

Kognitive Robotik

HK WS 05/06

Mit Dank an

Heiko Bittner
Andreas Dittrich
Caspar Dzikus
Peter Haider
Daniela Mayr
Michael Peters

und viele andere aus dem
Proseminar WS02/03

Bewegung

Hans-Dieter Burkhard
Humboldt-Universität Berlin
Institut für Informatik



Bewegung

- Fortbewegung:
 - Erde-Wasser-Luft-Weltraum
 - Räder, Beine, Flügel, Propeller/Schrauben, Rakete, ...
 - kriechen, laufen, fahren, gleiten, schwimmen,...
- Manipulatoren
- Ausrichtung von Sensoren

Kinematik

- Kinematik (Vorwärts-Kinematik):
 - Wohin läuft er?
 - Wo befinden sich die Glieder?
- Inverse Kinematik (Rückwärts-Kinematik):
 - Wie kommt er dahin?
 - Wie sollen sich die Glieder bewegen?

Vereinfachung:
Vernachlässigung von Masse und Kraft

Hardware

- Energieversorgung
- Antriebe
- Kraftübertragung
- Gelenke
- Fortbewegung
- Manipulation

Antriebsarten

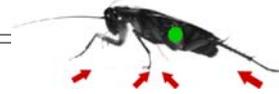
- Motoren
 - Elektro, Dampf, Gas, Sprit, ...
- Hydraulische Antriebe
- Pneumatische Antriebe
- Raketenantrieb
- ...
- Künstliche Muskeln
- Nano-Technik

... was die eigene oder eine fremde Form verändern kann ...

Bionik

TU Berlin (Ingo Rechenberg)
<http://lautaro.bionik.tu-berlin.de/institut/s2foshow/>

Sprawlita



- Inspiration: Insekten
- Pneumatischer Antrieb (semi-autonom)
- Adaption durch Fußkontakt (binäre Sensoren)
- Einfacher Patterngenerator

cdr.stanford.edu.biomimetrics

Bewegung

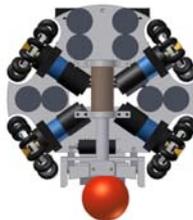
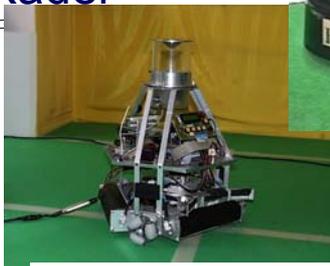
- Manipulatoren
 - stoßen, drücken, greifen, ...
 - ...

Pneumatischer Antrieb
Problem:
Steuerung für
unterschiedliche Aufgaben

Räder

- Angetriebene Räder oder Ketten
- Mitlaufende Räder (Stützräder)
- Steuernde Räder
- Messende Räder (Odometrie)
- Omnidirektionale Räder
- Ketten

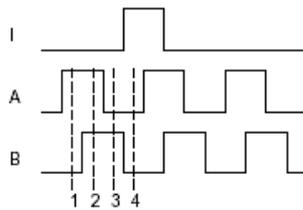
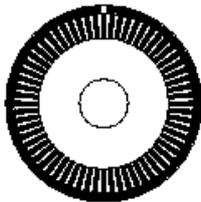
Räder



Geschwindigkeitsmessung

Beispiel:

- optische Messung
- Phasenverschiebung erlaubt Richtungsbestimmung



State	Ch A	Ch B
S ₁	High	Low
S ₂	High	High
S ₃	Low	High
S ₄	Low	Low

Gelenke

meist je Gelenk 1 Freiheitsgrad

aktiv bewegbar: Motoren

passiv: Ausgleich

Stellung abfragen: Körpersensoren

Einsatz: Beine, Arme, Hände, Sensorenbewegung, ...

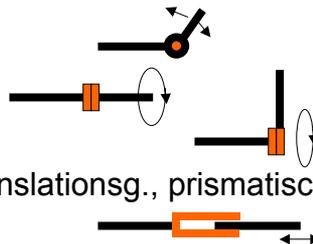
- Gelenkarten

- Rotationsgelenk

- Torsionsgelenk

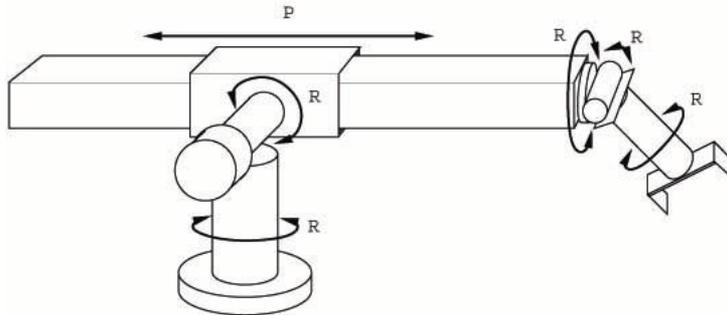
- Revolvergelenk

- Lineargelenk (Translationsg., prismatisches G.)

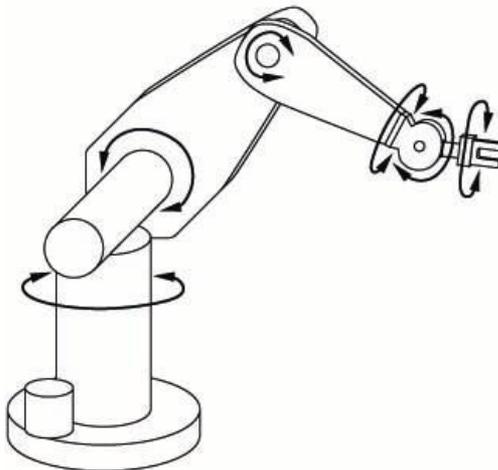


Stanford Manipulator

- P: Prismatisches Gelenk
- R: Rotationsgelenke



Puma



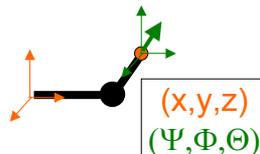
Ansteuerung von Bewegungen

- Modellbasiert
 - Kinematik, inverse Kinematik
- Reproduzieren („teachen“)
 - Protokollieren und reproduzieren eigener Abläufe
- Nachahmen („motion capturing“)
 - Vorgeführte Bewegungen nachvollziehen
- Adaption/Lernen
 - Optimieren gemäß „trial and error“
- Biologische Prinzipien
 - Sensor-Aktor-Schleifen, Patterngenerator etc.

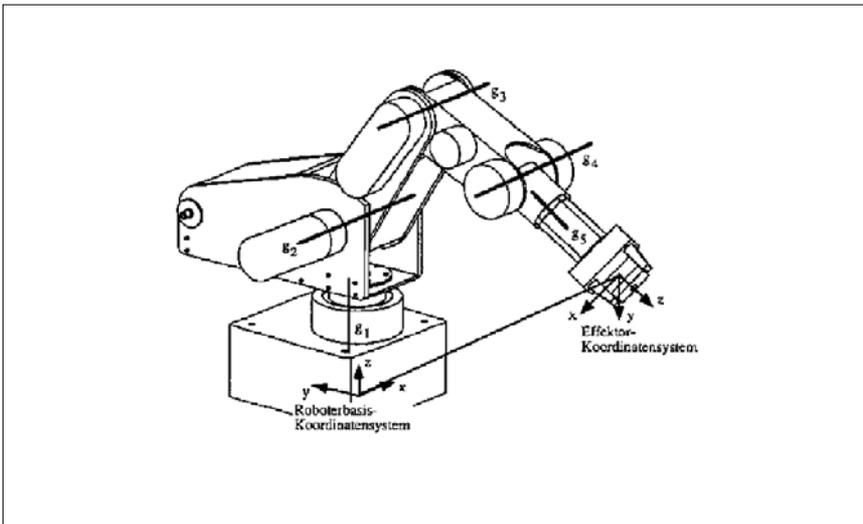
Modellbasiert: (Inverse) Kinematik

Grundlagen: Analytische Geometrie

- Orthonormierte Koordinatensysteme (3D-Raum)
 - Drehung (3 Parameter)
 - Translation (3 Parameter)
 - Homogene Koordinaten
- Position P des Roboters:
 - Ort (3 Parameter)
 - Orientierung (3 Parameter)
- Bewegung des Roboters (Ortsänderung) dP/dt
- Analog: Positionen/Bewegungen der Körperteile



Modellbasiert: (Inverse) Kinematik



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

17

Zustandsparameter (statisch)

Lage-Koordinaten $\mathbf{P} = \{P_1, \dots, P_n\}$ im realen Raum bezogen auf ein festgelegtes Basis-Koordinatensystem für

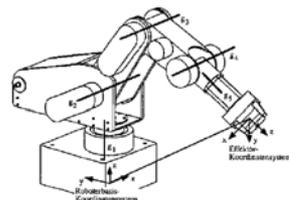
- Roboter
- Körperteile (z.B. Kopf, Schulter, Fuß, ...)

Jeweils Angabe von

- Ort (2 bzw. 3 Werte)
- Orientierung (1 bzw. 3 Werte)

Stellung $\mathbf{S} = \{S_1, \dots, S_n\}$ von Gelenken

- Winkel bzw. Länge



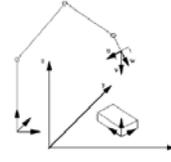
H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

18

Zustandsparameter (statisch)

- Gegenseitige Abhängigkeiten:
„Kinematische Kette“
- Steuerung über Antriebe/Gelenke



$$\textbf{Kinematik:} \quad \textbf{P} = \textbf{f}(\textbf{S})$$

$$\textbf{Inverse Kinematik:} \quad \textbf{S} = \textbf{f}^{-1}(\textbf{P})$$

Umkehrung i.a. nicht eindeutig

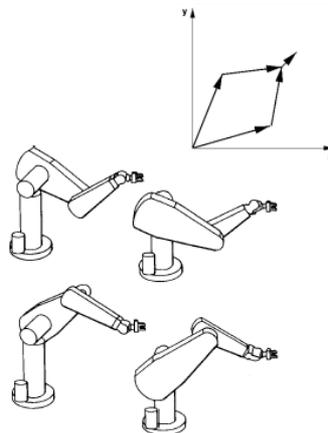
Probleme der Inversen Kinematik

Keine eindeutigen Lösungen:

i.a. führen unterschiedliche

Gelenkstellungen (**S**)

zur gewünschten Lage ($\textbf{P}' \subseteq \textbf{P}$)
eines Effektors



Probleme der Inversen Kinematik

Es kann **Unerreichbare Stellungen** geben:
Punkte, die nicht erreicht werden können

Es kann **Unzulässige Stellungen** geben:
Nicht alle mathematischen Lösungen (prinzipiell erreichbare Stellungen) sind realisierbar, z.B. wegen

- Beschränkung von Gelenkwinkeln
- Kollision mit Objekten bzw. dem Roboter selbst
- Unzulässigen Kräften

Probleme der Inversen Kinematik

Es gibt **keine allgemein anwendbaren Verfahren**:

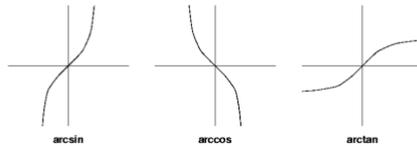
- $f^{-1}(P)$ ist evtl. nicht geschlossen darstellbar

Komplexität:

- Die Berechnung muss
 - mit den verfügbaren Mitteln
 - mit hinreichender Exaktheit
 - in annehmbarer Zeit (ggf. wenigen Millisekunden)erfolgen

Probleme der Inversen Kinematik

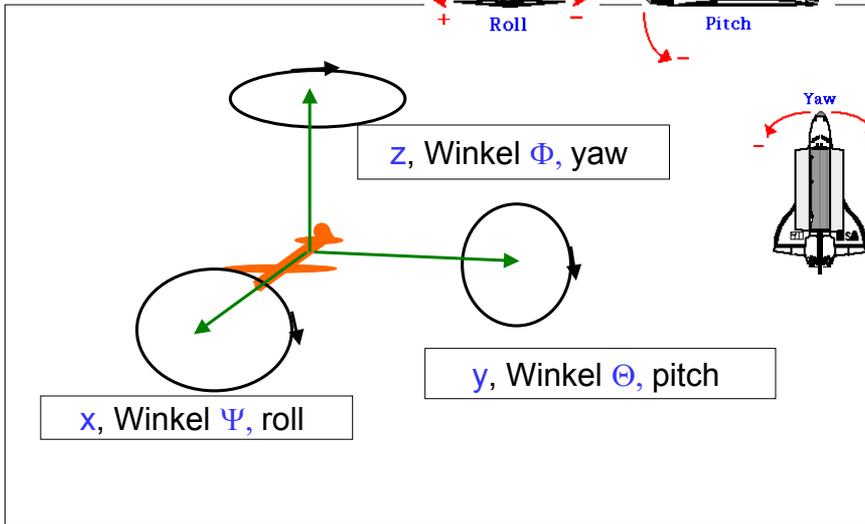
- Ausnutzen spezieller Eigenschaften der Konstruktion
- Drehungen jeweils in einer Ebene (um Gelenkachse)
- Auswahl „einfacher“ Lösungen
- Auswahl „einfacher“ Zusammenhänge
(z.B. bei n Gleichungen für m Unbekannte, wobei häufig $n > m$ wegen Abhängigkeiten zwischen den Elementen der Matrizen)
- \arctan statt \arcsin bzw. \arccos verwenden



Konfiguration

- Konfiguration: Konkrete Werte der Parameter
- Konfigurationsraum: Mögliche Werte
- Pose: Konkrete Werte für Raumparameter P
 - Position
 - Orientierung

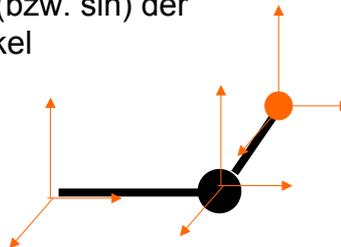
Orientierung



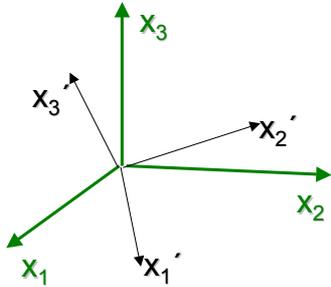
Koordinaten-Transformationen

Positionen berechnen mittels Transformationen

- Rotation im R_n :
 - n-reihige Matrix, Komponenten:
 - Skalarprodukte der Einheitsvektoren der beiden Systeme
 - Darstellbar mittels \cos (bzw. \sin) der eingeschlossenen Winkel
- Translation:
 - Verschiebungsvektor



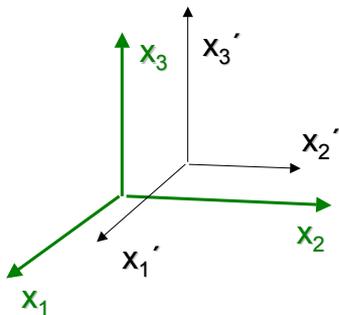
Rotation des Koordinatensystems im R_3



$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

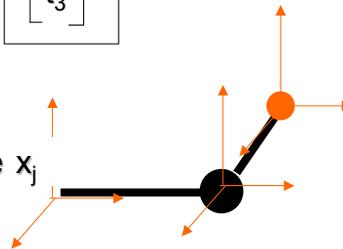
a_{ij} = Skalarprodukt der Einheitsvektoren e_i und e_j
= Cosinus des eingeschlossenen Winkels

Translation des Koordinatensystems im R_3

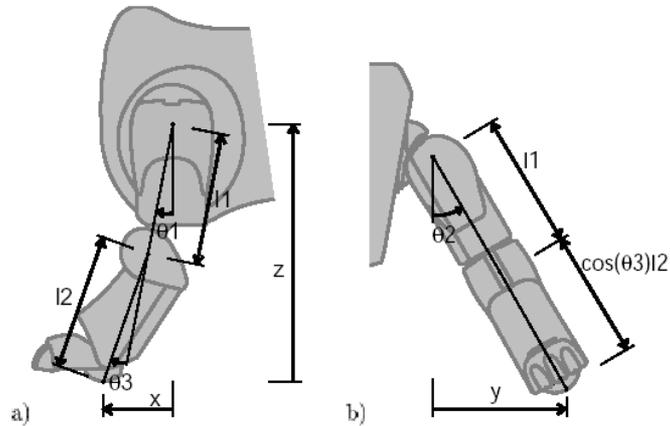


$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

t_i = Verschiebung der Achse x_j



Transformationen



Diplomarbeit Düffert

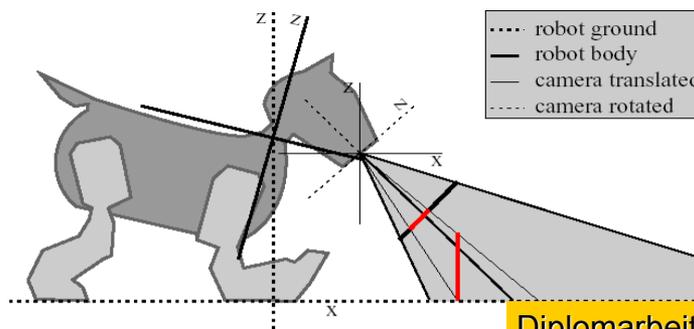
H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

29

Transformationen

- Kamera-Position (Abbildungsebene) berechnen
- Rückrechnung auf reale Welt



Diplomarbeit Jüngel

Figure 3.1: *Coordinate Systems*. Axes: The different 3D coordinate systems in relation to the robot. Gray cone: The angle of view, the optical axis, and the virtual projection plane. Red lines: An object and its projection to the projection plane.

Transformationen

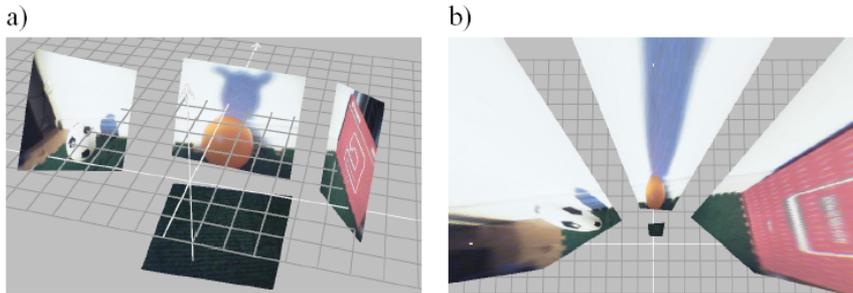


Figure 2.6: Images displayed visualizing the direction of view. The four images were taken from the same position of the robot but with different directions of view of the head: left, ahead, right, down. The white arrows are the axes of the coordinate system of the robot. The gray grid is the plane the robot stands on. a) Images displayed on virtual projection planes. b) The same images projected to the ground.

Diplomarbeit Jünger

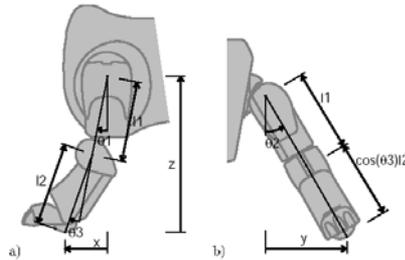
Homogene Koordinaten im R_n

- Zusätzliche Dimension:
- $(n+1)$ -dimensionale Vektoren beschreiben Lage
 $(x_1/w, x_2/w, \dots, x_n/w, w)$
 - repräsentiert (x_1, x_2, \dots, x_n) , $w \neq 0$, beliebig
 - normiert für $w=1$: $(x_1, x_2, \dots, x_n, 1)$
- $(n+1)$ -dimensionale Matrizen beschreiben Translationen und Rotationen

$$\begin{bmatrix} \left[\begin{array}{c} \text{Rot} \\ 0 \ 0 \ \dots \ 0 \end{array} \right] & \left[\begin{array}{c} \text{Tr} \\ 1 \end{array} \right] \end{bmatrix}$$

Homogene Koordinaten

- Sequenz von Koordinatentransformationen als Matrizenmultiplikation
 - Vorwärtskinematik als Resultat
 - Inverse Kinematik als Lösung einer Matrixgleichung



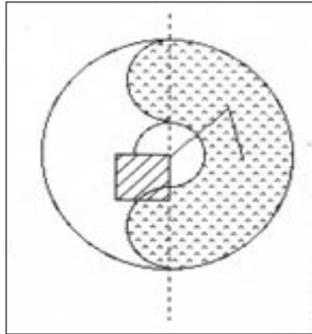
Bewegungen=Konfigurationsübergänge

Übergänge zwischen Konfigurationen können nur innerhalb des Konfigurationsraumes ausgeführt werden:

- Beschränkungen durch Umgebung
- Beschränkungen durch Körperbau

Erreichbarer Bereich

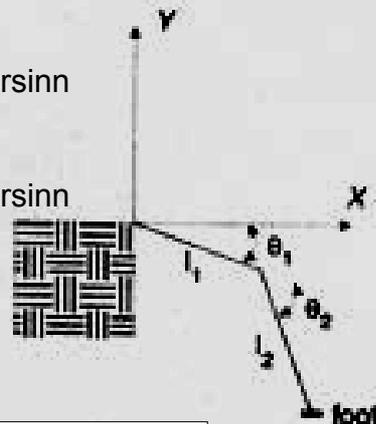
- durch Bein erreichbare Punkte (am Boden)
- beschränkt durch
Gelenkwinkel, Gliedlängen, Behinderungen



Bein (2D)

• Vorwärtskinematik:

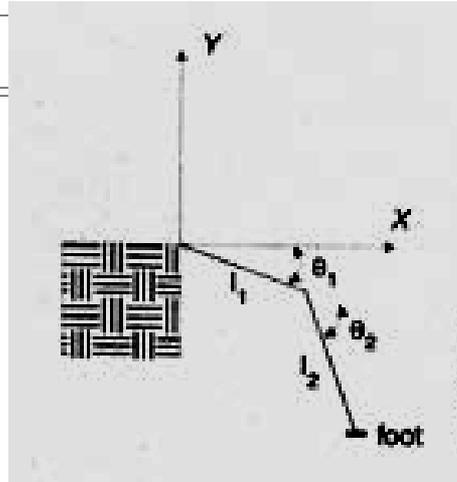
- Drehung um Θ_1 im Uhrzeigersinn
- Translation um l_1
- Drehung um Θ_2 im Uhrzeigersinn
- Translation um l_2



$$\chi = \text{Rot}[-\Theta_1] (\text{Tr}[l_1] (\text{Rot}[-\Theta_2] (\text{Tr}[l_2] (0))))$$

Bein (2D)

- Vorwärtskinematik:



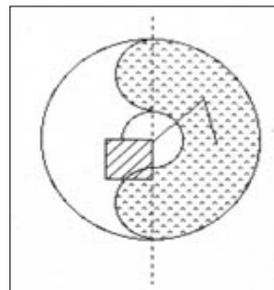
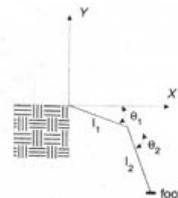
$$\begin{bmatrix} x \\ y \end{bmatrix} = l_1 \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} + l_2 \begin{bmatrix} \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) \end{bmatrix}$$

Bein (2D)

- Rückwärtskinematik:
(hier mit Kosinussatz)

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}$$

$\cos(\theta_1)$ z.B. aus Formel für
Vorwärtskinematik bestimmen



Erreichbarer Bereich

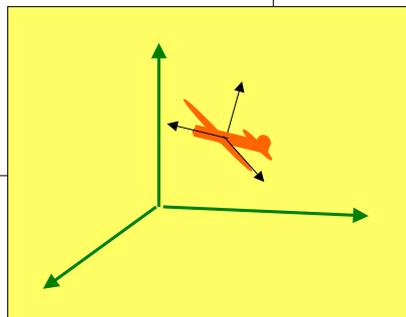
zusätzliche Gelenke bei Behinderungen verwenden

(Effektive) Freiheitsgrade (DOF)

Möglichkeiten der Positionierung im Raum

Lage, Orientierung:

- 3 in der Ebene
- 6 im Raum



Freiheitsgrade



AIBO hat
„20 Freiheitsgrade“
(DOF)

gemeint sind 20 aktiv ansteuerbare Gelenke:
20 dimensionaler **Konfigurationsraum** (der Gelenkwinkel)

Zusammenhang

Umgebungsraum (6DOF) und Konfigurationsraum:

Kinematik:

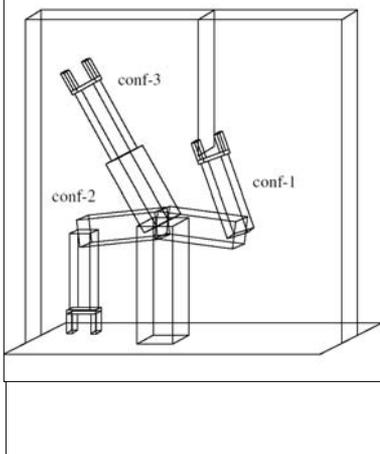


Inverse Kinematik:



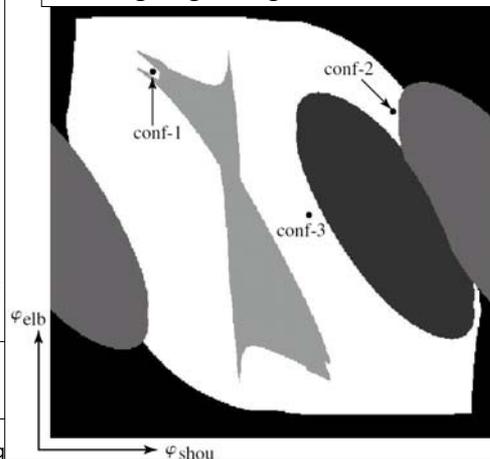
Transformation

Koordinaten: Raum (2D)



Koordinaten: Gelenkstellungen

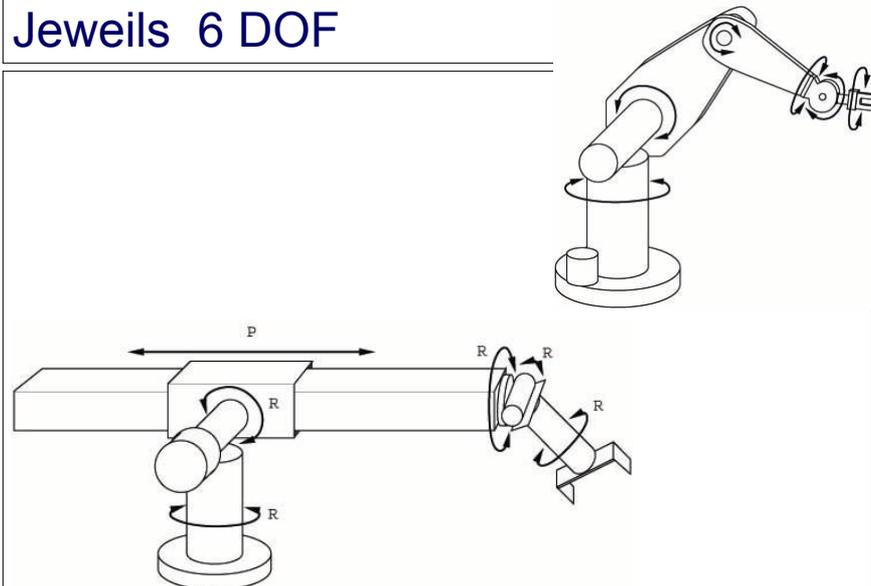
Freie Bereiche entsprechend
Bewegungsmöglichkeiten



Freiheitsgrade (DOF)

- Bezogen auf Umgebung:
 $\text{DOF}_{\text{effektiv}} = \text{Anzahl der möglichen unabhängigen Bewegungs-/Orientierungsrichtungen}$
- Bezogen auf Körper:
 $\text{DOF}_{\text{steuerbar}} = \text{Anzahl der (steuerbaren) Gelenkachsen bzw. Bewegungsrichtungen}$
- Falls $\text{DOF}_{\text{steuerbar}} < \text{DOF}_{\text{effektiv}}$:
Probleme bei der Steuerung (nichtholonome Systeme)

Jeweils 6 DOF



AIBO von SONY

20 Motoren:

- 3 pro Bein
- 3 für den Kopf
- 2 für den Schwanz
- 1 für das Maul
- 1 pro Ohr

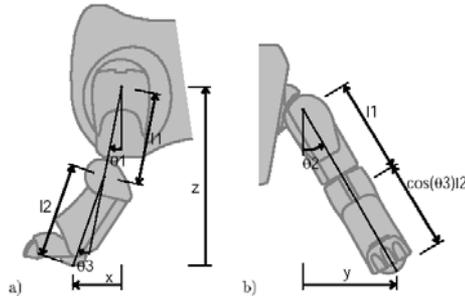


3 Motoren pro Bein



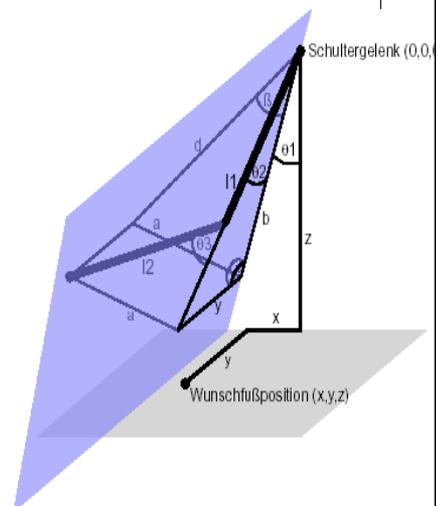
Vorwärtskinematik

- Stellung des Fußes relativ zur Schulter
- Berechnet als Sequenz von Koordinatentransformationen durch Matrizenmultiplikation (homogene Koordinaten)



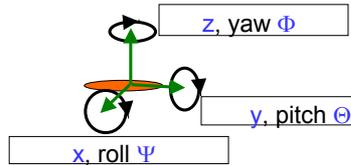
Rückwärtskinematik

- Notwendige Gelenkwinkel zum Erreichen einer vorgegebene Stellung des Fußes relativ zur Schulter
- Berechnet als Sequenz von Koordinatentransformationen durch Matrizenmultiplikation (homogene Koordinaten)



Ausführung von Drehbewegungen

- xyz-System: Bezug auf Originalsystem



- zx'z''-System:

Drehung um z-Achse : Winkel Φ

Drehung um neue x-Achse (x') : Winkel Ψ

Drehung um z-Achse (z'') : Winkel Θ

- Angabe der Rotations-Achse

• und weitere

Reihenfolge von Bewegungen

- Bewegungen um Achsen an unterschiedlichen Punkten (z.B. unterschiedliche Gelenke von Armen bzw. Beinen), können prinzipiell in beliebiger Reihenfolge realisiert werden (soweit der Raum frei ist).
- Bei Bewegungen am gleichen Punkt ist die Reihenfolge der Drehungen wichtig.
(Matrizenmultiplikationen sind nicht kommutativ.)
Jede mögliche Bewegungsfolge kann dabei durch eine einzige Drehung realisiert werden.

Rotationen in einem Punkt

Wenn der Körper in jeder Richtung (!) frei beweglich ist, können beliebige Drehungen um die Achsen durch **eine einzige** Drehung (1 Parameter) um eine Achse (2 Parameter) realisiert werden:

„Eigenachse“

Berechnung:

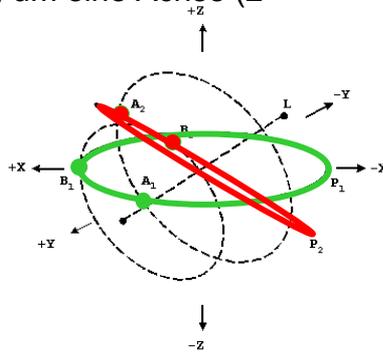
Lösung des Eigenwertproblems

$$R \mathbf{e} = \mathbf{e}$$

R = Rotationsmatrix

\mathbf{e} = Einheitsvektor in Richtung Eigenachse

Lösbarkeit für orthogonale Koordinatensysteme gesichert



Dynamisches Modell

- für Bewegung von Körperteilen relativ zum Körper.
- für Bewegung des Roboters relativ zur Umgebung.
- Steuerparameter: Bewegung von Gelenken, Rädern, ...

- Kinematik der Bewegungen:

$$dP/dt = df(S)/dt = df/dS \cdot dS/dt$$

(Positionsänderung P aus Änderung der Steuer-Parameter S bestimmen)

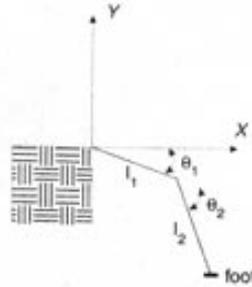
- Inverse Kinematik der Bewegungen:

$$dS/dt = df^{-1}(P)/dt = df^{-1}(P)/dP \cdot dP/dt$$

(Änderung der Steuer-Parameter aus gegebener Bewegungsanforderung bestimmen)

$$P = f(S)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = l_1 \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} + l_2 \begin{bmatrix} \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) \end{bmatrix}$$



$$P = f(S) = \begin{bmatrix} l_1 \cos(\theta_1) & l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) & l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

Jacobi-Matrix $df(S)/dS$

$$P = f(S) = \begin{bmatrix} l_1 \cos(\theta_1) & l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) & l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$df(S)/dS = \begin{bmatrix} -l_1 \sin(\theta_1) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

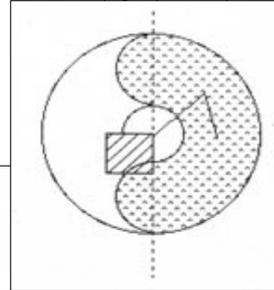
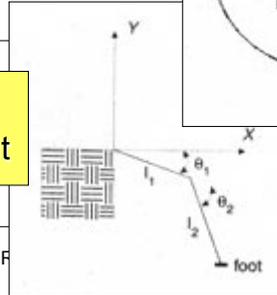
Singularitäten

$$\left| \frac{df(\mathbf{S})}{d\mathbf{S}} \right| = \begin{vmatrix} -l_1 \sin(\Theta_1) & -l_2 \sin(\Theta_1 + \Theta_2) \\ l_1 \cos(\Theta_1) & l_2 \cos(\Theta_1 + \Theta_2) \end{vmatrix}$$

$$= l_1 l_2 |\sin(\Theta_2)|$$

$$= 0 \quad \text{für } \Theta_2 = 0, \pi \text{ oder } -\pi$$

Singularitäten entsprechen eingeschränkter Beweglichkeit



Fahrende Roboter

- Angetriebene Räder oder Ketten
- Mitlaufende Räder (Stützräder)
- Steuernde Räder
- Messende Räder (Odometrie)

Idealisierende Annahmen:

Rad läuft geradeaus (rechtwinklig zur Rad-Achse)

Vorwärtsbewegung pro Umdrehung: $2r\pi$ bei Radius r

Messung (Odometrie) an nicht getriebenem, unbelastetem Rad

Bewegung in der Ebene

- 3 Freiheitsgrade in der Ebene: (x,y,θ)
 - 2 bzgl. Koordinaten: (x,y)
 - 1 bzgl. Ausrichtung: θ
 - i.a. nicht unabhängig voneinander steuerbar
- Positionsänderung bei Bewegung mit Geschwindigkeit $V(t)$ in Richtung $\theta(t)$

$$x(t) = \int_0^t V(t) \cos [\theta(t)] dt$$

$$y(t) = \int_0^t V(t) \sin [\theta(t)] dt$$

$$\theta(t) = \int_0^t \omega(t) dt$$

Normierung:

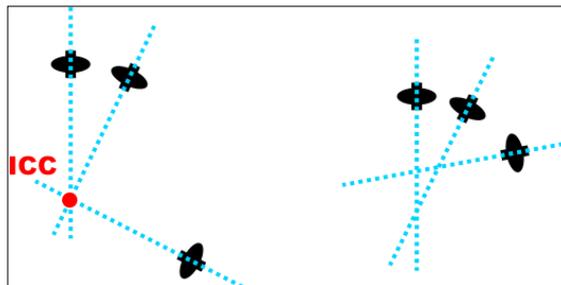
$\theta = 0$ in x-Richtung

Rotationszentrum

Gemeinsamer Schnittpunkt der Achsen der Räder mit Bodenkontakt:

„Rotationszentrum“

ICC = instantaneous center of curvature



Rotationszentrum

Roboter-Bewegung

(als Resultat der Bewegung der Räder):

- kreisförmig um das Rotationszentrum
- geradeaus: ICC unendlich weit

Rotationszentrum verlagern, z.B. durch:

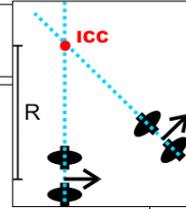
- veränderte Achsenstellung
- unterschiedliche Geschwindigkeiten
- . . .

Constraints

- Rollende Bewegung möglich,
falls für alle Räder mit Bodenkontakt gilt:
 - 1) Rotationszentrum (ICC) existiert
(d.h. die Achsen haben gemeinsamen Schnittpunkt)
 - 2) Geschwindigkeiten der (angetriebenen) Räder
konsistent
- Sonst::
 - Roboter verliert Bodenhaftung
 - Roboter schlittert, Bewegung unberechenbar

Kinematik

- Berechnung der Position des Roboters nach Zeit δt
- Roboter rotiert um den Winkel $\omega \delta t$ um den ICC



$$\text{ICC} = [x - R \sin(\theta), y + R \cos(\theta)]$$

Position zur Zeit t : (x, y, θ)

Position zur Zeit $t + \delta t$:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \delta t) & -\sin(\omega \delta t) & 0 \\ \sin(\omega \delta t) & \cos(\omega \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - \text{ICC}_x \\ y - \text{ICC}_y \\ \theta \end{bmatrix} + \begin{bmatrix} \text{ICC}_x \\ \text{ICC}_y \\ \omega \delta t \end{bmatrix}$$

Inverse Kinematik

- Welche Geschwindigkeiten \mathbf{V} sind notwendig für eine
 - Positionsänderung von $\mathbf{P}(0) = [0, 0, 0]$
nach $\mathbf{P}(t) = [x(t), y(t), \theta(t)]$ bzw.
 - eine Bewegung längs vorgegebener Trajektorie?
- Auflösung nach \mathbf{V} in Abhängigkeit von \mathbf{P}
- bzw. Angabe als Funktion $\mathbf{P}(t)$.

$$x(t) = \int_0^t V(t) \cos[\theta(t)] dt$$

$$y(t) = \int_0^t V(t) \sin[\theta(t)] dt$$

$$\theta(t) = \int_0^t \omega(t) dt$$

Holonome Systeme

Hindernisse im Raum und in der Beweglichkeit des Roboters schränken den Konfigurationsraum ein.

D.h. bestimmte Constraints müssen erfüllt sein:

$$C(P(t), dP/dt, d^2P/dt^2, d^3P/dt^3, \dots) = 0$$

Falls sich die Constraints allein in der Form

$$C(P(t)) = 0$$

ausdrücken lassen, heißt das System holonom.

Holonome Roboter sind aufwendiger konstruiert.

Wegplanungsaufgaben können einfacher gelöst werden.

Differentialantrieb

- Räder auf einer Achse
- Geschwindigkeit unterschiedlich

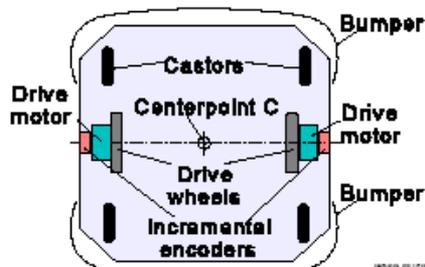
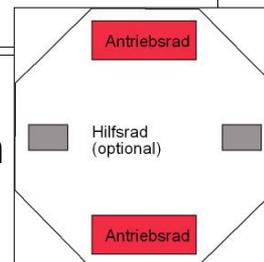


Figure 1.6: A typical differential-drive mobile robot (bottom view).

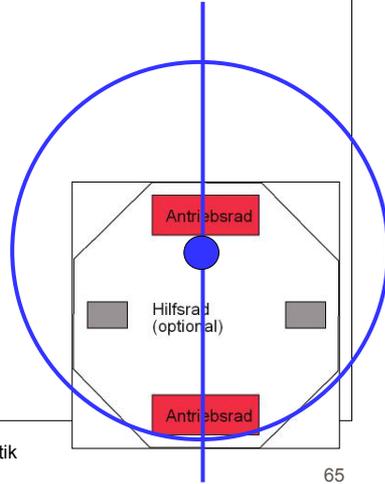
hohe Empfindlichkeit bzgl. Richtung



Differentialantrieb

Rotationszentrum (ICC) auf gemeinsamer Achse,
genaue Lage entsprechend Geschwindigkeiten v_l, v_r

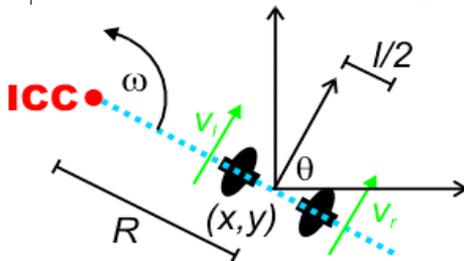
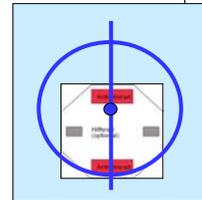
- $v_l = v_r$ Vorwärtsbewegung
- $v_l = -v_r$ Drehung auf der Stelle
- sonst: bogenförmige Bewegung



Differentialantrieb: Vorwärtskinematik

Geschwindigkeiten v_l, v_r der Räder (**Steuerparameter**)
bestimmen Position (x,y) und Ausrichtung θ
Berechnung benutzt ICC

- $v_l = v_r$, $R = \infty$ Vorwärtsbewegung
- $v_l = -v_r$, $R = 0$ Drehung auf der Stelle



$$\omega(R + l/2) = v_r$$

$$\omega(R - l/2) = v_l$$

$$R = \frac{l}{2} \frac{(v_l + v_r)}{(v_r - v_l)}$$

$$\omega = \frac{v_r - v_l}{l}$$

Differentialantrieb: Vorwärtskinematik

$$\begin{aligned}x(t) &= \int_0^t V(t) \cos(\Theta(t)) dt \\y(t) &= \int_0^t V(t) \sin(\Theta(t)) dt \\ \Theta(t) &= \int_0^t \omega(t) dt\end{aligned}$$

- Gleichungen für differentialgetriebenen Roboter:

$$\begin{aligned}x(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt + x_0 \\y(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt + y_0 \\ \theta(t) &= \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt + \theta_0\end{aligned}$$

Differentialantrieb: Vorwärtskinematik

Kein holonomes System:

Gleichungen (Constraints) für differentialgetriebenen Roboter nicht ohne die Geschwindigkeiten formulierbar

$$\begin{aligned}x(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt + x_0 \\y(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt + y_0 \\ \theta(t) &= \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt + \theta_0\end{aligned}$$

Differentialantrieb: Inverse Kinematik

Bewegungen berechnen zum Erreichen einer Position
(Ausgehend von Vorwärtskinematik)

$$x(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt + x_0$$

$$y(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt + y_0$$

$$\theta(t) = \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt + \theta_0$$

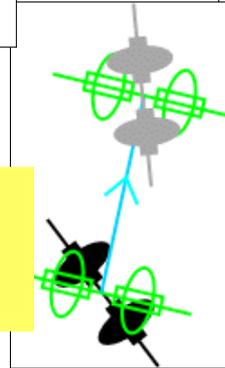
Problem: Gleichungen i.a. nicht auflösbar
aber Lösungen für Spezialfälle

Viele verschiedene Lösungen
(Fahrmöglichkeiten)

z.B. : drehen-fahren-drehen

H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung



Differentialantrieb: Inverse Kinematik

$$x(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt + x_0$$

$$y(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt + y_0$$

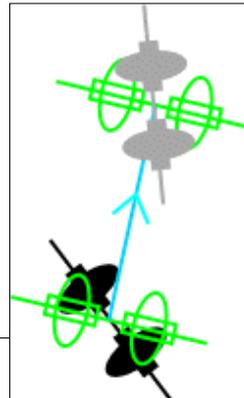
$$\theta(t) = \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt + \theta_0$$

$$v = v_l = -v_r :$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + 2v\delta t/l \end{pmatrix}$$

$$v = v_l = v_r :$$

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta)\delta t \\ y + v \sin(\theta)\delta t \\ \theta \end{pmatrix}$$



H.D.
Win

70

Differentialantrieb: Vorteile/Nachteile

- Drehen auf der Stelle möglich
- *** Grundsätzlich: Einfache Konstruktion und Steuerung
- Bewegung in Richtung Antriebsachse nicht möglich
- Stützräder erforderlich
- Richtungs-Empfindlichkeit bei
 - Unebenheiten
 - Rutschen(Verbesserung durch engere Radstellung)
- *** Tatsächlich: Ausgleich durch Steuerung kompliziert
- Antriebsräder können Bodenkontakt verlieren
- *** Tatsächlich: Stützräder müssen ausgleichen können

Synchroner Antrieb (SynchroDrive)

- Alle Räder synchron angetrieben
- Richtung aller Räder synchron steuerbar

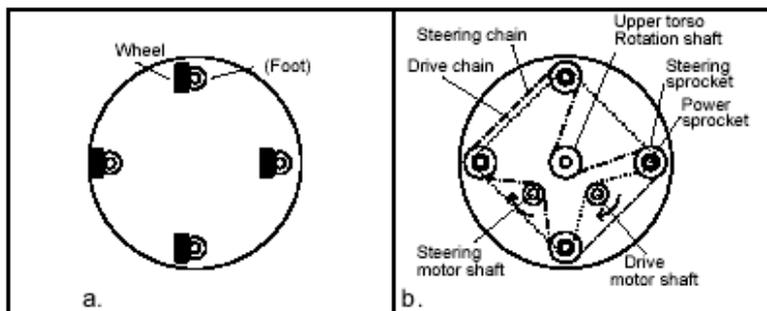


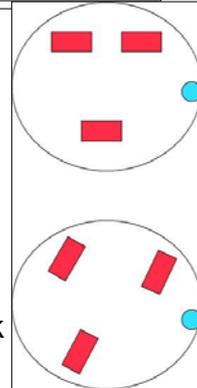
Figure 1.9: A four-wheel synchro-drive configuration: a. Bottom view. b. Top view.
(Adapted from Holland [1983].)

Synchroner Antrieb (SynchroDrive)

- Alle Räder synchron angetrieben
- Typischerweise 3 Räder
- Richtung aller Räder synchron steuerbar
d.h. Fahrtrichtung ist direkt steuerbar

Direkte Steuerbarkeit für Lauf und Drehung:
einfache Vorwärtskinematik/Inverse Kinematik
ICC unendlich weit (in Achsenrichtung)

Steuerparameter: Geschwindigkeit v der Räder
Ausrichtung α der Räder



Synchroner Antrieb Vorteile/Nachteile

- Hohe Beweglichkeit
- Drehung um eigene Achse möglich
- Aber: Roboterkörper zeigt nach Drehung noch in alte Richtung
 - zusätzliche Drehung des Körpers notwendig
(weiterer Konstruktionsaufwand)
 - Oder: Symmetrischer Aufbau bzgl. aller Richtungen,
d.h. 2 Freiheitsgrade (holonomes System)
- Problem bei Unebenheiten für synchronen Lauf
(Räder müssen unterschiedliche Wege zurücklegen)

Gesteuerte Räder: Dreirad usw.

Einige Räder steuerbar, andere fest
(Dreirad, Zweirad, Waggon)

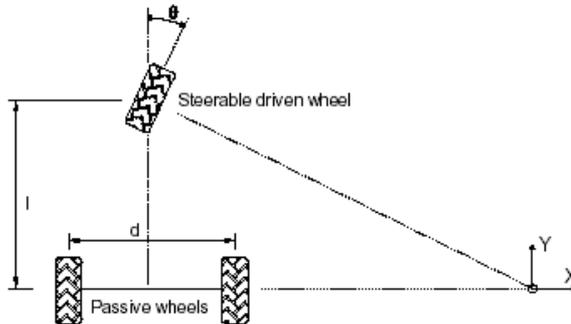


Figure 1.7: Tricycle-drive configurations employing a steerable driven wheel and two passive trailing wheels can derive heading information directly from a steering angle encoder or indirectly from differential odometry [Everett, 1995].

Gesteuerte Räder: Dreirad usw.

ICC liegt auf Achse der fixierten Räder

Lage des ICC abhängig von Achse der gesteuerten Räder

Steuerparameter: Geschwindigkeit v der Räder
Ausrichtung α der Räder

Drehung auf der Stelle nur möglich bei $\alpha = 90^\circ$

Komplizierte inverse Kinematik

Instabilität bei engen Kurven

Ackermann-Steuerung: Automobil

Frontsteuerräder getrennt steuerbar.

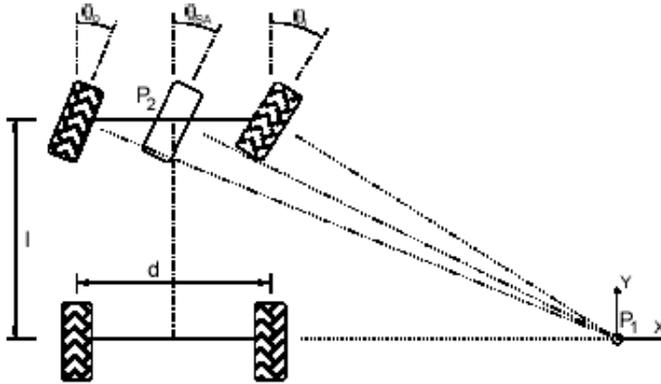


Figure 1.8: In an Ackerman-steered vehicle, the extended axes for all wheels intersect in a common point. (Adapted from [Byrne et al., 1992].)

Ackermann-Steuerung: Automobil

Frontsteuerräder getrennt steuerbar.

- ICC liegt auf Achse der fixierten Hinterräder
- ICC abhängig von Achsen der gesteuerten Vorderräder
- Vorderräder mit unterschiedlichen Winkeln gesteuert
- **Steuerparameter:**
 - Geschwindigkeit v_1 eines Rades
(andere Geschwindigkeiten dadurch festgelegt)
 - Ausrichtung α_1 eines Steuerrades
(α_2 des anderen Steuerrades dadurch festgelegt)

Weitere Freiheitsgrade für hohe Belastung

- Schrägstellung der Räder
- Ausgleich von Unebenheiten

Komplizierte Kinematik

Komplizierte Manöver (Einparken!)

Steuerung fahrender Roboter

i.a. nur 2 steuerbare Parameter für 3 DOF

- Differentialantrieb: Geschwindigkeiten v_l und v_r
- Andere Antriebe: Geschwindigkeit v und Winkel α

z.T. zusätzliche Abhängigkeiten zwischen v und Winkel α
(z.B. bei Ackermannsteuerung)

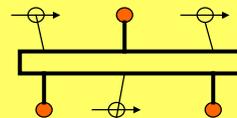
Dadurch

- komplexe inverse Kinematik
- komplexe Manöver zum Erreichen einer Position

Laufroboter

- Unwegsames Terrain (z.B. Waldboden)
- Hindernisse (z.B. Treppen)
- unbekannte Geländebeschaffenheit
- Greiffüße

Stabiler Lauf:
Stets 3 Beine am Boden



Lauron III (Laufender Roboter, neuronal gesteuert)
FZI Karlsruhe

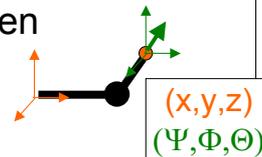
Wieviel Beine ?

- Gerade Anzahl ermöglicht Ausnutzung der Symmetrie
- 4 Beine ermöglichen stabile Gangart
 - dabei muss das Gewicht verlagert werden
 - es kann nur je ein Bein angehoben werden
- 6 Beine ermöglichen stabilen Lauf mit
 - gleichzeitiger Bewegung von 3 Beinen
 - ohne Verlagerung des Gewichts
- Patterngeneratoren zur Erzeugung von einfachen Laufmustern auch bei mehr Beinen

Gelenk-Kombinationen

- Mehrere Freiheitsgrade ermöglichen

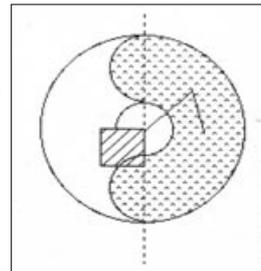
- 6 Freiheitsgrade für Objekt im Raum
(Position, Orientierung: je 3)



- Rotationsgelenke für Drehung erforderlich

- Erreichbarer Bereich:

- durch Bein erreichbare Punkte (am Boden)
- beschränkt durch Gelenkwinkel, Gliedlängen, Behinderungen
- zusätzliche Gelenke bei Behinderungen



Vorwärtsbewegung

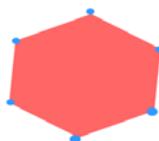
- Verlagerung des Roboters
(modelliert z.B. als Bewegung seines Schwerpunktes)
- Laufmuster:
Vollständiger Zyklus von Beinbewegungen
- Beinbewegung
Abheben, Vorwärtsbewegen, Aufsetzen
- Duty-Faktor:
Prozentualer Zeitanteil des Bodenkontakts der Beine
z.B. Trab:
von 4 Beinen jeweils 2 am Boden, d. h. Duty-Faktor = 0,5

Laufroboter

Statische Stabilität

Roboter kann jederzeit stehen bleiben

Schwerpunkt innerhalb der konvexen Hülle der Bodenkontakt-Punkte („Stütz-Polygon“)



(aber: wie groß sind die Füße)

Laufroboter

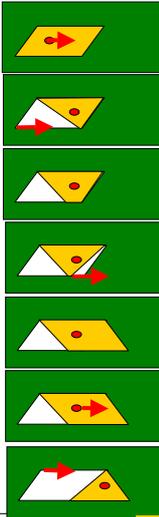
- Statischer Gang
 - langsam
 - unebene Flächen
 - nicht alle Gliedmaßen dienen der Fortbewegung
 - außerhalb der Bewegung stabil
 - geringe Anforderung an die Verarbeitungsgeschwindigkeit
 - Keine Regelung zur Stabilität erforderlich
 - mehr Gliedmaßen = mehr Material
 - Minimale Anzahl Beine: 4

Stabiler Lauf eines 4-beinigen Roboters

Mindestens 3 Beine stets am Boden.

1. Ein Bein anheben: Schwerpunkt muss innerhalb des Dreiecks der am Boden stehenden Beine sein.
2. Bein weiter vorn aufsetzen: 4 Beine am Boden.
3. Schwerpunkt ggf. nach vorn verlagern, um neues Vorsetzen zu ermöglichen.

Stabiler Lauf eines 4-beinigen Roboters



R1. Schwerpunkt nach vorn verlagern

R2. Rechtes Hinterbein in der Luft

R3. Rechtes Hinterbein am Boden

R4. Rechtes Vorderbein in der Luft

R5. Rechtes Vorderbein am Boden

L1. Schwerpunkt nach vorn verlagern

L2. Linkes Hinterbein in der Luft

H.D.Burkhard, HU
Winter-Semester 2

Und so weiter ... Bei L5. ist der Roboter fast eine Körperlänge vorwärts gekommen

Laufroboter

Dynamische Stabilität

Schwerpunkt auch außerhalb des Stütz-Polygons

Kontrollierte Bewegung verhindert Umfallen

Laufroboter

- Dynamischer Gang
 - schnell
 - glatte Flächen
 - Alle Gliedmaßen dienen gleichermaßen der Fortbewegung
 - außerhalb der Bewegung instabil
 - hohe Anforderung an Steuerung und Regelung
 - schnelle Verarbeitung von Sensordaten notwendig
 - weniger Gliedmaßen = weniger Material
 - funktioniert schon mit einem Bein

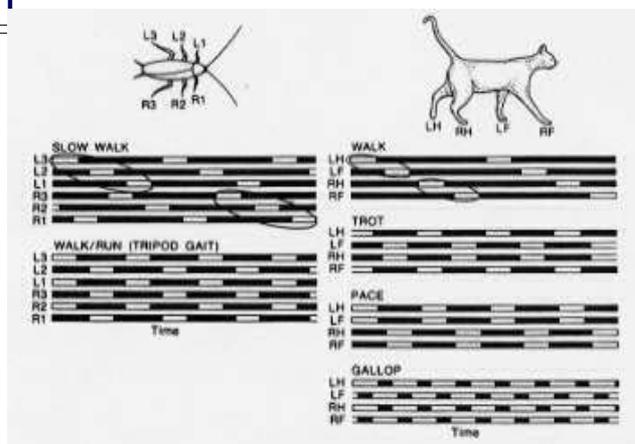
Gangarten

- Lauf (walk)
- Passgang (amble)
- Trab (trot)
- (rack)
- Langsamer Galopp (canter)
- Galopp (galopp)
- (ricochet)

Gangarten

- Trade-off zwischen
 - Geschwindigkeit
 - Energie
 - Stabilität
 - Berechenbarkeit
- 2 Phasen für jedes Bein:
 - Stützphase (Bodenkontakt)
 - Gelenkwinkel bestimmt durch Körperansatz/Aufsatzpunkt
 - Transferphase (freie Bewegung)
 - Gelenkwinkel bestimmt durch nächsten Aufsatzpunkt
 - Zyklus: Aufeinanderfolge beider Phasen
 - Periodische Bewegung im freien Gelände

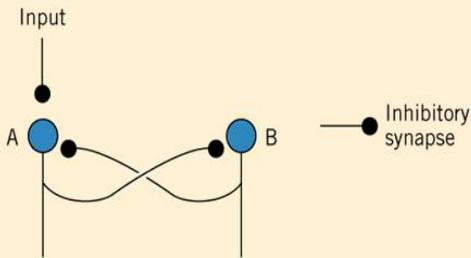
Gangarten



Duty-Faktor:

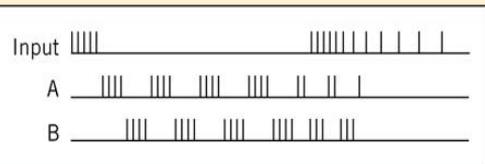
Prozentualer Zeitanteil des Bodenkontakts der Beine

Erzeugung rhythmischer Impulse



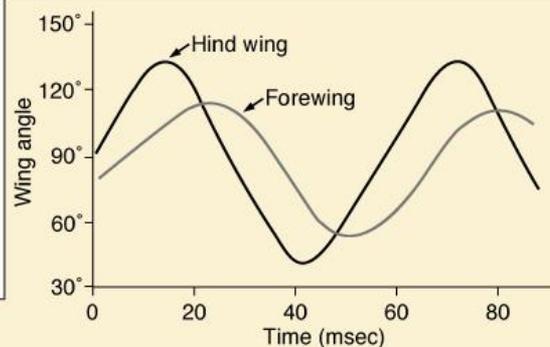
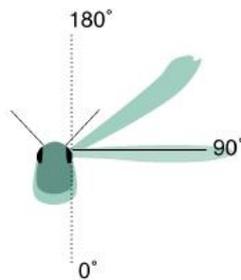
Zyklus:

1. A wird aktiv und hemmt B
2. A ist erschöpft, B nicht mehr gehemmt
3. B wird aktiv und hemmt A
4. B ist erschöpft, A nicht mehr gehemmt

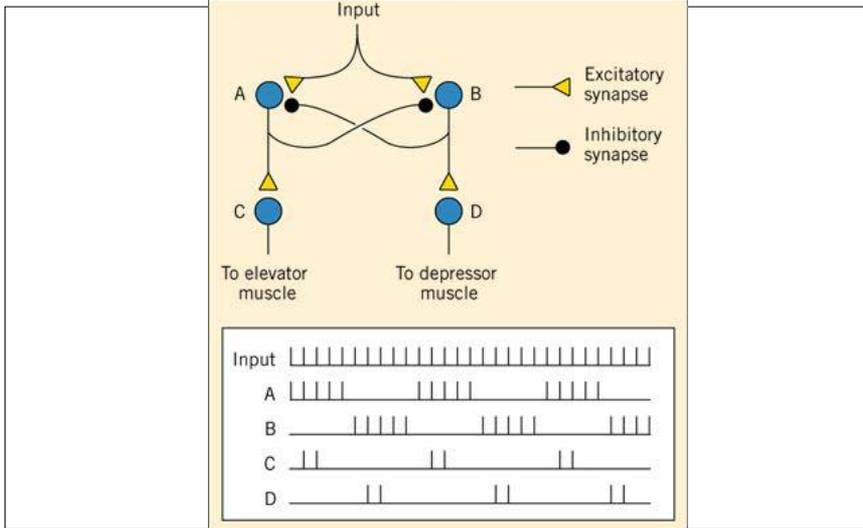


Tatsächlich sind natürliche Formen komplexer:
Ansteuerung von Muskeln.
Verarbeitung sensorischer Impulse
...

Rhythmische Impulse: Heuschrecke



Erzeugung rhythmischer Impulse



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

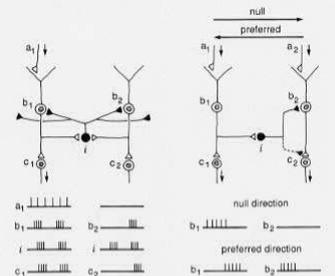
Vorlesung Kognitive Robotik
Bewegung

95

Erzeugung rhythmischer Impulse

Churchland and Sejnowski (1992):
The Computational Brain,
MIT Press (pbk, 1994)

A. RHYTHM GENERATION C. DIRECTION SELECTIVITY



B. SPATIAL CONTRAST

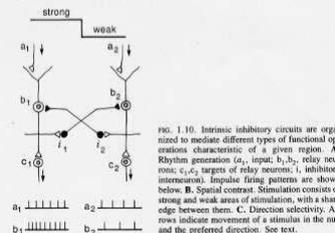


FIG. 1.10. Intrinsic inhibitory circuits are organized to mediate different types of functional operations characteristic of a given region. A. Rhythm generation (a_1 , input; b_1, b_2 , relay neurons; c_1, c_2 , targets of relay neurons; i , inhibitory interneuron). Impulse firing patterns are shown below. B. Spatial contrast. Stimulation consists of strong and weak areas of stimulation, with a sharp edge between them. C. Direction selectivity. Arrows indicate movement of a stimulus in the null and the preferred direction. See text.

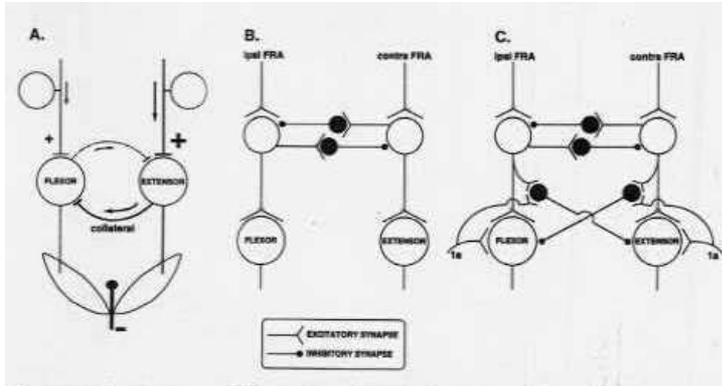
H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive
Bewegung

16

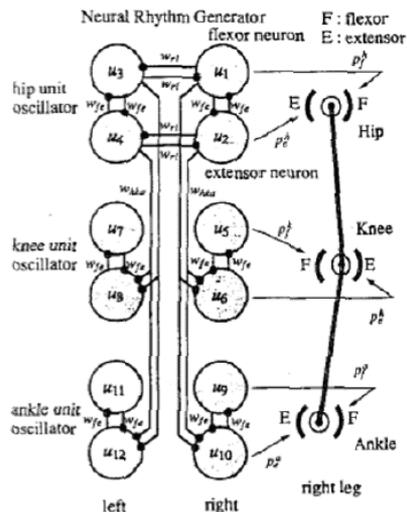
Central pattern generator

Grundlage: Oszillation neuronaler Aktivitäten mittels aktivierender bzw. unterdrückender Verbindungen



Central pattern generator

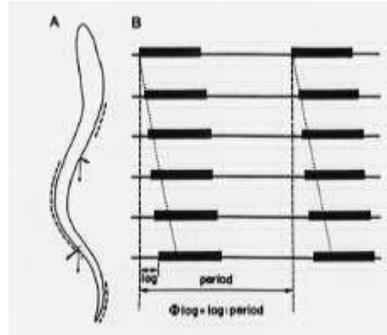
Verschaltung zu komplexeren Netzen mit zeitlichen Verzögerungen



Central pattern generator

- Verschaltung zu komplexeren Netzen mit zeitlichen Verzögerungen

(Bsp.: Neunauge mit Schwimmbewegungen unterschiedlicher Frequenz)



- Kombination mit sensorischen Signalen
- Hierarchische Steuerungen

Evolution von Bewegungsabläufen

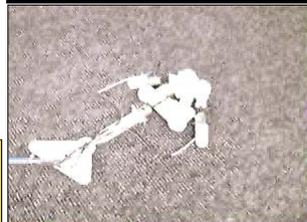
Karl Sims -- Virtual Creatures

**Evolved Virtual
Creatures**

**Examples from
work in progress**

Golem-Projekt Cornell University

**Anwendung evolutionärer Verfahren
für Design und
Steuerung durch Neuronale Netze**



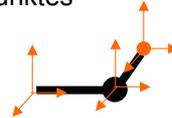
Gelenk-Kinematik: Statisch (Wdh.)

- **Kinematik:**

Aus Steuerparametern (einstellbare Gelenkwinkel) resultierende Endpunkte

- **Inverse Kinematik:**

Stellung der Gelenke zum Erreichen eines Endpunktes
(i.a. nicht eindeutig, evtl. keine Lösung)



- Koordinaten-Ursprung (polar, kartesisch) jeweils im Gelenk
- Zusammenhänge der Koordinatensysteme mittels Rotationen und Translationen
- Bewegungen entsprechen Rotationen und Translationen
- Homogene Koordinatensysteme verwenden

Berechnung von Laufbewegungen

- Zeitliche Bewegung des *Schwerpunktes*
evtl. dynamische Stabilität
- Zeitliche Bewegung des Körpers um Schwerpunkt
Ansatzpunkte der Beine damit fixiert
- Gangart (Zeitabhängige Zielpunkte am Boden)
- Für jedes Bein berechnen:
 - bei Bodenkontakt:
Ansatzpunkt + Fußpunkt ergibt Gelenkstellungen
 - ohne Bodenkontakt
Trajektorie zum folgenden Aufsatzpunkt ergibt Gelenkstellungen
- *Viele (keine) Lösungen (verbleibende Freiheitsgrade)*
- *Optimierung*

Berechnung von Laufbahnen

- Bahnplanung gemäß
 - Ausgangsstellung, Zielstellung, Hindernissen
 - Punkt-zu-Punkt-Bewegungen
 - Planungs-/Such-Problem im Raum
(Situationskalkül, STRIPS, A*, ...)
 - „Teachen“
 - Kraftbedingungen, Geschwindigkeiten, sanfte Bewegungen, Korrekturen
 - schrittweise Berechnen (Realzeitproblem)
 - Regelungsproblem
 - simultane Bewegungen (Fuzzy-Regler)
 - „Dual dynamics“

Berechnungen für Inverse Kinematik

- Analytische Verfahren
 - Stellparameter aus Matrixgleichung explizit ermitteln
 - nur in einfachen Fällen
- Spezielle Verfahren
 - roboterspezifisch gemäß Anordnung der Gelenke:
einfache Transformationen
z.B. Rotationen um eine Koordinatenachse
- Numerische Verfahren
- Lernverfahren

Fallstudie: Optimierter Lauf für AIBO

Diplomarbeit Uwe Düffert 2004

- Optimieren des omnidirektionalen Laufens
- Kalibrieren der Laufbewegungen (korrekte Ansteuerung)

- Automatisieren:
 - Lernverfahren
 - Testumgebung

AIBO: Anforderungen

- Omnidirektionaler Lauf:
Roboter soll in beliebige Richtung (vorwärts, rückwärts, seitwärts, schräg) laufen und sich dabei drehen können
- Fließende Übergänge zwischen den Richtungen (ohne „Anhalten“ oder „Umschalten“)
- Hohe Geschwindigkeiten
- Korrekte Umsetzung der angeforderten Bewegungen

Laufanforderungen:

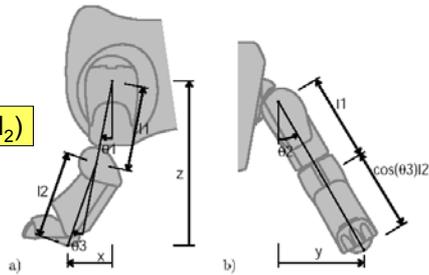
Vorwärtsgeschwindigkeit dx/dt

Seitwärtsgeschwindigkeit dy/dt

Drehgeschwindigkeit $d\phi/dt$

AIBO: Kinematik (Statisches Modell)

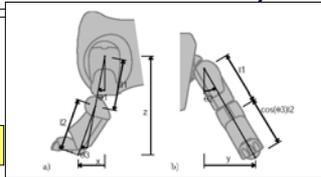
Rot(- θ_1)Rot(θ_2) Trans(l_1)Rot(- θ_3) Trans(l_2)



1. Drehung im Uhrzeigersinn um Schultergelenkwinkel 1 um die y-Achse
2. Drehung gegen Uhrzeigersinn um Schultergelenkwinkel 2 um die x-Achse
3. Verschiebung um Oberschenkellänge l_1 in Richtung der negativen z-Achse
4. Drehung im Uhrzeigersinn um Kniegelenkwinkel 3 um die y-Achse
5. Verschiebung um Unterschenkellänge l_2 in Richtung der negativen z-Achse

AIBO: Kinematik (Statisches Modell)

Rot(- θ_1) Rot(θ_2) Trans(l_1) Rot(- θ_3) Trans(l_2)



$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \text{Rot}_y(-\theta_1) \cdot \text{Rot}_x(\theta_2) \cdot \text{Trans} \begin{pmatrix} 0 \\ 0 \\ -l_1 \end{pmatrix} \cdot \text{Rot}_y(-\theta_3) \cdot \text{Trans} \begin{pmatrix} 0 \\ 0 \\ -l_2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

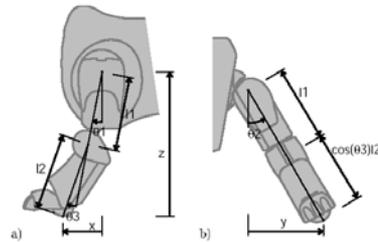
$$= \begin{pmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) & 0 \\ 0 & \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} l_2 \cos(\theta_1) \sin(\theta_3) + l_2 \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) + l_1 \sin(\theta_1) \cos(\theta_2) \\ l_1 \sin(\theta_2) + l_2 \sin(\theta_2) \cos(\theta_3) \\ l_2 \sin(\theta_1) \sin(\theta_3) - l_2 \cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - l_1 \cos(\theta_1) \cos(\theta_2) \\ 1 \end{pmatrix}$$

AIBO: Inverse Kinematik (Stat. Modell)

Berechnung von $\theta_1, \theta_2, \theta_3$
für angestrebte
Fußstellung (x,y,z)



$$\theta_3 = \arccos\left(\frac{x^2 + y^2 + z^2 - l_1^2 - l_2^2}{2l_1l_2}\right)$$

$$\theta_2 = \arcsin\left(\frac{y}{l_1 + l_2 \cos(\theta_3)}\right)$$

$$\theta_1 = \arctan(z, x) - \arctan\left(\frac{(l_1 + l_2 \cos(\theta_3)) \cos(\theta_2), l_2 \sin(\theta_3)}{y}\right)$$

AIBO: Inverse Kinematik (Stat. Modell)

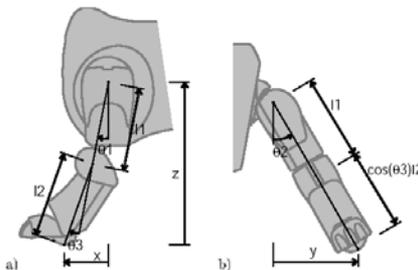
θ_3 nach Cosinus-Satz:

$$\cos(\pi - \theta_3) = \frac{l_1^2 + l_2^2 - (x^2 + y^2 + z^2)}{2l_1l_2} \quad \text{, und damit}$$

$$\theta_3 = \pi \pm \arccos\left(\frac{l_1^2 + l_2^2 - (x^2 + y^2 + z^2)}{2l_1l_2}\right)$$

$$= \mp \arccos\left(\frac{(x^2 + y^2 + z^2) - l_1^2 - l_2^2}{2l_1l_2}\right)$$

Vorzugsweise Beugung
nach vorn:
Positive Lösung

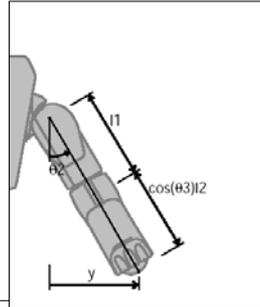


AIBO: Inverse Kinematik (Stat. Modell)

θ_2 nach Sinus-Definition:

$$y = \sin(\theta_2) \cdot (l_1 + l_2 \cdot \cos(\theta_3)) \quad , \text{ also}$$

$$\theta_2 = \arcsin\left(\frac{y}{l_1 + l_2 \cos(\theta_3)}\right) \quad , \text{ da anatomiebedingt } |\theta_2| \leq \frac{\pi}{2}.$$



AIBO: Inverse Kinematik (Stat. Modell)

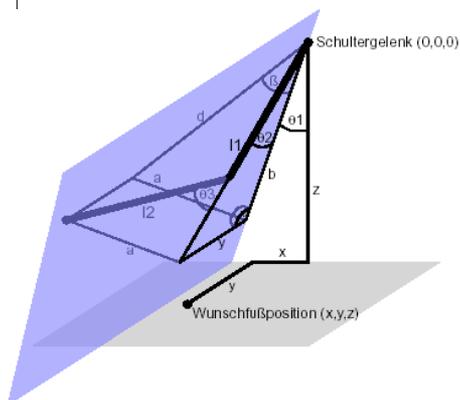
θ_1 mit weiteren Variablen:

$$a = l_2 \sin(\theta_3)$$

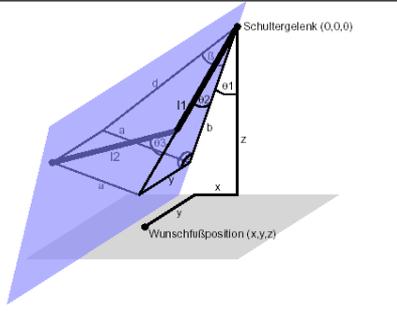
$$b = (l_1 + l_2 \cos(\theta_3)) \cos(\theta_2)$$

$$d = \sqrt{a^2 + b^2} \quad \text{und}$$

$$\beta = \arctan(b, a) \quad .$$



Kinematik (Stat. Modell)



$$a = l_2 \sin(\theta_3)$$

$$b = (l_1 + l_2 \cos(\theta_3)) \cos(\theta_2)$$

$$d = \sqrt{a^2 + b^2} \quad \text{und}$$

$$\beta = \arctan(b, a) \quad .$$

$$a = d \cos(\beta) \quad \text{und}$$

$$b = d \sin(\beta) \quad .$$

Mit diesen Hilfsvariablen ergibt sich:

$$x = l_2 \cos(\theta_1) \sin(\theta_3) + l_2 \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) + l_1 \sin(\theta_1) \cos(\theta_2)$$

$$= a \cos(\theta_1) + b \sin(\theta_1)$$

$$= d \cos(\theta_1) \cos(\beta) + d \sin(\theta_1) \sin(\beta)$$

$$= d \cos(\theta_1 + \beta)$$

$$z = d \sin(\theta_1 + \beta) \quad .$$

$$\theta_1 + \beta = \arctan(z, x) \quad , \quad \text{und somit}$$

$$\theta_1 = \arctan(z, x) - \beta \quad .$$

AIBO: Einfaches Laufmodell

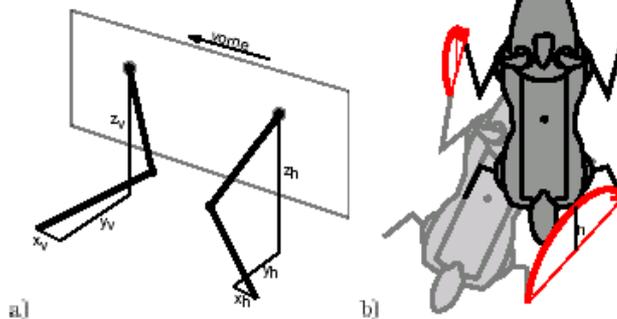
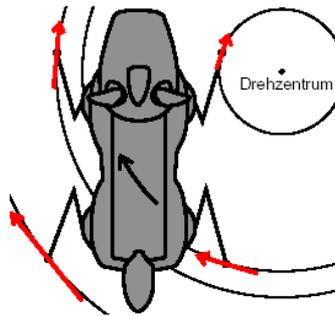


Abbildung 2-5: Das erste Laufmodell und seine Parameter: **a)** Die Seitenansicht enthält alle räumlichen Parameter. **b)** Die Draufsicht zeigt, dass die Schwerpunktsverlagerung des Roboters das Ziel der in Halbellipsen erfolgenden Fußbewegungen bei einem Schritt impliziert.

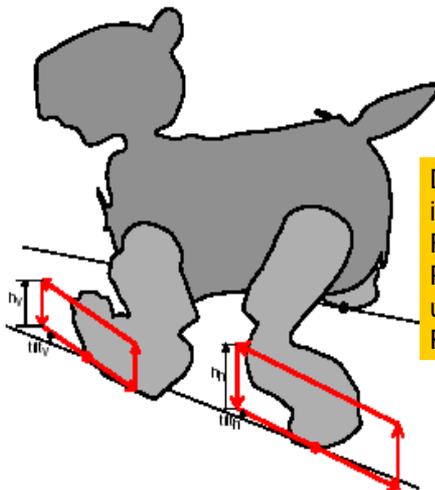
AIBO: „Rädermodell“

Kurvenbewegungen können nach dem Prinzip des Differentialantriebs realisiert werden (unterschiedliche Geschwindigkeiten der Beine)



Gleichzeitiges Laufen und Drehen mit „Rädermodell“: Die Ruhepositionen der Füße bewegen sich auf Kreisbahnen um ein gemeinsames Drehzentrum. Realisierung durch die aus den gedachten Radbewegungen abgeleiteten Schritte (rot) tangential zu den optimalen Kreisbahnen.

AIBO: Fußbewegungen im „Rädermodell“



Die Parameter der Fußbewegung im verwendeten Laufmodell: Fußbewegungen erfolgen in Parallelogrammen, deren Richtung und Länge sich nach dem Rädermodell ergibt.

AIBO: Parameter für Optimierung

Um die Optimierung zu beschleunigen, werden Parameter ausgewählt, die erfahrungsgemäß großen Einfluss haben und bereits „voreingestellt“ werden können:

1. Parameter zur Bestimmung der Ruheposition der Füße relativ zum Körper
2. Parameter zur Beschreibung der Trajektorie der Füße
3. Parameter zur Bestimmung der Gangart

Optimierungsaufgabe:

Parameterwerte $P = (p_1, \dots, p_n)$ bestimmen, der optimales Laufen realisiert.

AIBO: Zerlegung der Aufgabe

- Erfahrung: Parametersätze für schnelles Vorwärtslaufen haben nur mittlere Güte beim Rückwärtslaufen
- Deshalb:
Unterschiedlichen Parametersätzen $P_i = (p_{i1}, \dots, p_{in})$ für unterschiedliche Laufforderungen A_i
- 127 verschiedene Anforderungen unterteilt nach
 - Laufrichtung (8 Werte)
 - Dreh-Lauf-Verhältnis (7 Werte)
 - Gesamtgeschwindigkeit (3 Werte)(nicht alle Kombinationen treten auf)
Diese Aufteilung ist gleichmäßiger als eine Aufteilung nach Vorwärts-/Seitwärts-/Drehgeschwindigkeit

AIBO: Zerlegung der Aufgabe

$$\text{Laufrichtung } \alpha = \arctan(\dot{x}, \dot{y})$$

$$\rightarrow \left[-\pi, -\frac{3\pi}{4}, -\frac{\pi}{2}, -\frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4} \right]$$

$$\text{Dreh-Lauf-Verhältnis } \delta = \frac{2}{\pi} \arctan \left(\frac{v}{v_{max}}, \frac{\dot{\varphi}}{\dot{\varphi}_{max}} \right)$$

$$\rightarrow \left[\underbrace{-1}_{\text{Rechtsdrehung}}, -\frac{3}{10}, -\frac{1}{10}, 0, \frac{1}{10}, \frac{3}{10}, \underbrace{1}_{\text{Linksdrehung}} \right]$$

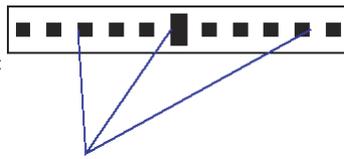
$$\text{Gesamtgeschwindigkeit } r = \sqrt{\left(\frac{v}{v_{max}} \right)^2 + \left(\frac{\dot{\varphi}}{\dot{\varphi}_{max}} \right)^2}$$

$$\rightarrow [\text{langsam, mittelschnell, schnell}]$$

AIBO: Lernumgebung

- Optimierung mittels evolutionärer Verfahren:
 - Experimente: Güte (Fitness) von Parameterwerten $P = (p_1, \dots, p_n)$ in realer Umgebung bestimmen
 - Fitness bestimmt durch Übereinstimmung mit geforderter Bahn und Zeit
 - Rekombination von Parametersätzen $P = (p_1, \dots, p_n)$
- Weitgehende Automatisierung der Experimente in einer speziell entwickelten Lernumgebung:
 - Roboter versucht auf geforderter Bahn zu laufen
 - Roboter ermittelt Fitness (Korrektheit und Zeit)
 - Steuerungsprogramm mittels XABSL

AIBO: Lernumgebung



- Orientierungshilfe
 - Mit ihrer Hilfe erfolgen Positionsbestimmungen und Bahnkorrekturen.
 - Die Erkennung geht in die Fitness ein

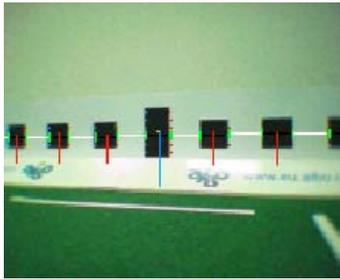


Bild aus der Kamera eines ERS-210 mit als erkannt markierten Teilen der Lokalisierungshilfe

AIBO: Fitnessfunktion

- dx/dt durchschnittliche Laufgeschwindigkeit in x-Richtung (entlang des Parcours auf die Lokalisierungshilfe zu beziehungsweise von ihr weg) in mm/sec
- Δy : gemittelter Betrag der Abweichung der y-Position, also durchschnittlicher Abstand zur Optimallinie
- $\Delta\phi$: gemittelter Betrag der Abweichung der Roboterrichtung, also durchschnittlicher Abstand zur Optimalrichtung
- d^2z/dt^2 : gemittelter Betrag der Beschleunigung in z-Richtung, diese ist unbeabsichtigt und als unangenehm hartes Stampfen zu erkennen
- p_{blind} : Anteil der Bilder an der Gesamtbildzahl, in denen nicht eindeutig die Lokalisierungshilfe erkannt werden konnte, die verwendeten Laufparameter führten bei hohem p_{blind} also offenbar zu stärkerer Abweichung der Realität von der Ansteuerung oder zu stärkerer Erschütterung als akzeptabel

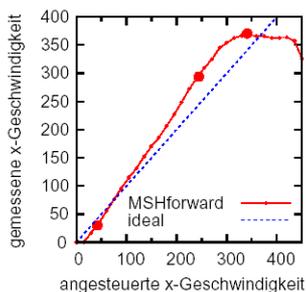
AIBO: Vermessung

- Es wurden nicht für alle 127 Anforderungen neue Parametersätze durch evolutionäre Techniken optimiert.
- Stattdessen wurden weitere (bekannte) Parametersätze hinsichtlich ihrer Eignung vermessen
- Daraus ergibt sich eine Zuordnung günstiger Parametersätze $P_i = (p_{i1}, \dots, p_{in})$ für unterschiedliche Laufanforderungen A_i
- Bei der Auswahl wurde auch auf gute Übergänge zwischen benachbarten Anforderungen geachtet.

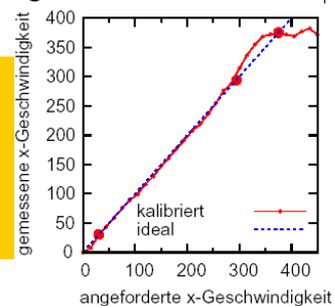
AIBO: Kalibrierung

Damit angeforderte und tatsächlich erzielte Geschwindigkeit möglichst gut übereinstimmen, ist eine Kalibrierung notwendig:

- Vermessung der erzielbaren Geschwindigkeiten
- Korrekturwerte für die Ansteuerung



Vermessung für
niedrige/mittlere/
hohe
Geschwindigkeit
ohne bzw. mit
Kalibrierung



Sensor-Aktor-Kopplung

Beispiel Insekten

Einfache Verhaltensweisen:

- Futter aufnehmen
- Hindernissen ausweichen, ...
- Speziell: Geruchssinn ausnutzen
 - Insekten sondern Duftstoffe (Pheromone) ab
 - Andere Insekten folgen den Pheromonen

Anwendung:

- Partnersuche
- Optimale Wege zwischen Futter und Nest
- Nestbau

Sensor-Aktor-Kopplung

Optimale Wege zwischen Futter und Nest

- Hindernissen ausweichen.
- Unterwegs nach Möglichkeit Futter aufnehmen.
- Im Nest Futter ablegen.
- Fortbewegung:
 - Zufällige Bewegungen,
bevorzugt: Richtung von Pheromon-Konzentration.
 - Dabei selbst Wege mit Pheromon markieren.

Mit der Zeit entsteht ein optimaler spannender Baum zwischen Futterplätzen und Nest.

Schwarmverhalten

- Komplexes kollektives Verhalten, dass aus dem Verhalten der Individuen entsteht (“emergiert”), ohne dass es einen Plan dafür gibt.
- Wesentliche Voraussetzungen im Beispiel:
 - Exploration (durch zufällige Bewegungen)
 - Stabilität (durch kollektives Bevorzugen bestimmter Verhaltensweisen)
 - Flexibilität (Pheromon verflüchtigt sich: “Vergessen”)
 - Robustheit (Individuelles Fehlverhalten belanglos)

Grundlage: Einfache Verkopplungen von Reizaufnahme und Aktionen

Sensor-Aktor-Kopplung

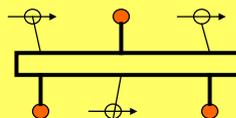
Beispiel:

Tastsinn für Fußkontakt mit Beinbewegung koppeln

Stabiler Lauf eines Hexapoden



Stabiler Lauf:
Stets 3 Beine am Boden



Sensor-Aktor-Kopplung

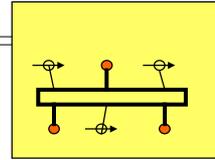
Einfaches Laufmuster:

Patterngenerator

Oszillierend: Aktivierung, Inhibition

Beherrschung von unebenen Gelände

- Solange Fußkontakt (Taster) fehlt:
Bein tiefer setzen
- Bei Hindernis während Vorwärtsbewegung:
Bein zurück ziehen, höher heben, nach vorn setzen



Ohne Internes Modell:

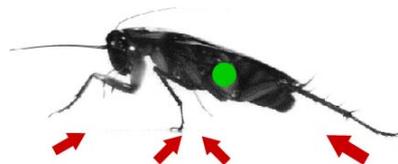
The best model of the world is the world

„Emergentes Verhalten“

Fallstudie: Sprawlita

Orientierung an der Natur (Arthropoden, z.B. Kakerlake)

- Laufen funktioniert automatisch
- Schnelligkeit: 10 x Körperlänge/sec)
- Ohne relevanten Geschwindigkeitsverlust Hindernisse überwinden (3 x Hüfthöhe)

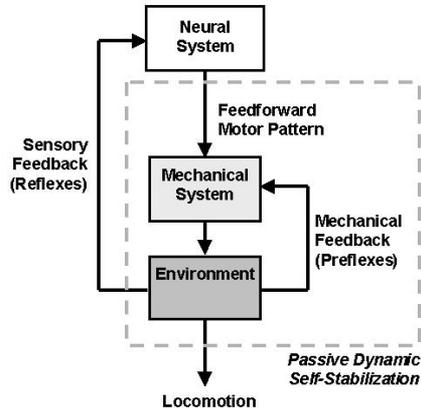


Kräfteverteilung

- 150 mm lang
- 290 g schwer
- Preis ca. 1400 \$

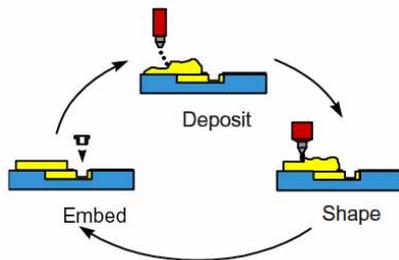
Sprawlita: Biologisches Modell der Fortbewegung

- Vernetzung aufeinander abgestimmter mechanischer, sensorischer und neuronaler Funktionen



Sprawlita: Design und Fertigung

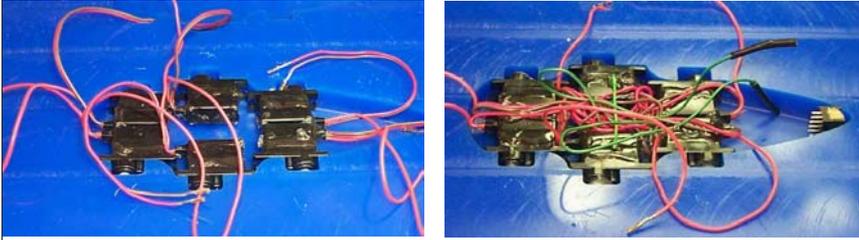
Shape Deposition Manufacturing:



- Formen
 - Teile werden in Gußform in Lage gesetzt
- Depositionieren
 - Ausgießen & fixieren
 - Bearbeiten des Körpers um weiter darauf aufzubauen

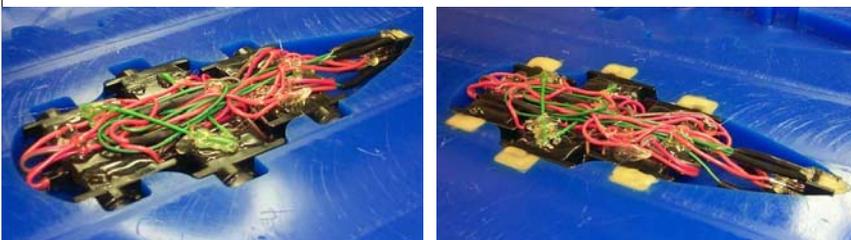
- Perfekte Befestigung/ Fixierung
- Schutz der sensiblen Komponenten (Kabel, Kabelkontaktpunkte, Microcontroller)
- Kostengünstige Herstellung

Sprawlita: Design und Fertigung



- Ausrichtung der Servomotoren und des Mikrocontrollers in der Form
- Verkabelung
 - Kabel führen zu Schnittstelle am Hinterteil
 - Alles wird möglich flach gehalten und verflochten

Sprawlita: Design und Fertigung



- Geringe Fixierung der Kabel mit Silikon
 - Schutz vor Verschiebung während des Ausgusses
- Enden der Motoren werden abgeschlossen

Sprawlita: Design und Fertigung



Ausgießen

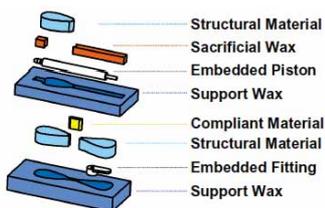


H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

135

Sprawlita: Design und Fertigung



- Beine einzeln in Gußtechnik geformt
- Starre Enden, flexibles Gelenk
- Air-Piston und Kontaktstück für Servomotoren mit eingegossen
- Keine späteren Bohrungen

H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

136

Sprawlita: Design und Fertigung

Pneumatik:

- 2 Ventile als Aufsatz auf dem Hinterteil
- Anstecken der Beine
- Verbindung der Beine mit der Pneumatik
 - Über Kreuz je 3 Beine an einem Ventil
- Anschluß an PC am Heck



Sprawlita: Verhaltens-Steuerung



Kakerlake beim Überqueren
einer Hindernisstrecke

Biologisches Vorbild

- Impulse zu den Muskeln ändern sich praktisch nicht bei unwegsamem Gelände
- Gang paßt sich ohne zentrale Informationsverarbeitung automatisch an

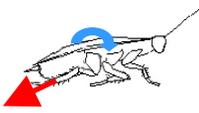


Impulse zu den Muskeln



Sprawlita: Beinkonstruktion

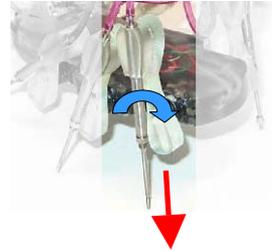
Kakerlaken-Modell Theoretische Adaption Implementation bei Sprawlita



Passives Hüftgelenk
Effektive
Beschleunigungskraft



Drehgelenk
Prismatisches
Gelenk



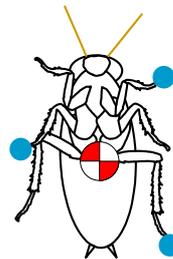
Flexible, gedämpfte Hüftbefestigung
Integrierter Luftdruck-Kolben

H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

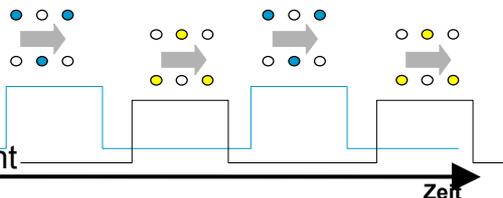
Vorlesung Kognitive Robotik
Bewegung

139

Sprawlita: Bewegungsmuster



- Drei Beine bewegen
sich immer gleichzeitig
- Schrittsequenz konstant

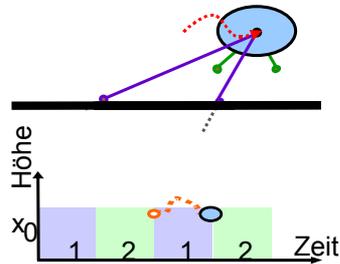
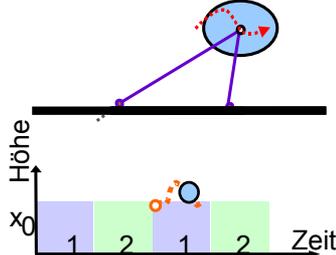
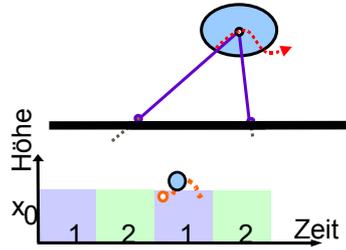
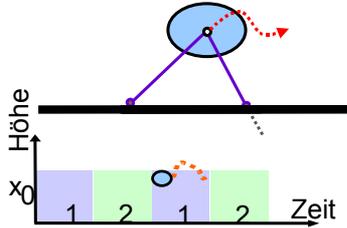


H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

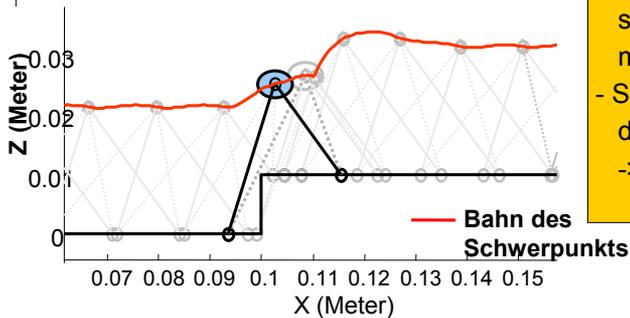
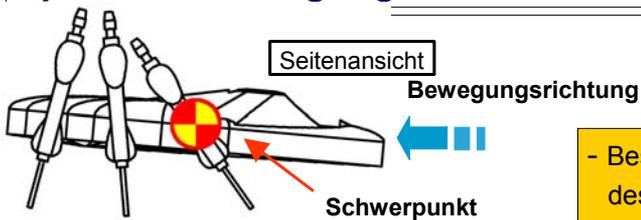
140

Sprawlita: Schrittsequenz



H.D.Burkhard,
Winter-Semes

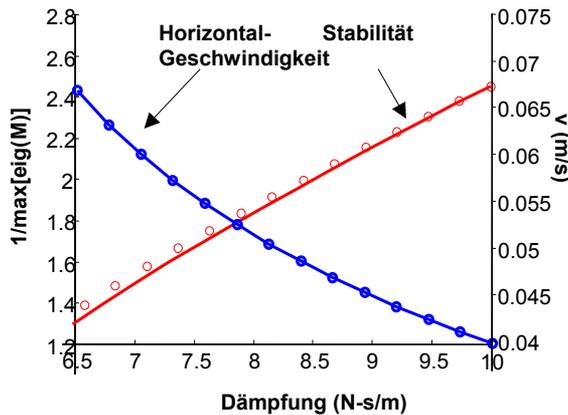
Sprawlita: Bewegung in unebenem Terrain



- Beschleunigung entlang des Hinterbeines
- Bei Hindernis steht es steiler, Schub erfolgt also mehr nach oben
- Schwerpunkt im Drehpunkt des Hinterbeines
- > kein Drehmoment, höhere Stabilität

Sprawlita: Auswirkung von Dämpfung

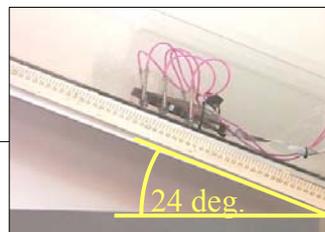
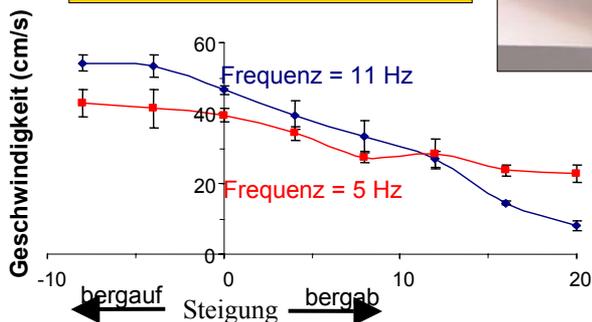
- niedrigere Dämpfung steigert Geschwindigkeit durch Speicherung von Energie im elastischen Hüftgelenk
- höhere Dämpfung steigert Stabilität durch bessere Bodenhaftung



143

Sprawlita: Auswirkung der Schrittfrequenz

Geschwindigkeit bei verschiedenen Steigungen und Schrittfrequenzen



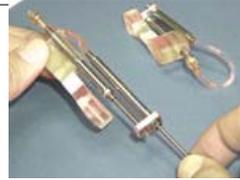
Sprawlita: Entwicklungsgeschichte



Sprawley-Davidson
(Ventile auf dem Kolben)



SprawlEx
(end-effector)



Cascaded Pistons
(kaskadierter Kolben)

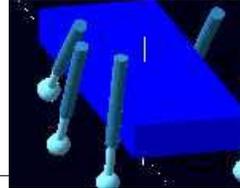
Pseudo-Sprawl
(konfigurierbar)



Sprawlettes
(Serienteile)



VSprawl
(Simulation)



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

145

Sprawlita



www-cdr.stanford.edu/biomimetics/

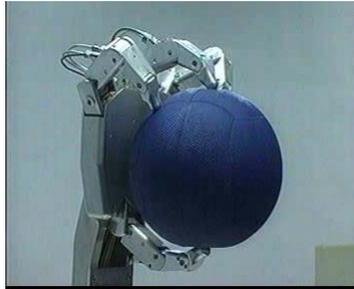
H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

146

Hand

Zusammenspiel von Sensoren (Druck)
und Aktoren (Bewegung)



Regelung

Zeitlichen Ablauf eines Prozesses so beeinflussen, dass
vorgegebene Ziele erreicht bzw. eingehalten werden.

- Endpunkt
 - z.B. Bewegung zu einem Zielpunkt
- Führung
 - z.B. Gleichgewicht halten
- Steuerung: Beeinflussung allgemein
- Regelung: Steuerung mit Rückkopplung
(Regelschleife)

Regelschleife

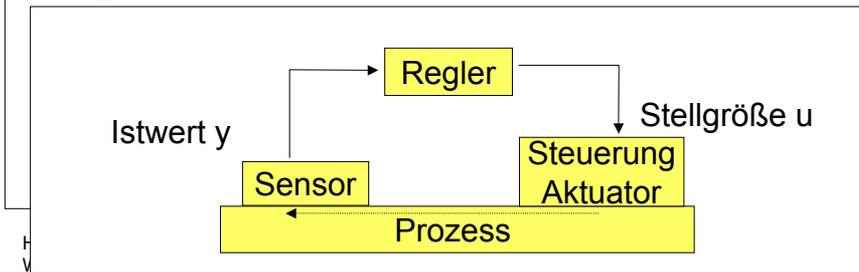
- Regelstrecke: Steuerung, Prozess: Istwert y
- Regler: Sensor, Regler: Stellgröße u

Ziel:

y soll mit vorgegebener Führungsgröße w übereinstimmen

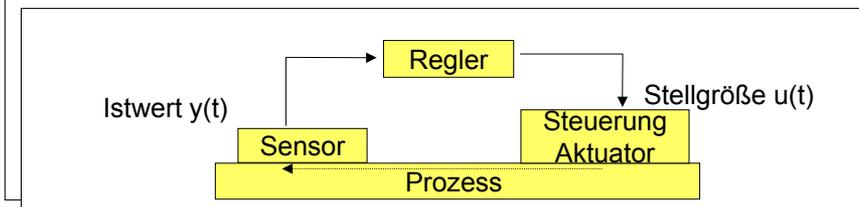
Verfahren:

Nach Messung von y soll Regler jeweils u geeignet vorgeben (fortlaufende Korrekturen)



Regelschleife

- **Führungsregelung:** $y(t)$ soll ständig mit vorgegebener Führungsgröße $w(t)$ übereinstimmen, d.h. die Differenz $e(t)=y(t)-w(t)$ soll minimal sein.
 - z.B. Abstand von der Wand einhalten
- **Endwertregelung:** $y(t)$ soll zum Zeitpunkt T einen vorgegebenen Wert $w(T)$ erreichen,
 - z.B. an der Tür ankommen



Regelung bei Robotern

Interne Regelkreise

- Interne Sensoren überwachen inneren Zustand
 - Gleichgewicht, Gelenkstellungen, Temperatur, ...

Regelkreise bzgl. Umwelt

- Interne/Externe Sensoren überwachen Beziehungen zur Umgebung
 - Bewegung, Abstand, ...

Management zur Koordination notwendig (z.B. Schichten)

Regler

Der Regler muss geeignete Stellwerte u für die Steuerung bereitstellen.

z.B. Stellwert u proportional zur Abweichung $e(t)=y(t)-w(t)$

Probleme:

- Zeitverzögerung
 - dadurch können Oszillationen entstehen
- Verrauschte Daten
 - ungenauere Steuerung

Regler soll stabil arbeiten:

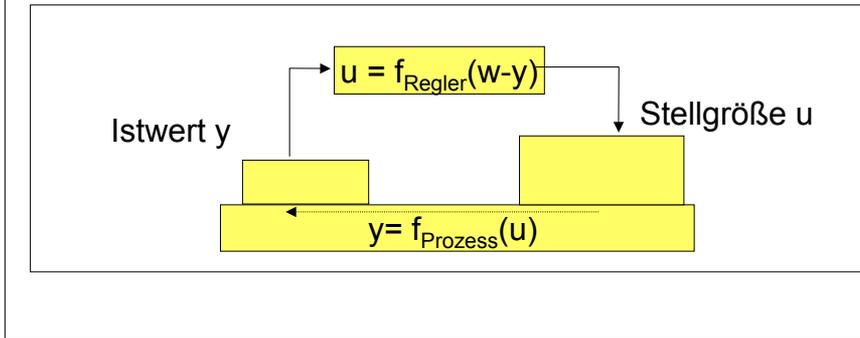
Trägheit bei der Ansteuerung, Filterung der Sensorsignale,

...

Modellierung

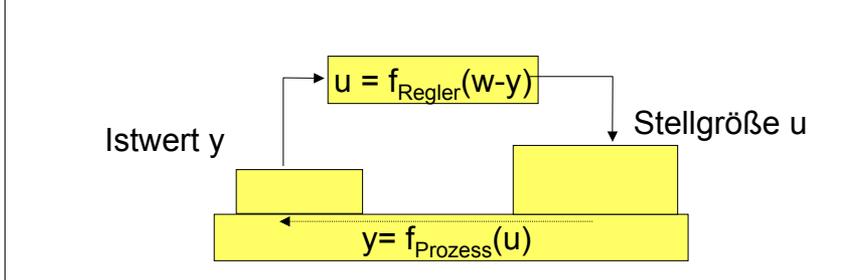
Zielstellung:

- $e(t) := w(t) - y(t) = 0$
- $f_{\text{Prozess}}(f_{\text{Regler}}(w(t) - y(t))) = w(t+1)$ (bzw. Annäherung)



Modellierung

1. Beschreibung des Prozesses: $f_{\text{Prozess}}(y)$
 2. Beschreibung des Regelkreises
 $y(t+1) = f_{\text{Prozess}}(f_{\text{Regler}}(y(t)))$
 3. Entwurf eines Reglers mit $w(t) = y(t)$
- Modelle z.B. mittels Differentialgleichungen



Proportionalregler

Abweichung: $e(t) := w(t) - y(t)$

Stellgröße \sim Abweichung

$$u(t) = K \cdot e(t)$$

mit Proportionalitätsfaktor K

Eigenschaften abhängig von K :

- Kleines K : langsames Herantasten an Sollwert
- Großes K : übersteuern bei nachlassendem Fehler e
Gefahr von Oszillationen

Integralregler

Abweichung: $e(t) := w(t) - y(t)$

Stellgröße \sim Dauer/Größe der Abweichung

$$u(t) = K \cdot 1/T_I \cdot T_0 \cdot [e(t-1) + e(t-2) + \dots + e(0)]$$

mit $T_0 :=$ Messintervall jeweils zwischen t und $t+1$
und konstanter „Nachstellzeit“ bzw. „Integrierzeit“ T_I

Eigenschaften:

Kann zu geringe Proportional-Regelung ausgleichen,
erhöht aber anschließend noch gewisse Zeit weiter

Differentialregler

Abweichung: $e(t) := w(t) - y(t)$

Stellgröße \sim Änderung der Abweichung

$$u(t) = K \cdot 1/T_D \cdot [e(t) - e(t-1)]$$

mit konstanter „Vohaltezeit“ bzw. „Differenzierzeit“ T_D

Eigenschaften:

Reagiert kurzfristig auf einen „Sprung“ der Abweichung, aber nicht auf dauerhaft konstante Abweichung.

Als alleiniger Regler nicht einsetzbar.

Problematisch bei Rauschen der Messwerte.

Kombinationen

PI-Regler

PD-Regler

PID-Regler

PID-Regler:

$$u(t) = K(e(t) + 1/T_I \cdot T_0 \cdot [e(t-1) + e(t-2) + \dots + e(0)] + 1/T_D \cdot [e(t) - e(t-1)])$$

Bei Realisierung durch elektronische Bauelemente auch in kontinuierlicher Form:

Integralregler: $K/T_I \cdot \int e(t) dt$

Differentialregler: $K \cdot de(t)/dt$

Bei Bedarf Filterung, z.B. zur Rauschunterdrückung

Physik ausnutzen: Sensor-Aktor-Kopplung

- Regelkreis mittels geeigneter Sensoren
 - Messung von Körperwerten, z.B. Gelenkstellungen oder
 - Umweltwerten (z.B. Bewegungsfortschritt)
- Servo-Motor
- Braitenberg-Vehikel
- Basis für „Emergentes Verhalten“
- Basis effizienter Verhaltensweisen in der Natur

Physik ausnutzen: Sensor-Aktor-Kopplung

- Regelung: Gleichgewicht + Bewegung



Segway



Physik ausnutzen: Energie

- Passiv Walker
- „Federung“: Rückgewinnung von Energie
- Bein als „Pendel“
- Arme als „Gegenpendel“



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

161

Humanoide Roboter

Anpassung an menschengerechte Umgebung

Prothetik

Kommunikation: Gestik

H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

162

Humanoide Roboter im RoboCup 2002

Tao-Pie-Pie (Neuseeland)
FootsPrint (Japan)



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

163

Humanoide Roboter im RoboCup 2003

Firststep (Honda)



H.D.Burkhard, HU Berlin
Winter-Semester 2005/06

Vorlesung Kognitive Robotik
Bewegung

164

Humanoide Roboter im RoboCup 2004

- V-Stone
- Vision



Robosapien von Mark Tilden (Kanada)

Wow-Wee's RoboSapien
Demonstration with
Mark W. Tilden

February 15, 2004
New York City Toy Fair

(C) Solarbotics, 2004

Controlling & Programming the
RoboSapien with the Remote
Control

February 15, 2004
New York City Toy Fair

(C) Solarbotics, 2004



7 DOF
Schulter (x 2)
Ellenbogen (x 2)
Hüfte
Bein (x 2)

Fernsteuerung

- Übertragung von sensorischen Daten
- Übertragung von (höheren) Befehlen

Probleme:

Beispiel: Marsrover 2004

- Sensorische Rückkopplung
- Komplexität von Bewegungen
- Verzögerung

• Telepräsenz

- Datenhandschuh
- Weiteres ...



Zukunft?

- Energie
- Kraftübertragung
- Endo-Skelett vs. Exo-Skelett
- Muskeln mit paralleler Ansteuerung der Aktuatoren
- Haut mit „Empfindungen“
 - Druck, Wärme, Feuchtigkeit
- Steuerungsprobleme: Lokal vs. Global
- Lernen (Belehren, Training, trial and error, ...)

- Materialien
- Design