

Robotik I: Einführung in die Robotik

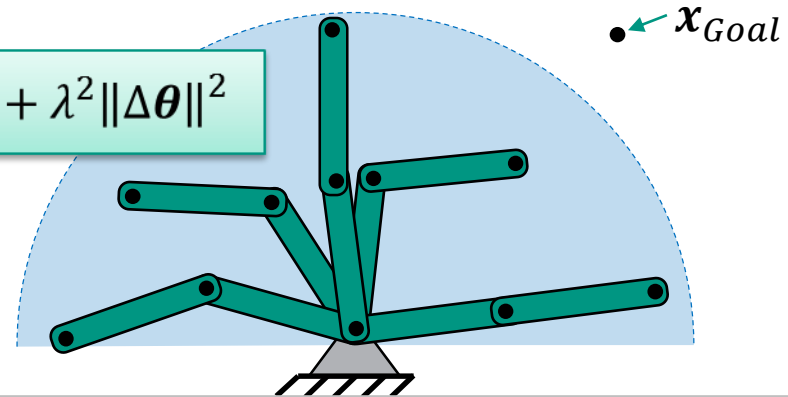
Kapitel 3 – Inverse Kinematik

Tamim Asfour

<http://www.humanoids.kit.edu>



$$\min_{\Delta\theta} \|J_f(\theta)\Delta\theta - \Delta x\|^2 + \lambda^2 \|\Delta\theta\|^2$$



Inhalt

- **Inverses Kinematisches Problem**
- Geschlossene Methoden
 - Geometrische Methoden
 - Algebraische Methoden
- Numerische Methoden
 - Gradientenabstieg
 - Jacobi Matrix und Pseudoinverse
- Zusammenfassung

Vorwärtskinematik

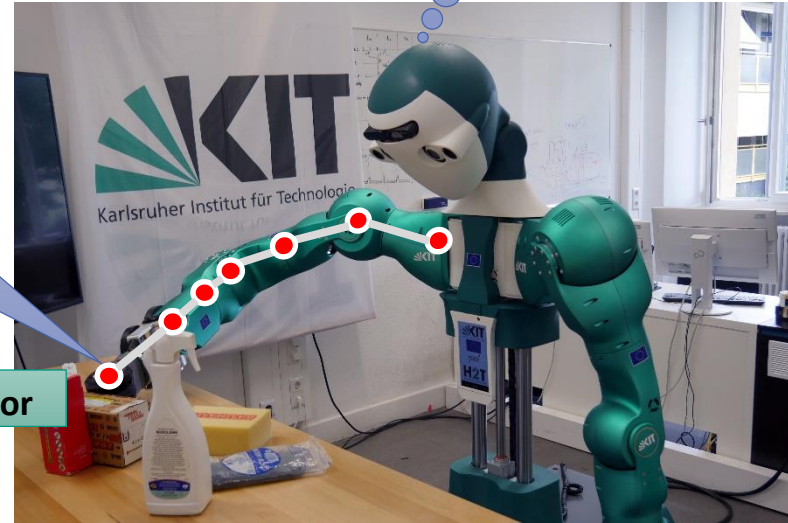
Direktes kinematisches Problem

- Eingabe: Gelenkwinkelstellungen des Roboters
- Ausgabe: **Pose des Endeffektors**

**Direkte Kinematik:
HIER!**

Endeffektor

**Wo ist meine
Hand?**



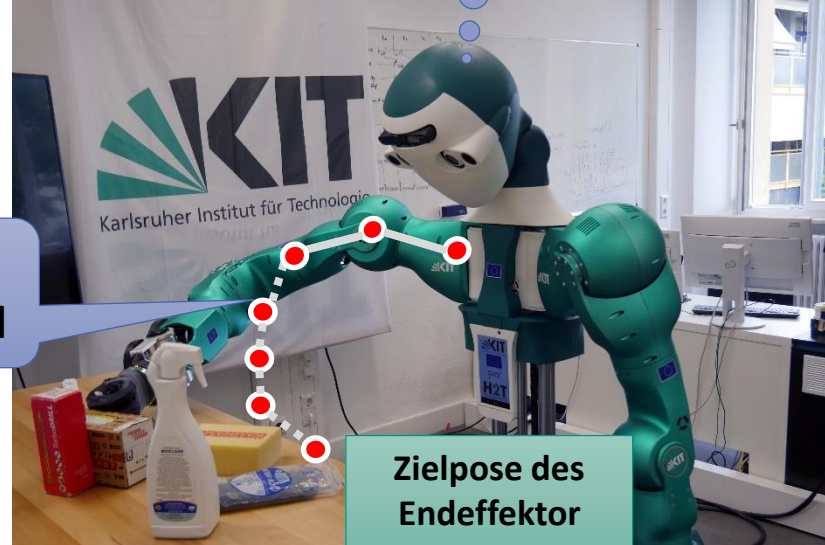
Inverse Kinematik

Inverses kinematisches Problem:

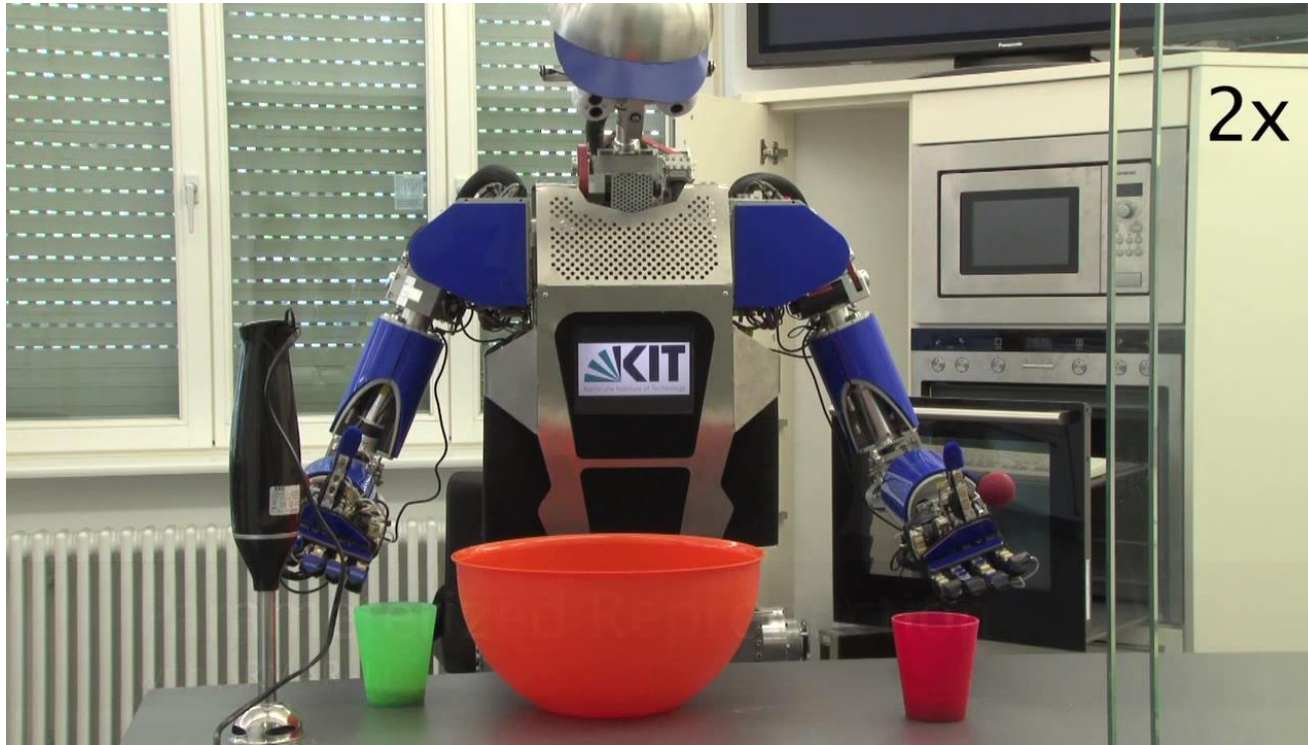
- Eingabe: Zielpose des Endeffektors
- Ausgabe: **Gelenkwinkelstellungen**

Wie bewege ich meine
Hand zum Ziel?

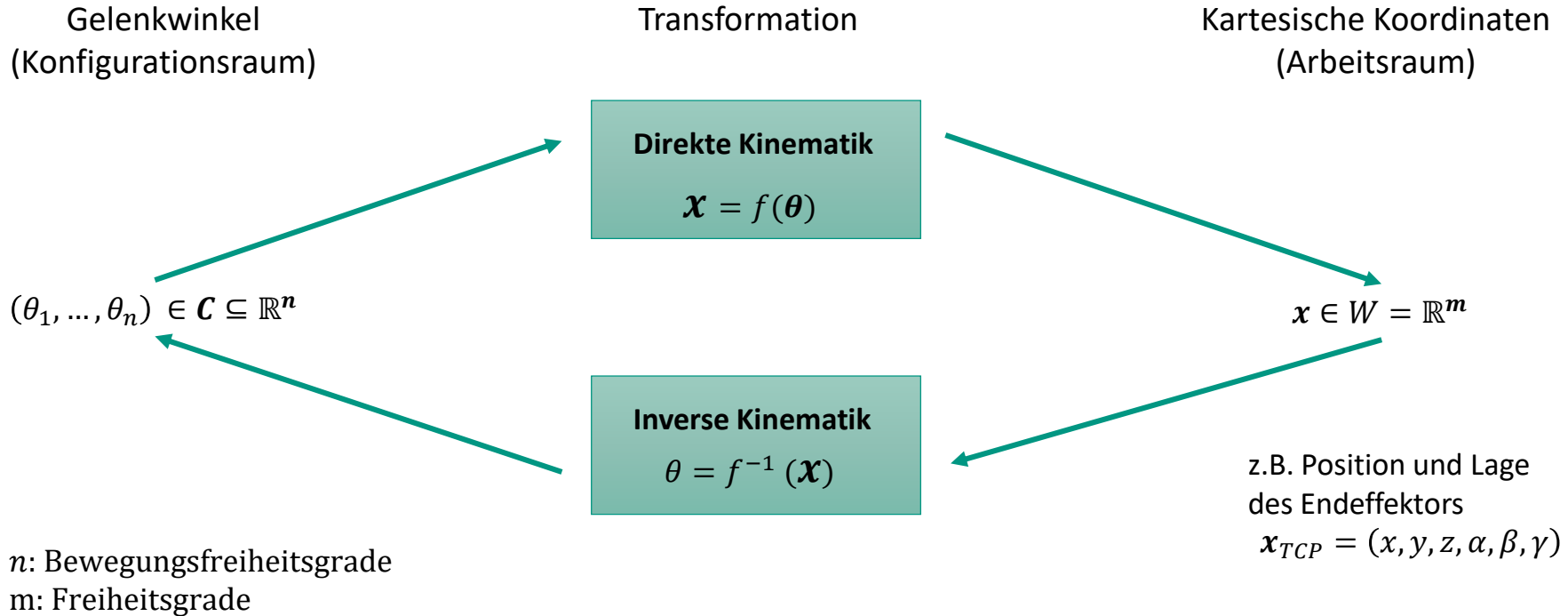
Inverse Kinematik:
Bestimmt die Gelenkwinkel



Inverse Kinematik



Übersicht: Direkte und Inverse Kinematik



Inverse Kinematik: Problemstellung

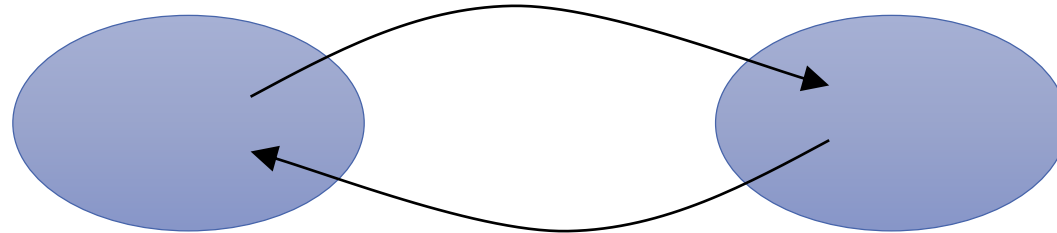
Gelenkwinkelraum
(Configuration Space)

$$(\theta_1, \dots, \theta_n) \in \mathcal{C} \subseteq \mathbb{R}^n$$

Kartesischer Raum
(Workspace)

$$W = \mathbb{R}^m$$

Direkte Kinematik



Inverse Kinematik

n : Bewegungsfreiheitsgrade

m : Freiheitsgrade

Inverse Kinematik: Bijektion

■ Direkte Kinematik: $x = f(\theta), \quad x \in W, \theta \in C$

■ Inverse Kinematik: $\theta = f^{-1}(x)$

■ Inverse Funktion f^{-1} existiert nur wenn f **bijektiv** ist (injektiv und surjektiv)

■ Funktion $f: C \rightarrow W$ ist **injektiv**, wenn zu jedem Wert in W höchstens ein Element aus C (gar keins, genau eins, aber nicht mehrere) existiert

$$f(\theta_1) = f(\theta_2) \Rightarrow \theta_1 = \theta_2$$

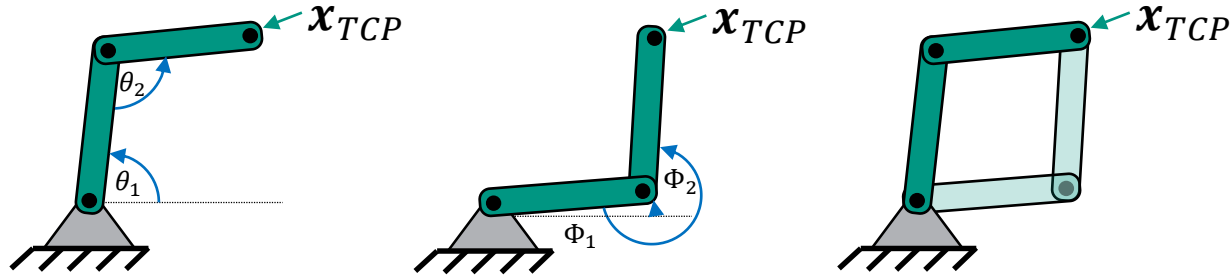
■ Funktion $f: C \rightarrow W$ ist **surjektiv**, wenn für alle Wert in W mindestens ein Element aus C existiert

$$\forall x \in W: \exists \theta \in C: f(\theta) = x$$

■ Im Allgemeinen ist die Vorwärtskinematik f **nicht bijektiv**

Inverse Kinematik: Injektion

Vorwärtskinematik ist i. A. nicht injektiv ($f(x_1) = f(x_2) \Rightarrow x_1 = x_2$)



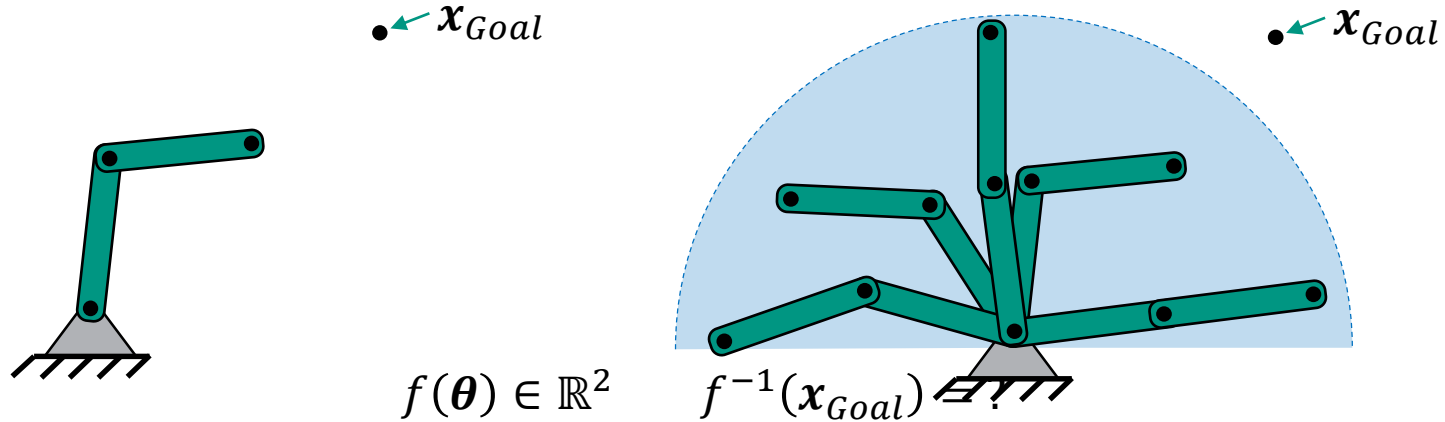
$$\theta = (\theta_1, \theta_2) \in \mathcal{C}, \quad \Phi = (\Phi_1, \Phi_2) \in \mathcal{C}$$

$$f(\theta) = f(\Phi) = x_{TCP}$$

$$f^{-1}(x_{TCP}) = ?, \quad \theta \text{ oder } \Phi?$$

Inverse Kinematik: Surjektion

Vorwärtskinematik ist i. A. nicht surjektiv ($\forall x \in W: \exists \theta \in C: f(\theta) = x$)



Es existiert kein $\theta \in C$, für das $f(\theta) = x_{Goal}$.

Teilweise behebbar durch Definition des Arbeitsraums $W \subset \mathbb{R}^2$.

Inverse Kinematik: Beispiel eines 2-DoF Roboters (1)

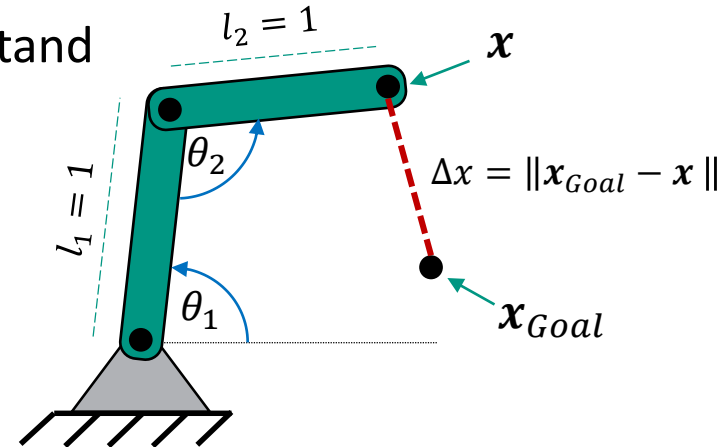
- Position des Endeffektors (Vorwärtskinematik)

$$\mathbf{x} = f(\boldsymbol{\theta}) = \begin{pmatrix} \cos \theta_1 + \cos(\theta_1 + \theta_2) \\ \sin \theta_2 + \sin(\theta_1 + \theta_2) \end{pmatrix}$$

- Bei gegebener Zielposition $\mathbf{x}_{Goal}$ beträgt der Abstand von der aktuellen Position zum Ziel:

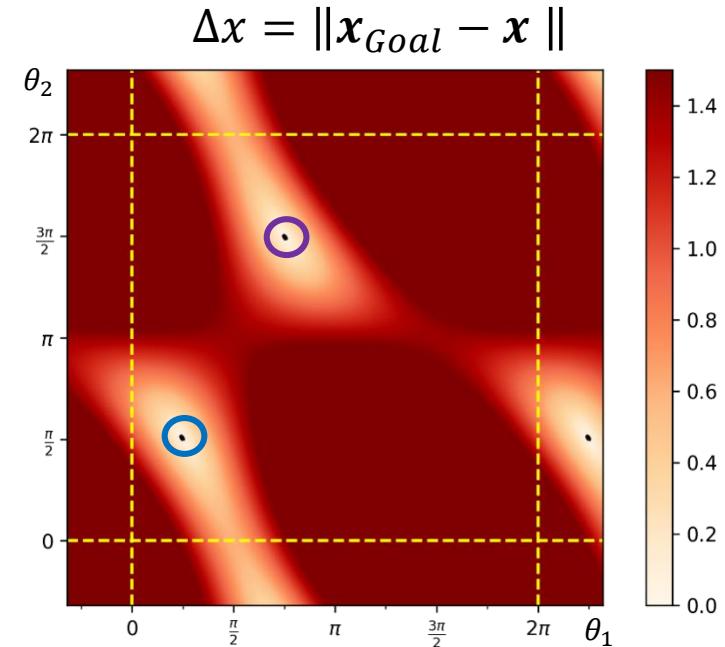
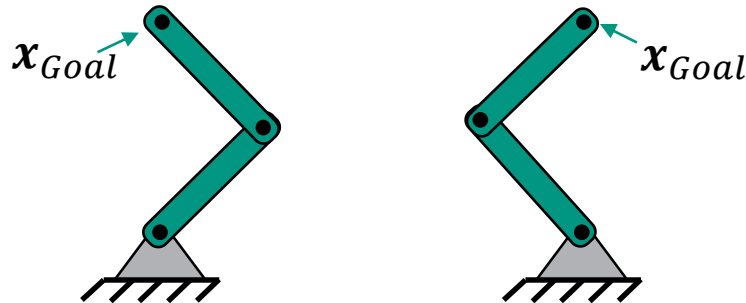
$$\Delta x = \|\mathbf{x}_{Goal} - \mathbf{x}\|$$

- Inverse Kinematik: Finde $\boldsymbol{\theta}$, für das $\Delta x = 0$.



Inverse Kinematik: Beispiel eines 2-DoF Roboters (2)

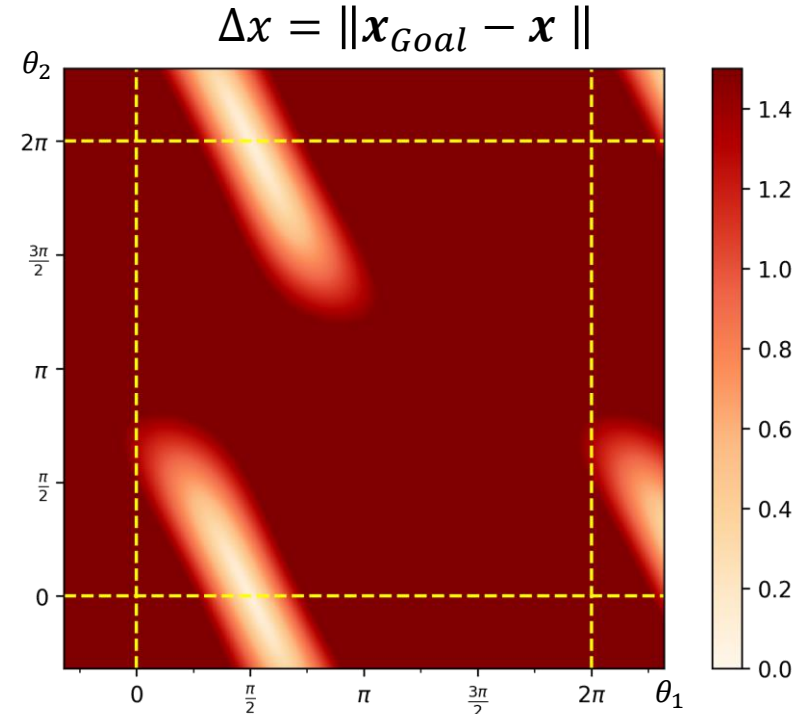
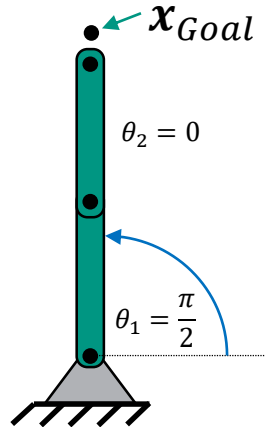
- Wie verändert sich der Abstand Δx
 - für unterschiedliche Gelenkwinkel $\theta = (\theta_1, \theta_2)$?
 - für unterschiedliche Zielpositionen $\mathbf{x}_{Goal}$?
- Konkret: $\mathbf{x}_{Goal} = (0, \sqrt{2})^T$
- Lösungen: $\theta_1 = \left(\frac{\pi}{4}, \frac{\pi}{2}\right)$ oder $\theta_2 = \left(\frac{3\pi}{4}, \frac{3\pi}{2}\right)$



Inverse Kinematik: Beispiel eines 2-DoF Roboters (4)

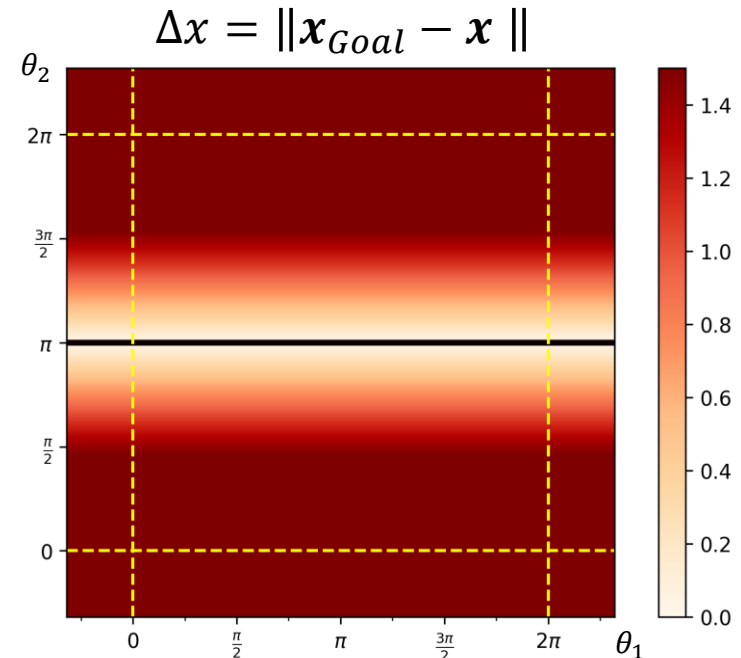
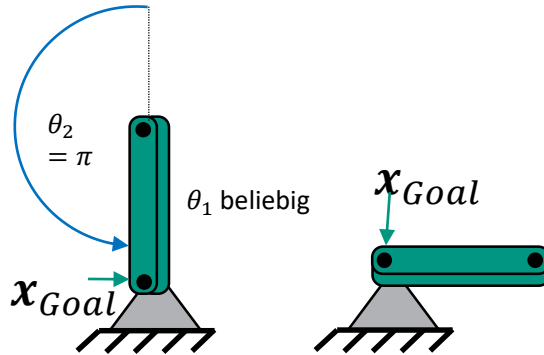
Was passiert bei $\mathbf{x}_{Goal} = (0, 2.1)^T$ außerhalb des Arbeitsraums?

Keine Lösungen



Inverse Kinematik: Beispiel eines 2-DoF Roboters (5)

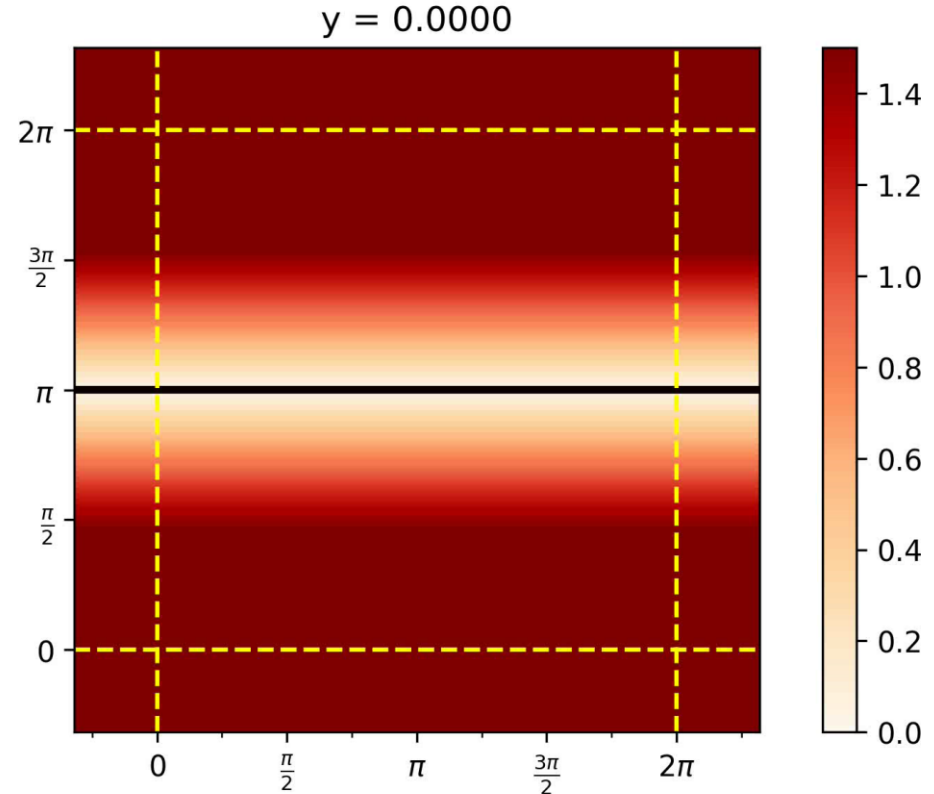
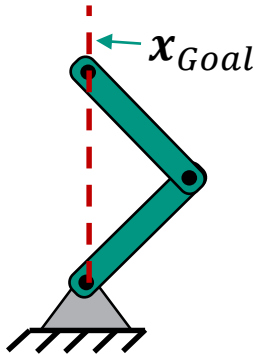
- Wie verändert sich der Abstand Δx für $\mathbf{x}_{Goal} = (0, 0)^T$?
- Es gibt unendlich viele Lösungen: $\boldsymbol{\theta} = (\theta_1, \pi)$
 - $\theta_2 = \pi$: Der zweite Arm ist eingeklappt auf den ersten Arm
 - θ_1 kann beliebig gewählt werden



Inverse Kinematik: Beispiel eines 2-DoF Roboters (6)

$$\mathbf{x}_{Goal} = (0, y)^T$$

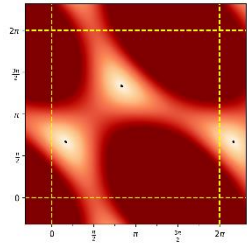
$$\Delta x = \|\mathbf{x}_{Goal} - \mathbf{x}\|$$



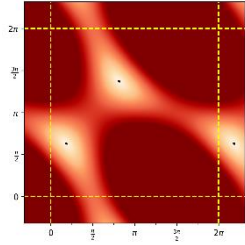
Inverse Kinematik: Beispiel eines 2-DoF Roboters (3)

- Wie verändert sich Δx für unterschiedliche Zielpositionen $\mathbf{x}_{Goal} = (0, y_{goal})^T$

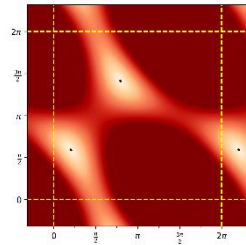
$y_{goal} = 1.0$



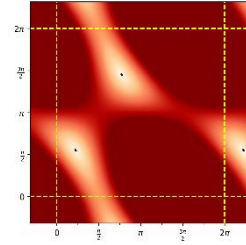
$y_{goal} = 1.1$



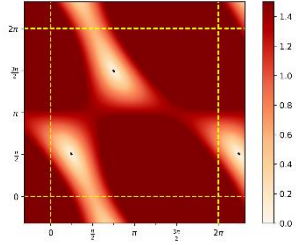
$y_{goal} = 1.2$



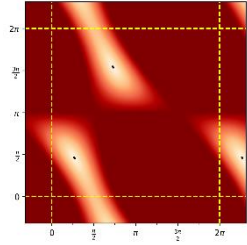
$y_{goal} = 1.3$



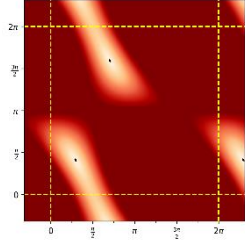
$y_{goal} = 1.4$



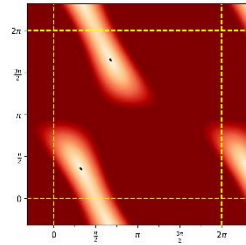
$y_{goal} = 1.5$



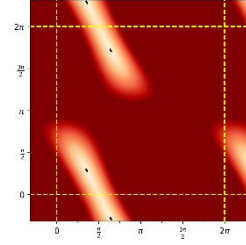
$y_{goal} = 1.6$



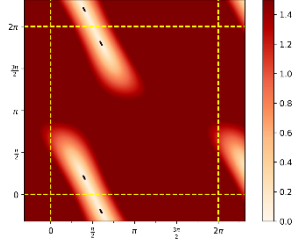
$y_{goal} = 1.7$



$y_{goal} = 1.8$



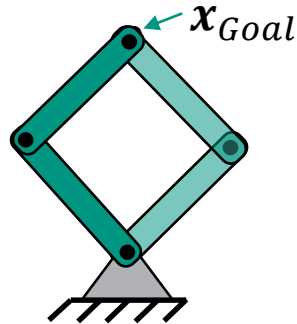
$y_{goal} = 1.9$



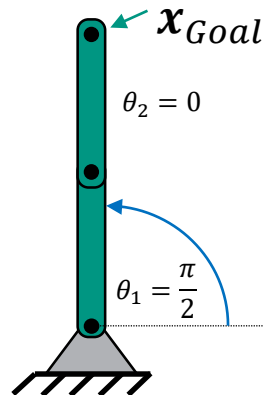
Inverse Kinematik: Beispiel eines 2-DoF Roboters (6)

Im Fall eines 2-DoF planaren Roboters gibt es vier verschiedene Fälle:

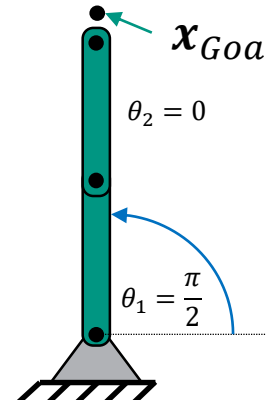
- Es gibt **zwei unabhängige Lösungen** (Normalfall).
- Es gibt **genau eine Lösung** (Rand des Arbeitsraums).
- Es gibt **keine Lösung** (Außerhalb des Arbeitsraums).
- Es gibt **unendlich viele Lösungen** (Zielpunkt in der Basis).



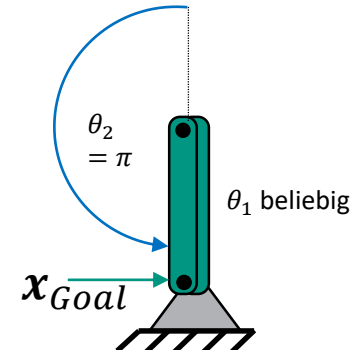
Zwei Lösungen



Eine Lösung



Keine Lösung



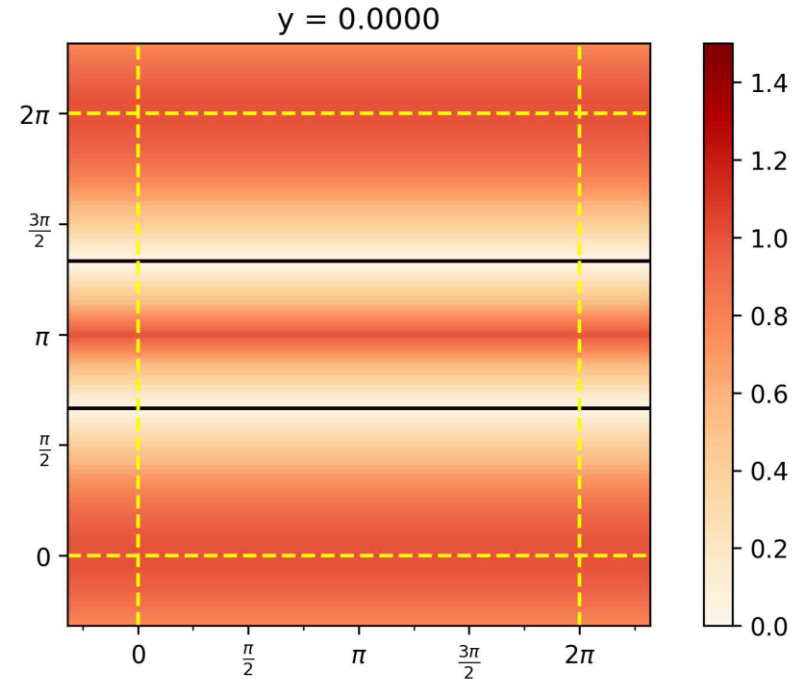
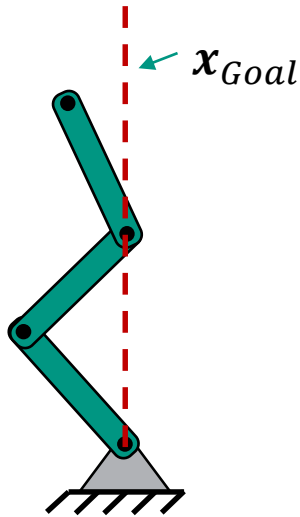
Unendlich viele Lösungen

Inverse Kinematik: Beispiel eines 3-DoF Roboters

3-DoF Roboter: Wie sieht der Lösungsraum aus?

$$\mathbf{x}_{Goal} = (0, y)^T$$

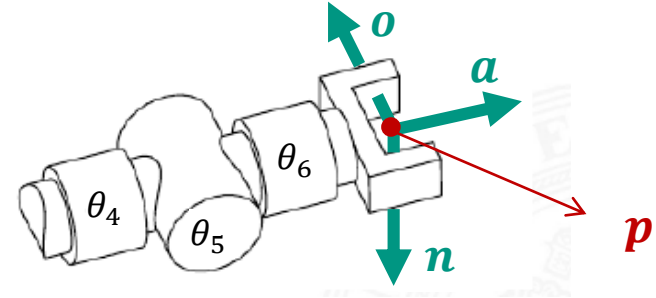
$$\Delta x = \|\mathbf{x}_{Goal} - \mathbf{x}\|$$



Inverse Kinematik: Vorgehensweise

■ Lage des TCP

$$T_{TCP} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



■ Kinematisches Modell:

$$T_{TCP} = {}^{BKS}T_{TCP}(\boldsymbol{\theta}) = A_{0,1}(\theta_1) \cdot A_{1,2}(\theta_2) \cdot \dots \cdot A_{n-2,n-1}(\theta_{n-1}) \cdot A_{n-1,n}(\theta_n)$$

■ Gegeben: T_{TCP} , Gesucht: $\boldsymbol{\theta}$

■ Ansatz: Gleichung nach $\boldsymbol{\theta}$ auflösen (**nichtlineares Problem**)

Inhalt

- Inverses Kinematisches Problem
- **Geschlossene Methoden**
 - Geometrische Methoden
 - Algebraische Methoden
- Numerische Methoden
- Zusammenfassung

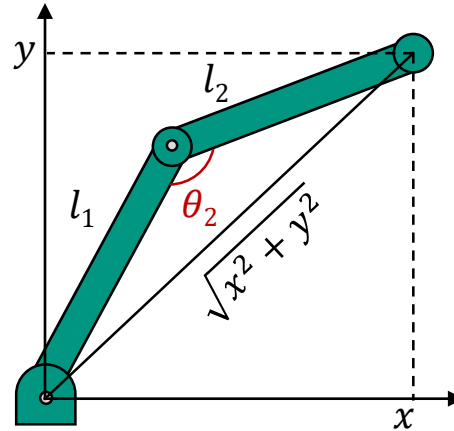
Geometrische Methode: Vorgehen

- Nutze **geometrische Beziehungen**, um die Gelenkwinkel θ aus der T_{TCP} zu bestimmen.
- Das kinematische Modell wird dabei nicht direkt verwendet.

Anwendung von:

- Trigonometrischen Funktionen
- Sinus- / Kosinussätzen

Geometrische Methode: Beispiel

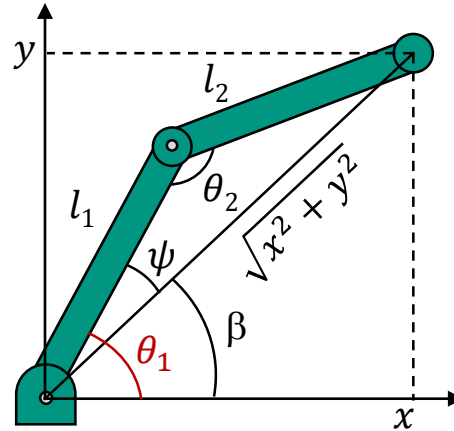


■ Mit Kosinussatz:

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

$$\cos(\theta_2) = - \underbrace{\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}}_u \rightarrow \theta_2 = \arccos(u)$$

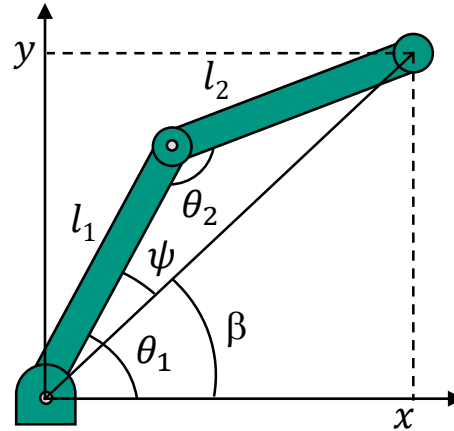
Geometrische Methode: Beispiel (2)



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

$$\rightarrow \cos(\psi) = \underbrace{\frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1 \sqrt{x^2 + y^2}}}_w \rightarrow \psi = \text{acos}(w)$$

Geometrische Methode: Beispiel (3)



$$\tan(\beta) = \frac{y}{x} \rightarrow \beta = \text{atan}\left(\frac{y}{x}\right)$$

$$\theta_1 = \psi + \beta$$

Geometrische Methode: Polynomialisierung

Transzendente Gleichungen sind in der Regel schwer zu lösen, da die Variable θ gewöhnlich in der Form $\cos \theta$ bzw. $\sin \theta$ auftritt.

Werkzeug: **Substitution (Weierstraß-Substitution)**

$$u = \tan\left(\frac{\theta}{2}\right)$$

unter Verwendung von:

$$\cos \theta = \frac{1-u^2}{1+u^2} \qquad \sin \theta = \frac{2u}{1+u^2}$$

➔ **Auflösen von Polynomgleichungen**

Inverses Kinematisches Problem

Geschlossene Methoden

Geometrische Methoden

Algebraische Methoden

Numerische Methoden

Zusammenfassung

Algebraische Methoden

- Gleichsetzen von TCP-Pose T_{TCP} und Transformation ${}^{BKS}T_{TCP}$ aus dem kinematischen Modell:

$$T_{TCP} = {}^{BKS}T_{TCP}(\theta)$$

- **Koeffizientenvergleich** der beiden Matrizen

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \dots & b_{nn} \end{pmatrix} \Rightarrow a_{ij} = b_{ij} \quad \forall i, j \in [1:n]$$

- 16 Gleichungen bei homogenen Matrizen in 3D (4 trivial: $0 = 0, 1 = 1$)
➔ **12 nicht-triviale Gleichungen**

Algebraische Methoden: Beispiel (1)

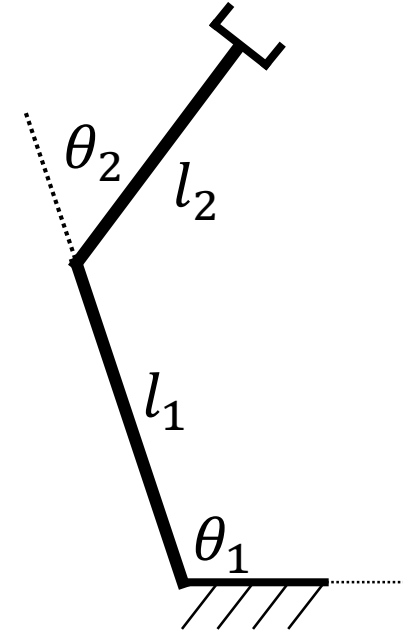
■ Aus kinematischem Modell

$${}^{BKS}T_{TCP} = \begin{pmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$c_{12} = \cos(\theta_1 + \theta_2); \quad s_{12} = \sin(\theta_1 + \theta_2)$$

■ Gewünschte Lage des Endeffektors im Raum: Position (x, y) , Orientierung (ϕ)

$$P_{TCP} = \begin{pmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Algebraische Methoden: Beispiel (2)

Koeffizientenvergleich

$$\begin{pmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

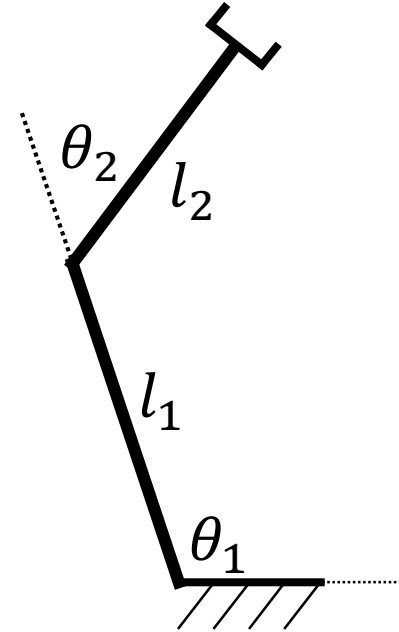
$$c_\phi = c_{12} \quad (1)$$

$$s_\phi = s_{12} \quad (2)$$

$$x = l_1 c_1 + l_2 c_{12} \quad (3)$$

$$y = l_1 s_1 + l_2 s_{12} \quad (4)$$

→ Nach θ auflösen



Algebraische Methoden: Beispiel (3)

Summe der Quadrate von (3) und (4)

$$x^2 = l_1^2 c_1^2 + 2l_1 c_1 l_2 c_{12} + l_2^2 c_{12}^2$$

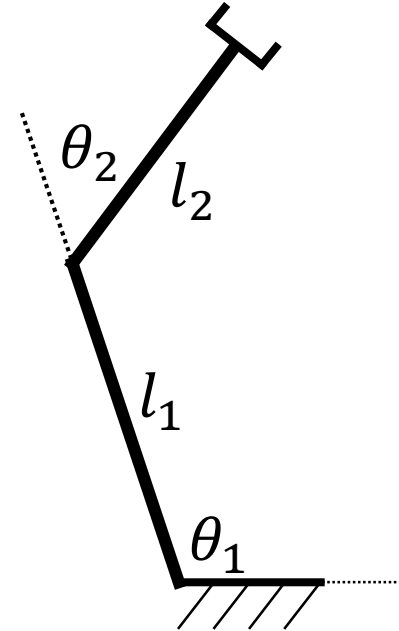
$$y^2 = l_1^2 s_1^2 + 2l_1 s_1 l_2 s_{12} + l_2^2 s_{12}^2$$

$$s_1^2 + c_1^2 = 1; \quad s_{12}^2 + c_{12}^2 = 1$$

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 (c_1 c_{12} + s_1 s_{12}) = l_1^2 + l_2^2 + 2l_1 l_2 c_2$$

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \rightarrow \theta_2$$

Zwei Lösungen für θ_2 sind möglich. Warum?
→ Redundanz



Algebraische Methoden: Beispiel (4)

Berechnung von θ_1

- Koeffizientenvergleich:

$$x = l_1 c_1 + l_2 c_{12}, \quad y = l_1 s_1 + l_2 s_{12}$$

- Additionstheorem: $\cos(\theta_1 + \theta_2) = \cos(\theta_1) \cos(\theta_2) - \sin(\theta_1) \sin(\theta_2)$

$$x = l_1 c_1 + l_2 (c_1 c_2 - s_1 s_2)$$

$$y = l_1 s_1 + l_2 (s_1 c_2 + s_2 c_1)$$

- Vereinfachen:

$$x = (l_1 + l_2 c_2) c_1 - (l_2 s_2) s_1$$

$$y = (l_1 + l_2 c_2) s_1 + (l_2 s_2) c_1$$

Auflösung schwierig.

Hilfe durch Schablonen für typische Gleichungen oder symbolische Mathematik in Matlab, Maple, Mathematica.

Algebraische Methoden: Lösungsalgorithmus

■ Problem:

Oft können nicht alle Gelenkwinkel aus den 12 Gleichungen bestimmt werden.

■ Ansatz:

Kenntnis der Transformationen erhöht Chance, die Gleichungen zu lösen.

■ Gegeben:

Die Transformationsmatrizen $A_{0,1} \cdot A_{1,2} \cdot \dots \cdot A_{n-1,n}$ und T_{TCP}

■ Gesucht:

Die Gelenkwinkel θ_1 bis θ_n

$$T_{TCP} = A_{0,1}(\theta_1) \cdot A_{1,2}(\theta_2) \cdot A_{2,3}(\theta_3) \cdot A_{3,4}(\theta_4) \cdot A_{4,5}(\theta_5) \cdot A_{5,6}(\theta_6)$$

Algebraische Methoden: Vorgehensweise

■ Ausgangspunkt: die Matrix-Gleichung

$$T_{TCP} = A_{0,1}(\theta_1) \cdot A_{1,2}(\theta_2) \cdot A_{2,3}(\theta_3) \cdot A_{3,4}(\theta_4) \cdot A_{4,5}(\theta_5) \cdot A_{5,6}(\theta_6)$$

■ Vorgehensweise:

1. Invertiere $A_{0,1}(\theta_1)$ und multipliziere beide Seiten der Gleichung mit $A_{0,1}^{-1}$
2. Versuche aus dem neu entstehenden Gleichungssystem eine Gleichung zu finden, die nur eine Unbekannte enthält und löse diese Gleichung nach der Unbekannten.
3. Versuche eine Gleichung im Gleichungssystem zu finden, die durch die Substitution der im letzten Schritt gefundenen Lösung nach eine Unbekannten lösbar ist.
4. Falls keine Lösungen mehr gefunden werden können, so muss eine weitere Matrix ($A_{1,2}(\theta_2)$) invertiert werden.
5. Wiederhole die Schritte 1 - 4 bis alle Gelenkwinkel ermittelt sind.

Algebraische Methoden: Gleichungen

$$T_{TCP} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3} \cdot A_{3,4} \cdot A_{4,5} \cdot A_{5,6}$$

$$A_{0,1}^{-1} \cdot T_{TCP} = A_{1,2} \cdot A_{2,3} \cdot A_{3,4} \cdot A_{4,5} \cdot A_{5,6}$$

$$A_{1,2}^{-1} \cdot A_{0,1}^{-1} \cdot T_{TCP} = A_{2,3} \cdot A_{3,4} \cdot A_{4,5} \cdot A_{5,6}$$

$$A_{2,3}^{-1} \cdot A_{1,2}^{-1} \cdot A_{0,1}^{-1} \cdot T_{TCP} = A_{3,4} \cdot A_{4,5} \cdot A_{5,6}$$

$$A_{3,4}^{-1} \cdot A_{2,3}^{-1} \cdot A_{1,2}^{-1} \cdot A_{0,1}^{-1} \cdot T_{TCP} = A_{4,5} \cdot A_{5,6}$$

$$A_{4,5}^{-1} \cdot A_{3,4}^{-1} \cdot A_{2,3}^{-1} \cdot A_{1,2}^{-1} \cdot A_{0,1}^{-1} \cdot T_{TCP} = A_{5,6}$$

$$T_{TCP} \cdot A_{5,6}^{-1} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3} \cdot A_{3,4} \cdot A_{4,5}$$

$$T_{TCP} \cdot A_{5,6}^{-1} \cdot A_{4,5}^{-1} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3} \cdot A_{3,4}$$

$$T_{TCP} \cdot A_{5,6}^{-1} \cdot A_{4,5}^{-1} \cdot A_{3,4}^{-1} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3}$$

$$T_{TCP} \cdot A_{5,6}^{-1} \cdot A_{4,5}^{-1} \cdot A_{3,4}^{-1} \cdot A_{2,3}^{-1} = A_{0,1} \cdot A_{1,2}$$

$$T_{TCP} \cdot A_{5,6}^{-1} \cdot A_{4,5}^{-1} \cdot A_{3,4}^{-1} \cdot A_{2,3}^{-1} \cdot A_{1,2}^{-1} = A_{0,1}$$

**12 nicht-triviale
Gleichungen aus
jeder Matrix-
Gleichung**

Inhalt

- Inverses Kinematisches Problem
- Geschlossene Methoden
 - Geometrische Methoden
 - Algebraische Methoden
- **Numerische Methoden**
- Zusammenfassung

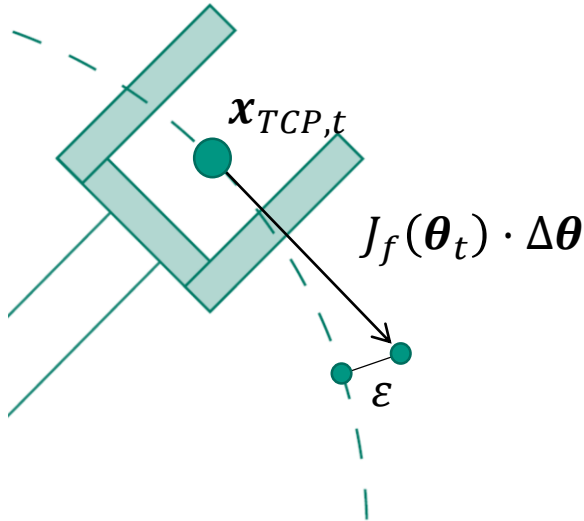
Numerische Methoden: Jacobi-Matrix (Wdh.)

- Gegeben eine differenzierbare Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- Die **Jacobi-Matrix** enthält sämtliche partiellen Ableitungen erster Ordnung von f . Für ein $\mathbf{a} \in \mathbb{R}^n$ gilt:

$$J_f(\mathbf{a}) = \left(\frac{\partial f_i}{\partial x_j}(\mathbf{a}) \right)_{i,j} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{a}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{a}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{a}) & \cdots & \frac{\partial f_m}{\partial x_n}(\mathbf{a}) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

- Es gilt:

$$\dot{\mathbf{x}}(t) = \frac{df(\theta(t))}{dt} = J_f(\theta(t)) \cdot \dot{\theta}(t)$$



TCP-Pose über **Vorwärtskinematik**:

$$\mathbf{x}_{TCP,t} = f(\boldsymbol{\theta}_t)$$

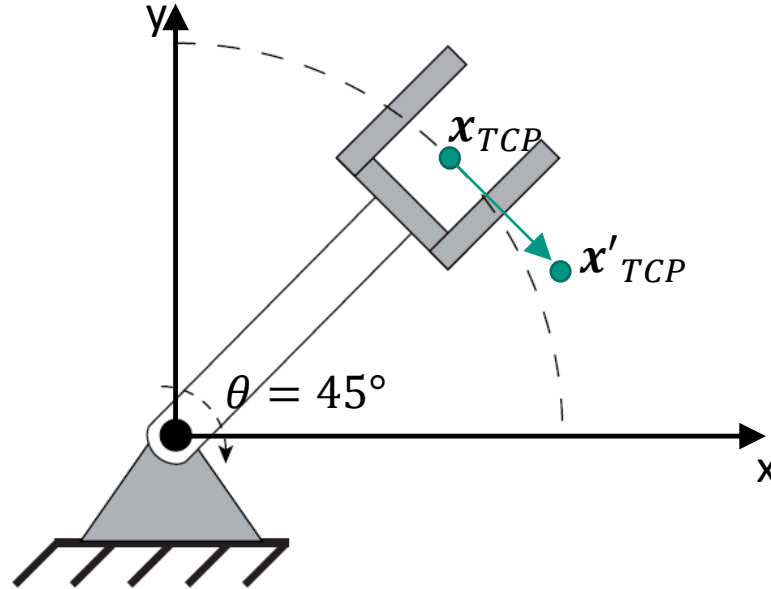
Jacobi-Matrix liefert **Bewegungstangenten** in der aktuellen Stellung $\boldsymbol{\theta}_t$:

$$J_f(\boldsymbol{\theta}_t) = \frac{\partial f(\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}$$

Annahme: Modell gültig für kleine $\Delta\theta$

Lineare Approximation der Bewegung

Approximationsfehler ε existiert



$$J_f(\theta) = J_f(45^\circ) = s \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix}, s \in \mathbb{R}$$

Inverses Kinematisches Problem

Geschlossene Methoden

Geometrische Methoden

Algebraische Methoden

Numerische Methoden

Gradientenabstieg

Pseudoinverse

Zusammenfassung

Gradientenabstieg: Optimierungsproblem

Vorwärtskinematik:

$$\boldsymbol{x} = f(\boldsymbol{\theta}), \quad \boldsymbol{x} \in W \subset \mathbb{R}^m, \quad \boldsymbol{\theta} \in C \subset \mathbb{R}^n$$

Fehlerfunktion für Zielpose $\boldsymbol{x}_{Goal} \in W$:

$$e(\boldsymbol{\theta}) = \|\boldsymbol{x}_{Goal} - f(\boldsymbol{\theta})\|^2$$

Lösungen für die Inverse Kinematik bei: $e(\boldsymbol{\theta}) = 0$

Ansatz: **Gradientenabstieg**

Gradientenabstieg: Ableitung der Fehlerfunktion

■ Fehlerfunktion für Zielpose $\mathbf{x}_{Goal} \in W$: $e(\boldsymbol{\theta}) = \|\mathbf{x}_{Goal} - f(\boldsymbol{\theta})\|^2$

■ Ableitung mit Kettenregel:

$$F(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|^2$$

$$\nabla F(\mathbf{x}) = 2 \mathbf{A}^T (\mathbf{Ax} - \mathbf{b})$$

$$\frac{\partial e}{\partial \boldsymbol{\theta}} = \frac{\partial (\|\mathbf{x}_{Goal} - f(\boldsymbol{\theta})\|^2)}{\partial (\mathbf{x}_{Goal} - f(\boldsymbol{\theta}))} \cdot \frac{\partial (\mathbf{x}_{Goal} - f(\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}}$$

Hinweis: $\frac{\partial e}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{1 \times m}$
ist ein Zeilenvektor

$$\frac{\partial e}{\partial \boldsymbol{\theta}} = -2 \cdot (\mathbf{x}_{Goal} - f(\boldsymbol{\theta}))^T \cdot J(\boldsymbol{\theta})$$

$$\left(\frac{\partial e}{\partial \boldsymbol{\theta}} \right)^T = 2 \cdot J^T(\boldsymbol{\theta}) \cdot (f(\boldsymbol{\theta}) - \mathbf{x}_{Goal})$$

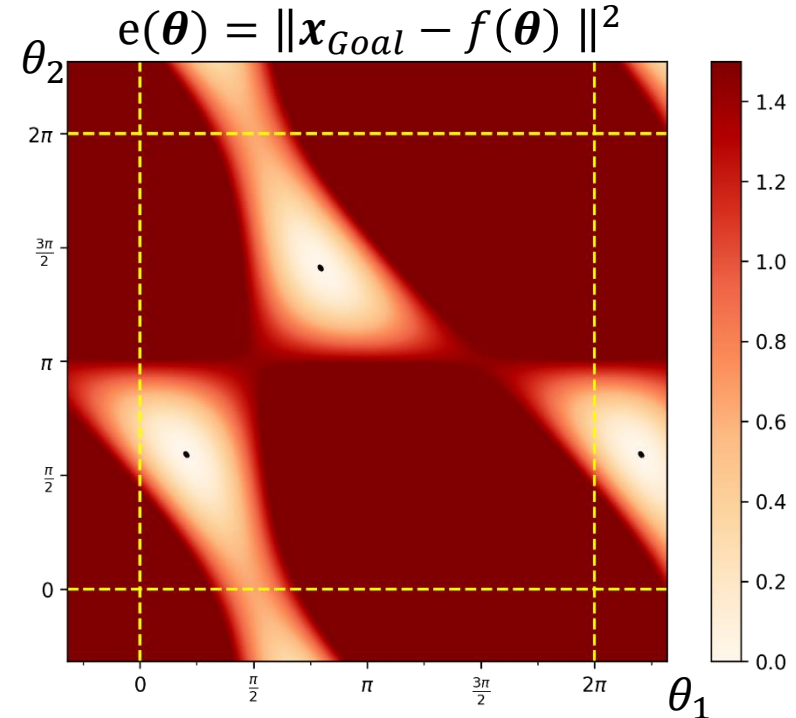
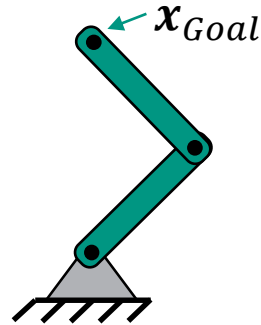
$$(\mathbf{x}^T \cdot \mathbf{A})^T = \mathbf{A}^T \cdot \mathbf{x}$$

Gradientenabstieg: Algorithmus

- Fehlerfunktion für Zielpose $\mathbf{x}_{Goal} \in W$: $e(\boldsymbol{\theta}) = \|\mathbf{x}_{Goal} - f(\boldsymbol{\theta})\|^2$
- Gradient: $grad(e) = \frac{\partial e}{\partial \boldsymbol{\theta}} = 2 \cdot J^T(\boldsymbol{\theta}) \cdot (f(\boldsymbol{\theta}) - \mathbf{x}_{Goal})$
- Startkonfiguration wählen: $\boldsymbol{\theta}_0 \in C, i = 0$
- Schrittlänge $\gamma \in \mathbb{R}$
- Solange $e(\boldsymbol{\theta}_i) > e_{Threshold}$: # Grenzwert
 - $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \gamma \cdot 2 \cdot J^T(\boldsymbol{\theta}_i) \cdot (f(\boldsymbol{\theta}_i) - \mathbf{x}_{Goal})$ # – Gradient
 - $i = i + 1$

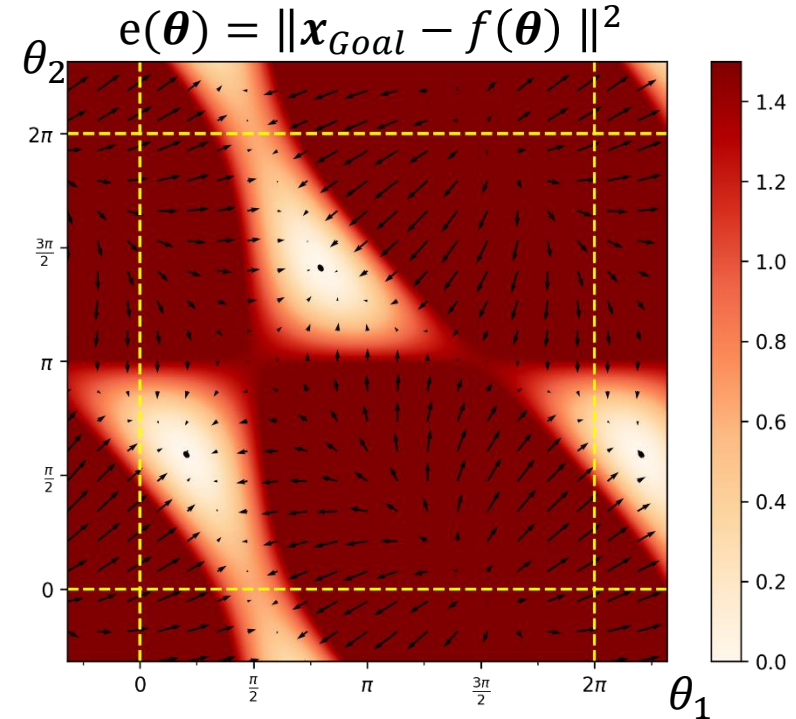
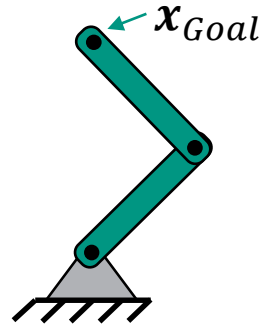
Gradientenabstieg: Beispiel (1)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$



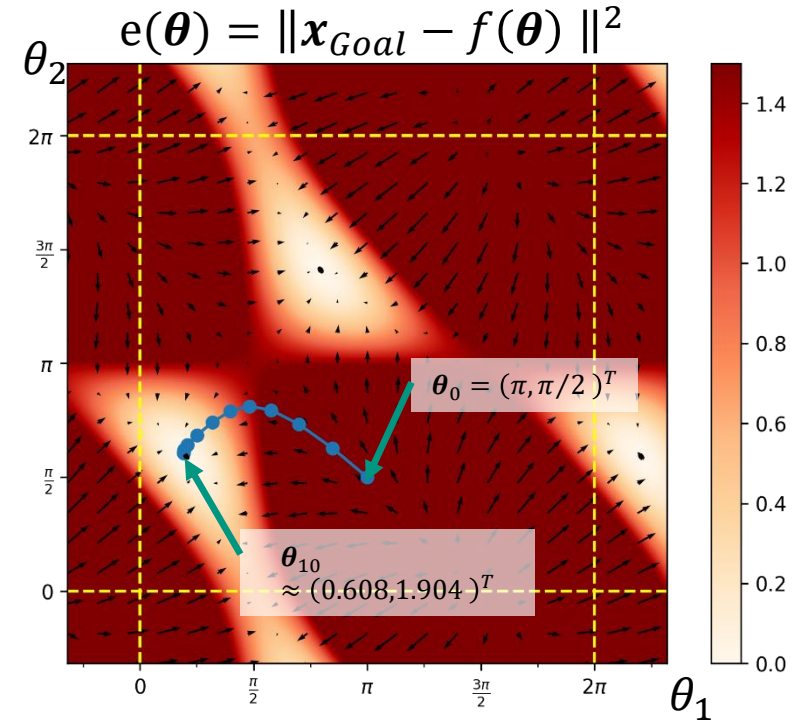
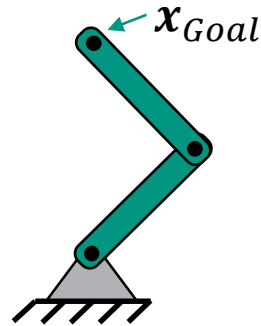
Gradientenabstieg: Beispiel (2)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$
- Gradientenfeld



Gradientenabstieg: Beispiel (3)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2)^T \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$
- Gradientenfeld
- Schrittlänge: $\gamma = 0.2$
- Start: $\theta_0 = \left(\pi, \frac{\pi}{2}\right)^T$



Gradientenabstieg: Beispiel (4)

2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2)^T \in \mathcal{C}$

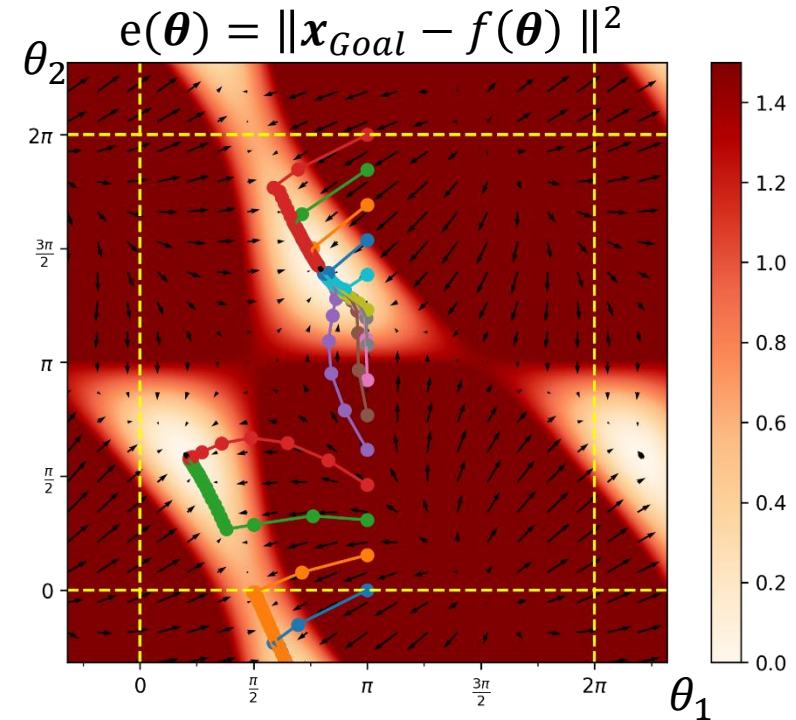
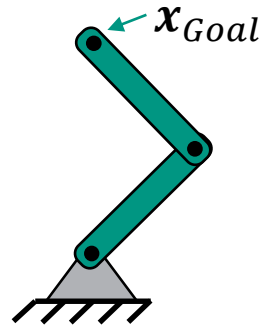
Zielpose $x_{Goal} = (0, 1.2)^T$

Gradientenfeld

Schrittlänge: $\gamma = 0.2$

Verschiedene Startpunkte:

$\theta_0 = (\pi, \theta_{2,start})^T$



Inhalt

- Inverses Kinematisches Problem
- Geschlossene Methoden
 - Geometrische Methoden
 - Algebraische Methoden
- **Numerische Methoden**
 - Gradientenabstieg
 - **Pseudoinverse**
- Zusammenfassung

Numerische Methoden: Differenzenquotient

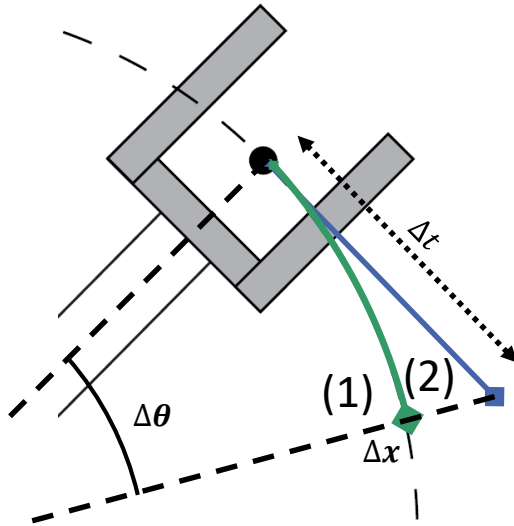
1) Tatsächliche Bewegung gemäß:

$$\dot{\mathbf{x}}(t) = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t)$$

2) Angenäherte Bewegung im Intervall Δt mittels Differenzenquotient:

$$\Delta \mathbf{x} \approx \mathbf{J}(\boldsymbol{\theta})\Delta \boldsymbol{\theta}$$

Approximation der Änderung durch Übergang vom Differentialquotienten zum **Differenzenquotienten**



Linearisierung des Problems

Numerische Methoden: Umkehrung

- Bisher erreicht: **Lokale, lineare Annäherung an die Vorwärtskinematik**

$$\Delta \mathbf{x} = \mathbf{f}(\boldsymbol{\theta} + \Delta \boldsymbol{\theta}) - \mathbf{f}(\boldsymbol{\theta}) \approx \mathbf{J}_f(\boldsymbol{\theta}) \cdot \Delta \boldsymbol{\theta}$$

- **Gesucht:** Lösung für das inverse Problem

$$\Delta \boldsymbol{\theta} \approx \mathbf{g}(\Delta \mathbf{x}) = \mathbf{J}_f^{-1}(\boldsymbol{\theta}) \cdot \Delta \mathbf{x}$$

- Invertierung ist möglich, wenn:
 - $\mathbf{J}_f(\boldsymbol{\theta})$ quadratisch ist (nicht-redundante Roboter, d.h. $n = m$)
 - $\mathbf{J}_f(\boldsymbol{\theta})$ vollen Rang hat

Numerische Methoden: Pseudoinverse

Pseudoinverse: Verallgemeinerung der inversen Matrix auf singuläre und nicht-quadratische Matrizen $A \in \mathbb{R}^{m \times n}$ (**redundante Roboter**)

Definition: Moore-Penrose Pseudoinverse (bei vollem Zeilenrang*)

$$A^+ = A^T (AA^T)^{-1}$$

Es gelten:

$$(A^+)^+ = A$$

$$(A^T)^+ = (A^+)^T$$

$$(\lambda A)^+ = \lambda^{-1} A^+, \text{ für ein } \lambda \neq 0$$

* Voller Zeilenrang ist bei J_f üblicherweise gegeben. Ausnahme: **Singularitäten!**

Pseudoinverse: Herleitung

- Berechne die im Sinne der Summe der Fehlerquadrate (least squares) bestmögliche Lösung eines Systems von linearen Gleichungen

$$\begin{array}{ll}
 A\mathbf{x} = \mathbf{b} & // A \text{ ist a rechteckige Matrix, nicht invertierbar} \\
 A^T A \mathbf{x} = A^T \mathbf{b} & // A^T A \text{ ist eine quadratische Matrix, invertierbar}
 \end{array}$$

$$\underbrace{(A^T A)^{-1} A^T A}_I \mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$$

$$\hat{\mathbf{x}} = \underbrace{(A^T A)^{-1} A^T}_{A^+} \mathbf{b} \quad // \hat{\mathbf{x}} \text{ ist eine Least-Squares-Lösung von } A\mathbf{x} = \mathbf{b}$$

$$\hat{\mathbf{x}} = A^+ \mathbf{b} \quad // A^+ \text{ ist die Pseudoinverse von } A$$

Pseudoinverse

- Die Pseudoinverse minimiert den Fehler $\|J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}\|^2$; sie findet die norm-minimale Lösung $\|\dot{\boldsymbol{\theta}}\|^2$

$$\min_{\dot{\boldsymbol{\theta}}} \|J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}\|^2 = \min_{\dot{\boldsymbol{\theta}}} (J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}})^T (J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}})$$

$$\nabla_{\dot{\boldsymbol{\theta}}} \|J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}\|^2 = 2 J_f^T (J_f \dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}) = 0$$

$$\dot{\boldsymbol{\theta}} = \underbrace{(J_f^T J_f)^{-1} J_f^T}_{J_f^+} \dot{\mathbf{x}}$$

$$A^+ = (A^T A)^{-1} A^T$$

Pseudoinverse: Zusammenfassung

1. Vorwärtskinematik als Funktion:

$$\mathbf{x}(t) = f(\boldsymbol{\theta}(t))$$

$\mathbf{x}(t) \in \mathbb{R}^6$: TCP-Pose

$\boldsymbol{\theta}(t) \in \mathbb{R}^n$: Gelenkwinkelstellungen

2. Ableitung nach der Zeit:

$$\frac{d\mathbf{x}(t)}{dt} = \dot{\mathbf{x}}(t) = J_f(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}(t)$$

$\dot{\mathbf{x}}(t) \in \mathbb{R}^6$: TCP-Geschwindigkeiten

$\dot{\boldsymbol{\theta}}(t) \in \mathbb{R}^n$: Gelenkgeschwindigkeiten

$J_f(\boldsymbol{\theta}) \in \mathbb{R}^{6 \times n}$: Jacobi Matrix

3. Übergang zum Differenzenquotienten:

$$\Delta \mathbf{x} \approx J_f(\boldsymbol{\theta})\Delta \boldsymbol{\theta}$$

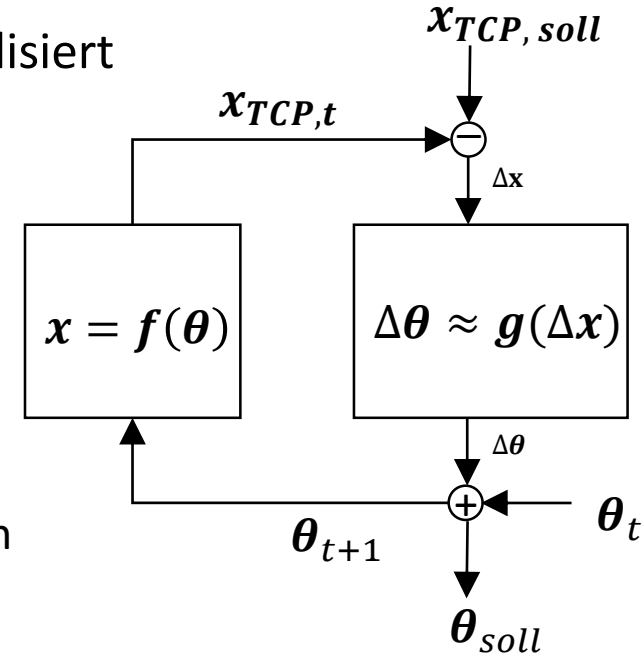
$\Delta \mathbf{x} \in \mathbb{R}^6$: Fehler in der TCP-Pose

$\Delta \boldsymbol{\theta} \in \mathbb{R}^n$: Fehler in Gelenkstellungen

4. Umkehrung: $\Delta \boldsymbol{\theta} \approx J_f^+(\boldsymbol{\theta})\Delta \mathbf{x}$

Pseudoinverse: Iteratives Vorgehen

- **Gegeben:** Sollpose des TCP $x_{TCP, soll}$
- **Gesucht:** Gelenkwinkelvektor θ , der $x_{TCP, soll}$ realisiert
- **Iterativer Ansatz** beginnend bei Initialkonfiguration θ_0 und $x_{TCP,0}$
 1. Berechne $x_{TCP,t}$ in Iteration t aus Gelenkwinkelstellungen θ_t
 2. Berechne Fehler Δx aus $x_{TCP, soll}$ und berechnetes $x_{TCP,t}$
 3. Benutze approximiertes inverses kinematisches Modell g , um Gelenkwinkelfehler $\Delta \theta$ zu berechnen
 4. Berechne $\theta_{t+1} = \theta_t + \Delta \theta$
 5. Fahre mit Iteration $t + 1$ fort



Pseudoinverse : Beispielberechnung (1)

■ Position:

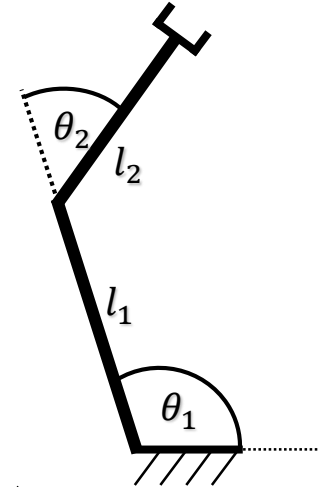
$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

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

■ Geschwindigkeit:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = J_f(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}}(t) = J_f(\boldsymbol{\theta}) \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} =$$

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



Pseudoinverse: Beispielberechnung (2)

- Die Jacobi-Matrix muss invertiert werden:

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

Abkürzungen:

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

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

$$c_i = \cos(\theta_i)$$

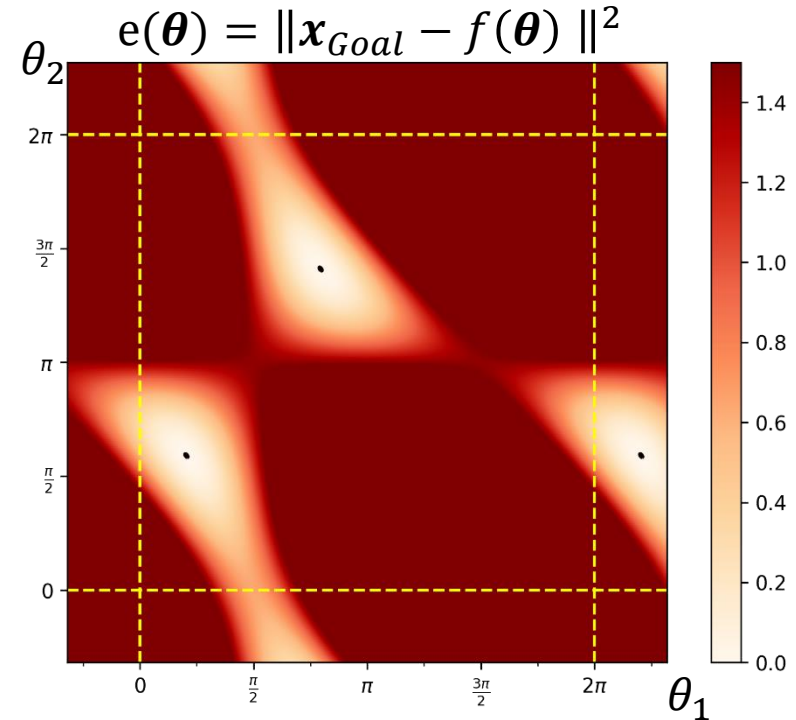
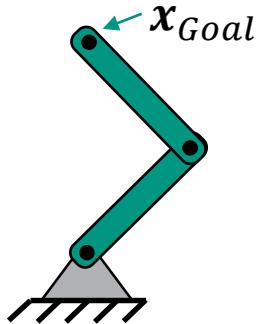
$$s_i = \sin(\theta_i)$$

- Für $\theta_2 = n \cdot \pi$, $n \in \mathbb{Z}$ ist $J_f(\boldsymbol{\theta})$ singulär!

Pseudoinverse: Numerisches Beispiel (1)

2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$

Zielpose $x_{Goal} = (0, 1.2)^T$



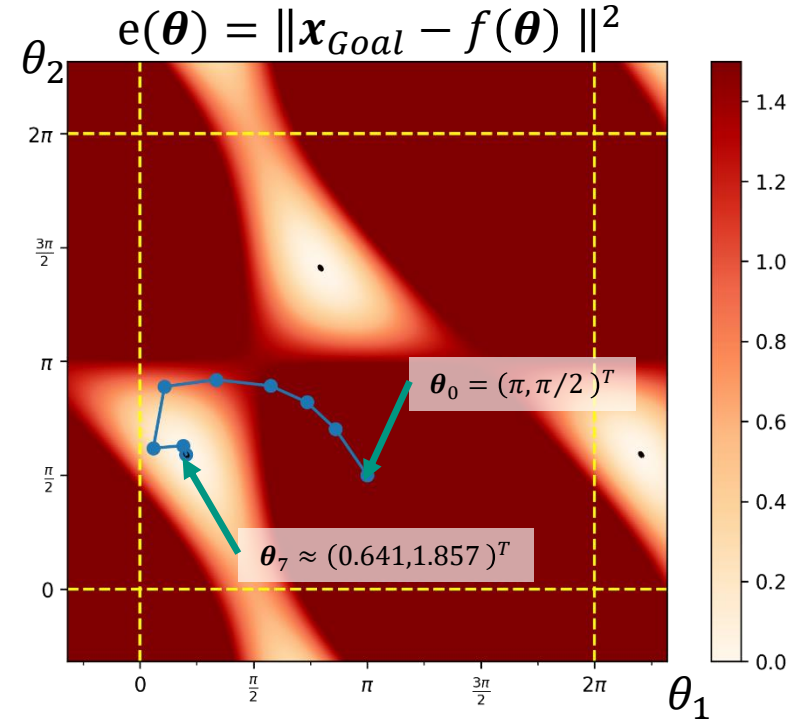
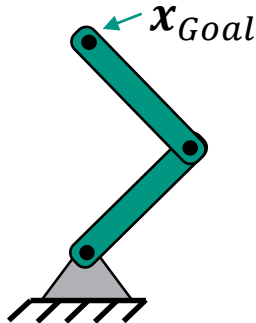
Pseudoinverse: Numerisches Beispiel (2)

2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$

Zielpose $x_{Goal} = (0, 1.2)^T$

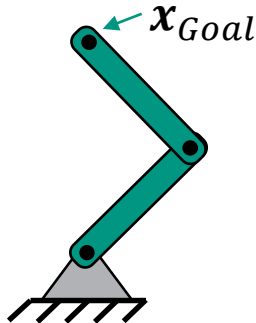
Schrittlänge: $\gamma = 0.2$

Start: $\theta_0 = (\pi, \frac{\pi}{2})^T$

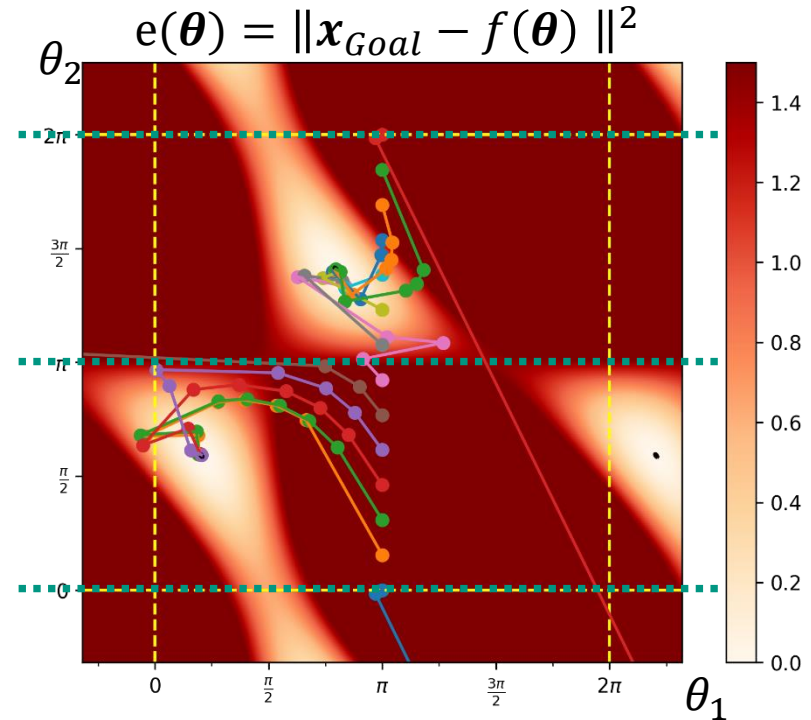


Pseudoinverse: Numerisches Beispiel (3)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$
- Schrittlänge: $\gamma = 0.2$
- Verschiedene Startpunkte:
 $\theta_0 = (\pi, \theta_{2,start})^T$



Singularitäten
bei $\theta_2 = n \cdot \pi$



Pseudoinverse: Singularitäten

- **Pseudoinverse ist in der Nähe von Singularitäten instabil**
- Umgang mit Singularitäten
 - Vermeidung von Singularitäten
(nicht immer möglich)
 - **Damped least squares**
(auch **Levenberg-Marquardt Minimierung**)

Pseudoinverse: Damped Least Squares (1)

■ Die Pseudoinverse $J_f^+(\boldsymbol{\theta})$ löst die Gleichung $J_f(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = \Delta\mathbf{x}$ optimal nach $\Delta\boldsymbol{\theta}$.

■ Optimal bezieht sich auf die Summe der Fehlerquadrate

$$\min_{\Delta\boldsymbol{\theta}} \|J_f(\boldsymbol{\theta})\Delta\boldsymbol{\theta} - \Delta\mathbf{x}\|^2$$

■ **Ansatz:** Minimiere stattdessen (Regularisierung einführen)

$$\min_{\Delta\boldsymbol{\theta}} \|J_f(\boldsymbol{\theta})\Delta\boldsymbol{\theta} - \Delta\mathbf{x}\|^2 + \lambda^2 \|\Delta\boldsymbol{\theta}\|^2$$

mit einer **Dämpfungskonstante** $\lambda > 0$.

Pseudoinverse: Damped Least Squares (2)

■ Ansatz:

$$\min_{\Delta \boldsymbol{\theta}} \|J_f(\boldsymbol{\theta})\Delta \boldsymbol{\theta} - \Delta \mathbf{x}\|^2 + \lambda^2 \|\Delta \boldsymbol{\theta}\|^2$$

■ Die zugehörige Lösung ist

$$(J^T J + \lambda^2 I)\Delta \boldsymbol{\theta} = J^T \Delta \mathbf{x}$$

■ Daraus ergibt sich

$$\Delta \boldsymbol{\theta} = \underbrace{(J^T J + \lambda^2 I)^{-1}}_{\in \mathbb{R}^{n \times n}} J^T \Delta \mathbf{x} = J^T \underbrace{(J J^T + \lambda^2 I)^{-1}}_{\in \mathbb{R}^{m \times m}} \Delta \mathbf{x}$$

■ Hier: $J = J_f(\boldsymbol{\theta})$, $m = 6$

Pseudoinverse: Damped Least Squares (3)

■ Lösung:

$$\Delta\theta = \underbrace{(J^T J + \lambda^2 I)^{-1}}_{\in \mathbb{R}^{n \times n}} J^T \Delta\mathbf{x} = J^T \underbrace{(J J^T + \lambda^2 I)^{-1}}_{\in \mathbb{R}^{m \times m}} \Delta\mathbf{x}$$

- Die Dämpfungskonstante $\lambda > 0$ muss anwendungsspezifisch gewählt werden
 - **Groß genug** für numerische Stabilität in Nähe von Singularitäten
 - **Klein genug** für schnelle Konvergenzrate

■ Hier: $J = J_f(\boldsymbol{\theta})$

Numerische Methoden: Stabilitätsbetrachtung (1)

Beide Ansätze (**Pseudoinverse** und **Damped Least Squares**) können durch Singularitäten instabil werden.

Eine Analyse der Stabilität ist über eine **Singulärwertzerlegung** (SVD, engl. „Singular Value Decomposition“) möglich

Singulärwertzerlegung: Eine Matrix $J \in \mathbb{R}^{m \times n}$ wird dargestellt durch zwei orthogonale Matrizen $U \in \mathbb{R}^{m \times m}$ und $V \in \mathbb{R}^{n \times n}$ und eine Diagonalmatrix $D \in \mathbb{R}^{m \times n}$, in der Form

$$J = UDV^T$$

O. B. d. A.: **Singulärwerte** σ_i auf der Diagonalen von D seien sortiert:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0$$

Numerische Methoden: Stabilitätsbetrachtung (2)

■ Singulärwertzerlegung: $J = UDV^T$

■ Die Singulärwertzerlegung von J **existiert immer** und sie erlaubt die folgende Darstellung von J

$$J = \sum_{i=1}^m \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T ,$$

$\mathbf{u}_i$ und $\mathbf{v}_i$ sind die Spalten von U und V , $r = \text{rang } J$.

■ Für die Pseudoinverse J^+ gilt (durch Orthogonalität von U und V):

$$J^+ = VD^+U^T = \sum_{i=1}^r \sigma_i^{-1} \mathbf{v}_i \mathbf{u}_i^T$$

Numerische Methoden: Stabilitätsbetrachtung (3)

■ Erinnerung **Damped-Least-Squares**: $\Delta\theta = J^T(JJ^T + \lambda^2 I)^{-1} \Delta x$

■ Für die innere (zu invertierende) Matrix gilt:

$$JJ^T + \lambda^2 I = (UDV^T)(VD^T U^T) + \lambda^2 I = U(DD^T + \lambda^2 I)U^T$$

■ $DD^T + \lambda^2 I$ ist eine **nicht-singuläre** Diagonalmatrix mit den Diagonaleinträgen $\sigma_i^2 + \lambda^2$. Daher ist $(DD^T + \lambda^2 I)^{-1}$ eine Diagonalmatrix mit den Diagonaleinträgen $(\sigma_i^2 + \lambda^2)^{-1}$.

■ Es folgt:

$$J^T(JJ^T + \lambda^2 I)^{-1} = V D^T (DD^T + \lambda^2 I)^{-1} U^T = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T$$

Numerische Methoden: Stabilitätsbetrachtung (4)

Pseudoinverse:

$$J^+ = \sum_{i=1}^r \underbrace{\frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T}_{\rightarrow \infty (\sigma_i \rightarrow 0)}$$

Damped-Least-Squares:

$$J^T (J J^T + \lambda^2 I)^{-1} = \sum_{i=1}^r \underbrace{\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \mathbf{u}_i^T}_{\rightarrow 0 (\sigma_i \rightarrow 0)}$$

Die Invertierung von J hat in beiden Fällen eine ähnliche Form.

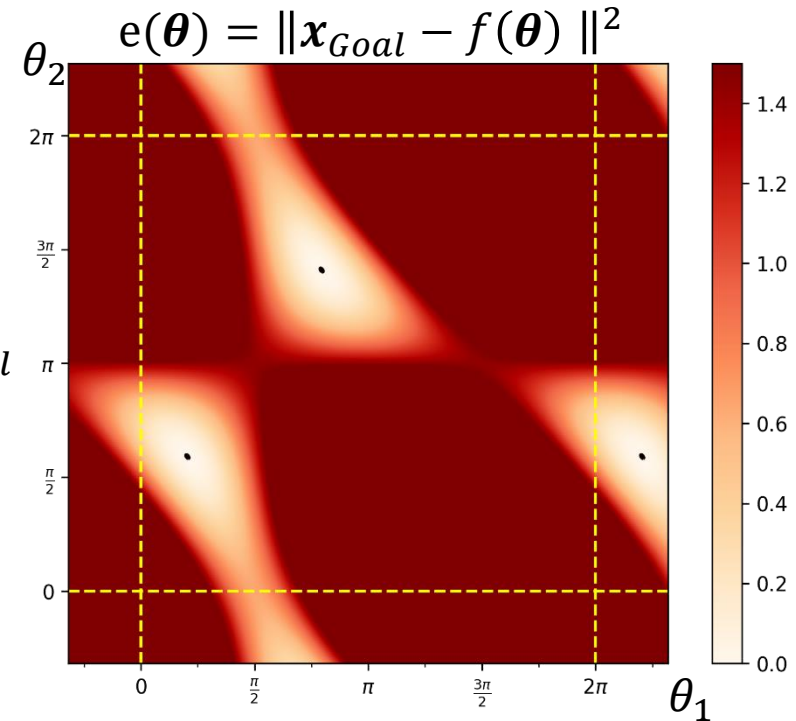
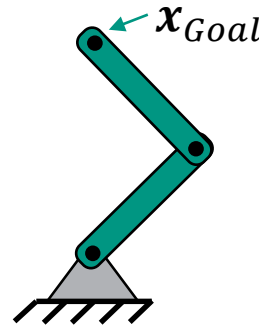
Die Pseudoinverse wird instabil, wenn ein $\sigma_i \rightarrow 0$ (Singularität)

Für große σ_i (verglichen mit λ) verhält sich Damped-Least-Squares wie die Pseudoinverse

Für $\sigma_i \rightarrow 0$ verhält sich Damped-Least-Squares wohldefiniert

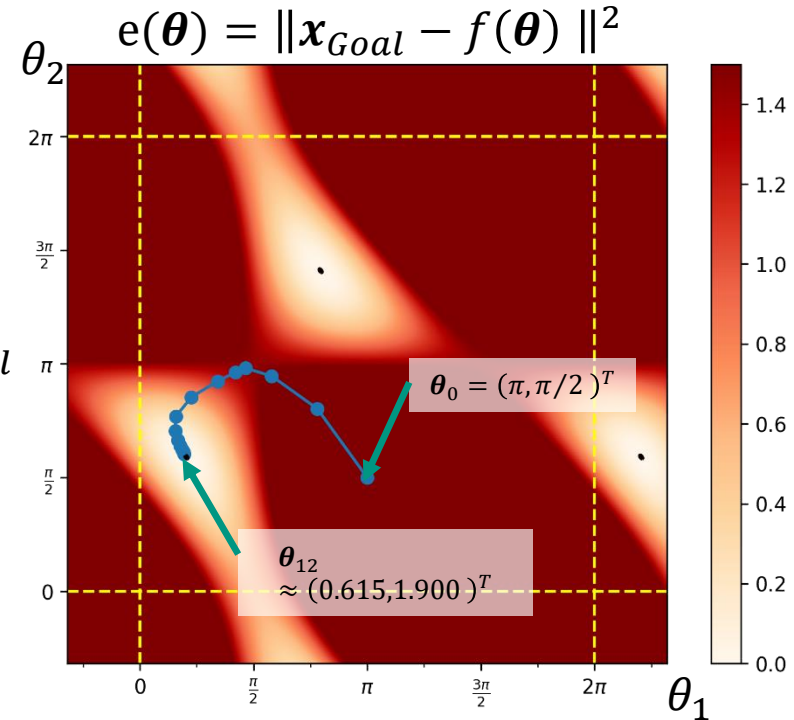
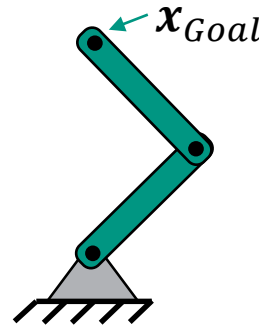
Damped Least Squares: Beispiel (1)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$



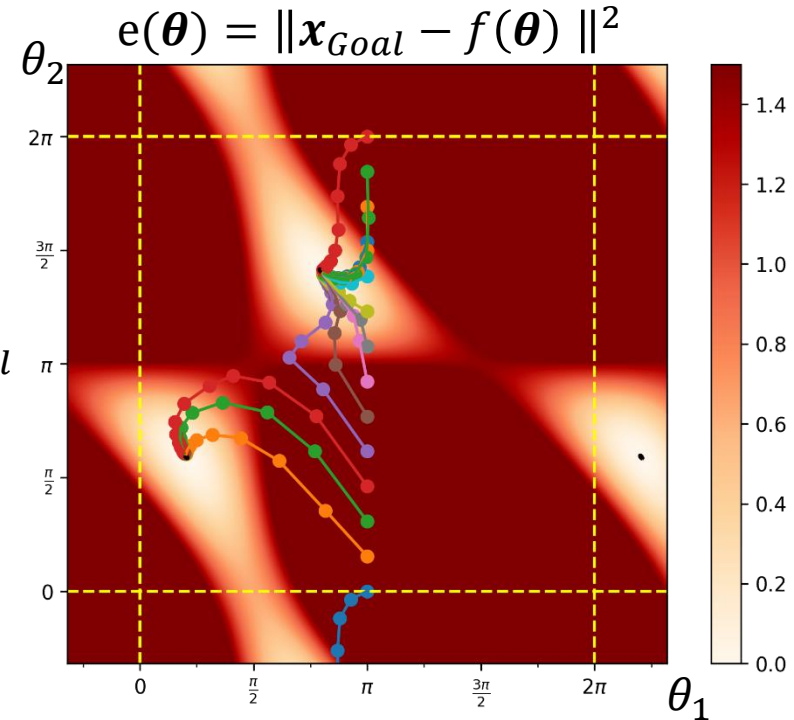
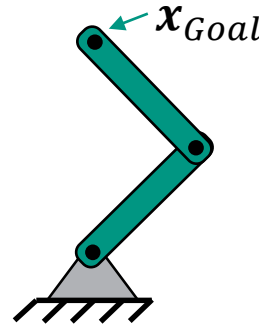
Damped Least Squares: Beispiel (2)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$
- Schrittlänge: $\gamma = 0.5$
- Dämpfung: $\lambda = 0.5$
- Start: $\theta_0 = (\pi, \frac{\pi}{2})^T$



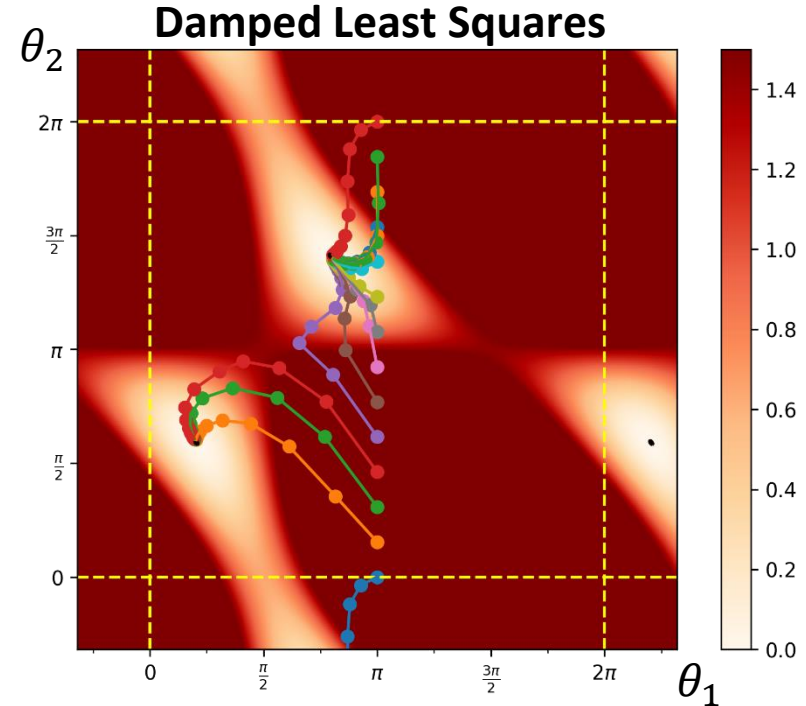
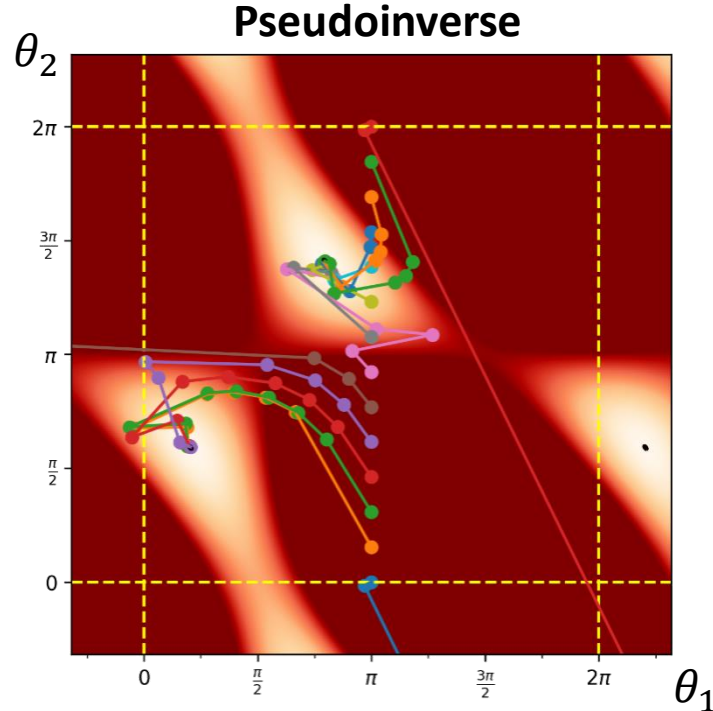
Damped Least Squares: Beispiel (3)

- 2-DoF planarer Roboter: $\theta = (\theta_1, \theta_2) \in \mathcal{C}$
- Zielpose $x_{Goal} = (0, 1.2)^T$
- Schrittlänge: $\gamma = 0.5$
- Dämpfung: $\lambda = 0.5$
- Verschiedene Startpunkte:
 $\theta_0 = (\pi, \theta_{2,start})^T$



Vergleich: Pseudoinverse vs. Damped Least Squares

Singularitäten bei $\theta_2 = n \cdot \pi, n \in \mathbb{Z}$



Inverses Kinematisches Problem

Geschlossene Methoden

Geometrische Methoden

Algebraische Methoden

Numerische Methoden

Zusammenfassung

Zusammenfassung: Kinematik

Direkte Kinematik:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m \quad \boldsymbol{x} = f(\boldsymbol{\theta})$$

Inverse Kinematik:

$$F: \mathbb{R}^m \rightarrow \mathbb{R}^n \quad \boldsymbol{\theta} = F(\boldsymbol{x})$$

Fälle:

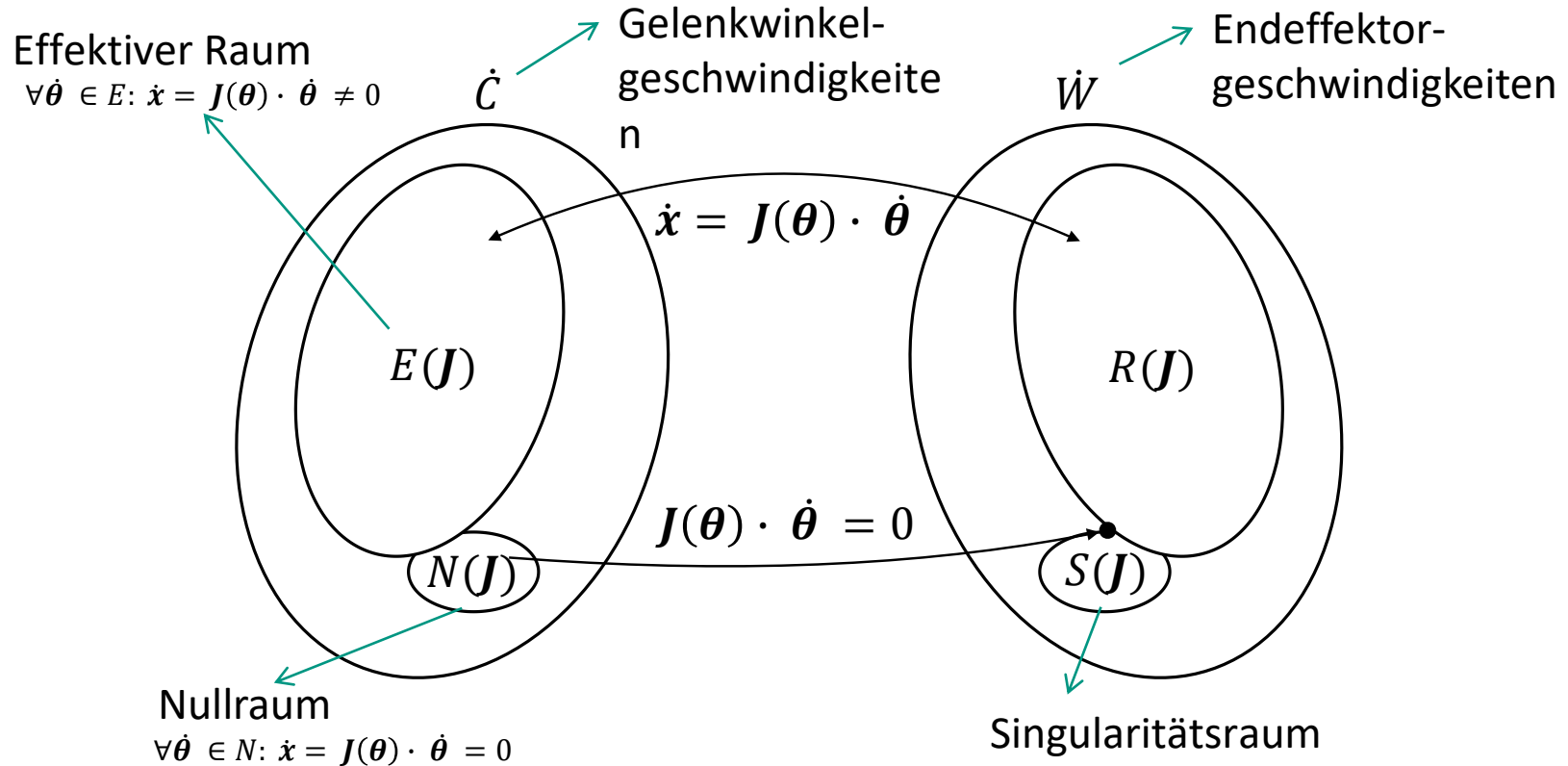
Es existiert eine eindeutige Lösung.

Es existiert eine endliche Menge von Lösungen.

Es existiert eine unendliche Menge von Lösungen.

Es existiert keine Lösung.

Wichtige Räume der Robotik



Important Spaces in Robotics (English)

