

Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und
Mathematik
Studiengang Ingenieurinformatik

REPRÄSENTATION VON
TRANSFORMATIONSKETTEN ZUR
VISUALISIERUNG UNIVERSELLER
ROBOTERKINEMATIKEN

Ben Raeke 1220577,
Sascha Nickel 1205844

23. Juni 2022

Betreuer:
Prof. Dr. Axel Schneider
Matthäus Engel

Für die Programmierung eines Robotersystems und zur Überprüfung dieser, wird eine adäquate Simulation dieses Systems benötigt. Speziell für die Bewegung von Aktoren müssen mathematische Modelle aufgestellt werden, welche das System abstrakt als kinematische Kette darstellen. Die kinematische Kette kann anschließend für Berechnungen der Vorwärtskinematik und der inversen Kinematik verwendet werden. Je nach Komplexität und der gegebenen Anforderungen an das dargestellte Robotersystem, variiert auch die Komplexität der kinematischen Kette und der enthaltenden Informationen. Relative Beziehungen von Gelenken in einer kinematischen Kette, also der Abstand und die Verdrehung zweier Gelenke zueinander, können in einer einheitlichen Transformationsmatrix mit Parametern abgebildet werden. Der strukturelle Aufbau der Matrix ist dabei immer identisch. Durch Verknüpfungen verschiedener Transformationsmatrizen können diese serialisiert werden, um ganze Pfade in der kinematischen Kette in einer einzelnen Transformationsmatrix darzustellen. Das projektbegleitende Programm soll eben diese relativen Beziehungen von Gelenken eines Robotersystems und die resultierende Struktur in Transformationsmatrizen umformen und als mathematisches Äquivalent der kinematischen Kette grafisch ausgeben. Des Weiteren sollen mithilfe der Vorwärtskinematik Änderungen an den Gelenken vorgenommen werden können, sodass eine Bewegung des Robotersystems dargestellt werden kann. Allgemein soll das Projekt die Überprüfung der Korrektheit einer Implementierung auf einem realen System digital ermöglichen und für Lehrzwecke relevante Konventionen übermitteln können.

For the programming of a robot system and its verification, an adequate simulation of this system is required. Especially for the movement of actors, mathematical models must be set up that abstractly represent the system as a kinematic chain. The kinematic chain can then be used for calculations of forward kinematics and inverse kinematics. Depending on the complexity and the given requirements of the represented robot system, the complexity of the kinematic chain and the information it contains also varies. Relative relationships of joints in a kinematic chain, i.e. the distance and rotation of two joints to each other, can be mapped in a uniform transformation matrix with parameters. The structural design of the matrix is always identical. By linking different transformation matrices, these can be serialised to represent entire paths in the kinematic chain in a single transformation matrix. The program accompanying the project should transform these relative relationships of joints of a robot system and the resulting structure into suitable transformation matrices and output them graphically as the mathematical equivalent of the kinematic chain. Furthermore, with the help of forward kinematics, it should be possible to make changes to the joints so that a movement of the robot system can be represented. In general, the project should make it possible to digitally check the correctness of an implementation on a real system and to transmit relevant conventions for teaching purposes.

INHALTSVERZEICHNIS

1. EINLEITUNG	2
2. EINFÜHRUNG IN DIE ROBOTERKINEMATIK	4
2.1. Überführung eines mechatronischen Systems anhand einer kinematischen Kette in einen mathematischen Ausdruck	4
2.2. Denavit-Hartenberg-Konvention	11
2.3. Besondere Arten von kinematischen Ketten	13
3. DARSTELLUNG VON BELIEBIGEN ROBOTERKINEMATIKEN	15
3.1. Allgemeine Anforderungen an die Klassen Robot und Joint	15
3.2. Verknüpfung der Gelenke in einer kinematischen Kette durch eine Baumstruktur	17
3.3. Abbildung eines Robotersystems	20
4. DISKUSSION UND AUSBLICK	22
QUELLENVERZEICHNIS	23
A. ANHANG	24
A.1. Überführen eines Robotersystems in das Programm	24
A.1.1. Grundaufbau der JSON-Datei	24
A.1.2. Serielle Kinematik aus drei Gelenken	26
A.1.3. Serielle Kinematik mit parallel verlaufenden Abzweigungen	28
A.1.4. Geschlossene kinematische Kette	30

1. EINLEITUNG

Die vorliegende Arbeit befasst sich mit der Visualisierung von kinematischen Ketten, welche Teile von realen Robotersystemen in einer abstrakten Form abbilden können. Das begleitende Projekt dieser Arbeit ist ein interaktiver dreidimensionaler Plotter, welcher die Berechnungen und die Methoden dieser theoretischen Ausarbeitung umsetzt. Das Programm soll als Grundlage für Überprüfungen von simplen Programmabläufen, wie der Vorwärtskinematik, auf realen Robotersystemen dienen können, beziehungsweise diese für Lehrzwecke veranschaulichen. Eine beispielhafte Darstellung der Version 1.0 des Programms ist in Abb. 1 gegeben. Im ersten Kapitel werden notwendige Grundlagen der Robotik vermittelt, dabei wird erläutert, wie Aktoren eines Robotersystems auf zwei Grundbestandteile reduziert werden: Gelenke und Zwischensegmente. Aus diesen werden kinematische Ketten zusammengesetzt, wobei je nach Komplexität und der gegebenen Anforderungen an das dargestellte Robotersystem auch die Komplexität der kinematischen Kette und der enthaltenden Informationen, variiert. Dabei gelten die Anzahl von Gelenken und die verwendeten Arten von kinematischen Ketten als Grundlage der Komplexität des Systems. Die Informationen der in dieser Arbeit verwendeten kinematischen Ketten belaufen sich nur auf die Positionierung und Orientierung von Gelenken. Es kann anschließend keine physikalische Simulation stattfinden, lediglich eine Visualisierung von eben jenen grundlegenden Daten findet statt. Diese Strukturen können jedoch als Grundlage für weiterführende Projekte und Arbeiten dienen, ausführliche Diskussionen und Bewertungen darüber werden im Ausblick geführt. Aufbauend auf die kinematischen Ketten, wird ein mathematisches Modell vorgestellt, wobei die Gelenke durch Koordinatensysteme ersetzt werden. Die realen Beziehungen, also die Verdrehung und Verschiebung zueinander zweier solcher Gelenke, wird in Form von Transformationsmatrizen festgehalten. Diese 4×4 Matrizen haben eine feste Struktur. Durch Variation der Parameter wird die resultierende Transformationsmatrix verändert. In dieser Arbeit wird speziell die Denavit-Hartenberg Konvention genutzt, wodurch die benötigte Parameterzahl von ursprünglich sechs, auf vier notwendige Parameter reduziert wird. Einmal für das ganze Robotersystem erzeugt, können die Transformationsmatrizen genutzt werden, um relative Änderungen von den Gelenken der kinematischen Kette in Bezug auf ein globales Basiskoordinatensystem zu berechnen. Dabei wird in dieser Arbeit das Vorgehen der Vorwärtskinematik behandelt, wobei Gelenkdrehungen und Vorschübe zu einer Positionierung des Endpunktes von einem Aktor führt. Die Berechnung von Gelenkkonfigurationen die zu einem bestimmten Zielpunkt des Endpunktes von einem Aktor führen, die sogenannte inverse Kinematik, wird hier nicht behandelt. Wie die Vorwärtskinematik objektorientiert für das begleitende Projekt umgesetzt wird, um Veränderungen in der kinematischen Kette zu visualisieren und die Zusammenhänge der Gelenkkonfigurationsparameter mit dem abgebildeten System darzustellen, wird in Kapitel 3 dieser Arbeit erläutert. Besonders eine Erweiterung der Konstruktion von verschiedenen Arten von kinematischen Ketten wird in jenem Kapitel näher betrachtet. Abschließend lassen sich im Anhang Beispiele der direkten Anwendung der erzeugten Software finden.

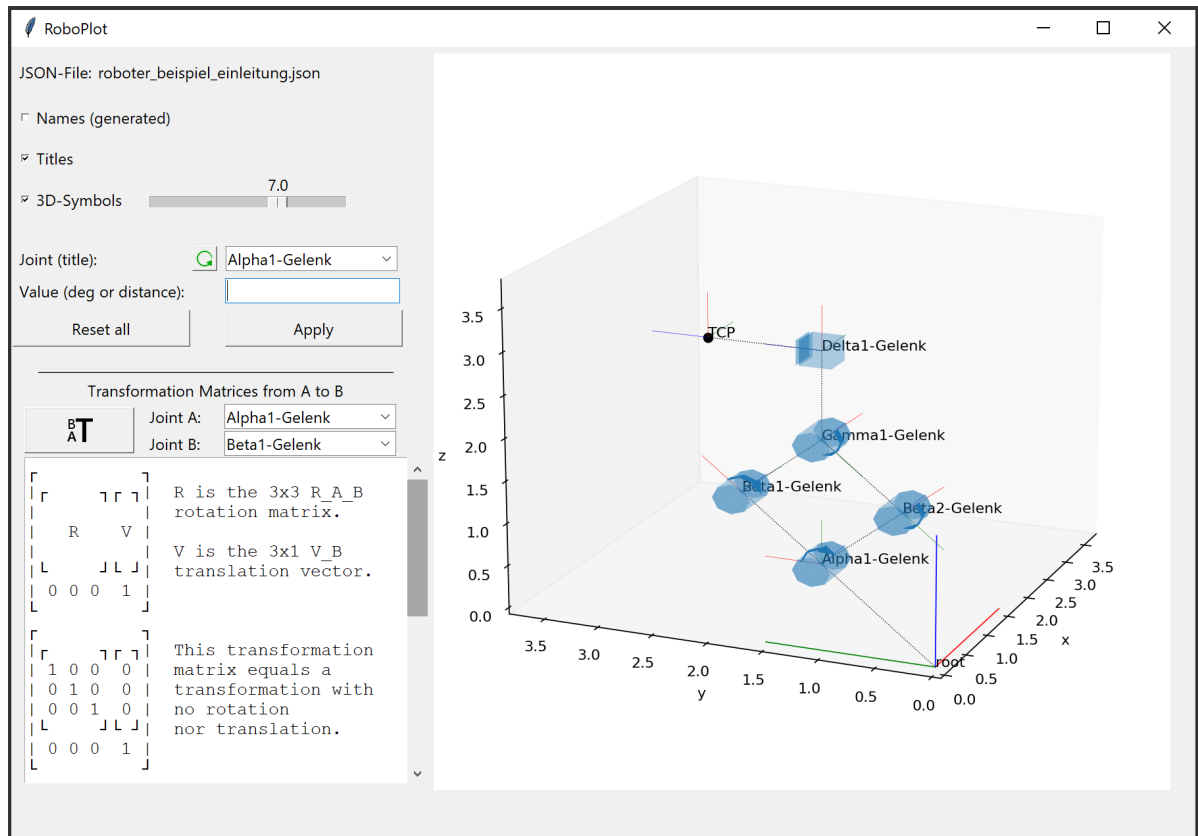


Abbildung 1.: Vorstellung des projektbegleitenden Programms in der Version 1.0 mit der Darstellung der kinematischen Kette eines Robotersystems (rechts). In dem Menü auf der linken Seite können verschiedene Kontrollkästchen für eine Ein- bzw. Ausblendung verschiedener Funktionen umgeschaltet werden. Zudem können für selektierte Gelenke die Gelenkkonfigurationen (Winkel oder Vorschub) gesetzt werden. Diese Änderungen werden nach einer Bestätigung im Plotter übernommen. Ferner kann das Programm Transformationsmatrizen zwischen zwei ausgewählten Gelenken berechnen. Diese Abbildung dient als Ausblick für verschiedene Aspekte und Begriffe, welche in der vorliegenden Ausarbeitung im Detail besprochen werden.

2. EINFÜHRUNG IN DIE ROBOTERKINEMATIK

Roboter, Manipulatoren und ähnliche mechatronische Systeme, im Folgenden zusammengefasst als Robotersysteme bezeichnet, besitzen Aktoren, welche für Bewegungen von Segmenten, zur Ausrichtung von Sensoren oder gar des ganzen Systems nötig sind. Diese Aktoren bestehen zumeist aus einer Aneinanderreihung von Gelenken und starren Zwischensegmenten, welche dementsprechend eine kinematische Kette bilden. Die Gelenke einer kinematischen Kette lassen Bewegungen abhängig ihrer Freiheitsgrade und daraus resultierende Relativbewegungen zwischen einzelnen Zwischensegmenten zu oder schränken sie ein [1]. Allgemein sind Berechnungen der einzelnen Gelenkkonfigurationen nötig, um gewünschte Positionen und Ausrichtungen, zum Beispiel eines Endeffektors, zu erreichen. Wie ein physikalisches Robotersystem zu kinematischen Ketten vereinfacht wird, welche Möglichkeiten für Berechnungen daraus resultieren und was Notwendig ist, um solche Systeme grafisch darzustellen, wird in den folgenden Abschnitten erläutert.

2.1. ÜBERFÜHRUNG EINES MECHATRONISCHEN SYSTEMS ANHAND EINER KINEMATISCHEN KETTE IN EINEN MATHEMATISCHEN AUSDRUCK

Für Berechnungen dieser Arbeit und des damit verbundenen Projekts sind nur wenige physikalische Eigenschaften des Robotersystems relevant. Die Eigenschaften werden auf das wesentliche reduziert. Es wird nach Gelenken und Zwischensegmenten unterschieden. Die Form, Funktionsweise und Daten wie Gewicht, Reibung, Trägheit und Ähnliches werden nicht näher untersucht, da lediglich eine Visualisierung von eingegangenen Daten stattfindet. Die Informationen zu den relativen Abständen und Orientierungen der Gelenke zueinander, sowie die Art eines Gelenks sind ausreichend um eine simple kinematische Kette des Systems abzubilden. Ziel ist es, diese relativen Beziehungen der Gelenke und der Gelenkkonfigurationen, wie exemplarisch den Rotationswinkel eines Rotationsgelenks selbst, in absolute Position mit Orientierung in einem Basiskoordinatensystem darzustellen. Ein Benutzer von industriellen Robotersystemen, wie zum Beispiel eines Industriemanipulators, kann Koordinaten zum Anfahren eines Endeffektors in diesem kartesischen Koordinatensystem angeben. Jedes Gelenk wird durch ein kartesisches Koordinatensystem ersetzt und die Zwischensegmente durch Strecken, welche jeweils zwei Gelenke miteinander verbinden, vgl. Abb. 2. Für eine genauere Abbildung kann die Form der Verbindungsstrecken grob der Form der Zwischensegmente nachgeahmt werden, dies findet in dieser Arbeit jedoch keine Anwendung.

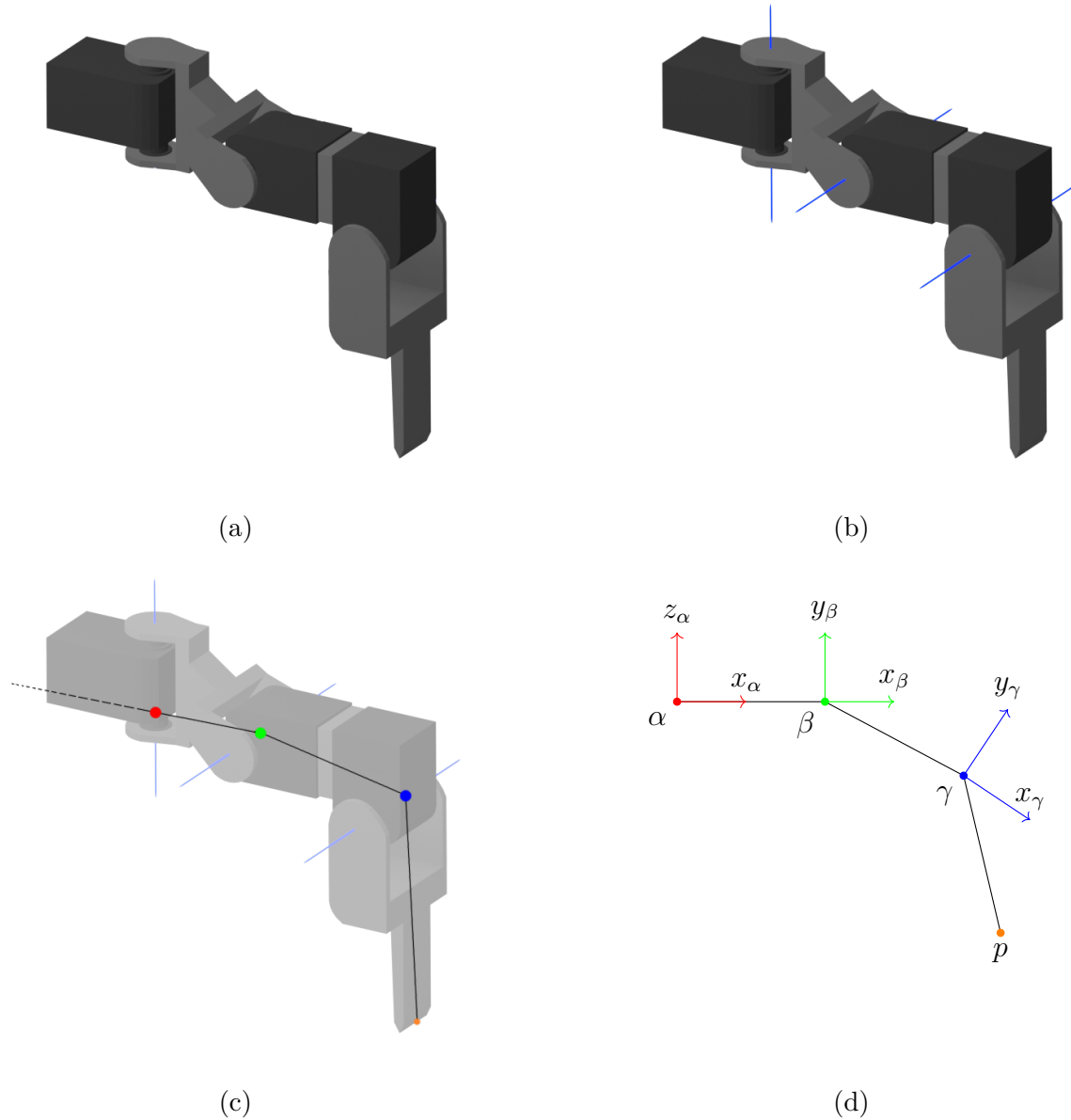


Abbildung 2.: Grafische Darstellungen eines Verlaufs von Arbeitsschritten zur Erzeugung einer kinematischen Kette aus einem Roboterbein, welches als Rendering vorliegt. In a ist das gegebene Roboterbein, mit drei Rotationsgelenken und einem Fußpunkt in Form eines dreidimensionalen Renderings abgebildet. In b wird das Rendering aus a grafisch mit Gelenkachsen erweitert, um den Freiheitsgrad der Gelenke zu visualisieren. In c werden die Gelenke (rot, grün, blau) und der Fußpunkt (orange) durch Punkte ersetzt. Die Punkte werden über Strecken verbunden. In d wird die in c erzeugte kinematische Kette skizzenhaft von der Seite mit Benennung der Gelenke (α , β , γ) erweitert, indem jedem Gelenkpunkt ein lokales Koordinatensystem zugewiesen wird. In diesem Fall wird die Denavit-Hartenberg-Konvention zur Zuweisung der Achsen genutzt.

In dem Basiskoordinatensystem werden die Gelenke als weitere kartesische Koordinatensysteme, mit Orientierung und Position abgebildet. Je nach Ansatz kann die Orientierung eines Gelenks frei gewählt werden oder wird so gewählt, dass Berechnungen an Komplexität verlieren. Welche Auswirkungen dies genau hat, wird in späteren Kapiteln näher betrachtet.

Ein allgemeines Ziel des Projekts ist es, reale Gelenke virtuell abzubilden. Diese sollen im Anschluss verschoben beziehungsweise gedreht werden können, wie bei einem realen Robotersystem. Speziell in diesem Zusammenhang werden bei der Vorwärtskinematik für die Gelenke bestimmte Konfigurationen (Drehwinkel, Vorschublänge) eingestellt, damit das letzte Glied der kinematischen Kette einen definierten Punkt im Basiskoordinatensystem erreicht. Notwendig für das Erreichen dieser Aufgabe ist zunächst ein Zwischenschritt. Relative, räumliche Beziehungen zwischen Koordinatensystemen müssen durch einen mathematischen Ausdruck beschrieben werden. Diese Transformation zum jeweils nächsten Koordinatensystem (Gelenk) in der kinematischen Kette wird in einer Transformationsmatrix festgehalten. Eine Transformationsmatrix T hat formal als linksseiteigen Index das Ursprungskoordinatensystem, von welchem aus die Transformation startet und als linksseitigen Exponenten das Zielkoordinatensystem. Also würde die Transformationsmatrix ${}^B_A T$ eine Transformation von einem Koordinatensystem A zu einem Koordinatensystem B beschreiben. Durch Multiplikation von solchen Matrizen in richtiger Reihenfolge, können ganze Ketten an Transformationen als eine Transformation zusammengefasst werden [2], das generelle Konzept wird in Abb. 3 dargestellt.

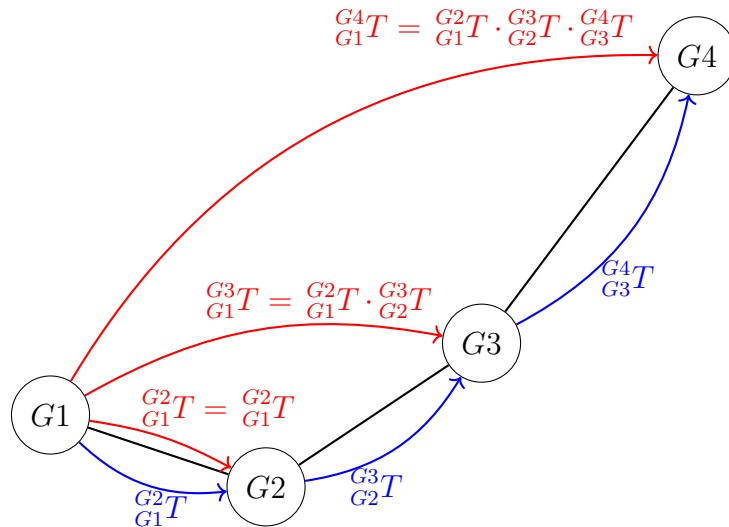
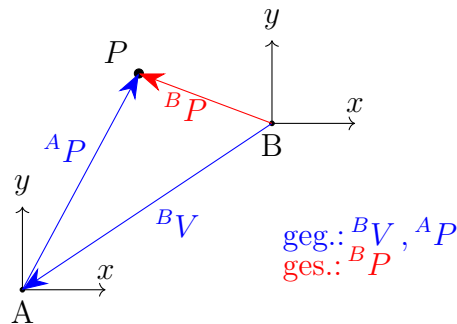


Abbildung 3.: Abgebildet ist eine kinematische Kette bestehend aus den vier Gelenken G1 bis G4. Die Transformationsmatrizen zwischen jeweils zwei Gelenken (blau) können durch Multiplikation den Pfad der kinematischen Kette vom ersten bis zum spezifizierten Gelenk bilden (rot).

Im Folgendem wird der Ansatz und Aufbau einer solchen Transformationsmatrix erläutert, mit Ausblick auf die Nutzung für später durch Koordinatensysteme ersetzte Gelenke. Der Ansatz dabei ist, den Ursprungspunkt eines Koordinatensystems mit Orientierung zu verschieben, allgemein können durch die Transformationsmatrix beliebige Punkte transformiert werden. Insgesamt werden zur Beschreibung der Position und Orientierung sechs Parameter für zwei zueinander verschobene und verdrehte kartesische Koordinatensysteme im dreidimensionalen Raum benötigt. Jeweils ein Parameter für die Rotation um eine der Achsen und jeweils einer für die Verschiebungen entlang der Achsen. Ferner werden Punkte und Vektoren zur Erläuterung dieser Thematik benötigt, welche in Koordinaten eines bestimmten Koordinatensystems gegeben beziehungsweise gesucht werden. Damit immer eindeutig bestimmbar ist, zu welchem Koordinatensystem diese jeweils relativ angegeben sind, wird hier die Konvention eingehalten, welche als linkseitigen Exponenten das zugrundeliegende Koordinatensystem angibt. Die Transformationsmatrix setzt sich aus einem Rotations- und einem Translationsanteil zusammen [2], welche zunächst unabhängig und daraufhin in Kombination untersucht werden.

Zunächst wird die Translation eines beliebigen Punktes P erläutert, wobei die zwei zugrundeliegenden Koordinatensysteme A (Ursprungs Koordinatensystem) und B (Zielkoordinatensystem) dementsprechend nicht zueinander verdreht sind. Wie in Abb. 4 angedeutet, müssen zwei Dinge gegeben sein, wenn die Koordinaten eines Punktes P in Koordinaten von einem Zielkoordinatensystem gesucht werden. Zum Einem werden der Ortsvektor des Punktes ${}^A P$ benötigt, also die Koordinaten des Punktes im Ursprungs Koordinatensystem. Zum Anderem muss der Verschiebungsvektor ${}^B V$ vorliegen, welcher von dem Ursprung von dem Koordinatensystem B zu dem Koordinatenursprung von dem Koordinatensystem A zeigt, gegeben in Koordinaten von B .



Transformation: Translation

Abbildung 4.: Grafische Darstellung bekannter und unbekannter Größen für die Transformation eines Punktes von einem Koordinatensystem zu einem anderem ohne Rotation. Abgebildet sind zwei zueinander verschobene zweidimensionale Koordinatensysteme A und B und ein Punkt P . Gesucht sind die Koordinaten ${}^B P$ des Punktes P im Zielkoordinatensystem B , bekannt ist der Ortsvektor ${}^A P$ des Punktes P im Ursprungs Koordinatensystem A und der Translationsvektor ${}^B V$, der vom Ursprung von B auf den Ursprung von A zeigt in Koordinaten von B .

Liegt, wie hier, eine reine Translation vor, gilt für diesen Vektor ${}^B V$ [3]:

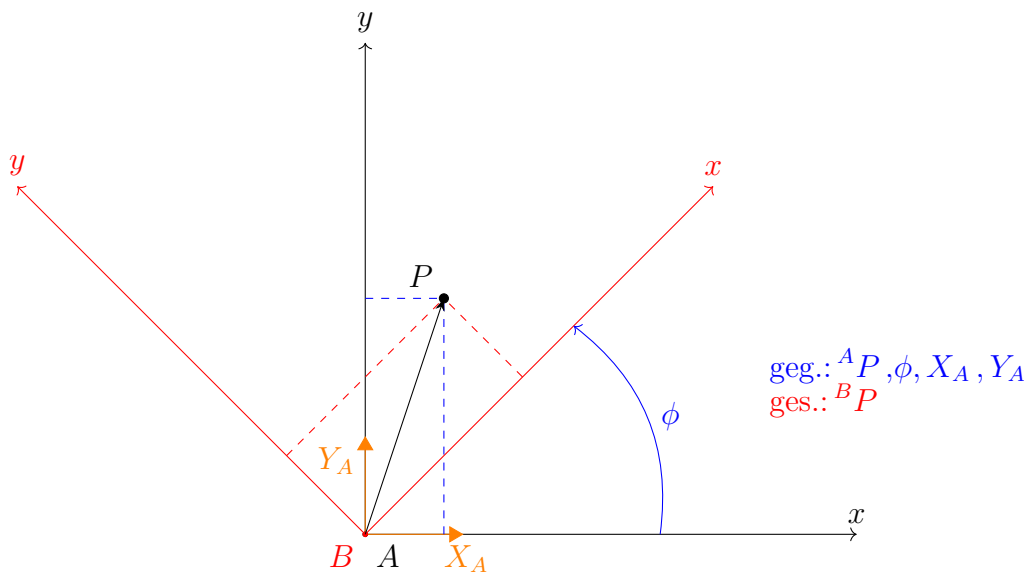
$${}^B V = {}^A -V. \quad (1)$$

Sobald jedoch ein Rotationsanteil vorhanden ist, gilt diese Gleichung (1) nicht mehr [3], deshalb wird dieser Ansatz nicht weiter verfolgt. Es gilt ferner [2]:

$${}^B P = {}^A P + {}^B V, \quad (2)$$

für die Transformation ohne Rotation.

Im Weiteren wird die Transformation mit Rotation, aber ohne Translation der Koordinatensysteme dargestellt. Begleitendes Beispiel ist dafür Abb. 5.



Transformation: Rotation

Abbildung 5.: Grafische Darstellung bekannter und unbekannter Größen für die Transformation mit Rotation eines Punktes von einem Koordinatensystem zu einem anderen ohne Translation. Abgebildet sind zwei zueinander rotierte zweidimensionale Koordinatensysteme A und B, ein Punkt P mit zugehörigem Ortsvektor, der Winkel ϕ , um den beide Koordinatensysteme zueinander verdreht sind und die Einheitsvektoren des Ursprungskoordinatensystems A X_A und Y_A . Gesucht ist der Ortsvektor ${}^B P$ des Punktes P im Zielkoordinatensystem B. Bekannt sind der Ortsvektor des Punktes P in Koordinaten von A, der Winkel ϕ und die Einheitsvektoren X_A und Y_A in Koordinaten von A.

Zunächst werden geltende Bedingungen aufgezeigt. Der Winkel ϕ beschreibt den Winkel vom Ursprungskoordinatensystem zum Zielkoordinatensystem und hat im mathematischen Sinne eine positive Drehrichtung gegen den Uhrzeigersinn. X_A und Y_A seien die Einheitsvektoren des Ursprungskoordinatensystems A . Für Berechnungen werden genau diese Einheitsvektoren auch in Koordinaten von B gesucht, genannt ${}^B X_A$ und ${}^B Y_A$. Die gesuchte Rotationsmatrix ${}^B R_A$ verknüpft diese Größen, es gilt [3]:

$${}^B X_A = {}^B R \cdot X_A \quad (3)$$

$${}^B Y_A = {}^B R \cdot Y_A \quad (4)$$

Ferner werden die gegebenen Einheitsvektoren durch Zahlenwerte ersetzt:

$${}^B X_A = {}^B R \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5)$$

$${}^B Y_A = {}^B R \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (6)$$

Die Rotationsmatrix ${}^B R$ ergibt sich direkt aus den Einheitsvektoren ${}^B X_A$ und ${}^B Y_A$ [3]:

$${}^B R = \begin{bmatrix} {}^B X_A & {}^B Y_A \end{bmatrix} \quad (7)$$

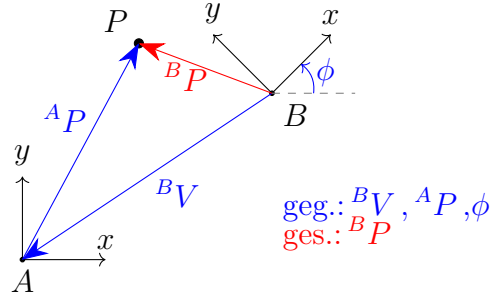
Aus dem geometrischen Zusammenhang in Abb. 5 ergibt sich dafür:

$${}^B R = \begin{bmatrix} {}^B X_A & {}^B Y_A \end{bmatrix} = \begin{pmatrix} \cos(-\phi) & -\sin(-\phi) \\ \sin(-\phi) & \cos(-\phi) \end{pmatrix} = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (8)$$

Bei der Kombination beider Transformationen, vgl. Abb. 6, ist das Ziel, wie Eingangs beschrieben, ein Operand in Matrixform zu berechnen. Diese soll mit dem gegebenen Punkt multipliziert werden können, sodass als Ergebnis dieser Rechnung der gesuchte Punkt berechnet wird. Es wird eine 4×4 Matrix, für dreidimensionale Probleme definiert, welche als homogene Transformationsmatrix bezeichnet wird [2]. So wird bei einem Problem auf einer Ebene (\mathbb{R}^2), wie in den gegebenen Beispielen, die Dimension von zwei auf drei angehoben - analog wird ein Problem im Raum (\mathbb{R}^3) auf eine vierte Dimension gehoben. Ferner können durch diese Erweiterung perspektivische Transformationen und Skalierungen durchgeführt werden [4, 2]. In diesem Zusammenhang der vorliegenden Arbeit gilt diese Erweiterung als Notwendigkeit für die Kombination der allgemeinen Rotation und Translation in einer einzelnen Matrix [2]. Gleichung (2) und Gleichung (8) werden verknüpft [3, 2]:

$${}^B P = {}^B R \cdot {}^A P + {}^B V \quad (9)$$

$${}^B P = \begin{bmatrix} {}^B X_A & {}^B Y_A \end{bmatrix} \cdot {}^A P + {}^B V \quad (10)$$



Transformation: Translation & Rotation

Abbildung 6.: Grafische Darstellung bekannter und unbekannter Größen für die Transformation mit Rotation und Translation eines Punktes von einem Koordinatensystem zu einem anderen. Abgebildet sind zwei zueinander verschobene und verdrehte zweidimensionale Koordinatensysteme A und B , ein Punkt P mit zugehörigem Ortsvektor und der Winkel ϕ um die beide Koordinatensysteme zueinander verdreht sind. Gesucht sind der Ortsvektor ${}^B P$ des Punktes P im Zielkoordinatensystem B . Bekannt sind der Ortsvektor ${}^A P$ des Punktes P im Ursprungskordinatensystem A , der Translationsvektor ${}^B V$, der vom Ursprung von B auf den Ursprung von A zeigt in Koordinaten von B und der Winkel ϕ .

Zur Veranschaulichung folgt ein Beispiel in der Ebene mit der zusammengesetzten Transformationsmatrix:

$$\begin{aligned} \begin{pmatrix} {}^B P_x \\ {}^B P_y \\ 1 \end{pmatrix} &= \begin{bmatrix} {}^B R & {}^B V \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} {}^A P_x \\ {}^A P_y \\ 1 \end{pmatrix} \\ &= \begin{bmatrix} \cos(\phi) & \sin(\phi) & {}^B V_x \\ -\sin(\phi) & \cos(\phi) & {}^B V_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} {}^A P_x \\ {}^A P_y \\ 1 \end{pmatrix} \end{aligned} \quad (11)$$

Verschiedene Transformationsmatrizen, folglich immer als homogen zu deuten, können nun miteinander verkettet werden, vgl. Abb. 3. Die Transformationsmatrizen sind immer gleich aufgebaut und können je nach Dimensionierung des Problems erweitert werden [3, 2]:

$${}^B_A T = \left[\begin{array}{ccc|c} & ({}^B_A R) & & {}^B V \\ \hline - & - & - & - \\ 0 & \dots & 0 & 1 \end{array} \right] \quad (12)$$

Für 2-D Transformationen gilt:

$${}^B_A T^{(2)} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & {}^B V_x \\ -\sin(\phi) & \cos(\phi) & {}^B V_y \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Für dreidimensionale Transformationen gelten andere Rotationsmatrizen, je nachdem, um welche der drei Achsen der Punkt rotiert wird [5, 6]:

$${}^B_A R_{(x,\alpha)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (14)$$

$${}^B_A R_{(y,\beta)} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \quad (15)$$

$${}^B_A R_{(z,\gamma)} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (16)$$

Diese können aus gegebenen Gründen ebenfalls verkettet werden. Die in dieser Arbeit verwendeten Gelenke haben jedoch nur einen Freiheitsgrad und können genau entlang oder um eine Achse verschoben beziehungsweise rotiert werden. Dementsprechend wird hier auf die Kombination mehrerer Rotationsmatrizen verzichtet. In der Robotik bietet sich der Ansatz von Denavit-Hartenberg an. Wie dies genau umgesetzt wird und warum die Komplexität der Rechnungen dadurch reduziert wird, soll in Abschnitt 2.2 näher untersucht werden.

2.2. DENAVIT-HARTENBERG-KONVENTION

Wie im letzten Abschnitt angedeutet, wird in der Robotik im Zusammenhang mit Transformationsmatrizen häufig die Denavit-Hartenberg-Konvention (DH-Konvention) genutzt. Im Allgemeinen gilt es hierbei die Anzahl der Transformationsparameter zu reduzieren. Dies geschieht in dem Arbeitsschritt, bei dem die Gelenke der kinematischen Kette durch das mathematische Modell mit Koordinatensystemen ersetzt werden. Hier können die Koordinatensysteme beliebig positioniert und rotiert werden, dabei verändern sich dementsprechend auch die Parameter für folgende Gelenke in der kinematischen Kette. Dabei kann es zu unnötig komplizierten Parameterwerten kommen. Um dies möglichst simpel zu halten, bietet es sich deshalb an, die Koordinatensysteme entlang der Rotationsachse des Gelenks zu positionieren. Die DH-Konvention erweitert dieses Vorgehen mit festen Regeln, wobei die Gelenke vom Anfang der kinematischen Kette bis zum Ende nacheinander nummeriert ersetzt werden [2]:

1. Die z-Achse eines Koordinatensystems muss auf der Rotations-/Translationsachse des Gelenks liegen.
2. Die x-Achse eines Koordinatensystems muss senkrecht zur z-Achse des vorherigen Koordinatensystems der kinematischen Kette liegen.
3. Die x-Achse eines Koordinatensystems muss die z-Achse des vorherigen Koordinatensystems in der kinematischen Kette schneiden.

Damit diese Regeln eingehalten werden können, müssen die Koordinatensysteme nicht zwingend an der gleichen Position wie die zu ersetzenden Gelenke liegen, solange die Verschiebungen in folgenden Gelenken ausgeglichen werden [3]. Werden jene Regeln für alle Ersetzungen der Gelenke eingehalten, wird die Anzahl der Parameter im dreidimensionalen Raum von sechs auf vier Parameter reduziert [2, 3, 7]:

- Angle (Gelenkwinkel) Θ : Winkel um die vorherige z-Achse, von der vorherigen x-Achse zur aktuellen x-Achse.
- Twist (Verdrehung) α : Winkel um die aktuelle x-Achse, von der alten z-Achse zur aktuellen z-Achse.
- Length (Segmentlänge) a : Abstand auf der aktuellen x-Achse von dem aktuellen Koordinatenursprung bis zum vorherigen Koordinatenursprung.
- Offset (Verschiebung/Vorschub) d : Verschiebung auf der vorherigen z-Achse von dem einem Koordinatenursprung zum anderem.

Wenn die Gelenke nummeriert werden, ergibt sich für das i -te Gelenk mathematisch [5]:

$${}^i_{i-1}T = {}^i_{i-1}R_{(Z_{i-1}, \Theta_i)} \cdot {}^i_{i-1}T_{(Z_{i-1}, d_i)} \cdot {}^i_{i-1}T_{(X_i, a_i)} \cdot {}^i_{i-1}R_{(X_i, \alpha_i)} \quad (17)$$

$$\begin{aligned} {}^i_{i-1}T = & \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \\ & \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (18)$$

aus dieser Matrixmultiplikation ergibt sich [5]:

$${}^i_{i-1}T = \begin{bmatrix} \cos(\Theta_i) & -\sin(\Theta_i) \cos(\alpha_i) & \sin(\Theta_i) \sin(\alpha_i) & a_i \cos(\Theta_i) \\ \sin(\Theta_i) & \cos(\Theta_i) \cos(\alpha_i) & -\cos(\Theta_i) \sin(\alpha_i) & a_i \sin(\Theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Bewegt sich das Robotersystem und verändert dementsprechend den Gelenkwinkel Θ_i eines Rotationsgelenks oder den Vorschub d_i eines Translationsgelenks, werden diese stetig aktualisiert. Die Parameter α_i und a_i , also die Verdrehung sowie die Segmentlänge sind hingegen immer konstant. Wie diese stetigen Veränderungen in dem begleitenden Projekt beachtet werden, soll in Abschnitt 3.3 erläutert werden. Mit diesen DH-Konvention konformen Transformationsmatrizen können folglich kinematische Ketten beschrieben werden.

2.3. BESONDERE ARTEN VON KINEMATISCHEN KETTEN

Als Grundlage für Problemstellungen, welche in den folgenden Abschnitten gelöst, beziehungsweise für das begleitende Projekt vorbereitet wird, folgen einige Sonderfälle für kinematische Ketten und weitere Unregelmäßigkeiten der Thematik. Für kinematische Ketten gibt es eine Unterscheidung in offen und geschlossen. Jedes Zwischensegment kann mit bis zu zwei Gelenken verbunden sein. Wenn jedes Segment einer kinematischen Kette mit genau zwei Gelenken verbunden ist, gilt sie als geschlossen, andernfalls als offen [8]. Darauf aufbauend können die Bezeichnungen seriell und parallel eingeführt werden, welche in Abb. 7 veranschaulicht werden. Eine serielle Kinematik ist eine offene kinematische Kette, jedes Zwischensegment ist an genau einer Stelle mit dem nachfolgenden Segment verbunden. Das Ende der kinematischen Kette bildet der Effektor, welcher mit dem letzten Segment verbunden ist. Die kinematische Kette hat nur eine Verbindung zum Sockel. Anders die parallele Kinematik, diese zeichnet sich durch eine geschlossene kinematische Kette aus, welche kein explizites Ende aufweist, dabei gibt es mehrere Verbindungsstücke zum Sockel. Der Effektor lässt sich, ausgehend vom Sockel, auf mindestens zwei Wegen erreichen [5]. Eine weitere abstraktere Darstellung dieser Unterschiede soll Abb. 8 aufzeigen.

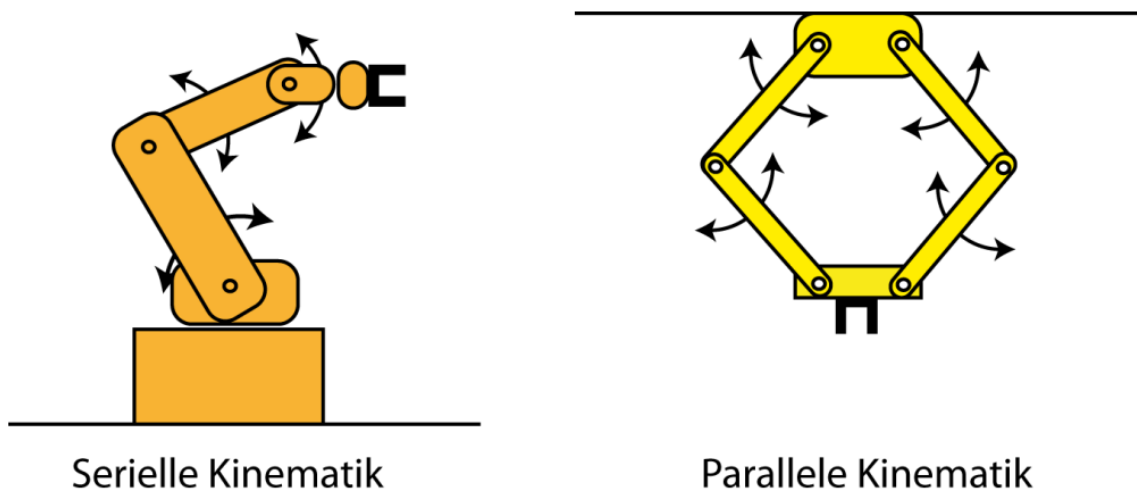


Abbildung 7.: Beispielhafte Darstellung von Industrierobotern zur Unterscheidung von seriellen (links) und parallelen (rechts) Kinematiken, entnommen aus [5].

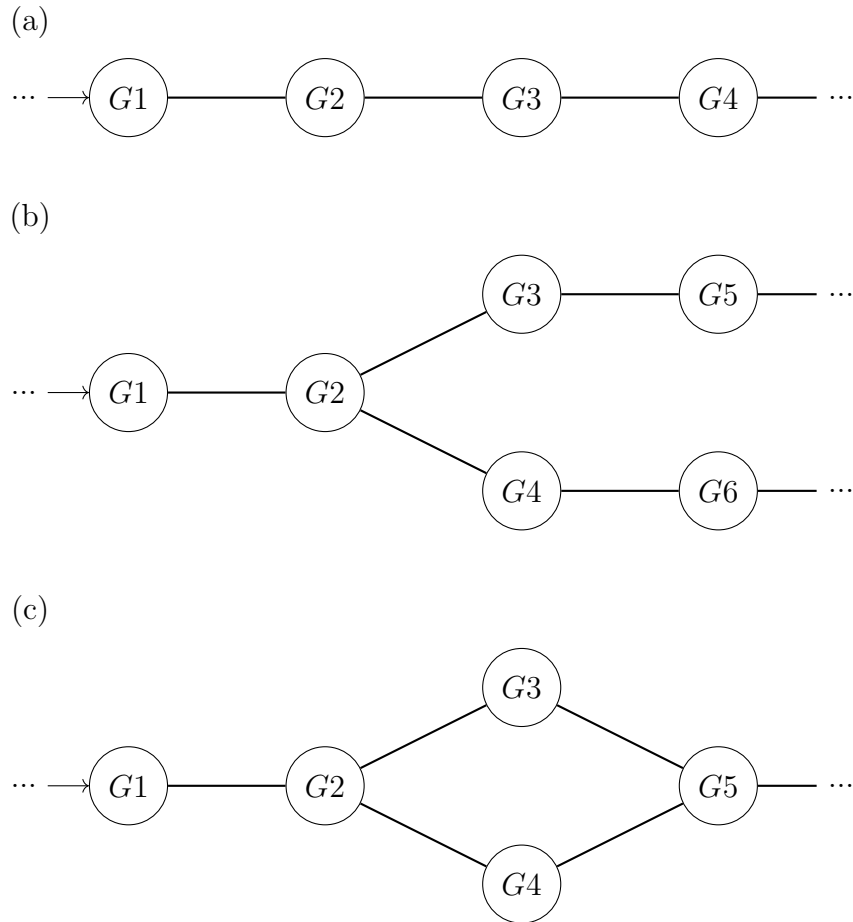


Abbildung 8.: Abstrakte Darstellungen von verschiedenen kinematischen Ketten mit G_i als Gelenke, zur Visualisierung der Begriffe *einfache kinematische Kette*, *parallele kinematische Kette* und *geschlossene kinematische Kette*. In (a) ist ein Beispiel für eine einfache kinematische Kette abgebildet, bei dem jedes Gelenk maximal einen Vorgänger und einen Nachfolger hat. In (b) hat das Gelenk G_2 zwei Nachfolnergelenke G_3 und G_4 , es bildet sich eine offene, parallele kinematische Kette ab diesem Punkt. In (c) teilt sich die kinematische Kette wie in (b) ab dem Gelenk G_2 in zwei weitere Gelenke G_3 und G_4 auf. Diese werden jedoch wieder zusammengeführt in Gelenk G_5 , sodass es sich anders als in (b) um eine geschlossene kinematische Kette in diesem Bereich ($G_2 - G_5$) handelt.

3. DARSTELLUNG VON BELIEBIGEN ROBOTERKINEMATIKEN

Für eine adäquate Darstellung und der Möglichkeit einer Erweiterung der kinematischen Kette, wird ein objektorientierter Ansatz gewählt. Welche Unterteilung in Klassen dafür notwendig ist, wie diese Verknüpft sind und wie mit der daraus entstandenen Datenstruktur gearbeitet wird, soll in diesem Kapitel erläutert werden.

3.1. ALLGEMEINE ANFORDERUNGEN AN DIE KLASSEN ROBOT UND JOINT

In diesem Abschnitt wird erläutert, wie die Klassen `Robot` und `Joint` zusammenhängen und welche Datenstrukturen für die Verwaltung einer kinematischen Kette verwendet werden. In Bezug auf die Anforderung an das begleitende Projekt werden dafür zwei Eigenschaften von kinematischen Ketten beachtet. Zum Einem müssen Parameter einzelner Gelenke für Berechnungen erfasst werden. Also die bereits in Kapitel 2 genannten Transformationsparameter, hier als vier DH-Parameter angegeben und weitere Attribute, wie die Art der Freiheitsgrade des Gelenks. Diese Arbeit beschränkt sich hierbei auf Rotations- und Translationsgelenke mit nur einem Freiheitsgrad, das System kann jedoch erweitert werden. Zum Anderem wird eine Verknüpfung zweier solcher Gelenke erfasst, um eine zusammenhängende Struktur erzeugen zu können. Diese Unterteilung ermöglicht eine Erweiterung und eine Wiederverwendbarkeit einer kinematischen Kette, sodass diese zum Beispiel grafisch dargestellt werden können. Wie in Abb. 9 dargestellt, werden dementsprechend zwei Klassen für eine kinematische Kette erzeugt. `Joint`, eine Gelenkklassse als Parameterspeicher und `Robot` als verwaltende Roboterklasse. Eine Instanz von `Robot` besitzt ein `Joint`-Objekt als Startpunkt der kinematischen Kette, welches das Basiskoordinatensystem des Systems widerspiegelt. Ein `Robot`-Objekt instanziiert Objekte der Klasse `Joint` und hängt sie an die vorgesehene Stelle der kinematischen Kette, benennt sie und setzt Parameter. Dabei müssen jeweilige Elter-Kind-Beziehungen berücksichtigt werden, sodass einerseits ein Objekt von `Joint` nur die nötigen Beziehungen zu seinem Elter und Kindern kennt und andererseits insgesamt eine Struktur erzeugt wird, durch die iteriert werden kann. Vorgesehen sind kinematische Ketten, welche sich beliebig in mehrere Pfade unterteilen können und gegebenenfalls auch wieder zusammengeführt werden und dann eine geschlossene kinematische Kette bilden, vgl. Abb. 8 und Abschnitt 2.3. Die verwaltende `Robot`-Klasse bildet im Allgemeinen eine Baumstruktur ab, mit `Joint`-Objekten als Knoten. Dabei kann sich die Komplexität der Struktur ändern, je nach Art der kinematischen Kette. Bei einer seriellen Kette würde jeder Knoten nur eine Abzweigung aufweisen und somit eine doppelt verkettete Liste darstellen. Bei einer mehrfach verzweigten parallelen Kette hingen würde die Anzahl der Abzweigung an bestimmten Knotenpunkten variieren und eine Baumstruktur bilden.

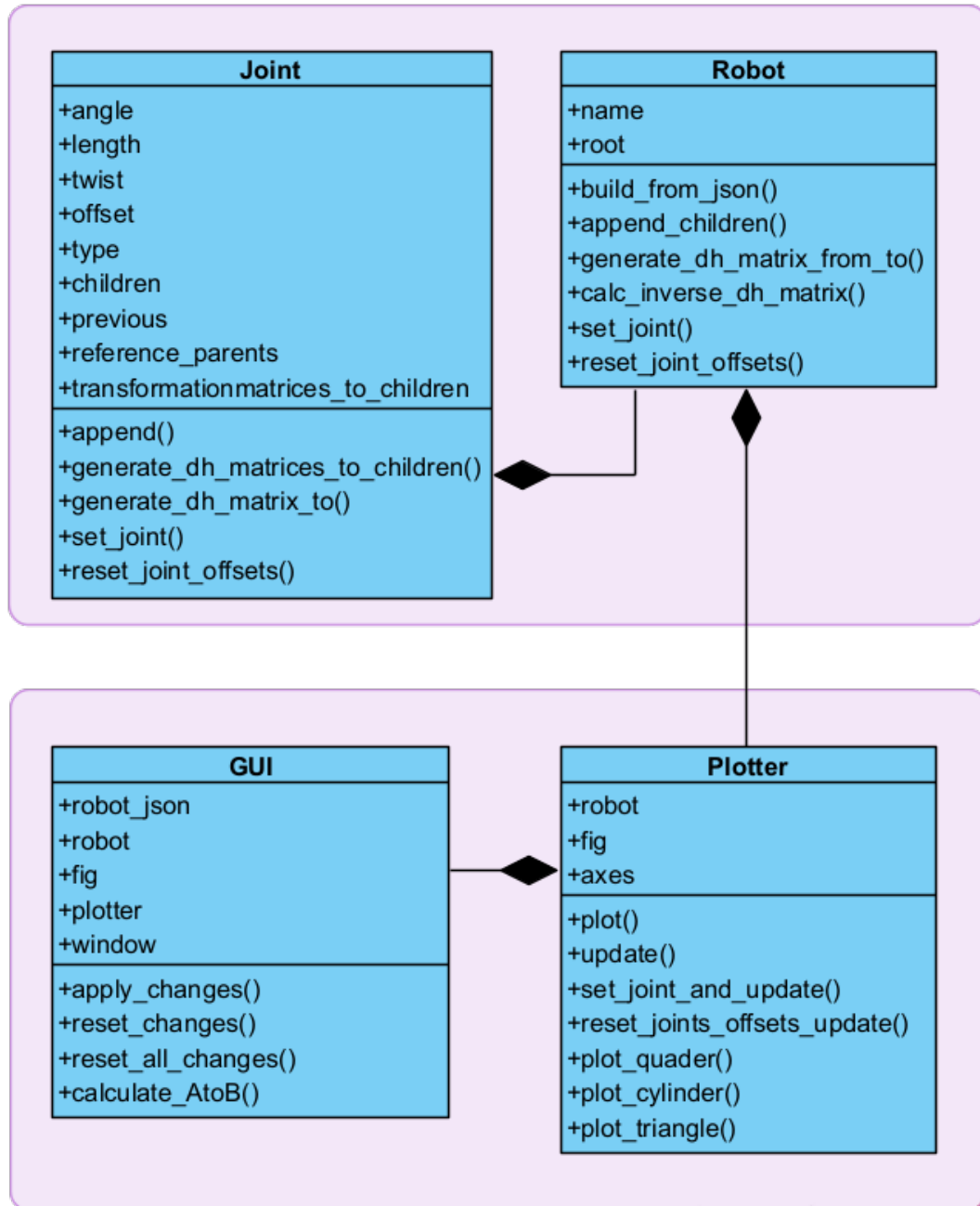


Abbildung 9.: UML-Klassendiagramm des Visualisierungsprogramms dieser Studienarbeit. Dargestellt sind die Klassen `Robot`, `Joint` und `Plotter`. Die Klasse `Plotter` besitzt ein Objekt der Klasse `Robot`. Die Klasse `Robot` instanziert und verwaltet ein `Joint`-Objekt `Root`, von dem weitere `Joint`-Objekte als Kinder abzweigen können. Abgebildet sind die Attribute und Methoden dieser Klassen. Die Klassen `Robot` und `Joint` sind grafisch von der Klasse `Plotter` abgegrenzt um zu symbolisieren, dass diese unabhängig voneinander genutzt werden können.

3.2. VERKNÜPFUNG DER GELENKE IN EINER KINEMATISCHEN KETTE DURCH EINE BAUMSTRUKTUR

Im Folgenden wird die verwaltende Datenstruktur der `Robot`-Klasse genauer beschrieben. Dabei werden vor allem die in Abschnitt 2.3 behandelten Elter-Kind-Beziehungen und Probleme der geschlossenen beziehungsweise parallelen kinematischen Ketten aufgegriffen und im Zusammenhang der Datenstruktur erläutert. Wie in Abschnitt 3.1 behandelt, werden die Gelenkobjekte bei der Initialisierung des virtuellen Robotersystems nacheinander an die bereits bestehende Struktur gehangen. Dies ist im direkten Vergleich der DH-Konvention zu sehen, dort werden die DH-Parameter für das jeweils folgende Glied der Kette benutzt, um die jeweilige Transformationsmatrix ${}^i_{i-1}T$ zu erzeugen. Ein Gelenkobjekt benötigt dementsprechend die DH-Parameter des Folgegelenks um eine Transformationsmatrix berechnen zu können. Diese benötigten Parameter stehen dem neuen Gelenkobjekt als Attribute zu Verfügung. Daraus resultiert unmittelbar, dass ein Gelenk ein Folgegelenk kennen muss, um Zugriff auf diese Parameter zu erhalten. Ebenso muss das neu angehangene Gelenk das Eltergelenk (das vorherige Gelenk der Kette) kennen, um eine möglichst schnelle Iteration durch die Struktur zu ermöglichen. Folgend gelte die Abb. 10 als Grundlage zur Erläuterung der unterschiedlichen Bedingungen für eine Elter-Kind-Beziehung für verschiedene Arten von Strukturen. Die Abbildung ist als Erweiterung der bereits behandelten Abb. 8 zu erachten, erweitert um die nun eingeführten Elter-Kind-Beziehungen, unabhängig von der Art der kinematischen Kette.

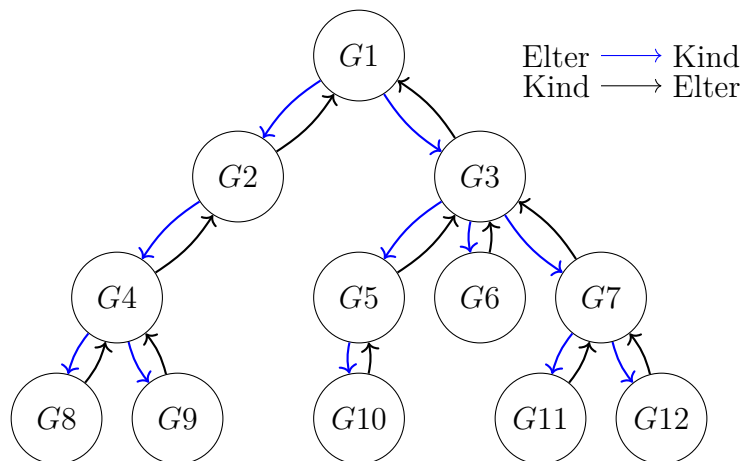


Abbildung 10.: Darstellung einer resultierenden Baumstruktur durch Nutzung von offenen, parallel verlaufenden kinematischen Ketten. Die Beziehung von zwei Gelenken in einer einzelnen kinematischen Kette (z.B. $G1$ - $G2$ - $G4$ - $G8$) basiert auf einer doppelt verketteten Liste in Form von Eltern-Kind-Beziehungen. Jedes Kind (jedes Gelenk) hat dabei maximal einen Elter und beliebig viele Kinder.

Solange die gesamte kinematische Kette keine geschlossenen Teilstrukturen aufweist, gelten stets die aufgezeigten Beziehungen der Gelenke untereinander. Werden jedoch, wie in Abb. 11 angedeutet, bestimmte Gelenke unmittelbar miteinander verbunden, sodass sich eine geschlossene kinematische Kette bildet, muss die Elter-Kind-Beziehung einzelner Gelenkpaare erweitert werden.

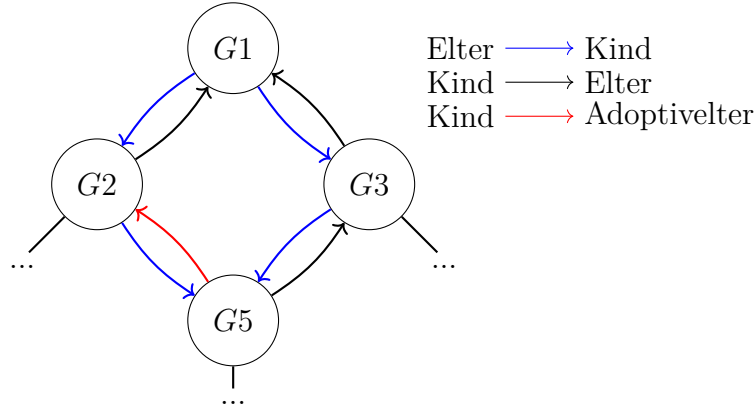


Abbildung 11.: Erweiterung der Abb. 10, die Gelenke $G2$ und $G5$ werden verbunden, daraus resultiert eine geschlossene kinematische Kette mit den Gelenken $G1$, $G2$, $G3$ und $G5$

Diese Problematik soll nun anhand des Beispiels aus Abb. 11 untersucht werden. Das Problem besteht darin, dass ein Gelenk, welches n ($n > 1$) Elterobjekte hat, auch automatisch n mal als Kindobjekt eingetragen ist. Dadurch muss es dementsprechend im Beispiel zweimal einen Eintrag zu DH-Parametern in diesem Gelenk $G5$ geben. Einmal die Parameter in der Kette von Gelenk $G2$ zu Gelenk $G5$ und einmal von Gelenk $G3$ zu Gelenk $G5$. Solange die Eltergelenke $G2$ und $G3$ nicht identisch sind und die gleiche räumliche Beziehung zu Gelenk $G5$ aufweisen, sind diese Parameter jeweils unterschiedlich. Zudem entsteht ein weiteres Problem, sobald das Gelenk $G5$ räumlich verschoben werden soll. Die zu verändernden Gelenkeinstellungen (Gelenkwinkel bzw. Vorschublänge) in den Eltergelenken sind nicht zwingend die gleichen. Dies hängt von den schon vorher differenten räumlichen Beziehungen zwischen den Elter-Kind Paaren ab. Für das Lösen dieser Problematik gibt es verschiedene Ansätze. Eine Möglichkeit wäre es zum Beispiel, die Reihenfolge der Gelenke in der kinematischen Kette zu ändern, sodass kein Gelenk mehr als ein Elterobjekt besitzt. Dafür würde in dem Beispiel das Gelenk $G5$ nicht weiter ein Kind von Gelenk $G3$ sein, sondern genau entgegengesetzt. Ferner müsste Gelenk $G3$ Elter von Gelenk $G1$ werden, sodass sich ein geschlossener Kreis über die Gelenke $G1$, $G2$, $G5$ und $G3$ zurück zu $G1$ bildet. Ersichtlich wird, dass bei diesem Ansatz zumeist auch eigentlich konforme Beziehungen zwischen Gelenken, verändert werden müssen. Dadurch wird die kinematische Kette grundlegend verändert. Die Abbildung der kinematischen Kette soll möglichst der vorliegenden kinematischen Kette entsprechen, weil sie eben diese darstellen soll. Des Weiteren entstünden Probleme in der später durchzuführenden Iteration durch die Baumstruktur. Hier könnte es

zu Endlosschleifen kommen, weil es kein direktes Ende in einer solchen Struktur geben würde. Dafür wären dann zusätzliche Maßnahmen, zum Beispiel Markierungen für ein Ende notwendig, was zusätzliche Rechenzeit in Abfragen bedeuten würde. Aus genannten Gründen wird dieser Ansatz deshalb nicht weiterverfolgt. Wie in dem Beispiel angedeutet wird ein anderer Ansatz in dieser Ausarbeitung zur Lösung der Problematik angewendet. Alle Elterobjekte desselben Kindes, bis auf eins, erhalten eine andere Art von Elter-Kind-Beziehung mit ihrem Kind. Im Beispiel bedeutet dies, dass alle Gelenke bis auf Gelenk G3 zu dem Kind G5 keine direkten Eltern mehr wären. Die Beziehung von Gelenk G2 zu Gelenk G5 würde dementsprechend gesondert unterschieden werden. Dadurch würde jedes Gelenk wieder genau einen direkten Elter besitzen. Die andere Beziehung beruht darauf, dass in dem Kindobjekt nur ein Gelenkobjekt als direkter Elter einzutragen ist, alle weiteren Gelenke, welche eigentlich auch als Elter einzutragen wären, bekommen einen Sonderstatus. Im Beispiel würde also Gelenk G2 in Gelenk G5 mit eben diesem Sonderstatus als Elterobjekt eingetragen werden. Dieser Status soll für Erklärungen ferner Adoptivelter genannt werden. Durch diese Änderung entstehen neue Möglichkeiten der Iteration, wenn die Gelenke später nacheinander abgebildet werden sollen. Um dies zu erläutern wird der Algorithmus zur Darstellung der Baumstruktur aufgeführt, das Wurzelobjekt wird explizit, unabhängig von dem Algorithmus abgebildet. Start des Algorithmus ist also nachdem das Gelenk G1 abgebildet wurde, es gilt die restlichen Gelenke darzustellen. Grundlegendes Konzept ist eine rekursive Tiefensuche:

1. Iteration durch jedes eingetragene Kindobjekt in einer Kinderliste, sobald kein Kind mehr vorhanden ist → Sprung zurück zum Elter
2. Abbruch, wenn die Kinderliste vollständig abgearbeitet wurde
3. Prüfen, ob das aktuelle Gelenk als Elter oder Adoptivelter in dem aktuellen Kind eingetragen ist:
 - a) Eingetragen als Elter:
 - i. DH-Parameter zum Kind anfragen
 - ii. Transformationsmatrix bestimmen
 - iii. Kind und Verbindungsstrecke abbilden
 - iv. Aufrufen von Schritt 1 für das Kind (Rekursion)
 - b) Eingetragen als Adoptivelter:
 - i. Bestimmen der eigenen Position in der Adoptivelterliste des Kindes
 - ii. Passende Adoptiv-DH-Parameter zum Kind aus der Adoptivelter-DH-Parameter-Liste abfragen (Stelle ergibt sich aus der zuvor bestimmten Position in der Liste)
 - iii. Transformationsmatrix bestimmen
 - iv. Verbindungsstrecke abbilden (*nicht* das Kind selbst)
 - v. Sprung zurück zu Schritt 1

Im allgemeinen handelt es sich um eine normale rekursive Tiefensuche durch die Baumstruktur, es wird jedoch stetig die Beziehung vom einem Kindobjekt zu sich selbst geprüft und je nach Art, ein anderes Vorgehen gewählt. Mit dieser Erweiterung können geschlossene kinematische Ketten abgebildet werden, ohne die Struktur der Kette selbst grundlegend zu verändern. Das Kindobjekt mit Adoptivelter muss selbstverständlich dennoch verschiedene DH-Parameter für verschiedene Elter aufweisen, sonst gelten gleiche Probleme, wie in dem vorher angeführten Ansatz.

3.3. ABBILDUNG EINES ROBOTERSYSTEMS

Im vorherigen Kapitel wurde bereits das Grundkonzept für eine Abbildung des stationären Systems dargestellt. Ergänzend zu Kapitel 2 und diesem Grundkonzept, soll folgendes Kapitel beschreiben, welche Möglichkeiten durch die berechneten Transformationsmatrizen entstanden sind und wie diese auch für Bewegungen des Robotersystems in der Darstellung genutzt werden können. Die erzeugten Transformationsmatrizen zwischen zwei Gelenken sind nicht auf die Transformation von einem Koordinatenursprung zu einem anderem beschränkt. Das bedeutet, dass ein Punkt, gegeben im Zielkoordinatensystem mit der gleichen Transformationsmatrix transformiert werden kann, wie der Ursprung des Systems. Für diese Anwendung bedeutet dies, dass Grafiken, wie zum Beispiel Koordinatenachsen oder dreidimensionale Symbole zur Unterscheidung von Gelenkarten transformiert werden können ohne größeren mathematischen Aufwand als ohnehin schon gegeben. Jene Symbole, Grafiken, Beschriftungen oder Ähnliches können demnach relativ zum eigenem Koordinatenursprung angegeben werden. Dabei können identische Plotfunktionen für verschiedene Gelenkobjekte genutzt werden, unabhängig von der Position des Koordinatensystems im Raum. In Abschnitt 2.2 wurde bereits die Transformationsmatrix für einen stationären Fall berechnet. Zudem wurde erläutert, dass der Gelenkwinkel Θ_i bei einem Rotationsgelenk beziehungsweise der Vorschub d_i bei einem Translationsgelenk variabel sind. Wenn diese Parameter demnach in der Transformationsmatrix verändert werden, spiegeln diese eine Transformation, hier eine Bewegung, des Systems wider. Die `Joint`-Klasse kann um solche Änderungsparameter erweitert werden, sodass diese unmittelbar Einfluss auf die Transformationsmatrix haben, ohne die ursprünglichen DH-Parameter zu verändern. Dies ermöglicht eine einfache Reset-Funktion.

Die Transformationsmatrix aus Gleichung (19) wird dementsprechend erweitert:

$${}_{i-1}^iT = \begin{bmatrix} \cos(\Theta_{i2}) & -\sin(\Theta_{i2})\cos(\alpha_i) & \sin(\Theta_{i2})\sin(\alpha_i) & a_i\cos(\Theta_{i2}) \\ \sin(\Theta_{i2}) & \cos(\Theta_{i2})\cos(\alpha_i) & -\cos(\Theta_{i2})\sin(\alpha_i) & a_i\sin(\Theta_{i2}) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_{i2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

wobei gilt:

$$\Theta_{i2} = \Theta_i + \Theta_{io} \quad (21)$$

und

$$d_{i2} = d_i + d_{io} \quad (22)$$

dabei sind Θ_{io} und d_{io} die Offsetparameter der jeweiligen Größe. Im Allgemeinen können mithilfe der Transformationsmatrizen alle Koordinatenursprünge, also die Gelenke selbst, sowie zusätzliche Grafiken in Koordinaten des Basiskoordinatensystems dargestellt werden. Dieses Basiskoordinatensystem stellt ein dreidimensionalen Plotbereich da, welcher das vorliegende Robotersystem adäquat nach beschriebenem Algorithmus abbilden und Bewegungen aktiv darstellen kann.

4. DISKUSSION UND AUSBLICK

In dieser Arbeit konnte ein simpler, interaktiver dreidimensionaler Plotter für Robotersysteme realisiert werden. Dieser stellt kinematische Ketten dynamisch als dreidimensionale Objekte dar. Gelenkparameter wie Winkel und Vorschub können interaktiv verändert und so deren Einflüsse veranschaulicht werden. Dafür wurde die theoretische Grundlage von Roboterkinematiken, mit verschiedenen Arten von kinematischen Ketten in Kapitel 2 erläutert. Ferner wurden in diesem Zusammenhang Transformationsmatrizen eingeführt, um ein mathematisches Modell von kinematischen Ketten abbilden zu können. Die Modellbildung wurde mit der Denavit-Hartenberg-Konvention erweitert. Dafür müssen Regeln bei der Definition der Gelenkkoordinatensysteme eingehalten werden, wodurch die benötigte Parameterzahl reduziert werden kann. Im Weiteren wurden Probleme und Lösungsansätze für geschlossene kinematische Ketten aufgezeigt. Die dargestellten Informationen wurden in Kapitel 3 genutzt, um einen Ansatz für objektorientierte Darstellungen von Robotersystemen zu realisieren. Ferner wurde hier eine Baumstruktur erzeugt, in welcher jedes Gelenk ein Knoten in der Struktur widerspiegelt. Das Projekt ist beschränkt auf die DH-Konvention, demnach müssen die Bedingungen dieser Konvention bei dem Überführen eines Robotersystems in das Programm zwingend eingehalten werden. Des Weiteren sind in der ersten Version nur Rotations- und Translationsgelenke, mit nur einem Freiheitsgrad zulässig. Ist die Implementierung von anderen Gelenken erwünscht, müssten komplexere Abfragen während des Abbildens im stationären sowie im bewegten Zustand stattfinden. Die gegebenen Informationen einer kinematischen Kette beschränken sich zudem nur auf die relative Position und Orientierung untereinander. Hier gibt es bis jetzt keine Anwendung von weiteren physikalischen Eigenschaften, was eine realitätsnähere Simulation zulassen würde. Durch einen objektorientierten Ansatz ist der Zugang für eine Erweiterung solcher und ähnlicher Funktionen generell gewährleistet und kann in Zukunft untersucht und implementiert werden.

QUELLENVERZEICHNIS

- [1] M. Webhofer. *Verwendung kinematischer Ketten bei der Generierung von Finite-Elemente-Modellen*. 2005.
- [2] J. J. Craig. *Introduction to robotics. Mechanics and control*. Prentice Hall, 2003.
- [3] M. Hulse. *Folien Robotik 02*. 2021.
- [4] Jianwei Zhang. *Einführung in die Robotik*. Hamburg, 13.04.2010.
- [5] Prof. Dr. Klaus Wüst. *Grundlagen der Robotik: Skriptum*. 25.06.2018.
- [6] W. Globke. *Koordinaten, Transformationen und Roboter*.
- [7] S. Hesse und G. Seitz. *Robotik: Grundwissen für die berufliche Bildung*. Viewegs Fachbücher der Technik. Braunschweig und Wiesbaden: Vieweg, 1996.
- [8] Anonym. *Roboterkinematik: Roboterarm und Gelenke*. TU Chemnitz. 2013.

A. ANHANG

Im vorliegenden Anhang sollen Einblicke und Hilfestellungen zur Nutzung des begleitenden Programms gestellt werden. Darüber hinaus werden Arbeitsschritte für eine erfolgreiche Abbildung beliebiger Roboterkinematiken aufgezeigt.

A.1. ÜBERFÜHREN EINES ROBOTERSYSTEMS IN DAS PROGRAMM

Wie im Hauptteil dieser Ausarbeitung beschrieben, benötigt das Programm die DH-Parameter aller Gelenke in der kinematischen Kette. Des Weiteren muss eine Angabe über die Art der Gelenke (Rotations- oder Translationsgelenk) stattfinden. Für eine benutzerfreundlichere Anzeige, muss ein fiktiver Name jedes Gelenks eingetragen werden („Alpha-1-Gelenk“ usw.). Darüber hinaus dient diese Angabe für die Darstellung geschlossener Kinematiken als Markierung für Duplikate. Sobald ein solcher `title` mehrfach vorkommt, werden die Angaben der Parameter, nach dem ersten Mal, nicht für das Hinzufügen eines neuen Gelenks, sondern für die Adoptivelter-Kind-Beziehungen eines bestehenden Gelenks genutzt. Die entstehende Struktur der kinematischen Kette wird wie in Kapitel 2 und Kapitel 3 beschrieben, eine Baumstruktur bilden. Eben diese Struktur muss von der Programm nutzenden Person direkt angegeben werden. Dies geschieht über ein JSON-File, eine Datei für die Angabe von Objektstrukturen. Folgend soll der Grundaufbau einer solchen Datei, welche von dem Plotterprogramm gelesen werden kann, dargestellt werden.

A.1.1. GRUNDAUFBAU DER JSON-DATEI

Die Standardstruktur eines jeden Roboters in der JSON-Datei ist immer gleich. Wie in folgender Abb. 12 zu sehen, beginnt die Datei mit einer öffnenden geschweiften Klammer und endet mit einer schließenden geschweiften Klammer. Die Beschreibung der Kinematik wird unter dem Namen *robot*, als Liste der Folgegelenke, hinterlegt. Innerhalb dieser Liste werden im späteren Verlauf die Struktur und die Parameter, des zu implementierenden Roboters, angegeben. Bei jedem Roboter wird mindestens ein Gelenkobjekt wie beschrieben erzeugt. Dabei handelt es sich um ein Platzhalter Element welcher den Körper beziehungsweise den Sockel des Roboters darstellen soll, dieses liegt im Ursprung des Basiskoordinatensystems. Demnach wird bei der Einspeisung der gegebenen JSON-Datei folgender Plot, Abb. 13 angezeigt.

```
{  
  "robot": [  
  ]  
}
```

Abbildung 12.: JSON-Code für die Darstellung eines leeren Roboters. Erzeugt wird damit der in Abb. 13 aufgezeigte Plotter, sofern diese Datei als Quelle angegeben wird.

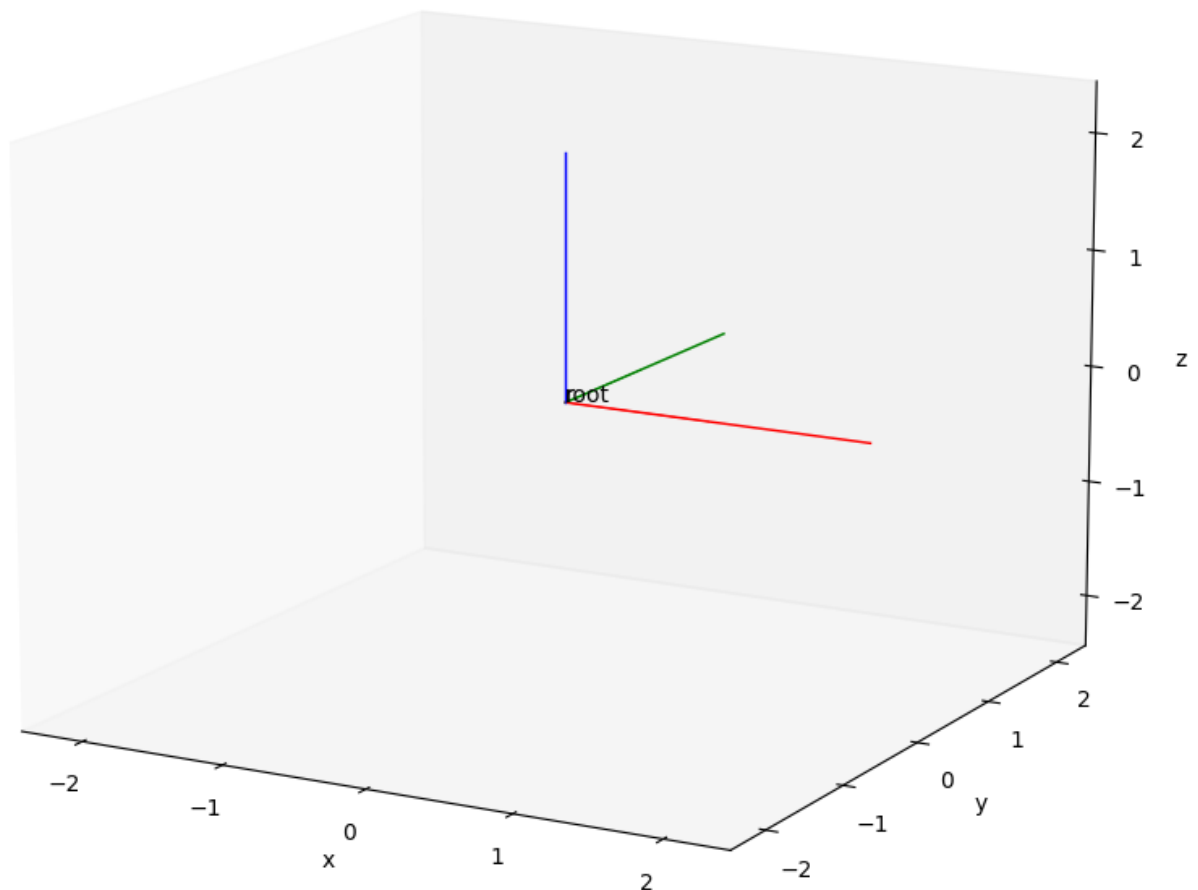


Abbildung 13.: Plot des Roboters im Programm unter der Angabe der JSON-Datei aus Abb. 12. Es handelt sich um einen leeren Roboter, demnach wird nur ein Sockeläquivalent beziehungsweise Basisgelenk im Basiskoordinatensystemursprung dargestellt.

A.1.2. SERIELLE KINEMATIK AUS DREI GELENKEN

Folgend wird die im letzten Abschnitt beschriebene Struktur erweitert. Dargestellt werden soll eine kinematische Kette mit drei Gelenken, welche seriell nacheinander geschaltet sind. Hierfür werden wie in Abb. 14 gezeigt, die Parameter der einzelnen Gelenke angegeben. Ein einzelnes Gelenk wird erzeugt mit Angabe folgender Parameter:

title Der angezeigte Titel, über welchem der Zugriff auf ein Gelenk in verschiedenen Funktionen erfolgt (Bsp: `generate_dh_matrix_from_to(_from, _to)`, mit `_from` als Titel von Gelenk A und `_to` als Titel von Gelenk B). Falls Duplikate für geschlossene kinematische Ketten erzeugt werden sollen, muss für diese der gleiche Titel verwendet werden, andernfalls müssen diese einzigartig sein.

type Hier wird der Typ des Gelenks eingetragen. Zur Auswahl stehen „translation“ für ein Lineargelenk, „rotation“ für ein Drehgelenk mit jeweils einem Freiheitsgrad entlang der Z-Achse und „TCP“ für einen Endeffektor

angle Attribut für den Winkel in Radiant des gleichnamigen DH-Parameters (Θ). Dieser kann als float sowie in Verbindung mit pi gewählt werden, z.B. „-0.34195“, „-pi/2“, „2*pi“.

length Attribut für den gleichnamigen DH-Parameter (a)

offset Attribut für den gleichnamigen DH-Parameter (d)

twist Attribut für den Winkel in Radiant des gleichnamigen DH-Parameters (α). Dieser kann als float sowie in Verbindung mit pi gewählt werden, z.B. „-0.34195“, „-pi/2“, „2*pi“.

children Liste aus Folgegelenken, hier wird unmittelbar die Elter-Kind-Beziehung der Baumstruktur angegeben.

Bei dem Import der vorliegenden Datei, Abb. 14 wird folgender Plot, Abb. 15 dargestellt.

```
{
  "robot": [
    {
      "title": "Alpha1-Gelenk",
      "type": "translation",
      "angle": "pi/2",
      "length": "1.2",
      "offset": "1.3",
      "twist": "0",
      "children": [
        {
          "title": "Beta1-Gelenk",
          "type": "rotation",
          "angle": "0",
          "length": "2.3",
          "offset": "1",
          "twist": "pi",
          "children": [
            {
              "title": "Gamma1-Gelenk",
              "type": "rotation",
              "angle": "0",
              "length": "1",
              "offset": "3",
              "twist": "0.7854"
            }
          ]
        }
      ]
    }
  ]
}
```

Abbildung 14.: JSON-Code für die Darstellung einer seriellen Kinematik mit drei Gelenken. Erzeugt wird damit der in Abb. 15 abgebildete Plot.

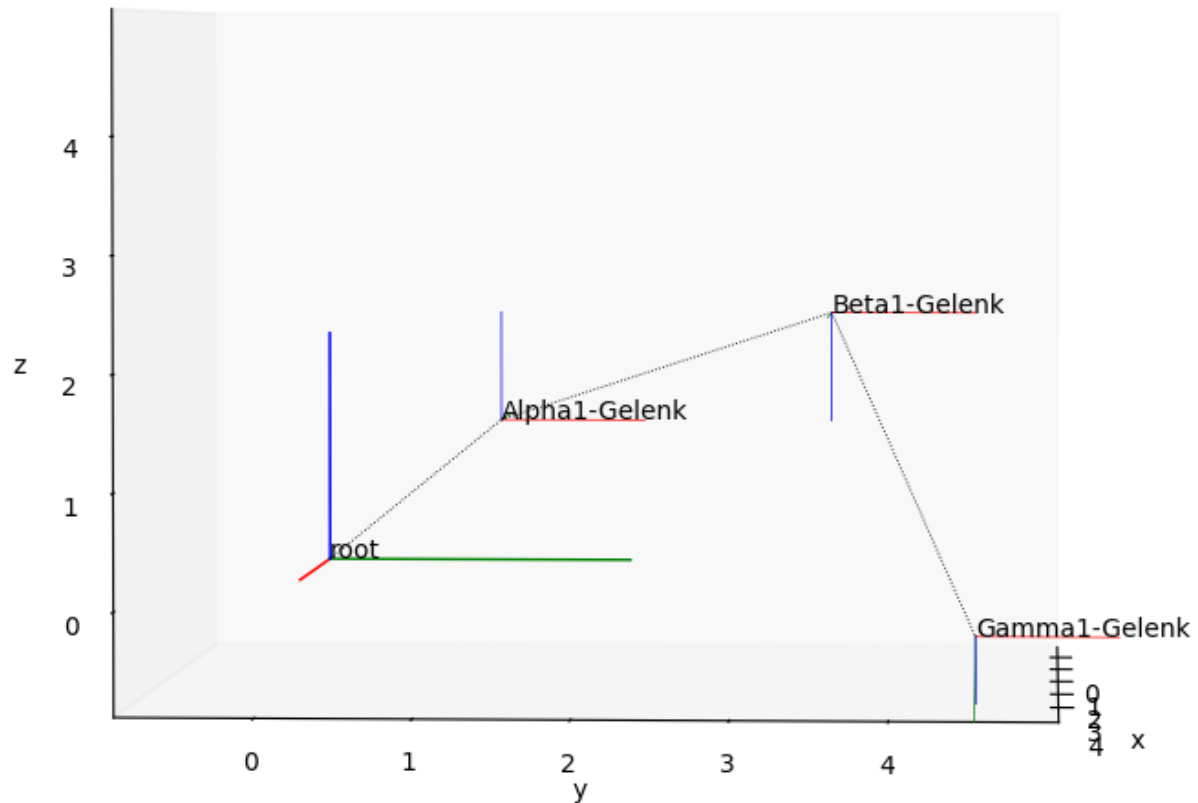


Abbildung 15.: Plot einer seriellen kinematischen Kette mit drei Gelenken „Alpha1-Gelenk“, „Beta1-Gelenk“ und „Gamma1-Gelenk“ im Programm unter Angabe des in Abb. 14 angegebenen Quellcodes.

A.1.3. SERIELLE KINEMATIK MIT PARALLEL VERLAUFENDEN ABZWEIGUNGEN

Die kinematische Kette aus vorherigem Abschnitt wird erweitert. Ein Gelenk, hier „Alpha1-Gelenk“, erhält ein zweites Folgegelenk, sodass sich hier die kinematische Kette in zwei parallel verlaufende Ketten teilt. Beide kinematische Ketten sind ab jenem Gelenk unabhängig voneinander zu sehen. Um einen Plot, vgl. Abb. 17 für eine solche Struktur zu erzeugen, wird folgender Quellcode Abb. 16 benötigt.

```
{
  "robot": [
    {
      "title": "Alpha1-Gelenk",
      "type": "rotation",
      "angle": "pi/2",
      "length": "1.2",
      "offset": "1.3",
      "twist": "0",
      "children": [
        {
          "title": "Beta1-
            Gelenk",
          "type": "rotation",
          "angle": "0",
          "length": "2.3",
          "offset": "1",
          "twist": "pi",
          "children": [
            {
              "title": "Gamma
                1-Gelenk",
              "type": "
                rotation",
              "angle": "0",
              "length": "1",
              "offset": "3",
              "twist": "0.785
                4"
            }
          ]
        }
      ]
    },
    {
      "title": "Beta2-
        Gelenk",
      "type": "rotation",
      "angle": "0.7854",
      "length": "1.5",
      "offset": "0.2",
      "twist": "-0.7854",
      "children": [
        {
          "title": "Gamma
            2-Gelenk",
          "type": "
            rotation",
          "angle": "-1.57
            0796",
          "length": "0.5",
          "offset": "2",
          "twist": "0"
        }
      ]
    }
  ]
}
```

Abbildung 16.: JSON-Code für die Darstellung einer seriellen Kinematik mit parallel verlaufenden Abzweigungen. Es kann der in Abb. 17 dargestellte Plot erzeugt werden.

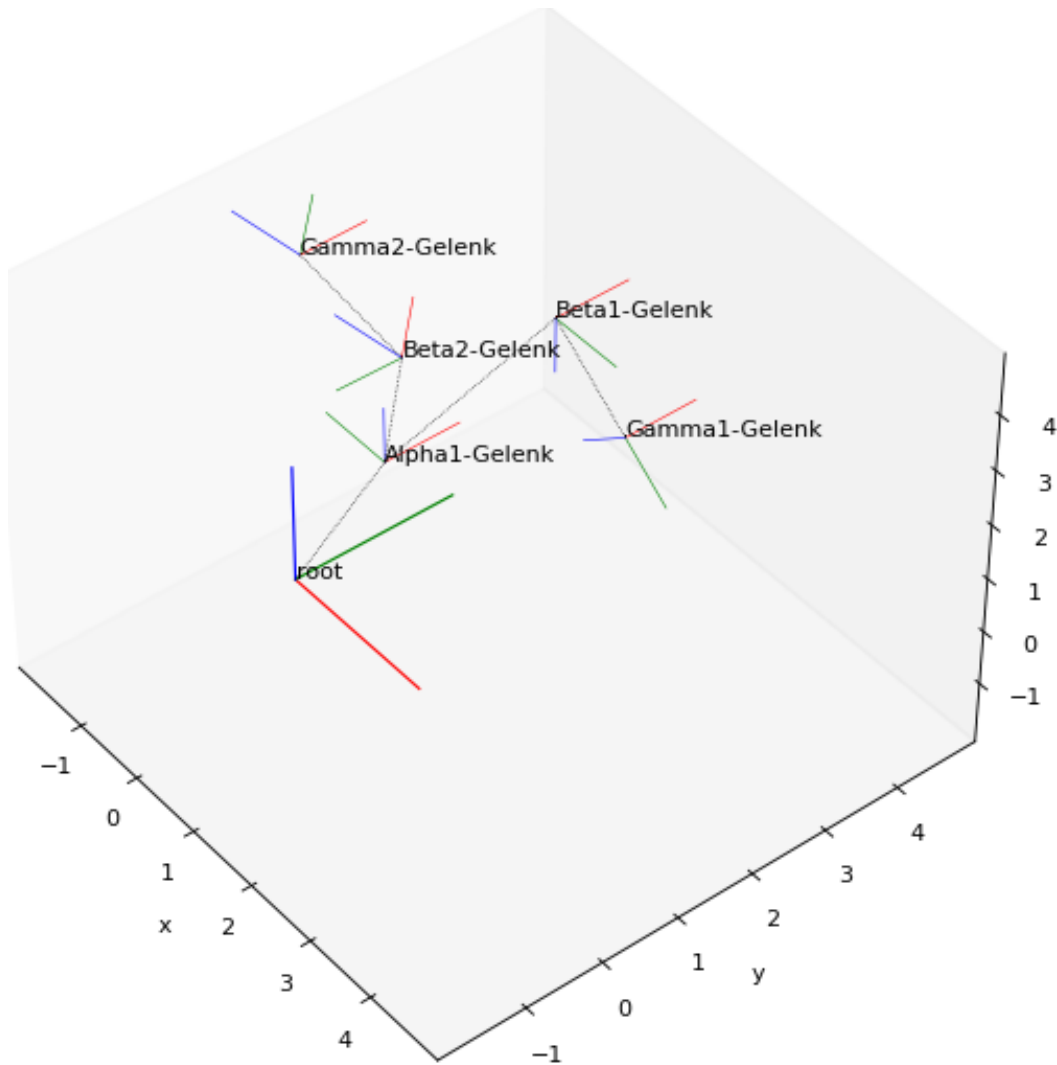


Abbildung 17.: Plot einer seriellen Kette mit parallel verlaufenden Abzweigung ab dem „Alpha1-Gelenk“ im Programm unter Angabe des in Abb. 16 gegebenen Quellcodes.

A.1.4. GESCHLOSSENE KINEMATISCHE KETTE

Wie beschrieben, können auch geschlossene kinematische Kette dargestellt werden. Das Gelenk, an dem zwei parallel verlaufende Ketten wieder zusammengeführt werden sollen, muss dafür ein zweites Mal angegeben werden (gleicher „title“, die anderen Parameter dürfen abweichen). Jeweils einmal als Kind von beiden Gelenken, die in ein Gelenk zusammengeführt werden sollen. Die erste Angabe des Gelenks erzeugt das Gelenkobjekt und stellt dieses dar. Die zweite Angabe setzt das zweite vorherige Gelenk als Adoptivelter und nutzt die angegebenen Parameter zur Abbildung der Verbindungsstrecke zwischen Adoptivelter und Kind. Als Beispiel für eine solche Anwendung soll Abb. 18 dienen, der aus diesem Quellcode erzeugte Plot ist in Abb. 19 dargestellt.

```
{
  "robot": [
    {
      "title": "Alpha1-Gelenk",
      "type": "rotation",
      "angle": "pi/2",
      "length": "1.2",
      "offset": "1.3",
      "twist": "0",
      "children": [
        {
          "title": "Beta1-Gelenk",
          "type": "rotation",
          "angle": "0",
          "length": "2",
          "offset": "1",
          "twist": "0",
          "children": [
            {
              "title": "Gamma1-Gelenk",
              "type": "rotation",
              "angle": "0",
              "length": "2",
              "offset": "-1",
              "twist": "0"
            }
          ]
        }
      ]
    },
    {
      "title": "Beta2-Gelenk",
      "type": "rotation",
      "angle": "0",
      "length": "2",
      "offset": "-1",
      "twist": "0",
      "children": [
        {
          "title": "Gamma1-Gelenk",
          "type": "rotation",
          "angle": "0",
          "length": "2",
          "offset": "1",
          "twist": "0"
        }
      ]
    }
  ]
}
```

Abbildung 18.: JSON-Code für die Darstellung einer geschlossenen kinematischen Kette. Das „Gamma1-Gelenk“ wird mit verschiedenen Parametern zweimal angegeben. Das „Beta1-Gelenk“ wird dadurch als Elter und das „Beta2-Gelenk“ wird dadurch als Adoptivelter gesetzt. Es kann der in Abb. 19 dargestellte Plot erzeugt werden.

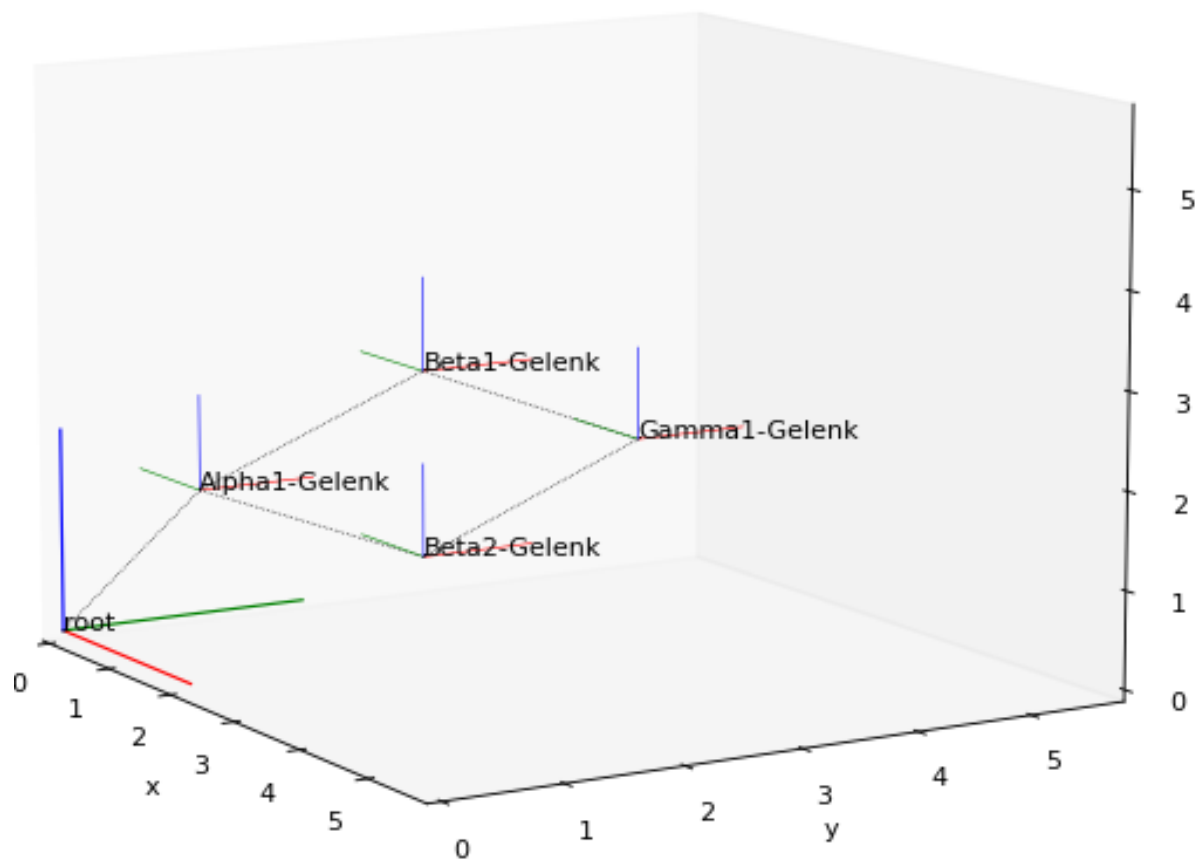


Abbildung 19.: Plot einer geschlossenen kinematischen Kette unter Angabe des in Abb. 18 gegebenen Quellcodes.

EIGENSTÄNDIGKEITSERKLÄRUNG

Wir, Ben Raeke und Sascha Nickel, haben diese Arbeit eigenständig und ohne Hilfe Dritter angefertigt. Fremde Inhalte haben wir als Zitat kenntlich gemacht.

Ben Raeke
Studiengang Ingenieurinformatik
Fachhochschule Bielefeld
Matrikelnummer: 1220577

Bielefeld den 23. Juni 2022

Sascha Nickel
Studiengang Ingenieurinformatik
Fachhochschule Bielefeld
Matrikelnummer: 1205844

Bielefeld den 23. Juni 2022