

# Musterlösungen zur Klausur

Robotik I: Einführung in die Robotik

am 24. Juli 2017, 18:00 – 19:00 Uhr

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

Aufgabe 1	von 4 Punkten
Aufgabe 2	von 11 Punkten
Aufgabe 3	von 8 Punkten
Aufgabe 4	von 7 Punkten
Aufgabe 5	von 7 Punkten
Aufgabe 6	von 8 Punkten

Gesamtpunktzahl:	45 von 45 Punkten
------------------	-------------------

Note:	<b>1,0</b>
-------	------------

## Aufgabe 1    *Quaternionen*

1. Winkel  $\theta$  und Rotationsachse  $\mathbf{u}$ :

$$\begin{aligned}\mathbf{q} &= \left( \cos\left(\frac{\theta}{2}\right), \mathbf{u} \cdot \sin\left(\frac{\theta}{2}\right) \right) = (0.7, 0, 0.7, 0) \\ \cos\left(\frac{\theta}{2}\right) &= 0.7 \Rightarrow \frac{\theta}{2} \approx \frac{\pi}{4} \Rightarrow \theta \approx \frac{\pi}{2} \\ \mathbf{u} \cdot \sin\left(\frac{\theta}{2}\right) &= \mathbf{u} \cdot \sin\left(\frac{\pi}{4}\right) \approx \mathbf{u} \cdot 0.7 = (0, 0.7, 0) \Rightarrow \mathbf{u} = (0, 1, 0)\end{aligned}$$

Es gibt zwei äquivalente Lösungen (nur eine Lösung wird erwartet):

$$\mathbf{u} = (0, 1, 0), \theta = \frac{\pi}{2} \quad \mathbf{u} = (0, -1, 0), \theta = -\frac{\pi}{2}$$

2. Inverses Quaternion  $\mathbf{q}^{-1}$ :

Da  $\mathbf{q}$  ein Einheitsquaternion ist, ist das inverse Quaternion gleich dem konjugierten Quaternion:

$$\mathbf{q}^{-1} = \mathbf{q}^* = (0.7, 0, -0.7, 0)$$

3. SLERP (Vorteil und Probleme):

- Vorteil: Konstante Winkelgeschwindigkeit bei der Interpolation
- Probleme: Orientierungen in  $SO(3)$  werden durch Einheitsquaternionen doppelt abgedeckt. SLERP berechnet deshalb nicht immer die kürzeste Rotation.

## Aufgabe 2 *Roboterkinematik*

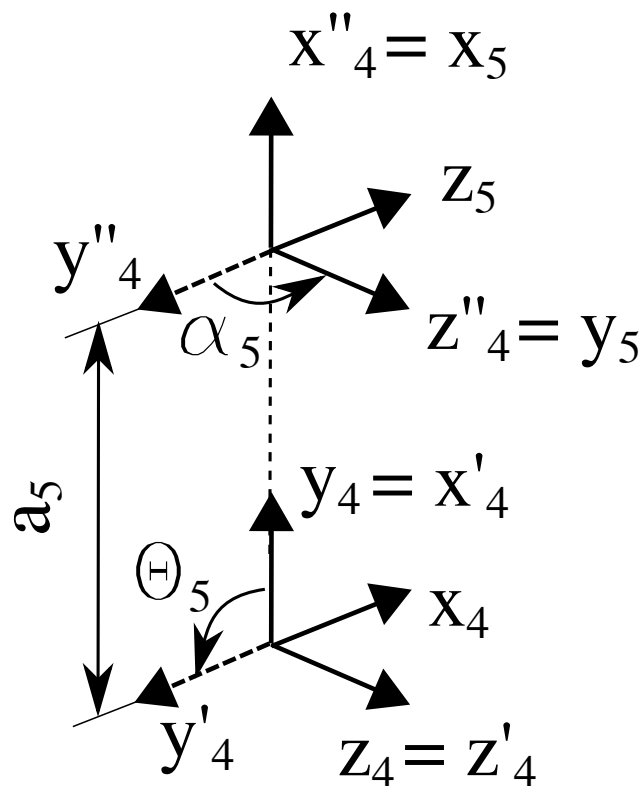
1. DH-Parameter des Roboters:

<i>Gelenk</i>	$\theta_i$ [°]	$d_i$ [mm]	$a_i$ [mm]	$\alpha_i$ [°]
G1	0	$d_1$	110	-90
G2	-90	$d_2$	50	90
G3	180	$d_3$	0	90
G4	$\theta_4 + 90$	0	100	0

2. Anzahl der Rotationsgelenke: 1  
Anzahl der Translationsgelenke: 3

3. Arbeitsraum: Quader

4. Transformation zwischen  $(x_4, y_4, z_4)$  und  $(x_5, y_5, z_5)$ :



5. Dimension:  $6 \times 4$  (6 Zeilen, 4 Spalten)

## Aufgabe 3 *Bahnsteuerung und Bewegungsplanung*

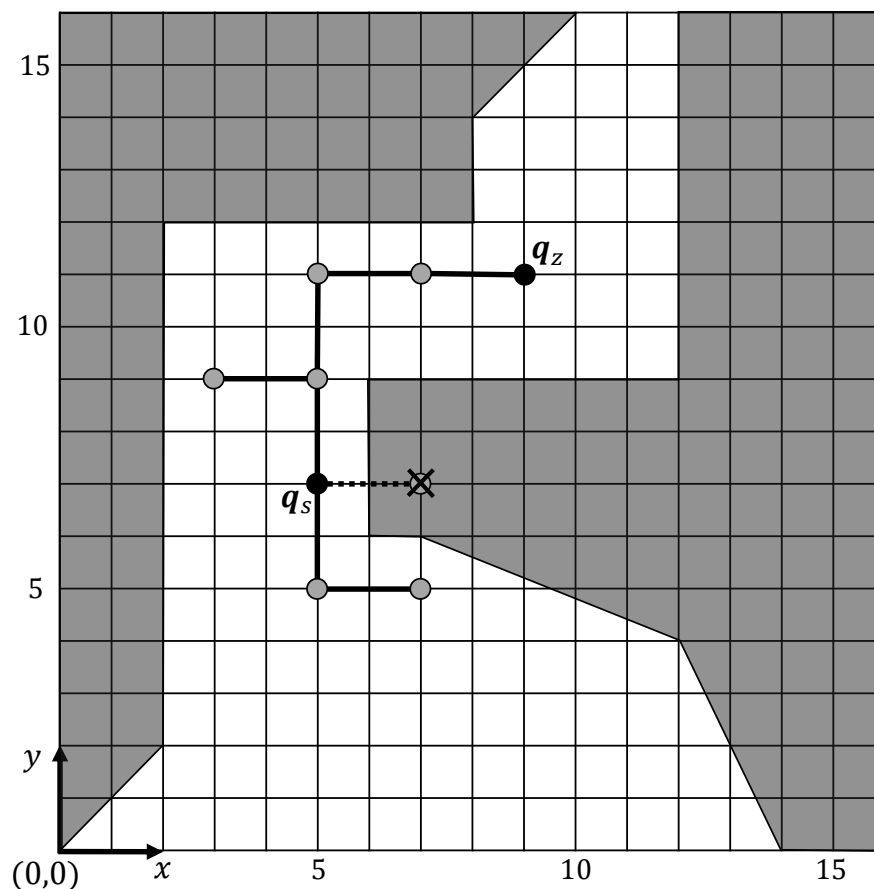
### 1. Synchron und asynchrone Punkt-zu-Punkt Bahnen / Leitachse

Bei asynchronen Punkt-zu-Punkt Bahnen wird jedes Gelenk sofort mit der vorgegebenen (maximalen) Geschwindigkeit angesteuert, jede Gelenkbewegung endet unabhängig von den anderen. Bei synchronen Punkt-zu-Punkt Bahnen wird für jedes Gelenk eine individuelle maximale Geschwindigkeit berechnet, so dass alle Gelenkbewegungen zum gleichen Zeitpunkt enden. Die Leitachse bezeichnet dabei die Achse bzw. das Gelenk mit der längsten Fahrzeit.

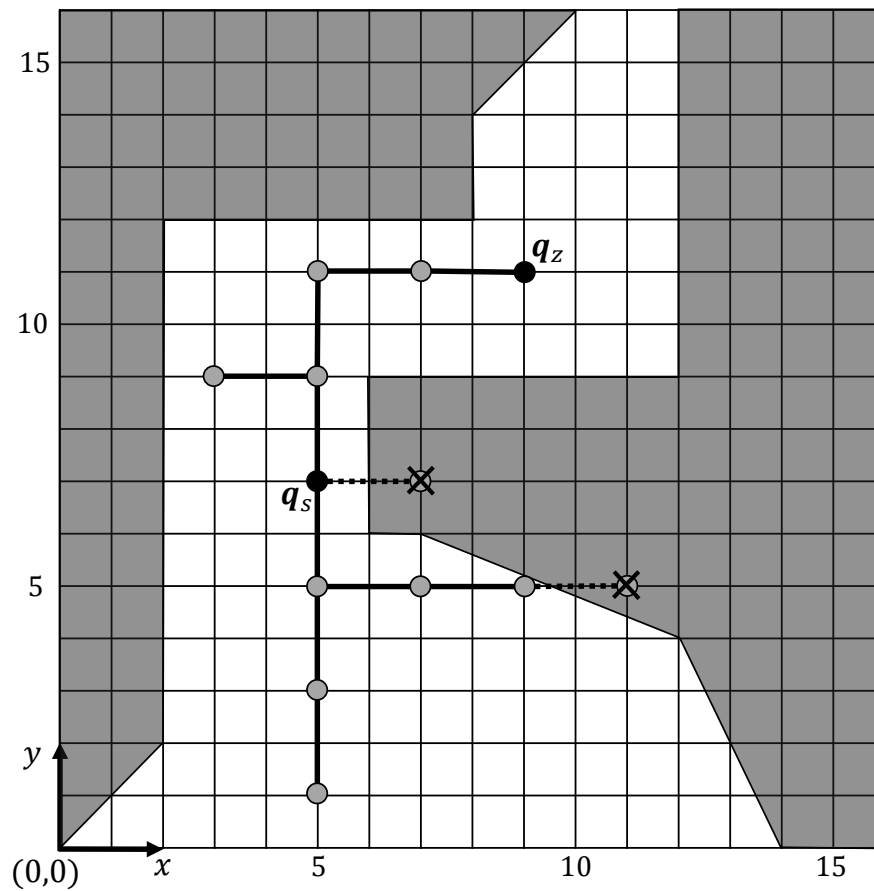
### 2. Inverse Kinematik bei Bahninterpolation in Weltkoordinaten

Die inverse Kinematik muss nach der Interpolation für jeden anzusteuernenden Punkt der interpolierten Bahn gelöst werden, um von dem jeweils interpolierten Punkt in Weltkoordinaten auf eine Konfiguration im Gelenkwinkelraum zu schließen.

### 3. RRT-Algorithmus



*Anmerkung:* Wenn der Hinweis, nur einen Expansionschritt durchzuführen, missachtet wird, ergibt sich das folgende Bild:



#### 4. Enge Passagen

Klassische randomisierte Algorithmen zur Bewegungsplanung (z.B. RRT oder PRM) bestimmen neue Stichproben durch eine gleichverteilte Zufallswahl im Konfigurationsraum. Durchgänge durch *enge Passagen* können nur mit großem Zeitaufwand geplant werden, da die Wahrscheinlichkeit, eine Stichprobe innerhalb der Passage zu wählen, gering ist.

## Aufgabe 4    *Greifplanung*

1. Grifftypen der ersten Ebene bei der Griff-taxonomie nach Cutkosky:

In der Griff-taxonomie nach Mark Cutkosky wird auf der ersten Ebene nach Kraft- und Präzisionsgriffen unterschieden.

2. Griffanalyse und Griffsynthese:

### *Griffanalyse*

Gegeben: Objekt und eine Menge von Kontaktpunkten

Gesucht: Aussagen zur Stabilität des Griffs

### *Griffsynthese*

Gegeben: Objekt und eine Menge von Nebenbedingungen

Gesucht: Eine Menge von Kontaktpunkten

3. Kontaktmodelle:

### *Punktkontakt ohne Reibung:*

Eine an einem Punktkontakt ohne Reibung auf eine Fläche eines Objektes angreifende Kraft wirkt ausschließlich normal zur Fläche.

### *Starrer Punktkontakt mit Reibung:*

Eine an einem starren Punktkontakt mit Reibung auf eine Fläche eines Objektes angreifende Kraft wirkt sowohl normal als auch tangential zur Fläche. Die beiden Kräfte sind über das Coulombsche Reibungsgesetz miteinander verknüpft.

### *Nicht starrer Punktkontakt mit Reibung (Soft-Contact):*

Eine an einem nicht starren Punktkontakt mit Reibung auf eine Fläche eines Objektes angreifende Kraft wirkt sowohl normal als auch tangential. Zusätzlich wirken auch axiale Momente. Es gilt ebenfalls das Coulombsche Reibungsgesetz.

4. Unterschied zwischen Kraftschluss und Formschluss:

### *Kraftschluss:*

Die Kinematik der Hand kann aktiv Kräfte erzeugen, um einer externen Störung zu widerstehen.

### *Formschluss:*

Die Kontakte an sich verhindern, dass sich das Objekt bewegen kann.

5. Unterteilungsverfahren bei der Griffsynthese auf Teilobjekten:

Formprimitive (shape primitives), Unterteilung in Quader (box decomposition), Superquadriken, Unterteilung mit der Medialen-Achsen-Transformation, Unterteilung anhand der Oberflächennormalen

## Aufgabe 5 *Bildverarbeitung*

1. Mathematischer Operator zu  $P_x$ :

Die Filtermatrix approximiert den Gradient in x Richtung:

$$P_x = \frac{\partial g(x, y)}{\partial x}.$$

2. Ergebnis der Mittelwert-Filterung:

$$B' = \begin{pmatrix} 2 & 2\frac{2}{3} & 3\frac{1}{3} & 4 \\ 2 & 2\frac{2}{3} & 3\frac{1}{3} & 4 \end{pmatrix}$$

3. Objekthöhe im Lochkamera-Modell: Aus dem Lochkameramodell folgt die folgende Beziehung:

$$v = \frac{f}{z}y \Rightarrow y = \frac{z}{f}v$$

Für die konkrete Problemstellung ergibt sich daraus:

$$h = 0,5mm \frac{100m}{20mm} = \frac{100}{40}m = 2,5m$$

4. Morphologische Operatoren:

- **Dilatation:** Dilatation erweitert einzelne Pixel bzw. Pixelgruppen zu größeren Bereichen
- **Erosion:** Erosion entfernt vereinzelte Pixel und schwach zusammenhängende Pixelgruppen

## Aufgabe 6    *Symbolisches Planen*

1. Aktionssequenz:

```
move(R, S, T)
pickup(R, P, G, T)
move(R, T, S)
putdown(R, P, G, S)
```

2. Neuer Aktionsoperator:

```
moveAndPickup(A, O, H, L, M)
  Preconditions: Agent(A), Hand(H), Location(L), Location(LM),
               At(A, L), Empty(H), On(O, M)
  Effects:      !Empty(H), !At(A, L), At(A, M), !On(O, M), InHand(O, H)
```

3. Weltzustand nach `moveAndPickup(R, P, G, S, T)`:

```
Table(T), Stove(S), Location(T), Location(S),
Gripper(G), Hand(G), Robot(R), Agent(R), Pan(P),
InHand(P, G), At(A, T)
```