätzung) ist „Melissa“ (siehe Abbildung 49).

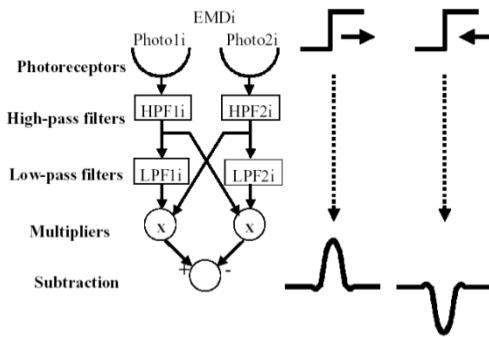


Abbildung 49: Melissa

**Optischer Fluss für Roboter & Flugdrohnen** (vgl. Abbildung 50):

- Optischer Fluss basiert auf Feature detection und Tracking
  - Markante Features werden im Bild erkannt (SIFT, SURF, ...) <sup>1</sup>
  - Bewegung des Features von Frame zu Frame erzeugt Flussvektor
- Optischer Fluss für mobile Robotern, Autos und Flugdrohnen
  - Hardware (Kamera) sehr günstig + Optischen Flusses algorithmisch einfach
  - Nur relative Lokalisierungsmethode (mit Drift),
  - Keine Lokalisierung in featurelosen, homogenen Umgebungen
- Optical Flow Sensoren als kompakte, günstige „Fertig“- Sensoren verfügbar

<sup>1</sup>siehe z.B. Nourani-Vatan2012, *A Study of Feature Extraction Algorithms for Optical Flow Tracking*

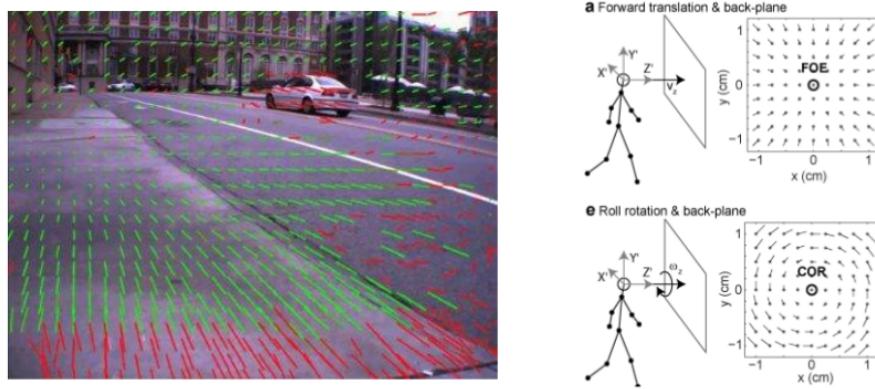


Abbildung 50: Optischer Fluss

## 5 Bewegungssteuerung

*Nichts ist so komplex wie die Bewegungssteuerung in der Biologie... und doch auch so einfach.*

### 5.1 Neuronale Kontrolle in der Natur

#### 5.1.1 Struktur vs. Parametrisierung

Grundsätzlich zwei Teile bei einer Steuerung:

1. Der Aufbau, bzw. das Steuerungsparadigma z.B. Verhaltensbasiertes System, KNN
2. Die Methode zum Parametrisieren des Systems z.B. Empirisch, Reinforcement Learning,  
→ Leider nicht immer eindeutig trennbar!

**Beispiel: PID - proportional–integral–derivative controller:**

- Reglerstruktur:
  - Differentialgleichung:  $u(t) = K_P \left[ e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau + T_V \frac{d}{dt} e(t) \right]$
  - Übertragungsfunktion:  $\frac{U}{E}(s) = K_P + \frac{K_P}{T_N \cdot s} + K_P \cdot T_V \cdot s = K_P(1 + \frac{1}{T_N \cdot s} + T_V) \cdot s$
- Parametrisierung: Bestimmen der Werte  $K_P, T_V, T_N$ 
  - Empirisch / Einstellen durch Ausprobieren
  - Einstellregel nach Ziegler/Nichols
  - Einstellregel nach Chien, Hrones und Reswick

**Elementare Bewegungssteuerung in der Natur:** Bewegungsabläufe

- Willkürmotorik
- Rhythmische Bewegungsabläufe
- Reflexe

**Elementare Bewegungssteuerung in der Robotik:** Verhaltensbasierte Steuerung der sechsbeinigen Laufmaschine LAURON (vgl. Abbildung 51)

- Höhere Verhalten
- Haltungskontrolle
- Reflexe

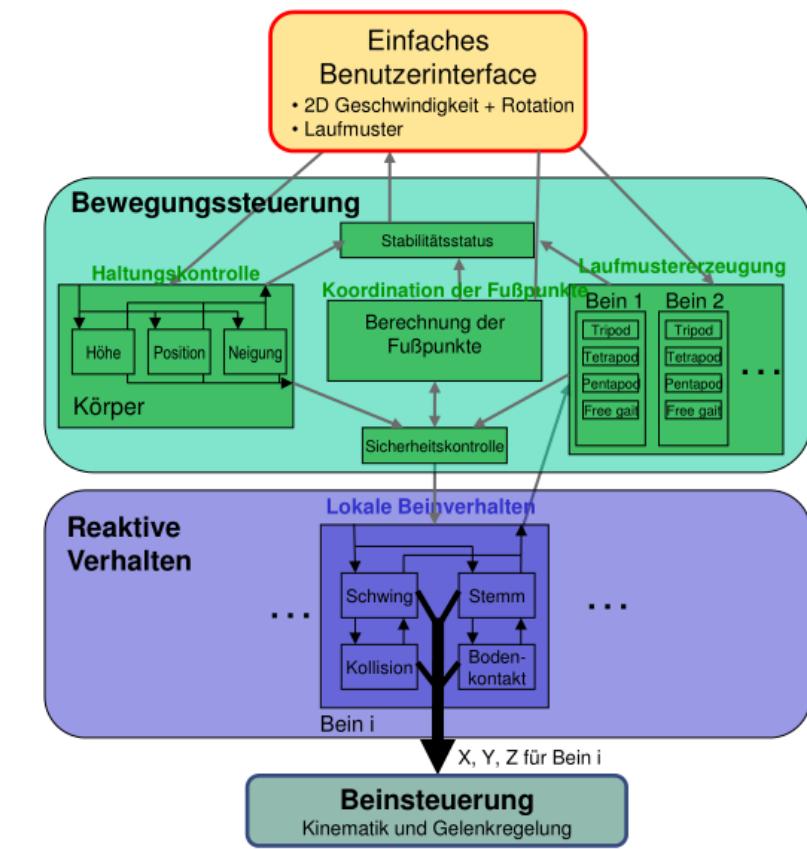


Abbildung 51: Steuerung von LAURON

### 5.1.2 Zentraler Mustergenerator vs. Reflexe

Als **zentraler Mustergenerator** (Central Pattern Generator - CPG) wird ein neuronales Netzwerk verstanden, welches bei isoliertem zentralen Netzwerk (also ohne jede Art Sensorfeedback) die zeitlichen Impulse der rhythmischen Motoraktionen liefert (nach Baessler86). Der Begriff des (rhythmischen) Mustergenerators schließt darauf aufbauend den zentralen Mustergenerator sowie die modulierende Sensorinformation ein.

**Reflexe** sind zielgerichtete, sensorgekoppelte Verhaltensweisen, bei denen motorische sowie vegetative Prozesse ablaufen können. Die Reflexantwort ist dabei hinsichtlich Latenz, Stärke und Muster eng an die Intensität der Sensorerregung gekoppelt (nach Schmidt95).

Beispiele für Mustergeneratoren und Reflexe:

- Rhythmogenese der Atmung (Respiratory CPG)
  - respiratorische Neuronen in verlängerten Rückenmark
  - 3 rhythmische Aktivitäten
  - Inspirationsphase: Einatmungsmuskulatur wird durch Nervenimpulse zur Kontraktion veranlasst
  - Postinspirationsphase: Neuronale Aktivierung der Einatmungsmuskulatur lässt nach. Somit wird eine passive Ausatmung ermöglicht.
  - Aktive Exspirationsphase: Die Ausatmungsmuskulatur aktiviert.
- Pupillenlichtreflex
  - Veränderung der Weite der Iris in Abhängigkeit vom Lichteinfall
  - Beide Augen reagieren auch auf einseitigen Reiz

### 5.1.3 Das Neuron: Funktionale Approximation, KNN

- Eingabe durch eine Propagierungsfunktion mit Gewichten verknüpft
- Aktivierungsfunktion definiert Reaktion auf Eingabe

#### Künstliche Neuronale Netze - Historischer Abriss:

- 1943: McCulloch und Pitts beschreiben erstes neurologisches Netz
- 1949: Hebb'sche Lernregel: aktivierende oder hemmende Wirkung einer Synapse lässt sich als Produkt der prä- und postsynaptischen Aktivität berechnen
- 1958: Frank Rosenblatt präsentiert das Perzeptron, welches die Grundlage der heutigen KNNs bildet
- Heute: Große Vielfalt an Topologien (Strukturen der Netze) und Verbindungsarten (z.B. „spiking neural networks“), spezialisierte Netze

#### Anwendungsgebiete von KNNs in biologisch motivierter Robotik:

- Robotersteuerungen
- Regelung komplexer Bewegungsabläufe
- Sprachgenerierung (Beispiel: NETtalk)
- Bildverarbeitung und Mustererkennung
- Spracherkennung
- virtuelle Agenten

#### KNNs – Prinzipien:

- Einfaches Einzelneuron – komplexes Gesamtsystem
- Hohe Fehlertoleranz
- Integration sensorischer Massendaten
- Lernvorgang (bestärkendes, (un)-überwachtes, stochastisches Lernen)
- Direkte Generierung kontinuierlicher Steuersignale
- Komplizierte nichtlineare Funktionen erlernbar

#### Netztopologien:

- Einschichtiges Feed-Forward-Netzwerk (azyklischer gerichteter Graph): eine Eingabeschicht, eine Ausgabeschicht (azyklischer gerichteter Graph)
- Mehrschichtiges Feed-Forward Netzwerk: 1-n verdeckte Schichten, verbesserte die Abstraktionsfähigkeit, z.B. XOR Problem lösbar
- Rekurrentes (rückgekoppeltes Netz):
  - direkte Rückkopplungen (*direct feedback*)
  - indirekte Rückkopplungen (*indirect feedback*)
  - seitliche Rückkoppelungen (*lateral feedback*)
  - vollständige Verbindungen (jedes Neuron ist mit jedem anderen Neuron verbunden)

### Neuro-Oszillatoren:

- Paare aus sich gegenseitig hemmenden und erregenden Neuronen
- Der Antagonistische Aufbau der Muskeln setzt sich in der Steuerung fort
- Dienen auf höheren Ebenen als Mustergeneratoren

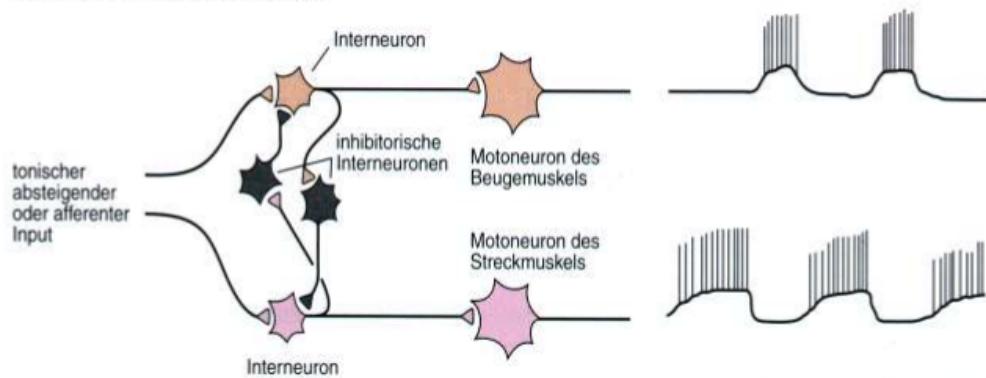


Abbildung 52: Neuro-Oszillator: rhythmische alternierende Aktivität

### Verhaltensbasierte Systeme:

- Komplexe Verhalten entstehen nicht notwendigerweise aus komplexen Steuerungssystemen
- Kein Modell sondern direkte Interaktion mit der Umwelt
- Charakteristika:
  - Situatedness: Der Roboter ist als Einheit in seiner Umgebung eingebunden
  - Embodiment: Der Roboter hat eine physikalische Präsenz
  - Emergence: Intelligenz entsteht durch Interaktion mit der Umwelt

**Hierarchie biologisch motivierter Steuerstrukturen** (zunehmende Abstraktion und Komplexität):

0. Mustergeneratoren → CPGs, zyklische Bewegungen
  1. Reflexe → reaktive Systeme in Robotern
  2. Neuronen → KNNs
  3. Verhalten → verhaltensbasierte Systeme (vgl. Kap. 9)
- es werden aber auch mehr und mehr Parameter!

**Steuerungsstrukturen parametrieren:** *Parameteridentifikation und Tuning stellt bei komplexen Robotern und anspruchsvollen Aufgaben eine zentrale Herausforderung der Entwicklung dar.*  
Es gibt verschiedene Verfahren und Methoden:

- Modellbasierte Parametrierung – „Regler Einstellregeln“
- Biologische Untersuchungen (z.B. KNN bei Insekten)
- Empirisches Parametrieren anhand von umfangreichen Versuchen und Expertenwissen
- Systematische, lernende Ansätze:
  - Maschinelles Lernen: finde iterativ Optimum im Parameterraum über mehrdimensionalen Gradientenabstieg; verschiedene Verfahren zum Beschleunigen des Lernvorgangs
  - Genetische Algorithmen: Problemrepräsentation durch „Genom“, Mutation und Rekombination

## 5.2 Neuronale Steuerung der Stabheuschrecke

### 5.2.1 Warum Stabheuschrecke als Vorbild?

Stabheuschrecke (*carausius morosus*):

- großer 6-Beiner → Statisch stabiles Laufen, Überwinden komplexer Hindernisse, läuft auch Wände hoch, sehr robust
- Langsam → Realisierbar mit verfügbarer Aktorik (z.B. Elektromotoren)
- Einfacher kinematischer Aufbau → Einfache, gut beherrschbare mechanische Konstruktion
- Trotzdem sehr flexibel → Viele Forschungsbereiche und Herausforderungen
- Weitreichende biologische Untersuchungen vorhanden (z.B. Prof. H. Cruse, Prof. A. Büschges)  
→ Koordinationsmechanismen („Cruse Regeln“)
- Verteilte dezentrale Steuerung → Verteile Hardware

### 5.2.2 Laufmuster und Cruse-Regeln

Mehrbeinige Laufmuster werden oft durch die Anzahl der am Boden verbleibenden (tragenden) Beine klassifiziert:

- Tripod: zu jedem Zeitpunkt mind. 3 Bein auf Boden
  - Tetrapod: zu jedem Zeitpunkt mind. 4 Beine auf Boden
  - Pentapod: zu jedem Zeitpunkt mind. 5 Beine auf Boden
- Verwendung der Begriffe Tetrapod und Tripod verleitet zu der Annahme der Existenz *fester Gangmustern*.
- Es ist aber bekannt, dass Insekten einen *freien Gang* verwenden.

**Koordinationsregeln nach Cruse:**

1. Hintere Schwingphase hemmt Start vorderer Schwingphase: wenn hinteres Bein in der Luft ist soll Vorderes nicht auch in der Luft sein (vgl. Abb. 53a)
2. Beginn der Stemmpphase löst vorderes Schwingen aus (vgl. Abb. 53b)
3. Beinposition während der Stemmpphase reizt Start der Schwingphase: je weiter das Bein sich am Ende seines Arbeitsraumes befindet, desto stärker wird der Start der Schwingphase des Vorderbeins motiviert (vgl. Abb. 53c)
4. Position beeinflusst die Position am Ende der Schwingphase („targeting“)
5. Erhöhter Widerstand erfordert erhöhte Kraft (Stemmen)
6. Tread-On-Tarsus Reflex: wenn man der Heuschrecke über den Fuß streift hebt sie ihn, was verhindern soll, dass sie sich selbst auf den Fuß tritt

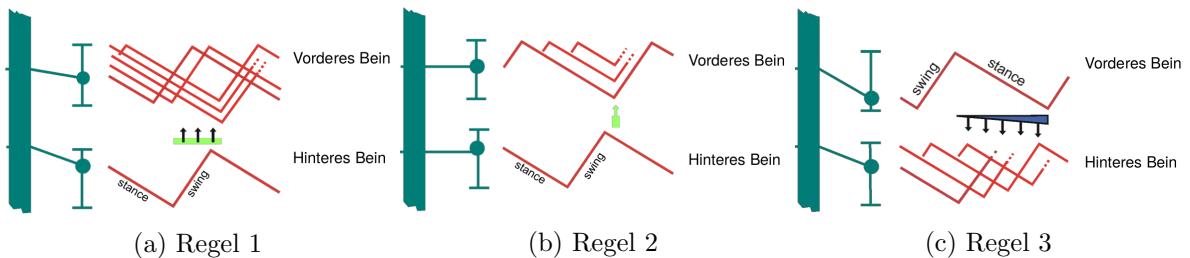


Abbildung 53: Cruse-Regeln 1 - 3

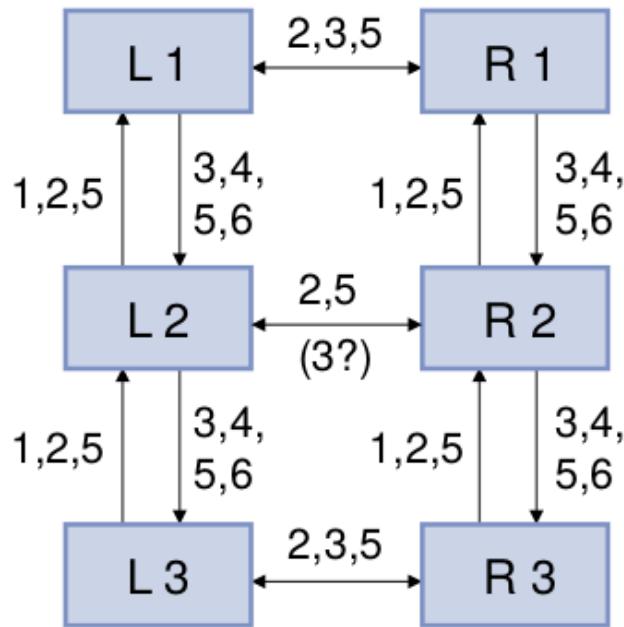


Abbildung 54: Cruse-Regeln

### 5.3 Oszillatoren als Grundbausteine für natürliche Bewegungssteuerung

#### 5.3.1 Rhythmusgeneratoren, Erweiterung des Neuronen Modells

Als **zentraler Mustergenerator** (Central Pattern Generator - CPG) wird ein neuronales Netzwerk verstanden, welches bei isoliertem zentralen Netzwerk (also ohne jede Art Sensorfeedback) die zeitlichen Impulse der rhythmischen Motoraktionen liefert (nach Baessler86). Der Begriff des (rhythmischen) Mustergenerators schließt darauf aufbauend den zentralen Mustergenerator sowie die modulierende Sensorinformation ein.

- neuronales Netz ohne externe Sensorinformation
- in der Lage rhythmische Muster zu generieren
- genutzt um rhythmische Motoransteuerungen zu erzeugen

**Reflexe** sind schnelle, automatische und unbewusste Reaktionen, die von einem sensorischen Stimulus ausgelöst werden. Es sind zielgerichtete, sensorgekoppelte Verhaltensweisen, bei denen motorische sowie vegetative Prozesse ablaufen können. Die Reflexantwort ist dabei hinsichtlich Latenz, Stärke und Muster eng an die Intensität der Sensorerregung gekoppelt (nach Schmidt95).

- Beispiel: Muskelkontraktion bei Hitze (taktil), Ausweichen vor einem Ball (visuell)

**Taxen** (*sg. Taxis*) sind Verhalten, die ein Tier in Richtung (attraktive) oder entgegen (aversive) eines Stimulus orientieren.

- Beispiel: Orientierung von Bienen nach Sonnenstand, Bewegung von Lichtquelle weg bei Asseln

**Feste Muster** sind Reaktionen auf Stimuli, die länger anhalten als die Stimuli selbst.

- Fluchtreflex hält länger an als Stimulus selbst („Reflex“ ist hier eine irreführende Bezeichnung)

Drei Modelle von Rhythmusgeneratoren:

- **Half-Center Modell** (Abb. 55a):

- ursprünglich entwickelt zur Beschreibung der Beuger-Strecker Aktivitäten bei Katzen
- Durch andere „Driver“-Neuronen ständig angeregt
- Zentren inhibieren sich gegenseitig

- **Closed-Loop Modell** (Abb. 55b):

- feinere Aufteilung einzelner Flexor- und Extensor-Aktivitäten (Beuger, Strecker)

- **Schrittmachermodell** (Abb. 55c):

- intrinsische rhythmische Eigenschaften eines einzelnen Neuron erzeugt Oszillation

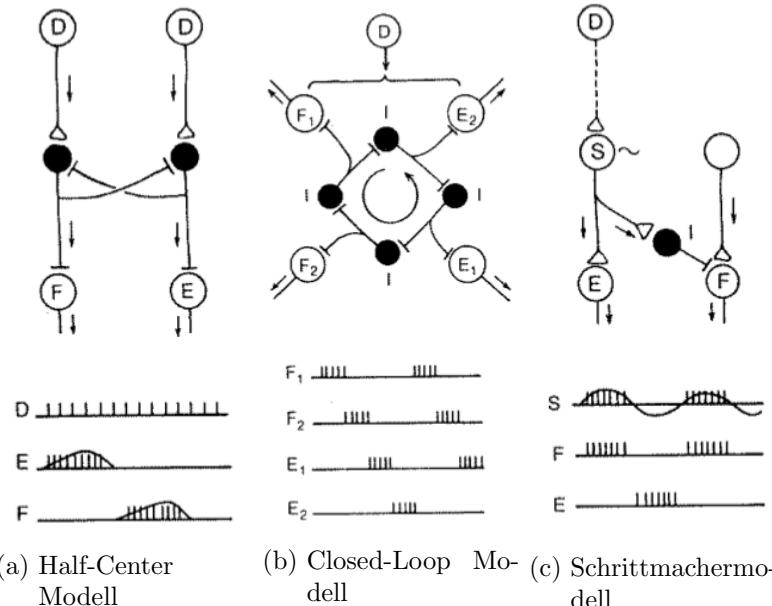


Abbildung 55: Modelle von Rhythmusgeneratoren

### Das Neuron – Funktionale Approximation:

- Eingabe durch eine Propagierungsfunktion mit Gewichten verknüpft
- Aktivierungsfunktion definiert Reaktion auf Eingabe
- Allgemeine Gleichung:  $y = F(\sum_i w_i x_i)$  mit
  - $\vec{X}$  : Eingabevektor
  - $w_i$  : Gewichte ( $> 0$  anregend,  $< 0$  hemmend)
  - $y$  : Ausgabe
  - $F$  : Übertragungsfunktion, z.B. Sigmoidfunktion  $y = \left(1 + e^{-\sum_i w_i x_i}\right)^{-1}$

### 5.3.2 Oszillatormodelle: Leaky Integrator, Matsuoka

„Leaky Integrator“ - Neuron:

Verwendung einer Zustandsvariable  $m_i$  (Membranpotential), vgl. Abb.

- Membran Differentialgleichung:  $\tau_j \frac{dm_j}{dt} = -m_j + \sum_i w_{ij}x_i$
- Ausgabefunktion:  $x_j = (1 + e^{-(m_j + b_j)})^{-1}$

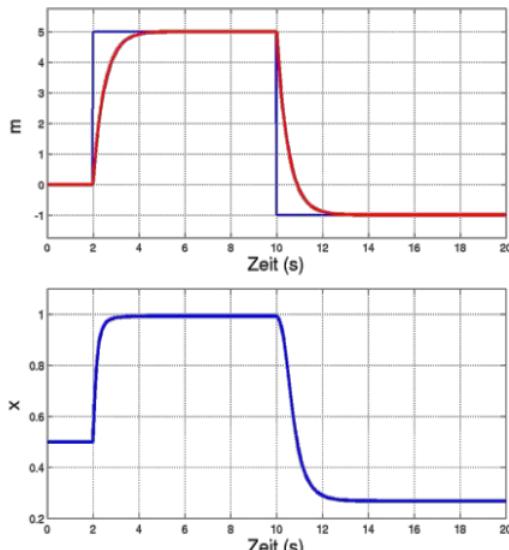


Abbildung 56: „Leaky Integrator“ Modell

**Matsuoka Oszillatoren<sup>2</sup>** – Formale Beschreibung des Half-Center Modells

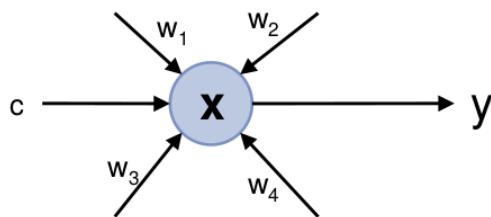
- einzelnes Neuron:

$$T_r \dot{x}_i + x_i = - \sum_{j=1, j \neq i}^n w_{ij} y_j + c - bf$$

$$T_a \dot{f} + f = y, y = \max(0, x)$$

mit

- $y_i$ : Activation ( $i = 0$  eigene)
- $x_i$ : Membrane potential
- $c$ : External excitation
- $f$ : Degree of adaptation
- $b$ : Parameter, steady-state firing
- $T_r, T_a$ : Zeitkonstanten,  $r$ =rise,  $a$ =adapt



<sup>2</sup>Mechanisms of frequency and pattern control in the neural rhythm generators, K. Matsuoka, Biol. Cybern. 56, 1987  
Sustained oscillations generated by mutually inhibiting neurons with adaptation, K. Matsuoka, Biol. Cybern. 52, 1985

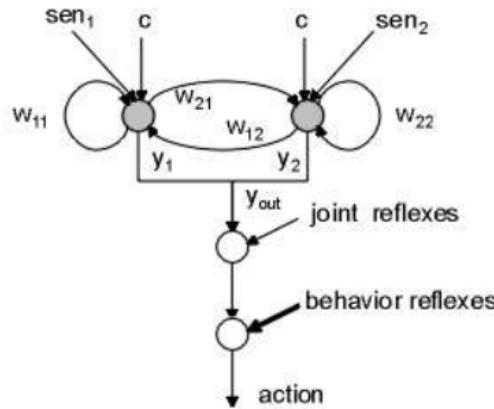
- Oszillatoren:

$$T_r \dot{x}_i + x_i = - \sum_{j=1, j \neq i}^n w_{ij} y_j + c - b f + \sum_{k=1}^m v_k s_{en_k}$$

$$T_a \dot{f}_i + f_i = y_i, y = \max(0, x_i)$$

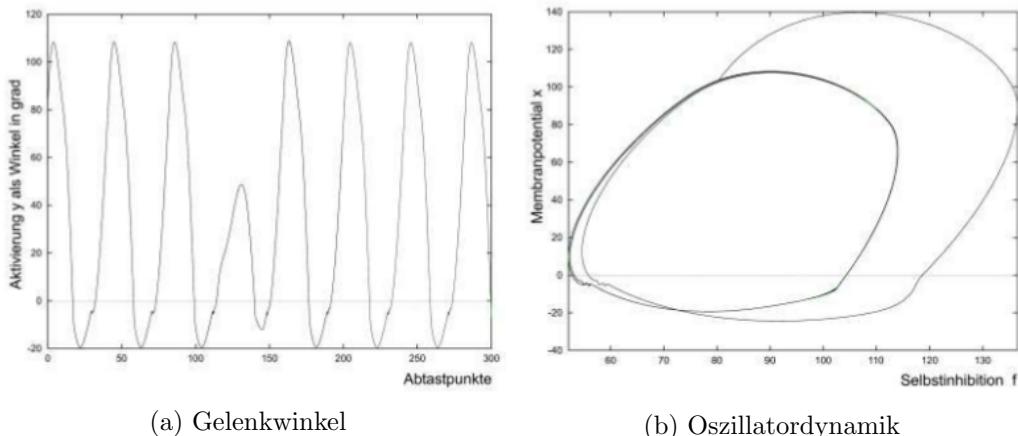
mit

- $s_{en_i}$  : Sensor input

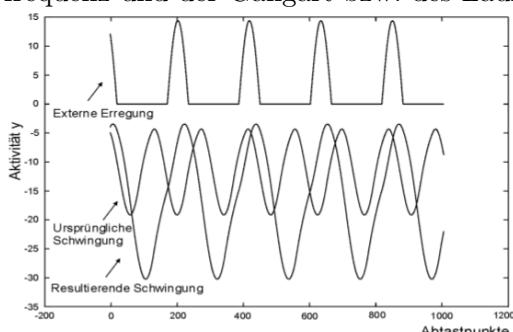


- Attraktorverhalten:

- Nach Auftreten der Störung schwingt sich der Oszillatator wieder auf den Attraktor ein und das System kehrt in einen stabilen Zustand zurück (siehe Abb. 57a)
- Abb. 57b veranschaulicht die Dynamik des Neuro-Oszillators: Aktivierung/Membranpotential  $x$  ist auf der Y-Achse abgetragen, die Selbstinhibition  $f$  auf der X-Achse



- Aufzwingen einer Frequenz: Anpassungsfähigkeit des Oszillators hinsichtlich der durch externe Erregung (durch sensorische Information) aufgezwungenen Frequenz → Änderung der Schritt-frequenz und der Gangart bzw. des Laufmusters



- Oszillatoren in der Natur:

- Der Antagonistische Aufbau der Muskeln setzt sich in der Steuerung fort, zum Beispiel hemmen und erregen die Motoneuronen der Gegenspieler sich wechselseitig.
- Die Existenz von Mustergeneratoren (MG) in der Natur gilt als unzweifelhaft. Umstritten ist die Frage, ob es einen Zentralen Mustergenerator (ohne Sensorfeedback) gibt.
- Preflexe und Reflexe dienen bei der Bewegung als Mittel zur Kompensation von externen Störungen, das heißt sie überlagern die Ausgaben der CPGs. Der gestörte Fall bei der Lokomotion ist aber die Regel, nicht die Ausnahme. Kann es deshalb nicht auch sein, dass der „klassische“ Neuro-Oszillator das Artefakt einer verkümmerten Reflexkette ist?

## 5.4 Anwendungsbeispiele für Oszillatoren

### 5.4.1 Neunauge, Salamander – Schwimmen und Laufen, Tekken – Vierbeiniges Laufen

- **Neunauge** (Lamprey) ist einer der primitivsten Vertebraten. Es existieren weitreichende neurobiologische Untersuchungen
  - Künstliches Neunauge (Grillner, Ekkeberg, Ijspeert): 2D-Biomechanisches Modell, Multi-Segment Körper mit Feder-Dämpfer Muskeln, Leaky-Integrator-Neuronen-Netzwerk
- **Schwimmender Roboter (Ayers)**: Mit Nitinol-Muskeln bewegter Roboter nach Neunauge-Vorbild; Bewegung basierend auf 4-Neuron Oszillatoren
- Salamander (A.J. Ijspeert): Vierbeiniges Landwirbeltier (Vertebrate), amphibienartiges/reptilienartiges Laufen
  - Künstlicher Salamander: 2D/3D-Biomechanisches Modell, Multi-Segment Körper und 1DOF Beine mit Feder-Dämpfer Muskeln, „klassisches“ Leaky-Integrator-Neuronen-Netzwerk, Netzwerk durch evolutionäre Algorithmen generiert, jeweils 1 Motoneuron und 3 Interneuronen
  - Aktueller Stand: auch bei Katzen, nicht nur bei einfachen Wirbellosen vorhanden
    - sehr viel Vorkodierung im Rückenmark (decerebrierte Katzen laufen auf Laufband)
  - Laufroboter Tekken

### 5.4.2 Zusammenfassung & Diskussion

- + Angenehmes Attraktor-Verhalten: Rückkehr zu stabilen Rhythmus
- + Einfacher Aufbau bei großer Funktionalität
- + Kann ohne Sensorfeedback eingesetzt werden
- Ohne Erweiterung anfällig für größere Störungen (mit Reflexen auch robustes Steuerung möglich)
  - Schwer zu parametrisieren
  - Komplexe mathematische Modellierung (DGL, evtl. rechenintensiv)
  - Bei komplexeren Kinematiken nur noch schwer beherrschbar

## 6 Mehrbeinige Lokomotion

*Kaum etwas sieht so einfach aus und ist im Detail doch so komplex*

## 6.1 Lokomotionsarten

*Lokomotion beschreibt die aktive Fortbewegung eines Lebewesens*

- **Kriechen:** Fortbewegungsart, bei der die Körperunterseite immer Bodenkontakt hat  
Natürliche Vertreter: (beinlose) Vertebrate (Schlangen, Reptilien, Amphibien,...)
- **Hüpfen:** Fortbewegungsart mit vollständiger Flugphase (keinerlei Bodenkontakt), typischerweise ergibt sich eine ballistische Flugbahn, lange Flugphase  
Natürliche Vertreter: (eher selten) z.B. Känguru, Frosch
- **Laufen:** Fortbewegungsart mit wiederkehrendem Wechsel des Bodenkontakte, Unterscheidung zwischen „Gehen“ und „Rennen“, Unterschiedliche Laufmuster (je nach Anzahl der Beine)  
Natürliche Vertreter: Invertebrate, Vertebrate, eigentlich alle

## 6.2 Stabilität beim Laufen

- Laufroboter müssen in der Regel AKTIV stabilisiert werden
  - Nicht zu viele Beine anheben
  - Körperschwerpunkt ausregeln
- Stabilität wird maßgeblich von Laufmuster beeinflusst
- Unterscheidung: Statische Stabilität vs. Dynamische Stabilität
- Faustregel: Je mehr Beine ein Roboter hat, desto einfacher ist es diesen stabil zu halten

### 6.2.1 Statische Stabilität

- Laufbewegung kann zu JEDEM Zeitpunkt angehalten werden → System bleibt stehen (z.B. ein rennender Gepard ist also NICHT statisch stabil)
- Mindestens 3 Beine haben Bodenkontakt
- Beine mit Bodenkontakt bilden Aufstandspolygon (stark abhängig von der Geometrie des Roboters, z.B. lange Beine = großes AP)
- Projizierter Schwerpunkt (CoM = Center of Mass) muss innerhalb des Aufstandspolygons liegen → 6-beinige sind statisch stabil
- Stabilitätsmaße definieren Wert für Stabilität:
  - **Stability Margin:** Minimaler Abstand zum Support-Polygon Rand (vgl. Abbildung 58b)
  - **Longitudinal Stability Margin:** Minimaler Abstand zum Rand in Laufrichtung (vgl. Abbildung 58c)

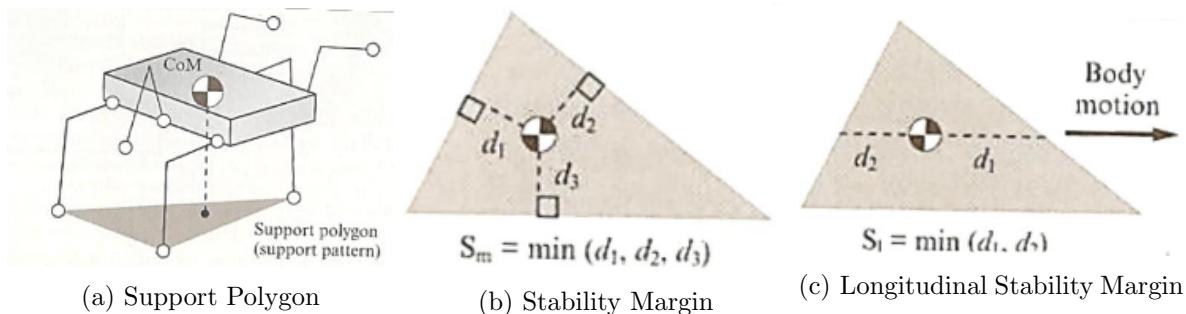


Abbildung 58: Stabilitätsmaße

### 6.2.2 Dynamische Stabilität

- Zyklische Bewegung ohne umzufallen
- Deutlich komplexere Stabilität, da dynamische Effekte (Geschwindigkeiten, Trägheitsmomente) berücksichtigt werden
- Bei fortschreitender Bewegung bleibt das System aufrecht/stabil
- Zahlreiche (komplexe) Stabilitätskriterien:
  - Center of pressure method (EMC)
  - Dynamic stability margin (DSM)
  - Tumble stability margin (TSM)
  - Force-angle stability margin (FASM): *A new measure of tipover stability margin for mobile manipulators*, Papadopoulos1996
  - Zero Moment Point (ZMP): *Zero-moment point – Thirty five years of its life*, Vukobratovic2005
  - Normalized dynamic energy stability margin (NDESM)

### 6.2.3 Stabilitätskriterien

#### Vukobratovic - Zero Moment Point (ZMP)

- Punkt, in dem sich, wenn er auf den Boden projiziert wird, alle Kräfte auslöschen
  - anschaulich: ZMP  $\approx$  „Drehpunkt“ auf dem Boden
  - Wenn der ZMP innerhalb des Aufstandspolygons / Fußfläche liegt ist der Roboter stabil
  - Fußgröße ist hier wichtig; Bodenkontaktkräfte verteilen sich anteilig über den gesamten Fuß<sup>3</sup>
- flächiger Bodenkontakt: alle punktuellen Kontakte können zu einem zentralen Kontaktspunkt zusammengefasst werden, gewichtet anhand der Abstände und der Kräfte
  - $\mathbf{p}_i$ : Gewichte der  $yxz$ -Position auf der Ebene
  - $\mathbf{f}_i$ : Kraftvektoren (ergeben sich aus Sensorwerten, normal jeweils vier Kraftsensoren vorne und hinten auf Fußfläche)
- Seien  $\mathbf{p}_i$  die Fußpunkte, mit den Kräften  $f_i = [f_{ix}, f_{iy}, f_{iz}]$ , dann ist der ZMP definiert als

$$\mathbf{p} := \frac{\sum_{i=1}^N \mathbf{p}_i f_{iz}}{\sum_{i=1}^N \mathbf{p}_i f_{iz}}$$

- Drehmomente berechenbar durch

$$\tau = \sum_{i=1}^N (\mathbf{p}_i - \mathbf{p}) \times \mathbf{f}_i$$

---

<sup>3</sup>Daher ist es so beachtlich, dass *Pacman* von Boston Dynamics normale Menschen-Turnschuhe trägt; viele 2-beinige Roboter haben sehr große Füße.

- Anhand der Definitionen werden die Drehmomente in  $x$ - und  $y$ -Achse jeweils Null:  $\tau_x = \tau_y = 0$  [Zero Moment Point]

$$\begin{aligned}\tau_x &= \sum_{i=1}^N (p_{iy} - p_y) f_{iz} - \sum_{i=1}^N (p_{iz} - p_z) f_{iy} && \text{zweiter Term wird 0} \\ \tau_y &= \sum_{i=1}^N (p_{iz} - p_z) f_{ix} - \sum_{i=1}^N (p_{ix} - p_x) f_{iz} && \text{erster Term wird 0} \\ \tau_x &= \sum_{i=1}^N (p_{ix} - p_x) f_{iy} - \sum_{i=1}^N (p_{iy} - p_y) f_{ix}\end{aligned}$$

- Im statischen Fall kann der Schwerpunkt einfach auf den Boden projiziert werden
- Über Roboterdynamik kann ein virtueller / computed ZMP berechnet werden
- ZMP bereits 1968 von Vukobratovic in Moskau präsentiert
- ZMP macht recht starke Annahme: finde den Punkt, an dem sich alle Kräfte ausgleichen → Kriterium funktioniert im Grunde nur auf flachem Boden, in unebenem Gelände nicht so gut anwendbar
- Dennoch weitverbreitetes dynamisches Stabilitätskriterium insbesondere für zweibeinige Laufroboter und weltweit angewandtes Konzept in der humanoiden Robotik (z.B. laufender *ASIMO* von Honda beweist, dass ZMP ein gutes Kriterium ist)

### Force-Angle Stability Measure (FASM)

- Ursprünglich für Baufahrzeuge entwickelt
- Winkel zur Kippachse als Stabilitätsmaß

$$\begin{aligned}\mathbf{x}_{cog} &= \frac{1}{\sum_{i=1}^N M_i} \sum_{n=0}^N m_i \cdot \mathbf{x}_{cog} \\ \alpha &= \min(\theta_i) \|\mathbf{f}_r\|\end{aligned}$$

- Wenn Winkel  $\theta = 0$  wird kippt man um
- Berücksichtigt die Höhe
- Kann die externe Störkräfte berücksichtigen
- schnell und effektiv berechenbar
- Eingangsgrößen: CoM, Gewichtskraft auf CoM, externe Kräfte und Momente (auf CoM), Fußpunkte

## 6.3 Laufmuster

### Gait-Diagramme (siehe z.B. Abbildung 61)

- Zur übersichtlichen Darstellung von Laufmustern
- Weit verbreitet: sowohl in Biologie wie Robotik
- X-Achse: zeitlicher Verlauf
- Y-Achse (pro Bein): Darstellung der Schwing- bzw. Stemmpphase  
(meist: schwarzer Balken = Bodenkontakt/Stemmpphase)

- Laufmuster aus Gait-Diagramm „ablesbar“

### Phase-Shift

- Phase-Shift des  $Bein_i$  (Phasenverschiebung): Laufmuster können über Phasenverschiebung zwischen den Beinen beschrieben/definiert werden

$$\phi_i = \frac{\text{BeginnStemmphase}Bein_i}{T}$$

### Duty-Factor

- Duty-Factor des  $Bein_i$  (Belastungsfaktor): Beschreibt den relativen Anteil der Stemmphase zu der Gesamtzykluszeit ( $T = \text{Schwingzeit} + \text{Stemmzeit}$ )

$$\beta_i = \frac{\text{Stemmzeit}Bein_i}{T}$$

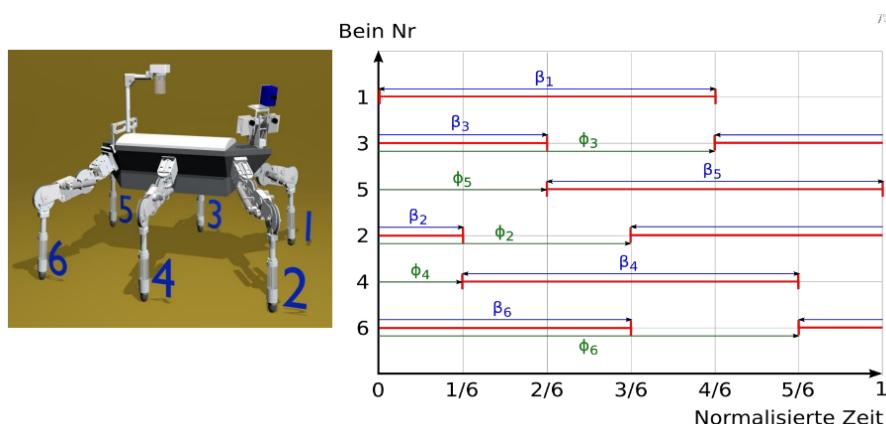


Abbildung 59: Phase-Shift und Duty-Factor

#### 6.3.1 Unterscheidungskriterien für Laufmuster

Laufmuster haben wesentlichen Einfluss auf die Leistungsfähigkeit von Laufrobotern:

- Stabilität
- Geschwindigkeit
- Effizienz

Wenn man über Laufmuster spricht sind die folgenden Kriterien in dieser Reihenfolge anzugeben:

1. **Anzahl der Beine:** vgl. Abbildung 60
2. **Symmetrie der Beinpaare:** Wie ist die Phasenverschiebung innerhalb eines Beinpaars? (vgl. Abbildung 61)
3. **Zyklisches Laufmuster:** Ist ein wiederkehrendes Muster erkennbar? (vgl. Abbildung 62)

Bemerkung: Wie unterscheiden sich Kriterien 2 und 3?

- Kriterium 2 bezieht sich auf die Phasenverschiebung innerhalb der Beinpaare (z.B. vordere Beine: LF, RF); symmetrisch: 0.5, ansonsten asymmetrisch
- Bei Kriterium 3 bleiben die Phasenverschiebung zwischen den Beinen und der Duty-Factor konstant; Muster (kann aus mehreren Schrittfolgen bestehen) bleibt gleich

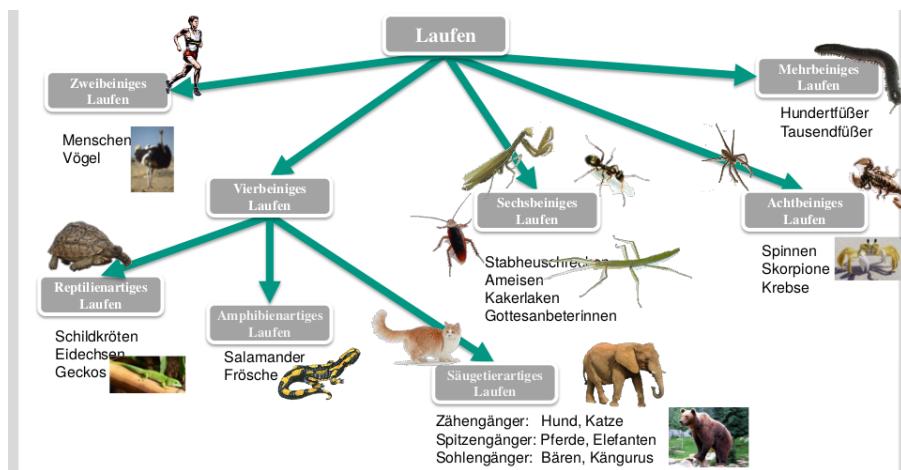


Abbildung 60: Laufarten

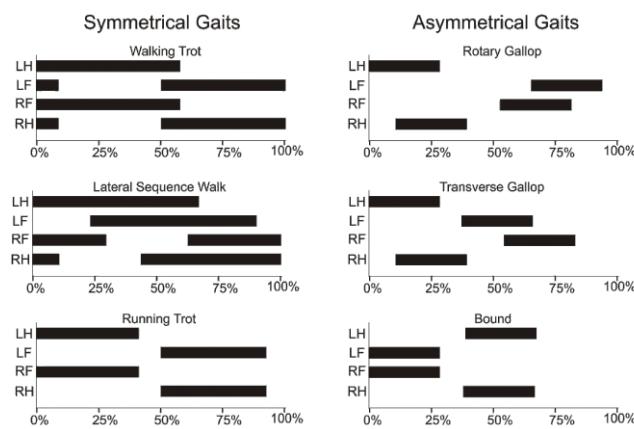


Abbildung 61: Symmetrie der Beinpaare: Beim asymmetrischen Gang schwingen RH und LH in Phase (gleichzeitig), während beim symmetrischen Gang eine klare Phasenverschiebung von 0.5 auftritt.

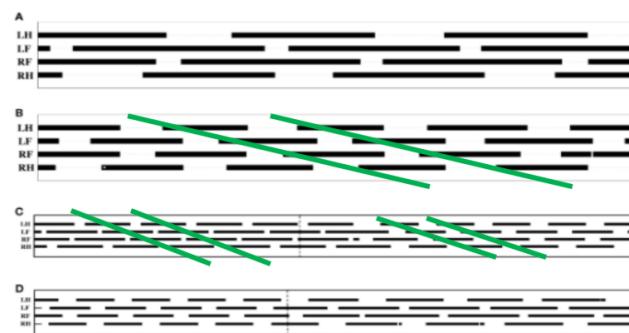


Abbildung 62: Zyklen

#### 4. Anzahl der Bodenkontakte/Beine auf dem Boden:

- Laufmuster wird über die Mindestzahl an Beinen mit Bodenkontakt definiert
- Mehr Beine im Kontakt mit Boden → größere Stabilität
- Meist handelt es sich um zyklische Laufmuster
- Typische Beispiele sind: Tripod (immer mind. 3 Beine auf dem Boden), Tetrapod (immer mind. 4 Beine auf dem Boden), Pentapod (immer mind. 5 Beine auf dem Boden)

### 6.3.2 Sechsbeinige, vierbeinige, zweibeinige Laufmuster

#### Hexapod/6-Beiner:

- Schnellstes statisch stabiles Laufmuster: Tripod (abwechselnd 3 Beine auf Boden, zu jedem Zeitpunkt statisch stabil), siehe Abbildung 63 A
- Weitere typische Laufmuster: Tetrapod, Pentapod (4 bzw. 5 Beine auf dem Boden, zu jedem Zeitpunkt statisch stabil), siehe Abbildung 63 B
- Cruse-Regel basiertes Laufmuster (Mix aus den obigen), siehe Abbildung 63 Mischung aus A und B
- Free-Gait – „Freies Laufmuster“: große Flexibilität, besonders für schwieriges Terrain gut geeignet, siehe Abbildung 63 C

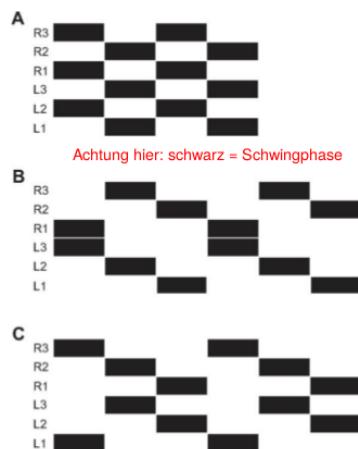


Abbildung 63: Gangarten 6-beiniger Insekten

**Free Gait am Beispiel der Stabheuschrecke** Die Verwendung der Begriffe Tetrapod und Tripod verleitet zu der Annahme der Existenz fester Gangmustern. Es ist aber bekannt, dass Insekten einen freien Gang verwenden. Die Koordination der Beine erfolgt durch mehrere Koordinationsregeln (vgl. Unterunterabschnitt 5.2.2). Bei der Stabheuschrecke gilt die einfache Regel: *Wenn der linke und rechte Nachbar auf dem Boden sind, dann darf das Bein anfangen zu schwingen* (vgl. Abbildung 64). Es ist

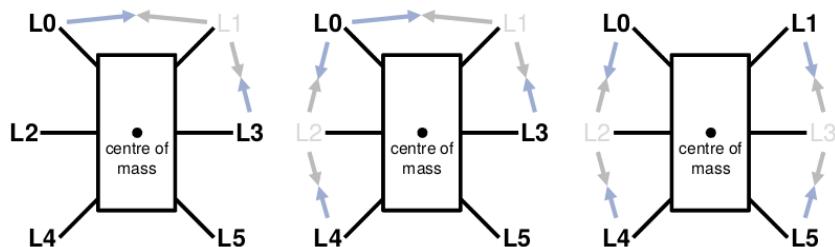


Abbildung 64: Gang der Stabheuschrecke. Die Konstellation im dritten Bild ist durch die Cruse-Regeln eigentlich nicht erlaubt.

ein Wechsel/Übergang zwischen verschiedenen Laufmustern möglich:

$$\text{Tripod: } \beta \approx 0.65, \text{ Tetrapod: } \beta \approx 0.80, \text{ Pentapod: } \beta \approx 0.9 \\ (\beta := \text{Schwing- zu Stemmzeit Verhältnis})$$

#### Quadruped/4-Beiner:

- Zahlreiche, sehr unterschiedliche Laufmuster

- Laufende Muster (Walking Gaits)
  - Mind. 1 Bein immer auf dem Boden
  - Walk, Passgang (Amble)
  - Trott - Walking
  - Creeping Gait (immer nur 1 Bein schwingt)
  - Crawl Gait (Creeping Gait mit max. Stabilität)
- Rennende Muster (Running Gaits)
  - Flugphase vorhanden (kein Bein auf Boden)
  - Passgang (Pace)
  - Leichter Galopp (Canter)
  - Galopp (Gallop)
  - Trab (Trot - Running)

#### Zweibeinige Laufmuster/Menschliches Laufen:

„Laufen“ (run)	„Gehen“ (walk)
<ul style="list-style-type: none"> <li>• Schneller als „Gehen“</li> <li>• Mit Flugphase</li> <li>• Duty-Factor <math>\leq 0.5</math></li> </ul>	<ul style="list-style-type: none"> <li>• Langsamer „Laufen“</li> <li>• Keine Flugphase</li> <li>• Duty-Factor <math>&gt; 0.5</math></li> </ul>

Tabelle 3: Wichtigste Unterscheidung: „Laufen“ vs. „Gehen“

## 6.4 Mehrbeiniges Laufen

### 6.4.1 Zentrale Muster Generatoren (CPG)

- Bio-inspirierte Mechanik und bio-inspirierte Steuerung (CPGs)
- Vier gekoppelte Oszillatoren erzeugen unterschiedliche Laufmuster (vgl. Abbildung 65)

### 6.4.2 Verhaltens- und regelbasierte Steuerung

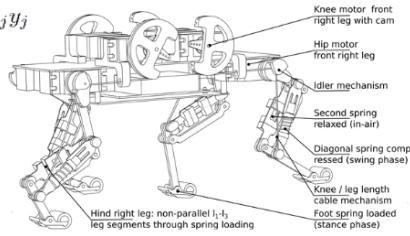
#### Beispiel: LAURON 4/5

- Modulare Architektur: Wart- und erweiterbar
- Verschiedene Laufmuster: Tripod, Tetrapod, Pentapod sowie Free-Gait
- Schrittgrößen und Zykluszeiten werden automatisch angepasst (z.B. bei Gang über Rampe)
- Hierarchische Verhaltsarchitektur
  - Koordinierte Verhalten (langsam)
  - Reaktive Verhalten (schnell)
  - Hardware Abstraction Layer (schnell)

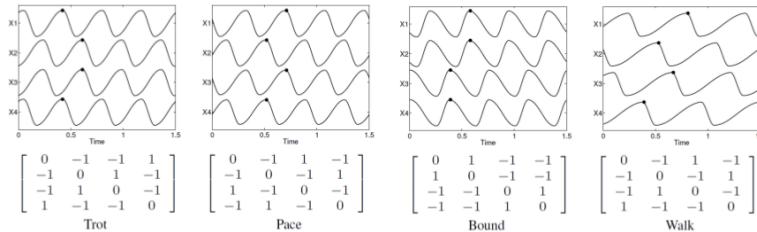
#### Passive Walking:

Rein mechanisches Laufen (kein Strom etc.) durch simple Regelungsalgorithmen und intelligente Mechanik (z.B. *Cornell Biped*)

$$\begin{aligned}\dot{x}_i &= \alpha(\mu - r_i^2)x_i - \omega_i y_i \\ \dot{y}_i &= \beta(\mu - r_i^2)y_i + \omega_i x_i + \sum k_{ij}y_j \\ \omega_i &= \frac{\omega_{stance}}{e^{-by} + 1} + \frac{\omega_{swing}}{e^{by} + 1}\end{aligned}$$



(a)



(b)

Abbildung 65: Beispiel CPGs

## 6.5 Zweibeiniges Laufen

### 6.5.1 Biomechanische Ansätze menschliches Laufen zu beschreiben

#### Selbststabilisierende Systeme beim Laufen

- Informationsfluss auf Nervenbahnen deutlich kleiner als Schallgeschwindigkeit  
⇒ Säugetiere tasten etwa mit  $50ms - 100ms$  ab.
- Trotzdem kann sich z.B. ein Gepard mit über  $80km/h$  bewegen, das bedeutet, dass er sich pro „Abtastschritt“ etwa  $1 - 2m$  bewegt.  
⇒ Informations-Vorverarbeitung in der „Mechanik“

Daraus lassen sich biomechanisch folgende Schlüsse ziehen:

- Bei schnellerer Lokomotion kann das Muskel-Skelett-System den Schwerpunkt nicht mehr kontinuierlich stabilisieren.
- Die „Bein-Mechanik“ muss so eingestellt werden, dass ein so genannter *Selbststabilisierender Zyklus* eingestellt wird.
- Das Nervensystem stellt die „Mechanik“ anhand eines *kinematischen Programms* immer wieder neu ein (Reflex-Preflex)

Beispiele:

- detailliertes neuronales Muskel-Skelett-Modell nach Hatze
- verallgemeinertes mechanisches Modell nach Günther
- „Virtual Model Control“ nach Henze bzw. Pratt
- Feder-Masse-System nach Blickhan
- Erweiterte Untersuchungen des Feder-Masse-Systems nach Seyfarth/Geyer/Blickhan

#### Bewegungsbereich und Selbststabilisation

Bewegungsbereich und Selbststabilisation sind beschreibende Attribute der energetischen Topographie (siehe Abbildungen 66 und 67).

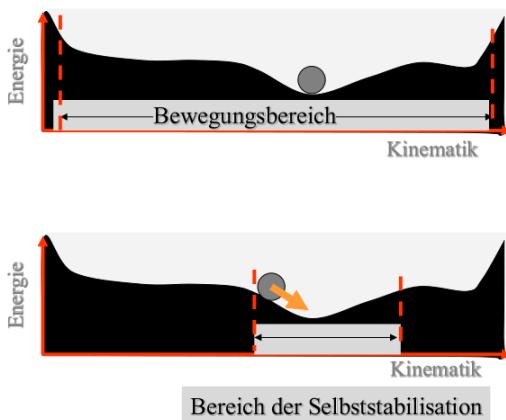


Abbildung 66: Bewegungsbereich und Selbststabilisation

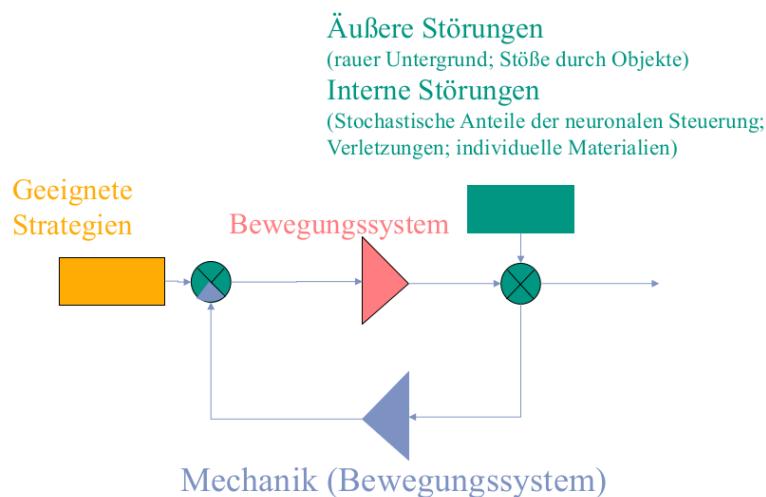


Abbildung 67: Selbststabilität

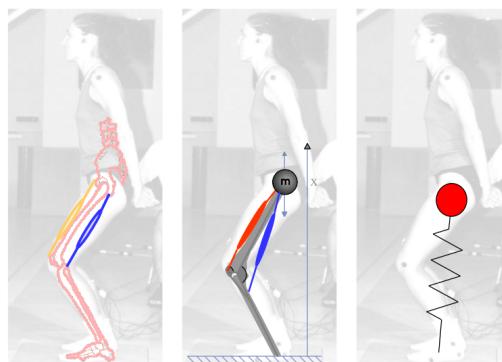


Abbildung 68: Starke Vereinfachung als Punktmasse an Feder

### Feder-Masse-System nach Blickhan

Modellierung selbststabilisierender Systeme:

- Modellierung menschlichen Laufens durch sehr einfache physikalische Modelle; Modellierung nähert das Feder-Dämpfer Verhalten der biologischen Muskeln an
- Abstraktion des gesamten muskulären Apparates auf eine Feder
- Schwerpunkt bewegt sich auf sinesoider Bahn. Bein „federt ein“ beim Auffußen und „federt aus“ beim Abfußen

Abbildung 69 zeigt die Modellierung, hierbei ist:

- $l_0$ : Initiale Beinlänge (Beim Auffußen und Abfußen)
- $\alpha_0$ : *Angle of Attack* (Aufsetzwinkel)
- $m$ : Systemmasse
- $k_{leg}$ : Federkonstante des Beines
- $y_i$ : Apexhöhe (max. Schulterhöhe)
- $v_{x,0}$ : Geschwindigkeit in  $x$ -Richtung

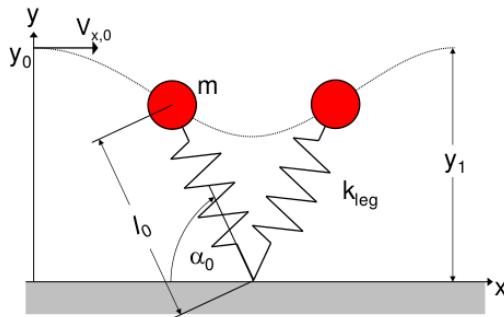


Abbildung 69: Modellierung Selbststabilisierender Systeme

Abbildung 70 zeigt ein attraktives Masse-Federsystem. Dieses Modell kann 1:1 auf den Menschen übertragen werden. In der rechten Abbildung sieht man den stabilen Zyklus. Hierbei ist der Aufsetzwinkel  $\alpha$  als aktiv zu regelnde Stellgröße besonders wichtig.  $69^\circ$  sind zu groß und der stabile Zyklus wird verlassen. Abbildung 71 zeigt den Ablauf. Hierbei ist:

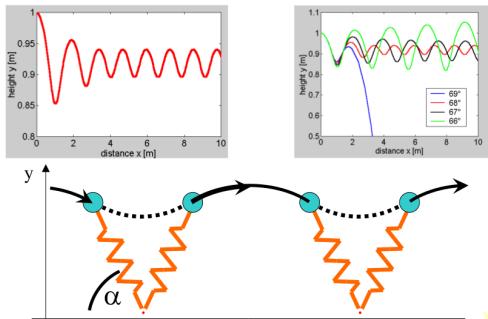


Abbildung 70: Attraktives Masse-Feder-System

- Konstante Geschwindigkeit  $\omega_R$
  - Start am Apex mit Angriffs-winkel  $\alpha_R$
  - Aufsetzwinkel  $\alpha_0$
  - Anpassen von  $\alpha$  als Funktion über die Flugzeit:  $\alpha = \alpha(t)$
  - Konstante Steifheit  $k$  im Bein
- ⇒ Verschiedene „Kinematische Programme“ im ZNS

### Rennen und Gehen

In Abbildung 72 ist elastisches Rennen dargestellt. Hierbei zeigt die linke Abbildung das Kraftprofil mit Laufrichtung als  $x$ -Komponente und Kraft als  $z$ -Komponente. Stabile Winkel sind  $67^\circ$  und  $68^\circ$ , nicht aber  $66^\circ$  und  $69^\circ$ . Bei letzteren kommt es zum Aufschwingen bzw. Abbruch.

Abbildung 73 zeigt Unterschiede zwischen Rennen und Gehen:

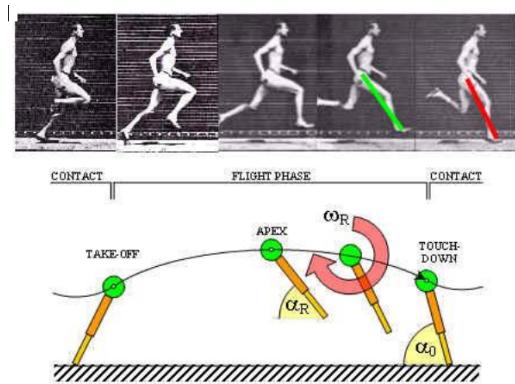


Abbildung 71: Selbststabilisierende Systeme – Ablauf

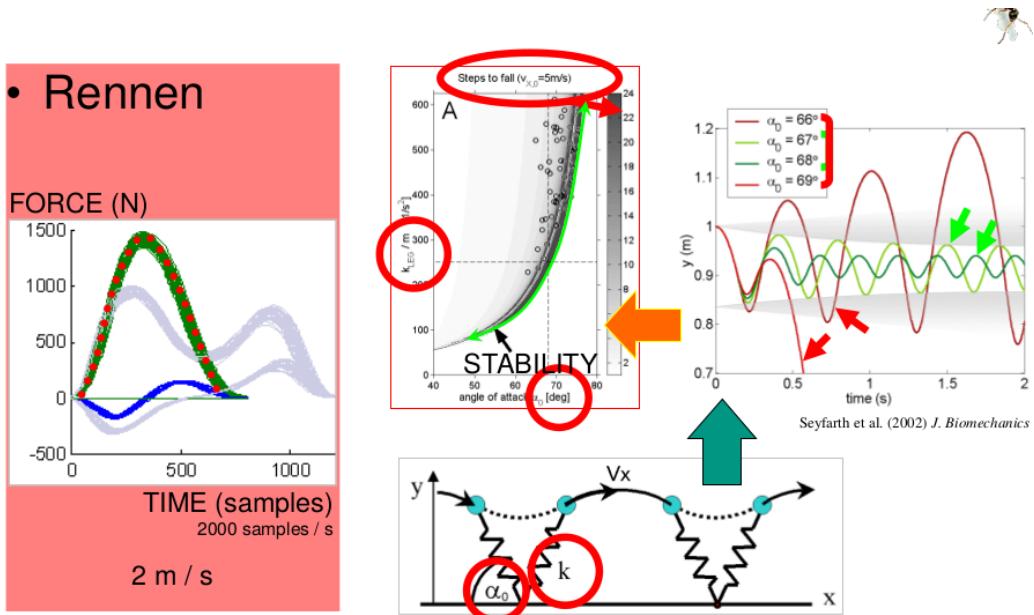


Abbildung 72: Elastisches Rennen

- Verhältnis Schwingen/Stemmen  $< 0.5 \leftrightarrow > 0.5$
- single-support  $\leftrightarrow$  double-support
- Einhügiges  $\leftrightarrow$  zweihügiges Kraftmuster
- Kurzer  $\leftrightarrow$  langer Bodenkontakt
- hüpfende  $\leftrightarrow$  glatte Bewegung
- Elastische  $\leftrightarrow$  steife Beine

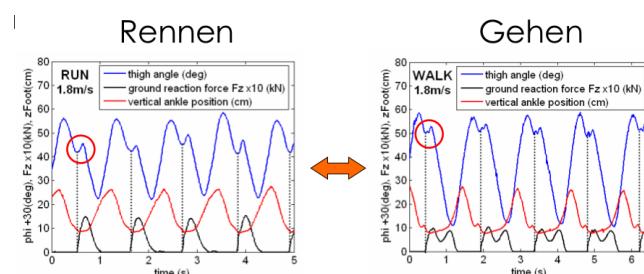


Abbildung 73: Rennen vs. Gehen

Abbildung 74 zeigt den Übergang vom Rennen zum Gehen. Auffällig ist die große Lücke zwischen Gehen und Rennen, es gibt keinen kontinuierlichen Übergang sondern der Geschwindigkeitswechsel verändert die Laufmuster grundlegend.

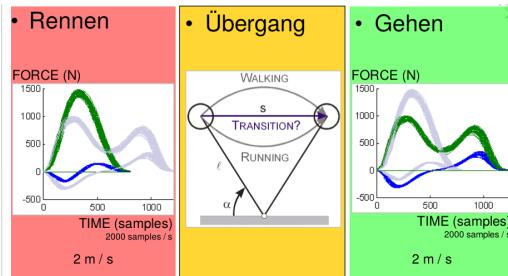


Abbildung 74: Übergang vom Rennen zum Gehen

Abbildungen 75 und 76 geben Informationen über elastisches Gehen.

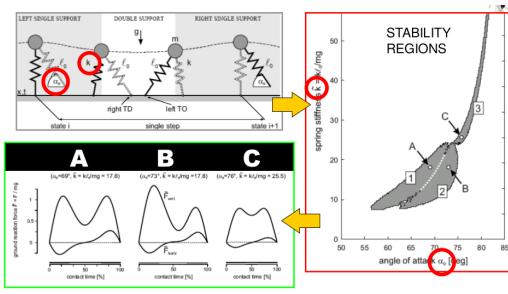


Abbildung 75: Ablauf des elastischen Gehens

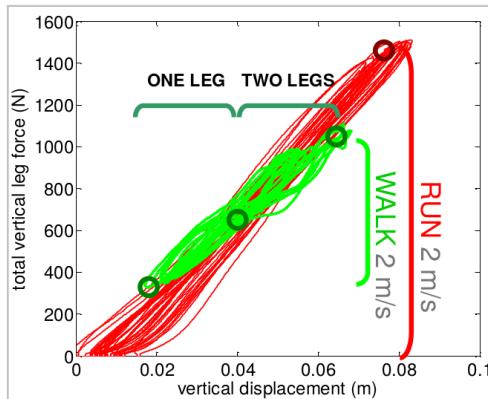


Abbildung 76: Elastizität beim Rennen und Gehen

### 6.5.2 Technische Umsetzung dieser Prinzipien auf reale Systeme

Abbildung 77 stellt die Schritte zur Technischen Umsetzung dar.

#### 1. Analyse, Übertragung und Adaption der biomechanischen Laufmodelle

Siehe Abbildung 78.

#### 3. Entwurf eines Algorithmus zur elastischen Beinlängen-Schwingungsregelung

Siehe Abbildung 79.

- Adaptives Einstellen der Eigenfrequenz über Systemgrößen
  - Kontrolliertes Anregen mit Resonanzfrequenz
- Optimale energieeffiziente Bewegung

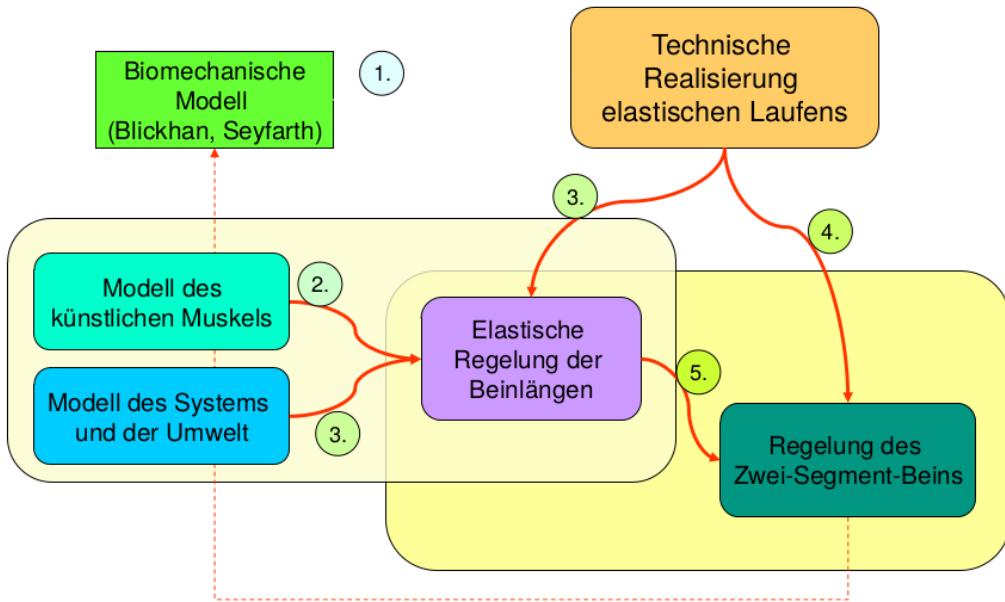


Abbildung 77: Vorgehen bei der technischen Umsetzung

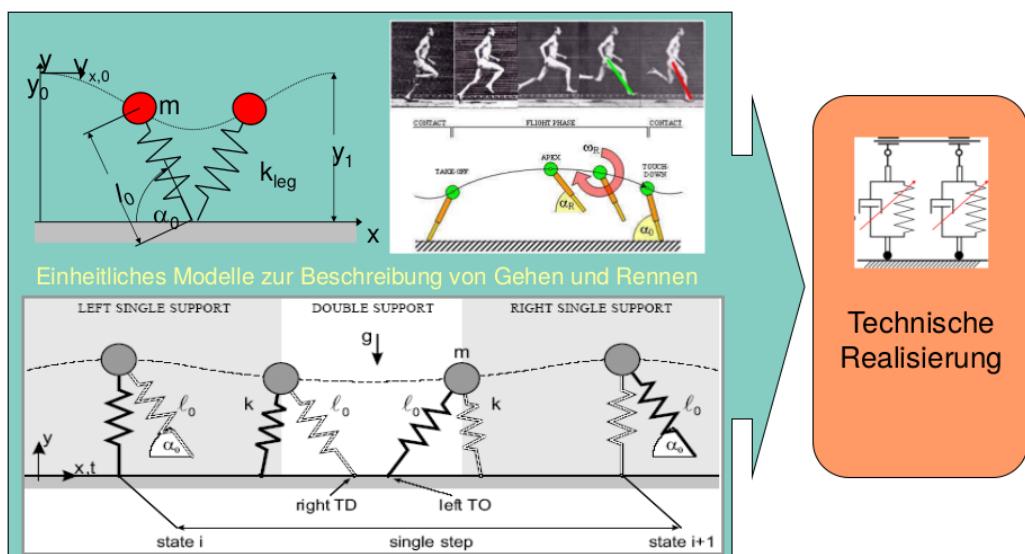


Abbildung 78: Schritt 1

→ Unterschiedliche Lauffrequenzen durch Modifikation von Systemgrößen

Ein-Dimensionaler Versuchsstand (EDV):

- Entwurf eines Versuchsstandes zur Entwicklung der Beinlängen Regelung
- Flexibler Aufbau - Tests mit unterschiedlichen Muskellängen, Gewichten, Untergründen
- Aufzeichnung aller relevanten Größen des Muskels und der Bewegung des Gesamtsystems

#### 4. Modellierung und Analyse der dynamischen Eigenschaften des gesamten Beins 5. Übertragung der elastischen Beinlängen-Regelung auf Mehrsegmentbein; Kopplung und Synchronisation des Hüftantriebs mit der Beinlängenschwingung

- 4 DOF pro Bein (2 DOF pro Hüfte) für erste Tests
- Auslegung für Geschwindigkeiten von bis zu  $3m/s$
- Integrierte Winkel-, Drehmomentsensorik

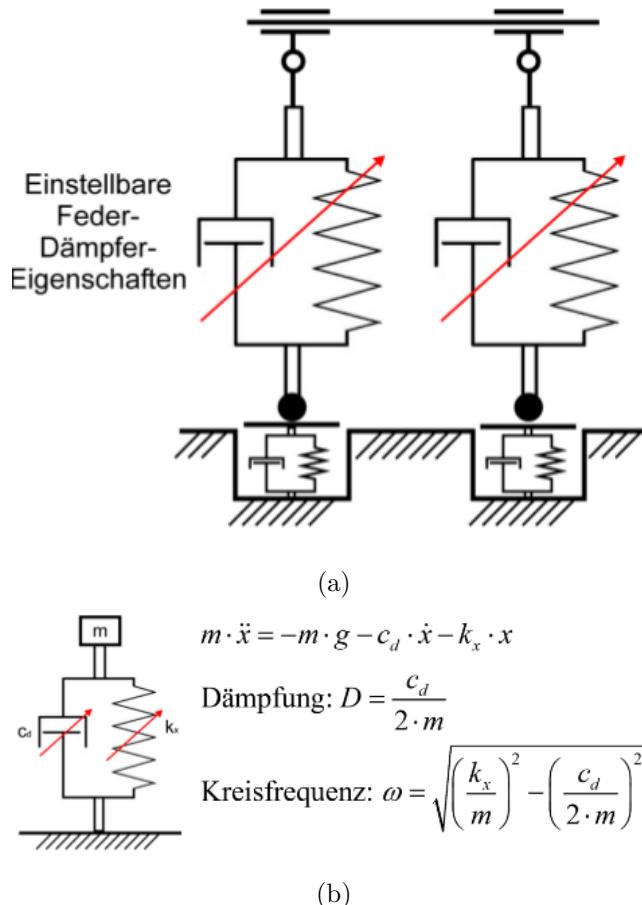


Abbildung 79: Schritt 3

- Beinlänge ca. 1m
- Gesamtgewicht Unterkörper ca. 20kg
- Winkelbereich Hüfte: Vorwärts 135°, Rückwärts 45°, Seitwärts je 30°

Regelungsansatz: Virtual Model Control (VMC) – nach Pratt '95

- Kontrolle durch „Virtual Components“ (VC)
  - Mit Hilfe virtueller externer Kräfte
  - Meist virtuelle Feder-Dämpfer-Systeme
  - Transformation auf Gelenkmomente mittels Jakobi Matrix
  - Keine inverse Kinematik
  - Keine Matrixinversion
  - Intuitive Kontrollmethode
  - Komplex wenn viele VCs von Nöten
- Kontrolle ähnelt Marionettenspiel
- Virtuelle Komponenten für Zweibeiniges System:
    - Ähnelt reaktivem Ansatz
    - Es werden 23 VCs benötigt
    - Abhängig von Ereignissen werden VCs aktiviert, deaktiviert oder neu parametrisiert
    - Drei Grundlegende Zustände der Muskelgetriebenen Gelenke: Aktiv, Passiv, Frei

## 7 Rechnerarchitektur

*The last Frontier of the biological science: Understanding the biological basis of consciousness!*

### 7.1 Neuronen, Synapsen und Nervenfasern

#### 7.1.1 Das Neuron als Basisrecheneinheit

- Zellkörper: enthält Zellkern und biochemischen Apparat (Versorgung der Zelle), von Zellhaut/Membran umgeben
- Dendriten: (altgr. Dendron = „Baum“) kurze, rohrförmige, stark verzweigte Fortsätze des Zellkörpers zum Empfang von Signalen
- **Nervenfaser:** langer, dünner Nervenfortsatz, meist nur eine Nervenfaser pro Nervenzelle (bis zu 1m lang sein beim Menschen), Weiterleiten der Signale
- **Myelin:** besteht aus fettartiger Substanz und Proteinen, dient zur Beschleunigung des Signal-Transports
- **Mitochondrien:** erzeugen die benötigte Energie
- **Endoplasmatische Retikulum:** synthetisiert die benötigten Eiweißstoffe

#### Rezeptoren und Reizverarbeitung

Siehe Abbildung 29 in Kapitel 4

#### Reizverarbeitung und Informationskodierung

Siehe Abbildung 30 in Kapitel 4

#### Signale auf Nervenfasern

Abbildung 80 zeigt die Zellmembran eines **Axons** (Nervenfaser). In der Fasermembran (1) befinden sich Ionenpumpen (2), welche für den aktiven Transport von Ionen entgegen ihrer elektrischen Gradienten zuständig sind. Des Weiteren enthält die Membran spezifische Ionenkanäle für Kalium ( $K^+$ )-Ionen (3) und Natrium ( $Na^+$ )-Ionen (4). Im Ruhezustand ist die Innenseite der Membran um 70mV negativer als die Außenseite. **Synapsen** sind Verbindungsstellen einer Nervenfaser (Sender) und Dendriten (Empfänger). Man unterscheidet erregende und hemmende Synapsen. Erstere depolarisieren die postsynaptische Membran (d.h. die Wahrscheinlichkeit für das Auslösen eines Aktionspotentials wird erhöht) und zweitere hyperpolarisieren sie.

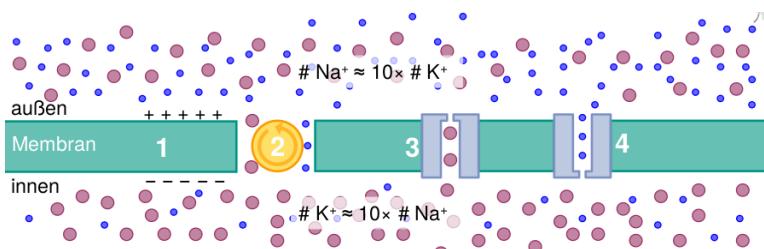


Abbildung 80: Signale auf Nervenfasern

dritten (Empfänger). Man unterscheidet erregende und hemmende Synapsen. Erstere depolarisieren die postsynaptische Membran (d.h. die Wahrscheinlichkeit für das Auslösen eines Aktionspotentials wird erhöht) und zweitere hyperpolarisieren sie.

### 7.2 Hierarchien der Informationsverarbeitung im Gehirn

#### 7.2.1 Das Zentrale Nervensystem

Siehe Abbildung 81.

#### 7.2.2 Aufteilung der Rindenfeldes

Siehe Abbildung 82.

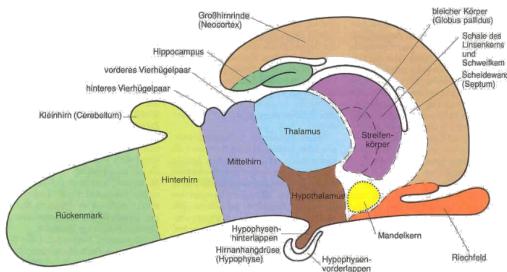


Abbildung 81: ZNS

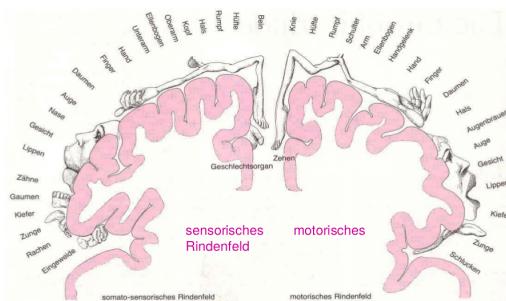


Abbildung 82: Rindenfeld

### 7.2.3 Aufteilung der Großhirnhälften

Siehe Abbildung 83.

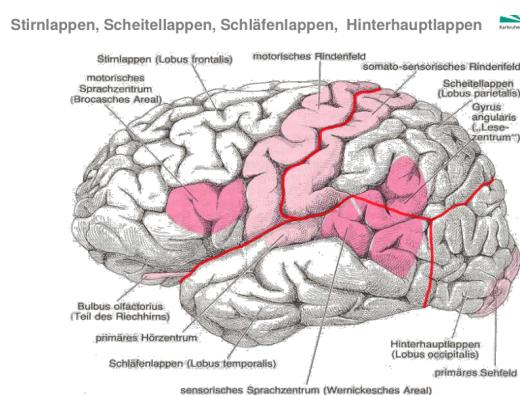


Abbildung 83: Grosshirn

## 7.2.4 Stabheuschrecke – Abstraktion des Steuerungskonzeptes

- Intelligente Einheiten
  - Definition der Einheiten über gefordertes Steuerungsverhalten
  - Adaptivität auf den Verhaltensebenen
  - Kommunikation der Einheiten auf einer Verhaltensebene
  - Hierarchische, verteilte Steuerung

## 7.3 Rechnerarchitektur für biologisch motivierte Roboter

### 7.3.1 Anforderungen an die Rechnerarchitektur

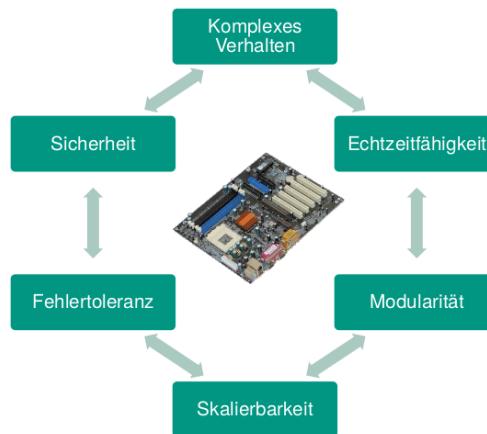


Abbildung 84: Anforderungen an die Rechnerarchitektur

#### Probleme bei der Umsetzung:

- Platzmangel
- Gewicht der Komponenten
- Energieversorgung
- Energieeffizienz
- Resistenz gegen elektromagnetische Störungen

#### Energieversorgung:

Energieversorgung im Falle von Rechnerbauteilen durch elektrische Energie  
→ Problem: elektrische Energie lässt sich schlecht kompakt speichern

- Batterien:
  - Schwer, groß
  - Kurze Laufzeit, lange Ladezeit
  - Verbesserung durch LiPo-Akkus, Grundproblem bleibt bestehen
- Brennstoffzellen:
  - werden zunehmend interessanter
  - noch kein breiter kommerzieller Einsatz
  - noch in der Erprobung

**Vergleich – Energiebedarf:** Siehe Tabelle 4.

### 7.3.2 Systemkomponenten – die einzelnen Bausteine

#### Mikrocontroller:

Ein Mikrocontroller besteht aus (vgl. Abbildung 85)

- Mikroprozessor
- Speicher

Herkömmlicher PC	Im Vergleich dazu
<ul style="list-style-type: none"> <li>• PC-System Intel Core i3, 3 GHz mit Monitor ca. 300 Watt</li> <li>• Pentium M ohne Monitor ca. 150 Watt</li> <li>• Intel Core i5-3450 (3,1 GHz) ca. 77 Watt</li> </ul>	<ul style="list-style-type: none"> <li>• Mikrocontroller ca. 0,4 Watt (C167 Infineon)</li> <li>• DSP ca. 0,55 Watt (DSP56F803 Motorola)</li> <li>• CPLD / FPGA ca. 0,7 Watt (Altera Flex 10k10)</li> <li>• PC/104-System (Pentium M @1 GHz) ca. 20 Watt (Digital-Logic)</li> <li>• PC/104-System (Core 2 Duo @ 2×1,5 GHz) ca. 25 Watt</li> <li>• Raspberry Pi (ARM @700 MHz) 3,5 Watt</li> <li>• BeagleBoard (ARM Cortex A8 @1 GHz) 6–10 Watt</li> </ul>

Tabelle 4: Vergleich – Energiebedarf

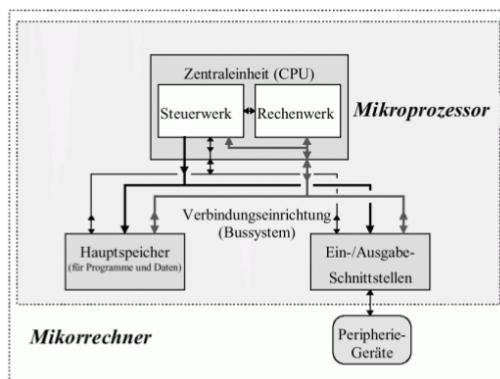


Abbildung 85: Komponenten eines Mikrocontrollers

- Ein-/Ausgabe-Schnittstellen
- Peripheriegeräten

### Digitale Signalprozessoren (DSP)

- Die Mikrorechnerarchitektur von DSPs ist auf die Verarbeitung digitaler Signale optimiert
- DSPs sind im Bereich der Arithmetik, speziell bei der Multiplikation, leistungsfähiger als Mikrocontroller
- Anwendungsbereiche für Signalprozessoren: Digitale Filter, Signalaufbereitung (Bsp. Sensordaten), Spracherkennung, Audio-/Videokompression

### Programmable Logic Device (PLD)

- Im Allgemeinen zweistufige Logik (Und/Oder-Gatter)
- Programmierbare Logik-Operationen
- Sehr schnell bei parallelen Prozessen

### Universal Controller Module (UCoM)

- Regelung von bis zu drei Motoren

- Auslesen von verschiedenen Sensoren
- Echtzeitfähig
- Größe:  $80\text{mm} \times 70\text{mm} \times 20\text{mm}$
- $80\text{MHz}$  DSP: Motorola
- Externes RAM
- FPGA: Altera
- Interfaces: CAN, SCI, SPI, JTAG
- DSP programmierbar über CAN-Bus
- Bis zu 3 Motoren (24 V) bis zu 5 A
- 6 Encoder-Ports (6-pin; 2 Spannungsversorgung, 4 IO)
- Differenzielle Motorstrommessung für jeden Motor

### Formfaktoren

- Netbook- und Multimedia-PC-Boom führt zur Verkleinerung der PC-Formfaktoren
- Früher:
  - Consumer: ATX ( $305 \times 244\text{mm}$ ), Micro-ATX ( $244 \times 244\text{mm}$ )
  - Roboter/Industrie: PC/104 ( $90 \times 96\text{mm}$ ) dafür aber höher!
- Aktuell:
  - Consumer: Mini-ITX ( $170 \times 170\text{mm}$ ) weit verbreitet
  - Roboter/Industrie: Pico-ITX ( $100 \times 72\text{mm}$ ), ETX ( $114 \times 95\text{mm}$ ), PC/104 ( $90 \times 96\text{mm}$ )

### PC/104

- Modulares Stecksystem
- PCI-Bus bei PC/104-Plus Systemen
- PCI/104-Express: x16 PCI-Express-Bus
- Basisplatine mit Prozessor (bis 2 GHz)
- Erweiterungsplatinen: Framegrabberkarten, CAN-Schnittstellenkarte, Karte für PCMCIA-Steckplatz (WLAN, Speicherkarten-Slots), ...

### Raspberry Pi 2

- Kompakte, kleine Größe (Kreditkarte)
- Prozessor: Quad-Core ARM Cortex-A7 mit 900 MHz
- RAM: 1024MB
- Schnittstellen: USB, LAN, HDMI, I2C, SPI, ...
- Erweiterungskarten: Kamera, WLAN, NFC, GPIO, ...
- Versorgung: 5V
- Kostenlose Betriebssysteme
- Preis: nur ca. 40 Euro

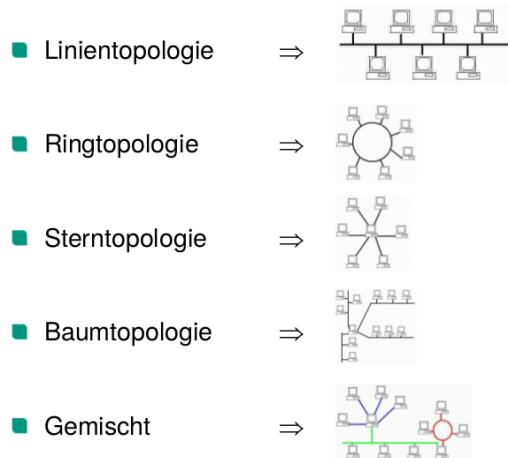


Abbildung 86: Topologien von Bus-Systemen

### 7.3.3 Kopplung – Bus-Systeme

**Topologien von Bus-Systemen:** Siehe Abbildung 86.

#### Übertragungsarten

- Elektrisch: Signale werden durch elektrische Impulse über Kabel übertragen
  - Vorteile: Einfache Verkabelung, flexible Leitungen
  - Nachteile: Anfälligkeit gegen elektromagnetische Störung
- Optisch: Signale werden über Lichtwellenleiter durch gepulstes Licht übertragen
  - Vorteile: Unempfindlichkeit gegenüber elektromagnetischen Störungen, geringes Gewicht der Lichtwellenleiter
  - Nachteile: Aufwendiges Konfektionieren und Verlegen der Lichtwellenleiter, begrenzter Biegeradius der Lichtwellenleiter

#### Beispiele für serielle Bus-Systeme:

- **CAN – Controller Area Network** (elektrisch):
  - Asynchrones, serielles Bussystem (1983 von Bosch entwickelt)
  - Vernetzung von Steuergeräten in Automobilen
  - Zweidrahtleitung → einfache Verkabelung
  - Linientopologie
  - Multimaster-Netzwerk mit Broadcastübertragung
  - Sehr robust, geringe Störanfälligkeit, daher auch weite Verbreitung in der Industrie
  - Echtzeitfähig – es kann eine maximale Latenzzeit angegeben werden
  - Bandbreite maximal  $1\text{ Mbit/s}$
  - Leitungslänge beträgt bei  $1\text{ Mbit/s} 40\text{ m}$  und bei  $125\text{ kbit/s} 500\text{ m}$
- **IEEE1394** oder **FireWire** oder **iLink** (elektrisch oder optisch)
  - serielles Bussystem (von Apple entwickelt)
  - asynchrone oder isochrone Datenübertragung: Paketlänge variabel mit Empfangsbestätigung; Broadcast-Pakete variabler Länge in festen Intervallen ohne Empfangsbestätigung)
  - Integrierte Spannungsversorgung ( $8V...33V$  bei max.  $1,5A$ )
  - Firewire 400 / IEEE1394a:  $100, 200$  oder  $400\text{ Mbit/s}$  Übertragungsbandbreit

- Firewire 800 / IEEE1394b:  $800\text{Mbit/s}$  Übertragungsbandbreite
- Multimedia-Anwendung (Video-Kamera)
- Shielded Twisted Pair → Komplexe Verkabelung: vier Adern für Daten, zwei für Versorg.-Spannung
- IEEE1394a: Beliebige Topologie außer Ring, IEEE1394b: auch Ringtopologie
- „Daisy Chain“ möglich

- **USB 1.1, 2.0, 3.0**

- Universal Serial Bus: serielles Bussystem (1996 von Intel entwickelt)
- Zweidrahtleitung → Einfache Verkabelung
- Zusätzliche 5V Versorgungsspannung (max.  $500mA$ )
- USB-Host für Kommunikation erforderlich
- Baumtopologie
- Robust und weit verbreitet (Festplatten, Tastatur, Maus, ...)
- NICHT echtzeitfähig
- USB 1.1:  $12\text{MBit/s}$ , USB 2.0:  $480\text{MBit/s}$ , USB 3.0:  $5\text{Gbit/s}$
- Maximale Leitungslänge  $5m$  und mit Repeatern max.  $25m$

- **Thunderbolt**

- parallele Kanälen zur seriellen Datenübertragung (2011 von Intel und Apple entwickelt)
- Aktive Kabel mit integrierter Elektronik
- Elektrische wie später optische Übertragung geplant
- Unterstützung mehrerer Busprotokolle: PCI Express, DisplayPort
- Direkte Konkurrenz zu USB 3.0
- Maximal 7 Busteilnehmer in Daisy chain verbindbar
- Erste Kabel-Version mit 20 Pins: Zwei Kanäle je  $10\text{GBit/s}$
- Max. Leitungslänge  $3m$  elektrisch, mind.  $10m$  optisch möglich

- Industrial Ethernet (elektrisch)
- Profibus (elektrisch)
- FlexRay (elektrisch)
- SERCOS – Serial Realtime Communication System (optisch)

## 7.4 Beispielarchitekturen

*It's alive! It's alive!!!*

### 7.4.1 Rechnerarchitektur von LAURON IVc

Siehe Abbildung 87.

- Autonom
- Steuerung auf LAURON
- Erfassung und Verarbeitung der Sensorwerte
- 7 UCoMs

- 2×PC/104 System
- CAN-Bus
- MCA2

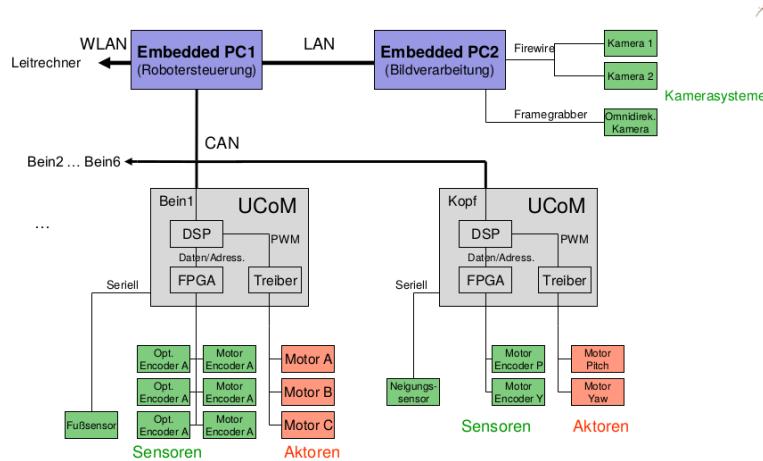


Abbildung 87: Rechnerarchitektur auf LAURON IVC

#### 7.4.2 Rechnerarchitektur von LAURON V

Siehe Abbildung 88.

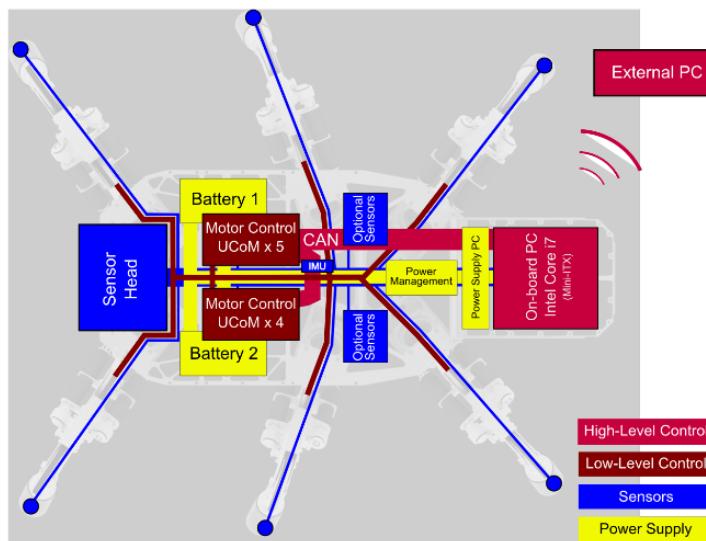


Abbildung 88: HW-Architektur auf LAURON V

#### 7.4.3 Rechnerarchitektur von KAIRO 3

Siehe Abbildung 89.

- Modularer schlängelartiger Roboter
- Inspektion von unzugänglichen Bereichen
- Veränderliche Konfiguration
- 3 bis 11 UCoMs (je nach Konfiguration)

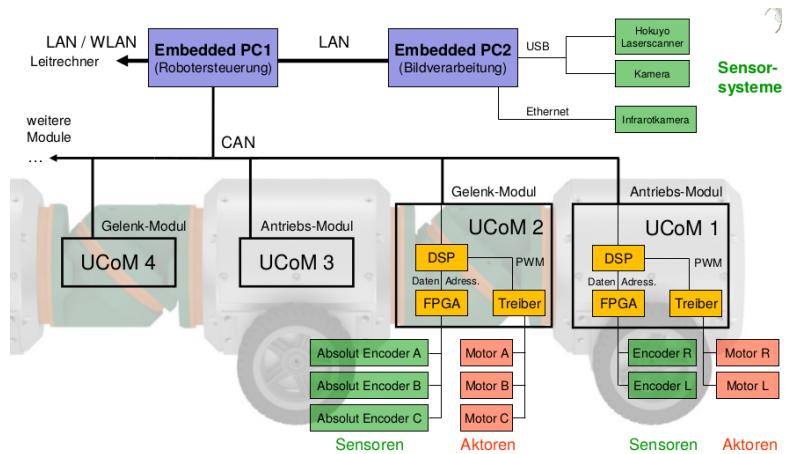


Abbildung 89: Verteilte SW/HW-Rechnerarchitektur auf KAIRO 3

- Bussysteme: 2×CAN, I2C, Ethernet
- Embedded PC: ARM Prozessor-Board
- MCA2: adaptive Steuerung

#### Eigenschaften der SW/HW-Architektur:

- Modularität
- Flexibilität
- Effizienz
- Ressourcen
- Re-Konfigurierbarkeit

#### 7.5 Gegenüberstellung Rechnerarchitektur in Natur und Technik

Siehe Abbildung 90.

	Natur	Technik
Parallelität	sehr hoch	niedrig–mittel
Redundanz	sehr hoch	niedrig
Fehlertoleranz	sehr hoch	niedrig
Rekonstruktion	möglich	nicht möglich
Verarbeitungsgeschwindigkeit	niedrig	hoch
Übertragungsgeschwindigkeit	niedrig	hoch
Schnittstellen	hoch	niedrig

Abbildung 90

## 8 Verhalten

Planen bedeutet zu Vermeiden herauszufinden, was man als nächstes zu tun hat

## 8.1 Verhalten im Tierreich: Neurologisch, Psychologisch, Ethologisch

*It is said that the limbic system of the brain controls the four Fs: Feeding, Fighting, Fleeing and Reproduction*

### 8.1.1 Neurologisch – Vektorfeldaddition im Frosch-Rückenmark

- Neurobiologische Hypothese der motorischen Kontrolle durch Vektorfelder
- Bizzi (MIT) hat gezeigt, dass Gliedmaßenbewegung in Fröschen im Rückenmark codiert abgelegt sind
- Mikrostimulation im Rückenmark generiert eine Bewegung zu einem spezifischen Punkt
- Planung im ZNS wird auf solche Reize umgesetzt

### 8.1.2 Psychologisch – Verhaltens-, Gestalt- und Kognitive Psychologie

- **Verhaltens-Psychologie:** Verhalten wird durch Beobachtung definiert. Alles wird auf ein Stimulus-Antwort Schema zurückgeführt.
- **Gestalt-Psychologie:** Einbeziehung der Physik. Verhalten entsteht als direkte Konsequenz aus der Struktur der physischen Umgebung.
- **Kognitive Psychologie:** Einbeziehung des Wissens. Vereinheitlichte Methode, um die Beziehung von Aktion und Perzeption zu erklären.

### 8.1.3 Ethologisch – Reflexe, Taxe und feste Muster

- Ethologie bezeichnet die Beobachtung von tierischem Verhalten in seinem natürlichen Lebensraum.
- Das Tier wird als Teil des Gesamtsystems angesehen, das auch die Umgebung mit einbeziehen muss.
- Klassifizierung in **Reflexe, Taxe und Feste Muster**.
- Motivierte Verhalten als Antwort auf interne Stimuli.
- Schematheorie (Lorenz): Zusammenfassung von komplizierten Kombinationen aus Reflexen, Taxen und Mustern zu Schemata.
- Prinzip der „Ökologische Nische“ (McFarland). Ein Tier überlebt, weil es eine Nische gefunden hat.

Drei Klassen von Verhalten im Tierreich:

- **Reflexe:** schnelle, automatische und unbewusste Reaktionen, die von einem sensorischen Stimulus ausgelöst werden.
- **Taxe:** Verhalten, die ein Tier in Richtung (attractive) oder entgegen (aversive) eines Stimulus orientieren.
- **Feste Muster:** Reaktionen auf Stimuli, die länger anhalten als die Stimuli selbst.

## 8.2 Verhaltensbasierte Robotik – Übersicht und Einführung

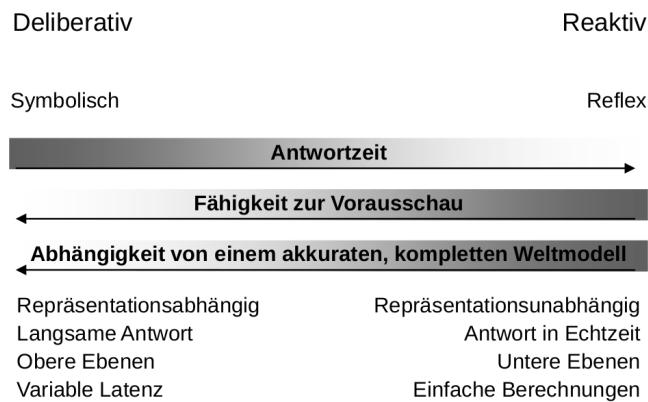
*Es gibt zwei Möglichkeiten beim Entwurf von Robotersteuerungen: Erstens, so simpel, dass es keine offensichtlichen Mängel gibt; Zweitens, so komplex, dass es keine offensichtlichen Mängel gibt.*

### 8.2.1 Motivation – Warum verhaltensbasierte Architekturen

- Komplexe Verhalten entstehen nicht notwendigerweise aus komplexen Steuerungssystemen
- Die Natur ist das beste Vorbild
- Simplizität ist eine Tugend
- Robustheit bei verrauschten Sensorwerten ist ein Designziel
- Systeme sollten inkrementell aufgebaut werden können
- Alle Berechnungen Onboard
- Roboter sollten billig sein

### 8.2.2 Roboter Steuerungs-Spektrum – Von reaktiv bis deliberativ

Roboter Steuerungs-Spektrum:



Hierarchische Steuerung:

- Hierarchische Anordnung von Steuerungselementen, wobei höhere Ebenen Unterziele für niedrigere erzeugen
- Eigenschaften
  - Es wird vorausgesetzt, dass die zu erfüllende Aufgabe zerlegbar ist
  - Ein Weltmodell auf jedem Level garantiert die korrekte Ausführung

Reaktive Steuerung:

- Bedeutet vereinfacht eine enge Kopplung zwischen Perzeption und Aktion
- Definierte Schlüsselemente:
  - **Embodiment:** Der Roboter hat eine physikalische Präsenz
  - **Situatedness:** Der Roboter ist als Einheit in seiner Umgebung eingebunden
  - **Emergence:** Intelligenz entsteht durch Interaktion mit der Umwelt

### 8.2.3 Die Anfänge

- Walters Machina Speculatrix 1953
- Braitenberg Vehicles 1984

## 8.3 Beispiele für verhaltensbasierte Architekturen

### 8.3.1 Anforderung an die Architektur / Bewertungsschema

- Unterstützung der **Parallelisierung**: Verhalten auf verschiedenen Prozessoren berechnen
- Abstimmung auf die Hardware / **Hardwareabhängigkeit**: Verhalten speziell auf Hardware abgestimmt
- **Modularität**: Einfache klare Module, einfache System-Erweiterbarkeit
- **Robustheit**: Gesamtsystem sollte stets stabil bleiben
- Fortschritt der **Umsetzung**: Gibt es eine funktionierende Implementierung?
- **Flexibilität zur Laufzeit**: Kann sich das System anpassen?
- **Effektivität** in Hinsicht der Erfüllung der Aufgabe: System Overhead?

### 8.3.2 Subsumption (MIT)

- Anordnung der Verhalten auf horizontalen Schichten
- Alle Verhalten greifen auf alle Sensordaten zu und generieren Aktionen für alle Aktoren
- Interaktion der einzelnen Verhalten durch Inhibitieren von Eingängen und Überstimmen von Ausgängen
- Rückkopplung primär über die Umgebung
- Eigenes C Derivat: Interactive C

Siehe Abbildung 91.

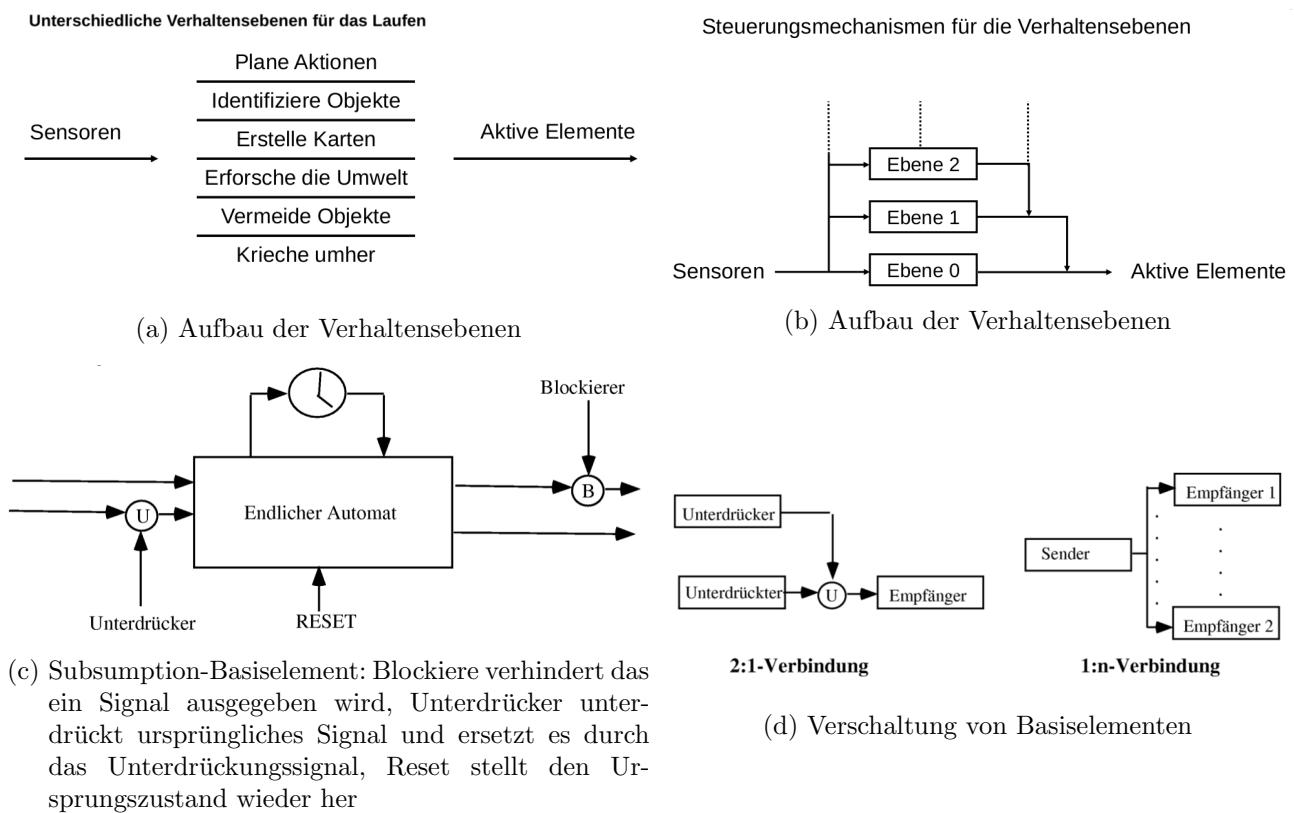


Abbildung 91: Subsumptionsarchitektur

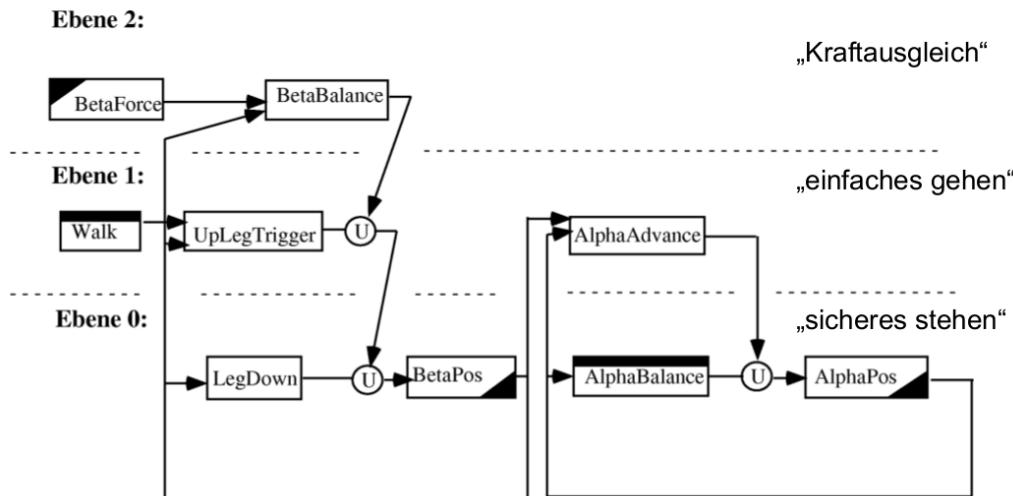


Abbildung 92: Subsumption: Beispiel der MIT Laufroboter Ghengis/Attila – Ebenen

Ein Beispiel für diese Architektur zeigt Abbildung 92.

#### Subsumption Bewertung

- Parallelisieren möglich
- Hardwareabhängig
- Modular erweiterbare Struktur teilweise vorhanden
- Robustheit von Modulen abhängig
- Prototypisch umgesetztes System
- Keine Flexibilität zur Laufzeit
- Teilweise effektive Erfüllung der Aufgabe

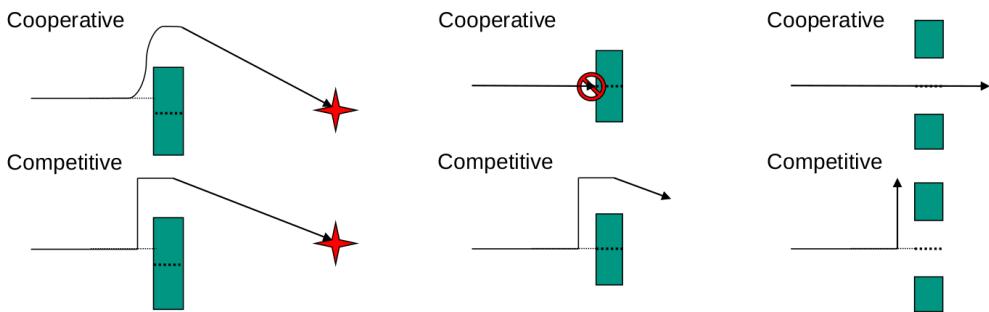
#### 8.3.3 Verhaltensnetzwerke/ Verhaltenssteuerung

##### Vorteile

- Flexibel, adaptiv
- Robust, Ausfall eines Verhaltens führt nicht zu Totalausfall
- Können auf unvorhergesehene Situationen besser reagieren
- Asynchron, jedes Verhalten kann eigene Zykluszeiten verwenden
- Viele kleine Module, die separat testbar sind
- Unterstützt Parallelisierung

##### Nachteile

- Verhalten des gesamten Systems schwer zu überschauen
  - Viel Aufwand durch Tuning des Systems, speziell der Fusion
- schwer zu debuggen, jedes einzelne Verhalten ja, aber wenn zu großem Gesamtverhalten zusammen geschaltet **unvorhersehbar!!!**



(a) Beispiel: Drei Verhalten: weiche nach links/rechts aus, fahre gerade zum Ziel

	Cooperative	Competitive
Priority-based		X
Activity-based	X	X
Fuzzy-based	X	X
Election-based	X	
State-based		X

(b) Unterschiedliche Methoden zur Fusion

Abbildung 93: Verhaltensbasierte Steuerungen: Ausgabe-Fusion

Die Vor- und Nachteile hängen von konkreten Architektur-Ausprägungen ab. Mögliche Varianten und deren Charakteristika sind:

- Fusionsprinzip/Coordinator: von Kooperativ bis Konkurrierend
- Abstaktionsniveau/Granularity: von Reaktiv bis Deliberativ
- Datenformat/Action-encoding: von Diskret bis Kontinuierlich

#### Ausgabe-Fusion:

Die Fusion oder Koordination der Ausgabe der einzelnen Verhalten:

- **Cooperative**
  - Alle Verhalten bestimmen zusammen eine Aktion oder
  - Alle Verhalten bestimmen die Stärke vieler Aktionen
  - ⇒ Sprungfreie Aktionen
- **Competitive**
  - Ein Verhalten bestimmt die zu ausführende Aktion
  - ⇒ Einfacher zu überschauen und zu testen

Problem: beste Wahl hängt stark von Situation und konkreten Verhalten ab. Dies ist in Abbildung 93a an einem Beispiel verdeutlicht. Abbildung 93b stellt die unterschiedlichen Methoden zur Fusion gegenüber.

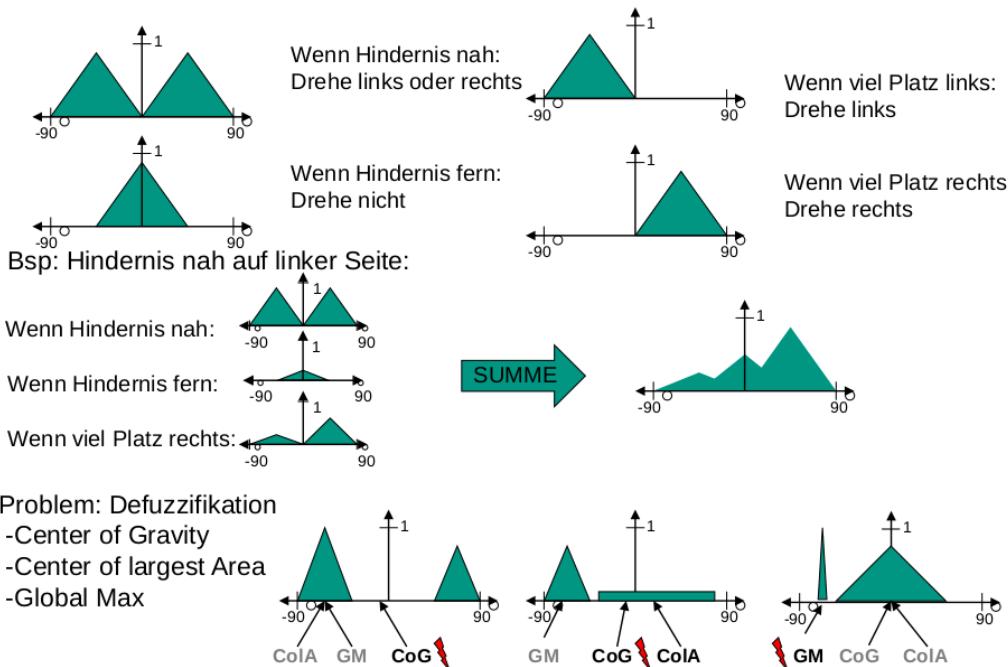
#### Priorität-basierte Fusion

- Jedem Verhalten wird eine Priorität zugewiesen
- Prioritäten sind (meist) dynamisch
- Zuweisung erfolgt durch höhere Verhalten oder Arbiter, Automaten, Planer, etc.
- Konkurrierend: Verhalten mit höchster Priorität bestimmt Aktion

### Aktivität-basierte Fusion

- Jedes Verhalten bestimmt selbst seine Aktivität
  - Normierte Stellgröße
  - Normierte Soll-Ist Differenz ...
- Entsprechend der Aktivität wird gewählt
  - Konkurrierend: Das Verhalten mit der höchsten Aktivität wählt eine Aktion
  - Kooperative: Jede Aktion wird mit der Summe der Aktivitäten gewichtet ausgeführt

### Fuzzy-basierte Fusion:



### Wahl-basierte Fusion

- Endliche Menge definierter Aktionen
- Jedes Verhalten stimmt für eine Aktion
  - Priority Based: Jedes Verhalten hat ein Stimmrecht gleich seiner Priorität
  - Activity Based: Jedes Verhalten hat ein Stimmrecht gleich seiner Aktivität
- Entsprechend der Stimmen wird ausgewählt
  - Die Aktivität mit den meisten Stimmen wird ausgeführt
  - Alle Aktivitäten werden mit ihren Stimmen gewichtet ausgeführt
- Variante: Fusion besitzt ein Gedächtnis. Verhalten kann sowohl für als auch gegen eine Aktion stimmen. Die Stimmen werden über die Zyklen akkumuliert.  
Vor- und Nachteil: Schnelle Aktionswechsel können verhindert werden.

### Status-basierte Fusion

- Ein endlicher Automat kontrolliert die Verhalten
- Je nach Zustand bestimmt der Automat, welches Verhalten die Aktion auswählen kann
- Wird verwendet um Verhalten Prioritäten zuweisen zu können
- Wird verwendet um zwischen verschiedenen Fusionsmethoden zu wechseln

### 8.3.4 Verhaltensnetzwerke (FZI)

- Einheitliche Schnittstelle der Reflexeinheiten, bestehend aus Sensor- und Steuerungsdaten sowie Aktivierung, Aktivität und Bewertung
- Modularer Aufbau
- Hierarchische Anordnung innerhalb der Architektur abhängig von der Roboter-Kinematik
- Trennung von Daten- und Kontrollfluss
- Implementierung in MCA2

**BISAM:**

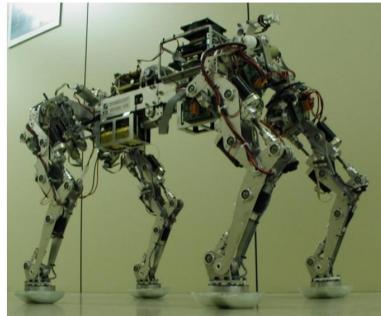


Abbildung 94 veranschaulicht BISAMs Verhalten. Zu Abbildung 94d:

- Verteilung von reaktiv nach deliberativ
- Zustandsbewertung und Aktivität  $a$  als virtuelle Sensoren
- Motivation  $t$  als virtueller Aktor
- Regionen durch virtuelle Akteure
- Stress und Angst einer Region

$$Angst_T(A) = \int_{t=t_n-T}^{t_n} \left( \sum_{A_t \in R(A)} r_i(t) \right) dt \quad (2)$$

$$Stress_T(a) = \int_{t=t_n-T}^{t_n} \left( \sum_{A_t \in R(A)} a_i(t) \right) dt \quad (3)$$

**Verhalten – Entwurfsverfahren:**

1. Definition der Verhalten und Reflexe
2. Finden von kinematischen Gruppen
3. Verteilen der Verhalten und Reflexe auf die Gruppen
4. Top-Down Sortierung deliberativ nach reaktiv (Regionen)
5. Bottom-Up Implementierung und schrittweise Validierung

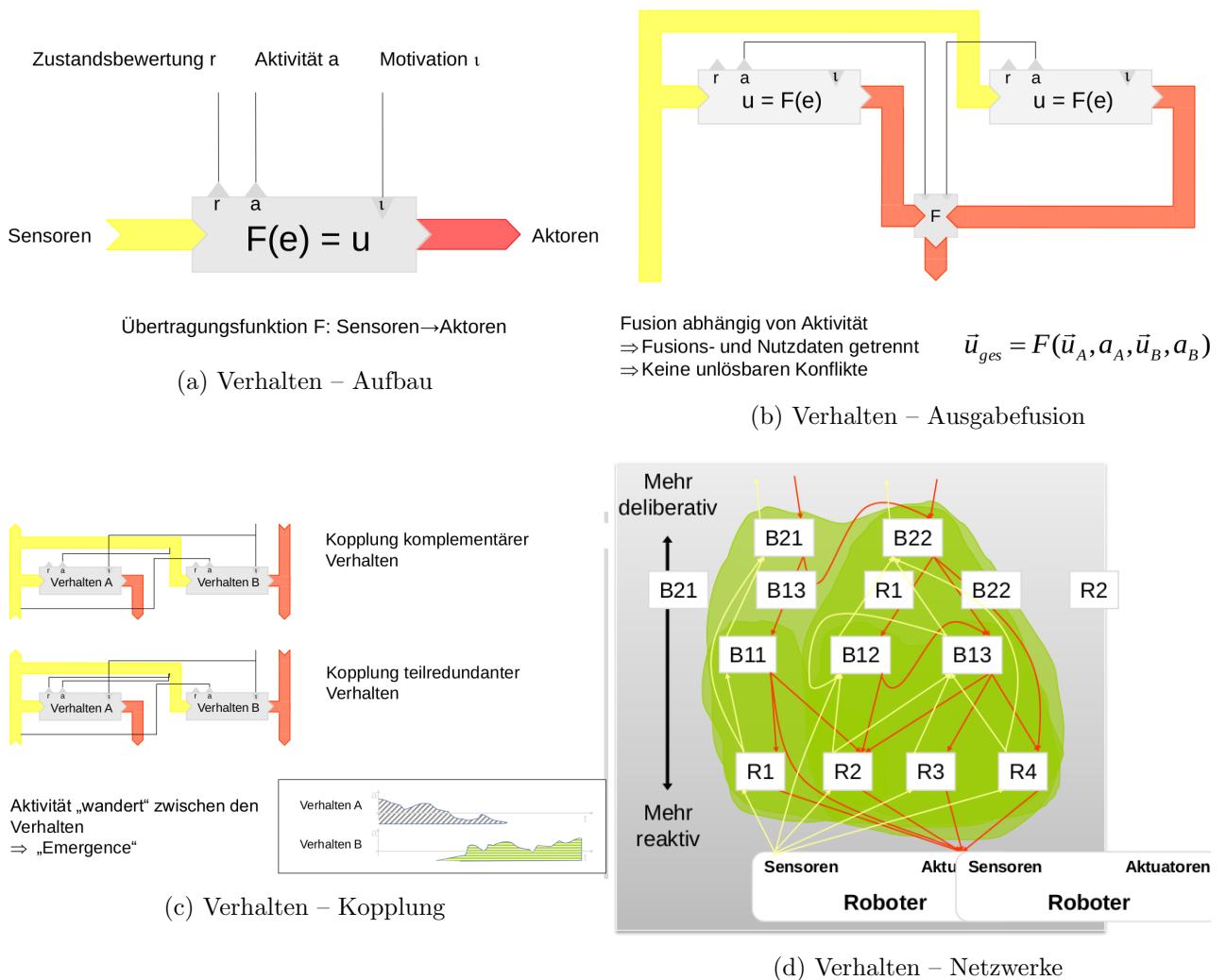


Abbildung 94

**Statisch stabiles Laufen:**

- Variabler Untergrund
- Adaption an Hindernisse
- Entlastung des nächsten Schwungbeines
- Stabilität während des Schwungzyklus

**BISAM Bewertung:**

- Parallelisieren möglich
- Hardwareunabhängig
- Modulare, erweiterbare Strukturen sind vorhanden
- Robustheit ist abhängig von Einzelverhalten
- System ist prototypisch umgesetzt
- Beschränkte Flexibilität der Laufzeit
- Effektive Erfüllung der Aufgaben

**InBOT** (selbstfahrender Einkaufswagen, der mit Menschen interagiert) als Beispiel für verhaltensbasierte Steuerung für Fahrzeuge: Hierarchisches Modell von Fußgängerbewegungen:

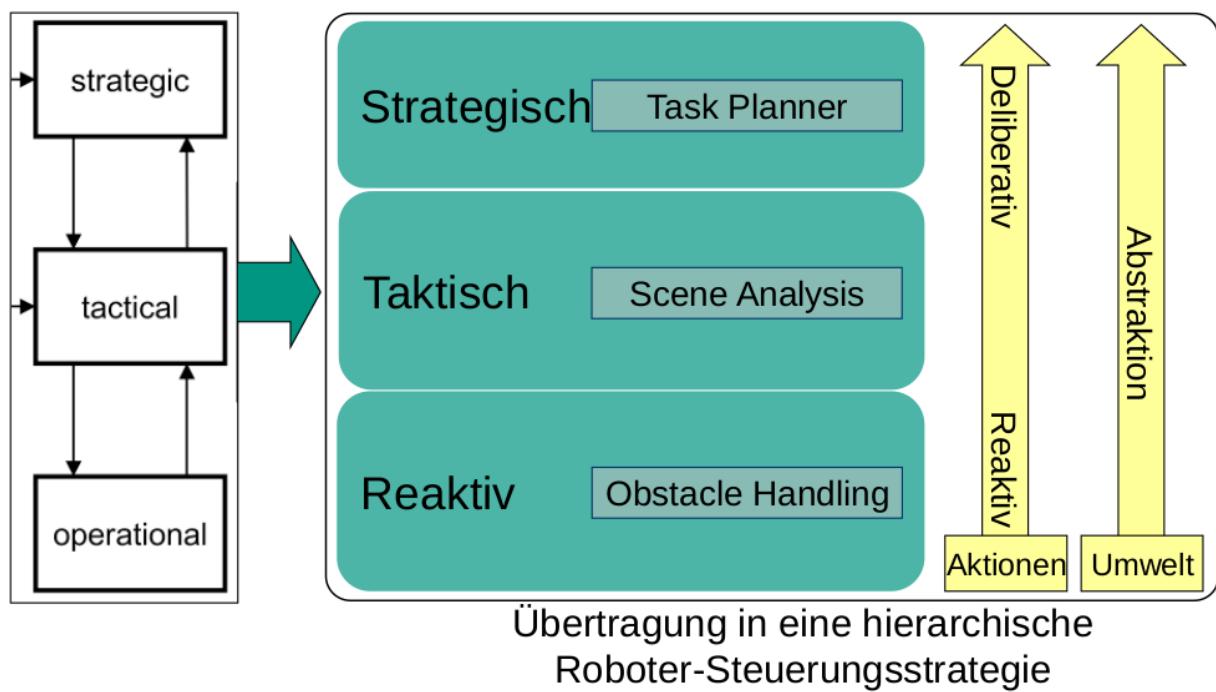


Abbildung 95: Verhaltensbasierte Steuerung für Fahrzeuge Beispiel InBOT

**• Strategische Verhalten:**

- Grobe Routenplanung
- Basiert auf Vorlieben
- Erzeugt Knotenpunkte (KP)

- Taktische Verhalten
  - Erzeugt Unter-KP
  - Geometrie der Umgebung
- Ausführende Verhalten
  - Bewegungserzeugung
  - Pfad wird an Hindernisse und Menschen angepasst

Vorteile:

- Nutzen erfolgreicher Navigationsstrategien
- Geringer A-Priori Informationsbedarf (keine detaillierte Karten)

Nachteile:

- Das Verhalten des Roboters ist unscharf
- D.h. schwieriger zu kontrollieren

**Welt Modell** – Drei Ebenen:

- Topologische Karte
  - Invariant
  - Semantische Erweiterung
- Objekt Datenbank
  - Segmentierte Hindernisse
  - Tracking Informationen
- Occupancy Maps
  - Basislevel
  - Eignet sich gut für Datenfusion
  - Diskretes Welt Modell

**Aufgaben-Orientiertes Level:** Vgl. Abbildung 96.

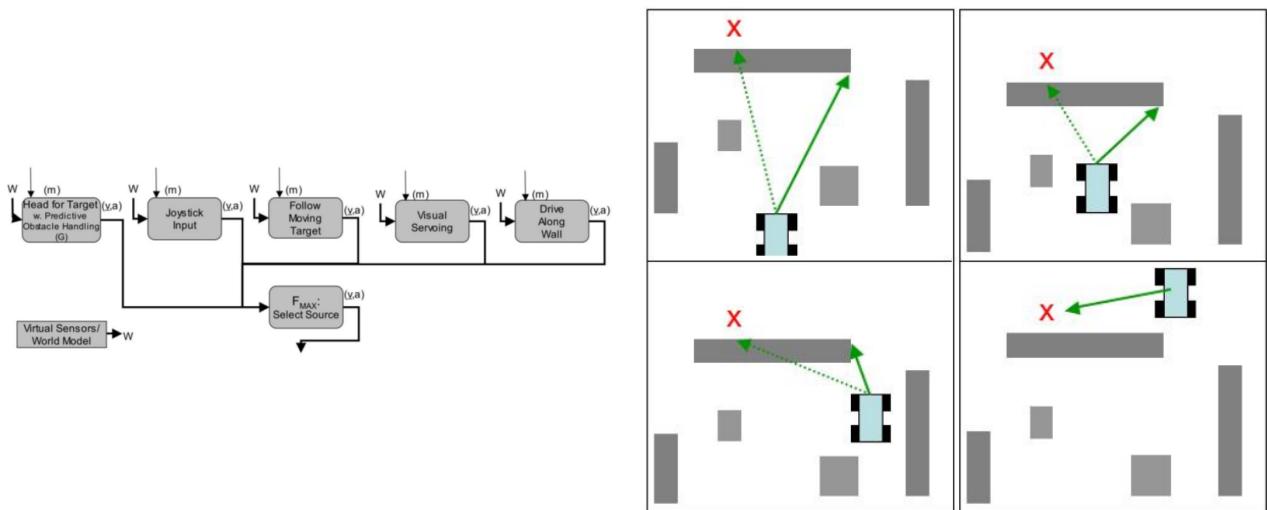


Abbildung 96: Aufgaben-Orientiertes Level

- Fusion der Task-Orientierten Verhalten
- Auswahl eines Verhaltens durch eine Maximum Fusion (konkurrierend)
- Vorausschauende Hindernis-Behandlung
- Sub-Ziele werden generiert, wenn der direkte Weg zum Ziel blockiert ist
- Ecken der Hindernisse dienen als Sub-Ziele
- Der Roboter bewegt sich ähnlich zu einem Sichtgraph
- Alle sichtbaren Eckpunkte werden nach ihrer Qualität gewichtet
- Ecke mit bester Qualität wird ausgewählt
- Bei Sackgassen/Lokalen Minima:
  - Günstigste Ecke wird als Sub-Ziel gewählt
  - Befindet sich der Roboter bereits in der Sackgasse ist das Sub-Ziel hinter ihm
  - ⇒ Der Roboter kann um Sackgassen herumfahren oder aus ihnen entkommen
  - ⇒ Klassisches Problem der Lokalen Minima der Potentialfeldermethoden wird vermieden

**Objekt-Orientiertes Level:** Vgl. Abbildung 97.

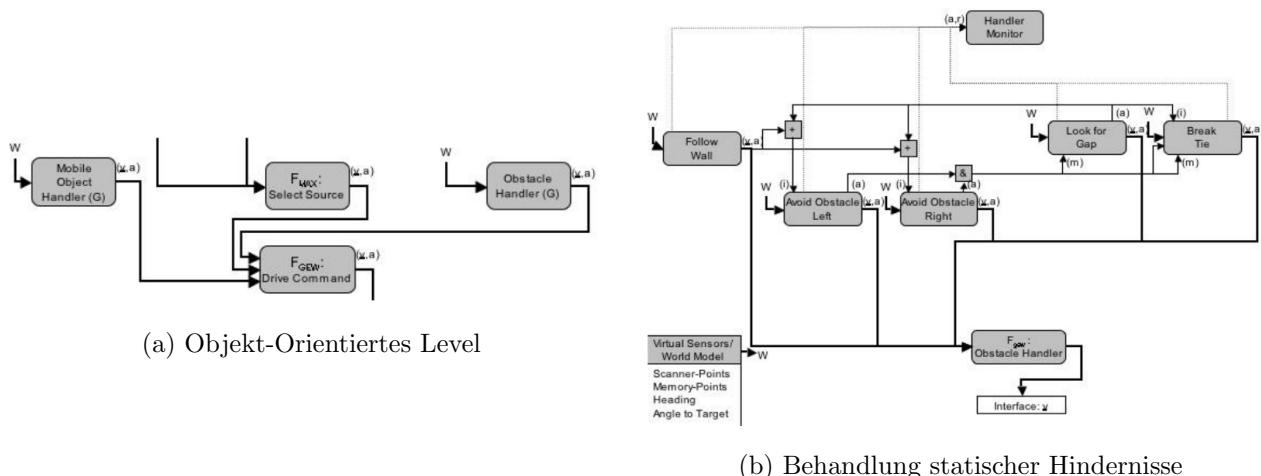


Abbildung 97

- Fusion der Aktion aus dem Aufgaben-Orientierten Level mit den Ausgaben der Verhaltensgruppen zur Hindernisbehandlung
- Kooperative Fusion (alle drei Aktionen werden gewichtet und gleichzeitig ausgeführt)

Objekt-Orientiertes Level: Avoid Obstacles (vgl. Abbildung 98).

- Dynamische Vektorfeld Methode
- Erzeugt Vektor, der vom Hindernis-Schwerpunkt weg zeigt
- Jeweils ein abstoßender Vektor für Links- und Rechtsseitige Hindernisse
- Fusion (kooperativ) dieser beiden Vektoren mit einem anziehenden Vektor Richtung Ziel

Bewertung InBOT:

- Parallelisieren möglich

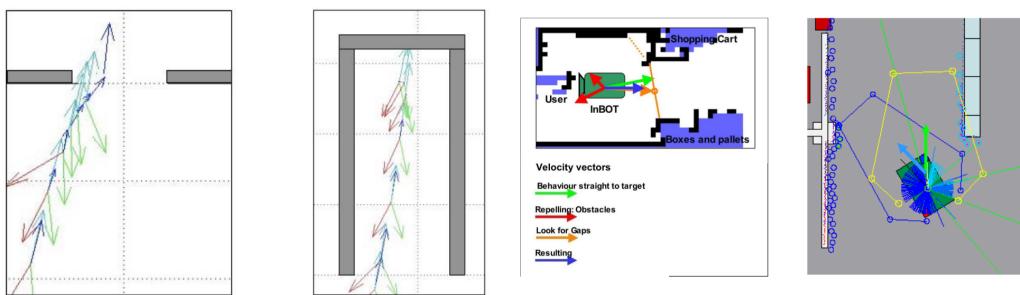


Abbildung 98: Aufgaben-Orientiertes Avoid Obstacles

- Hardwareunabhängig
- Modularer, erweiterbare Struktur vorhanden
- Robustheit ist abhängig von Einzelverhalten
- System ist sehr weit umgesetzt
- Flexibilität zur Laufzeit
- Effektive Erfüllung der Aufgabe

## 9 Lernende Ansätze

*Jedes Kind lernt irgendwann, dass heiße Herdplatten gefährlich sind. Können Roboter das auch?*

### 9.1 Motivation und Einführung in lernende Systeme

*Maschinelles Lernen ist genau genommen nur eine Optimierung mit clever parametrisiertem Gradientenabstieg.* Wenn kein Modell des zu steuernden Prozesses vorhanden ist, die Modellberechnung sehr komplex oder schwierig ist und/oder nicht modellierbare Umwelteinflüsse existieren. Dann muss der Zustands-/Aktionsraum experimentell „erforscht“ werden. Im Falle eines Roboters muss dies zwangsläufig (oder zumindest größtenteils) *online* sein.

**Subsymbolische (low-level) Lernverfahren:**

- Echtzeit Verarbeitung
  - Hohe Fehlertoleranz und Generalisierung
  - Leichte Erweiterbarkeit
  - Integration sensorischer Massendaten
  - Direkte Generierung kontinuierlicher Steuersignale
- Neuronale Netze sind als Lösungsansatz geeignet
- online/offline Lernen
  - überwachtes/uniüberwachtes Lernen

### 9.1.1 Gradientenabstieg

Optimierung einer Gütfunktion (häufig Minimum):

- Gradientenabstieg in mehrdimensionalem Feld, Bewegung entlang des Gradienten im > 3-Dimensionalen: lokale Minima und flache Plateaus sind ein Problem!
- Lernverfahren hören terminieren im Grunde nicht, sondern es wird eine Anzahl von Iterationen oder ein sonstiges Abbruchkriterium künstlich definiert
- Strategien: z.B. aggressivere Lernrate, aber dabei kann man über das Minimum hinaus schießen (in Abbildung 99d dargestellt)

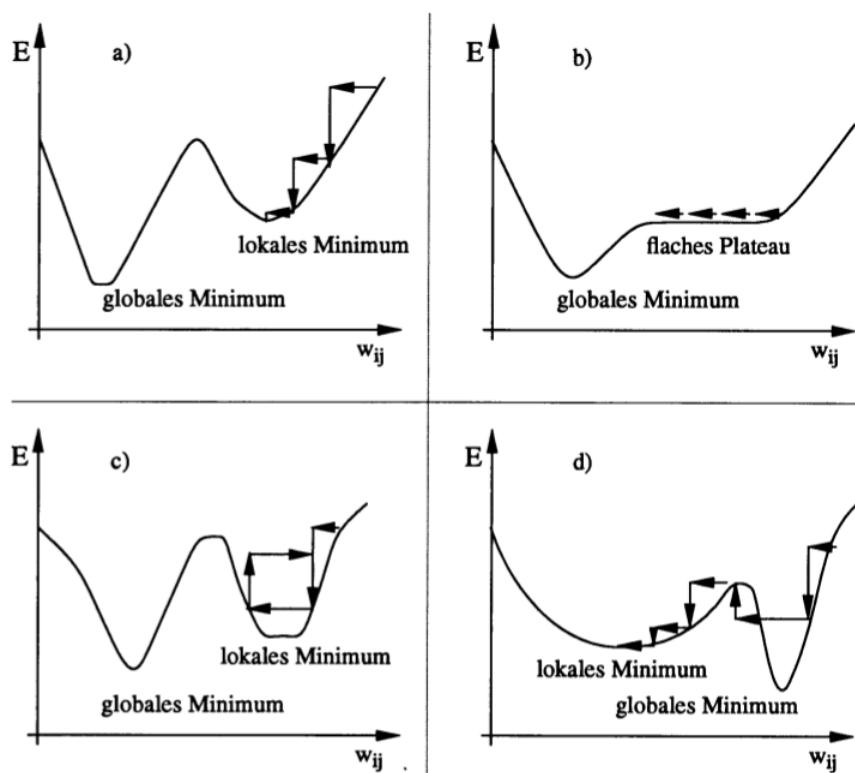


Abbildung 99: Gradientenabstieg

### 9.1.2 Reinforcement Learning – Zuckerbrot und Peitsche in der Robotik

- Online-Lernen
- Inkrementelles Lernen
- Sicherheits- und Robustheitsaspekte
- Lernen von Aktionssequenzen
- Kontinuierliche Aktionen

Unter verschiedenen Reaktionen, die auf dieselbe Situation hin ausgeführt werden, werden – bei Gleichheit anderer Bedingungen – diejenigen stärker mit der Situation verknüpft, die von einem für das Tier befriedigenden Zustand begleitet oder innerhalb kurzer Zeit gefolgt werden ... ; diejenigen Reaktionen, die von einem für das Tier unangenehmen Zustand begleitet oder dicht verfolgt werden, werden dagegen – wiederum bei Gleichheit anderer Bedingungen – in ihrer Verknüpfung mit der gegebenen Situation geschwächt.

**Formalisierung:**

Betrachte Zustände  $s$ , Aktionen  $a$ , Übergänge und Bewertung  $r$

$$S_1 \xrightarrow[r_1]{a_1} S_2 \xrightarrow[r_2]{a_2} \dots \xrightarrow[r_{n-1}]{a_{n-1}} S_n \quad (4)$$

$$\delta : (S \times A) \rightarrow S; \delta(s_t, a_t) = s_{t+1} \quad (5)$$

$$r : (S \times A) \rightarrow R; r(s_t, a_t) = r_t \quad (6)$$

Lernen entspricht dem Finden der (optimalen) Zielfunktion  $\pi : S \rightarrow A, \tau(s_t) = a_t$

wobei  $\pi :=$  optimale Handlungsstrategie, sodass die akkumulierte Bewertung (zum Ziel hin)

$$V^\pi(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \quad (7)$$

maximiert wird, mit  $V :=$  Value Function,  $s_t :=$  aktueller Zustand,  $r_t :=$  erhaltende Bewertung beim Übergang zu  $s_{t+1}$  und Gewichtung der Bewertung (Diskontierungsfaktor)  $0 \leq \gamma \leq 1$  ( $0$  : aktuelle Aktionsbewertung ist wichtig,  $> 0$  : zukünftige (letzte) Bewertungen werden bevorzugt ( $n$ -step)).

**Lernarchitektur** (siehe Abbildung 100):

Schleife über Roboter, z.B. für pneumatische Muskeln: Bewegung eines Beins

- beliebige Random-Steuersignale, Bewegung in richtige/vorgegebene Richtung
- kein Modell notwendig
- unabhängig von Technologie
- aber: bei sehr empfindlichen oder sehr großen Robotern nicht zu empfehlen, da der Roboter oder seine Umgebung dabei kaputt gehen könnten

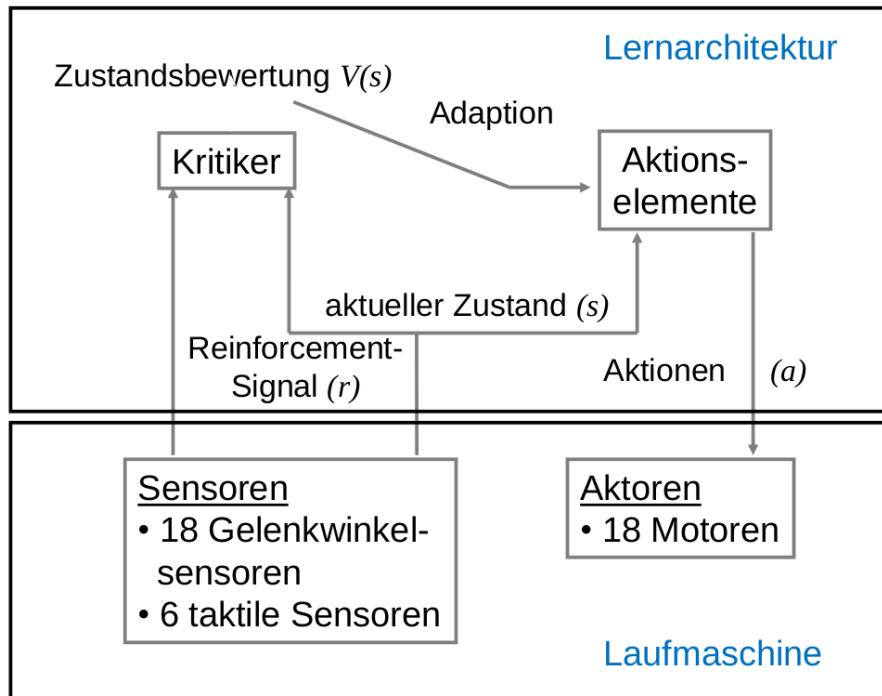


Abbildung 100

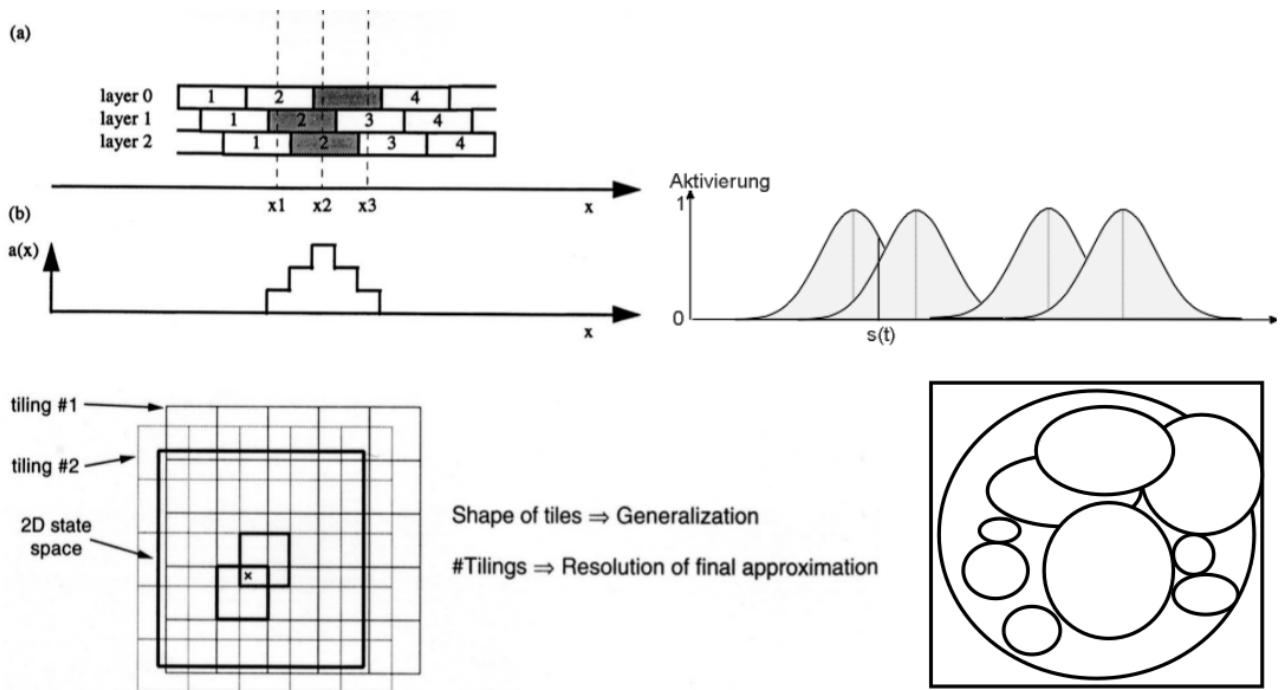


Abbildung 101: CMAC vs. RBF-Netze

## 9.2 Verschiedene Möglichkeiten der Wissensrepräsentation

### 9.2.1 Cerebellar Model Articulation Controller (CMAC)

Lokale Wissensrepräsentation mit mehreren assoziativen Netzen/Layer. Parameter sind die Anzahl der Gitter (Netze) und die Auflösung der Kacheln (Neuronen pro Netz). Hierbei entspricht die Anzahl der Gitter den Schichten des Netzes, die Auflösung der Kacheln entspricht Anzahl der Neuronen im Netz.

### 9.2.2 Radial Basis Functions (RBF)

- Lokale rezeptive Felder

$$a_i(\vec{s}) = e^{-\frac{\|\vec{s} - \vec{c}_i\|^2}{\sigma_i^2}} \quad (8)$$

$$out_j = \sum_i a_i w_{ij} \quad (9)$$

- Dynamische Ausbildung
  - Einfügestrategien: Aktivitätskriterium, Fehlerkriterium
  - Überlappung beim Einfügen

**CMAC vs. RBF-Netze** siehe Abbildung 101

## 9.3 Algorithmen zum Online-Lernen

### 9.3.1 Adaptive Heuristic Critic (AHC) mit Stochastic Real Value Exploration (SRV)

**Aktor/Kritiker-Ansatz** siehe Abbildung 102.

- NN wird über Sensorwerte (z.B. Gelenkwinkelstellungen) angeregt
- zwei Teilnetze, eines für Kritiker, anderes für Generation von Aktionen

- auch Kritiker wird bewertet → versuche, zukünftige Bewertungen für Kritiker zu prädizieren
- bewusste Variation der Aktionen, also nicht genau was Netz sagt sondern Exploration/Varianz (stochastischer Zufall), werden dann auf Roboter gegeben und dann wird wieder geschaut wie es läuft
- externe Bewertungsfunktion (vorher vorgegeben) + interne Bewertung über Kritiker: schaue von einem Zustand zum nächsten, wie Variation sich auswirkt, falls gut, adaptiere Gewichtung im Netz

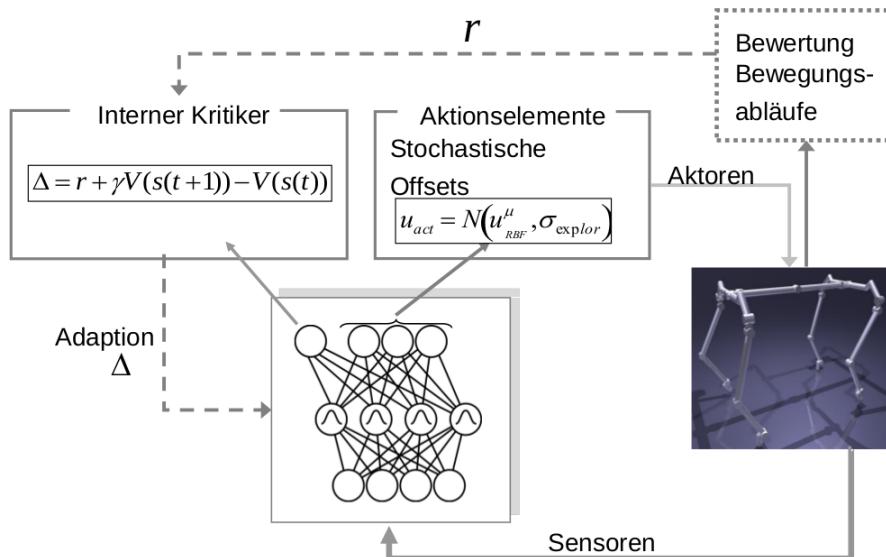


Abbildung 102: Aktor/Kritiker–Ansatz

### Stochastic Real-Valued Reinforcement Learning (Abbildung 103)

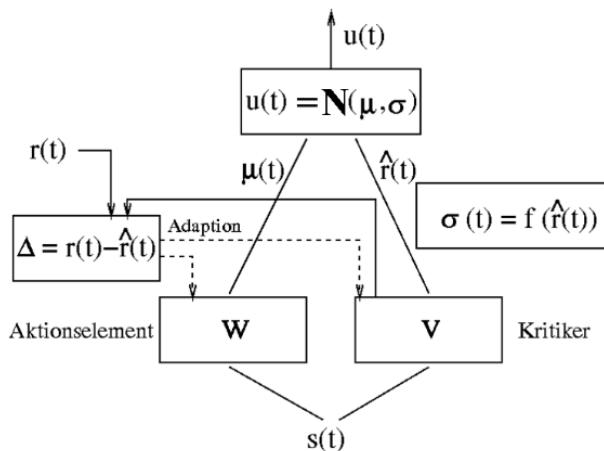


Abbildung 103: Reinforcement Learning

- Netzausgabe W – prototypische Aktion

$$\mu(t) = \sum_{i=1}^n w_i(t)x_i(t) + w_{thres}(t)$$

- Netzausgabe V – interner Kritiker

$$\hat{r}(t) = \sum_{i=1}^n v_i(t)x_i(t) + v_{thres}(t)$$

- Explorationsvarianz – erwartete Bewertung

$$\sigma(t) = f(\hat{r}(t))$$

$$f(\hat{r}(t)) = \max \left( \left( \frac{1.0 - \hat{r}(t)}{5.0} \right), 0.0 \right)$$

- Stochastische Aktionsauswahl (wird bewusst verrauscht)

$$u(t) \leftarrow N(\mu(t), \sigma(t))$$

- **Adaption der Aktionsgewichte abhängig von Aktionen und Zustand**

$$w_t(t+1) = w_t(t) + \alpha \Delta_w(t) s_t(t), \text{Lernfaktor } \alpha$$

$$\Delta_w(t) = (r(t) - \hat{r}(t)) \left( \frac{a(t) - \mu(t)}{\sigma(t)} \right) \text{ Bewertungsfunktion}$$

- **Adaption der Kritikergewichte**

$$v_t(t+1) = v_t(t) + \beta \Delta_v(t) s_t(t)$$

$$\Delta_v(t) = r(t) - \hat{r}(t)$$

### 9.3.2 Zustandsraum Randwertkontrolle mit Boundary Controllern (BC)

**Problematik:** Der Zustandsraum eines realen Roboters ist begrenzt (durch Mechanik, Motortreiber/Elektronik, Leistung). Die stochastische Exploration führt zum Verlassen des erlaubten Zustandsraumes

- Basissteuerung des Roboters verhindert Aktion
- Zustand des Roboters ändert sich nicht
- kein  $\Delta$  mehr
- kein Lernen mehr möglich

**Lösung:** Aktive Kontrolle des Randbereiches und Eingriff in den Lernvorgang mit Hilfe eines *Boundary Controller*: soll erkennen, wenn Zustandsraum verlassen wird

- Randbereichsbewertung  $b(s_t)$ :

$$r_{t_{gesamt}} = r_{t_{sensor}} - b(s_t)$$

$$b : S \Rightarrow R$$

$$s_t \in S \text{ Zustandsraum}$$

- Identifikation der für das Verlassen des Aktionsraumes verantwortlichen Stellgrößen  $u_k$

$$b_k^r = \begin{cases} 1 & : u_k \text{ verantwortlich für Randaufenthalt} \\ 0 & : \text{sonst} \end{cases}$$

$$b_r \in R^k$$

$$k = \dim(u)$$

- Modifikation der Stochastischen Exploration

$$(b_r^k == 1) \wedge ((u_k - \bar{u}_k) > 0) \Rightarrow u_k := u_k - |u_k - u_k^\mu|$$

- Modifikation der Adaption der Aktionsgewichte

$$w_t(t+1) = \begin{cases} w_t(t) + \alpha \Delta_w(t) s_t(t) & : \forall b_r^k : b_r^k \neq 1 \\ w_t(t) & : \text{sonst} \end{cases}$$

## 9.4 Anwendung

### 9.4.1 Zweibeiniges Laufen mit Hilfe von CMAC und SRV

Biped dynamic walking using Reinforcement Learning:

- Ziele
  - Experimentelles Lernen bzw. Optimieren der Bewegungsabläufe
- Roboter
  - 6 aktive Freiheitsgrad in der Sagitalebene: Hüfte, Knie, Knöchel
  - Passive Freiheitsgrade in Schulter und Händen
  - Wagen zur Restriktion in der Sagitalebene
- Ziele für das Gehen
  - Aufrecht
  - Elegant, flüssig
  - Kein Schlurfen oder Nachziehen der Beine
- Vorgaben für die Steuerung
  - Körperhöhe und -neigung innerhalb fester Grenzen
  - Anheben des Schwingbeines über eine vorgeschriebene Höhe
  - Periodische Bewegungen
  - Energieeffizienz
- Diese Vorgaben spezifizieren das Laufmuster nicht vollständig, legen aber die Bereiche fest, in denen das Lernen der Bewegungsmuster stattfinden soll – Roboter lernt nicht gesamten Aktionsraum; verhindert langes Lernen und Feststecken in lokalem Minimum (z.B. „Schlurfen“)
- Steuerungsstrategien
  - 1 Hauptgangregler
  - verschiedenen Regler, die situationsabhängig ausgewählt werden
  - Melting Pot
- Zentraler Mustergenerator Abbildung 104a
  - Realisiert mit CMAC
  - Adaption durch die Kritikerbewertungen der peripheren Controller
- Haltungskontrolle Abbildung 104b
  - Sehr wichtige Steuerungskomponente, da kleine Neigungen im Körper schon zu großen Momenten auf das tragende Bein führen können
  - Nur bei kleinen Abweichungen ist die Beeinflussung z.B. durch Knöchel und Hüfte möglich
  - Bei größeren Abweichungen müssen Zwischenschritte eingefügt werden
- Körperhöhe Abbildung 104c
- Zusammensetzung Abbildung 104d
- Lernarchitektur Abbildung 104e
- Gesamtübersicht Aktor Abbildung 104f

- CPG Kritiker Abbildung 104g
  - Die einzelnen peripheren Kritiker erzeugen eigene Reinforcement Signale, die dann gewichtet zusammengefasst werden
- Pre-training
  - Vorgabe geeigneter Fußtrajektorien Abbildung 104h
  - Konstante Körperhöhe, so dass am Ende der Schritte die Beine ausgestreckt sind
  - Der CPG CMAC wird mit den so erzeugten Gelenkwinkelsequenzen für verschiedene Schrittängen und Schrittdauern eintrainiert
  - Für das Einlernen der Haltungskontrolle werden 2 PD Regler eingesetzt
  - Kritiker-Vorlernen
  - Einlernen der Kritiker ohne Aktionsvarianz

#### 9.4.2 Online-Lernen auf der vierbeinigen Laufmaschine BISAM

**Ziele:** Experimentelles Lernen zur Optimierung der Haltung (Schwerpunkt) im Kreuzgang

**BISAM:** 21 aktive Freiheitsgrade (4 pro Bein, 5 im Körper), 2 Achsen Lagesensor, 3 Achsen Winkelbeschleunigung, 3 Komponenten Kraftsensor pro Fuß, Onboard PC

##### BISAM – Lernen Gewichtsverlagerung im Schritt

- Problemstellung: Optimierung der Schwerpunktsverschiebung zur Unterstützung des Schritts
- Bewertungskriterien: Lage des Schwerpunktes (static stability margin, SSM)
- Netzeingaben: SSM<sub>x</sub>, SSM<sub>y</sub>, letzte Verschiebung x, Verschiebung y
- Netzausgaben: Kritiker, Verschiebung x, Verschiebung y

##### BISAM – Lernen der Schwerpunktsverschiebung im Kreuzgang

- Problem: Lernen der Schwerpunktsverschiebung, SSM<sub>y</sub> (links/rechts)
- Netzeingabe: SSM<sub>y</sub>(k-1), SSM<sub>y</sub>(k), shift Y(k)
- Netzausgabe: interner Kritiker, shift Y(k+1)
- Inkrementelles Lernen: Stehen/Laufen, Ohne/mit Exploration

#### 9.4.3 Diskussion

- Ist experimentelles Lernen bei relevanten (technischen) Problemstellungen anwendbar?
  - sehr lange Lernprozesse
  - kaum verlässliche Aussagen über den Lernprozess
- Wenn hinreichendes Prozessmodell vorhanden ist, kann RL erfolgreich eingesetzt werden (z.B. Fahrstuhlsteuerung)
- Ohne Prozessmodell sind Optimierungsaufgaben denkbar, wobei beim Lernen nicht optimale aber keine gefährlichen oder schädlichen Prozesszustände eingenommen werden
- Bei Robotern noch aktuelle Grundlagenforschung

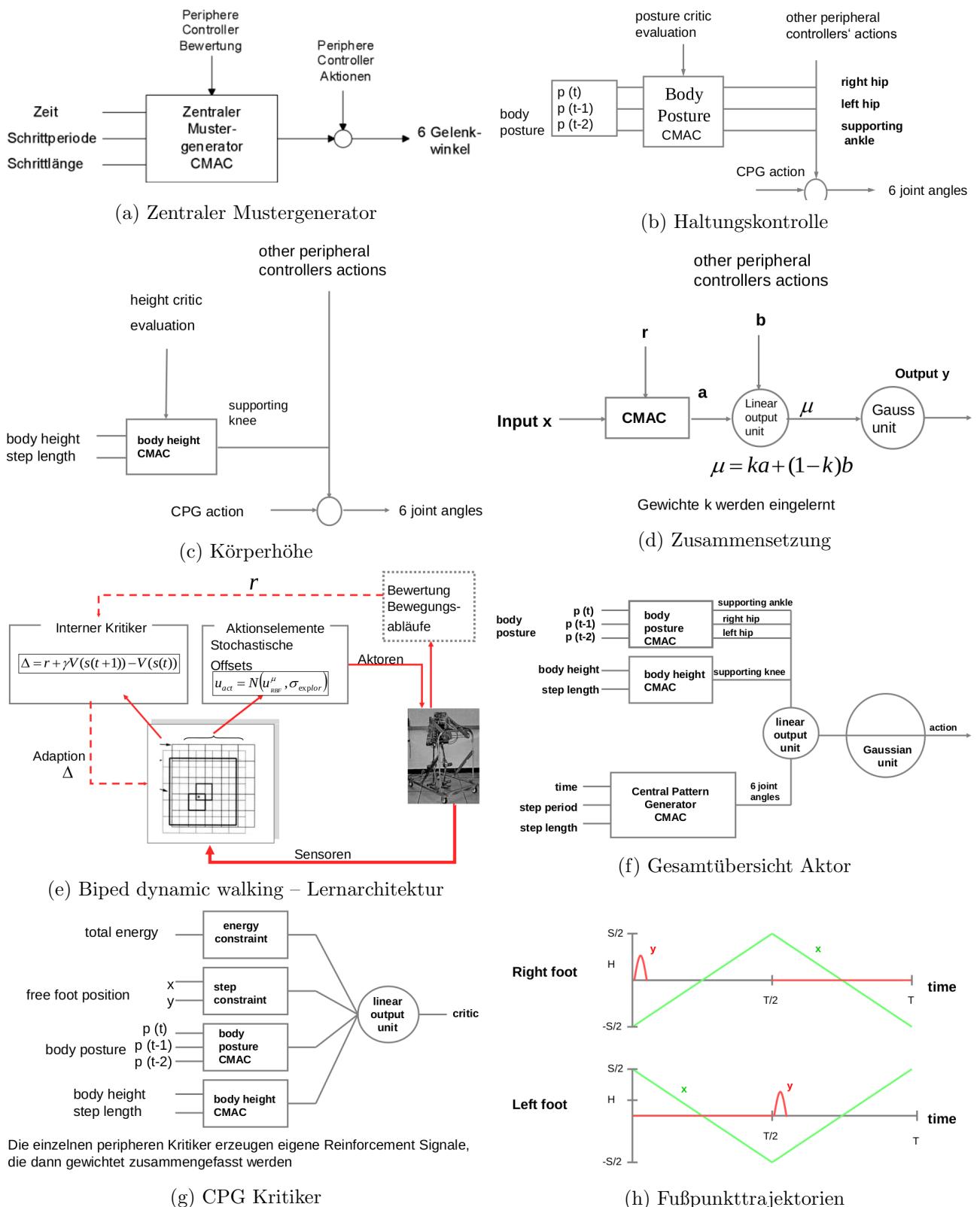


Abbildung 104

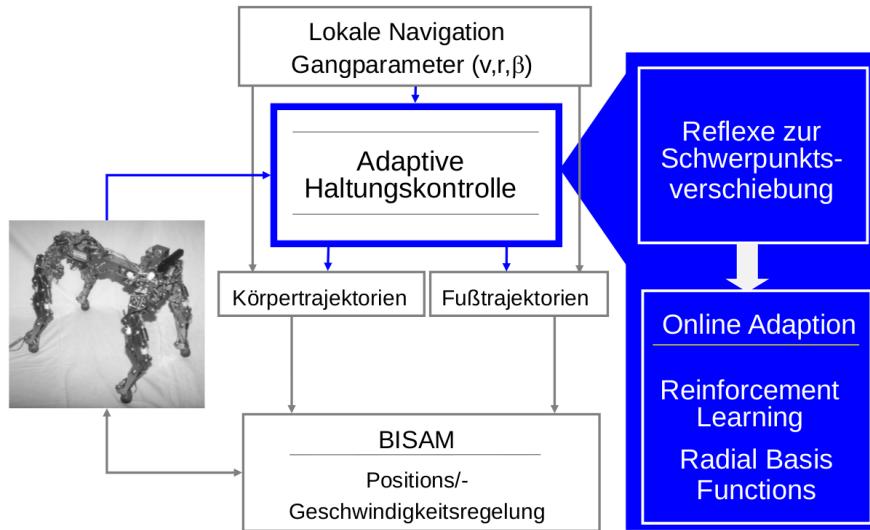
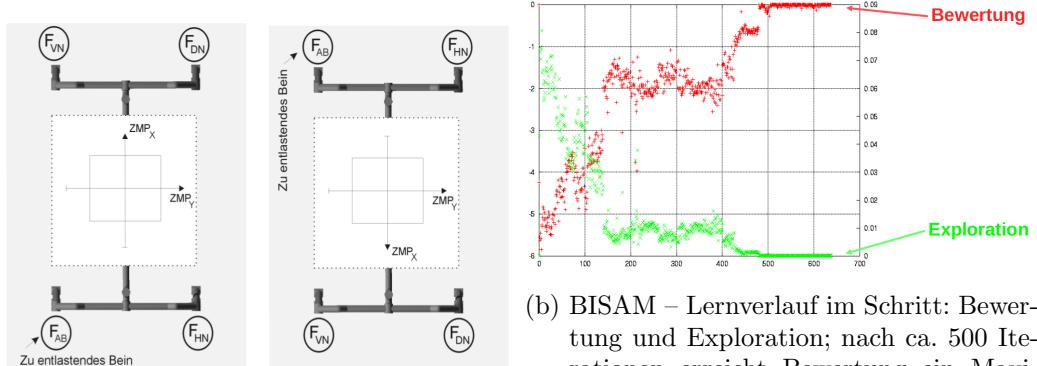


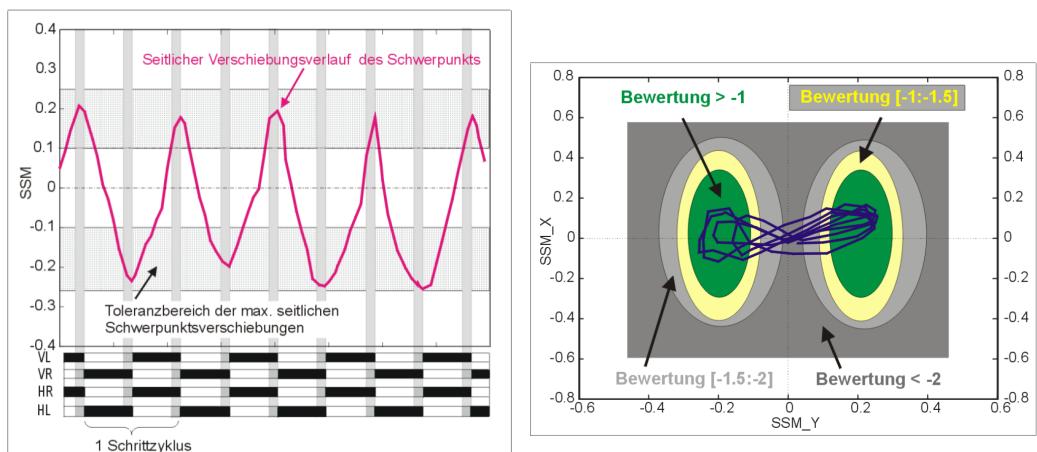
Abbildung 105: BISAM – Steuerungsarchitektur: lerne Reflexe, entwickle System welches Roboter selbst stabilisiert



(a) BISAM – Lernen der Gewichtsverlagerung im Schritt

(b) BISAM – Lernverlauf im Schritt: Bewertung und Exploration; nach ca. 500 Iterationen erreicht Bewertung ein Maximum (stabiler Zustand) und Exploration wird aufgrund der guten Bewertung eingestellt

Abbildung 106: Lernen von BISAM



(a) BISAM – Schwerpunktverlauf Kreuzgang: Grenzen: ca. -0.25, außerhalb dieses Bereiches würde Roboter umfallen; Exploration in diesem Bereich früh genug verhindern (Boundary)

(b) BISAM – Bewertung des Schwerpunktverlaufes im Kreuzgang: stabile Bereiche für Schwerpunkt in  $x$ - und  $y$ -Richtung in grün

Abbildung 107

## 10 **Humanoide Service- und Assistenzroboter**

### 10.1 Übersicht aktueller, humanoider Robotersysteme

#### 10.1.1 Fragestellungen bei der Entwicklung Humanoider

- Welche physiologischen Eigenschaften braucht ein humanoider Roboter? (z.B. Haut nötig? Anzahl der Arme?)
- Welche Fähigkeiten müssen existieren um jede Art von Objekt zu manipulieren?
- Wie können wir diese Fähigkeiten finden und optimieren?
- Wie zuverlässig ist das Robotersystem und wie sicher in Kooperation mit dem Menschen?
- Welche Information benötigt man über das Einsatzumfeld des Roboters und den menschlichen Operator?
- Wie kann der Roboter mit dem menschlichen Operator kommunizieren und umgekehrt?
- Wie kann sich der Roboter selbst an neue Anwendungen anpassen?

#### 10.1.2 Beispiele humanoider Roboter

Mechatronik schon sehr gutm Autonomie und Perzeption noch sehr basal.

- Honda **Asimo**, Japan: sehr stabiles Laufen, kann auch auf einem Bein hüpfen
- HRP4, Japan
- **ARMAR III**, Deutschland: hybrides system (keine Beine), aber von Perzeption und Planungskomponenten sehr weit
- **Walking Trumpet** (Toyota), Japan: Entertainment
- **Wabian**, Japan: sehr natürliches menschenähnliches Laufen
- Shadow Biped, UK: pneumatische Muskeln
- iCub, Italien, UK, Schweiz: Mechatronik sehr weit
- Rollin' Justin, DLR, Deutschland: kann z.B. Ball im Flug aus der Luft fangen, regelungstechnisch sehr weit
- Elvis, Uni Gothenburg, Schweden: viele kleine Servomotoren, wenige Freiheitsgrade in Beinen
- HoLLiE, FZI, Deutschland
- ARMAR 4, KIT, Deutschland
- REEM-C, PAL, Spanien: sehr weit kommerzialisiert (→ Öl-Schaich), auch hybride Version
- TORO, DLR, Deutschland: Weiterentwicklung von Justin, sehr fortschrittliche regelungstechnische Modelle, gutes Laufen
- SCHAFT, Japan (DARPA): Gewinner im Vorentscheid, dann von Google gekauft und von DRC abgemeldet
- ATLAS, USA (DARPA): entwickelt von Boston Dynamics für DARPA (10 an Unis, 2 Mio. das Stück)
- HUBO, Japan (DARPA)

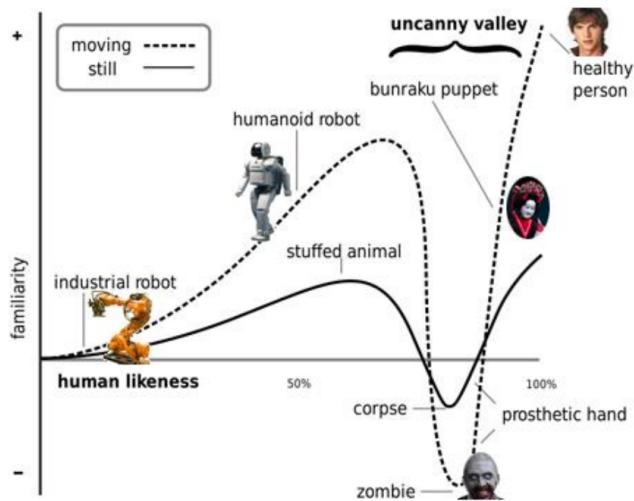


Abbildung 108: Uncanny Valley

- Valkyrie, USA (DARPA): JPL, Entwickler von Curiosity; nicht fertig geworden
- Teilkomponenten:
  - DLR-Arm II + III
  - Fa. Otto Bock, Duderstadt
  - Frauenhofer, Magdeburg
  - Schunk SVH: 5-Finger-Hand, schon im industriellen Einsatz, mechatronisch sehr fortschrittlich

### 10.1.3 Uncanny Valley

Je ähnlicher Roboter dem Menschen werden, desto unheimlicher wirken sie auf uns. Wenn sie zu ähnlich sind vergleicht man sie direkt mit echten Menschen und wirken daher freakig. Das Uncanny Valley ist ein stark untersuchtes wissenschaftliches Phänomen. Sollte man als Forscher bewusst abstrahieren und Roboter lieber eckig und kantig gestalten, um ihre Akzeptanz zu erhöhen? Dieses „Tal“ muss überwunden werden (vgl. Abbildung 108).

## 10.2 ARMAR der humanoide Roboter aus Karlsruhe

- ARMAR, 2000
- ARMAR-II, 2002
- ARMAR-IIIa, 2006
- ARMAR-IV, 2012

### 10.2.1 Zielsetzung des ARMAR Projekts

- Humanoide Roboter als Service-Roboter
- Niedrige Herstellungskosten, Leichtbaukonstruktion mit niedrigem Energieverbrauch
- Einfache Programmierung von Manipulationsaufgaben
- Direkte Mensch-Roboter-Interaktion
- Einfacher, direkter Transfer von menschlichen Fertigkeiten
- Adäquate Benutzerschnittstelle

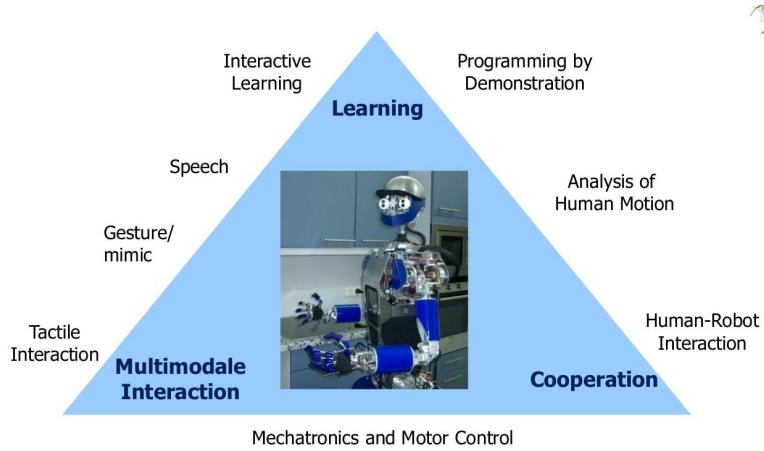


Abbildung 109: ARMARx, PdV

### 10.2.2 ARMAR-III

- 7 DOF Lightweight arms
  - Position, velocity and torque controlled
  - Force-Torque Sensors
  - Sensitiv Skin
- 7 DOF robot head
  - 2 cameras in each eye
  - 6 microphones
- 3 DOF torso
  - 2 embedded PCs
  - 10 DSP/FPGA Units
- Holonomic mobile platform
  - 3 laser scanner
  - 3 embedded PCs
  - 2 batteries
- Head
  - 3 DOF eyes (Pan left, Pan right, Tilt)
  - 3 DOF neck (roll, pitch, yaw)
  - incremental encoders for all DOF
  - 6 microphones
  - 2 stereo camera pairs
- Cameras
  - PointGrey DragonFly
  - Connection: Firewire 400
  - Framerate: 30fps max.
  - Resolution: 640x480 max.
  - Focal length foveal pair: 12mm

- Focal length perspective pair: 2,9mm
- Mechatronics and Control
  - Speech input and output
  - Collision-free navigation
  - Acoustic localization
  - Visual user tracking
  - Visual color-based object
  - Redundancy resolving (Redundanz in inverser Kinematik, plötzlich ein df mehr)
  - Hybrid force position control
  - Coordination of task execution

## 10.3 Programmieren humanoider Roboter

### 10.3.1 Einführung

- Artificial Skin for the Support of Safe MMI
  - Multimodale künstliche Haut mit Distanzsensoren: erhöht Sicherheit; direktes Feedback um Roboter irgendwo hin zu ziehen
  - Japaner haben schon Babysitter-Roboter (aber schon mutig)
  - Beschleunigungssensoren
  - Kraft-Moment-Messung
  - Finger mit speziellen Tastsensoren
- Inverse Kinematik
  - brauche 6 DoF um jede Position im Raum anzufahren (3 Richtungen + 3 Orientierungen), bei 7 ist ein Gelenk nicht bestimmt → mathematische Probleme beim Lösen der inversen Kinematik
  - Keine Invertierung der Jacobi-Matrix
  - Redundanz ist parametrisiert als Kreis im kartesischen Raum
  - Direkte Kontrolle von allen Punkten im Arbeitsraum

### 10.3.2 Programmierung von Manipulationsaufgaben

- Adaption von humanoiden Armbewegungen
  - Vermessung von Position und Orientierung von Ellbogen und Handgelenk
  - Simulation der natürlichen Armbewegung mit 3D-Modell des menschlichen Arms
  - Transfer der Bewegungen
  - Inverse Kinematik
- Adaption von Arm- und Körperbewegung basierend auf Manipulationsaufgabe und Umweltsituation

Human detection in Smart rooms:

- Sensoren:
  - On-board: Stereo-camera + 2...8 microphones
  - Extern: 2...n monocular cameras + 4-channel microphone arrays

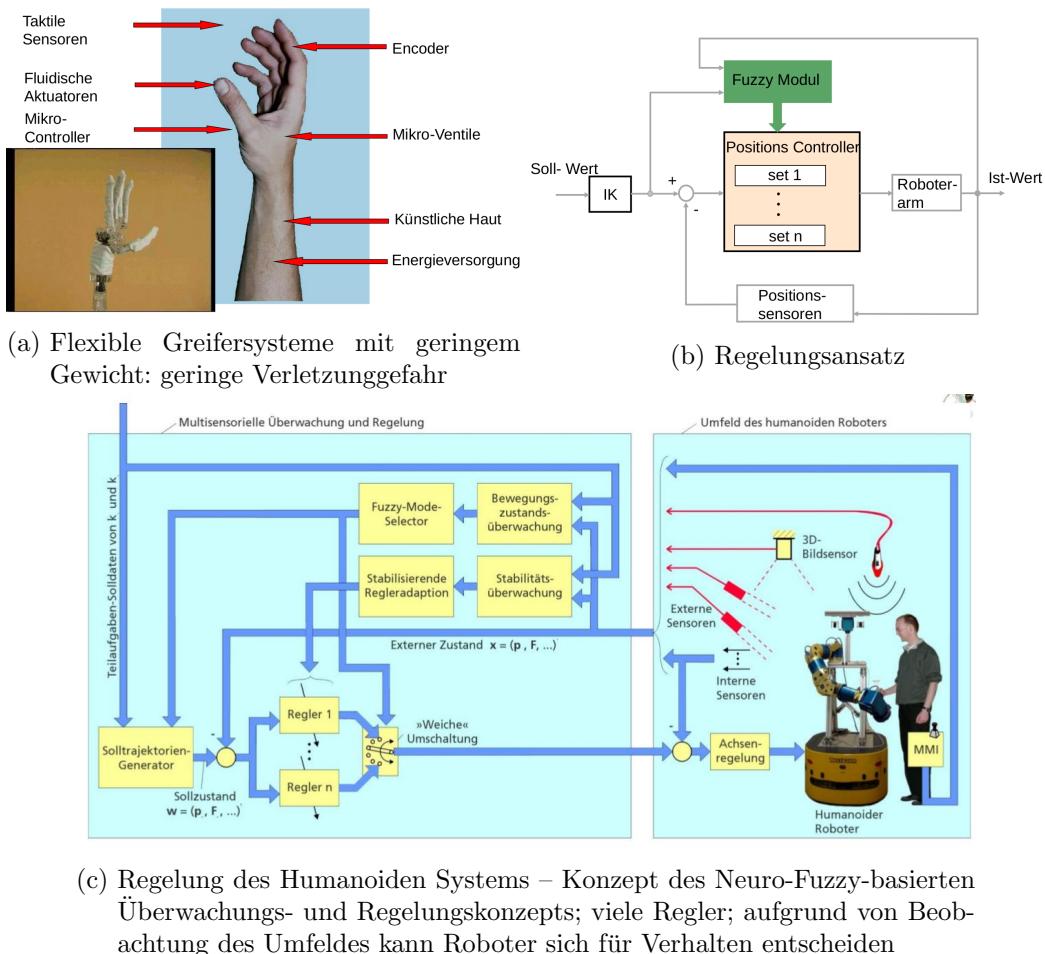


Abbildung 110

- Features:
  - Individual color models, detectors for head and upper body, difference images
- Methode:
  - Shared particle filter for A/V features, local calculation of features achieving real-time computation

### Programmieren durch Vormachen:

- Demonstration Environment und Execution Environment.
- Multimodale Perzeption: mit Kameras, Kinects, Datenhandschuh usw.
- Interaktives Lernen von humanoiden Robotern:  
Menschlicher Operator als Lehrer: Demonstration, Kontrollabsichten, Kommentare zu ausgeführten Aktionen  
Roboter: Hypothesen bilden, Nachfragen von Hypothesen
- Simulation eines humanoiden Roboters: VR für Roboter-Mensch Interaktion. Simulation eines komplexen mechatronischen Systems mit unterschiedlichen Sensoren.

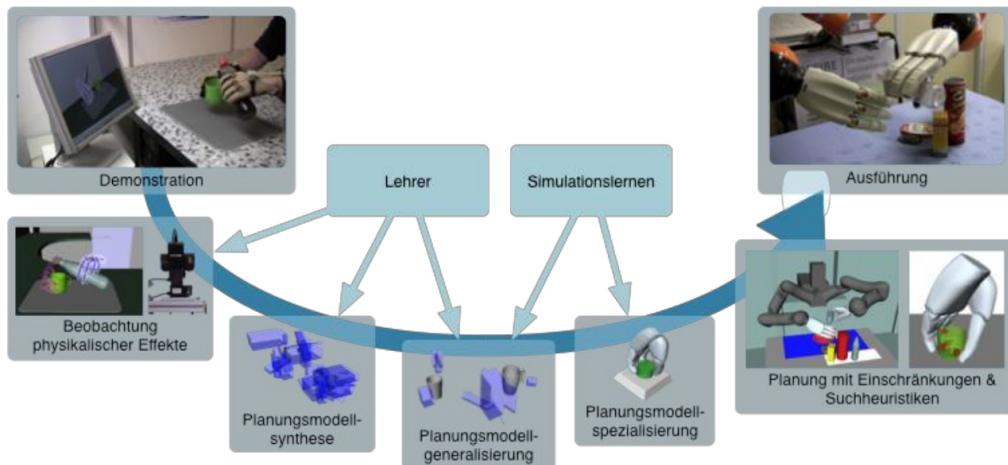


Abbildung 111: Programmieren durch Vormachen

## 10.4 Perzeption für humanoide Roboter

### 10.4.1 Detektion und Modellierung menschlicher Operator und Umwelt

- Entwicklung eines erweiterten Umfeldmodells, mit interaktiven Mechanismen
- Kontextsensitives Model für die Beschreibung und Voraussage von menschlichen Operator Bewegungen
- Präzise Modelle menschlicher Handbewegungen für die Manipulation
- Unterstützung von Mensch-Roboter-Interaktionen basierend auf einem umfassenden Grundwissen der Umweltsituation und der Intention des menschlichen Operators
- Robuste Detektion und Segmentierung
- Audio-visuelles Tracking des menschlichen Operators
- Detektion von bestimmten Körperteilen
- Bestimmung des „focus of attention“ des menschlichen Operators
- Audio-visuelle Kontrolle des „focus of attention“

### 10.4.2 Akustische Szenen-Analyse und Stereo Vision

Akustische Szenen-Analyse:

- Reduktion von Lärm zwecks Spracherkennung
- Lokalisation und Segmentierung von Lärmquellen
- Identifikation von Lärmquellen
- Erweiterung des Umweltmodells

Stereo Vision:

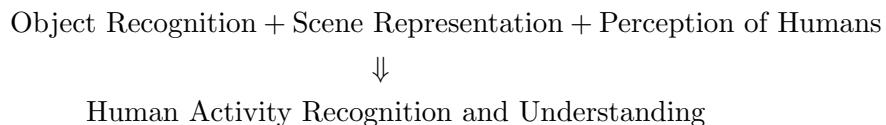
- Correlation based depth reconstruction
- Texture mapping
- Color based object recognition methods
- Stereo based object localization methods

### 10.4.3 Modeling of Scenes and Objects

- Development of a flexible environment model database
- Generic object and scene representations
- Interactive object modelling methods and tools
  - high accuracy 3D laser scanner
  - rotation joint with camera bracket
  - turn table
  - high resolution stereo camera system
  - software controlled lights for environment variation

### 10.4.4 Colored Object Recognition

- Segmentation by color
- Appearance-based recognition using a global approach
- Model-based generation of viewsets
- Combination of stereo vision and stored orientation information for 6D-localization (with respect to a given 3D model)



## 10.5 Anwendungen und Einsatz

- DESIRE (Deutsche Service Roboter Initiative) – Functions and components for everyday use
  - Perception
    - \* Safe and fast detection and modeling of objects and obstacles in 3D
    - \* Module for the analysis of everyday scenes
    - \* Algorithms for the robust recognition and identification of people and interaction partners
  - Mobile Manipulation
    - \* Safe and adaptive grasping and manipulation of everyday items
    - \* Robust, collision-free control of highly complex kinematics
  - Mechatronic Components
    - \* Hardware for the technology platform and the specific service robots and applications
    - \* Modules for multimodal commands, navigation and localization components for both the indoor and outdoor navigation of mobile robots
- Car-O-Bot – Assistenz und Pflegeroboter
  - Mobile Roboterbasis mit 1 bzw. 2 Leichtbauarmen des Fraunhofer IPA, Stuttgart
  - Transportaufgaben im Haushalt und Pflegeeinrichtungen
  - Unterstützung im Alltag
  - Fokus: Mensch-Maschine Interaktion
  - Mittlerweile 4. Generation (4. Generation: komplett modulares Design)

- Rollin Justin
- AIS Rob
- HoLLiE (House of Living Labs intelligent Escort) – Service Roboter
  - Kombination industrieller Komponenten zu innovativen Roboter
  - Zwei 6 DOF Arme
  - Zwei 5 Finger Hände
  - Omnidirektionale Basis
  - 2 Knickgelenke im Torso
  - 2 Freiheitsgrade im Kopf
  - Großer Arbeitsraum (erreicht Fußboden)
- Asimo
- Wabian-2R (Zweibeiner)
- 5-Finger Hand an 6-Achs Arm
- Petman – boston Dynamics 2009, 2011, 2013

## 11 Mutation

### 11.1 Vererbung und Evolution in der Biologie

#### 11.1.1 Die Mendelschen Gesetze

Nomenklatur (vgl. Abbildung 112)

- Homozygotie: Reinerbigkeit im Bezug auf ein genetisches Merkmal
- Heterozygotie: Mischerbigkeit im Bezug auf ein genetisches Merkmal
- Allel: mögliche Ausprägung eines Gens, das sich an einem bestimmten Ort auf einem Chromosom befindet
- Gen: Abschnitt auf der DNA
- Chromosom: Strukturen, die Gene und damit Erbinformationen enthalten. Sie bestehen aus DNA, die mit vielen Proteinen verpackt ist
- Genom (Erbgut): Gesamtheit der vererbaren Informationen einer Zelle oder die Gesamtheit der materiellen Träger der vererbaren Informationen, also die Gesamtheit der Chromosomen, der Gene oder DNA
- Diploider Organismus: besitzt von jedem Gen, das z.B. die Blutgruppe oder Haarfarbe kodiert, zwei Kopien, im Normalfall eine von jedem Elternteil (Mensch ist diploid)

#### Vererbung von Merkmalen in der Natur – Mendelsche Gesetze

1. **Uniformitätsgesetz:** Kreuzt man zwei homozygote Linien, die sich in einem oder mehreren Allelpaaren unterscheiden, so sind alle F1-Hybriden uniform.
  - Es werden zwei reinerbige (homozygote) Eltern verpaart
  - Die Eltern unterscheiden sich
  - Die Nachkommen der ersten Generation (F1-Hybriden) sind dann gleich (uniform) bezogen auf das untersuchte Merkmal

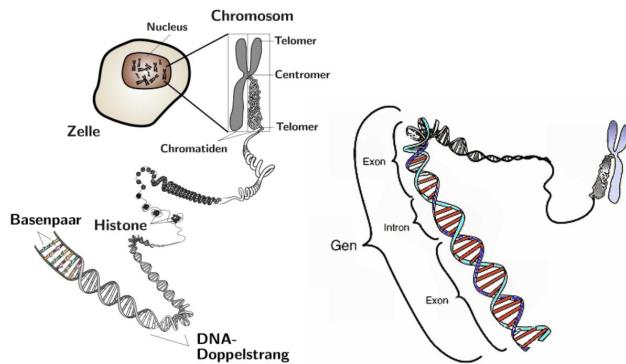


Abbildung 112: Vererbung Nomenklatur

- Unterscheidung je nach Erbgang:
  - dominant-rezessiv: ein Merkmal dominant, ein Merkmal rezessiv
  - intermediär: Merkmale bilden Mischform
  - Kodominant: beide Merkmale separat ausgebildet
- 2. **Spaltungsgesetz:** Kreuzt man (F1-)Hybriden, die in einem Allelpaar heterozygot sind, so ist die F1-Generation nicht uniform.
  - Es werden zwei mischerbige (heterozygote) Eltern verpaart
  - Die Eltern besitzen beide die gleichen Merkmalsanlagen
  - Die Nachkommen der ersten Generation (F1-Hybriden) sind dann nicht gleich bezogen auf das untersuchte Merkmal
  - Auswirkung durch Erbgang: dominant-rezessive Vererbung:
    - $\frac{1}{4}$  rein rezessive
    - $\frac{2}{4}$  mischerbig
    - $\frac{1}{4}$  rein dominant
- 3. **Unabhängigkeitsregel** (gilt eingeschränkt): Kreuzt man zwei homozygote Linien untereinander, die sich in zwei oder mehreren Allelpaaren voneinander unterscheiden, so werden die einzelnen Allele unabhängig voneinander, entsprechend den beiden ersten Mendelschen Gesetzen vererbt.

### 11.1.2 Zellteilung, Vererbung und Mutation

#### Zellteilung

- **Meiose** = Reifeteilung = Reduktionsteilung
  - besondere Art der Zellkernteilung
  - Zahl der Chromosomen halbiert sich
  - vollzieht sich immer in zwei Teilungsschritten
  - Rekombination der elterlichen Chromosomen
  - Grundlage der geschlechtlichen Fortpflanzung (Säugetiere, Fische, Vögel..)
  - Haploidie: Chromosomensatz einer Zelle nur einfach vorhanden, Zellkern enthält von allen Chromosomentypen nur ein einziges Exemplar. Typischerweise sind die Chromosomensätze der Eizellen und Spermien haploid. Ihre haploiden Chromosomensätze verschmelzen bei der Befruchtung zum doppelten Chromosomensatz einer diploiden Zelle.
  - Homolog: zwei Chromosomen enthalten gleiche Gene.

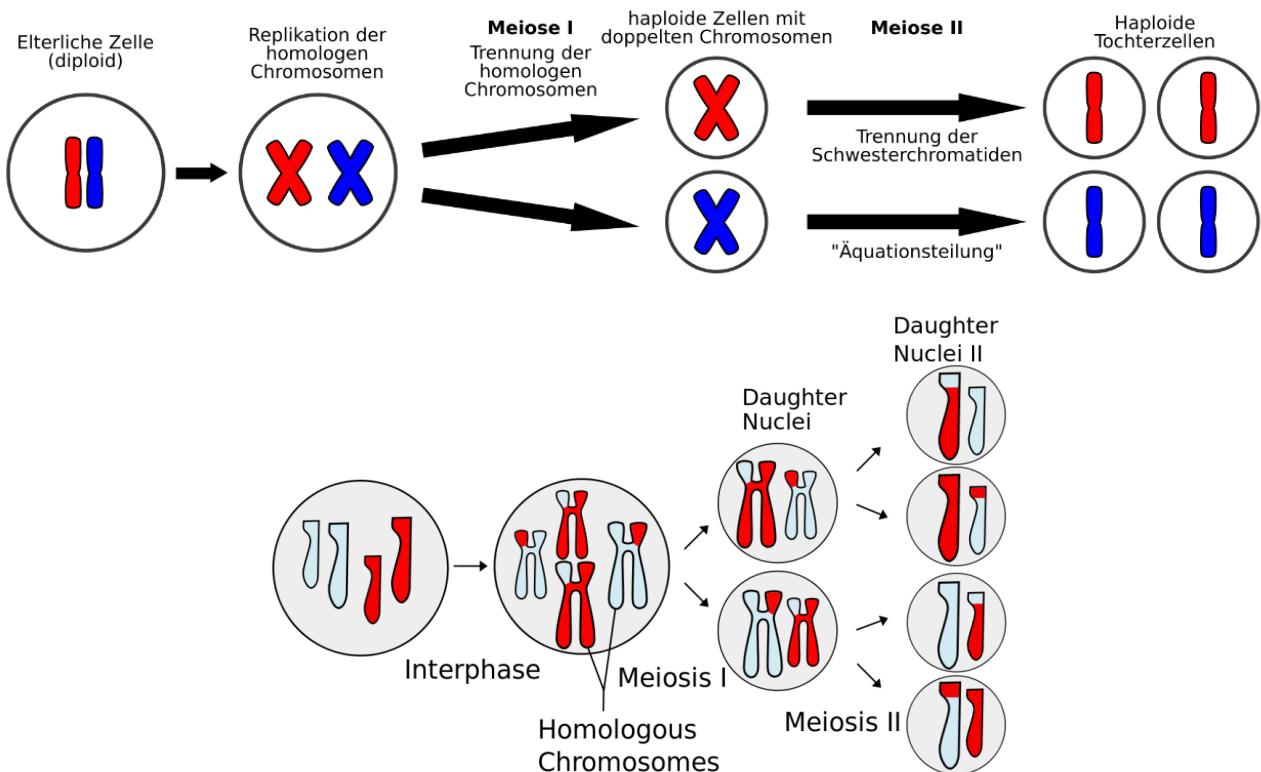


Abbildung 113: Meiose – Zellteilung

- **Mitose** = Karyokinese = indirekte Kernteilung
  - gewöhnliche Kernteilung
  - Keine Rekombination der Chromosomen
  - Beispiele: Bakterien, manche Pflanzen, Gewebe, ...

## Mutationen

- Genommutationen (Monosomie, Trisomie)
- Chromosommutation (Verlust, Verdopplung, Hinzufügung, Drehung 180°)
- Genmutationen (ähnlich wie Chromosommutation nur auf Genbasis)

## 11.2 Genetische Algorithmen – „Evolution im Rechner“

### 11.2.1 Genetische Algorithmen – Motivation und Nomenklatur

Biologisches Evolutionsmodell nach Darwin: *survival of the fittest* als Synonym *natural selection*; Selektion als treibende Kraft

Evolution als Optimierung komplexer künstlicher Systeme: Zufall und Selektion ⇒ hoch komplexe und adaptive Arbeits- bzw. Lebensumgebung hervorragend angepasste Systeme erzeugen lassen

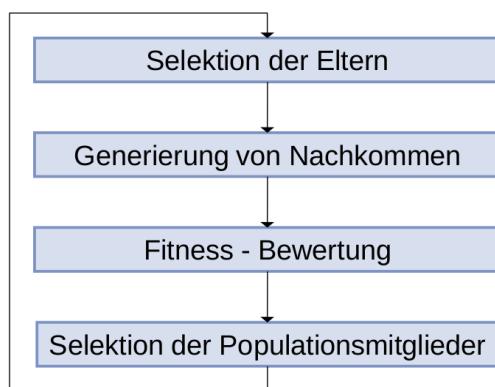
## Nomenklatur

- Fitnessfunktion  $\triangleq$  zu optimierende Bewertungsfunktion
- Individuum oder Hypothese  $\triangleq$  Einzelne Lösung
- Population und Generation  $\triangleq$  Lösungsmenge
- Erzeugung von Nachkommen  $\triangleq$  Generierung neuer Lösungen

- Genetische Operationen  $\triangleq$  Rekombination, Mutation
- Veränderter Nachfolger, Kind, Nachkomme  $\triangleq$  neue Lösung
- Selektion der Besten  $\triangleq$  Auswahl der Lösungen, welche die beste Optimierung erzeugen

### 11.2.2 Grundalgorithmus

Solange nicht optimal



### 11.2.3 Generierung von Nachkommen

Exploration  $\Leftrightarrow$  Exploitation

Untersuchen des Raumes  $\Leftrightarrow$  lokale Optimierung

- Je stärker und zufälliger Änderungen sind, desto geringer ist die Wahrscheinlichkeit, einen besseren Nachkommen zu erzeugen.
- Bei lokalen Verbesserungsmethoden ist die Gefahr der lokalen Minima gegeben.
- Explorationsfaktor muss gemäß der aktuellen Fitness der Generation ausgewählt werden.

### Kodierung der Individuen

- Wissen/Information muss strukturiert repräsentiert werden
  - Wie viel von dieser Strukturinformation soll genutzt werden?
    - \* kein Einsatz, ausschließlich Anwenden der Algorithmen auf den Binärsequenzen
    - \* volle Ausnutzung der Strukturinformation  $\Rightarrow$  Spezielles Zuschneiden des Optimierungs-algorithmus
- Kodieren der Gene
  - Binärcodierung  $\Leftrightarrow$  Genetische Algorithmen
  - Reelle Zahlen, Vektoren  $\Leftrightarrow$  Evolutionäre Strategien (Beispiel: 1010|1110|1111|1010|1000|1010)

**Mutation:** Der Nachkomme stammt von einem Elternteil ab

- Mutation einzelner Bits: Bitinversion

Elternteil: 1010110

Nachkomme: 1011110

- Konzepte:

- Alle Bits einer Sequenz werden unabhängig voneinander mit einer bestimmten Wahrscheinlichkeit invertiert
- Für eine bestimmte (oder zufällige) Anzahl von Bits werden die Indizes zufällig ausgewählt
- Stochastisch bei kontinuierlicher Repräsentation

$$x_i := x_i + z$$

wobei  $z$  eine Zufallsvariable (z.B. Nach  $N(0, \sigma)$ ) Normalverteilung um 0 mit Varianz  $\sigma$

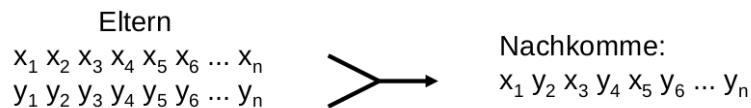
- Mutationsoperator bei Sequenzen
  - Herausnehmen einer Teilsequenz und Einfügen an anderer Stelle
  - Invertiertes Einfügen der Teilsequenz

$$\begin{aligned} \text{Elternteil: } & x_1 \dots x_i \mathbf{x_{i+1}} \dots \mathbf{x_j} x_{j+1} \dots x_n \\ \text{Nachkomme: } & x_1 \dots x_i \mathbf{x_j} \dots \mathbf{x_{i+1}} x_{j+1} \dots x_n \end{aligned}$$

- Spezielle Mutationsoperationen → anwendungsspezifisch

**Rekombination:** Der Nachkomme stammt von zwei oder mehr Eltern ab

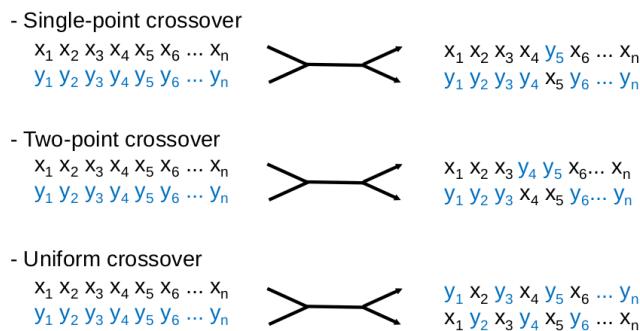
- Diskrete Rekombination



- Intermediäre Rekombination: Sei  $x :=$  Elternteil<sub>1</sub> und  $y :=$  Elternteil<sub>2</sub>, dann ist Nachkomme  $z$  definiert durch

$$z_i := (x_i + y_i)/2$$

- Crossover



#### 11.2.4 Selektion

##### Herausforderung einer guten Selektion

- Selektion:
  - der Population in jeder Iteration
  - der Eltern für jeweilige Erzeugung von Nachkommen (Mating)
- Problem:
  - Genetischer Drift: Individuen vermehren sich zufällig mehr als andere
  - Crowding, Ausreißerproblem: „fitte“ Individuen und ähnliche Nachkommen dominieren die Population

- Vielfalt der Population wird eingeschränkt
- Entwicklung der Individuen (Konvergenz) wird verlangsamt

- Lösung:
  - Unterschiedliche Populationsmodelle und Selektionsmethoden
  - Populationsgröße optimieren

### Populationsmodelle

- Eine einfache Menge: die global Besten entwickeln sich rasch weiter, andere Entwicklungslinien werden unterdrückt
- Inselmodell: die Evolution läuft weitgehend getrennt, nur manchmal werden Individuen ausgetauscht
- Nachbarschaftsmodell: Nachkommen dürfen nur von Individuen erzeugt werden, die in ihrer Nachbarschaft die beste Fitness besitzen
- Nomenklatur
  - $\lambda$  Anzahl von Nachkommen
  - $\mu$  Populationsgröße
  - $f$  Fitness-Funktion

### Populationsmitglieder

- Populationsgröße
  - Soll sie konstant bleiben?  $\mu$
  - Wie viele neu erzeugte Nachkommen?  $\lambda$
- Mitgliederselektion: stochastisch ausgewählt  $\Rightarrow$  die besten  $\mu$  Individuen
  - $(\mu, \lambda)$  Strategie: Auswahl bezieht sich nur auf die Nachkommen (bessere Exploration)
  - $(\mu + \lambda)$  Strategie: Auswahl zieht auch Eltern mit ein (die Besten werden berücksichtigt, geeignet für gut berechenbare Fitness)
- Ersetzungsregel für Mitglieder:
  - Nachkommen ersetzen alle Eltern (Generationen-Modus)
  - Nachkommen ersetzen einen Teil der Eltern
  - Nachkommen ersetzen die ihnen am ähnlichsten Eltern
  - Geographische Ersetzung
  - Bestes Elternindividuum überlebt (Elitist-Modus)
- Daumenregel: Das beste Viertel der Population sollte drei Viertel der Nachkommen erzeugen

### Selektionsmethoden

- Fitness Based Selection

$$P(x) \approx \frac{f(x)}{\sum_{x' \in Pop} f(x')} \text{ mit}$$

$P(x)$  : Wahrscheinlichkeit der Auswahl von Individuum x

$\lambda$  : Anzahl von Nachkommen

$\mu$  : Populationsgröße

$f$  : Fitness-Funktion

abhängig von der Fitnessfunktion z.B. im Laufe der Evolution nur noch geringe Änderungen in  $f(x)$  und damit in  $P(x)$

- Ranking Based Selection

$$P(x) \approx \frac{g(r(x))}{\sum_{x' \in Pop} g(r(x'))} \text{ mit}$$

$P(x)$  : Wahrscheinlichkeit der Auswahl von Individuum x

$r(x)$  : Ranking von x in der aktuellen Population gemäß Fitnessfunktion

$g$  : mit der Güte des Ranges monoton steigende Funktion größer 0

- Exponentiell:  $g(x) = a^{-x}$
- Hyperbolisch:  $g(x) = x^{-a}$
- Die besten k:  $g(x) := \begin{cases} \frac{1}{k} & , x \leq k \\ 0 & , sonst \end{cases}$

weniger abhängig von dem Betrag der Fitness

- Tournament Selection (Tunier)

- wähle für jedes zu erzeugende Individuum  $n (= 2)$  Individuen
- belohne davon, gemäß der Fitness das beste Individuum
- wähle Individuen mit höchster Bewertung

wenig abhängig von dem Betrag der Fitness;

Wahl der Selektionsmethode oft anwendungsspezifisch

## Evolutionstheorien in der Biologie

- Lamarksche Evolution

- Individuen ändern sich (lernen) nach der Erzeugung
- Genotyp wird verändert und anschließend vererbt

- Baldwinsche Evolution

- Individuen ändern sich (lernen) nach der Erzeugung
- Genotyp bleibt unverändert

- Hybride Verfahren

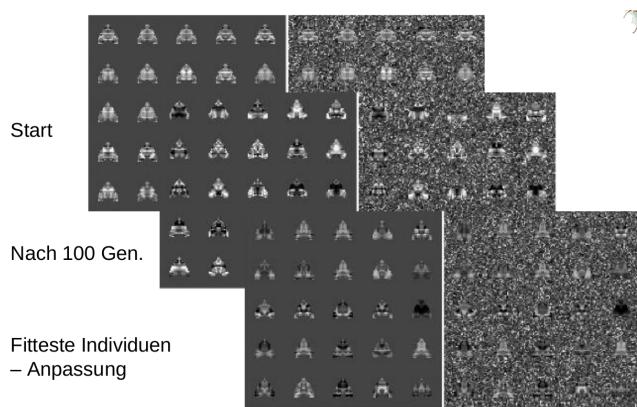
- es gibt veränderbare und fixe Phänotypen

- Anwendung: Suche nach optimalen neuronalen Netzen

### 11.2.5 Anwendungen

- Cybermotten: Wer sich tarnt, wird seltener gefressen. Doch was ist eine gute Tarnung?

- Konstante Population 200
- Fitness – Zeit in der die Motten vom Vogel entdeckt werden
- Rekombination und Mutation (auf dem Graycode der Grauwertpixel)
- Eltern erzeugen mehrere Nachkommen
- Eltern „überleben“ nur eine Generation



- Optionale Steuerung von Laufmaschinen
  - Unterschiedliche Morphologie
  - Beispiel – Erzeugung von Beinsteuerung
  - Steuerungsparameter
    - \* Gelenknummer: *integer* [0; 3]
    - \* Startzeitpunkt: *real* [0;  $t$ ]
    - \* Dauer: *real* [0;  $t$ ]
    - \* Kräfte: *real* [- $max$ ;  $max$ ]
  - Individuum = Sequenz von Instruktionen variabler Länge
  - Mutation
    - \* Micro-Mutation (Veränderung einzelner Werte einer Instruktion)
    - \* Macro-Mutation (Austausch ganzer Instruktionsblöcke)
  - Operatoren (Anpassung)
    - \* Crossover
    - \* Micro-Crossover (Änderung eines Parameters einer Instruktion)
    - \* Macro-Crossover (Austausch einer Instruktion)
    - \* Homologes-Crossover (Austausch von Instruktionen eines speziellen Gelenks)
  - Fitness = Abweichung von einer vorgegebenen Trajektorie
- AIBO – Autonomous Evolution of Dynamic Gaits
  - Ähnlicher Ansatz
  - Fitness = Mittelwert aus drei Läufern; Funktion aus Richtung und Geschwindigkeit
- Snakebot
  - Biologische Inspiration: Seitenwinder Klapperschlange: 5km/h (schnellste Schlange) und die Schwarze Mamba: bis 16km/h
  - Individuum = Steuerung
  - Fitness = Geschwindigkeit in Fortbewegung und Klettern, Befreien aus Hindernissen etc.
  - Adaption: ein oder zwei Gelenke unbeweglich
  - Hinderniserkennung

### 11.3 Künstliche Ontogenese: Entwicklung von Agenten durch künstliche Zellen

**Ontogenese:** struktureller Wandel einer Einheit ohne Verlust ihrer Organisation, Entwicklung des einzelnen Lebenswesens von der befruchteten Eizelle zu erwachsenen Lebewesen.

**Ziel:** Entwicklung von Agenten durch „künstliche Zellen“

#### Methode

- Repräsentation einer Zelle durch Kugel
- Operatoren
  - Wachsen der Kugel im Durchmesser
  - bei Erreichen der maximalen Größen – Teilen in zwei Kugeln, die durch Rotationsgelenk oder ein starres Element verbunden sind
- Jede Kugel besitzt ein neuronales Netz mit Motoneuronen, Interneuronen und Perzeptionsneuronen
- Kodierung als Gleitkomma Genom

#### Genobots – Automatisches Modulares Design

- Individuen = Regeln und Algorithmen für die Erzeugung (Generative Repräsentation)

#### Golem – „life as it could be“

- Individuum = Repräsentation der Morphologie
- Roboter: Stangen (Länge, D, Steifigkeit), Aktuatoren, Gelenktypen, Neuronen (threshold, synaptische Verbindung)
- Beschränkungen
  - Komplexe Strukturen (wie Muskeln) nicht möglich
  - Design Beschränkung
  - Evaluation (Fitness)
  - Übertragbarkeit in die Realität

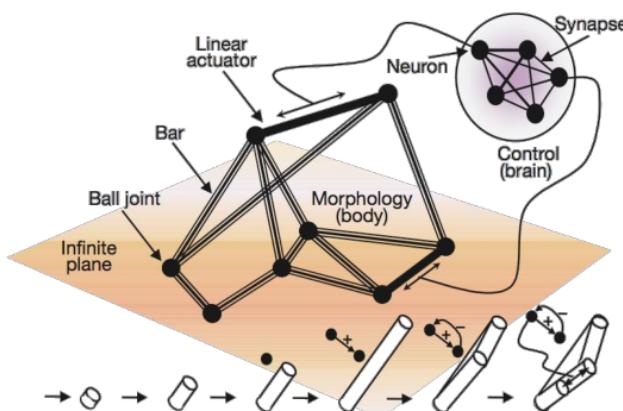


Abbildung 114: Project Golem

## 11.4 Selbstkonfigurierende Systeme – Mutierende Roboter

### 11.4.1 Kurzer Überblick (AIST, Japan) – Selbstkonfiguration = Evolution? Mutation?

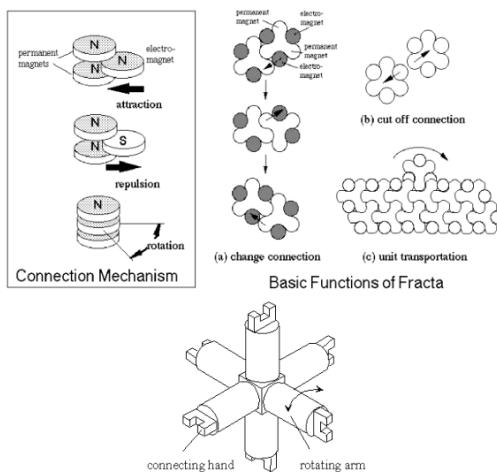


Abbildung 115: Selbstkonfiguration & -reparatur

### 11.4.2 Reales 2D System

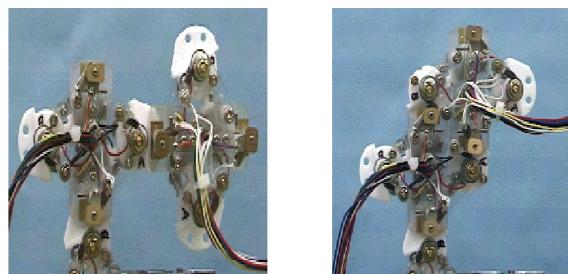
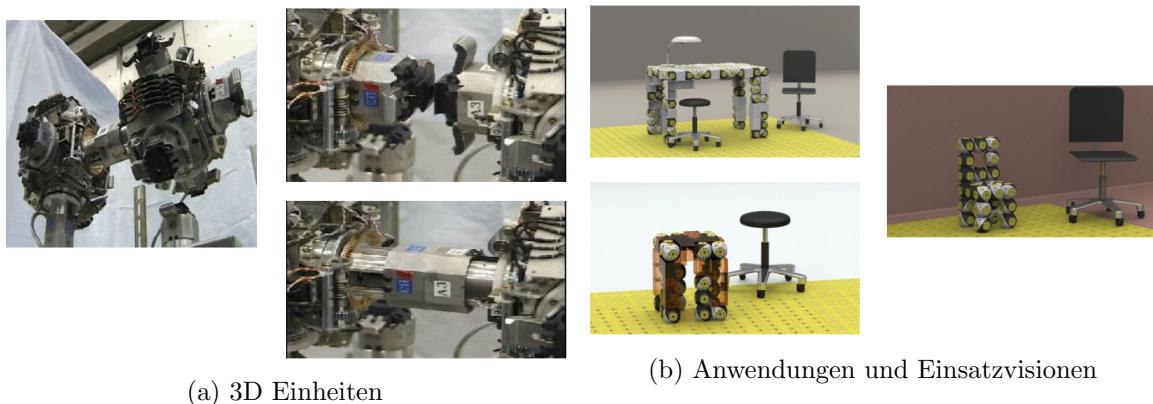


Abbildung 116: 2D Einheiten

### 11.4.3 3D Selbstkonfigurierendes Robotersystem



(a) 3D Einheiten

(b) Anwendungen und Einsatzvisionen

## 12 Anwendungen

- Inspektion von Stahltanks oder Betonbauwerken
- Reinigungsroboter für Glasbauten
- Einsatz in der Schiffindustrie
- Arbeitsmaschinen für die Land- und Forstwirtschaft
- Landminensuche
- Erkundung von Vulkanen
- Planetenerkundung (z.B. Mars Rover *Curiosity*)
- Erkundung von Feuchtgebieten
- Meeresbodenvermessung, Wartungs- und Inspektionsarbeiten auf dem Meeresgrund
- Rohrroboter, Kanalinspektion
- Prothetik (Prothesen, Orthesen)
- Rehabilitation
- Altenpflege (z.B. *Paro Robbe*)
- Entertainmentroboter (z.B. Sony *Aibo*)
- Militärische/Surveillance Anwendungen