

# Seminararbeit

## Verbesserung der Pfad- und Trajektorienplanung am UR5

Vorgelegt von:	Nik Julin Nowoczyn
Matrikelnummer:	8529776
Studiengang:	Ingenieurinformatik
Prüfer:	Prof. Dr.-Ing. Johannes Schilp
Betreuer:	Ludwig Vogt
Ausgabedatum:	01. 10. 2022
Abgabedatum:	31. 03. 2023



# Aufgabenstellung



# Kurzfassung



# Inhaltsverzeichnis

<b>Aufgabenstellung</b>	<b>I</b>
<b>Kurzfassung</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>1 Räumliche Beschreibung (optional ??)</b>	<b>1</b>
1.1 Koordinatensysteme . . . . .	1
1.2 Translation . . . . .	1
1.3 Rotation . . . . .	1
1.4 Homogene Transformationsmatrix . . . . .	1
<b>2 Direkte Kinematik</b>	<b>3</b>
2.1 DH-Konvention . . . . .	3
2.2 URDF . . . . .	5
2.3 Robotics API . . . . .	5
2.4 UR5 in DH . . . . .	5
<b>3 Inverse Kinematik</b>	<b>7</b>
3.1 Problembeschreibung . . . . .	7
3.2 Analytische Lösung . . . . .	7
3.3 Numerische Lösung . . . . .	7
3.4 Geometrische Lösung . . . . .	7
3.5 Singularitäten . . . . .	7
3.6 Geschwindigkeitskinematik . . . . .	7
<b>4 Pfadplanung</b>	<b>9</b>
4.1 Formale Beschreibung . . . . .	9
4.2 Konfigurationsraum . . . . .	9
4.3 Berechnungsmethoden . . . . .	9
4.3.1 Kürzester Weg . . . . .	9
4.3.2 Zellendekomposition . . . . .	9
4.3.3 Sampling-Verfahren . . . . .	9

4.4	Constraints und Praxisbezug . . . . .	9
<b>5</b>	<b>Trajektorienplanung</b>	<b>11</b>
5.1	Profile . . . . .	11
5.2	Synchronität . . . . .	11
5.3	Mehrsegment-Trajektorien . . . . .	11
<b>6</b>	<b>Fazit und Ausblick</b>	<b>13</b>



# Abbildungsverzeichnis

- 2.1 Denavit-Hartenberg-Konvention (DH-Konvention) zwischen zwei Gelenken ?? <https://de.wikipedia.org/wiki/Datei:Denavit-Hartenberg-Transformation.svg> 5



# Tabellenverzeichnis



# Abkürzungsverzeichnis

**DH-Konvention** Denavit-Hartenberg-Konvention



# 1 Räumliche Beschreibung (optional ??)

ggf kürzen und/oder am Ende ergänzen ??

## 1.1 Koordinatensysteme

## 1.2 Translation

## 1.3 Rotation

Euler / current frame, Fixed Frame, Axis-Angle, Quaternion

## 1.4 Homogene Transformationsmatrix





## 2 Direkte Kinematik

Die direkte Kinematik ist dafür verantwortlich aus den verschiedenen Winkeln und Positionen der Gelenke die Rotation und Position des Endeffektors im Raum zu berechnen. Dazu wird zunächst in jedem Gelenk ein Koordinatenursprung gelegt, der eine Nullstellung jedes Gelenks beschreibt. Um alle Gelenke in einer kinematischen Kette abzubilden, kann dann beispielsweise mithilfe der homogenen Transformationsmatrix (Abschnitt 1.4) und einem Parameter in den Freiheitsgraden des entsprechenden Gelenks eine Rechenvorschrift aufgebaut werden, um den Roboter zu beschreiben und die Position des Endeffektors schnell bestimmen zu können. (?? Quelle)

### 2.1 DH-Konvention

(?? Quelle)

Die Konvention, die in der Regel verwendet wird, um Rotation und Translation eines Gliedes der kinematischen Kette darzustellen ist die sog. DH-Konvention oder DH-Transformation. DH-Konvention beschreibt, wie die Koordinatensysteme basieren auf dem vorherigen Koordinatensystem beschrieben werden. Um nun ausgehend von Gelenk  $n - 1$  das Koordinatensystem von Gelenk  $n$  zu beschreiben, müssen die folgenden Regeln befolgt werden:

1. Achse  $z_n$  liegt entlang der Bewegungsachse von Gelenk des Gelenks
2. Achse  $x_n$  liegt auf der kürzesten Verbindung zwischen Achsen  $z_{n-1}$  und  $z_n$ .
3. Die  $y_n$  Achse wird rechtshändig ergänzt.

Dabei sind die Ursprünge der Gelenkkordinatensysteme oftmals nicht im Gelenkursprung, was Komplexität für die Berechnung von Transformationen verringert. Aus der Beziehung der zwei Koordinatensysteme können die DH-Parameter abgeleitet werden (siehe auch Abbildung 2.1):

- $\theta_n$  Winkel zwischen  $x_{n-1}$  und  $x_n$  mit Rotationsachse  $z_{n-1}$
- $d_n$ : Kleinster Abstand zwischen  $x_{n-1}$  und  $x_n$
- $a_n$ : Abstand zwischen den Achsen  $z_{n-1}$  und  $z_n$

- $\alpha_n$  Winkel zwischen  $z_{n-1}$  und  $z_n$  mit Rotationsachse  $x_n$

Dies entspricht den folgenden Transformationsmatrizen (Gleichungen 2.1, 2.1, 2.1, 2.1):

$$T_{\theta_n} = \begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & 0 \\ \sin(\theta_n) & \cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{d_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{a_n} = \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{\alpha_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_n) & -\sin(\alpha_n) & 0 \\ 0 & \sin(\alpha_n) & \cos(\alpha_n) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Um nun Koordinatensystem  $n - 1$  in Koordinatensystem  $n$  zu überführen, kann die

Transformationsmatrix  $T_{n-1,n}$  verwendet werden (Gleichung 2.1).

$$T_{n-1,n} = T_{\theta_n} \cdot T_{d_n} \cdot T_{a_n} \cdot T_{\alpha_n} = \begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) \cos(\alpha_n) & \sin(\theta_n) \sin(\alpha_n) & a_n \cos(\theta_n) \\ \sin(\theta_n) & \cos(\theta_n) \cos(\alpha_n) & -\sin(\theta_n) \sin(\alpha_n) & a_n \sin(\theta_n) \\ 0 & \sin(\alpha_n) & \cos(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

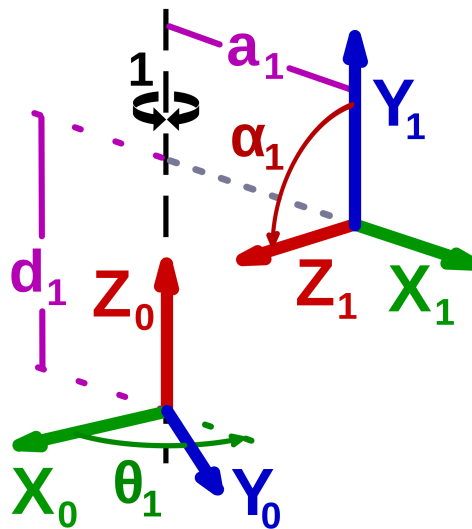


Abbildung 2.1: DH-Konvention zwischen zwei Gelenken ??  
<https://de.wikipedia.org/wiki/Datei:Denavit-Hartenberg-Transformation.svg>

## 2.2 URDF

## 2.3 Robotics API

## 2.4 UR5 in DH



## 3 Inverse Kinematik

### 3.1 Problembeschreibung

### 3.2 Analytische Lösung

### 3.3 Numerische Lösung

### 3.4 Geometrische Lösung

Handgelenk entkoppeln, nicht anwendbar auf UR5

### 3.5 Singularitäten

Kuka vs UR5.

Theorie (rundungsfehler), Praxis (große Geschwindigkeiten)  $\alpha_2$   $\alpha_1$   $\alpha_5$

Elbow-Up / Elbow-Down

### 3.6 Geschwindigkeitskinematik



## 4 Pfadplanung

### 4.1 Formale Beschreibung

### 4.2 Konfigurationsraum

Konfigurationsraum und Kollisionsfreier Konfigurationsraum

### 4.3 Berechnungsmethoden

#### 4.3.1 Kürzester Weg

#### 4.3.2 Zellendekomposition

Zellendekomposition (Vollständige Abtastung)

#### 4.3.3 Sampling-Verfahren

Single-Query Unidirektional vs. Bidirektional RRT biased / unbiased to exploration  
Q-space muss nicht vollständig bekannt sein RRT\* Multi-Query Probabilistic Roadmaps  
Potentialfeldmethode / Gradientenverfahren Genetische Algorithmen

### 4.4 Constraints und Praxisbezug

Kinematisch (Winkelbegrenzung) Dynamisch (Geschwindigkeit) Einbezug der Constraints in den Algorithmen Praxis: OMPL (MoveIt+ ROS / CoppeliaSim Plugin)





## **5 Trajektorienplanung**

### **5.1 Profile**

Trapez, 7-Segment

### **5.2 Synchronität**

Vollsynchon, Teilsynchon, Asynchon

### **5.3 Mehrsegment-Trajektorien**

Ggf besser in Pfadplanung ??

Mehrsegment-Trajektorien (z.B. Bezier, Überschleifen)



## **6 Fazit und Ausblick**



# Erklärung

Die vorliegende Arbeit habe ich selbstständig ohne Benutzung anderer als der angegebenen Quellen angefertigt. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer oder anderer Prüfungen noch nicht vorgelegt worden.

Augsburg, den 31.03.2023

Nik Julin Nowoczyn