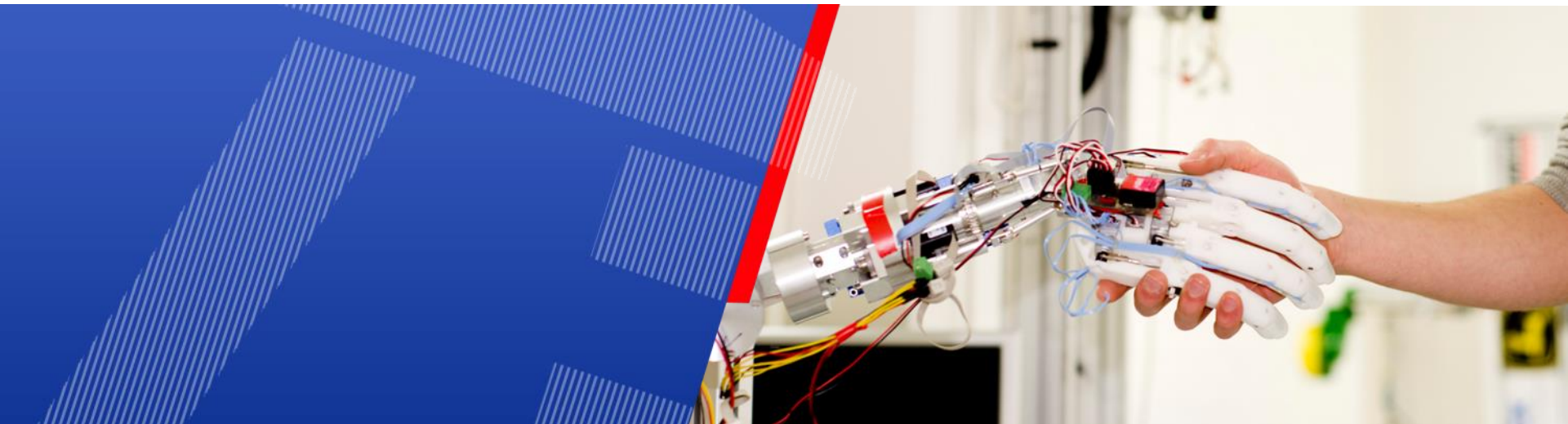


Robotermodellierung II



Prof. Karsten Berns

Robotics Research Lab

Department of Computer Science

University of Kaiserslautern, Germany

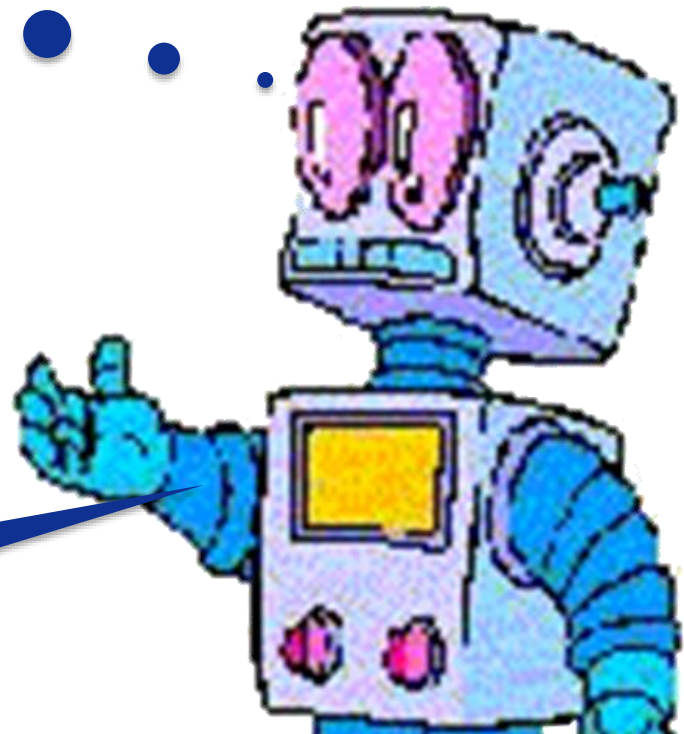
Inhalt

- Freiheitsgrade eines Robotersystems
- Geometrisches Modell
- Kinematisches Modell
- Direktes Kinematisches Problem
- Inverses Kinematisches Problem
- Dynamisches Modell

Inverses kinematisches Problem

Wie soll ich meine Hand
dorthin bewegen?

Bestimmt die Gelenkwinkel

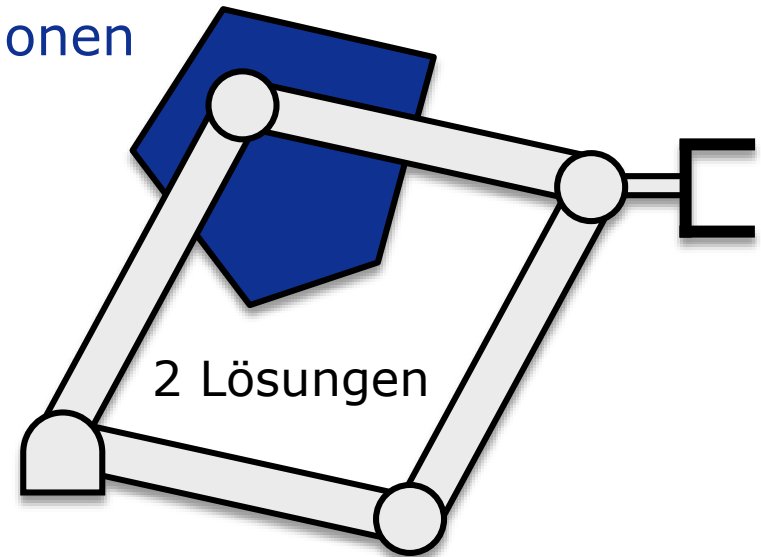


Inverses kinematisches Problem (IK)

- Aus DH-Parametern und Stellung des Greifers sollen die Gelenkwinkel bestimmt werden → Gleichung nach $\vec{\theta}$ auflösen
$$\text{BASIS}_{\text{TCP}}^A = \text{BASIS}_1^A(\theta_1) \cdot {}^1_2A(\theta_2) \cdots {}^{n-2}_{n-1}A(\theta_{n-1}) \cdot {}^{n-1}_nA(\theta_n)$$
- Ergibt 12 Gleichungen mit n Unbekannten
- Bei Puma 260: 12 Gleichungen, 6 Unbekannte

IK-Probleme

- Zulässige Konfigurationen: Nicht alle mathematischen Lösungen sind auch mechanisch realisierbar
 - Gelenkwinkelanschläge
 - Singuläre Konfigurationen
 - Ziel gehört nicht zum Arbeitsraum
 - ...
- Eindeutigkeit: Mehrere Konfigurationen (Gelenkwinkelkombinationen) führen zur selben Stellung des Endeffektors
- Wie kann eine geeignete Lösung gewählt werden?

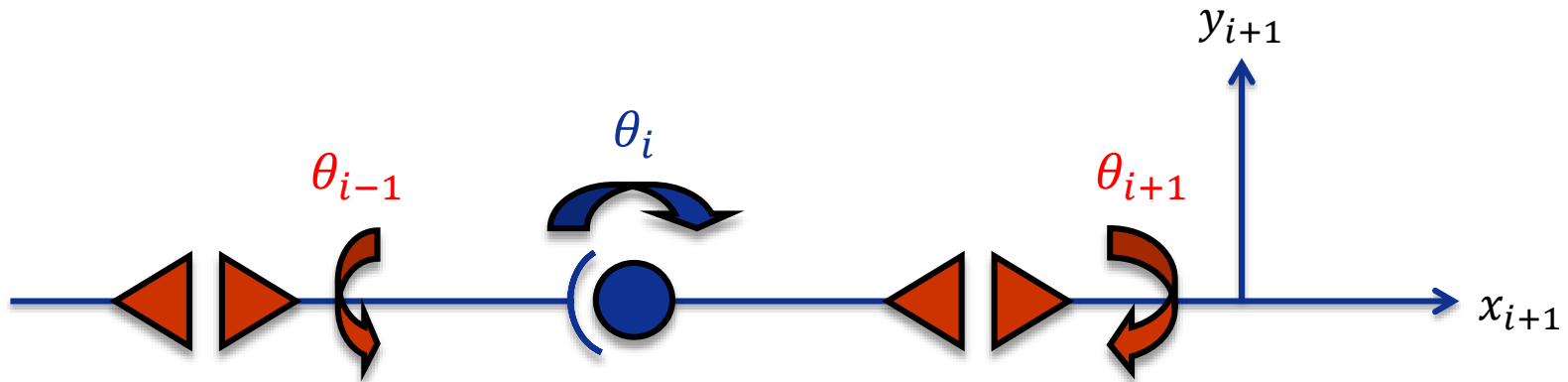


IK-Probleme: Singuläre Stellung

Angabe einer Nebenbedingung, z.B.

$$f(\theta_{i-1}, \theta_{i+1}) = a({}^{t-1}\theta_{i-1} - {}^t\theta_{i-1})^2 + b({}^{t-1}\theta_{i+1} - {}^t\theta_{i+1})^2 = \min$$

mit Gewichtungsfaktoren a und b



IK-Probleme

- Kein allgemein anwendbares Verfahren
- Geschwindigkeit:
Gelenkwinkelberechnung muss schnell erfolgen
- Methoden
 - Algebraische/Geometrische Methoden
(Lösungen in geschlossener Form)
 - Numerische Methoden

Beispiel: Planarer 2-Glied Roboterarm

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

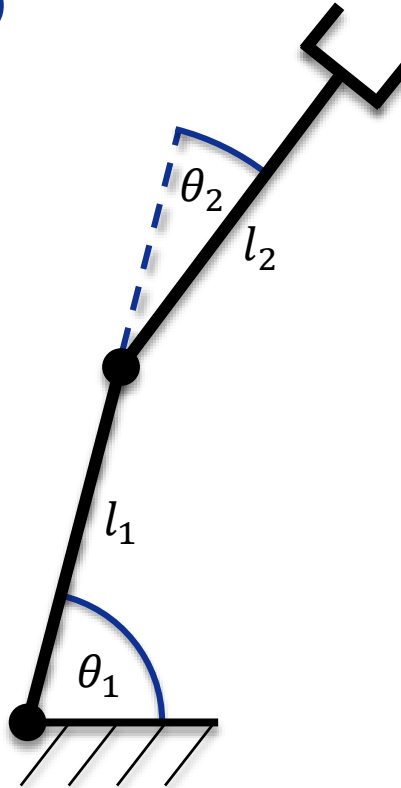
$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$\phi = \theta_1 + \theta_2$$

Abkürzungen

$$c_{12} = \cos(\theta_1 + \theta_2)$$

$$s_{12} = \sin(\theta_1 + \theta_2)$$



Algebraische Lösung

- Durch die Vorwärtskinematik erhält man

$${}^0_2T = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Position und Orientierung des Endeffektors gegeben durch

$${}^{BASIS}_{TCP}T = \begin{bmatrix} c\varphi & -s\varphi & 0 & x \\ s\varphi & c\varphi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Algebraische Lösung

- Koeffizientenvergleich

$$c\varphi = c_{12}$$

$$s\varphi = s_{12}$$

$$x = l_1 c_1 + l_2 c_{12}$$

$$y = l_1 s_1 + l_2 s_{12}$$

- Summe der Quadrate der letzten beiden Gleichungen

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2$$

- Daraus folgt $c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}$

Algebraische Lösung

- Wann existiert eine Lösung?
- Warum zwei Lösungen für θ_2 ?

Berechnung von θ_1

$$\cos(\theta_1 + \theta_2) = c_{12} = c_1 c_2 - s_1 s_2$$

$$\sin(\theta_1 + \theta_2) = s_{12} = c_1 s_2 + c_2 s_1$$

$$x = k_1 c_1 - k_2 s_1$$

$$y = k_1 s_1 + k_2 c_1$$

$$k_1 = l_1 + l_2 c_2$$

$$k_2 = l_2 s_2$$

$$T_{0,2} = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Setze $r = \sqrt{k_1^2 + k_2^2}$ und $\gamma = \text{ATAN2}(k_2, k_1)$

Algebraische Lösung

$$\begin{aligned}k_1 &= r \cdot \cos \gamma \\k_2 &= r \cdot \sin \gamma \\x/r &= \cos \gamma \cos \theta_1 - \sin \gamma \sin \theta_1 \\y/r &= \cos \gamma \sin \theta_1 + \sin \gamma \cos \theta_1\end{aligned}$$

oder

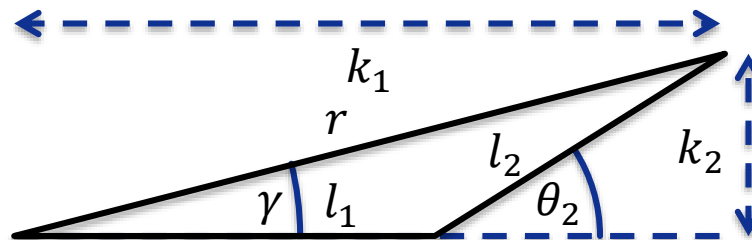
$$\begin{aligned}x/r &= \cos(\gamma + \theta_1) \\y/r &= \sin(\gamma + \theta_1)\end{aligned}$$

$$\begin{aligned}\gamma + \theta_1 &= \text{ATAN2}\left(\frac{y}{r}, \frac{x}{r}\right) = \text{ATAN2}(y, x) \\ \rightarrow \theta_1 &= \text{ATAN2}(y, x)\end{aligned}$$

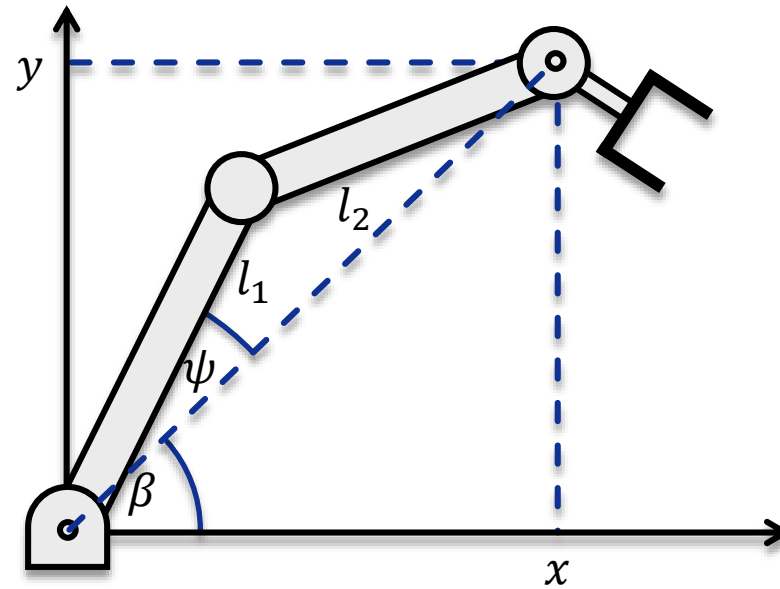
$$\begin{aligned}k_1 &= l_1 + l_2 c_2 \\k_2 &= l_2 s_2 \\r &= \sqrt{k_1^2 + k_2^2} \\ \gamma &= \text{ATAN2}(k_2, k_1) \\x &= k_1 c_1 - k_2 s_1 \\y &= k_1 s_1 + k_2 c_1\end{aligned}$$

Bedingung für Winkel:

$$\phi = \theta_1 + \theta_2$$



Geometrische Lösung

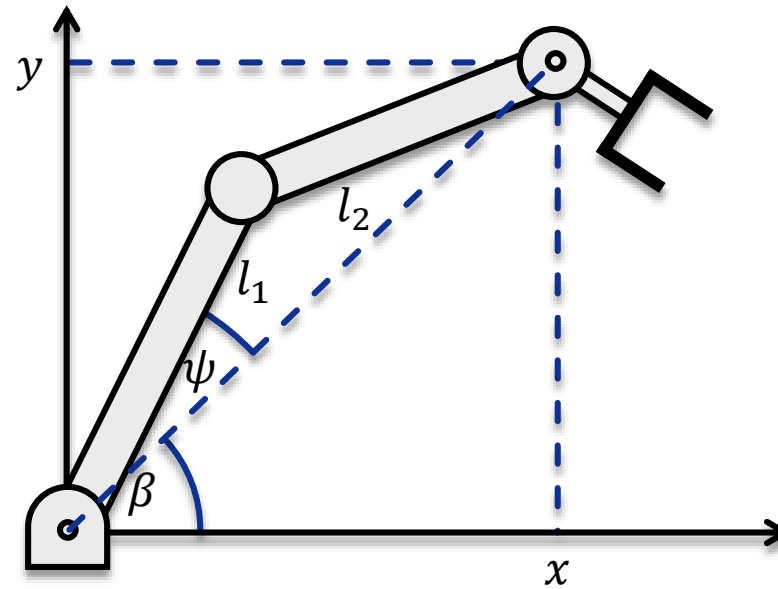


Kosinussatz:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos(180 - \theta_2)$$

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

Geometrische Lösung



$$l_2^2 = x^2 + y^2 + l_1^2 - 2l_1\sqrt{x^2 + y^2} \cos \psi$$

$$\cos \psi = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}}$$

$$\theta_1 = \beta \pm \psi \text{ für } 0 \leq \psi \leq 180$$

Das IK-Problem

- Bestimmung von T_{TCP} durch Multiplikation der homogenen Transformationsmatrizen

$$\text{BASIS}_{TCP}^T = \text{BASIS}_1^T A(\theta_1) \cdot {}_1^2A(\theta_2) \cdots {}_{n-1}^{n-2}A(\theta_{n-1}) \cdot {}_{n-1}^nA(\theta_n) \quad (1)$$

- T_{TCP} ist eine homogene 4×4 Matrix zur Beschreibung der gewünschten Position und Orientierung des Endeffektors

$$\text{BASIS}_{TCP}^T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Algorithmus zur Lösung des IK-Problems

- Gegeben: Transformationsmatrix (mit z.B. $n = 6$)
 - Gesucht: Gelenkwinkel θ_1 bis θ_n
-
1. Invert. ${}^0_1A(\theta_i)$ und multipliziere beide Seiten von (1) mit ${}^0_1A^{-1}$
 2. Finde im entstandenen Gleichungssystem eine Gleichung, die nur eine Unbekannte enthält und berechne diese
 3. Finde Gleichung im Gleichungssystem, die durch Substitution des im letzten Schritt gefundenen Gelenkwinkels lösbar ist
 4. Falls keine Gleichung gefunden, invertiere die Matrix ${}_{i+1}^iA(\theta_{i+1})$
 5. Wiederhole Schritte 1 – 4 bis alle Gelenkwinkel ermittelt

Geschlossene Lösungsverfahren

Durch Invertieren einzelner Transformationsmatrizen und Multiplizieren von rechts bzw. links erhält man weitere Matrixgleichungen, aus deren Elementen evtl. geschlossene Lösungen für einzelne Winkel gewonnen werden können.

$$BASIS_1 A^{-1} \cdot Basis_{TCP} T = {}^1_2 A \cdot {}^2_3 A \cdot {}^3_4 A \cdot {}^4_5 A \cdot {}^5_6 A$$

$${}^1_2 A^{-1} \cdot BASIS_1 A^{-1} \cdot Basis_{TCP} T = {}^2_3 A \cdot {}^3_4 A \cdot {}^4_5 A \cdot {}^5_6 A$$

$${}^2_3 A^{-1} \cdot {}^1_2 A^{-1} \cdot BASIS_1 A^{-1} \cdot Basis_{TCP} T = {}^3_4 A \cdot {}^4_5 A \cdot {}^5_6 A$$

$${}^3_4 A^{-1} \cdot {}^2_3 A^{-1} \cdot {}^1_2 A^{-1} \cdot BASIS_1 A^{-1} \cdot Basis_{TCP} T = {}^4_5 A \cdot {}^5_6 A$$

$${}^4_5 A^{-1} \cdot {}^3_4 A^{-1} \cdot {}^2_3 A^{-1} \cdot {}^1_2 A^{-1} \cdot BASIS_1 A^{-1} \cdot Basis_{TCP} T = {}^5_6 A$$

$$Basis_{TCP} T \cdot {}^5_6 A^{-1} = BASIS_1 A \cdot {}^1_2 A \cdot {}^2_3 A \cdot {}^3_4 A \cdot {}^4_5 A$$

$$Basis_{TCP} T \cdot {}^5_6 A^{-1} \cdot {}^4_5 A^{-1} = BASIS_1 A \cdot {}^1_2 A \cdot {}^2_3 A \cdot {}^3_4 A$$

$$Basis_{TCP} T \cdot {}^5_6 A^{-1} \cdot {}^4_5 A^{-1} \cdot {}^3_4 A^{-1} = BASIS_1 A \cdot {}^1_2 A \cdot {}^2_3 A$$

$$Basis_{TCP} T \cdot {}^5_6 A^{-1} \cdot {}^4_5 A^{-1} \cdot {}^3_4 A^{-1} \cdot {}^2_3 A^{-1} = BASIS_1 A \cdot {}^1_2 A$$

$$Basis_{TCP} T \cdot {}^5_6 A^{-1} \cdot {}^4_5 A^{-1} \cdot {}^3_4 A^{-1} \cdot {}^2_3 A^{-1} \cdot {}^1_2 A^{-1} = BASIS_1 A$$

Numerische Methoden

$$\vec{x}(t) = f(\vec{\theta}(t)) \Rightarrow \frac{d\vec{x}(t)}{dt} = \dot{\vec{x}}(t) = J(\vec{\theta})\dot{\vec{\theta}}(t)$$

$$J(\vec{\theta}) \in R^{n \times m} \quad J_{ij} = \frac{df_i}{d\theta_j} \quad 1 \leq i \leq m \quad 1 \leq j \leq n$$

- Kartesische Freiheitsgrade m
- Anzahl der Gelenke n
- Translations- und Winkelgeschwindigkeiten des TCP (z.B. differenzielle zeitliche Änderung der Euler-Winkel)

$$\dot{\vec{x}}(t) = (\dot{p}_x, \dot{p}_y, \dot{p}_z, \dot{\alpha}, \dot{\beta}, \dot{\gamma})^T$$

- Differenzenquotient statt Differentialquotient

$$\Delta\vec{\theta} = J(\vec{\theta})^{-1} \Delta\vec{x}$$

Numerische Methoden

- Bestimme Änderung des Beschreibungsvektors $\Delta \vec{x}$
- Berechne nötige Änderung der Gelenkwinkel $\Delta \vec{\theta}$ mittels inverser Jacobi-Matrix
- Näherungslösung, da bei Berechnung der zu den $\Delta \vec{x}$ -Änderungen gehörenden $\Delta \vec{\theta}$ -Änderungen die Jacobi-Matrix als konstant angenommen wird
- Nach Berechnung von $\Delta \vec{\theta}$ wird die Jacobi-Matrix aktualisiert
- Entstehende Fehler iterativ verkleinern

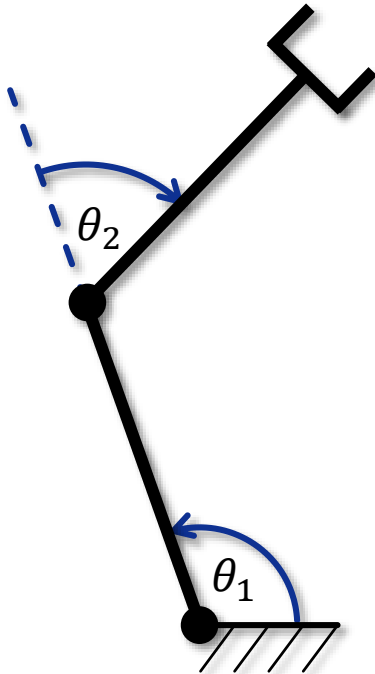
Numerische Methoden

- $n = m$: Nicht redundanter Manipulator
 - Jacobi Matrix quadratisch und kann invertiert werden
- $n > m$: Unterbestimmtes System
 - Redundanter Manipulator
 - Inverse Jacobi-Matrix existiert nicht
 - Generalisierte Inverse der Jacobi-Matrix „Pseudoinverse“
- $n < m$: Überbestimmtes System
 - Oft keine Lösung oder Unterraum
 - Inverse der Jacobi-Matrix existiert nicht
 - Generalisierte Inverse der Jacobi-Matrix „Pseudoinverse“

Numerische Methoden

- Verfahren prinzipiell für alle Robotertypen mit beliebigem Freiheitsgrad einsetzbar
- Weitere Probleme
 - Anfälligkeit für Singularitäten
 - Hohe Laufzeit
 - Es wird irgendeine Lösung gefunden

Beispiel: Zweigelenk-Roboterarm



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

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = J(\vec{\theta}) \dot{\vec{\theta}}$$

$$= J(\vec{\theta}) \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

$$= \begin{bmatrix} -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

Beispiel: Zweigelenk-Roboterarm

- Die Jacobi-Matrix muss invertiert werden

$$\begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{pmatrix} = \frac{1}{\underbrace{l_1 l_2 \sin \theta_2}_{J(\vec{\theta})^{-1}}} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_{12} - l_1 c_1 & -l_1 s_{12} - l_1 s_1 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

- Für $\theta_2 = 0, \pm 180$ ist $J(\theta)$ singulär!
- Alle singulären Konfigurationen auf dem Arbeitsraumrand
- Abkürzungen
 - $c_{12} = \cos(\theta_1 + \theta_2)$
 - $s_{12} = \sin(\theta_1 + \theta_2)$
 - $c_i = \cos \theta_i$
 - $s_i = \sin \theta_i$

Numerische Methoden: Optimierung

Überbestimmtes System ($n < m$)

- Zu wenig Freiheitsgrade, um Pose zu erreichen
- Approximative Lösung vermöge $\|J\Delta\vec{\theta} - \Delta\vec{x}\|^2 \rightarrow \min$
- Mittels Pseudo-Inversen $J^+ := (J^T J)^{-1} J^T$ erhält man $\Delta\vec{\theta} = J^+ \Delta\vec{x}$
- Mit Differenzvektor $\Delta\vec{x} = \overrightarrow{x_{\text{ziel}}} - \overrightarrow{x_{\text{aktuell}}} = (\Delta p_x, \Delta p_y, \Delta p_z, \Delta\alpha, \Delta\beta, \Delta\gamma)^T$

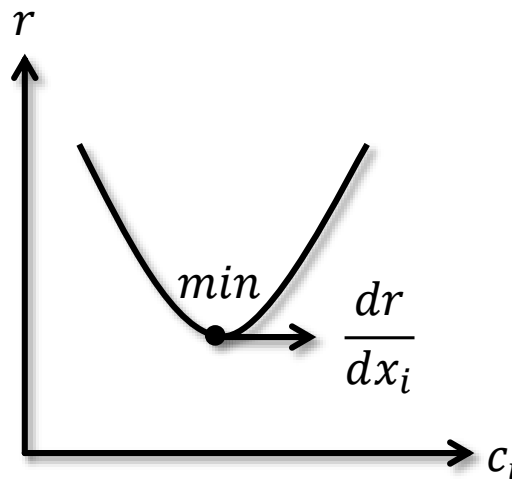
Optimierung

Minimierung der Fehlerquadrate (Gauß)

- Überbestimmtes System $J \cdot \Delta \vec{x} = \Delta \vec{\theta}$ mit $i = 1, \dots, m > n$

$$\sum_{k=1}^m j_{ik} \Delta x_k = \theta_i$$

- Minimiere $r = \sum_{i=1}^m (\sum_{k=1}^n j_{ik} \Delta x_k - \Delta \theta_i)^2 \rightarrow \min$
- (Notwendige) Bedingung ist $\nabla r = 0$, d.h. $\frac{dr}{dx_i} \stackrel{!}{=} 0$ mit $i = 1, \dots, n$



Optimierung

- Minimierung der Fehlerquadrate

$$\begin{aligned}
 \frac{dr}{dx_i} &= \sum_{l=1}^m \frac{d}{dx_i} \left(\sum_{k=1}^n j_{lk} \Delta x_k - \Delta \theta_l \right)^2 \quad (\text{Kettenregel}) \\
 &= \sum_{l=1}^m 2 \left(\sum_{k=1}^n j_{lk} \Delta x_k - \Delta \theta_l \right) \frac{d}{dx_i} \sum_{k=1}^n j_{lk} \Delta x_k \\
 &= 2 \sum_{l=1}^m \left(\sum_{k=1}^n j_{lk} \Delta x_k - \Delta \theta_l \right) j_{li} \\
 &= 2 \sum_{l=1}^m \sum_{k=1}^n j_{li} j_{lk} \Delta x_k - 2 \sum_{l=1}^m j_{li} \Delta \theta_l \stackrel{!}{=} 0
 \end{aligned}$$

- Das System lautet nun $J^T \cdot J \cdot \Delta \vec{x} = J^T \cdot \Delta \vec{\theta}$
oder $\vec{x} = (J^T \cdot J)^{-1} \cdot J^T \cdot \vec{\theta}$

Numerische Methoden: Optimierung

Unterbestimmtes System ($n > m$)

- Meist liegen zu viele Freiheitsgrade vor
- Zusätzliche Bedingungen, z.B. „natürlichste“ Gelenkstellung $\Delta\theta'_i$
- Optimierungsproblem unter Nebenbedingung $J\Delta\vec{\theta} = \Delta\vec{x}$:

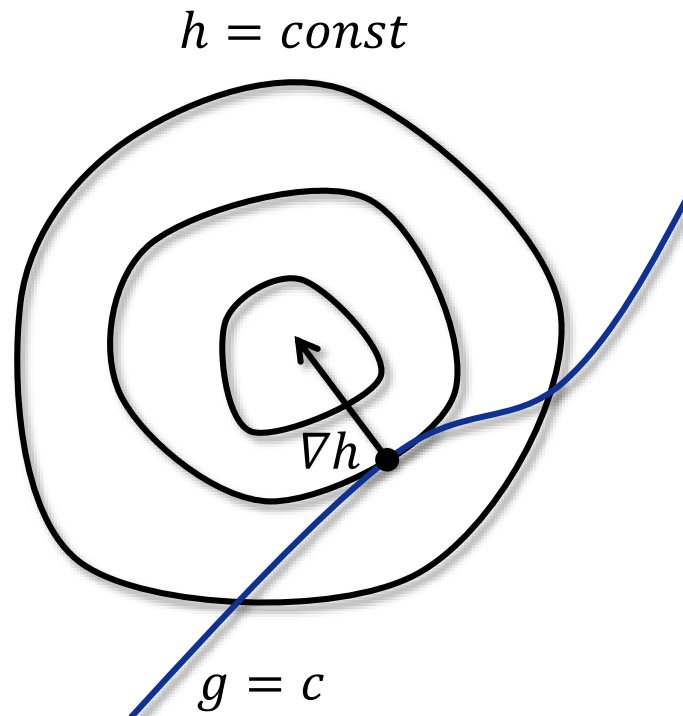
$$h := \sum_{i=1}^n w_i (\Delta\theta_i - \Delta\theta'_i)^2 \rightarrow \min$$

- Lösung mittels Lagrange-Multiplikatoren

$$r := h + \lambda^T (J\Delta\vec{\theta} - \Delta\vec{x}) \rightarrow \min$$

Optimierung: Lagrange-Multiplikatoren

- Idee: Finde Extremwert einer Funktion $h(x_1, \dots, x_n)$ unter Nebenbedingung $g(x_1, \dots, x_n) = c$
- Lösung: $g \wedge \nabla h = \lambda h \nabla g$



Optimierung: Lagrange-Multiplikatoren

Gegeben:

- m „harte“ Bedingungen (Endeffektor, vgl. $g = c$)
der Form $J \cdot \Delta \vec{x} = \Delta \vec{\theta}$

$$m \begin{array}{|c|} \hline J \\ \hline \end{array} \begin{array}{c} \left| \right. \\ n \end{array} \Delta x = \begin{array}{c} \left| \right. \\ m \end{array} \Delta \theta$$

- l „weiche“ Bedingungen („natürliche“ Haltung, $h \rightarrow \min$) $A \cdot \Delta \vec{x} = \vec{q}$

$$l \begin{array}{|c|} \hline A \\ \hline \end{array} \begin{array}{c} \left| \right. \\ n \end{array} \Delta x = \begin{array}{c} \left| \right. \\ l \end{array} q$$

- Mit Unbekannten $x_i, (i = 1, \dots, n), n > m,$

Optimierung: Lagrange-Multiplikatoren

Vorgehensweise:

- Minimiere

$$r = \sum_{k=1}^l \left(\sum_{r=1}^n a_{kr} \Delta x_r - q_k \right)^2 + \sum_{k=1}^m \lambda_k \left(\sum_{r=1}^n j_{kr} \Delta x_r - \Delta \theta_k \right) \rightarrow \min$$

- r hängt von $x_i, (i = 1, \dots, n)$ und $\Delta_k, (k = 1, \dots, m)$ ab

$$\frac{dr}{dx_i} = 2 \sum_{k=1}^l \left(\sum_{r=1}^n a_{kr} \Delta x_r - q_k \right) a_{ki} + \sum_{k=1}^m \lambda_k j_{ki} \stackrel{!}{=} 0$$

(Weiche Bedingungen, $\nabla h = \lambda \cdot \nabla g$)

$$\frac{dr}{d\lambda_i} = \sum_{r=1}^n j_{ir} \Delta x_r - \Delta \theta_i \stackrel{!}{=} 0$$

(Harte Bedingungen, $g = c$)

Optimierung: Lagrange-Multiplikatoren

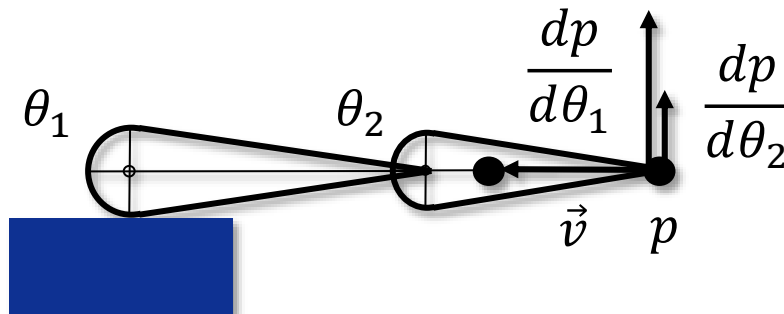
- Matrixform:
$$\begin{pmatrix} 2A^T A & J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} \Delta \vec{x} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} 2A^T \vec{q} \\ \Delta \vec{\theta} \end{pmatrix}$$
- Bemerkungen
 - Lösung besteht nur aus den Werten x_i
 - Lagrange-Multiplikatoren λ_k dienen zum Auffinden der Lösung

Optimierung: Algorithmus

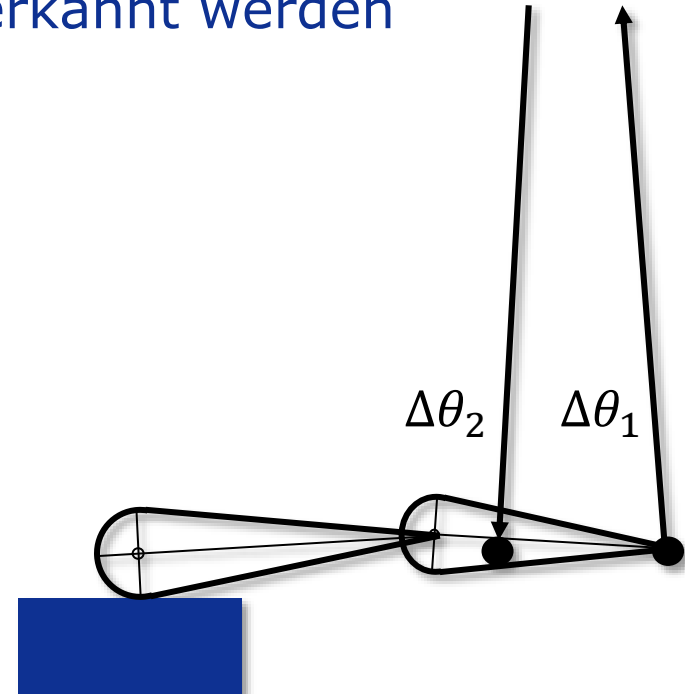
- (1) Startwerte für Θ und Translation \vec{v}
- (2) Bestimme $J(\Theta)$
- (3) Eliminiere ggf. redundante Parameter (Singularität)
- (4) Löse nach $\Delta\Theta$ (Optimierung ggf. mit Nebenbedingung)
- (5) Aktualisiere Θ und \vec{v}
- (6) Falls Lösung noch nicht erreicht, gehe zu (2)

Optimierung: Probleme

- Singuläre Situationen müssen erkannt werden
- Beschränkung von $\Delta\theta$



Blockierter Manipulator



Umgebung der Singularität

Direkte und Inverse Kinematik

- Direkte Kinematik: $f: R^n \rightarrow R^m, \vec{x} = f(\vec{\theta})$
- Inverse Kinematik: $f^{-1}: R^m \rightarrow R^n, \vec{\theta} = f^{-1}(\vec{x})$
- Es existiert ...
 - ... eine eindeutige Lösung
 - ... eine endliche Menge von Lösungen
 - ... eine unendliche Menge von Lösungen
 - ... keine Lösung

IK-Probleme

	Allgemeine Verfahren	Spezielle Verfahren
Vorgehen	Iteration, allgemeine Lösungsverfahren für Gleichungssysteme	Auf trigonometrischen Beziehungen basierende graphische Verfahren
Vorteile	Allgemein	Schnell
Nachteile	Hoher Aufwand, lange Zeitdauer	Nur für spezielle Robotertypen

Nächste Vorlesung

Robotermodellierung

- Geschwindigkeit starrer Körper
- Berechnung statischer Kräfte/Momente