

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

Aufgabenstellung	I
Kurzfassung	III
Abbildungsverzeichnis	VII
Tabellenverzeichnis	IX
1 Räumliche Beschreibung (optional ??)	1
1.1 Koordinatensysteme	1
1.2 Translation	1
1.3 Rotation	1
1.4 Homogene Transformationsmatrix	1
2 Direkte Kinematik	3
2.1 DH-Konvention	3
2.2 Unified Robot Description Format	5
2.3 Beschreibung des UR5e	6
3 Inverse Kinematik	9
3.1 Problembeschreibung	9
3.2 Analytische Lösung	9
3.3 Numerische Lösung	9
3.4 Geometrische Lösung	9
3.5 Singularitäten	9
3.6 Geschwindigkeitskinematik	9
4 Pfadplanung	11
4.1 Formale Beschreibung	11
4.2 Konfigurationsraum	11
4.3 Berechnungsmethoden	11
4.3.1 Kürzester Weg	11
4.3.2 Zellendekomposition	11
4.3.3 Sampling-Verfahren	11
4.4 Constraints und Praxisbezug	11

5	Trajektorienplanung	13
5.1	Profile	13
5.2	Synchronität	13
5.3	Mehrsegment-Trajektorien	13
6	Fazit und Ausblick	15

Abbildungsverzeichnis

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

Tabellenverzeichnis

2.1	DH-Parameter des UR5e Roboters von Universal Robots	6
2.2	Beschreibung der dynamischen Eigenschaften des UR5e Roboters von Universal Robots	7

Abkürzungsverzeichnis

DH-Konvention Denavit-Hartenberg-Konvention

URDF Unified Robot Description Format

ROS das Robot Operating System

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

?? Quelle

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.

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):

- θ_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
- α_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).

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

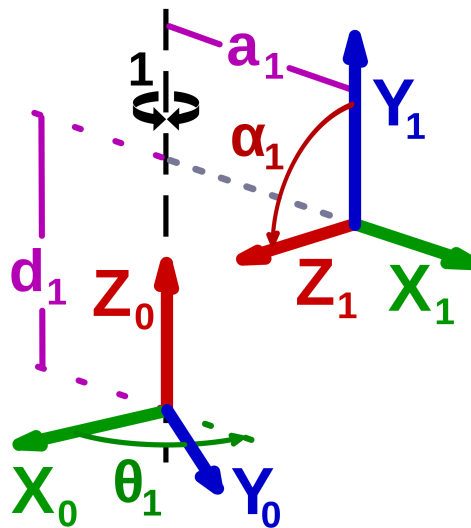


Abbildung 2.1: DH-Konvention zwischen zwei Gelenken ??
<https://de.wikipedia.org/wiki/Datei:Denavit-Hartenberg-Transformation.svg>

2.2 Unified Robot Description Format

?? Quelle Unified Robot Description Format (URDF) ist ein Standard entwickelt für das Robot Operating System (ROS), um sowohl die geometrischen als auch physische und visuelle Eigenschaften eines Roboters zu beschreiben. Für die Beschreibung ist eine auf XML basierende Datei nötig, die mit Tags geschachtelt den Roboter mithilfe von sog. „links“ und „joints“ beschreibt.

Links beschreiben die physischen Verbindungen zwischen zwei Gelenken. Dabei wird unterschieden zwischen visuellen Eigenschaften („visual“), Kollisionseigenschaften („col-

Tabelle 2.1: DH-Parameter des UR5e Roboters von Universal Robots

Joint	θ [rad]	a [m]	d [m]	α [rad]
Joint 1	0	0	0.1625	$\pi/2$
Joint 2	0	-0.425	0	0
Joint 3	0	-0.3922	0	0
Joint 4	0	0	0.1333	$\pi/2$
Joint 5	0	0	0.0997	$-\pi/2$
Joint 6	0	0	0.0996	0

lision“) und Trägheitseigenschaften („inertial“). Jede dieser Kategorien kann so die Geometrie des Links aus verschiedenen Perspektiven unterschiedlich beschreiben Joints bezeichnen Gelenke, also die Verbindungen zweier Links und beschreiben die mögliche relative Bewegung zwischen diesen. Dabei sind nicht nur translatorische („prismatic“) und rotatorische Gelenke („revolute“) beschreibbar, sondern auch feste („fixed“), schwebende („floating“) und planare Verbindungen („planar“). Für die Beschreibung eines Joints wird der vorhergehende Link als „parent“ und der nächste Link als „child“ bezeichnet. Zudem müssen im Feld „origin“ Translation und Rotation des Ursprungs im Koordinatensystem des Parent Links, sowie im Feld „axis“ je nach Gelenk die Rotationsachse, Translationsachse oder Normale der Bewegungsoberfläche angegeben werden. Desweiteren ist es möglich, Schnittstellen für die Bewegung der Motoren zu definieren und Bewegungslimits für die Gelenke anzugeben, um die Ansteuerung des Roboters und die Bewegungsplanung zu vereinfachen.

2.3 Beschreibung des UR5e

?? Quelle <https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/>

Der in dieser Arbeit betrachtete Roboter ist der UR5e von Universal Robots. Dieser ist ein vergleichsweise günstiger Roboter mit verringerter Zahl an Singularitäten (siehe Abschnitt 3.5). Offiziell wird der Roboter mit den DH-Parametern aus Tabelle 2.1 und dynamischen Eigenschaften der Links aus Tabelle 2.2 beschrieben.

?? verweis auf anhang

?? Bild ergänzen

Tabelle 2.2: Beschreibung der dynamischen Eigenschaften des UR5e Roboters von Universal Robots

Link	Mass [kg]	Center of Mass [m]
Link 1	3.761	[0, -0.02561, 0.00193]
Link 2	8.058	[0.2125, 0, 0.11336]
Link 3	2.846	[0.15, 0.0, 0.0265]
Link 4	1.37	[0, -0.0018, 0.01634]
Link 5	1.3	[0, 0.0018, 0.01634]
Link 6	0.365	[0, 0, -0.001159]

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) α_2 α_1 α_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