



Technische Universität Brunn

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTÄT FÜR MASCHINENBAU

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUT FÜR MATHEMATIK

ÚSTAV MATEMATIKY

ANWENDUNGEN VON QUATERNIONEN IN ROBOTERN KINEMATIK

APLIKACE QUATERNIONŮ V KINEMATICE ROBOTU

Masterarbeit

DIPLOMOVÁ PRÁCE

AUTOR

AUTOR PRÁCE

Diana Doktor

AUFSICHT

VEDOUCÍ PRÁCE

Dok. Ing. Radomil Matoušek, Ph.D.

BRÜNN 2019



Spezifikation Masterarbeit

Abteilung: Institut für Mathematik
Student: Diana Doktor
Studienprogramm: Angewandte Wissenschaften im Ingenieurwesen
Studienrichtung: Mathematische Ingenieurwissenschaften
Aufsicht: Dok. Ing. Radomil Matoušek, Ph.D.
Akademisches Jahr: 2018/19

Gemäß Gesetz Nr. 111/1998 über die Hochschulen und die ABER-Studien- und Prüfungsordnung, Sie haben von der Institutsleitung folgendes Thema für die Masterarbeit erhalten:

Anwendungen von Quaternionen in der Roboterkinematik

Prägnante Beschreibung der Aufgabe:

Diese Arbeit befasst sich mit der Nützlichkeit der Quaternionendarstellung der Kinematik für die Robotik Anwendungen. Sowohl die direkte als auch die inverse Kinematik für einen Roboter manipulator werden mithilfe abgeleitet Quaternionen.

Ziele Masterarbeit:

Die Überprüfung von 6DOF-Robotern und direkten und inversen kinematischen Modellen für das Steuerungsdesign.
Stand der Technik in der Robotersteuerung mittels Quaternion-Algebra.
Simulation eines Roboters (Universal Robot) und praktische Programmieranwendung.

Empfohlene Bibliographie:

SICILIANO, Bruno a KHATIB, Oussama, Hrsg. Springer-Handbuch zur Robotik. 2. Auflage. Berlin: Springer, [2016]. ISBN 978-3-319-32550-7.

Die Frist für die Einreichung der Masterarbeit ergibt sich aus dem Studienplan des Studienjahres 2018/19

In Brünn,

LS

Prof. RNDr. Josef Šlapal, CSc.
Direktor des Instituts

Dok. Ing. Jaroslav Katolický, Ph.D.
FME-Dekan

Abstrakt

Diese Arbeit befasst sich mit der Nützlichkeit der Anwendung von Quaternionen bei der Darstellung der Roboterkinematik. Zunächst wird die Beziehung von Quaternionen zu den bekannteren komplexen Zahlen aufgezeigt und gezeigt, wie sie Drehungen in drei Dimensionen darstellen können. Anschließend werden die dualen Quaternionen eingeführt, um sowohl die dreidimensionale Rotation als auch die Translation darzustellen. Es wird dann verwendet, um die Vorwärts- und Rückwärtskinematik abzuleiten, insbesondere für den Universal Robot UR3, einen 6-DOF-Roboterarm. Abschließend wird eine tatsächliche Anwendung dualer Quaternionen in der Roboterpro

Schlüsselwörter

Quaternionen, duale Quaternionen, Robotik, Universalroboter, UR3, Vorwärtskinematik, Inverse Kinematik, dreidimensionale Rotationen

DOCTOR, DB Anwendungen von Quaternionen in der Roboterkinematik. Brunn: Technische Universität Brunn, Fakultät für Maschinenbau, 2019. 105 S. Betreuer der Diplomarbeit Prof. Dok. Ing. Radomil Matoušek, Ph.D.

Ich erkläre, dass ich diese Diplomarbeit „Anwendungen von Quaternionen in der Roboterkinematik“ unter der Leitung meines Betreuers, Prof. Dr. Dok. Ing. Radomil Matoušek, Ph.D., und Unterstützung von Ing. Roman Parák und unter Verwendung der in der Bibliographie aufgeführten Quellen.

31. Mai 2019

Diana B. Doktor

Ich möchte diese Dissertation zunächst meinem Vater Mateo widmen, der mich zu der Person erzogen hat, die ich heute voller Hoffnungen und Träume bin, und den Menschen hinter dem InterMaths-Programm, die mir mehrere Chancen gegeben haben, diese Träume zu verwirklichen.

Diana B. Doktor

Inhalt

1. Einleitung	13
2 Komplexe Zahlen 2.1	15
Geschichte der komplexen Zahlen	15
2.2 Das komplexe Zahlensystem	16
2.2.1 Definition	16
2.2.2 Komplexe Zahlenmultiplikation	16
2.3 Grafische Darstellung von $z \in \mathbb{C}$	17
2.4 Eigenschaften komplexer Zahlen	18
2.4.1 Norm und Argument von $z \in \mathbb{C}$	18
2.4.2 Komplexes Konjugat von $z \in \mathbb{C}$	18
2.4.3 Algebraische Regeln für komplexe Zahlen	19
2.4.4 Addition komplexer Zahlen	20
2.4.5 Multiplikation komplexer Zahlen	20
2.5 Rotationen in der komplexen Ebene	22
3 Quaternionen 3.1	23
Geschichte der Quaternionen	23
3.2 Das Quaternion-Zahlensystem	26
3.2.1 Definition	26
3.2.2 Quaternion-Darstellungen	26
3.2.3 Quaternion-Multiplikationsgesetze	27
3.3 Eigenschaften von Quaternionen	28
3.3.1 Quaternion-Konjugat.	28
3.3.2 Norm eines Quaternions	28
3.3.3 Einheitsquaternion	28
3.3.4 Umkehrung einer Quaternion	28
3.3.5 Addition und Subtraktion von Quaternionen	29
3.3.6 Multiplikation und Division von Quaternionen	29
3.3.7 Zusammengefasste Eigenschaften von Quaternionen	30
3.4 3D-Rotationen mit Quaternionen	31

3.4.1 3D-Rotationsdarstellungen .				31
3.4.2 Quaternion-Konvertierungen .				36
3.5 Duale Quaternionen .				38
3.5.1 Definition .				38
3.5.2 Umkehrung einer dualen Quaternion .				38
3.5.3 Multiplikation eines dualen Quaternions . . .				39
3.5.4 Transformation durch ein duales Quaternion . .				39
4 Anwendung von Quaternionen auf den Universalroboter UR3				41
4.1 Universalroboter – UR3 . .				41
4.2 Vorwärtskinematik durch duale Quaternionen .				42
4.2.1 Darstellungen der Gelenktransformation .				42
4.2.2 UR3-Vorwärtskinematikberechnung . .				44
4.3 Inverse Kinematik durch duale Quaternionen .				45
4.3.1 Darstellungen von Gelenktransformationen .				45
4.3.2 Gelenktransaktionsprodukte .				46
4.3.3 Transformationen von Basis zu Handgelenksrahmen.				47
4.3.4 Transformation von Schulter-zu-Handgelenk-Rahmen .				53
4.3.5 Gelenkwinkellösungen . .				56
4.4 Praktische Programmieranwendung.				57
4.4.1 Denavit-Hartenberg-Parameter .				57
4.4.2 Python-Programmieranwendung.				58
4.5 Überprüfung der Ergebnisse . .				71
5 Schlussfolgerungen und Empfehlungen				75
5.1 Fazit .				75
5.2 Empfehlung .				77
Anhänge				79
Ein Universalroboter UR3				
Technische Spezifikationen				81
B Transformationsprodukte für Base toWrist 3				85
C Transformationsprodukte für Schulter zu Handgelenk 1				97
D Tabellen- und Abbildungsverzeichnis				101

Kapitel 1

Einführung

Das Konzept der Robotik existierte schon in der Antike. Der Begriff „Roboter“ wurde jedoch erst im 20. Jahrhundert erfunden und hat seinen Ursprung in der Tschechischen Republik. Es wurde zuerst von den Tschechen eingeführt

Dramatiker Karel Capek in seinem Stück „Rossum's Universal Robots (RUR)“, das in den 1920er Jahren in der Stadt Prag uraufgeführt wurde. Der eigentliche Erfinder des Begriffs „Roboter“ war jedoch Karels Bruder Josef Čapek, der ihn vom slawischen Wort „robota“ ableitete, was „Untergebene“ oder „Zwangsarbeit“ bedeutet [2]. Das Stück gewann schnell an Einfluss, so dass der Begriff „Roboter“ in die englische Sprache aufgenommen wurde und seitdem auch in Belletristik und wissenschaftlichen Forschungen verwendet wird. In der Neuzeit bezeichnen wir Roboter als eine von Menschen geschaffene Maschine, die Arbeit verrichten und von Menschen ausgeführte Handlungen nachahmen kann. Wir haben verschiedene Anwendungen von Robotern gesehen, die am häufigsten in der Fertigungsindustrie vorkommen. Die fortschrittlichste Anwendung wäre die Erforschung des Meeres und des Weltraums.

Bei der Modellierung eines Roboters kommt die Roboterkinematik ins Spiel. Die Formulierung der richtigen Kinematik eines Roboters ist wichtig, um sein Verhalten zu analysieren und ihn zu manipulieren. Roboter bestehen aus Gelenken und Gliedern. Die beiden häufigsten Gelenkarten sind Revoluten- und Prismengelenke. Diese Gelenke führen Rotationen bzw. Translationen aus. Wir können die Drehgelenke mit menschlichen Gelenken vergleichen, beispielsweise mit Gelenken in der Schulter, im Ellenbogen, im Handgelenk und im Knie. Ein Roboter wird auch durch seine Freiheitsgrade definiert. Dies entspricht allen möglichen Bewegungen seiner Gelenke. Es wird beschrieben, dass ein planarer Roboter über drei Freiheitsgrade verfügt. Dies liegt daran, dass es sich nur in einer Ebene bewegen kann. Diese sind links oder rechts und entweder vorwärts oder rückwärts. Der andere Freiheitsgrad ist die Drehung, die in einer Ebene durchgeführt werden kann. Wenn ein Roboter auch eine Auf- und Abbewegung ausführen kann, die eine zusätzliche Bewegungsachse darstellt, verfügt er nun über vier Freiheitsgrade. Wenn ein Roboter auch seine Pose (Position) ändern kann

und Ausrichtung) durch Drehung um eine andere Achse in der Ebene, dann erhöht es auch seine Freiheitsgrade. Diese Drehungen werden am häufigsten als Nicken, Gieren und Rollen bezeichnet. Wenn ein Roboter alle diese Verschiebungen und Drehungen ausführen kann, verfügt er über die maximalen sechs Freiheitsgrade.¹

Unter Roboterkinematik versteht man die Untersuchung aller möglichen Bewegungen eines Roboters, ohne die Kräfte zu berücksichtigen, die diese Bewegungen verursachen, wie in [21] dargestellt. Die Kinematik kann in Vorwärts- und Rückwärtskinematik unterteilt werden. Die

Vorwärtskinematik bestimmt die Endeffektorhaltung anhand der Gelenkwinkel. Dies lässt sich leicht ableiten und die Lösung ist immer eindeutig. Allerdings ist ihre Ergänzung, die Inverse Kinematik, im Allgemeinen viel schwieriger abzuleiten, da sie die für eine bestimmte Pose erforderlichen Gelenkwinkel bestimmt. Die Gleichungen sind nichtlinear und können Singularitäten aufweisen. Abgesehen davon kann es auch mehrere Lösungen geben. Unsere Anwendung beinhaltet die Verifizierung dieser Lösungen; Die Ermittlung der optimalen Lösung ist jedoch nicht Gegenstand der Studie.

Die Roboterkinematik wird hauptsächlich durch die Transformation beschrieben, die von jedem Gelenk des Roboters durchgeführt wird. Diese Transformationen werden in Rotationen und Translationen zerlegt. Ein Robotergelenk kann nur eine Translation durchführen, wie es bei prismatischen Gelenken der Fall ist, was bedeutet, dass es die Ausrichtung des Endeffektors nicht beeinflusst, sondern nur seine Position. Es gibt auch ein Gelenk, das beides leisten kann, beispielsweise ein Drehgelenk. Diese Gelenke verändern nicht nur die Ausrichtung des Endeffektors, sondern können auch seine Position verändern. Ein Beispiel wäre, wenn dieses Gelenk mit einer Verbindung mit dem Endeffektor verbunden wäre. Diese Verbindung stellt die Übersetzung zwischen diesem Gelenkrahmen und dem Endeffektorrahmen dar. Somit beeinflusst die Drehung des Gelenks auch die Länge der Translation in Bezug auf den Basiskoordinatenrahmen sowie

Für Rotationen gibt es verschiedene Darstellungsmöglichkeiten. Die gebräuchlichsten Methoden sind Rotationsmatrizen, Eulerwinkel, Achsenwinkeldarstellung und Einheitsquaternionen. Diese Studie konzentriert sich auf Einheitsquaternionen, was uns eine elegante und kompakte Darstellung der Rotation liefert. Während eine 3×3 -Rotationsmatrix 9 Elemente benötigt, um eine Rotation darzustellen, benötigen Quaternionen nur 4. Allerdings wird diese Darstellung in der Robotik immer noch nicht so häufig verwendet wie homogene Transformationen. Das liegt vielleicht an dem geringen Wissen, dass Quaternionen eine Erweiterung komplexer Zahlen sind. Um die Grundidee von Quaternionen zu verstehen, müssen wir daher zunächst das Konzept der komplexen Zahlen verstehen. Der Zweck dieser Forschung besteht darin, das Verständnis der Beziehung zwischen Quaternionen und komplexen Zahlen zu erleichtern und zu zeigen, wie Quaternionen, insbesondere duale Quaternionen, Transformationen im dreidimensionalen Raum am besten darstellen können.

¹Aus dem edX-Kurs: „Robotik: Kinematik und mathematische Grundlagen“ in [1]

Kapitel 2

Komplexe Zahlen

Bevor wir mit dem Studium der Quaternionen beginnen, müssen wir uns an die Grundideen und einige wichtige Eigenschaften komplexer Zahlen erinnern, die für das Verständnis des Konzepts der Quaternionen sehr hilfreich sind. Zunächst erklären wir den Grund für die Formulierung komplexer Zahlen und fahren dann mit der Beschreibung dieses Zahlensystems fort. Anschließend werden wir die wichtigen Eigenschaften der komplexen Zahlen diskutieren, indem wir sie im algebraischen und auch im geometrischen Sinne definieren.

2.1 Geschichte der komplexen Zahlen

Das reelle Zahlensystem $\mathbb{R}$ entstand aus der Suche nach einem System, das die rationalen Zahlen umfasst und Lösungen für verschiedene Polynomgleichungen liefert. Es gibt jedoch einige Polynomgleichungen wie $x^2 + 2x + 2 = 0$, die von keiner reellen Zahl x erfüllt werden können. Mit der quadratischen Formel $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ erhält man formale Ausdrücke für die beiden Lösungen für das System $ax^2 + bx + c = 0$; aber in den Fällen der obigen Gleichungen handelt es sich bei dieser Formel um Quadratwurzeln negativer Zahlen, die zuvor nicht definiert wurden. Im 16. Jahrhundert begann der italienische Mathematiker Geronimo Cardano, über diese Art von Gleichungen nachzudenken und betonte die Notwendigkeit sowohl negativer als auch „komplexer Zahlen“. Er bemerkte, dass diese Ausdrücke, wenn sie durch $\sqrt{-1}$ mit der Eigenschaft $\sqrt{-1} \cdot \sqrt{-1} = -1$ faktorisiert würden, tatsächlich als Lösungen dieser Gleichungen erfüllten. Wir verwenden nun die allgemein akzeptierte Bezeichnung $i = \sqrt{-1}$ für diese Ausdrücke, so dass $i^2 = -1$.

¹Der Inhalt dieses Kapitels, einschließlich der Bilder, bezieht sich alle auf das Buch „Basic Complex Analysis“ von JE Marsden und MJ Hoffman [18]

2.2 Das komplexe Zahlensystem

Wir beginnen mit der Definition einer komplexen Zahl und der Erläuterung der Multiplikationsregel für komplexe Zahlen, die zum Verständnis der Quaternion-Multiplikation notwendig ist.

2.2.1 Definition

Das mit $\mathbb{C}$ bezeichnete System komplexer Zahlen ist die Menge $\mathbb{R}^2$ der xy -Ebene bestehend aus allen geordneten Paaren (x, y) reeller Zahlen zusammen mit den üblichen Regeln der Vektoraddition,

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \quad (2.1)$$

und Skalarmultiplikation mit einer reellen Zahl a ,

$$a(x, y) = (ax, ay) \quad (2.2)$$

und mit der zusätzlichen Operation der komplexen Multiplikation, definiert durch:

$$(x_1, y_1)(x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + y_1x_2) \quad (2.3)$$

Für die Darstellung komplexer Zahlen identifizieren wir zunächst die reellen Zahlen x als Punkte auf der reellen Achse x und y als Punkte auf der imaginären Achse y , wobei der Einheitspunkt $(0,1)$ i darstellt. Nun können wir eine komplexe Zahl als $(x, y) = x + iy$ darstellen, was der Standardschreibweise entspricht. Daher kann ein einzelnes Symbol wie $z = a + ib$ auch zur Angabe einer komplexen Zahl verwendet werden, wobei $\operatorname{Re} z = a$ als Realteil und $\operatorname{Im} z = b$ als Imaginärteil bezeichnet wird. Daher ist eine reelle Zahl eine komplexe Zahl z mit $\operatorname{Im} z = b = 0$.

Wenn wir das Gegenteil haben, $\operatorname{Re} z = a = 0$, dann haben wir eine reine imaginäre Zahl. Die Notation $z \in \mathbb{C}$ impliziert, dass z zur Menge der komplexen Zahlen gehört.

2.2.2 Komplexe Zahlenmultiplikation

Um die Regel für die Multiplikation komplexer Zahlen abzuleiten,

$$(a, b)(c, d) = (ac - bd, ad + bc) \quad (2.4)$$

Wir verwenden die Standardschreibweisen für die komplexen Zahlen (a, b) und (c, d) und wenden dann die Eigenschaft an, dass $i^2 = -1$, wir erhalten:

$$\begin{aligned} (a + ib)(c + id) &= ac + iad + ibc + i^2bd \\ &= ac - bd + i(ad + bc) \end{aligned} \quad (2.5)$$

2.3 Grafische Darstellung von $z \in \mathbb{C}$

Wie wir in der Definition angegeben haben, kann man sich eine komplexe Zahl auch als zweidimensionalen Vektor in $\mathbb{R}^2$ vom Ursprung zu einem Punkt in der xy -Ebene vorstellen, dessen Koordinaten durch den Real- und den Imaginärteil der komplexen Zahl gegeben sind. Siehe Abbildung 4.6.

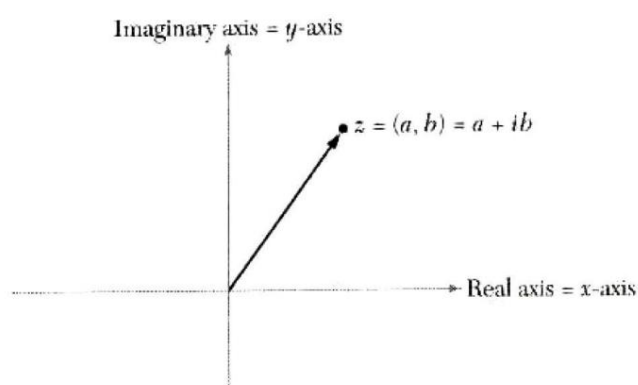


Abbildung 2.1: Vektordarstellung von $z = a + ib$ mit $\operatorname{Re} z = a$ und $\operatorname{Im} z = b$.

Polarkoordinaten- und Exponentialdarstellung. Wir schreiben eine komplexe Zahl in Polarkoordinatenform, indem wir die Länge r des Vektors $(a, b) = a + ib$ durch $r = \sqrt{a^2 + b^2}$ nehmen. Dann nehmen wir an, dass der Vektor einen positiven Winkel θ zur realen Achse bildet, wobei $0 \leq \theta < 2\pi$. Siehe Abbildung 2.2.

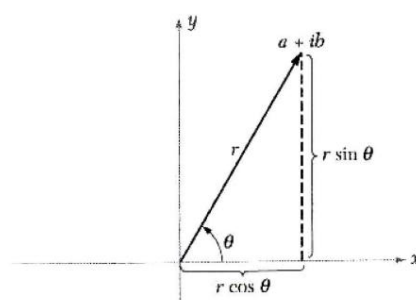


Abbildung 2.2: Polarkoordinatendarstellung der komplexen Zahl $z = a + ib$ mit Länge r und Winkel θ .

Durch SOHCAHTOA erhalten wir $\tan \theta = \frac{b}{a}$, wobei $a = r \cos \theta$ und $b = r \sin \theta$, also gilt:

$$z = a + ib = r \cos \theta + i(r \sin \theta) = r(\cos \theta + i \sin \theta) = re^{i\theta} \quad (2.6)$$

²Wir haben die Exponentialform mithilfe der Euler-Formel abgeleitet, $e^{i\theta} = \cos \theta + i \sin \theta$.

2.4 Eigenschaften komplexer Zahlen

In diesem Abschnitt stellen wir alle grundlegenden und wichtigen Eigenschaften komplexer Zahlen vor, die für das Verständnis der entsprechenden Eigenschaften von Quaternionen notwendig sind.

2.4.1 Norm und Argument von $z \in \mathbb{C}$

In Abbildung 2.2 hat der Vektor, der die komplexe Zahl $z = a + ib$ darstellt, die Länge r , die auch als $|z|$ bezeichnet werden kann und heißt Norm von z . Die formale Berechnung ergibt sich aus:

$$|z| = \sqrt{a^2 + b^2} \quad (2.7)$$

Der Winkel φ wird als Argument von z bezeichnet und mit $\arg z = \varphi$ bezeichnet. Die formale Berechnung ergibt sich aus:

$$\varphi = \tan^{-1} \frac{b}{a} \quad (2.8)$$

2.4.2 Komplexes Konjugat von $z \in \mathbb{C}$

Gegeben eine komplexe Zahl $z = a + ib$, dann ist $\bar{z}$, das komplex konjugiert von z genannt wird, definiert als $\bar{z} = a - ib$. Geometrisch lässt sich dies als Spiegelung in der reellen Achse x darstellen, die in Abbildung 2.3 dargestellt ist.

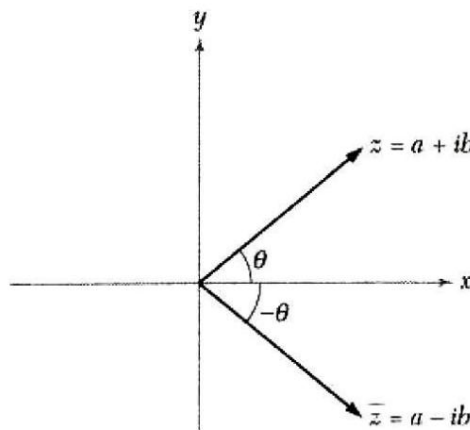


Abbildung 2.3: Komplexe Konjugation.

Daher kann die komplexe Konjugation in Polarkoordinatenform auch durch die Verwendung derselben Größe, aber entgegengesetzten Winkels dargestellt werden.

2.4.3 Algebraische Regeln für komplexe Zahlen

In diesem Abschnitt werden wir zeigen, dass alle üblichen algebraischen Regeln zur Manipulation reeller Zahlen auch auf komplexe Zahlen angewendet werden können.

Additionsregeln

Komplexe Zahlen z und w erfüllen die folgenden Additionsregeln:

1. $z + w = w + z$
2. $z + (w + s) = (z + w) + s$
3. $z + 0 = z$
4. $z + (\bar{z}) = 0$

Multiplikationsregeln

Komplexe Zahlen z und w erfüllen die folgenden Multiplikationsregeln:

1. $zw = wz$
2. $(zw)s = z(ws)$
3. $1z = z$
4. $z(\bar{z}) = 1$ für $z \neq 0$

Wir können nach dem multiplikativen Inversen z auflösen $\bar{z}z = z\bar{z} = a^2 + b^2$ durch:

$$\frac{1}{z} = \frac{a}{a^2 + b^2} \quad \text{und} \quad \frac{1}{\bar{z}} = \frac{b}{a^2 + b^2} \quad (2.9)$$

Daher können wir für $z = a + ib \neq 0$ schreiben:

$$\frac{1}{z} = \frac{a}{a^2 + b^2} - \frac{ib}{a^2 + b^2} \quad (2.10)$$

Verteilungsrecht

Komplexe Zahlen z , w und s erfüllen die folgenden Verteilungsregeln:

$$z(w + s) = zw + zs \quad (2.11)$$

Es ist hier wichtig zu beachten, dass komplexe Zahlen ebenso wie die reellen Zahlen auch die assoziativen, kommutativen und distributiven Eigenschaften der Addition und Multiplikation erfüllen. Wir werden diesen Fall mit dem Fall von Quaternion-Operationen vergleichen, die im nächsten Kapitel besprochen werden.

2.4.4 Addition komplexer Zahlen

Wir können die Addition komplexer Zahlen als Addition zweidimensionaler Vektoren darstellen, die die beiden komplexen Zahlen darstellen. Dieser Vorgang ist in Abbildung 2.4 dargestellt.

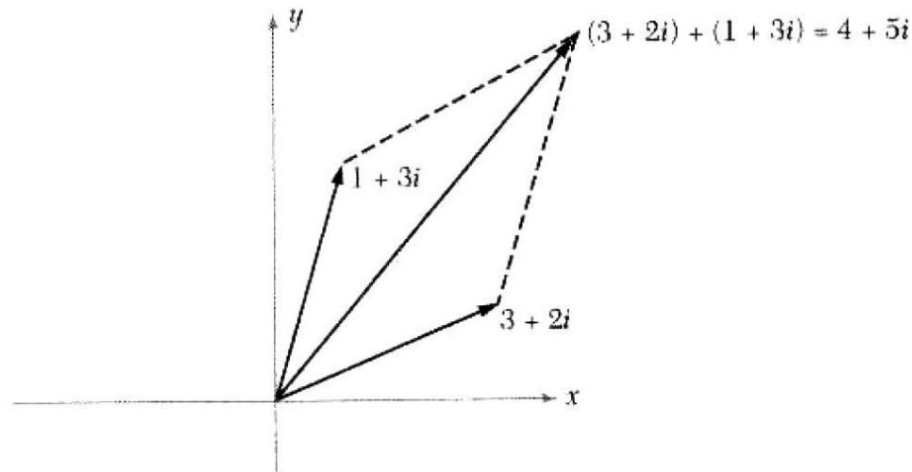


Abbildung 2.4: Addition der komplexen Zahlen $3 + 2i$ und $1 + 3i$.

Dieselbe Darstellung kann auch für die Subtraktion komplexer Zahlen verwendet werden, wobei auch eine Addition durch Addition des ersten Vektors zur Spiegelung im Ursprung des anderen Vektors möglich ist.

2.4.5 Multiplikation komplexer Zahlen

Die Darstellung einer komplexen Zahl in Polarkoordinaten- oder Exponentialform hilft, das Produkt zweier komplexer Zahlen im geometrischen Sinne zu verstehen.

Wenn wir $z_1 = r_1(\cos \varphi_1 + i \sin \varphi_1)$ und $z_2 = r_2(\cos \varphi_2 + i \sin \varphi_2)$ setzen, gilt:

$$\begin{aligned} z_1 z_2 &= r_1(\cos \varphi_1 + i \sin \varphi_1) \cdot r_2(\cos \varphi_2 + i \sin \varphi_2) = \\ &= r_1 r_2 [(\cos \varphi_1 \cdot \cos \varphi_2 - \sin \varphi_1 \cdot \sin \varphi_2) + i(\cos \varphi_1 \cdot \sin \varphi_2 + \sin \varphi_1 \cdot \cos \varphi_2)] \\ &= r_1 r_2 [\cos(\varphi_1 + \varphi_2) + i \sin(\varphi_1 + \varphi_2)] = r_1 r_2 e^{i(\varphi_1 + \varphi_2)} \quad (2.12) \end{aligned}$$

Beachten Sie, dass wir die folgenden trigonometrischen Formeln für die Summe und Differenz der Produkte der Kosinus- und Sinuswerte der Winkel angewendet haben:

$$\cos(\varphi_1 + \varphi_2) = \cos \varphi_1 \cos \varphi_2 - \sin \varphi_1 \sin \varphi_2 \quad (2.13)$$

$$\sin(\varphi_1 + \varphi_2) = \sin \varphi_1 \cos \varphi_2 + \cos \varphi_1 \sin \varphi_2 \quad (2.14)$$

2.4. Eigenschaften komplexer Zahlen

Vorschlag. Für jede komplexe Zahl z_1 und z_2 schlagen wir Folgendes vor:

$$|z_1 z_2| = |z_1| \cdot |z_2| \text{ und } \arg(z_1 z_2) = \arg z_1 + \arg z_2 \pmod{2\pi} \quad (2.15)$$

Dies bedeutet, dass das Produkt zweier komplexer Zahlen eine komplexe Zahl ist, deren Länge den Produkten der Längen der beiden multiplizierten komplexen Zahlen entspricht und deren Argument der Summe der Argumente dieser beiden komplexen Zahlen entspricht. Dieses Produkt kann in Abbildung 2.5 und ein Beispiel in 2.6 dargestellt werden.

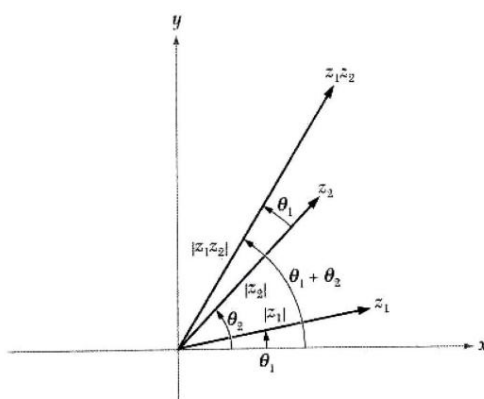


Abbildung 2.5: Multiplikation der komplexen Zahlen z_1 und z_2 .

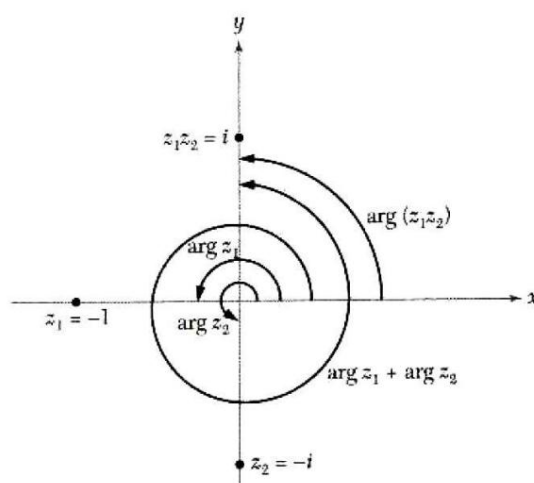


Abbildung 2.6: Multiplikation der komplexen Zahlen z_1 und z_2 .

2.5 Rotationen in der komplexen Ebene

Die Multiplikation komplexer Zahlen kann auch durch eine Abbildung γ_z analysiert werden, die die Multiplikation mit z darstellt, sodass $\gamma_z : \mathbb{C} \rightarrow \mathbb{C}$ und durch $\gamma_z(w) = wz$ definiert ist. Nach dem Satz besteht die Wirkung dieser Karte darin, eine komplexe Zahl w um einen Winkel, der gleich $\arg z$ ist, entgegen dem Uhrzeigersinn zu drehen und ihre Länge um den Faktor $|z|$ zu strecken. Dies ist in Abbildung 2.7 und auch im vorherigen Beispiel in Abbildung 2.6 zu sehen.

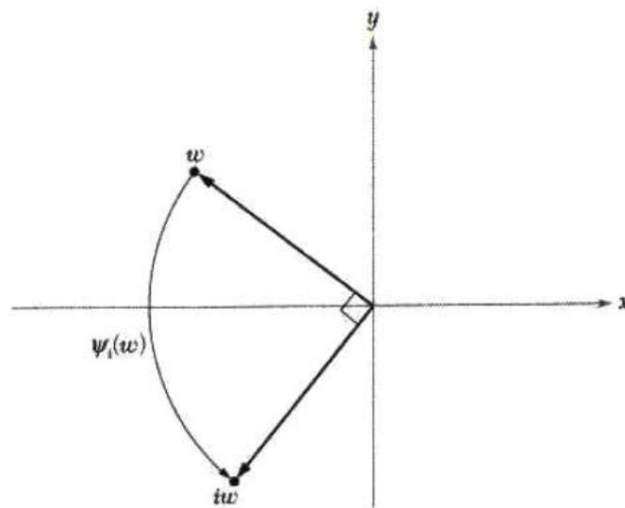


Abbildung 2.7: Multiplikation von wz mit $z = i$.

Im nächsten Kapitel beginnen wir mit der Diskussion von Quaternionen und wie ihre Multiplikation auch Drehungen in drei Dimensionen darstellen kann.

Kapitel 3

Quaternionen

Dieses Kapitel befasst sich mit der Hauptdiskussion von Quaternionen und ihren grundlegenden Eigenschaften. Wir beginnen mit einer detaillierten Beschreibung der dramatischen Entdeckung von Quaternionen und wie es hat im Laufe seiner Zeit die Welt der Mathematik erschüttert. Anschließend werden wir mit der Definition und den wichtigen Eigenschaften von Quaternionen fortfahren wird uns ein grundlegendes Verständnis dieses Zahlensystems vermitteln. Es ist auch wichtig um zu zeigen, wie Quaternionen Rotationen in drei Dimensionen darstellen können und wie es sich auf andere Rotationsdarstellungen bezieht. Abschließend werden wir diskutieren über die dualen Quaternionen und wie sie in der Robotik verwendet werden.

3.1 Geschichte der Quaternionen

Wie wir im vorherigen Kapitel gesehen haben, modellieren komplexe Zahlen den zweidimensionalen Raum, indem sie einen entsprechenden Real- und Imaginärteil haben zur xy-Ebene, in der die Multiplikation und Addition dieser Zahlen noch erfolgt im selben Raum auftreten. Seit Jahren der irische Mathematiker William Rowan Hamilton hatte versucht, einen dreidimensionalen Raum zu modellieren, der dies bietet gleiche Struktur. Er versuchte es real und zweidimensional darzustellen Imaginärteil, aber es entsprach nicht der Struktur, die er erreichen wollte. Sein Sohn William Edward Hamilton fragte ihn oft nach seinen Fortschritten in der Multiplikation von Triolen, auf die er antworten würde, dass er nur addieren könne oder subtrahiere sie, aber multipliziere sie niemals. [24]

Der berühmteste Akt des mathematischen Vandalismus. Der Durchbruch kam am 16. Oktober 1843, als Hamilton den Fluss entlang spazierte Brougham Bridge (heute Broom Bridge) in Dublin. Er hatte einen Geistesblitz und konnte nicht anders, als die folgenden Gleichungen in die Brücke einzuritzen:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (3.1)$$

Kapitel 3. Quaternionen

In seinem Tagebuch [10] beschrieb Hamilton diesen Moment wie folgt:

„Am 16. Oktober – der zufällig ein Montag und ein Ratstag der Royal Irish Academy war – kam ich herein, um einer Vorlesung beizuwohnen, und Ihre Mutter ging mit mir den Royal Canal entlang, zu dem sie war vielleicht gefahren; Und obwohl sie hin und wieder mit mir redete, ging in meinem Kopf doch eine unterschwellige Gedankenströmung durch, die schließlich zu einem Ergebnis führte, dessen Bedeutung nicht übertrieben ist, um zu sagen, dass ich sofort die Wichtigkeit spürte. Ein Stromkreis schien sich zu schließen; Und ein Funke blitzte auf, der Vorbote (wie ich sofort voraussah) für viele lange Jahre, die noch vor mir liegen würden, in denen ich bestimmt und zielstrebig gedacht und gearbeitet habe, von mir selbst, wenn ich verschont bliebe, und auf jeden Fall von Seiten anderer, wenn es mir überhaupt gestattet werden sollte leben lange genug, um die Entdeckung deutlich zu kommunizieren. Auch ich konnte dem Impuls – so unphilosophisch er auch gewesen sein mag – nicht widerstehen, mit einem Messer in einen Stein der Brougham Bridge, als wir daran vorbeikamen, die Grundformel mit den Symbolen i, j, k zu schneiden; nämlich $i^2 = j^2 = k^2 = ijk = -1$, das die Lösung des Problems enthält.“

$$i^2 = j^2 = k^2 = ijk = -1$$

Seine Entdeckung dieser Multiplikationsgesetze war auch ein entscheidender Moment in der Geschichte der Algebra. In diesem Moment erkannte er, dass eine vierte Dimension erforderlich war, und prägte später den Begriff Quaternion, um den realen Raum darzustellen, der von den Elementen $1, i, j$ und k aufgespannt wird, die den oben genannten Multiplikationsgesetzen unterliegen.

Seine Schnitzereien sind mit der Zeit abgenutzt, aber als Ersatz und zur Erinnerung an die Entdeckung in der Mathematikgeschichte wurde an der Brücke eine Gedenktafel angebracht.



Abbildung 3.1: Ein Porträt von Sir William Rowan Hamilton¹

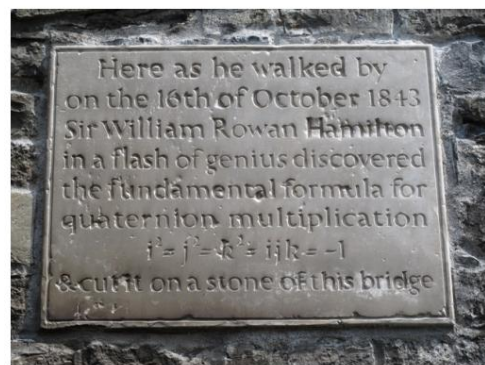


Abbildung 3.2: Gedenktafel zur Erinnerung an die Entdeckung der Quaternionen.²

12Gemäß den genauen Worten von Hamilton [10]:

„Ein Quaternion kann symbolisch als ein Quadrino-Mial-Ausdruck der Form $q = w + ix + jy + kz$ definiert werden, wobei w, x, y, z vier Skalare oder gewöhnliche algebraische Größen sind, während i, j und k sind drei neue Symbole, die allen in der obigen Formel enthaltenen Gesetzen gehorchen und daher nicht allen üblichen Regeln der Algebra unterliegen.“

Hamilton definierte die Quaternionen als Quotient zweier Vektoren. Als Quaternionen ein Zahlensystem sind, das die übliche kommutative Regel der Multiplikation nicht erfüllt, stieß es in der mathematischen Welt zwangsläufig auf Widerstand, als es erstmals von Hamilton entdeckt und erklärt wurde. Die Rivalität zwischen denen, die Quaternionen bevorzugen, und denen, die Vektoren als physikalische Notationen bevorzugen, entbrannte in einem Krieg zwischen denen, die auf der Seite der Quaternionen waren, und denen, die auf der Seite der Vektornotation standen, wobei einige sogar Quaternionen als reines Übel bezeichneten.

Mit der Einführung von Quaternionen eröffnete es neue algebraische Möglichkeiten, wie etwa die Wiederentdeckung von Oktaven oder Oktonionen, einer Algebra in acht Dimensionen, durch John T. Graves durch die erfolgreiche Verallgemeinerung von Quaternionen. Ein weiteres bekanntes Ergebnis von Quaternionen sind ihre Verallgemeinerungen auf höhere Dimensionen durch William Clifford, der eine Methode zum Aufbau von Algebren aus quadratischen Formen entwickelte, so dass die Hamilton-Quaternionen als diese Algebra entstehen, die besser als Clifford-Algebra bekannt ist.

Ab Mitte der 1880er Jahre begann man, Quaternionen durch die Vektoranalyse zu verdrängen, die dieselben Phänomene wie Quaternionen beschrieb, jedoch konzeptionell einfacher und notatorisch sauberer. Aufgrund ihrer Nützlichkeit bei der Beschreibung räumlicher Rotationen erlebten Quaternionen jedoch seit dem späten 20. Jahrhundert ein Revival. Dies liegt daran, dass Quaternionen eine elegante Möglichkeit bieten, Drehungen zu beschreiben und zu berechnen, die kompakter und rechnerisch effizienter ist und Gimbal Lock vermeidet, der auftritt, wenn bei der Anwendung bestimmter Drehungen ein Freiheitsgrad verloren geht. Insbesondere Euler-Winkel sind anfällig für dieses mechanische Problem.

Aus diesen Gründen werden Quaternionen auch heute noch in der Computergrafik, Computer Vision, Robotik, Kontrolltheorie, Signalverarbeitung, Lagekontrolle, Quantenmechanik und vielen anderen Bereichen verwendet, in denen es um die Arbeit im dreidimensionalen Raum geht.

1https://en.wikipedia.org/wiki/William_Rowan_Hamilton

2<http://www.irelandtravelkit.com/broome-bridge-and-sir-hamiltons-eureka-moment-cabra-co-dublin/broombridge-plaque-bdolan-2010-800-Label/>

3.2 Das Quaternion-Zahlensystem

Wir beginnen mit der Definition der Quaternion und ihren erweiterten Multiplikationsgesetzen, wie auch in [13] dargelegt.

3.2.1 Definition

Die Quaternionen sind die Menge, die definiert ist als:

$$H = \{w + ix + jy + kz \mid w, x, y, z \in \mathbb{R}\} \quad (3.2)$$

Das ist die Menge aller Zahlen, die in der Form vorliegen können,

$$q = w + ix + jy + kz \quad (3.3)$$

wobei w, x, y und z reelle Zahlen sind und i, j und k die sogenannten fundamentalen Quaternioneinheiten sind, die die drei imaginären Dimensionen darstellen.

3.2.2 Quaternion-Darstellungen

Standard-Quaternionalform. Eine Quaternion wird im Allgemeinen in ihrer standardmäßigen quaternionalen Form dargestellt, geschrieben als:

$$q = q_0 + q_1i + q_2j + q_3k \quad (3.4),$$

in dem alle Variablen den gleichen Interpretationen wie in der Definition entsprechen.

Skalar- und Vektorform. Abgesehen von der Standardform kann ein Quaternion auch so beschrieben werden, dass es aus einem Skalarteil und einem Vektorteil besteht, wobei im obigen Fall der Quaternionteil $q_1i + q_2j + q_3k$ den Vektorteil und q_0 den Skalarteil darstellt. Dies kann in der sogenannten Skalarvektorform dargestellt werden, geschrieben als:

$$q = (q_0, \mathbf{q}) \quad (3.5)$$

wobei q_0 als Skalarteil bezeichnet wird und der Vektor $\mathbf{q}$ den dreidimensionalen Vektor darstellt, der durch den Imaginärteil der Quaternion gebildet wird, und als Vektorteil bezeichnet wird.

Eine Quaternion ohne Realteil wird als reine Quaternion oder Vektorquaternion bezeichnet, und eine Quaternion ohne Imaginärteile ist dann äquivalent zu ihrer entsprechenden reellen Zahl, was bedeutet, dass die reellen und komplexen Zahlen eine Teilmenge der Quaternionen sind. ($\mathbb{R} \subset \mathbb{C} \subset H$)

3.2. Das Quaternion-Zahlensystem

3.2.3 Quaternion-Multiplikationsgesetze

Diese Quaternioneeinheiten i , j und k erfüllen, wie bereits erwähnt, Folgendes Multiplikationsregel:

$$i^2 = j^2 = k^2 = -1 \quad (3.6)$$

welches zu den folgenden Multiplikationsgesetzen erweitert wird:

$$ij = k = -ji \quad (3.7)$$

$$jk = i = -kj \quad (3.8)$$

$$ki = j = -ik \quad (3.9)$$

Diese Gesetze sind in Tabelle 4.1 zusammengefasst.

$\times$	1	i	j	k
1	1	i	j	k
i	$-i$	-1	k	$-j$
j	$-j$	$-k$	-1	i
k	$-k$	i	j	-1

Tabelle 3.1: Quaternion-Multiplikationstabelle

Diese können auch im Diagramm in Abbildung 3.3 visualisiert werden.

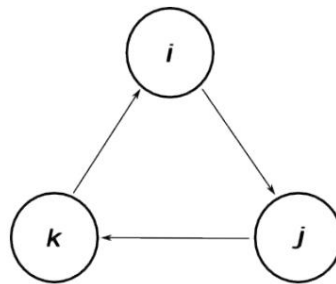


Abbildung 3.3: Quaternion-Multiplikationsdiagramm

Wie wir aus der Tabelle und dem Diagramm oben sehen können, ist die Multiplikation von Quaternioneeinheiten nicht kommutativ ($ij \neq ji$). Wenn die Multiplikation dieser Einheiten gegen den Fluss im Diagramm erfolgt, ist das Produkt das Negative des entsprechenden Produkts im Normalfluss. Dies wird besser verstanden, wenn wir die Darstellung von Rotationen durch Quaternionen besprochen haben.

³<http://hforsten.com/quaternion-roots-of-modern-vector-calculus.html>

3.3 Eigenschaften von Quaternionen

In diesem Abschnitt stellen wir alle grundlegenden und wichtigen Eigenschaften eines Quaternions mit der Formel $q = q_0 + q_1i + q_2j + q_3k$ vor, wie auch in [17] angegeben.

3.3.1 Quaternion-Konjugat

Das Quaternion-Konjugat von q ergibt sich durch Negation seines Vektorteils, dargestellt als:

$$Q^\vee = q_0 - q_1i - q_2j - q_3k = (q_0, -\mathbf{q}) \quad (3.10)$$

3.3.2 Norm eines Quaternions

Die Quaternionnorm, auch Magnitude genannt, ist die Länge des Quaternions vom Ursprung. Es ist definiert durch:

$$\|q\| = \sqrt{q q^\vee} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.11)$$

3.3.3 Einheitsquaternion

Eine Quaternion wird als Einheitsquaternion bezeichnet, wenn sie eine Norm von 1 hat, d. h.

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.12)$$

Normalisierung. Jede Quaternion kann normalisiert werden, um sie in eine Einheitsquaternion umzuwandeln. Dies geschieht durch Division der Quaternion durch ihre Größe oder Norm, wie zum Beispiel:

$$\hat{q} = \frac{q}{\|q\|} = \frac{q_0 + q_1i + q_2j + q_3k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \quad (3.13)$$

3.3.4 Umkehrung einer Quaternion

Der Kehrwert (Kehrwert) einer Quaternion kann berechnet werden durch:

$$q^{-1} = \frac{Q^\vee}{\|q\|^2} = \frac{q_0 - q_1i - q_2j - q_3k}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.14)$$

Dies kann anhand der Definition der Norm ($\|q\|^2 = q q^\vee$) überprüft werden.

$$q q^{-1} = q \frac{Q^\vee}{\|q\|^2} = \frac{q Q^\vee}{\|q\|^2} = \frac{q q^\vee}{\|q\|^2} = 1 \quad (3.15)$$

3.3.5 Addition und Subtraktion von Quaternionen

Die Addition von Quaternionen erfolgt nach der gleichen Regel wie bei den komplexen Zahlen, nämlich komponentenweise. Wenn wir die gegebene Quaternion q und eine andere Quaternion p als gegeben betrachten,

$$p = p_0 + p_1i + p_2j + p_3k \quad (3.16)$$

Dann ist ihre Summe gegeben durch:

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \quad (3.17)$$

Die gleiche Regel wird auch auf die Subtraktion von Quaternionen angewendet,

$$p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k \quad \text{sodass (3.18)}$$

3.3.6 Multiplikation und Division von Quaternionen

Das Produkt zweier Quaternionen muss den eingangs genannten Grundgesetzen sowie seiner Nichtkommutativitätseigenschaft für die Multiplikation genügen. Somit ist das Produkt der Quaternionen p und q gegeben durch:

$$\begin{aligned} pq &= (p_0 + p_1i + p_2j + p_3k)(q_0 + q_1i + q_2j + q_3k) \quad (3.19) \\ &= p_0q_0 + (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k \\ &\quad + (p_1q_2)ij + (p_2q_1)ji + (p_2q_3)jk + (p_3q_2)kj + (p_3q_1)ki + (p_1q_3)ik + (p_1q_1)i^2 = \\ &\quad p_0q_0 + \quad^2 + (p_2q_2)j^2 + (p_3q_3)k^2 \\ &\quad (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k + (p_1q_2)(ik) + \\ &\quad (p_2q_1)(jk) + (p_2q_3)(i) + (p_3q_2)(ji) + (p_3q_1)(j) + (p_1q_3)(ji) + (p_1q_2)(j) + (p_2q_2)(ji) + \\ &\quad (p_3q_3)(ji) = p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 + \\ &\quad (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)i \\ &\quad + (p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3)j + (p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1)k \end{aligned}$$

Dies ist eine komplizierte Methode, das Produkt zweier Quaternionen zu bilden. Es gibt einen viel einfacheren Weg, der darin besteht, die Skalar- und Vektorform einer Quaternion zu verwenden und das Skalar- und Kreuzprodukt für Vektoren anzuwenden. Dies geschieht wie

$$\text{folgt: } pq = (p_0, \mathbf{p})(q_0, \mathbf{q}) = p_0q_0 - \mathbf{p} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} \quad (3.20)$$

Für die Division zwischen Quaternionen kann dies erreicht werden, indem die multiplikative Umkehrung des Divisors gebildet wird, sodass

$$\frac{P}{Q} = pq^{-1} \quad (3.21)$$

und Anwenden der obigen Methode auf die Multiplikation zweier Quaternionen.

⁴Auf diese Addition und Multiplikation von Quaternionen wird verwiesen in [6]

3.3.7 Zusammengefasste Eigenschaften von Quaternionen

Hier sind die zusammengefassten Eigenschaften von Quaternionen aufgeführt.

Zusatzeigenschaften

Die Quaternionen q , p und r erfüllen die folgenden Additionseigenschaften:

- Abschluss: $p + q \in H$
- Assoziativität: $p + (q + r) = (p + q) + r$
- Kommutativität: $p + q = q + p$
- Identität: $p + 0 = 0 + p = p$
- Summe: $p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k$
- Differenz: $p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k$

Multiplikationseigenschaften

Die Quaternionen q , p und r erfüllen die folgenden Multiplikationseigenschaften:

- Abschluss: $pq \in H$
- Assoziativität: $(pq)r = p(qr)$
- Nichtkommutativität: $pq \neq qp$
- Distributivität: $p(q + r) = pq + pr$
- Invers: Wenn $q \neq 0$, dann $\exists q^{-1} \in H$ mit $qq^{-1} = q^{-1}q = 1$
- Produkt: $pq = p_0q_0 - p \cdot q + p_0q + q_0p + p \times q$
- Division: $p/q = pq^{-1}$

Konjugiert: $\bar{q} = q_0 - q_1i - q_2j - q_3k$

Norm: $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$

Einheitsquaternion: $\|q\| = 1$ und $\bar{q}p = 0$, $\|p\| = 1$ ist eine Einheitsquaternion

Invers: $q^{-1} = \frac{\bar{q}}{\|q\|^2}$ für eine Einheitsquaternion u , $u \|q\|^2 = \bar{q}$

Skalarprodukt: $q \cdot p = q_0p_0 + q_1p_1 + q_2p_2 + q_3p_3$

3.4 3D-Rotationen mit Quaternionen

In diesem Abschnitt zeigen wir, wie Einheitsquaternionen Rotationen in drei Dimensionen darstellen können und welche Beziehung sie zu anderen Rotationsdarstellungen haben.

3.4.1 3D-Rotationsdarstellungen

Das Konzept, eine Rotation in drei Dimensionen als mathematische Formulierung auszudrücken, erstreckt sich auf die klassische Mechanik, wo Rotationskinematik die Wissenschaft ist, die reine Rotationsbewegung eines Objekts mit Zahlen zu beschreiben.

Nach dem Rotationssatz von Euler wird eine allgemeine Verschiebung eines starren Körpers mit einem festen Punkt durch eine Drehung um eine Achse beschrieben. Darüber hinaus kann diese Rotation durch mindestens drei Parameter eindeutig beschrieben werden. Es gibt verschiedene Möglichkeiten, 3D-Rotationen darzustellen, wobei einige mehr als nur drei Parameter benötigen, selbst wenn alle nur drei Freiheitsgrade haben. Die am häufigsten verwendeten Rotationsdarstellungen werden hier detailliert beschrieben.

Darstellung der Rotationsmatrix

Wir wissen, dass zweidimensionale Rotationen durch die Matrix dargestellt werden können,

$$R = \begin{pmatrix} \cos \tilde{\gamma} & \sin \tilde{\gamma} \\ -\sin \tilde{\gamma} & \cos \tilde{\gamma} \end{pmatrix} \quad (3.22)$$

wobei $\tilde{\gamma}$ der Rotationswinkel um den Ursprung ist. Ein einziger Winkelparameter bedeutet auch, dass Ebenenrotationen nur einen Freiheitsgrad haben.

In drei Dimensionen haben wir es mit Drehungen um die Hauptachsen x , y und z zu tun. Eine Drehung um die x -Achse entspricht einer Drehung in der yz -Ebene. Andererseits entspricht eine Drehung um die y -Achse einer Drehung in der xz -Ebene und eine Drehung um die z -Achse einer Drehung in der xy -Ebene.

Diese Rotationen werden durch die folgenden Matrizen dargestellt:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \tilde{\gamma} & \sin \tilde{\gamma} \\ 0 & -\sin \tilde{\gamma} & \cos \tilde{\gamma} \end{pmatrix} \quad (3.23)$$

⁵Dieser Unterabschnitt ist eine Zusammenstellung von Inhalten aus den folgenden Quellen: [5][6][8][9][17]

Kapitel 3. Quaternionen

$$R_y = \begin{pmatrix} \cos \tilde{y} & 0 & \sin \tilde{y} \\ 0 & 1 & 0 \\ -\sin \tilde{y} & 0 & \cos \tilde{y} \end{pmatrix} \quad (3.24)$$

$$R_z = \begin{pmatrix} \cos \tilde{y} & \sin \tilde{y} & 0 \\ -\sin \tilde{y} & \cos \tilde{y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.25)$$

Im Allgemeinen betrachten wir einen starren Körper mit drei daran befestigten orthogonalen Einheitsvektoren (auch als Basis bezeichnet). Durch Angabe der Koordinaten dieser Basis in ihrer aktuellen gedrehten Position in Bezug auf die nicht gedrehten Referenzkoordinatenachsen wird die auf sie angewendete Drehung beschrieben. Diese drei Einheitsvektoren r_1 , r_2 und r_3 , bestehend aus jeweils 3 Koordinaten, ergeben insgesamt 9 Parameter. Diese Parameter bilden eine 3x3-Matrix, die als Rotationsmatrix bezeichnet wird.

Die Elemente einer Rotationsmatrix sind wie folgt definiert:

$$R = [r_1 \ r_2 \ r_3] \quad (3.26)$$

$$= \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad (3.27)$$

Eine Rotationsmatrix hat folgende besondere Eigenschaften:

1. Die Spalten einer Rotationsmatrix sind orthogonale Einheitsvektoren.
2. Die Zeilen einer Rotationsmatrix sind ebenfalls orthogonale Einheitsvektoren.
3. Die Transponierte einer Rotationsmatrix ist ihre Umkehrung, $R^T = R^{-1}$. Daher, $R^T R = R R^T = I$.
4. Die Determinante einer Rotationsmatrix ist +1.

Eine allgemeine dreidimensionale Rotation wird erreicht, indem diese Rotationsmatrizen durch Matrixmultiplikation zu einer einzigen Matrix kombiniert werden. Da Matrixmultiplikationen nicht kommutativ sind, ist es wichtig, die Reihenfolge der Rotationen zu beachten, wie zum Beispiel:

$$R_x R_y R_z = R_z R_y R_x \quad (3.28)$$

Dies bedeutet, dass eine Drehung zuerst um die x-Achse, dann um die y-Achse und dann um die z-Achse nicht die gleiche Drehung erzeugt wie eine Drehung zuerst um die z-Achse, dann um die y-Achse und dann um die x-Achse. Diese Eigenschaft stimmt darin überein, dass Drehungen auch nicht kommutativ sind. Die endgültige Matrix enthält drei Winkelparameter aus jeder Rotationsmatrix, was drei Freiheitsgraden entspricht (diese Winkelparameter werden oft als Euler-Winkel $[\tilde{y}, \tilde{\theta}, \tilde{\psi}]$ bezeichnet).

3.4. 3D-Rotationen mit Quaternionen

Die Durchführung einer Matrix-Vektor-Multiplikation zwischen der Rotationsmatrix und einem Vektor im Raum führt zu einer Rotation dieses Vektors im Raum, während seine Länge erhalten bleibt. Dies kann ausgedrückt werden als:

$$\vec{z}' = R \vec{z} \quad (3.29)$$

$$\begin{pmatrix} z'_1 \\ z'_2 \\ z'_3 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

wo $\vec{z}$ ist der Vektor z , der durch die Rotationsmatrix R gedreht wird.

Die Möglichkeit, mehrere Rotationen in einer einzigen Matrix zu kombinieren und die Rotation einfach auf einen Vektor anzuwenden, macht die Rotationsmatrix zu einer nützlichen und beliebten Methode zur Darstellung von Rotationen, auch wenn sie weniger präzise als andere Darstellungen ist.

Euler-Winkel-Darstellung

Euler-Winkel sind historisch gesehen die beliebteste Rotationsdarstellung, bei der eine allgemeine Rotation als eine Folge von Rotationen um drei orthogonale Koordinatenachsen, nämlich die x-, y- und z-Achse, beschrieben wird. Die Idee hinter Euler-Winkeln besteht darin, eine Drehung in drei einfachere Drehungen $[\gamma, \beta, \alpha]$ aufzuteilen, die Drehungen auf den Hauptachsen entsprechen und üblicherweise als Nicken, Gieren und Rollen bezeichnet werden.

Die Definition der Euler-Winkel ist nicht eindeutig und die Konventionen hängen vom Bezugssystem, den Achsen und der Reihenfolge ab, in der die Drehungen angewendet werden. Daher kann für einen globalen Bezugsrahmen eine Drehung von γ auf der x-Achse, gefolgt von einer Drehung von β auf der festen y-Achse und dann einer Drehung von α auf der festen z-Achse in der Rotationsmatrix dargestellt werden Form durch Vormultiplikation der nachfolgenden Rotationen (für den lokalen Bezugsrahmen erfolgt die Nachmultiplikation).

Dies kann ausgedrückt werden als: (3.30)

$$R_{xyz} = R_z(\alpha) R_y(\beta) R_x(\gamma) = \begin{pmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \beta & \cos \alpha \cos \beta \sin \gamma + \sin \alpha \sin \beta & \sin \alpha \cos \beta \\ \sin \alpha \cos \beta \cos \gamma + \cos \alpha \sin \beta & \sin \alpha \cos \beta \sin \gamma - \cos \alpha \sin \beta & \cos \alpha \cos \beta \\ -\sin \beta \cos \gamma & -\sin \beta \sin \gamma & \cos \beta \end{pmatrix}$$

Euler-Winkel wurden zur gebräuchlichsten Methode zur Darstellung der Drehung eines starren Körpers im dreidimensionalen Raum, da sie am einfachsten zu verwenden und zu verstehen sind. Allerdings führten die Hauptnachteile der Euler-Winkel, wie die Anfälligkeit für eine kardansche Blockierung und die geringe Genauigkeit bei der Integration inkrementeller Änderungen über die Zeit, dazu, dass Forscher andere Rotationsdarstellungen wie Einheitsquaternionen als Möglichkeit zur effektiven Darstellung von Rotationen verwer-

Kapitel 3. Quaternionen

Achsenwinkeldarstellung

Der Rotationssatz von Euler impliziert, dass jede Drehung als einzelne Drehung um eine Achse ausgedrückt werden kann. Ein allgemeinerer Ausdruck einer Drehung ist daher die Achsenwinkeldarstellung, die einfach ausgedrückt wird als:

$$\text{rot}(\mathbf{n}, \tilde{\gamma}), \text{ wobei } \mathbf{n} = n_x \mathbf{i} + n_y \mathbf{j} + n_z \mathbf{k} \text{ und } \tilde{\gamma} \in \mathbb{R} \quad (3.31)$$

Die Rotationsachse $\mathbf{n}$ ist als jeder Einheitsvektor im dreidimensionalen Raum definiert, der durch die Rotation nicht verändert wird, und der Winkel $\tilde{\gamma}$ stellt den Betrag der Rotation um diese Achse dar, der der Rechten-Hand-Regel folgt, um die positive Richtung der Rotation zu bestimmen.

Um einen Vektor von $\mathbf{z}$ nach $\mathbf{z}'$ zu drehen, Unter Verwendung der Achse-Winkel-Darstellung wenden wir die an. Die Rotationsformel von Rodrigues ist definiert als:

$$\mathbf{z}' = \mathbf{z} \cos \tilde{\gamma} + \mathbf{n}(\mathbf{n} \cdot \mathbf{z})(1 - \cos \tilde{\gamma}) + (\mathbf{z} \times \mathbf{n}) \sin \tilde{\gamma} \quad (3.32)$$

Beachten Sie, dass es normalerweise schwierig ist, die Rotationsachse zu identifizieren, wenn es sich bei der Achse nicht um eine der Hauptachsen handelt. Darüber hinaus ist die Kombination zweier aufeinanderfolgender Drehungen, die durch einen Achsenwinkel dargestellt werden, nicht einfach und es ist besser, eine Rotationsmatrix oder eine Quaternion-Multiplikation zu verwenden, um die endgültige Drehung zu erhalten und sie dann in die Achsenwinkeldarstellung umzuwandeln.

Einheiten-Quaternion-Darstellung

Eine Einheitsquaternion $\mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$ ist als Quaternion mit einer Einheitsnorm definiert, so dass

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.33)$$

Die Darstellung einer Rotation durch eine Einheitsquaternion erfolgt im Allgemeinen durch Konvertierung anderer Rotationsdarstellungen in Quaternionen und deren Normalisierung. Für eine Achsenwinkeldarstellung $\text{rot}(\mathbf{n}, \tilde{\gamma})$, bei der die Drehachse durch einen Einheitsvektor $\mathbf{n}$ und der Drehwinkel durch $\tilde{\gamma}$ gegeben ist, wird die entsprechende Quaterniondarstellung wie folgt formuliert:

$$\mathbf{q} = \cos \frac{\tilde{\gamma}}{2} + \sin \frac{\tilde{\gamma}}{2} (\mathbf{n}_x \mathbf{i} + \mathbf{n}_y \mathbf{j} + \mathbf{n}_z \mathbf{k}) \quad (3.34)$$

Es ist wichtig zu beachten, dass $\mathbf{q}$ und $\tilde{\gamma} \mathbf{q}$ der gleichen Drehung entsprechen, auch wenn ihre Achsenwinkeldarstellungen unterschiedlich sind. Dies liegt daran, dass eine Drehung um einen Winkel $\tilde{\gamma}$ um eine Achse $\mathbf{n}$ dasselbe ist wie eine Drehung um einen Winkel von $2\tilde{\gamma}$ um die Achse $\tilde{\gamma} \mathbf{n}$.

3.4. 3D-Rotationen mit Quaternionen

Für Rotationssequenzen, die durch mehrere Quaternionen dargestellt werden, ist die Ableitung der endgültigen Quaternionrotation genauso einfach wie mit Rotationsmatrizen. Wenn wir eine Folge von Quaternion-Rotationen $q_1, q_2, \dots, q_n$ erhalten, wird die endgültige Rotation q_F berechnet, indem wir einfach das Produkt der Quaternionen erhalten und dabei die Rotationsreihenfolge und den Typ des Referenzrahmens notieren.

Für einen lokalen Referenzrahmen gilt also:

$$q_F = q_1 q_2 \dots q_n \quad (3.35)$$

Um einen dreidimensionalen Vektor p um q nach p zu drehen, wird der Anfangsvektor p zunächst als reines Quaternion dargestellt, dessen Vektorteil nur der Imaginärteil und der Realteil Null ist. Um p zu erhalten, wird Folgendes auf den Anfangsvektor p angewendet:

$$P' = qpq^{-1}, \text{ wobei } q^{-1} \text{ ist die Umkehrung von } q \quad (3.36)$$

Als Beispiel haben wir die folgende Drehung in Abbildung 3.4 dargestellt, bei der wir den Vektor drehen möchten, der den Punkt $(2,1,1)$ entlang der Z-Achse darstellt, $\text{rot}(z, 90^\circ)$.

Der gedrehte Vektor wird dann mithilfe der obigen Formel um 90° und unter Anwendung der Multiplikationsregeln der Quaternioneneinheiten i, j und k bestimmt. Dadurch entsteht ein weiteres reines Quaternion, das dann den gedrehten Vektor darstellt.

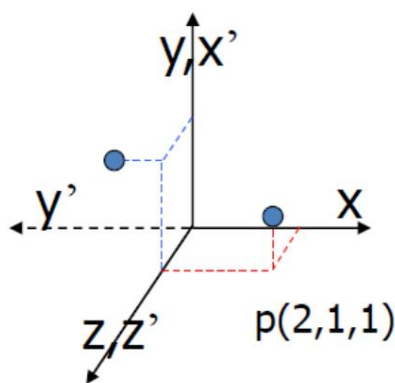


Abbildung 3.4: Drehung des Vektors $(2,1,1)$ aus [22]

$$p = 0 + 2i + 1j + 1k \quad (3.37)$$

$$q = \cos \frac{90^\circ}{2} + 0i + 0j + \sin \frac{90^\circ}{2} k = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} k \quad (3.38)$$

$$q^{-1} = \cos \frac{90^\circ}{2} + 0i + 0j - \sin \frac{90^\circ}{2} k = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} k \quad (3.39)$$

$$P' = qpq^{-1} = \sqrt{2}i + 2j + k \quad (3.40)$$

Daher sind die Koordinaten des um 90° entlang der Z-Achse gedrehten Vektors $(2,1,1)$ $(-1,2,1)$.

3.4.2 Quaternion-Konvertierungen

Im Umgang mit Einheitsquaternionen ist es notwendig zu lernen, andere Rotationsdarstellungen in Einheitsquaternionen und zurück umzuwandeln. In dieser Studie beschränken wir unsere Konvertierungen nur auf das, was wir zur Ableitung der Roboterkinematik unter Verwendung des Konzepts der Quaternionen verwenden werden.

6

Einheitsquaternion $\hat{q}$ Rotationsmatrix

Die Umwandlung von einer Quaternion $q = q_0 + q_1i + q_2j + q_3k$ in eine Rotationsmatrix R ist durch die folgende Matrix gegeben:

$$R(q) := \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (3.41)$$

Jedes Element der Matrix R wird mit r_{ij} für $i, j \in 1, 2, 3$ bezeichnet.

Einheitsquaternion $\hat{q}$ Rotationsmatrix

Um die umgekehrte Abbildung zu erhalten, führt die Untersuchung der Beziehung der Elemente von R zu den Quaternionkoeffizienten (q_0, q_1, q_2, q_3) zu vier verschiedenen inversen Abbildungen. Einige dieser Zuordnungen führen zu komplexen Ergebnissen. Daher verwenden wir die folgenden Bedingungen (abhängig von den Parametern von R), um zu bestimmen, welche Zuordnung verwendet werden soll:

a) Wenn $r_{22} > r_{33}$, $r_{11} > r_{22}$, $r_{11} > r_{33}$, dann gilt:

$$q(R) = \frac{1}{2} \begin{pmatrix} \sqrt{(1 + r_{11} + r_{22} + r_{33})} (r_{23} - r_{32}) \\ \sqrt{(1 + r_{11} + r_{22} + r_{33})} (r_{31} + r_{13}) \\ \sqrt{(1 + r_{11} + r_{22} + r_{33})} (r_{12} + r_{21}) \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad (3.42)$$

b) Wenn $r_{22} < r_{33}$, $r_{11} > r_{22}$, $r_{11} > r_{33}$, dann gilt:

$$q(R) = \frac{1}{2} \begin{pmatrix} \sqrt{(r_{23} - r_{32})/(1 + r_{11} - r_{22} - r_{33})} (1 + r_{11}) \\ \sqrt{(r_{23} - r_{32})/(1 + r_{11} - r_{22} - r_{33})} (r_{22} - r_{33}) \\ \sqrt{(r_{23} - r_{32})/(1 + r_{11} - r_{22} - r_{33})} (r_{12} + r_{21}) \\ \sqrt{(r_{23} - r_{32})/(1 + r_{11} - r_{22} - r_{33})} (r_{31} + r_{13}) \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad (3.43)$$

⁶In unserer Anwendung haben wir insbesondere die Umrechnungen von Rotationsmatrix und Achsenwinkeldarstellung in Einheitsquaternionen genutzt.

3.4. 3D-Rotationen mit Quaternionen

c) Wenn $r_{22} > r_{33}$, $r_{11} < r_{22}$, $r_{11} < r_{33}$, dann,

$$q(R) = 2 \begin{pmatrix} 1 & \frac{r_{31} - r_{13}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{12} + r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{23} - r_{32}}{(r_{11} + r_{22} + r_{33})} \\ \frac{r_{31} - r_{13}}{(r_{11} + r_{22} + r_{33})} & 1 & \frac{r_{12} - r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{23} + r_{32}}{(r_{11} + r_{22} + r_{33})} \\ \frac{r_{12} + r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{12} - r_{21}}{(r_{11} + r_{22} + r_{33})} & 1 & \frac{r_{31} + r_{13}}{(r_{11} + r_{22} + r_{33})} \\ \frac{r_{23} - r_{32}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{23} + r_{32}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{31} + r_{13}}{(r_{11} + r_{22} + r_{33})} & 1 \end{pmatrix} \quad (3.44)$$

d) Wenn $r_{22} < r_{33}$, $r_{11} < r_{22}$, $r_{11} < r_{33}$, dann gilt:

$$q(R) = 2 \begin{pmatrix} \frac{r_{12} - r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{31} - r_{13}}{(r_{11} + r_{22} + r_{33})} & 1 & \frac{r_{23} + r_{32}}{(r_{11} + r_{22} + r_{33})} \\ \frac{r_{31} - r_{13}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{12} + r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{23} - r_{32}}{(r_{11} + r_{22} + r_{33})} & 1 \\ \frac{r_{12} + r_{21}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{12} - r_{21}}{(r_{11} + r_{22} + r_{33})} & 1 & \frac{r_{31} + r_{13}}{(r_{11} + r_{22} + r_{33})} \\ \frac{r_{23} - r_{32}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{23} + r_{32}}{(r_{11} + r_{22} + r_{33})} & \frac{r_{31} + r_{13}}{(r_{11} + r_{22} + r_{33})} & 1 \end{pmatrix} \quad (3.45)$$

Die Auswahl einer dieser Zuordnungen hilft, Ungenauigkeiten zu reduzieren, indem Situationen vermieden werden, in denen der Nenner nahe Null liegt.

Einheitsquaternion $\hat{q}$ = Achsenwinkel

Das Quaternion q -Äquivalent einer Drehung $\hat{q}$ um einen Einheitsvektor n im dreidimensionalen Raum, dargestellt durch den Achsenwinkel $\text{rot}(n, \hat{q})$, ist definiert als:

$$q(\hat{q}, n) := \left(\cos\left(\frac{\hat{q}}{2}\right), n \sin\left(\frac{\hat{q}}{2}\right) \right) \quad (3.46)$$

Beachten Sie, dass wir nur Einheitsvektoren $\|n\| = 1$, in dieser Konvertierung.

Einheit Quaternion = $\hat{q}$ Achsenwinkel

Für die inverse Abbildung von der Einheitsquaternion q auf die Achsenwinkel_Tdarstellung $\text{rot}(n, \hat{q})$ betrachten wir zunächst $q = q_0 + q_1 i + q_2 j + q_3 k$, den Vektorteil von q . Die Achsenwinkel_Tdarstellung kann dann wie folgt berechnet werden:

$$\hat{q} = 2 \arccos(q_0) \quad (3.47)$$

$$n = \frac{\mathbf{q}}{\|\mathbf{q}\|} \quad (3.48)$$

Somit kann der Drehwinkel $\hat{q}$ leicht aus dem ersten Term der gegebenen Einheitsquaternion berechnet werden, während die Drehachse n durch einfache Normalisierung des Vektorteils (Imaginärteils) der Einheitsquaternion abgeleitet werden kann.

3.5 Duale Quaternionen

Quaternionen können dreidimensionale Rotationen darstellen, wie wir in der vorherigen Diskussion gesehen haben. Wenn es jedoch um Transformationen in drei Dimensionen geht, kann es auch zu Verschiebungen oder Verschiebungen kommen, die unsere aktuelle Form der Quaternion nicht darstellen kann. In diesem Abschnitt stellen wir das Konzept der Dualen Quaternionen vor, das diese Grenze überwindet und während der gesamten Dauer der Studie verwendet wird.

3.5.1 Definition

Duale Quaternionen bieten eine Möglichkeit, sowohl Rotation als auch Translation in einem Transformationsvektor darzustellen. Für eine Einheitsquaternion q , die die entsprechende Drehung darstellt, und für einen als reine Quaternion geschriebenen Vektor p , der die entsprechende Verschiebung darstellt, ist die in der Robotik am häufigsten verwendete Form einer dualen Quaternion das Quaternion-Vektor-Paar⁷

$$Q(q, p) = \cos \frac{\tilde{y}}{2} + \frac{\tilde{y}}{2} \text{ „Sünde“ } \frac{\tilde{y}}{2} \langle kx, ky, kz \rangle + \langle px, py, pz \rangle \quad (3.49)$$

Für den Quaternion-Teil verwenden wir die Umwandlung von einer Achsenwinkel-Rotationsdarstellung in eine Einheitsquaternion. Somit gibt $\tilde{y}$ den Drehwinkel an und $\langle kx, ky, kz \rangle$ bezeichnet die Drehachse. Der Vektorteil ist einfach der durch $p = \langle px, py, pz \rangle$ gegebene Verschiebungsvektor.

Für eine Transformation ohne Rotation und nur Verschiebung wird die duale Quaternion eindeutig dargestellt durch:

$$Q(q, p) = ([1, \langle 0, 0, 0 \rangle], \langle px, py, pz \rangle) \quad (3.50)$$

wobei $[1, \langle 0, 0, 0 \rangle]$ die Einheitsidentitätsquaternion darstellt.

3.5.2 Umkehrung einer dualen Quaternion

Die Umkehrung einer dualen Quaternion $Q(q, p)$ wird wie folgt bezeichnet:

$$Q^{\tilde{y}1} = ([q^{\tilde{y}1}], \langle \tilde{y}q^{\tilde{y}1} \tilde{y} p \tilde{y} q \rangle) \quad (3.51)$$

$q^{\tilde{y}1}$ bezeichnet die übliche Umkehrung der Quaternion q . Daher kann für a mit Quaternion $q = [s, v]$ der Vektorteil auch geschrieben werden als:

$$Q^{\tilde{y}1} \tilde{y} p \tilde{y} q = \tilde{y} p + [\tilde{y} 2s(v \times (\tilde{y} p)) + 2v \times (v \times (\tilde{y} p))] \quad (3.52)$$

Beachten Sie, dass p ein reines Quaternion (Realteil Null) und somit auch ein Vektor ist.

⁷Duale Quaternionen werden im Allgemeinen mithilfe dualer Zahlen definiert, sodass $Q = q + 2$ wobei $\tilde{y} \tilde{y} p = 0$. Dies wird in [14] klar beschrieben.

3.5.3 Multiplikation eines dualen Quaternions

In dualen Quaternionen werden Sequenzen von Drehungen und Verschiebungen durch Multiplikation ihrer entsprechenden dualen Quaternionendarstellungen dargestellt. Daher ist für Transformationen $Q_1(q_1, p_1)$, gefolgt von $Q_2(q_2, p_2)$ für einen lokalen Koordinatenrahmen, ihr duales Quaternionenprodukt gegeben durch:

$$Q_1 Q_2 = (q_1, p_1) \tilde{\cdot} (q_2, p_2) = ([q_1 \tilde{\cdot} q_2], \langle q_1 \tilde{\cdot} p_2 \tilde{\cdot} q \tilde{\cdot} \frac{\tilde{y}^1}{1} + p_1 \rangle) \quad (3.53)$$

wobei der Quaternionenteil einfach eine Multiplikation zwischen den Quaternionen q_1 und q_2 ist und der Vektorteil auch geschrieben werden kann als:

$$q_1 \tilde{\cdot} p_2 \tilde{\cdot} q \tilde{\cdot} \frac{\tilde{y}^1}{1} + p_1 = p_2 + 2s_1(v_1 \times p_2) + 2v_1 \times (v_1 \times p_2) + p_1 \quad (3.54)$$

wobei $q_1 = [s_1, v_1]$ und p_2 ein reines Quaternion ist und auch als Vektor fungiert.

Die Analyse der Operation für den Übersetzungsteil der dualen Quaternion zeigt, dass die nachfolgende Übersetzung zuerst durch die vorherige Drehung gedreht wird und die resultierende gedrehte Übersetzung dann zur aktuellen Übersetzung hinzugefügt wird.

3.5.4 Transformation durch ein duales Quaternion

Die Transformation eines Vektors mithilfe dualer Quaternionen erfolgt fast nach dem gleichen Prozess wie die Multiplikation zweier dualer Quaternionen. Wenn wir die endgültige Transformation haben, die durch die duale Quaternion gegeben ist,

$$QF = ([w, \langle a, b, c \rangle], \langle p_x, p_y, p_z \rangle) \quad (3.55)$$

wobei $[w, \langle a, b, c \rangle] = qF$ und $\langle p_x, p_y, p_z \rangle = pF$ und ein zu transformierender Anfangsvektor gegeben durch:

$$v = \langle v_x, v_y, v_z \rangle \quad (3.56)$$

Wir erhalten den transformierten (gedrehten und verschobenen) Vektor $v' = \langle v'_x, v'_y, v'_z \rangle$ durch:

$$\tilde{y} v' = (qF \tilde{\cdot} v \tilde{\cdot} q \tilde{\cdot} \frac{\tilde{y}^1}{F}) + pF \quad (3.57)$$

Daher wird ein Vektor mithilfe von Dual-Quaternionen transformiert, indem man ihn zunächst mit dem Quaternion-Teil qF (der die Rotation darstellt) des Dual-Quaternions dreht und ihn dann verschiebt, indem man ihm den Vektorteil pF (der die Translation darstellt) des Dual-Quaternions hinzufügt. Dieser Prozess ähnelt der Anwendung einer Transformation mithilfe von Transformationsmatrizen.

Im nächsten Kapitel werden wir die Anwendung dualer Quaternionen demonstrieren, indem wir sie bei der Ableitung sowohl der Vorwärts- als auch der Rückwärtskinematik eines Roboters mit praktischer Anwendung auf dem Universal Robot UR3 verwenden.

Kapitel 4

Anwendung von Quaternionen auf Universalroboter UR3

Das Hauptziel dieser Studie besteht darin, das Konzept der Quaternionen bei der Manipulation von Roboterbewegungen anwenden zu können. Für diese Studie werden wir mit dem Universalroboter UR3 arbeiten und seine Vorwärts- und Rückwärtskinematik mithilfe von Quaternionen ableiten. Am Ende dieses Kapitels wird eine tatsächliche Programmieranwendung für die Kinematik von UR3 getestet und vorgestellt.

4.1 Universalroboter – UR3

Der Universal Robot UR3 ist ein kollaborativer Roboterarm. Es verfügt über sechs Freiheitsgrade (6DOF) und besteht aus sechs Drehgelenken, die üblicherweise als (1) Basis, (2) Schulter, (3) Ellenbogen, (4) Handgelenk 1, (5) Handgelenk 2 und bezeichnet werden (6) Handgelenk 3, jeweils mit einem Arbeitsbereich von 360 Grad.



Abbildung 4.1: Universalroboter UR3

¹Technische Daten des Roboters finden Sie im Anhang A.

4.2 Vorwärtskinematik durch duale Quaternionen

Die Vorwärtskinematik ist im Allgemeinen einfacher zu berechnen als ihr Gegenstück, und das Gleiche gilt für die Berechnung durch Duale Quaternionen. Unser Ziel ist es, die endgültige Ausrichtung und Position des Endeffektors unter Berücksichtigung aller Gelenkwinkel zu bestimmen. Grundsätzlich gibt es mindestens zwei Möglichkeiten, die Vorwärtskinematik mithilfe dualer Quaternionen abzuleiten. Es kann entweder von einer Rotationsmatrix oder von einer Achsen-Winkel-Darstellung ausgehen. Jede dieser beiden führt zur gleichen Enddrehung, verwendet jedoch jeweils unterschiedliche Koordinaten für die anfängliche TCP-Position. In dieser Studie verwenden wir den Ansatz mit der Achse-Winkel-Darstellung.²

4.2.1 Darstellungen der Gelenktransformation

Angesichts der kinematischen Struktur des UR3-Roboters in der Nullposition leiten wir die dualen Quaternionen ab, die die Transformation in jedem Gelenk darstellen, indem wir zunächst die Achsenwinkeldarstellung jedes Gelenks in Bezug auf den Grundrahmen ableiten und diese als ersten Teil unseres Duals festlegen Quaternion und den zweiten Teil, indem die Übersetzung vom aktuellen gemeinsamen Rahmen³ zum nächsten abge-

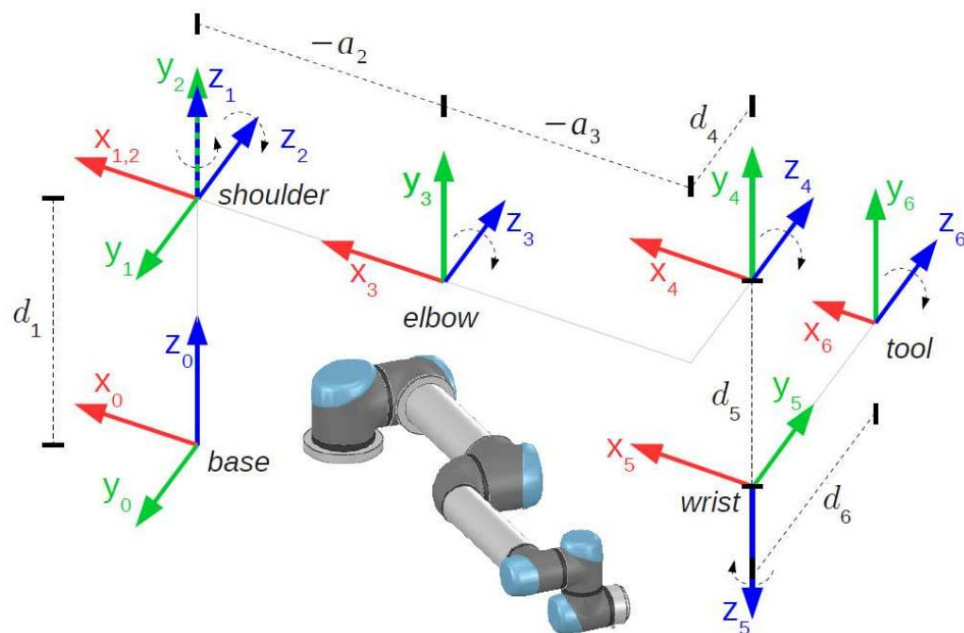


Abbildung 4.2: Kinematische Struktur von UR3 in Nullposition ($\alpha_{1,2,3,4,5,6} = 0$) [3].

²Diese Studie folgt dem Ansatz von [4] und [15]

³Die übliche Konvention zur Ableitung der Kinematik für UR3 findet sich in [3] und [11]

4.2. Vorwärtskinematik durch duale Quaternionen

Beachten Sie, dass die Z-Achse die Rotationsachse darstellt und wir dieser folgen Rechtshänder-Regel, um jeweils die positive Drehrichtung zu bestimmen Rotationsachse. Wir werden die Z-Achsen in Bezug auf die Koordinate darstellen Achse des Grundrahmens.

Für den UR3 erhalten wir außerdem die folgenden DH-Parameter:4

ich	$\ddot{y}_i$	$a_i \ddot{y}_i$	$d_i \ddot{y}_i$
1	0	0	$d_1 \ddot{y}_1$
2 $\ddot{y}_1 = 90^\circ$	0	0	$0 \ddot{y}_2$
3 $\ddot{y}_3$			
$a_2 \ddot{y}_4 = 90^\circ$	0	$a_3 \ddot{y}_4$	
5 $\ddot{y}_4 = 90^\circ$	0	$d_5 \ddot{y}_5$	
6 $\ddot{y}_5 = 90^\circ$	0	$d_6 \ddot{y}_6$	

Tabelle 4.1: UR3 Denavit-Hartenberg-Parameter

Für das erste Gelenk im Rotationsteil haben wir den Rotationswinkel $\ddot{y}_1$ und wir haben unsere Rotationsachse z_1 auf die Basisachse z_0 ausgerichtet. Für die Übersetzungsteil, beginnend vom Grundrahmen bis zum Schulterrahmen, dort ist eine Verschiebung von d_1 in der z-Achse. Daher ist unsere duale Quaternion für die Das erste Gelenk wird vertreten durch:

$$Q_1 = \cos \frac{\ddot{y}_1}{2}, \text{Sünde} \frac{\ddot{y}_1}{2} < 0, 0, 1 >, < 0, 0, d_1 >$$

Für das zweite Gelenk beträgt der Drehwinkel $\ddot{y}_2$ und unsere Drehachse z_2 jetzt parallel zur $\ddot{y}_0$ - Achse des Grundrahmens. Für den Übersetzungsteil: Wir müssen analysieren, dass das Drehen des zweiten Gelenks zu einer Translation führt in der x_0 -Achse und der z_0 -Achse, die von a_2 und $\ddot{y}_2$ abhängig ist. Daher, wir haben:

$$Q_2 = \cos \frac{\ddot{y}_2}{2}, \text{Sünde} \frac{\ddot{y}_2}{2} < 0, \ddot{y}_1, 0 >, < a_2 \cos \ddot{y}_2, 0, a_2 \sin \ddot{y}_2 >$$

Für das dritte Gelenk beträgt der Drehwinkel $\ddot{y}_3$ und unsere Drehachse z_3 , wie z_2 , ist parallel zur $\ddot{y}_0$ -Achse des Grundrahmens. Für die Übersetzung Teilweise führt die Drehung des dritten Gelenks zu einer Translation in der x_0 -Achse und die z_0 -Achse, die von a_3 und $\ddot{y}_3$ abhängt. Abgesehen davon, um zum zu gelangen Im vierten Gelenk gibt es auch eine Verschiebung von d_4 in der $\ddot{y}_0$ -Achse. Daher:

$$Q_3 = \cos \frac{\ddot{y}_3}{2}, \text{Sünde} \frac{\ddot{y}_3}{2} < 0, \ddot{y}_1, 0 >, < a_3 \cos \ddot{y}_3, \ddot{y}_4, a_3 \sin \ddot{y}_3 >$$

4DH-Parameter werden in Abschnitt 4.4.1 besprochen und dargestellt

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Für das vierte Gelenk beträgt der Drehwinkel $\ddot{y}_4$ und unsere Drehachse z_4 ist, genau wie z_2 und z_3 , parallel zur $\ddot{y}_0$ -Achse des Grundrahmens. Für den Translationsteil führt die Drehung des vierten Gelenks zu einer Translation in der x_0 -Achse und der z_0 -Achse, die von d_5 und $\ddot{y}_5$ abhängt. Deshalb haben wir:

$$Q_4 = \cos \frac{\ddot{y}_4}{2} \text{ „Sünde“ } \frac{\ddot{y}_4}{2} < 0, \ddot{y}_1, 0 > \quad , < d_5 \sin \ddot{y}_4, 0, \ddot{y}_5 \cos \ddot{y}_4 >$$

Für das fünfte Gelenk beträgt der Drehwinkel $\ddot{y}_5$ und unsere Drehachse z_5 ist parallel zur $\ddot{y}_0$ -Achse des Grundrahmens. Für den Translationsteil führt die Drehung des fünften Gelenks zu einer Translation in der x_0 -Achse und der y_0 -Achse, die von d_6 und $\ddot{y}_5$ abhängt. Deshalb haben wir:

$$Q_5 = \cos \frac{\ddot{y}_5}{2} \text{ „Sünde“ } \frac{\ddot{y}_5}{2} < 0, 0, \ddot{y}_1 > \quad , < \ddot{y}_6 \sin \ddot{y}_5, \ddot{y}_6 \cos \ddot{y}_5, 0 >$$

Für das sechste und letzte Gelenk beträgt der Drehwinkel $\ddot{y}_6$ und unsere Drehachse z_6 ist parallel zur $\ddot{y}_0$ -Achse des Grundrahmens. Es gibt keine Übersetzung, da es sich um den letzten Frame handelt. Deshalb haben wir:

$$Q_6 = \cos \frac{\ddot{y}_6}{2} \text{ „Sünde“ } \frac{\ddot{y}_6}{2} < 0, \ddot{y}_1, 0 > \quad , < 0, 0, 0 >$$

Die Darstellung der Transformationen durch duale Quaternionen hat ein ähnliches Konzept wie die Transformationsmatrizen. Der Hauptunterschied besteht darin, dass duale Quaternionen kompakter sind, sodass wir nur 7 Komponenten benötigen, während wir für die Transformationsmatrizen $4 \times 4 = 16$ Komponenten verwenden.

4.2.2 UR3-Vorwärtskinematikberechnung

Nachdem wir die entsprechenden Doppelquaternionen für jedes Gelenk abgeleitet haben, nehmen wir deren Produkte vom Grundrahmen bis zum endgültigen Rahmen, um die endgültige Doppelquaternion zu erhalten, die die endgültige Transformation und die Vorwärtskinematik des Roboters darstellt. Somit haben wir,

$$Q_{FK} = Q_1 \ddot{y} Q_2 \ddot{y} Q_3 \ddot{y} Q_4 \ddot{y} Q_5 \ddot{y} Q_6 \quad (4.1)$$

Bei diesem Ansatz müssen die Koordinaten des zu transformierenden Punkts auch in Bezug auf das Basissystem angegeben werden. Daher wird der Punkt während der Transformation um den Rotationsteil von Q_{FK} gedreht und dann um seinen Translationsteil ausgehend vom Basisrahmen verschoben.

4.3 Inverse Kinematik durch duale Quaternionen

Die Ableitung der inversen Kinematik erfordert mehr Arbeit und Analyse als die Vorwärtskinematik. Wir müssen das Konzept der Transformationen durch duale Quaternionen anwenden, uns mit komplexen trigonometrischen Gleichungen befassen und die Transformationen, die jedes Gelenk am Roboter durchführt, geometrisch analysieren und wie sich jede von ihnen auf die Koordinaten der nachfolgenden Gelenksysteme auswirkt. Unser Ziel in der inversen Kinematik ist es, die Gelenkwinkel zu bestimmen, die erforderlich sind, um den Endeffektor in die gewünschte Position und Ausrichtung zu bewegen, die durch die endgültige Transformation $Q_{FK} = ([w, < a, b, c >])$ gegeben ist.

4.3.1 Darstellungen von Gelenktransformationen

Um mit der Ableitung der inversen Kinematik zu beginnen, benötigen wir die dualen Quaternionen, die die von jedem Gelenk durchgeführte Transformation Q_i für $1 \leq i \leq 6$ darstellen, die wir oben bereits für die Vorwärtskinematik abgeleitet haben, sowie ihre entsprechenden dualen Quaternion-Inversen. Q_i^{-1} ist

In Zukunft werden wir die folgenden Darstellungen für längere Gleichungen verwenden: $\cos \theta_i = c_i$, $\sin \theta_i = s_i$, $\cos \frac{\theta_i}{2} = c_{i/2}$, $\sin \frac{\theta_i}{2} = s_{i/2}$.

Daher haben wir für jedes Gelenk:

$$\begin{aligned} Q_1 &= ([c_1 + s_1 k], < d_1 k >) \\ Q_2 &= ([c_2 \tilde{y} s_2 j], < a_2 c_2 i + a_2 s_2 k >) \\ Q_3 &= ([c_3 \tilde{y} s_3 j], < a_3 c_3 i \tilde{y} d_4 j + a_3 s_3 k >) \\ Q_4 &= ([c_4 \tilde{y} s_4 j], < d_5 s_4 i \tilde{y} d_5 c_4 k >) \\ Q_5 &= ([c_5 \tilde{y} s_5 k], < d_6 s_5 i \tilde{y} d_6 c_5 j >) \\ Q_6 &= ([c_6 \tilde{y} s_6 j], < 0 >) \end{aligned}$$

und die Umkehrungen als:

$$\begin{aligned} Q_1^{-1} &= ([c_1 \tilde{y} s_1 k], < d_1 k >) = \\ Q_2^{-1} &= ([c_2 + s_2 j], < a_2 i >) \\ Q_3^{-1} &= ([c_3 + s_3 j], < a_3 i + d_4 j >) = ([c_4 \\ Q_4^{-1} &+ s_4 j], < d_5 k >) \\ Q_5^{-1} &= ([c_5 + s_5 k], < d_6 i >) = \\ Q_6^{-1} &= ([c_6 + s_6 j], < 0 >) \end{aligned}$$

Wir werden die dualen Quaternionendarstellungen dieser beiden Gelenke verwenden, um die Beziehung zwischen den aufeinanderfolgenden Transformationskombinationen zu ermitteln.

4.3.2 Gelenktransaktionsprodukte

Verwendung der dualen Quaternionen Q_i ($1 \leq i \leq 6$), die die Kinematik bezeichnen Transformationen, die die räumlichen Beziehungen zwischen aufeinander folgenden beschreiben Koordinatenrahmen und Q_i^{-1} als ihre entsprechenden Umkehrungen definieren wir zunächst das duale Quaternionenprodukt M_i als:

$$M_i = Q_i \cdot M_{i+1} \text{ mit } 1 \leq i \leq 6 \quad (4.2)$$

Für $i = 6$ gilt $M_6 = Q_6$. Daher haben wir M_i als,

$$\begin{aligned} M_6 &= Q_6 \\ M_5 &= Q_5 \cdot M_6 = Q_5 \cdot Q_6 \\ M_4 &= Q_4 \cdot M_5 = Q_4 \cdot Q_5 \cdot Q_6 \\ M_3 &= Q_3 \cdot M_4 = Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 \\ M_2 &= Q_2 \cdot M_3 = Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 \\ M_1 &= Q_1 \cdot M_2 = Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 \end{aligned}$$

Dann definieren wir ein weiteres duales Quaternionprodukt N_i wie folgt:

$$N_{i+1} = Q_i^{-1} \cdot N_i \text{ mit } 1 \leq i \leq 6 \quad (4.3)$$

Für $i = 0$ definieren wir N_1 als letzte Transformationsquaternion, was bedeutet:

$$N_1 = [R_w, T_w] = ([w, < a, b, c >], < p_x, p_y, p_z >)$$

Daher haben wir N_i als,

$$\begin{aligned} N_1 &= [R_w, T_w] \\ N_2 &= Q_1^{-1} \cdot N_1 = Q_1^{-1} \cdot [R_w, T_w] \\ N_3 &= Q_2^{-1} \cdot N_2 = Q_2^{-1} \cdot Q_1^{-1} \cdot [R_w, T_w] \\ N_4 &= Q_3^{-1} \cdot N_3 = Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot [R_w, T_w] \\ N_5 &= Q_4^{-1} \cdot N_4 = Q_4^{-1} \cdot Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot [R_w, T_w] \\ N_6 &= Q_5^{-1} \cdot N_5 = Q_5^{-1} \cdot Q_4^{-1} \cdot Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot [R_w, T_w] \end{aligned}$$

Wir können deutlich erkennen, dass M_1 und N_1 genau der Vorwärtskinematik entsprechen QFK. Eine weitere Analyse der Zusammensetzung jeder dieser Transformationen zwischen den dualen Quaternionprodukten M und N impliziert, dass $M_i = N_i$ ist. Daher, ihre entsprechenden Begriffe sollten ebenfalls gleich sein.

Der nächste Schritt besteht darin, alle entsprechenden dualen Quaternion-Terme dazwischen gleichzusetzen M_i und N_i , die uns mehrere Gleichungen liefern, die wir zur Berechnung verwenden können für die inverse Kinematik.

4.3.3 Transformationen von Basis zu Handgelenksrahmen

Die Berechnung der Transformationsprodukte M_i und N_i mit $(a \dot{y} i \dot{y} b)$ führt uns zu $((b \dot{y} a) + 1) \times 7 \times 2$ Gleichungen, in denen die Multiplikatoren die Anzahl der Transformationen $(b \dot{y} (a \dot{y} 1))$ darstellen.), die Anzahl der Terme im dualen Quaternion (7) und entsprechend sowohl für M als auch N . Wir setzen die Terme für jedes entsprechende Transformationsprodukt gleich und erhalten $((b \dot{y} a) + 1) \times 7$ Gleichungen und versuchen, eine Beziehung zwischen einigen dieser Gleichungen zu finden und sie zu manipulieren, um explizit nach den Gelenkwinkeln auflösen zu können.

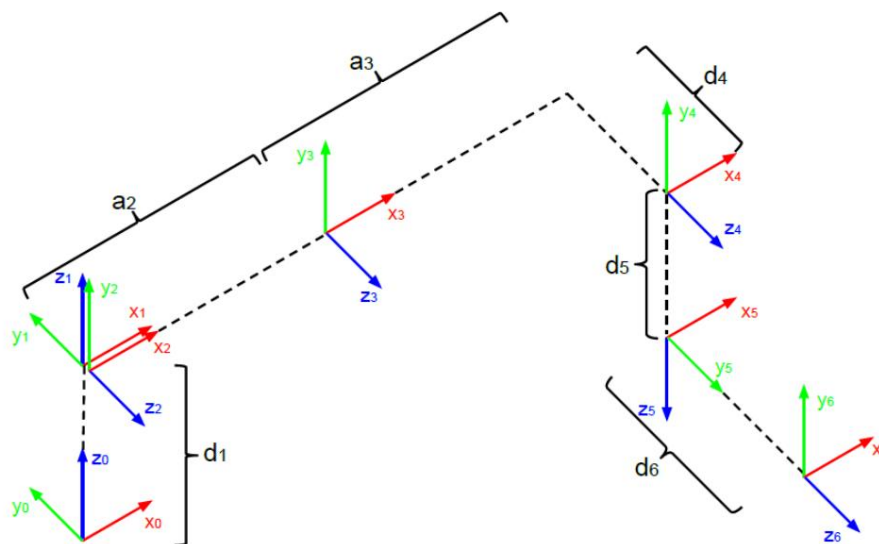


Abbildung 4.3: Koordinatensysteme von UR3 ($\dot{y}_{1,2,3,4,5,6} = 0$) aus [11].

Transformation von Rahmen 1 (Basis) zu Rahmen 6 (Handgelenk 3).

Nach [4] und [15] sollten wir zunächst mit der Basis-zu-Handgelenk-3-Transformation beginnen. Dies führt uns zu 84 Gleichungen mit $(1 \dot{y} i \dot{y} 6)$. Wir setzen die entsprechenden Transformationsprodukte und Terme zwischen M_i und N_i gleich. Diese Gleichungen sind in Anhang B aufgeführt. Wie einfach es ist, eine einfache Gleichung für die Gelenkwinkel zu berechnen, hängt von der Roboterstruktur ab. Im Fall von UR3 gibt es keine Gleichungen, mit denen sich der Wert eines Gelenkwinkels direkt ableiten lässt. Wir können eine Gleichung für $\dot{y}_5$ und dann für $\dot{y}_6$ finden, aber wir brauchen $\dot{y}_1$, um sie zu lösen. Wir können numerische Methoden anwenden, um die Werte der Gelenkwinkel abzuschätzen, aber in unserem Fall benötigen wir genaue Werte. Um dieses Problem zu lösen, können wir dann den geometrischen Ansatz integrieren, der üblicherweise bei der Ableitung der inversen Kinematik eines Roboters verwendet wird, und mit der nächsten Transformation fortfahren, der Base-to-Wrist-2-

Transformation von Rahmen 1 (Basis) zu Rahmen 5 (Handgelenk 2).

Unsere Hauptmotivation, mit dieser Transformation fortzufahren, ist die Lösung nach $\ddot{y}_1$. Der geometrische Ansatz besteht darin, die Beziehung zwischen Gelenkwinkeln und Verschiebungen zu ermitteln. Die folgende Abbildung zeigt die Beziehung zwischen dem Gelenkwinkel $\ddot{y}_1$ und den Koordinaten von Rahmen 1 (Basis) und Rahmen 5 (Handgelenk 2).

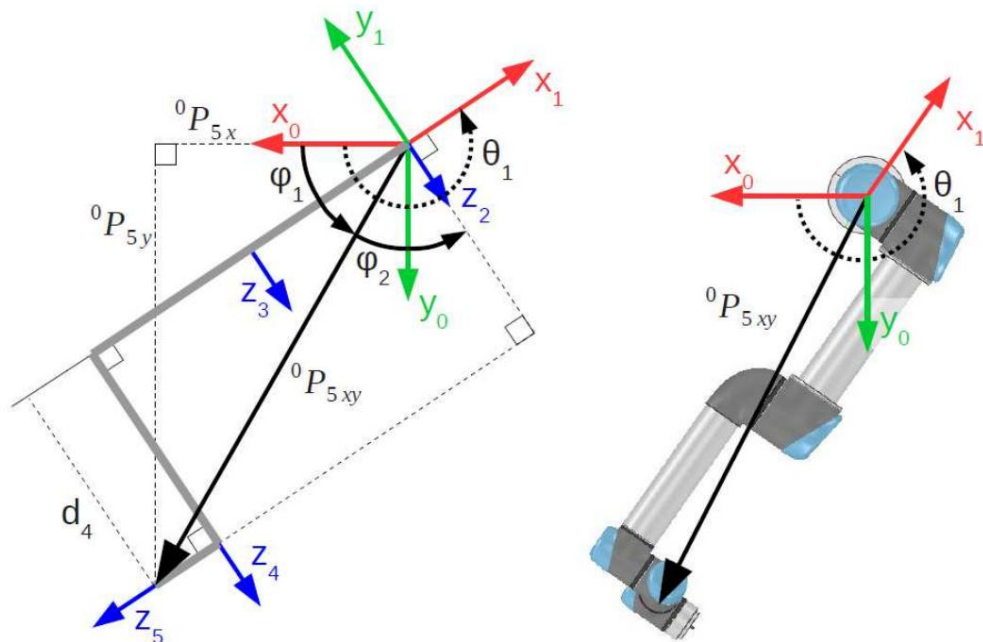


Abbildung 4.4: UR3-Roboterstruktur von Frame 1 bis Frame 5 (aus [3])

Wie wir sehen können, ist es möglich, den Wert von $\ddot{y}_1$ mithilfe der Verschiebung zwischen Frame 1, Q_1 , und Frame 5, Q_5 , zu bestimmen. Die Base-to-Wrist-2-Transformation führt uns zu 70 Gleichungen mit $(1 \ddot{y}_i \ddot{y}_5)$. Glücklicherweise können wir, anstatt diese Transformationen neu zu berechnen, die vorherige Transformation verwenden und N_1 bis N_5 nehmen, da sie einfach äquivalent sind. Der einzige Unterschied besteht darin, dass wir an den Koordinaten bei Frame 5 arbeiten. Daher müssen wir nur eine Übersetzung von Frame 6-Koordinaten zu Frame 5 durchführen. Was die Vorwärtstransformationsprodukte betrifft, sollten wir mit M_5 statt mit M_6 beginnen. Dies kann einfach durch Setzen von $M_5 = Q_5$ und der Übersetzung auf Null erfolgen. Wenn wir die Terme von M_5 mit den entsprechenden Termen in N_5 gleichsetzen und trigonometrische Identitäten anwenden, erhalten wir unsere Formel für $\ddot{y}_1$.

Dies wird es uns dann ermöglichen, $\ddot{y}_5$ und $\ddot{y}_6$ mithilfe der Gleichungen aufzulösen, die wir aus der Transformation von Rahmen 1 (Basis) zu Rahmen 6 (Handgelenk 3) erhalten haben.

Ableitung des Gelenkwinkels $\ddot{y}_1$ (Basis)

Um eine Formel für $\ddot{y}_1$ abzuleiten, können wir die Koordinaten zuerst im letzten Frame (Handgelenk 3) zurück in Frame 5 (Handgelenk 2) übertragen und diese in der Formel verwenden, die aus den Transformationsprodukten extrahiert werden soll. Um zurück zu Frame 5 zu übersetzen, analysieren wir die Transformationen von Wrist 3 (Q6) und Wrist 2 (Q5) und ihre Beiträge zur endgültigen Transformation QFK.

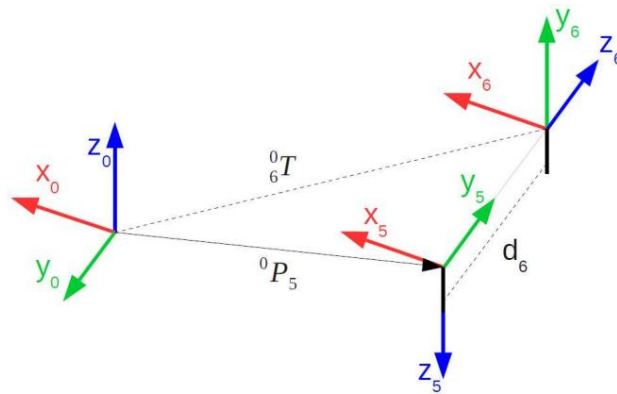


Abbildung 4.5: Übersetzung von Rahmen 6 (Handgelenk 3) zu Rahmen 5 (Handgelenk 2)[3]

Frame 5 (Handgelenk 2) Transformation:

$$Q5 = ([\bar{c}5\ddot{y} \ s^{-5}k], <\ddot{y}d6s5i \ \ddot{y} \ d6c5j >)$$

Frame 6 (Handgelenk 3) Transformation:

$$Q6 = ([\bar{c}6\ddot{y} \ s^{-6}j], <0 >)$$

Endgültige Transformation: Rahmen 1 (Basis) zu Rahmen 6 (Handgelenk 3)

$$QFK = Q1 \ddot{y} Q2 \ddot{y} Q3 \ddot{y} Q4 \ddot{y} Q5 \ddot{y} Q6 = Q1 \quad 6$$

Offensichtlich führt Q6 keine Verschiebungen durch, sondern nur Drehungen. Daher ist unsere Endeffektorposition bereits in der Transformation bis Q5 festgelegt. Somit gilt für den Übersetzungsteil (p) von QFK:

$$\begin{aligned} (QFK)p &= (Q1 \ 6)p = (Q15)p \\ &= (Q1 \ddot{y} Q2 \ddot{y} Q3 \ddot{y} Q4 \ddot{y} Q5)p \end{aligned} \quad (4.4)$$

Beachten Sie, dass wir die Koordinaten in Frame 5 benötigen und uns nur mit dem Übersetzungsteil der dualen Quaternion befassen.

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Wir müssen die von Q5 vorgenommene Übersetzung rückgängig machen. Wir tun dies, indem wir die Umkehrung verwenden:

$$F_5^{\tilde{y}_1} = ([-c_5 + s_5 k], d_5 i)$$

Daher,

$$\begin{aligned} (Q_{FK} \tilde{y} Q_{\tilde{y}_1}^{-1})^p &= (Q_1 \tilde{y} Q_2 \tilde{y} Q_3 \tilde{y} Q_4 \tilde{y} Q_5 \tilde{y} Q_{\tilde{y}_1}^{-1})^p \\ &= (Q_1 \tilde{y} Q_2 \tilde{y} Q_3 \tilde{y} Q_4)^p \\ &= (Q_1)^S \end{aligned} \quad (4.5)$$

Wir haben noch keinen Wert für $\tilde{y}_5$, daher können wir Q5 nicht vollständig erstellen, aber da wir uns nur mit dem Übersetzungsteil befassen, brauchen wir ihn nicht wirklich. Stattdessen können wir einen beliebigen Wert für seinen Quaternion-Teil festlegen und nur den Übersetzungsteil definieren. Denken Sie daran, dass die Bestimmung des Übersetzungsteils aus einer dualen Quaternion-Multiplikation ohne Verwendung von q_B definiert wird.

$$(Q_A \tilde{y} Q_B)^p = \langle q_A \tilde{y} p_B \tilde{y} q_A^{\tilde{y}_1} + p_A \rangle \quad (4.6)$$

Da $(Q_{FK})^p = (Q_1)^S$, gilt:

$$(Q_{FK} \tilde{y} Q_{\tilde{y}_1}^{-1})^p = \langle q_{FK} \tilde{y} p = \langle x_5 i + y_5 j + \tilde{y}_5 k + q_{FK}^{\tilde{y}_1} \tilde{y} q_{FK}^{\tilde{y}_1} + p_{FK} \rangle \quad (4.7)$$

$$z_5 k \rangle = (Q_1)^S \quad (4.8)$$

wobei $\langle x_5, y_5, z_5 \rangle$ die Koordinaten in Frame 5 in Bezug auf Frame 1 sind.

Um nun $\tilde{y}_1$ zu berechnen, benötigen wir die Dual-Quaternion-Transformationsprodukte für die Base-to-Wrist-2-Transformation. Wie wir in unserer Analyse festgestellt haben, können wir einfach (B.13) $M_{56} = N_{56}$ verwenden, wobei der $\cos \tilde{y}_5$ -Term ausgeschlossen ist.

Daher,

$$\begin{aligned} 0 &= y \cos \tilde{y}_1 \tilde{y} x \sin \tilde{y}_1 + d_4 y \\ \cos \tilde{y}_1 \tilde{y} x \sin \tilde{y}_1 &= \tilde{y} d_4 \end{aligned} \quad (4.9)$$

Wir wenden die folgende trigonometrische Identität an:

$$a \sin \tilde{y} + b \cos \tilde{y} = c$$

$$\tilde{y} = \text{atan2}(a, b) \pm \text{atan2}(\tilde{y} a^2 + b^2 - c^2, c) \quad (4.10)$$

Und jetzt verwenden wir die Koordinaten aus Bild 5, um die Formel für $\tilde{y}_1$ abzuleiten,

$$\tilde{y}_1 = \text{atan2}(\tilde{y} x_5, y_5) \pm \text{atan2}\left(x_5^2 + y_5^2 + d_4^2 - J_5^2 \tilde{y}^2, \tilde{y} d_4\right) \quad (4.11)$$

Wir betrachten zwei Werte für $\tilde{y}_1$, die der Rotation des Basisgelenks entsprechen, wenn die Schulter entweder nach links oder nach rechts eingestellt wird.

Ableitung des Gelenkwinkels $\dot{\gamma}_5$ (Handgelenk 2)

Da uns nun $\dot{\gamma}_1$ gegeben ist, können wir dasselbe Gleichungspaar aus dem Dual verwenden Quaternion-Transformationsprodukte für die Base-to-Wrist-3-Transformation (B.13), aber ohne einen Term zur Ableitung der Gleichung für $\dot{\gamma}_5$ auszuschließen,

$$\underline{M56 = N56}$$

$$\dot{\gamma}_6 \cos \dot{\gamma}_5 = \dot{\gamma} \cos \dot{\gamma}_1 \dot{\gamma} x \sin \dot{\gamma}_1 + d_4 \quad (4.12)$$

Wir können $\dot{\gamma}_5$ einfach berechnen, indem wir

$$\dot{\gamma}_5 = \pm \arccos \frac{x_6 \sin \dot{\gamma}_1 \dot{\gamma} y_6 \cos \dot{\gamma}_1 \dot{\gamma} d_4}{d_6} \quad (4.13)$$

Wir betrachten zwei Werte für $\dot{\gamma}_5$ für die entsprechende Rotation des Handgelenks 2 dass das Handgelenk entweder oben oder unten ist.

Ableitung des Gelenkwinkels $\dot{\gamma}_6$ (Handgelenk 3)

Wir haben jetzt $\dot{\gamma}_1$ und $\dot{\gamma}_5$, wir können dann die Gleichungspaare (B.29), (B.30) verwenden, (B.31) und (B.32) auch aus den Dual-Quaternion-Transformationsprodukten für die Basis-zu-Handgelenk-3-Transformation, um die Gleichung für $\dot{\gamma}_6$ abzuleiten,

$$\underline{M21 = N21}$$

$$\cos \frac{\dot{\gamma}_5}{2} \cos \frac{\dot{\gamma}_2 + \dot{\gamma}_3 + \dot{\gamma}_4 + \dot{\gamma}_6}{2} = w \cos \frac{\dot{\gamma}_1}{2} + c \sin \frac{\dot{\gamma}_1}{2} \quad (4.14)$$

$$\underline{M22 = N22}$$

$$\sin \frac{\dot{\gamma}_5}{2} \sin \frac{\dot{\gamma}_2 + \dot{\gamma}_3 + \dot{\gamma}_4 + \dot{\gamma}_6}{2} = \sin \cos \frac{\dot{\gamma}_1}{2} + b \sin \frac{\dot{\gamma}_1}{2} \quad (4.15)$$

$$\underline{M23 = N23}$$

$$\dot{\gamma} \cos \frac{\dot{\gamma}_5}{2} \sin \frac{\dot{\gamma}_2 + \dot{\gamma}_3 + \dot{\gamma}_4 + \dot{\gamma}_6}{2} = \dot{\gamma} \sin \frac{\dot{\gamma}_1}{2} + b \cos \frac{\dot{\gamma}_1}{2} \quad (4.16)$$

$$\underline{M24 = N24}$$

$$\dot{\gamma} \sin \frac{\dot{\gamma}_5}{2} \cos \frac{\dot{\gamma}_2 + \dot{\gamma}_3 + \dot{\gamma}_4 + \dot{\gamma}_6}{2} = \dot{\gamma} w \sin \frac{\dot{\gamma}_1}{2} + c \cos \frac{\dot{\gamma}_1}{2} \quad (4.17)$$

Unter Verwendung trigonometrischer Identitäten (Winkelsumme) entsprechen diese:

$$c^{-(2+3+4)}c^{-6}\ddot{y}s^{-(2+3+4)}s^{-6} = \frac{wc^{-1}+cs^{-1}}{c^{-5}} = \ddot{y} \quad (4.18)$$

$$s^{-(2+3+4)}c^{-6}\ddot{y}c^{-(2+3+4)}s^{-6} = \frac{ac^{-1}+bs^{-1}}{s^{-5}} = \ddot{y} \quad (4.19)$$

$$s^{-(2+3+4)}c^{-6}+c^{-(2+3+4)}s^{-6} = \frac{as^{-1}\ddot{y}bc^{-1}}{c^{-5}} = \ddot{y} \quad (4.20)$$

$$c^{-(2+3+4)}c^{-6}+s^{-(2+3+4)}s^{-6} = \frac{ws^{-1}\ddot{y}cc^{-1}}{s^{-5}} = \ddot{y} \quad (4.21)$$

Die Multiplikation von (4.18) mit s^{-6} ergibt:

$$c^{-(2+3+4)}c^{-6}s^{-6}\ddot{y}s^{-(2+3+4)}s^{-6} = \ddot{y}s^{-6} \quad (4.22)$$

Und die Multiplikation von (4.19) mit c^{-6} ergibt:

$$s^{-(2+3+4)}c^{-6}\ddot{y}c^{-(2+3+4)}s^{-6} = \ddot{y}c^{-6} \quad (4.23)$$

Die Summe dieser beiden Gleichungen ergibt:

$$s^{-(2+3+4)}c^{-6}\ddot{y}s^{-6} = -s^{-(2+3+4)}c^{-6} = \ddot{y}s^{-6} + \ddot{y}c^{-6} \quad (4.24)$$

Dann erhalten wir durch Multiplikation von (4.20) mit c^{-6}

$$s^{-(2+3+4)}c^{-6} + c^{-(2+3+4)}s^{-6} = \ddot{y}c^{-6} \quad (4.25)$$

Und die Multiplikation von (4.21) mit $\ddot{y}s^{-6}$ ergibt:

$$\ddot{y}c^{-(2+3+4)}c^{-6}s^{-6}\ddot{y}s^{-(2+3+4)}s^{-6} = \ddot{y}\ddot{y}s^{-6} \quad (4.26)$$

Die Summe dieser beiden Gleichungen ergibt:

$$s^{-(2+3+4)}c^{-6}\ddot{y}s^{-6} = -s^{-(2+3+4)}c^{-6} = \ddot{y}c^{-6}\ddot{y}s^{-6} \quad (4.27)$$

Wir können die Gleichungen (4.24) und (4.27) subtrahieren, um Folgendes zu vereinfachen Gleichung, in der nur $\ddot{y}$ unbestimmt ist:

$$\ddot{y}s^{-6} + \ddot{y}c^{-6} = \ddot{y}c^{-6}\ddot{y}s^{-6}\ddot{y}s^{-6}(\ddot{y} + \ddot{y}) = c^{-6}(\ddot{y}\ddot{y})$$

Wenn wir diese Gleichung neu anordnen, um sie nach $\ddot{y}$ aufzulösen, erhalten wir die folgende Formel:

$$\ddot{y} = 2\operatorname{atan2}(\ddot{y}\ddot{y}, \ddot{y} + \ddot{y}) \quad (4.28)$$

Es gibt nur einen Wert für $\ddot{y}$, der der Drehung des Handgelenks 3 entspricht.

4.3.4 Transformation von Schulter-zu-Handgelenk-Rahmen

Jetzt müssen nur noch $\ddot{y}_2$, $\ddot{y}_3$ und $\ddot{y}_4$ bestimmt werden. Beachten Sie, dass ihre Die entsprechenden Gelenke bilden einen 3R-planaren Manipulator, wie in der Abbildung dargestellt unten,

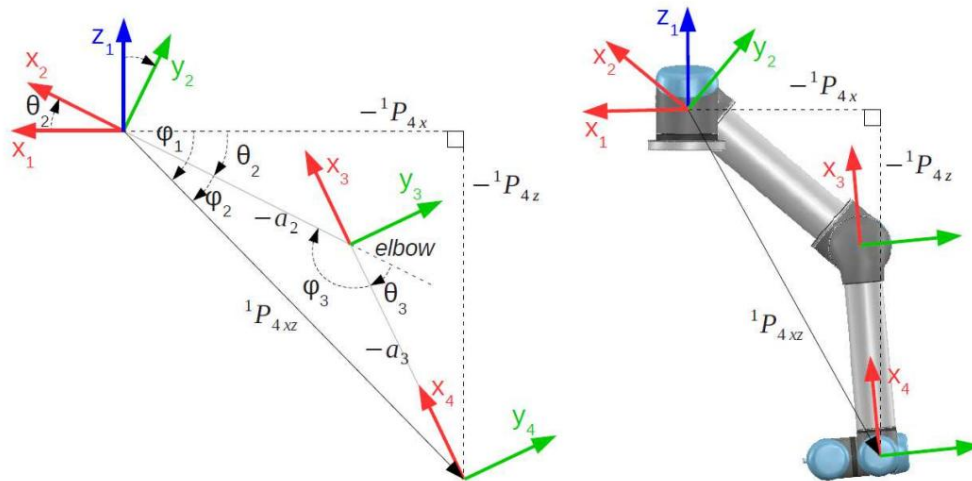


Abbildung 4.6: 3R-Planar-Manipulator, gebildet aus den UR3-Gelenken 2 bis 4 aus [3].

Daher können wir die Transformationsprodukte M_i und N_i erhalten, wobei ($2 \leq i \leq 4$). Dies führt uns zu 42 Gleichungen. Wenn wir die Terme für jedes entsprechende Transformationsprodukt gleichsetzen, erhalten wir $\frac{42}{2} = 21$ Gleichungen und wir versuchen, eine Beziehung zwischen einigen dieser Gleichungen zu finden und manipulieren sie, um explizit nach unseren verbleibenden Gelenkwinkeln suchen zu können bestimmt.

Wir haben somit die folgende duale Quaternion-Darstellung für jedes Gelenk:

$$Q_2 = ([c_2 \ddot{y}_2 \ s^{-2}], \langle a_2 c_2 i + a_2 s_2 k \rangle)$$

$$Q_3 = ([c_3 \ddot{y}_3 \ s^{-3}], \langle a_3 c_3 i + d_4 j + a_3 s_3 k \rangle)$$

$$Q_4 = ([c_4 \ddot{y}_4 \ s^{-4}], \langle 0 \rangle)$$

und die Umkehrungen als:

$$Q_2^{\ddot{y}_1} = ([c_2 + s_2], \langle \ddot{y}_2 a_2 i \rangle)$$

$$Q_3^{\ddot{y}_1} = ([c_3 + s_3], \langle \ddot{y}_2 a_3 i + d_4 j \rangle)$$

$$Q_4^{\ddot{y}_1} = ([c_4 + s_4], \langle 0 \rangle)$$

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Daher wird unser Dual-Quaternion-Transformationsprodukt M_i wie folgt formuliert:

$$M_4 = Q_4$$

$$M_3 = Q_3 \cdot M_4 = Q_3 \cdot Q_4$$

$$M_2 = Q_2 \cdot M_3 = Q_2 \cdot Q_3 \cdot Q_4$$

Und das duale Quaternionenprodukt N_i als:

$$N_2 = Q_2 = ([a_2, b_2, c_2, d_2], [x_2, y_2, z_2])$$

$$N_3 = Q_3 \cdot N_2 = Q_3 \cdot Q_2 \quad 2 \quad 4$$

$$N_4 = Q_4 \cdot N_3 = Q_4 \cdot Q_3 \cdot Q_2 \quad 3 \quad 4$$

Diese Gleichungen sind in Anhang C aufgeführt.

Abgesehen davon müssen wir auch unsere Basiskoordinaten und Endkoordinaten übersetzen, da wir nur von Rahmen 2 (Schulter) bis Rahmen 4 (Handgelenk 1) arbeiten. Um die Koordinaten zu übersetzen, verwenden wir dieselbe Technik wie bei der Bestimmung der Koordinaten für $\tilde{y}_1$.

$$Q_{FK} = Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 = Q_1 \quad 6 \quad (4,29)$$

$$Q_1 \cdot Q_{FK} = Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 \quad = Q_2 \quad 6 \quad (4,30)$$

$$Q_1 \cdot Q_{FK} \cdot Q_1^{-1} \quad 6 \quad = Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6 \quad = Q_2 \quad 5 \quad (4,31)$$

$$Q_1 \cdot Q_{FK} \cdot Q_1^{-1} \quad 6 \quad Q_1^{-1} \quad 5 \quad = Q_2 \cdot Q_3 \cdot Q_4 \quad = Q_2 \quad 4 \quad (4,32)$$

Beachten Sie, dass wir Q_1 , Q_5 und Q_6 bereits vollständig bilden können. Für Q_4 müssen wir nur die Übersetzung zurücksetzen. Dazu verwenden wir seine Umkehrung Q_4^{-1} auf die gleiche Weise wie in der vorherigen Übersetzung. Wir haben,

$$Q_4^{-1} = ([-c_4, -s_4], [d_4])$$

Und wir machen die Übersetzung rückgängig, indem wir

$$(Q_2 \cdot Q_4 \cdot Q_4^{-1})^p = (Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_4^{-1})^p \quad (4.33)$$

$$= (Q_2 \cdot Q_3)^p$$

$$= (Q_2 \cdot Q_3)^p$$

Wir erhalten die Koordinaten durch den Übersetzungsteil dieses Produkts.

$$(Q_2 \cdot Q_4 \cdot Q_4^{-1})^p = \left[q_2^2 + s_4^2 \cdot (q_4^{-1})^2 + s_4^2 \right] \quad (4.34)$$

$$= [x_4 i + y_4 j + z_4 k]$$

wobei $[x_4, y_4, z_4]$ die Koordinaten in Frame 4 in Bezug auf Frame 2 sind.

4.3. Inverse Kinematik durch duale Quaternionen

Ableitung des Gelenkwinkels $\ddot{\gamma}_3$ (Ellenbogen)

Um eine Formel für $\ddot{\gamma}_3$ abzuleiten, verwenden wir die Gleichungen (C.19) und (C.21) in die Schulter-zu-Handgelenk-1-Transformation zusammen mit der zuvor berechneten Koordinaten $\langle x_4, y_4, z_4 \rangle$.

$$\underline{M_{25} = N_{25}}$$

$$a_3 \cos(\ddot{\gamma}_2 + \ddot{\gamma}_3) + a_2 \cos \ddot{\gamma}_2 = x_4 \quad (4.35)$$

$$\underline{M_{27} = N_{27}}$$

$$a_3 \sin(\ddot{\gamma}_2 + \ddot{\gamma}_3) + a_2 \sin \ddot{\gamma}_2 = z_4 \quad (4.36)$$

Wenn wir diese beiden Gleichungen quadrieren und addieren, erhalten wir:

$$a_3^2 + 2a_2a_3 \cos \ddot{\gamma}_3 + a_2^2 = x_4^2 + z_4^2 \quad (4.37)$$

Jetzt können wir $\ddot{\gamma}_3$ direkt berechnen, indem wir

$$\ddot{\gamma}_3 = \pm \arccos \frac{x_4^2 + z_4^2 - a_2^2}{2a_2a_3} \quad (4.38)$$

Wir betrachten zwei Werte für $\ddot{\gamma}_3$, die der Drehung des Ellenbogengelenks entsprechen entweder oben oder unten zu sein.

Ableitung des Gelenkwinkels $\ddot{\gamma}_2$ (Schulter)

Da wir bereits $\ddot{\gamma}_3$ haben, können wir $\ddot{\gamma}_2$ mithilfe der Gleichungen (C.12) berechnen. und (C.14) auch in der Schulter-zu-Handgelenk-1-Transformation unter Verwendung derselben Koordinaten.

$$\underline{M_{35} = N_{35}}$$

$$a_3 \cos \ddot{\gamma}_3 = y_4 + x_4 \cos \ddot{\gamma}_2 + z_4 \sin \ddot{\gamma}_2 \quad (4.39)$$

$$\underline{M_{37} = N_{37}}$$

$$a_3 \sin \ddot{\gamma}_3 = y_4 x_4 \sin \ddot{\gamma}_2 + z_4 \cos \ddot{\gamma}_2 \quad (4.40)$$

Und mit ähnlichen Manipulationen können wir $\ddot{\gamma}_2$ direkt berechnen, indem wir

$$\ddot{\gamma}_2 = \arctan2(z_4(a_2 + a_3 \cos \ddot{\gamma}_3) - x_4 a_3 \sin \ddot{\gamma}_3, x_4(a_2 + a_3 \cos \ddot{\gamma}_3) + z_4 a_3 \sin \ddot{\gamma}_3) \quad (4.41)$$

Wir betrachten einen Wert für $\ddot{\gamma}_2$, der der Schulterrotation entspricht.

Ableitung des Gelenkwinkels γ_4 (Handgelenk 1)

Wenn γ_3 und γ_2 gegeben sind, können wir γ_4 mithilfe der Gleichungen (C.1) und (C.3) und des Rotationsteils der Transformation Q_2 berechnen, der durch $q_4 = w_4 + a_4i + b_4j + c_4k$ gegeben ist.

$$\underline{M_{41} = N_{41}}$$

$$\cos \frac{\gamma_4}{2} = w_4 \cos \frac{\gamma_2 + \gamma_3}{2} - b_4 \sin \frac{\gamma_2 + \gamma_3}{2} \quad (4.42)$$

$$\underline{M_{43} = N_{43}}$$

$$\sin \frac{\gamma_4}{2} = w_4 \sin \frac{\gamma_2 + \gamma_3}{2} + b_4 \cos \frac{\gamma_2 + \gamma_3}{2} \quad (4.43)$$

Wir können γ_4 einfach berechnen, indem wir

$$\gamma_4 = 2 \arctan \left(\frac{w_4 \sin \frac{\gamma_2 + \gamma_3}{2} - b_4 \cos \frac{\gamma_2 + \gamma_3}{2}}{w_4 \cos \frac{\gamma_2 + \gamma_3}{2} + b_4 \sin \frac{\gamma_2 + \gamma_3}{2}} \right) \quad (4.44)$$

Wir betrachten einen Wert für γ_4 , der der Drehung des Handgelenks 1 entspricht.

4.3.5 Gelenkwinkelösungen

Mithilfe der Gleichungen (4.11), (4.13), (4.28), (4.38), (4.41) und (4.44) können wir für eine Winkelkombination $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6)$ berechnen, dass dies der Fall ist. Geben Sie uns unsere gewünschte Transformation Q_1 . Jeder unserer Gelenkwinkel aus diesen Gleichungen hat die folgende Anzahl möglicher Werte:

$$\begin{array}{ccc} \gamma_1 \times 2 & \gamma_2 \times 1 & \gamma_3 \times 2 \\ \gamma_4 \times 1 & \gamma_5 \times 2 & \gamma_6 \times 1 \end{array}$$

Somit gibt es insgesamt $2 \times 1 \times 2 \times 1 \times 2 \times 1 = 8$ mögliche Gelenkwinkelkombinationen. Es ist immer noch notwendig, diese Werte in unserer praktischen Programmieranwendung etwas zu filtern, da es immer noch vorkommen kann, dass nur einige dieser Kombinationen genau die gewünschte Transformation liefern. In dieser Studie wird dazu das Prinzip der Vorwärtskinematik genutzt. Jetzt erhalten wir als Eingabe die Gelenkwinkel aus unserer aktuellen Kombination und erhalten die entsprechende Transformation.

$$Q_1(\gamma_1) \cdot Q_2(\gamma_2) \cdot Q_3(\gamma_3) \cdot Q_4(\gamma_4) \cdot Q_5(\gamma_5) \cdot Q_6(\gamma_6) = Q_{FK} \quad (4.45)$$

Wir prüfen, ob diese Transformation unserer gewünschten Transformation entspricht, wobei wir auch einige kleine Abweichungen aufgrund numerischer Fehler berücksichtigen müssen. Dieses Verfahren wird in unserer Programmieranwendung demonstriert.

4.4 Praktische Programmieranwendung

Die Python-Programmieranwendung zum Ableiten der Vorwärts- und Rückwärtsbewegung
Die Kinematik von UR3 wird in diesem Abschnitt vorgestellt.

4.4.1 Denavit-Hartenberg-Parameter

Die Denavit-Hartenberg-Parameter von UR3 sind in der folgenden Tabelle aufgeführt:

UR3				
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1519	$\pi/2$
Joint 2	0	-0.24365	0	0
Joint 3	0	-0.21325	0	0
Joint 4	0	0	0.11235	$\pi/2$
Joint 5	0	0	0.08535	$-\pi/2$
Joint 6	0	0	0.0819	0

Tabelle 4.2: Denavit-Hartenberg-Parameter für UR3 aus [20]

Diese Parameter werden durch das folgende Diagramm beschrieben:⁵

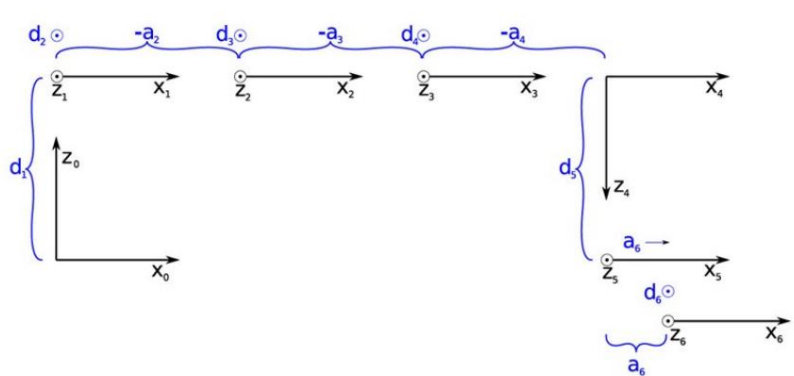


Abbildung 4.7: UR3 DH-Parameter Diagramm aus [20]

Die Einrichtung der DH-Parameter im Programm folgt der Reihenfolge in der Tabelle
4.1 unter Verwendung der in Tabelle 4.2 angegebenen Werte.

```
1 # UR3 Denavit-Hartenberg Parameter
2 d = [0.1519, 0.0, 0.0, 0.11235, 0.08535, 0.0819]
3 a = [0.0, -0.24365, -0.21325, 0.0, 0.0, 0.0]
alpha = [0, pi/2, 0, 0, pi/2, pi/2]
```

⁵Die Verwendung und Ableitung der Denavit-Hartenberg-Parameter findet sich in [23]

4.4.2 Python-Programmieranwendung

Wir verwenden die urx-Python-Bibliothek zur Kommunikation und Steuerung des UR3-Roboters.

```
1 Urx importieren
2 von urx import urrobot
3 rob = urx . Roboter ( " 1 4 7. 2 2 9. 1 3 2. 2 4 9" ) #IP-Adresse von UR3
```

Die IP-Adresse des UR3-Roboters ist auf der Registerkarte „Über“ im UR3 zu sehen

Grafische Benutzeroberfläche von Polyscope.

Duale Quaternion-Klasse

Wir definieren zunächst unser duales Quaternionobjekt. Wie wir beschrieben haben, besteht es von zwei Quaternionen, wobei die erste Quaternion die Rotation darstellt und die zweite Quaternion repräsentiert die Übersetzung. Die Pyquaternion-Bibliothek ist zur Definition dieser Quaternionobjekte verwendet und wir haben eine Klasse für das Dual erstellt Quaternion, das den Eigenschaften folgt, die wir im Kapitel Quaternion aufgeführt haben. Der Multiplikation dualer Quaternionen wird in dieser Klasse auch im Anschluss an definiert Eigenschaften, die wir im Kapitel über duale Quaternionen aufgelistet haben.

```
1 Klasse DualQuaternion:
2 # Ich initialisiere ein duales Quaternion
3     def __init__(self, q=Quaternion(), p=0):
4         self.q = q
5         self.p = p
6
7 # Duale Quaternion-Multiplikation
8     def mult(self, dq):
9         # Für den Rotationsteil
10            q1 = self.q
11            q2 = dq.q
12
13        # Für den Übersetzungsteil
14            p1 = self.p
15            p2 = dq.p
16
17        Rückgabe DualQuaternion ( q1*q2 , q1*p2 + q2*p1 . konjugiert e+p1 )
18
19 # Prüft die Gleichheit von Dualquaternen
20     def gleich(self, dq):
21         # Für den Rotationsteil
22            q1 = self.q
23            q2 = dq.q
24
25        # Für den Übersetzungsteil
26            p1 = self.p
27            p2 = dq.p
```

4.4. Praktische Programmieranwendung

```

29 # Prüft die Gleichheit
30 print ( „\nVergleich von Dual-Quaternionen:“ )
31 drucken ( „Drehung:“, q1, print, „Übersetzung:“, p1,
32 ( , p2 )
33
34 if ( ( np . allclose ( p1 , p2 , atol=1e-03)
35         oder np . allclose ( p2 , p1 , atol=1e-03) )
36     und ( np . allclose ( q1 , q2 , atol=1e-03)
37         oder np . allclose ( q2 , q1 , atol=1e-03) ) ):
38     Rückgabe True
39     return False
40
41 # Verwandelt sich im Raum
42 def transform (self, pose):
43     # Gedrehte Pose
44     rotpose = self . Q . drehen (posieren)
45     print ( „\nAfte r Rotation: rotpose)
46
47     # Übersetzte Pose
48     tranpose = self . P . Vektor + rotpose
49     print ( „Nach der Translation: tranpose)
50
51     Rückübertragung _

```

Beim Drehen eines Vektors haben wir die Pyquaternion-Methode `Rotieren` verwendet, da dies auch der Fall ist **normalisiert die Quaternion, bevor die Rotation angewendet wird. Dies erfolgt im Folgende Codes in der Pyquaternion-Bibliothek:**

```

1 def rotieren (Selbstvektor):
2     Drehen Sie einen 3D-Vektor um die im gespeicherte Drehung
3     Quaternion-Objekt.
4
5     Parameter:
6         Vektor: Ein 3yV -Ektor, der als beliebige geordnete Sequenz spezifiziert ist
7         von 3 reellen Zahlen entsprechen Di-Ngtox- und Z-Werten.
8         Einige Typen, die erkannt werden, sind: Numpy-Arrays
9         , Listen und Tuplel.
10        Ein 3yVektor kann auch durch a dargestellt werden
11        Quaternion-Objekt, dessen Skalarteil 0 und Vektorteil ist
12        der erforderliche rote 3yV ektor.
13        Somit ist ein Anruf möglich ' Quaternion . drehen ( q
14        ) ' mit einem oder mehreren Objekten als Eingabe .
15
16    Kehrt zurück :
17        Der gedrehte Vektor wurde im selben Typ zurückgegeben, den er hatte
18        angegebeninput .
19
20    Raises:
21        TypeError: Wenn eines der Vektorelemente nicht möglich ist
22        umgewandelt in eine reale Zahl.

```

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

```

15 Wertfehler: if ' vector 3yv ' kann nicht so interpretiert werden
    ectorora Quaternion object .
16
17
18 if isinstance (Vektor, Quaternion):
19     selbst zurückgeben . Rotationsquaternion (Vektor)
20     q = Quaternion (Vektor=Vektor)
21     a = selbst . Rotationsquaternion ( q ) . Vektor
22     if isinstance (Vektorliste ) : ,
23         l = [ x für x in a ]
24         gib l zurück
25     elif isinstance ( vectorl = [ x for , tu pl e ):
26         x in a ]
27         return tu pl e ( l )
28     sonst :
29         zurückgeben a
30
31 def _rotatequaternion (self, q):
32     Drehen Sie Wasser auf Vektorbasis mit der gespeicherten Drehung.
33
34     Parameter:
35     q: Der zu drehende Vektor , in quaterni on auf Form ( 0 +
    xi + yj + kz )
36
37     Kehrt zurück :
38     Ein Quaternion-Objekt, das den gedrehten Vektor darstellt
    in quaterni on from ( 0 + xi + yj + kz )
39
40     selbst. norm ali se ( )
41     return self y q y self . konjugieren
42
43 def norm ali se (self):
44     """ Das Objekt muss danach gültig sein
    ruft dies an
45     Oper ti on, ES SEI DENN, das objektive ui v al entto Quaternion
    ( 0 )
46
47     wenn nicht selbst . isunit() :
48         n = selbst . Norm
49         wenn n > 0 :
50             selbst. q = selbst . q / n

```

Wir stellen außerdem fest, dass wir im vorherigen Code definiert haben, wie ermittelt werden kann, ob zwei vorhanden sind. Duale Quaternionen sind ungefähr gleich. Wir haben die Allclose-Methode verwendet, die die Numpy-Bibliothek, die prüft, ob zwei Arrays innerhalb von a elementweise gleich sind Toleranz. Wir machen dies für die inverse Kinematik, wenn wir herausfiltern möchten die resultierenden Winkelkombinationen durch ihre entsprechenden Transformationen und Wir müssen die numerischen Fehler berücksichtigen, die bei Berechnungen auftreten können.

4.4. Praktische Programmieranwendung

Vorwärtskinematik

Für die Vorwärtskinematik ist es unser Ziel, die endgültige Transformation zu bestimmen (Ausrichtung und Position) des Endeffektors des Roboters aufgrund seiner Gelenke Winkeldrehungen. Zuerst müssen wir die aktuellen Gelenkwinkel unseres Roboters ermitteln. Wir tun dies, indem wir die Methode `getj` aus der `urx`-Bibliothek verwenden.

```
1 # Ruft die aktuellen Jointtangs ab
2 Jangles = rauben . getj ()
```

Dies gibt den Wert der sechs Gelenkwinkel von der Basis bis zum Handgelenk zurück
3. Wir definieren außerdem zwei Ansätze zur Ableitung einer dualen Quaternion. Einer ist vorbei der Achsen-Winkel-Darstellung (AA) und durch die Transformationsmatrix (TM). Für jeden dieser Ansätze definieren wir unterschiedliche anfängliche TCP-Koordinaten.

```
1 # Setzt die TCP-Anfangsposition im Raum basierend auf dem entsprechenden Wert
   rahmen
2 TCP_endf = np . array ( [ 0 3 TCP_basef = np . array ( 0 , 0 , 2 ] ) # basierend auf Endframe ( 0 , 0 , 1 )
( [ 0 - , y0 . 2 , 0 ] ) # basierend auf Basisrahmen ( 0 , y1 , 0 )
```

Der Grund dafür ist, dass wir den ersten Ansatz von AA berücksichtigen müssen die TCP-Koordinaten in Bezug auf den Basisrahmen. Und für das TM Ansatz müssen wir die TCP-Koordinaten in Bezug auf seinen Rahmen berücksichtigen.

Die Methode Vorwärtskinematik nennt nun die Ableitung des Dualen Quaternionen für jeweils sechs Gelenke basierend auf dem gewählten Ansatz und berechnet Die letzte duale Quaternion stellt die endgültige Transformation durch Multiplikation dar sie folgen der Reihenfolge der Transformation (Gelenke).

```
1 # Ruft die aktuelle Position unter Verwendung von Vorwärtskinematiken ab
2 def getforward_Kinematiken (ans, jangles):
3
4     print ( „\nAktuelle Verbindungswinkel: “ , klimpert)
5
6 # Ergibt die entsprechenden Dual-Quaternionen für jedes Gelenk
   aus ihrer Winkelachsendarstellung
7     if ( ans == „AA“ ) :
8         print ( „\nBerechnung der Vorwärtskinematik aus der Winkelachse
   Repräsentation. . . “ )
9         DualQuatList = das aktuelle Quat von AA (Jangles) -
10        initialpose = TCP_basef -
11 # Ergibt die entsprechenden Dual-Quaternion-Sequenzen für jedes Gelenk
   aus ihren Transformation Matrices
12     elif ( ans == „TM“ ) :
13         print ( „\nAbleitung von Vorwärtskinematiken aus Transformation
   Matrix. . . “ )
14         DualQuatList = Ableiten von Dualquatern aus TM (Jangles) -
15         initialpose = TCP_endf -
16     sonst :
17         print ( „Falsche Eingabe!“ )
```

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

```

18     zurückkehren
19
20 # Berechnet die endgültige duale Quaternion durch Multiplikation der
    Dual Quaternionen
21 FKDualQuat = DualQuaternion ( )
22 für dq in DualQuatList:
23     FKDualQuat = FKDualQuat . mult (dq)
24
25 print ( "\nForward Kinematics (by Dual Quaternion) : \n( [ "
    FKDualQuat . q , " ] , , FKDualQuat . p , "> )" )
26
27 # Anfangsposition durch Quaternion-Vektor- Multiplikation transformieren „
28 print ( „\n Initiale TCP-Koordinaten (atzerop se): initialpose) „ ,
    -
29 currpose = FKDualQuat . transformieren (initialisieren) -
30 print ( „\nTransformierte TCP-Koordinaten: \n“ , np . Round ( currpose -
    , 3 ) )
31
32 gebe FKDualQuat zurück

```

Die Ausgabe dieser Methode ist die aktuelle Pose oder die TCP-Koordinaten von der Roboter, nachdem er transformiert (gedreht und verschoben) wurde und mit in Bezug auf den Grundrahmen.

Duale Quaternionen aus der Achsen-Winkel-Darstellung

Die Ableitung der Vorwärtskinematik durch den Achsenwinkel-Ansatz ist Dies geschieht mit der Methode, Dual-Quat von AA abzuleiten:

```

1 # Leiten Sie das duale Quaternion aus Winkel und Drehachse ab
2 def derivDualQuatFromAA (Jangles):
3 # Ruft Verbindungswinkel aus der aktuellen Position ab
4     Theta = Klirren
5
6 # Stellen Sie die Übersetzung für jedes Gelenk ein
7     trans = ( [ [ 0 d [ 0, ] ] , 0 [ a [ 2 ]
8                 y cos ( theta [ 1 ] ) , [ a [ 3 ]
9                 y cos ( theta [ 2 ] ) , [ d [ 4 ]
10                y sin ( theta [ 3 ] ) , [ yd [ 5 ]
11                y sin ( theta [ 4 ] ) , [ 0 0 ] ]
12                , 0 ,
13
14 # Baut die Dual-Quaternionen auf
15     DQList = [ ]
16     al p = 0
17
18     für mich in range ( 6 ):
19         # Baut das Dual-Quartier auf
20         qscalar = cos ( theta [ i ] / 2 )

```

4.4. Praktische Programmieranwendung

```

21     alp += alph a [ i ]
22     qvector = np . m u l t i p l y (sin(theta[i]/2),alp))          [ 0 , ȳsin ( alp ) ,      weil (
23
24     quatRot = np . verketten ([[qscalar], qvector))
25
26     DualQuat = DualQuaternion ( Quaternion ( quatRot ) , Quaternion (
27     Vektor=trans [ i ] ) )
28     print ( "Dual Quaternion : ( [ "          , DualQuat . q , " ] , , DualQuat . p , ">"
29     " ) )
30
31     # Speichern Sie die Dual-Quaternion in der Liste
32     DQList . anhängen (DualQuat)
33
34     DQList zurückgeben

```

Um die endgültige Transformation zu berechnen, definieren wir zunächst die Übersetzung für jedes Gelenk. Dies entspricht dem Übersetzungsteil unserer dualen Quaternionen in dem wir die Denavit-Hartenberg-Parameter (a , d , $\dot{y}$) verwenden. Dann wir Erstellen Sie die Rotationsquaternionen für alle sechs Gelenke mithilfe der Winkel vom Roboter und seine entsprechende Übersetzungsquaternion durch die abgeleiteten Übersetzungen. Der Die Variable alp folgt dem Alpha-Parameter in den DH-Parametern und wir verwenden dies bei der Bestimmung der Drehachse. Dies kann die Werte 0 oder $\frac{\pi}{2}$ which haben macht einen der Terme $-\sin(alp)$ und $\cos(alp)$ zu Null oder ± 1 . Dann, Wir erstellen das Dual-Quaternion-Objekt, indem wir diese beiden Quaternionen kombinieren. Zuletzt speichern wir diese dualen Quaternionen in einer Liste und geben sie an die vorherige zurück Methode, die die Transformation durchführt.

Duale Quaternionen aus der Transformationsmatrix

Ein weiterer Ansatz zur Ableitung der Vorwärtskinematik kann eine Transformation sein Matrix. Dies geschieht mit der Methode Dual-Quat aus TM ableiten. Das Übliche Die Konvention zur Ableitung der Vorwärtskinematik besteht in der Verwendung von Rotationsmatrizen und Übersetzungsvektoren, die zu einer (4x4)-Transformationsmatrix zusammengefasst werden. Aus dieser Transformationsmatrix können wir sie in eine duale Quaternion umwandeln durch Umwandlung der Rotationsmatrizen (3x3) in Einheitsquaternionen und der Translationsvektoren (3x1) als reine Quaternion. Die Rotationsmatrix zur Quaternion Die Konvertierung wurde in Abschnitt 3.4.2 besprochen. Für die Transformationsmatrix wir Folgen Sie seiner Herleitung in [3] und [16].

```

1 # Ruft die Dual-Quadrat-Elemente für jedes Gelenk ab
2 def leiten Dual.Qu a te rni on von TM (jangles) ab:
3     print ( „Computing T r a n s f o r m a t i o n Matrix von Denavit-Ha r tenbe rg
4     Parameter . . . “ )
5
6 # Ruft Verbindungswinkel aus der aktuellen Position ab
7     Theta = Klirren
8
9 # Ruft Transformationsmatrixen für alle 6 Frames ab

```

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

```

7 T = np . Nullen ( Form = ( 6
                                , 4 , 4 ) )
8
9 # Errichtet die Dualquaternionen
10 DQList = []
11
12 für mich in range ( 6 ):
13     T[i] = ( [[ cos ( theta [ i ] ) , ysin ( theta [ i ] ) , a [ i ] ] [ sin ( theta [ i ] ) y cos (
14                 ( alpha a [ i ] ) , cos ( Theta [ i ] ) y cos (
15     alph a [ i ] ) , ysin ( alph a [ i ] ) , ysin ( alph a [ i ] ) yd [ i ] ] [ sin ( theta [ i ] ) ,
16                 y sin ( alph a [ i ] ) , cos ( theta [ i ] ) y sin (
17     alpha a [ i ] ) , cos ( alph a [ i ] ) , cos ( alph a [ i ] ) yd [ i ] ] ,
18                 [ 0 , 0 , 0 , 1 ] ] )
19
20     print ( „\nTransformation Matrix for Frame“ , „Zu“ , i + 1 , „\n“
21           , T[i] )
22
23     # Leitet die Duale Quaternion aus der Transformation ab
24     Matrix
25     DualQuat = DualQuaternion ( Quaternion ( matrix = T[i] ) ,
26     Quaternion ( Vektor = T[i] [ 0 : 3 , 3 ] ) )
27     print ( "Dual Quaternion : ( " , DualQuat . q , " , " , DualQuat . p , ">"
28           " )
29
30     # Speichern Sie die Dual-Quaternion in der Liste
31     DQList . anhängen ( DualQuat )
32
33 DQList zurückgeben

```

Wie beim vorherigen Ansatz speichern wir die dualen Quaternionen in einer Liste und übergeben Sie es an die erste Methode, die die Transformation durchführt. Das letzte Dual Quaternion, das die endgültige Transformation darstellt, wird sich von der unterscheiden anderer Ansatz. Dies liegt daran, dass wir in der Ableitung von AA das zugrunde legen dualen Quaternionen nur auf dem Grundrahmen, während für den TM-Ansatz jeweils Dualen Quaternionen werden in Bezug auf ihre eigenen Rahmen abgeleitet. In jedem von Bei diesen Ansätzen sind die endgültigen TCP-Koordinaten immer dieselben.

Praktische Anwendung

Für die Anwendung verwenden wir eine Methode Record Pose, die den Strom annimmt Pose des Roboters und zeichnet sie in einer Textdatei auf. Die Koeffizienten dieses Duals Die aus den Gelenkwinkeln des Roboters abgeleitete Quaternion wird dann in einem Text gespeichert Datei. In der Textdatei können mehrere Posen aufgezeichnet werden. Abgesehen davon ist die Die Vorwärtskinematik wird auch zum Filtern der abgeleiteten Winkelkombinationen verwendet die inverse Kinematik, indem überprüft wird, ob ihre Transformationen dem entsprechen gewünschte Transformation in die Textdatei geschrieben.

Inverse Kinematik

Für die inverse Kinematik besteht unser Ziel darin, die entsprechenden Gelenkwinkel zu bestimmen. Geben Sie uns unsere gewünschte Pose (Rotation und Translation). Somit ist es uns nun gegeben die endgültige Transformation und daraus leiten wir die Werte der entsprechenden sechs Gelenkwinkel ab. Zur Durchführung verwenden wir die Methode „Get Inverse Kinematics“. die Ableitung. Wir beginnen mit der Definition der Koordinaten bei Frame 6 (Handgelenk 3). und die entsprechenden Koordinaten bei Frame 5 (Handgelenk 2).

```

1 # Gibt Gelenkwinkel von inversen Kinematiken aus
2 def getinversekinematics (gewünschte Pose):
3     print ( "\nDe si red Pose: gewünschte Pose. q, - gewünschte Pose. P ) ;
4 # Gewünschte Transformation von Frame 1 bis 6
5 DQ16 = gewünschte Pose
6 # Rotation Teil
7     w6 = DQ16. q [ 0 ]
8     a6 = DQ16. q [ 1 ]
9     b6 = DQ16. q [ 2 ]
10    c6 = DQ16. q [ 3 ]
11 # Translation Teil
12    x6 = DQ16. p[1];
13    y6 = DQ16. p[2];
14    z6 = DQ16. p [3];
15
16 # Getriebe aus Rahmen 5 entfernen
17    DQ5i = DualQuaternion ( p=Quaternion ( Vektor =[0 , d [ 5 ] , 0 ] ) )
18    DQ15p = DQ16. mult (DQ5i )
19    x5 = DQ15p . p [ 1 ]
20    y5 = DQ15p . p [ 2 ]
21    z5 = DQ15p . p [ 3 ]

```

Die Übersetzung von Handgelenk 3 zu Handgelenk 2 wurde in Abschnitt 4.3.3 demonstriert.

Berechnung für Basis ($\ddot{y}_1$)

Um $\ddot{y}_1$ zu berechnen, haben wir die Gleichung (4.11) verwendet:

$$\ddot{y}_1 = \text{atan2}(\ddot{y}_x5, y_5) \pm \text{atan2} \left(x_5 \sqrt{\ddot{y}_x5^2 + y_5^2} \ddot{y}_d [3] \ddot{y}_d [2] \ddot{y}_d [3] \right) + J \sqrt{\ddot{y}_x5^2 + y_5^2} \ddot{y}_d [4] \ddot{y}_d [4]$$

Dies wird durch den folgenden Code ausgedrückt:

```

1 # Berechnen Sie mögliche Werte für Theta 1
2     Theta 1 =[]
3     t 1 1 = atan2 ( \ddot{y}_x5 , y_5 )+atan2 ( sqrt ( x_5\ddot{y}_x5^2+y_5^2\ddot{y}_d [ 3 ] \ddot{y}_d [ 2 ] , \ddot{y}_d [ 3 ] ) )
4     t 1 2 = atan2 ( \ddot{y}_x5 , y_5 )\ddot{y}_d [ 3 ] \ddot{y}_d [ 2 ] , \ddot{y}_d [ 3 ] )
5 # (Basis) Theta 1: 2 Werte
6     Theta 1 . anhängen ( t 1 1 )
7     Theta 1 . anhängen ( t 1 2 )

```

Dies ergibt höchstens zwei mögliche Werte für $\ddot{y}_1$.

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Berechnung für Handgelenk 2 ($\ddot{\gamma}_5$)

Um $\ddot{\gamma}_5$ zu berechnen, haben wir die Gleichung (4.13) verwendet:

$$\ddot{\gamma}_5 = \pm \arccos \frac{x_6 \sin \ddot{\gamma}_1 \ddot{\gamma}_6 \cos \ddot{\gamma}_1 \ddot{\gamma}_4}{d_6}$$

Dies wird durch den folgenden Code ausgedrückt:

```

1 # Berechnen Sie mögliche Werte für Theta 5
2   Theta 1 5 = []
3   für t 1 in Theta 1: # Für jeden Wert von Theta 1
4       # Ev a te express für Theta 5
5       exp5 = ( x6 * sin ( t 1 ) * y6 * cos ( t 1 ) ) / d [ 3 ]
6       # Prüft, ob mit der Domain for arccos
7       if ( exp5 <= 1 und exp5 >= -1 ):
8           t 5 = acos ( exp5 )
9       elif ( round ( exp5 ) <= 1 und Round ( exp5 ) >= -1 ):
10          t 5 = acos ( round ( exp5 ) )
11       Sonst : Fortsetzung
12   # (Handgelenk 2) Theta 5: 2 Werte
13   Theta 1 5 . anhängen ( [ t 1 t 5 ] )
14   Theta 1 5 . append ( [ t 1 , t 5 ] )

```

Wir gehen davon aus, dass Arccos nur eine Domäne von $[-1, 1]$ hat. Daher zur Vorbeugung Bei Laufzeitfehlern runden wir die Werte ab und prüfen, ob sie nun darin enthalten sind Domain. Dies ergibt wiederum zwei mögliche Werte für $\ddot{\gamma}_5$.

Berechnung für Handgelenk 3 ($\ddot{\gamma}_6$)

Um $\ddot{\gamma}_6$ zu berechnen, haben wir die Gleichung (4.28) verwendet:

$$\ddot{\gamma}_6 = 2 \arctan 2(\ddot{\gamma}_1 \ddot{\gamma}_5, \ddot{\gamma}_1 + \ddot{\gamma}_5)$$

Wo,

$$\ddot{\gamma}_1 = \frac{w c^{-1} + c s^{-1}}{c^{-5}} \quad \ddot{\gamma}_5 = \frac{a c^{-1} + b s^{-1}}{s^{-5}} \quad \ddot{\gamma}_1 = \frac{a s^{-1} \ddot{\gamma}_1 b c^{-1}}{c^{-5}} \quad \ddot{\gamma}_5 = \frac{w s^{-1} \ddot{\gamma}_1 c c^{-1}}{s^{-5}}$$

Dies wird durch den folgenden Code ausgedrückt:

```

1 # Berechnen Sie mögliche Werte für Theta 6
2   Theta 1 5 6 = []
3   für t 1 t 5 in Theta 1 5 :
4       t 1 = t 1 t 5 [ 0 ]
5       t 5 = t 1 t 5 [ 1 ]
6
7   # Vermeidet Singularität für Theta 6
8   if ( t 5 == 0 ) :

```

4.4. Praktische Programmieranwendung

```

9      t156 = [ t1 , t5 , 0 ]
10     wenn t156 nicht in Theta156 :
11         Theta156 . anhängen ( t156 )
12     weitermachen
13
14     # Evaluate express für Theta 6
15     tm1 = ( ( w6y cos ( t1 / 2 ) + c6 y sin ( t1 / 2 ) ) / cos ( t5 / 2 ) )
16     tm2 = ( ( a6y cos ( t1 / 2 ) + b6y sin ( t1 / 2 ) ) / sin ( t5 / 2 ) )
17     tm3 = ( ( a6y sin ( t1 / 2 ) - b6y cos ( t1 / 2 ) ) / cos ( t5 / 2 ) )
18     tm4 = ( ( w6y sin ( t1 / 2 ) - c6 y cos ( t1 / 2 ) ) / sin ( t5 / 2 ) )
19
20     # (Handgelenk 3) Theta 6: 1 Wert
21     t6 = 2y atan2 ( tm3, tm2 )
22     Theta156 . anhängen ( [ t1 t6 ] ) ,

```

Dies ergibt nur einen möglichen Wert für y_6 .

Nun zur Definition der nächsten Winkel. Wir müssen eine Übersetzung durchführen wieder. Wir müssen die Transformation von Rahmen 2 (Schulter) zu Rahmen durchführen 4 (Handgelenk 1), daher benötigen wir ~~4.2~~ Dies wurde in Abschnitt 4.3.4 besprochen.

```

1 # Transformation von Frame 2 zu Frame 4 ableiten
2 DQ24f = [ ]
3 für t156 in Theta156 :
4     t1 = t156 [ 0 ]
5     t5 = t156 [ 1 ]
6     t6 = t156 [ 2 ]
7     # Derives das inverse Dualquadrat aus Joints 1 6 , 5 ,
8     q1 = np . verketten ( ( [ cos ( t1 / 2 ) ] , np . multipl y ( sin ( t1 / 2 ) ,
9     [ 0 , 0 , y1 ] ) ) )
10    DQ1i = DualQuaternion ( Quaternion ( q1 ) , Quaternion ( vector
11    =[ 0 , 0 , yd [ 0 ] ] ) )
12    q5 = np . verketten ( ( [ cos ( t5 / 2 ) ] , np . multipl y ( sin ( t5 / 2 ) ,
13    [ 0 , 0 , 1 ] ) ) )
14    DQ5i = DualQuaternion ( Quaternion ( q5 ) , Quaternion ( Vektor =[ 0 ,
15    d [ 5 ] , 0 ] ) )
16    q6 = np . verketten ( ( [ cos ( t6 / 2 ) ] , np . multipl y ( sin ( t6 / 2 ) ,
17    [ 0 , 1 , 0 ] ) ) )
18    DQ6i = DualQuaternion ( Quaternion ( q6 ) , Quaternion ( vector
19    =[ 0 , 0 , 0 ] ) )
20    # Derives Transformation von Frame 2 bis Frame 6
21    DQ26 = DQ1i . multi ( DQ16 )
22    # Derives Transformation Von Frame 2 bis Frame 5
23    DQ25 = DQ26 . mult ( DQ6i )
24    # Derives Transformation von Frame 2 bis Frame 4
25    DQ24 = DQ25 . mult ( DQ5i )
26    # Entfernen Sie das Getriebe von Rahmen 4
27    DQ4i = DualQuaternion ( p=Quaternion ( vector =[ 0 DQ24f . append ( 0 , d [ 4 ] ) ) )
28    ( [ DQ24 . mult ( DQ4i ) , t156 ] )

```

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Für jede Winkelkombination von $\ddot{y}_1$, $\ddot{y}_5$ und $\ddot{y}_6$ leiten wir die entsprechende abduale Quaternion $Q_{\ddot{y}_1}^{-1}$, $Q_{\ddot{y}_5}^{-1}$ und $Q_{\ddot{y}_6}^{-1}$ ab. Wir verwenden diese Umkehrungen zur Ableitung der Transformation Q_2 . 4. Dann machen wir die von Q_4 vorgenommene Übersetzung um ebenfalls rückgängig unter Verwendung seiner Umkehrung Q_4^{-1} . Die Transformationen werden dann in einer Liste gespeichert mit dem Winkelsatz $[\ddot{y}_1, \ddot{y}_5, \ddot{y}_6]$, der sie erzeugt hat. All dies wird der Fall sein wird bei der Berechnung der verbleibenden Gelenkwinkel verwendet.

Berechnung für Ellenbogen ($\ddot{y}_3$)

Um $\ddot{y}_3$ zu berechnen, haben wir die Gleichung (4.38) verwendet:

$$\ddot{y}_3 = \pm \arccos \frac{x_4^2 + z_4^2 - \ddot{y}_a^2 - \ddot{y}_{a3}^2}{2a_2a_3}$$

Dies wird durch den folgenden Code ausgedrückt:

```

1 # Berechnen Sie mögliche Werte für Theta 3
2   das ta 1 3 5 6 q = []
3   für dq in DQ24f :
4       w4 = dq [ 0 ] . q [ 0 ]
5       b4 = dq [ 0 ] . q [ 2 ]
6       x4 = dq [ 0 ] . p [ 1 ]
7       y4 = dq [ 0 ] . p [ 2 ]
8       z4 = dq [ 0 ] . p [ 3 ]
9       q4 = [ w4 b4 , x4 , y4 , z4 ]
10
11      t 1 = dq [ 1 ] [ 0 ]
12      t 5 = dq [ 1 ] [ 1 ]
13      t 6 = dq [ 1 ] [ 2 ]
14
15  # Ev a te express für Theta 3
16  exp3 = ( ( x42+z42- $\ddot{y}_a^2$ - $\ddot{y}_{a3}^2$  ) / ( 2 $\ddot{y}_a$  [ 2 ]  $\ddot{y}_a$  [ 3 ] ) )
17
18  # Prüft, ob mit der Domain forarccos
19  if ( exp3<=-1 und exp3>=1 ):
20      t 3 = acos ( exp3 )
21  elif ( round ( exp3 )<=-1 und Round ( exp3 )>=1 ):
22      t 3 = acos ( round ( exp3 ) )
23  Sonst : Fortsetzung
24
25  # (Ellenbogen) Theta 3: 2 Werte
26  die ta 1 3 5 6 q . append ( [ t1 t5 t3 t4 , q4 ] )
27  3 5 6 q . append ( [ t1 , t3 , t5 t6 , q4 ] )

```

Das Gleiche gilt für $\ddot{y}_5$, wir gehen davon aus, dass die Ergebnisse innerhalb des Arccos-Bereichs liegen. Dies ergibt höchstens zwei mögliche Werte für $\ddot{y}_3$.

4.4. Praktische Programmieranwendung

Berechnung für Schulter ($\ddot{\gamma}_2$)

Um $\ddot{\gamma}_2$ zu berechnen, haben wir die Gleichung (4.41) verwendet:

$$\ddot{\gamma}_2 = \text{atan2}(z_4(a_2 + a_3 \cos \ddot{\gamma}_3) \ddot{\gamma} x_4 a_3 \sin \ddot{\gamma}_3, x_4(a_2 + a_3 \cos \ddot{\gamma}_3) + z_4 a_3 \sin \ddot{\gamma}_3)$$

Dies wird durch den folgenden Code ausgedrückt:

```
1 # Berechnen Sie mögliche Werte für Theta 2
2   die ta 1 2 3 5 6 q =[]
3   für t1356q im ta 1 3 5 6 q :
4       t 1 = t1356q [ 0 ]
5       t 3 = t1356q [ 1 ]
6       t 5 = t1356q [ 2 ]
7       t 6 = t1356q [ 3 ]
8       x4 = t1356q [ 4 ] [ 2 ]
9       z4 = t1356q [ 4 ] [ 4 ]
10      # Es handelt sich um einen Ausdruck für Theta 2
11      t 2 = atan2 ( ( z4 \ddot{\gamma}( a [ 2]+ a [ 3 ] \ddot{\gamma} \cos ( t 3 ) )\ddot{\gamma}x4\ddot{\gamma}a [ 3 ] \ddot{\gamma} \sin ( t 3 ) ) , ( x4 \ddot{\gamma}( a
[ 2]+ a [ 3 ] \ddot{\gamma} \cos ( t 3 ) )+z4 \ddot{\gamma}a [ 3 ] \ddot{\gamma} \sin ( t 3 ) ) )
12      # (Schulter) Theta 2: 1 Wert
13      die ta 1 2 3 5 6 q . anhängen ( [ t1 t5 t2 , t3 , , t6 , t1356q [ 4 ] ] )
```

Dies ergibt nur einen möglichen Wert für $\ddot{\gamma}_2$.

Berechnung für Handgelenk 1 ($\ddot{\gamma}_4$)

Um $\ddot{\gamma}_4$ zu berechnen, haben wir die Gleichung (4.44) verwendet:

$$\ddot{\gamma}_4 = \ddot{\gamma}_2 \text{atan2}(w_4 s^{-(2+3)} + b_4 c^{-(2+3)}, w_4 c^{-(2+3)} \ddot{\gamma} b_4 s^{-(2+3)})$$

Dies wird durch den folgenden Code ausgedrückt:

```
1 # Berechnen Sie mögliche Werte für Theta 4
2   das ta 1 2 3 4 5 6 =[]
3   für t12356q im ta 1 2 3 5 6 q :
4       t 1 = t12356q [ 0 ]
5       t 2 = t12356q [ 1 ]
6       t 3 = t12356q [ 2 ]
7       t 5 = t12356q [ 3 ]
8       t 6 = t12356q [ 4 ]
9       w4 = t12356q [ 5 ] [ 0 ]
10      b4 = t12356q [ 5 ] [ 1 ]
11      # Ev a te express für Theta 4
12      t 4 = \ddot{\gamma}_2 \ddot{\gamma} \text{atan2} ( ( w4 \ddot{\gamma} \sin ( ( t 2+t 3 ) / 2 ) )+(b4 \ddot{\gamma} \cos ( ( t 2+t 3 ) / 2 ) ) , ( w4 \ddot{\gamma}
\cos ( ( t 2+t 3 ) / 2 ) )\ddot{\gamma}(b4 \ddot{\gamma} \sin ( ( t 2+t 3 ) / 2 ) ) )
13      # (Handgelenk 1) Theta 4: 1 Wert
14      die ta 1 2 3 4 5 6 . anhängen ( [ t1 t3 t2 , , t4 , t5 , t 6 ] )
```

Dies ergibt nur einen möglichen Wert für $\ddot{\gamma}_4$.

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Filtration von Winkelkombinationen und praktische Anwendung

Um die resultierenden Winkelkombinationen zu filtern und zu verifizieren, leiten wir ihre entsprechende Vorwärtskinematik ab, die die endgültige durchgeführte Transformation darstellt durch diese Winkel. Wenn diese Transformation nicht annähernd unserer entspricht gewünschte Transformation, dann filtern wir diese Winkel heraus und lassen nur diese übrig die unsere Gleichheit innerhalb der Toleranz von $1e^{-3}$ erfüllen. Dies geschieht unten.

```

1 # Holen Sie sich die entsprechende endgültige duale Quaternion basierend auf
   Kombination von Winkeln
2 klimpert=[]
3 print ( „\nSatz möglicher Verbindungswinkel: \n“ für Theta , die ta 1 2 3 4 5 6 )
4 im ta 1 2 3 4 5 6 :
5     FinalDQ = getforward kinematiken ( „AA“ , Theta)
6
7     print ( „\nVergleich von Dual-Quaternionen:“ )
8     drucken ( " Gewünschte Pose: \n ( [ " , DQ16. q , " ] , , DQ16. p , ">)" )
9     print ( " Aktuelle Pose: \n ( [ " , FinalDQ . q , " ] , , FinalDQ . p , ">)" )
10
11     if (DQ16. equals ( FinalDQ ) ):
12         print ( „Kriterien BESTANDEN!“ )
13         klirr. anhängen (Theta)
14     else : print ( „FEHLGESCHLAGENE Kriterien!“ )
15
16     print ( "\nEndgültige Verbindungswinkel: " , klimpert )
17
18     Rückkehr jangles

```

Für die Anwendung erstellen wir zunächst das durch das dargestellte duale Quaternion Koeffizienten aus der Textdatei, die mit der Record-Pose-Methode generiert wurde. Dann, Wir wenden die inverse Kinematik an und verwenden die ersten resultierenden Gelenkwinkel als Eingabe für die Bewegung des Roboters. (Ein zusätzliches Kriterium kann ebenfalls erstellt werden um die optimale Gelenkwinkelkombination zu finden)

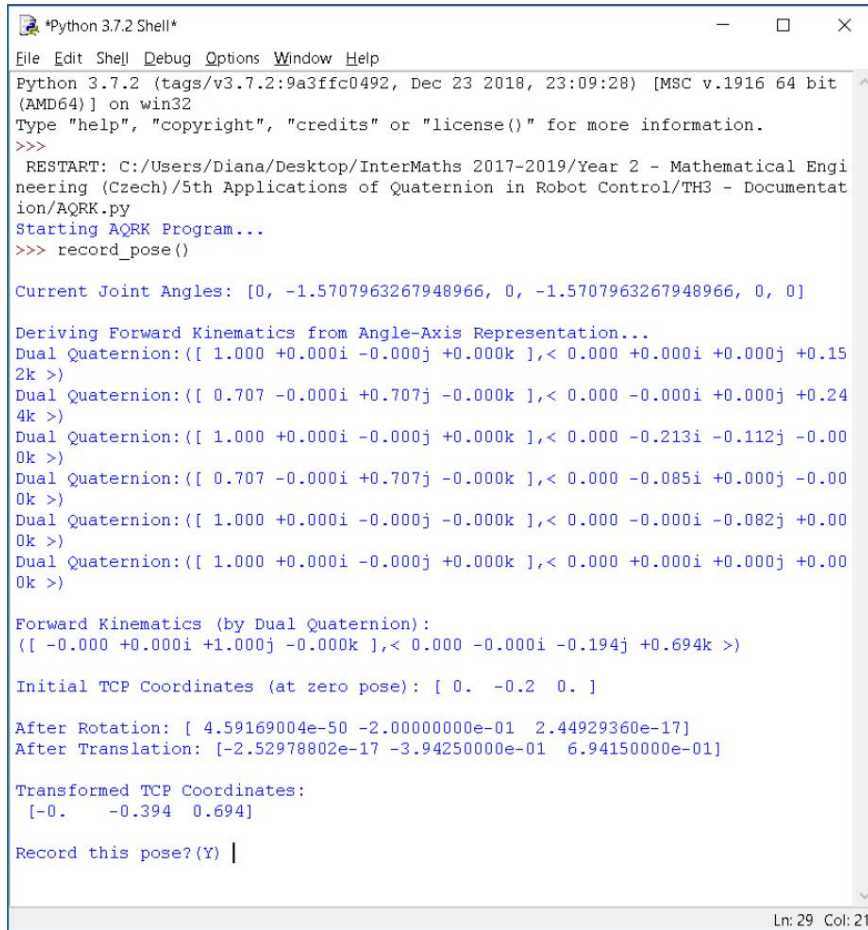
```

1 # Bewegt den Roboter aus der gegebenen Dual-Quaternion-Transformation
2 def move robot (dqcoeff): -
3     # Nimmt die Dual-Quater-Ni-Onkoeffizienten und erstellt die
   du al quater ni on
4     DQ de si redpose = DualQuaternion ( Quaternion ( array=dqcoeff
   [ 0 : 4 ] ), Quaternion ( array=dqcoeff [ 4 : 8 ] ) )
5
6     # Wendet die umgekehrte Kinematik an, um die Gelenkwinkel anzutreiben
7     jointangles = getinversekinematics ( DQ de si redpose )
8
9     print ( "\nEs gibt " , len (Gelenkwinkel) , " mögliche Posen ! "
   );
10
11     # Bewegt den Roboter anhand der Gelenkwinkel
12     rauben . movej ( jointangles [ 0 ] acc =1, v el =1, wai t=F al se )

```

4.5 Überprüfung der Ergebnisse

Das Programm beginnt mit der Funktion „Pose aufzeichnen“:



```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Diana/Desktop/InterMaths 2017-2019/Year 2 - Mathematical Engi
neering (Czech)/5th Applications of Quaternion in Robot Control/TH3 - Documentat
ion/AQRK.py
Starting AQRK Program...
>>> record_pose()

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.15
2k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.000i +0.000j +0.24
4k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 -0.213i -0.112j -0.00
0k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.085i +0.000j -0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j -0.000k ],< 0.000 -0.000i -0.082j +0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.00
0k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ],< 0.000 -0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [ 4.59169004e-50 -2.00000000e-01 2.44929360e-17]
After Translation: [-2.52978802e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0. -0.394 0.694]

Record this pose?(Y) |
```

Dies ist die Grundhaltung des Roboters, die sich aus den folgenden Winkeln ergibt:

$$\begin{array}{lll}
 \text{Basis: } \ddot{\gamma}_1 = 0 & \text{Schulter: } \ddot{\gamma}_2 = \frac{\ddot{\gamma}}{2} & \text{Ellenbogen: } \ddot{\gamma}_3 = 0 \\
 \text{Handgelenk 1: } \ddot{\gamma}_4 = \frac{\ddot{\gamma}}{2} & \text{Handgelenk 2: } \ddot{\gamma}_5 = 0 & \text{Handgelenk 3: } \ddot{\gamma}_6 = 0
 \end{array}$$

Die obigen dualen Quaternionen werden in Bezug auf diese Winkel zusammen mit ihrer Rotationsachse und ihren entsprechenden Übersetzungen abgeleitet. Dies kann in Abschnitt 4.2.1 überprüft werden. Diese werden dann multipliziert und das Produkt wird als duale Quaternion festgelegt, die die endgültige Transformation darstellt. Der Werkzeugmittelpunkt des Endeffektors wird dann von den Koordinaten (0, -0,2, 0) in (0, -0,394, 0,694) in Bezug auf den Basisrahmen transformiert.

Kapitel 4. Anwendung von Quaternionen auf den Universalroboter UR3

Die Funktion „Pose aufzeichnen“ ist so eingestellt, dass sie die Achsenwinkeldarstellung verwendet. Zu Demonstrationszwecken kann die Ableitung der Vorwärtskinematik aus einer Transformationsmatrix auch durch Aufrufen der Funktion „Get Forward Kinematics“ erfolgen, indem die Gelenkwinkel als Parameter und TM als Transformationsmodus festgelegt werden. Die Implementierung wird unten demonstriert:

```
>>> get_forward_kinematics("TM", rob.getj())

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Transformation Matrix...
Computing Transformation Matrix by Denavit-Hartenberg Parameters...

Transformation Matrix for Frame 0 to 1
[[ 1.    -0.    0.    0.   ]
 [ 0.    1.    -0.    -0.   ]
 [ 0.    0.    1.    0.1519]
 [ 0.    0.    0.    1.   ]]
Dual Quaternion: ([ 1.000 +0.000i +0.000j +0.000k ], < 0.000 +0.000i -0.000j +0.152k >)

Transformation Matrix for Frame 1 to 2
[[ 6.12323400e-17  1.00000000e+00  0.00000000e+00  0.00000000e+00]
 [-6.12323400e-17  3.74939946e-33 -1.00000000e+00 -0.00000000e+00]
 [-1.00000000e+00  6.12323400e-17  6.12323400e-17  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion: ([ 0.500 +0.500i +0.500j -0.500k ], < 0.000 +0.000i -0.000j +0.000k >)

Transformation Matrix for Frame 2 to 3
[[ 1.    -0.    0.    -0.24365]
 [ 0.    1.    -0.    -0.   ]
 [ 0.    0.    1.    0.   ]
 [ 0.    0.    0.    1.   ]]
Dual Quaternion: ([ 1.000 +0.000i +0.000j +0.000k ], < 0.000 -0.244i -0.000j +0.000k >)

Transformation Matrix for Frame 3 to 4
[[ 6.123234e-17  1.000000e+00  0.000000e+00 -2.132500e-01]
 [-1.000000e+00  6.123234e-17 -0.000000e+00 -0.000000e+00]
 [ 0.000000e+00  0.000000e+00  1.000000e+00  1.123500e-01]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Dual Quaternion: ([ 0.707 +0.000i +0.000j -0.707k ], < 0.000 -0.213i -0.000j +0.112k >)

Transformation Matrix for Frame 4 to 5
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17 -1.00000000e+00 -8.53500000e-02]
 [ 0.00000000e+00  1.00000000e+00  6.12323400e-17  5.22618022e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion: ([ 0.707 +0.707i +0.000j +0.000k ], < 0.000 +0.000i -0.085j +0.000k >)

Transformation Matrix for Frame 5 to 6
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17  1.00000000e+00  8.19000000e-02]
 [-0.00000000e+00 -1.00000000e+00  6.12323400e-17  5.01492864e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion: ([ 0.707 -0.707i +0.000j +0.000k ], < 0.000 +0.000i +0.082j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ 0.000 +0.000i +0.707j -0.707k ], < 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [0.  0.  0.2]

After Rotation: [ 0.  -0.2  0. ]
After Translation: [ 0.  -0.39425  0.69415]

Transformed TCP Coordinates:
[ 0.  -0.394  0.694]
```

Die duale Quaternion, die die endgültige Transformation in beiden Ansätzen darstellt, unterscheidet sich deutlich, aber die transformierten Koordinaten sind dieselben. Beachten Sie, dass wir auch andere anfängliche TCP-Koordinaten verwenden.

Die Record Pose-Funktion speichert die Posen in einer Datei und die Run Poses-Funktion ruft die Koeffizienten der in der Datei gespeicherten Dual-Quaternionen ab und generiert die Winkel, die diesen Transformationen entsprechen. Dies wird im Folgenden demonstriert:

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> run_poses()

Desired Pose: -0.000 +0.000i +1.000j -0.000k 0.000 -0.000i -0.194j +0.694k

Set of Possible Joint Angles:
[[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4.7123889644920745,
0.0, 0], [6.283185307179586, -1.5707963128851894, -2.9802322387695312e-08, 4.712388996
277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08, -1.570796342687512
, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.5707963109022816, 0.
0, 0]]

Current Joint Angles: [6.283185307179586, -1.570796340704604, 2.9802322387695312e-08,
4.7123889644920745, 0.0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion: ([ -1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.152k >)
Dual Quaternion: ([ 0.707 -0.000i +0.707j -0.000k ], < 0.000 +0.000i +0.000j +0.244k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 -0.213i -0.112j -0.000k >)
Dual Quaternion: ([ -0.707 +0.000i -0.707j +0.000k ], < 0.000 -0.085i +0.000j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j -0.000k ], < 0.000 -0.000i -0.082j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [-4.8985872e-17 -2.0000000e-01 2.4492936e-17]
After Translation: [-3.60044906e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0. -0.394 0.694]

Comparing Dual Quaternions:
Desired Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 -0.000i -0.194j +0.694k >)
Current Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)
PASSED criteria!
```

Es gab 4 mögliche Winkelkombinationen. Dies wird weiter verifiziert, indem überprüft wird, ob die entsprechende Transformation (dargestellt durch die duale Quaternion) ungefähr unserer gewünschten Transformation entspricht, wie auch oben gezeigt.

Die endgültigen Ergebnisse am Ende sind gegeben durch:

```
Final Joint Angles: [[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4.
7123889644920745, 0.0, 0], [6.283185307179586, -1.5707963128851894, -2.98023223876953
12e-08, 4.712388996277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08,
-1.570796342687512, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.57
07963109022816, 0.0, 0]]

There are 4 possible pose(s)!
>>>
```

Daher waren alle Kombinationen gültig und alle transformieren den Werkzeugmittelpunkt unseres Roboters in die gewünschte Pose (gleiche transformierte TCP-Koordinaten), was die Lösung der inversen Kinematik darstellt.

Kapitel 5

Schlussfolgerungen und Empfehlungen

5.1 Fazit

Die Arbeit konnte den Einsatz der Quaternionenalgebra erfolgreich demonstrieren in der Darstellung von Kinematiken für Roboteranwendungen insbesondere mit der 6-DOF-Universalroboter UR3. Sowohl seine direkte als auch seine inverse Kinematik wurden nach den Prinzipien der Quaternionen abgeleitet. Diese sind enthalten in Abschnitt 4.2 Vorwärtskinematik durch duale Quaternionen und Abschnitt 4.3 Invers Kinematik durch duale Quaternionen. Eine praktische Programmieranwendung war Entwickelt wurde auch die in Abschnitt 4.4.2 gezeigte Python-Programmieranwendung bestätigen die Genauigkeit der abgeleiteten Kinematik.

Abgesehen davon können wir aufgrund unserer Ergebnisanalyse auch eine Schlussfolgerung ziehen die folgenden Aussagen:

1. Duale Quaternionen stellen eine kompaktere Möglichkeit zur Darstellung dreidimensionaler Transformationen dar. Bietet somit auch einen erheblichen Vorteil hinsichtlich der Speichereffizienz.
 - (a) Dies ist darauf zurückzuführen, dass Quaternionen im Vergleich zur am häufigsten verwendeten Rotation nur vier Elemente zur Darstellung einer Rotation benötigen Matrizen, die 9 Elemente umfassen.
 - (b) Für die Transformation in drei Dimensionen, die Rotation und Translation beinhaltet. Eine homogene Transformationsmatrix besteht aus 16 Elementen, von denen vier Elemente trivial und dual sind Quaternion benötigt nur 7 Elemente, um dieselbe Transformation darzustellen.

2. Die Durchführung mehrerer Transformationen durch duale Quaternionen bietet auch Vorteile hinsichtlich der Rechenstabilität und der numerischen Genauigkeit, insbesondere im Umgang mit der Kinematik eines Roboters.

- (a) Das Kombinieren von Transformationen unter Verwendung homogener Transformationsmatrizen erfordert die Multiplikation von 4×4 -Matrizen, was mehrere arithmetische und trigonometrische Operationen erfordert, die die numerische Präzision der Lösung erheblich verringern können.
- (b) Die Durchführung einer Quaternion-Multiplikation ist vergleichbar mit einer Polynommultiplikation, berücksichtigt aber auch die Multiplikationsgesetze für die Quaternion-Einheiten. Dies macht es effizienter und auch numerisch präziser, da weniger arithmetische und trigonometrische Operationen erforderlich sind.

3. Quaternionen bieten auch einen großen Vorteil hinsichtlich der Normalisierung und einfachen Ableitung.

- (a) Quaternionen sind einfacher zu normalisieren als Matrizen. Eine Rotationsmatrix muss orthogonal sein und eine Determinante von 1 haben. Bei Quaternionen muss es sich um eine Einheitsquaternion handeln, was bedeutet, dass sie eine Einheitsnorm haben muss. Dies kann einfach dadurch erreicht werden, dass alle vier Elemente der Quaternionen durch ihre Größe dividiert werden.
- (b) Duale Quaternionen lassen sich leicht mithilfe der Achsenwinkeldarstellung ableiten, wobei wir lediglich die Beziehung der aktuellen Drehachse zum Basiskoordinatensystem bestimmen müssen.

4. Duale Quaternionen sind nicht anfällig für eine kardanische Sperre.

- (a) Obwohl in dieser Studie nicht diskutiert, stellt die kardanische Sperre ein ernstes Problem beim Umgang mit Rotationen im dreidimensionalen Raum dar. Stellen Sie sich drei Kardanringe vor, die Drehungen um jeweils drei Achsen darstellen. stellt somit die drei Freiheitsgrade dar. Diese werden als Rollen, Nicken und Gieren definiert. Dies geschieht, wenn die äußeren und inneren Kardanringe aufeinander ausgerichtet sind und ein Freiheitsgrad verloren geht. Dies ist auf die Tatsache zurückzuführen, dass die Rotation, die von jedem dieser beiden Kardanringe ausgeführt werden kann, nun als gleich angesehen wird. Euler Angles ist dafür bekannt, anfällig für dieses seltene Ereignis zu sein.
- (b) Quaternionen sind ebenso wie Rotationsmatrizen bekanntermaßen nicht anfällig für eine kardanische Sperre. Dies liegt daran, dass sie die Orientierung als einen Wert darstellen und nicht als drei separate und verwandte Werte.

5. Wie einfach die Darstellung der Roboterkinematik durch duale Quaternionen ist, hängt von der Art der abgeleiteten Kinematik und der kinematischen Struktur des Roboters ab.

(a) Für die Vorwärtskinematik empfiehlt es sich, die Rotationen statt ausgehend von einer Rotationsmatrix durch einen Achsen-Winkel-Ansatz abzuleiten und anschließend in Einheitsquaternionen umzuwandeln. Auf diese Weise müssen im Vergleich zur herkömmlichen homogenen Transformation deutlich weniger Operationen durchgeführt werden.

(b) Für die inverse Kinematik bleibt die Schwierigkeit, Formeln für die Winkel abzuleiten, in dualen Quaternionen weiterhin bestehen. Dies hängt im Allgemeinen von der kinematischen Struktur des Roboters ab, wie in unserem Fall. Es bezieht sich auf das Verhältnis der Transformation, die von jedem Gelenk durchgeführt werden kann, zur endgültigen Transformation. In unserem Fall mussten wir die Transformationen in drei Teile aufteilen. In einigen Fällen kann es jedoch vorkommen, dass bestimmte Formeln bereits nur aus den in der endgültigen Transformation generierten Gleichungen

abgeleitet werden können. (c) Um das Konzept der Quaternionen weiterhin in der inversen Kinematik anwenden zu können, mussten wir auch mehrere Gleichungen analysieren und berechnen, die die von den einzelnen Gelenken durchgeführten Transformationen darstellen können. Dies macht es im Vergleich zur normalen Konvention, die normalerweise mit dem geometrischen Lösungsansatz durchgeführt

Wir hoffen, dass diese Vorteile als ausreichend erachtet werden, um duale Quaternionen nicht nur in der Robotik und Computergrafik, sondern auch in anderen Bereichen, in denen es um die Arbeit im dreidimensionalen Raum geht, umfassender einzusetzen.

5.2 Empfehlung

Für zukünftige Forschung wird empfohlen, herauszufinden, ob es eine bessere und effizientere Möglichkeit gibt, die Roboterkinematik, insbesondere für die inverse Kinematik, darzustellen. Für den Universal Robot UR3 kann die inverse Kinematiklösung weiter gefiltert werden, indem nur ein bestimmter Bewegungsbereich berücksichtigt wird. Es können auch bestimmte Bedingungen angewendet werden, um die optimale Lösung zu ermitteln, die die gewünschte Pose ergibt. Da sich diese Forschung nur auf die Anwendung von Quaternionen in der Roboterkinematik konzentrierte, kann sie weiter auf ihre Anwendung in der Robotersteuerung im Allgemeinen ausgeweitet werden. Dies kann einen zusätzlichen Schritt der Durchführung einer Quaternion-Interpolation beinhalten. Dies trägt dazu bei, den optimalen Weg für die Flugbahn des Roboters zu finden. Dies wird für die zukünftige Forschung von Vorteil sein, da es sich auch um eine der nützlichen und bekannten Anwendungen von Quater

Anhänge

Anhang A

Universalroboter UR3

Technische Spezifikationen

UR3

Leistung	
Wiederholbarkeit	±0,1 mm / ±0,0039 Zoll (4 Mil)
Umgebungstemperaturbereich	0-50° *
Energieverbrauch	Min. 90 W, typisch 125 W, max. 250 W
Kooperationsbetrieb	15 erweiterte einstellbare Sicherheitsfunktionen. Vom TÜV NORD geprüfte Sicherheitsfunktion Geprüft nach: EN ISO 13849:2008 PL d

Spezifikation	
Nutzlast	3 kg / 6,6 Pfund
Erreichen	500 mm / 19,7 Zoll
Freiheitsgrade	6 Drehgelenke
Programmierung	Grafische Polyscope-Benutzeroberfläche auf 12-Zoll-Touchscreen mit Halterung

Bewegung		
Roboterarm mit Achsenbewegung	Arbeitsbereich	Maximale Geschwindigkeit
Base	± 360°	± 180°/Sek.
Schulter	± 360°	± 180°/Sek.
Ellbogen	± 360°	± 180°/Sek.
Handgelenk 1	± 360°	± 360°/Sek.
Handgelenk 2	± 360°	± 360°/Sek.
Handgelenk 3	Unendlich	±360°/Sek.
Typisches Werkzeug		1 m/Sek. / 39,4 Zoll/Sek.

Merkmale		
IP-Klassifizierung	IP64	
Reinraum der ISO-Klasse	5	
Lärm	70 dB	
Robotermontage	Beliebig	
I/O-Ports	Digital in	2
	Digitaler Ausgang	2
	Analoger Eingang	2
	Analoger Ausgang	0
I/O-Stromversorgung im Werkzeug	12 V/24 V 600 mA im Werkzeug	

Körperlich	
Fußabdruck	Ø 128mm
Materialien	Aluminium, PP-Kunststoffe
Werkzeuganschlusstyp	M8
Kabellänge Roboterarm	6 m / 236 Zoll
Gewicht mit Kabel	11 kg

* Der Roboter kann in einem Temperaturbereich von 0-50°C arbeiten. Bei hoher kontinuierlicher Fügegeschwindigkeit wird die Umgebungstemperatur reduziert.

KONTROLLKASTEN

Merkmale		
IP-Klassifizierung	IP20	
Reinraum der ISO-Klasse	6	
Lärm	<65dB(A)	
I/O-Ports	Digital in	16
	Digitaler Ausgang	16
	Analoger Eingang	2
	Analoger Ausgang	2
I/O-Stromversorgung	24V 2A	
Kommunikation	TCP/IP 100 Mbit, Modbus TCP, Profinet,EthernetIP	
Energiequelle	100–240 VAC, 50–60 Hz	
Umgebungstemperaturbereich	0-50°	
Körperlich		
Größe des Steuerkastens (BxHxT)	475mm x 423mm x 268mm / 18,7 x 16,7 x 10,6 Zoll	
Gewicht	15 kg / 33,1 Pfund	
Materialien	Stahl	

TEACH-ANHÄNGER

Merkmale	
IP-Klassifizierung	IP20
Körperlich	
Materialien	Aluminium, PP
Gewicht	1,5 kg / 3,3 Pfund
Kabellänge	4,5 m / 177 Zoll



Anhang B

Transformationsprodukte für Basis bis Handgelenk 3

Dieser Teil zeigt die Transformationsprodukte für die UR3-Roboterkinematik Struktur von Rahmen 1 (Basis) bis Rahmen 6 (Handgelenk 3).

Wir definieren das duale Quaternionenprodukt M_i als:

$$M_i = Q_i \cdot M_{i+1} \text{ mit } 1 \leq i \leq 6$$

$$M_6 = Q_6$$

$$M_5 = Q_5 \cdot M_6 = Q_5 \cdot Q_6$$

$$M_4 = Q_4 \cdot M_5 = Q_4 \cdot Q_5 \cdot Q_6$$

$$M_3 = Q_3 \cdot M_4 = Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6$$

$$M_2 = Q_2 \cdot M_3 = Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6$$

$$M_1 = Q_1 \cdot M_2 = Q_1 \cdot Q_2 \cdot Q_3 \cdot Q_4 \cdot Q_5 \cdot Q_6$$

Wir definieren das duale Quaternionenprodukt N_i wie folgt:

$$N_{i+1} = Q_i^{-1} \cdot N_i \text{ mit } 1 \leq i \leq 6$$

$$N_1 = QFK = ([w, a, b, c], [x, y, z])$$

$$N_2 = Q_1^{-1} \cdot N_1 = Q_1^{-1} \cdot QFK$$

$$N_3 = Q_2^{-1} \cdot N_2 = Q_2^{-1} \cdot Q_1^{-1} \cdot QFK$$

$$N_4 = Q_3^{-1} \cdot N_3 = Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot QFK$$

$$N_5 = Q_4^{-1} \cdot N_4 = Q_4^{-1} \cdot Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot QFK$$

$$N_6 = Q_5^{-1} \cdot N_5 = Q_5^{-1} \cdot Q_4^{-1} \cdot Q_3^{-1} \cdot Q_2^{-1} \cdot Q_1^{-1} \cdot QFK$$

Transformationsprodukte von der Basis zum Handgelenk 3 (M6 = N6)

$$\mathbf{M6} = ([\mathbf{M61} + \mathbf{M62i} + \mathbf{M63j} + \mathbf{M64k}], < \mathbf{M65i} + \mathbf{M66j} + \mathbf{M67k} >)$$

$$\mathbf{N6} = ([\mathbf{N61} + \mathbf{N62i} + \mathbf{N63j} + \mathbf{N64k}], < \mathbf{N65i} + \mathbf{N66j} + \mathbf{N67k} >)$$

M61 = N61

$$\begin{array}{lcl} \cos \frac{\ddot{y}_6}{2} = w \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} & (B.1) \\ & \text{+eine Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 + \ddot{y}_5}{2} \\ & \text{\ddot{y}b Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 + \ddot{y}_5}{2} \\ & \text{+c cos} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \end{array}$$

M62 = N62

$$\begin{array}{lcl} 0 = \ddot{y}w \sin \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 + \ddot{y}_5}{2} & (B.2) \\ & \text{+a cos} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \\ & \text{+b cos} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \\ & \text{+c Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 + \ddot{y}_5}{2} \end{array}$$

M63 = N63

$$\begin{array}{lcl} \ddot{y} \text{ Sünde} \frac{\ddot{y}_6}{2} = w \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 + \ddot{y}_5}{2} & (B.3) \\ & \text{\ddot{y}a cos} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \\ & \text{+b cos} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \cos \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \\ & \text{+c Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \text{Sünde} \frac{\ddot{y}_1 + \ddot{y}_5}{2} \end{array}$$

$$\underline{M64 = N64}$$

$$\begin{aligned}
 0 = \ddot{y}_w \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \quad & \text{Sünde} \quad \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} & (B.4) \\
 \ddot{y}_{\text{eine Sünde}} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \quad & \cos \quad \frac{\ddot{y}_1 + \ddot{y}_5}{2} \\
 \ddot{y}_b \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \quad & \text{Sünde} \quad \frac{\ddot{y}_1 + \ddot{y}_5}{2} \\
 +c \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \quad & \cos \quad \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2}
 \end{aligned}$$

$$\underline{M65 = N65}$$

$$\begin{aligned}
 0 = \ddot{y} d_1 \cos \ddot{y}_5 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \ddot{y} d_4 \sin \ddot{y}_5 \ddot{y} a_2 \cos (\ddot{y}_3 + \ddot{y}_4) \cos \ddot{y}_5 & (B.5) \\
 \ddot{y} a_3 \cos \ddot{y}_4 \cos \ddot{y}_5 + x [\cos \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \cos \ddot{y}_5 + \sin \ddot{y}_1 \sin \ddot{y}_5] \\
 + y [\cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \cos \ddot{y}_5 \sin \ddot{y}_1 \ddot{y} \cos \ddot{y}_1 \sin \ddot{y}_5] \\
 + z \cos \ddot{y}_5 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4)
 \end{aligned}$$

$$\underline{M66 = N66}$$

$$\begin{aligned}
 0 = \ddot{y} d_1 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 + d_4 \cos \ddot{y}_5 \ddot{y} a_2 \cos (\ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 & (B.6) \\
 \ddot{y} a_3 \cos \ddot{y}_4 \sin \ddot{y}_5 + x [\cos \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 \ddot{y} \cos \ddot{y}_5 \sin \ddot{y}_1] \\
 + y [\cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_1 \sin \ddot{y}_5 + \cos \ddot{y}_1 \cos \ddot{y}_5] \\
 + z \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 + d_6
 \end{aligned}$$

$$\underline{M67 = N67}$$

$$\begin{aligned}
 0 = \ddot{y} d_1 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \ddot{y} x \cos \ddot{y}_1 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \ddot{y} y & (B.7) \\
 \sin \ddot{y}_1 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) + z \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \\
 + a_2 \sin (\ddot{y}_3 + \ddot{y}_4) + a_3 \sin \ddot{y}_4 + d_5
 \end{aligned}$$

Anhang B. Transformationsprodukte für **Base toWrist 3**

Transformationsprodukte von der Basis zum Handgelenk 3 (M5 = N5)

$$\mathbf{M5} = ([\mathbf{M51} + \mathbf{M52i} + \mathbf{M53j} + \mathbf{M54k}], < \mathbf{M55i} + \mathbf{M56j} + \mathbf{M57k} >)$$

$$\mathbf{N5} = ([\mathbf{N51} + \mathbf{N52i} + \mathbf{N53j} + \mathbf{N54k}], < \mathbf{N55i} + \mathbf{N56j} + \mathbf{N57k} >)$$

$$\mathbf{M51} = \mathbf{N51}$$

$$\begin{aligned} \cos \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ \text{eine Sünde} \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &\ddot{y}b \cos \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ c \text{ Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \end{aligned} \quad (\text{B.8})$$

$$\mathbf{M52} = \mathbf{N52}$$

$$\begin{aligned} \ddot{y} \text{ Sünde} \frac{\ddot{y}_5}{2} \text{ Sünde} \frac{\ddot{y}_6}{2} &= \ddot{y}w \sin \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ a \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ b \text{ Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ c \cos \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \end{aligned} \quad (\text{B.9})$$

$$\mathbf{M53} = \mathbf{N53}$$

$$\begin{aligned} \ddot{y} \cos \frac{\ddot{y}_5}{2} \text{ Sünde} \frac{\ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &\ddot{y} \text{ eine Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ b \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \\ &+ c \text{ Sünde} \frac{\ddot{y}_1}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} \end{aligned} \quad (\text{B.10})$$

$$\underline{M54 = N54}$$

$$\begin{aligned} \ddot{y} \sin \ddot{\gamma} \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_6}{2} &= \ddot{y} w \sin \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & (B.11) \\ \ddot{y} a \cos \frac{\ddot{y}_1}{2} \sin \ddot{\gamma} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \\ \ddot{y} b \sin \ddot{\gamma} \frac{\ddot{y}_1}{2} \sin \ddot{\gamma} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \\ + c \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} & \end{aligned}$$

$$\underline{M55 = N55}$$

$$\begin{aligned} \ddot{y} d_6 \sin \ddot{\gamma}_5 &= \ddot{y} d_1 \sin (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \ddot{y} a_2 \cos (\ddot{\gamma}_3 + \ddot{\gamma}_4) \ddot{y} a_3 \cos \ddot{\gamma}_4 & (B.12) \\ &+ x \cos \ddot{\gamma}_1 \cos (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) + y \sin \ddot{\gamma}_1 \cos (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \\ &+ z \sin (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \end{aligned}$$

$$\underline{M56 = N56}$$

$$\ddot{y} d_6 \cos \ddot{\gamma}_5 = y \cos \ddot{\gamma}_1 \ddot{y} x \sin \ddot{\gamma}_1 + d_4 \quad (B.13)$$

$$\underline{M57 = N57}$$

$$\begin{aligned} 0 &= \ddot{y} d_1 \cos (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) + d_5 + a_2 \sin (\ddot{\gamma}_3 + \ddot{\gamma}_4) + a_3 \sin \ddot{\gamma}_4 \ddot{y} x & (B.14) \\ &\cos \ddot{\gamma}_1 \sin (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \ddot{y} y \sin \ddot{\gamma}_1 \sin (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \\ &+ z \cos (\ddot{\gamma}_2 + \ddot{\gamma}_3 + \ddot{\gamma}_4) \end{aligned}$$

Anhang B. Transformationsprodukte für **Base toWrist 3**

Transformationsprodukte von der Basis zum Handgelenk 3 (M4 = N4)

$$\mathbf{M4} = ([\mathbf{M41} + \mathbf{M42i} + \mathbf{M43j} + \mathbf{M44k}], < \mathbf{M45i} + \mathbf{M46j} + \mathbf{M47k} >)$$

$$\mathbf{N4} = ([\mathbf{N41} + \mathbf{N42i} + \mathbf{N43j} + \mathbf{N44k}], < \mathbf{N45i} + \mathbf{N46j} + \mathbf{N47k} >)$$

$$\mathbf{M41} = \mathbf{N41}$$

$$\begin{aligned} \cos \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_4 + \ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ \text{eine Sünde} \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &\ddot{y}b \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ c \text{Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \end{aligned} \quad (\text{B.15})$$

$$\mathbf{M42} = \mathbf{N42}$$

$$\begin{aligned} \text{Sünde} \frac{\ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_4 \ddot{y} \ddot{y}_6}{2} &= \ddot{y}w \sin \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ a \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ b \text{Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ c \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \end{aligned} \quad (\text{B.16})$$

$$\mathbf{M43} = \mathbf{N43}$$

$$\begin{aligned} \ddot{y} \cos \frac{\ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_4 + \ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &\ddot{y} \text{eine Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ b \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ &+ c \text{Sünde} \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3}{2} \end{aligned} \quad (\text{B.17})$$

$$\underline{M44 = N44}$$

$$\begin{aligned} \ddot{y} \sin \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_4 \ddot{y} \ddot{y}_6}{2} &= \ddot{y} \sin \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} & (B.18) \\ \ddot{y}_a \cos \frac{\ddot{y}_1}{2} \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ \ddot{y}_b \sin \frac{\ddot{y}_1}{2} \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} \\ + c \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \end{aligned}$$

$$\underline{M45 = N45}$$

$$\begin{aligned} \ddot{y} d_6 \cos \ddot{y}_4 \sin \ddot{y}_5 + d_5 \sin \ddot{y}_4 &= \ddot{y} d_1 \sin (\ddot{y}_2 + \ddot{y}_3) \ddot{y} a_2 \cos \ddot{y}_3 \ddot{y} a_3 + & (B.19) \\ x \cos \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3) \\ + y \sin \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3) \\ + z \sin (\ddot{y}_2 + \ddot{y}_3) \end{aligned}$$

$$\underline{M46 = N46}$$

$$\ddot{y} d_6 \cos \ddot{y}_5 = y \cos \ddot{y}_1 \ddot{y} x \sin \ddot{y}_1 + d_4 \quad (B.20)$$

$$\underline{M47 = N47}$$

$$\begin{aligned} \ddot{y} d_6 \sin \ddot{y}_4 \sin \ddot{y}_5 \ddot{y} d_5 \cos \ddot{y}_4 &= \ddot{y} d_1 \cos (\ddot{y}_2 + \ddot{y}_3) + a_2 \sin \ddot{y}_3 \ddot{y} & (B.21) \\ x \cos \ddot{y}_1 \sin (\ddot{y}_2 + \ddot{y}_3) \\ \ddot{y} y \sin \ddot{y}_1 \sin (\ddot{y}_2 + \ddot{y}_3) \\ + z \cos (\ddot{y}_2 + \ddot{y}_3) \end{aligned}$$

Anhang B. Transformationsprodukte für Base toWrist 3

Transformationsprodukte von der Basis zum Handgelenk 3 (M3 = N3)

$$\mathbf{M3} = ([\mathbf{M31} + \mathbf{M32i} + \mathbf{M33j} + \mathbf{M34k}], < \mathbf{M35i} + \mathbf{M36j} + \mathbf{M37k} >)$$

$$\mathbf{N3} = ([\mathbf{N31} + \mathbf{N32i} + \mathbf{N33j} + \mathbf{N34k}], < \mathbf{N35i} + \mathbf{N36j} + \mathbf{N37k} >)$$

$$\mathbf{M31} = \mathbf{N31}$$

$$\begin{aligned} \cos \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ &+ \text{eine Sünde} \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \\ &\ddot{y}b \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \\ &+c \text{Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \end{aligned} \quad (\text{B.22})$$

$$\mathbf{M32} = \mathbf{N32}$$

$$\begin{aligned} \text{Sünde} \frac{\ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_3 + \ddot{y}_4 \ddot{y} \ddot{y}_6}{2} &= \ddot{y}w \sin \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \\ &+a \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ &+b \text{Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ &+c \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \end{aligned} \quad (\text{B.23})$$

$$\mathbf{M33} = \mathbf{N33}$$

$$\begin{aligned} \ddot{y} \cos \frac{\ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} &= w \cos \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \\ &\ddot{y} \text{eine Sünde} \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ &+b \cos \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ &+c \text{Sünde} \frac{\ddot{y}_1}{2} \text{Sünde} \frac{\ddot{y}_2}{2} \end{aligned} \quad (\text{B.24})$$

$$\underline{M34 = N34}$$

$$\begin{aligned} \ddot{y} \sin \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_3 + \ddot{y}_4}{2} \ddot{y} \ddot{y}_6 &= \ddot{y} \sin \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} \\ \ddot{y}_a \cos \frac{\ddot{y}_1}{2} \sin \frac{\ddot{y}_2}{2} & \\ \ddot{y}_b \sin \frac{\ddot{y}_1}{2} \cos \frac{\ddot{y}_2}{2} & \\ + c \cos \frac{\ddot{y}_1}{2} \sin \frac{\ddot{y}_2}{2} & \end{aligned} \quad (B.25)$$

$$\underline{M35 = N35}$$

$$\begin{aligned} \ddot{y} d_6 \cos(\ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 + d_5 \sin(\ddot{y}_3 + \ddot{y}_4) + a_3 \cos \ddot{y}_3 &= \ddot{y} \\ d_1 \sin \ddot{y}_2 \ddot{y} a_2 + x \cos \ddot{y}_1 \cos \ddot{y}_2 & \\ + y \sin \ddot{y}_1 \cos \ddot{y}_2 + z \sin \ddot{y}_2 & \end{aligned} \quad (B.26)$$

$$\underline{M36 = N36}$$

$$\ddot{y} d_6 \cos \ddot{y}_5 \ddot{y} d_4 = y \cos \ddot{y}_1 \ddot{y} x \sin \ddot{y}_1 \quad (B.27)$$

$$\underline{M37 = N37}$$

$$\begin{aligned} \ddot{y} d_6 \sin(\ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 \ddot{y} d_5 \cos(\ddot{y}_3 + \ddot{y}_4) + a_3 \sin \ddot{y}_3 &= \\ \ddot{y} d_1 \cos \ddot{y}_2 \ddot{y} x \cos \ddot{y}_1 \sin \ddot{y}_2 & \\ \ddot{y} y \sin \ddot{y}_1 \sin \ddot{y}_2 + z \cos \ddot{y}_2 & \end{aligned} \quad (B.28)$$

Anhang B. Transformationsprodukte für Base toWrist 3

Transformationsprodukte von der Basis zum Handgelenk 3 (M2 = N2)

$$\mathbf{M2} = ([\mathbf{M21} + \mathbf{M22i} + \mathbf{M23j} + \mathbf{M24k}], < \mathbf{M25i} + \mathbf{M26j} + \mathbf{M27k} >)$$

$$\mathbf{N2} = ([\mathbf{N21} + \mathbf{N22i} + \mathbf{N23j} + \mathbf{N24k}], < \mathbf{N25i} + \mathbf{N26j} + \mathbf{N27k} >)$$

$$\mathbf{M21} = \mathbf{N21}$$

$$\cos \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} = w \cos \frac{\ddot{y}_1}{2} + c \text{ Sünde} \frac{\ddot{y}_1}{2} \quad (\text{B.29})$$

$$\mathbf{M22} = \mathbf{N22}$$

$$\text{Sünde} \frac{\ddot{y}_5}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 \ddot{y}_6}{2} = \text{ein} \cos \frac{\ddot{y}_1}{2} + b \text{ Sünde} \frac{\ddot{y}_1}{2} \quad (\text{B.30})$$

$$\mathbf{M23} = \mathbf{N23}$$

$$\ddot{y} \cos \frac{\ddot{y}_5}{2} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} = \ddot{y} \text{eine Sünde} \frac{\ddot{y}_1}{2} + b \cos \frac{\ddot{y}_1}{2} \quad (\text{B.31})$$

$$\mathbf{M24} = \mathbf{N24}$$

$$\ddot{y} \text{ Sünde} \frac{\ddot{y}_5}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 \ddot{y}_6}{2} = \ddot{y} w \sin \frac{\ddot{y}_1}{2} + c \cos \frac{\ddot{y}_1}{2} \quad (\text{B.32})$$

$$\mathbf{M25} = \mathbf{N25}$$

$$\begin{aligned} \ddot{y} d_6 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 + d_5 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \\ + a_3 \cos (\ddot{y}_2 + \ddot{y}_3) + a_2 \cos \ddot{y}_2 = x \cos \ddot{y}_1 + y \sin \ddot{y}_1 \end{aligned} \quad (\text{B.33})$$

$$\mathbf{M26} = \mathbf{N26}$$

$$\ddot{y} d_6 \cos \ddot{y}_5 \ddot{y} d_4 = y \cos \ddot{y}_1 \ddot{y} x \sin \ddot{y}_1 \quad (\text{B.34})$$

$$\mathbf{M27} = \mathbf{N27}$$

$$\begin{aligned} \ddot{y} d_6 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 \ddot{y} d_5 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \\ + a_3 \sin (\ddot{y}_2 + \ddot{y}_3) + a_2 \sin \ddot{y}_2 = \ddot{y} d_1 + z \end{aligned} \quad (\text{B.35})$$

Transformationsprodukte von der Basis zum Handgelenk 3 ($M1 = N1$)

$$M1 = ([M11 + M12i + M13j + M14k], < M15i + M16j + M17k >)$$

$$N1 = ([N11 + N12i + N13j + N14k], < N15i + N16j + N17k >)$$

$$M11 = N11$$

$$\cos \frac{\ddot{y}_1 \ddot{y} \ddot{y}_5}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} = w \quad (B.36)$$

$$M12 = N12$$

$$\text{Sünde} \frac{\ddot{y}_1 + \ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 \ddot{y} \ddot{y}_6}{2} = a \quad (B.37)$$

$$M13 = N13$$

$$\ddot{y} \cos \frac{\ddot{y}_1 + \ddot{y}_5}{2} \text{Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 + \ddot{y}_6}{2} = b \quad (B.38)$$

$$M14 = N14$$

$$\ddot{y} \text{Sünde} \frac{\ddot{y}_5 \ddot{y} \ddot{y}_1}{2} \cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4 \ddot{y} \ddot{y}_6}{2} = c \quad (B.39)$$

$$M15 = N15$$

$$\begin{aligned} \ddot{y} d6 \cos \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 + d6 \cos \ddot{y}_5 \sin \ddot{y}_1 + d4 \sin \ddot{y}_1 \\ + d5 \cos \ddot{y}_1 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) + a3 \cos \ddot{y}_1 \cos (\ddot{y}_2 + \ddot{y}_3) \\ + a2 \cos \ddot{y}_1 \cos \ddot{y}_2 = x \end{aligned} \quad (B.40)$$

$$M16 = N16$$

$$\begin{aligned} \ddot{y} d6 \cos \ddot{y}_1 \cos \ddot{y}_5 \ddot{y} d4 \cos \ddot{y}_1 + d6 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_1 \sin \ddot{y}_5 \\ + d5 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) + a3 \cos (\ddot{y}_2 + \ddot{y}_3) + a2 \cos \ddot{y}_2 = y \end{aligned} \quad (B.41)$$

$$M17 = N17$$

$$\begin{aligned} \ddot{y} d6 \sin (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) \sin \ddot{y}_5 \ddot{y} d5 \cos (\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4) + a3 \sin (\ddot{y}_2 + \ddot{y}_3) \\ + a2 \sin \ddot{y}_2 + d1 = z \end{aligned} \quad (B.42)$$

Anhang C

Transformationsprodukte für Schulter bis Handgelenk 1

Dieser Teil zeigt die Transformationsprodukte für die kinematische Struktur des UR3-Roboters von Rahmen 2 (Schulter) bis Rahmen 4 (Handgelenk 1).

Wir definieren das duale Quaternionenprodukt M_i als:

$$M_i = Q_i \cdot M_{i+1} \text{ mit } 2 \leq i \leq 4$$

$$M_4 = Q_4$$

$$M_3 = Q_3 \cdot M_4 = Q_3 \cdot Q_4$$

$$M_2 = Q_2 \cdot M_3 = Q_2 \cdot Q_3 \cdot Q_4$$

Wir definieren das duale Quaternionenprodukt N_i wie folgt:

$$N_{i+1} = Q_i^{-1} \cdot N_i \text{ mit } 2 \leq i \leq 4$$

$$N_2 = Q_2^{-1} \cdot N_4 = Q_2^{-1} \cdot [w_4, < a_4, b_4, c_4 >, < x_4, y_4, z_4 >]$$

$$N_3 = Q_3^{-1} \cdot N_2 = Q_3^{-1} \cdot Q_2^{-1} \cdot N_4$$

$$N_4 = Q_4^{-1} \cdot N_3 = Q_4^{-1} \cdot Q_3^{-1} \cdot Q_2^{-1} \cdot N_4$$

Anhang C. Transformationsprodukte für **Schulter** zu **Handgelenk 1**

Transformationsprodukte von **Schulter** zu **Handgelenk 1** ($M4 = N4$)

$$M4 = ([M41 + M42i + M43j + M44k], < M45i + M46j + M47k >)$$

$$N4 = ([N41 + N42i + N43j + N44k], < N45i + N46j + N47k >)$$

$$\underline{M41 = N41}$$

$$\cos \frac{\ddot{y}_4}{2} = w_4 \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \ddot{y} b_4 \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} \quad (C.1)$$

$$\underline{M42 = N42}$$

$$0 = a_4 \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} + c_4 \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} \quad (C.2)$$

$$\underline{M43 = N43}$$

$$\ddot{y} \sin \frac{\ddot{y}_4}{2} = w_4 \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} + b_4 \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \quad (C.3)$$

$$\underline{M44 = N44}$$

$$0 = \ddot{y} a_4 \sin \frac{\ddot{y}_2 + \ddot{y}_3}{2} + c_4 \cos \frac{\ddot{y}_2 + \ddot{y}_3}{2} \quad (C.4)$$

$$\underline{M45 = N45}$$

$$0 = \ddot{y} a_2 \cos \ddot{y}_3 \ddot{y} a_3 + x_4 \cos (\ddot{y}_2 + \ddot{y}_3) + z_4 \sin (\ddot{y}_2 + \ddot{y}_3) \quad (C.5)$$

$$\underline{M46 = N46}$$

$$0 = y_4 + d_4 \quad (C.6)$$

$$\underline{M47 = N47}$$

$$0 = a_2 \sin \ddot{y}_3 \ddot{y} x_4 \sin (\ddot{y}_2 + \ddot{y}_3) + z_4 \cos (\ddot{y}_2 + \ddot{y}_3) \quad (C.7)$$

Transformationsprodukte von Schulter zu Handgelenk 1 (M3 = N3)

$$\mathbf{M3} = ([\mathbf{M31} + \mathbf{M32i} + \mathbf{M33j} + \mathbf{M34k}], < \mathbf{M35i} + \mathbf{M36j} + \mathbf{M37k} >)$$

$$\mathbf{N3} = ([\mathbf{N31} + \mathbf{N32i} + \mathbf{N33j} + \mathbf{N34k}], < \mathbf{N35i} + \mathbf{N36j} + \mathbf{N37k} >)$$

$$\mathbf{M31} = \mathbf{N31}$$

$$\cos \frac{\ddot{y}_3 + \ddot{y}_4}{2} = w_4 \cos \frac{\ddot{y}_2}{2} \ddot{y} b_4 \sin \frac{\ddot{y}_2}{2} \quad (\text{C.8})$$

$$\mathbf{M32} = \mathbf{N32}$$

$$0 = a_4 \cos \frac{\ddot{y}_2}{2} + c_4 \sin \frac{\ddot{y}_2}{2} \quad (\text{C.9})$$

$$\mathbf{M33} = \mathbf{N33}$$

$$\ddot{y} \sin \frac{\ddot{y}_3 + \ddot{y}_4}{2} = w_4 \sin \frac{\ddot{y}_2}{2} + b_4 \cos \frac{\ddot{y}_2}{2} \quad (\text{C.10})$$

$$\mathbf{M34} = \mathbf{N34}$$

$$0 = \ddot{y} a_4 \sin \frac{\ddot{y}_2}{2} + c_4 \cos \frac{\ddot{y}_2}{2} \quad (\text{C.11})$$

$$\mathbf{M35} = \mathbf{N35}$$

$$a_3 \cos \ddot{y}_3 = \ddot{y} a_2 + x_4 \cos \ddot{y}_2 + z_4 \sin \ddot{y}_2 \quad (\text{C.12})$$

$$\mathbf{M36} = \mathbf{N36}$$

$$\ddot{y} d_4 = y_4 \quad (\text{C.13})$$

$$\mathbf{M37} = \mathbf{N37}$$

$$a_3 \sin \ddot{y}_3 = \ddot{y} x_4 \sin \ddot{y}_2 + z_4 \cos \ddot{y}_2 \quad (\text{C.14})$$

Anhang C. Transformationsprodukte für **Schulter** zu **Handgelenk 1**

Transformationsprodukte von **Schulter** zu **Handgelenk 1** ($M2 = N2$)

$$\mathbf{M2} = ([\mathbf{M21} + M22i + M23j + \mathbf{M24k}], < M25i + M26j + M27k >)$$

$$\mathbf{N2} = ([\mathbf{N21} + N22i + N23j + \mathbf{N24k}], < N25i + N26j + N27k >)$$

$$\mathbf{M21} = \mathbf{N21}$$

$$\cos \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} = w_4 \quad (\text{C.15})$$

$$\mathbf{M22} = \mathbf{N22}$$

$$0 = a_4 \quad (\text{C.16})$$

$$\mathbf{M23} = \mathbf{N23}$$

$$\ddot{y} \text{ Sünde} \frac{\ddot{y}_2 + \ddot{y}_3 + \ddot{y}_4}{2} = b_4 \quad (\text{C.17})$$

$$\mathbf{M24} = \mathbf{N24}$$

$$0 = c_4 \quad (\text{C.18})$$

$$\mathbf{M25} = \mathbf{N25}$$

$$a_3 \cos (\ddot{y}_2 + \ddot{y}_3) + a_2 \cos \ddot{y}_2 = x_4 \quad (\text{C.19})$$

$$\mathbf{M26} = \mathbf{N26}$$

$$\ddot{y}d_4 = y_4 \quad (\text{C.20})$$

$$\mathbf{M27} = \mathbf{N27}$$

$$a_3 \sin (\ddot{y}_2 + \ddot{y}_3) + a_2 \sin \ddot{y}_2 = z_4 \quad (\text{C.21})$$

Anhang D

Verzeichnis der Tabellen und Abbildungen

Liste der Tabellen

3.1 Quaternion-Multiplikationstabelle.				27
4.1 UR3 Denavit-Hartenberg-Parameter				43
4.2 Denavit-Hartenberg-Parameter für UR3 aus [20]				57

Abbildungsverzeichnis

2.1 Vektordarstellung von $z = a + ib$ mit $\text{Re} z = a$ und $\text{Im} z = b$.	17
2.2 Polarkoordinatendarstellung der komplexen Zahl $z = a + ib$ mit Länge r und Winkel φ .	17
2.3 Komplexe Konjugation.	18
2.4 Addition der komplexen Zahlen $3 + 2i$ und $1 + 3i$.	20
2.5 Multiplikation komplexer Zahlen z_1 und z_2 .	21
2.6 Multiplikation komplexer Zahlen $\bar{y}_1$ und $\bar{y}_i$.	21
2.7 Multiplikation von wz mit $z = i$.	22
3.1 Ein Porträt von Sir William Rowan Hamilton	24
3.2 Gedenktafel zur Erinnerung an die Entdeckung der Quaternionen	24
3.3 Quaternion-Multiplikationsdiagramm.	27
3.4 Drehung des Vektors $(2,1,1)$ aus [22]	35
4.1 Universalroboter UR3	41
4.2 Kinematische Struktur von UR3 in Nullposition ($\ddot{y}_{1,2,3,4,5,6} = 0$)	42
4.3 Koordinatensysteme von UR3 ($\ddot{y}_{1,2,3,4,5,6} = 0$) aus [11].	47
4.4 UR3-Roboterstruktur von Frame 1 bis Frame 5 (aus [3])	48
4.5 Übersetzung von Rahmen 6 (Handgelenk 3) zu Rahmen 5 (Handgelenk 2)	49
4.6 3R-Planar-Manipulator, gebildet durch UR3-Gelenke 2 bis 4 aus [3].	53
4.7 UR3 DH-Parameter Diagramm aus [20]	57

Literaturverzeichnis

- [1] Robotik: Kinematik und mathematische Grundlagen.
<https://www.edx.org/course/robotics-kinematics-mathematical-pennx-robo1x>.
- [2] M. Ahr. Das Erbe von Rossums Universalrobotern.
<https://www.denofgeek.com/us/culture/279175/the-legacy-of-rossums-universal-robots>, 2019.
- [3] R. Andersen. Kinematik eines UR5. Technischer Bericht, Aalborg Univer
Stadt, 2018.
- [4] Y. Aydin und S. Kucuk. Quaternionbasierte inverse Kinematik für industrielle
Roboter manipulatoren mit Euler-Handgelenk. 2006 IEEE International
Konferenz über Mechatronik, ICM, 2006.
- [5] J. Barbic. Quaternionen und Rotationen. Technischer Bericht, Universität
Südkalifornien, 2011.
- [6] M. Ben-Ari. Ein Tutorial zu Eulerwinkeln und Quaternionen. 2018.
- [7] Z. Chen und J. Hung. Anwendung von Quaternion in der Robotersteuerung.
10. alle drei Jahre stattfindender IFAC-Weltkongress, 1987.
- [8] J. Diebel. Darstellung der Haltung: Euler-Winkel, Einheitsquaternionen und
Rotationsvektoren. Technischer Bericht, Stanford University, 2006.
- [9] PR Evans. Rotationen und Rotationsmatrizen. Technischer Bericht, MRC
Labor für Molekularbiologie, 2001.
- [10] WR Hamilton. Elemente von Quaternionen. Cambridge University Press,
1866.
- [11] K. Hawkins. Analytische inverse Kinematik für die Universal Robots UR
5/UR-10 Waffen. Technischer Bericht, 2013.

- [12] AO Hill. Kinematik: Warum Roboter sich so bewegen, wie sie es tun. <https://blog.robotiq.com/kinematics-why-robots-move-like-they-do>, 2015.
- [13] J. Huerta. Einführung in die Quaternionen. Technischer Bericht, Fullerton College, 2010.
- [14] B. Kenwright. Ein Anfängerleitfaden zu Dual-Quaternionen: Was sie sind, wie sie funktionieren und wie man sie für 3D verwendet. Technischer Bericht, Newcastle University, Vereinigtes Königreich.
- [15] S. Kucuk und Z. Bingul. Roboterkinematik: Vorwärts- und Rückwärtskinematik. Technischer Bericht, Kocaeli University, 2006.
- [16] S. LaValle. Planungsalgorithmen – Die homogene Transformationsmatrix. <http://planning.cs.uiuc.edu/node111.html>, 2006.
- [17] A. Lerios. Rotationen und Quaternionen. Technischer Bericht, Stanford University, 1995.
- [18] JE Marsden und MJ Hoffman. Grundlegende komplexe Analyse. WHFreeman & Co Ltd, New York, Vereinigte Staaten, 3. Auflage, 1999.
- [19] Mataric. Die Einführung in die Robotik. Massachusetts Institute of Technology, London, England, 2007.
- [20] U. Robots. Parameter für Berechnungen der Kinematik und Dynamik. <https://www.universal-robots.com/how-tos-and-faqs/faq/ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/>.
- [21] B. Siciliano und O. Khatib. Springer-Handbuch der Robotik. Springer, Berlin, 2. Auflage, 2016.
- [22] M. Sunardi. 3D-Kinematik. Technischer Bericht, 2006.
- [23] Tekkotsu. Kinematik. <http://www.tekkotsu.org/Kinematics.html>, 2010.
- [24] J. Voight. Quaternion-Algebren, Band 0.9.14. 2018.
- [25] A. Watt und M. Watt. Fortgeschrittene Animations- und Rendering-Techniken. ACM Press, New York, Vereinigte Staaten, 1992.