KIT-Fakultät für Informatik
Prof. Dr.-Ing. Tamim Asfour

# Musterlösungen zur Klausur

Robotik I: Einführung in die Robotik

am 23. Februar 2023

| Name: | Vorname: | Matrikelnummer: |
|---|---|---|
| Denavit | Hartenberg | $\frac{\pi}{2}$ |

| | |
|---|---|
| Aufgabe 1 | von 7 Punkten |
| Aufgabe 2 | von 8 Punkten |
| Aufgabe 3 | von 10 Punkten |
| Aufgabe 4 | von 7 Punkten |
| Aufgabe 5 | von 8 Punkten |
| Aufgabe 6 | von 5 Punkten |

| | |
|---|---|
| **Gesamtpunktzahl:** | 45 von 45 Punkten |

| | |
|---|---|
| | **Note:** **1,0** |

# Aufgabe 1 *Transformationen*

1. Die homogene Transformationsmatrix $\mathbf{T}_1$, welche die Pose von $\mathcal{F}_1$ im WKS $\mathcal{F}_w$ beschreibt, ist gegeben durch $\boxed{\text{1 P.}}$

$$\mathbf{T}_1 = \begin{pmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 & 0.4 \\ 0 & 0 & 1 & 0.5 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Die Rotationsmatrix $\mathbf{R}$ von $\mathcal{F}_1$ nach $\mathcal{F}_2$ ist $\boxed{\text{1 P.}}$

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. Da die Rotation $\mathbf{R}$ im Koordinatensystem $\mathcal{F}_1$ dargestellt wird, wird sie von rechts mit der Rotationsmatrix $\mathbf{R}_1$ multipliziert, um die Orientierung von $\mathcal{F}_2$ zu erhalten. $\boxed{\text{2 P.}}$

$$\mathbf{R}_2 = \mathbf{R}_1\mathbf{R} = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

4. Die Zielposition der Flasche im Weltkoordinatensystem ist $\boxed{\text{2 P.}}$

$$\mathbf{p}_w = \mathbf{R}_2\mathbf{p}_2 + \mathbf{t}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} -0.1 \\ 0.1 \\ 0.2 \end{pmatrix} + \begin{pmatrix} 0.4 \\ 0.5 \\ 1.0 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 0.7 \\ 0.9 \end{pmatrix}$$

5. Nachteil von Rotationsmatrizen gegenüber Quaternionen: $\boxed{\text{1 P.}}$

   - Redundant (9 Werte statt 4)
   - Interpolation ist schwieriger

# Aufgabe 2 *Kinematik*

1. Jacobi-Matrix: **3 P.**

$$\mathbf{J}(\boldsymbol{\theta}) = \left(\frac{\delta f}{\delta d_1}, \frac{\delta f}{\delta \theta_2}, \frac{\delta f}{\delta d_3}\right) = \begin{pmatrix} 1 & -\sin(\theta_2) & 0 \\ 0 & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. Matrix der Manipulierbarkeit: **2 P.**

$$\mathbf{A}(\boldsymbol{\theta}) = \mathbf{J}(\boldsymbol{\theta}) \cdot \mathbf{J}(\boldsymbol{\theta})^\top = \begin{pmatrix} 1 & -\sin(\theta_2) & 0 \\ 0 & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 + \sin(\theta_2)^2 & -\cos(\theta_2) \cdot \sin(\theta_2) & 0 \\ -\cos(\theta_2) \cdot \sin(\theta_2) & \cos(\theta_2)^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. Matrix der Manipulierbarkeit: **1 P.**

$$\mathbf{A}((5, 180^\circ, 10)^\top) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4. Die Bewegungen des Endeffektors in alle Richtungen (gleichmäßig) ist uneingeschränkt möglich sind. **1 P.**

   Mögliche Begründungen: Alle Singulärwerte sind 1. Das Ellipsoid ist eine Kugel. Die inverse Kondition ist 1.

5. Beziehung: **1 P.**

   Die Transponierte der Jacobi-Matrix $\mathbf{J}^\top(\boldsymbol{\theta})$ bildet Kräfte und Momente am Endeffektor $\mathbf{F}(t)$ auf Drehmomente in den Gelenken $\boldsymbol{\tau}(t)$ ab.

$$\boldsymbol{\tau}(t) = \mathbf{J}^\top(\boldsymbol{\theta}) \cdot \mathbf{F}(t)$$

# Aufgabe 3 *Bewegungsplanung*

1. Definitionen: 3 P.

   - Ein **vollständiger Algorithmus** findet für ein Problem mindestens eine Lösung oder erkennt in endlicher Zeit, dass keine Lösung existiert

     Beispiele: A*, Sichtgraphen, Voronoi-Diagramme

   - Ein **probabilistisch vollständiger Algorithmus** findet (gegeben unendlich langer Laufzeit) mindestens eine Lösung, falls sie existiert . Allerdings kann mit probabilistisch vollständigen Algorithmen nicht ermittelt werden, ob keine Lösung existiert.

     Beispiele: RRT, PRM, DRM, RRT + Varianten

2. Voronoi-Region: Für alle Punkte der Voronoi-Region gilt, dass deren Abstand zum Hindernis geringer ist als zu allen anderen Hindernissen. 1 P.

3. Vor- und Nachteil: 1 P.

   - Mögliche Vorteile:
     - Probabilistisch-vollständig
     - geeignet für hochdimensionale Räume
     - keine Vorverarbeitung (geeignet für dynamische Umgebung)
   - Mögliche Nachteile:
     - Im Allgemeinen nicht optimal
     - Nicht deterministisch
     - Nicht vollständig, terminiert nicht, wenn keine Lösung existiert
     - Im Allgemeinen nicht gut bei Narrow Path-Problem

4. Rewiring-Schritt: Optimieren des Suchbaums durch Hinzufügen und Entfernen von Kanten um den optimale Pfad zwischen Start- und Zielkonfiguration zu erhalten. 1 P.

5. **Algorithmus für statische Umgebungen**: 2 P.

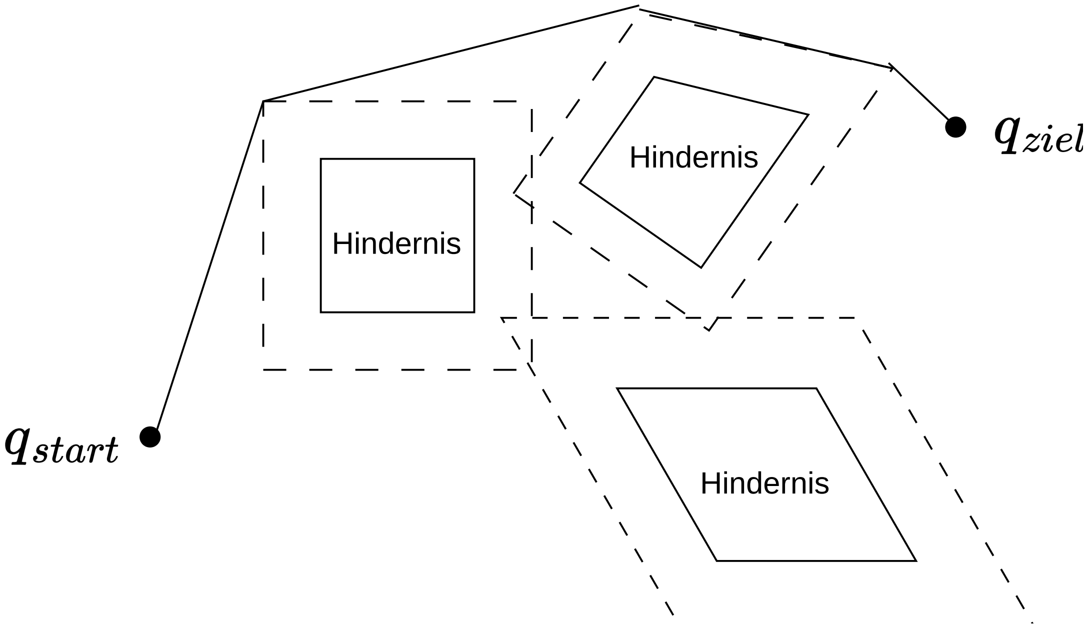
   Geeignet sind sowohl Algorithmen aus der Kategorie der *vollständigen* sowie *probabilistisch vollständigen Algorithmen*. *Vollständige Algorithmen* finden die optimale Lösung, z.B. A*. Jedoch steigt der Berechnungsaufand stark mit der Anzahl der Freiheitsgrade des Roboters. *Probabilistisch vollständige Algorithmen* können hierfür ebenso verwendet werden, z.B. single-query Algorithmen (z.B. RRT) sowie multi-query Algorithmen (z.B. PRM), da sie effizient eine Lösung für hochdimensionale Bewegungsplanungsprobleme finden.

   **Algorithmus für dynamische Umgebungen**:

   Geeignet sind RRT sowie RRT-Varianten (RRT*, BiRRT, ...), da es sich um ein Algorithmus zur einmaligen Anfrage handelt, also auch mit sich veränderten Umgebungen gut umgegangen werden kann.

   Ebenso geeignet sind Potentialfeldmethoden, da das Potentialfeld in jedem Schritt neu evaluiert wird.

6. Sichtgraph. 2 P.



# Aufgabe 4 *Greifen*

1. Bei Kraftgeschlossenheit werden starre Punktkontakte **mit** Reibung angenommen, während Formgeschlossenheit **keine** Reibung annimmt. $\boxed{1\text{ P.}}$

2. Griff:

   (a) Wrenches: $\boxed{2\text{ P.}}$

   Normalkraft $\mathbf{f}_1$ wirkt in Richtung $(1,1)^\top$ mit $|\mathbf{f}_1| = 1$

$$\Longrightarrow \mathbf{f}_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\top \quad \text{(nicht gefragt)}$$

   Mit Reibung $\Longrightarrow$ GWS von $C_1$ wird von 2 Wrenches $\mathbf{w}_{1,1}, \mathbf{w}_{1,2}$ aufgespannt.
   Reibungskoeffizient $\mu = 1 \Longrightarrow$ Öffnung der Reibungskegel $= 45°$

$$\Longrightarrow \text{Kräfte } \mathbf{f}_{1,1} = \left(0, \sqrt{2}\right)^\top, \quad \mathbf{f}_{1,2} = \left(\sqrt{2}, 0\right)^\top \quad \text{(siehe Abbildung 1)}$$

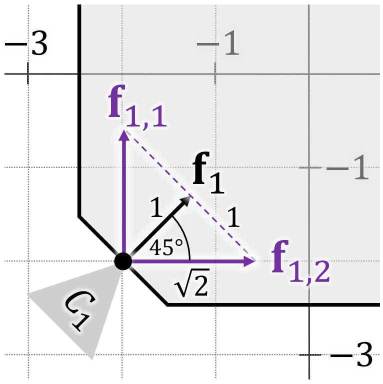



Abbildung 1: Reibungsdreieck von Griff $G$

   Damit ergeben sich die Drehmomente $\tau_{1,1}, \tau_{1,2}$ zu:

$$\tau_{1,1} = \mathbf{d}_1 \times \mathbf{f}_{1,1} = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \times \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix} = -2 \cdot \sqrt{2} - (-2) \cdot 0 = -2\sqrt{2}$$

$$\tau_{1,2} = \mathbf{d}_1 \times \mathbf{f}_{1,2} = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \times \begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix} = -2 \cdot 0 - (-2) \cdot \sqrt{2} = 2\sqrt{2}$$

$$\Longrightarrow \mathbf{w}_{1,1} = \left(0,\ \sqrt{2},\ -2\sqrt{2}\right)^\top, \quad \mathbf{w}_{1,2} = \left(\sqrt{2},\ 0,\ 2\sqrt{2}\right)^\top$$

   *Mit **falscher** Annahme* $\mathbf{f}_1 = (1,1)^\top$:

$$\Longrightarrow \text{Kräfte } \mathbf{f}_{1,1} = (0,1)^\top, \quad \mathbf{f}_{1,2} = (1,0)^\top$$

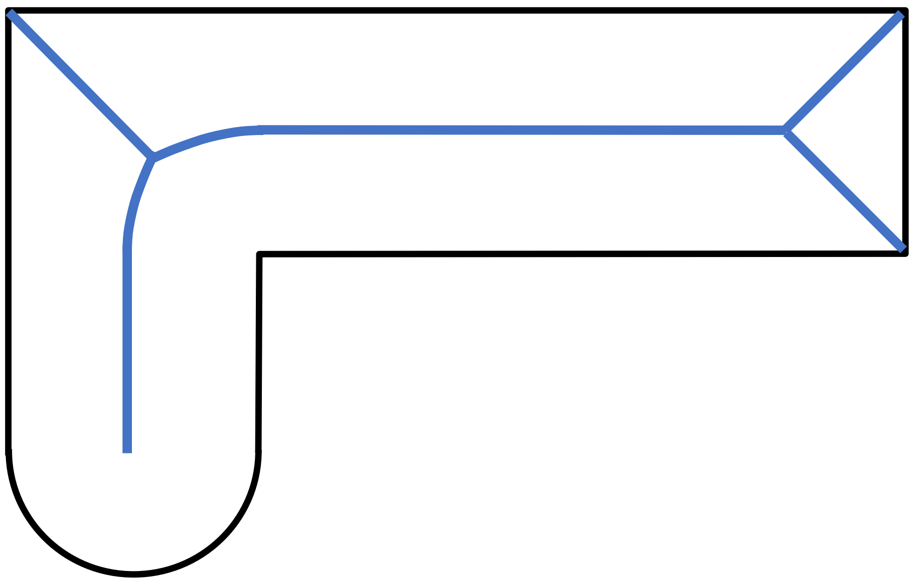
$$\tau_{1,1} = \mathbf{d}_1 \times \mathbf{f}_{1,1} = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -2 \cdot 1 - (-2) \cdot 0 = -2$$

$$\tau_{1,2} = \mathbf{d}_1 \times \mathbf{f}_{1,2} = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = -2 \cdot 0 - (-2) \cdot 1 = 2$$

$$\Longrightarrow \mathbf{w}_{1,1} = (0,1,-2)^\top, \quad \mathbf{w}_{1,2} = (1,0,2)^\top$$

(b) Schranke: $\varepsilon \leq 0$ (äquivalent: $\varepsilon = 0$)
Begründung: Kein Wrench erzeugt Kräfte in Richtung $-y = (0, -1)$.

3. Objekthülle: 2 P.



# Aufgabe 5 *Bildverarbeitung*

1. Rauschunterdrückung: 1 P.

   Das Median-Filter kann zur Unterdrückung von Salz- & Pfefferrauschen verwendet werden, solange die Anzahl der Störpixel nicht überwiegt, da der mittlere Wert aller Pixel im Filterbereich verwendet wird und Ausreißer so eliminiert werden können.

2. Korrelation vs. Faltung: 1 P.

   Korrelation ist eine Filteroperation, bei der eine Filtermaske über das Bild bewegt und die Summe der Produkte an jedem Bildpunkt berechnet wird. Die Faltung ist eine Filteroperation, bei der das Filter zuerst um 180° gedreht wird.

3. Segmentierung: 1 P.

   Begriff: Segmentierung ist die Aufteilung einer Menge in aussagekräftige Segmente.

   Schwellenwertfilterung: Ist die Konvertierung eines Grauwertbildes in ein binäres Bild. Hierfür wird die Intensität von jedem Pixel $(u, v)$ mit einem vordefinierten Schwellenwert T verglichen und der Wert des Pixels auf 0 gesetzt falls der Pixelwert den Schwellenwert unterschreitet.

4. Erweitertes Kameramodell:

   (a) Unterschied: Das erweiterte Kameramodell verwendet unabhängige Brennweiten $f_x$ und $f_y$ in $u$ und $v$ Richtung (rechteckige Pixel) und der Hauptpunkt $(c_x, c_y)$, ist nicht identisch mit dem Ursprung des Kamerakoordinatensystems . 1 P.

   (b) Bildkoordinaten: 2 P.

   Erweitertes Kameramodell:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \frac{1}{z} \begin{pmatrix} f_x \cdot x \\ f_y \cdot y \end{pmatrix}$$

$$\mathbf{px}_1 = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \qquad \mathbf{px}_2 = \begin{pmatrix} 6 \\ 68 \end{pmatrix}$$

5. Visual Servoing: 2 P.

   (1) Positionsbasiert: Aktuelle Pose der Hand wird aus Bildmerkmalen extrahiert. Die Regeldifferenz wird im kartesischen Raum gebildet (3D).

   (2) Bildbasiert: Regeldifferenz wird direkt aus der Position der Bildmerkmale (im Bildraum) berechnet (z.B. Kamera-In-Hand System)

# **Aufgabe 6** *Roboterprogrammierung*

1. Symbolisch vs. sub-symbolisch: 2 P.

   **Subsymbolisch**: Segmentierung von Bewegungstrajektorie, d.h. die Identifikation von Schlüsselpunkten (Key Points) der Demonstration

   Ergebnis: Beschreibung der resultierenden Segmente in einer generalisierten Form (Funktionsapproximation)

   **Symbolisch**: Lernen einer Sequenz von Aktionen, die die Demonstration repräsentiert. Diese stellt eine Repräsentation auf höherer Ebene dar.

   Ergebnis: Task-Modell

2. **Bewegungssegmentierung**: Unterteilung der Bewegungstrajektorie(n) in einfach darstellbare Teile. Hierzu müssen Schlüsselpunkte (key points) der Demonstration bestimmt werden. 2 P.

   Kriterien:

   - Änderung der Bewegung
   - Suchen nach bestimmten Mustern
   - Lokale Minima und Maxima sowie Pausen im Aufgabenraum
   - Lokale Minima und Maxima sowie Pausen im Konfigurationsraum
   - Minimierung des Approximationsfehlers zwischen der Demonstration und einer bekannten Grundfunktion

3. Eigenschaften: 1 P.

   - Hierarchisch
   - Interlevel-Transitionen
   - Orthogonalität
   - Zustandsaktionsphasen: Entry, Exit, Throughout