

Zusammenfassung Robotik 1

Cand.-Inform. Michael Aschke

16. Oktober 2000

Kapitel 1

Mathematische Grundlagen der Kinematik

1.1 Beschreibung von Objekten und Objektlagen im 3-dim. euklidischen Raum

- BKS: Basiskoordinatensystem (wird durch orthogonale Einheitsvektoren aufgespannt)
- OKS: Objektkoordinatensystem (ebenfalls durch orthogonale Einheitsvektoren definiert)
- EKS: Effektorkoordinatensystem (TCP (Tool Center Point))
- SKS: Sensorkoordinatensystem
- **Ort** eines Objektes: Ortsvektor vom Ursprung des BKS zum Ursprung des OKS
- **Orientierung** bezogen auf BKS: Rotationsmatrix zur Abbildung der Einheitsvektoren des OKS auf die Einheitsvektoren des BKS
- **Lage**: Ortsvektor und Rotationsmatrix des OKS bezogen auf das BKS
- **Freiheitsgrade**: Anzahl möglicher unabhängiger Bewegungen im Bezug auf das BKS (bestehend aus *Translationen* und *Rotationen*. Für Objekte, welche im 3-dim. Raum beweglich sind: $f = 6$ (3T und 3R))
- **Bewegungsfreiheitsgrade** eines Roboters: Freiheitsgrad Rotationsgelenk: $F_R \leq 3$
Freiheitsgrad Translationsgelenk: $F_T = 1$
Anzahl n

Bewegungsfreiheitsgrade:
$$F = \sum_{i=1}^n (F_{R_i} + F_{T_i})$$

- $F \geq f$

1.2 6-dim. Beschreibungsvektor

- Der Ort kann durch ein Tupel von sechs Zahlen beschrieben werden: $\vec{v} = (x, y, z, \alpha, \beta, \gamma)$, wobei x, y, z , die Koordinaten des Ursprungs des OKS bezogen auf das BKS und α, β, γ die Drehwinkel sind, welche sich auf die Drehachsen beziehen.
- Zwei Arten der Festlegung der Drehachsen und ihrer Reihenfolge: Euler-Winkel und Roll-Pitch-Yaw
- Euler: Drehung um **veränderte** Achsen: Drehung α um z-Achse des BKS, Drehung β um neue y-Achse y' , Drehung γ um neue z-Achse z''
- Roll-Pitch-Yaw: Drehung um **unveränderte** Achsen. Erste um x-Achse, zweite um y-Achse und dritte um z-Achse.
- Zusammenfassung von Vektor- und Matrizenmanipulation durch Verwendung von homogenen 4x4-Matrizen.

1.3 Homogene 4x4-Matrix

- Die Verwendung von homogenen Koordinaten im 3-dimensionalen euklidischen Raum erlaubt die Entwicklung von 4x4-Transformationsmatrizen, die die Rotation, die Translation, die Skalierung sowie die perspektivische Transformationen enthalten.

•

$$\begin{pmatrix} R & u \\ P & s \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & u_x \\ n_y & o_y & a_y & u_y \\ n_z & o_z & a_z & u_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

mit R = Orientierung (= 3 Einheitsvektoren (n, o, a) in x, y, z -Richtung des OKS), u = Ort, P = Projektionsmatrix (=Perspektive), s = Skalierungsfaktor

1.4 Orientierungsbeschreibung mit 3x3 Matrizen

- Siehe Isometrien/Drehmatrizen

•

$$R_z(\alpha) \cdot \vec{u} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \vec{u}'$$

- Die 1 steht bei der Rotationmatrix immer in der Diagonale in derjenigen Zeile, um deren Achse gedreht wird.
- Bei Eulerwinkeln: Matrixmultiplikation **von links nach rechts**, da linksstehende Matrix immer das Bezugssystem darstellt. Bei erster Matrix bildet das BKS das Bezugssystem.
- Bei Roll-Pitch-Yaw: Matrixmultiplikation **von rechts nach links**, da sich das Bezugssystem nicht mitändert.

1.5 Transformation: Beschreibungsvektor in 4x4 Matrix

- 6-dim. Beschreibungsvektor (siehe oben).
- Ort: $x,y,z \rightarrow$ Transformation in homogenen Raumpunkt $(x, y, z, 1)^\top$
- Orientierung: $\alpha, \beta, \gamma \rightarrow$ Berechnung der zugehörigen 3x3 Orientierungsmatrix im \mathbb{R}^3 (abh. von Euler, oder RPY).

1.6 Transformation: 4x4 Matrix in Beschreibungsvektor

Siehe Skript

1.7 Rotation und Translation von Punkten

Siehe Skript

1.8 Verkettete Lagebeschreibung

- Beschreibung oft nicht in Bezug auf BKS, sondern relative Definition
- Umrechnen von Koordinaten notwendig
- Vorteile:
 - Verringerung des NACHführaufwandes
 - Koordinatenangaben beschränken sich auf kürzere Distanzen
- Klarere Schreibweise durch Angabe der Bezugskoordinatensysteme:

$${}^{\text{BKS}}H_{B,\text{neu}} = {}^{\text{BKS}}H_A \cdot {}^{H_A}H_B$$

Kapitel 2

Robotermodellierung

Drei Modellarten:

- Geometrie: Beschreibt Oberflächen, Volumen eines Körpers im 3-dim. euklidischen Raum
- Kinematik: Beschreibt Bewegungszustände von Punkten, Körpern, sowie kinematischen Ketten
- Dynamik: Untersucht Bewegung von Körpern als Folge der auf sie wirkenden Kräfte und Momente

2.1 Geometrisches Modell

- Dient als Grundlage zur Modellierung und Animation von Objekt, Roboter und Umwelt.
- erlaubt Abstandsberechnungen und Kollisions- bzw. Durchdringungsberechnungen
- ist Basis für automatische Bahnplanungsverfahren
- Oberflächenpräsentation mit Zentralperspektive, Approximierte Modelle (Drahtmodell, Volumenmodell). Bilder siehe Skript (Folien 5-8)

2.2 Kinematisches Modell

- Zweck:
 - Roboterentwurf
 - Erreichbarkeitsanalyse
 - Bahnplanung
 - Programmierung
 - Simulation
 - Analyse und Synthese kollisionsfreier Bahnen
 - Zusammenhang Konfigurationsraum in Roboterkoordinaten → Konfigurationsraum in Weltkoordinaten

- Kinematische Kette: wird von mehreren Körpern gebildet, die durch Gelenke kinematisch verbunden sind. (z.B. Roboterarm). Offene kinematische Kette (z.B. Ein-Arm), geschlossene kinematische Kette (z.B. beidarmiges Szenario)
- Ansatz zur kinematischen Modellierung von Robotern
 - Bestimmung der Gelenkwinkelfreiheitsgrade (Translation/Rotation)
 - Best.: Anzahl der Gelenke n ($\theta_1, \theta_2, \dots, \theta_n$) und Indizierung
 - Best.: kinematische Parameter zwischen Gelenkachsen und lokalen Koordinatensysteme
 - Best.: Basistransformation zur Überführung benachbarter Koordinatensysteme (3 Translationen, 3 Rotationen)
 - Durch geeignete Konstruktion von BKS kann Anzahl der Transformationen verringert werden (Denavit-Hartenberg) (2 Translationen, 2 Rotationen) \rightarrow 4 kinematische Parameter, davon lediglich 1 variabel.

2.2.1 Denavit-Hartenberg

- Transformation vom OKS des i -ten Armelementes in das OKS des $(i-1)$ -ten Armelementes nach der **Denavit-Hartenberg-Konvention**.
- Voraussetzungen: Ursprung des Koordinatensystems A_{i-1} ist fest mit Armteil $i-1$ verbunden und liegt im Schnittpunkt der gemeinsamen Normalen der Gelenkachsen $i-1$ und i mit der Gelenkachse i .
- Prinzip: Effektorstellung wird schrittweise ermittelt; In jedes Robotergelenk wird ein Rechtskoordinatensystem gelegt; Die Systeme werden nach bestimmten DH-Regeln konstruiert, so daß die Transformationsmatrizen eine einheitliche Struktur erhalten.
- **Denavit-Hartenberg-Regeln:**
 1. Achse z_i des Systems S_i wird in Bewegungsachse des Gelenks G_i gelegt. Richtung: positive Bewegungsrichtung von G_i .
 2. Achse x_i wird durch Normale von z_{i-1} nach z_i definiert.
 3. $y_i = z_i \times x_i$ (Rechtssystem)
- **Spezialfälle:**
 - Schneiden sich zwei Gelenkachsen \rightarrow Ursprung im Schnittpunkt der beiden Achsen. (Kann auch außerhalb des Arms liegen)
 - Gelenkachsen parallel \rightarrow Ursprung so wählen, dass Versatz zum Koordinatenursprung im nächsten Gelenk des nächsten Armteils zu 0 wird.
 - Koordinatensystem A_0 bildet Basissystem und ist fest mit der Basis und dem Ursprung in der 1. Achse verbunden.
 - Koordinatensystem D_n wird in den Greifer gelegt. x_n muss normal zu z_{n-1} liegen.
- **Überführen OKS(i-1) auf OKS(i):**
 1. Rotation θ_i um die z_{i-1} -Achse, damit x_{i-1} -Achse parallel zur x_i -Achse liegt:

$$R_{z_{i-1}}(\theta_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Translation s_i entlang der z_{i-1} -Achse zu dem Punkt, in dem sich z_{i-1} und x_i schneiden:

$$T_{z_{i-1}}(s_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Translation a_i entlang der x_i -Achse, um die Ursprünge in Deckung zu bringen:

$$T_{x_i}(a_i) = \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4. Rotation α_i um die x_i -Achse, um die z -Achse und die y -Achse in Übereinstimmung zu bringen:

$$R_{x_i}(\alpha_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

• In Matrixschreibweise:

$$\begin{aligned} D_{i-1,i} &= R_{z_{i-1}}(\theta_i) \bullet T_{z_{i-1}}(s_i) \bullet T_{x_i}(a_i) \bullet R_{x_i}(\alpha_i) \\ &= \begin{pmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

• Unterschied zwischen Translations- und Rotationsgelenk:

- Translationsgröße: $s_i = s_k + s_v$; s_k : konstanter Abstand vom Ursprung U_i zum Schnittpunkt z_i mit Normalen $s_v = 0$; s_v : Gelenkvariable.
- Drehwinkel: $\theta_i = \theta_k + \theta_v$; θ_k : konstanter Winkel zur Überführung von x_i nach x_{i+1} bei $\theta_v = 0$; θ_v : Gelenkvariable.
- a_i und α_i sind immer konstant

• Armgleichung:

$$E_{\text{Basis, Greifer}} = D_{0,1} \bullet D_{1,2} \bullet D_{2,3} \cdots \bullet D_{n-1,n} = \begin{pmatrix} x_{e_x} & y_{e_x} & z_{e_x} & x_s \\ x_{e_y} & y_{e_y} & z_{e_y} & y_s \\ x_{e_z} & y_{e_z} & z_{e_z} & z_s \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

• **Direktes kinematisches Problem:**

- Armgleichung ausrechnen.
- Gegeben $\theta_1, \theta_2, \dots, \theta_n \rightarrow \text{DkP} \rightarrow x, y, z$ (Gesucht)

• **Inverses kinematisches Problem:**

- Aus DH-Parametern und Stellung des Greifers sollen Gelenkwinkel bestimmt werden.
- Mit bekannten Kinematikmodellaten und gegebener Matrix E unbekannte θ_i aus Armgleichung ermitteln.
- Gegeben $x, y, z \rightarrow \text{IkP} \rightarrow \theta_1, \theta_2, \dots, \theta_n$ Gesucht
- **Lösung:**
 - * allgemeine Verfahren (algebraische Lösungen, Iteration, Reduktion der Ordnung der Armgleichung, Substitutionsverfahren): Nachteile: Aufwand hoch und lange Zeitdauer
 - * spezielle Verfahren (auf trigonoetrische/geometrische Beziehungen aufbauend (lokale Verfahren)): Vorteil: schnell; Nachteil: nur für spezielle Robotertypen
- Kinematisch überbestimmte Roboterkonfigurationen (siehe Folie 31)

2.3 Dynamisches Modell

- Bewegungsgleichung:

$$Q = M(q) \cdot \ddot{q} + n(\dot{q}, q) + g(q) + R \cdot \dot{q}$$

mit: Q : Vektor der allgemeinen Stellkräfte und -momente

$M(q)$: Trägheitsmatrix

$n(.,.)$: Vektor mit Zentrifugal- und Corioliskomponenten

$g(q)$: Vektor mit Gravitationskomponenten

R : Disgonalmatrix zur Beschreibung der Reibungskräfte

q : Vektor mit Winkellagen des Manipulators

- **Direktes dynamisches Problem:**

- Aus Masse, äußeren Kräften und Momenten sowie der Lage, Anfangsgeschwindigkeit und Anfangsbeschleunigung werden die sich ergebenden Bewegungsänderungen berechnet:
- Bewegungsgleichung nach $q(t), \dot{q}(t), \ddot{q}(t)$ auflösen.
- Gegeben $Q(t) \rightarrow \text{DdP}(q(t_0), \dot{q}(t_0), \ddot{q}(t_0)) \rightarrow q(t), \dot{q}(t), \ddot{q}(t)$ Gesucht

- **Inverses dynamisches Problem:**

- Aus den gewünschten Bewegungsparametern sowie den kinematischen Parametern die dazu erforderlichen Stellkräfte ermitteln:
- Bewegungsgleichung ausrechnen
- **Lösung:**
 - * Lagrange: (zur Ermittlung der Bewegungsgleichungen müssen nur die kinetische und die potentielle Energie aufgestellt werden. Die Bewegungsgleichungen folgen dann formal durch differenzieren.)
 - Vorteile: einfaches Aufstellen der Gleichungen, geschlossenes Modell, analytisch auswertbar
 - Nachteile: $O(n^4)$ (n = Anzahl der Gelenke)
 - * Newton-Euler: (Rekursive Berechnung der Kinematik von der Basis zum Greifer (Vorwärtsgleichungen), dann rekursive Berechnung der Dynamik vom Greifer zur Basis (Rückwärtsgleichungen))
 - Vorteile: für beliebige Anzahl von Gelenken, Belastungen der Armelemente werden berechnet, Aufwand $O(n)$
 - Nachteile: Rekursion

Kapitel 3

Bahnplanung

3.1 Grundlagen der Bahnplanung

- Bewegungen eines Roboters werden als **Trajektorien** (Zustandsänderungen über die Zeit relativ zu einem Koordinatensystem aufgefasst).
- Häufig sind Gütekriterien, Neben- und Randbedingungen zu berücksichtigen.
- Bekannt: Startzustand, Endzustand; Gesucht: Zwischenzustände, welchen den Startzustand in den Endzustand überführen.
- Zustandsräume: Gelenkwinkelzustandsraum (Konfigurationsraum), 3-dim. euklidischer Raum, Sensorzustandsraum, Objektzustandsraum, ...
- Unterscheidung der Bahnplanungsverfahren nach Art des Roboters: für Manipulatoren, für mobile Roboter, für Laufmaschinen und antropomorphe Systeme, Greif- und Montageplanung.
- **Bahnplanung im Gelenkwinkelraum:**
 - Spezifikation der Trajektorie als Funktion der Gelenkwinkelzustände des Roboters. Ausführung entweder Punkt-zu-Punkt (Steuerung der Achsen unabhängig voneinander) oder durch achsinterpolierte Steuerung (Bewegung aller Achsen beginnt und endet zum gleichen Zeitpunkt). Verlauf der punktweise in Gelenkwinkel spezifizierten Bahn muss im kartesischen Koordinatensystem nicht notwendigerweise bekannt sein.
 - Vorteile: Berechnung der Gelenkwinkel ist einfach und erzeugt keine Probleme mit Singularitäten ($\det(J(q(t))) = 0$)
 - Randbedingungen: Start- und Zielzustand bekannt, z.B. Geschwindigkeit zu Beginn und am Ende = 0, begrenzte Beschleunigung und begrenzter Gelenkwinkelvorrat.
- **Bahnplanung im 3-dim. euklidischen Raum:**
 - Angabe der Trajektorie als Funktion der Zustände des Roboters.
 - Funktionen: z.B.: lineare Bahnen, Polynombahnen, Splines
 - Vorteile: Definition des Verlaufs der Trajektorie erfolgt explizit im euklidischen Raum
 - Nachteile: Für jeden Trajektorienpunkt muss eine Gelenkwinkelrücktransformation berechnet werden; Geplante Trajektorie nicht immer ausführbar (begrenzter Arbeitsraum, Singularitäten des Roboters)

- **Bahntypen** (Bilder siehe Folien 12-17):
 - Punkt-zu-Punkt:
 - * Jedes Gelenk wird sofort mit der maximalen Beschleunigung angesteuert. Die Gelenkbewegung endet unabhängig von den anderen.
 - Punkt-zu-Punkt mit Achsinterpolation:
 - * Alle Gelenke beginnen und beenden ihre Bewegungen gleichzeitig.
 - Punkt-zu-Punkt linear:
 - * Die Robotersteuerung interpoliert die Bahn zwischen 2 Teiltrajektorien
 - Punkt-zu-Punkt mit Überschleifen (Glättung von Unstetigkeiten):
 - * Zum Zeitpunkt $t_j - \epsilon$ wird begonnen die Parameter (Richtung und Geschwindigkeit) der Teiltrajektorie $j - 1$ auf die Parameter der Teiltrajektorie j zu überführen. Normalerweise wird der Stützpunkt nicht erreicht.
 - Bahnapproximation:
 - * Bei n Zwischenpunkten wird die Bahn durch ein Polynom von $(n-1)$ -ter Ordnung approximiert.
 - Segmentweise Interpolation:
 - * Endbedingungen der Teiltrajektorie $j-1$ (Richtung, Geschwindigkeit, Beschleunigung) und Anfangsbedingungen der Teiltrajektorie j aneinander angleichen.

3.2 Problemklassen

- A. Bekannt: vollständiges Umweltmodell, vollständige Nebenbedingungen, Rand- und Zwangsbedingungen; Gesucht: Bahn vom Start- zum Zielzustand
- B. Bekannt: unvollständiges Umweltmodell, unvollständige Neben-, Rand- und Zwangsbedingungen; Gesucht: Bahn Start \rightarrow Ziel; Problem: Kollision mit unbekanntem Objekt.
- C. Bekannt: zeitvariantes Umweltmodell (bewegliche Hindernisse); Gesucht: s.o.; Problem: Hindernisse in Ort und Zeit variant.
- D. Bekannt: kein Umweltmodell, Gesucht: s.o.; Problem: Kartographieren
- E. Bekannt: zeitvariantes Umweltmodell; Gesucht: Bahn zu beweglichem Ziel (Rendez-Vouz-Problem); Problem: Zielzustand in Ort und Zeit beweglich.

3.3 Schema zur kollisionsfreien Bahnplanung

- Geg.: Modell des Roboters (Geometrie und Kinematik) und der Umwelt
 1. Berechnung Konfigurationsraum
 2. Berechnung Hindernisraum
 3. Berechnung Freiraum
 4. Zerlegung des Freiraums in Unterräume
 5. Bahnplanung im Unterraum
 6. Integration der lokalen Lösungen zur Gesamtlösung.

3.4 Verfahren zur kollisionsfreien Bahnplanung

- **Konfiguration K** : Beschreibt den Zustand eines Roboters; (im 3-dim. euklidischen Raum durch Lage und Orientierung; im Gelenkwinkelraum durch die Werte der Gelenke)
- **Konfigurationsraum \mathbb{K} des Roboters A** : Raum aller möglichen Konfigurationen von A
- **Weg**: für Roboter A von K_{Start} nach K_{Ziel} ist eine stetige Abbildung $\tau : [0, 1] \mapsto \mathbb{K}$ mit $\tau[0] = K_{\text{Start}}, \tau[1] = K_{\text{Ziel}}$
- **Arbeitsraumhindernis H** : Raum, welcher von einem Objekt im Arbeitsraum eingenommen wird.
- **Konfigurationsraumhindernis \mathbb{K}_H** : $\mathbb{K}_H = \{K \in \mathbb{K} \mid K \in H_i\}$
- **Hindernisraum**: $\mathbb{K}_{\text{Block}} = \bigcup_{i=1}^n \mathbb{K}_{H_i}$
- **Freiraum**: $\mathbb{K}_{\text{Frei}} = \{K \in \mathbb{K} \mid K \notin \mathbb{K}_{\text{Block}}\} = \mathbb{K} / \mathbb{K}_{\text{Block}}$
- Aufwand zur Berechnung des Freiraums $O(m^n)$ mit n : Anzahl der Freiheitsgrade des Roboters und m : Anzahl Hindernisse.

3.5 Planungsverfahren

- **Exakte Verfahren**: basiert auf exaktem (in algebraischer Form beschriebenen) Umweltmodell. Beispiel: Retraktion, Voronoi.
- **Approximierte Verfahren**: Umwelt durch Näherung beschrieben (Kuben, Zylinder, Polyeder, ...). Beispiel: Sichtgraph.
- **Straßenkarten**:
 - Gegeben: 2-dim. Weltmodell, Start und Ziel
 - Aufgabe: Suche günstigsten Weg von Start zu Ziel
 - Lösung:
 1. konstruiere Netz von Wegen \mathbb{W} in \mathbb{K}_{Frei}
 2. bilde K_{Start} auf K_{Ziel} ab: $(\mathbb{W}(K_{\text{Start}}), \mathbb{W}(K_{\text{Ziel}}))$
 - Wegkonstruktion mit: Retraktionsverfahren (z.B. Voronoidiagramm)
 - Suche im Netz mit z.B.: Baumsuche, wuklidischer Abstand, Potentialfeld
 - **Retraktion**: Sei X eine Menge und $Y \subset X$. Eine surjektive Abbildung $p : X \mapsto Y$ heisst Retraktion gdw. p stetig und $p(y) = y; \forall y \in Y$. D.h. Abbildung der Menge X auf ihre Teilmenge Y , wobei die Menge Y auf sich selbst abgebildet wird.
 - Für die Bahnplanung gilt: Y ist ein Netz von eindimensionalen Kurven (Wegenetz); Retraktionsmethoden unterscheiden sich in der Wahl von p .
 - Voronoi-Diagramm ist ein Beispiel für p : mittlerer Abstand (Folie 30)
 - **Sichtgraphen**: Konstruktion:
 - * Verbinde jedes Paar von Eckpunkten auf dem Rand von \mathbb{K}_{Frei} durch ein gerades Liniensegment, wenn das Segment kein Hindernis schneidet.

- * Verbinde q_{Start} mit q_{Ziel} analog dazu. Beispiele (Folie 32)
- Anmerkungen zur Sichtgraphmethode:
 1. Wege sind “halbfrei” (nicht kollisionsfrei), da Hinderniskanten auch Wegsegmente darstellen können. Abhilfe durch Erweiterung der Hindernisse (durch Kreis (um Roboter und Kreisradius um Hindernisse) oder durch Rechteck).
 2. Wenn ein Weg gefunden ist, ist es auch der kürzeste Weg (!!)
 3. Methode ist exakt, wenn der Roboter nur 2 translatorische Freiheitsgrade hat und alles durch polygone darstellbar ist.
 4. auch im \mathbb{R}^3 anwendbar. (Dann allerdings sind gefundene Wege nicht mehr unbedingt die kürzesten Wege).
- **Zellenzerlegungsmethoden:**
 - Lösung:
 1. Zerlege \mathbb{K}_{Frei} in solche Zellen, dass ein Weg zwischen 2 Konfigurationen innerhalb einer Zelle leicht zu finden ist
 2. Stelle Adjazenz (Nachbarschaft) Relation in einem Graphen dar
 3. Suche optimalen Weg von $q(\text{Start})$ nach $q(\text{Ziel})$ in dem Graphen
 - Exakte Zellenzerlegung:
 - * Lösung: Zerlege Freiraum so, dass Zellen disjunkt und Zerlegung vollständig.
 - * Exakte Zellenzerlegung mit “line-sweep”-Operator (Senkrechte Linie von jeder Ecke aus bis zum nächsten Hindernis oder zur “Wand”) Zellen nummerieren.
 - * Verbindungsgraph der freien Unterräume: Zellen = Knoten, Kanten = freie Übergänge zwischen benachbarten Zellen.
 - Approximative Zellenzerlegung:
 - * Lösung:
 1. Zerlege Freiraum in Zellen vordefinierter Form (z.B. Rechtecke, Quadrate)
 2. Wenn Zelle nicht vollständig im Freiraum liegt, verringere Größe und zerlege Zelle weiter (z.B. Quadtree).
 3. Diese Schritte nur bis zu einer Minimalgröße der Zelle anwenden.
 - * Vorteil: einfache Zerlegung → einfache Wegsuche
 - * Nachteil: Freiraum kann i.A. nur annähernd beschrieben werden.
 - * Approximative Zellenzerlegung: Graphengenerierung: Quadtree: Zerlege Quadrat in vier gleich grosse Teiquadrate. Baumstruktur mit Nummern 0-3 (Nummerierung im Gegenuhrzeigersinn, Anfang unten links im Quadrat; Bei Unterteilung eines Quadrates → akt. Knoten wird Vaterknoten von neuer Vierergruppe). (Siehe Folie 41-44)
- **Potentialfelder:**
 - Roboter bewegt sich unter dem Einfluss von Kräften, welche ein potentialfeld auf ihn ausübt. (Abbildung U von \mathbb{K}_{Frei} nach \mathbb{R}).
 - Ziel hat Anziehungspotential
 - Hindernisse haben Abstossungspotential
 - Die Summe der einwirkenden Kräfte bestimmen die Bewegung.
 - Eigenschaften Abstossungspotential:

- * In geringem Abstand zu Hindernissen sollen keine Potentialbarrieren (lokale Minima) entstehen, welche der Roboter nicht durchqueren kann.
- * In großem Abstand zu Hindernissen soll der Roboter nicht beeinflusst werden.
- Eigenschaften Anziehungspotential:
 - * Es soll nur *ein* Minimum in $q(\text{Ziel})$ geben
- Lokale Minima:
 - * Problem: Durch Summation kann U lokale Minima besitzen, in denen der Roboter steckenbleiben kann.
 - * Lösung: An-/Abstossungspotentiale so wählen, dass kein lokales Minimum (ausser $q(\text{Ziel})$) existiert; Im Suchalgorithmus Techniken zur Flucht aus lokalen Minima anwenden.
- Andere Bahnplanungsverfahren: Sensorgestützte Bahnplanung (Klassen B,D); Algebraische Methoden (bei einfachen Umweltmodellen); Konnektionistische Methode (Wissen über Umwelt wird implizit in neuronalen Netzen repräsentiert).

3.6 Komplexität des Wegsuche-Problems

Verringerung der Komplexität durch:

1. Projektion in \mathbb{K}' . (Verringern der Dimension; Entspricht dem fixieren eines oder mehrerer Parameter).
2. Unterräume in \mathbb{K} .
3. Kombination obiger Verfahren.
4. Zerlegung des Arbeitsraumes in Teilräume mit: vielen Hindernissen (=exakt planen) oder wenig Hindernissen (=approximativ planen)
5. Geometrische Vereinfachungen: Erweiterung des Hindernisraumes und Minimierung des Roboters durch:
 - (a) Referenzpunkt auf Rand des Roboters beliebig aber fest
 - (b) Erweitere jeden Hindernisraum durch Spur des Referenzpunktes, wenn Roboter um das Hindernis herumbewegt wird.
 - (c) Schrumpfe Roboterraum zu einem Punkt
6. (Siehe Folie 58)

3.7 Erweiterung des Wegsuche-Problems

- Zeitvariantes Umweltmodell (Hindernisse bewegen sich); Lösung: Mache \mathbb{K} zeitabhängig ($\mathbb{K}T(t_i) = \mathbb{K}$ zum Zeitpunkt t_i).
- Mehrere Roboter:
 - Zentralisierte Planung: Bilde Gesamtkonfigurationsraum (durch Kreuzprodukt aller beteiligter Konfigurationsräume), plane in \mathbb{K}_{ges} mit bekannten Methoden.
 - Entkoppelte Planung: Plane für jeden Roboter in seinem Konfigurationsraum; Untersuche mögliche Interaktionen zwischen den Wegen.

Kapitel 4

Greifplanung

- Die menschliche Hand besitzt 16 Gelenke. Daraus resultieren 22 Freiheitsgrade. Modellierung durch kinematisches oder flächenbasiertes Modell (siehe Folie 4,5)

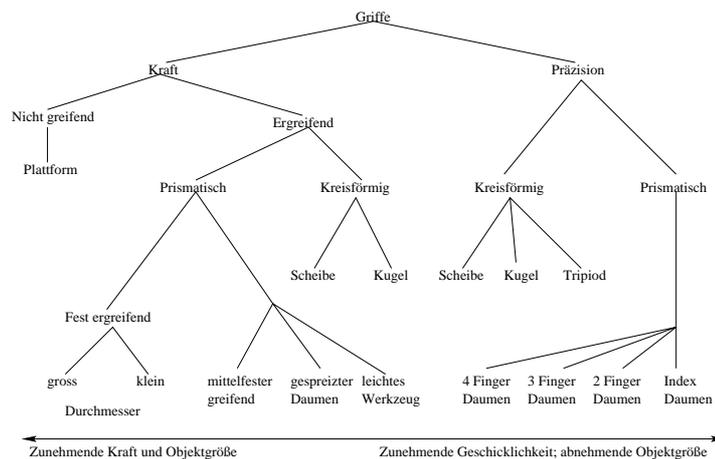


Abbildung 4.1: Cutkosky Griff-taxonomie

4.1 Grundlagen

- Es lassen sich folgende Bewegungstypen unterscheiden:
 - **Greifen bzw. Loslassen eines Objektes:** Auswahl eines sicheren Griffes d.h. Bestimmung einer geeigneten geometrischen Relation der Greiffinger zum gegriffenen Objekt; Dabei sind Kollisionen zu vermeiden.
 - **Anrückbewegung bzw. Abrückbewegung des Greifers:** Planung der Bewegung, um Greiffinger entsprechend zu positionieren und zu orientieren; Dabei sind Kollisionen zu vermeiden.
 - **Anrückbewegung bzw. Abrückbewegung des Greifers mit gegriffenem Objekt:** Kollisionen sind zu vermeiden.

- **Verbinden des gegriffenen Objektes mit anderen Objekten:** Verwendung sensorüberwachter und/oder sensorgeführter Bewegungen.
- **Transferbewegung des Greifers ohne/mit gegriffenem Objekt:** höhere Ausführungsgeschwindigkeit und geringere Genauigkeitsanforderungen.
- **Interne Nebenbedingungen des Greifvorganges (I):**
 1. **Gültigkeit des Griffes:** Überlappung zwischen den Greifmerkmalen des zu greifenden Objektes und den Greifmerkmalen der Greiffinger.
 2. **Kollisionsfreiheit eines Griffes:** keine Kollisionen zwischen Greifer und gegriffenem Objekt.
 3. **Zugänglichkeit eines Griffes:** Griff ist für den Greifer kollisionsfrei zu erreichen.
- **Externe Nebenbedingungen (E):**
 1. **Kollisionsfreie Anrückbewegung des Greifers:** keine Kollisionen zwischen Roboterarm, Greifer, Objekt, Arbeitsfläche.
 2. **Kollisionsfreie Abrückbewegung des Greifers:** s.o.
 3. **Berücksichtigung der Roboterkinematik:** selektierte Griff muss im Arbeitsraum des Roboters liegen und zugehörige Trajektorien durch Roboter abfahrbar.
 4. **Stabilität des Griffes:** die relative Lage und Orientierung des zu greifenden bzw. gegriffenen Objektes zum Greifer ändert sich während Greif- und Transferbewegung nicht.
 5. **Stabilität der Szene:** Abrückbewegung des Greifers sollte die Stabilität der Szene nicht beeinflussen.
 6. **Aufgabenabhängigkeit eines Griffes:** Wählen eines zur Aufnahme- und Ablagekonfiguration des zu greifenden Objektes kompatiblen Griffes. Evtl. Konsequenzen:
 - Kann kein Griff bestimmt werden → geeignete Umgreifsequenz
 - Erfordert ein Griff die Ausübung von Prozesskräften und -momenten, so muss ein entsprechend kompetenter Greifer gewählt werden.
- **Planungsschritte zur Generierung von Greifoperationen:**
 - Siehe Folie 17: Kreislauf: Griffanalyse → interne und externe Nebenbedingungen → Griffsynthese → Griff ◯
- **Fingerspitzengriff-Modell:**
 - Vereinfachung der Algorithmen zur Synthese möglicher Griffe, da nur I1 bestimmt werden muss.
 - Nachteil: Nichtbeachtung fundamentaler Nebenbedingungen I2 und I3
- **Fingerspitzenkontakte mit der Oberfläche (man unterscheidet verschiedene Kontakte unter folgenden Annahmen):**
 - Punktkontakt ohne Reibung: Kraft wirkt ausschliesslich normal zur Fläche.
 - Starrer Punktkontakt mit Reibung: Kraft wirkt sowohl normal als auch tangential zur Fläche. Beide Kräfte über das Coulombsche Reibungsgesetz miteinander verknüpft.
 - Nichtstarrer Punktkontakt mit Reibung: Kraft wirkt sowohl normal als auch tangential zur Fläche.

4.2 Griffhierarchie

- **Wrenchvektor:** \vec{w} er beinhaltet die an einem Punkt \vec{p}_i wirkenden Kräfte f_i und Momente τ_i mit $i \in [x, y, z]$. Für diesen gilt:
 - planarer Griff: $\vec{w} = (f_x, f_y, \tau_z)^\top \in \mathbb{R}^3$
 - räumlicher Griff: $\vec{w} = (f_x, f_y, f_z, \tau_x, \tau_y, \tau_z)^\top \in \mathbb{R}^6$
 - Im Folgenden werden Wrenchvektoren aufgeführt, welche die am Kontaktpunkt wirkenden normalen (n) und tangentialen (t) Kräfte und axialen Momente (θ) beschreiben. Bzgl. des i-ten Kontaktpunktes werden sie bezeichnet: ${}^i\vec{w}_n, {}^i\vec{w}_t, {}^i\vec{w}_\theta$, die dazu korrespondierenden Skalare sind mit ${}^i c_n, {}^i c_t, {}^i c_\theta$ bezeichnet.
- **Greifmatrix:**
 - Bei m Kontaktpunkten für ienen räumlichen Griff: $6 \times 3m$ -Matrix $G = [{}^1\vec{w}_n, {}^1\vec{w}_t, {}^1\vec{w}_\theta, \dots, {}^m\vec{w}_n, {}^m\vec{w}_t, {}^m\vec{w}_\theta]$ Für die Skalare erhält man den Vektor: $\vec{c} = ({}^1 c_n, {}^1 c_t, {}^1 c_\theta, \dots, {}^m c_n, {}^m c_t, {}^m c_\theta)^\top \in \mathbb{R}^{3m}$
 - Auf G wirkendes externes Moment (meist Erdanziehung etc.): $\vec{\ell} = (f_x, f_y, f_z, \tau_x, \tau_y, \tau_z)^\top \in \mathbb{R}^6$
- **Gleichgewichtsgriff:**
 - Wenn die Summe aller Kräfte f_i und Momente τ_i , die auf das gegriffene Objekt wirken = 0 ist:
 1. $\forall i \in [1, m] : {}^i c_n \geq 0, {}^i \mu_t \star {}^i c_n \geq |{}^i c_t|, {}^i \mu_\theta \star {}^i c_n \geq |{}^i c_\theta|$
 2. $\exists \vec{c} \in \mathbb{R}^{3m}, \vec{c} \neq \vec{0} : G\vec{c} + \vec{\ell} = \vec{0}$
 - 1. besagt: c_t und c_θ sind durch die korrespondierenden Normalenkräfte und die Coulombschen Reibungskoeffizienten nach oben begrenzt.
 - Beispiel: Folie 26
- **Kraftgeschlossene Griffe:**
 - Die Stabilität eines Griffes erfordert, dass das gegriffene Objekt im Kräftegleichgewicht bleibt, d.h. Kräfte und Momente, welche durch die Greiffinger auf das gegriffene Objekt ausgeübt werden, müssen sämtliche externen Kräfte und Momente kompensieren können,
 1. Es gilt Gleichung 2 vom Gleichgewichtsgriff, welche $\forall \vec{c} \in \mathbb{R}^6$ gelten muss.
 - Für die Anzahl der benötigten Kontaktpunkte eines kraftgeschlossenen Griffes gilt:
 - * basierend auf Punktkontakten *ohne* Reibung: Wenn Objekt keine Rotationssymmetrie besitzt: mindestes 4 Kontaktpunkte, für beliebige 3d-Objekte höchstens 12 Kontaktpunkt, beschränkt auf Klasse der Polyeder: höchstens 7 Kontaktpunkte.
 - * basierend auf Kontaktpunkten *mit* Reibung: planare Objekte: auf 3 Kontaktpunkte basierender kraftgeschlossener Fingerspitzengriff; räumliche Objekte: untere Grenze von 4 Kontaktpunkten.
 - Beispiel: Folie 31
- **Formgeschlossene Griffe:**
 - Es werden lediglich ${}^i\vec{w}_n$ in G berücksichtigt. Also $G' \in \mathbb{R}^{6 \times m}$.

- Für einen formgeschlossenen planaren Griff sind mindestens 4 Kontaktpunkte erforderlich. Bei beliebigen 3d-Objekten mindestens 6 Kontaktpunkte (im Skript stehen 7, aber z.B. Würfel?).
- Hier gilt das gleiche Kriterium, wie bei kraftgeschlossenen Griffen, nur in reduzierter Form (da nur eine von drei Kraftkomponenten berücksichtigt wird.)

- **Stabile Griffe:**

- Zur Modellierung von Fingerkräften, welche durch ihre Nachgiebigkeit kleine Änderungen von der Nominallage des gegriffenen Objektes kompensieren können, folgende Definition einer Potentialfunktion: $V : \mathbb{R}^6 \rightarrow \mathbb{R}$. Sie spezifiziert die im Griff gespeicherte potentielle Energie in Abhängigkeit von Lage und Orientierung des gegriffenen Objektes.
- Sei Objekt in Gleichgewichtgriff, $\delta q = (\delta_x, \delta_y, \delta_z, \delta_{\phi_x}, \delta_{\phi_y}, \delta_{\phi_z})^\top \in \mathbb{R}^6 \neq \vec{0}$ eine infinitesimale Lageänderung des gegriffenen Objektes und δV die resultierende Änderung der potentiellen Energie, so ist der Griff **stabil**, falls: $\forall \delta q \in \mathbb{R}^6 : \delta V > 0$
- Es gibt Griffe (siehe Folie 38), welche stabil sind, aber nicht kraftgeschlossenen.

- Mengentheoretische Darstellung der Grifffhierarchie: formgeschlossene Griffe \subset kraftgeschlossene Griffe \subset stabile Griffe \subset Gleichgewichtsgriffe.

4.3 Klassifikation von Greifplanungssystemen

Greifplanungssysteme können nach folgenden Kriterien klassifiziert werden:

1. Typ des Greifers (Zweifinger-, Dreifinger-, Parallelbackengreifer)
 2. Typ der zugrunde liegenden Greifplanungsalgorithmen (geometrisch basiert (berücksichtigen CAD-Daten), physikalisch basiert (Kräfte, Momente))
 3. Typ der zu greifenden Objekte (Polygone, Polyeder, ...)
 4. Typ der Szenen (deterministisch (Lage und Orientierung aller Objekte in Szene bekannt), nicht deterministisch,...)
 5. Einsatz von Sensorik (keine -, taktile -, visuelle Sensoren)
- Suchraum: Dimension des Suchraums sowohl für den physikalischen als auch geometrischen Ansatz = $6 + n$, mit n = Anzahl der Konfigurationsparameter der Greiffinger. (6 = Lage und Orientierung der Hand im Raum). Beim Parallelbackengreifer: 7-dim. Suchraum. bei Karlsruher Hand 15-dim. ($3 \cdot 3 + 6$) und bei einer menschlichen Hand 22-dim..
 - Polyedrische Approximation, Geometrieparameter und Greiferreferenzsystem: Greiferreferenzsystem/TCP wird so gewählt, dass sich der Greiferreferenzpunkt zwischen den Fingerspitzen befindet. Die Vektoren bilden ein orthonormales Rechtssystem (siehe Folie 45).
 - **Greifobjekte:**
 - starre, polyedrische Körper (Ecke, Fläche, Kante). Keine Einschränkung, da Körper, welche ausserhalb dieser Klasse existieren approximiert werden können.
 - **Objektmodell:**

- Zur Bestimmung möglicher Griffe ist eine Repräsentation notwendig, welche explizit die begrenzenden Flächen, Kanten und Ecken beschreibt. Das zugrundeliegende Objektmodell beinhaltet die geometrische und topologische Beschreibung des Polyeders.
- Somit kann ein Objekt als zusammenhängender Graph modelliert werden. Die Knoten bilden die Ecken, Kanten und Flächen und die Kanten beschreiben die Topologie.
- An Eckpunkten laufen 3 oder mehr Flächen zusammen.
- Eine Kante ist der Schnitt von zwei Flächen
- Eine Fläche wird durch ein planares Polygon beschrieben, nur begrenzt von den Kanten des Polyeders.
- Graph siehe Folie 49.

- **Symbolische Griffe:**

- Hauptaufgabe der modellbasierten Greifplanung ist die Generierung möglicher Griffe eines Objektes O . Da unendlich viele Griffe existieren, werden während der Offline-Phase **Griffklassen** eines Objektes bestimmt.
- Erster Schritt zur Generierung von Griffklassen ist die Generierung symbolischer Griffe. Sie ergeben sich aufgrund von Kombinationem geometrischer Primitive (Fläche, Kante, Ecke), die im Griff mit dem Greiffinger in Kontakt treten. Die geometrischen Primitive $gP(O)$ werden **Greifmerkmale** genannt.
- Ein symbolischer Griff besteht aus der Kombination zweier Greifmerkmale, die **geometrischen Greifbedingungen** (folgen aus der Greifer- und Objektgeometrie) genügen müssen.
- Die Menge der symbolischen Griffe eines Objektes O , welches mit dem Greifer G gegriffen werden soll, ist: $S(O, G) = \{(M_1, M_2) | M_1, M_2 \in gP(O) \wedge M_1 \neq M_2 \wedge \text{geometrische-Greifbedingung}(M_1, M_2, G) = \text{erfüllt}\}$

- **Geometrische Greifbedingungen:**

- Basierend auf der Objektgeometrie:
 - * Parallele Zugänglichkeit einer Greifmerkmalskombination
 - * Lokale Zugänglichkeit eines Greifmerkmals
 - * Planare lokale Zugänglichkeit
- Basierend auf Objekt- und Greifergeometrie:
 - * Kontakt mit Greiferfingern
 - * Minimale und maximale Öffnungsweite des Greifers
- Siehe Bild auf Folie 54 (konvexe und konkave Kanten (zugänglich/nicht zugänglich))

4.4 Planung von Umgreifoperationen

- Nebenbedingungen einer Umgreifoperation:
 1. U-operation muss kollisionsfrei durchführbar sein
 2. U-operation muss kinematisch durchführbar sein
- Beispiel: Folie 56 (Parallelbackengreifer soll ein Stück um 90° drehen und an ein anderes mit Kontakt dranstellen)

4.5 Szenenstabilität

- Wird die Szenenstabilität beeinflusst, so sollte eine andere Reihenfolge der Greifoperationen gewählt werden. Beeinflusst die Ausführung die Szenenstabilität nicht, so kann auf Einsatz von Sensorik verzichtet werden.
- **Kontaktmodell:**
 - Kontaktgeometrie der zu manipulierenden Szene ist Grundlage der Stabilitätsanalyse. Diese Geometrie wird durch die Kontaktpunkte mit den korrespondierenden Normalenvektoren beschrieben.
 - Typen polyedrischer Kontakte: nicht-paralleler Kanten-Kanten-Kontakt, paralleler Kanten-Kanten-Kontakt, Kanten-Ecke-Kontakt
- **Stabilitätsanalyse mit einem Kontaktgraphen:**
 - Kontaktgraph: Dieser ist ein einfacher gerichteter Graph, wobei die Knoten $K_i \in [1, m]$, zu den Komponenten und die Kanten $e_i \in [1, n]$ zu den physikalischen Kontakten der Baugruppe korrespondieren. (Siehe BEispiel auf Folie 61).
- **Stabilitätsanalyse über virtuelle Verschiebungen:**
 - Dieses Verfahren beschreibt die Änderung der potentiellen Energie einer Konfiguration starrer Körper unter Wirkung von Gravitationskräften. Dies führt im Falle einer instabilen Konfiguration zu einer Erhöhung der kinetischen Energie.
 - Grundidee: Bestimme diejenige Verschiebung aus der Menge aller gültigen, virtuellen Verschiebungen aller Körper, die die größte Verminderung der potentiellen Energie bewirkt. Kann keine gefunden werden, so wird die Konfiguration als stabil klassifiziert.
- Planung von Greifoperationen:
 1. Greifklassenselektion
 2. Kollisionsfreiheit der An-/Abrückbewegung
 3. Bestimmung von Fingerhindernissen
 4. Bestimmung von Basishindernissen
 5. zugängliche Griffe
 6. erreichbare Griffe
 7. Griffselektion

Kapitel 5

Architektur von Robotersystemen

5.1 Grundlegende Fähigkeiten

- Umwelt, Perzeption, (Erfassung, Planung, Lernen, Ausnahmebehandlung, Ausführung, Motorik(Wissensbasis)), Aktion, Umwelt...

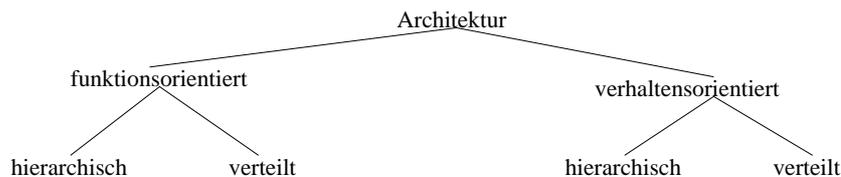


Abbildung 5.1: Klassifikation von Architekturen für Roboter

5.2 Schematische Darstellung der vier Basisarchitekturen

- **hierarchisch funktionsorientiert:** z.B. NASREM, Mehrere Ebenen (Detaillierung/Abstraktion), mehrere Spalten (Sensorik, Modellierung, Planung, Ausführung)
- **verteilt funktionsorientiert:** die vier Spalten (s.o.) sind mit einem zentralen Kommunikationsmechanismus verbunden.
- **hierarchisch verhaltensorientiert:** Aufgabenteilung als Stapel (mehrere Schichten, pro Schicht ein Verhalten). z.B. Erforsche Deine Umwelt, Bewege Dich ziellos umher, Vermeide Kollisionen.
- **verteilt verhaltensorientiert:** mehrere Felder, pro Feld Verhaltensvorrat (s.o.)

5.3 Hierarchisch funktionsorientierte Architekturen

- Mehrere Schichten (oben: komplexe Ebene, unten: primitive Ebene); drei Spalten (Sensoren, Weltmodell, Planung)
- Aufteilung in 4-6 Ebenen

- je Ebene drei “Module”
 - G_i : Sensorverarbeitungsmodul
 - M_i : Weltmodell- und Referenzmodul
 - H_i : Taskzerlegungs-, Planungs- und Ausführungsmodul
- jede Modul-Art ist durch die Ebeneneinteilung hierarchisch geordnet
- Abstrakte Anwendungen werden in zeitliche Aktionsfolgen zerlegt. (z.B. Komplexe Fertigungsaufgabe \rightarrow Einzelaufgaben \rightarrow Task \rightarrow Elementaroperation \rightarrow Bewegungsprimitive \rightarrow Koordinatentransformation, Servoregelung)
- Informationsfluss: In Ebene Interaktion mit Weltmodell (M-Modul), in G-Spalte von unten nach oben (Input = elementare Sensordaten), in H-Spalte von oben nach unten (Aufgabenzerlegung)
- Bei H-Spalte: Ausgabevektor: $P_i \sim C_{i-1}$ (nach unten), Anweisungsvektor: C_i (von oben), Sensorvektor F_i (von M-Modul), Steuerungsanweisung: C_i , Eingabevektor $S_i = C_i + F_i$

5.3.1 H-Modul

1. Das H-Modul erhält eine Aufgabe aus der oberen Ebene.
 2. Mit den Daten aus dem M-Modul wird die Aufgabe in Teilaufgaben zerlegt.
 3. Das M-Modul wird aktualisiert.
 4. Die Teilaufgaben werden (einzeln) an die untere Ebene weitergereicht.
- Aufgabenzerlegung in konkurrierende oder parallele Prozesse, diese werden lokal geplant und in Teilpläne zerlegt, welche dann abgearbeitet werden.

5.3.2 G-Modul

1. Das G-Modul bekommt Sensor-Informationen aus der unteren Ebene.
 2. Mit dem M-Modul werden diese Daten verarbeitet.
 3. Das M-Modul wird aktualisiert.
 4. Die Informationen werden der oberen Ebene zur Verfügung gestellt.
- elementare Sensordaten werden mit Vorhersage lokal verglichen, Abweichungen werden zeitlich integriert, alle lokalen Daten werden dann räumlich zu einem Gesamtergebnis integriert, dieses wird über das Weltmodell klassifiziert und das Weltmodell ggf. aktualisiert.

5.3.3 M-Modul

1. Es besteht aus einem einzigen Modul, aus welchem die G- und H-Module Daten mit dem Abstraktionsgrad der jeweiligen Ebene extrahieren.
2. Aufgrund einer Differenz zwischen den Solldaten (H-Modul) und den Ist-Daten (G-Modul) wird ein Fehler erkannt.

5.3.4 Operationen des H-Moduls

- Der Interpreter erhält eine Taskbeschreibung und reicht Teilaufgaben an Planer weiter. Diese geben unter Verwendung eines lokalen Güteindex Subtasksequenzen an lokale Ausführungskomponenten weiter. Diese erhalten lokal Sensordaten und geben Anweisungsfolgen nach unten weiter und erhalten von dort Statusinformationen.
- Zerlegung einer Aufgabe in Teilaufgaben (mit Zwischenzielen).
- Die Eingabefolge als Graph (Befehle = Knoten, Sensordaten = Kanten/Transitionen) kann entweder einfach, linear sein (Folie 24) oder ein verzweigtes Netz (Folie 25) darstellen. Die Sensordaten bestimmen dann, in welchen Folgeknoten gesprungen wird.
- $S=C+F$, beim Servomechanismus kommt noch eine Regelgröße P (skalar) hinzu. C ist die Sollgröße. Der H -Operator beinhaltet den Regler.
- Es ergibt sich dann Trajektorie (Zustandsänderung über die Zeit) von C , über (mehrere) F nach Trajektorie von S , diese über H nach Trajektorie von P

5.3.5 Operationen des G-Moduls

- Sensorverarbeitung ohne Kontextinformation: Abbildung von Sensormustern E auf Klassen Q
- Sensorverarbeitung mit Kontextinformation: Abbildung vom D -Raum (Sensormuster + Kontextinformation) in den Klassenraum. Einzelpunkte auf Trajektorie in Klassenraum korrespondieren mit mehreren Punkten auf Trajektorie in D -Raum.

5.4 Informationshierarchie

- Über Schnittstellen.
- Schnittstelle einer elementaren Steuerungsebene:
 - Generische Steuerungsebene hat jeweils eine Schnittstelle zur über- und unter ihr liegenden Generischen Steuerungsebene, zur Wissensbasis (ebenenübergreifend) und zur Sensorverarbeitung und Weltmodell (ebenfalls ebenenübergreifend).
- Hierarchisch gegliederte Abstraktionsebenen (auch Bauebenen), (* = nachfolgend Schnittstelle zwischen Tasksteuerungsebene und Robotergelenkebene möglich):
 - Task
 - Elementare Bewegung *
 - Einfache Bewegung *
 - Koordinierte Gelenke *
 - Servo

5.5 Verteilte funktionsorientierte Architekturen

- Menge spezialisierter Teilsysteme
- Kommunikation über zentrale Kompetenz
- Beispiel: Nav-Lab System von CMN(?)

5.6 Hierarchisch verhaltensorientierte Architekturen

- Steuerung durch Verhaltensmuster bzw. Reflexe
- Muster: Reaktion des Systems auf bestimmte Sensorstimuli
- Anordnung in Verhaltensebenen
- Hierarchische Struktur der Kompetenzebenen
- Beispiel: Subsumption Architecture (R.Brook, MIT)

5.7 Verteilt verhaltensorientierte Architekturen

- Menge von unabhängigen Teilsystemen mit identischen Verhaltensmustern
- Koordination über ein Verhaltensmuster
- Beispiel: Multi-Agenten-Systeme:
 - Eigenschaften: Selbstorganisation, assoziative Speicherung von Information, Erkennung bestimmter Muster.

Kapitel 6

Planungssysteme

- Ein Planungssystem ist ein System, das ausgehend von einem Anfangszustand und der Beschreibung eines gewünschten Zielzustandes eine Folge von Aktionen generiert, welche das betrachtete System von seinem Anfangszustand in den gewünschten Zielzustand überführen.

6.1 Beispiel eines Planungssystems

- Planungsphasen:
 - Weltmodellierung
 - Aufgabenspezifikation
 - Aufgabenanalyse
 - Erstellen von Vorranggraphen
 - Planung von Ausführungsdetails
 - Ausführung

6.1.1 Cranfield-Montage-Benchmark

wurde bei “Programmieren durch Vormachen” verwendet. Siehe Folien 6-7.

6.1.2 Planung und Überwachung der Montage

Produkt- und Aufgabenbeschreibung → Vorranggraph → Erweiterter Vorranggraph → Strategische Planung → Symbolische Aktionsplan → Geometrische Planung → Geometrische detaillierte Planung → Generierung → Roboteranweisung.

6.1.3 Aufgabenspezifikation

- Modellierung der Anfangs- und Endkonfiguration der Aufgabe
- Konfiguration besteht aus: [Geometrie (3D), physikalische Eigenschaften (Gravitationszentrum, Material), kartesischen Koordinaten] des Werkstücks.
- raphischer Editor zur: Spezifizierung der Greifkonfigurationen und des Ortes des Werkstücks.

6.1.4 Aufgabenanalyse

Kriterien: Keine Durchdringung, keine Seiteneffekte, Stabilität.

6.1.5 Vorranggraph des Cranfield-Benchmarks

bei Bedarf Folie 15.

6.1.6 Detailplanung

- Roboterauswahl
- Layoutplanung: (Finden einer Roboterposition (Kriterien: Erreichbarkeit, Genauigkeit, Effizienz))
- Bewegungsplanung (kollisionsfrei.)
 - Bewegungen zum Greifen und Loslassen eines Werkstückes.
 - Feinbewegungen um eine gezielte Berührung zweier Werkstücke hervorzurufen.
 - Grobe Manipulatorbewegungen zum Transport eines Werkstücks von einem Ort zum anderen.

6.1.7 Synthese der Vorranggraphen

- Verschmelzung des Regelsatzes, der aus den Analysekriterien abgeleitet wird.
- Auswahl des “besten” Vorranggraphens
- Eliminierung redundanter Kanten
- Detektion von Zyklen

6.1.8 Das gesamte System

Aufgabenspezifikation → Anfangs- und Endzustand → Analyse 1 .. Analyse n → Einschränkungen
→ Synthese → Vorranggraph → Detailplaner → Operationsparameter → Ausführung.

Kapitel 7

Laufmaschinen

- Vorteile von Laufmaschinen:
 - nur wenig erreichbare Aufsetzpunkte, hohe Mobilität, Schonung des Untergrunds, bessere Traktion bei weichen Böden, dynamische Stabilität
- Schwierigkeiten:
 - Echtzeit
 - hohe Materialbelastung
 - hoher Energieverbrauch
 - geringe Nutzlast
 - hohe Entwicklungskosten
- Einsatzgebiete:
 - unstrukturiertes Gelände, Meeresboden, Erkundung fremder Planeten, Katastropheneinsatz, Arbeitsmaschinen, Serviceroboter, Inspektionsroboter, Beinprothesen, Unterhaltungsindustrie.
- Schwerpunkt der Laufmaschinenforschung der 90er Jahre:
 - Anwendungen (Planetenerkundung, sehr unzugängliches Terrain)
 - Kletterroboter
 - Natur als Vorbild für Laufmaschinen
 - Vierbeiner mit dynamisch stabilen Gang
 - Optimierung der Zweibeiner
- Sinnesorgane:
 - Reize von Aussen:
 - * Druck, Zug
 - * Verformung der Körperdecke
 - * Vibration
 - * Luft- und Wasserströmung
 - * Schwerkraft

- * Schall, Licht, Linear- und Drehbeschleunigungen
- Körpereigene Reize:
 - * Druck und Spannung von der Körperdecke
 - * Dehnung und Länge der Muskeln
 - * Gelenkstellungen
 - * Bewegungen
- Natur als Vorbild:
 - Übernahmefähig:
 - * Anordnung und Aufbau der Beine, Gelenkprinzip, Muskelähnliche Antriebe
 - * Bewegungsablauf, Verarbeitungsprinzip von Sinnesorganen
 - * Prinzip der Sensorfusion, Konzept der Informationsverarbeitung
 - Nicht übernahmefähig:
 - * Materialien, Energieversorgung, Aufbau der Sinnesorgane
 - * Genaues Funktionsprinzip der neuronalen Informationsverarbeitung
- Pantograph-Mechanismus:
 - entkoppelte Bewegungen
 - einfache Steuerung von Trajektorien
- Sensorik für Laufmaschinen:
 - Messung der Beinzustände (Zustand der Antriebe, Gelenkwinkelmessung, Abstandsmessung, Taktile Information, Kraftmessung im Fuß, Verspannungsmessung)
 - Zustandsbestimmung des Zentralkörpers (Lage, Beschleunigung und Geschwindigkeit)
 - Sensoren zur Umwelterkennung (Abstandsbilder, einfache optische und akustische Signale)
- Messung der Beinzustände.
 - Aufgabe: Messung von (Biegekräften, Bodenkontakt, Abstand vom Boden, Stellung, Kraftverteilung, Kontakt des Beins mit Objekten aus der Umwelt)
 - Ziel: energetisch günstige Beinbewegung, keine Zerstörung der Mechanik, Kraft im Fuß so, dass Bewegung möglich.
- Definitionen zur Beschreibung des Laufens.
 - **Stemmphase**: in dieser Phase befindet sich das Bein auf dem Boden, stützt den Körper und schiebt ihn vorwärts. In der **Schwingphase** hebt das Bein vom Boden ab und schwingt zur Anfangsposition der nächsten Stemmphase.
 - Die **Stemmzeit** s_i ist die Dauer der Stemmphase von Bein i . Analog **Schwingzeit**.
 - **Beinschlag** ist die Entfernung von Anterior Extreme Position bis Posterior Extreme Position.
 - **Beinabstand** ist der Abstand zwischen den Mittelpunkten des Beinschlags zweier benachbarter Beine derselben Körperseite.
 - **Zykluszeit** = Stemmphase + Schwingphase

– ...

- **Steuerungskonzepte:**

- Neuroethologische Steuerungskonzept. Es basiert auf Lernen und Anpassung als Schlüsselkomponenten bei der Bewegungssteuerung.
- Bei Laufmaschinen. Nachteile: Sehr langsame Lerngeschwindigkeit, Vorteil: wenig Speicherbedarf, hohe Fehlertoleranz, Echtzeitfähigkeit erfüllt.

- **Reinforcement Learning:**

- Online-Lernen, inkrementelles Lernen, Sicherheits- und Robustheitsaspekte. Behalten von “angenehmen” und “unangenehmen” Situationen.

- Lernarchitektur: selbstorganisierende Neuronale Netze.

Kapitel 8

Service-Roboter

- Ein Serviceroboter ist eine freiprogrammierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dies sind Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen an Menschen und Einrichtungen dienen.
- Der Entwicklungstrend geht immer mehr in Richtung unstrukturierterer Umwelt und steigender Autonomie; in Richtung "Personal Robot".
- Einsatzbereiche: Baugewerbe, Medizin, Rehabilitation, Kommunalwesen, Hotel und Gastronomie.
- z.B. ABM (Müllaster, teilautomatisierte Aufnahme von Behältern, Detektion mit Hilfe von Laserscannern)
- Baugewerbe: Mauerroboter, Betonspritzroboter
- Medizin: Robodoc, Laparobot, Caspar
- **Basisfunktionen eines Serviceroboters:**
 - Orientierung in Bewegung
 - Umgebungserfassung
 - Umweltmodellierung
 - Ortung
 - Bahnplanung
 - Bewegungsführung
 - Sicherheit und Kollisionsschutz
 - Handhabung
- **Teilsysteme von Servicerobotern:**
 - handhabungsarm
 - Endeffektor
 - Mobile Plattform
 - Antriebstechnik

- Steuerung
 - Sensorik
 - Mensch-Maschine-Schnittstelle
 - Sicherheitssystem
- Kanalroboter