



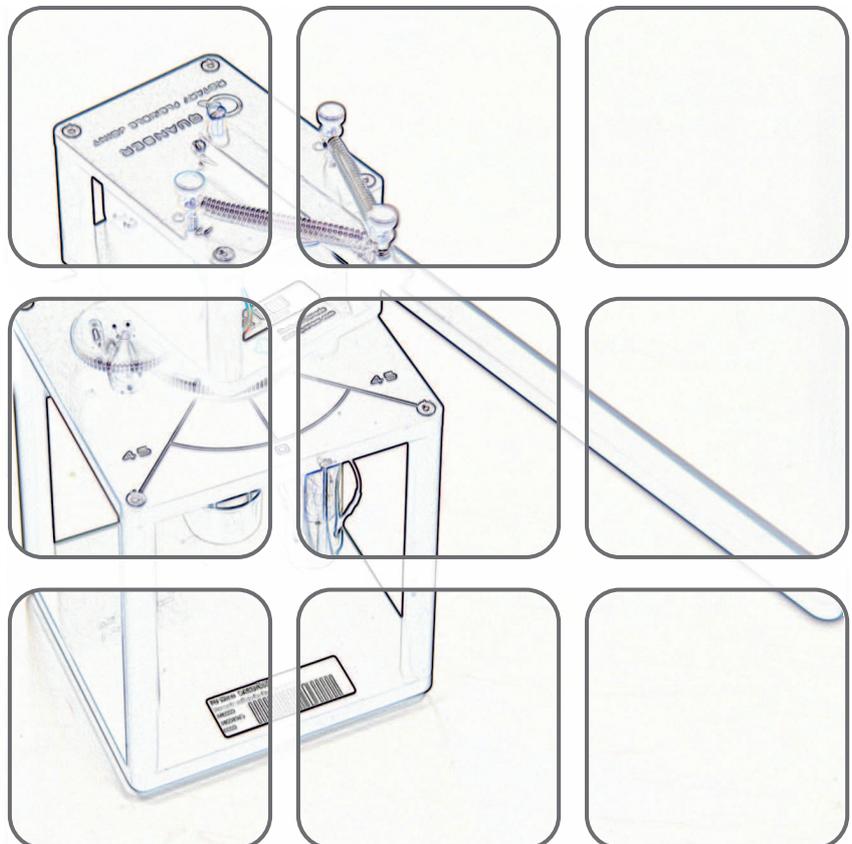
TECHNISCHE
UNIVERSITÄT
WIEN



Vorlesung
WS 2024/2025

Univ.Prof. Dipl.-Ing. Dr.-Ing. Christian Ott
Univ.Ass. Dipl.-Ing. Dr.techn. Christian Hartl-Nesic

GRUNDLAGEN DER ROBOTIK



Grundlagen der Robotik

Vorlesung
WS 2024/2025

Univ.Prof. Dipl.-Ing. Dr.-Ing. Christian Ott
Univ.Ass. Dipl.-Ing. Dr.techn. Christian Hartl-Nesic

TU Wien
Institut für Automatisierungs- und Regelungstechnik
Gruppe für komplexe dynamische Systeme

Gußhausstraße 27–29
1040 Wien
Telefon: +43 1 58801 – 37615
Internet: <https://www.acin.tuwien.ac.at>

Inhaltsverzeichnis

1	Einführung	1
1.1	Grundbegriffe	2
1.2	Roboterarten	4
1.3	Konfigurationsraum	10
1.3.1	Freiheitsgrade eines Starrkörpers und eines Robotersystems	10
1.3.2	Konfiguration	11
1.3.3	Gelenke	11
1.3.4	Generalisierte Koordinaten	12
1.3.5	Kinematische Ketten	13
1.3.6	Formel von Grübler	13
1.3.7	Topologien	15
1.4	Aufgabenraum	17
1.4.1	Arbeitsraum	17
1.4.2	Aufgabenraum	18
1.4.3	Absolutgenauigkeit und Wiederholgenauigkeit	19
1.5	Literatur	21
2	Roboterkomponenten	23
2.1	Sensoren	23
2.1.1	Absolutencoder	24
2.1.2	Inkrementalencoder	25
2.1.3	Drehmomentsensoren	26
2.1.4	Kraft/Momentensensoren (FTS)	26
2.1.5	Inertiale Messeinheiten (IMU)	28
2.1.6	Abstandssensoren	29
2.1.7	Kameras	30
2.2	Antriebssystem	31
2.2.1	Überblick	31
2.2.2	Getriebe	32
2.3	Antriebsarten	37
2.3.1	Hydraulische Antriebe	38
2.3.2	Pneumatische Antriebe	39
2.3.3	Elektrische Antriebe	40
2.3.4	Drehmomentgeregelte Antriebe	42
2.3.5	Series Elastic Actuators (SEA)	44
2.3.6	Steuerungsarchitekturen	45
2.4	Literatur	46

3	Starrkörperbewegungen	47
3.1	Translation	48
3.2	Rotation	49
3.2.1	Rotation um eine Achse	50
3.2.2	Rotationsmatrix	51
3.2.3	Elementare Rotationsmatrizen	53
3.3	Homogene Transformation	54
3.3.1	Pose eines Starrkörpers	54
3.3.2	Wechsel des Bezugskordinatensystems	54
3.3.3	Eigenschaften	56
3.3.4	Elementare homogene Transformationen	57
3.3.5	Versatz eines Vektors oder Koordinatensystems	57
3.4	Bewegte Koordinatensysteme	58
3.4.1	Drehwinkelgeschwindigkeiten	59
3.4.2	Räumliche Geschwindigkeit	62
3.4.3	Transformation räumlicher Geschwindigkeiten	63
3.5	Parametrierungen der Orientierung	64
3.5.1	Euler-Winkel	65
3.5.2	Exponentielle Koordinaten für Orientierungen	67
3.5.3	Einheitsquaternionen	68
3.6	Literatur	70
4	Roboterkinematik	71
4.1	Serielle Kinematik	71
4.1.1	Vorwärtskinematik	71
4.1.2	Kinematikrepräsentationen	75
4.1.3	Differentielle Kinematik	78
4.1.4	Inverse Kinematik	91
4.1.5	Differentielle inverse Kinematik	97
4.1.6	Kinematische Redundanz	99
4.2	Parallele Kinematik	102
4.2.1	Eigenschaften	102
4.2.2	Bauformen	103
4.2.3	Stewart-Gough-Parallelroboter	104
4.3	Literatur	108
5	Dynamik	111
5.1	Newton-Euler-Gleichung für einen Starrkörper	111
5.1.1	Newton-Euler-Gleichungen im Inertialsystem	112
5.1.2	Newton-Euler-Gleichungen in körperfesten Koordinaten	113
5.2	Lagrange-Formulierung	115
5.2.1	Kinetische und potentielle Energie	115
5.2.2	Lagrange-Gleichungen	116
5.2.3	Berücksichtigung der Rotorträgheiten	118
5.3	Eigenschaften der Dynamikkomponenten	118

5.4	Rekursive Newton-Euler-Formulierung	121
5.4.1	Formulierung im Inertialsystem	121
5.4.2	Formulierung in körperfesten Koordinaten	124
5.5	Vorwärtsdynamik	125
5.6	Zwangsbedingungen (Bindungen)	126
5.6.1	Baumgarte Stabilisierung holonomer Bindungen	128
5.6.2	Reduktion der Freiheitsgrade	129
5.7	Literatur	132
6	Regelung	133
6.1	Regelungstechnische Grundlagen	133
6.1.1	Stabilität nichtlinearer Systeme	133
6.1.2	Passivität	135
6.1.3	Zustandsraumdarstellung der Roboterdynamik	135
6.2	Regelung im Gelenkraum	136
6.2.1	Stabilisierung eines Arbeitspunktes	136
6.2.2	Trajektorienfolgeregelung	139
6.2.3	Computed-Torque-Regler	139
6.2.4	PD+ Regler	140
6.2.5	Zusammenfassung der Gelenkregler	141
6.3	Regelung im Aufgabenraum	142
6.3.1	Nichtredundante Kinematik	142
6.3.2	Redundante Kinematik	145
6.3.3	Zusammenfassung der Regler im Aufgabenraum	147
6.4	Kraft- und Impedanzregelung	148
6.4.1	Kartesische Impedanzregelung	148
6.4.2	Kartesische Nachgiebigkeitsregelung	149
6.4.3	Orientierungssteifigkeit	150
6.4.4	Kraftregelung	150
6.5	Ausblick: Unteraktuierte Systeme	151
6.6	Anhang	153
6.6.1	Anhang I: Lyapunov-Funktion des PD-Reglers im Gelenkraum	153
6.6.2	Anhang II: Kartesische Nachgiebigkeitsregelung um eine Solltrajektorie	154
6.7	Literatur	155
7	Bewegungsplanung	157
7.1	Trajektorienplanung	158
7.1.1	Pfade und Trajektorien	159
7.1.2	Geradlinige Pfade	160
7.1.3	Zeitparametrierung	163
7.2	Bewegungsplanung	174
7.2.1	Grundlagen	174
7.2.2	Definition	177
7.2.3	Kategorien und Eigenschaften	177
7.2.4	Methoden	180

7.3	Literatur	186
8	Mobile Roboter	187
8.1	Starrkörperbewegungen in der Ebene	187
8.2	Omnidirektionale mobile Roboter	188
8.2.1	Kinematisches Modell	189
8.2.2	Planung	194
8.2.3	Regelung	194
8.3	Nichtholonome mobile Roboter	194
8.3.1	Nichtholonome Zwangsbedingungen	195
8.3.2	Kinematisches Modell	197
8.3.3	Erreichbarkeit	198
8.3.4	Planung	203
8.3.5	Regelung	204
8.4	Odometrie	205
8.5	Literatur	206

1 Einführung

Die *Robotik* befasst sich mit der Erforschung jener Maschinen, die den Menschen bei der Ausführung einer Aufgabe ersetzen können – sowohl in Bezug auf körperliche Aktivität als auch beim Füllen von Entscheidung. Robotik hat tiefe kulturelle Wurzeln. Im Laufe der Jahrhunderte haben Menschen ständig versucht, Maschinen zu erschaffen, die in der Lage sind, sie nachzuahmen und ihr Verhalten in den verschiedenen Interaktionen mit der Umgebung auszuführen. Diese kontinuierliche Suche wird auf unterschiedliche Ebenen getrieben, die auf philosophischen, wirtschaftlichen, sozialen und wissenschaftlichen Prinzipien basieren.

Frühe robotische Konzepte finden sich unter anderem in Leonardo Da Vincis *mechanischem Ritter* (ca. 1500) und den mechanisierten Puppen *karakuri-ningyo* aus Japan (siehe Abb. 1.1). Der Begriff „Roboter“ geht auf das tschechische Wort „robota“ (dt. Zwangsarbeit) zurück. In dem 1920 veröffentlichten Bühnenstück „R.U.R. - Rossums Universal Robots“ (siehe Abb. 1.2) beschreibt der tschechische Schriftsteller Karel Čapek eine futuristische Gesellschaft in der Menschen unangenehme Arbeiten von ihren robotischen Sklaven verrichten lassen.

Der von den US-Amerikanern George Devol und Joseph Engelberger entwickelte Roboter „Unimate“ (siehe Abb. 1.3) gilt als der weltweit erste Industrieroboter. Bereits Anfang der 60er-Jahre wurde dieser Roboter in der Fertigungsstraße bei dem Automobilhersteller General Motors für Schweiß- und Lackieraufgaben eingesetzt.

In den 80er Jahren wurde eine auch heute noch genutzte abstrakte Definition der Robotik geprägt als „Wissenschaft, welche sich mit der intelligenten Verbindung von Perzeption und Aktion beschäftigt“. Neben der Perzeption (d.h. der Messung des Zustandes und der



(a)



(b)

Abbildung 1.1: Beispiele für frühzeitige robotische Konzepte. (a) Leonardo Da Vincis mechanischer Ritter, (b) Mechanisierte Puppe *karakuri-ningyo* aus Japan



Abbildung 1.2: Tschechisches Bühnenstück „R.U.R. - Rossums Universal Robots“

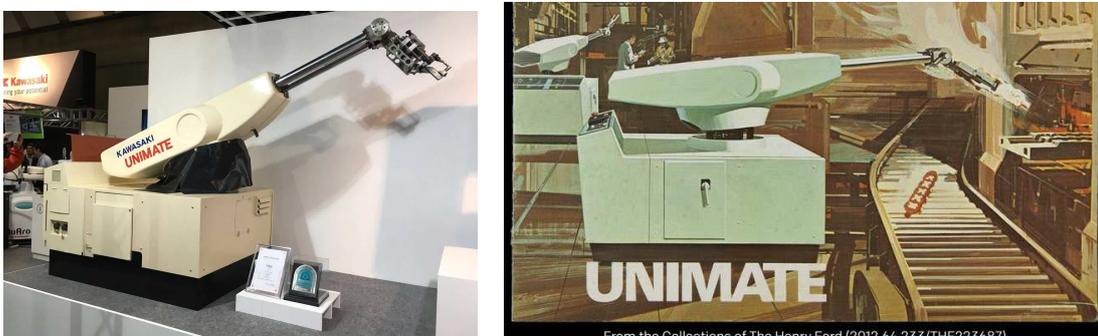


Abbildung 1.3: Der 1956 vorgestellte erste Industrieroboter „Unimate“.

Umgebung) und der Aktion (Manipulation, Lokomotion) ist hierbei auch die „intelligente Verbindung“ ein wesentliches Merkmal welches sich auf die Programmierung, Planung, sowie Steuerung und Regelung bezieht.

Dieses einleitende Kapitel beschäftigt sich mit einer Reihe von Grundbegriffen und gibt einen breiten Überblick über das große Forschungs- und Themenfeld der Robotik. Weiters werden der Konfigurationsraum und der Aufgabenraum begrifflich eingeordnet und im Detail erklärt.

1.1 Grundbegriffe

Ein *Roboter* ist im Allgemeinen ein mechatronisches System bestehend aus einer Reihe von Körpern, den *Gliedern* bzw. *Segmenten* (engl. *Links*), welche durch *Gelenke* (engl. *Joints*) miteinander verbunden sind. Während es Roboter in vielen unterschiedlichen Bauformen gibt – von Nanorobotern bis Großraummanipulatoren, siehe Abbildung 1.4 –, sind diese Begriffe auf alle Roboter gleichermaßen anwendbar und werden im Folgenden einführend erklärt.

Glied Die Glieder (auch „Segmente“) eines Roboters sind die einzelnen Körper, aus denen

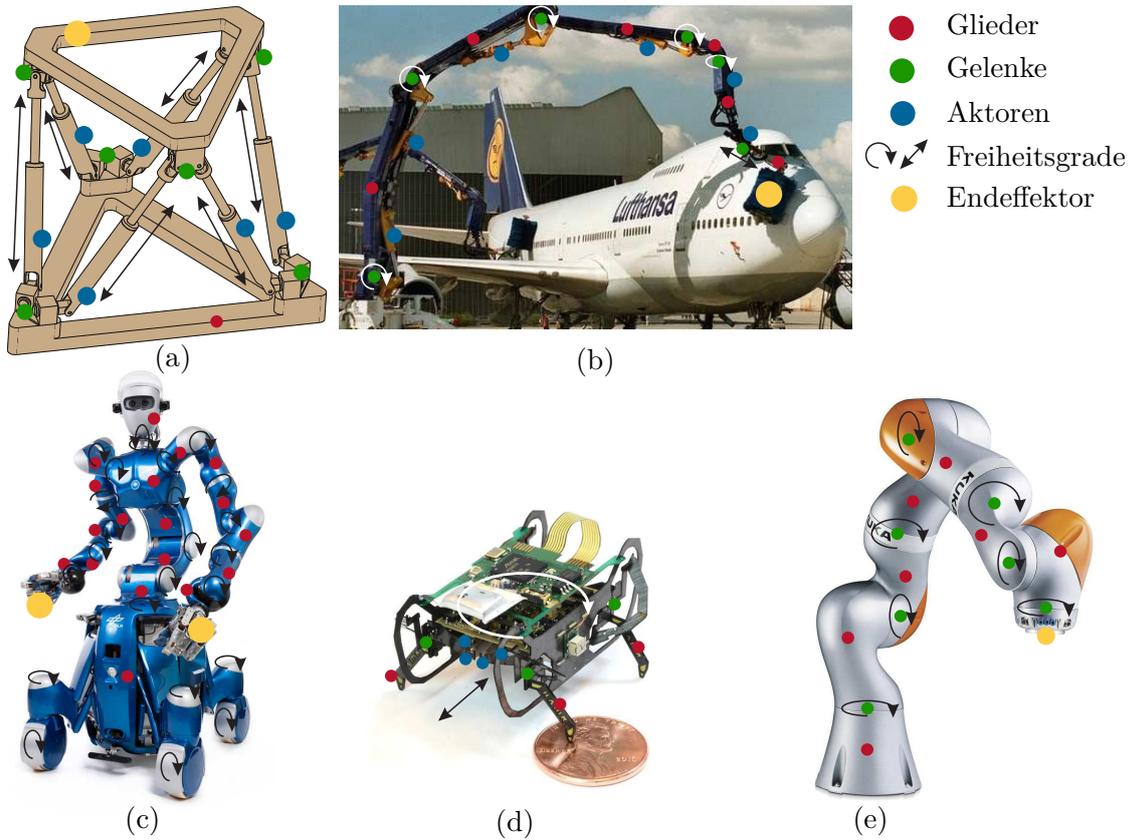


Abbildung 1.4: Beispiele für Roboter mit Markierungen für Glieder, Gelenke, Aktoren, Freiheitsgrade und Endeffektoren. (a) Steward-Gough-Plattform [1.1], (b) Flugzeugreinigungsroboter „Skywash“, (c) humanoider Roboter „Rollin’ Justin“, (d) Haward’s Ambulatory Micro Robot (HAMR), (e) kollaborativer Roboter KUKA LBR iiwa

der Roboter zusammengesetzt ist. Im einfachsten Fall werden sämtliche Glieder des Roboters als Starrkörper aufgefasst. Diese Annahme ist für klassische Industrieroboter hinreichend erfüllt und wird in dieser Vorlesung für alle betrachteten Systeme vorausgesetzt. Die Glieder von Großraummanipulatoren unterliegen hingegen häufig einer elastischen Verbiegung aufgrund des Eigengewichts und werden daher als Biegebalken mithilfe der Balkentheorie modelliert.

Gelenk Ein Gelenk verbindet genau zwei Glieder eines Roboters miteinander. Diese Verbindung hat zur Folge, dass die Bewegungsfreiheit der beiden Körper zueinander eingeschränkt ist. Die am häufigsten eingesetzten Gelenke erlauben nur eine Drehung um eine bestimmte Achse oder eine Verschiebung entlang einer bestimmten Achse. Gelenke können aktiv angetrieben (aktuiert) werden, d. h. mit einem Aktor versehen sein, oder passiv sein, d. h. frei beweglich sein. Während in seriellen Manipulatorarmen wie z.B. dem KUKA LBR iiwa (Abbildung 1.4e) alle Gelenke als Rotationsgelenke ausgelegt und angetrieben sind, besteht die Steward-

Gough-Plattform (Abbildung 1.4a) aus sechs Schubgelenken und vielen passiven Drehgelenken.

Aktor Ein im Roboter verbauter Antrieb wird als Aktor (auch „Aktuator“) bezeichnet. In der Robotik werden viele unterschiedliche Arten von Aktoren eingesetzt. Klassische Industrieroboter verwenden überwiegend Elektromotoren, während bei Großraummanipulatoren häufig hydraulische Antriebe zum Einsatz kommen. Bei den Mikro- und Nanorobotern kommen häufig Piezoantriebe bzw. mikro-elektromechanische Systeme (MEMS) vor. Auch pneumatische und hydraulische Antriebe sind in der Industrie verbreitet. Weiters kann ein Aktor entweder direkt auf ein Gelenk wirken, wie z.B. beim KUKA LBR iiwa (Abbildung 1.4e), oder auch indirekt, wie z.B. beim Flugzeugreinigungsroboter „Skywash“ (Abbildung 1.4b). Bei letzterem wird die linear Schubbewegung der Hydraulikzylinder mithilfe einer passiven Gelenksstruktur auf eine Rotationsbewegung umgesetzt.

Freiheitsgrad Durch die Verbindung der Glieder eines Roboters mithilfe von Gelenken wird die Bewegungsfreiheit des Gesamtsystem eingeschränkt. Jede verbleibende – unabhängige – Bewegungsfreiheit wird als Freiheitsgrad bezeichnet. Für jedes Robotersystem kann ein Satz von Freiheitsgraden gewählt werden, z.B. die Drehwinkel der Rotationsgelenke und die Verschiebepositionen der Schubgelenke. Ein vollständiger Satz von Freiheitsgraden für ein Robotersystem erfüllt daher folgende Eigenschaften:

- Das System bzw. die Lage aller Glieder ist durch Festlegen von Zahlenwerten für alle Freiheitsgrade eindeutig bestimmt.
- Die Eindeutigkeit geht verloren, sobald ein Freiheitsgrad weggelassen wird.
- Der Zahlenwert eines einzelnen Freiheitsgrades kann verändert werden, ohne dass sich damit die Zahlenwerte von anderen Freiheitsgraden unmittelbar ändern.

Endeffektor Einerseits wird als Endeffektor ein ausgezeichnetes Glied eines Roboters bezeichnet, welches für die Interaktion mit der Umgebung verwendet wird. Andererseits wird mit dem Begriff auch das Werkzeug bzw. der Greifer bezeichnet, welches bzw. welcher am Roboter montiert ist. Industrieroboter weisen dazu häufig einen standardisierten Montageflansch auf, an dem unterschiedliche Werkzeuge und Greifer montiert werden können.

1.2 Roboterarten

Das Feld der Robotik ist äußerst umfangreich und umfasst viele unterschiedliche Disziplinen. Es überrascht daher nicht, dass bereits ein breites Spektrum an Roboterarten existiert. Einige wichtige Roboterarten in Bezug auf Bauform und Anwendung werden im Folgenden kurz beschrieben.

Industrieroboter Ein *Industrieroboter* ist eine Maschine mit sehr hoher Vielseitigkeit und Flexibilität. Gemäß der Norm ISO 8373:2021-11 ist ein Industrieroboter definiert als ein „automatisch gesteuerter, frei programmierbarer Mehrzweck-Manipulator,



Abbildung 1.5: Industrieroboter in unterschiedlichen Baugrößen. Die Roboter unterscheiden sich in der maximalen Traglast, Reichweite sowie maximale Beschleunigungen und Geschwindigkeiten am Endeffektor.

der in drei oder mehr Achsen programmierbar ist und zur Verwendung in der Automatisierungstechnik entweder an einem festen Ort oder beweglich angeordnet sein kann.” [1.2] Zu den wichtigsten Eigenschaften zählen

- Vielseitigkeit beim Einsatz von unterschiedlichen Werkzeugen am Endeffektor des Roboters,
- Positioniergenauigkeit durch den Einsatz einer computergestützten numerischen Steuerung,
- Wiederholgenauigkeit bei der Ausführung in Bezug auf die Programmierung von unterschiedlichen Aufgaben.

In Abbildung 1.5 ist eine Reihe von Industrierobotern in unterschiedlichen Baugrößen dargestellt. Die meisten Industrieroboter haben die gleiche Bauform, nämlich sind dies 6-Achs-Roboter mit ausschließlich rotierenden Achsen, einem sogenannten *sphärischen Handgelenk* (engl. *spherical wrist*) und starren Robotergliedern.

Selective Compliance Assembly Robot Arm (SCARA) Ein *SCARA* ist eine spezielle Bauform eines Industrieroboters, welche für schnelle Abfolgen von *Pick-and-Place* (dt. *Aufnehmen-und-Ablegen/Fügen*) konzipiert sind. Beispiele von SCARA sind in Abbildung 1.6 dargestellt. Sie verfügen meist über mehrere Drehgelenke mit parallelen Achsen und einem Schubgelenk. Dadurch wird eine hohe Dynamik mit einem relativ großen Arbeitsraum erreicht.

Parallelroboter Bei einem *Parallelroboter* gibt es mehrere Reihen von Gliedern und Gelenken, die die Roboterbasis mit dem Endeffektor verbinden. Beispiele von Parallelrobotern zeigt Abbildung 1.7. Der grundsätzliche Vorteil dieser Roboterbauform ist die hohe Struktursteifigkeit und die verhältnismäßig geringe bewegte Masse. Dadurch ergibt sich die Möglichkeit, hohe Arbeitsgeschwindigkeiten und eine hohe Wiederholgenauigkeit zu erreichen mit dem Nachteil eines stark reduzierten Arbeitsraumes. Je nach Ausführung können 3-Achs-Parallelroboter reine Positionieraufgaben durchführen, ohne den Endeffektor zu drehen, während 6-Achs-Parallelroboter den Endeffektor in allen drei Achsen verschieben und drehen können.



Abbildung 1.6: Beispiele für SCARA. (a) ABB IRB 910SC, (b) Yamaha YK-X-Serie, (c) KUKA KR 6 R700 Z200

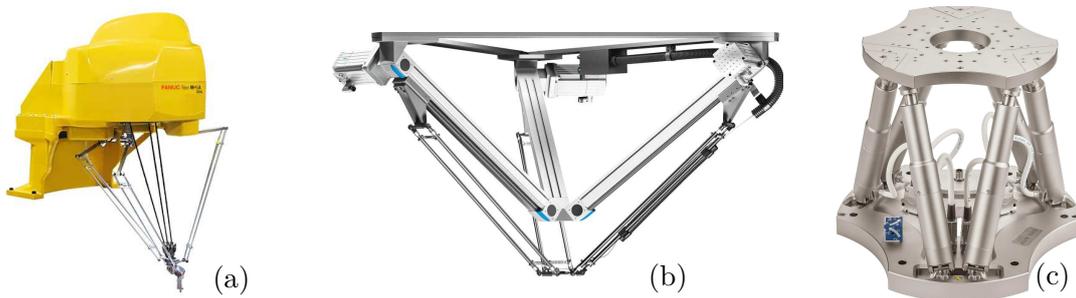


Abbildung 1.7: Beispiele für Parallelroboter. (a) Fanuc Robot M-1iA, (b) Festo Tripod, (c) PI 6-Achsen-Hexapod

Kollaborative Roboter Die Kategorie der *kollaborativen Roboter* (engl. *Collaborative Robot, Cobot*) umfasst Industrieroboter, die für die direkte (physische) Zusammenarbeit mit dem Menschen konzipiert wurden. Durch eine Vielzahl an integrierten Sensoren sowie leichter und runder Bauweise können Sicherheitsgarantien gegeben werden, wodurch Schutzzäune zwischen Mensch und Roboter entfallen können. Beispiele für direkt verbaute Sensoren sind Drehmomentsensoren in allen Achsen, ein Kraft-/Drehmomentsensor am Endeffektor, sowie 2D- und 3D-Kameras am Endeffektor. Abbildung 1.8 zeigt häufig eingesetzte kollaborative Roboter.

Mikro- und Nanoroboter Das Gebiet der *Mikrorobotik* umfasst die robotische Manipulation von Objekten mit Abmessungen im Millimeter- bis Mikrometerbereich sowie das Design und die Herstellung von autonomen Robotern, die in diesen Größenbereich fallen. *Nanorobotik* wird auf die gleiche Weise nur für Dimensionen kleiner als ein Mikrometer definiert. Mit der Fähigkeit, Objekte mit Abmessungen im Mikro- und Nanometerbereich zu positionieren und auszurichten, kann die Montage von Mikro- und Nanosystemen (einschließlich der Roboter selbst) ermöglicht werden. Aktuelle Beispiele aus der Literatur sind in Abbildung 1.9 gezeigt.

(Autonome) mobile Roboter Autonome mobile Roboter sind Robotersysteme, die sich in ihrer Umgebung selbstständig orientieren, bewegen und agieren können. Bekannte Beispiele umfassen autonome Fahrzeuge, Saug-Wisch-Roboter für den Haushalt sowie Forschungsroboter für die Erkundung von Planeten, siehe Abbildung

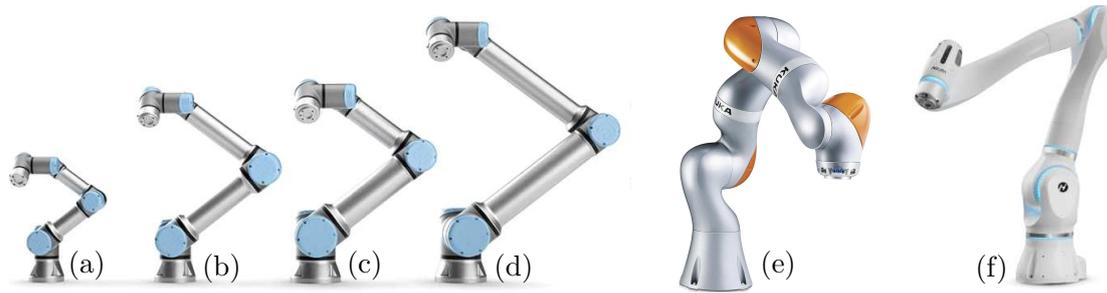


Abbildung 1.8: Beispiele für kollaborative Roboter. (a) Universal Robots UR3, (b) Universal Robots UR5, (c) Universal Robots UR10, (d) Universal Robots UR16, (e) KUKA LBR iiwa, (f) Neura Robotics MAiRA (Multi-Sensing Intelligent Robotic Assistant)

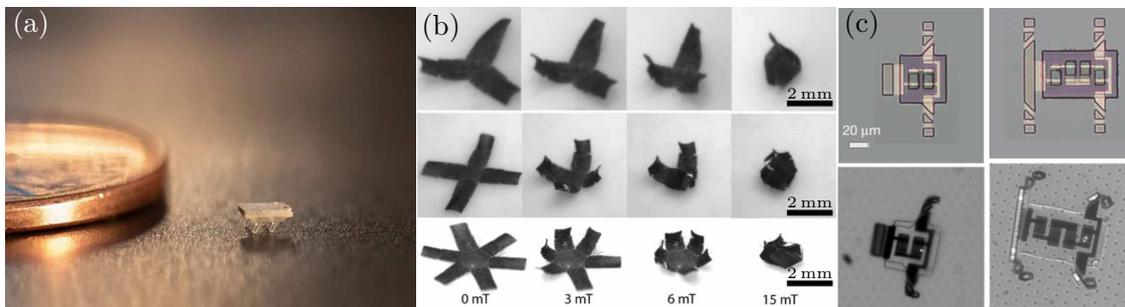


Abbildung 1.9: Beispiele für Mikro- und Nanoroboter. (a) Bürstenroboter der Größe einer Ameise, angetrieben durch externe Schallwellen (Allison Carter, Georgia Tech), (b) Miniaturisierter 3-, 4- und 6-Finger-Greifer durch äußeres Magnetfeld in Bewegung versetzt (Tianqi Xu, University of Toronto), (c) Laser-betriebener, krabbelfähige Mikroroboter, welche für die Massenproduktion auf Siliziumwafern konzipiert sind (Marc Z. Miskin, Cornell University)

1.10. Die Autonomie des Roboters (d. h. die Selbstbestimmtheit) kann dabei in unterschiedlichen Abstufungen realisiert werden. Bei autonomen Fahrzeugen sind dazu beispielsweise sechs „Stufen“ definiert, von Stufe 0 (Selbstfahrer) über Stufe 2 (Teilautomatisierung) bis hin zu Stufe 5 (Vollautomatisierung).

Service-Roboter Die Klasse der *Service-Roboter* umfasst semi- oder vollautonome Roboter, welche Dienstleistungen für den Menschen erbringen. Beispiele für Dienstleistungsroboter für den privaten Haushalt sind Saug-Wisch-Roboter, Rasenmäherroboter und Spielzeugroboter. Auch im professionellen Bereich finden Service-Roboter immer mehr Verbreitung, z. B. in der Gastronomie für den Transport von Speisen und Getränken zum Gast oder in öffentlichen Gebäuden für die Kommunikation mit Besucher:innen, siehe Abbildung 1.11.

Humanoide Roboter Humanoide Roboter ahmen selektiv Aspekte der menschlichen



Abbildung 1.10: Beispiele für autonome mobile Roboter. (a) Autonomes Fahrzeug (Waymo, Google), (b) Saug-Wisch-Roboter, (c) Mars Science Laboratory Mission's Curiosity Rover (NASA)



Abbildung 1.11: Beispiele für Serviceroboter. (a) Toyota Human Support Robot (HSR), (b) Pudu Robotics BellaBot, (c) Aldebaran Robotics Pepper

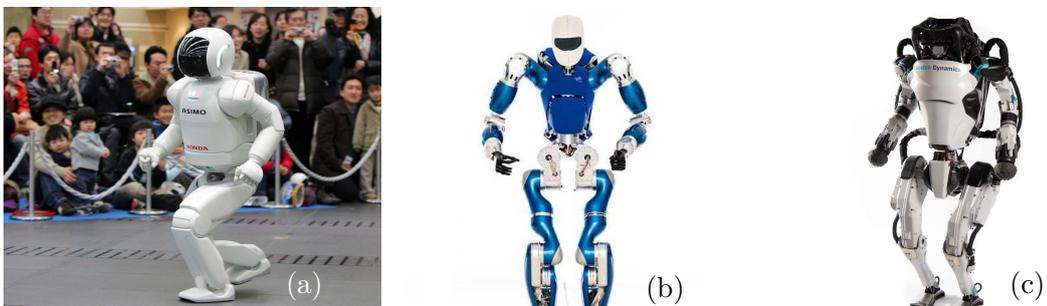


Abbildung 1.12: Beispiele für humanoide Roboter. (a) Honda ASIMO (Advanced Step in Innovative Mobility), (b) DLR TORO (Torque-controlled humanoid Robot), (c) Boston Dynamics Atlas

Form und des menschlichen Verhaltens nach. Humanoide Roboter gibt es in einer Vielzahl von Formen und Größen, von kompletten Robotern mit Beinen in Menschengröße bis hin zu separaten Roboterköpfen mit menschenähnlicher Wahrnehmung und Ausdruck. Beispiele für humanoide Roboter sind in [Abbildung 1.12](#) dargestellt.

Laufroboter Mehrbeinige Laufroboter werden in unstrukturierten Umgebungen eingesetzt. In vierbeinigen Laufrobotern werden unter anderem biologisch inspirierte



Abbildung 1.13: Beispiele für vierbeinige Laufroboter. (a) ANYmal (ANYbotics), (b) Spot Mini (Boston Dynamics)



Abbildung 1.14: Beispiele für Flugroboter. (a) MQ-9A „Reaper“ (U.S. Air Force), (b) DJI Phantom 3, (c) Lieferung per Drohne (Deutsche Post DHL Group)

Gangmuster wie das Gehen, Traben, Galoppieren, oder der Passgang genutzt. Typischerweise sind vierbeinige Laufroboter mit relativ leichten Beinen aufgebaut, um eine schnelle Schrittadaption zu erlauben. Abbildung 1.13 zeigt zwei kommerziell erhältliche Laufroboter.

Flugroboter *Flugroboter* (engl. *Unmanned Aerial Vehicle*, UAV) haben ihre Wurzeln in den ersten Lenkflugkörpern. Heutzutage bezieht sich der Begriff jedoch auf eine Vielzahl fortschrittlicher intelligenter Systeme. Nach Angaben des American Institute of Aeronautics and Astronautics (AIAA) [1.3] ist ein UAV definiert als „ein Luftfahrzeug, das konstruiert oder modifiziert wurde, um *keinen* menschlichen Piloten zu transportieren und entweder vom Fluglotsen oder durch eine autonome Flugsteuerung an Bord des Roboters gesteuert wird.“ Wie allgemein in der Robotik werden Flugroboter immer komplexere Systeme, die zu fortgeschrittener Entscheidungsfindung und Planung fähig sind. Basierend auf der Wahrnehmung Ihrer Umwelt mithilfe von On-Board-Sensorik können herausfordernde Aufgaben gelöst werden.

1.3 Konfigurationsraum

Die Bewegungen eines Roboters können sowohl im sogenannten *Konfigurationsraum* als auch im *Aufgabenraum* beschrieben werden. Im folgenden Abschnitt wird zunächst der Konfigurationsraum behandelt. Zunächst werden die Freiheitsgrade eines einzelnen Starrkörpers sowie eines gesamten Robotersystems diskutiert und die Begriffe der *Konfiguration* und des Konfigurationsraums eingeführt. Mithilfe von Gelenken werden kinematische Ketten zusammengesetzt und mithilfe der *Formel von Grübler* analysiert. Abschließend werden Konfigurationsräume topologisch betrachtet. Abschnitt 1.4 widmet sich dem Aufgabenraum.

1.3.1 Freiheitsgrade eines Starrkörpers und eines Robotersystems

Ein frei beweglicher Starrkörper wird im dreidimensionalen Raum durch 6 *Freiheitsgrade* beschrieben, nämlich 3 Freiheitsgrade für die *translatorische* Verschiebung und 3 Freiheitsgrade zur Beschreibung der Orientierung (*rotatorische* Freiheitsgrade) des Starrkörpers zum Inertialsystem. Eine Punktmasse im dreidimensionalen Raum hingegen weist nur die 3 translatorischen Freiheitsgrade auf. Im zweidimensionalen Raum besitzt ein Starrkörper insgesamt 3 Freiheitsgrade, das sind zwei für die Position des Starrkörpers in der Ebene und ein Freiheitsgrad für die Orientierung. Die räumliche Lage eines Starrkörpers (Kombination von Position und Orientierung) oder die Position einer Punktmasse im zwei- oder dreidimensionalen Raum, wird auch als *Pose* bezeichnet.

Die Bewegung eines Starrkörpers unterliegt aber im Allgemeinen *Zwangsbedingungen*, die die Bewegungsfreiheit des Starrkörpers einschränken. Zum Beispiel, eine Münze im dreidimensionalen Raum ist ein Starrkörper mit 6 Freiheitsgraden, siehe Abbildung 1.15¹. Wird diese Münze aber auf einen Tisch gelegt, so entstehen durch den Tisch drei Zwangsbedingungen: Es wird ein translatorischer Freiheitsgrad (d. h. die vertikale Position) und zwei rotatorische Freiheitsgrade (d. h. die Rotation um die beiden Achsen der Tischebene) durch den Tisch vorgegeben. Es verbleiben daher zwei translatorische Freiheitsgrade, das ist die zweidimensionale Position der Münze am Tisch, und ein rotatorischer Freiheitsgrad, das ist die Rotation um den Oberflächennormalenvektor der Tischplatte.

Hinter diesem Beispiel steht die allgemeine Regel für die Bestimmung der Anzahl der Freiheitsgrade n eines Starrkörper- bzw. Robotersystems

$$n = (\text{Summe der Freiheitsgrade der Starrkörper}) - (\text{Anzahl der unabhängigen Zwangsbedingungen}) . \quad (1.1)$$

Die Anzahl der Freiheitsgrade n ist die minimale Anzahl an reellwertigen Koordinaten die erforderlich sind, um die Lage aller Starrkörper – und damit jedes Massepunktes aller Starrkörper – in einem mechanischen System festzulegen.

¹In Abbildung 1.15 wird die Farbkonvention für Koordinatensysteme in diesem Skriptum eingeführt: Die x -Achse ist rot, die y -Achse ist grün und die z -Achse ist blau ($xyz \Leftrightarrow RGB$).

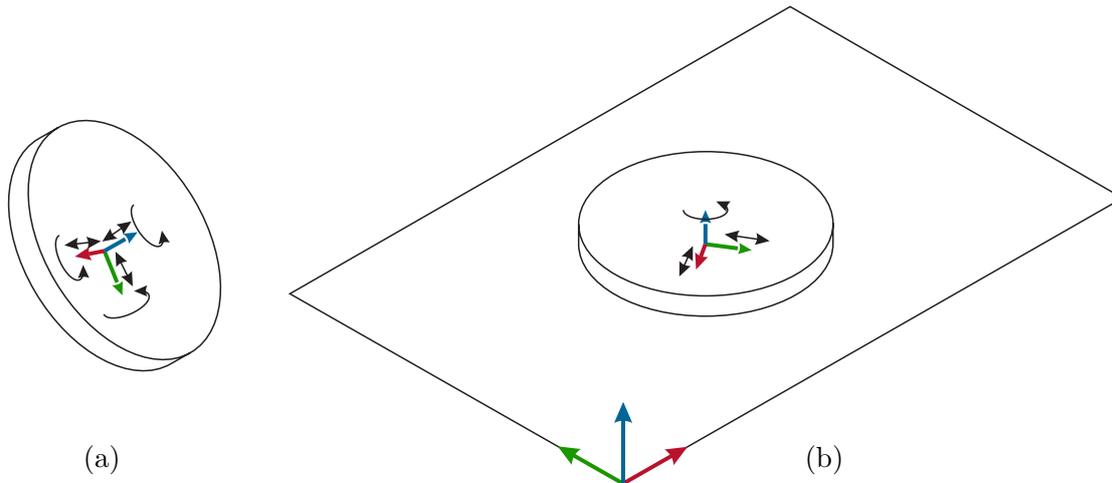


Abbildung 1.15: Freiheitsgrade einer Münze. (a) Münze ohne Zwangsbedingungen mit 6 Freiheitsgraden, (b) Münze auf einem Tisch mit 3 Freiheitsgraden

1.3.2 Konfiguration

Die obige konzeptionelle Beschreibung der Freiheitsgrade führt zur formalen Definition der beiden Begriffe *Konfiguration* (engl. *Configuration*) und *Konfigurationsraum* (engl. *Configuration Space*, C-Space).

Definition 1.1. Die *Konfiguration* eines Roboters spezifiziert die räumliche Lage sämtlicher Körper des Roboters. Eine Konfiguration wird durch die n reellwertigen Koordinaten der Freiheitsgrade beschrieben.

Die Menge aller Konfigurationen, die ein Roboter einnehmen kann, wird als *Konfigurationsraum* bezeichnet und ist n -dimensional. Jede Konfiguration wird durch einen Punkt im Konfigurationsraum repräsentiert.

Die Dimension des Konfigurationsraumes entspricht also genau der Anzahl an Freiheitsgraden n des Robotersystems.

1.3.3 Gelenke

In Abbildung 1.16 sind einige Gelenkstypen dargestellt, welche häufig in Robotern verbaut werden. Das *Rotationsgelenk* (auch *Drehgelenk*, engl. *Revolute*, R) ermöglicht die Rotation um eine Achse, während das *Schubgelenk* (engl. *Prismatic*, P) eine lineare Translationsbewegung ausführt. Mit dem *Schraubgelenk* (engl. *Helical*, H) wird eine simultane Rotation und lineare Translation entlang einer Schraubenachse ausgeführt. Diese drei Gelenkstypen bieten genau einen Freiheitsgrad. Weiters gibt es auch Gelenkstypen mit mehreren Freiheitsgraden. Dazu zählt das *Walzengelenk* (engl. *Cylindrical*, C), welches zwei Freiheitsgrade besitzt und im Gegensatz zum Schraubgelenk eine unabhängige Rotations- und Translationsbewegung entlang einer Gelenksachse erlaubt. Das *Kardangelenk* (engl. *Universal*, U) besteht aus zwei sich orthogonal schneidenden Rotationsgelenken und wird durch zwei Freiheitsgrade beschrieben. Drei Freiheitsgrade hat das *Kugelgelenk* (engl.

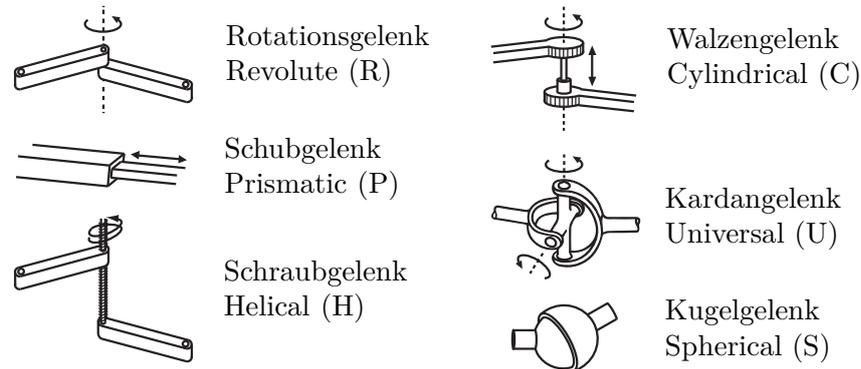


Abbildung 1.16: Unterschiedliche Gelenkstypen [1.4].

Gelenktyp	Freiheitsgrade	Zwangsbedingungen zwischen zwei Starrkörpern im zweidimensionalen Raum	Zwangsbedingungen zwischen zwei Starrkörpern im dreidimensionalen Raum
Rotationsgelenk	1	2	5
Schubgelenk	1	2	5
Schraubgelenk	1	–	5
Walzengelenk	2	–	4
Kardangelenk	2	–	4
Kugelgelenk	3	–	3

Tabelle 1.1: Freiheitsgrade und Zwangsbedingungen für unterschiedliche Gelenkstypen.

Spherical, S), welches eine Bewegung ähnlich dem menschlichen Schultergelenk ermöglicht.

Jedes Gelenk verbindet genau zwei Glieder eines Roboters miteinander und erlaubt eine Bewegung entsprechend der Anzahl an Freiheitsgraden des Gelenkes. Auf der anderen Seite ruft das Gelenk durch die mechanische Verbindung auch Zwangsbedingungen zwischen den beiden Körpern hervor und schränkt dadurch die Bewegungsfreiheit ein. Zum Beispiel erlaubt das Rotationsgelenk eine Drehung um eine Achse, also einen Freiheitsgrad, aber führt auch zu insgesamt fünf Zwangsbedingungen für die Bewegung zwischen den beiden Körpern. Die genannten Gelenkstypen mit der zugehörigen Anzahl an Freiheitsgraden und Zwangsbedingungen im zwei- und dreidimensionalen Raum sind in Tabelle 1.1 aufgelistet.

1.3.4 Generalisierte Koordinaten

Ein Starrkörpersystem, welches aus N Gliedern besteht, benötigt im Allgemeinen $6N$ Koordinaten zur Beschreibung der Posen aller Glieder in Bezug auf ein Inertialkoordinatensystem. Da die Glieder durch Gelenke verbunden sind, gibt es eine Reihe von Zwangsbedingungen zwischen den allgemeinen $6N$ Koordinaten, siehe Abschnitt 1.3.3.

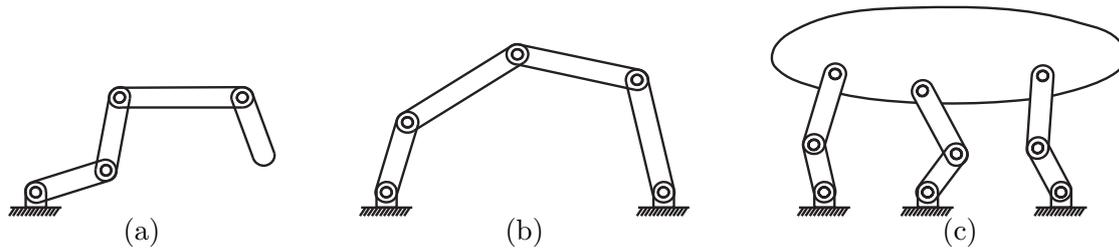


Abbildung 1.17: Kinematische Ketten [1.4]: (a) Offene bzw. serielle Kette, (b) geschlossene Kette, (c) Parallele Kinematik.

Dies bedeutet, dass die $6N$ Koordinaten als Funktion einer geringeren Anzahl n an Koordinaten $\mathbf{q} \in \mathbb{R}^n$, welche unabhängig voneinander sind, ausgedrückt werden können. Diese unabhängigen Koordinaten \mathbf{q} werden *generalisierte Koordinaten* genannt und alle Bewegungen zufolge dieser Koordinaten erfüllen stets alle Zwangsbedingungen.

1.3.5 Kinematische Ketten

Werden mehrere Glieder durch Gelenke zu einem Mechanismus verbunden, so bilden diese eine *kinematische Kette*.

Eine kinematische Kette wird als *offen* bezeichnet, wenn sie frei von Schleifen ist. Das heißt, sie lässt sich als Baumstruktur darstellen, wie das Beispiel in Abbildung 1.17a zeigt. Eine kinematische Kette ohne Verzweigungen nennt man auch *seriell*.

Eine *geschlossene* kinematische Kette enthält eine oder mehrere Schleifen, siehe Abbildung 1.17b. Hier ist zu beachten, dass auch der inertialfeste Boden (schraffiert dargestellt) als Teil des Mechanismus zählt.

Ein Spezialfall der geschlossenen kinematischen Kette stellt die *parallele Kinematik* (Abbildung 1.17c) dar. Diese besteht aus einer bewegten Basis, welche über mehrere gleichartige *Stränge* (engl. *legs*) mit dem inertialfesten Boden verbunden sind. Die Stränge sind wiederum kinematische Ketten, welche für sich eine serielle oder parallele Kinematik aufweisen können. Roboter mit paralleler Kinematik weisen eine hohe strukturelle Steifigkeit auf und ermöglichen daher höhere Bewegungsgeschwindigkeiten im Vergleich zu Robotern mit serieller Kinematik. Als Nachteil erweist sich allerdings eine deutliche Einschränkung in der Bewegungsfreiheit des Roboters.

1.3.6 Formel von Grübler

Mithilfe der Formel von Grübler kann die Anzahl der Freiheitsgrade n einer kinematischen Kette bestehend aus Gliedern und Gelenken bestimmt werden und ist im folgenden Satz zusammengefasst.

Satz 1.1 (Formel von Grübler). *Es sei J die Anzahl der Gelenke und N die Anzahl der Glieder einer kinematischen Kette, wobei der inertialfeste Boden auch als Glied zählt. Weiters sei m die Anzahl der Freiheitsgrade jedes Starrkörpers ($m = 3$ im zweidimensionalen Raum, $m = 6$ im dreidimensionalen Raum), f_i die Anzahl an*

Freiheitsgraden des Gelenks i und c_i die Anzahl der durch das Gelenk i hervorgerufenen Zwangsbedingungen, wobei gilt $f_i + c_i = m \forall i, i = 1, \dots, J$. Für diesen Mechanismus lautet die Formel von Grübler

$$\begin{aligned}
 n &= \underbrace{m(N-1)}_{\text{Freiheitsgrade der Starrkörper}} - \underbrace{\sum_{i=1}^J c_i}_{\text{Zwangsbedingungen der Gelenke}} \\
 &= m(N-1-J) + \sum_{i=1}^J f_i, \tag{1.2}
 \end{aligned}$$

unter der Voraussetzung, dass die Zwangsbedingungen der Gelenke unabhängig voneinander sind.

Diese Formel entspricht damit (1.1) angewendet auf kinematische Ketten wie sie in Abschnitt 1.3.5 beschrieben wurden. Im Folgenden wird die Formel von Grübler anhand von drei Beispielen veranschaulicht.

Beispiel 1.1 (Drei kinematische Ketten).

Vier-Balken-Kette Die Vier-Balken-Kette ist in Abbildung 1.18a dargestellt und besteht aus vier Rotationsgelenken und vier Balken, wobei einer der Balken dem inertialfesten Boden entspricht. Es handelt sich um einen zweidimensionalen Mechanismus, daher gilt $m = 3$. Weiters ist $J = 4$, $N = 4$ und $f_i = 1$ für $i = 1, \dots, 4$ und die Formel von Grübler (1.2) liefert $n = 1$, also einen Freiheitsgrad.

Schieber-Kurbel-Mechanismus Abbildung 1.18b zeigt einen Schieber-Kurbel-Mechanismus, welcher aus zwei Balken, dem Schieber und dem inertialfesten Boden besteht. Es ist daher wieder $N = 4$. Der Mechanismus weist $J = 4$ Gelenke mit jeweils $f_i = 1$ ($i = 1, \dots, 4$) auf, nämlich drei Rotationsgelenke und ein Schubgelenk. Durch die Formel von Grübler (1.2) folgt die Anzahl der Freiheitsgrade des Mechanismus $n = 1$.

Delta-Roboter Der Delta-Roboter, schematisch skizziert in Abbildung 1.18c, hat eine starre Plattform (oben) und eine bewegte Plattform (unten), welche durch drei Stränge mit Parallelkinematik verbunden sind. Jeder Strang besteht aus drei Rotationsgelenken, vier Kugelgelenken und fünf Gliedern. Zusammen mit den beiden Plattformen weist dieser Mechanismus also $N = 17$ Glieder und $J = 21$ Gelenke auf. Nach der Formel von Grübler (1.2) hat dieser Mechanismus $n = 15$ Freiheitsgrade. Dabei rufen aber nur drei Freiheitsgrade eine Bewegung in der mobilen Plattform hervor, nämlich die Translationen in die drei Raumrichtungen. Durch die Anordnung der Parallelkinematik als Parallelogramme in den drei Strängen wird eine Rotation der mobilen Plattform verhindert und diese bleibt stets parallel zur starren Plattform. Die verbleibenden zwölf Freiheitsgrade sind Drehungen aller Stäbe in den Parallelogrammen um die jeweils

eigene Längsachse. Werden statt den Kugelgelenken (S) mit $f_i = 3$ lediglich Kardangelenke (U) mit $f_i = 2$ verwendet, so entfallen diese zwölf Freiheitsgrade und es ergeben sich $n = 3$ Freiheitsgrade für den gesamten Delta-Roboter.

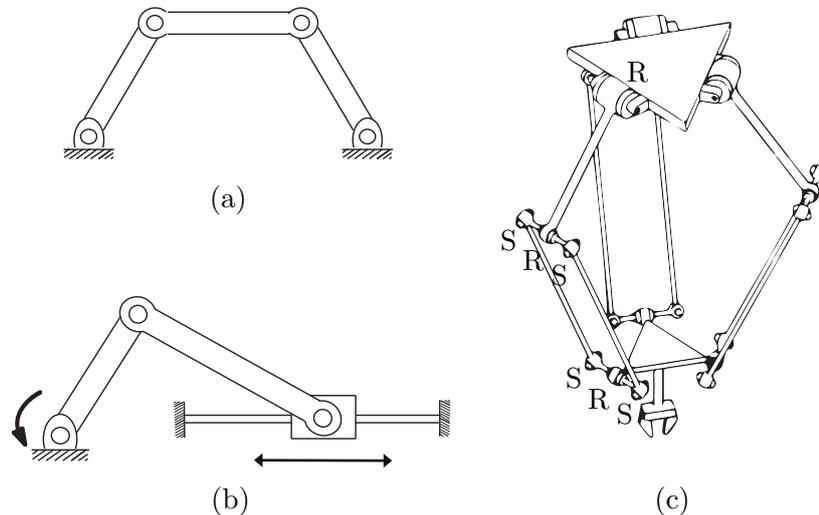


Abbildung 1.18: Beispiele für kinematische Ketten [1.4]: (a) Vier-Balken-Kette (der inertialfeste Boden ist einer der Balken), (b) Schieber-Kurbel-Mechanismus, (c) Delta-Roboter.

1.3.7 Topologien

Während bisher nur die Dimension des Konfigurationsraums diskutiert wurde, wird im Folgenden die Form des Konfigurationsraums betrachtet.

Ein Punkt auf der Oberfläche einer Kugel kann durch einen zweidimensionalen Konfigurationsraum beschrieben werden, z. B. durch den Längen- und Breitengrad. Ein Punkt auf einer Ebene wird durch einen zweidimensionalen Konfigurationsraum beschrieben. Dennoch ist die Form der beiden Konfigurationsräume unterschiedlich. Während die Ebene unendlich ausgedehnt ist, wird eine endliche Fläche zu einer Kugel „zusammengewickelt“ und „geklebt“. Es zeigt sich, dass die beiden Konfigurationsräume eine unterschiedliche *Topologie* besitzen. Dabei ist hervorzuheben, dass die Topologie eines Konfigurationsraumes eine fundamentale Eigenschaft des Raumes selbst ist, welche völlig unabhängig von der Wahl der Koordinaten für einen Punkt in diesem Raum ist. Zwei Konfigurationsräume sind *topologisch äquivalent*, wenn eine stetige Verformung ausreicht – ohne zu schneiden oder zu kleben –, um von der einen Form zur anderen zu gelangen.

Die beiden genannten Beispiele sind in Tabelle 1.2 dargestellt. Der Punkt auf einer Ebene wird topologisch als zweidimensionaler euklidischer Raum \mathbb{E}^2 dargestellt. Jede Dimension dieses Raums wird mit einer reellen Zahl dargestellt. Ein Beispiel für Koordinaten des Raumes sind die kartesischen Koordinaten mit $(x, y) \in \mathbb{R}^2$. Die Spitze eines sphärischen Pendels entspricht einem Punkt auf einer Kugeloberfläche. Topologisch ist

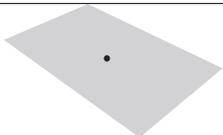
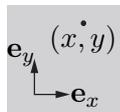
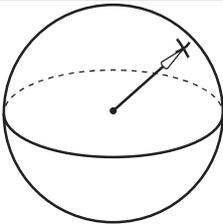
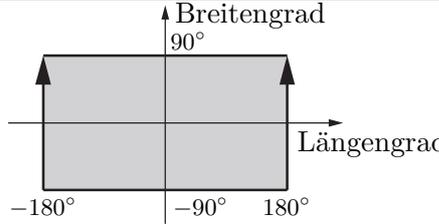
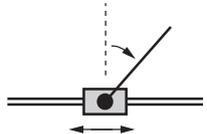
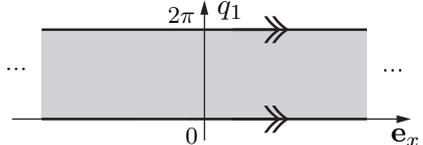
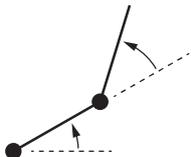
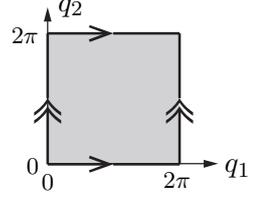
System	Topologie	Darstellungsbeispiel
 Punkt auf Ebene	 \mathbb{E}^2	 \mathbb{R}^2
 Sphärisches Pendel	 S^2	 Breitengrad 90° Längengrad -180° -90° 180° $[-180^\circ, -180^\circ] \times [-90^\circ, -90^\circ]$
 Wagen-Pendel-System	 $\mathbb{E}^1 \times S^1$	 2π q_1 0 e_x $\mathbb{R}^1 \times [0, 2\pi)$
 2R-Roboter	 $T^2 = S^1 \times S^1$	 2π q_2 0 0 2π q_1 $[0, 2\pi) \times [0, 2\pi)$

Tabelle 1.2: Vier unterschiedliche Systeme, deren Topologie und ein Beispiel für den Konfigurationsraum und dessen Koordinaten. Jene Kanten mit den gleichen Pfeilen sind „zusammengeklebt“ und es entsteht eine Hülle bzw. Umwicklung.

der Konfigurationsraum eine Einheitskugel S^2 , d. h. die zweidimensionale Kugeloberfläche im dreidimensionalen Raum². Ein Beispiel für Koordinaten des S^2 sind Längen- und Breitengrade. Diese Beschreibung weist allerdings sogenannte *Koordinatensingularitäten* auf, da die Koordinaten an den Polen der Kugeloberfläche nicht eindeutig sind. Es ist aber zu beachten, dass diese Singularitäten erst durch die Wahl der Koordinaten auftreten – die Kugel selbst hat keine ausgezeichneten Punkte.

Im Allgemeinen können Konfigurationsräume als kartesisches Produkt von zwei oder mehr Räumen dargestellt werden. In Tabelle 1.2 sind zwei weitere Beispiele angeführt, nämlich das Wagen-Pendel-System und ein 2R-Roboter. Das Wagen-Pendel-System erlaubt eine (unendliche) lineare Bewegung des Wagens entlang einer Achse und das Pendel kann um eine Achse drehen, d. h. die Spitze des Pendels beschreibt einen Einheitskreis.

²Allgemein ist eine n -dimensionale Einheitskugel S^n beschrieben durch $S^n = \{\mathbf{x} \in \mathbb{R}^{n+1} \mid \|\mathbf{x}\| = 1\}$.

Dadurch ergibt sich die Topologie in Form eines Rohrs als kartesisches Produkt eines eindimensionalen euklidischen Raumes und dem Einheitskreis $E^1 \times S^1$. Die beiden Achsen des 2R-Roboters können topologisch ebenfalls als Einheitskreise aufgefasst werden. Der Konfigurationsraum folgt zu $S^1 \times S^1 = T^2$, d. h. eine zweidimensionale Oberfläche eines Torus im dreidimensionalen Raum³. Dieser Raum kann mit den Koordinaten $(q_1, q_2) \in [0, 2\pi) \times [0, 2\pi)$ ausgestattet werden. Werden die gegenüberliegenden Kanten dieser Koordinatendarstellung jeweils zusammengeklebt, so entsteht der Torus, siehe Tabelle 1.2.

1.4 Aufgabenraum

Im *Aufgabenraum* (engl. *task space*) werden die Positionen und Bewegungen des Endeffektors des Roboters beschrieben, während sich der Konfigurationsraum auf die Beschreibung des gesamten Roboters mit all seinen Starrkörpern bezieht. Im Folgenden werden die Begriffe Arbeitsraum, Aufgabenraum sowie die Absolut- und Wiederholgenauigkeit eines Roboters allgemein erklärt.

1.4.1 Arbeitsraum

Der *Arbeitsraum* (engl. *workspace*, auch Arbeitsbereich) eines Roboters umfasst nun den gesamten Bereich, den der Roboter mit seinem Endeffektor erreichen kann. Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsräumen sind in Abbildung 1.19 dargestellt.

Bei vielen Endeffektoren kann ein sogenannter *Werkzeugzentrumspunkt* (engl. Tool Center Point, TCP) angegeben werden, welcher einen ausgezeichneten Punkt samt zugehöriger Orientierung am Werkzeug angibt, z. B. der Greifpunkt an einem Greifer oder der Schweißpunkt eines Schweißwerkzeuges. Der Werkzeugzentrumspunkt enthält zwar das Wort „Punkt“ im Namen, wird aber im Allgemeinen als Pose (siehe Abschnitt 1.3.1) relativ zum Roboterflansch charakterisiert.

Beim Arbeitsraum wird häufig zwischen dem *erreichbaren* Arbeitsraum (engl. *reachable workspace*) und dem *gewandten* Arbeitsraum (engl. *dexterous workspace*) unterschieden. Letzterer beschreibt jenen Bereich des Arbeitsraums, den der Roboter mit dem TCP mit unterschiedlichen Orientierungen erreichen kann. In manchen Definitionen wird sogar gefordert, dass der Roboter die Punkte des gewandten Arbeitsraumes mit *allen* Orientierungen anfahren kann. Klassische Industrieroboter besitzen aufgrund der Bauform und den vorhandenen Gelenkwinkelbeschränkungen von realen Gelenken keinen gewandten Arbeitsraum. Im Gegensatz dazu, umfasst der erreichbare Arbeitsraum lediglich jene Punkte, die der Roboter mit zumindest einer Orientierung anfahren kann. Anhand der Definitionen ist nachvollziehbar, dass der gewandte Arbeitsraum ein Unterraum des erreichbaren Arbeitsraumes ist.

³Im Allgemeinen ist $S^1 \times S^1 \times \dots \times S^1 = T^n$, d. h. ein n -dimensionaler Torus in einem $(n+1)$ -dimensionalen Raum. Dieser ist *nicht* äquivalent zu S^n .

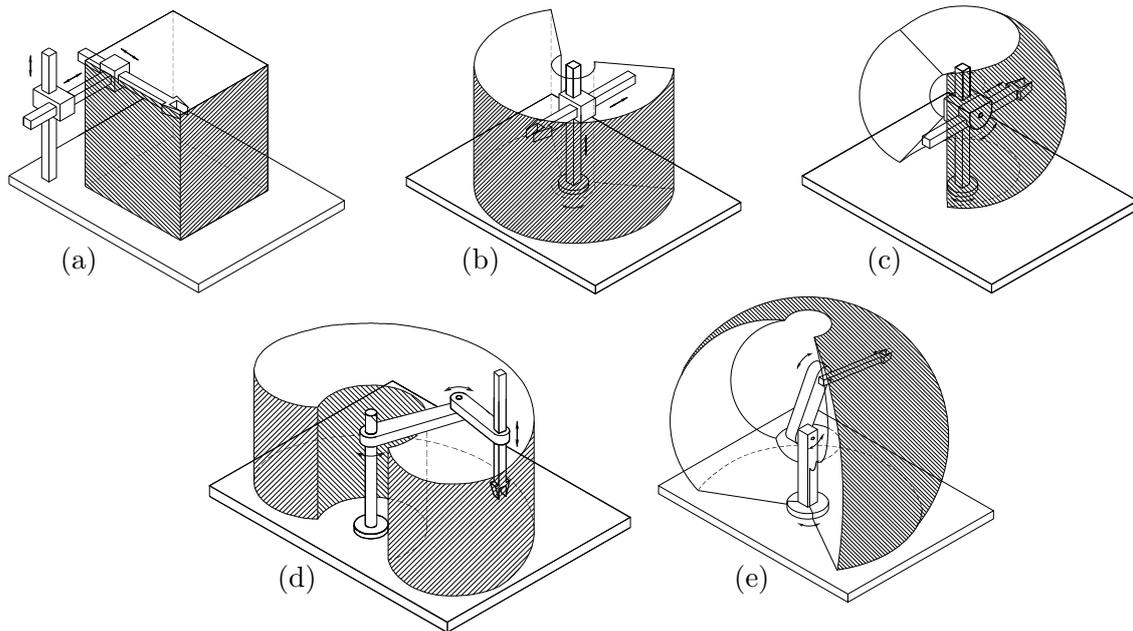


Abbildung 1.19: Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsräumen [1.5]: (a) Kartesischer Roboter (PPP-Roboter), (b) Roboter mit zylindrischem Arbeitsraum (RPP-Roboter), (c) Roboter mit kugelförmigem Arbeitsraum (RRP-Roboter), (d) SCARA-Roboter (Selective Compliance Assembly Robot Arm, RRP-Roboter), (e) Kinematik eines klassischen Industrieroboters mit drei Achsen (RRR-Roboter).

1.4.2 Aufgabenraum

Der Aufgabenraum wird so gewählt, dass die Aufgabe des Roboters geeignet beschrieben werden kann. Im Folgenden wird die Dimension des Aufgabenraumes mit m bezeichnet. Für das Zeichnen mit einem Stift auf ein Blatt Papier ist beispielsweise $m = 2$ ausreichend um die Position des Stiftes am Papier zu charakterisieren. Soll der Roboter mit einer Schneidklinge entlang einer Kontur auf dem Papier schneiden, so wird auch die Orientierung der Klinge und somit $m = 3$ benötigt. Um einen Starrkörper mit einem Greifer im dreidimensionalen Raum zu manipulieren, sind alle Freiheitsgrade dieses Raums notwendig, also $m = 6$.

An dieser Stelle sei noch betont, dass sich der Aufgabenraum eines Roboters konzeptionell vom Konfigurationsraum unterscheidet: Ein Punkt im Aufgabenraum ist keine vollständige Spezifikation für die Konfiguration des Roboters. Auch für klassische Industrieroboter mit $m = n = 6$ kann jede Pose im Aufgabenraum durch mehrere unterschiedliche Konfigurationen erreicht werden.

Falls der Roboter mehr Freiheitsgrade besitzt als im Aufgabenraum benötigt wird, d. h. $n > m$, so nennt man den Roboter *kinematisch redundant*. Das bedeutet, dass der Roboter in einem $(n - m)$ -dimensionalen *Nullraum* des Konfigurationsraums Bewegungen durchführen kann, ohne dass sich die Aufgabenkoordinaten (also z.B. die Pose des Endeffektors

im Aufgabenraum) ändern.

1.4.3 Absolutgenauigkeit und Wiederholgenauigkeit

Für einen Industrieroboter und dessen Einsatz in einem industriellen Prozess sind die beiden Kenngrößen *Absolutgenauigkeit* und *Wiederholgenauigkeit* von großer Bedeutung. Die Absolutgenauigkeit ist ein Maß dafür, wie genau der Roboter einen vorgegebenen kartesischen Punkt in einem inertialfesten Koordinatensystem innerhalb des Aufgabenraumes anfahren kann. Im Gegensatz dazu gibt die Wiederholgenauigkeit lediglich an, wie genau ein Roboter den selben Punkt im Aufgabenraum wiederholt anfahren kann.

Die beiden Begriffe sind in Abbildung 1.20 illustriert: Ein Roboter wird wiederholt von einer Startkonfiguration zur vorgegebenen kartesischen Position $\mathbf{x}_{e,d}$ und wieder zurück gefahren. Die erreichten Positionen im Raum werden dabei aufgezeichnet und es entsteht eine Punktwolke mit dem Mittelpunkt $\bar{\mathbf{x}}_e$ und der dreifachen Standardabweichung r . Für Roboter mit guter Wiederholgenauigkeit ist r sehr klein. Die mittlere erreichte Position $\bar{\mathbf{x}}_e$ weicht in Abbildung 1.20 um den Abstand Δl von der vorgegebenen Position $\mathbf{x}_{e,d}$ ab. Bei Robotern mit guter Absolutgenauigkeit ist Δl für beliebige Sollpositionen $\mathbf{x}_{e,d}$ im Aufgabenraum sehr klein.

Im Allgemeinen kann die Endeffektorpose bei Robotern nicht direkt gemessen werden. Lediglich die Gelenkwinkel können über die Inkrementalencoder in den Gelenken gemessen werden und auch Positionierfehler können nur über diese Messung ermittelt werden. Darüber hinaus kann nur über die als bekannt angenommenen Geometrie und dem starren Aufbau des Roboters auf die Endeffektorpose geschlossen werden. Die Absolutgenauigkeit wird daher von der Fertigungsgenauigkeit des Roboters, der Elastizität und dem Spiel im Getriebe, den Lasten am Endeffektor und vielen anderen statischen und dynamischen Effekten beeinflusst. Aus diesem Grund werden klassische Industrieroboter meist mit sehr hoher Steifigkeit in den Gliedern konstruiert. Andernfalls wäre die Modellannahme eines Starrkörpersystems verletzt.

In den Datenblättern der Industrieroboter ist meist nur die Wiederholgenauigkeit angegeben und eine Absolutgenauigkeit wird nicht spezifiziert. Weiters ist die Wiederholgenauigkeit bei Industrierobotern häufig deutlich besser als die Absolutgenauigkeit, d. h. es gilt $r \ll \Delta l$.

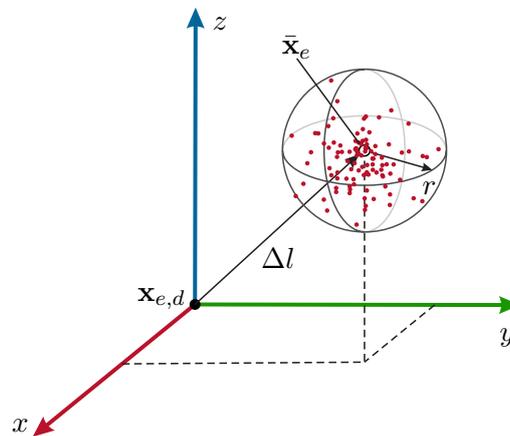


Abbildung 1.20: Absolutgenauigkeit und Wiederholgenauigkeit [1.6].

1.5 Literatur

- [1.1] B. Siciliano und O. Khatib, Hrsg., *Springer Handbook of Robotics*. Springer International Publishing, 2016. Adresse: <https://doi.org/10.1007/978-3-319-32552-1>.
- [1.2] *ISO 8373:2021-11, Robotics – Vocabulary*, Norm, Nov. 2021.
- [1.3] K. Nonami, F. Kendoul, S. Suzuki, W. Wang und D. Nakazawa, *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer Science & Business Media, 2010.
- [1.4] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [1.5] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [1.6] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik und M. Munih, *Robotics (Intelligent Systems, Control and Automation: Science and Engineering)*. Springer, Dordrecht, 2010, Bd. 43.

2 Roboterkomponenten

Dieses Kapitel gibt einen Überblick über die wichtigsten Komponenten eines Robotersystems. Hierbei wird das Hauptaugenmerk auf die mechatronischen Komponenten gelegt. Neben Sensoren und Aktoren werden auch typische Getriebebaupformen erläutert. Abbildung 2.1 zeigt beispielhaft eine Schnittdarstellung eines Gelenkes des DLR Leichtbauroboters. Die einzelnen Teilkomponenten müssen gut aufeinander abgestimmt sein, um eine hohe Performanz des Gesamtsystems zu erreichen. In dem DLR-Leichtbauroboter z.B. kommen Harmonic-Drive-Getriebe und abtriebsseitige Drehmomentsensoren zum Einsatz. Neben der antriebsseitigen Positionsmessung der Rotorwelle wird auch die abtriebsseitige Gelenkwinkelposition mit einem Potentiometer gemessen.

2.1 Sensoren

Es ist grob zu unterscheiden zwischen propriozeptiven und exterozeptiven Sensoren. Propriozeptive Sensoren messen den internen Zustand des Roboters. Hierzu gehören, z.B.

- Gelenkwinkelsensoren, wie z.B. Absolut- und Inkrementalencoder,
- Gelenkgeschwindigkeitssensoren,
- Drehmomentsensoren,

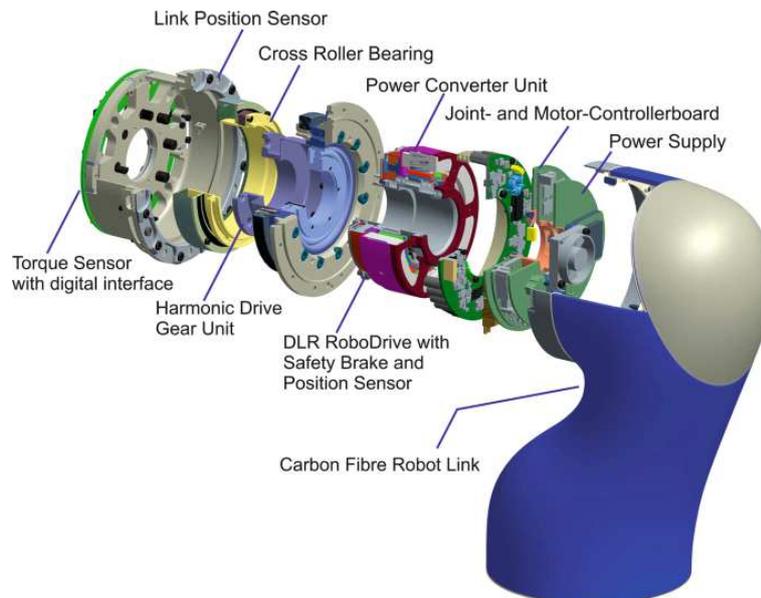


Abbildung 2.1: Schnittdarstellung eines Gelenkes des DLR Leichtbauroboters.

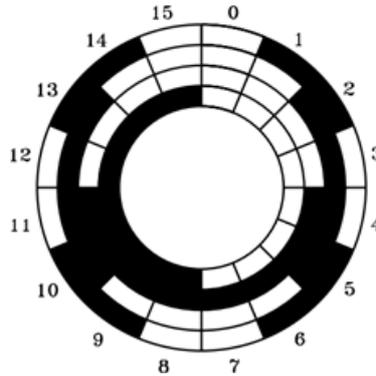


Abbildung 2.2: Prinzipskizze der Spuren eines Absolutencoders [2.1].

- Beschleunigungssensoren und inertielle Messinstrumente (IMU),
- Temperatursensoren.

Exterozeptive Sensoren messen Informationen der Umgebung. Das Gesamtbild aller Sensordaten liefert wichtige Informationen für die Aufgabenplanung und Autonomie. In diesem Sinne ermöglichen exterozeptive Sensoren die eigentliche intelligente Verbindung zwischen Perzeption und Aktion. Zu den exterozeptiven Sensoren gehören unter anderem:

- Abstandsensoren über Triangulation oder Laufzeit-Messungen,
- Kameras (Mono, Stereo),
- Kraft-Momentensensoren,
- Taktile Sensoren,
- Anwendungsbezogene Sensoren: z.B. Mikrofone, Feuchtigkeit, Rauch, Umgebungsdruck, Temperatur, etc...

2.1.1 Absolutencoder

Absolutencoder (auch „Absolutwertgeber“) liefern zu jeder Gelenkwinkelstellung ein eindeutiges Messsignal. Grundsätzlich können Absolutencoder auf optischen, magnetischen, oder induktiven Messprinzipien basieren. Optische Absolutencoder sind in unterschiedliche Spuren aufgeteilt. In den Spuren wechseln sich transparente und matte Felder in unterschiedlichen Mustern ab, so dass sich in jeder Position eine eindeutige Konfiguration ergibt (siehe Abbildung 2.2). Durch eine Anordnung der transparenten und matten Felder in der sich zwischen zwei Konfigurationen jeweils nur eines der Felder ändert, lässt sich hierbei eine hohe Fehlertoleranz erreichen. Die Anzahl der Spuren bestimmt die Messgenauigkeit. Typischerweise liefert ein Encoder mit n Spuren eine Auflösung von $360^\circ/2^n$.

Abbildung 2.3 zeigt ein magnetisches Wirkprinzip eines auf Hall-Sensoren basierenden Winkelsensors. Hallensoren liefern ein Messsignal proportional zur magnetischen Flussdichte. Durch Anordnung mehrerer Hall-Sensoren um einen sich drehenden Magneten lässt sich aus den Messwerten auf die Stellung des Magneten zurückschließen.

Absolutencoder werden typischerweise als abtriebsseitige Sensoren (d.h. nach dem Getriebe) eingesetzt. Bei Verwendung als motorseitige Sensoren (d.h. vor dem Getriebe) muss die Anzahl der Motorumdrehungen durch die Sensorelektronik mitgezählt werden.

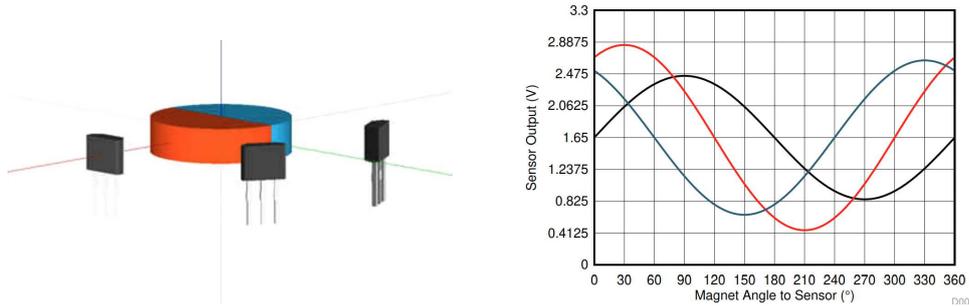


Abbildung 2.3: Prinzipskizze eines auf Hall-Sensoren basierenden Winkelsensors [2.2].

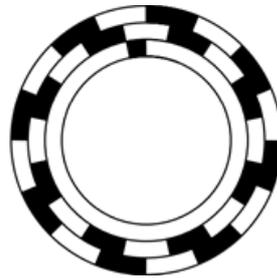


Abbildung 2.4: Prinzipskizze der Spuren eines Inkrementalencoders [2.1].

2.1.2 Inkrementalencoder

Inkrementalencoder sind einfacher aufgebaut und somit in der Regel auch kostengünstiger. Im Gegensatz zu Absolutencodern beinhalten Inkrementalencoder nur zwei Spuren in denen sich die Muster alle vier Schritte periodisch wiederholen (siehe Abbildung 2.2). Durch Verwendung von zwei Spuren ist es möglich auch die Drehrichtung eindeutig zu bestimmen:

$$\text{Vorwärts: } (0,0) - (1,0) - (1,1) - (0,1) - (0,0) \quad (2.1)$$

$$\text{Rückwärts: } (0,0) - (0,1) - (1,1) - (1,0) - (0,0) \quad (2.2)$$

Um aus dieser Musterabfolge die absolute Stellung des Gelenkes zu bestimmen, ist eine zusätzliche Elektronik erforderlich, welche die einzelnen Inkremente aufsummiert. Oftmals wird der Sensor durch eine dritte Spur mit einem einzigen matten Feld ergänzt. Durch diese Spur kann eine eindeutige Referenzposition bestimmt werden. In Robotergelenken werden Inkrementalencoder sowohl antriebsseitig wie auch abtriebsseitig eingesetzt.

Inkrementalencoder können auch zur Geschwindigkeitsmessung eingesetzt werden. Hierbei wird bei jeder Zustandsänderung im Messwert ein Puls generiert. Bei schnellen Drehgeschwindigkeiten lässt sich aus der Frequenz der Pulse die Geschwindigkeit ermitteln. Für langsame Geschwindigkeiten ist alternativ eine Messung der Zeiten zwischen zwei aufeinanderfolgenden Pulsen besser geeignet.

2.1.3 Drehmomentsensoren

Drehmomentsensoren in Robotergelenken werden in einigen kollaborativen Robotern eingesetzt, wie z.B. im KUKA LBR iiwa, Franka Panda, Neura Robotics Maira. Durch Messung des abtriebsseitigen Gelenkdrehmomentes, d.h. unabhängig von der Getriebereibung, kann in Folge eine sehr sensitive Kraft- und Interaktionsregelung erfolgen. Die Messung von Kräften und Drehmomenten erfolgt üblicherweise indirekt über die elastische Dehnung eines mechanischen Bauteils. Die Dehnung wird zum Beispiel über Dehnmessstreifen (DMS) gemessen. Dehnmessstreifen ändern ihren elektrischen Widerstand in Abhängigkeit von der Längenänderung. Die Auswertung der Widerstandsänderung erfolgt schließlich über eine Brückenschaltung (Wheatstone, siehe Abbildung 2.5).

$$V_o = \left(\frac{R_2}{R_1 + R_2} - \frac{R_s}{R_3 + R_s} \right) V_i \quad (2.3)$$

Abbildung 2.5 zeigt beispielhaft den Drehmomentsensor im DLR Leichtbauroboter III. Die einzelnen DMS sind hier an den Stegen eines verformbaren mechanischen Bauteils angebracht. In der mechanischen Auslegung des Roboters ist darauf zu achten, dass die Steifigkeit des Drehmomentsensors deutlich über der Getriebeelastizität liegt. Üblicherweise werden mehrere Dehnmessstreifen an unterschiedlichen Stellen eingesetzt, um die Sensitivität zu erhöhen und auch ein größeres Maß an Temperaturunempfindlichkeit zu erreichen.

2.1.4 Kraft/Momentensensoren (FTS)

6-Achs-Kraft/Momenten-Sensoren (engl. „force/torque-sensor“ (FTS)) erlauben die Messung von drei Kräften und drei Momenten. FTS werden üblicherweise im Handgelenk bzw. im Endeffektor des Roboters zur Messung der Prozesskräfte eingesetzt. Die Funktionsweise ist grundsätzlich ähnlich wie in einem Drehmomentsensor. Durch die wirkenden Kräfte und Momente wird eine Auslenkung in einem elastischen mechanischen Bauteil bewirkt und es werden die Dehnungen in den verschiedenen Raumrichtungen gemessen. Hierbei müssen hinreichend viele DMS benutzt werden, um alle translatorischen und rotatorischen Dehnungen messen zu können. Über eine Kalibriermatrix werden die einzelnen Messungen so kombiniert dass die wirkenden Kräfte und Momente im Sensorkoordinatensystem rekonstruiert werden können. Die gemessenen Kräfte und Momente müssen abschließend aus dem lokalen Koordinatensystem des Sensors in das Koordinatensystem am Endeffektor umgerechnet werden. Abbildung 2.6 zeigt eine Prinzipskizze zur Rekonstruktion der wirkenden Kräfte und Momente aus den gemessenen Dehnungen. Hierbei werden über DMS acht verschiedene Dehnungen an den Stegen zwischen dem inneren und dem äußeren Bauteil gemessen. Somit ergibt sich folgende Struktur der Kalibriermatrix:

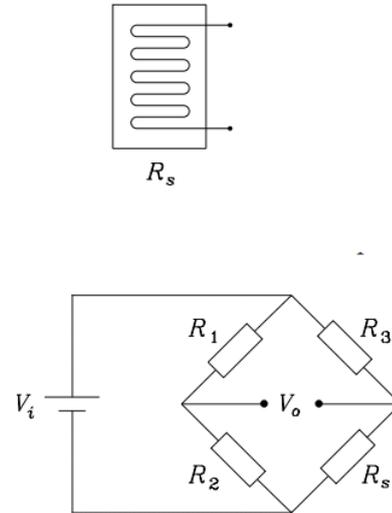


Abbildung 2.5: Links: Drehmomentsensor in den Gelenken des DLR Leichtbauroboters [2.3]. Durch eine äußere Krafteinwirkung kommt es zu einer Verdrehung zwischen dem inneren und äußeren Ring. Diese Verdrehung wird über die in den Stegen integrierten DMS erfasst. Rechts: Auswertung eines DMS über eine Wheatstone-Brückenschaltung [2.1].

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 & 0 & c_{15} & 0 & 0 & 0 \\ 0 & 0 & c_{23} & 0 & 0 & 0 & c_{27} & 0 \\ 0 & c_{32} & 0 & c_{34} & 0 & c_{36} & 0 & c_{38} \\ 0 & c_{42} & 0 & 0 & 0 & c_{46} & 0 & 0 \\ 0 & 0 & 0 & c_{54} & 0 & 0 & 0 & c_{58} \\ c_{61} & 0 & c_{63} & 0 & c_{65} & 0 & c_{67} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix} \quad (2.4)$$

Hierbei sind f_i und τ_i (für $i \in \{x, y, z\}$) die in den einzelnen Raumrichtungen wirkenden Kräfte und Momente, während w_i , wie in Abbildung 2.6 eingezeichnet, die Messwerte der einzelnen DMS darstellen. Die Parameter c_{ij} stellen die Einträge der Kalibriermatrix dar. Eine genaue Kalibrierung erfordert die messtechnische Ermittlung aller 48 Komponenten der Kalibriermatrix, während in (2.4) nur die Hauptterme als ungleich null eingetragen sind.

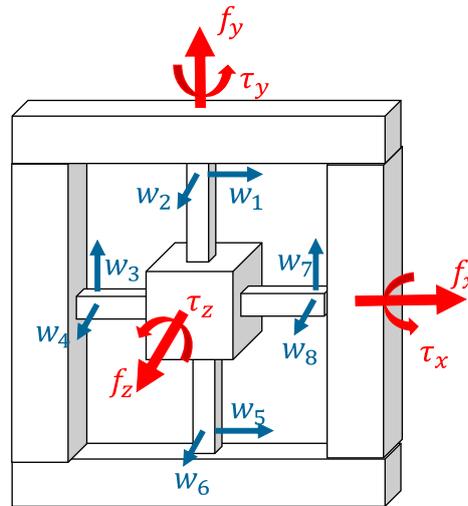


Abbildung 2.6: Prinzipskizze zur Rekonstruktion der wirkenden Kräfte f_i und Momente τ_i aus den gemessenen Dehnungen w_i .

2.1.5 Inertiale Messeinheiten (IMU)

Inertiale Messeinheiten (engl. „inertial measurement unit“ (IMU)) werden in mobilen Robotersystemen benutzt um die „Ego-Motion“, d.h. die eigene Bewegung des Roboters im Raum zu messen. Insbesondere für Flugroboter ist die IMU der zentrale Sensor zur Lagestabilisierung und Trajektorienfolgeregelung. Auch in humanoiden Robotern und mehrbeinigen Laufrobotern spielt die IMU eine wichtige Rolle und bildet ein mechatronisches Gegenstück zum Gleichgewichtssinn in Menschen und Tieren.

Eine IMU beinhaltet mehrere Teilkomponenten, insbesondere ein Gyroskop zur Messung der Winkelgeschwindigkeit und einen dreiachsigen translatorischen Beschleunigungssensor. Auch IMUs mit zusätzlicher Sensorik für das Erdmagnetfeld werden häufig eingesetzt. Viele IMUs sind bereits mit einer internen Elektronik ausgestattet, die aus diesen einzelnen Messwerten eine Schätzung der Lage und Orientierung der IMU im Raum ermittelt. Abbildung 2.7 zeigt die grundlegenden Berechnungsschritte einer IMU. Aus der Winkelgeschwindigkeit wird zunächst über eine nichtlineare Integration (Details hierzu finden Sie im folgenden Kapitel - „Starrkörperbewegungen“) die Orientierung ermittelt. Mittels dieser Information wird aus dem Beschleunigungssignal die Erdbeschleunigung abgezogen und durch zweimalige Integration eine Schätzung der translatorischen Geschwindigkeit und Position ermittelt. Hierbei ist anzumerken, dass diese Integration sehr sensitiv gegenüber Messfehlern in den einzelnen Teilsensoren ist. In der Praxis ergibt sich insbesondere in der Positionsbestimmung eine nicht unerhebliche Drift. Daher ist es üblich diese Messwerte mit zusätzlichen Sensorinformationen, wie z.B. GPS-Daten oder Messungen eines Magnetometers zu fusionieren. Weiters lässt sich aus dem Gleichanteil der Beschleunigungswerte eine Schätzung der vertikalen Richtung ermitteln. Hierdurch ist eine Reduktion der Drift um die Kippachsen möglich. Für diese Sensorfusion kommen überwiegend Kalman-Filter (siehe VU „Regelungssysteme“) zum Einsatz. Die Implementierungsdetails dieser Filter und die resultierenden Genauigkeitswerte bilden wichtige Unterscheidungsmerkmale für

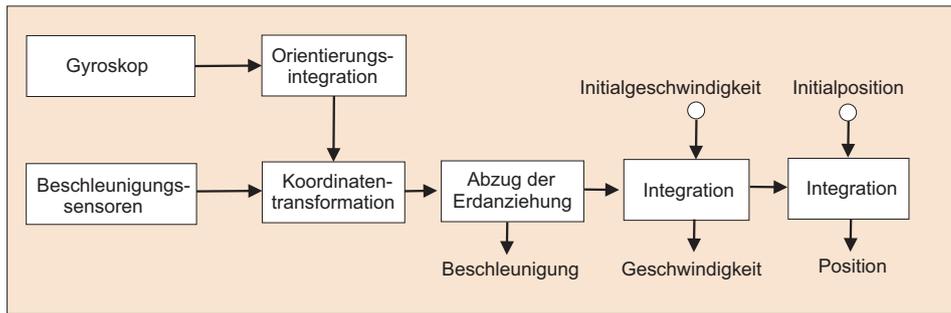


Abbildung 2.7: Grundlegende Berechnungsstruktur einer IMU.

die Auswahl einer kommerziellen IMU.

2.1.6 Abstandssensoren

Mittels Abstandssensoren werden wichtige Informationen über die dreidimensionale Umgebung des Roboters gesammelt. Somit erlauben diese Sensoren den Autonomiegrad des Robotersystems zu steigern und ermöglichen intelligente Funktionen wie die Hindernisvermeidung, das Aufbauen von Karten der Umgebung, die Navigation, oder das Erkennen von Objekten in der Umgebung. Abstandssensoren liefern eine „2.5-dimensionale“ Information über die dreidimensionale Umgebung in dem Sinne, dass zu jeder Richtung im Sichtbereich des Sensors der Abstand zum nächsten Objekt ermittelt wird. Somit „sieht“ der Roboter quasi nur eine Seite des Objektes. Neben Sensoren basierend auf Licht bzw. Laser sind auch Ultraschallsensoren (Sonar) als Beispiele zu nennen. Die zwei wichtigsten Messprinzipien in Abstandssensoren sind die Triangulation und die Laufzeit-Messung (engl. „Time-of-Flight“).

In der Triangulation (Abbildung 2.8) wird durch eine Laserdiode ein Laserstrahl in die Umgebung emittiert. Das reflektierte Laserlicht wird von einer Linse fokussiert und mittels CCD-Sensor detektiert. Hierbei lässt sich aus der Position des Laserstrahles auf dem CCD-Sensor der Abstand zum reflektierenden Objekt ermitteln. Dies erfolgt über eine trigonometrische Berechnung, wobei die Lage des CCD-Sensors relativ zur Laserdiode genau bekannt und kalibriert sein muss.

Abbildung 2.9 zeigt das Prinzip eines Time-of-Flight-Sensors. Hierbei wird aus der Laufzeit eines Laserpulses von der Quelle zum Objekt und zurück zum Detektor auf den Abstand zurückgeschlossen. Bei diesem Messprinzip müssen sehr kurze Zeitspannen gemessen werden. Um z.B. eine Genauigkeit von 1mm zu erreichen ist eine Genauigkeit in der Zeitmessung von 3ps erforderlich. Eine weitere Einschränkung dieses Messverfahrens ist durch das sogenannte Mehrdeutigkeitsintervall (engl. „ambiguity interval“) gegeben. Viele Time-of-flight-Sensoren senden periodisch Lichtpulse aus und ermitteln den Abstand aus einer Mittelung mehrerer Messungen innerhalb eines Zeitintervalls Δt . Allerdings können in diesem Zeitintervall auch frühere Pulse von einem weiter entfernten Objekt empfangen werden. Hieraus ergibt sich eine Uneindeutigkeit zu einem Vielfachen von $\frac{1}{2}c\Delta t$, wobei c die Lichtgeschwindigkeit darstellt. Typische Werte liegen hier im Bereich von $\frac{1}{2}c\Delta t = 20 - 40\text{m}$. Durch eine intelligente Auswerteelektronik kann dieses Problem

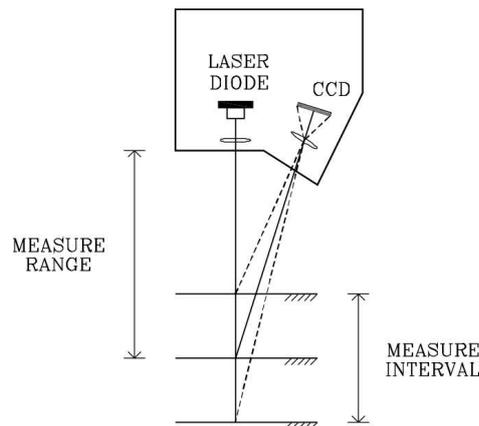


Abbildung 2.8: Prinzip der Triangulationsmessung [2.1].

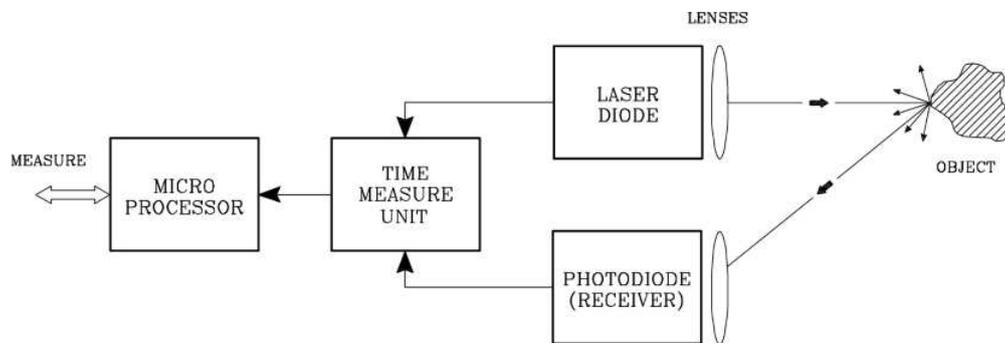


Abbildung 2.9: Prinzip der Time-of-Flight-Messung [2.1].

etwas entschärft werden unter der Annahme, dass die Änderung des Abstandes nicht sprunghaft, sondern nur graduell, erfolgt. Der Laserstrahl eines Time-of-Flight-Sensors kann über Spiegel in einer Linie oder in der Ebene geschwenkt werden. Somit lassen sich auch komplexere Szenen abbilden. Typische Time-of-Flight-Sensoren in der Robotik haben einen Arbeitsbereich von 5 – 100m, bei einer Genauigkeit von 5 – 10mm.

2.1.7 Kameras

Neben Tiefeninformationen aus Time-of-Flight-Sensoren stellen Mono- und Stereokameras die wohl wichtigsten Sensoren für die Autonomiefunktionen moderner Robotersysteme dar. Einen besonderen Stellenwert darin nimmt unter anderem das Thema der Kalibrierung der „intrinsischen“ und „extrinsischen“ Kameraparameter dar. Hierbei werden als intrinsische Parameter die Parameter der Kamera selbst bezeichnet, wie z.B. die Brennweite, Verzerrungskoeffizienten, Skalierungsfaktoren oder die Lage des optischen Bildzentrums. Hingegen bezeichnet man als extrinsische Parameter die Lageparameter der Kamera bezüglich eines Basiskoordinatensystems. Eine ausführliche Behandlung unterschiedlicher Kamera-Technologien geht jedoch über diese Vorlesung hinaus. Für eine detailliertere Beschreibung dieser Aspekte wird auf die VU „Machine Vision“ verwiesen.

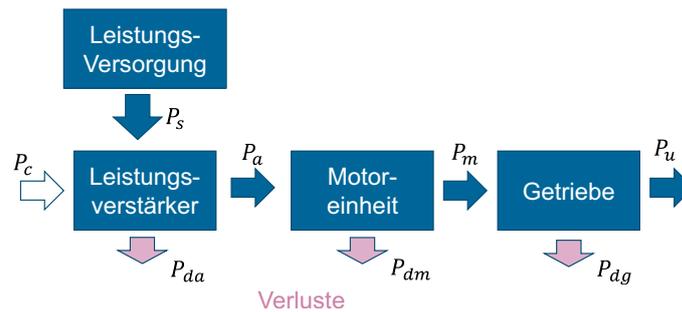


Abbildung 2.10: Hauptkomponenten in einem Roboterantrieb.

2.2 Antriebssystem

2.2.1 Überblick

In robotischen Systemen kommen je nach Anwendungsfall und Größenordnung eine Vielzahl unterschiedlicher Antriebe zum Einsatz. Neben elektrischen Motoren sind auch pneumatische und hydraulische Antriebe verbreitet. Im Allgemeinen besteht ein robotisches Antriebssystem aus folgenden Komponenten:

- Leistungsversorgung
- Leistungsverstärker
- Motoreinheit
- Getriebe

Abbildung 2.10 zeigt den Leistungsfluss zwischen diesen Komponenten. Hierbei bezeichnet P_u die in dem Gelenk erforderliche mechanische Leistung. Die Auslegung eines Antriebssystems startet üblicherweise von diesen Leistungsanforderungen. Die Leistungsversorgung stellt die nutzbare Eingangsleistung P_s zur Verfügung. In einem hydraulischen Antrieb z.B. wird die Leistungsversorgung durch das Ölreservoir zusammen mit der Pumpe zur Erzeugung des Systemdruckes repräsentiert.

In Abhängigkeit des Stellsignales P_c moduliert der Leistungsverstärker den Energiefluss P_a in die Motoreinheit. Im Falle von hydraulischen Antrieben nimmt das Servoventil zur Steuerung der Volumenströme in die Kammern eines Hydraulikzylinders die Rolle des Leistungsverstärkers ein. Die eigentliche Konvertierung der Eingangsleistung in mechanische Leistung P_m erfolgt durch die Motoreinheit. Getriebe stellen abschließend die Schnittstelle zwischen der durch den Motor erzeugten Bewegung und der (gewünschten) Bewegung des Robotergelenkes dar. In Abbildung 2.10 sind auch die wichtigsten Verlustleistungen P_{da} , P_{dm} , P_{dg} eingezeichnet, die sich unter anderem durch die Konvertierung zwischen unterschiedlichen Energieformen in der Antriebseinheit ergeben.

Die Abschnitte 2.2.2 und 2.3 geben einen Überblick über in der Robotik häufig eingesetzte Getriebe und Antriebe.

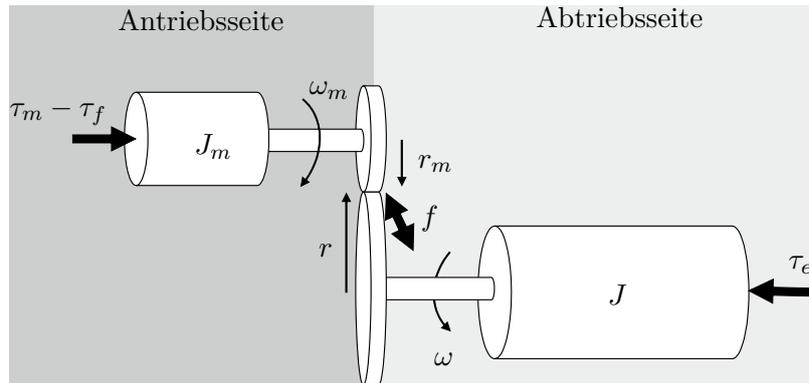


Abbildung 2.11: Skizze zu einem Antrieb mit Getriebe. Die Trägheit des Rotors J_m und die Trägheit der abtriebsseitigen Last J sind über ein Stirnradgetriebe verbunden. Durch die Getriebeübersetzung $i = r/r_m$ ergibt sich eine Skalierung zwischen den antriebsseitigen Größen (vor dem Getriebe) und den abtriebsseitigen Größen (hinter dem Getriebe).

2.2.2 Getriebe

Getriebe stellen die Schnittstelle zwischen der durch den Motor erzeugten Bewegung und der (gewünschten) Bewegung des Robotergelenkes dar. Hierbei erfüllt das Getriebe eine oder mehrere der folgenden drei Funktionen:

1. Skalierung der durch den Motor erzeugten Bewegung bzw. der wirkenden Kraft
2. Änderung der Bewegungsform zwischen Rotation und Translation
3. Überbrückung des räumlichen Abstandes zwischen Antrieb und Gelenk

Die in Industrierobotern eingesetzten elektrische Motoren sind üblicherweise darauf ausgelegt hohe Geschwindigkeiten (und somit geringe Drehmomente) zu erzeugen. Zur Verwendung in einem Roboterarm müssen die entsprechenden Drehmomente über ein Getriebe in den gewünschten Arbeitsbereich gebracht werden. Abbildung 2.11 zeigt beispielhaft einen einzelnen Antrieb mit Motor, Stirnradgetriebe, und Abtrieb. Für die folgende Betrachtung wird das Getriebe als ideal verlustfrei betrachtet. Zwischen der Winkelgeschwindigkeit auf Motorseite ω_m und im Abtrieb ω besteht der folgende Zusammenhang

$$\omega_m r_m = \omega r . \quad (2.5)$$

Der Faktor $i = r/r_m$ beschreibt die Getriebeübersetzung. In einem idealen verlustlosen Getriebe und ohne Einfluss von Reibung $\tau_f = 0$ wird die Antriebsleistung $P_m = \tau_m \omega_m$ direkt auf den Abtrieb übertragen, d.h. $P = \tau \omega = P_m$. Somit erhält man das am Abtrieb wirkende Drehmoment zu $\tau = \tau_m i$. Zwischen den an- und abtriebsseitigen Größen bestehen also die folgenden Zusammenhänge:

$$\omega = \omega_m / i \quad (2.6)$$

$$\tau = \tau_m i \quad (2.7)$$

Die Kraftübertragung zwischen An- und Abtrieb erfolgt in diesem Fall über die im Flankenkontakt der Sirnräder wirkende Kraft f , so dass das Kräftegleichgewicht für die An- und Abtriebsseite gemäß

$$J_m \dot{\omega}_m = \tau_m - \tau_f - r_m f \quad (2.8)$$

$$J \dot{\omega} = \tau_e + r f \quad (2.9)$$

angeschrieben werden kann. Durch Elimination der Kontaktkraft f ergibt sich die Gesamtdynamik in abtriebsseitigen Größen als

$$(J + J_m i^2) \dot{\omega} = \tau - i \tau_f + \tau_e . \quad (2.10)$$

Hier erkennt man, dass die Rotorträgheit quadratisch mit der Getriebeübersetzung verstärkt wird. Reibung und Störeffekte auf der Motorseite werden ebenso mit der Getriebeübersetzung verstärkt. Weiters erkennt man, dass für sehr große Getriebeübersetzungen die Gesamtdynamik von der Motordynamik (und der Getriebereibung) dominiert wird, während abtriebsseitige Terme, wie z.B. Trägheitskopplungen der nichtlinearen Mehrkörperdynamik (siehe Kapitel 6 „Dynamik“) für die Positionsregelung eine kleinere Rolle spielen.

Neben der Getriebeübersetzung und dem Nennmoment [Nm] sind insbesondere der Wirkungsgrad [%], das Spiel [arcmin] und die Torsionssteifigkeit [Nm/arcmin] wichtige Getriebeparameter. Letztere werden oftmals zu einem gemeinsamen Parameter, dem sogenannten „Positionsfehler“ (eng. „lost motion“) subsummiert:

$$\text{lost motion} = \text{Spiel} + \text{Nennmoment} / \text{Torsionssteifigkeit} . \quad (2.11)$$

Im Folgenden werden einige in der Robotik weit verbreitete Getriebearten und deren Wirkprinzipien aufgelistet:

Gewindespindel (engl. lead screw), werden benutzt um die Drehbewegung eines Motors in eine translatorische Bewegung umzuwandeln. In Kugelumlaufspindeln (engl. ball screw) werden Kugeln als Wälzkörper in Laufrillen zwischen Spindel und Mutter eingesetzt um eine reibungsarme Bewegungsübertragung zu ermöglichen. Die Kugeln wandern axial und werden in der Spindelmutter über Rückfuhrkanäle zurückgeleitet, siehe Abbildung 2.12. Aufgrund der Kugellagerung besitzen Kugelumlaufspindel eine deutlich geringere Reibung (Rollreibung). Es kommt zu einer geringeren Wärmeentwicklung, so dass letztendlich auch höhere Drehzahlen möglich sind und Wirkungsgrade im Bereich von 90% und höher erreicht werden können. Außerdem weisen Kugelumlaufspindel, im Gegensatz zu normalen Gewindespindeln, keine Selbsthemmung auf (siehe Abbildung 2.13).

Zahnriemen, Ketten, Schubstangen, und Seile werden benutzt um einen räumlichen Versatz zwischen der Motorachse und dem Robotergelenk zu überbrücken, siehe Beispiele in Abbildung 2.14.



Abbildung 2.12: Beispiel für den Aufbau einer Kugelumlaufspindel (Quelle: www.omc-stepperonline.com)

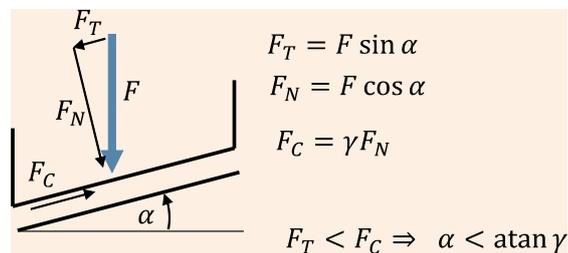


Abbildung 2.13: Skizze zum Effekt der Selbsthemmung in Gewindespindeln: Eine wirkende Kraft F in Richtung der Spindelachse kann mit der Gewindesteigung α in die Komponenten F_N und F_R normal und entlang zum Gewinde aufgeteilt werden. Die Haftreibung $F_C = \gamma F_N$ ergibt sich aus dem Reibkoeffizienten γ und F_N . Selbsthemmung entsteht bei $F_T < F_C$, und somit für $\alpha < \arctan(\gamma)$.

Planetengetriebe sind Zahnradgetriebe in denen einzelne Zahnräder auf beweglichen Achsen kreisförmig um die Antriebsachse umlaufen. Die umlaufenden Zahnräder werden auch Planetenräder genannt, während die Zahnräder mit feststehenden Achsen „Zentralräder“ oder „Sonnenräder“ genannt werden. Planetengetriebe weisen eine kompakte Bauweise auf, erlauben die Übertragung hoher Drehmomente, und haben im Allgemeinen sehr hohe Torsionssteifigkeiten. Ein Nachteil ist das üblicherweise relativ große Spiel, welches sich aus dem Zahneingriff ergibt. Zur Realisierung höherer Übersetzungen werden oft auch mehrstufige Planetengetriebe eingesetzt.

Wellengetriebe (nach dem Hersteller auch HD, „Harmonic-Drive-Getriebe“ genannt) bestehen aus drei Komponenten: Wave Generator (WG), Flexspline (FS), und Circular Spline (CS), siehe Abbildung 2.16. Der Wave Generator ist ein elliptisches Bauteil das starr mit dem Rotor verbunden wird. Um den Wave Generator befindet sich ein Wälzlager mit verformbaren Laufringen. Das Flexspline bezeichnet eine verformbare Stahlbüchse mit Außenverzahnung. Das Circular Spline ist ein Außenring mit Innenverzahnung. Hierbei besitzt das Circular Spline mehr Zähne als das Flexspline. Bei Drehung des Wave Generators greifen nun die Zahnflanken zwischen

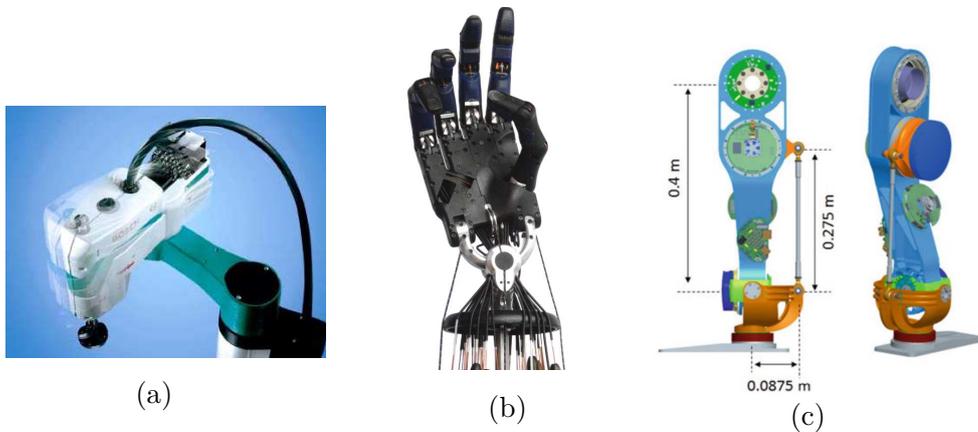


Abbildung 2.14: Beispiele für Roboter mit Antrieben abseits der Gelenkachse. (a) Im SCARA Bosch turboscaras SR6 befinden sich zwei Antriebe mit Harmonic-Drive-Getrieben in der Roboterbasis. Das zweite Gelenk wird über einen Riemen mit der Antriebseinheit gekoppelt. (b) Die Roboterhand der Firma „Shadow Robotics“ verwendet Seile zur Verbindung der Fingergelenke mit den Antrieben im Unterarm. (c) Im Unterschenkel des humanoiden Roboters TORO (DLR) wird eine Schubstange verwendet um den Antrieb mit dem Fußgelenk zu verbinden.



Abbildung 2.15: Konzeptskizze eines Planetengetriebes bestehend aus einem zentralen Sonnenrad (gelb), vier Planetenrädern (blau) und einem Hohlrad (rot) (Wikipedia). Die Umlaufbewegung der Planetenräder wird an den Steg (grün) abgegeben. Grundsätzlich kann entweder das Sonnenrad, das Hohlrad, oder der Steg feststehend sein, während sich die anderen beiden Komponenten mit unterschiedlichen Drehgeschwindigkeiten gegeneinander bewegen.

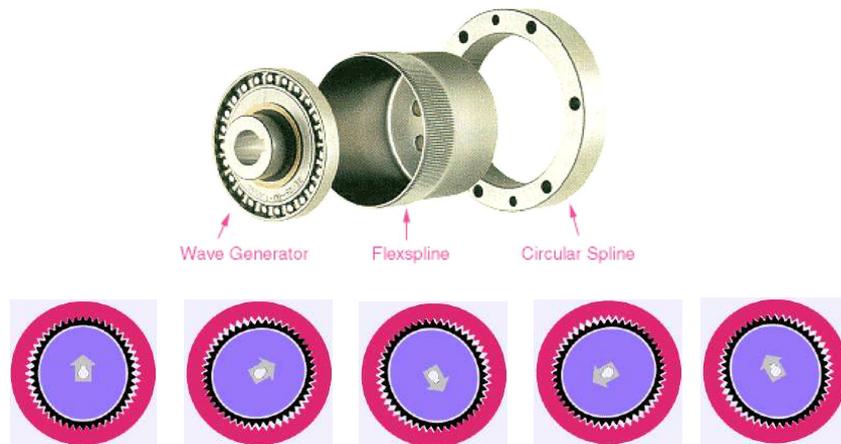


Abbildung 2.16: Konzeptskizze eines Wellengetriebes (Harmonic Drive).

Circular Spline und Flexspline nach und nach ineinander. Wenn das Circular Spline, z.B. zwei Zähne mehr als das Flexspline besitzt, so ergibt sich bei jeder vollen Umdrehung des Rotors eine Relativbewegung zwischen Circular Spline und Flexspline um zwei Zähne. Bei z.B. $n = 200$ Zähnen der Innenverzahnung und $m = 198$ Zähnen im Flexspline erhält man ein Übersetzungsverhältnis von $i = m/(n - m) = 198/2 = 99$. Wellengetriebe zeichnen sich durch besonders hohe Übersetzungen und geringes Spiel aus, besitzen aufgrund der Nachgiebigkeit im Flexspline aber eine nicht unerhebliche Elastizität. Wellengetriebe sind nicht selbsthemmend, erfordern jedoch aufgrund der hohen Übersetzung ein sehr großes Rücktreibmoment.

Zykloidgetriebe sind Exzentergetriebe ohne Zahnräder. Abbildung 2.17 zeigt eine Prinzipskizze eines Zykloidgetriebes mit den folgenden Komponenten: Exzenter (grün), Kurvenscheibe (gelb), Bolzenring (grau), Abtrieb (violett). Zunächst treibt der Rotor den innen liegenden Exzenter an. Dadurch wälzt die zykloidenförmige Kurvenscheibe an dem außen befindlichen feststehenden Bolzenring ab. Die sich daraus ergebende Rotationsbewegung der Kurvenscheibe um die Getriebeachse wird über die violett eingezeichneten Bolzen an den Abtrieb übertragen. Das Übersetzungsverhältnis ergibt sich aus der Anzahl N der Bolzen und der Anzahl n der Nocken in der Kurvenscheibe zu $i = n/(N - n)$. Verglichen mit Planetengetrieben weisen Zykloidgetriebe eine größere Kontaktfläche auf. Zykloidgetriebe sind üblicherweise relativ robust gegenüber externen Stößen, weisen geringes Spiel und eine hohe Torsionssteifigkeit auf. Zykloidgetriebe sind nicht selbsthemmend und eignen sich sehr gut für hohe Übersetzungen. Für sehr kleine Übersetzungen (<20) weisen Planetengetriebe jedoch üblicherweise einen höheren Wirkungsgrad auf.

Unter einem **Direktantrieb** (engl. *direct drive*) versteht man einen Antrieb in dem ein Motor direkt mit dem Abtrieb verbunden ist, d.h. komplett ohne Getriebe, Zahnriemen, oder Kette. Direktantriebe sind sehr energieeffizient in der Leistungsübertragung und erlauben eine gute Steuerung des Drehmomentes über den Motorstrom. Allerdings erfordert die Generierung großer Drehmomente direkt aus dem Motor ohne zusätzliches Getriebe

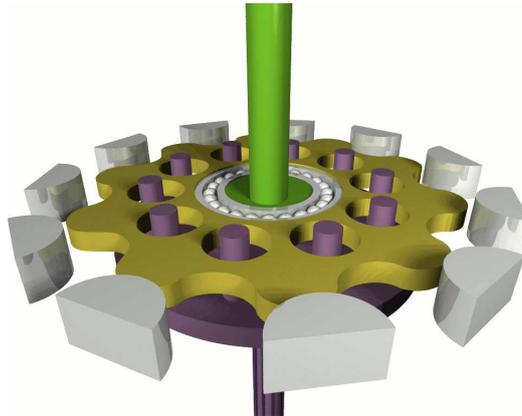


Abbildung 2.17: Konzeptskizze eines Zykloidgetriebes (Wikipedia).

besondere Bauformen und führt im Allgemeinen zu relativ schweren Antrieben.

2.3 Antriebsarten

Die Anforderungen an robotische Antriebe zeichnen sich durch folgende Charakteristika aus:

- Geringe Trägheit
- Gutes Verhältnis von Leistung zu Gewicht
- Überlastfähigkeit
- Erzeugung kurzfristig sehr hoher Drehmomente
- Hohe Beschleunigungen und Drehmomente
- Großer Geschwindigkeitsbereich (0-10 rad/s)
- Hohe Positioniergenauigkeit
- Geringe Drehmomentschwankungen auch bei kleinen Geschwindigkeiten

Die in der Robotik üblicherweise eingesetzten Antriebsarten lassen sich in folgende Klassen unterteilen:

- Hydraulische Antriebe
- Pneumatische Antriebe
- Elektrische Antriebe
- Neuartige Aktuatoren (werden hier nicht im Detail behandelt)
 - Piezoelektrische Aktuatoren
 - Shape-Memory Alloy
 - Elektroaktive Polymere
 - Twisted String Actuators (TSA)
 - etc...

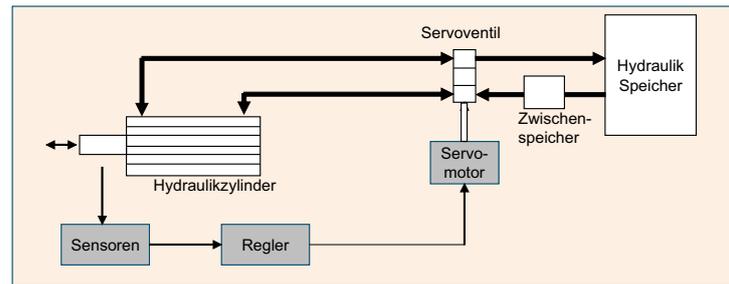


Abbildung 2.18: Grundsätzlicher Aufbau hydraulischer Antriebe. Über einen Servomotor wird ein Servoventil gesteuert um den Volumenstrom in einem Hydraulikzylinder einzustellen.

2.3.1 Hydraulische Antriebe

In hydraulischen Antrieben wird über eine (elektrisch angesteuerte) Pumpe die Energie in einem hydraulischen (Öl-)reservoir in mechanische Leistung umgewandelt. Der Druckbereich des Fluids liegt im Bereich von 100 bis 200bar.

Hydraulische Antriebe sind sehr gut für hohe Lasten geeignet und besitzen ein ausgezeichnetes Verhältnis von Leistung zu Gewicht. Sie sind auch robust gegenüber externen Stößen und erlauben auch ohne zusätzliches Getriebe große Drehmomente zu erzeugen. Im Gegensatz zu elektrischen Antrieben kommt es bei hydraulischen Antrieben zu keiner statischen Überhitzung. Ein Nachteil von hydraulischen Antrieben ist die recht aufwändige Systeminfrastruktur mit Ölreservoir, Pumpe und Hydraulikleitungen. Auch die Leckage stellt ein praktisches Problem dar, das den Anwendungsbereich hydraulischer Antriebe einschränkt. Schließlich ist das Systemverhalten, insbesondere die Viskosität des Öls, temperaturabhängig. Hydraulische Antriebe werden üblicherweise in sehr starken Robotergelenken eingesetzt, wie sie z.b. bei Großraummanipulatoren vorkommen.

Abbildung 2.18 zeigt eine schematische Darstellung von einem hydraulischen Antrieb. Die Volumenströme aus dem Speicher in die Kammern des Hydraulikzylinders werden über ein Servoventil gesteuert. Die Ansteuerung des Servoventils erfolgt über einen elektrischen Motor. Über die Drücke p_A und p_B in den beiden Kammern wird eine Kraft F auf den Kolben ausgeübt

$$F = A_k p_A - \alpha A_k p_B, \quad (2.12)$$

wobei A_k und αA_k die effektiven Kolbenflächen in Kammer A und B darstellen, siehe Abbildung 2.19.

Weiters wirken die Reibungskraft F_R und die externe Kraft F_{ext} auf den Kolben. Mit m für die Kolbenmasse und x als Position ergibt sich die Dynamik des Kolbens zu

$$m\ddot{x} = A_k(p_A - \alpha p_B) - F_R - F_{ext}, \quad (2.13)$$

Die beiden Drücke p_A und p_B sind hier weitere Zustandsgrößen, deren Dynamik über die Summe der Massenflüsse in den Kammern ermittelt werden kann. Die Volumenströme in die beiden Kammern werden mit Q_A und Q_B bezeichnet. Die Leckage Q_L zwischen

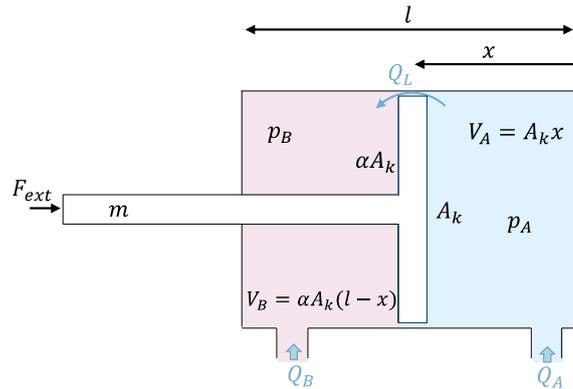


Abbildung 2.19: Skizze eines Hydraulikzylinders.

den Kammern wird unter Modellannahme einer laminaren Strömung proportional zur Druckdifferenz mit einem Systemparameter K_L angesetzt:

$$Q_L = K_L(p_A - p_B) . \quad (2.14)$$

Weiters seien $V_A = A_k x$ und $V_B = \alpha A_k (l - x)$ die Volumina der beiden Kammern. Die Änderung der Volumina ergibt sich für die beiden Kammern zu

$$\dot{V}_A = Q_A - Q_L - Q_{cA} , \quad (2.15)$$

$$\dot{V}_B = Q_B + Q_L - Q_{cB} , \quad (2.16)$$

Über die Terme $Q_{cA} = \gamma V_A \dot{p}_A$ und $Q_{cB} = \gamma V_B \dot{p}_B$ mit dem Kompressionskoeffizienten γ wird die Kompressibilität des Öls modelliert. Somit ergeben sich die Dynamikgleichungen für die Kammerdrücke zu

$$\dot{p}_A = 1/(\gamma V_A)(Q_A - K_L(p_A - p_B) - A_k \dot{x}) , \quad (2.17)$$

$$\dot{p}_B = 1/(\gamma V_B)(Q_B - K_L(p_B - p_A) + \alpha A_k \dot{x}) . \quad (2.18)$$

Die Gesamtdynamik besteht aus der mechanischen Gleichung (2.13) und der Druckdynamik (2.17)-(2.18) erster Ordnung. Weiters erkennt man den Einfluss der Reibungskomponente F_R in (2.13), welche über die Stellgrößen Q_A und Q_B nicht direkt kompensiert werden kann.

2.3.2 Pneumatische Antriebe

Pneumatische Antriebe nutzen Druckluft aus einem Kompressor und wandeln sie in mechanische Leistung um. Gegenüber hydraulischen Antrieben weisen pneumatische Antriebe den Vorteil auf, dass Druckluft relativ leicht gespeichert und übertragen werden kann. Der Systemdruck ist um Größenordnungen geringer als in hydraulischen Antrieben. Pneumatische Antriebe haben wenige bewegliche Teile und üblicherweise ein geringes Gewicht. Jedoch stellt sich eine genaue Positionierung aufgrund der Nachgiebigkeit der Druckluft oft als schwierig heraus. Typische Einsatzbereiche in der Robotik findet man als Aktuator für Greifer, als künstliche Muskel (PAM), oder in neuartigen weichen Robotern (engl. „soft robotics“), siehe Abbildung 2.20.

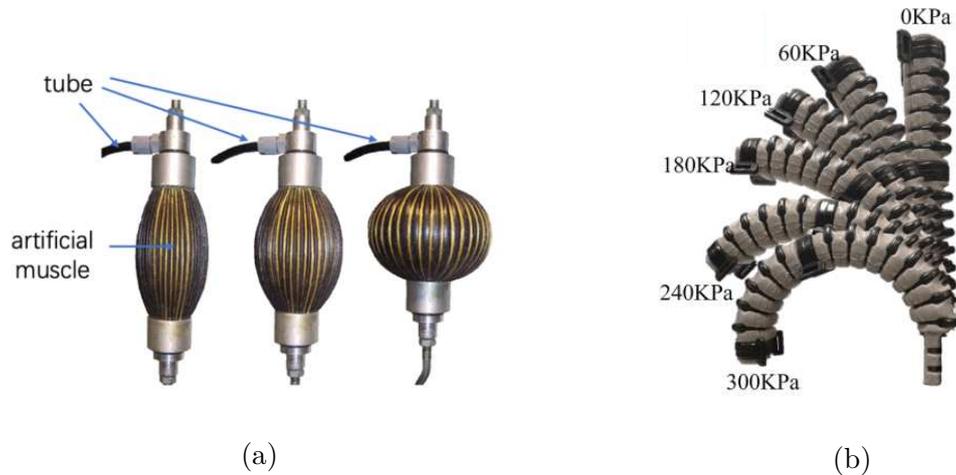


Abbildung 2.20: Beispiele für pneumatische Aktuatoren: (a) Künstlicher Muskel, (b) Beispiel für einen weichen Roboter mit pneumatischer Aktuierung [2.4]

2.3.3 Elektrische Antriebe

Die wichtigsten Typen von elektrischen Motoren in der Robotik umfassen bürstenlose und bürstenbehaftete Gleichstrommotoren, sowie Permanentmagnet-Synchronmotoren. Als „Servomotor“ bezeichnet man einen elektrischen Motor mit integrierter Sensorik zur Messung des Antriebswinkels. In Kombination mit einer Regeleinheit spricht man auch von einem Servoantrieb. Schrittmotoren bestehen aus einer Rotorwelle mit Permanentmagnet die sich in einem schrittweise rotierenden elektrischen Feld der Statorspulen bewegt. Durch Änderung der an den Statorspulen anliegenden Spannung bewegt sich der Rotor schrittweise weiter. Schrittmotoren können ohne zusätzliche Positionssensorik betrieben werden, jedoch kann es bei großen wirkenden Drehmomenten zu Schrittverlusten kommen.

Für eine umfassende Übersicht elektrischer Maschinen wird auf die VU „Maschinen und Antriebe“ verwiesen. Stellvertretend für die große Klasse elektrischer Antriebe, wird hier ein Gleichstrommotor mit beweglichem Rotor und feststehendem Stator betrachtet. Der Stator generiert über Permanentmagnete ein magnetisches Feld in dem sich der Rotor bewegt. Der Rotor trägt mehrere Wicklungen, welche in Abhängigkeit der relativen Verdrehung des Rotors zum Stator abwechselnd bestromt werden. Abbildung 2.21 zeigt beispielhaft den Aufbau eines Gleichstrommotors mit Encoder und integriertem Planetengetriebe.

Das im Gleichstrommotor auf den Rotor wirkende Drehmoment τ_m ist proportional zum Strom I , d.h. es gilt

$$\tau_m = k_t I . \quad (2.19)$$

Der Proportionalfaktor k_t [Nm/A] wird auch als *Drehmomentkonstante* bezeichnet und hängt vom detaillierten Aufbau des Motors ab. Der elektrische Widerstand R in den Wicklungen führt zu ohmschen Verlusten $P_\Omega = RI^2$ und somit zu Wärmeentwicklung. Um eine Überhitzung des Motors zu vermeiden muss daher der dauerhaft fließende Strom - und somit das Motordrehmoment - limitiert werden.

Abbildung 2.22 zeigt ein einfaches elektrisches Ersatzschaltbild eines Gleichstrommotors. Der elektrische Kreis ist durch den Widerstand R und die Induktivität L der Wicklung,

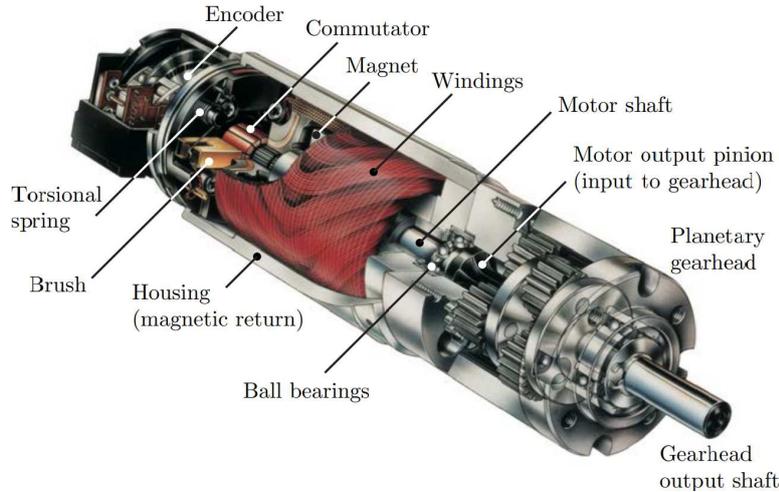


Abbildung 2.21: Bürstenbehalteter Gleichstrommotor, inklusive Encoder und Getriebe, der Firma Maxxon [2.5].

die Eingangsspannung U und die induzierte elektromotorische Kraft $U_{emk} = k_t \omega_m$ definiert. Hierbei stellt ω_m die Winkelgeschwindigkeit des Rotors dar. Diese Rückwirkung der Rotorgeschwindigkeit auf den elektrischen Kreis über U_{emk} stellt zusammen mit (2.19) die Verbindung zwischen der elektrischen Motordynamik und der mechanischen Roboterdynamik aus Kapitel 6 dar. Aus diesem Ersatzschaltbild ergibt sich die Motordynamik zu

$$U = U_{emk} + RI + L\dot{I} . \quad (2.20)$$

Wird nun der stationäre Zustand betrachtet, d.h. $\dot{I} = 0$, so erhält man den folgenden Zusammenhang zwischen der Rotorgeschwindigkeit und dem Drehmoment

$$\omega_m = \frac{U}{k_t} - \frac{R}{k_t^2} \tau_m . \quad (2.21)$$

Bei konstanter Eingangsspannung U ergibt sich die Winkelgeschwindigkeit ω_m als affine Funktion des Motordrehmomentes (mit einer negativen Steigung von $-R/k_t^2$). Betrachtet man nun die Maximalwerte der Eingangsspannung V_{max} und des Motorstromes I_{max} aufgrund der Leistungselektronik und Spannungsversorgung, so ergibt sich ein Arbeitsbereich wie in Abbildung 2.23 gezeigt. Im ersten und dritten Quadranten liefert der Motor eine positive Ausgangsleistung $P_m = \omega_m \tau_m$, während im zweiten und vierten Quadranten der Motor mechanische Leistung verbraucht und dieser somit als Generator fungiert $P_m < 0$. Die Geschwindigkeits-Drehmoment-Kurve wird im ersten Quadranten von zwei wesentlichen Punkten begrenzt. Die *Nominalgeschwindigkeit* $\omega_0 = V_{max}/k_t$ beschreibt die Geschwindigkeit des Motors im lastfreien Betrieb $\tau_m = 0$. In diesem Fall ist die induzierte elektromotorische Kraft U_{emk} gleich der Eingangsspannung. Das *Standmoment* (engl. „stall torque“) $\tau_{stall} = k_t V_{max}/R$ ist das maximale Drehmoment im eingespannten Zustand und somit für $U_{emk} = 0$. Der dunkel eingezeichnete Bereich in Abbildung 2.23 zeigt den dauerhaften Arbeitsbereich in dem das Motordrehmoment durch das dauerhaft mögliche Drehmoment τ_{cont} beschränkt ist. In diesem Bereich kann

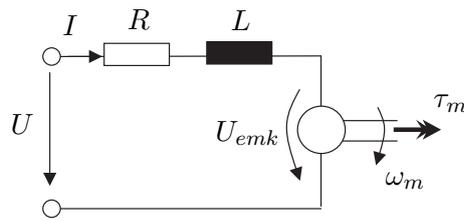
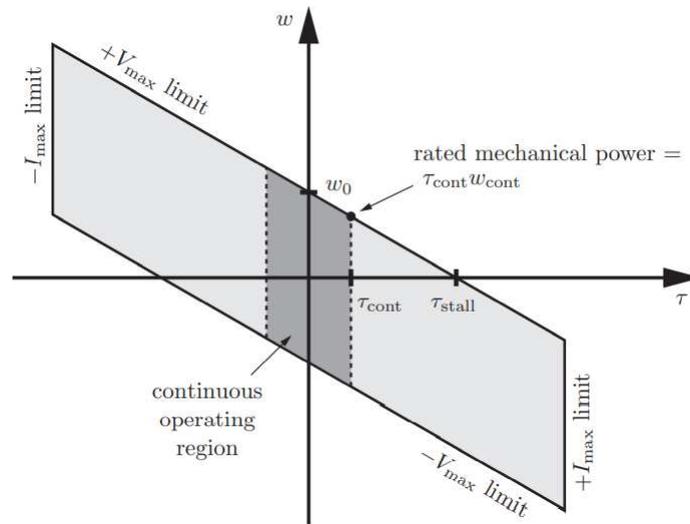


Abbildung 2.22: Einfaches Ersatzschaltbild eines permanentenerregten Gleichstrommotors.

Abbildung 2.23: Arbeitsbereich eines Gleichstrommotors unter Begrenzungen der maximalen Eingangsspannung V_{max} und des maximalen Motorstromes I_{max} . [2.5]

der Motor dauerhaft betrieben werden ohne zu überhitzen. Die nominelle Leistung des Motors wird durch den Wert $P_n = \tau_{cont}\omega_{cont}$ angegeben, wobei ω_{cont} der Wert der Geschwindigkeits-Drehmomenten-Kurve bei τ_{cont} ist.

Da Gleichstrommotoren üblicherweise für hohe Geschwindigkeiten und kleine Drehmomente ausgelegt sind, muss der Arbeitsbereich über ein Getriebe angepasst werden. Abbildung 2.24 zeigt skizzenhaft die Auswirkung eines (verlustlosen) Getriebes auf den Arbeitsbereich. Eine Erhöhung des Drehmomentes ist mit einer proportionalen Verringerung der Geschwindigkeit verbunden.

2.3.4 Drehmomentgeregelte Antriebe

Mit der Verbreitung von kollaborativen Robotern, wie dem KUKA LBR iiwa oder Franka Panda, gewinnen drehmomentgeregelte Robotergelenke eine immer größere Bedeutung. Abbildung 2.25 zeigt den typischen Aufbau eines Robotergelenkes mit Drehmomentsensor. Zur besseren Unterscheidung von antriebsseitigen (vor dem Getriebe) und abtriebsseitigen Größen (nach dem Getriebe) werden antriebsseitige Größen in diesem Abschnitt durch $(\cdot)'$ gekennzeichnet. Gemessen werden die antriebsseitige Gelenkwinkelposition θ' und das

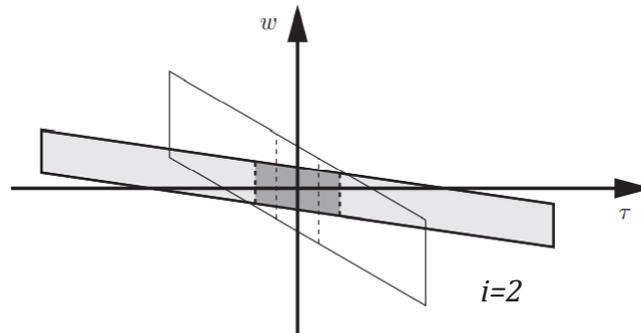


Abbildung 2.24: Skizze zur Veränderung des Arbeitsbereichs des Motors durch ein Getriebe mit Übersetzung $i = 2$. Eine Erhöhung des Drehmomentes bewirkt eine proportionale Verringerung der Winkelgeschwindigkeit. [2.5]

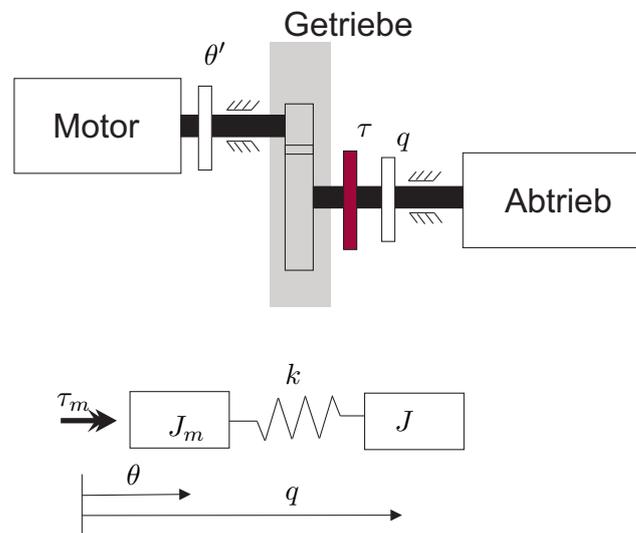


Abbildung 2.25: Drehmomenteregelter Antrieb und dessen translatorisches Ersatzmodell.

im Abtrieb wirkende Gelenkdrehmoment τ . Oftmals wird außerdem die abtriebsseitige Position q gemessen. Zunächst werden der Rotorwinkel θ' und das Motordrehmoment τ'_m über die Getriebeübersetzung i auf die Abtriebsseite umgerechnet, d.h. $\theta = \theta'/i$ und $\tau_m = \tau'_m i$. In einem idealen Drehmomentsensor liegt die Sensorsteifigkeit um mindestens eine Größenordnung über der Getriebebesteifigkeit k und kann somit vernachlässigt werden. Zwischen dem Gelenkdrehmoment τ und der Getriebebesteifigkeit k besteht dann der Zusammenhang

$$\tau = k(\theta - q) . \quad (2.22)$$

Die Bewegungsgleichungen zu dem in Abbildung 2.25 gezeigten Ersatzmodell mit der reflektierten Motorträgheit $J_m = J'_m i^2$ und der Abtriebsträgheit J ergeben sich zu

$$J\ddot{q} = k(\theta - q) \quad (2.23)$$

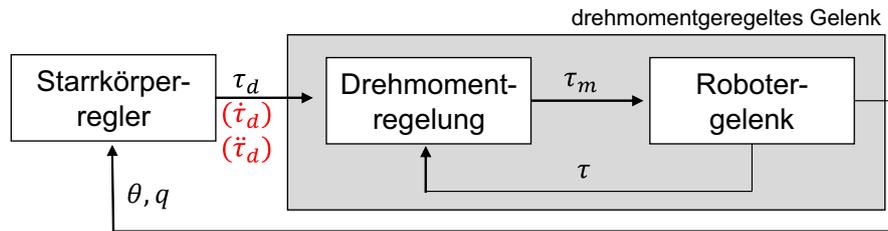


Abbildung 2.26: Kaskadierte Regulationsstruktur in Robotern mit drehmomentgeregelten Antrieben.

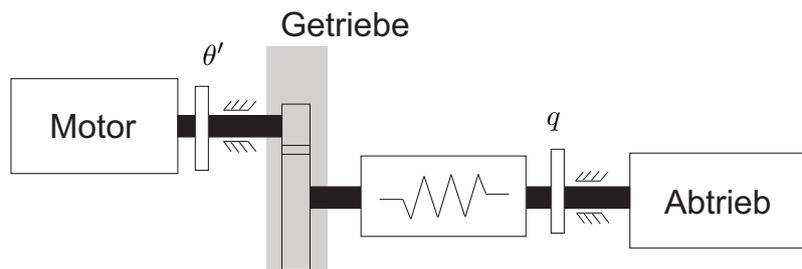


Abbildung 2.27: Prinzipieller Aufbau eines SEA.

$$J_m \ddot{\theta} + k(\theta - q) = \tau_m . \quad (2.24)$$

In einem drehmomentgeregelten Antrieb wird nun das Gelenkdrehmoment τ als Ausgangsgröße eines inneren Regelkreises betrachtet. Die Sollgröße τ_d für diesen inneren Regelkreis wird durch eine äußere Regelung für das Starrkörpermodell des Gesamtroboters ermittelt. Dies führt auf eine kaskadierte Regulationsstruktur wie sie in Abb. 2.26 gezeigt ist.

2.3.5 Series Elastic Actuators (SEA)

In einem seriell-elastischen Aktuator (engl. „Series Elastic Actuator“ (SEA)) wird ein elastisches Element gezielt in den Antriebsstrang integriert (siehe Abbildung 2.27). SEA werden oft als drehmomentgeregelte Antriebe eingesetzt. Durch Messung der Auslenkung des elastischen Bauteils wird auf das im Antrieb wirkende Gelenkdrehmoment zurück geschlossen und dieses kann über eine Regelung eingestellt werden. Die Dynamikgleichungen eines SEA haben grundsätzlich die selbe Struktur wie in (2.23)-(2.24), jedoch stammt in einem SEA die dominierende Systemsteifigkeit nicht aus dem Getriebe sondern von dem eingebrachten elastischen Bauteil. Eine weitere Motivation zur Einbringung von Elastizität in den Antriebsstrang ist die Möglichkeit das Getriebe über das elastische Bauteil gegenüber impulshaften externen Stößen zu schützen.

Eine ausführliche Behandlung zur Modellbildung und Regelung elastischer Roboter findet sich in der VU „Manipulation und Lokomotion“.

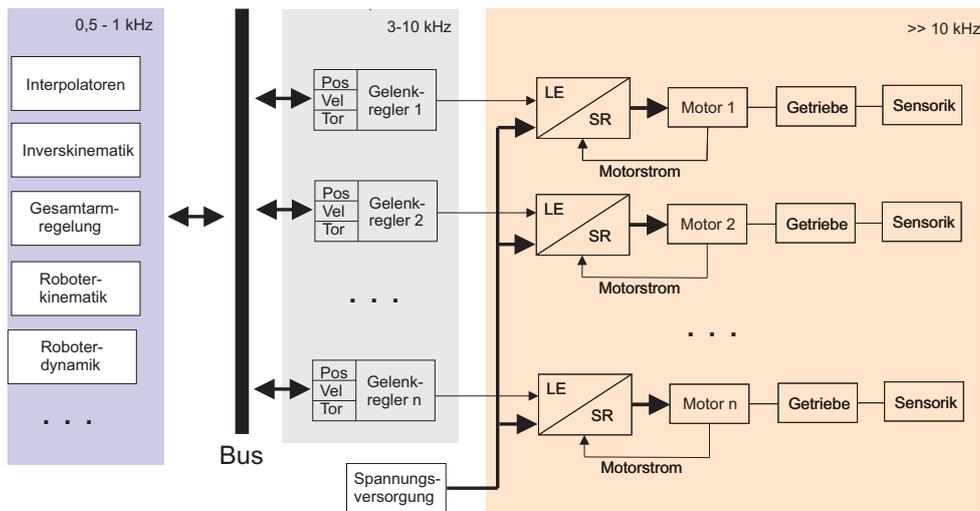


Abbildung 2.28: Komponenten einer Robotersteuerung mit Leistungselektronik (LE) und Stromregelung (SR)

2.3.6 Steuerungsarchitekturen

Abbildung 2.28 zeigt die Hauptkomponenten einer typischen Robotersteuerung. Die innersten schnellsten Regelungsschleifen bilden die Stromregler der einzelnen Motoren. In einer dezentralen Steuerungsarchitektur werden die Gelenkregler in lokalen Signalprozessoren implementiert. Diese Gelenkregler laufen üblicherweise in einer Taktrate von 3-10 kHz und liefern die Sollwerte für die Motorströme bzw. die Motordrehmomente. Ihre Referenzwerte erhalten sie von einer zentralen Robotersteuerung über einen Kommunikationsbus (z.B. EtherCAT, SERCOS). Je nach Betriebsart werden im Gelenk lokale Position-, Geschwindigkeits-, Drehmoment-, oder Impedanzregler (siehe Kapitel 7 „Regelung“) implementiert. Diese Regler beinhalten auch Komponenten zur Reibungskompensation sowie Störgrößenbeobachter. Regelungsalgorithmen welche auf die Informationen von allen Gelenken zurückgreifen müssen, wie auch die Berechnung der Roboterkinematik und -dynamik, werden in einem zentralen Echtzeitrechner implementiert, üblicherweise in einer Taktrate von 500 Hz - 1 kHz.

2.4 Literatur

- [2.1] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [2.2] M. Morse, „Linear Hall-Effect Sensor Angle Measurement Theory, Implementation, and Calibration,“ Texas Instruments Application Note, 2018–2021.
- [2.3] G. Hirzinger, N. Sporer, M. Schedl, J. Butterfaß und M. Grebenstein, „Torque-controlled Light Weight Arms and Articulated Hands - Do We Reach Technological Limits Now?“ In *Experimental Robotics VIII [ISER 2002, Sant’Angelo d’Ischia, Italy, 8-11 July 2002]*, B. Siciliano und P. Dario, Hrsg., Ser. Springer Tracts in Advanced Robotics, Bd. 5, Springer, 2002, S. 21–47. Adresse: https://doi.org/10.1007/3-540-36268-1%5C_2.
- [2.4] C. Q. Zhou, X. Q. Shi, Z. Li und K. Y. Tong, „3D Printed Soft Robotic Hand Combining Post-Stroke Rehabilitation and Stiffness Evaluation,“ in *Intelligent Robotics and Applications*, H. Liu u. a., Hrsg., Cham: Springer International Publishing, 2022, S. 13–23.
- [2.5] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.

3 Starrkörperbewegungen

Für die Beschreibung von Starrkörperbewegungen werden die betrachteten Starrkörper mit körperfesten Koordinatensystemen ausgestattet und ein inertialfestes Koordinatensystem festgelegt. Da die Lage eines Starrkörpers – und damit die Lage aller Massepunkte dieses Körpers – vollständig durch das zugehörige körperfeste Koordinatensystem festgelegt ist, reduzieren sich die Starrkörperbewegungen auf die Bewegungen der Koordinatensysteme selbst.

Abbildung 3.1 zeigt das Beispiel eines Starrkörpers, welcher in allgemeiner Lage in einem Inertialkoordinatensystem liegt. Das Inertialkoordinatensystem wird durch den Ursprung \mathbf{o}_w (\mathbf{o} für engl. *origin*, dt. *Ursprung*) und die Einheitsvektoren der drei Achsen \mathbf{e}_{x_w} , \mathbf{e}_{y_w} und \mathbf{e}_{z_w} beschrieben. Diese vier Elemente werden durch die Kurzbezeichnung $\{w\}$ (für engl. *world*, dt. *Welt*) zusammengefasst. Entsprechend beschreibt $\{k\}$ das körperfeste Koordinatensystem mit dem Ursprung \mathbf{o}_k und den Einheitsvektoren der Achsen \mathbf{e}_{x_k} , \mathbf{e}_{y_k} und \mathbf{e}_{z_k} .

In diesem Kapitel wird zunächst die Translation und die Rotation eines Starrkörpers in zwei separaten Abschnitten betrachtet und begleitend eine geeignete Notation eingeführt. Im Anschluss werden diese Komponenten zu einer homogenen Transformation zusammengefasst, um allgemeine Lagen eines Starrkörpers zu beschreiben. Im nächsten Schritt werden bewegte Koordinatensysteme betrachtet. Abschließend werden unterschiedliche Möglichkeiten, die Orientierung eines Starrkörpers zu beschreiben, aufgezeigt und diskutiert.

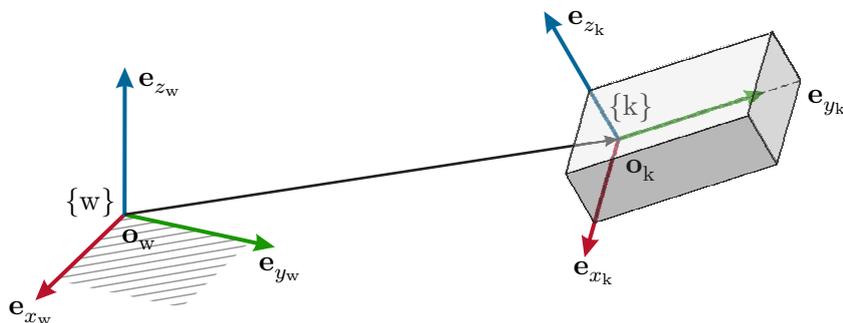


Abbildung 3.1: Inertialfestes Koordinatensystem $\{w\}$ und Starrkörper mit körperfestem Koordinatensystem $\{k\}$. Der Starrkörper ist gegenüber $\{w\}$ allgemein verdreht und verschoben [3.1].

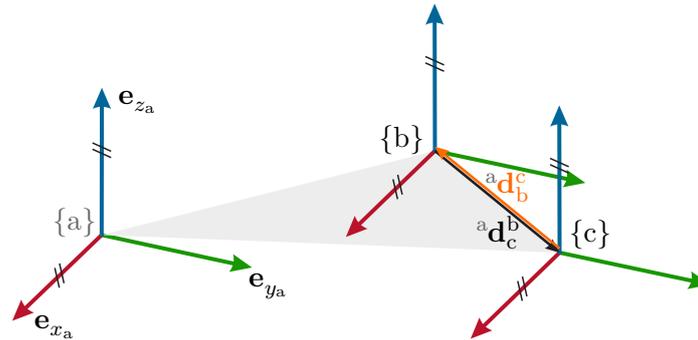


Abbildung 3.2: Verschiebungsvektor ${}^a\mathbf{d}_c^b$ vom Ursprung des Koordinatensystems $\{b\}$ nach $\{c\}$ und ${}^a\mathbf{d}_b^c$ von $\{c\}$ nach $\{b\}$. Die Vektoren werden im Koordinatensystem $\{a\}$ beschrieben.

3.1 Translation

Eine reine Translation eines Koordinatensystems $\{c\}$ gegenüber einem Koordinatensystem $\{b\}$ ist in Abb. 3.2 dargestellt. Die Koordinatensysteme $\{a\}$, $\{b\}$ und $\{c\}$ sind gleich orientiert, d. h. die jeweiligen Koordinatenachsen der x -, y - und z -Achsen sind parallel. Der dreidimensionale Verschiebungsvektor ${}^a\mathbf{d}_c^b$ (\mathbf{d} für engl. *distance*, dt. Distanz), gibt nun die Translation zwischen den Ursprüngen der beiden Koordinatensysteme $\{b\}$ und $\{c\}$ an, beschrieben im dritten Koordinatensystem $\{a\}$. Der Vektor ${}^a\mathbf{d}_c^b$ setzt sich gemäß

$${}^a\mathbf{d}_c^b = d_x {}^a\mathbf{e}_{x_a} + d_y {}^a\mathbf{e}_{y_a} + d_z {}^a\mathbf{e}_{z_a} \quad (3.1)$$

zusammen. Dabei zeigt der Index rechtsunten an, wo die Spitze des Vektors liegt, der Index rechtsoben gibt an, wo der Schaft des Vektors liegt und der Index linksoben gibt an, in welchem Koordinatensystem der Vektor beschrieben wird. Der Verschiebungsvektor ${}^a\mathbf{d}_c^b$ ist also die *Verschiebung des Ursprungs des Koordinatensystems $\{c\}$ vom Ursprung von $\{b\}$ beschrieben in $\{a\}$* . Auch die Einheitsvektoren ${}^a\mathbf{e}_{x_a}$, ${}^a\mathbf{e}_{y_a}$ und ${}^a\mathbf{e}_{z_a}$ werden in jenem Koordinatensystem beschrieben, welches linksoben angegeben ist. In (3.1) sei darauf hingewiesen, dass die Addition nur ausgeführt werden darf, wenn alle Vektoren der Addition im selben Koordinatensystem dargestellt sind. Die Notation kann hierbei unterstützen, nämlich die linken oberen Indizes („beschrieben in“) müssen bei der Addition übereinstimmen.

Kompakt kann ${}^a\mathbf{d}_c^b$ angeschrieben werden, wenn lediglich die Koordinaten der jeweiligen Achsen eines Koordinatensystems als Vektor

$${}^a\mathbf{d}_c^b = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad (3.2)$$

angeschrieben werden, wobei hier die Einheitsvektoren ${}^a\mathbf{e}_{x_a}$, ${}^a\mathbf{e}_{y_a}$ und ${}^a\mathbf{e}_{z_a}$ implizit enthalten sind.

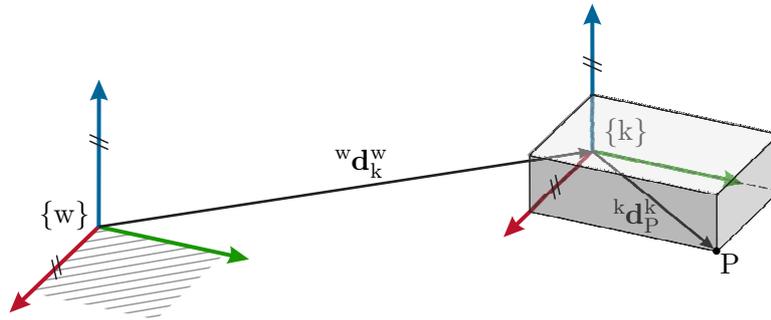


Abbildung 3.3: Inertialfestes Koordinatensystem $\{w\}$ und Starrkörper mit körperfestem Koordinatensystem $\{k\}$. Der Starrkörper ist gegenüber dem inertialfesten Koordinatensystem $\{w\}$ nur verschoben, aber nicht verdreht. Weiters ist ein Punkt P eingezeichnet [3.1].

Für den Verschiebungsvektor ${}^a\mathbf{d}_b^c$ von $\{c\}$ nach $\{b\}$ werden die Spitze und der Schaft im Vergleich zu ${}^a\mathbf{d}_c^b$ vertauscht, siehe Abb. 3.2. Dies entspricht einer Richtungsumkehr des Vektors und es gilt allgemein

$${}^a\mathbf{d}_c^b = -{}^a\mathbf{d}_b^c. \quad (3.3)$$

Im Folgenden wird der Starrkörper mit dem körperfesten Koordinatensystem $\{k\}$ in Abb. 3.3 betrachtet, welcher gegenüber einem inertialfesten Koordinatensystem $\{w\}$ verschoben ist. Für den Verschiebungsvektor ${}^w\mathbf{d}_k^w$ gilt

$${}^w\mathbf{d}_k^w = d_x {}^w\mathbf{e}_{x_w} + d_y {}^w\mathbf{e}_{y_w} + d_z {}^w\mathbf{e}_{z_w}. \quad (3.4)$$

Sind die oberen beiden Indizes gleich („Schaft“ gleich „beschrieben in“), so wird der rechte obere Index weggelassen, um die Lesbarkeit zu erhöhen und die Notation kompakter zu halten. D. h. es ist ${}^w\mathbf{d}_k^w = {}^w\mathbf{d}_k$ bzw. allgemein gilt

$${}^a\mathbf{d}_c^a = {}^a\mathbf{d}_c. \quad (3.5)$$

Ebenfalls kann die Position des Punktes P , siehe Abb. 3.3, als Verschiebung gegenüber einem Ursprung aufgefasst werden. Der Verschiebungsvektor ${}^k\mathbf{d}_P^k = {}^k\mathbf{d}_P$ gibt die *Position des Punktes P vom Ursprung des Koordinatensystems $\{k\}$ beschrieben in $\{k\}$* an.

3.2 Rotation

In diesem Abschnitt wird eine reine Rotation, d. h. eine reine Verdrehung, zweier Koordinatensysteme betrachtet. Im Folgenden wird zunächst die Rotation eines Koordinatensystems um eine Achse als Matrix hergeleitet und im Anschluss die Rotationsmatrix allgemein eingeführt. Weiters werden die Eigenschaften der Rotationsmatrix formalisiert und diskutiert sowie elementare Rotationsmatrizen eingeführt.

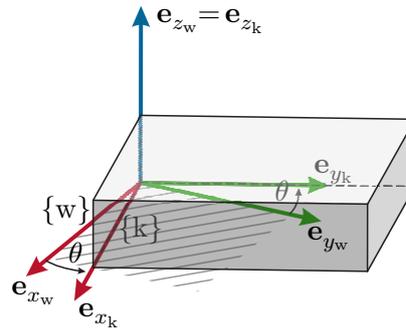


Abbildung 3.4: Starrkörper mit dem körperfesten Koordinatensystem $\{k\}$, welcher durch eine reine Rotation um die z -Achse um den Winkel θ gegenüber dem inertialfesten Koordinatensystem $\{w\}$ verdreht ist.

3.2.1 Rotation um eine Achse

Zunächst sei ein Starrkörper mit dem körperfesten Koordinatensystem $\{k\}$ durch eine reine Rotation um die z -Achse um den Winkel θ gegenüber dem inertialfesten Koordinatensystem $\{w\}$ verdreht, siehe Abb. 3.4. Darin ist ersichtlich, dass die beiden z -Achsen \mathbf{e}_{z_w} und \mathbf{e}_{z_k} zusammenfallen. Weiters ist \mathbf{e}_{x_k} gegenüber \mathbf{e}_{x_w} in der xy -Ebene um θ gedreht und \mathbf{e}_{y_k} ist gegenüber \mathbf{e}_{y_w} um θ gedreht.

Aus Abb. 3.4 können die mathematischen Beziehungen zwischen den Einheitsvektoren der Koordinatensysteme abgelesen werden. Zunächst werden die Einheitsvektoren \mathbf{e}_{x_w} , \mathbf{e}_{y_w} und \mathbf{e}_{z_w} der Koordinatenachsen des inertialfesten Koordinatensystems $\{w\}$ im *selben* Koordinatensystem $\{w\}$ ausgedrückt, d. h. die Einheitsvektoren sind die Basisvektoren des Koordinatensystems $\{w\}$ gemäß

$${}^w\mathbf{e}_{x_w} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad {}^w\mathbf{e}_{y_w} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad {}^w\mathbf{e}_{z_w} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (3.6)$$

Diese Basisvektoren finden sich in allen Koordinatensystemen $\{a\}$ als Einheitsvektoren ${}^a\mathbf{e}_{i_a}$, $i \in \{x, y, z\}$. Mithilfe dieser Einheitsvektoren werden nun die Einheitsvektoren ${}^w\mathbf{e}_{x_k}$, ${}^w\mathbf{e}_{y_k}$ und ${}^w\mathbf{e}_{z_k}$ des körperfesten Koordinatensystems $\{k\}$ im inertialfesten Koordinatensystem $\{w\}$ durch elementare Operationen gemäß

$${}^w\mathbf{e}_{x_k} = \cos(\theta) {}^w\mathbf{e}_{x_w} + \sin(\theta) {}^w\mathbf{e}_{y_w} \quad (3.7a)$$

$${}^w\mathbf{e}_{y_k} = -\sin(\theta) {}^w\mathbf{e}_{x_w} + \cos(\theta) {}^w\mathbf{e}_{y_w} \quad (3.7b)$$

$${}^w\mathbf{e}_{z_k} = {}^w\mathbf{e}_{z_w} \quad (3.7c)$$

beschrieben. Der linke obere Index gibt wieder an, in welchem Koordinatensystem das jeweilige Element beschrieben ist¹. Die Einheitsvektoren (3.7) können als Spaltenvektoren

¹Es sei darauf hingewiesen, dass der linke obere Index bei den Vektoren in Abb. 3.4 nicht angegeben wurde. Es handelt sich um eine grafische Darstellung von geometrischen Objekten in der der mathematische Bezug („beschrieben in“) nicht immer klar dargestellt werden kann.

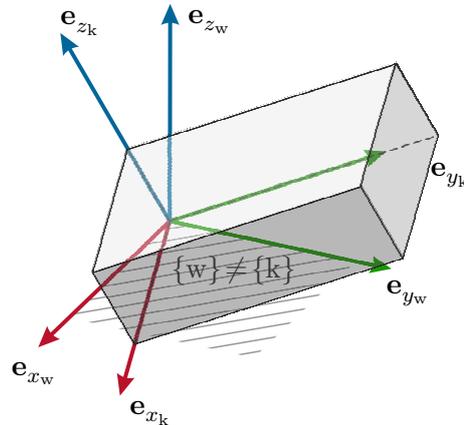


Abbildung 3.5: Rotation eines Starrkörpers mit dem körperfesten Koordinatensystem $\{k\}$, beschrieben durch die Einheitsvektoren \mathbf{e}_{x_k} , \mathbf{e}_{y_k} und \mathbf{e}_{z_k} , in Bezug auf ein inertialfestes Koordinatensystem $\{w\}$ [3.1].

einer Matrix in der Form

$${}^w\mathbf{R}_k = \begin{bmatrix} {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{z_k} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

aufgefasst werden. Es handelt sich dabei um eine *Rotationsmatrix*, welche die Orientierung des körperfesten Koordinatensystems $\{k\}$ beschreibt im inertialfesten Koordinatensystem $\{w\}$.

3.2.2 Rotationsmatrix

Mit (3.8) wurde gezeigt, dass die drei Spaltenvektoren einer Rotationsmatrix für eine einfache Drehung genau den Einheitsvektoren der Koordinatenachsen des gedrehten Koordinatensystems entsprechen. Dieser Zusammenhang gilt auch für allgemeine Drehungen im dreidimensionalen Raum. Dazu wird das körperfeste Koordinatensystem $\{k\}$ in Abb. 3.5 mit den Einheitsvektoren \mathbf{e}_{x_k} , \mathbf{e}_{y_k} und \mathbf{e}_{z_k} betrachtet, welches gegenüber dem inertialfesten Koordinatensystem $\{w\}$ allgemein verdreht ist. Die Rotationsmatrix ${}^w\mathbf{R}_k$, welche die Lage des körperfesten Koordinatensystems $\{k\}$ beschreibt, setzt sich aus den Einheitsvektoren zusammen und hat die Form

$${}^w\mathbf{R}_k = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{z_k} \end{bmatrix}. \quad (3.9)$$

In einem gültigen Koordinatensystem sind die Einheitsvektoren normiert, d. h.

$$\|{}^w\mathbf{e}_{x_k}\|^2 = r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \quad (3.10a)$$

$$\left\| {}^w\mathbf{e}_{y_k} \right\|^2 = r_{12}^2 + r_{22}^2 + r_{32}^2 = 1 \quad (3.10b)$$

$$\left\| {}^w\mathbf{e}_{z_k} \right\|^2 = r_{13}^2 + r_{23}^2 + r_{33}^2 = 1, \quad (3.10c)$$

und stehen orthogonal aufeinander, d. h.

$${}^w\mathbf{e}_{x_k}^T {}^w\mathbf{e}_{y_k} = 0 \quad (3.11a)$$

$${}^w\mathbf{e}_{x_k}^T {}^w\mathbf{e}_{z_k} = 0 \quad (3.11b)$$

$${}^w\mathbf{e}_{y_k}^T {}^w\mathbf{e}_{z_k} = 0. \quad (3.11c)$$

Die Bedingungen (3.10) und (3.11) können in kompakter Form als

$${}^w\mathbf{e}_{i_k}^T {}^w\mathbf{e}_{j_k} = \delta_{ij}, \quad \text{mit } i, j \in \{x, y, z\}, \quad (3.12)$$

bzw. in Matrixform

$${}^w\mathbf{R}_k^T {}^w\mathbf{R}_k = \begin{bmatrix} {}^w\mathbf{e}_{x_k}^T {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{x_k}^T {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{x_k}^T {}^w\mathbf{e}_{z_k} \\ {}^w\mathbf{e}_{y_k}^T {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{y_k}^T {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{y_k}^T {}^w\mathbf{e}_{z_k} \\ {}^w\mathbf{e}_{z_k}^T {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{z_k}^T {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{z_k}^T {}^w\mathbf{e}_{z_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{E}, \quad (3.13)$$

mit dem KRONECKER-Delta δ_{ij} und der Einheitsmatrix \mathbf{E} angeschrieben werden. Eine Rotationsmatrix mit neun Einträgen unterliegt entsprechend sechs Bedingungen und nur drei Einträge können unabhängig voneinander gewählt werden. Aus (3.13) folgen direkt die äquivalenten Eigenschaften

$${}^w\mathbf{R}_k^T {}^w\mathbf{R}_k = \mathbf{E}, \quad (3.14a)$$

$${}^w\mathbf{R}_k^T = {}^w\mathbf{R}_k^{-1} \quad \text{und} \quad (3.14b)$$

$${}^w\mathbf{R}_k^T = {}^k\mathbf{R}_w. \quad (3.14c)$$

Bemerkung 3.1. Die Gleichung (3.14c) ist nicht unmittelbar ersichtlich. Die Spaltenvektoren der Rotationsmatrix in (3.9) können als Projektion der Einheitsvektoren eines Koordinatensystems $\{b\}$ (beschrieben in $\{w\}$) auf die Einheitsvektoren eines Koordinatensystems $\{a\}$ (beschrieben in $\{w\}$) aufgefasst werden, vgl. Abb. 3.4 und (3.7). Die Rotationsmatrix ${}^a\mathbf{R}_b$ kann also in der Form

$${}^a\mathbf{R}_b = \begin{bmatrix} {}^w\mathbf{e}_{x_a}^T {}^w\mathbf{e}_{x_b} & {}^w\mathbf{e}_{x_a}^T {}^w\mathbf{e}_{y_b} & {}^w\mathbf{e}_{x_a}^T {}^w\mathbf{e}_{z_b} \\ {}^w\mathbf{e}_{y_a}^T {}^w\mathbf{e}_{x_b} & {}^w\mathbf{e}_{y_a}^T {}^w\mathbf{e}_{y_b} & {}^w\mathbf{e}_{y_a}^T {}^w\mathbf{e}_{z_b} \\ {}^w\mathbf{e}_{z_a}^T {}^w\mathbf{e}_{x_b} & {}^w\mathbf{e}_{z_a}^T {}^w\mathbf{e}_{y_b} & {}^w\mathbf{e}_{z_a}^T {}^w\mathbf{e}_{z_b} \end{bmatrix} \quad (3.15)$$

angeschrieben werden. Durch die Eigenschaften des Skalarproduktes ${}^w\mathbf{e}_{i_a}^T {}^w\mathbf{e}_{j_b} = {}^w\mathbf{e}_{j_b}^T {}^w\mathbf{e}_{i_a}$ für alle $i, j \in \{x, y, z\}$, und die Transposition der Matrix folgt

$${}^a\mathbf{R}_b = \begin{bmatrix} {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{x_a} & {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{x_a} & {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{x_a} \\ {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{y_a} & {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{y_a} & {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{y_a} \\ {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{z_a} & {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{z_a} & {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{z_a} \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{x_a} & {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{y_a} & {}^w\mathbf{e}_{x_b}^T {}^w\mathbf{e}_{z_a} \\ {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{x_a} & {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{y_a} & {}^w\mathbf{e}_{y_b}^T {}^w\mathbf{e}_{z_a} \\ {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{x_a} & {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{y_a} & {}^w\mathbf{e}_{z_b}^T {}^w\mathbf{e}_{z_a} \end{bmatrix}^T = {}^b\mathbf{R}_a^T, \quad (3.16)$$

womit (3.14c) gezeigt ist.

Weiters werden im Rahmen dieser Vorlesung nur rechtshändige Koordinatensysteme betrachtet. Es gilt daher ${}^w\mathbf{e}_{x_k} \times {}^w\mathbf{e}_{y_k} = {}^w\mathbf{e}_{z_k}$ und damit wird $\det({}^w\mathbf{R}_k) = 1$ gefordert.

Definition 3.1. Die *spezielle orthogonale Gruppe im 3-dimensionalen Raum* $SO(3)$ umfasst alle reellen 3×3 -Matrizen \mathbf{R} , die die Bedingungen $\mathbf{R}^T \mathbf{R} = \mathbf{E}$ und $\det(\mathbf{R}) = 1$ erfüllen. Diese Matrizen werden *Rotationsmatrizen* genannt.

3.2.3 Elementare Rotationsmatrizen

In Abschnitt 3.2.1 bzw. Abb. 3.4 wurde die Lage eines Starrkörpers bezüglich des Inertialkoordinatensystems beschrieben. Darin ist das körperfeste Koordinatensystem $\{k\}$ um den Winkel θ um die \mathbf{e}_{z_w} -Achse bzw. \mathbf{e}_{z_k} -Achse verdreht. Die dabei hergeleitete Rotationsmatrix (3.8) ist eine der drei elementaren Rotationsmatrizen

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.17a)$$

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.17b)$$

$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (3.17c)$$

Die elementaren Rotationsmatrizen (3.17) beschreiben eine einfache Drehung um eine Koordinatenachse x , y bzw. z um den Winkel θ . Eine Drehung um eine allgemeine Achse, angegeben durch den Einheitsvektor $\mathbf{e} = [e_1 \ e_2 \ e_3]^T$, wird mithilfe der Rotationsmatrix

$$\mathbf{R}_{\mathbf{e},\theta} = \begin{bmatrix} c_\theta + e_1^2(1-c_\theta) & e_1e_2(1-c_\theta) - e_3s_\theta & e_1e_3(1-c_\theta) + e_2s_\theta \\ e_1e_2(1-c_\theta) + e_3s_\theta & c_\theta + e_2^2(1-c_\theta) & e_2e_3(1-c_\theta) - e_1s_\theta \\ e_1e_3(1-c_\theta) - e_2s_\theta & e_2e_3(1-c_\theta) + e_1s_\theta & c_\theta + e_3^2(1-c_\theta) \end{bmatrix} \quad (3.18)$$

beschrieben, wobei $s_x = \sin(x)$ und $c_x = \cos(x)$ kompakte Notationen für die trigonometrischen Funktionen sind. Es gilt $\mathbf{R}_{\mathbf{e},\theta} = \mathbf{R}_{-\mathbf{e},-\theta}$. Der Zusammenhang (3.18) wird später mit Satz 3.1 hergeleitet.

3.3 Homogene Transformation

In den Abschnitten 3.1 und 3.2 wurde die *Pose* eines Starrkörpers, d. h. die Position und Orientierung, mithilfe eines Verschiebungsvektors und einer Rotationsmatrix dargestellt. Mit den sogenannten *homogenen Transformationen* \mathbf{H} werden diese beiden Komponenten zu einer Matrix zusammengefasst und es können damit alle Starrkörperlagen und -bewegungen abgebildet werden.

Definition 3.2. Die *spezielle euklidische Gruppe im 3-dimensionalen Raum* $SE(3)$ umfasst alle reellen 4×4 -Matrizen der Bauform

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (3.19)$$

mit der Rotationsmatrix $\mathbf{R} \in SO(3)$ und dem Vektor $\mathbf{d} \in \mathbb{R}^3$. Diese Matrizen werden *homogene Transformationen* bzw. *Transformationsmatrizen* genannt.

3.3.1 Pose eines Starrkörpers

Abbildung 3.1 zeigt einen Starrkörper, der mit dem körperfesten Koordinatensystem $\{\mathbf{k}\}$ ausgestattet ist und in allgemeiner Lage relativ zum inertialfesten Koordinatensystem $\{\mathbf{w}\}$ liegt. Die Verschiebung des Ursprungs von $\{\mathbf{k}\}$ gegenüber $\{\mathbf{w}\}$ wird durch ${}^w\mathbf{d}_k$ beschrieben, siehe (3.1), und die Orientierung von $\{\mathbf{k}\}$ bezüglich $\{\mathbf{w}\}$ wird durch die Rotationsmatrix ${}^w\mathbf{R}_k$ angegeben. Die zugehörige homogene Transformation

$${}^w\mathbf{H}_k = \begin{bmatrix} {}^w\mathbf{R}_k & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.20)$$

beschreibt somit die *Pose* des körperfesten Koordinatensystems $\{\mathbf{k}\}$ in Bezug auf das Inertialkoordinatensystem $\{\mathbf{w}\}$ ausgedrückt in $\{\mathbf{w}\}$, d. h. dessen Position und Orientierung.

3.3.2 Wechsel des Bezugskordinatensystems

In diesem Abschnitt wird Abb. 3.6 mit den Koordinatensystemen $\{\mathbf{w}\}$, $\{\mathbf{b}\}$ und $\{\mathbf{k}\}$ betrachtet. Während die Koordinatensysteme in dieser Abbildung nur in der Ebene verschoben und verdreht sind, können mithilfe der homogenen Transformationen auch allgemeinere Lagen mit (3.17) und (3.18) beschrieben werden – die folgenden Beziehungen zwischen den Koordinatensystemen bleiben bestehen. Es gilt

$${}^w\mathbf{H}_k = \begin{bmatrix} {}^w\mathbf{R}_k & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{mit} \quad {}^w\mathbf{R}_k = \begin{bmatrix} {}^w\mathbf{e}_{x_k} & {}^w\mathbf{e}_{y_k} & {}^w\mathbf{e}_{z_k} \end{bmatrix} = \mathbf{R}_{z,\phi}$$

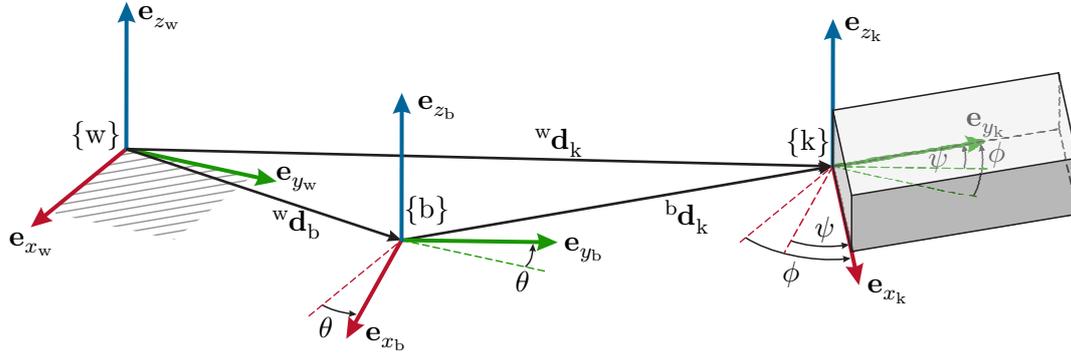


Abbildung 3.6: Drei Koordinatensysteme $\{w\}$, $\{k\}$ und $\{b\}$, welche durch die Verschiebungsvektoren ${}^w\mathbf{d}_k$ und ${}^b\mathbf{d}_k$ verschoben sind und um die Winkel θ und ψ zueinander verdreht sind. Das Koordinatensystem $\{w\}$ ist inertialfest [3.1].

und

$${}^w\mathbf{H}_b = \begin{bmatrix} {}^w\mathbf{R}_b & {}^w\mathbf{d}_b \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{mit} \quad {}^w\mathbf{R}_b = \begin{bmatrix} {}^w\mathbf{e}_{x_b} & {}^w\mathbf{e}_{y_b} & {}^w\mathbf{e}_{z_b} \end{bmatrix} = \mathbf{R}_{z,\theta}$$

Jedes Koordinatensystem kann bezüglich eines anderen Koordinatensystems dargestellt werden, solange das jeweilige Bezugskoordinatensystem berücksichtigt wird. Aus Abb. 3.6 kann

$${}^b\mathbf{H}_k = \begin{bmatrix} {}^b\mathbf{R}_k & {}^b\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{mit} \quad {}^b\mathbf{R}_k = \begin{bmatrix} {}^b\mathbf{e}_{x_k} & {}^b\mathbf{e}_{y_k} & {}^b\mathbf{e}_{z_k} \end{bmatrix} = \mathbf{R}_{z,\psi}$$

abgelesen werden. Hierbei ist zu beachten, dass die Elemente ${}^b\mathbf{R}_k$ und ${}^b\mathbf{d}_k$ bezüglich des Koordinatensystems $\{b\}$ ausgedrückt werden müssen. Dies wird durch den linken oberen Index angezeigt. Schließlich kann mit der Multiplikation mit ${}^w\mathbf{H}_b$ gemäß

$${}^w\mathbf{H}_b {}^b\mathbf{H}_k = \begin{bmatrix} {}^w\mathbf{R}_b & {}^w\mathbf{d}_b \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{R}_k & {}^b\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{R}_b {}^b\mathbf{R}_k & {}^w\mathbf{d}_b + {}^w\mathbf{R}_b {}^b\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix}$$

das Bezugskoordinatensystem von ${}^b\mathbf{H}_k$ auf $\{w\}$ gewechselt werden, d. h. es wird ${}^w\mathbf{H}_k$ mit ${}^w\mathbf{R}_k = {}^w\mathbf{R}_b {}^b\mathbf{R}_k$ und ${}^w\mathbf{d}_k = {}^w\mathbf{d}_b + {}^w\mathbf{R}_b {}^b\mathbf{d}_k$ gefunden. Die Notation unterstützt dabei den Rechenweg, denn der obere Index „b“ kürzt sich gedanklich mit dem unteren Index „b“ und es bleiben nur „w“ links oben und „k“ rechts unten übrig. Im Detail lautet der Rechenweg

$${}^w\mathbf{H}_b {}^b\mathbf{H}_k = {}^w\mathbf{H}_b {}^b\mathbf{H}_k = {}^w\mathbf{H}\mathbf{H}_k = {}^w\mathbf{H}_k. \quad (3.21)$$

Für Rotationsmatrizen gilt

$${}^w\mathbf{R}_b {}^b\mathbf{R}_k = {}^w\mathbf{R}_b {}^b\mathbf{R}_k = {}^w\mathbf{R}\mathbf{R}_k = {}^w\mathbf{R}_k \quad (3.22)$$

3.3.4 Elementare homogene Transformationen

Eine allgemeine Translation um den Vektor \mathbf{d} wird durch die homogene Transformation

$$\mathbf{H}_{T,\mathbf{d}} = \begin{bmatrix} \mathbf{E} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.27)$$

beschrieben. Die elementare homogene Transformation $\mathbf{H}_{T_i,d}$ beschreibt eine Translation um die skalare Distanz d entlang der lokalen Koordinatenachse $i \in \{x, y, z\}$, d. h.

$$\mathbf{H}_{T_i,d} = \begin{bmatrix} \mathbf{E} & d\mathbf{e}_i \\ \mathbf{0} & 1 \end{bmatrix}, \quad (3.28)$$

mit den Basisvektoren analog zu (3.6). Eine Rotation um die lokale Achse $i \in \{x, y, z\}$ mit dem Winkel θ wird im Folgenden mit der homogenen Transformation $\mathbf{H}_{R_i,\theta}$ bezeichnet, welche sich gemäß

$$\mathbf{H}_{R_i,\theta} = \begin{bmatrix} \mathbf{R}_{i,\theta} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.29)$$

mit der zugehörigen Rotationsmatrix $\mathbf{R}_{i,\theta}$ aus (3.17) zusammensetzt. Wird auch hier die Rotationsmatrix für eine allgemeine Drehung um eine Achse \mathbf{e} um den Winkel θ aus (3.18) verwendet, so wird die zugehörige homogene Transformation mit $\mathbf{H}_{R_e,\theta}$ bezeichnet. Schließlich kann jede homogene Transformation auch als Produkt einer Translation $\mathbf{H}_{T,\mathbf{d}}$ und einer Rotation $\mathbf{H}_{R_e,\theta}$ gemäß

$$\mathbf{H} = \mathbf{H}_{T,\mathbf{d}}\mathbf{H}_{R_e,\theta} = \begin{bmatrix} \mathbf{E} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{e,\theta} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{e,\theta} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.30)$$

zusammengesetzt werden.

3.3.5 Versatz eines Vektors oder Koordinatensystems

Homogene Transformationen können verwendet werden, um einen Starrkörper bzw. ein Koordinatensystem zu versetzen, d. h. eine Rotation und Translation auszuführen. Bei den homogenen Transformationen (wie auch bei den Rotationsmatrizen) ist es essenziell, ob eine Transformation von links oder von rechts multipliziert wird, da diese im Allgemeinen nicht kommutieren, vgl. Abschnitt 3.3.3.

Im Folgenden wird dazu das körperfeste Koordinatensystem $\{\mathbf{k}\}$, beschrieben durch ${}^w\mathbf{H}_{\mathbf{k}}$, betrachtet und darauf eine allgemeine homogene Transformation $\mathbf{H} = \mathbf{H}_{T,\mathbf{d}}\mathbf{H}_{R_e,\theta}$ angewendet, siehe Abb. 3.7. Wird diese von rechts multipliziert, so ergibt sich

$$\begin{aligned} {}^w\mathbf{H}_{\mathbf{k}'} &= {}^w\mathbf{H}_{\mathbf{k}}\mathbf{H} = {}^w\mathbf{H}_{\mathbf{k}}\mathbf{H}_{T,\mathbf{d}}\mathbf{H}_{R_e,\theta} \\ &= \begin{bmatrix} {}^w\mathbf{R}_{\mathbf{k}} & {}^w\mathbf{d}_{\mathbf{k}} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{e,\theta} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{R}_{\mathbf{k}}\mathbf{R}_{e,\theta} & {}^w\mathbf{R}_{\mathbf{k}}\mathbf{d} + {}^w\mathbf{d}_{\mathbf{k}} \\ \mathbf{0} & 1 \end{bmatrix}. \end{aligned} \quad (3.31)$$

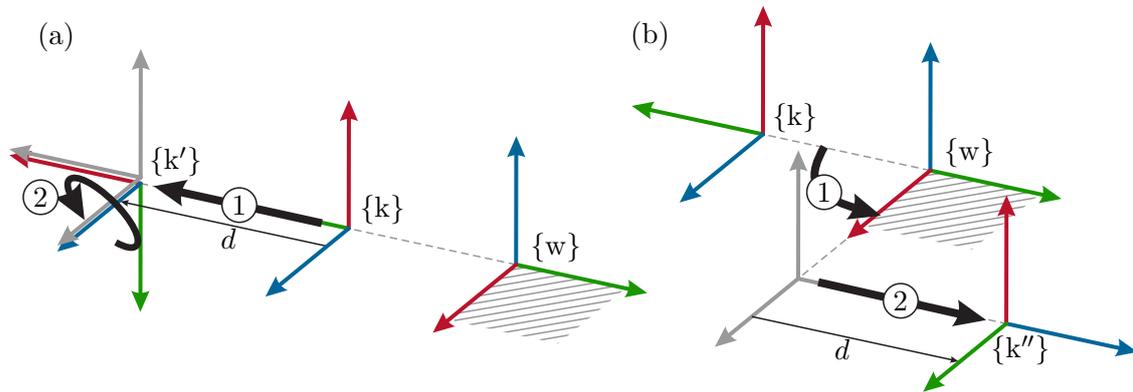


Abbildung 3.7: Versatz des Koordinatensystems $\{k\}$ mit der homogenen Transformation $\mathbf{H} = \mathbf{H}_{T,d} \mathbf{H}_{R,e,\theta}$. Für die Darstellung ist $\mathbf{d} = [0 \ d \ 0]^T$, $\mathbf{e} = [0 \ 0 \ 1]^T$ und $\theta = 90^\circ$ gewählt. (a) Transformation bezüglich des körperfesten Koordinatensystems ${}^w\mathbf{H}_{k'} = {}^w\mathbf{H}_k \mathbf{H}$, (b) Transformation bezüglich des inertialfesten Koordinatensystems ${}^w\mathbf{H}_{k''} = \mathbf{H} {}^w\mathbf{H}_k$ [3.1].

Dieses Ergebnis kann so interpretiert werden, dass zunächst eine Translation um den Vektor \mathbf{d} , beschrieben im körperfesten Koordinatensystem $\{k\}$, ausgeführt wird und im Anschluss eine reine Rotation um die Achse \mathbf{e} , beschrieben im verschobenen Koordinatensystem, um den Winkel θ ausgeführt wird. Dies führt zum neuen körperfesten Koordinatensystem $\{k'\}$, siehe Abb. 3.7a. Wird im Gegensatz dazu eine Multiplikation von links verwendet, führt dies zu

$$\begin{aligned} {}^w\mathbf{H}_{k''} &= \mathbf{H} {}^w\mathbf{H}_k = \mathbf{H}_{T,d} \mathbf{H}_{R,e,\theta} {}^w\mathbf{H}_k \\ &= \begin{bmatrix} \mathbf{R}_{e,\theta} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^w\mathbf{R}_k & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{e,\theta} {}^w\mathbf{R}_k & \mathbf{R}_{e,\theta} {}^w\mathbf{d}_k + \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}, \end{aligned} \quad (3.32)$$

d. h. ein Versatz bezüglich des inertialfesten Koordinatensystems wird ausgeführt. Abb. 3.7b zeigt, dass in diesem Fall zuerst das körperfeste Koordinatensystem $\{k\}$ um die Achse \mathbf{e} , beschrieben in $\{w\}$, um den Winkel θ gedreht wird. Dabei wird im Allgemeinen auch der Ursprung von $\{k\}$ bewegt. Im Anschluss wird eine Translation um den Vektor \mathbf{d} , ebenfalls beschrieben in $\{w\}$, ausgeführt. Das Resultat des Versatzes ist das körperfeste Koordinatensystem $\{k''\}$.

3.4 Bewegte Koordinatensysteme

In diesem Abschnitt werden bewegte Koordinatensysteme betrachtet. Diese können sich sowohl geradlinig mit einer linearen Geschwindigkeit als auch rotatorisch mit einer Drehwinkelgeschwindigkeit bewegen. Zunächst werden reine Rotationsbewegungen betrachtet und im Anschluss werden allgemeine Bewegungen mithilfe des *Vektors der räumlichen Geschwindigkeit* (engl. *twist*) beschrieben.

$$= \left[{}^w\boldsymbol{\omega}_k \times {}^w\mathbf{e}_{x_k} \quad {}^w\boldsymbol{\omega}_k \times {}^w\mathbf{e}_{y_k} \quad {}^w\boldsymbol{\omega}_k \times {}^w\mathbf{e}_{z_k} \right] =: {}^w\boldsymbol{\omega}_k \times {}^w\mathbf{R}_k. \quad (3.35)$$

Definition 3.3. Die Matrix

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3.36)$$

ist die *schiefsymmetrische Matrixdarstellung* des Vektors $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T$, d. h. es gilt $\mathbf{S}(\boldsymbol{\omega}) = -\mathbf{S}^T(\boldsymbol{\omega})$.

Wie durch direktes Nachrechnen bestätigt wird, kann das Kreuzprodukt in (3.35) umgeschrieben werden als ${}^w\boldsymbol{\omega}_k \times {}^w\mathbf{R}_k = \mathbf{S}({}^w\boldsymbol{\omega}_k) {}^w\mathbf{R}_k$. Entsprechend ergibt sich der Vektor der Drehwinkelgeschwindigkeiten im Inertialkoordinatensystem ${}^w\boldsymbol{\omega}_k$ aus (3.35) zu

$$\mathbf{S}({}^w\boldsymbol{\omega}_k) = {}^w\dot{\mathbf{R}}_k {}^w\mathbf{R}_k^{-1} = {}^w\dot{\mathbf{R}}_k {}^w\mathbf{R}_k^T \quad (3.37)$$

unter Verwendung der Eigenschaft (3.14b).

Im nächsten Schritt werden die obigen Betrachtungen für das körperfeste Koordinatensystem $\{k\}$ wiederholt, d. h. es wird der Vektor der Drehwinkelgeschwindigkeit ${}^k\boldsymbol{\omega}_k^w$ für die Drehbewegung des Koordinatensystems $\{k\}$ gegenüber $\{w\}$ beschrieben in $\{k\}$ gesucht. Analog zu (3.23) kann auch für Vektoren der Drehwinkelgeschwindigkeit das Bezugssystem gemäß

$${}^w\mathbf{R}_k {}^k\boldsymbol{\omega}_k^w = {}^w\mathbf{R}_k {}^k\boldsymbol{\omega}_k^w = {}^w\mathbf{R}_k {}^k\boldsymbol{\omega}_k^w = {}^w\boldsymbol{\omega}_k^w \quad (3.38)$$

gewechselt werden. Zunächst wird der Ausdruck

$${}^w\mathbf{R}_k^T \mathbf{S}({}^w\boldsymbol{\omega}_k) {}^w\mathbf{R}_k = {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k {}^w\mathbf{R}_k^T {}^w\mathbf{R}_k = {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k. \quad (3.39)$$

mithilfe von (3.37) ausgewertet. Das Kreuzprodukt $\mathbf{S}({}^w\boldsymbol{\omega}_k) {}^w\mathbf{R}_k = {}^w\boldsymbol{\omega}_k \times {}^w\mathbf{R}_k$ auf der linken Seite wird zu

$$\begin{aligned} {}^w\mathbf{R}_k^T \mathbf{S}({}^w\boldsymbol{\omega}_k) {}^w\mathbf{R}_k &= {}^w\mathbf{R}_k^T ({}^w\boldsymbol{\omega}_k \times {}^w\mathbf{R}_k) \\ &= ({}^w\mathbf{R}_k^T {}^w\boldsymbol{\omega}_k^w) \times ({}^w\mathbf{R}_k^T {}^w\mathbf{R}_k) = {}^k\boldsymbol{\omega}_k^w \times \mathbf{E} = \mathbf{S}({}^k\boldsymbol{\omega}_k^w) \end{aligned} \quad (3.40)$$

vereinfacht. Folglich ist mit

$$\mathbf{S}({}^k\boldsymbol{\omega}_k^w) = {}^w\mathbf{R}_k^{-1} {}^w\dot{\mathbf{R}}_k = {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k \quad (3.41)$$

der Vektor der Drehwinkelgeschwindigkeit ${}^k\boldsymbol{\omega}_k^w$ im körperfesten Koordinatensystem $\{k\}$ gefunden.

Schließlich sei noch darauf hingewiesen, dass der Vektor der Drehwinkelgeschwindigkeiten im Inertialkoordinatensystem ${}^w\boldsymbol{\omega}_k^w$ nicht von der Wahl des körperfesten Koordinatensystems $\{k\}$ abhängt. Gleichung (3.37) beinhaltet zwar die Rotationsmatrix ${}^w\mathbf{R}_k$ und deren

Ableitung, welche beide sowohl vom Inertialkoordinatensystem $\{w\}$ als auch vom körperfesten Koordinatensystem $\{k\}$ abhängen. Allerdings ist das *Produkt* ${}^w\mathbf{R}_k {}^w\mathbf{R}_k^T$ unabhängig von der Wahl von $\{k\}$. Der Vektor ${}^w\boldsymbol{\omega}_k^w$ beschreibt lediglich eine momentane Drehrate z. B. eines Starrkörpers bezüglich des Inertialkoordinatensystem $\{w\}$ ausgedrückt in $\{w\}$. Diese momentane *Drehrate* ist in allen Punkten des Starrkörpers gleich³ und damit ist sie unabhängig von der Platzierung des Koordinatensystems $\{k\}$. Umgekehrt ist auch ${}^k\boldsymbol{\omega}_k^w$ (3.41) *unabhängig* von der Wahl des Inertialkoordinatensystems $\{w\}$.

Bemerkung 3.2 (Drehwinkelgeschwindigkeit eines Starrkörpers.). Im Folgenden wird ein allgemeiner Starrkörper betrachtet, welcher mit zwei Koordinatensystemen $\{a\}$ und $\{b\}$ ausgestattet ist, siehe Abb. 3.9. Dabei wird zunächst angenommen, dass das Koordinatensystem $\{a\}$ mit der Drehwinkelgeschwindigkeit ${}^w\boldsymbol{\omega}_a$ und das Koordinatensystem $\{b\}$ mit einer anderen Drehwinkelgeschwindigkeit ${}^w\boldsymbol{\omega}_b \neq {}^w\boldsymbol{\omega}_a$ momentan rotiert. Über die eingezeichneten Verschiebungsvektoren können die Position ${}^w\mathbf{d}_P$ und die Geschwindigkeit ${}^w\dot{\mathbf{d}}_P$ des Punktes P sowohl über das Koordinatensystem $\{a\}$ als auch über $\{b\}$ gleichermaßen beschrieben werden, nämlich

$${}^w\mathbf{d}_P = {}^w\mathbf{d}_a + {}^w\mathbf{d}_P^a = {}^w\mathbf{d}_b + {}^w\mathbf{d}_P^b \quad (3.42)$$

$${}^w\dot{\mathbf{d}}_P = {}^w\dot{\mathbf{d}}_a + {}^w\boldsymbol{\omega}_a \times {}^w\mathbf{d}_P^a = {}^w\dot{\mathbf{d}}_b + {}^w\boldsymbol{\omega}_b \times {}^w\mathbf{d}_P^b . \quad (3.43)$$

Die Geschwindigkeit des Koordinatensystems $\{b\}$ kann ebenfalls über $\{a\}$ in der Form

$${}^w\dot{\mathbf{d}}_b = {}^w\dot{\mathbf{d}}_a + {}^w\boldsymbol{\omega}_a \times {}^w\mathbf{d}_b^a = {}^w\dot{\mathbf{d}}_a + {}^w\boldsymbol{\omega}_a \times ({}^w\mathbf{d}_P^a - {}^w\mathbf{d}_P^b) \quad (3.44)$$

beschrieben werden, wobei ${}^w\mathbf{d}_b^a = {}^w\mathbf{d}_P^a - {}^w\mathbf{d}_P^b$ verwendet wurde, siehe Abb. 3.9. Wird nun (3.44) in (3.43) eingesetzt, so ergibt sich

$${}^w\dot{\mathbf{d}}_a + {}^w\boldsymbol{\omega}_a \times {}^w\mathbf{d}_P^a = {}^w\dot{\mathbf{d}}_a + {}^w\boldsymbol{\omega}_a \times ({}^w\mathbf{d}_P^a - {}^w\mathbf{d}_P^b) + {}^w\boldsymbol{\omega}_b \times {}^w\mathbf{d}_P^b \quad (3.45)$$

bzw. vereinfacht

$$\mathbf{0} = ({}^w\boldsymbol{\omega}_b - {}^w\boldsymbol{\omega}_a) \times {}^w\mathbf{d}_P^b . \quad (3.46)$$

Da (3.46) für alle beliebigen Vektoren ${}^w\mathbf{d}_P^b$ gilt, muss der Ausdruck in der Klammer verschwinden und es folgt daraus ${}^w\boldsymbol{\omega}_b = {}^w\boldsymbol{\omega}_a$. Dies steht im Widerspruch zur ursprünglichen Annahme (${}^w\boldsymbol{\omega}_b \neq {}^w\boldsymbol{\omega}_a$) und beweist, dass die momentane *Drehrate* in allen Punkten des Starrkörpers gleich ist.

³Die *translatorische Geschwindigkeit* zufolge der Drehrate ${}^w\boldsymbol{\omega}_k^w$ hingegen hängt von der Position der Drehachse ab und steigt linear mit dem Abstand zur Drehachse für jeden Punkt des Starrkörpers.

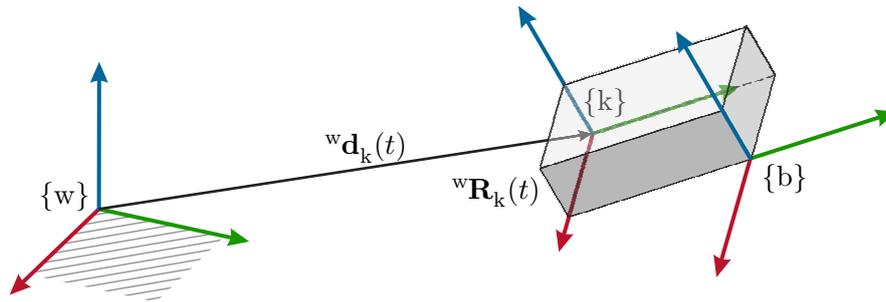


Abbildung 3.10: Bewegter Starrkörper mit dem körperfesten Koordinatensystem $\{k\}$, welches durch den zeitabhängigen Ursprung ${}^w \mathbf{d}_k(t)$ und der zeitabhängigen Orientierung ${}^w \mathbf{R}_k(t)$ beschrieben wird. Ein Eckpunkt des Starrkörpers ist mit einem weiteren Koordinatensystem $\{b\}$ ausgestattet.

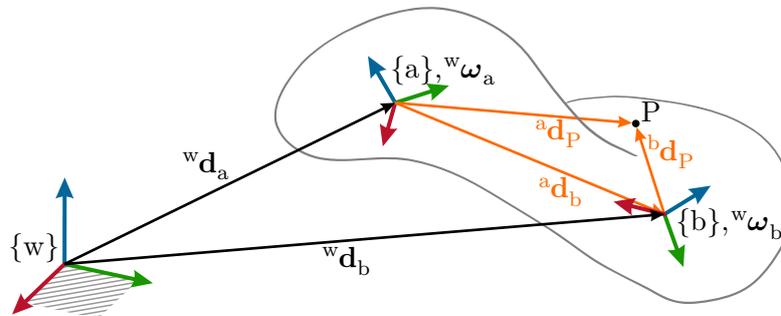


Abbildung 3.9: Allgemeiner Starrkörper mit zwei Koordinatensystemen $\{a\}$ und $\{b\}$.

3.4.2 Räumliche Geschwindigkeit

Die reinen Drehbewegungen von Koordinatensystemen werden in diesem Abschnitt auf allgemeine Starrkörperbewegungen erweitert. Dazu wird ein Starrkörper mit einem körperfesten Koordinatensystem $\{k\}$ betrachtet, welcher sich gegenüber einem Inertialkoordinatensystem $\{w\}$ allgemein bewegt, siehe Abb. 3.10. Dies wird mit der homogenen Transformation ${}^w \mathbf{H}_k(t)$ als Funktion der Zeit in der Form

$${}^w \mathbf{H}_k(t) = \begin{bmatrix} {}^w \mathbf{R}_k(t) & {}^w \mathbf{d}_k(t) \\ \mathbf{0} & 1 \end{bmatrix} \quad (3.47)$$

beschrieben. Analog zu Abschnitt 3.4.1, in dem die Drehwinkelgeschwindigkeiten ${}^w \boldsymbol{\omega}_k$ bzw. ${}^k \boldsymbol{\omega}_k^w$ mithilfe von ${}^w \mathbf{R}_k$ und ${}^w \dot{\mathbf{R}}_k$ berechnet wurden, wird nun ein Ausdruck für die räumliche Geschwindigkeit mithilfe von ${}^w \mathbf{H}_k$ und ${}^w \dot{\mathbf{H}}_k$ ermittelt. Dazu wird zunächst der

Ausdruck ${}^w\mathbf{H}_k^{-1}{}^w\dot{\mathbf{H}}_k$ mithilfe von (3.26)

$$\begin{bmatrix} {}^w\mathbf{R}_k^T & -{}^w\mathbf{R}_k^T{}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^w\dot{\mathbf{R}}_k & {}^w\dot{\mathbf{d}}_k \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{R}_k^T{}^w\dot{\mathbf{R}}_k & {}^w\mathbf{R}_k^T{}^w\dot{\mathbf{d}}_k \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{S}({}^k\boldsymbol{\omega}_k^w) & {}^k\mathbf{v}_k^w \\ \mathbf{0} & 0 \end{bmatrix} \quad (3.48)$$

ausgewertet, wobei ${}^w\dot{\mathbf{d}}_k^w$ die lineare Geschwindigkeit des Ursprungs von $\{k\}$ gegenüber $\{w\}$ ausgedrückt in $\{w\}$ beschreibt. Durch die Transformation mit ${}^w\mathbf{R}_k^T$ ist der Ausdruck ${}^k\mathbf{v}_k^w = {}^w\mathbf{R}_k^T{}^w\dot{\mathbf{d}}_k^w$ die momentane lineare Geschwindigkeit des körperfesten Koordinatensystems $\{k\}$ beschrieben in $\{k\}$. Weiters wurde der Zusammenhang (3.41) für den Vektor der Drehwinkelgeschwindigkeit ${}^k\boldsymbol{\omega}_k^w$, ausgedrückt in $\{k\}$, verwendet. Die obigen Geschwindigkeiten werden zum *Vektor der räumlichen Geschwindigkeit im körperfesten Koordinatensystem* (engl. *body twist*) in der Form

$${}^k\mathbf{t}_k^w = \begin{bmatrix} {}^k\mathbf{v}_k^w \\ {}^k\boldsymbol{\omega}_k^w \end{bmatrix} \in \mathbb{R}^6 \quad (3.49)$$

zusammengefasst. Zur einfacheren Notation wird häufig für die Matrix (3.48) auch $\begin{bmatrix} {}^k\mathbf{t}_k^w \end{bmatrix}$ verwendet.

3.4.3 Transformation räumlicher Geschwindigkeiten

Vektoren der räumlichen Geschwindigkeit (3.49) können auch in andere Koordinatensysteme mit der sogenannten *adjungierten Darstellung* $\mathbf{Ad}(\mathbf{H})$ einer *homogenen Transformation* \mathbf{H} transformiert werden. Diese wird anhand folgender Definition eingeführt.

Definition 3.4. Es sei $\mathbf{H} \in \text{SE}(3)$ eine homogene Transformation gemäß Definition 3.2. Dann ist die Matrix

$$\mathbf{Ad}(\mathbf{H}) = \begin{bmatrix} \mathbf{R} & \mathbf{S}(\mathbf{d})\mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{6 \times 6} \quad (3.50)$$

die *adjungierte Darstellung der homogenen Transformation* \mathbf{H} . Die zugehörige *adjungierte Abbildung* eines Vektors $\mathbf{t} \in \mathbb{R}^6$ lautet

$$\mathbf{t}' = \mathbf{Ad}(\mathbf{H})\mathbf{t} . \quad (3.51)$$

Für eine genaue Herleitung dieser Matrix sei auf die Literatur am Ende des Kapitels verwiesen [3.2].

Als Beispiel wird nochmals der bewegte Starrkörper in Abb. 3.10 betrachtet. Es sei angenommen, dass die momentane Bewegung des Starrkörpers als ${}^w\mathbf{H}_k(t)$, zusammengesetzt aus ${}^w\mathbf{d}_k(t)$ und ${}^w\mathbf{R}_k(t)$, sowie die *konstante* homogene Transformation ${}^k\mathbf{H}_b$ bekannt sind. Der Vektor der räumlichen Geschwindigkeit (3.49) wird mithilfe der adjungierten Darstellung von ${}^k\mathbf{H}_b$ gemäß

$${}^b\mathbf{t}_b^w = \begin{bmatrix} {}^b\mathbf{v}_b^w \\ {}^b\boldsymbol{\omega}_b^w \end{bmatrix} = \mathbf{Ad}({}^k\mathbf{H}_b^{-1}){}^k\mathbf{t}_k^w = \mathbf{Ad}({}^b\mathbf{H}_k){}^k\mathbf{t}_k^w \quad (3.52)$$

in das körperfeste Koordinatensystem $\{b\}$ transformiert. Hier ist zu beachten, dass sich in (3.52) auch der rechte untere Index auf „b“ ändert, wie in folgender Bemerkung erläutert wird.

Bemerkung 3.3. Bei (3.52) handelt es sich um den Spezialfall einer allgemeineren Transformation

$${}^b\mathbf{t}_b^w = \mathbf{Ad}({}^b\mathbf{H}_k) {}^k\mathbf{t}_k^w + {}^b\mathbf{t}_b^k = \mathbf{Ad}({}^b\mathbf{H}_k) {}^k\mathbf{t}_k^w + {}^b\mathbf{t}_b^k = \mathbf{Ad}({}^b\mathbf{H}) \mathbf{t}_b^w + \mathbf{t}_b, \quad (3.53)$$

welche für ein bewegtes Koordinatensystem ${}^b\mathbf{H}_k(t)$ mit dem zugehörigen Geschwindigkeitsvektor ${}^b\mathbf{t}_b^k$ gilt. In dieser Transformation werden zwei Geschwindigkeitsvektoren aneinander gefügt, ähnlich (3.24) und Abb. 3.3. Da die homogene Transformation ${}^b\mathbf{H}_k$ in (3.52) als konstant angenommen wurde, fällt der zweite Term ${}^b\mathbf{t}_b^k$ in (3.53) weg.

Soll lediglich der linke obere Index gewechselt werden („beschrieben in“), so wird eine vereinfachte Transformation gemäß

$${}^b\mathbf{t}_k^w = \begin{bmatrix} {}^b\mathbf{R}_k & \mathbf{0} \\ \mathbf{0} & {}^b\mathbf{R}_k \end{bmatrix} {}^k\mathbf{t}_k^w = \begin{bmatrix} {}^b\mathbf{R}_k & \mathbf{0} \\ \mathbf{0} & {}^b\mathbf{R}_k \end{bmatrix} {}^k\mathbf{t}_k^w = \begin{bmatrix} {}^b\mathbf{R} & \mathbf{0} \\ \mathbf{0} & {}^b\mathbf{R} \end{bmatrix} \mathbf{t}_k^w \quad (3.54)$$

im obigen Beispiel angewendet. In (3.54) werden die Komponenten der räumlichen Geschwindigkeit (3.49) mithilfe der Rotationsmatrix ${}^b\mathbf{R}_k$ im Koordinatensystem $\{b\}$ dargestellt. Mithilfe der Notation wird dieser Vorgang unterstützt.

Abschließend werden die adjungierten Darstellungen der beiden homogenen Transformationen \mathbf{H}_1 und \mathbf{H}_2 betrachtet. Es gelten die Zusammenhänge

$$\mathbf{Ad}(\mathbf{H}_1)\mathbf{Ad}(\mathbf{H}_2) = \mathbf{Ad}(\mathbf{H}_1\mathbf{H}_2), \quad (3.55)$$

$$\mathbf{Ad}(\mathbf{H})^{-1} = \mathbf{Ad}(\mathbf{H}^{-1}), \quad (3.56)$$

welche durch direktes Nachrechnen bewiesen werden können.

3.5 Parametrierungen der Orientierung

Im dreidimensionalen Raum besitzt ein Starrkörper 6 Freiheitsgrade, wovon 3 Freiheitsgrade die Orientierung beschreiben. Bis hierhin ist eine Rotation immer mithilfe einer Rotationsmatrix beschrieben worden. Die Parametrierung dieser Orientierung kann aber auf unterschiedliche Weise mit unterschiedlicher Anzahl an Parametern erfolgen. Bei einer *Minimalkoordinatendarstellung* wird die Orientierung ebenfalls mit genau drei Parametern angegeben. Beispiele dafür sind die *EULER-Winkel* und die *exponentiellen Koordinaten*. Darstellungen mit mehr als drei Parametern sind *Einheitsquaternionen* (4 Parameter) und die *Achse-Winkel-Darstellung* (engl. *axis-angle representation*, 4 Parameter) sowie die Rotationsmatrix (9 Parameter, siehe (3.9)).

Alle Minimalkoordinatendarstellungen der Orientierung haben aber die gemeinsame Eigenschaft, dass sogenannte *Repräsentationssingularitäten* bei deren Verwendung auftreten.

Dabei handelt es sich um ausgezeichnete Punkte oder Bereiche im Raum der Orientierungen $SO(3)$, bei denen eine Umrechnung auf die gewählte Koordinatendarstellung nicht möglich oder nicht eindeutig ist. Auf Kosten von weiteren Parametern, d. h. unter Verwendung von Nicht-Minimalkoordinatendarstellungen, kann dieses Problem vermieden werden.

Im Folgenden werden die wichtigsten Darstellungen zusammengefasst und die Zusammenhänge mit der Rotationsmatrix aufgezeigt. Da die Zusammenhänge allgemein gelten und um die Notation einfach zu halten, werden keine Bezugskoordinatensysteme eingeführt. Die Rotationsmatrix wird folglich mit \mathbf{R} notiert und eine Rotationsachse mit \mathbf{e}_ω bezeichnet.

3.5.1 Euler-Winkel

Die EULER-Winkel werden in der industriellen Robotik am häufigsten verwendet und praktisch jede Robotersteuerung verwendet sie, um Endeffektorposen im Raum zu parametrieren. Im Allgemeinen setzen sich die EULER-Winkel aus drei Winkeln $\phi = [\varphi \ \theta \ \psi]^T \in \mathbb{R}^3$ zusammen. Die Lage eines Koordinatensystems wird durch Drehen um diese Winkel nacheinander um Koordinatenachsen $i, j, k \in \{x, y, z\}$ erzeugt, d. h. einer Drehsequenz i - j - k . Insgesamt gibt es zwölf unterschiedliche Drehsequenzen, die den gesamten Raum der Orientierungen $SO(3)$ abdecken [3.3]. Diese werden in zwei Gruppen eingeteilt:

1. Bei den *klassischen* EULER-Winkeln gilt $i = k$, d. h. die erste und dritte Rotation erfolgt um dieselbe Achse. Es gibt es sechs unterschiedliche Sequenzen, wobei die bekanntesten Sequenzen z - x - z und z - y - z sind.
2. Sind alle drei Rotationsachsen verschieden, so spricht man von TAIT-BRYAN-Winkeln oder Kardan-Winkeln. Auch hier gibt es sechs unterschiedliche Sequenzen. Die bekanntesten Vertreter dieser Gruppe sind die Roll-Nick-Gier-Winkel (engl. Roll-Pitch-Yaw angles, RPY angles), welche aus der Luftfahrt stammen. Diese haben die Sequenz z - y - x .

Im Gegensatz dazu sind jene Minimalkoordinatendarstellungen *degeneriert*, bei denen nacheinander um dieselbe Achse gedreht wird, d. h. $i = j$ oder $j = k$. Diese können nicht den gesamten $SO(3)$ beschreiben.

Ganz allgemein kann zu jeder Drehsequenz i - j - k die zugehörige Rotationsmatrix \mathbf{R} angegeben werden. Sie besteht aus einer Sequenz von drei Rotationen, d. h.

$$\mathbf{R}(\phi) = \mathbf{R}_{i,\varphi} \mathbf{R}_{j,\theta} \mathbf{R}_{k,\psi} . \quad (3.57)$$

mit den EULER-Winkeln ϕ um die Achsen $i, j, k \in \{x, y, z\}$. Weiters wird formal die Umkehrfunktion für die Sequenz i - j - k bei gegebener Matrix \mathbf{R} gemäß

$$\phi = \phi_{ijk}(\mathbf{R}) \quad (3.58)$$

eingeführt. Während (3.57) direkt ausgewertet werden kann, verbergen sich hinter (3.58) komplexere Zusammenhänge mit Fallunterscheidungen. Daher ist die Umkehrfunktion (3.58) nicht eindeutig.

Im Folgenden werden die *klassischen* EULER-Winkel mit der Drehsequenz $i = z, j = y$ und $k = z$ betrachtet. Die zugehörige Rotationsmatrix $\mathbf{R}(\phi)$ lautet, siehe (3.57)

$$\mathbf{R}(\phi) = \mathbf{R}_{z,\varphi} \mathbf{R}_{y,\theta} \mathbf{R}_{z,\psi} = \begin{bmatrix} c_\varphi c_\theta c_\psi - s_\varphi s_\psi & -c_\varphi c_\theta s_\psi - s_\varphi c_\psi & c_\varphi s_\theta \\ s_\varphi c_\theta c_\psi + c_\varphi s_\psi & -s_\varphi c_\theta s_\psi + c_\varphi c_\psi & s_\varphi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (3.59)$$

wobei die Abkürzungen s_x und c_x aus (3.18) verwendet wurden. Nun ist umgekehrt die Rotationsmatrix \mathbf{R} wie in (3.59) gegeben und die zugehörigen EULER-Winkel sind gesucht. Unter der Annahme $r_{13} \neq 0$ und $r_{23} \neq 0$ lassen sich die EULER-Winkel für $0 < \theta < \pi$ aus den Elementen der Drehmatrix (3.59) mit

$$\phi_{zyz}(\mathbf{R}) = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(r_{23}, r_{13}) \\ \text{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \\ \text{atan2}(r_{32}, -r_{31}) \end{bmatrix} \quad (3.60)$$

berechnen. Für den Fall $-\pi < \theta < 0$ folgt

$$\phi_{zyz}(\mathbf{R}) = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(-r_{23}, -r_{13}) \\ \text{atan2}\left(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \\ \text{atan2}(-r_{32}, r_{31}) \end{bmatrix}. \quad (3.61)$$

In (3.60) und (3.61) ist $\text{atan2}(y, x)$ die vorzeichenkorrekte Auswertung des Arkustangens $\arctan(y/x)$ in allen vier Quadranten.

Die RPY-Winkel verwenden die Konvention $\phi_{zyx}(\cdot)$. Statt den Winkeln (3.58) wird aber häufig die Winkelkonvention $\phi^T = [\alpha \ \beta \ \gamma]$ verwendet. Die Rotationsmatrix $\mathbf{R}(\phi)$ ergibt sich damit zu

$$\mathbf{R}(\phi) = \mathbf{R}_{z,\alpha} \mathbf{R}_{y,\beta} \mathbf{R}_{x,\gamma} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (3.62)$$

Die Lösung für $\phi_{zyx}(\cdot)$ für den Fall $-\frac{\pi}{2} < \beta < \frac{\pi}{2}$ mit $r_{32} \neq 0$ und $r_{33} \neq 0$ ist

$$\phi_{zyx}(\mathbf{R}) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \text{atan2}(r_{21}, r_{11}) \\ \text{atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \\ \text{atan2}(r_{32}, r_{33}) \end{bmatrix} \quad (3.63)$$

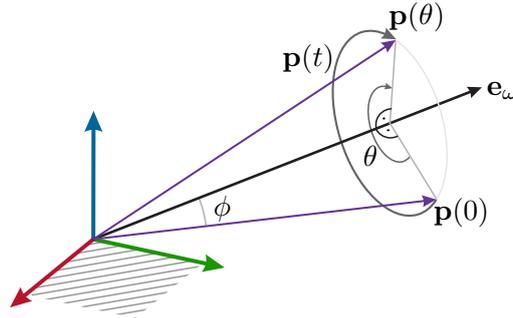


Abbildung 3.11: Drehung eines Vektors $\mathbf{p}(0)$ um den Einheitsvektor \mathbf{e}_ω um den Winkel θ [3.1].

und für $\frac{\pi}{2} < \beta < \frac{3\pi}{2}$ folgt

$$\phi_{zyx}(\mathbf{R}) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \text{atan2}(-r_{21}, -r_{11}) \\ \text{atan2}\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right) \\ \text{atan2}(-r_{32}, -r_{33}) \end{bmatrix}. \quad (3.64)$$

3.5.2 Exponentielle Koordinaten für Orientierungen

Die exponentiellen Koordinaten für die Orientierung zählen ebenfalls zu den Minimalkoordinatendarstellungen. Für die Parametrierung der Orientierung wird dabei das Produkt $\mathbf{e}_\omega \theta \in \mathbb{R}^3$ verwendet, welches aus einem (normierten) Vektor für die Rotationsachse \mathbf{e}_ω und einem Rotationswinkel θ besteht. Werden im Gegensatz dazu beide Größen separat verwendet, d. h. $\begin{bmatrix} \mathbf{e}_\omega^T & \theta \end{bmatrix}^T \in \mathbb{R}^4$, so ist dies die Achse-Winkel-Darstellung.

Als einleitendes Beispiel wird der Vektor $\mathbf{p}(0)$ in Abb. 3.11 betrachtet, welcher um den Einheitsvektor \mathbf{e}_ω gedreht wird. Wird eine konstante Drehrate von 1 rad/s in der Zeit von $t = 0$ bis $t = \theta$ angenommen, so überstreicht die Spitze des Vektors $\mathbf{p}(0)$ den Kreisbogen $\mathbf{p}(t)$. Die Geschwindigkeit der Vektorspitze ist, vgl. (3.34)

$$\dot{\mathbf{p}} = \mathbf{e}_\omega \times \mathbf{p}. \quad (3.65)$$

Mithilfe von (3.36) erhält die obige Differentialgleichung die Bauform $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, nämlich $\dot{\mathbf{p}} = \mathbf{S}(\mathbf{e}_\omega)\mathbf{p}$ mit der Anfangsbedingung $\mathbf{p}(0)$. Die Lösung dieser Differentialgleichung ist direkt mit

$$\mathbf{p}(t) = e^{\mathbf{S}(\mathbf{e}_\omega)t} \mathbf{p}(0) \quad (3.66)$$

bzw. für $t = \theta$

$$\mathbf{p}(\theta) = e^{\mathbf{S}(\mathbf{e}_\omega)\theta} \mathbf{p}(0) \quad (3.67)$$

bekannt [3.4]. Darin ist der Ausdruck $e^{\mathbf{S}(\mathbf{e}_\omega)\theta}$ die Matrixexponentialfunktion der Matrix $\mathbf{S}(\mathbf{e}_\omega)\theta$, welche in Reihenform gemäß

$$e^{\mathbf{S}(\mathbf{e}_\omega)\theta} = \mathbf{E} + \mathbf{S}(\mathbf{e}_\omega)\theta + \mathbf{S}^2(\mathbf{e}_\omega)\frac{\theta^2}{2!} + \mathbf{S}^3(\mathbf{e}_\omega)\frac{\theta^3}{3!} + \dots \quad (3.68)$$

angeschrieben werden kann. Mit den Zusammenhängen $\mathbf{S}^3(\mathbf{e}_\omega) = -\mathbf{S}(\mathbf{e}_\omega)$, $\mathbf{S}^4(\mathbf{e}_\omega) = -\mathbf{S}^2(\mathbf{e}_\omega)$, $\mathbf{S}^5(\mathbf{e}_\omega) = \mathbf{S}^3(\mathbf{e}_\omega)$, ... kann die Reihenform neu geordnet werden und es folgt

$$e^{\mathbf{S}(\mathbf{e}_\omega)\theta} = \mathbf{E} + \underbrace{\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots\right)}_{=\sin(\theta)} \mathbf{S}(\mathbf{e}_\omega) + \underbrace{\left(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} - \dots\right)}_{=1-\cos(\theta)} \mathbf{S}^2(\mathbf{e}_\omega). \quad (3.69)$$

Satz 3.1 (Rodrigues-Formel). Es sei der Vektor $\mathbf{e}_\omega\theta \in \mathbb{R}^3$ ein Produkt aus einem Einheitsvektor \mathbf{e}_ω und einem Skalar θ . Dann lautet die Matrixexponentialfunktion

$$e^{\mathbf{S}(\mathbf{e}_\omega)\theta} = \mathbf{E} + \sin(\theta)\mathbf{S}(\mathbf{e}_\omega) + (1 - \cos(\theta))\mathbf{S}^2(\mathbf{e}_\omega) \in \text{SO}(3) \quad (3.70)$$

und es beschreibt $\mathbf{R}_{\mathbf{e}_\omega,\theta} = e^{\mathbf{S}(\mathbf{e}_\omega)\theta}$ eine Drehung um die Achse \mathbf{e}_ω um den Winkel θ , vgl. (3.18). Der Zusammenhang (3.70) wird RODRIGUES-Formel genannt.

Beweis von Satz 3.1. Um zu beweisen, dass $\tilde{\mathbf{R}} := e^{\mathbf{S}(\mathbf{e}_\omega)\theta} \in \text{SO}(3)$ gilt, müssen die Eigenschaften (3.14) für $\tilde{\mathbf{R}}$ gezeigt werden. Zunächst gilt

$$\tilde{\mathbf{R}}^{-1} = \left(e^{\mathbf{S}(\mathbf{e}_\omega)\theta}\right)^{-1} = e^{-\mathbf{S}(\mathbf{e}_\omega)\theta} = e^{\mathbf{S}^T(\mathbf{e}_\omega)\theta} = \left(e^{\mathbf{S}(\mathbf{e}_\omega)\theta}\right)^T = \tilde{\mathbf{R}}^T \quad (3.71)$$

Die zweite und vorletzte Gleichung in (3.71) können mit (3.70) direkt nachgerechnet werden und die dritte Gleichung ergibt sich aus Definition 3.3. Diese Zusammenhänge zeigen $\tilde{\mathbf{R}}^{-1} = \tilde{\mathbf{R}}^T$, womit $\tilde{\mathbf{R}}^T\tilde{\mathbf{R}} = \mathbf{E}$ gezeigt ist. Daraus folgt auch $\det(\tilde{\mathbf{R}}) = \pm 1$. Aus der Stetigkeit der Determinante als Funktion der Matrixeinträge, der Stetigkeit der Matrixexponentialfunktion und der Relation $\det(e^{\mathbf{0}}) = 1$, folgt $\det(\tilde{\mathbf{R}}) = +1$. \square

3.5.3 Einheitsquaternionen

Im Folgenden wird die Rotationsmatrix $\mathbf{R} = e^{\mathbf{S}(\mathbf{e}_\omega)\theta}$ mit $\|\mathbf{e}_\omega\| = 1$ und $\theta \in [0, \pi]$ betrachtet.

Definition 3.5 (Einheitsquaternion). Eine Einheitsquaternion definiert als ein Vektor $\tilde{\mathbf{q}} \in S^3 \subset \mathbb{R}^4$ in der Form

$$\tilde{\mathbf{q}} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{e}_\omega \sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \quad (3.72)$$

welcher die Normierungsbedingung $\|\tilde{\mathbf{q}}\| = 1$ erfüllt. Damit ist eine Einheitsquaternion $\tilde{\mathbf{q}}$ ein Punkt auf der Oberfläche einer dreidimensionalen Kugel $S^3 = \{\mathbf{x} \in \mathbb{R}^4 \mid \|\mathbf{x}\| = 1\}$ im vierdimensionalen Raum \mathbb{R}^4 .

Durch die Normierungsbedingung können nur drei Koordinaten unabhängig von einander gewählt werden. Der Zusammenhang zur Rotationsmatrix \mathbf{R} wird über die RODRIGUES-Formel (3.70) mit dem Drehwinkel $\theta = 2 \arccos(q_0)$ und der Rotationsachse

$\mathbf{e}_\omega = [q_1 \ q_2 \ q_3]^T / \sin(\theta/2)$ gemäß

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.73)$$

hergestellt⁴. Hier sei erwähnt, dass sowohl das Quaternion $\tilde{\mathbf{q}} \in S^3$ als auch der „gegenüberliegende“ Punkt $-\tilde{\mathbf{q}} \in S^3$ auf exakt dieselbe Rotationsmatrix \mathbf{R} führen. Dies kann durch direktes Nachrechnen bestätigt werden.

Der umgekehrte Weg, d. h. von einer Rotationsmatrix in der Form (3.9) zu einer Einheitsquaternion der Form (3.72) zu gelangen, erfolgt durch

$$q_0 = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}} \quad (3.74a)$$

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{4q_0} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \quad (3.74b)$$

Für eine Herleitung sei auf die Literatur am Ende des Kapitels verwiesen.

⁴Hierbei werden für $\theta = 2 \arccos(q_0)$ die trigonometrischen Identitäten $\sin(\theta) = 2q_0 \sqrt{1 - q_0^2}$, $\sin(\theta/2) = \sqrt{1 - q_0^2}$ und $\cos(\theta) = 2q_0^2 - 1$ verwendet.

3.6 Literatur

- [3.1] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [3.2] R. M. Murray, Z. Li und S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, 1994. Adresse: <http://www.cds.caltech.edu/>.
- [3.3] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [3.4] A. Kugi, *Skriptum zur VU Automatisierung (WS 2023/2024)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. Adresse: <https://www.acin.tuwien.ac.at/bachelor/automatisierung/>.

4 Roboterkinematik

Die *Kinematik* (engl. *kinematics*, altgr. *kinema*, dt. *Bewegung*) beschäftigt sich ausschließlich geometrisch mit der Bewegung von Körpern unter Verwendung der Größen Zeit, Position, Geschwindigkeit und Beschleunigung. Dabei bleiben wirkende Kräfte sowie die Massen und Massenträgheiten der Körper unberücksichtigt – diese sind Teil der *Dynamik* (engl. *dynamics*).

Dieses Kapitel behandelt die Geometrie von Robotern mit serieller und paralleler Kinematik. Dabei werden die fundamentalen Begriffe der Vorwärtskinematik, inversen Kinematik, differentiellen Kinematik sowie der kinematischen Redundanz eingeführt und mathematisch aufgearbeitet.

4.1 Serielle Kinematik

Im ersten Abschnitt dieses Kapitels werden die oben erwähnten Begriffe für Roboter mit serieller Kinematik, d. h. mit offener kinematischer Kette, behandelt.

4.1.1 Vorwärtskinematik

Die *Vorwärtskinematik* eines Roboters

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \quad (4.1)$$

beschreibt die Pose des Endeffektors $\mathbf{x}_e \in \mathbb{R}^m$ des Roboters im m -dimensionalen Aufgabenraum als Funktion der Konfiguration $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T \in \mathbb{R}^n$. Dies ist ein fundamentaler Zusammenhang eines Roboters, da über die Vorwärtskinematik die (im Allgemeinen nichtlineare) Beziehung zwischen den Koordinaten des Konfigurationsraumes und des Aufgabenraumes hergestellt wird.

In Abb. 4.1 ist schematisch eine allgemeine serielle kinematische Kette bestehend aus n Rotations- und Schubgelenken mit dem Inertialkoordinatensystem $\{w\}$ und dem Endeffektorkoordinatensystem $\{e\}$ dargestellt. Gemäß (3.19) wird die homogene Transformation des Endeffektorkoordinatensystems $\{e\}$ in Bezug auf das Inertialkoordinatensystem $\{w\}$ durch

$${}^w\mathbf{H}_e(\mathbf{q}) = \begin{bmatrix} {}^w\mathbf{R}_e(\mathbf{q}) & {}^w\mathbf{d}_e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.2)$$

als Funktion der Konfiguration \mathbf{q} beschrieben. Weiters wird jedes Glied des Roboters mit einem Koordinatensystem $\{i\}$, $i = 0, \dots, n$, ausgestattet. Dabei ist es von Vorteil, jeweils eine Achse der Koordinatensysteme in die jeweilige Rotations- bzw. Schubachse zu legen. Das spezielle Koordinatensystem für die *Roboterbasis* $\{b\}$ wird mit $\{b\} = \{0\}$ identifiziert.

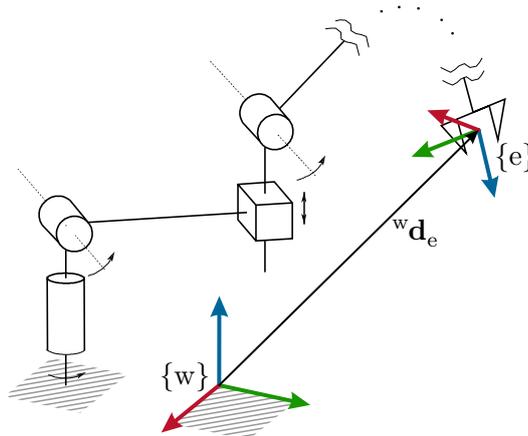


Abbildung 4.1: Schematische Darstellung einer seriellen kinematischen Kette [4.1].

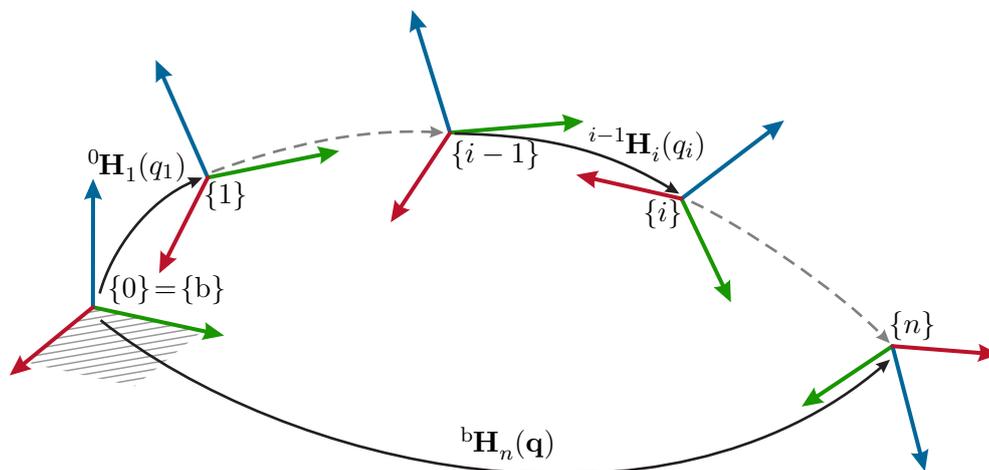


Abbildung 4.2: Koordinatensysteme eines Roboters mit serieller Kinematik [4.1].

Wie in Abb. 4.2 gezeigt, entsteht dadurch eine Kette von Koordinatensystemen, welche mithilfe der homogenen Transformationen in der Form

$${}^b\mathbf{H}_n(\mathbf{q}) = {}^b\mathbf{H}_1(q_1) {}^1\mathbf{H}_2(q_2) \cdots {}^{i-1}\mathbf{H}_i(q_i) \cdots {}^{n-1}\mathbf{H}_n(q_n) \quad (4.3)$$

beschrieben wird, siehe (3.31). Jede der Transformationen ${}^{i-1}\mathbf{H}_i(q_i)$, $i = 1, \dots, n$, in (4.3) beschreibt dabei die Kinematik von genau einem Freiheitsgrad q_i und ist aus den elementaren Transformationen für Rotation und Translation zusammengesetzt. Weiters werden die (üblicherweise konstanten) Transformationen ${}^w\mathbf{H}_b$ und ${}^n\mathbf{H}_e$ zur seriellen kinematischen Kette (4.3) hinzugefügt, welche das Koordinatensystem der Roboterbasis $\{b\}$ in Bezug auf das Inertialkoordinatensystem $\{w\}$ bzw. das Endeffektorkoordinatensystem $\{e\}$ in Bezug auf das Koordinatensystem des letzten Robotergliedes $\{n\}$ beschreibt. Damit folgt die Beschreibung der Vorwärtskinematik als homogene Transformation zu

$${}^w\mathbf{H}_e(\mathbf{q}) = {}^w\mathbf{H}_b {}^b\mathbf{H}_n(\mathbf{q}) {}^n\mathbf{H}_e = \begin{bmatrix} {}^w\mathbf{R}_e(\mathbf{q}) & {}^w\mathbf{d}_e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix}. \quad (4.4)$$

Um die Vorwärtskinematik in der Form (4.1) zu erhalten, wird ${}^w\mathbf{H}_e(\mathbf{q})$ schließlich in Form der $m = 6$ Koordinaten des Aufgabenraums

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} {}^w\mathbf{d}_e(\mathbf{q}) \\ \phi({}^w\mathbf{R}_e(\mathbf{q})) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_e \\ \phi_e \end{bmatrix} \quad (4.5)$$

dargestellt. In (4.5) setzt sich die Pose des Endeffektors \mathbf{x}_e im Aufgabenraum aus dem Vektor ${}^w\mathbf{d}_e$ zum Ursprung des Endeffektorkoordinatensystems und einer Minimalkoordinatendarstellung der Orientierung $\phi(\cdot)$ zusammen. Weiters wurden die Kurzbezeichnungen \mathbf{p}_e und ϕ_e für die Position und Orientierung des Endeffektors im Aufgabenraum eingeführt. Die Vorwärtskinematik wurde in (4.5) mit $m = 6$ eingeführt und kann aber durch Weglassen von nicht benötigten Koordinaten mit $m < 6$ definiert werden. Wird beispielsweise für eine Aufgabe nur die Position des Endeffektors im dreidimensionalen Raum relevant, so können die Koordinaten der Orientierung ϕ_e weggelassen werden und es ist $m = 3$. Für eine Aufgabe im *zweidimensionalen* Raum ($m = 3$) ist $\mathbf{p}_e \in \mathbb{R}^2$ und $\phi_e \in \mathbb{R}$.

Beispiel 4.1 (6-Achs-Roboter mit sphärischem Handgelenk). In diesem Beispiel wird die Vorwärtskinematik des 6-Achs-Roboters mit sphärischem Handgelenk hergeleitet. Die kinematische Kette mit den zugehörigen Koordinatensystemen $\{i\}$, $i = 0, \dots, 6$, ist schematisch in Abb. 4.3 dargestellt. Weiters ist das Inertialkoordinatensystem $\{w\}$ und das Endeffektorkoordinatensystem $\{e\}$ eingezeichnet. Die Koordinatensysteme $\{i\}$ liegen körperfest im Glied i des Roboters. Die sechs Rotationsgelenke mit den Koordinaten $\mathbf{q}^T = [q_1 \ q_2 \ \dots \ q_6]$ verbinden jeweils das Glied $i - 1$ mit dem Glied i . Aus Abb. 4.3 können die geometrischen Zusammenhänge als homogene Transformationen abgelesen werden, welche sich aus den elementaren Translationen und Rotationen

$$\begin{aligned} {}^w\mathbf{H}_b &= \mathbf{E} & {}^3\mathbf{H}_4(q_4) &= \mathbf{H}_{Tz,d_4} \mathbf{H}_{Rz,q_4} \\ {}^b\mathbf{H}_0 &= \mathbf{E} & {}^4\mathbf{H}_5(q_5) &= \mathbf{H}_{Ry,q_5} \\ {}^0\mathbf{H}_1(q_1) &= \mathbf{H}_{Rz,q_1} & {}^5\mathbf{H}_6(q_6) &= \mathbf{H}_{Tz,d_6} \mathbf{H}_{Rz,q_6} \\ {}^1\mathbf{H}_2(q_2) &= \mathbf{H}_{Tz,d_1} \mathbf{H}_{Ry,q_2} & {}^6\mathbf{H}_e &= \mathbf{E} \\ {}^2\mathbf{H}_3(q_3) &= \mathbf{H}_{Tz,d_2} \mathbf{H}_{Ry,q_3} & & \end{aligned}$$

zusammensetzen. Analog zu (4.3) und (4.4) folgt mit

$$\begin{aligned} {}^w\mathbf{H}_e(\mathbf{q}) &= \begin{bmatrix} {}^w\mathbf{R}_e(\mathbf{q}) & {}^w\mathbf{d}_e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \\ &= {}^w\mathbf{H}_b {}^b\mathbf{H}_0 {}^0\mathbf{H}_1(q_1) {}^1\mathbf{H}_2(q_2) {}^2\mathbf{H}_3(q_3) {}^3\mathbf{H}_4(q_4) {}^4\mathbf{H}_5(q_5) {}^5\mathbf{H}_6(q_6) {}^6\mathbf{H}_e \quad (4.6) \end{aligned}$$

die Vorwärtskinematik gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} {}^w\mathbf{d}_e(\mathbf{q}) \\ \phi_{zyz}({}^w\mathbf{R}_e(\mathbf{q})) \end{bmatrix}, \quad (4.7)$$

wobei mit $\phi_{zyz}(\cdot)$ die klassischen EULER-Winkel zur Parametrierung der Orientierung gewählt wurden.

Das obige Ergebnis kann durch Einsetzen einfacher Konfigurationen \mathbf{q} überprüft werden. Mit $\mathbf{q} = \mathbf{0}$ steht der 6-Achs-Roboter aufrecht und das Endeffektorkoordinatensystem $\{e\}$ liegt auf der z -Achse des Inertialkoordinatensystems $\{w\}$ und hat die selbe Orientierung wie $\{w\}$. Die Vorwärtskinematik für $\mathbf{q}_0 = \mathbf{0}$ ergibt

$${}^w\mathbf{H}_e(\mathbf{q}_0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 + d_2 + d_4 + d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8a)$$

$$\mathbf{x}_e^T = \begin{bmatrix} 0 & 0 & d_1 + d_2 + d_4 + d_6 & 0 & 0 & 0 \end{bmatrix} \quad (4.8b)$$

Für $\mathbf{q}_1^T = \begin{bmatrix} \frac{\pi}{2} & 0 & \frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}$ folgt

$${}^w\mathbf{H}_e(\mathbf{q}_1) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & d_4 + d_6 \\ -1 & 0 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9a)$$

$$\mathbf{x}_e^T = \begin{bmatrix} 0 & d_4 + d_6 & d_1 + d_2 & \frac{\pi}{2} & \frac{\pi}{2} & 0 \end{bmatrix}. \quad (4.9b)$$

Für diesen Fall ist das Gelenk 1 um $\frac{\pi}{2}$ gedreht und das Gelenk 3 um $\frac{\pi}{2}$, womit die Glieder 1 und 2 des Roboters parallel zur z -Achse des Inertialkoordinatensystems $\{w\}$ sind und der restliche Arm in Richtung der y -Achse von $\{w\}$ zeigt. Das Endeffektorkoordinatensystem $\{e\}$ ist bezüglich $\{w\}$ entsprechend gedreht. Die z -Achse von $\{e\}$ ist parallel zur y -Achse von $\{w\}$ und die x -Achse von $\{e\}$ ist antiparallel zur z -Achse des Inertialkoordinatensystems $\{w\}$.

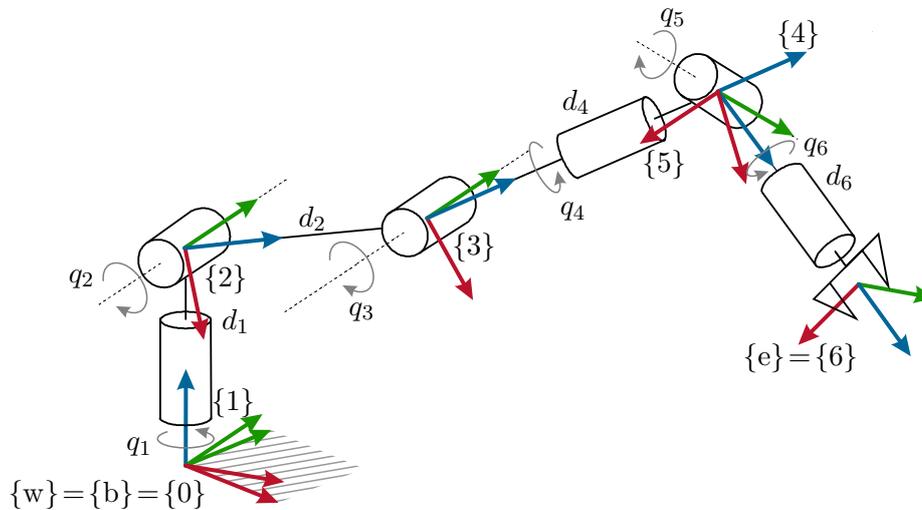


Abbildung 4.3: 6-Achs-Roboter mit sphärischem Handgelenk [4.1].

4.1.2 Kinematikrepräsentationen

In diesem Abschnitt werden zwei Methoden für die *systematische* Darstellung der Kinematik eines Roboters mit serieller Kinematik vorgestellt. Einerseits ist dies die sogenannte *Denavit-Hartenberg-Konvention* (DH-Konvention) [4.2], welche 1955 für die einheitliche Beschreibung der Koordinatensysteme von mechanischen Systemen eingeführt wurde und bis heute speziell bei Industrierobotern verwendet wird. Seit der Einführungen wurden allerdings eine Reihe von Abwandlungen der DH-Konvention publiziert, welche aber meist trotzdem als DH-Konvention bezeichnet werden. Andererseits wird das *Universal Robot Description Format* (URDF, dt. universelles Format zum Beschreiben von Robotern) vorgestellt als ein aktueller Ansatz für die Beschreibung von beliebigen seriellen und baumartigen kinematischen Ketten.

Denavit-Hartenberg-Konvention

Im vorigen Abschnitt wurde die Vorwärtskinematik eines Roboters mithilfe einer Reihe von homogenen Transformationen oder Schraubachsen beschrieben. Dabei wurden die Koordinatensysteme frei gewählt und die Lage dieser geometrischen Objekte stets mit allen notwendigen sechs Freiheitsgraden bzw. Parametern beschrieben. Durch geeignete Wahl der Koordinatensysteme für die Roboterglieder kann mithilfe der DH-Konvention die kinematische Beschreibung eines Roboters systematisch erfolgen und vereinfacht werden. Bei der DH-Konvention sind nur vier Parameter notwendig, um die Lage der Koordinatensysteme von zwei Gliedern zu beschreiben, welche durch ein Gelenk verbunden sind.

In diesem Skriptum wird die DH-Konvention nach Craig [4.3] verwendet und vorgestellt. Die Platzierung der Koordinatensysteme erfolgt nach einfachen Regeln und wird im

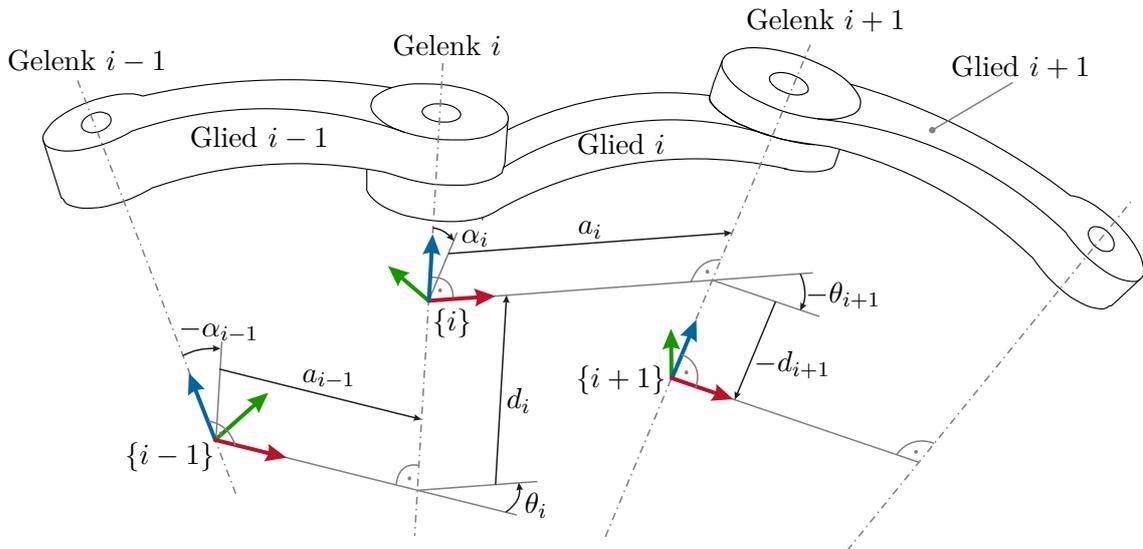


Abbildung 4.4: Schematische Darstellung von drei Gliedern und drei Gelenken. Die Koordinatensysteme $\{k\}$, $k = i - 1, i, i + 1$ liegen körperfest im zugehörigen Glied k und wurden anhand den Regeln der DH-Konvention mit den Parametern α_k , a_k , θ_k und d_k , $k = i - 1, i, i + 1$ gewählt. Parameter mit negativem Vorzeichen zeigen eine umgekehrte Zählrichtung an (angepasst von [4.1, 4.3]).

Folgendes anhand von Abb. 4.4 erklärt. Im ersten Schritt wird die z -Achse jedes Koordinatensystems $\{i\}$ in die jeweilige Gelenksachse gelegt, wobei die positive Drehrichtung des Gelenks anhand der Rechten-Hand-Regel die Lage der z -Achsen eindeutig festlegt. Jedes Koordinatensystem $\{i\}$ liegt körperfest im zugehörigen Glied i . Im nächsten Schritt werden zwei aufeinanderfolgende Gelenksachsen $i - 1$ und i mit einer Linie so verbunden, dass diese die beiden Achsen normal schneidet (Normalenlinie). Auf den Schnittpunkt zwischen der Normalenlinie und der Gelenksachse $i - 1$ wird der Ursprung des Koordinatensystems $\{i - 1\}$ gelegt. Die x -Achse von $\{i - 1\}$ wird auf die Normalenlinie in Richtung der Gelenksachse i gelegt und die y -Achse ergibt sich aus dem Kreuzprodukt gemäß, vgl. Definition 3.1

$$\mathbf{e}_{y_{i-1}} = \mathbf{e}_{z_{i-1}} \times \mathbf{e}_{x_{i-1}}. \quad (4.10)$$

Durch dieses Vorgehen werden die Koordinatensysteme in Abb. 4.4 eindeutig gelegt und durch vier Parameter spezifiziert:

- α_{i-1} : Verdrehung der z -Achse von $\{i\}$ gegenüber $\{i - 1\}$ als Rotation um die x -Achse von $\{i - 1\}$.
- a_{i-1} : Länge der Normalenlinie
- θ_i : Verdrehung der x -Achse von $\{i\}$ gegenüber $\{i - 1\}$ um die Gelenksachse i .
- d_i : Verschiebung zwischen dem Schnittpunkt der Normalenlinie mit der Gelenksachse i und dem Ursprung von $\{i\}$. Positive Zählrichtung in Richtung der z -Achse.

In zwei Spezialfällen ist die Normalenlinie zwischen zwei Gelenksachsen nicht eindeutig definiert. Diese werden wie folgt behandelt:

1. Zwei aufeinanderfolgende Gelenksachsen schneiden sich: In diesem Fall ist $a_{i-1} = 0$ und die x -Achse von $\{i-1\}$ wird so gewählt, dass diese senkrecht auf die von den z -Achsen von $\{i-1\}$ und $\{i\}$ aufgespannte Ebene steht.
2. Zwei aufeinanderfolgende Gelenksachsen sind parallel: In diesem Fall kann die Normalenlinie frei gewählt und auf eine günstige Position gelegt werden, sodass weitere Parameter zu Null werden.

Die obigen Parameter α_{i-1} , a_{i-1} , θ_i und d_i beschreiben vollständig die Lage des Koordinatensystems $\{i\}$ gegenüber dem Koordinatensystem $\{i-1\}$. Diese werden in die homogene Transformation ${}^{i-1}\mathbf{H}_i$ gemäß

$${}^{i-1}\mathbf{H}_i = \mathbf{H}_{R_x, \alpha_{i-1}} \mathbf{H}_{T_x, a_{i-1}} \mathbf{H}_{R_z, \theta_i} \mathbf{H}_{T_z, d_i}$$

$$= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i) \cos(\alpha_{i-1}) & \cos(\theta_i) \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i \sin(\alpha_{i-1}) \\ \sin(\theta_i) \sin(\alpha_{i-1}) & \cos(\theta_i) \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

eingesetzt und in (4.3) zum Berechnen der Vorwärtskinematik verwendet.

In (4.11) ist ersichtlich, dass mit den Koordinatensystemen $\{i\}$ nach der DH-Konvention für die homogene Transformation ${}^{i-1}\mathbf{H}_i$ nur vier Parameter benötigt werden. Dies ist eine nützliche Eigenschaft, wenn wenig Rechenleistung in der Robotersteuerung zur Verfügung steht, wie dies in der Zeit der Veröffentlichung dieser Konvention der Fall war. Dieser Aspekt ist in heutigen Robotersteuerungen nicht mehr relevant. Durch die reduzierte Zahl an Parametern können die körperfesten Koordinatensysteme nicht frei gewählt werden und kommen in manchen Fällen an ungünstigen Stellen oder auch außerhalb des Gliedes zu liegen.

Universal Robot Description Format (URDF)

Das URDF ist ein Dateiformat, welches in ROS¹ (Robot Operating System) für die Beschreibung der Kinematik, Geometrie und Dynamik (siehe Kapitel 6 „Dynamik“) von seriellen und baumartigen kinematischen Ketten verwendet wird. In einer URDF-Datei werden die Glieder und Gelenke eines Roboters beschrieben, vgl. Abb. 4.5:

Glieder Jedes Roboterglied ist ein Starrkörper und damit eine bewegte Masse. Für jedes Glied kann entsprechend mit `origin` die Lage des Massenschwerpunktes, mit `mass` die Masse und mit `inertia` das Massenträgheitsmoment (siehe Kapitel 6 „Dynamik“) angegeben werden.

¹ROS ist eine Open-Source-Middleware, welche in komplexen Robotersystemen eingesetzt wird. Eine Middleware wird verwendet, um die Komplexität und Heterogenität von Hardware und den Anwendungen zu verwalten. Weiters kann damit die Softwareentwicklung durch Modularität vereinfacht und die Integration von Aktoren und Sensoren erleichtert werden, indem die Low-Level-Funktionen verborgen und abstrahiert werden.

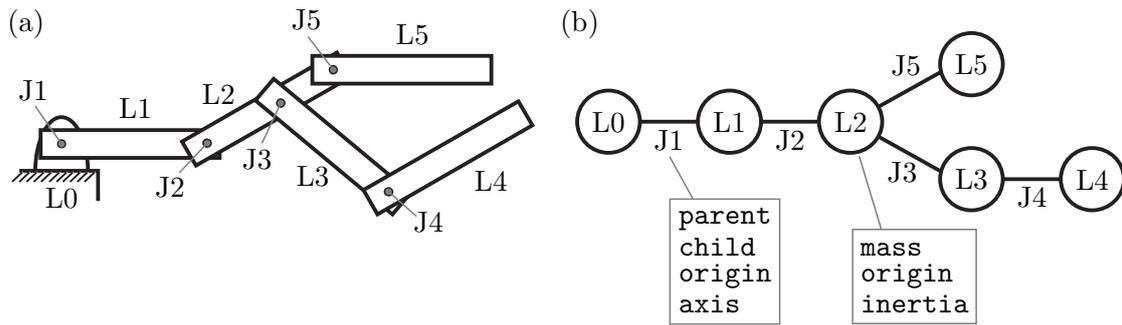


Abbildung 4.5: Baumartig aufgebauter Roboter mit sechs Gliedern L_0, \dots, L_5 und fünf Gelenken J_1, \dots, J_5 [4.4]. (a) Schematische Zeichnung des Roboters, (b) Darstellung des Roboters als Baum mit den Gliedern als Knoten und den Gelenken als Kanten.

Gelenke Ein Gelenk in einer URDF-Datei verbindet zwei Glieder, welche mit `parent` und `child` spezifiziert werden. Weiters wird mit `origin` die Lage des `child`-Gliedes relativ zum `parent`-Glieder angegeben und mit `axis` die Rotations- bzw. Schubachse definiert.

Ein Beispiel für einen baumartig aufgebauten Roboter mit sechs Gliedern L_0, \dots, L_5 und fünf Gelenken J_1, \dots, J_5 zeigt Abb. 4.5.

4.1.3 Differentielle Kinematik

Der vorige Abschnitt 4.1.1 zur Vorwärtskinematik hat den Zusammenhang zwischen den Punkten im Konfigurationsraum $\mathbf{q} \in \mathbb{R}^n$ und dem Aufgabenraum eines Roboters $\mathbf{x}_e \in \mathbb{R}^m$ hergestellt. Die *differentielle Kinematik* beschäftigt sich nun mit den *Geschwindigkeiten* im Konfigurations- und Aufgabenraum und den mathematischen Beziehungen zwischen den Räumen. Die wichtigsten Werkzeuge dabei sind die *analytische* und die *geometrische Jacobi-Matrix*, sowie die *körperbezogene Jacobi-Matrix* (auch *Body-Jacobi-Matrix*, engl. *body Jacobian*). Zusätzlich können mit diesen Matrizen kinematische Eigenschaften von Robotern analysiert werden, wie z. B. Singularitäten und Redundanzen, und auch Lösungen der inversen Kinematik für Roboterbewegungen im Aufgabenraum gefunden werden.

Analytische Jacobi-Matrix

Die Pose des Endeffektors wird durch den Vektor \mathbf{x}_e im m -dimensionalen Aufgabenraum beschrieben. Die Geschwindigkeit des Endeffektors $\dot{\mathbf{x}}_e$ wird durch zeitliches Ableiten der Vorwärtskinematik (4.5) gemäß

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix} = \underbrace{\frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q})}_{\mathbf{J}_a(\mathbf{q})} \dot{\mathbf{q}} = \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}} \quad (4.12)$$

mit der *analytischen Jacobi-Matrix* (engl. *analytical Jacobian*)

$$\mathbf{J}_a(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) \in \mathbb{R}^{m \times n} \quad (4.13)$$

berechnet. In (4.12) wurde die translatorische Endeffektorgeschwindigkeit $\dot{\mathbf{p}}_e$ und die rotatorische Endeffektorgeschwindigkeit in Minimalkoordinatendarstellung $\dot{\boldsymbol{\phi}}_e$ eingeführt. Die analytische Jacobi-Matrix gibt also an, welche zeitlichen Änderungsraten sich in den Koordinaten des Aufgabenraumes $\dot{\mathbf{x}}_e$ zufolge der Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$ ergeben. Direktes Berechnen von $\partial \mathbf{f}(\mathbf{q}) / \partial \mathbf{q}$ ist aber für die Koordinaten der Orientierung meist nicht möglich, weil $\boldsymbol{\phi}_e(\mathbf{q})$ nicht in Form von geschlossenen Ausdrücken vorliegt, sondern über die Rotationsmatrix ${}^w\mathbf{R}_e(\mathbf{q})$ berechnet werden muss, siehe (3.58) bzw. (3.60) und (3.61).

Geometrische Jacobi-Matrix

Die momentanen Endeffektorgeschwindigkeiten können auch in Form der translatorischen Geschwindigkeit $\dot{\mathbf{p}}_e = {}^w\dot{\mathbf{d}}_e$ und dem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e = {}^w\boldsymbol{\omega}_e$, beschrieben im Inertialkoordinatensystem $\{w\}$, ausgedrückt werden. Identisch zu (4.12) berechnet sich die translatorische Geschwindigkeit gemäß

$$\dot{\mathbf{p}}_e = {}^w\dot{\mathbf{d}}_e(\mathbf{q}) = \sum_{i=1}^n \left(\frac{\partial}{\partial q_i} {}^w\mathbf{d}_e(\mathbf{q}) \right) \dot{q}_i, \quad (4.14)$$

während sich die schiefsymmetrische Matrix zum Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ mithilfe von (3.37) zu

$$\mathbf{S}(\boldsymbol{\omega}_e) = {}^w\dot{\mathbf{R}}_e(\mathbf{q}) {}^w\mathbf{R}_e^T = \sum_{i=1}^n \left(\frac{\partial}{\partial q_i} {}^w\mathbf{R}_e(\mathbf{q}) \right) {}^w\mathbf{R}_e^T \dot{q}_i \quad (4.15)$$

ergibt. Bei den obigen Gleichungen (4.14) und (4.15) wird sichtbar, dass sich die Geschwindigkeiten $\dot{\mathbf{p}}_e$ und $\boldsymbol{\omega}_e$ als Produkt einer Matrix, welche nur von \mathbf{q} abhängig ist, und den Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$ zusammensetzen. D. h. diese Gleichungen sind lediglich linear in den Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$. Folglich kann Gleichung (4.14) in der Form

$$\dot{\mathbf{p}}_e = \frac{\partial \mathbf{p}_e}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{v,1} & \mathbf{J}_{v,2} & \cdots & \mathbf{J}_{v,n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \sum_{i=1}^n \mathbf{J}_{v,i} \dot{q}_i = \mathbf{J}_v(\mathbf{q}) \dot{\mathbf{q}} \quad (4.16)$$

mit einer Matrix $\mathbf{J}_v(\mathbf{q})$

$$\mathbf{J}_v(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{v,1} & \mathbf{J}_{v,2} & \cdots & \mathbf{J}_{v,n} \end{bmatrix} \in \mathbb{R}^{3 \times n} \quad (4.17)$$

dargestellt werden. Ausgehend von (4.15) wird die Matrix $\mathbf{J}_\omega(\mathbf{q})$ gemäß

$$\boldsymbol{\omega}_e = \frac{\partial \boldsymbol{\omega}_e}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{\omega,1} & \mathbf{J}_{\omega,2} & \cdots & \mathbf{J}_{\omega,n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \sum_{i=1}^n \mathbf{J}_{\omega,i} \dot{q}_i = \mathbf{J}_\omega(\mathbf{q}) \dot{\mathbf{q}} \quad (4.18)$$

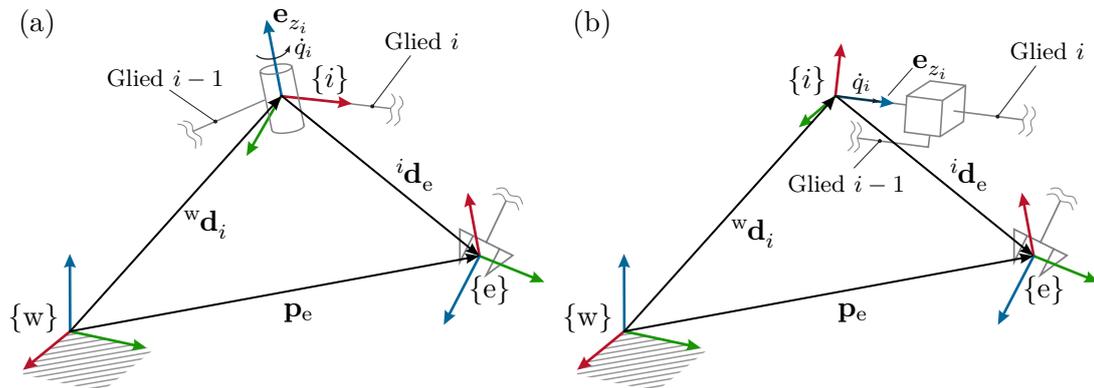


Abbildung 4.6: Geometrische Relationen für ein einzelnes Gelenk i mit der Geschwindigkeit \dot{q}_i , (a) Rotationsgelenk, (b) Schubgelenk.

mit

$$\mathbf{J}_\omega(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{\omega,1} & \mathbf{J}_{\omega,2} & \cdots & \mathbf{J}_{\omega,n} \end{bmatrix} \in \mathbb{R}^{3 \times n} \quad (4.19)$$

eingeführt. Die obigen Matrizen werden schließlich zur *geometrischen Jacobi-Matrix* (engl. *geometric Jacobian*) $\mathbf{J}_g(\mathbf{q}) \in \mathbb{R}^{m \times n}$ zu

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_g(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_v(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \\ \frac{\partial \boldsymbol{\omega}_e}{\partial \mathbf{q}} \end{bmatrix} \dot{\mathbf{q}} \quad (4.20)$$

zusammengefasst. Die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ unterscheidet sich bei den rotatorischen Geschwindigkeiten von der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$. Während bei der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ in (4.13) implizit eine Minimalkoordinatendarstellung $\phi_e = \phi_{ijk}({}^w\mathbf{R}_e)$ gewählt wurde, siehe (3.58), ist hingegen die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ aus (4.20) unabhängig von der Wahl der Koordinatendarstellung der Orientierung.

Mithilfe von (4.20) mit (4.16) und (4.18) kann die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ aus analytischen Zusammenhängen aus den Geschwindigkeiten $\dot{\mathbf{p}}_e$ und $\boldsymbol{\omega}_e$ berechnet werden. Im Folgenden wird ein weiterer Zugang vorgestellt, in welchem die einzelnen Spaltenvektoren der (Teil-)Jacobi-Matrizen (4.17) und (4.19) aus geometrischen Überlegungen hergeleitet werden. Aus (4.16) bzw. (4.18) ist ersichtlich, dass sich die Endeffektorgeschwindigkeiten als Summe von n Geschwindigkeitskomponenten zufolge einzelner Gelenke i mit den Geschwindigkeiten \dot{q}_i , $i = 1, \dots, n$, zusammensetzen lässt. Anhand von Abb. 4.6 werden diese Geschwindigkeitskomponenten für einzelne Rotations- und Schubgelenke berechnet. Dabei wird angenommen, dass nur das Gelenk i momentan (differentiell) mit der Geschwindigkeit \dot{q}_i in Bewegung ist, während alle anderen Gelenke fest sind, d. h. es ist momentan $\mathbf{q} = \text{const.}$

Rotationsgelenk Die geometrischen Relationen für ein einzelnes Rotationsgelenk i mit der Geschwindigkeit \dot{q}_i sind in Abb. 4.6a mit dem Endeffektorkoordinatensystem $\{e\}$ dargestellt. Darin ist ersichtlich, dass

$$\mathbf{p}_e = {}^w\mathbf{d}_i + {}^w\mathbf{R}_i {}^i\mathbf{d}_e \quad (4.21)$$

gilt. Hier ist zu beachten, dass die Verschiebungsvektoren ${}^w\mathbf{d}_i$ und ${}^i\mathbf{d}_e$ im Allgemeinen Funktionen von \mathbf{q} sind. Aufgrund der obigen Annahme ($\mathbf{q} = \text{const}$) sind diese aber konstant und das Rotationsgelenk führt lediglich eine Rotationsbewegung des Vektors ${}^i\mathbf{d}_e$ um die Achse \mathbf{e}_{z_i} zufolge \dot{q}_i gemäß

$$\begin{aligned}\dot{\mathbf{p}}_e &= \mathbf{0} + {}^w\dot{\mathbf{R}}_i {}^i\mathbf{d}_e = \mathbf{S}({}^w\boldsymbol{\omega}_i) {}^w\mathbf{R}_i \dot{{}^i\mathbf{d}}_e \\ &= {}^w\boldsymbol{\omega}_i \times {}^w\mathbf{d}_e^i = \dot{q}_i {}^w\mathbf{e}_{z_i} \times {}^w\mathbf{d}_e^i = \dot{q}_i {}^w\mathbf{e}_{z_i} \times (\mathbf{p}_e - {}^w\mathbf{d}_i)\end{aligned}\quad (4.22)$$

aus. Daraus folgt die Jacobi-Matrix für die translatorische Geschwindigkeitskomponente zu $\mathbf{J}_{v,i} = {}^w\mathbf{e}_{z_i} \times (\mathbf{p}_e - {}^w\mathbf{d}_i)$.

Der Vektor der Drehwinkelgeschwindigkeit des Endeffektors $\boldsymbol{\omega}_e$ kann auch als Summe von Drehwinkelgeschwindigkeiten zufolge jeder Achse – aufsummiert im Koordinatensystem $\{w\}$ – gesehen werden, vgl. (4.18)

$$\boldsymbol{\omega}_e = {}^w\boldsymbol{\omega}_1^w + {}^w\boldsymbol{\omega}_2^1 + \dots + {}^w\boldsymbol{\omega}_n^{n-1} = \sum_{i=1}^n {}^w\boldsymbol{\omega}_i^{i-1} = \sum_{i=1}^n \mathbf{J}_{\omega,i} \dot{q}_i \quad (4.23)$$

Das Rotationsgelenk i liefert den Beitrag ${}^w\boldsymbol{\omega}_i^{i-1}$ in der Form

$${}^w\boldsymbol{\omega}_i^{i-1} = \dot{q}_i {}^w\mathbf{e}_{z_i}, \quad (4.24)$$

womit die Jacobi-Matrix der rotatorischen Komponente $\mathbf{J}_{\omega,i} = {}^w\mathbf{e}_{z_i}$ gefunden ist.

Schubgelenk Das Schubgelenk i in Abb. 4.6b verschiebt das Glied i entlang der Achse ${}^w\mathbf{e}_{z_i}$ mit der Geschwindigkeit \dot{q}_i , ohne eine Drehung hervorzurufen. Aus (4.21) folgt daher die Endeffektorgeschwindigkeit

$$\dot{\mathbf{p}}_e = {}^w\dot{\mathbf{d}}_i = \dot{q}_i {}^w\mathbf{e}_{z_i} \quad (4.25)$$

und aus (4.23) folgt

$${}^w\boldsymbol{\omega}_i^{i-1} = \mathbf{0}. \quad (4.26)$$

Damit sind die jeweiligen Jacobi-Matrizen in der Form $\mathbf{J}_{v,i} = {}^w\mathbf{e}_{z_i}$ und $\mathbf{J}_{\omega,i} = \mathbf{0}$ gefunden.

Zusammengefasst können die Spaltenvektoren der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ in (4.20) direkt in der Form

$$\begin{bmatrix} \mathbf{J}_{v,i} \\ \mathbf{J}_{\omega,i} \end{bmatrix} = \begin{cases} \begin{bmatrix} {}^w\mathbf{e}_{z_i} \times (\mathbf{p}_e - {}^w\mathbf{d}_i) \\ {}^w\mathbf{e}_{z_i} \end{bmatrix} & \text{für ein Rotationsgelenk} \\ \begin{bmatrix} {}^w\mathbf{e}_{z_i} \\ \mathbf{0} \end{bmatrix} & \text{für ein Schubgelenk} \end{cases} \quad (4.27)$$

angegeben werden. Hier ist zu beachten, dass die Vektoren \mathbf{p}_e , ${}^w\mathbf{d}_i$ und ${}^w\mathbf{e}_{z_i}$ selbst wiederum Funktionen von \mathbf{q} sind. Mithilfe der homogenen Transformationen können diese berechnet werden. Die Endeffektorposition $\mathbf{p}_e = {}^w\mathbf{d}_e(\mathbf{q})$ ist in ${}^w\mathbf{H}_e(\mathbf{q})$ enthalten, siehe

(4.4). Der Ursprung ${}^w\mathbf{d}_i$ des Koordinatensystems $\{i\}$ findet sich als Verschiebungsvektor der homogenen Transformation ${}^w\mathbf{H}_i(\mathbf{q})$ gemäß

$${}^w\mathbf{H}_i(\mathbf{q}) = \begin{bmatrix} {}^w\mathbf{R}_i(\mathbf{q}) & {}^w\mathbf{d}_i(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} = {}^w\mathbf{H}_b {}^b\mathbf{H}_1(q_1) {}^1\mathbf{H}_2(q_2) \cdots {}^{i-1}\mathbf{H}_i(q_i). \quad (4.28)$$

Der Einheitsvektor ${}^w\mathbf{e}_{z_i}$ findet sich als dritter Spaltenvektor in der Rotationsmatrix ${}^w\mathbf{R}_i(\mathbf{q})$, vgl. (3.9), welche in der homogenen Transformation ${}^w\mathbf{H}_i(\mathbf{q})$ enthalten ist.

Body-Jacobi-Matrix

Wie in Abschnitt 3.4.2 gezeigt, kann die Bewegung jedes Koordinatensystems mithilfe einer räumlichen Geschwindigkeit beschrieben werden. Die Bewegung des Koordinatensystems des Endeffektors $\{e\}$ gegenüber dem inertialfesten Koordinatensystem $\{w\}$ wird mithilfe der homogenen Transformation ${}^w\mathbf{H}_e(\mathbf{q}(t))$ bzw. dem Vektor der räumlichen Geschwindigkeit ${}^e\mathbf{t}_e^w$ beschrieben, vgl. (3.49). Dieser Vektor ist damit eine weitere Möglichkeit (neben der Endeffektorgeschwindigkeit $\dot{\mathbf{x}}_e$ und dem Vektor $[\dot{\mathbf{p}}_e^T \ \boldsymbol{\omega}_e^T]^T$), die Endeffektorgeschwindigkeit im Aufgabenraum zu beschreiben. Analog zu (4.12) und (4.20) kann die räumliche Geschwindigkeit mithilfe der sogenannten *Body-Jacobi-Matrix* $\mathbf{J}_b(\mathbf{q})$ in der Form

$${}^e\mathbf{t}_e^w = \begin{bmatrix} {}^e\mathbf{v}_e^w \\ {}^e\boldsymbol{\omega}_e^w \end{bmatrix} = \mathbf{J}_b(\mathbf{q})\dot{\mathbf{q}} \quad (4.29)$$

dargestellt werden und wird gemäß

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_g(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} {}^w\dot{\mathbf{d}}_e \\ {}^w\boldsymbol{\omega}_e \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{R}_e & {}^e\dot{\mathbf{d}}_e^w \\ {}^w\mathbf{R}_e & {}^e\boldsymbol{\omega}_e^w \end{bmatrix} = \underbrace{\begin{bmatrix} {}^w\mathbf{R}_e & \mathbf{0} \\ \mathbf{0} & {}^w\mathbf{R}_e \end{bmatrix}}_{\mathbf{T}_b(\mathbf{q})} \underbrace{\begin{bmatrix} {}^e\mathbf{v}_e^w \\ {}^e\boldsymbol{\omega}_e^w \end{bmatrix}}_{{}^e\mathbf{t}_e^w} = \underbrace{\mathbf{T}_b(\mathbf{q})\mathbf{J}_b(\mathbf{q})}_{\mathbf{J}_g(\mathbf{q})} \dot{\mathbf{q}} \quad (4.30)$$

direkt aus der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ hergeleitet. In (4.30) ist ersichtlich, dass eine ähnliche Transformationsmatrix wie in (3.54) verwendet wird, um die Geschwindigkeiten zwischen dem inertialfesten Koordinatensystem $\{w\}$ und dem Endeffektorkoordinatensystem $\{e\}$ zu transformieren.

Zusammenhang zwischen den Jacobi-Matrizen

Sowohl $\dot{\boldsymbol{\phi}}_e$ als auch $\boldsymbol{\omega}_e$ beschreiben Drehgeschwindigkeiten des Endeffektorkoordinatensystems. Das zeitliche Integral von $\boldsymbol{\omega}_e$, also $\int \boldsymbol{\omega}_e dt$, führt allerdings zu keinem sinnvollen Ausdruck für die Orientierung. Im Gegensatz dazu ist $\boldsymbol{\phi}_e = \int \dot{\boldsymbol{\phi}}_e dt$ ein gültiger Zusammenhang, denn gemäß (4.12) ist $\dot{\boldsymbol{\phi}}_e$ eine echte Zeitableitung der Minimalkoordinatendarstellung $\boldsymbol{\phi}_e$ und beschreibt die momentane Drehwinkelgeschwindigkeit in diesen Koordinaten. Dies illustriert das folgende Beispiel.

Beispiel 4.2 (Integral des Vektors der Drehwinkelgeschwindigkeiten). Es sei die Orientierung eines Starrkörpers zum Zeitpunkt $t = 0$ bekannt. Beginnend bei dieser Orientierung wird der Starrkörper mit zwei unterschiedlichen Zeitverläufen für den Vektor der Drehwinkelgeschwindigkeit gedreht, nämlich

$$\omega_1(t) = \begin{cases} \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^T & \text{für } 0 \leq t < 1 \\ \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^T & \text{für } 1 \leq t \leq 2 \end{cases} \quad (4.31)$$

und

$$\omega_2(t) = \begin{cases} \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^T & \text{für } 0 \leq t < 1 \\ \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^T & \text{für } 1 \leq t \leq 2 \end{cases}. \quad (4.32)$$

Das Integral über den gesamten Zeitverlauf $0 \leq t \leq 2$ liefert für beide Fälle

$$\int_0^2 \omega_i(t) dt = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & 0 \end{bmatrix}^T, \quad i = 1, 2, \quad (4.33)$$

jedoch kommt in beiden Fällen eine ganz unterschiedliche Lage für den Starrkörper zum Zeitpunkt $t = 2$ heraus, wie dies in Abb. 4.7 dargestellt ist. Dies führt zu dem Schluss, dass das Integral eines Vektors der Drehwinkelgeschwindigkeit keine eindeutige Aussage über die Orientierung eines Starrkörpers zulässt.

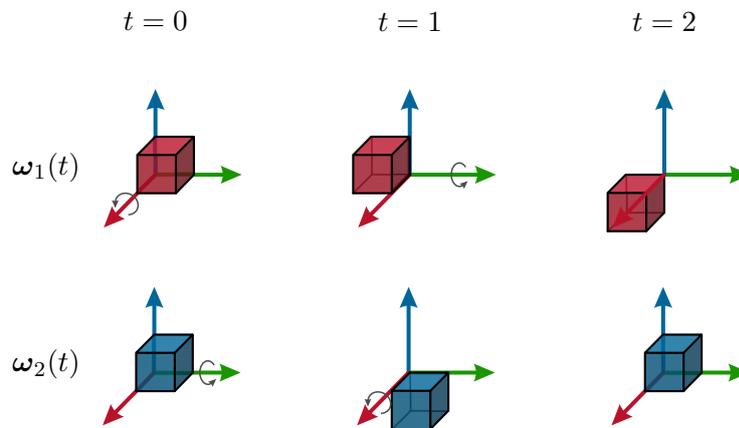


Abbildung 4.7: Bewegung des Starrkörpers, welcher mit $\omega_1(t)$ bzw. mit $\omega_2(t)$ rotiert wird [4.1].

Im Folgenden wird der Zusammenhang zwischen $\dot{\phi}_e$ und ω_e diskutiert. Zunächst wird der Vektor der Drehwinkelgeschwindigkeiten des Endeffektors ω_e mithilfe von (3.57), (3.37) und gemäß

$$\mathbf{S}(\omega_e) = \dot{\mathbf{R}}(\phi_e) \mathbf{R}^T(\phi_e) \quad (4.34)$$

ausgedrückt. Wie in (4.15) zeigt sich, dass ω_e linear in $\dot{\phi}_e$ ist. Dies bedeutet, dass ein linearer Zusammenhang zwischen $\dot{\phi}_e$ und ω_e mit einer Transformationsmatrix $\mathbf{T}_\phi(\phi_e)$

gemäß

$$\boldsymbol{\omega}_e = \mathbf{T}_\phi(\boldsymbol{\phi}_e) \dot{\boldsymbol{\phi}}_e \quad (4.35)$$

existieren muss. Diese Matrix $\mathbf{T}_\phi(\boldsymbol{\phi}_e) = \mathbf{T}_{\phi,ijk}(\boldsymbol{\phi}_e)$ folgt für die gewählte Konvention der Minimalkoordinatendarstellung $i, j, k \in \{x, y, z\}$ aus dem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ in der Form

$$\mathbf{T}_\phi(\boldsymbol{\phi}_e) = \mathbf{T}_{\phi,ijk}(\boldsymbol{\phi}_e) = \frac{\partial}{\partial \dot{\boldsymbol{\phi}}_e} \boldsymbol{\omega}_e \quad (4.36)$$

und lautet für die klassischen EULER-Winkel

$$\mathbf{T}_{\phi,zyz}(\boldsymbol{\phi}_e) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi) \sin(\theta) \\ 0 & \cos(\varphi) & \sin(\varphi) \sin(\theta) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \quad (4.37)$$

und für die RPY-Winkel

$$\mathbf{T}_{\phi,zyx}(\boldsymbol{\phi}_e) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi) \cos(\theta) \\ 0 & \cos(\varphi) & \sin(\varphi) \cos(\theta) \\ 1 & 0 & -\sin(\theta) \end{bmatrix}. \quad (4.38)$$

Schließlich folgt mit (4.35) der Zusammenhang zwischen den Jacobi-Matrizen $\mathbf{J}_g(\mathbf{q})$ und $\mathbf{J}_a(\mathbf{q})$

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \mathbf{T}_\phi(\boldsymbol{\phi}_e) \dot{\boldsymbol{\phi}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\phi(\boldsymbol{\phi}_e) \end{bmatrix} \underbrace{\begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix}}_{\dot{\mathbf{x}}_e} = \underbrace{\begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\phi(\boldsymbol{\phi}_e) \end{bmatrix}}_{\mathbf{J}_g(\mathbf{q})} \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}}. \quad (4.39)$$

Der Zusammenhang (4.39) zeigt also, dass die geometrische und die analytische Jacobi-Matrix für translatorische Geschwindigkeiten $\dot{\mathbf{p}}_e$ identisch sind. Da die beiden Matrizen lediglich unterschiedliche Darstellungen für die Drehrate verwenden, wird hierfür die Matrix $\mathbf{T}_\phi(\boldsymbol{\phi}_e)$ zum Umrechnen benötigt, siehe (4.35).

Zusammenfassend sind in Abb. 4.8 die Zusammenhänge zwischen der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ aus (4.13), der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ aus (4.20) und der Body-Jacobi-Matrix $\mathbf{J}_b(\mathbf{q})$ aus (4.29) dargestellt.

Beispiel 4.3 (Konstante Drehwinkelgeschwindigkeiten). Dieses Beispiel untersucht den nichtlinearen Zusammenhang (4.35) zwischen dem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ und der Drehrate $\dot{\boldsymbol{\phi}}_e$ in Minimalkoordinatendarstellung mit RPY-Winkel. Dazu wird ein Starrkörper mit der Orientierung $\boldsymbol{\phi}_e^T(0) = [0, 0, 0]$ zum Zeitpunkt $t = 0$ in Rotation mit konstanter Drehrate versetzt.

Zunächst wird eine konstante Drehrate $\dot{\boldsymbol{\phi}}_e^T = [1, 0.5, 0]$ rad/s in Minimalkoordinatendarstellung vorgegeben. Der Zeitverlauf der Orientierung des Starrkörpers $\boldsymbol{\phi}_e(t)$

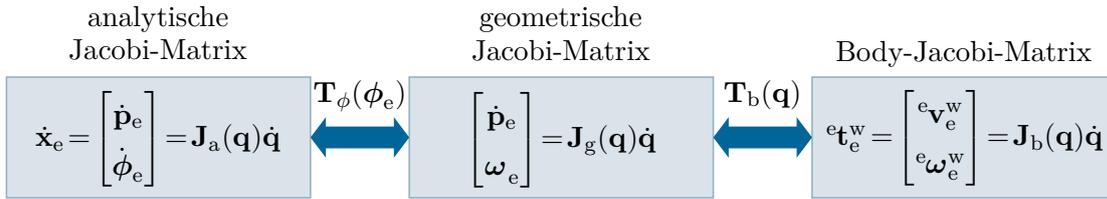


Abbildung 4.8: Mathematische Zusammenhänge zwischen den Jacobi-Matrizen.

ist direkt das Integral von $\dot{\phi}_e$, d.h.

$$\phi_e(t) = \int_0^t \dot{\phi}_e(\tau) d\tau = \begin{bmatrix} t \\ 0.5t \\ 0 \end{bmatrix} \text{ rad}, \quad (4.40)$$

während $\omega_e(t)$ zu jedem Zeitpunkt mithilfe von $\phi_e(t)$ und $\dot{\phi}_e$ aus (4.35) berechnet werden kann. Die Zeitverläufe dieser drei Vektoren sind in Abb. 4.9 dargestellt. Durch die nichtlineare Abbildung über die Matrix $\mathbf{T}_{\phi,zyx}(\phi_e)$ aus (4.38) schwankt der Vektor der Drehwinkelgeschwindigkeit $\omega_e(t)$ sinusförmig zufolge der konstanten Drehrate $\dot{\phi}_e$ in RPY-Winkel.

Im zweiten Schritt wird ein konstanter Vektor der Drehwinkelgeschwindigkeiten $\omega_e = [1, 0.5, 0]^T \text{ rad/s}$ vorgegeben und die zugehörige Drehung in Minimalkoordinatendarstellung berechnet. Dazu wird (4.35) auf $\dot{\phi}_e$ umgeformt und gemäß

$$\phi_e(t) = \int_0^t \dot{\phi}_e(\tau) d\tau = \int_0^t \mathbf{T}_{\phi,zyx}^{-1}(\phi_e(\tau)) \omega_e d\tau \quad (4.41)$$

integriert. In Abb. 4.10 sind die Zeitverläufe für diese Drehung veranschaulicht. Darin ist deutlich zu sehen, dass der konstante Vektor der Drehwinkelgeschwindigkeiten ω_e auf eine schwankende Drehrate $\dot{\phi}_e$ in Minimalkoordinatendarstellung führt. Weiters entsteht durch die Verkopplung über die Matrix $\mathbf{T}_{\phi,zyx}(\phi_e)$ in (4.41) eine Drehung in allen drei Dimensionen von ϕ_e .

Dieses Beispiel zeigt, dass durch den nichtlinearen Zusammenhang (4.35) konstante Geschwindigkeiten ω_e *nicht* auf konstante Geschwindigkeiten $\dot{\phi}_e$ abgebildet werden und umgekehrt.

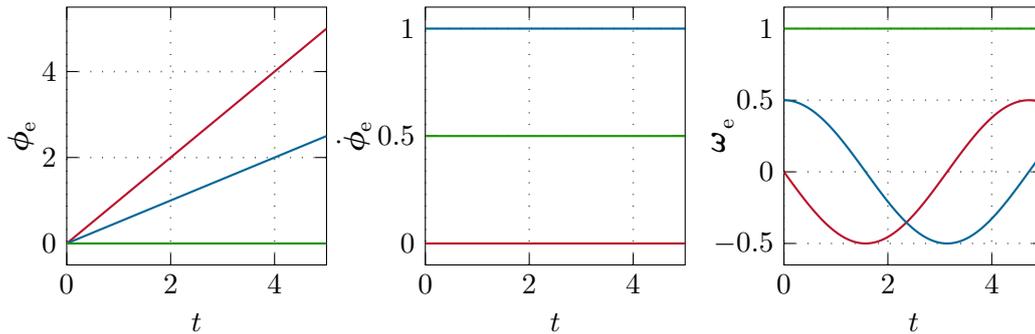


Abbildung 4.9: Drehung eines Starrkörpers mit konstanter Drehrate $\dot{\phi}_e$ in Minimalkoordinatendarstellung mit RPY-Winkel.

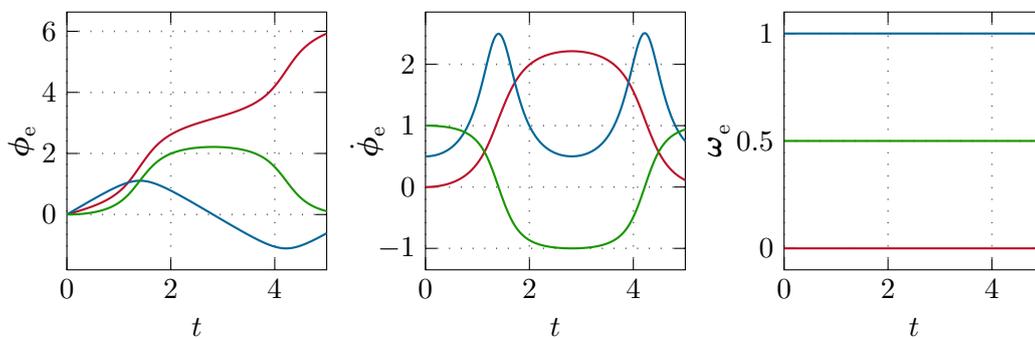


Abbildung 4.10: Drehung eines Starrkörpers mit konstantem Vektor der Drehwinkelgeschwindigkeiten ω_e .

Singularitäten

In der Robotik bezeichnet der Begriff *Singularität* jene Konfigurationen des Roboters bei denen eine Bewegungseinschränkung auftritt. Es gibt eine Reihe von unterschiedlichen Arten von Singularitäten, welche nach der Ursache kategorisiert sind. Im Folgenden werden die häufigsten Arten von Singularitäten für Roboter mit serieller Kinematik aufgelistet und diskutiert.

Kinematische Singularität: Wie in (4.20) gezeigt, bildet die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ die Geschwindigkeiten im Konfigurationsraum $\dot{\mathbf{q}}$ linear auf die Geschwindigkeiten im Aufgabenraum $[\dot{\mathbf{p}}_e^T \ \dot{\omega}_e^T]^T$ ab, wobei die Jacobi-Matrix selbst im Allgemeinen *nicht-linear* von der Konfiguration \mathbf{q} abhängt. *Kinematische Singularitäten* treten bei jenen Konfigurationen \mathbf{q} auf, an denen es zu einem Rangverlust im Zeilenrang der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ kommt. Das bedeutet, dass die Abbildung (4.20) nicht mehr alle Geschwindigkeiten im Aufgabenraum $[\dot{\mathbf{p}}_e^T \ \dot{\omega}_e^T]^T$ linear unabhängig voneinander erzeugen kann. Für einen kinematisch nichtredundanten Roboter mit $m = n$ ist $\mathbf{J}_g(\mathbf{q})$ quadratisch und Konfigurationen \mathbf{q} mit Rangverlust können direkt aus der Gleichung $\det(\mathbf{J}_g(\mathbf{q})) = 0$

ermittelt werden. In der Nähe einer Singularität hat die geometrische Jacobi-Matrix zwar vollen Rang, ist aber schlecht konditioniert. Dadurch können in der Nähe einer Singularität große Geschwindigkeiten $\dot{\mathbf{q}}$ im Konfigurationsraum in einer oder mehreren Achsen auftreten, um bestimmte Bewegungen $\dot{\mathbf{p}}_e$ bzw. $\boldsymbol{\omega}_e$ im Aufgabenraum hervorzurufen.

- **Singularität am Rand des Arbeitsraums:** Diese Art von Singularität tritt immer dann auf, wenn der Roboter ganz ausgestreckt oder ganz eingezogen ist. Mechanisch bedingt kann der Roboter ohnehin nicht über den Rand seines Arbeitsraums hinaus fahren. Durch die Singularität ist aber auch die entgegengesetzte Richtung nicht mehr möglich. Daher wird der Rand des Arbeitsraums eines Roboters in der Praxis gemieden.
- **Singularität im Inneren des Arbeitsraums:** Kinematische Singularitäten können auch im Inneren des Arbeitsraumes auftreten, z. B. wenn die Rotationsachsen von zwei oder mehreren Rotationsgelenken kollinear werden. Eine übergeordnete Pfadplanung muss daher im Vorhinein sicherstellen, dass der Roboter nicht in solche kinematischen Singularitäten hineinfährt oder in die Nähe von solchen Singularitäten kommt.

Repräsentationssingularität: Eine sogenannte *Repräsentationssingularität* ist eine Singularität, die aufgrund der gewählten Koordinatendarstellung im Aufgabenraum auftritt. Die Ursache für diese Art der Singularität liegt daher nicht in der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ (die ausschließlich die kinematische Kette beschreibt), sondern in der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ bzw. in der Transformationsmatrix $\mathbf{T}_\phi(\phi)$.

Es lässt sich zeigen, dass bei allen Minimalkoordinatendarstellungen der Orientierung Repräsentationssingularitäten auftreten. Zur Analyse der Singularitäten wird die Determinante von $\mathbf{T}_\phi(\phi)$ betrachtet. Für die klassischen EULER-Winkel lautet diese, siehe (4.37)

$$\det(\mathbf{T}_{\phi,zyz}(\phi_e)) = -\sin(\theta) \quad (4.42)$$

und für die RPY-Winkel, siehe (4.38), folgt

$$\det(\mathbf{T}_{\phi,zyx}(\phi_e)) = -\cos(\theta) . \quad (4.43)$$

Eine Repräsentationssingularität liegt vor, wenn $\det(\mathbf{T}_\phi(\phi_e)) = 0$ gilt und damit die Inverse $\mathbf{T}_\phi^{-1}(\phi_e)$ nicht existiert. In diesem Fall kann aus einem gegebenen Vektor der Drehwinkelgeschwindigkeiten des Endeffektors $\boldsymbol{\omega}_e$ keine zugehörige Winkelgeschwindigkeit der EULER-Winkel $\dot{\phi}_e$ berechnet werden, siehe (4.39). Zusätzlich degenerieren in einer Repräsentationssingularität auch die Gleichungen (3.60) und (3.61) bzw. (3.63) und (3.64). Es können φ und ψ nicht mehr unabhängig voneinander berechnet werden – nur noch die Summe bzw. die Differenz von φ und ψ . Diese Singularitäten lassen sich durch eine Nicht-Minimalkoordinatendarstellung der Orientierung (z. B. Quaternionen) beheben.

Beispiel 4.4 (Kinematische Singularitäten eines 3-Achs-Roboters). Der in Abb. 4.11 gezeigte 3-Achs-Roboter in RRR-Struktur entspricht dem 6-Achs-Roboter aus Abb. 4.3 ohne sphärischem Handgelenk, d. h. ohne die Gelenke 4, 5 und 6. Das Endeffektorkoordinatensystem des 3-Achs-Roboters $\{e\} = \{4\}$ liegt dort, wo beim 6-Achs-Roboter

der Handgelenkspunkt (engl. *wrist*, W) zu finden ist. Weitere ausgezeichnete Punkte des 3-Achs-Roboters sind die *Schulter* (engl. *shoulder*, S) und der *Ellenbogen* (engl. *elbow*, E), in Analogie zum menschlichen Arm.

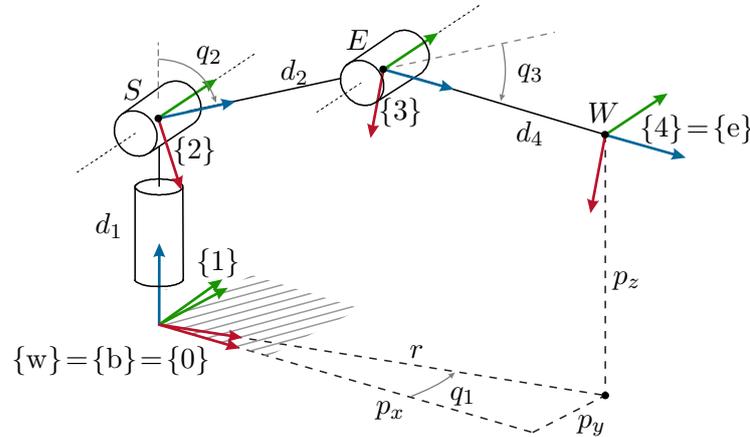


Abbildung 4.11: 3-Achs-Roboter in RRR-Struktur [4.1].

Die körperfesten Koordinatensysteme $\{i\}$ des 3-Achs-Roboters, $i = 0, \dots, 4$, sind in Abb. 4.11 dargestellt. Aus dieser Abbildung werden die homogenen Transformationen zwischen den Koordinatensystemen in der Form

$$\begin{aligned} {}^w\mathbf{H}_b &= \mathbf{E} & {}^2\mathbf{H}_3(q_3) &= \mathbf{H}_{Tz,d_2} \mathbf{H}_{Ry,q_3} \\ {}^b\mathbf{H}_0 &= \mathbf{E} & {}^3\mathbf{H}_4 &= \mathbf{H}_{Tz,d_4} \\ {}^0\mathbf{H}_1(q_1) &= \mathbf{H}_{Rz,q_1} & {}^4\mathbf{H}_e &= \mathbf{E} \\ {}^1\mathbf{H}_2(q_2) &= \mathbf{H}_{Tz,d_1} \mathbf{H}_{Ry,q_2} \end{aligned}$$

bestimmt und es folgt aus

$${}^w\mathbf{H}_e(\mathbf{q}) = {}^w\mathbf{H}_b {}^b\mathbf{H}_0 {}^0\mathbf{H}_1(q_1) {}^1\mathbf{H}_2(q_2) {}^2\mathbf{H}_3(q_3) {}^3\mathbf{H}_4 {}^4\mathbf{H}_e = \begin{bmatrix} {}^w\mathbf{R}_e(\mathbf{q}) & {}^w\mathbf{d}_e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.44)$$

die Vorwärtskinematik des 3-Achs-Roboters gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = {}^w\mathbf{d}_e(\mathbf{q}) = \begin{bmatrix} (d_4 \sin(q_2 + q_3) + d_2 \sin(q_2)) \cos(q_1) \\ (d_4 \sin(q_2 + q_3) + d_2 \sin(q_2)) \sin(q_1) \\ d_4 \cos(q_2 + q_3) + d_2 \cos(q_2) + d_1 \end{bmatrix}. \quad (4.45)$$

Um die kinematischen Singularitäten beurteilen zu können, wird die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$

$$\mathbf{J}_g(\mathbf{q}) = \frac{\partial {}^w\mathbf{d}_e}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) = \mathbf{J}_a(\mathbf{q}) \quad (4.46)$$

und deren Determinante

$$\det(\mathbf{J}_g(\mathbf{q})) = d_2 d_4 \underbrace{\sin(q_3)}_{S_1} \left(\underbrace{d_4 \sin(q_2 + q_3) + d_2 \sin(q_2)}_{S_2} \right) \quad (4.47)$$

berechnet. Da die Koordinaten des Aufgabenraums \mathbf{x}_e nur die Endeffektorposition ${}^w\mathbf{d}_e(\mathbf{q})$ und keine Orientierung beinhalten, ist in (4.46) ersichtlich, dass die geometrische und die analytische Jacobi-Matrix identisch sind. In (4.47) wurden die beiden Ausdrücke S_1 und S_2 eingeführt: Wird einer der beiden Ausdrücke gleich Null, so verschwindet die Determinante und es liegt eine kinematische Singularität vor. Der Ausdruck S_1 verschwindet, wenn der Roboterarm entweder ganz ausgestreckt ist ($q_3 = 2k\pi$, $k \in \mathbb{Z}$, siehe Abb. 4.12a) oder ganz eingezogen ist ($q_3 = (2k+1)\pi$, $k \in \mathbb{Z}$). Für bestimmte Konstellationen aus den Winkeln q_2 und q_3 liegt der Endeffektor des Roboters auf der z_0 -Achse ($p_x = p_y = 0$, siehe Abb. 4.11) und damit ist der Ausdruck S_2 gleich Null. Dies ist in Abb. 4.12b dargestellt.

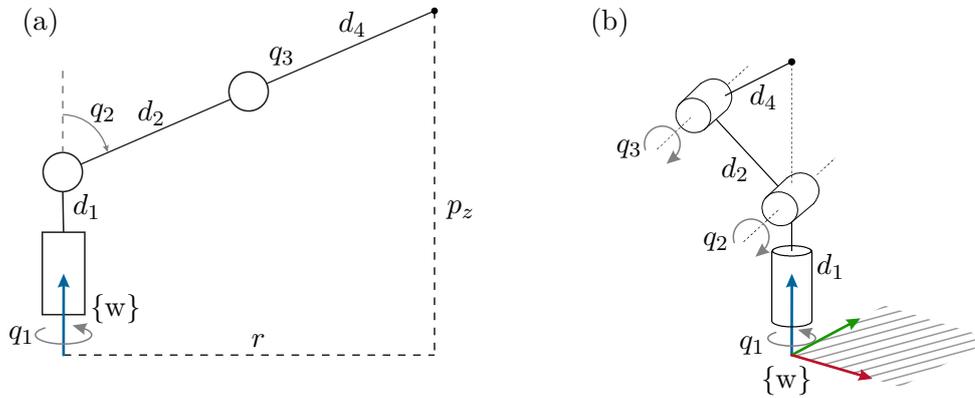


Abbildung 4.12: Kinematische Singularitäten des 3-Achs-Roboters: (a) Ausgestreckter Roboterarm ($S_1 = 0$), (b) Endeffektor liegt auf z -Achse des Koordinatensystems $\{w\}$ ($S_2 = 0$) [4.1].

Beispiel 4.5 (Kinematische Singularitäten des sphärischen Handgelenks). Abb. 4.13 zeigt eine schematische Darstellung eines sphärischen Handgelenks und den zugehörigen Koordinatensystemen $\{i\}$, welche jeweils körperfest im Glied i liegen, $i = 4, 5, 6$. Die Glieder sind durch die drei Rotationsgelenke mit den Koordinaten $\mathbf{q}^T = [q_4 \ q_5 \ q_6]$ verbunden. Die kinematische Kette wird nun anhand der homogenen Transformationen

$${}^0\mathbf{H}_6(\mathbf{q}) = {}^0\mathbf{H}_4(q_4) {}^4\mathbf{H}_5(q_5) {}^5\mathbf{H}_6(q_6) = \begin{bmatrix} {}^0\mathbf{R}_6(\mathbf{q}) & {}^0\mathbf{d}_6(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.48)$$

mit

$${}^0\mathbf{H}_4(q_4) = \mathbf{H}_{Rz,q_4} \mathbf{H}_{Tz,d_4} \quad (4.49a)$$

$${}^4\mathbf{H}_5(q_5) = \mathbf{H}_{Ry,q_5} \quad (4.49b)$$

$${}^5\mathbf{H}_6(q_6) = \mathbf{H}_{Rz,q_6} \mathbf{H}_{Tz,d_6} \quad (4.49c)$$

beschrieben, welche direkt aus Abb. 4.13 abgelesen werden können.

Ein sphärisches Handgelenk weist kinematische Singularitäten auf, welche bei einem 6-Achs-Roboter mit sphärischem Handgelenk auch im Inneren des Arbeitsraums auftreten können. Zur Analyse der kinematischen Kette wird daher der Vektor der Drehwinkelgeschwindigkeiten des Endeffektorkoordinatensystems $\boldsymbol{\omega}_e$ mithilfe von (3.37) aus ${}^0\mathbf{R}_6(\mathbf{q})$ und ${}^0\dot{\mathbf{R}}_6(\mathbf{q})$ zu

$$\boldsymbol{\omega}_e = {}^0\boldsymbol{\omega}_6(\mathbf{q}) = \begin{bmatrix} \cos(q_4) \sin(q_5) \dot{q}_6 - \dot{q}_5 \sin(q_4) \\ \sin(q_4) \sin(q_5) \dot{q}_6 + \dot{q}_5 \cos(q_4) \\ \dot{q}_6 \cos(q_5) + \dot{q}_4 \end{bmatrix} \quad (4.50)$$

berechnet. Es folgt die zugehörige geometrische Jacobi-Matrix zu

$$\mathbf{J}_g(\mathbf{q}) = \begin{bmatrix} 0 & -\sin(q_4) & \cos(q_4) \sin(q_5) \\ 0 & \cos(q_4) & \sin(q_4) \sin(q_5) \\ 1 & 0 & \cos(q_5) \end{bmatrix} \quad (4.51)$$

mit der Determinante $\det(\mathbf{J}_g(\mathbf{q})) = -\sin(q_5)$. Singularitäten treten also für $q_5 = k\pi$, $k \in \mathbb{Z}$ auf. In diesen Fällen sind die Rotationsachsen der Gelenke 4 und 6 kollinear und die Bewegungsfreiheit des Roboters ist damit eingeschränkt: Ist $\dot{q}_4 = -\dot{q}_6$, so wird keine Bewegung am Endeffektor hervorgerufen. Weiters kann in dieser Singularität keine Rotationsgeschwindigkeit um die x -Achse von $\{5\}$ (siehe Abb. 4.13 für $q_5 = 0$) erzeugt werden, da alle Rotationsgelenke orthogonal zu dieser x -Achse stehen.

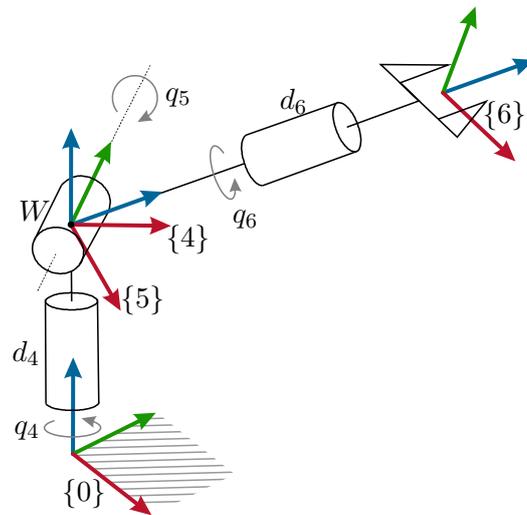


Abbildung 4.13: Sphärisches Handgelenk [4.1].

Manipulierbarkeit

Die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ kann auch dazu benutzt werden, um die sogenannte *Manipulierbarkeit* (engl. *manipulability*) des Roboters in der Form

$$m(\mathbf{q}) = \sqrt{\det(\mathbf{J}_g(\mathbf{q})\mathbf{J}_g^T(\mathbf{q}))} \quad (4.52)$$

als Funktion der Konfiguration \mathbf{q} zu definieren. Das Maß der Manipulierbarkeit sagt aus, wie gut der Roboter in seiner aktuellen Konfiguration \mathbf{q} Bewegungen in *allen* Dimensionen des Aufgabenraums ausführen kann – dies schließt also Translationen und Rotationen ein. Der Ausdruck (4.52) wird an den kinematischen Singularitäten Null, siehe Abschnitt 4.1.3. Anstelle der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ kann in (4.52) auch die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ verwendet werden. Dadurch werden die Repräsentationssingularitäten mitberücksichtigt.

4.1.4 Inverse Kinematik

Für Roboter mit serieller Kinematik kann die Lage des Endeffektors $\{e\}$ mithilfe von (4.4) eindeutig für jede Konfiguration \mathbf{q} berechnet werden. Das Problem der *inversen* Kinematik ist aber wesentlich komplexer, da die Vorwärtskinematik (4.1) invertiert werden muss, um für eine gegebene Pose im Aufgabenraum $\mathbf{x}_{e,d}$ bzw. ${}^w\mathbf{H}_{e,d}$ (Index d für *desired*, dt. *gewünscht*) die zugehörige Konfiguration \mathbf{q} zu berechnen. Die Gleichung (4.1) wird *formal* zu

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_{e,d}) \quad (4.53a)$$

$$\text{bzw. } \mathbf{q} = \mathbf{f}^{-1}({}^w\mathbf{H}_{e,d}) \quad (4.53b)$$

invertiert. Sie ist im Allgemeinen nichtlinear und es ist nicht immer möglich, geschlossene analytische Ausdrücke als Lösung zu finden. Dies ist nur für einfache kinematische Ketten bzw. für kinematische Ketten mit spezieller Struktur der Fall. Es benötigt daher entweder eine algebraische Intuition, um geeignete Gleichungen für die unbekanntenen Größen \mathbf{q} zu finden oder eine geometrische Intuition, um geeignete Zwischenpunkte zu finden, damit das Problem der inversen Kinematik analytisch lösbar wird. Aufgrund der nichtlinearen Zusammenhänge können keine, eine, mehrere oder unendlich viele Lösungen für eine gegebene Konfiguration \mathbf{q} existieren. Im Folgenden werden häufig verwendete Methoden kurz beschrieben und danach an klassischen Industrierobotern angewendet.

Geschlossene analytische Lösungen

Mithilfe von geschlossenen analytischen Ausdrücken ist es möglich, alle existierenden Lösungen einer seriellen kinematischen Kette direkt zu finden. Nachteilig ist, dass die gefundenen analytischen Ausdrücke nur für einen speziellen Roboter(typ) gelten und sich nicht verallgemeinern lassen. Methoden zum Finden von geschlossenen analytischen Ausdrücken können in zwei Gruppen eingeteilt werden, nämlich algebraische und geometrische Methoden.

Algebraische Methoden: Bei den algebraischen Methoden werden die maßgeblichen Gleichungen identifiziert, welche die generalisierten Koordinaten des Roboters enthalten. Anschließend werden diese in eine geschlossen lösbare Form umgeformt. Häufig werden dabei transzendente Gleichungen in einer Variable q_i der Form

$$C_1 \cos(q_i) + C_2 \sin(q_i) + C_3 = 0, \quad (4.54)$$

mit den Konstanten C_1 , C_2 und C_3 gebildet. Die Lösung dieser Gleichung lautet

$$q_i = 2 \arctan \left(\frac{C_2 \pm \sqrt{C_2^2 - C_3^2 + C_1^2}}{C_1 - C_3} \right), \quad (4.55)$$

unter Verwendung der WEIERSTRASS-Substitution [4.1]

$$x = \tan \left(\frac{q_i}{2} \right), \quad \sin(q_i) = \frac{2x}{1+x^2}, \quad \cos(q_i) = \frac{1-x^2}{1+x^2}. \quad (4.56)$$

Eine weitere häufige Form ist das Gleichungspaar

$$C_1 \cos(q_i) + C_2 \sin(q_i) + C_3 = 0 \quad (4.57a)$$

$$C_1 \sin(q_i) - C_2 \cos(q_i) + C_4 = 0, \quad (4.57b)$$

für welches die eindeutige Lösung

$$q_i = \operatorname{atan2}(-C_1 C_4 - C_2 C_3, C_2 C_4 - C_1 C_3) \quad (4.58)$$

existiert.

Geometrische Methoden: Bei den geometrischen Methoden werden ausgezeichnete Punkte am Roboter identifiziert, die durch eine reduzierte Anzahl an generalisierten

Koordinaten beschrieben werden können. Ein dreidimensionales Problem kann so häufig in mehrere ebene (zweidimensionale) Probleme zerlegt werden. Die resultierenden Gleichungen können mithilfe der obigen algebraischen Methoden gelöst werden.

Industrieroboter sind meist mit einer einfachen Kinematik aufgebaut, damit die inverse Kinematik analytisch lösbar bleibt, siehe z. B. Abb. 4.3. Es kann gezeigt werden, dass die inverse Kinematik eines Roboters in zwei Teilprobleme getrennt werden kann, wenn

- sich drei aufeinanderfolgende Achsen in einem einzelnen Punkt schneiden, wie z. B. beim sphärischen Handgelenk, oder wenn
- drei aufeinanderfolgende Rotationsgelenke zueinander parallele Rotationsachsen aufweisen.

In diesem Fall ist die Lösung der Endeffektor*position* von der Lösung der *Orientierung* des Endeffektors entkoppelt. Es lässt sich ein Punkt im Koordinatensystem $\{3\}$ finden, welcher durch

$${}^w\mathbf{H}_{e,d} \left({}^3\mathbf{H}_e(q_4, q_5, q_6) \right)^{-1} = {}^w\mathbf{H}_3(q_1, q_2, q_3) \quad (4.59)$$

beschrieben werden kann, siehe (4.6) sowie Abb. 4.3.

Numerische Lösungen

Im Gegensatz zu geschlossenen analytischen Lösungen sind numerische Lösungen nicht abhängig vom Roboter(typ) und können daher auf jede kinematische Kette angewendet werden. Sie sind jedoch häufig langsamer und liefern nicht alle (bzw. nur eine der) existierenden Lösungen der inversen Kinematik. Zu den wichtigsten numerischen Lösungen zählen die *differentielle inverse Kinematik*, welche in Abschnitt 4.1.5 ausführlich behandelt werden, und die *iterativen Methoden*.

Bei den iterativen Methoden wird die Vorwärtskinematik (4.1) als Nullstellensuche in der Form

$$\mathbf{g}(\mathbf{q}) = \mathbf{x}_{e,d} - \mathbf{f}(\mathbf{q}) = \mathbf{0} \quad (4.60)$$

aufgefasst. Eine TAYLOR-Reihenentwicklung erster Ordnung um einen Startpunkt \mathbf{q}_0 ergibt

$$\mathbf{g}(\mathbf{q}) = \mathbf{g}(\mathbf{q}_0) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{q}}(\mathbf{q}) \right|_{\mathbf{q}=\mathbf{q}_0} (\mathbf{q} - \mathbf{q}_0) + \text{Terme höherer Ordnung} , \quad (4.61)$$

wobei aus (4.60) mit (4.13) der Zusammenhang

$$\frac{\partial \mathbf{g}}{\partial \mathbf{q}}(\mathbf{q}) = -\mathbf{J}_a(\mathbf{q}) \quad (4.62)$$

folgt. Durch das Vernachlässigen der Terme höherer Ordnung kann daraus die Lösung \mathbf{q} gemäß

$$\mathbf{q} = \mathbf{q}_0 + \mathbf{J}_a^{-1}(\mathbf{q}_0)\mathbf{g}(\mathbf{q}_0) \quad (4.63)$$

gefunden werden. Im Allgemeinen ist die Vorwärtskinematik $\mathbf{f}(\mathbf{q})$ nichtlinear in \mathbf{q} . Daher ist die obige Lösung \mathbf{q} nur eine Approximation, welche aber erwartungsgemäß näher an der richtigen Lösung liegen sollte. Entsprechend kann \mathbf{q} als neuer Startpunkt \mathbf{q}_0 verwendet werden und die obigen Gleichung erneut angewendet werden. Es ergibt sich die iterative Vorschrift

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}_a^{-1}(\mathbf{q}_k) \mathbf{g}(\mathbf{q}_k) \quad (4.64)$$

mit $k = 0, 1, 2, \dots$, welche durch ein geeignetes Abbruchkriterium, z. B.

$$\|\mathbf{q}_{k+1} - \mathbf{q}_k\| \leq \varepsilon \quad (4.65)$$

mit einem geeigneten $\varepsilon > 0$, beendet werden kann. Die resultierende Lösung wird mit \mathbf{q}^* bezeichnet. Dieses iterative Verfahren, angewendet als numerische inverse Kinematik eines Roboters, ist unter dem Namen NEWTON-RAPHSON-Verfahren bekannt. Hier ist zu beachten, dass für numerische Verfahren eine Startlösung \mathbf{q}_0 benötigt wird, welche großen Einfluss auf die gefundene Lösung \mathbf{q}^* hat. Für weiterführende Informationen sei auf die Vorlesung *Optimierung* [4.5] verwiesen.

Beispiele

Im Folgenden wird die analytische Lösung der inversen Kinematik für einige Beispiele gezeigt, die häufig in der Industrie vorkommen.

Beispiel 4.6 (Inverse Kinematik eines 3-Achs-Roboters in RRR-Struktur). In diesem Beispiel wird die analytische inverse Kinematik des 3-Achs-Roboters aus Beispiel 4.4 betrachtet. Der dreidimensionale Konfigurationsraum $\mathbf{q}^T = [q_1 \ q_2 \ q_3]$, $n = 3$, erlaubt es, einen dreidimensionalen Aufgabenraum ($m = 3$) zu wählen. Als Koordinaten im Aufgabenraum wird der Ursprung des Endeffektorkoordinatensystems $\{e\}$ mit $\mathbf{x}_e = \mathbf{p}_e = {}^w\mathbf{d}_e = [p_x \ p_y \ p_z]^T$ gewählt, siehe Abb. 4.11. In der xy -Ebene von $\{w\}$ bilden die Geradenstücke p_x und p_y ein rechtwinkliges Dreieck mit der Hypotenuse $r = \sqrt{p_x^2 + p_y^2}$. Aus struktureller Sicht handelt es sich bei dem 3-Achs-Roboter um einen planaren 2-Achs-Roboter, dessen Bewegungsebene (aufgespannt durch die z -Achse von $\{w\}$ und die Gerade r) mit q_1 gedreht werden kann.

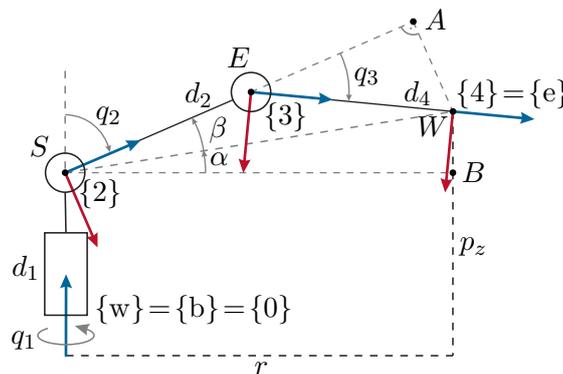


Abbildung 4.14: Ansicht der Ebene, welche durch die z -Achse von $\{w\}$ und die Gerade r des 3-Achs-Roboters aufgespannt ist.

Für $p_x \neq 0$ und/oder $p_y \neq 0$ folgt damit der erste Freiheitsgrad zu

$$q_1 = \text{atan2}(p_y, p_x) . \quad (4.66)$$

Für das Dreieck (*SEW*), siehe Abb. 4.14, lautet der Kosinussatz für den Winkel $\pi - q_3$

$$r^2 + (p_z - d_1)^2 = d_2^2 + d_4^2 - 2d_2d_4 \cos(\pi - q_3) . \quad (4.67)$$

Unter Berücksichtigung von $\cos(\pi - q_3) = -\cos(q_3)$ ergibt sich

$$c_{q_3} = \cos(q_3) = \frac{r^2 + (p_z - d_1)^2 - d_2^2 - d_4^2}{2d_2d_4} \quad (4.68)$$

und damit

$$q_3 = \pm \arccos(c_{q_3}) , \quad (4.69)$$

unter der Voraussetzung, dass der Punkt \mathbf{x}_e im Arbeitsraum des 3-Achs-Roboters liegt, also $\sqrt{r^2 + (p_z - d_1)^2} \leq d_2 + d_4$ gilt. Auf ähnliche Weise kann der verbleibende Winkel q_2 berechnet werden. Aus dem Dreieck (*SBW*) wird die Zwischengröße α gemäß

$$\alpha = \text{atan2}(p_z - d_1, r) \quad (4.70)$$

berechnet und der Zusammenhang

$$\cos(\beta) \sqrt{r^2 + (p_z - d_1)^2} = d_2 + d_4 \cos(q_3) \quad (4.71)$$

ist durch zweimalige Projektion im Dreieck (*SWA*) ersichtlich. Unter Verwendung von (4.68) berechnet sich die Zwischengröße β zu

$$\beta = \arccos\left(\frac{r^2 + (p_z - d_1)^2 + d_2^2 - d_4^2}{2d_2 \sqrt{r^2 + (p_z - d_1)^2}}\right) \quad (4.72)$$

und es folgt der verbleibende Winkel q_2 gemäß

$$q_2 = \frac{\pi}{2} - \alpha \mp \beta . \quad (4.73)$$

Bisher wurden also zwei unterschiedliche Lösungen gefunden: Die *Elbow-Up*-Lösung hat ein negatives Vorzeichen in (4.73) und ein positives Vorzeichen in (4.69) und für die *Elbow-Down*-Lösung ist es umgekehrt. Insgesamt besitzt der 3-Achs-Roboter mit RRR-Struktur vier unterschiedliche Lösungen, wie in Abb. 4.15 gezeigt ist. Wird die erste Achse um 180° gedreht mit $q'_1 = q_1 + \pi$, der zweite Winkel zu $q'_2 = -q_2$ und der dritte Winkel zu $q'_3 = -q_3$ gewählt, so können die zwei weiteren Lösungen gefunden werden.

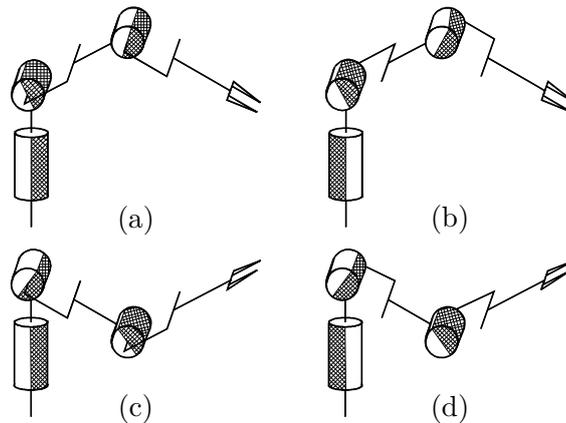


Abbildung 4.15: Die vier Lösungen der inversen Kinematik des 3-Achs-Roboters: (a) Shoulder-Right/Elbow-Up, (b) Shoulder-Left/Elbow-Up, (c) Shoulder-Right/Elbow-Down, (d) Shoulder-Left/Elbow-Down [4.1].

Beispiel 4.7 (Inverse Kinematik des sphärischen Handgelenks). In diesem Beispiel wird die inverse Kinematik des sphärischen Handgelenks aus Abb. 4.13 betrachtet, d. h. es werden die Gelenkwinkel q_4 , q_5 und q_6 für eine vorgegebene Orientierung ${}^w\mathbf{R}_{e,d}$ bzw. $\phi_{e,d}$ des Endeffektorkoordinatensystems $\{e\} = \{6\}$ gesucht. Für die Berechnung wird zunächst die Rotationsmatrix ${}^0\mathbf{R}_6(\mathbf{q})$ in (4.48) zu

$${}^0\mathbf{R}_6(\mathbf{q}) = \begin{bmatrix} c_{q_4}c_{q_5}c_{q_6} - s_{q_4}s_{q_6} & -c_{q_4}c_{q_5}s_{q_6} - s_{q_4}c_{q_6} & c_{q_4}s_{q_5} \\ s_{q_4}c_{q_5}c_{q_6} + c_{q_4}s_{q_6} & -s_{q_4}c_{q_5}s_{q_6} + c_{q_4}c_{q_6} & s_{q_4}s_{q_5} \\ -s_{q_5}c_{q_6} & s_{q_5}s_{q_6} & c_{q_5} \end{bmatrix} \quad (4.74)$$

ausgewertet. Diese weist die gleiche Form auf wie die Rotationsmatrix für die klassischen EULER-Winkel in (3.59) mit $\varphi = q_4$, $\theta = q_5$ und $\psi = q_6$. Der Zusammenhang kommt durch die Wahl der Koordinatensysteme und durch die gewählte Sequenz von Drehungen in der kinematischen Struktur zustande. Durch diese mathematische Übereinstimmung folgt sofort mit den Gleichungen (3.60) und (3.61) die inverse Kinematik für das sphärische Handgelenk eines Roboters für eine gegebene Orientierung $\phi_{e,d}$ bzw. für eine gegebene Rotationsmatrix ${}^w\mathbf{R}_{e,d}$ in der Form (3.59). Für jede Orientierung im Aufgabenraum existieren somit zwei Sätze von Lösungen \mathbf{q} .

Beispiel 4.8 (Inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk). Aufbauend auf den drei Beispiele 4.1, 4.6 und 4.7, soll in diesem Beispiel die inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk in Abb. 4.3 analytisch gelöst werden.

Im 6-dimensionalen Aufgabenraum wird nun eine Endeffektorpose $\mathbf{x}_{e,d}^T = [\mathbf{p}_{e,d}^T \ \phi_{e,d}^T]$

vorgegeben und die zugehörigen Konfigurationen $\mathbf{q}^T = [q_1 \dots q_6]$ werden gesucht. Es ist ${}^w\mathbf{R}_{e,d} = \mathbf{R}(\phi_{e,d})$ gemäß (3.59) und ${}^w\mathbf{d}_{e,d} = \mathbf{p}_{e,d}$. Für die inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk wird der Handgelenkspunkt ${}^w\mathbf{d}_4$, d. h. der Ursprung des Koordinatensystems $\{4\}$, als Bezugspunkt gewählt und die Position dieses Punktes folgt zu

$${}^w\mathbf{d}_4 = {}^w\mathbf{d}_{e,d} - {}^w\mathbf{R}_{e,d} \begin{bmatrix} 0 & 0 & d_6 \end{bmatrix}^T. \quad (4.75)$$

Wie anhand von (4.6) nachvollzogen werden kann, ist der Vektor ${}^w\mathbf{d}_4$ ausschließlich eine Funktion der ersten drei Gelenkwinkel, also gilt ${}^w\mathbf{d}_4 = {}^w\mathbf{d}_4(q_1, q_2, q_3)$. Es ist zu beachten, dass dies nicht für die Rotationsmatrix ${}^w\mathbf{R}_4 = {}^w\mathbf{R}_4(q_1, q_2, q_3, q_4)$ gilt, wie auch in Abb. 4.3 ersichtlich ist: q_4 dreht um die Längsachse des Gliedes 4 und bewegt damit den Ursprung des Koordinatensystems $\{4\}$ nicht.

Damit kann die inverse Kinematik in folgenden Schritten gelöst werden.

1. Berechnen des Handgelenkspunktes ${}^w\mathbf{d}_4(q_1, q_2, q_3)$ gemäß (4.75).
2. Lösen der inversen Kinematik für (q_1, q_2, q_3) analog zu Beispiel 4.6.
3. Berechnen von ${}^w\mathbf{R}_3(q_1, q_2, q_3)$ mithilfe von (4.6).
4. Berechnen von ${}^3\mathbf{R}_e(q_4, q_5, q_6) = {}^w\mathbf{R}_3^T(q_1, q_2, q_3) {}^w\mathbf{R}_{e,d}$.
5. Lösen der inversen Kinematik für (q_4, q_5, q_6) analog zu Beispiel 4.7.

4.1.5 Differentielle inverse Kinematik

Mit der inversen Kinematik in Abschnitt 4.1.4 werden Konfigurationen \mathbf{q} im Konfigurationsraum für eine gegebene Pose $\mathbf{x}_{e,d}$ im Aufgabenraum gesucht. Bei der *differentiellen inversen Kinematik* werden nun zusätzlich die Geschwindigkeiten und Beschleunigungen berücksichtigt. Dadurch können nicht nur Lösungen für statische Posen $\mathbf{x}_{e,d}$ gefunden werden, sondern es können gewünschte Zeitverläufe $\mathbf{x}_{e,d}(t)$, $\dot{\mathbf{x}}_{e,d}(t)$ und $\ddot{\mathbf{x}}_{e,d}(t)$ in Zeitverläufe im Konfigurationsraum $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ und $\ddot{\mathbf{q}}(t)$ übertragen werden.

Im Folgenden wird ein numerisches Verfahren für die differentielle inverse Kinematik zunächst für kinematisch nichtredundante Roboter, d. h. für $n = m$, vorgestellt. Dieses Verfahren erlaubt es, die Roboterbewegung im Konfigurationsraum $\mathbf{q}(t)$ und $\dot{\mathbf{q}}(t)$ zu einer gegebenen Bewegung des Endeffektors im Aufgabenraum $\mathbf{x}_{e,d}(t)$ und $\dot{\mathbf{x}}_{e,d}(t)$ zu berechnen. Im darauffolgenden Abschnitt wird das Verfahren auf kinematisch redundante Roboter mit $n > m$ erweitert.

Wenn die Anzahl $m = \dim(\mathbf{x}_e)$ der Koordinaten des Endeffektors im Aufgabenraum \mathbf{x}_e gleich der Anzahl $n = \dim(\mathbf{q})$ der mechanischen Freiheitsgrade \mathbf{q} des Roboters ist, d. h. $m = n$, dann ist die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ quadratisch. Außerhalb von Singularitäten ist $\mathbf{J}_a(\mathbf{q})$ regulär und es können daher die Geschwindigkeiten im Konfigurationsraum $\dot{\mathbf{q}}$ direkt durch Inversion von $\mathbf{J}_a(\mathbf{q})$ in der Form, siehe (4.12)

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q}) \dot{\mathbf{x}}_{e,d} \quad (4.76)$$

aus den gewünschten Geschwindigkeiten im Aufgabenraum $\dot{\mathbf{x}}_{e,d}$ bestimmt werden. Die Trajektorie der generalisierten Koordinaten $\mathbf{q}(t)$ wird nun einfach über Zeitintegration aus (4.76) mit der Anfangsposition $\mathbf{q}(0)$ gemäß

$$\mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\tau) d\tau + \mathbf{q}(0) \quad (4.77)$$

berechnet. Bei der Implementierung in einem Digitalrechner muss allerdings eine Zeitdiskretisierung eingeführt werden und (4.77) mit numerischen Methoden integriert werden, z. B. dem *expliziten EULER-Verfahren*. Die resultierende Gleichung lautet

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + T_a \mathbf{J}_a^{-1}(\mathbf{q}(t_k)) \dot{\mathbf{x}}_{e,d} , \quad (4.78)$$

mit der Abtastzeit T_a und dem Zeitindex k . Nach kurzer Zeit weicht allerdings die diskrete Integration $\mathbf{q}(t_k)$ von der zeitkontinuierlichen Trajektorie $\mathbf{q}(t)$ ab zufolge einer *Drift* bei der numerische Integration.

Um das obige Problem zu beheben, wird der *Fehler in den Koordinaten des Aufgabenraums* \mathbf{e} gemäß

$$\mathbf{e} = \mathbf{x}_{e,d} - \mathbf{x}_e = \mathbf{x}_{e,d} - \mathbf{f}(\mathbf{q}) \quad (4.79)$$

eingeführt. Die Zeitableitung von \mathbf{e} ist

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_{e,d} - \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}} , \quad (4.80)$$

unter Verwendung der differentiellen Kinematik (4.13). Anstatt die Gelenksgeschwindigkeit $\dot{\mathbf{q}}$ gemäß (4.76) vorzugeben, wird nun der Fehler \mathbf{e} in der Form

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\mathbf{e}) \quad (4.81)$$

mit der Verstärkungsmatrix \mathbf{K} rückgeführt. Eingesetzt in (4.80) ergibt sich das lineare Fehlersystem

$$\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = \mathbf{0} . \quad (4.82)$$

Dieses Fehlersystem ist asymptotisch stabil, wenn die Verstärkungsmatrix \mathbf{K} positiv definit gewählt wird und die Konvergenzrate hängt von den Eigenwerten dieser Matrix ab. Für systemtheoretische Betrachtungen sei auf die Vorlesungen *Automatisierung* [4.6] und *Regelungssysteme 2* [4.7] verwiesen.

Analog zu (4.78) kann auch für (4.81) eine zeitdiskrete Formulierung mithilfe des expliziten EULER-Integrationsverfahrens gefunden werden. Mit (4.79) folgt daraus die *zeitdiskrete differentielle inverse Kinematik*

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + T_a \mathbf{J}_a^{-1}(\mathbf{q}(t_k)) \left(\dot{\mathbf{x}}_{e,d}(t_k) + \mathbf{K}(\mathbf{x}_{e,d}(t_k) - \mathbf{f}(\mathbf{q}(t_k))) \right) , \quad (4.83)$$

mit der positiv definiten Verstärkungsmatrix \mathbf{K} . Ein Vergleich mit (4.60) und (4.64) zeigt, dass die differentielle inverse Kinematik genau dem NEWTON-RAPHSON-Verfahren für stationäre Punkte im Aufgabenraum $\dot{\mathbf{x}}_{e,d} = \mathbf{0}$ mit der Verstärkung $T_a \mathbf{K}$ entspricht. Die Laufvariable k bezeichnet in (4.83) den Zeitindex und beim NEWTON-RAPHSON-Verfahren (4.64) den Iterationsindex.

Abschließend sei noch darauf hingewiesen, dass die obigen Konzepte, im Speziellen (4.81) und (4.83), auch auf Ableitungen höherer Ordnung erweitert werden können [4.1]. Für weiterführende Informationen sei auf die Literatur am Ende des Kapitels hingewiesen.

4.1.6 Kinematische Redundanz

Kinematisch redundante Roboter besitzen mehr Freiheitsgrade im Konfigurationsraum $\mathbf{q} \in \mathbb{R}^n$ als für den gewünschten Aufgabenraum bzw. die gewünschte Aufgabe $\mathbf{x}_e \in \mathbb{R}^m$ benötigt wird. Durch den Aufgabenraum werden gemäß (4.1) nur m Bedingungen vorgegeben, während ein $(n - m)$ -dimensionaler Unterraum entsteht, der sogenannte *Nullraum* (engl. *null space*). Innerhalb dieses Nullraums kann der Roboter kontinuierliche Eigenbewegungen durchführen, ohne die m Bedingungen des Aufgabenraumes zu verletzen. Diese Nullraumbewegungen können genutzt werden, um neben der eigentlichen Roboterbewegung noch weitere (sekundäre) Ziele bestmöglich zu erfüllen. Für die Analyse von kinematisch redundanten Robotern sind die Jacobi-Matrizen die wichtigsten Werkzeuge.

Aufbauend auf den Ergebnissen der kinematisch nichtredundanten Roboter werden im Folgenden alle Aspekte der Roboterkinematik nochmals für redundante Roboter erweitert und diskutiert.

Vorwärtskinematik

Die Vorwärtskinematik eines kinematisch redundanten Roboters mit serieller Kinematik kann wie auch im nichtredundanten Fall in der Form (4.1) aus Abschnitt 4.1.1 berechnet werden. Der vorgestellten Ansatz mit homogenen Transformationen (4.4) kann mit einer beliebigen Anzahl an Freiheitsgraden n angewendet werden.

Differentielle Kinematik

In Abschnitt 4.1.3 wurden drei unterschiedliche Jacobi-Matrizen vorgestellt. Durch die kinematische Redundanz $n > m$ werden die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q}) \in \mathbb{R}^{m \times n}$ und die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q}) \in \mathbb{R}^{m \times n}$ zu Matrizen mit mehr Spalten als Zeilen. Anhand der Definition (4.29) ist ersichtlich, dass auch die Body-Jacobi-Matrix $\mathbf{J}_b(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ für $n > 6$ zu einer rechteckigen Matrix mit mehr Spalten als Zeilen wird.

Für die Analyse der Singularitäten mithilfe des Zeilenrangs $\text{rang}(\cdot)$ hat dies keine Auswirkung, da dafür weiterhin die geometrische oder die analytische Jacobi-Matrix verwendet werden kann. Während die Determinante $\det(\cdot)$ nur für quadratische Matrizen definiert ist, kann die Manipulierbarkeit $m(\mathbf{q})$ gemäß (4.52) für auch nichtquadratische Jacobi-Matrizen ausgewertet werden.

Inverse Kinematik

Für kinematisch redundante Roboter ist (4.53) zusätzlich zur Nichtlinearität auch unterbestimmt. Bei einem m -dimensionalen Aufgabenraum werden entsprechend nur m Gleichungen für die $n > m$ Unbekannten \mathbf{q} des Konfigurationsraums vorgegeben.

Um die inverse Kinematik für kinematisch redundante Roboter mit $n > m$ analytisch lösen zu können, wird ein $(n - m)$ -dimensionaler Redundanzparameter $\mathbf{r} \in \mathbb{R}^{n-m}$ eingeführt. Die gewünschte Pose im Aufgabenraum $\mathbf{x}_{e,d} \in \mathbb{R}^m$ zusammen mit dem Redundanzparameter \mathbf{r} haben daher wieder die Dimension n des Konfigurationsraums und es kann eine eindeutige Lösung gefunden werden. Häufig werden noch weitere binäre

bzw. diskrete Redundanzparameter benötigt, um die Eindeutigkeit zu gewährleisten. Damit eine analytische Lösung gefunden werden kann, sind jedoch spezielle Kinematiken bzw. Roboterbauformen notwendig. Beispielsweise kann die inverse Kinematik bei vielen kommerziellen 7-Achs-Robotern noch analytisch gelöst werden. Beispiele dafür sind der KUKA LBR iiwa [4.8, 4.9], der Franka Emika Panda [4.10, 4.11], der OB7 [4.12] und Neura Robotics LARA [4.13].

Für Roboter mit allgemeiner Kinematik oder besonders vielen Freiheitsgraden können nur noch numerische Methoden zum Lösen der inversen Kinematik angewendet werden, z. B. das NEWTON-RAPHSON-Verfahren (4.64). Dies ist allerdings mit erheblichem Rechenaufwand verbunden und die gefundene Lösung \mathbf{q} hängt sehr stark von der gegebenen Anfangslösung \mathbf{q}_0 ab.

Differentielle Inverse Kinematik

Für den Fall eines kinematisch redundanten Roboters, d. h. $n > m$, hat die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ gemäß (4.13) n Spalten und m Zeilen und kann nicht mehr wie in (4.76) bzw. (4.81) direkt invertiert werden. Die zusätzlichen Freiheitsgrade des kinematisch redundanten Roboters können nun so genutzt, dass nach wie vor das Gleichungssystem (4.12) erfüllt wird, aber gleichzeitig auch die Minimierungsaufgabe

$$\min_{\dot{\mathbf{q}}} V_n(\dot{\mathbf{q}}) = \min_{\dot{\mathbf{q}}} \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n) \quad (4.84)$$

mit dem noch frei zu wählenden Vektor der Gelenksgeschwindigkeiten $\dot{\mathbf{q}}_n$ gelöst wird. Diese Minimierungsaufgabe mit der Gleichungsnebenbedingung (4.12) wird typischerweise durch Minimierung der Funktion

$$L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T(\dot{\mathbf{q}} - \dot{\mathbf{q}}_n) + \boldsymbol{\lambda}^T(\dot{\mathbf{x}}_{e,d} - \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}) \quad (4.85)$$

bezüglich $\dot{\mathbf{q}}$ und $\boldsymbol{\lambda}$ gelöst. Die Größe $\boldsymbol{\lambda}$ wird auch als *Lagrange-Multiplikator* bezeichnet. Die notwendige Bedingung für ein Minimum ist durch die Ableitungen erster Ordnung in der Form

$$\frac{\partial}{\partial \dot{\mathbf{q}}} L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T - \boldsymbol{\lambda}^T \mathbf{J}_a(\mathbf{q}) = \mathbf{0} \quad (4.86a)$$

$$\frac{\partial}{\partial \boldsymbol{\lambda}} L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = (\dot{\mathbf{x}}_{e,d} - \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}) = \mathbf{0} \quad (4.86b)$$

gegeben. Wird (4.86a) nach $\dot{\mathbf{q}}$ gemäß

$$\dot{\mathbf{q}} = \mathbf{J}_a^T(\mathbf{q})\boldsymbol{\lambda} + \dot{\mathbf{q}}_n \quad (4.87)$$

aufgelöst und dies in (4.86b) eingesetzt, so folgt

$$\dot{\mathbf{x}}_{e,d} = \mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q})\boldsymbol{\lambda} + \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_n. \quad (4.88)$$

Außerhalb von Singularitäten ist die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ für einen redundanten Roboter *zeilenregulär*, d. h. die m Zeilen sind im Aufgabenraum linear unabhängig und

es gilt $\text{rang}(\mathbf{J}_a(\mathbf{q})) = m$. Daraus folgt, dass die quadratische $m \times m$ -Matrix $\mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q})$ regulär und damit invertierbar ist. Somit lässt sich der Lagrange-Multiplikator $\boldsymbol{\lambda}$ in (4.88) gemäß

$$\boldsymbol{\lambda} = \left(\mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q}) \right)^{-1} (\dot{\mathbf{x}}_{e,d} - \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_n) \quad (4.89)$$

berechnen. Eingesetzt in (4.87) folgt damit die Lösung der Minimierungsaufgabe (4.84) unter der Gleichungsnebenbedingung (4.12) zu

$$\dot{\mathbf{q}} = \mathbf{J}_a^\dagger(\mathbf{q})\dot{\mathbf{x}}_{e,d} + \left(\mathbf{E} - \mathbf{J}_a^\dagger(\mathbf{q})\mathbf{J}_a(\mathbf{q}) \right) \dot{\mathbf{q}}_n \quad (4.90)$$

mit der *rechten Pseudoinversen* der analytischen Jacobi-Matrix

$$\mathbf{J}_a^\dagger(\mathbf{q}) = \mathbf{J}_a^T(\mathbf{q}) \left(\mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q}) \right)^{-1}. \quad (4.91)$$

Bemerkung 4.1. Die rechte Pseudoinverse $\mathbf{J}_a^\dagger(\mathbf{q})$ in (4.90) wird oft auch durch $\mathbf{J}_a^*(\mathbf{q}) = \mathbf{J}_a^T(\mathbf{q}) \left(\mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q}) + k^2\mathbf{E} \right)^{-1}$ mit dem Dämpfungsfaktor k ersetzt, damit die Invertierbarkeit von $\mathbf{J}_a(\mathbf{q})\mathbf{J}_a^T(\mathbf{q})$ in der Nähe von Singularitäten sichergestellt wird.

Der Term $\left(\mathbf{E} - \mathbf{J}_a^\dagger(\mathbf{q})\mathbf{J}_a(\mathbf{q}) \right) \dot{\mathbf{q}}_n$ in (4.90) beschreibt nun die Bewegungen des kinematisch redundanten Roboters im Nullraum, ohne dass sich dabei die Pose des Endeffektors ändert. Beispielsweise können bei einem 7-Achs-Roboter alle 6 Freiheitsgrade der Pose des Endeffektors fixiert werden, während der gesamte Roboter mit dem Ellenbogen eine kontinuierliche Bewegung ausführen kann. Diese zusätzliche Bewegungsfreiheit kann nun dazu genutzt werden, um neben der eigentlichen Endeffektorbewegung $\mathbf{x}_{e,d}(t)$ ein zusätzliches (sekundäres) Kriterium bestmöglich zu erfüllen. Als allgemeiner Ansatz für $\dot{\mathbf{q}}_n$ in (4.90) wird z. B.

$$\dot{\mathbf{q}}_n = k_0 \left(\frac{\partial}{\partial \mathbf{q}} w(\mathbf{q}) \right)^T, \quad k_0 > 0 \quad (4.92)$$

mit der anwendungsspezifischen Funktion $w(\mathbf{q})$ gewählt. Damit bewegt sich der Roboter in Richtung des Gradienten des Kriteriums $w(\mathbf{q})$ und der Roboter versucht dieses sekundäre Kriterium *lokal* zu maximieren, während das primäre Ziel der vorgegebenen Endeffektorbewegung $\mathbf{x}_{e,d}(t)$ mit (4.90) erfüllt wird. Um beispielsweise die Manipulierbarkeit (und damit den Abstand zu Singularitäten) zu maximieren, wird $w(\mathbf{q}) = m(\mathbf{q})$, siehe (4.52), gewählt.

Analog zum kinematisch nichtredundanten Fall (4.81) kann nun mithilfe von (4.90) mit der Rückführung des Fehlers in den Koordinaten des Aufgabenraums \mathbf{e} aus (4.79) die *differentielle inverse Kinematik für kinematisch redundante Roboter* gemäß

$$\dot{\mathbf{q}} = \mathbf{J}_a^\dagger(\mathbf{q})(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\mathbf{e}) + \left(\mathbf{E} - \mathbf{J}_a^\dagger(\mathbf{q})\mathbf{J}_a(\mathbf{q}) \right) \dot{\mathbf{q}}_n \quad (4.93)$$

mit der positiv definiten Verstärkungsmatrix \mathbf{K} zusammengesetzt werden. Obige Gleichung kann ebenfalls analog zu (4.83) zeitdiskret implementiert werden.

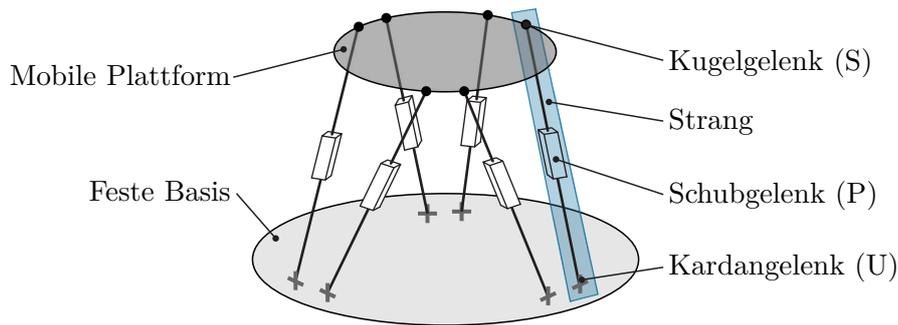


Abbildung 4.16: Beispiel eines Parallelroboters mit sechs Freiheitsgraden [4.14].

4.2 Parallele Kinematik

Roboter mit paralleler Kinematik (*Parallelroboter*, engl. *parallel robot*), siehe Abb. 1.7, sind eine spezielle Bauform von geschlossener Kinematik. Ein Parallelroboter besteht aus einer *festen Basis* (engl. *fixed base*) und einer *mobilen Plattform* (engl. *mobile platform*), welche über mehrere *gleichartige* kinematische Ketten, den sogenannten *Strängen* (engl. *legs*), miteinander verbunden sind. Durch diese Struktur entsteht eine geschlossene Kinematik über jeweils zwei Stränge, der festen Basis und der mobilen Plattform. Jeder Strang für sich ist wiederum entweder eine serielle oder eine geschlossene kinematische Kette und kann über Rotations- oder Schubgelenke aktuiert werden. Gleichzeitig enthalten Parallelroboter eine Reihe von nichtaktuierten Gelenken. Diese sind notwendig, um die Bewegungen der anderen Stränge aufzunehmen bzw. auszugleichen. Die Anzahl der Freiheitsgrade eines Parallelroboters kann mithilfe der Formel von Grübler berechnet werden, siehe Abschnitt 1.3.6. Ein Beispiel eines Parallelroboters mit sechs Freiheitsgraden ist in Abb. 4.16 dargestellt.

4.2.1 Eigenschaften

Im Folgenden werden Roboter mit serieller und paralleler Kinematik anhand von vier grundlegenden Eigenschaften miteinander verglichen.

Arbeitsraum Der wichtigste Nachteil eines Parallelroboters ist der relativ kleine Arbeitsraum. Dieser wird durch die gemeinsame Schnittmenge der Arbeitsräume aller Stränge bestimmt.

Verhältnis von Traglast zu Eigengewicht Roboter mit serieller Kinematik bewegen den Endeffektor bzw. eine Lastmasse, welche sich an einem ausladenden Ende einer (langen) Kette befindet. Dementsprechend muss jeder Aktor dieser seriellen Kette nicht nur die Lastmasse, sondern auch alle nachfolgenden Glieder und Aktoren tragen und bewegen können. Dies führt zu einem schlechten Verhältnis von Traglast zu Eigengewicht.

Bei Parallelrobotern wird die Lastmasse direkt durch alle Aktoren bzw. Stränge getragen. Zusätzlich können die Aktoren direkt in der (oder nahe der) festen Basis angebracht werden. Dadurch können die Roboterglieder, die die mobile Plattform

mit der festen Basis verbinden, deutlich leichter konstruiert werden und es wird ein geringeres Eigengewicht bei gleicher Traglast erreicht.

Genauigkeit und Wiederholgenauigkeit Bei Robotern mit serieller Kinematik akkumulieren sich Positionsfehler von einem Gelenk zum nächsten und werden somit bis zum Endeffektor hin über den Hebelarm „verstärkt“. Dieser Effekt ist umso größer, je weiter ein Gelenk vom Endeffektor entfernt ist. Mechanisches Spiel, Reibung und Elastizität in den Gliedern und den Gelenken sind Quellen von Ungenauigkeit in einer seriellen Kette, welche maßgeblich die Absolutgenauigkeit eines Roboters mit serieller Kinematik begrenzen.

Parallelroboter weisen diese Nachteile nicht auf, da die leichte Bauweise durch die parallele Struktur dennoch eine hohe Steifigkeit bietet.

Dynamik Durch das hervorragende Verhältnis zwischen Traglast und Eigengewicht bei den Parallelrobotern, können diese Bewegungen mit hoher Dynamik am Endeffektor ausführen.

4.2.2 Bauformen

Grundsätzlich werden Parallelroboter in zwei Kategorien eingeteilt, nämlich die *planaren* Parallelroboter (engl. *planar parallel robots*) und die *räumlichen* Parallelroboter (engl. *spatial parallel robots*). Diese werden im Folgenden diskutiert und zusätzlich auch die *Delta-Roboter* (engl. *Delta robots*) als Spezialfall der räumlichen Parallelroboter vorgestellt.

Planare Parallelroboter Bei einem planaren Parallelroboter besitzt die mobile Plattform drei Freiheitsgrade. Das sind zwei translatorische und ein rotatorischer Freiheitsgrad um jene Achse, die normal zur mobilen Plattform steht. Für einen nicht-redundanten Parallelroboter muss die mobile Plattform mit der fixen Basis über drei Stränge verbunden werden. Ein Beispiel eines planaren Parallelroboters ist in Abb. 4.17a dargestellt. Für die drei Stränge des Roboters sind drei unterschiedliche Strukturen möglich, vgl. Abschnitt 1.3.3:

- RPR-Struktur, bei der das Schubgelenk aktuiert ist, siehe Abb. 4.17a.
- PRR-Struktur, bei der das Schubgelenk aktuiert ist.
- RRR-Struktur, bei der jenes Drehgelenk aktuiert ist, welches der festen Basis am nächsten ist.

Räumliche Parallelroboter Bei dieser Bauform besitzt die mobile Basis einen räumlichen Arbeitsraum. Zu dieser Kategorie gehören Roboter mit ausschließlich translatorischen Freiheitsgraden und auch Roboter mit sechs Freiheitsgraden an der mobilen Plattform. Letzterer wird auch STEWART-GOUGH-Parallelroboter bzw. *Hexapod* genannt, siehe Abb. 4.17b.

Delta-Roboter Der Delta-Roboter hat im einfachsten Fall drei translatorische Freiheitsgrade in der mobilen Plattform. Werden alle drei Motoren inertialfest verbaut, so hat der Delta-Roboter eine besonders geringe bewegte Masse und kann leichte Objekte

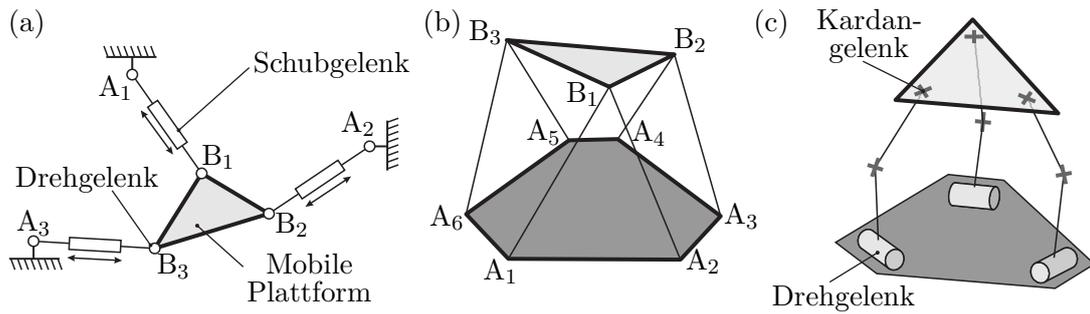


Abbildung 4.17: Unterschiedliche Bauformen von Parallelrobotern. (a) Planarer Parallelroboter mit drei Freiheitsgraden, (b) Räumlicher Parallelroboter mit sechs Freiheitsgraden, (c) Delta-Roboter [4.14].

mit sehr kurzer Zykluszeit manipulieren. Im Vergleich zum STEWART-GOUGH-Parallelroboter besitzt diese Roboterbauform einen relativ großen Arbeitsraum. Diese Bauform wird besonders häufig in der Industrie eingesetzt, siehe Abb. 4.17c. Häufig wird der Delta-Roboter auch mit einem vierten Freiheitsgrad, nämlich einer Rotation um die vertikale Achse, eingesetzt. Der rotatorische Freiheitsgrad kann entweder über einen Motor in der mobilen Plattform oder durch eine flexible Welle realisiert werden.

4.2.3 Stewart-Gough-Parallelroboter

In diesem Abschnitt wird die Kinematik sowie die differentielle Kinematik eines STEWART-GOUGH-Parallelroboters hergeleitet. Die hier vorgestellten Methoden können auch auf den Delta-Roboter angewendet werden.

Geometrie

Die Geometrie eines STEWART-GOUGH-Parallelroboters ist in Abb. 4.18 dargestellt. Für die sechs Stränge der Parallelkinematik wird eine RR-P-RRR-Struktur (=U-P-S-Struktur) angenommen, d. h. in den Punkten A_i , $i = 1, \dots, 6$, befindet sich ein Kardangelenk (U), in den Punkten B_i , $i = 1, \dots, 6$, sind Kugelgelenke (S) und die Längen der Stränge werden mithilfe von Schubgelenken (P) aktuiert. Die generalisierten Koordinaten $\mathbf{q}^T = [q_1 \dots q_6]$ sind die Längen der Stränge $\overline{A_i B_i}$, $i = 1, \dots, 6$.

Die Kinematik des Roboters wird mithilfe der beiden Koordinatensysteme der festen Basis $\{b\}$ und der mobilen Plattform $\{m\}$ beschrieben. Weiters werden noch das inertialfeste Koordinatensystem $\{w\}$ und das Endeffektorkoordinatensystem $\{e\}$ eingeführt. Diese sind wie folgt definiert:

- Der Ursprung von $\{b\}$ liegt im Punkt A_1 und die x -Achse zeigt in Richtung des Punktes A_2 . Die xy -Ebene von $\{b\}$ wird durch die Punkte A_1 , A_2 und A_6 bestimmt.
- Der Ursprung von $\{m\}$ liegt im Punkt B_1 und die x -Achse zeigt in Richtung des Punktes B_2 . Die xy -Ebene von $\{m\}$ wird durch die Punkte B_1 , B_2 und B_6 bestimmt.

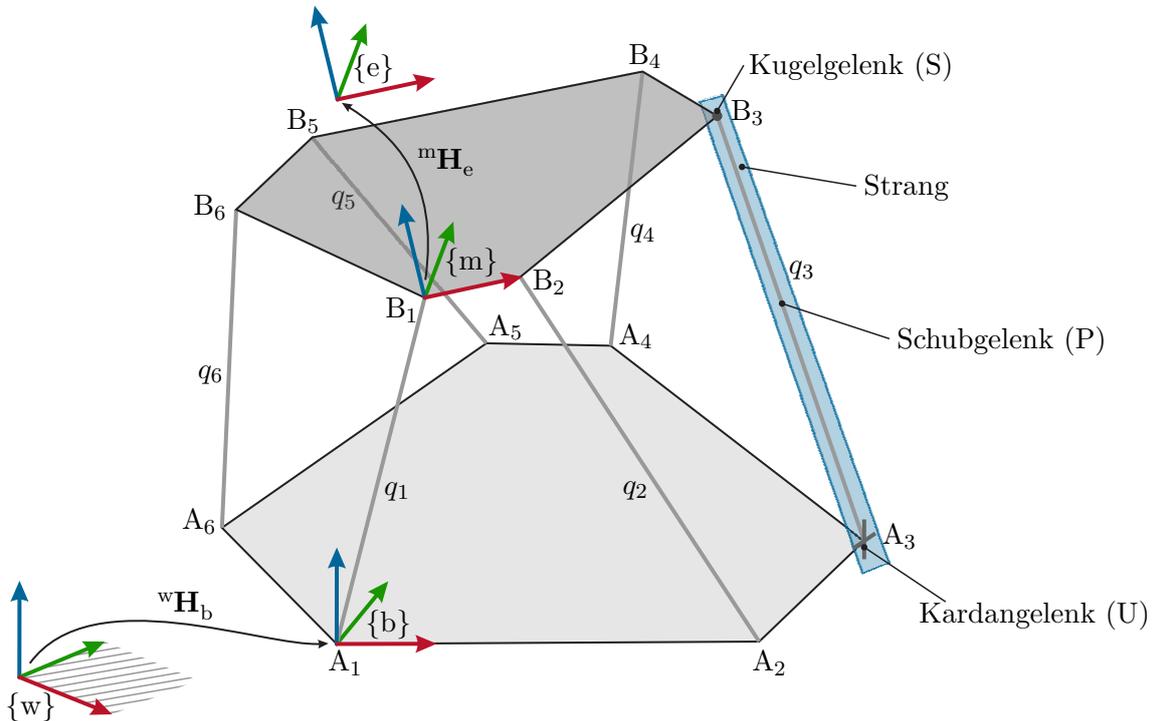


Abbildung 4.18: Schematische Darstellung eines STEWART-GOUGH-Parallelroboters. Jeder der sechs Stränge besteht aus einem passiven Kardangelenkelement (U), einem aktuierten Schubgelenkelement (P) und einem passiven Kugelgelenkelement (S) [4.14].

- Die konstante homogene Transformation ${}^w\mathbf{H}_b$ beschreibt die Lage der festen Basis $\{b\}$ bezüglich des Inertialkoordinatensystems $\{w\}$.
- Die konstante homogene Transformation ${}^m\mathbf{H}_e$ beschreibt die Lage des Endeffektorkoordinatensystems $\{e\}$ bezüglich Koordinatensystems der mobilen Plattform $\{m\}$, ausgedrückt in $\{m\}$.

Der Zusammenhang zwischen den Koordinatensystemen lautet entsprechend

$${}^w\mathbf{H}_e(\mathbf{q}) = {}^w\mathbf{H}_b {}^b\mathbf{H}_m(\mathbf{q}) {}^m\mathbf{H}_e . \quad (4.94)$$

Inverse Kinematik

Bei der inversen Kinematik des Parallelroboters wird die gewünschte Lage des Endeffektorkoordinatensystems $\{e\}$ mit $\mathbf{x}_{e,d}$ bzw. ${}^w\mathbf{H}_{e,d}$ vorgegeben und der Vektor der generalisierten Koordinaten \mathbf{q} gemäß (4.53) berechnet. Da die Positionen der Punkte A_i , in der festen Basis $\{b\}$ bzw. der Punkte B_i , $i = 1, \dots, 6$, auf der mobilen Plattform $\{m\}$ bekannt sind, können diese in das Inertialkoordinatensystem $\{w\}$ gemäß, vgl. (3.24) und (3.25)

$$\begin{bmatrix} {}^w\mathbf{d}_{A_i} \\ 1 \end{bmatrix} = {}^w\mathbf{H}_b \begin{bmatrix} {}^b\mathbf{d}_{A_i} \\ 1 \end{bmatrix} \quad (4.95a)$$

$$\begin{bmatrix} {}^w\mathbf{d}_{B_i} \\ 1 \end{bmatrix} = {}^w\mathbf{H}_{e,d} {}^m\mathbf{H}_e^{-1} \begin{bmatrix} {}^m\mathbf{d}_{B_i} \\ 1 \end{bmatrix} \quad (4.95b)$$

transformiert werden. Die generalisierten Koordinaten der Schubgelenke q_i ergeben sich direkt als geometrische Länge $\overline{A_i B_i}$, $i = 1, \dots, 6$ als

$$q_i^2 = ({}^w\mathbf{d}_{B_i} - {}^w\mathbf{d}_{A_i})^T ({}^w\mathbf{d}_{B_i} - {}^w\mathbf{d}_{A_i}), \quad (4.96)$$

wobei nur $q_i > 0$ physikalisch sinnvoll ist. Aus den obigen Gleichungen ist ersichtlich, dass die inverse Kinematik des STEWART-GOUGH-Parallelroboters eindeutig ist und einfach und mit geschlossenen analytischen Ausdrücken bestimmt werden kann. Dies verhält sich genau reziprok zur seriellen Kinematik: Während die Vorwärtskinematik bei Robotern mit serieller Kinematik einfach und eindeutig auszuwerten ist, ist die inverse Kinematik schwieriger lösbar und liefert im Allgemeinen eine Vielzahl an Lösungen.

Differentielle Kinematik

Ähnlich wie bei der inversen Kinematik, ist auch die differentielle inverse Kinematik eines Parallelroboters einfacher zu lösen. Bei der inversen differentiellen Kinematik werden die Endeffektorgeschwindigkeiten $\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix}^T$ bezüglich des Inertialkoordinatensystems $\{w\}$ vorgegeben und die zugehörigen Geschwindigkeiten im Konfigurationsraum \mathbf{q} bestimmt. Der gesuchte Zusammenhang ist also invers zu (4.20) und lautet

$$\dot{\mathbf{q}} = \mathbf{J}_g^{-1}(\mathbf{q}) \begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} \quad (4.97)$$

mit der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$. Für die Herleitung dieses Zusammenhangs werden zunächst die translatorischen Geschwindigkeiten der Punkte B_i , $i = 1, \dots, 6$, bezüglich des Inertialkoordinatensystems $\{w\}$ gemäß

$${}^w\dot{\mathbf{d}}_{B_i} = \dot{\mathbf{p}}_e + \boldsymbol{\omega}_e \times {}^w\mathbf{d}_{B_i}^e \quad (4.98)$$

berechnet, wobei der Vektor ${}^w\mathbf{d}_{B_i}^e$ zwischen dem Ursprung von $\{e\}$ und dem Punkt B_i in $\{w\}$ ausgedrückt ist. Da zunächst nur die Vektoren ${}^m\mathbf{d}_{B_i}$ bekannt sind, müssen die Vektoren ${}^e\mathbf{d}_{B_i}$, $i = 1, \dots, n$, mit

$$\begin{bmatrix} {}^e\mathbf{d}_{B_i} \\ 1 \end{bmatrix} = {}^m\mathbf{H}_e^{-1} \begin{bmatrix} {}^m\mathbf{d}_{B_i} \\ 1 \end{bmatrix} \quad (4.99)$$

berechnet werden. Die benötigte Geschwindigkeit \dot{q}_i im Schubgelenk i findet sich nun als Projektion der Geschwindigkeit des Punktes B_i auf die Achse des Schubgelenks \mathbf{u}_i in der Form

$$\dot{q}_i = \mathbf{u}_i^T {}^w\dot{\mathbf{d}}_{B_i} \quad (4.100)$$

mit dem Einheitsvektor

$$\mathbf{u}_i = \frac{{}^w\mathbf{d}_{B_i} - {}^w\mathbf{d}_{A_i}}{\|{}^w\mathbf{d}_{B_i} - {}^w\mathbf{d}_{A_i}\|}. \quad (4.101)$$

Das Einsetzen von (4.98) in (4.100) ergibt

$$\begin{aligned} \dot{q}_i &= \mathbf{u}_i^T \dot{\mathbf{p}}_e + \mathbf{u}_i^T (\boldsymbol{\omega}_e \times {}^w \mathbf{d}_{B_i}^e) \\ &= \mathbf{u}_i^T \dot{\mathbf{p}}_e + \left({}^w \mathbf{d}_{B_i}^e \times \mathbf{u}_i \right)^T \boldsymbol{\omega}_e, \end{aligned} \quad (4.102)$$

worin die Eigenschaft des Spatproduktes $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ verwendet wurde. Dieses Zwischenergebnis für eine einzelne Achse q_i wird schließlich für alle sechs Achsen $i = 1, \dots, 6$ zur Inversen der geometrischen Jacobi-Matrix in der Form

$$\mathbf{J}_g^{-1}(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_1^T & \left({}^w \mathbf{d}_{B_1}^e \times \mathbf{u}_1 \right)^T \\ \vdots & \dots \\ \mathbf{u}_6^T & \left({}^w \mathbf{d}_{B_6}^e \times \mathbf{u}_6 \right)^T \end{bmatrix} \quad (4.103)$$

zusammengesetzt.

Die differentielle (Vorwärts-)Kinematik gemäß der Definition (4.20) wird direkt durch Inversion von (4.97) mit (4.103) gefunden.

Vorwärtskinematik

Die Vorwärtskinematik von Parallelrobotern für eine gegebene Roboterkonfiguration \mathbf{q} , analog zu (4.1), ist deutlich schwieriger zu berechnen im Vergleich zu Robotern mit serieller Kinematik.

Eine analytisch geschlossene Lösung für die Vorwärtskinematik existiert im Allgemeinen nicht, sondern wurde nur für spezielle Bauformen hergeleitet. Beispielsweise [4.15] kann die Lösung der STEWART-GOUGH-Plattform mithilfe der WEIERSTRASS-Substitution (4.56) auf ein Polynom 40. Ordnung in einer einzelnen Variable zurückgeführt werden, welches aber schwierig und nur mit großem Aufwand lösbar ist. Aus den numerischen Lösungen dieses Polynoms können alle anderen Freiheitsgrade der Roboterkonfiguration eindeutig bestimmt werden. Für die Gruppe der Delta-Roboter kann gezeigt werden, dass die Vorwärtskinematik durch das Lösen eines Polynoms 2. Ordnung berechnet werden kann [4.16].

In der Praxis wird die Vorwärtskinematik häufig numerisch gelöst. Dies erfolgt entweder mithilfe von iterativen Algorithmen, analog zu Abschnitt 4.1.4, oder mithilfe von Integrationsverfahren, analog zu Abschnitt 4.1.5.

4.3 Literatur

- [4.1] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [4.2] J. Denavit und R. S. Hartenberg, „A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,“ *Journal of Applied Mechanics*, Jg. 22, Nr. 2, S. 215–221, 1955, ISSN: 0021-8936.
- [4.3] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3. Aufl. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005.
- [4.4] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [4.5] A. Steinböck, *Skriptum zur VU Optimierung (WS 2023/24)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. Adresse: <https://www.acin.tuwien.ac.at/master/optimierung/>.
- [4.6] A. Kugi, *Skriptum zur VU Automatisierung (WS 2023/2024)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. Adresse: <https://www.acin.tuwien.ac.at/bachelor/automatisierung/>.
- [4.7] A. Kugi, *Skriptum zur VO Nichtlineare dynamische Systeme und Regelung (SS 2023)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. Adresse: <https://www.acin.tuwien.ac.at/master/nichtlineare-dynamische-systeme-und-regelung/>.
- [4.8] KUKA Roboter GmbH, *LBR iiwa Betriebsanleitung*, Manual, 2015.
- [4.9] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki und K. Kosuge, „Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution,“ *IEEE Transactions on Robotics*, Jg. 24, Nr. 5, S. 1131–1142, 2008.
- [4.10] Y. He und S. Liu, „Analytical Inverse Kinematics for Franka Emika Panda – a Geometrical Solver for 7-DOF Manipulators with Unconventional Design,“ in *Proceedings of the International Conference on Control, Mechatronics and Automation (ICCMA)*, 2021, S. 194–199.
- [4.11] Franka Emika, *PANDA*, (24. Oktober, 2022). Adresse: <https://www.franka.de/>.
- [4.12] productive robotics, *OB7*, (24. Oktober, 2022). Adresse: <https://www.productiverobotics.com/>.
- [4.13] NEURA Robotics, *LARA*, (24. Oktober, 2022). Adresse: <https://neura-robotics.com>.
- [4.14] W. Khalil und E. Dombre, *Modeling, Identification and Control of Robots*. London, UK: Hermes Penton Ltd., 2002.
- [4.15] M. Husty, „An algorithm for solving the direct kinematics of general Stewart-Gough platforms,“ *Mechanism and Machine Theory*, Jg. 31, Nr. 4, S. 365–379, 1996.

- [4.16] F. Pierrot, P. Dauchez und A. Fournier, „Fast parallel robots,“ *Journal of Robotic Systems*, Jg. 8, Nr. 6, S. 829–840, 1991.

5 Dynamik

In diesem Kapitel wird die Roboterdynamik, also der Zusammenhang zwischen der Roboterbewegung und den wirkenden Kräften und Momenten beschrieben. Für einen Roboter mit den generalisierten Koordinaten $\mathbf{q} \in \mathbb{R}^n$ haben die Bewegungsgleichungen die Form einer Differentialgleichung zweiter Ordnung

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} . \quad (5.1)$$

Hierbei bezeichnet $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ die Massenmatrix, der Term $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ beinhaltet die Zentrifugal- und Coriolisterme, und $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ die Gravitationsterme. Die in den Gelenken wirkenden generalisierten Gelenkmomente sind in dem Vektor $\boldsymbol{\tau} \in \mathbb{R}^n$ zusammengefasst.

Ähnlich wie im Bereich der Kinematik wird in der Dynamik zwischen der Vorwärtsdynamik (engl. forward dynamics) und der Inversdynamik (engl. inverse dynamics) unterschieden. Hierbei bezeichnet die Vorwärtsdynamik eine Berechnung der Gelenkbeschleunigung $\ddot{\mathbf{q}}$ bei gegebenem Zustand $(\mathbf{q}, \dot{\mathbf{q}})$ und bekannten Gelenkdrehmomenten $\boldsymbol{\tau}$, d.h.

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})) := \mathbf{fdyn}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}) . \quad (5.2)$$

Die Vorwärtsdynamik wird unter anderem benötigt, um das dynamische Verhalten des Roboters zu simulieren. Als Inversdynamik bezeichnet man hingegen die Berechnung der Gelenkdrehmomente $\boldsymbol{\tau}$ bei vorgegebener Gelenkkonfiguration, Geschwindigkeiten und Beschleunigungen via (5.1). Im Rahmen der Vorlesung werden zwei Methoden zur Dynamikberechnung besprochen. Der Lagrange-Formalismus eignet sich sehr gut für Roboter mit wenigen Freiheitsgraden und um die prinzipiellen Eigenschaften der Roboterdynamik abzuleiten. Für komplexere Robotersysteme mit einer größeren Anzahl an Freiheitsgraden ($n > 7$), sowie zur direkten Berechnung der Vorwärtsdynamik, eignen sich effiziente rekursive Algorithmen besser.

5.1 Newton-Euler-Gleichung für einen Starrkörper

Wie in Kapitel 3 beschrieben, kann die Pose (Position und Orientierung) eines starren Körpers mit einem körperfesten Koordinatensystems $\{k\}$ durch eine homogene Transformation $\mathbf{H} \in \text{SE}(3)$ beschrieben werden. Diese beinhaltet eine Rotationsmatrix $\mathbf{R} \in \text{SO}(3)$ zur Beschreibung der Orientierung sowie einen Vektor ${}^w\mathbf{d}_k \in \mathbb{R}^3$ zur Beschreibung des Ursprungs von $\{k\}$ gegenüber einem inertialfesten Basiskoordinatensystem $\{w\}$.

Das dynamische Verhalten eines starren Körpers ist durch seine Masse m , die Position seines Schwerpunktes $\mathbf{c} = {}^k\mathbf{d}_c \in \mathbb{R}^3$, und seine Rotationsträgheit (auch „Massenträgheitsmoment“) ${}^k\mathcal{I}_k \in \mathbb{R}^{3 \times 3}$ definiert. Diese Größen ergeben sich aus der Masseverteilung innerhalb des Körpers, welche wiederum durch die ortsabhängige Dichte $\rho : \mathbb{R}^3 \rightarrow \mathbb{R} \geq 0$

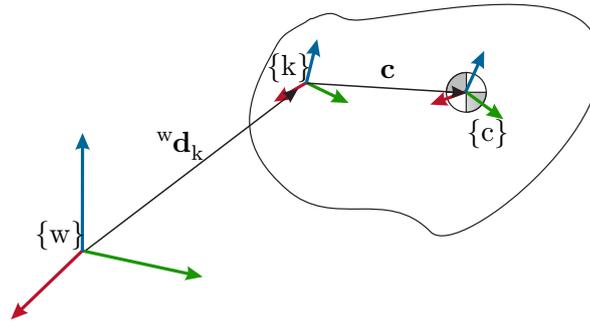


Abbildung 5.1: Koordinatensysteme in der Dynamikbeschreibung eines starren Körpers

spezifiziert ist. Zur Ermittlung dieser Größen werden die Dichte, das erste und das zweite mechanische Moment über das Körpervolumen V mit dem Positionsvektor \mathbf{r} (bezüglich $\{k\}$) des Volumenelements dV aufintegriert:

$$m = \int_V \rho(\mathbf{r}) dV \quad (5.3)$$

$$\mathbf{c} = \frac{1}{m} \int_V \mathbf{r} \rho(\mathbf{r}) dV \quad (5.4)$$

$${}^k\mathcal{I}_k = \int_V \rho(\mathbf{r}) \mathbf{S}(\mathbf{r})^T \mathbf{S}(\mathbf{r}) dV . \quad (5.5)$$

In dem körperfesten Koordinatensystem $\{k\}$ sind diese Größen konstant. Die Rotationsträgheit ${}^k\mathcal{I}_k$ ist bezüglich des körperfesten Koordinatensystems $\{k\}$ angegeben. Sie ist symmetrisch und positiv definit und kann somit auf sechs unabhängige Parameter reduziert werden (drei Diagonalelemente und drei Außerdiagonalelemente). Für die Beschreibung der Dynamik ist es zweckmäßig, ein Koordinatensystem $\{c\}$ zu definieren, das seinen Ursprung im Schwerpunkt hat und die gleiche Orientierung wie $\{k\}$ aufweist, d.h.

$${}^k\mathbf{H}_c = \begin{bmatrix} \mathbf{E} & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} .$$

5.1.1 Newton-Euler-Gleichungen im Inertialsystem

Die Bewegung des Körpers mit dem körperfesten Koordinatensystem $\{c\}$ ergibt sich nun aus der initialen Pose und Geschwindigkeit und der über die Zeit auf den Körper wirkenden Kräfte ${}^w\mathbf{f}_c \in \mathbb{R}^3$ und Momente ${}^w\mathbf{m}_c \in \mathbb{R}^3$. Als grundlegende Axiome kommen der Impuls- und Drallsatz zum Einsatz. Im Folgenden wird der translatorische Impuls mit $\mathbf{h}_v = m\mathbf{v}_c$ bezeichnet, wobei $\mathbf{v}_c = {}^w\mathbf{v}_c$ die Geschwindigkeit des Schwerpunktes in Inertialkoordinaten darstellt. Die zeitliche Änderung des Impulses ist gleich der Summe der wirkenden Kräfte ${}^w\mathbf{f}_c$ (Impulssatz, Newton-Gleichung):

$$\dot{\mathbf{h}}_v = {}^w\mathbf{f}_c \quad (5.6)$$

$$m\dot{\mathbf{v}}_c = {}^w\mathbf{f}_c . \quad (5.7)$$

Der Drehimpuls (auch Drall) $\mathbf{h}_\omega \in \mathbb{R}^3$ ist durch $\mathbf{h}_\omega = {}^w\mathcal{I}_c \boldsymbol{\omega}_c$ gegeben, wobei ${}^w\mathcal{I}_c \in \mathbb{R}^{3 \times 3}$ die Rotationsträgheit bezüglich des Schwerpunktes in inertialfesten Koordinaten darstellt, d.h. ${}^w\mathcal{I}_c = {}^w\mathbf{R}_c {}^c\mathcal{I}_c {}^c\mathbf{R}_w$. Die Winkelgeschwindigkeit des Körpers in Inertialkoordinaten ist durch $\boldsymbol{\omega}_c = {}^w\boldsymbol{\omega}_c$ gegeben. Die Änderung des Drehimpulses entspricht der Summe der wirkenden Momente (Drallsatz, Euler-Gleichung):

$$\dot{\mathbf{h}}_\omega = {}^w\mathbf{m}_c \quad (5.8)$$

Durch Einsetzen von $\mathbf{h}_\omega = {}^w\mathcal{I}_c \boldsymbol{\omega}_c$ erhält man die Euler-Gleichung in inertialfesten Koordinaten:

$$\begin{aligned} {}^w\mathcal{I}_c \dot{\boldsymbol{\omega}}_c + \dot{{}^w\mathcal{I}_c} \boldsymbol{\omega}_c &= {}^w\mathbf{m}_c \\ {}^w\mathcal{I}_c \dot{\boldsymbol{\omega}}_c + \left({}^w\dot{\mathbf{R}}_c {}^c\mathcal{I}_c {}^c\mathbf{R}_w + {}^w\mathbf{R}_c {}^c\mathcal{I}_c {}^c\dot{\mathbf{R}}_w \right) \boldsymbol{\omega}_c &= {}^w\mathbf{m}_c \\ {}^w\mathcal{I}_c \dot{\boldsymbol{\omega}}_c + \left({}^w\dot{\mathbf{R}}_c {}^c\mathbf{R}_w {}^w\mathcal{I}_c + {}^w\mathcal{I}_c {}^w\mathbf{R}_c {}^c\dot{\mathbf{R}}_w \right) \boldsymbol{\omega}_c &= {}^w\mathbf{m}_c \\ {}^w\mathcal{I}_c \dot{\boldsymbol{\omega}}_c + \mathbf{S}(\boldsymbol{\omega}_c) {}^w\mathcal{I}_c \boldsymbol{\omega}_c &= {}^w\mathbf{m}_c . \end{aligned} \quad (5.9)$$

Die letzte Zeile erhält man durch Ersetzen von ${}^w\dot{\mathbf{R}}_c {}^c\mathbf{R}_w = \mathbf{S}(\boldsymbol{\omega}_c)$ und Ausnutzen der Eigenschaft $\mathbf{S}(\boldsymbol{\omega}_c) \boldsymbol{\omega}_c = \mathbf{0}$ (Kreuzprodukt paralleler Vektoren). Die Gleichungen (5.7) und (5.9) stellen die Newton-Euler-Gleichungen für einen Starrkörper in inertialfesten Koordinaten dar.

5.1.2 Newton-Euler-Gleichungen in körperfesten Koordinaten

Oftmals ist es hilfreich, die Newton-Euler-Gleichungen in den körperfesten Geschwindigkeiten ${}^c\mathbf{v}_c = {}^c\mathbf{v}_c^w$ und ${}^c\boldsymbol{\omega}_c = {}^c\boldsymbol{\omega}_c^w$ anzuschreiben. Zwischen den inertialfesten und körperfesten Geschwindigkeiten besteht der folgende Zusammenhang

$$\begin{aligned} \mathbf{v}_c &= {}^w\mathbf{R}_c {}^c\mathbf{v}_c \\ \boldsymbol{\omega}_c &= {}^w\mathbf{R}_c {}^c\boldsymbol{\omega}_c . \end{aligned}$$

Einsetzen in (5.7) ergibt

$$m {}^w\mathbf{R}_c {}^c\dot{\mathbf{v}}_c + m {}^w\dot{\mathbf{R}}_c {}^c\mathbf{v}_c = {}^w\mathbf{f}_c \quad (5.10)$$

$$m {}^c\dot{\mathbf{v}}_c + m {}^c\mathbf{R}_w {}^w\dot{\mathbf{R}}_c {}^c\mathbf{v}_c = {}^c\mathbf{R}_w {}^w\mathbf{f}_c \quad (5.11)$$

$$m {}^c\dot{\mathbf{v}}_c + m \mathbf{S}({}^c\boldsymbol{\omega}_c) {}^c\mathbf{v}_c = {}^c\mathbf{f}_c , \quad (5.12)$$

wobei die Kraft ${}^w\mathbf{f}_c$ über ${}^c\mathbf{f}_c = {}^c\mathbf{R}_w {}^w\mathbf{f}_c$ in das körperfeste Schwerpunktkoordinatensystem $\{c\}$ gedreht wurde.

Analoges Vorgehen für (5.8) mit $\mathbf{h}_\omega = {}^w\mathbf{R}_c {}^c\mathcal{I}_c {}^c\boldsymbol{\omega}_c$ führt zu

$${}^w\mathbf{R}_c {}^c\mathcal{I}_c {}^c\dot{\boldsymbol{\omega}}_c + {}^w\dot{\mathbf{R}}_c {}^c\mathcal{I}_c {}^c\boldsymbol{\omega}_c = {}^w\mathbf{m}_c \quad (5.13)$$

$${}^c\mathcal{I}_c {}^c\dot{\boldsymbol{\omega}}_c + {}^c\mathbf{R}_w {}^w\dot{\mathbf{R}}_c {}^c\mathcal{I}_c {}^c\boldsymbol{\omega}_c = {}^c\mathbf{R}_w {}^w\mathbf{m}_c \quad (5.14)$$

$${}^c\mathcal{I}_c {}^c\dot{\boldsymbol{\omega}}_c + \mathbf{S}({}^c\boldsymbol{\omega}_c) {}^c\mathcal{I}_c {}^c\boldsymbol{\omega}_c = {}^c\mathbf{m}_c , \quad (5.15)$$

Gleichung (5.12) und (5.15) können mit der räumlichen Geschwindigkeit in körperfesten Koordinaten ${}^c\mathbf{t}_c = (({}^c\mathbf{v}_c)^T, ({}^c\boldsymbol{\omega}_c)^T)^T$ zu

$$\underbrace{\begin{bmatrix} m\mathbf{E} & \mathbf{0} \\ \mathbf{0} & {}^c\mathcal{I}_c \end{bmatrix}}_{\mathcal{M}_c} {}^c\mathbf{t}_c + \begin{bmatrix} \mathbf{S}({}^c\boldsymbol{\omega}_c) & \mathbf{0} \\ \mathbf{0} & \mathbf{S}({}^c\boldsymbol{\omega}_c) \end{bmatrix} \begin{bmatrix} m\mathbf{E} & \mathbf{0} \\ \mathbf{0} & {}^c\mathcal{I}_c \end{bmatrix} {}^c\mathbf{t}_c = {}^c\mathbf{w}_c, \quad (5.16)$$

zusammengefasst werden, wobei die Kraft ${}^c\mathbf{f}_c$ und das Drehmoment ${}^c\mathbf{m}_c$ zu einem 6D-Vektor

$${}^c\mathbf{w}_c = \begin{pmatrix} {}^c\mathbf{f}_c \\ {}^c\mathbf{m}_c \end{pmatrix} \quad (5.17)$$

zusammengefasst wurden. In der englischsprachigen Literatur wird dieser sechsdimensionale Kraftvektor als *body wrench*¹ bezeichnet.

Bemerkung 5.1. In Kapitel 3 wurde gezeigt, dass zwischen dem body twist zweier körperfester Koordinatensystemen $\{k\}$ und $\{c\}$ der Zusammenhang ${}^k\mathbf{t}_k = \mathbf{Ad}({}^k\mathbf{H}_c) {}^c\mathbf{t}_c$ besteht. Aus der Forderung, dass ein Paar ${}^c\mathbf{t}_c, {}^c\mathbf{w}_c$ in beiden Koordinatensystemen die gleiche Leistung ergeben soll, d.h. $({}^c\mathbf{t}_c)^T {}^c\mathbf{w}_c = ({}^k\mathbf{t}_k)^T {}^k\mathbf{w}_k$ erhält man direkt den Zusammenhang

$${}^c\mathbf{w}_c = \mathbf{Ad}({}^k\mathbf{H}_c)^T {}^k\mathbf{w}_k. \quad (5.18)$$

Die Matrix $\mathcal{M}_c \in \mathbb{R}^{6 \times 6}$ in (5.16) ist die *räumliche Trägheitsmatrix* des Starrkörpers in seinem Schwerpunkt. Die kinetische Energie \mathcal{T} des Starrkörpers mit der generalisierten Geschwindigkeit ${}^c\mathbf{t}_c$ ist durch

$$\mathcal{T} = \frac{1}{2} ({}^c\mathbf{t}_c)^T \mathcal{M}_c {}^c\mathbf{t}_c \quad (5.19)$$

gegeben. Durch Transformation des Twists ${}^c\mathbf{t}_c$ in ein anders körperfestes Koordinatensystem $\{k\}$ über ${}^k\mathbf{t}_k = \mathbf{Ad}({}^k\mathbf{H}_c) {}^c\mathbf{t}_c$ erhält man die räumliche Trägheitsmatrix und die kinetische Energie zu

$$\mathcal{T} = \frac{1}{2} ({}^k\mathbf{t}_k)^T \mathcal{M}_k {}^k\mathbf{t}_k \quad (5.20)$$

$$\mathcal{M}_k = \mathbf{Ad}({}^c\mathbf{H}_k)^T \mathcal{M}_c \mathbf{Ad}({}^c\mathbf{H}_k) \quad (5.21)$$

$$= \begin{bmatrix} m\mathbf{E} & m\mathbf{S}(\mathbf{c})^T \\ m\mathbf{S}(\mathbf{c}) & {}^c\mathcal{I}_c + m\mathbf{S}(\mathbf{c})^T \mathbf{S}(\mathbf{c}) \end{bmatrix} \quad (5.22)$$

Abbildung 5.2 gibt einen anschaulichen Überblick zu den Transformationen der räumlichen Geschwindigkeiten (Twists) und Kräften (Wrenches) zwischen zwei körperfesten Koordinatensystemen.

¹Im Deutschen gibt es hierfür den Begriff *Kraftwinder*. Dieser ist jedoch nicht sehr weit verbreitet.

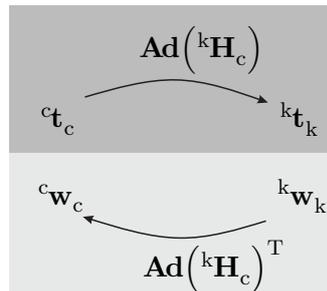


Abbildung 5.2: Transformation der räumlichen Geschwindigkeiten (Twists) und Kräfte (Wrenches) zwischen zwei körperfesten Koordinatensystemen. Die Kräfte werden „dual“ zu den Geschwindigkeiten transformiert, so dass die Leistung in beiden Koordinatensystemen gleich ist, d.h. $({}^c \mathbf{t}_c)^T {}^c \mathbf{w}_c = ({}^k \mathbf{t}_k)^T {}^k \mathbf{w}_k$.

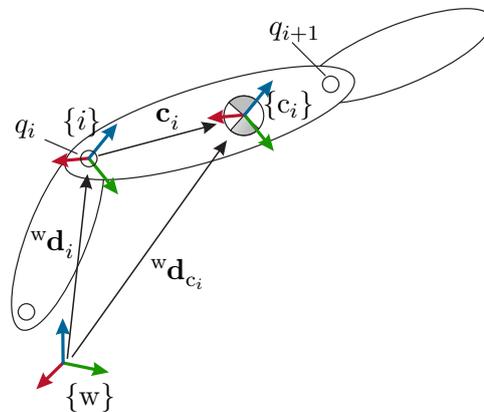


Abbildung 5.3: Skizze zur Formulierung der kinetischen und potentiellen Energie der Glieder.

5.2 Lagrange-Formulierung

Im Folgenden wird ein Roboter mit n generalisierten Koordinaten q_i (für $i = 1 \dots n$) betrachtet. Im Falle eines Drehgelenks ist q_i der Gelenkwinkel, und für ein Schubgelenk, die Linearbewegung des Schubgelenks. Der Roboter besteht aus n Gliedern² welche über die Gelenke miteinander verbunden sind. Die Glieder werden als unverformbare Starrkörper modelliert. In jedem Glied i wird ein körperfestes Koordinatensystem $\{i\}$ festgelegt (siehe Abbildung 5.3). Zusätzlich wird in jedem Körper ein körperfestes Koordinatensystem $\{c_i\}$ in den Schwerpunkt gelegt.

5.2.1 Kinetische und potentielle Energie

Die kinetische Energie \mathcal{T} ergibt sich aus der Summe der Beiträge aller n Glieder

²Das feststehende Glied 0 in der Roboterbasis wird in der Dynamikbeschreibung nicht berücksichtigt.

$$\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^n \frac{1}{2} ({}^i\mathbf{t}_i)^T \mathcal{M}_i {}^i\mathbf{t}_i . \quad (5.23)$$

Über die Body-Jacobi-Matrix (siehe Definition 4.29) wird zu jedem Koordinatensystem $\{i\}$ der body twist ${}^i\mathbf{t}_i$ bestimmt. Um eine einheitliche Dimension aller Body-Jacobi-Matrizen als $6 \times n$ -Matrizen zu erhalten, sind die letzten $n - i$ Spalten durch Nullvektoren aufzufüllen. Mit der Formel

$${}^i\mathbf{t}_i = \mathbf{J}_i(\mathbf{q})\dot{\mathbf{q}} \quad (5.24)$$

ergibt sich die gesamte kinetische Energie zu

$$\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^n \frac{1}{2} \dot{\mathbf{q}}^T \underbrace{\mathbf{J}_i(\mathbf{q})^T \mathcal{M}_i \mathbf{J}_i(\mathbf{q})}_{\mathbf{M}(\mathbf{q})} \dot{\mathbf{q}} . \quad (5.25)$$

Hierbei ist $\mathbf{M}(\mathbf{q})$ die symmetrisch und positive definite *Massenmatrix*.

Die potentielle Energie \mathcal{V} ergibt sich aus der Summe der Gravitationspotentiale aller Glieder. Mit der Position ${}^w\mathbf{d}_i(\mathbf{q})$ und Orientierung ${}^w\mathbf{R}_i(\mathbf{q})$ des körperfesten Koordinatensystems $\{i\}$ (berechnet über die Vorwärtskinematik aus Kapitel 4.1.1) erhält man

$$\mathcal{V}(\mathbf{q}) = \mathcal{V}_g(\mathbf{q}) = \sum_{i=1}^n -m_i g \mathbf{e}_g^T \underbrace{({}^w\mathbf{d}_i(\mathbf{q}) + {}^w\mathbf{R}_i(\mathbf{q})\mathbf{c}_i)}_{{}^w\mathbf{d}_{c_i}(\mathbf{q})} , \quad (5.26)$$

wobei g und \mathbf{e}_g Betrag und Richtung der Gravitationsbeschleunigung darstellen.

5.2.2 Lagrange-Gleichungen

Die Lagrangefunktion $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ ist definiert als Differenz zwischen kinetischer und potentieller Energie:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) := \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{V}(\mathbf{q}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} - \mathcal{V}_g(\mathbf{q}) . \quad (5.27)$$

Die Bewegungsgleichungen erhält man aus den Lagrange-Gleichungen

$$\frac{d}{dt} \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_i} = Q_i \quad \forall i = 1, \dots, n \quad (5.28)$$

wobei $Q_i \in \mathbb{R}$ eine generalisierte Kraft in Gelenk i darstellt. Für ein rotatorisches Gelenk ist Q_i ein Drehmoment, während es für ein Schubgelenk die translatorische Kraft in Richtung der Linearbewegung darstellt. Die n Lagrange-Gleichungen lassen sich in Vektorform wie folgt zusammenfassen:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right)^T - \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} \right)^T = \mathbf{Q} . \quad (5.29)$$

Die generalisierten Kräfte $\mathbf{Q} = (Q_1, \dots, Q_n)^T \in \mathbb{R}^n$ setzen sich aus den generalisierten Antriebsmomenten $\boldsymbol{\tau}$ und weiteren, auf den Roboter extern einwirkenden, generalisierten Kräften $\boldsymbol{\tau}_{ext}$ (z.B. Kräften aus der Interaktion mit der Umgebung) zusammen, d.h. $\mathbf{Q} = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}$. Durch Einsetzen von kinetischer und potentieller Energie erhält man die Bewegungsgleichungen in der Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad (5.30)$$

mit dem nichtlinearen geschwindigkeitsabhängigen Term $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ und den Gravitationsmomenten $\mathbf{g}(\mathbf{q})$

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} - \frac{1}{2} \left(\frac{\partial}{\partial \mathbf{q}} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right)^T \quad (5.31)$$

$$\mathbf{g}(\mathbf{q}) = \left(\frac{\partial \mathcal{V}_g(\mathbf{q})}{\partial \mathbf{q}} \right)^T. \quad (5.32)$$

Der Term $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ beinhaltet Zentrifugal- und Corioliskräfte und kann weiter analysiert werden. Der i -te Eintrag von $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ ist

$$\begin{aligned} n_i(\mathbf{q}, \dot{\mathbf{q}}) &= \sum_{j=1}^n \dot{m}_{ij}(\mathbf{q}) \dot{q}_j - \frac{1}{2} \frac{\partial}{\partial q_i} \left(\sum_{j=1}^n \dot{q}_j \sum_{k=1}^n m_{jk}(\mathbf{q}) \dot{q}_k \right) \\ &= \sum_{j=1}^n \sum_{k=1}^n \frac{\partial m_{ij}(\mathbf{q})}{\partial q_k} \dot{q}_k \dot{q}_j - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial m_{jk}(\mathbf{q})}{\partial q_i} \dot{q}_k \dot{q}_j \\ &= \sum_{j=1}^n \sum_{k=1}^n \underbrace{\frac{1}{2} \left(\frac{\partial m_{ij}(\mathbf{q})}{\partial q_k} + \frac{\partial m_{ik}(\mathbf{q})}{\partial q_j} - \frac{\partial m_{jk}(\mathbf{q})}{\partial q_i} \right)}_{:=\Gamma_{ijk}(\mathbf{q})} \dot{q}_k \dot{q}_j. \end{aligned} \quad (5.33)$$

Die Ausdrücke $\Gamma_{ijk}(\mathbf{q})$ werden als *Christoffelsymbole* bezeichnet. Wie aus (5.33) ersichtlich ist, lassen sich die Zentrifugal- und Coriolisterme mit einer Matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, deren Elemente über

$$C_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{k=1}^n \Gamma_{ijk}(\mathbf{q}) \dot{q}_k \quad (5.34)$$

gegeben sind, wie folgt ausdrücken

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}. \quad (5.35)$$

Die Matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ wird in der Literatur mitunter auch als *Coriolismatrix* bezeichnet - ein allerdings etwas irreführender Name, wenn man bedenkt, dass $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ neben den Coriolistermen ja auch Zentrifugalterme beinhaltet.

Somit erhält man die Bewegungsgleichungen in der endgültigen Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}. \quad (5.36)$$

5.2.3 Berücksichtigung der Rotorträgheiten

In der Berechnung der Lagrange-Gleichungen wurde angenommen, dass die Massen der Rotoren der Antriebsmotoren bereits in den Dynamikparametern der Glieder berücksichtigt wurde. Da sich der Rotor aber aufgrund der Getriebeübersetzung sehr schnell dreht, muss (wie in Kapitel 2 erläutert) die Getriebeübersetzung in der effektiven Rotorträgheit mitberücksichtigt werden. Die einfachste Art diese erhöhte Rotorträgheit in dem Modell zu berücksichtigen, ist, die Diagonalelemente der Massenmatrix $m_{ii}(\mathbf{q})$ um die effektiven Rotorträgheiten J_i zu ergänzen, $m_{ii}(\mathbf{q}) \rightarrow m_{ii}(\mathbf{q}) + J_i$. Dieses Vorgehen ist durchaus üblich. Es sei jedoch angemerkt, dass dabei die gyroskopischen Rotorkräfte vernachlässigt werden, d.h. man approximiert die kinetische Energie des Rotors durch $\frac{1}{2}J_i\dot{q}_i^2$ und vernachlässigt den Anteil der Starrkörperbewegung des Roboters. Diese Approximation führt für Roboter mit sehr großen Getriebeübersetzungen nur zu sehr kleinen Modellfehlern. Eine genauere Modellierung ist durchaus möglich, indem man in der Lagrange-Funktion die gesamte kinetische Energie des Rotors ergänzt.

5.3 Eigenschaften der Dynamikkomponenten

Aus der Herleitung der Bewegungsgleichungen im vorherigen Abschnitt ergeben sich direkt einige wichtige, strukturelle Eigenschaften der Dynamikkomponenten.

Satz 5.1. Die Massenmatrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ ist symmetrisch und positiv definit, d.h.

$$\mathbf{M}(\mathbf{q}) = \mathbf{M}(\mathbf{q})^T > 0 \quad \forall \mathbf{q} \in \mathbb{R}^n .$$

Satz 5.2. Sind alle Gelenke eines Roboters als Drehgelenke ausgeführt - abgesehen von Schubgelenken in der Basis - so sind die Eigenwerte der Massenmatrix nach oben beschränkt, d.h. $\exists \lambda_{min}, \lambda_{max} > 0$ so dass

$$\lambda_{min} < \text{eig}(\mathbf{M}(\mathbf{q})) < \lambda_{max} \quad \forall \mathbf{q} \in \mathbb{R}^n \quad (5.37)$$

gilt.

Die Positivdefinitheit der Massenmatrix stellt eine Generalisierung der Tatsache dar, dass die Masse eines Starrkörpers immer positiv ist. Die Kraft auf den Schwerpunkt eines Starrkörpers ist immer kollinear mit der resultierenden Beschleunigung des Schwerpunktes, und die Trägheit (Masse) ist in alle Richtungen gleich (siehe (5.7)). Die Massenmatrix eines Roboters hingegen repräsentiert eine unterschiedliche Trägheit in verschiedene Beschleunigungsrichtungen, und $\ddot{\mathbf{q}}$ ist (auch im Fall $\dot{\mathbf{q}} = \mathbf{0}, \mathbf{g}(\mathbf{q}) = \mathbf{0}$) nicht kollinear mit $\boldsymbol{\tau}$. Um die Richtungsabhängigkeit der Trägheit zu visualisieren, kann man untersuchen wie der Einheitsball der Gelenkbeschleunigungen $\{\ddot{\mathbf{q}} | \ddot{\mathbf{q}}^T \ddot{\mathbf{q}} = 1\}$ über die Massenmatrix $\mathbf{M}(\mathbf{q})$ in generalisierte Gelenkmomente abgebildet wird. Abbildung 5.4 zeigt die resultierende Ellipse der generalisierten Kräfte für das Beispiel eines Roboters mit zwei Gelenken. Nur entlang der Hauptachsen der Trägheitsellipse sind $\ddot{\mathbf{q}}$ und $\boldsymbol{\tau}$ kollinear. Aufgrund der Konfigurationsabhängigkeit der Massenmatrix, ändert die Trägheitsellipse über den Arbeitsraum ihre Form.

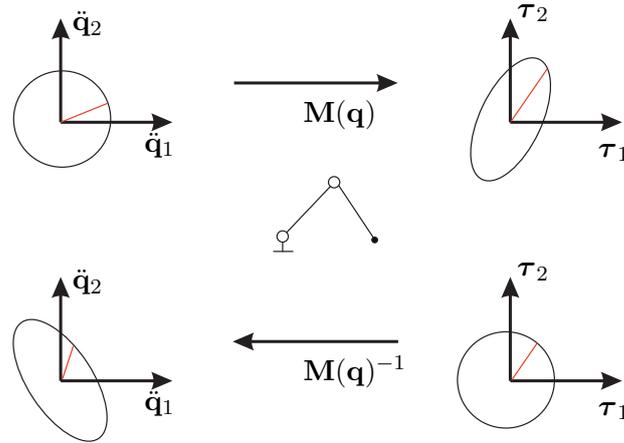


Abbildung 5.4: Skizze zur Interpretation der Massenmatrix. Die Trägheitsellipse gibt an wie der Einheitsball der Gelenkbeschleunigungen (für $\dot{\mathbf{q}} = \mathbf{0}, \mathbf{g}(\mathbf{q}) = \mathbf{0}$) über $\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}$ in generalisierte Gelenkmomente abgebildet wird.

Satz 5.3 (Passivitätseigenschaft). Die Matrix der Zentrifugal- und Coriolisterme $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ erfüllt die Eigenschaft

$$\dot{\mathbf{q}}^T (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{q}} = 0 \quad \forall \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n \quad (5.38)$$

Gleichung (5.38) entspricht der Eigenschaft, dass die Gesamtenergie $\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{V}(\mathbf{q})$ des Roboters sich nur durch Anwendung generalisierter Kräfte $\boldsymbol{\tau} + \boldsymbol{\tau}_{ext}$ ändert:

$$\begin{aligned} \frac{d}{dt} \mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{d}{dt} \left(\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right) + \dot{\mathcal{V}}_g(\mathbf{q}) \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{\partial \mathcal{V}_g(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} + \boldsymbol{\tau}_{ext} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})) + \mathbf{g}(\mathbf{q})^T \dot{\mathbf{q}} \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} + \boldsymbol{\tau}_{ext} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}) \\ &= \dot{\mathbf{q}}^T \left(\frac{1}{2} \dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} + \boldsymbol{\tau}_{ext}) \\ &= \dot{\mathbf{q}}^T (\boldsymbol{\tau} + \boldsymbol{\tau}_{ext}) . \end{aligned}$$

Satz 5.4. Wird die Matrix der Zentrifugal- und Coriolisterme $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ entsprechend (5.34) über die Christoffelsymbole generiert, so gilt die Schiefsymmetrie der Matrix $(\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}))$, d.h.

$$\mathbf{v}^T (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \mathbf{v} = 0 \quad \forall \mathbf{v} \in \mathbb{R}^n \quad (5.39)$$

Die Schiefsymmetrie der Matrix $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ ist äquivalent zu

$$\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})^T .$$

Beweis. Zu zeigen ist die Schiefsymmetrie der Matrix $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$. Die Einträge dieser Matrix ergeben sich mit (5.34) zu

$$\begin{aligned} s_{ij}(\mathbf{q}, \dot{\mathbf{q}}) &= \sum_{k=1}^n \frac{\partial m_{ij}(\mathbf{q})}{\partial q_k} \dot{q}_k - 2 \sum_{k=1}^n \Gamma_{ijk}(\mathbf{q}) \dot{q}_k \\ &= \sum_{k=1}^n \left(\frac{\partial m_{jk}(\mathbf{q})}{\partial q_i} - \frac{\partial m_{ik}(\mathbf{q})}{\partial q_j} \right) \dot{q}_k , \end{aligned}$$

woraus die Eigenschaft $s_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = -s_{ji}(\mathbf{q}, \dot{\mathbf{q}})$ direkt ersichtlich ist. \square

Man beachte, dass die Faktorisierung der Zentrifugal- und Coriolisterme $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ nicht eindeutig ist, und die Eigenschaft (5.39) nur für die spezielle Wahl (5.34) gilt. Die Passivitätseigenschaft (5.38) gilt hingegen für jede beliebige Wahl von $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$.

Die Dynamikgleichung hängt von den Elementen der räumlichen Trägheitsmatrizen \mathcal{M}_i der einzelnen Glieder ab. Für jedes Glied sind dies zehn unabhängige Parameter, und zwar die Masse $m_i \in \mathbb{R}$, das erste mechanische Moment $m_i \mathbf{c}_i \in \mathbb{R}^3$ und die 6 unabhängigen Einträge der Rotationsträgheit bezüglich $\{i\}$: $(I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz})$. Fasst man alle diese Parameter zusammen, so erhält man einen $10n$ -dimensionalen Parametervektor $\boldsymbol{\pi} \in \mathbb{R}^{10n}$.

Satz 5.5 (Linearität in den Starrkörperparametern). Die Dynamikgleichung hängt in linearer Weise von den Starrkörperparametern $\boldsymbol{\pi}$ ab, lässt sich also in der Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi} \quad (5.40)$$

schreiben. Die Matrix $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n \times (10n)}$ wird als Regressormatrix bezeichnet.

Die Linearität der Dynamikgleichungen in den Starrkörperparametern $\boldsymbol{\pi}$ ist eine wichtige Eigenschaft für die Parameteridentifikation und die adaptive Regelung. Es ist jedoch anzumerken, dass der $10n$ -dimensionale Vektor $\boldsymbol{\pi}$ keinen minimalen Satz von Parametern darstellt. Zur Identifikation und adaptiven Regelung wird der Parametersatz daher zunächst auf einen minimalen Satz unabhängiger Parameter reduziert. Hierfür kommen sowohl analytische, wie auch numerische Methoden zum Einsatz.

Die folgenden beiden Eigenschaften sind in der Stabilitätsanalyse nichtlinearer Regler und Zustandsbeobachter nützlich.

Satz 5.6. Die Coriolismatrix (5.34) erfüllt die Bedingung

$$\mathbf{C}(\mathbf{q}, \mathbf{v})\mathbf{w} = \mathbf{C}(\mathbf{q}, \mathbf{w})\mathbf{v} \quad \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^n .$$

Satz 5.7. Für Roboter, in denen, abgesehen von translatorischen Gelenken in der Basis, alle Gelenke als Drehgelenke ausgeführt sind, können endliche Werte $\alpha >$

$0, \beta > 0, \gamma > 0$ gefunden werden, so dass die folgenden Ungleichungen erfüllt sind.

$$\begin{aligned}\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\| &\leq \alpha \|\dot{\mathbf{q}}\|^2 \\ \|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\| &\leq \beta \|\dot{\mathbf{q}}\| \\ \|\mathbf{g}(\mathbf{q})\| &\leq \gamma\end{aligned}$$

5.4 Rekursive Newton-Euler-Formulierung

Die Lagrange-Formulierung aus dem letzten Kapitel eignet sich für Systeme mit wenigen Freiheitsgraden. Für komplexe Robotersysteme mit vielen Freiheitsgraden ($n > 7$) wird die analytische Berechnung der Ableitungen sehr umfangreich und auch eine computergestützte Vereinfachung der Ausdrücke wird mit steigender Zahl der Freiheitsgrade zunehmend inpraktikabel. In diesen Fällen eignen sich rekursive Algorithmen, wie der rekursive Newton-Euler-Algorithmus (RNEA), besser. Dieser basiert auf einer Formulierung der Newton- und Euler-Gleichung für jeden einzelnen Körper. Die Berechnung erfolgt hier in zwei Schritten, einer Vorwärtsiteration der Kinematik und einer Rückwärtsiteration der Kräfte.

5.4.1 Formulierung im Inertialsystem

Im Folgenden wird die Herleitung des rekursiven Newton-Euler-Algorithmus der Dynamik in inertialfesten Größen gezeigt. Es wird angenommen, dass jedem Körper ein körperfestes Koordinatensystem $\{i\}$ zugeordnet ist, dessen Ursprung auf der Achse von Gelenk i liegt. Wir beschränken uns auf den Fall einer seriellen Roboterkinematik³ (d.h. jeder Körper i hat in der kinematischen Kette genau einen Vorgänger $i - 1$ und ist über Gelenk i mit diesem verbunden), sowie auf eindimensionale prismatische und rotatorische Gelenke. Die Gelenkachse wird durch den Vektor ${}^w\mathbf{a}_i \in \mathbb{R}^3$ bezeichnet. Ausgedrückt in $\{i\}$ ist die Gelenkachse konstant ${}^i\mathbf{a}_i = {}^i\mathbf{R}_w {}^w\mathbf{a}_i$. Für jedes Gelenk gibt der Wert $\sigma_i \in \{0, 1\}$ an, ob das Gelenk prismatisch ($\sigma_i = 0$) oder rotatorisch ($\sigma_i = 1$) ist. Zur Vereinfachung der Schreibweise definieren wir auch den komplementären Wert $\bar{\sigma}_i = 1 - \sigma_i$.

In Abschnitt 5.1 wurden die Newton-Euler-Gleichungen bezüglich eines Koordinatensystems $\{c_i\}$ im Schwerpunkt eingeführt. Diese werden nun zunächst mit Bezug auf das Koordinatensystem $\{i\}$ transformiert. Der Ursprung und die Orientierung von $\{i\}$ bezüglich des Inertialsystems $\{w\}$ werden mit ${}^w\mathbf{d}_i$ und ${}^w\mathbf{R}_i$ bezeichnet. Abbildung 5.5(a) zeigt die wirkenden Kräfte im Schwerpunkt, bzw. im Koordinatensystem $\{i\}$. Für die Translation erhält man

$$\begin{aligned}{}^w\mathbf{d}_{c_i} &= {}^w\mathbf{d}_i + {}^w\mathbf{c}_i \\ \mathbf{v}_{c_i} &= \mathbf{v}_i + \frac{d}{dt}({}^w\mathbf{R}_i \mathbf{c}_i) \\ &= \mathbf{v}_i + \mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathbf{c}_i\end{aligned}$$

³Der resultierende Algorithmus lässt sich allerdings auch leicht auf baumartige Roboterstrukturen erweitern.

$$\dot{\mathbf{v}}_{c_i} = \dot{\mathbf{v}}_i + \mathbf{S}(\dot{\boldsymbol{\omega}}_i) {}^w\mathbf{c}_i + \mathbf{S}(\boldsymbol{\omega}_i)\mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathbf{c}_i,$$

wobei der Schwerpunktvektor \mathbf{c}_i ausgedrückt im Inertialkoordinatensystem als ${}^w\mathbf{c}_i = {}^w\mathbf{R}_i\mathbf{c}_i$ geschrieben wurde. Durch Einsetzen in (5.7) erhält man mit ${}^w\mathbf{f}_i = {}^w\mathbf{f}_{c_i}$ (da $\{i\}$ und $\{c_i\}$ die gleiche Orientierung haben)

$$\begin{aligned} m_i \dot{\mathbf{v}}_{c_i} &= {}^w\mathbf{f}_{c_i} \\ m_i \dot{\mathbf{v}}_i + m_i \mathbf{S}(\dot{\boldsymbol{\omega}}_i) {}^w\mathbf{c}_i + m_i \mathbf{S}(\boldsymbol{\omega}_i)\mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathbf{c}_i &= {}^w\mathbf{f}_i. \end{aligned} \quad (5.41)$$

Die Euler-Gleichung (5.9) ergibt sich mit ${}^w\mathbf{m}_i = {}^w\mathbf{m}_{c_i} + \mathbf{S}({}^w\mathbf{c}_i) {}^w\mathbf{f}_{c_i}$ zu

$$\begin{aligned} {}^w\mathcal{I}_i \dot{\boldsymbol{\omega}}_i + \mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathcal{I}_i \boldsymbol{\omega}_i &= {}^w\mathbf{m}_{c_i} \\ {}^w\mathcal{I}_i \dot{\boldsymbol{\omega}}_i + \mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathcal{I}_i \boldsymbol{\omega}_i + \mathbf{S}({}^w\mathbf{c}_i) {}^w\mathbf{f}_{c_i} &= {}^w\mathbf{m}_i \\ {}^w\mathcal{I}_i \dot{\boldsymbol{\omega}}_i + \mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathcal{I}_i \boldsymbol{\omega}_i + m_i \mathbf{S}({}^w\mathbf{c}_i)(\dot{\mathbf{v}}_i + \mathbf{S}(\dot{\boldsymbol{\omega}}_i) {}^w\mathbf{c}_i + \mathbf{S}(\boldsymbol{\omega}_i)\mathbf{S}(\boldsymbol{\omega}_i) {}^w\mathbf{c}_i) &= {}^w\mathbf{m}_i \\ \underbrace{({}^w\mathcal{I}_i + m_i \mathbf{S}({}^w\mathbf{c}_i) {}^T \mathbf{S}({}^w\mathbf{c}_i))}_{{}^w\bar{\mathcal{I}}_i} \dot{\boldsymbol{\omega}}_i + \mathbf{S}(\boldsymbol{\omega}_i) {}^w\bar{\mathcal{I}}_i \boldsymbol{\omega}_i + m_i \mathbf{S}({}^w\mathbf{c}_i) \dot{\mathbf{v}}_i &= {}^w\mathbf{m}_i \end{aligned} \quad (5.42)$$

Zur Ermittlung der letzten Zeile wurde die Jacobi-Identität⁴ des Kreuzproduktes ausgenutzt.

1) Vorwärtsrekursion der Kinematik: Zur Auswertung von (5.41) und (5.42) werden $\dot{\boldsymbol{\omega}}_i$, $\boldsymbol{\omega}_i$ und $\dot{\mathbf{v}}_i$ benötigt. Diese Größen können iterativ berechnet werden. Die Winkelgeschwindigkeiten der rotatorischen Gelenke geben die relativen Winkelgeschwindigkeiten zwischen aufeinanderfolgenden Gliedern vor:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \sigma_i \dot{q}_i {}^w\mathbf{a}_i \quad (5.43)$$

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \sigma_i (\ddot{q}_i {}^w\mathbf{a}_i + \dot{q}_i \mathbf{S}(\boldsymbol{\omega}_{i-1}) {}^w\mathbf{a}_i) \quad (5.44)$$

Für die translatorischen Größen erhält man mit ${}^w\mathbf{d}_i^{i-1} \in \mathbb{R}^3$ als Vektor zwischen $\{i-1\}$ und $\{i\}$ (siehe Abbildung 5.5)

$$\begin{aligned} {}^w\mathbf{d}_i &= {}^w\mathbf{d}_{i-1} + {}^w\mathbf{d}_i^{i-1} \\ \mathbf{v}_i &= \mathbf{v}_{i-1} + \mathbf{S}(\boldsymbol{\omega}_{i-1}) {}^w\mathbf{d}_i^{i-1} + \bar{\sigma}_i \dot{q}_i {}^w\mathbf{a}_i \\ \dot{\mathbf{v}}_i &= \dot{\mathbf{v}}_{i-1} + \mathbf{S}(\dot{\boldsymbol{\omega}}_{i-1}) {}^w\mathbf{d}_i^{i-1} + \mathbf{S}(\boldsymbol{\omega}_{i-1})\mathbf{S}(\boldsymbol{\omega}_{i-1}) {}^w\mathbf{d}_i^{i-1} + \\ &\quad \bar{\sigma}_i (\ddot{q}_i {}^w\mathbf{a}_i + 2\dot{q}_i \mathbf{S}(\boldsymbol{\omega}_{i-1}) {}^w\mathbf{a}_i). \end{aligned} \quad (5.45)$$

Die Gleichungen (5.43), (5.44) und (5.45) werden iterativ für alle Glieder $i = 1 \dots n$ berechnet. Als Initialwerte der Iteration setzt man (für eine stillstehende Roboterbasis) $\boldsymbol{\omega}_0 = \mathbf{0}$, $\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$ und $\dot{\mathbf{v}}_0 = \mathbf{0}$.

2) Rekursive Berechnung der Kräfte: Mit den Ergebnissen der kinematischen Vorwärtsrekursion können über (5.41) und (5.42) die Trägheitskräfte ${}^w\mathbf{f}_i$ und Momente ${}^w\mathbf{m}_i$ aller Glieder berechnet werden. Die in den Gelenken zwischen zwei Gliedern wirkenden Kräfte und Momente ${}^w\mathbf{f}_i$ und ${}^w\bar{\mathbf{m}}_i$ werden nun über eine Rückwärtsiteration

⁴Man beachte, dass das Kreuzprodukt nicht assoziativ ist. Die Jacobi-Identität besagt $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) + \mathbf{b} \times (\mathbf{c} \times \mathbf{a}) + \mathbf{c} \times (\mathbf{a} \times \mathbf{b}) = \mathbf{0}$ für drei Vektoren $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$. Hieraus folgt $\mathbf{S}(\mathbf{c})\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\boldsymbol{\omega})\mathbf{c} = -\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\mathbf{S}(\boldsymbol{\omega})\mathbf{c})\mathbf{c} - \mathbf{S}(\mathbf{S}(\boldsymbol{\omega})\mathbf{c})\mathbf{S}(\boldsymbol{\omega})\mathbf{c} = -\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\mathbf{c})\mathbf{S}(\boldsymbol{\omega})\mathbf{c} = \mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\mathbf{c}) {}^T \mathbf{S}(\boldsymbol{\omega})\mathbf{c}$.

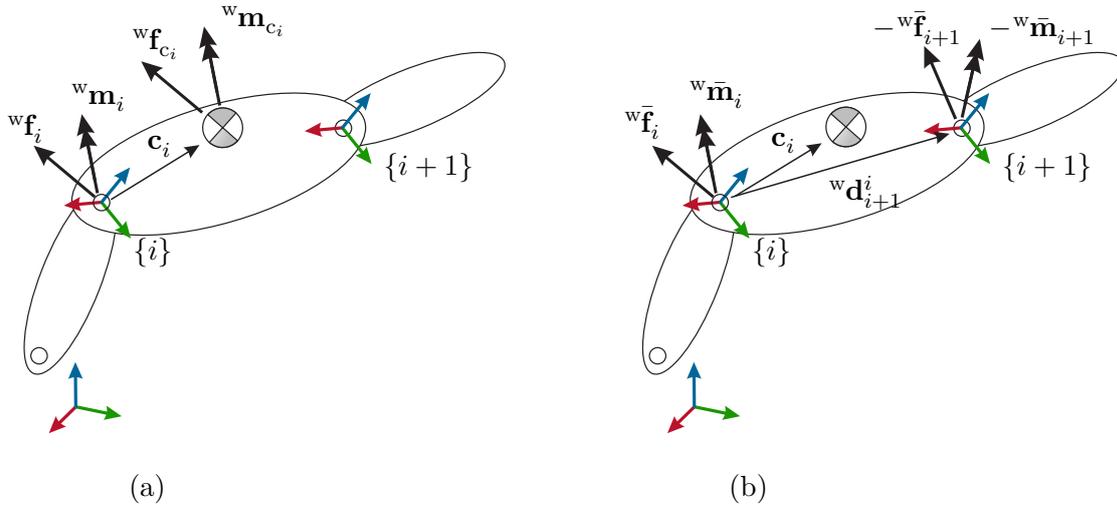


Abbildung 5.5: Skizze zur Transformation der Kräfte im Newton-Euler-Formalismus. (a) Zwischen den wirkenden Kräften und Momenten im Schwerpunkt und deren Repräsentation in $\{i\}$ gilt der Zusammenhang ${}^w\mathbf{f}_i = {}^w\mathbf{f}_{c_i}$ und ${}^w\mathbf{m}_i = {}^w\mathbf{m}_{c_i} + \mathbf{S}({}^w\mathbf{c}_i){}^w\mathbf{f}_{c_i}$. (b) In den Gelenken wirkende Kräfte und Momente. Im Gelenk i wirkt die Kraft ${}^w\bar{\mathbf{f}}_i$ auf Glied i , während die Reaktionskraft $-{}^w\bar{\mathbf{f}}_i$ auf Glied $i - 1$ wirkt.

$i = n \dots 1$ bestimmt. Hierbei bezeichnen ${}^w\bar{\mathbf{f}}_i$ und ${}^w\bar{\mathbf{m}}_i$ jene Kräfte und Momente, die auf das Glied i wirken. Die entsprechenden Reaktionskräfte und -momente auf Glied $i - 1$ sind durch $-{}^w\bar{\mathbf{f}}_i$ und $-{}^w\bar{\mathbf{m}}_i$ gegeben (siehe dazu auch Abbildung 5.5(b)). Neben diesen Kräften wirkt auch noch die Gravitationskraft $m_i g \mathbf{e}_g$. Somit erhält man das Kräfte- und Momentengleichgewicht zu

$${}^w\mathbf{f}_i = {}^w\bar{\mathbf{f}}_i - {}^w\bar{\mathbf{f}}_{i+1} + m_i g \mathbf{e}_g \quad (5.46)$$

$${}^w\mathbf{m}_i = {}^w\bar{\mathbf{m}}_i - {}^w\bar{\mathbf{m}}_{i+1} - \mathbf{S}({}^w\mathbf{d}_{i+1}^i){}^w\bar{\mathbf{f}}_{i+1} + m_i g \mathbf{S}({}^w\mathbf{c}_i)\mathbf{e}_g \quad (5.47)$$

Weitere, wirkende externe Kräfte und Momente können auf die selbe Art und Weise in den Formalismus integriert werden wie die Gravitationskraft. Die Gravitationskraft kann auch indirekt über eine Beschleunigung der Roboterbasis, d.h. $\dot{\mathbf{v}}_0 = -g\mathbf{e}_g$, berücksichtigt werden. Die Rekursionsgleichungen ergeben sich in diesem Fall zu

$${}^w\bar{\mathbf{f}}_i = {}^w\bar{\mathbf{f}}_{i+1} + {}^w\mathbf{f}_i \quad (5.48)$$

$${}^w\bar{\mathbf{m}}_i = {}^w\bar{\mathbf{m}}_{i+1} + {}^w\mathbf{m}_i + \mathbf{S}({}^w\mathbf{d}_{i+1}^i){}^w\bar{\mathbf{f}}_{i+1} \quad (5.49)$$

Die Rückwärtsrekursion wird mit ${}^w\bar{\mathbf{f}}_{n+1} = \mathbf{0}$ und ${}^w\bar{\mathbf{m}}_{n+1} = \mathbf{0}$ gestartet. Aus den Ergebnissen für ${}^w\bar{\mathbf{f}}_i$ und ${}^w\bar{\mathbf{m}}_i$ erhält man schließlich das generalisierte Gelenkmoment zu

$$\tau_i = (\bar{\sigma}_i {}^w\bar{\mathbf{f}}_i + \sigma_i {}^w\bar{\mathbf{m}}_i)^T {}^w\mathbf{a}_i. \quad (5.50)$$

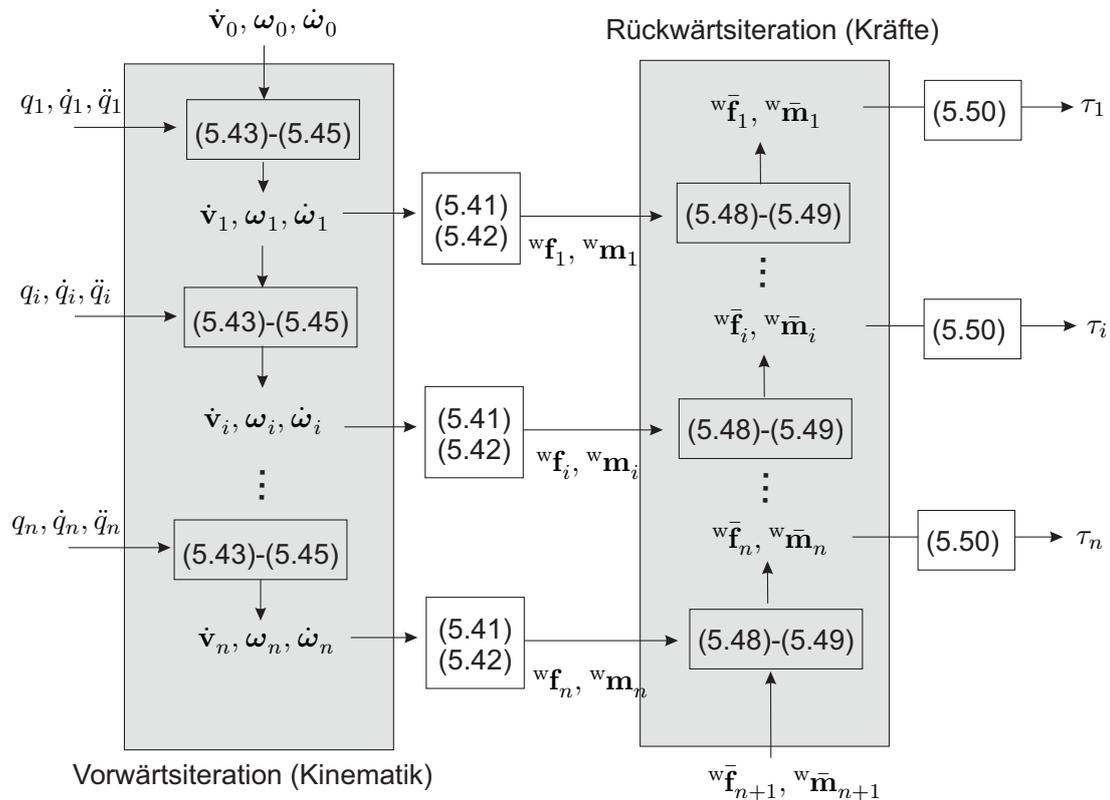


Abbildung 5.6: Berechnungsstruktur des rekursiven Newton-Euler-Algorithmus bestehend aus einer Vorwärtsiteration der Kinematikgleichungen und einer Rückwärtsiteration der Kräfte.

Der gesamte Algorithmus besteht somit aus den Gleichungen (5.43), (5.44) und (5.45) für die Vorwärtsiteration, sowie Gleichung (5.48), (5.49) und (5.50) für die Rückwärtsiteration. Abbildung 5.6 zeigt die Berechnungsstruktur in diesen beiden Iterationen.

Der rekursive Newton-Euler-Algorithmus führt auf eine Berechnung der inversen Dynamik mit einer Komplexität von $O(n)$. Als Ergebnis erhält man für vorgegebene Vektoren \mathbf{q} , $\dot{\mathbf{q}}$ und $\ddot{\mathbf{q}}$ direkt die Gelenkmomente $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ ohne explizite Berechnung der Dynamikkomponenten $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ und $\mathbf{g}(\mathbf{q})$. Die Rotorträgheiten der Antriebsmotoren können auf die gleiche Weise eingearbeitet werden wie in der Lagrange-Formulierung.

5.4.2 Formulierung in körperfesten Koordinaten

Zur besseren Anschauung der einzelnen Berechnungsschritte im RNEA wurde bisher eine Darstellung in Inertialkoordinaten gewählt. Da die Rotationsträgheit und der Schwerpunkt jedes Gliedes in einem körperfesten Koordinatensystem konstant sind, ist für eine effiziente Implementierung hingegen eine Darstellung in körperfesten Koordinaten vorteilhaft. Als Referenz für eine praktische Implementierung sind im Folgenden die Gleichungen (bewusst ohne detaillierte Herleitung) für eine körperfeste Darstellung angeführt [5.1]. Die

Vorwärtsrekursion für $i = 1 \dots n$ nimmt die folgende Form an:

$$\begin{aligned}
{}^i\boldsymbol{\omega}_{i-1}^w &= {}^i\mathbf{R}_{i-1} {}^{i-1}\boldsymbol{\omega}_{i-1} & (5.51) \\
{}^i\boldsymbol{\omega}_i &= {}^i\boldsymbol{\omega}_{i-1}^w + \sigma_i(\dot{q}_i {}^i\mathbf{a}_i) \\
{}^i\dot{\boldsymbol{\omega}}_i &= {}^i\mathbf{R}_{i-1} {}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} + \sigma_i(\ddot{q}_i {}^i\mathbf{a}_i + \dot{q}_i \mathbf{S}({}^i\boldsymbol{\omega}_{i-1}^w) {}^i\mathbf{a}_i) \\
{}^i\dot{\mathbf{v}}_i &= {}^i\mathbf{R}_{i-1} ({}^{i-1}\dot{\mathbf{v}}_{i-1} + \mathbf{U}_{i-1} {}^i\mathbf{d}_i^{i-1}) + \bar{\sigma}_i(\ddot{q}_i {}^i\mathbf{a}_i + 2\dot{q}_i \mathbf{S}({}^i\boldsymbol{\omega}_{i-1}^w) {}^i\mathbf{a}_i) \\
{}^i\mathbf{f}_i &= m_i {}^i\dot{\mathbf{v}}_i + m_i \mathbf{U}_i \mathbf{c}_i \\
{}^i\mathbf{m}_i &= {}^i\mathcal{Z}_i {}^i\dot{\boldsymbol{\omega}}_i + \mathbf{S}({}^i\boldsymbol{\omega}_i) {}^i\mathcal{Z}_i {}^i\boldsymbol{\omega}_i + m_i \mathbf{S}(\mathbf{c}_i) {}^i\dot{\mathbf{v}}_i \\
\mathbf{U}_i &= \mathbf{S}({}^i\dot{\boldsymbol{\omega}}_i) + \mathbf{S}({}^i\boldsymbol{\omega}_i) \mathbf{S}({}^i\boldsymbol{\omega}_i)
\end{aligned}$$

Die Rückwärtsrekursion für $i = n \dots 1$ ist durch

$$\begin{aligned}
{}^i\bar{\mathbf{f}}_i &= {}^i\mathbf{f}_i + {}^i\bar{\mathbf{f}}_{i+1} \\
{}^{i-1}\bar{\mathbf{f}}_i &= {}^{i-1}\mathbf{R}_i {}^i\bar{\mathbf{f}}_i \\
{}^i\bar{\mathbf{m}}_i &= {}^i\mathbf{m}_i + {}^i\mathbf{R}_{i+1} {}^{i+1}\bar{\mathbf{m}}_{i+1} + \mathbf{S}({}^i\mathbf{d}_{i+1}) {}^i\mathbf{f}_{i+1} \\
\tau_i &= (\bar{\sigma}_i {}^i\bar{\mathbf{f}}_i + \sigma_i {}^i\bar{\mathbf{m}}_i)^\top {}^i\mathbf{a}_i
\end{aligned}$$

gegeben.

Bemerkung 5.2. Der in diesem Abschnitt beschriebene rekursive Newton-Euler-Algorithmus wurde hierin für eine serielle Roboterkinematik formuliert. Er lässt sich jedoch auch sehr leicht auf eine baumartige Roboterstruktur anwenden, indem man in der Vorwärtsrekursion die Reihenfolge der Glieder in der Baumstruktur beachtet und in der Rückwärtsrekursion die Kräfte und Momente über alle nachfolgenden Glieder aufsummiert.

5.5 Vorwärtsdynamik

Die bisher diskutierten Methoden erlauben die Berechnung der Inversdynamik $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. In der Vorwärtsdynamik $\mathbf{fdyn}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau})$ sind die Gelenkmomente $\boldsymbol{\tau}$ und die Zustandsgrößen \mathbf{q} und $\dot{\mathbf{q}}$ gegeben, während die Beschleunigung $\ddot{\mathbf{q}}$ zu ermitteln ist. Grundsätzlich lässt sich aus den Dynamikkomponenten $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ und $\mathbf{g}(\mathbf{q})$ eine Vorwärtsdynamik berechnen, indem man das lineare Gleichungssystem

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})$$

nach den Beschleunigungen $\ddot{\mathbf{q}}$ löst⁵. Hierbei kann der Ausdruck $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$ effizient über den rekursiven Newton-Euler-Algorithmus aus dem vorherigen Abschnitt berechnet werden, indem man darin die Gelenkbeschleunigungen zu null setzt. Ebenso kann die Massenmatrix spaltenweise aus dem Newton-Euler-Algorithmus ermittelt werden, indem man

⁵Eine Berechnung der Vorwärtsdynamik über Matrixinversion sollte vermieden werden, sofern die inverse Massenmatrix nicht auch für andere Anwendungen (z.B. in der Regelung) explizit benötigt wird.

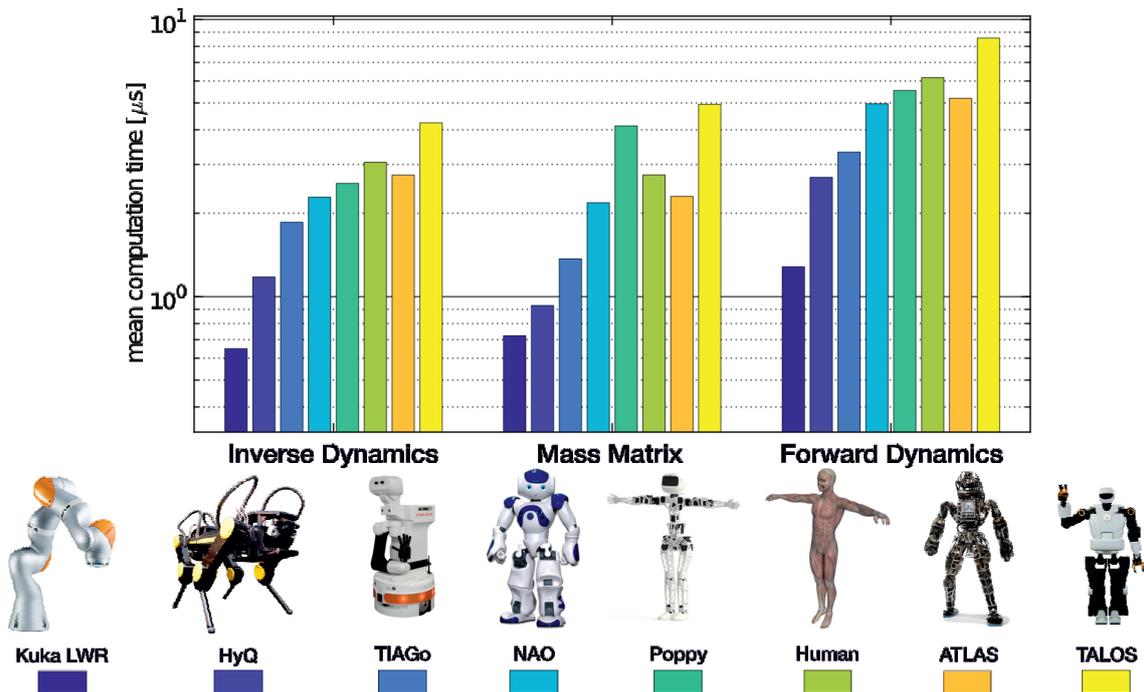


Abbildung 5.7: Vergleich der Ausführungszeiten der Dynamikberechnung mittels der Open Source Dynamikbibliothek Pinocchio (CNRS-LAAS, <https://github.com/stack-of-tasks/pinocchio>). Diese Ergebnisse wurden mit einem Standard-Laptop mit Intel Core i7 CPU @ 2.4 GHz erzielt.

die einzelnen Gelenkbeschleunigungen zu $\ddot{q}_i = 1$ setzt und alle anderen Beschleunigungen sowie $\dot{\mathbf{q}}$ und g gleich null setzt.

Es sei jedoch darauf hingewiesen, dass es für die Vorwärtsdynamik in der Literatur auch effiziente Berechnungsmethoden der Ordnung $O(n)$ gibt. Diese bauen auf einer geschickten Umformung der Gleichungen des Newton-Euler-Formalismus auf, erfordern jedoch 3 Rekursionen, im Gegensatz zur Inversdynamik, welche über eine Vorwärts- und eine Rückwärtsiteration berechnet werden kann [5.1]. Ein bekanntes Beispiel hierfür ist der *Articulated Body Algorithm (ABA)* [5.2]. Zur effizienten Berechnung der Massenmatrix $\mathbf{M}(\mathbf{q})$ findet man Algorithmen mit einer Komplexität von $O(n^2)$ (z.B. *Composite Rigid Body Algorithm*). Viele dieser Algorithmen sind in frei verfügbaren Dynamikbibliotheken implementiert [5.3]. Als Beispiel für die erreichbare Rechenperformanz zeigt Abbildung 5.7 die mit der Open-Source-Dynamik-Bibliothek „Pinocchio“ [5.4] erzielbaren Rechenzeiten der Invers- und Vorwärtsdynamikberechnung für verschiedene Robotermodelle.

5.6 Zwangsbedingungen (Bindungen)

Durch Zwangsbedingungen (auch „Bindungen“, engl. „constraints“) wird die Bewegungsfreiheit eines mechanischen Systems eingeschränkt. In der Robotik sind Zwangsbedingungen in zwei Fällen relevant. Einerseits treten sie auf, wenn der Roboter in Kontakt mit

einer starren Umgebung kommt oder (teilweise) eingespannt ist. Andererseits kann der Mechanismus selbst geschlossene kinematische Ketten beinhalten, welche über zusätzliche Bindungen modelliert werden. Dies ist z.B. bei Parallelrobotern der Fall. Hierbei wird zunächst über die Algorithmen des letzten Abschnittes die Dynamik für eine offene Baumstruktur berechnet. In einem zweiten Schritt werden dann die Bindungen, welche die geschlossenen Ketten beschreiben, eingearbeitet.

Grundsätzlich unterscheidet man zwischen *holonomen* und *nichtholonomen* Zwangsbedingungen. Holonome Zwangsbedingungen schränken die Bewegung des Systems auf eine Hyperebene des Konfigurationsraumes \mathbb{R}^n ein und lassen sich durch eine Vektorfunktion $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^k$ repräsentieren:

$$\phi(\mathbf{q}) = \mathbf{0} . \quad (5.52)$$

Die vektorwertige Funktion $\phi(\mathbf{q})$ beinhaltet k skalare Bindungen $\phi_i(\mathbf{q}) = 0$ für $i = 1, \dots, k$. Im Folgenden wird angenommen, dass diese einzelnen Zwangsbedingungen linear unabhängig sind. Eine Einschränkung des Konfigurationsraumes bedeutet gleichzeitig auch eine Einschränkung der realisierbaren Geschwindigkeiten. Durch zeitliche Differentiation ergibt sich

$$\dot{\phi}(\mathbf{q}) = \frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{0} . \quad (5.53)$$

Die Einschränkung auf Geschwindigkeitsebene kann in der Form

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (5.54)$$

mit $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{k \times n}$ geschrieben werden. Gleichung (5.54) hat die Form einer *Pfaff'schen Zwangsbedingung*, welche linear in den Gelenkwinkelgeschwindigkeiten $\dot{\mathbf{q}}$ ist.

Im Gegensatz dazu schränken *nichtholonome* Zwangsbedingungen „nur“ die Geschwindigkeit des Systems ein, aber nicht die Konfiguration. Ein typisches Beispiel für ein System mit nichtholonomen Bindungen sind mobile Roboter. Auf Geschwindigkeitsebene lassen sich holonome und nichtholonome Zwangsbedingungen in gleicher Art und Weise durch eine Pfaff'sche Zwangsbedingung der Form (5.54) ausdrücken.

Eine weitere Unterscheidung bezieht sich auf die Zeitabhängigkeit. Zeitabhängige Bindungen $\phi(\mathbf{q}, t)$ werden als *rheonom* (griechisch für „fließend“), zeitunabhängige Bindungen werden als *skleronom* (griechisch für „starr“) bezeichnet.

Im Folgenden wird angenommen, dass die Zwangsbedingungen zeitunabhängig und linear unabhängig sind, und somit die Matrix $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{k \times n}$ vollen Zeilenrang besitzt. Die Zwangsbedingung (5.54) wird durch eine externe, auf den Roboter einwirkende, generalisierte Kraft $\boldsymbol{\tau}_\lambda \in \mathbb{R}^n$ bewirkt. Die Zwangskräfte aus skleronomen Zwangsbedingungen leisten per Definition keine Arbeit, d.h. für die Leistung P_λ gilt

$$P_\lambda = \boldsymbol{\tau}_\lambda^T \dot{\mathbf{q}} = 0 .$$

Es ist zweckmäßig, diese Eigenschaft in der Modellierung zu berücksichtigen, indem die generalisierte Kraft $\boldsymbol{\tau}_\lambda$ durch eine Zwangskraft $\boldsymbol{\lambda} \in \mathbb{R}^k$ repräsentiert wird. Durch

$$\boldsymbol{\tau}_\lambda = \mathbf{A}(\mathbf{q})^T \boldsymbol{\lambda} \quad (5.55)$$

wird $P_\lambda = 0$ sichergestellt.

Mit (5.55) nimmt die Dynamikgleichung (5.36) die Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})}_{\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})} = \boldsymbol{\tau} + \mathbf{A}(\mathbf{q})^T \boldsymbol{\lambda} \quad (5.56)$$

an. Gleichsam kann die Zwangsbedingung (5.54) durch zeitliche Ableitung auf Beschleunigungsebene gebracht werden:

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} . \quad (5.57)$$

Die Dynamikgleichungen eines Robotersystems mit Zwangsbedingungen lassen sich damit in der folgenden Form als Gleichungssystem anschreiben:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & -\mathbf{A}(\mathbf{q})^T \\ \mathbf{A}(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} \end{pmatrix} \quad (5.58)$$

Gleichung (5.58) erlaubt durchaus eine explizite Berechnung der Zwangskraft $\boldsymbol{\lambda}$. Hierzu wird aus der oberen Gleichung ein Ausdruck für $\ddot{\mathbf{q}}$ bestimmt und in die untere Gleichung eingesetzt. Dies führt auf

$$\boldsymbol{\lambda} = \left(\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T \right)^{-1} \left\{ \mathbf{A}\mathbf{M}^{-1}(-\boldsymbol{\tau} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})) - \dot{\mathbf{A}}\dot{\mathbf{q}} \right\} . \quad (5.59)$$

Die erforderliche Invertierbarkeit der Matrix $\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T$ ist durch die Annahme von linear unabhängigen Zwangsbedingungen sichergestellt. Man beachte, dass das Vorgehen bisher sowohl für holonome wie auch für nicht-holonome Zwangsbedingungen in gleicher Weise angewendet werden kann.

5.6.1 Baumgarte Stabilisierung holonomer Bindungen

Grundsätzlich könnte man nun bei gegebenen Gelenkmomenten $\boldsymbol{\tau}$ mit Kenntnis der Zwangskraft (5.59) aus (5.56) die Beschleunigung $\ddot{\mathbf{q}}$ ermitteln und diese aufintegrieren, um das Systemverhalten zu simulieren. Dieses Vorgehen ist jedoch sehr sensitiv gegenüber numerischen Berechnungsfehlern und führt durch die zweifache Integration sehr schnell zu einer Verletzung der Zwangsbedingung $\boldsymbol{\phi}(\mathbf{q}) = \mathbf{0}$. Eine Abhilfe verschafft die sogenannte *Baumgarte-Stabilisierung* der Zwangsbedingung. Hierbei wird statt (5.57) die folgende stabile Fehlerdynamik zweiter Ordnung angesetzt

$$\ddot{\boldsymbol{\phi}}(\mathbf{q}) + \alpha\dot{\boldsymbol{\phi}}(\mathbf{q}) + \beta\boldsymbol{\phi}(\mathbf{q}) = \mathbf{0} , \quad (5.60)$$

wobei $\alpha > 0$ und $\beta > 0$ zusätzliche Parameter sind, welche beeinflussen wie schnell ein etwaiger numerischer Integrationsfehler in der Zwangsbedingung wieder zu null konvergiert. Die Fehlerdynamik in (5.60) lässt sich realisieren, indem (5.57) durch

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} = -\dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} - \alpha\dot{\boldsymbol{\phi}}(\mathbf{q}) - \beta\boldsymbol{\phi}(\mathbf{q}) \quad (5.61)$$

ersetzt wird.

5.6.2 Reduktion der Freiheitsgrade

In einfachen Fällen ist es möglich, die k Bindungen im Modell zu berücksichtigen, indem man die generalisierten Koordinaten $\mathbf{q} \in \mathbb{R}^n$ auf einen Satz von $n - k$ Minimalkoordinaten $\hat{\mathbf{q}} \in \mathbb{R}^{n-k}$ reduziert. Zwischen den Koordinaten \mathbf{q} und $\hat{\mathbf{q}}$ besteht der Zusammenhang

$$\mathbf{q} = \boldsymbol{\psi}(\hat{\mathbf{q}}) \quad (5.62)$$

$$\dot{\mathbf{q}} = \mathbf{J}_\psi(\hat{\mathbf{q}})\dot{\hat{\mathbf{q}}} \quad (5.63)$$

$$\ddot{\mathbf{q}} = \mathbf{J}_\psi(\hat{\mathbf{q}})\ddot{\hat{\mathbf{q}}} + \dot{\mathbf{J}}_\psi(\hat{\mathbf{q}})\dot{\hat{\mathbf{q}}} \quad (5.64)$$

mit der Koordinatentransformation $\boldsymbol{\psi} : \mathbb{R}^{n-k} \rightarrow \mathbb{R}^n$ und der entsprechenden analytischen Jacobimatrix $\mathbf{J}_\psi(\hat{\mathbf{q}}) := \partial\boldsymbol{\psi}(\hat{\mathbf{q}})/\partial\hat{\mathbf{q}}$. Hierbei ist eine wesentliche Anforderung, dass die Transformation zwischen \mathbf{q} und $\hat{\mathbf{q}}$ kompatibel zu den Bindungen (5.54) ist, d.h. es wird die Eigenschaft

$$\mathbf{A}(\boldsymbol{\psi}(\hat{\mathbf{q}}))\mathbf{J}_\psi(\hat{\mathbf{q}}) = \mathbf{0} \quad (5.65)$$

gefordert, so dass (5.54) für jede beliebige Minimalgeschwindigkeit $\dot{\hat{\mathbf{q}}}$ erfüllt wird.

Ersetzt man die neuen Koordinaten $\hat{\mathbf{q}}$ in die Dynamikgleichungen (5.56), so ergibt sich zunächst

$$\mathbf{M}\mathbf{J}_\psi\ddot{\hat{\mathbf{q}}} + \mathbf{M}\dot{\mathbf{J}}_\psi\dot{\hat{\mathbf{q}}} + \mathbf{C}\mathbf{J}_\psi\dot{\hat{\mathbf{q}}} + \mathbf{g} = \boldsymbol{\tau} + \mathbf{A}^T\boldsymbol{\lambda} . \quad (5.66)$$

Um die Zwangskraft $\boldsymbol{\lambda}$ zu eliminieren, wird die Dynamikgleichung nun mit $\mathbf{J}_\psi(\hat{\mathbf{q}})^T$ vormultipliziert

$$\underbrace{\mathbf{J}_\psi^T\mathbf{M}\mathbf{J}_\psi}_{\hat{\mathbf{M}}}\ddot{\hat{\mathbf{q}}} + \underbrace{\mathbf{J}_\psi^T(\mathbf{M}\dot{\mathbf{J}}_\psi + \mathbf{C}\mathbf{J}_\psi)}_{\hat{\mathbf{C}}}\dot{\hat{\mathbf{q}}} + \underbrace{\mathbf{J}_\psi^T\mathbf{g}}_{\hat{\mathbf{g}}} = \underbrace{\mathbf{J}_\psi^T\boldsymbol{\tau}}_{\hat{\boldsymbol{\tau}}} . \quad (5.67)$$

Hierbei sind die Massenmatrix $\hat{\mathbf{M}} \in \mathbb{R}^{(n-k) \times (n-k)}$, die Coriolismatrix $\hat{\mathbf{C}} \in \mathbb{R}^{(n-k) \times (n-k)}$ und der Gravitationsvektor $\hat{\mathbf{g}} \in \mathbb{R}^{(n-k)}$ für $\mathbf{q} = \boldsymbol{\psi}(\hat{\mathbf{q}})$ und $\dot{\mathbf{q}} = \mathbf{J}_\psi(\hat{\mathbf{q}})\dot{\hat{\mathbf{q}}}$ auszuwerten.

Der Hauptvorteil in dieser Vorgehensweise liegt in der Tatsache, dass nur die reduzierte Dynamikgleichung (5.67) aufintegriert werden muss, und es hierbei zu keiner Drift in den Bindungsgleichungen kommen kann. Jedoch ergibt sich kein expliziter Ausdruck für die Kontaktkraft $\boldsymbol{\lambda}$. Um diese zu ermitteln, muss weiterhin (5.59) ausgewertet werden.

Bemerkung 5.3. Man beachte, dass dasselbe Vorgehen auch angewendet werden kann, um für die Dynamikgleichungen eine vollständige Koordinatentransformation zwischen zwei Sätzen von generalisierten Koordinaten $\mathbf{q}_1 \in \mathbb{R}^n$ und $\mathbf{q}_2 \in \mathbb{R}^n$ zu berechnen.

Beispiel 5.1 (Eingeschränkte Bewegung des Endeffektors).

Abbildung

5.8 zeigt einen einfachen planaren Mechanismus, in dem die vertikale Bewegung des Endeffektors durch eine Zwangsbedingung gesperrt ist. Wir nehmen nun an, dass das System zunächst ohne diese Zwangsbedingung als serielle kinematische Kette modelliert wurde. Die Position des Endeffektors ergibt sich in Abhängigkeit der

Koordinaten q_1 und q_2 zu

$$\mathbf{p}_e = \begin{pmatrix} l \sin(q_1) + l \sin(q_1 + q_2) \\ l \cos(q_1) + l \cos(q_1 + q_2) \end{pmatrix}$$

Durch die Einsperrung in vertikaler Richtung erhält man die skalare Zwangsbedingung ($k = 1$)

$$\phi(\mathbf{q}) = l \cos(q_1) + l \cos(q_1 + q_2) = 0 ,$$

und somit die Matrix $\mathbf{A}(\mathbf{q})$ zu

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} -l \sin(q_1) - l \sin(q_1 + q_2) & -l \sin(q_1 + q_2) \end{bmatrix} \in \mathbb{R}^{1 \times 2} .$$

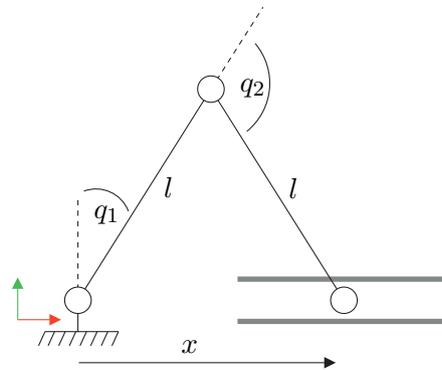


Abbildung 5.8: Planarer Manipulator mit einer Bindung am Endeffektor.

Mithilfe der Matrix $\mathbf{A}(\mathbf{q})$ lässt sich über (5.59) die Zwangskraft λ bestimmen. Wählt man nun den Winkel q_1 als Minimalkoordinate, d.h. $\hat{q} = q_1$, dann wird noch ein Ausdruck von q_2 in Abhängigkeit der Minimalkoordinate q_1 benötigt. Aus $\phi(\mathbf{q}) = 0$ erhält man

$$\begin{aligned} \cos(q_1) &= -\cos(q_1 + q_2) \\ &= \cos(\pi - q_1 - q_2) , \end{aligned}$$

woraus $q_2 = \pi - 2q_1$ folgt^a. Somit erhält man den Zusammenhang (5.62) zu

$$\mathbf{q} = \psi(\hat{q}_1) = \begin{pmatrix} \hat{q}_1 \\ \pi - 2\hat{q}_1 \end{pmatrix}$$

mit der entsprechenden Jacobimatrix

$$\mathbf{J}_\psi(\hat{q}_1) = \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

Die notwendige Eigenschaft (5.65) kann leicht durch Einsetzen verifiziert werden:

$$\begin{aligned} \mathbf{A}(\psi(\hat{\mathbf{q}}))\mathbf{J}_\psi(\hat{\mathbf{q}}) &= -l \begin{bmatrix} \sin(\hat{q}_1) + \sin(\hat{q}_1 + \pi - 2\hat{q}_1) & \sin(\hat{q}_1 + \pi - 2\hat{q}_1) \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \\ &= -l \begin{bmatrix} 2\sin(\hat{q}_1) & \sin(\hat{q}_1) \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \\ &= 0 \end{aligned}$$

Alternativ zur Wahl $\hat{q} = q_1$ hätte man statt der Gelenkkordinate q_1 auch die horizontale Position des Endeffektors nutzen können, d.h.

$$\hat{q} = x = l \sin(q_1) + l \sin(q_1 + q_2).$$

Mit $q_2 = \pi - 2q_1$ (wie vorhin aus $\phi(\mathbf{q}) = 0$) erhält man nun

$$\mathbf{q} = \psi(x) = \begin{pmatrix} \arcsin(x/2) \\ \pi - 2\arcsin(x/2) \end{pmatrix}$$

Durch Anwendung der Koordinatenreduktion (5.67) erhält man dann direkt die Dynamikgleichung in der Koordinate x .

^aDie Auflösung der Gleichung über den arccos ist aufgrund der Periodizität mit 2π nicht eindeutig. Die Wahl $q_2 = \pi - 2q_1$ entspricht der in Abbildung 5.8 gezeigten Konfiguration.

5.7 Literatur

- [5.1] W. Khalil und E. Dombre, *Modeling and Control of Robots*. Hermes Penton Science, 2002.
- [5.2] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer Verlag, 2008.
- [5.3] S. M. Neuman, T. Koolen, J. Drean, J. E. Miller und S. Devadas, „Benchmarking and Workload Analysis of Robot Dynamics Algorithms,“ in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, S. 5235–5242.
- [5.4] J. Carpentier u. a., „The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,“ in *SII 2019 - International Symposium on System Integrations*, Paris, France, Jan. 2019. Adresse: <https://hal.laas.fr/hal-01866228>.

6 Regelung

In diesem Kapitel werden verschiedene grundlegende Regelungsansätze für die in Kapitel 5 hergeleitete Roboterdynamik

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (6.1)$$

beschrieben. Beim Entwurf und der Analyse von Roboterregelungen unterscheidet man zwischen der Stabilisierung eines Arbeitspunktes (engl. *regulation*) und einer Trajektorienfolgeregelung (engl. *tracking*), bei der der Roboter einer gegebenen Solltrajektorie asymptotisch stabil folgen soll. Einleitend werden die wichtigsten benötigten regelungstechnischen Grundlagen in Erinnerung gerufen.

6.1 Regelungstechnische Grundlagen

6.1.1 Stabilität nichtlinearer Systeme

In diesem Abschnitt werden die benötigten Begriffe der Stabilitätstheorie nichtlinearer Systeme kurz eingeführt. Für eine ausführlichere Darstellung wird auf die LVA „Nichtlineare dynamische Systeme und Regelung“ verwiesen. Gegeben sei ein nichtlineares autonomes dynamisches System mit Zustand¹ $\bar{\mathbf{x}} \in \mathbb{R}^N$ in der Form

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}) . \quad (6.2)$$

Im Folgenden wird angenommen, dass der Punkt $\bar{\mathbf{x}}^* = \mathbf{0}$ eine Ruhelage des Systems ist, d.h. $\mathbf{f}(\bar{\mathbf{x}}^*) = \mathbf{0}$.

Satz 6.1 (Stabilität). *Das dynamische System (6.2) nennt man stabil (im Sinne von Lyapunov), wenn zu jedem $\epsilon > 0$ ein $\delta(\epsilon) > 0$ existiert, so dass*

$$\|\bar{\mathbf{x}}(t_0) - \bar{\mathbf{x}}^*\| < \delta \implies \|\bar{\mathbf{x}}(t) - \bar{\mathbf{x}}^*\| < \epsilon \quad \forall t \geq t_0 . \quad (6.3)$$

Gibt es außerdem ein $\eta > 0$, so dass

$$\|\bar{\mathbf{x}}(t_0) - \bar{\mathbf{x}}^*\| < \eta \implies \lim_{t \rightarrow \infty} \bar{\mathbf{x}}(t) = \bar{\mathbf{x}}^* \quad (6.4)$$

gilt, dann nennt man das System asymptotisch stabil (Stabilität + Attraktivität = asymptotische Stabilität).

In vielen Fällen lässt sich die lokale Stabilität des Systems anhand seiner Linearisierung beurteilen:

¹Zur besseren Unterscheidung von den Aufgabenkoordinaten \mathbf{x} in Abschnitt (6.3) wird der Zustand eines allgemeinen Systems hier (abweichend von der in der regelungstechnischen Literatur üblichen Notation) mit $\bar{\mathbf{x}}$ bezeichnet.

Satz 6.2 (Indirekte Methode von Lyapunov). Gegeben sei das dynamische System (6.2) mit einer differenzierbaren Funktion $\mathbf{f}(\bar{\mathbf{x}})$. Haben alle Eigenwerte der Matrix

$$\mathbf{A} = \frac{\partial}{\partial \bar{\mathbf{x}}} \mathbf{f}(\bar{\mathbf{x}})|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^*} \quad (6.5)$$

negativen Realteil, so ist das System lokal asymptotisch stabil. Wenn ein Eigenwert einen Realteil größer null aufweist, so ist das System instabil. Im Falle von Eigenwerten mit Realteil gleich null, lässt sich keine Aussage treffen.

Ein wichtiges Hilfsmittel zur Stabilitätsanalyse nichtlinearer Systeme stellt die Lyapunov-Theorie dar. Diese beruht auf der Analyse von positiv definiten Funktionen entlang der Lösung der Systemdynamik.

Satz 6.3 (Positiv Definitheit einer skalaren Funktion). Eine skalare Funktion $V(\bar{\mathbf{x}})$ nennt man lokal positiv definit, wenn in einer offenen Umgebung $\mathcal{D} \subseteq \mathbb{R}^N$ des Ursprunges

$$\begin{aligned} V(\mathbf{0}) &= 0 \\ V(\bar{\mathbf{x}}) &> 0 \quad \forall \bar{\mathbf{x}} \in \mathcal{D} \setminus \{\mathbf{0}\} \end{aligned} \quad (6.6)$$

gilt. Gilt dies für den gesamten \mathbb{R}^N und gibt es eine Konstante $r > 0$, so dass die Bedingung

$$\inf_{\|\bar{\mathbf{x}}\| > r} V(\bar{\mathbf{x}}) > 0$$

erfüllt ist, dann nennt man die Funktion positiv definit. Gilt in (6.6) nur die schwächere Ungleichung $V(\bar{\mathbf{x}}) \geq 0$, so nennt man die Funktion positiv semi-definit. Eine Funktion $V(\bar{\mathbf{x}})$, für die $-V(\bar{\mathbf{x}})$ positiv (semi-)definit ist, nennt man negativ (semi-)definit.

Satz 6.4 (Direkte Methode von Lyapunov). Gegeben sei das dynamische System (6.2). Die Ruhelage $\bar{\mathbf{x}}^* = \mathbf{0}$ ist stabil im Sinne von Lyapunov, wenn in einer offenen Umgebung $\mathcal{U} \subset \mathbb{R}^N$ von $\bar{\mathbf{x}}^*$ eine stetig differenzierbare, positiv definite Funktion $V(\bar{\mathbf{x}})$ existiert, so dass die Relation

$$\dot{V} = \frac{\partial V(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} \mathbf{f}(\bar{\mathbf{x}}) \leq 0 \quad \forall \bar{\mathbf{x}} \in \mathcal{U} \quad (6.7)$$

gilt. Die Funktion $V(\bar{\mathbf{x}})$ wird Lyapunov-Funktion genannt. Ist die zeitliche Ableitung $\dot{V}(\bar{\mathbf{x}})$ negativ definit, d.h. $\dot{V}(\bar{\mathbf{x}}) < 0$, so ist die Ruhelage asymptotisch stabil. Man spricht in diesem Fall von einer strikten Lyapunov-Funktion. Gelten diese Eigenschaften im gesamten \mathbb{R}^N und ist die Funktion $V(\bar{\mathbf{x}})$ radial unbeschränkt^a, so ist das System global stabil (bzw. global asymptotisch stabil).

^aEine skalare Funktion $V(\bar{\mathbf{x}})$ nennt man radial unbeschränkt, wenn $\lim_{\|\bar{\mathbf{x}}\| \rightarrow \infty} V(\bar{\mathbf{x}}) \rightarrow \infty$ gilt.

Sowohl die Stabilitätsdefinition, wie auch die Lyapunov-Analyse lassen sich auf den nicht-

autonomen Fall erweitern (siehe LVA „Nichtlineare dynamische Systeme und Regelung“). Insbesondere in der Trajektorienfolgeregelung führen einige Regelungsmethoden auf ein zeitvarianten Verhalten des geschlossenen Regelkreises.

6.1.2 Passivität

Während die Stabilität im Sinne von Lyapunov das autonome Verhalten eines Systems ohne Eingang betrifft, bezieht sich die Passivität auf das Ein-/Ausgangsverhalten. Gegeben sei ein dynamisches System mit Eingang $\mathbf{u} \in \mathbb{R}^m$ und Ausgang $\mathbf{y} \in \mathbb{R}^m$

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \mathbf{u}) \quad (6.8)$$

$$\mathbf{y} = \mathbf{h}(\bar{\mathbf{x}}) . \quad (6.9)$$

Satz 6.5 (Passivität). *Das System (6.8)-(6.9) wird als passiv bezüglich des Einganges \mathbf{u} und Ausganges \mathbf{y} bezeichnet, sofern eine nach unten beschränkte Funktion $S(\bar{\mathbf{x}})$, genannt Speicherfunktion, existiert, so dass die Ungleichung*

$$\dot{S}(\bar{\mathbf{x}}) \leq \mathbf{y}^T \mathbf{u} \quad \forall \bar{\mathbf{x}} \in \mathbb{R}^N, \mathbf{u} \in \mathbb{R}^m \quad (6.10)$$

erfüllt ist.

Passivität bedeutet, dass das System von sich aus keine positive „Leistung“ \dot{S} erzeugt. In der Robotik wird als Speicherfunktion oft die Gesamtenergie des Systems angesetzt. Passivitätseigenschaften können insbesondere in einer Systemdarstellung mit mehreren Subsystemen genutzt werden. Die Rückkopplung und die Parallelschaltung zweier passiver Teilsysteme ergibt wiederum ein passives Gesamtsystem. Dies gilt jedoch nicht für die Serienschaltung! Für weiterführende Eigenschaften wie die strikte Eingangs- (Ausgangs-) Passivität sei an dieser Stelle auf die VO „Fortgeschrittene Methoden der nichtlinearen Regelung“ verwiesen.

6.1.3 Zustandsraumdarstellung der Roboterdynamik

Die Roboterdynamik in (6.1) ist eine Dynamik zweiter Ordnung, in dem Sinne, dass sie die zweite zeitliche Ableitung der Gelenkkonfiguration $\mathbf{q} \in \mathbb{R}^n$ beinhaltet. In der Robotik ist es üblich, direkt mit Gleichung (6.1) zu arbeiten. Es sollte jedoch darauf hingewiesen werden, dass der Systemzustand $\bar{\mathbf{x}} \in \mathbb{R}^{2n}$ durch $\bar{\mathbf{x}} = (\mathbf{q}^T, \dot{\mathbf{q}}^T)^T$ gegeben ist. Gleichung (6.1) kann mit $\mathbf{u} \hat{=} \boldsymbol{\tau}$ leicht in die generische Form (6.8) gebracht werden:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})) \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\tau} \end{pmatrix}$$

Beispiel 6.1 (Passivität der Roboterdynamik). Für einen Roboterarm mit ausschließlich rotatorischen Gelenken ist die potentielle Energie $\mathcal{V}_g(\mathbf{q})$ beschränkt. In diesem Fall besteht $\mathcal{V}_g(\mathbf{q})$ lediglich aus trigonometrischen Termen in den Gelenkwinkeln q_i $i = 1, \dots, n$. Somit ist die Gesamtenergie $\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{V}(\mathbf{q})$ des Roboters von unten beschränkt und kann als zulässige Speicherfunktion benutzt werden. Die Passivität der Roboterdynamik folgt mit Satz 5.3 aus

$$\begin{aligned}
 \frac{d}{dt} \mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{d}{dt} \left(\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} \right) + \dot{\mathcal{V}}_g(\mathbf{q}) \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{\partial \mathcal{V}_g(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})) + \mathbf{g}(\mathbf{q})^T \dot{\mathbf{q}} \\
 &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}) \\
 &= \dot{\mathbf{q}}^T \left(\frac{1}{2} \dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \boldsymbol{\tau} \\
 &= \dot{\mathbf{q}}^T \boldsymbol{\tau}
 \end{aligned}$$

6.2 Regelung im Gelenkraum

6.2.1 Stabilisierung eines Arbeitspunktes

Regler mit PD- und PID-Struktur können zur Stabilisierung eines Arbeitspunktes verwendet werden, d.h. es wird eine konstante Sollkonfiguration \mathbf{q}_d vorgegeben. Hierbei handelt es sich um sogenannte dezentrale Regler. Diese Regler können für jedes Gelenk getrennt implementiert werden.

Dezentrale Regler mit PD- und PID-Struktur

Ein PD-Regler hat die folgende Form

$$\boldsymbol{\tau} = - \underbrace{\mathbf{K}_d \dot{\mathbf{e}}_q}_{\text{D-Anteil}} - \underbrace{\mathbf{K}_p \mathbf{e}_q}_{\text{P-Anteil}} \quad (6.11)$$

mit dem Regelfehler in den Gelenkpositionen $\mathbf{e}_q = \mathbf{q} - \mathbf{q}_d$ und -geschwindigkeiten $\dot{\mathbf{e}}_q = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d = \dot{\mathbf{q}}$, sowie den positiv definiten und symmetrischen Reglermatrizen $\mathbf{K}_d \in \mathbb{R}^{n \times n}$ und $\mathbf{K}_p \in \mathbb{R}^{n \times n}$. Für eine dezentrale Regelung müssen \mathbf{K}_p und \mathbf{K}_d weiters als Diagonalmatrizen gewählt werden. Die Reglermatrizen haben eine nützliche physikalische Interpretation als virtuelle Steifigkeit und Dämpfung. Der Proportionalanteil lässt sich auch als Differential eines Steifigkeitspotentials $V_k(\mathbf{e}_q)$ wie folgt anschreiben:

$$V_k(\mathbf{e}_q) = \frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q \quad (6.12)$$

$$\boldsymbol{\tau}_p = - \left(\frac{\partial}{\partial \mathbf{q}} V_k(\mathbf{e}_q) \right)^T = -\mathbf{K}_p \mathbf{e}_q \quad (6.13)$$

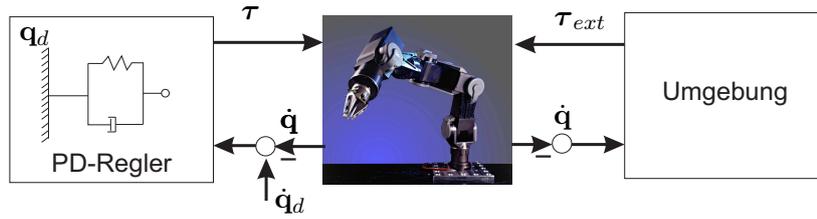


Abbildung 6.1: Interpretation der PD-Regelung als Zusammenschaltung passiver Teilsysteme.

Durch Einsetzen von (6.11) in (6.1) erhält man die folgende Systemdynamik des geschlossenen Regelkreises

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = -\mathbf{K}_d\dot{\mathbf{e}}_q - \mathbf{K}_p\mathbf{e}_q . \quad (6.14)$$

Die Gleichung für die Ruhelage des Systems ergibt sich aus der Bedingung $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0}$ zu

$$\mathbf{g}(\mathbf{q}) = -\mathbf{K}_p(\mathbf{q} - \mathbf{q}_d) . \quad (6.15)$$

Wie man sieht, ist der gewünschte Punkt $\mathbf{q} = \mathbf{q}_d$ für den PD-Regler keine Ruhelage des Systems. Es lässt sich aber zeigen, dass sich für hinreichend große Werte der diagonalen Reglermatrix \mathbf{K}_p aus (6.15) eine eindeutige Ruhelage ($\mathbf{q}^*, \dot{\mathbf{q}}^* = \mathbf{0}$) in der Nähe von \mathbf{q}_d ergibt. Eine hinreichende Bedingung hierfür ist, dass die Eigenwerte der Verstärkungsmatrix \mathbf{K}_p größer sind als die Eigenwerte der Hessematrix des Gravitationspotentials. Unter der Bedingung

$$\min(\text{eig}(\mathbf{K}_p)) > \max\left(\text{eig}\left(\frac{\partial^2 \mathcal{V}_g(\mathbf{q})}{\partial \mathbf{q}^2}\right)\right) \quad \forall \mathbf{q} \in \mathbb{R}^n \quad (6.16)$$

besitzt die Gleichung (6.15) eine eindeutige Lösung \mathbf{q}^* . In diesem Fall lässt sich für den PD-geregelten Roboter eine Lyapunov-Funktion angeben, bestehend aus der kinetischen und potentiellen Energie des Roboters und der Potentialfunktion der virtuellen Steifigkeit. Eine detaillierte Stabilitätsanalyse für diesen Regler ist im Anhang I (Abschnitt 6.6.1) gegeben.

Es erscheint auf den ersten Blick erstaunlich, dass die komplexe nichtlineare Roboterdynamik (6.1) durch einen einfachen linearen PD-Regler in jedem Gelenk stabilisiert werden kann. Der Grund hierfür liegt in den bereits erwähnten Passivitätseigenschaften der Roboterdynamik. Der geschlossene Regelkreis kann als Zusammenschaltung zweier passiver Subsysteme² betrachtet werden. In der Tat lässt sich diese passivitätsbasierte Betrachtung erweitern, wenn man auch externe, auf den Roboter einwirkende, Drehmomente τ_{ext} betrachtet (siehe Abbildung 6.1).

Im Prinzip lässt sich der resultierende stationäre Regelfehler $\mathbf{q}^* - \mathbf{q}_d$ durch Wahl einer hinreichend großen Reglerverstärkung reduzieren. In der Praxis ist dieses Vorgehen allerdings durch Rauschen in den Messsignalen und die Auswirkung von unmodellierter Dynamik eingeschränkt. Ein anderer Ansatz, um die bleibende Regelabweichung zu reduzieren, ist die Verwendung eines Integralgliedes im Regler

²Man beachte, dass die Passivität der Roboterdynamik streng genommen die Beschränktheit des Gravitationspotentials erfordert. Dies ist jedenfalls für Roboter mit nur Drehgelenken sichergestellt.

$$\boldsymbol{\tau} = - \underbrace{\mathbf{K}_d \dot{\mathbf{e}}_q}_{\text{D-Anteil}} - \underbrace{\mathbf{K}_p \mathbf{e}_q}_{\text{P-Anteil}} - \underbrace{\int \mathbf{K}_i \mathbf{e}_q dt}_{\text{I-Anteil}} . \quad (6.17)$$

Für den Roboter mit PID-Regelung kann eine lokale Stabilitätsanalyse durch Berechnung der Linearisierung entsprechend Satz 6.2 erfolgen. Man beachte jedoch, dass die Systemdynamik durch die Konfigurationsabhängigkeit von $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ und $\mathbf{g}(\mathbf{q})$ über den Arbeitsraum üblicherweise stark variiert. Beispielsweise schwanken die Gravitationsterme in der Praxis durchaus zwischen null und bis zu 30% der maximalen Gelenkdrehmomente. Somit lässt sich mit konstanten Reglerparametern nur bedingt ein gutes Systemverhalten im gesamten Arbeitsraum erzielen.

Eine weitere Einschränkung in der Nutzung von Integralgliedern liegt in dem Einfluss der Haftreibung, welche in Kombination mit Integralanteilen zu Grenzyklen (Oszillationen) führen. In der Praxis kann man diesem Problem begegnen, indem der Integralanteil eine Totzone enthält, d.h. der Integrator wird in der unmittelbaren Nähe des stationären Arbeitspunktes deaktiviert. Auch bei zu großen Regelabweichungen wird der Integralanteil üblicherweise deaktiviert, um ein zu großes Überschwingen zu vermeiden.

PD-Regler mit Gravitationskompensation

Der stationäre Regelfehler der PD-Regelung resultiert aus den wirkenden Gravitationstermen $\mathbf{g}(\mathbf{q})$. Diese können auch direkt im Regler berücksichtigt werden

$$\boldsymbol{\tau} = \underbrace{\mathbf{g}(\mathbf{q})}_{\text{Gravitationskomp.}} - \underbrace{\mathbf{K}_d \dot{\mathbf{e}}_q}_{\text{D-Anteil}} - \underbrace{\mathbf{K}_p \mathbf{e}_q}_{\text{P-Anteil}} \quad (6.18)$$

Aufgrund des Terms $\mathbf{g}(\mathbf{q})$ in (6.18) muss das Regelgesetz von einer zentralen Recheneinheit ausgewertet werden, in welcher die Messwerte aller Gelenkpositionen \mathbf{q} vorliegen. Damit zählt dieses Verfahren zu den zentralen Reglern im Konfigurationsraum.

Setzt man (6.18) in (6.1) ein, so erhält man für den geschlossenen Regelkreis das folgende Systemverhalten

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = -\mathbf{K}_d \dot{\mathbf{q}} - \mathbf{K}_p \mathbf{e}_q . \quad (6.19)$$

Aus (6.19) ist unmittelbar einsichtig, dass die Ruhelage nun mit dem gewünschten Sollwert \mathbf{q}_d übereinstimmt, d.h. $\mathbf{q}^* = \mathbf{q}_d$ und $\dot{\mathbf{q}}^* = \mathbf{0}$. Zur Stabilitätsanalyse kann die Lyapunov-Funktion

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \underbrace{\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}}_{\text{kinetische Energie}} + \underbrace{\frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q}_{\text{potentielle Energie der Feder}} \quad (6.20)$$

mit der potentiellen Energie der Feder V_k aus (6.12) angesetzt werden. Bildet man die

zeitliche Ableitung von (6.20), so folgt

$$\begin{aligned}
 \dot{V}(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} \ddot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{e}}_q^T \mathbf{K}_p \mathbf{e}_q + \frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{e}}_q \\
 &= \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{e}}_q^T \mathbf{K}_p \mathbf{e}_q \\
 &= \dot{\mathbf{q}}^T \underbrace{(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{K}_d \dot{\mathbf{q}} - \mathbf{K}_p \mathbf{e}_q)}_{\text{gemäß (6.19)}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{K}_p \mathbf{e}_q \quad (6.21) \\
 &= -\dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} + \underbrace{\frac{1}{2} \dot{\mathbf{q}}^T (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{q}}}_{=0} \leq 0.
 \end{aligned}$$

Man erkennt, dass wegen der positiven Definitheit von \mathbf{K}_d die Lyapunov-Funktion negativ semi-definit ist, woraus die Stabilität des geschlossenen Regelkreises folgt. Auch in diesem Fall kann über das Invarianzprinzip von Krasovskii-LaSalle auch asymptotische Stabilität gezeigt werden. In der Literatur findet man für diesen Fall auch strikte Lyapunov-Funktionen.

Bemerkung 6.1. Man kann die Stabilität auch zeigen, wenn statt $\mathbf{g}(\mathbf{q})$ nur der gewünschte Arbeitspunkt bei der Gravitationskompensation $\mathbf{g}(\mathbf{q}_d)$ in (6.18) eingesetzt wird.

6.2.2 Trajektorienfolgeregelung

In der Trajektorienfolgeregelung sind eine zweifach stetig differenzierbare Solltrajektorie $\mathbf{q}_d(t)$ und deren zeitliche Ableitungen $\dot{\mathbf{q}}_d(t)$ und $\ddot{\mathbf{q}}_d(t)$ gegeben. Durch die Regelung soll eine Konvergenz von $\mathbf{q}(t)$ zu $\mathbf{q}_d(t)$ erreicht werden. Hierbei wird angenommen, dass die Solltrajektorie realisierbar ist. Dies bedeutet, dass die Gelenkgeschwindigkeit $\dot{\mathbf{q}}_d$ und das erforderliche Gelenkmoment

$$\boldsymbol{\tau}_d = \mathbf{M}(\mathbf{q}_d) \ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) \quad (6.22)$$

innerhalb der zulässigen Minimal- und Maximalwerte bleiben, d.h. $\dot{q}_{i,min} < \dot{q}_{d,i} < \dot{q}_{i,max}$ und $\tau_{i,min} < \tau_{d,i} < \tau_{i,max}$, $i = 1, \dots, n$.

Eine naheliegende Art, die PD-Regelung auf einen Trajektorienfolgeregler zu erweitern, ist es, den PD-Regler um eine Vorsteuerung entlang der Solltrajektorie zu ergänzen:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}_d) \ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) - \mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q \quad (6.23)$$

Für diesen Regler lassen sich im Allgemeinen nur lokale Stabilitätsaussagen treffen. Im Folgenden werden stattdessen zwei nichtlineare Regler vorgestellt, die globale asymptotische Stabilität sicherstellen.

6.2.3 Computed-Torque-Regler

In dem Computed-Torque-Regler werden die nichtlinearen Terme der Roboterdynamik (6.1) kompensiert, mit dem Ziel, ein lineares Systemverhalten zu erzeugen. Der Computed-Torque-Regler lautet

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d - \mathbf{K}_2\dot{\mathbf{e}}_q - \mathbf{K}_1\mathbf{e}_q) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) , \quad (6.24)$$

mit dem Trajektorienfehler $\mathbf{e}_q = \mathbf{q} - \mathbf{q}_d(t)$ bzw. $\dot{\mathbf{e}}_q = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d(t)$ und den positiv definiten Reglermatrizen $\mathbf{K}_1 \in \mathbb{R}^{n \times n}$ und $\mathbf{K}_2 \in \mathbb{R}^{n \times n}$. Ein Vergleich zwischen (6.24) und (6.1) zeigt die strukturelle Ähnlichkeit des Computed-Torque-Reglers zum Robotermodell im Konfigurationsraum. Deshalb wird dieses Verfahren auch als *Invers-Dynamik-Regelung* bezeichnet. Weiters handelt es sich beim Computed-Torque-Regler ebenfalls um ein zentrales Regelungskonzept im Konfigurationsraum.

Setzt man (6.24) in (6.1) ein, so ergibt sich

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d - \mathbf{K}_2\dot{\mathbf{e}}_q - \mathbf{K}_1\mathbf{e}_q) \quad (6.25)$$

$$\ddot{\mathbf{e}}_q + \mathbf{K}_2\dot{\mathbf{e}}_q + \mathbf{K}_1\mathbf{e}_q = \mathbf{0} . \quad (6.26)$$

Wählt man für \mathbf{K}_1 und \mathbf{K}_2 Diagonalmatrizen in der Form $\mathbf{K}_1 = \text{diag}(k_{1,1}, \dots, k_{1,n})$ und $\mathbf{K}_2 = \text{diag}(k_{2,1}, \dots, k_{2,n})$ mit positiven Einträgen $k_{1,j}, k_{2,j} > 0$, $j = 1, \dots, n$, dann sind die Fehlerdynamiken der einzelnen Gelenkkordinaten q_j mit

$$\ddot{e}_{q,j} + k_{2,j}\dot{e}_{q,j} + k_{1,j}e_{q,j} = 0 , \quad j = 1, \dots, n \quad (6.27)$$

entkoppelt und asymptotisch stabil.

Der Computed-Torque-Regler resultiert somit in einer exakt entkoppelten linearen Fehlerdynamik. Um die nichtlinearen Terme exakt zu kompensieren, müssen allerdings alle statischen und dynamischen Parameter (Abmessungen, Schwerpunkte, Massen, Massenträgheitsmomente) des Roboters sehr genau bekannt sein, was in der Praxis eine Herausforderung darstellt. Außerdem wird über den gesamten Arbeitsraum das gleiche lineare Fehlerverhalten gefordert. Da die Werte der Massenmatrix über den Konfigurationsraum stark variieren können, sind zum Teil sehr große Stellgrößen erforderlich, um im gesamten Arbeitsraum das gleiche Systemverhalten zu erzielen. Im folgenden Abschnitt wird daher ein alternatives Regelgesetz für das Trackingproblem beschrieben, das eine exakte Linearisierung vermeidet.

6.2.4 PD+ Regler

In dem *PD+ Regler* wird die exakte Entkopplung der Fehlerdynamik des Computed-Torque-Reglers vermieden. Stattdessen werden nichtlineare Vorsteuerterme gewählt, welche eine globale nichtlineare Stabilitätsanalyse erlauben. Der PD+-Regler hat die folgende Form

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) - \mathbf{K}_d\dot{\mathbf{e}}_q - \mathbf{K}_p\mathbf{e}_q , \quad (6.28)$$

mit den positiv definiten Reglermatrizen $\mathbf{K}_p \in \mathbb{R}^{n \times n}$ und $\mathbf{K}_d \in \mathbb{R}^{n \times n}$. Der PD+-Regler führt auf den geschlossenen Regelkreis

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{e}}_q + (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}_d)\dot{\mathbf{e}}_q + \mathbf{K}_p\mathbf{e}_q = \mathbf{0} . \quad (6.29)$$

Man beachte, dass der geschlossene Regelkreis nun zeitvariant ist. Um den geschlossenen Kreis formal in dem Zustand $(\mathbf{e}_q, \dot{\mathbf{e}}_q)$ anzuschreiben, ist die Konfigurationsabhängigkeit in

$\mathbf{M}(\mathbf{q})$ und $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ durch $\mathbf{q} = \mathbf{e}_q + \mathbf{q}_d(t)$ und $\dot{\mathbf{q}} = \dot{\mathbf{e}}_q + \dot{\mathbf{q}}_d(t)$ zu ersetzen. Globale Stabilität kann mittels der Lyapunov-Funktion

$$V(\mathbf{e}_q, \dot{\mathbf{e}}_q, t) = \frac{1}{2} \dot{\mathbf{e}}_q^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{e}}_q + \frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q \quad (6.30)$$

gezeigt werden. Die zeitliche Ableitung ergibt sich zu

$$\begin{aligned} \dot{V}(\mathbf{e}_q, \dot{\mathbf{e}}_q, t) &= \frac{1}{2} \dot{\mathbf{e}}_q^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{e}}_q + \dot{\mathbf{e}}_q^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{e}}_q + \mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{e}}_q \\ &= \frac{1}{2} \dot{\mathbf{e}}_q^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{e}}_q - \dot{\mathbf{e}}_q^T ((\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}_d) \dot{\mathbf{e}}_q + \mathbf{K}_p \mathbf{e}_q) + \mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{e}}_q \\ &= \frac{1}{2} \dot{\mathbf{e}}_q^T (\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) \dot{\mathbf{e}}_q - \dot{\mathbf{e}}_q^T \mathbf{K}_d \dot{\mathbf{e}}_q \\ &= -\dot{\mathbf{e}}_q^T \mathbf{K}_d \dot{\mathbf{e}}_q \end{aligned}$$

woraus die globale Stabilität folgt. Für diesen Regler kann weiters auch eine strikte Lyapunov-Funktion aufgestellt werden, um auch globale asymptotische Stabilität nachzuweisen.

Im Gegensatz zum Computed-Torque-Regler ist die Dynamik des geschlossenen Regelkreises (6.29) weder linear noch entkoppelt. Man beachte aber, dass die globale Stabilität auch für eine zeitlich variable oder konfigurationsabhängige Reglermatrix \mathbf{K}_d gilt. Zur Erzielung guter Dämpfungseigenschaften im gesamten Arbeitsraum kann \mathbf{K}_d daher an die Werte der Massenmatrix angepasst werden. Weiters können (im Gegensatz zum Computed-Torque-Regler) die Reglermatrizen \mathbf{K}_d und \mathbf{K}_p wie in der PD-Regelung als virtuelle Steifigkeit und Dämpfung interpretiert werden.

6.2.5 Zusammenfassung der Gelenkregler

Die folgende Tabelle fasst alle bisher kennengelernten Regler im Konfigurationsraum zusammen. Wie man sieht, benötigen alle Regler für die Trajektorienfolge sämtliche Komponenten der Dynamikgleichungen (Massenmatrix, Zentrifugal- und Coriolisterme, Gravitationsterme). Im Falle des PD+ Reglers wird auch die Coriolis-Matrix benötigt.

Bezeichnung	Regelgesetz	Stabilisierung
PD	$\boldsymbol{\tau} = -\mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q$	Ruhelage
PID	$\boldsymbol{\tau} = -\mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q - \int \mathbf{K}_i \mathbf{e}_q dt$	Ruhelage
PDg	$\boldsymbol{\tau} = -\mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q + \mathbf{g}(\mathbf{q})$	Ruhelage
PDff	$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}_d) \ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) - \mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q$	Trajektorie
Comp. Torque	$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d - \mathbf{K}_2 \dot{\mathbf{e}}_q - \mathbf{K}_1 \mathbf{e}_q) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$	Trajektorie
PD+	$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q$	Trajektorie

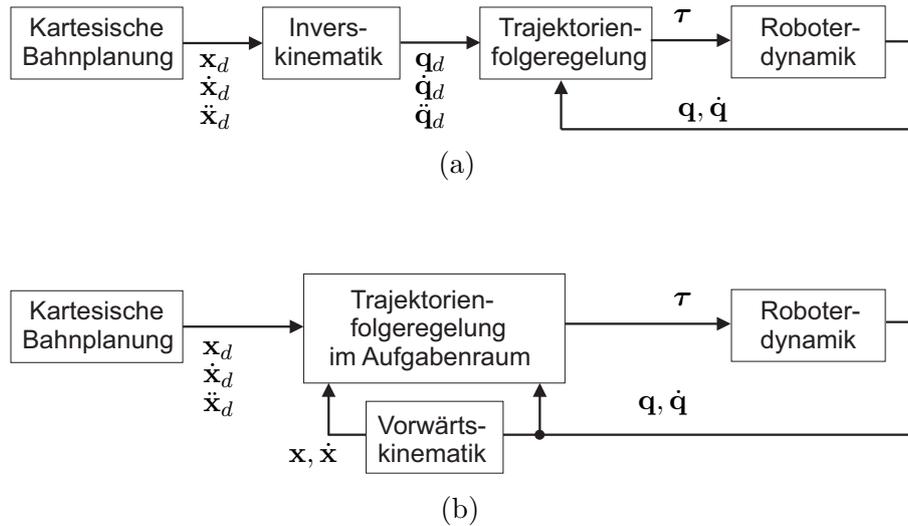


Abbildung 6.2: Zwei Regelungsarchitekturen zur Realisierung einer Trajektorienfolge- regelung in Aufgabenkoordinaten. (a) Über eine Inverskinematik wird die gewünschte Trajektorie aus dem Aufgabenraum in den Konfigurationsraum übertragen und mit einem Trajektorienfolge- regler im Gelenkraum realisiert. (b) Bei einem im Aufgabenraum formulierten Regler ist keine Inverskinematik erforderlich, stattdessen wird der Regler direkt in Aufga- benkoordinaten formuliert.

6.3 Regelung im Aufgabenraum

In vielen Anwendungen ist die gewünschte Trajektorie nicht im Konfigurationsraum, sondern im Aufgabenraum beschrieben. Man unterscheidet zwei Regelungsarchitekturen, wie in Abbildung 6.2 gezeigt. Einerseits kann die Trajektorie zunächst aus dem Aufgabenraum über eine Inverskinematik in den Konfigurationsraum übertragen werden. In diesem Fall können weiterhin die Regler aus dem vorherigen Abschnitt benutzt werden. Alternativ dazu, kann der Regler auch direkt in Aufgabenkoordinaten entworfen werden. Dieses Vorgehen wird im Folgenden näher beschrieben.

Es wird angenommen, dass ein Satz von $m \leq n$ Aufgabenkoordinaten x_i durch die Anwendung vorgegeben ist. Diese werden in dem Vektor $\mathbf{x} \in \mathbb{R}^m$ zusammengefasst. Hierbei handelt es sich oft um die Position und Orientierung des Endeffektors, z.B. $\mathbf{x} = (\mathbf{p}_e^T, \boldsymbol{\phi}_e^T)^T$.

6.3.1 Nichtredundante Kinematik

In diesem Abschnitt wird das dynamische Modell im Aufgabenraum für den Fall eines nichtredundanten Roboters, d.h. für $m = n$, hergeleitet.

Der Zusammenhang zwischen den Koordinaten im Konfigurationsraum und den Koordinaten im Aufgabenraum sowie deren Ableitungen sind durch

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \quad (6.31a)$$

$$\dot{\mathbf{x}} = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} \quad (6.31b)$$

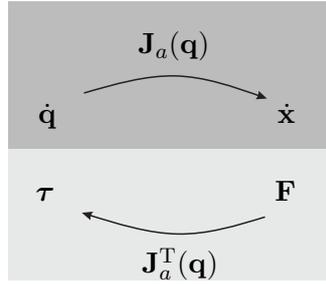


Abbildung 6.3: Transformation der Geschwindigkeiten und Kräfte zwischen Konfigurations- und Aufgabenraum. Die Kräfte werden „dual“ zu den Geschwindigkeiten transformiert, so dass die Leistung in beiden Koordinatensystemen gleich ist, d.h. $P = \boldsymbol{\tau}^T \dot{\mathbf{q}} = \mathbf{F}^T \dot{\mathbf{x}}$.

$$\ddot{\mathbf{x}} = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}} \quad (6.31c)$$

gegeben, wobei $\mathbf{J}_a(\mathbf{q})$ die analytische Jacobimatrix der Vektorfunktion $\mathbf{f}(\mathbf{q})$ ist. Aus (6.31c) folgt unter der Annahme der Regularität von $\mathbf{J}_a(\mathbf{q})$, d. h. außerhalb der Singularitäten, die Beziehung

$$\ddot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})\left(\ddot{\mathbf{x}} - \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}\right). \quad (6.32)$$

Setzt man (6.32) in (6.1) ein, so erhält man

$$\mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\ddot{\mathbf{x}} + \left(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\dot{\mathbf{J}}_a(\mathbf{q})\right)\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (6.33)$$

und durch Multiplikation der Gleichung mit $\mathbf{J}_a^{-T}(\mathbf{q})$ von links ergibt sich das *dynamische Modell des Roboters im Aufgabenraum* zu

$$\boldsymbol{\Lambda}(\mathbf{q})\ddot{\mathbf{x}} + \boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{F}_g(\mathbf{q}) = \mathbf{F} \quad (6.34)$$

mit

$$\boldsymbol{\Lambda}(\mathbf{q}) = \mathbf{J}_a^{-T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q}) \quad (6.35a)$$

$$\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_a^{-T}(\mathbf{q})\left(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\dot{\mathbf{J}}_a(\mathbf{q})\right)\mathbf{J}_a^{-1}(\mathbf{q}) \quad (6.35b)$$

$$\mathbf{F}_g(\mathbf{q}) = \mathbf{J}_a^{-T}(\mathbf{q})\mathbf{g}(\mathbf{q}) \quad (6.35c)$$

und der generalisierten Kraft \mathbf{F} im Arbeitsraum

$$\mathbf{F} = \mathbf{J}_a^{-T}(\mathbf{q})\boldsymbol{\tau}. \quad (6.36)$$

Man beachte, dass durch die Transformation von Geschwindigkeiten $\dot{\mathbf{x}} = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}$ und Kräften $\mathbf{F} = \mathbf{J}_a^{-T}(\mathbf{q})\boldsymbol{\tau}$ sichergestellt ist, dass die Leistung in beiden Darstellungen erhalten bleibt, d.h. $P = \boldsymbol{\tau}^T \dot{\mathbf{q}} = \mathbf{F}^T \dot{\mathbf{x}}$, siehe Abbildung 6.3.

Die Multiplikation von (6.33) mit $\mathbf{J}_a^{-T}(\mathbf{q})$ von links wurde ausgeführt, um beim Robotermodell im Arbeitsraum (6.34) eine ähnliche Struktur zu erhalten wie im Konfigurationsraum (6.1). In (6.34) ist $\boldsymbol{\Lambda}(\mathbf{q})$ die *kartesische Massenmatrix*, $\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})$ ist die *kartesische Coriolis-Matrix* und $\mathbf{F}_g(\mathbf{q})$ ist der Vektor der Gravitationskräfte im Arbeitsraum. Es kann gezeigt werden, dass folgende Eigenschaften auch für das Robotermodell (6.34) erhalten bleiben:

Satz 6.6. Die kartesische Massenmatrix $\Lambda(\mathbf{q}) \in \mathbb{R}^{m \times m}$ ist symmetrisch und positiv definit

$$\Lambda(\mathbf{q}) = \Lambda(\mathbf{q})^T > 0 \quad \forall \mathbf{q} \in \mathbb{R}^n .$$

Satz 6.7. Wird die Matrix der Zentrifugal- und Coriolisterme $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ in (6.35b) entsprechend (5.34) über die Christoffelsymbole generiert, so gilt die Schiefsymmetrie der Matrix $(\dot{\Lambda}(\mathbf{q}) - 2\Gamma(\mathbf{q}, \dot{\mathbf{q}}))$

$$\mathbf{v}^T (\dot{\Lambda}(\mathbf{q}) - 2\Gamma(\mathbf{q}, \dot{\mathbf{q}})) \mathbf{v} = 0 \quad \forall \mathbf{v} \in \mathbb{R}^m \quad (6.37)$$

Die Schiefsymmetrie der Matrix $\dot{\Lambda}(\mathbf{q}) - 2\Gamma(\mathbf{q}, \dot{\mathbf{q}})$ ist äquivalent zu

$$\dot{\Lambda}(\mathbf{q}) = \Gamma(\mathbf{q}, \dot{\mathbf{q}}) + \Gamma(\mathbf{q}, \dot{\mathbf{q}})^T .$$

Somit lassen sich grundsätzlich alle in Abschnitt 6.2 vorgestellten Regler auch auf das Robotermodell in Aufgabenkoordinaten anwenden. Die aus dem Regler resultierende Stellgröße \mathbf{F} wird über

$$\boldsymbol{\tau} = \mathbf{J}_a^T(\mathbf{q}) \mathbf{F} \quad (6.38)$$

aus dem Aufgabenraum in den Konfigurationsraum übertragen.

Beispiel 6.2 (PD-Regler mit Gravitationskompensation). Als Beispiel für einen Regler zur Stabilisierung eines Arbeitspunktes wird der PD-Regler mit Gravitationskompensation in Aufgabenkoordinaten betrachtet. Dieser lautet

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_g(\mathbf{q}) - \mathbf{K}_p(\mathbf{x} - \mathbf{x}_d) - \mathbf{K}_d \dot{\mathbf{x}} \\ \boldsymbol{\tau} &= \mathbf{J}_a^T(\mathbf{q}) \mathbf{F} \\ &= \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q}) (\mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) - \mathbf{K}_d \dot{\mathbf{x}}) \end{aligned}$$

mit positiv definiten Reglermatrizen $\mathbf{K}_p \in \mathbb{R}^{m \times m}$ und $\mathbf{K}_d \in \mathbb{R}^{m \times m}$. Man beachte, dass die Kompensation der Gravitationsterme sinnvollerweise direkt in Konfigurationsraum erfolgt.

Beispiel 6.3 (Computed-Torque-Regler). Der Computed-Torque-Regler für die Trajektorienfolgeregelung im Aufgabenraum lautet

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_g(\mathbf{q}) + \Gamma(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{x}} + \Lambda(\mathbf{q}) (\ddot{\mathbf{x}}_d - \mathbf{K}_2(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - \mathbf{K}_1(\mathbf{x} - \mathbf{x}_d)) \\ \boldsymbol{\tau} &= \mathbf{J}_a^T(\mathbf{q}) \mathbf{F} \\ &= \mathbf{g}(\mathbf{q}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{J}_a^T(\mathbf{q}) \Lambda(\mathbf{q}) (\ddot{\mathbf{x}}_d - \mathbf{K}_2(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - \mathbf{K}_1(\mathbf{x} - \mathbf{x}_d) - \dot{\mathbf{J}}_a(\mathbf{q}) \dot{\mathbf{q}}) \end{aligned}$$

Die Kompensation der Gravitations- sowie Zentrifugal- und Coriolisterme erfolgt auch hierbei besser im Konfigurationsraum. Weiters kann die Matrix $\mathbf{J}_a^T(\mathbf{q}) \Lambda(\mathbf{q})$ im nichtredundanten Fall zu $\mathbf{M}(\mathbf{q}) \mathbf{J}_a^{-1}(\mathbf{q})$ vereinfacht werden.

Man beachte, dass die Aufgabenkoordinaten $\mathbf{x} \in \mathbb{R}^m$ im nichtredundanten Fall einen

neuen Satz generalisierter Koordinaten darstellen und (6.34) im Wesentlichen einer Koordinatentransformation entspricht. Streng formal betrachtet, müsste man in (6.34) alle Abhängigkeiten von \mathbf{q} und $\dot{\mathbf{q}}$ durch $\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$ und $\dot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{f}^{-1}(\mathbf{x}))\dot{\mathbf{x}}$ ersetzen. Dies ist in der Robotikliteratur aber sehr unüblich, da die Messwerte des Roboters ja üblicherweise im Konfigurationsraum vorliegen und alle Dynamikkomponenten sinnvollerweise direkt in diesen Koordinaten berechnet werden.

6.3.2 Redundante Kinematik

Gleichung (6.34) stellt eine vollständige Beschreibung der Roboterdynamik in Aufgabenkoordinaten für den nichtredundanten Fall $m = n$ dar. Im Gegensatz dazu ist die Jacobi-Matrix im redundanten Fall $m < n$ nicht invertierbar und somit lässt sich die Herleitung nicht direkt auf den redundanten Fall anwenden. Man könnte verleitet sein, das Problem zu lösen, indem man in Analogie zur Inverskinematik die Inverse durch eine Pseudoinverse ersetzt. Hierbei ist jedoch Vorsicht angebracht, da die Wahl der Pseudoinversen im Bereich der Dynamik nicht beliebig ist. Man beachte, dass die in (4.91) definierte Pseudoinverse eine spezielle Wahl³ ist, die aus der Minimierung der Euklidischen Norm der Gelenkgeschwindigkeiten folgte. Wir werden sehen, dass im Bereich der Dynamik die Verwendung einer gewichteten Pseudoinversen⁴, mit der Massenmatrix als Gewichtungsmatrix, vorteilhafter ist.

Im Folgenden wird die Roboterdynamik in Aufgabenkoordinaten für den redundanten Fall hergeleitet. Im Gegensatz zum nichtredundanten Fall führt dies jedoch auf keine komplette Beschreibung der Systemdynamik, da die Dynamik der Nullraumbewegung gesondert behandelt werden muss.

Da $\mathbf{J}_a(\mathbf{q})$ nicht invertierbar ist, können (6.1) und (6.31c) kombiniert werden, indem die Gelenkbeschleunigung aus (6.1) in (6.31c) eingesetzt wird. Damit erhält man zunächst

$$\ddot{\mathbf{x}} = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}} \quad (6.39)$$

$$= \mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})) + \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}} \quad (6.40)$$

Das generalisierte Gelenkmoment $\boldsymbol{\tau} \in \mathbb{R}^n$ kann nun nicht mehr vollständig durch eine Kraft $\mathbf{F} \in \mathbb{R}^m$ in Aufgabenkoordinaten dargestellt werden und muss um eine Nullraumkomponente $\boldsymbol{\tau}_n \in \mathbb{R}^n$ ergänzt werden. Dieses Drehmoment $\boldsymbol{\tau}_n$ lässt sich über die Projektion eines beliebigen Gelenkmoments $\boldsymbol{\tau}_0 \in \mathbb{R}^n$ in den Nullraum anschreiben.

$$\boldsymbol{\tau} = \mathbf{J}_a^T(\mathbf{q})\mathbf{F} + \underbrace{(\mathbf{E} - \mathbf{J}_a^T(\mathbf{q})(\mathbf{J}_a^\#(\mathbf{q}))^T)}_{\boldsymbol{\tau}_n}\boldsymbol{\tau}_0 \quad (6.41)$$

Hierbei stellt $\mathbf{J}_a^\#(\mathbf{q})$ eine generalisierte Inverse der Jacobi-Matrix dar. Man beachte, dass diese Repräsentation der Nullraummomente über $\boldsymbol{\tau}_0 \in \mathbb{R}^n$ keine Minimalparametrierung darstellt. Eine Minimalrepräsentation würde nur $n - m$ Komponenten beinhalten.

Die in der Beschreibung der Nullraummomente verwendete Projektionsmatrix ist die Transponierte der Projektionsmatrix der Nullraumgeschwindigkeiten, siehe (4.90). So

³Die sogenannte *Moore-Penrose-Pseudoinverse*.

⁴Eine gewichtete rechte generalisierte Inverse mit Gewichtungsmatrix \mathbf{W} hat die Form $\mathbf{W}^{-1}\mathbf{J}_a^T(\mathbf{q})(\mathbf{J}_a(\mathbf{q})\mathbf{W}^{-1}\mathbf{J}_a^T(\mathbf{q}))^{-1}$.

lässt sich die Gelenkgeschwindigkeit als Summe zweier Komponenten anschreiben, wobei die erste Komponente den Anteil der Geschwindigkeit im Aufgabenraum und die zweite Komponente die Geschwindigkeit im Nullraum beschreibt

$$\dot{\mathbf{q}} = \mathbf{J}_a^\#(\mathbf{q})\dot{\mathbf{x}} + \left(\mathbf{E} - \mathbf{J}_a^\#(\mathbf{q})\mathbf{J}_a(\mathbf{q})\right)\dot{\mathbf{q}}. \quad (6.42)$$

Mit (6.41) nimmt (6.40) die folgende Form an

$$\ddot{\mathbf{x}} = \underbrace{\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{J}_a^\top(\mathbf{q})}_{\boldsymbol{\Lambda}(\mathbf{q})^{-1}}\mathbf{F} + \mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\tau}_n - \mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})) + \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}$$

Durch Multiplikation mit $\boldsymbol{\Lambda}(\mathbf{q})$ von links erhält man

$$\boldsymbol{\Lambda}(\mathbf{q})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}_g(\mathbf{q}) = \mathbf{F} + \boldsymbol{\Lambda}(\mathbf{q})\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\tau}_n, \quad ,$$

wobei die Dynamikkomponenten durch

$$\begin{aligned} \boldsymbol{\Lambda}(\mathbf{q}) &= \left(\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{J}_a^\top(\mathbf{q})\right)^{-1} \\ \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) &= \boldsymbol{\Lambda}(\mathbf{q})\left(\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \dot{\mathbf{J}}_a(\mathbf{q})\right)\dot{\mathbf{q}} \\ \mathbf{F}_g(\mathbf{q}) &= \boldsymbol{\Lambda}(\mathbf{q})\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{g}(\mathbf{q}) \end{aligned}$$

gegeben sind. Man beachte, dass die Dynamikkomponenten im Aufgabenraum für den Sonderfall $m = n$ mit den Komponenten aus dem letzten Abschnitt übereinstimmen. Solange die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ vollen Zeilenrang hat, d.h. abseits von Singularitäten, ist die Berechnung von $\boldsymbol{\Lambda}(\mathbf{q})$ wohldefiniert. Die Zentrifugal- und Coriolisterme sind nun in dem Vektor $\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})$ zusammengefasst.

Abschließend soll die Nullraumkomponente $\boldsymbol{\tau}_n$ genauer betrachtet werden. Für $\boldsymbol{\tau}_n$, und somit für die Wahl der generalisierten Inversen $\mathbf{J}_a^\#(\mathbf{q})$, gibt es eine eindeutige Wahl, welche bewirkt, dass durch ein beliebiges, in den Nullraum projiziertes, Drehmoment $\boldsymbol{\tau}_0$ keine Beschleunigung der Aufgabenkoordinaten erzeugt wird. Aus

$$\begin{aligned} \mathbf{0} &= \boldsymbol{\Lambda}(\mathbf{q})\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\tau}_n \\ \mathbf{0} &= \mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\left(\mathbf{E} - \mathbf{J}_a^\top(\mathbf{q})(\mathbf{J}_a^\#(\mathbf{q}))^\top\right) \\ \mathbf{0} &= \mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1} - \underbrace{\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{J}_a^\top(\mathbf{q})}_{\boldsymbol{\Lambda}(\mathbf{q})^{-1}}(\mathbf{J}_a^\#(\mathbf{q}))^\top \end{aligned}$$

folgt

$$\mathbf{J}_a^\#(\mathbf{q}) = \mathbf{M}(\mathbf{q})^{-1}\mathbf{J}_a^\top(\mathbf{q})\left(\mathbf{J}_a(\mathbf{q})\mathbf{M}(\mathbf{q})^{-1}\mathbf{J}_a^\top(\mathbf{q})\right)^{-1}. \quad (6.43)$$

Die mit der Massenmatrix gewichtete Pseudoinverse $\mathbf{J}_a^\#(\mathbf{q})$ in (6.43) wird in der Robotikliteratur als *dynamisch konsistente Pseudoinverse* bezeichnet. Die Dynamik des redundanten Roboters in Aufgabenkoordinaten ergibt sich zu

$$\Lambda(\mathbf{q})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}_g(\mathbf{q}) = \mathbf{F} . \quad (6.44)$$

Im Folgenden wird gezeigt, wie der Computed-Torque-Regler im Aufgabenraum um eine Nullraumregelung erweitert werden kann. Über (6.41) stehen die beiden Stellgrößen $\mathbf{F} \in \mathbb{R}^m$ und $\boldsymbol{\tau}_0 \in \mathbb{R}^n$ zur Verfügung. Durch \mathbf{F} wird die Dynamik in Aufgabenkoordinaten entkoppelt und linearisiert, wobei die Kompensation der Gravitationsterme direkt im Gelenkraum erfolgen sollte⁵

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_g(\mathbf{q}) + \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) + \Lambda(\mathbf{q})(\ddot{\mathbf{x}}_d - \mathbf{K}_2\dot{\mathbf{e}}_x - \mathbf{K}_1\mathbf{e}_x) \\ \boldsymbol{\tau} &= \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q})(\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) + \Lambda(\mathbf{q})(\ddot{\mathbf{x}}_d - \mathbf{K}_2\dot{\mathbf{e}}_x - \mathbf{K}_1\mathbf{e}_x)) + \boldsymbol{\tau}_n , \end{aligned}$$

mit dem Regelfehler $\mathbf{e}_x = \mathbf{x} - \mathbf{x}_d$ und den positiv definiten Reglermatrizen $\mathbf{K}_1 \in \mathbb{R}^{m \times m}$ und $\mathbf{K}_2 \in \mathbb{R}^{m \times m}$. Dies führt auf die lineare kartesische Fehlerdynamik

$$\ddot{\mathbf{e}}_x + \mathbf{K}_2\dot{\mathbf{e}}_x + \mathbf{K}_1\mathbf{e}_x = \mathbf{0} . \quad (6.45)$$

Für die Regelung der Nullraumbewegung sollte zumindest eine Gelenkdämpfung $\boldsymbol{\tau}_0 = -\mathbf{K}_0\dot{\mathbf{q}}$ mit einer positiv definiten Reglermatrix \mathbf{K}_0 angesetzt werden. Zusätzlich kann über die Minimierung einer Potentialfunktion⁶ $V_p(\mathbf{q}) \geq 0$ ein weiteres Regelziel im Nullraum eingebaut werden. Somit erhält man die Nullraumregelung zu

$$\boldsymbol{\tau}_n = \left(\mathbf{E} - \mathbf{J}_a^T(\mathbf{q})(\mathbf{J}_a^\#(\mathbf{q}))^T \right) \left(-\mathbf{K}_0\dot{\mathbf{q}} - \left(\frac{\partial V_p(\mathbf{q})}{\partial \mathbf{q}} \right)^T \right) .$$

6.3.3 Zusammenfassung der Regler im Aufgabenraum

Die folgende Tabelle fasst die Regler im Aufgabenraum zusammen. Die Kompensation der Gravitationsterme erfolgt jeweils im Gelenkraum. Die im Aufgabenraum formulierten Reglerkomponenten werden über die Transponierte der Jacobi-Matrix in den Gelenkraum abgebildet. Zusätzlich ist im Fall eines redundanten Roboters noch die Nullraumkomponente $\boldsymbol{\tau}_n$ zu ergänzen. Im nichtredundanten Fall kann $\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})$ durch $\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}}$ ersetzt werden. Der kartesische Impedanzregler wird im nächsten Abschnitt beschrieben.

⁵Bei Kompensation der Gravitationsterme über $\mathbf{F}_g(\mathbf{q})$ werden die im Nullraum wirkenden Gravitationsterme nicht berücksichtigt. Dies kann zu einer unerwünschten Driftbewegung im Nullraum führen.

⁶Ein einfaches Beispiel wäre hier die Minimierung der Abweichung von einer Referenzpose $\mathbf{q}_0 \in \mathbb{R}^n$ über $V_p(\mathbf{q}) = 1/2(\mathbf{q} - \mathbf{q}_0)^T \mathbf{K}_q(\mathbf{q} - \mathbf{q}_0)$ mit einer positiv definiten Reglermatrix \mathbf{K}_q .

Bezeichnung	Regelgesetz	Stabilisierung
PDg	$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q})\mathbf{F}_x + \boldsymbol{\tau}_n$ $\mathbf{F}_x = \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) - \mathbf{K}_d\dot{\mathbf{x}}$	Arbeitspunkt
Comp. Torque	$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q})\mathbf{F}_x + \boldsymbol{\tau}_n$ $\mathbf{F}_x = \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\Lambda}(\mathbf{q})(\ddot{\mathbf{x}}_d - \mathbf{K}_2\dot{\mathbf{e}}_x - \mathbf{K}_1\mathbf{e}_x)$	Trajektorie
Impedanzregelung	$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q})(\mathbf{F}_m + \mathbf{F}_f) + \boldsymbol{\tau}_n$ $\mathbf{F}_m = \boldsymbol{\Lambda}(\mathbf{q})\ddot{\mathbf{x}}_d + \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\Lambda}(\mathbf{q})\boldsymbol{\Lambda}_d^{-1}(\mathbf{D}_d\dot{\mathbf{e}}_x + \mathbf{K}_d\mathbf{e}_x)$ $\mathbf{F}_f = (\boldsymbol{\Lambda}(\mathbf{q})\boldsymbol{\Lambda}_d^{-1} - \mathbf{E})\mathbf{F}_{ext}$	Impedanz

6.4 Kraft- und Impedanzregelung

Den bisherigen Betrachtungen wurde das Robotermodell (6.1) ohne Einfluss von externen, auf den Roboter einwirkenden, Kräften zugrunde gelegt. Falls Messungen oder Schätzungen von externen Kräften, wie Reibungskomponenten oder anderen Störkräften, vorhanden sind, können diese im Sinne einer Vorsteuerung im Konfigurationsraum kompensiert werden. In diesem Abschnitt werden nun Regelungsaufgaben beschrieben, in denen die externen Kräfte nicht als Störung betrachtet werden, sondern im gewünschten dynamischen Verhalten explizit auftreten. Daher betrachten wir nun ein um externe Kräfte erweitertes Robotermodell

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad . \quad (6.46)$$

6.4.1 Kartesische Impedanzregelung

In der kartesischen⁷ Impedanzregelung wird nicht nur eine gewünschte Roboterbewegung $\mathbf{x}_d(t)$ vorgegeben, sondern ein gewünschtes dynamisches Verhalten bezüglich der Koordinaten im Arbeitsraum. Ein Beispiel hierfür ist das *Feder-Masse-Dämpfer-System*

$$\boldsymbol{\Lambda}_d\ddot{\mathbf{e}}_x + \mathbf{D}_d\dot{\mathbf{e}}_x + \mathbf{K}_d\mathbf{e}_x = \mathbf{F}_{ext} \quad (6.47)$$

mit $\mathbf{e}_x = \mathbf{x} - \mathbf{x}_d$ und den vorgegebenen positiv definiten Matrizen $\boldsymbol{\Lambda}_d \in \mathbb{R}^{m \times m}$, $\mathbf{D}_d \in \mathbb{R}^{m \times m}$ und $\mathbf{K}_d \in \mathbb{R}^{m \times m}$ für die Masse, Dämpfung bzw. Steifigkeit des Systems, siehe Abbildung 6.4. Die externe Kraft \mathbf{F}_{ext} in (6.47) wirkt zwischen dem Roboter und der Umgebung und lenkt die virtuelle Masse von der (zeitveränderlichen) virtuellen Ruhelage \mathbf{x}_d aus. Die Kraft \mathbf{F}_{ext} wirkt über die externen Drehmomente $\boldsymbol{\tau}_{ext}$ entsprechend $\boldsymbol{\tau}_{ext} = \mathbf{J}_a^T(\mathbf{q})\mathbf{F}_{ext}$ auf die Roboterdynamik.

Das Regelgesetz für die Impedanzregelung im Arbeitsraum lässt sich finden, indem man (6.47) mit der Dynamik des Roboters in Aufgabenkoordinaten vergleicht. Im nichtredun-

⁷Sind die Aufgabenkoordinaten aus der Position und Orientierung des Endeffektors zusammengesetzt, so spricht man auch von *kartesischen Koordinaten*.

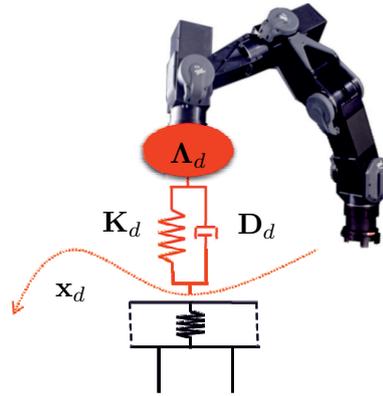


Abbildung 6.4: Konzept der Impedanzregelung: Das Sollverhalten ist im Sinne einer dynamischen Systemantwort auf externe Kräfte gegeben. Für den Fall der freien Bewegung soll der Roboter einer Trajektorie $\mathbf{x}_d(t)$ asymptotisch stabil folgen. Kommt der Roboter in Kontakt mit der Umgebung, so soll er sich entsprechend eines vorgegebenen Feder-Masse-Dämpfer-Systems verhalten.

danten Fall⁸ hat das Modell in Aufgabenkoordinaten (vgl. (6.34)), die Form

$$\mathbf{\Lambda}(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{F}_g(\mathbf{q}) = \mathbf{F} + \mathbf{F}_{ext} . \quad (6.48)$$

Aus (6.47) erhält man

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_d + \mathbf{\Lambda}_d^{-1}(\mathbf{F}_{ext} - \mathbf{D}_d\dot{\mathbf{e}}_x - \mathbf{K}_d\mathbf{e}_x) .$$

Der Ausdruck für $\ddot{\mathbf{x}}$ kann nun in die Dynamikgleichung (6.48) eingesetzt werden und man erhält den kartesischen Impedanzregler zu

$$\mathbf{F} = \mathbf{\Lambda}(\mathbf{q})\ddot{\mathbf{x}}_d + \mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{F}_g(\mathbf{q}) - \mathbf{\Lambda}(\mathbf{q})\mathbf{\Lambda}_d^{-1}(\mathbf{D}_d\dot{\mathbf{e}}_x + \mathbf{K}_d\mathbf{e}_x) + \left(\mathbf{\Lambda}(\mathbf{q})\mathbf{\Lambda}_d^{-1} - \mathbf{E}\right)\mathbf{F}_{ext} \quad (6.49)$$

Die kartesische Kraft \mathbf{F} wird wieder über $\boldsymbol{\tau} = \mathbf{J}_a^T(\mathbf{q})\mathbf{F}$ in den Konfigurationsraum abgebildet. Wie im Fall des Computed-Torque-Reglers bietet es sich an die Gravitationsterme direkt im Konfigurationsraum zu kompensieren.

6.4.2 Kartesische Nachgiebigkeitsregelung

Für die Impedanzregelung mit einer vorgegebenen Sollträgheit $\mathbf{\Lambda}_d$ ist die Messung (oder Schätzung) der externen Kräfte und Drehmomente \mathbf{F}_{ext} in allen m Dimensionen des Aufgabenraumes notwendig. Insbesondere im Kontakt mit sehr steifen Umgebungen

⁸Der Impedanzregler kann für den redundanten Fall auf die gleiche Weise hergeleitet werden, indem man als Robotermodell in Aufgabenkoordinaten anstatt (6.48) das Modell (6.44), ergänzt um die zusätzliche externe Kraft \mathbf{F}_{ext} , zugrunde legt. Dies hat zur Konsequenz, dass in (6.49) der Term $\mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}}$ durch $\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})$ zu ersetzen ist.

wirken sich kommunikationsbedingte Totzeiten in der Kraftmessung auf die Stabilität des geschlossenen Regelkreises aus.

Hingegen ist in vielen Fällen die Erzielung eines nachgiebigen Verhaltens mit vorgegebener Steifigkeit und Dämpfung ausreichend. Wie man in (6.49) sieht, entfällt für den Fall $\Lambda_d = \Lambda(\mathbf{q})$ die Notwendigkeit der Messung der externen Kräfte. In diesem Fall sollte auch auf die Kompensation der Zentrifugal- und Coriolisterme verzichtet werden um die Passivitätseigenschaften der Roboterdynamik im geschlossenen Regelkreis zu erhalten. Für den Fall einer konstanten Sollposition \mathbf{x}_d führt dieses Vorgehen zu einem Regelgesetz das genau dem PD-Regler mit Gravitationskompensation in Aufgabenkoordinaten entspricht.

Der allgemeinere Fall einer Nachgiebigkeitsregelung um eine Solltrajektorie führt auf einen PD+-Regler in Aufgabenkoordinaten und ist in Anhang II (Abschnitt 6.6.2) beschrieben.

6.4.3 Orientierungssteifigkeit

In der bisherigen Behandlung der Regelung im Aufgabenraum wurde von einer Aufgabenkoordinate $\mathbf{x} = \mathbf{f}(\mathbf{q})$ ausgegangen. In vielen Fällen beschreibt \mathbf{x} die Lage \mathbf{p}_e und Orientierung ${}^w\mathbf{R}_e \in SO(3)$ des Endeffektors, wobei für ${}^w\mathbf{R}_e$ ein Satz von Minimalkoordinaten (Euler-Winkel) ϕ_e gewählt werden muss. Ein grundsätzliches Problem in dieser Darstellung betrifft die Repräsentationssingularitäten der Euler-Winkel. Wie wir in Kapitel 4.1.3 gesehen haben, treten diese Repräsentationssingularitäten für jede mögliche Wahl von Minimalkoordinaten für die Orientierung auf. Da der Roboter mit seinem Endeffektor (innerhalb seines Arbeitsraumes) beliebige Bahnen realisieren können soll, können diese Singularitäten nur schwierig durch eine geschickte Bahnplanung vermieden werden. Eine mögliche Behandlung dieses Problems ist es, in der Regelung zwischen zwei Sätzen von Euler-Winkeln mit unterschiedlichen Repräsentationssingularitäten zu wechseln.

Im Rahmen der Positionsregelung gibt es allerdings eine bessere Herangehensweise. In der Positionsregelung wird der Regelfehler durch Wahl hinreichend großer Reglermatrizen gering gehalten⁹. Dann können die Repräsentationssingularitäten effektiv vermieden werden, indem man die Euler-Winkel nicht mehr aus der absoluten Endeffektor-Orientierung ${}^w\mathbf{R}_e$ bezüglich des Inertialkoordinatensystems, sondern aus der relativen Rotation ${}^d\mathbf{R}_e = {}^d\mathbf{R}_w {}^w\mathbf{R}_e$ zwischen dem Endeffektor ${}^w\mathbf{R}_e$ und der gewünschten Orientierung ${}^w\mathbf{R}_d(t)$ berechnet.

Ersetzt man nun in dem Sollverhalten (6.47) bzw. (6.45) die Komponente des Orientierungsfehlers von \mathbf{e}_x durch $\phi({}^d\mathbf{R}_e)$, so kann man die in diesem Kapitel vorgestellten Regler auf den Fall der relativen Rotation ${}^d\mathbf{R}_e$ verallgemeinern.

6.4.4 Kraftregelung

In der Kraftregelung soll die Interaktionskraft $\mathbf{F}_{int} = -\mathbf{F}_{ext}$ zwischen dem Roboter und seiner Umgebung, üblicherweise in einzelnen Komponenten des Aufgabenvektors \mathbf{x} , einer vorgegebenen Trajektorie $\mathbf{F}_d(t)$ folgen. Hierbei kann der Zusammenhang zwischen den Drehmomenten im Konfigurationsraum und im Aufgabenraum ausgenutzt werden. Da

⁹Man beachte, dass dies im Fall der Impedanzregelung nicht unbedingt gilt, da hier in einzelnen Raumrichtungen auch ein stark nachgiebiges Verhalten gefordert sein kann.

Messungen der externen Kräfte und Momente üblicherweise mit deutlichen Rauschanteilen versehen sind, wird auf einen D-Anteil im Regler verzichtet. Die Sollkraft $\mathbf{F}_d(t)$ variiert meist nur relativ langsam. Jedoch wird in der Kraftregelung meist eine hohe stationäre Genauigkeit gefordert. Dies motiviert eine Reglerstruktur mit Proportional- und Integralanteil. Ein einfacher Kraftregler mit PI-Struktur hat die folgende Form:

$$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q}) + \mathbf{J}_a^T(\mathbf{q}) \left(\mathbf{F}_d + \mathbf{K}_f \mathbf{e}_f + \mathbf{K}_i \int \mathbf{e}_f dt - \mathbf{K}_d \dot{\mathbf{x}} \right) \quad (6.50)$$

mit den positiv definiten Reglermatrizen $\mathbf{K}_f, \mathbf{K}_i, \mathbf{K}_d \in \mathbb{R}^{m \times m}$ und dem Kraftregelfehler $\mathbf{e}_f = \mathbf{F}_d - \mathbf{F}_{int}$. Neben der Gravitationskompensation im Konfigurationsraum und dem PI-Regler wird die Sollkraft \mathbf{F}_d auch direkt vorgesteuert. Ein zusätzlicher Dämpfungsterm $-\mathbf{K}_d \dot{\mathbf{x}}$ limitiert die resultierende Beschleunigung, falls der Endeffektor den Kontakt zur Umgebung verlieren sollte.

Wird der Kraftregler im Kontakt mit einer steifen Umgebung eingesetzt, so ist die Bewegung des Roboters im Aufgabenraum sehr klein $\ddot{\mathbf{x}} \approx \mathbf{0}, \dot{\mathbf{x}} \approx \mathbf{0}$. Die resultierende Dynamik des Kraftreglers ergibt sich dann mit (6.48) und (6.50) zu

$$(\mathbf{E} + \mathbf{K}_f) \mathbf{e}_f + \mathbf{K}_i \int \mathbf{e}_f dt = \mathbf{0} \quad . \quad (6.51)$$

Neben einer reinen Kraftregelung ist insbesondere auch die sogenannte *hybride Kraft/Positions-Regelung* [6.1, 6.2] von praktischer Bedeutung. Hierbei wird nur für einzelne Richtungen des Aufgabenraumes eine Sollkraft vorgegeben, während die restlichen Raumrichtungen in der Position geregelt werden.

6.5 Ausblick: Unteraktuierte Systeme

Die in diesem Kapitel vorgestellten Regelungsmethoden basieren auf der Dynamikdarstellung (6.1). Hierbei wurde angenommen, dass es zu jeder Koordinate q_i des Konfigurationsraumes eine generalisierte Kraft τ_i gibt, die man in der Regelung als Stellgröße nutzen kann, d.h. man hat ebenso viele Stellgrößen wie Gelenkkoordinaten. Dies ist z.B. für einen seriellen Manipulator mit feststehender Basis gegeben. Für viele Klassen von Robotersystemen stehen jedoch weniger Stellgrößen zur Verfügung. Man spricht in diesem Fall von einem *unteraktuierten mechanischen System*. Ein unteraktuiertes mechanisches System mit p Stellgrößen u_i lässt sich in der Form

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{B}(\mathbf{q}) \mathbf{u} \quad (6.52)$$

mit einer Matrix $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n \times p}$ und einem Vektor der Stellgrößen $\mathbf{u} \in \mathbb{R}^p$ schreiben. Für¹⁰ $p < n$ ist die Matrix $\mathbf{B}(\mathbf{q})$ nicht invertierbar und somit kann man keine beliebige generalisierte Kraft $\boldsymbol{\tau}$ auf den Roboter ausüben. In diesem Fall ist z.B. ein Computed-Torque-Regler, wie in Abschnitt 6.2.3 beschrieben, nicht mehr realisierbar.

¹⁰Oder für $p = n$, wenn $\mathbf{B}(\mathbf{q})$ keinen vollen Rang hat.

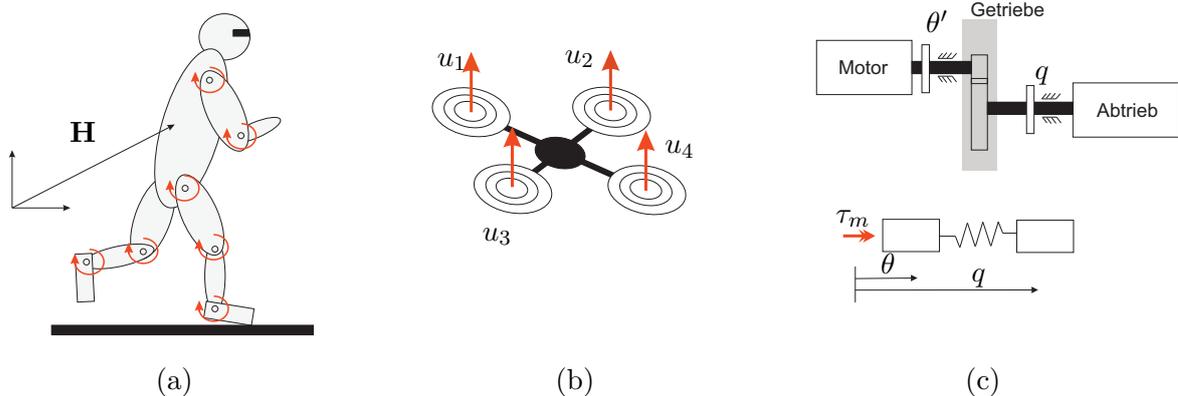


Abbildung 6.5: Beispiele für unteraktuierte Robotersysteme: (a) Laufroboter und humanoide Roboter, (b) Flugroboter, (c) Roboter mit elastischen Antrieben

Bemerkung 6.2. Zum Teil wird in der Literatur eine noch allgemeinere Definition für unteraktuierte Systeme benutzt, nach der ein System unteraktuiert ist, wenn die Abbildung $\mathbf{u} \rightarrow \boldsymbol{\tau}$ nicht surjektiv ist, d.h. dass nicht jeder beliebige Vektor $\boldsymbol{\tau} \in \mathbb{R}^n$ darstellbar ist. Bei dieser Definition inkludiert man z.B. auch Systeme mit Stellgrößenbeschränkungen $u_i < u_{i,max}$ und betrachtet sie als unteraktuiert.

Als Beispiele für unteraktuierte Robotersysteme können z.B. Laufroboter, Flugroboter, oder Roboter mit elastischen Gelenken genannt werden (siehe Abbildung 6.5). In zwei- und mehrbeinigen Laufrobotern gibt es einen zentralen Basiskörper (z.B. Torso, Hüfte) der als Ursprung der kinematischen Baumstruktur betrachtet wird. Alle weiteren Glieder sind über eindimensionale Gelenke entsprechend der Baumstruktur angebunden. Auf den Basiskörper können jedoch direkt keine generalisierten Kräfte aufgebracht werden. Diese können allerdings, in Abhängigkeit der wirkenden Kontaktbedingungen (single/double support) zumindest teilweise indirekt realisiert werden.

Als Beispiel für einen unteraktuierten Flugroboter können z.B. Quadrocopter betrachtet werden. Durch die vier Propeller kann jeweils eine Kraft in der Normalenrichtung des Propellers erzeugt werden, d.h. $\mathbf{u} \in \mathbb{R}^4$. Im Schwebeflug (mit vertikaler Orientierung) ist eine direkte Erzeugung einer horizontalen Kraftkomponente nicht möglich. Um den Quadrocopter horizontal zu bewegen, muss zunächst eine Orientierungsänderung durch ein Kippmoment erfolgen, so dass die Propellerkraft eine horizontale Komponente beinhaltet.

In elastischen Robotern sind die abtriebsseitigen Glieder über physikalische Federelemente mit den Rotormassen verbunden. Diese Federelemente können z.B. die Getriebeelastizität widerspiegeln. Aufgrund der Feder ist in der Modellierung zwischen den Motorwinkeln und den Gelenkwinkeln zu unterscheiden. Man erhält für einen Roboter mit n elastischen Gelenken also $2n$ generalisierte Koordinaten - bei allerdings nur n Motordrehmomenten als Stellgrößen.

Für diese doch sehr unterschiedlichen Klassen von Robotersystemen findet man in der Literatur auch unterschiedliche Regelungsansätze, in denen die jeweilige Form der Unteraktuierung gezielt behandelt wird. Einige dieser Systemklassen werden in der weiterführenden VU „Manipulation und Lokomotion“ behandelt.

6.6 Anhang

6.6.1 Anhang I: Lyapunov-Funktion des PD-Reglers im Gelenkraum

Wie in Abschnitt 6.2.1 gezeigt, ergibt sich für den PD-Regler bei hinreichend großer Reglerverstärkung \mathbf{K}_p eine Ruhelage \mathbf{q}^* aus der Lösung der statischen Gleichgewichtsbedingung (6.15). Die Stabilitätsanalyse kann für diesen Punkt mithilfe einer Lyapunov-Funktion erfolgen, welche sich aus der kinetischen und potentiellen Energie des Roboters und der Potentialfunktion der virtuellen Steifigkeit zusammensetzt

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \mathcal{V}_g(\mathbf{q}) + \frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q + V_0, \quad (6.53)$$

wobei V_0 ein konstanter Offsetterm ist. Dieser stellt sicher, dass $V(\mathbf{q}^*, \mathbf{0}) = 0$ gilt, d.h. $V_0 = -V_k(\mathbf{q}^* - \mathbf{q}_d) - \mathcal{V}_g(\mathbf{q}^*)$. Unter der Bedingung (6.16) ist die Funktion $V(\mathbf{q}, \dot{\mathbf{q}})$ im Arbeitspunkt $(\mathbf{q}^*, \dot{\mathbf{q}}^* = \mathbf{0})$ positiv definit. Die zeitliche Ableitung ergibt sich zu

$$\begin{aligned} \dot{V}(\mathbf{q}, \dot{\mathbf{q}}) &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{q}} + \frac{\partial \mathcal{V}_g(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T (-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \dot{\mathbf{e}}_q - \mathbf{K}_p \mathbf{e}_q) + \mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})^T \dot{\mathbf{q}} \\ &= \frac{1}{2} \dot{\mathbf{q}}^T \underbrace{(\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}))}_{=0} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} \\ &= -\dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} \leq 0, \end{aligned}$$

woraus mit positiv definiten Verstärkungsmatrizen \mathbf{K}_p und \mathbf{K}_d die Stabilität des geschlossenen Regelkreises gefolgert werden kann. Durch Anwendung des Invarianzprinzips von Krasovskii-LaSalle¹¹ kann für diesen Fall sogar die asymptotische Stabilität gezeigt werden.

¹¹siehe VO „Nichtlineare dynamische Systeme und Regelung“

6.6.2 Anhang II: Kartesische Nachgiebigkeitsregelung um eine Solltrajektorie

Das gewünschte Systemverhalten einer Nachgiebigkeitsregelung um eine Solltrajektorie lässt sich in der Form

$$\Lambda(\mathbf{q})\ddot{\mathbf{e}}_x + (\Gamma(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{D}_d)\dot{\mathbf{e}}_x + \mathbf{K}_d\mathbf{e}_x = \mathbf{F}_{ext} \quad (6.54)$$

ansetzen. Im Vergleich zu (6.47) wurde hierbei neben der Beibehaltung der konfigurationsabhängigen kartesischen Massenmatrix auch die entsprechende Coriolismatrix ergänzt. Das Regelgesetz der Impedanzregelung vereinfacht sich damit zu

$$\mathbf{F} = \Lambda(\mathbf{q})\ddot{\mathbf{x}}_d + \Gamma(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}}_d + \mathbf{F}_g(\mathbf{q}) - \mathbf{D}_d\dot{\mathbf{e}}_x - \mathbf{K}_d\mathbf{e}_x . \quad (6.55)$$

Man beachte, dass dies exakt einem PD+-Regler in Aufgabenkoordinaten entspricht! Durch die konfigurationsabhängige kartesische Massenmatrix erhält man ein zeitvariantes nichtlineares Systemverhalten. Passivität und Stabilität der Solldynamik lassen sich mit der positiv definiten Funktion

$$V(\mathbf{e}_x, \dot{\mathbf{e}}_x, t) = \frac{1}{2}\dot{\mathbf{e}}_x^T \Lambda(\mathbf{q})\dot{\mathbf{e}}_x + \frac{1}{2}\mathbf{e}_x^T \mathbf{K}_d\mathbf{e}_x \quad (6.56)$$

beurteilen. Für die zeitliche Ableitung von (6.56) erhält man durch Ausnutzen der Passivitätseigenschaft (6.37)

$$\dot{V}(\mathbf{e}_x, \dot{\mathbf{e}}_x, t) = -\dot{\mathbf{e}}_x^T \mathbf{D}_d\dot{\mathbf{e}}_x + \dot{\mathbf{e}}_x^T \mathbf{F}_{ext} , \quad (6.57)$$

woraus sich Passivität in der physikalischen Interaktion, sowie Stabilität für den Fall der freien Bewegung, d.h. für $\mathbf{F}_{ext} = \mathbf{0}$, folgern lässt. Für den Nachweis der asymptotischen Stabilität kann, wie beim PD+-Regler im Konfigurationsraum, auch eine strikte Lyapunov-Funktion gefunden werden.

6.7 Literatur

- [6.1] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [6.2] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.

7 Bewegungsplanung

Autonome Robotersysteme und Serviceroboter sind mittlerweile in der Lage, sehr komplexe Aufgaben zu bewältigen. Dazu müssen im Robotersystem viele Teilprozesse ablaufen: Der Roboter muss

- die beabsichtigten Aktionen planen,
- die beabsichtigten Auswirkungen dieser Aktion bestimmen,
- ungewünschte Nebeneffekte abschätzen,
- die Ausführbarkeit einer beabsichtigten Aktion beurteilen,
- die Erreichbarkeit der Ziele prüfen,
- die Entscheidung treffen, wie die gewünschte Aktion ausgeführt werden soll, . . .

Diese Vorgänge in der Software des Robotersystems werden in der englischsprachigen Literatur als *reasoning* (dt. *Gedankengang*) bezeichnet, wobei darunter *kein* menschenähnliches Denken verstanden wird [7.1]. In diesem Zusammenhang wird häufig maschinelles Lernen eingesetzt. Soll beispielsweise ein scheinbar einfacher Vorgang wie das Aufnehmen eines Objektes von einem Tisch durchgeführt werden, so müssen eine Reihe von Entscheidungen und Abschätzungen durch das Robotersystem getroffen werden, siehe Abbildung 7.1:

- Wo muss sich der Roboter platzieren, um das Objekt aufnehmen zu können?
- Welcher Roboterarm soll verwendet werden?
- Wie soll die Bewegung für die Annäherung zum Objekt durchgeführt werden?
- Welche Art des Griffs soll verwendet werden?
- Wo soll das Objekt gegriffen werden?
- Wieviel Greifkraft soll angewendet werden?
- Mit wieviel Kraft soll das Objekt angehoben werden?

Ein Roboterprogramm, das alle diese Entscheidungen in jeder erdenklichen Kombination von Objekt und Aufgabe treffen könnte, wäre sehr komplex – und wäre auch dann nicht ausreichend für ein überzeugendes Roboterverhalten in allen Situationen. Das Aufnehmen eines Objektes ist beispielsweise auch vom Kontext abhängig, also vom Zustand und der Lage des Objektes, von der aktuell ausgeführten Aufgabe und auch von der Umgebung, in der sich das Objekt befindet: Wenn ein Glas mit Wasser gefüllt ist, so muss es immer aufrecht gehalten werden. Wenn ein Roboter ein Glas aus einer Flasche befüllen möchte, dann sollte die Flasche nicht an der Öffnung gegriffen werden. Wird die Flasche aber von einem Regal gegriffen, so ist der Griff von oben möglicherweise die einzige Option. Die Komplexität wird nochmals deutlich erhöht, wenn nicht eine einzelne Aktion ausgeführt sondern eine zusammengesetzte Tätigkeit, wie z. B. das Abräumen eines Tisches. Für eine solche Tätigkeit muss der Roboter selbst die notwendigen Parameter für das Greifen der



Abbildung 7.1: Humanoider Roboter ARMAR in der Küche des „Collaborative Research Center“ in Karlsruhe, Deutschland [7.2].

Objekte ableiten und muss zuordnen können, wo jedes Objekt verstaut werden soll und in welcher Reihenfolge die Objekte abgeräumt werden sollen. Weiters muss der Roboter abschätzen, ob es besser ist, ein Tablett zu holen, um beim Abräumen effizienter zu sein, oder die Türen von Kästen während dem Ablauf der Tätigkeit offen zu lassen.

Um solche Aufgaben und Tätigkeiten mit einem Roboter bewerkstelligen zu können, ist ein gutes Zusammenspiel zwischen Hard- und Software notwendig. Auf der Software-Seite sind zusätzlich zur Aufgaben-, Bewegungs- und Trajektorienplanung auch noch eine geeignete *Wissensdarstellung* (engl. *knowledge representation*) sowie Algorithmen für das *Reasoning* notwendig [7.1]. Die letztgenannten Komponenten werden häufig mithilfe von maschinellem Lernen implementiert. Ein Beispiel [7.3] einer Architektur für die Bewegungsplanung eines humanoiden Service-Roboters zeigt Abbildung 7.2.

Die obige Einleitung für dieses Kapitel geht über die reine Aufgaben- und Bewegungsplanung hinaus. Sie soll einerseits einen Weitblick geben und soll andererseits die Komplexität, die Anforderungen sowie die Einflussfaktoren bei einfach erscheinenden Aufgaben verdeutlichen. In den folgenden Abschnitten werden die Grundlagen zur Bewegungsplanung diskutiert. Beginnend bei der untersten (hardwarenahen) Ebene – der Trajektorienplanung – wird das Niveau der Abstraktion schrittweise bis zur Aufgabenplanung erhöht. Für die Themen Reasoning und Wissensdarstellung sei auf die Literatur verwiesen [7.1–7.3].

7.1 Trajektorienplanung

Die Bewegung eines Roboters bzw. eines allgemeinen Starrkörpersystems als Funktion der Zeit wird als *Trajektorie* bezeichnet. Eine Trajektorie kann sowohl die gesamte Roboterbewegung im Konfigurationsraum als Funktion $\mathbf{q}(t)$ als auch die Endeffektorbewegung

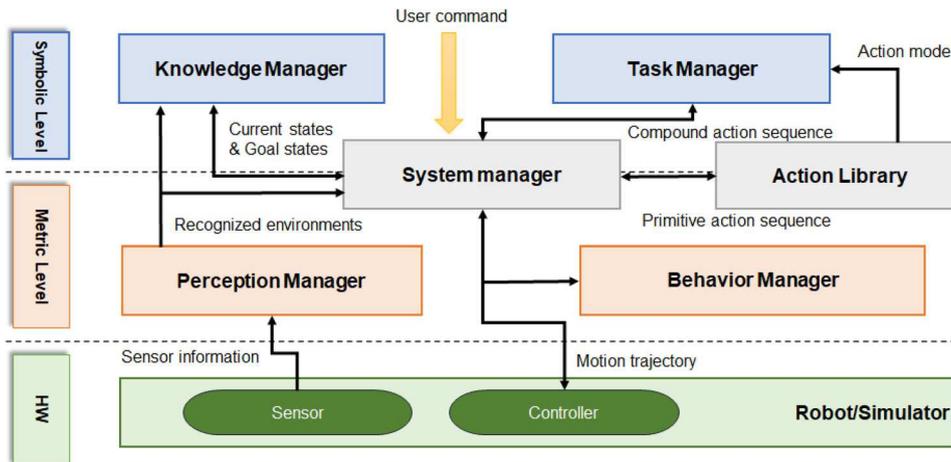


Abbildung 7.2: Beispiel einer Bewegungsplanung für einen humanoiden Service-Roboter [7.3].

im Aufgabenraum mit $\mathbf{x}_e(t)$ beschreiben. Bei manchen Anwendungen ist die Trajektorie des Roboters vollständig durch die zu lösende Aufgabe vorgegeben, wenn beispielsweise der Endeffektor einem Objekt folgen soll. Häufig wird aber auch nur eine Start- und Zielkonfiguration oder eine Start- und Zielpose für eine Roboterbewegung vorgegeben und die Bewegung zwischen diesen Punkten im Konfigurations- bzw. Aufgabenraum kann frei gewählt werden. Die Aufgabe der *Trajektorienplanung* ist es nun, Trajektorien zu berechnen, welche folgende Anforderungen erfüllen:

- Die Trajektorie muss eine hinreichend glatte Funktion der Zeit sein, damit sie durch das Starrkörpersystem mit den vorhandenen Aktuatoren bzw. den unterlagerten Regelkreisen ausgeführt werden kann.
- Die physikalischen Beschränkungen für die Gelenkpositionen und Gelenksgeschwindigkeiten sowie die Beschleunigungen und Drehmomente müssen eingehalten werden.
- Die Berechnung einer Trajektorie kann viel Zeit in Anspruch nehmen, wenn viele Randbedingungen wie Hindernisse im Aufgabenraum oder Zeitoptimalität berücksichtigt werden sollen. Die berechneten Trajektorien müssen hingegen einfach auszuwerten sein, damit dies während der Ausführung in Echtzeit passieren kann.

In diesem Abschnitt wird eine allgemeine Trajektorie aus zwei Teilen zusammengesetzt, nämlich einem *Pfad*, d. h. einer rein geometrischen Beschreibung der Bewegung im Konfigurations- oder Aufgabenraum, und einer *Zeitparametrierung*. In den weiteren Abschnitten werden einerseits geradlinige Bewegungen zwischen einem Start- und Zielpunkt berechnet und andererseits zwei Ansätze für die Zeitparametrierung vorgestellt.

7.1.1 Pfade und Trajektorien

Ein Pfad $\mathbf{q}(s)$ ist eine Abbildung des *Pfadparameters* s auf einen Punkt im Konfigurationsraum des Roboters. Analog kann auch ein Pfad $\mathbf{x}_e(s)$ im Aufgabenraum des Roboters definiert werden. Dabei wird angenommen, dass $s = 0$ beim Startpunkt gilt und mit

$s = 1$ der Zielpunkt erreicht wird. Weiters spezifiziert nun die Zeitparametrierung $s(t)$ den Zeitverlauf von s und weist jedem Zeitpunkt t zwischen dem Anfangszeitpunkt $t = 0$ und dem Endzeitpunkt $t = T$ einen Wert zwischen 0 und 1 zu. Damit ist die Zeitparametrierung als Funktion $s : [0, T] \rightarrow [0, 1]$ definiert.

Ein Pfad $x(s)$ zusammen mit einer Zeitparametrierung $s(t)$ spezifiziert eine *Trajektorie* $x(s(t))$ für eine Koordinate im Konfigurations- oder Aufgabenraum. Die zugehörige Geschwindigkeit und Beschleunigung als Funktion der Zeit werden unter Verwendung der Kettenregel gemäß

$$x = x(s(t)) \quad (7.1a)$$

$$\dot{x} = \frac{\partial x}{\partial s} \dot{s} \quad (7.1b)$$

$$\ddot{x} = \frac{\partial x}{\partial s} \ddot{s} + \frac{\partial^2 x}{\partial s^2} \dot{s}^2 \quad (7.1c)$$

berechnet.

Ohne Beschränkung der Allgemeinheit wird die Trajektorienplanung im Weiteren für eine skalare Funktion $x(s(t))$ vorgestellt. Die folgenden Konzepte können gleichermaßen auf vorgegebene Pfade und Zeitfunktionen im n -dimensionalen Konfigurationsraum $\mathbf{q}_d(s(t))$ und m -dimensionalen Arbeitsraum $\mathbf{x}_{e,d}(s(t))$ angewendet werden.

7.1.2 Geradlinige Pfade

Eine häufige Aufgabe ist es, einen Roboter von einer Konfiguration in eine andere Konfiguration zu bewegen, wobei der Roboter vor und nach der Bewegung im Stillstand sein soll und die genaue Bewegung dazwischen „frei“ ist. Diese Art der Bewegung wird *Punkt-zu-Punkt-Bewegung* (engl. *point-to-point motion*, PTP) genannt und ist im einfachsten Fall ein geradliniger Pfad im Konfigurationsraum. Mit dieser Wahl des Pfades ist immer gewährleistet, dass die mechanischen Beschränkungen der Gelenkpositionen $q_{i,min} \leq q_i \leq q_{i,max}$, $i = 1, \dots, n$, stets eingehalten werden. Solche geradlinigen Bewegungen sind auch im Aufgabenraum des Roboters möglich und diese werden als *Linearbewegung* (engl. *linear motion*, LIN) bezeichnet. Bei einer Linearbewegung ist hingegen nicht garantiert, dass die Beschränkungen in den Gelenkpositionen erfüllt sind.

Für eine skalare Koordinate x im Konfigurations- oder Aufgabenraum wird ein geradliniger Pfad $x(s)$ von einem Startpunkt x_0 zu einem Zielpunkt x_1 mit $s \in [0, 1]$ in der Form

$$x(s) = x_0 + s(x_1 - x_0), \quad s \in [0, 1] \quad (7.2)$$

mit den Ableitungen, vgl. (7.1)

$$\frac{\partial x}{\partial s} = x_1 - x_0 \quad (7.3a)$$

$$\frac{\partial^2 x}{\partial s^2} = 0 \quad (7.3b)$$

definiert. Eine geradlinige Bewegung im Konfigurationsraum führt im Allgemeinen *nicht* zu einer geradlinigen Bewegung im Aufgabenraum aufgrund des im Allgemeinen nichtlinearen

Zusammenhangs über die Roboterkinematik (4.5) bzw. (4.53). Dies soll im Folgenden anhand des Beispiels eines 2-Achs-Roboters gezeigt werden.

Beispiel 7.1 (Geradlinige Pfade eines 2-Achs-Roboters). In diesem Beispiel sollen geradlinige Pfade für einen 2-Achs-Roboter sowohl im Konfigurationsraum als auch im Aufgabenraum geplant werden. Der 2-Achs-Roboter mit den Gelenkwinkeln $\mathbf{q}^T = [q_1 \ q_2]$ und den Abmessungen d_1 und d_2 ist in Abbildung 7.3a dargestellt. Die Gelenkwinkel sind mechanisch mit $0^\circ \leq q_1 \leq 180^\circ$ und $0^\circ \leq q_2 \leq 150^\circ$ beschränkt. Der Startpunkt des Pfades ist $\mathbf{q}_0^T = [q_{0,1} \ q_{0,2}]$ und der Endpunkt wird mit $\mathbf{q}_1^T = [q_{1,1} \ q_{1,2}]$ bezeichnet.

Konfigurationsraum Im Konfigurationsraum werden geradlinige Pfade in der Form (7.2) für jedes Gelenk einzeln geplant, d. h.

$$\mathbf{q}(s) = \begin{bmatrix} q_1(s) \\ q_2(s) \end{bmatrix} = \begin{bmatrix} q_{0,1} + s(q_{1,1} - q_{0,1}) \\ q_{0,2} + s(q_{1,2} - q_{0,2}) \end{bmatrix}, \quad s \in [0, 1], \quad (7.4)$$

bzw. in Vektorschreibweise

$$\mathbf{q}(s) = \mathbf{q}_0 + s(\mathbf{q}_1 - \mathbf{q}_0), \quad s \in [0, 1]. \quad (7.5)$$

In den Abbildungen 7.3b und 7.3c ist der durch die Gelenkwinkelbeschränkungen zulässige Konfigurations- bzw. Aufgabenraum als hellgraue Fläche dargestellt. In Abbildung 7.3b ist der geradlinige Pfad zwischen den Punkten \mathbf{q}_0 und \mathbf{q}_1 ersichtlich. Im Gegensatz dazu entsteht durch diese geradlinige Bewegung im Konfigurationsraum eine kurvige Bewegung im Aufgabenraum, siehe Abbildung 7.3c.

Aufgabenraum Mithilfe der Vorwärtskinematik des 2-Achs-Roboters $\mathbf{f}(\mathbf{q})$ gemäß

$$\mathbf{x}_e = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{p}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} d_1 \cos(q_1) + d_2 \cos(q_1 + q_2) \\ d_1 \sin(q_1) + d_2 \sin(q_1 + q_2) \end{bmatrix} \quad (7.6)$$

werden zunächst der Start- und der Endpunkt \mathbf{q}_0 bzw. \mathbf{q}_1 in den Aufgabenraum gemäß

$$\mathbf{x}_0 = \mathbf{f}(\mathbf{q}_0) = [x_{0,1} \ x_{0,2}]^T \quad (7.7)$$

$$\mathbf{x}_1 = \mathbf{f}(\mathbf{q}_1) = [x_{1,1} \ x_{1,2}]^T \quad (7.8)$$

übertragen. Analog zu (7.2) wird nun im Aufgabenraum ein geradliniger Pfad

für den Endeffektor $\mathbf{x}_e(s)$

$$\mathbf{x}_e(s) = \begin{bmatrix} x_{e,1}(s) \\ x_{e,2}(s) \end{bmatrix} = \begin{bmatrix} x_{0,1} + s(x_{1,1} - x_{0,1}) \\ x_{0,2} + s(x_{1,2} - x_{0,2}) \end{bmatrix} = \mathbf{x}_0 + s(\mathbf{x}_1 - \mathbf{x}_0), \quad s \in [0, 1], \quad (7.9)$$

geplant. Um diese Bewegung auszuführen, kann einerseits die inverse differentielle Kinematik aus Abschnitt 4.1.5 verwendet werden. Damit wird zunächst die Roboterbewegung in den Konfigurationsraum umgerechnet und anschließend kann diese mithilfe eines Reglers im Konfigurationsraum ausgeführt werden. Andererseits kann die Regelung auch im Aufgabenraum erfolgen (siehe Kapitel 7 „Regelung“) und der Pfad (7.9) zusammen mit einer geeigneten Zeitparametrierung $s(t)$ kann direkt im geschlossenen Regelkreis vorgegeben werden.

Der geradlinige Pfad im Aufgabenraum des 2-Achs-Roboters ist in Abbildung 7.3e dargestellt, während der zugehörige, kurvige Pfad im Konfigurationsraum in Abbildung 7.3d zu sehen ist. In beiden Darstellungen ist deutlich zu sehen, dass der geplante Pfad den Arbeitsbereich des Roboters verlässt und die untere Gelenkwinkelbeschränkung von q_1 verletzt wird. Dies zeigt, dass besonders bei der Pfadplanung im Aufgabenraum auf die Einhaltung der mechanischen und physikalischen Beschränkungen der Gelenkwinkel, -geschwindigkeiten und -beschleunigungen geachtet werden muss.

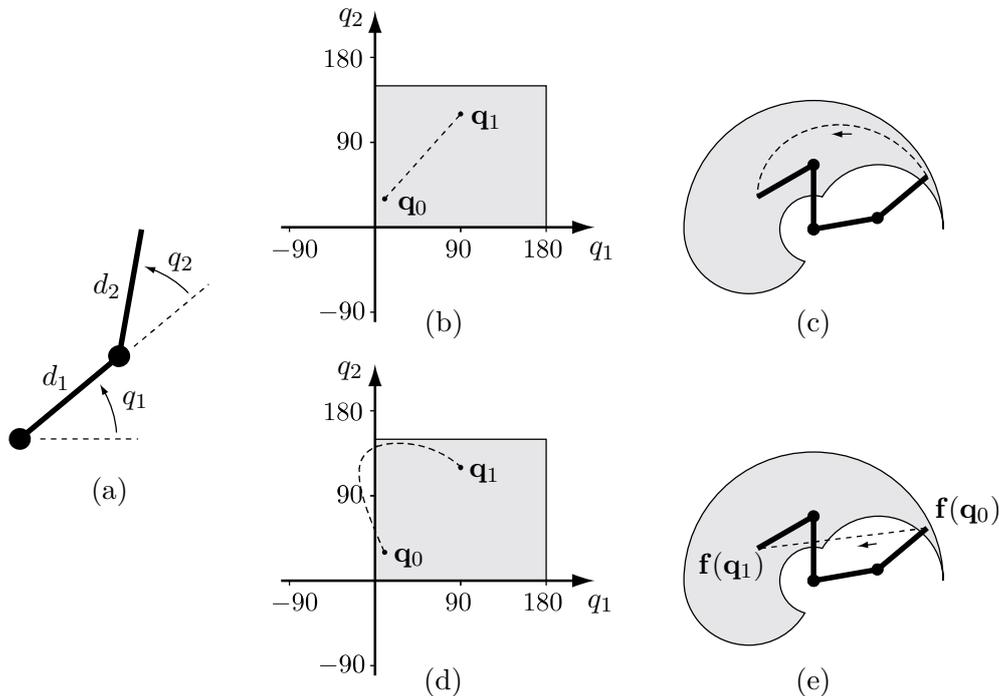


Abbildung 7.3: Geradlinige Pfade eines 2-Achs-Roboters: (a) Roboterkinematik und generalisierte Koordinaten \mathbf{q} , (b) geradliniger Pfad im Konfigurationsraum, (c) zugehöriger kurviger Pfad im Aufgabenraum, (d) kurviger Pfad im Konfigurationsraum, welcher aus einem geradlinigen Pfad im Aufgabenraum resultiert, (e) geradliniger Pfad im Aufgabenraum, welcher die Gelenkwinkelbeschränkungen verletzt [7.4].

7.1.3 Zeitparametrierung

Durch die Zeitparametrierung $s(t)$ eines Pfades $x(s)$ entsteht eine Trajektorie $x(s(t))$. Die Zeitparametrierung $s(t)$ gibt nun den zeitlichen Verlauf vom Anfangszeitpunkt $t = 0$ mit $s(0) = 0$ bis zum Endzeitpunkt $t = T$ mit $s(T) = 1$ vor. Durch die Wahl von $s(t)$ kann sichergestellt werden, dass die Roboterbewegung hinreichend glatt parametrisiert wird und auch die Beschränkungen für die Geschwindigkeit und die Beschleunigung aller Gelenke eingehalten werden.

Mit einer allgemeinen Zeitparametrierung $s(t)$ ergibt sich aus (7.1) für den geradlinigen Pfad (7.2) mit (7.3) die Trajektorie

$$x = x_0 + s(x_1 - x_0) \quad (7.10a)$$

$$\dot{x} = \dot{s}(x_1 - x_0) \quad (7.10b)$$

$$\ddot{x} = \ddot{s}(x_1 - x_0) . \quad (7.10c)$$

Im Folgenden werden zwei unterschiedliche Zeitparametrierungen $s(t)$ vorgestellt. Einerseits wird eine Zeitparametrierung mithilfe von Polynomen 3. und 5. Ordnung konstruiert,

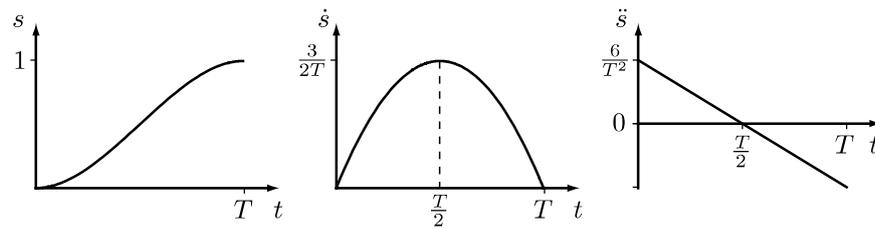


Abbildung 7.4: Schematische Darstellung der Funktionen $s(t)$, $\dot{s}(t)$ und $\ddot{s}(t)$ für die Zeitparametrierung mithilfe eines Polynoms 3. Ordnung [7.4].

mit welchen bestimmte Anfangs- und Endbedingungen erfüllt werden. Andererseits wird eine Bewegung intuitiv durch ein trapezförmiges Geschwindigkeitsprofil bestehend aus einer Beschleunigungsphase, einer Phase konstanter Geschwindigkeit und einer Verzögerungsphase beschrieben.

Polynom 3. Ordnung

Für die Zeitparametrierung mithilfe eines Polynoms 3. Ordnung in der Form

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (7.11a)$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (7.11b)$$

können insgesamt 4 Randbedingungen vorgegeben werden, um die vier Koeffizienten a_0, \dots, a_3 zu bestimmen. Für die Punkt-zu-Punkt-Bewegung entlang eines geradlinigen Pfades wird nun gefordert, dass der Roboter zu Beginn ($t = 0$) bei $s(0) = 0$ im Stillstand $\dot{s}(0) = 0$ startet und nach einer vorgegebenen Zeit $t = T$ bei $s(T) = 1$ wieder zum Stillstand kommt, d. h. $\dot{s}(T) = 0$. Aus diesen Bedingungen folgen die Koeffizienten gemäß

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = \frac{3}{T^2}, \quad a_3 = -\frac{2}{T^3}. \quad (7.12)$$

Die Trajektorie (7.10) zusammen mit (7.11) und (7.12) ist damit vollständig spezifiziert und in Abbildung 7.4 schematisch dargestellt. Aus (7.11b) wird der Zeitpunkt $t = T/2$ bestimmt, bei dem die maximale Geschwindigkeit

$$|\dot{x}_{max}| = \frac{3}{2T} |x_1 - x_0| \quad (7.13)$$

auftritt, vgl. (7.10b). Die maximale Beschleunigung ist

$$|\ddot{x}_{max}| = \frac{6}{T^2} |x_1 - x_0|, \quad (7.14)$$

welche bei $t = 0$ und $t = T$ auftritt. Diese einfachen analytischen Zusammenhänge erlauben es, die Geschwindigkeits- und Beschleunigungsbeschränkungen des Roboters bei einer vorgegebenen Endzeit T zu überprüfen. Umgekehrt kann die schnellstmögliche Roboterbewegung berechnet werden, indem T aus (7.13) bzw. (7.14) berechnet wird.

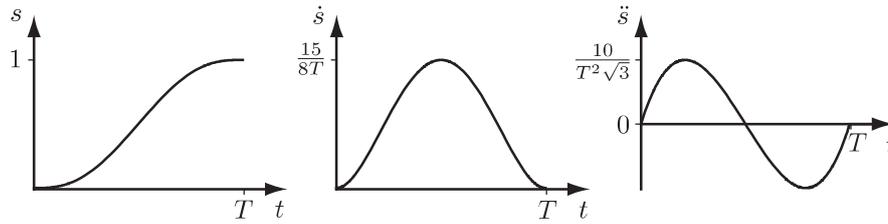


Abbildung 7.5: Schematische Darstellung der Funktionen $s(t)$, $\dot{s}(t)$ und $\ddot{s}(t)$ für die Zeitparametrierung mithilfe eines Polynoms 5. Ordnung [7.4].

Beim Polynom 3. Ordnung ist nur die erste Ableitung \dot{s} stetig, während die zweite Ableitung \ddot{s} bereits bei $t = 0$ und $t = T$ springt. Daher eignet sich dieser Ansatz nur für Systeme, bei denen die Geschwindigkeit der Systemeingang ist, aber nicht für Systeme mit einem Beschleunigungs- bzw. Drehmomenteingang. Weiters haben alle polynomialen Ansätze die Eigenschaft, dass die maximale Geschwindigkeit nur zu *einem* Zeitpunkt erreicht wird und sie eine lange Beschleunigungs- und Verzögerungsphase aufweisen. Damit wird die Leistungsfähigkeit des Robotersystems nicht optimal ausgenützt.

Polynom 5. Ordnung

Wird die Zeitparametrierung als Polynom 5. Ordnung in der Form

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (7.15a)$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (7.15b)$$

$$\ddot{s}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad (7.15c)$$

gewählt, so können im Vergleich zum Polynom 3. Ordnung noch zusätzlich die zweiten Ableitungen am Anfang und am Ende der Roboterbewegung mit $\ddot{s}(0) = \ddot{s}(T) = 0$ vorgegeben werden. D. h. es wird gefordert, dass die zweite Ableitung zu Beginn und am Ende stetig ist. Mit diesen Bedingungen werden die Koeffizienten a_0, \dots, a_5 des Polynoms (7.15a) gemäß

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = \frac{10}{T^3}, \quad a_4 = -\frac{15}{T^4}, \quad a_5 = \frac{6}{T^5} \quad (7.16)$$

berechnet. In der schematischen Darstellung in Abbildung 7.5 ist ersichtlich, dass die zweite Ableitung $\ddot{s}(t)$ nun ebenfalls mit Null beginnt und endet. Damit ist diese Zeitparametrierung geeignet, um Beschleunigungen bzw. Drehmomente für die Bewegungen eines Robotersystems vorzugeben.

Trapezförmige Bewegungsprofile

Bei den trapezförmigen Bewegungsprofilen weist die erste Ableitung der Zeitparametrierung $\dot{s}(t)$ oder die Geschwindigkeit der Trajektorie $\frac{d}{dt}x(s(t))$ eines geradlinigen Pfades $x(s)$ einen trapezförmigen Verlauf auf, wie Abbildung 7.6 zeigt. Trapezförmige Bewegungsprofile bestehen aus drei Phasen, nämlich einer Beschleunigungsphase ①, einer Phase konstanter

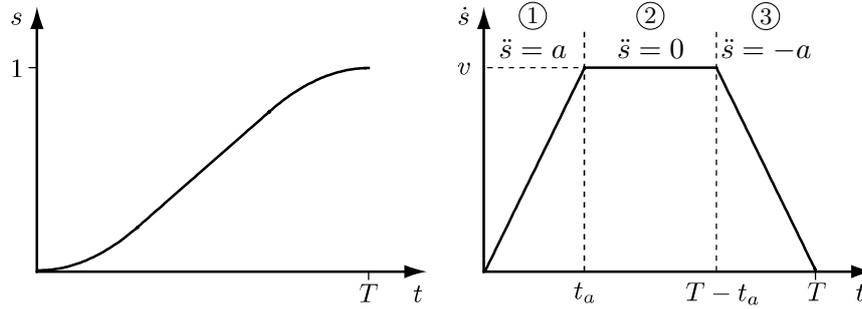


Abbildung 7.6: Zeitparametrierung mit trapezförmigen Bewegungsprofil: ① Beschleunigungsphase, ② Phase konstanter Geschwindigkeit v , ③ Verzögerungsphase [7.4].

Geschwindigkeit ② und einer Verzögerungsphase ③). Das zugehörige Beschleunigungsprofil mit dem konstanten Beschleunigungsparameter a lautet daher

$$\ddot{s}(t) = \begin{cases} a & 0 \leq t < t_a \\ 0 & t_a \leq t < T - t_a \\ -a & T - t_a \leq t \leq T \\ 0 & \text{sonst} \end{cases} \quad (7.17)$$

und durch einmalige Integration folgt das Geschwindigkeitsprofil zu

$$\dot{s}(t) = \begin{cases} at & 0 \leq t < t_a \\ v & t_a \leq t < T - t_a \\ a(T - t) & T - t_a \leq t \leq T \\ 0 & \text{sonst} , \end{cases} \quad (7.18)$$

wobei in (7.17) und (7.18) die Geschwindigkeit $v = at_a$ und die Beschleunigungs- und Verzögerungszeit t_a eingeführt wurde. Nochmalige Integration liefert die Zeitparametrierung $s(t)$ in der Form

$$s(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2}at^2 & 0 \leq t < t_a \\ vt - \frac{v^2}{2a} & t_a \leq t < T - t_a \\ vT - \frac{v^2}{a} - \frac{a}{2}(T - t)^2 & T - t_a \leq t \leq T \\ vT - \frac{v^2}{a} & \text{sonst} . \end{cases} \quad (7.19)$$

Das trapezförmige Bewegungsprofil bietet den Vorteil, dass die Geschwindigkeits- und Beschleunigungsbeschränkungen des Roboters optimal ausgenutzt werden können. Aus (7.17) und (7.18), eingesetzt in (7.10), lassen sich die Parameter v und a aus der maximalen Geschwindigkeit \dot{x}_{max} und der maximalen Beschleunigung \ddot{x}_{max} in der Koordinate x in der Form

$$|v(x_1 - x_0)| \leq \dot{x}_{max} \quad (7.20)$$

$$|a(x_1 - x_0)| \leq \ddot{x}_{max} \quad (7.21)$$

bestimmen.

Die geradlinige Trajektorie (7.10) mit dem trapezförmigen Bewegungsprofil (7.17)–(7.19) wird durch die drei Variablen v , a und T parametrisiert. Es können aber nur zwei davon unabhängig gewählt werden, denn es wird die Bedingungen $s(T) = 1$ gefordert und die Beschleunigungs- und Verzögerungszeit t_a steht über $t_a = v/a$ im Zusammenhang. Werden beispielsweise die Geschwindigkeit v und die Beschleunigung a vorgegeben – unter der Bedingung $v^2/a \leq 1$, sodass alle drei Phasen des Bewegungsprofils vorhanden sind – so ergibt sich aus der Bedingung $s(T) = 1$ die erforderliche Zeitdauer T für die Trajektorie zu

$$T = \frac{1}{v} + \frac{v}{a} . \quad (7.22)$$

Weiters kann auch durch Vorgabe von v und T die erforderliche Beschleunigung a bestimmt werden und wenn a und T vorgegeben wird, kann die zugehörige Geschwindigkeit v berechnet werden.

Das trapezförmige Bewegungsprofil mit drei Phasen weist allerdings den gleichen Nachteil auf wie die Zeitparametrierung mittels eines Polynoms 3. Ordnung, nämlich die Unstetigkeit der Beschleunigung. Um Vibrationen und Schwingungen im Robotersystem zu minimieren, wird meist die Änderungsrate der Beschleunigung, der sogenannte *Ruck* (engl. *jerk*), beschränkt. Analog zur Zeitparametrierung mit einem Polynom 5. Ordnung (siehe Abbildung 7.5) kann auch ein trapezförmiges Bewegungsprofil mit sieben Phasen konstruiert werden, welches diese Aufgabe erfüllt, siehe Abbildung 7.7. Anstatt sprunghaft die maximale Beschleunigung a aufzuschalten, wird die Beschleunigung linear mit beschränktem Ruck hoch- und heruntergefahren.

Zeitoptimale Parametrierung

Eine Roboterbewegung kann im Allgemeinen in die räumliche Bewegung im Konfigurationsraum $\mathbf{q}(s)$ mit einer Zeitparametrierung des Pfadparameters $s(t)$ zerlegt werden, siehe Abschnitt 7.1.1. Eine optimale Roboterbewegung kann nur gefunden werden, wenn beide Aspekte gleichzeitig betrachtet werden. Wenn allerdings die räumliche Bewegung $\mathbf{q}(s)$ durch die Anwendung bereits festgelegt ist, so reduziert sich das Bewegungsplanungsproblem auf das Finden der Zeitparametrierung $s(t)$. Besonders wichtig und häufig angewendet wird dabei die zeitoptimale Parametrierung, welche in diesem Abschnitt diskutiert wird.

Zustandsabhängige Beschränkungen: Roboter sind mechanische Systeme, welche durch eine Differentialgleichung zweiter Ordnung gemäß (5.1) beschrieben werden. Reale Robotersysteme weisen Beschränkungen bei den Gelenken (Position, Geschwindigkeit) und deren Antriebe (Drehmoment, Änderungsrate des Drehmoments) auf. Im Allgemeinen sind die Geschwindigkeitsbeschränkungen für die Gelenksgeschwindigkeit $\dot{\mathbf{q}}$ aber nicht konstant und auch konstante Beschränkungen bei der Beschleunigung $\ddot{\mathbf{q}}$ oder dem Ruck $\dddot{\mathbf{q}}$ lassen sich nicht angeben. Diese Beschränkungen sind folglich zustandsabhängig und werden bezüglich des Systemeingangs $\boldsymbol{\tau}^T = [\tau_1 \ \tau_2 \ \dots \ \tau_n]$ in der Form

$$\tau_i^{\min}(\mathbf{q}, \dot{\mathbf{q}}) \leq \tau_i \leq \tau_i^{\max}(\mathbf{q}, \dot{\mathbf{q}}) \quad (7.23)$$

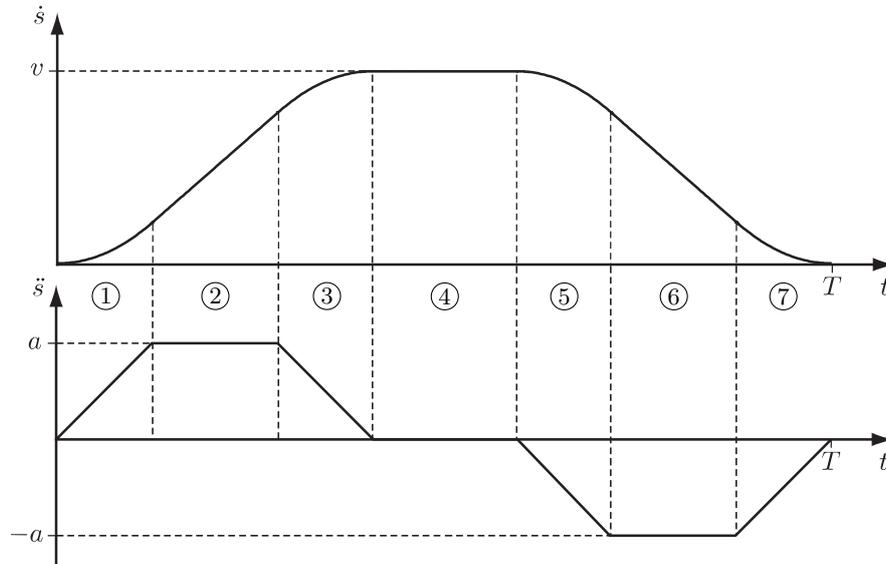


Abbildung 7.7: Zeitparametrierung für trapezförmiges Bewegungsprofil mit 7 Phasen:

- ① Hochfahren der Beschleunigung, ② konstante Beschleunigung a , ③ herunterfahren der Beschleunigung, ④ konstante Geschwindigkeit v , ⑤ hochfahren der Verzögerung, ⑥ konstante Verzögerung $-a$, ⑦ herunterfahren der Verzögerung [7.4].

mit der unteren Beschränkung τ_i^{\min} und der oberen Beschränkung τ_i^{\max} , $i = 1, \dots, n$, angegeben. Beispielsweise fällt das maximale Drehmoment eines DC-Motors linear mit der Motordrehzahl ab, siehe Abb. 2.23 in Abschnitt 2.3.3.

Pfadabhängige Dynamikgleichung und Beschränkungen: Im nächsten Schritt werden die Dynamikgleichung (5.1) und die Beschränkungen (7.23) als Funktion des Pfadparameters s dargestellt. Zunächst lässt sich zeigen, dass die Coriolis-Terme quadratisch in der Geschwindigkeit $\dot{\mathbf{q}}$ der generalisierten Koordinaten \mathbf{q} sind [7.4] und somit

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \dot{\mathbf{q}}^T \tilde{\Gamma}(\mathbf{q})\dot{\mathbf{q}} \quad (7.24)$$

gilt, wobei $\tilde{\Gamma}(\mathbf{q})$ ein dreidimensionaler Tensor der Christoffel-Symbole (Abschnitt 5.2.2) ist. Das Produkt auf der rechten Seite von (7.24) wird als Vektor-Matrix-Produkt gemäß

$$\dot{\mathbf{q}}^T \tilde{\Gamma}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}^T \Gamma_1(\mathbf{q})\dot{\mathbf{q}} \\ \dot{\mathbf{q}}^T \Gamma_2(\mathbf{q})\dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \Gamma_n(\mathbf{q})\dot{\mathbf{q}} \end{bmatrix} \quad (7.25)$$

dargestellt, wobei die $n \times n$ -Matrizen $\Gamma_i(\mathbf{q})$ mit den Christoffel-Symbolen $\Gamma_{ijk}(\mathbf{q})$ in den Matrixeinträgen (j, k) , $i, j, k \in \{1, \dots, n\}$, besetzt sind, siehe (5.33). Unter Verwendung von (7.24) und

$$\dot{\mathbf{q}} \rightarrow \frac{d\mathbf{q}}{ds} \dot{s} \quad (7.26)$$

$$\ddot{\mathbf{q}} \rightarrow \frac{d\mathbf{q}}{ds}\ddot{s} + \frac{d^2\mathbf{q}}{ds^2}\dot{s}^2 \quad (7.27)$$

kann die Dynamikgleichung (5.1) als Funktion des Pfadparameters s folglich als

$$\underbrace{\left(\mathbf{M}(\mathbf{q}(s))\frac{d\mathbf{q}}{ds}\right)}_{\mathbf{m}(s)}\ddot{s} + \underbrace{\left(\mathbf{M}(\mathbf{q}(s))\frac{d^2\mathbf{q}}{ds^2} + \left(\frac{d\mathbf{q}}{ds}\right)^T \tilde{\mathbf{I}}(\mathbf{q}(s))\frac{d\mathbf{q}}{ds}\right)}_{\mathbf{c}(s)}\dot{s}^2 + \underbrace{\mathbf{g}(\mathbf{q}(s))}_{\mathbf{g}(s)} = \boldsymbol{\tau} \quad (7.28)$$

ausgedrückt und kompakt als

$$\mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s) = \boldsymbol{\tau} \quad (7.29)$$

angeschrieben werden. Die Gleichung (7.29) stellt somit die Dynamik des Roboters dar, wenn dieser entlang des Pfades $\mathbf{q}(s)$ geführt wird, wobei $\mathbf{m}(s) \in \mathbb{R}^n$ die effektive Trägheit, $\mathbf{c}(s) \in \mathbb{R}^n$ die quadratischen Geschwindigkeitsterme und $\mathbf{g}(s) \in \mathbb{R}^n$ die Gravitationsterme bezeichnet. Mithilfe von (7.29) werden die Beschränkungen (7.23) elementweise für alle generalisierten Koordinaten mit $\tau_i = m_i(s)\ddot{s} + c_i(s)\dot{s}^2 + g_i(s)$, $i \in \{1, \dots, n\}$, als Funktion des Pfadparameters in der Form

$$\tau_i^{\min}(s, \dot{s}) \leq m_i(s)\ddot{s} + c_i(s)\dot{s}^2 + g_i(s) \leq \tau_i^{\max}(s, \dot{s}) \quad (7.30)$$

angeschrieben.

Abhängig vom Vorzeichen der effektiven Trägheit $m_i(s)$ können aus (7.30) die Beschränkungsfunktionen $U_i(s, \dot{s})$ für die maximale (engl. *upper*) Beschleunigung und $L_i(s, \dot{s})$ für die minimale (engl. *lower*) Beschleunigung (d. h. die maximale Verzögerung) abgeleitet werden. Diese lauten

$$L_i(s, \dot{s}) = \begin{cases} \frac{\tau_i^{\min} - c_i(s)\dot{s}^2 - g_i(s)}{m_i(s)} & \text{für } m_i(s) > 0 \\ \frac{\tau_i^{\max} - c_i(s)\dot{s}^2 - g_i(s)}{m_i(s)} & \text{für } m_i(s) < 0 \\ \text{Null-Trägheits-Punkt} & \text{für } m_i(s) = 0, \end{cases} \quad (7.31a)$$

$$U_i(s, \dot{s}) = \begin{cases} \frac{\tau_i^{\max} - c_i(s)\dot{s}^2 - g_i(s)}{m_i(s)} & \text{für } m_i(s) > 0 \\ \frac{\tau_i^{\min} - c_i(s)\dot{s}^2 - g_i(s)}{m_i(s)} & \text{für } m_i(s) < 0 \\ \text{Null-Trägheits-Punkt} & \text{für } m_i(s) = 0. \end{cases} \quad (7.31b)$$

Hier ist zu beachten, dass die Beschränkung (7.30) bei *Null-Trägheits-Punkten* für einzelne Achsen unabhängig von der Beschleunigung \ddot{s} wird. An diesen Punkten ist stattdessen die Geschwindigkeit \dot{s} beschränkt.

Die einzelnen Beschränkungen (7.31) werden über das Maximum von $L_i(s, \dot{s})$ bzw. das Minimum von $U_i(s, \dot{s})$, $i = 1, \dots, n$, gemäß

$$L(s, \dot{s}) = \max_i L_i(s, \dot{s}) \quad (7.32a)$$

$$U(s, \dot{s}) = \min_i U_i(s, \dot{s}) \quad (7.32b)$$

zusammengefasst und damit folgt die zustandsabhängige Beschränkung gemäß

$$L(s, \dot{s}) \leq \ddot{s} \leq U(s, \dot{s}). \quad (7.33)$$

Für die weitere Planung darf also nur eine Beschleunigung \ddot{s} gewählt werden, die die Beschränkungen aller Antriebe *gleichzeitig* erfüllt.

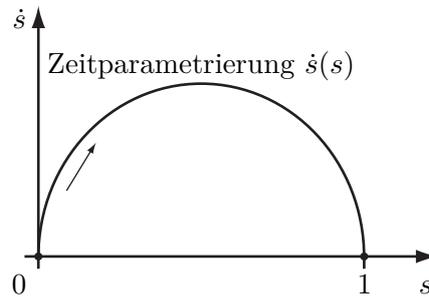


Abbildung 7.8: Die Zeitparametrierung $\dot{s}(s)$ in der (s, \dot{s}) -Ebene ist eine Kurve mit $\dot{s} \geq 0$, welche den Anfangszustand $(0, 0)$ mit dem Endzustand $(1, 0)$ verbindet [7.4].

Definition 7.1 (Zeitoptimale Parametrierung). Es sei ein Pfad $\mathbf{q}(s)$, $s \in [0, 1]$, der Anfangszustand $(s_0, \dot{s}_0) = (0, 0)$ und der Zielzustand $(s_1, \dot{s}_1) = (1, 0)$ gegeben. Die zeitoptimale Parametrierung $s : [0, T] \rightarrow [0, 1]$ ist eine monoton steigende, zweifach differenzierbare Funktion, welche

- (a) $s(0) = 0$, $\dot{s}(0) = 0$, $s(T) = 1$ und $\dot{s}(T) = 0$ erfüllt,
- (b) die Beschränkungen (7.33) erfüllt und
- (c) die gesamte Dauer der Roboterbewegung T minimiert.

Phasendiagramm: Das Planungsproblem aus Definition 7.1 ist als Phasendiagramm in der (s, \dot{s}) -Ebene in Abb. 7.8 dargestellt. Dabei verläuft s von 0 bis 1 und ist monoton steigend, weshalb $\dot{s}(t) \geq 0$ gilt. Eine gültige Zeitparametrierung $s(t)$ verläuft somit im Zustandsraum (s, \dot{s}) von $(0, 0)$ bis $(1, 0)$. Es erfüllen jedoch nicht alle Kurven die Beschränkungen der Beschleunigung (7.33).

Wie Abb. 7.9a zeigt, kann die Beschränkung (7.33) an jedem Punkt in der (s, \dot{s}) -Ebene als *Öffnungswinkel* eingezeichnet werden, welcher aus den Änderungsraten zufolge $L(s, \dot{s})$ und $U(s, \dot{s})$ und \dot{s} bestimmt wird. Für alle Punkte mit $L(s, \dot{s}) \geq U(s, \dot{s})$ verschwindet der Öffnungswinkel und es handelt sich um unzulässige Punkte, die der Roboter nicht realisieren kann. Diese sind als graue Flächen in Abb. 7.9a eingezeichnet. Bei Kurven mit $L(s, \dot{s}) = U(s, \dot{s})$ reduziert sich der Öffnungswinkel zu Null und es bleibt der Tangentialvektor als einzig zulässige Richtung. Diese Kurven werden im Weiteren als *Grenzgeschwindigkeitskurven* $\dot{s}_{\text{lim}}(s)$ bezeichnet. Bei der zeitoptimalen Parametrierung muss die Geschwindigkeit \dot{s} größtmöglich sein¹, aber innerhalb des Öffnungswinkels liegen. Ein Beispiel für eine ungültige Zeitparametrierung zeigt Abb. 7.9b, worin beim Verzögern

¹Mit der Substitution $ds/ds = 1$ wird die Gleichung für die Bewegungsdauer T gemäß

$$T = \int_0^T 1 dt = \int_0^T \frac{ds}{ds} dt = \int_0^1 \frac{dt}{ds} ds = \int_0^1 \dot{s}^{-1}(s) ds$$

umgeformt, wobei die Integrationsgrenzen durch die Änderung der Integrationsvariable von 0 bis T (Variable t) auf 0 bis 1 (Variable s) geändert wurden. Eine minimale Bewegungsdauer bedeutet somit eine maximale Geschwindigkeit des Pfadparameters \dot{s} unter der Beschränkung (7.33).

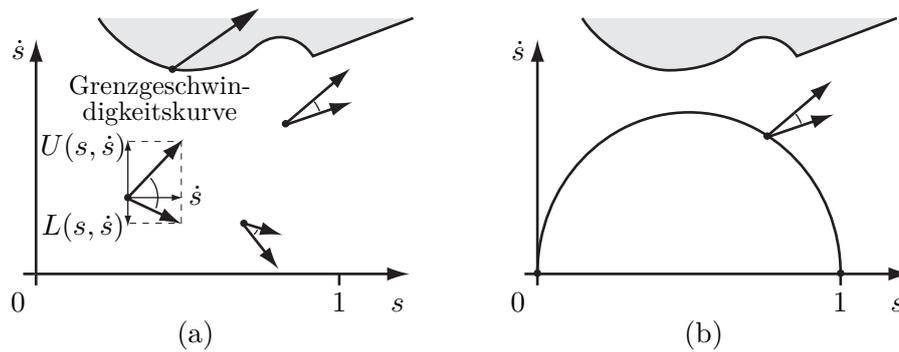


Abbildung 7.9: Punkte in der (s, \dot{s}) -Ebene [7.4]: (a) Beschleunigungsbeschränkungen als Öffnungswinkel an vier unterschiedlichen Punkten. Dieser Öffnungswinkel setzt sich aus vertikalen Pfeilen (Beschleunigung \ddot{s} , Änderung der Geschwindigkeit \dot{s}) und horizontalen Pfeilen (Geschwindigkeit \dot{s} , Änderung des Pfadparameters s) zusammen. (b) Ungültige Zeitparametrierung, da die gezeichnete Kurve nicht im Öffnungswinkel enthalten ist.

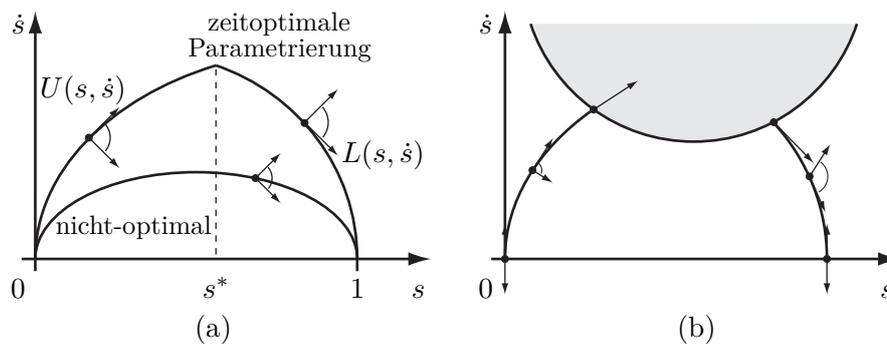


Abbildung 7.10: Zeitoptimale Bang-Bang-Steuerung (a) ohne und (b) mit Beschränkungen [7.4].

stärker abgebremst wird als es die Aktoren zulassen, d. h. (7.33) ist für den eingezeichneten Punkt verletzt.

Für eine zeitoptimale Parametrierung muss die Beschleunigung \ddot{s} stets entweder im unteren Limit $L(s, \dot{s})$ oder oberen Limit $U(s, \dot{s})$ gewählt werden, d. h. es wird eine sogenannte *Bang-Bang-Steuerung* gewählt, siehe auch [7.5]. Wenn keine Beschränkungen erreicht werden (siehe Abb. 7.10a) wird dabei zunächst die maximale Beschleunigung $U(s, \dot{s})$ aufgeschaltet bis zu einem optimalen Umschaltzeitpunkt s^* und danach auf die maximale Verzögerung $L(s, \dot{s})$ umgeschaltet. Dieser Umschaltzeitpunkt s^* wird als Schnittpunkt zwischen der Kurve der Vorwärtsintegration von $U(s, \dot{s})$ beginnend bei $(0, 0)$ und der Kurve der Rückwärtsintegration von $L(s, \dot{s})$ beginnend bei $(1, 0)$ gefunden. Verhindern allerdings Beschränkungen diesen einfachen Fall (graue Flächen in Abb. 7.10b), so sind mehrere Umschaltzeitpunkte notwendig, welche durch folgenden Algorithmus bestimmt werden.

Algorithmus:

1. Initialisiere die Liste der Umschaltunkte $\mathcal{S} = \{\}$ und den Zähler $i = 0$. Setze den Anfangszustand $(s_i, \dot{s}_i) = (0, 0)$.
2. Rückwärtsintegrieren der Gleichung $\ddot{s} = L(s, \dot{s})$ nach t vom Endzustand $(1, 0)$ bis eine Grenzgeschwindigkeitskurve erreicht wird ($L(s, \dot{s}) > U(s, \dot{s})$) oder $s = 0$ erreicht wird. Diese Kurve wird mit F bezeichnet.
3. Vorwärtsintegrieren der Gleichung $\ddot{s} = U(s, \dot{s})$ nach t vom Zustand (s_i, \dot{s}_i) bis ein Schnittpunkt mit F gefunden wird oder $U(s, \dot{s}) < L(s, \dot{s})$ (Grenzgeschwindigkeitskurve) erreicht wird. Diese Kurve wird A_i genannt. Hier werden zwei Fälle unterschieden: Einerseits, wenn A_i die Kurve F schneidet, dann ist das Planungsproblem gelöst. Erhöhe den Zähler i , setze (s_i, \dot{s}_i) auf die Werte von (s, \dot{s}) des Schnittpunktes und füge s_i zur Liste der Umschaltunkte \mathcal{S} hinzu. Bei diesem Umschaltpunkt wird von maximaler Beschleunigung auf maximale Verzögerung umgeschaltet. Andererseits, wenn eine Grenzgeschwindigkeitskurve erreicht wird, gibt $(s_{\text{lim}}, \dot{s}_{\text{lim}})$ den Schnittpunkt an.
4. Suche jene Geschwindigkeit $\dot{s}' \in [0, \dot{s}_{\text{lim}}]$, sodass die Vorwärtsintegration von $\ddot{s} = L(s, \dot{s})$ beginnend bei $(s_{\text{lim}}, \dot{s}')$ die Grenzgeschwindigkeitskurve genau tangential berührt. Dieser Punkt wird mit $(s_{\text{tan}}, \dot{s}_{\text{tan}})$ bezeichnet. An dieser Stelle ist $L(s, \dot{s}) = U(s, \dot{s})$ und der Öffnungswinkel ist Null. Diese Suche wird sehr effizient mit einem Bisektionsverfahren [7.5] ausgeführt.
5. Integriere $\ddot{s} = L(s, \dot{s})$ rückwärts von $(s_{\text{tan}}, \dot{s}_{\text{tan}})$ bis der Schnittpunkt mit A_i gefunden wird. Erhöhe i und setze (s_i, \dot{s}_i) auf die Werte (s, \dot{s}) des Schnittpunktes. Das Kurvensegment von (s_i, \dot{s}_i) bis $(s_{\text{tan}}, \dot{s}_{\text{tan}})$ wird mit A_i bezeichnet. Fügen von s_i zu \mathcal{S} hinzu. Bei diesem Umschaltpunkt wird von maximaler Beschleunigung auf maximale Verzögerung umgeschaltet.
6. Erhöhe i und setze (s_i, \dot{s}_i) auf $(s_{\text{tan}}, \dot{s}_{\text{tan}})$. Fügen von s_i zu \mathcal{S} hinzu. An diese Stelle wird von maximaler Verzögerung auf maximale Beschleunigung umgeschaltet. Gehe zu Schritt 3.

Beispiel 7.2 (Zeitoptimale Parametrierung mit Beschränkungen). Der obige Algorithmus wird für ein konkretes Beispiel einer Trajektorienplanung mit Beschränkungen diskutiert, siehe Abb. 7.11. Die Beschränkungen der Dynamik des Robotersystems (graue Flächen) verhindern den einfachen Fall und eine Zeitparametrierung mit mehreren Umschaltzeitpunkten ist notwendig:

- Abbildung 7.11a: Schritt 2, Rückwärtsintegration von $\ddot{s} = L(s, \dot{s})$ beginnend bei $(1, 0)$ bis die Grenzgeschwindigkeitskurve erreicht ist.
- Abbildung 7.11b: Schritt 3, Vorwärtsintegration von $\ddot{s} = U(s, \dot{s})$ beginnend bei $(0, 0)$ bis der Schnittpunkt $(s_{\text{lim}}, \dot{s}_{\text{lim}})$ an der Grenzgeschwindigkeitskurve gefunden wird.
- Abbildung 7.11c: Schritt 4, Finden von $\dot{s}' \in [0, \dot{s}_{\text{lim}}]$, sodass die Vorwärtsintegration von $\ddot{s} = L(s, \dot{s})$ beginnend bei $(s_{\text{lim}}, \dot{s}')$ die Grenzgeschwindigkeitskurve tangential berührt.

- Abbildung 7.11d: Schritt 5, Rückwärtsintegration von $(s_{\text{tan}}, \dot{s}_{\text{tan}})$, um den ersten Umschaltunkt s_1 zu finden.
- Abbildung 7.11e: Schritt 6, Festlegen des zweiten Umschaltpunktes bei $(s_2, \dot{s}_2) = (s_{\text{tan}}, \dot{s}_{\text{tan}})$.
- Abbildung 7.11f: Schritt 3, Vorwärtsintegration von $\ddot{s} = U(s, \dot{s})$ beginnend bei (s_2, \dot{s}_2) . Dies führt zum Schnittpunkt mit der Kurve F bei (s_3, \dot{s}_3) .

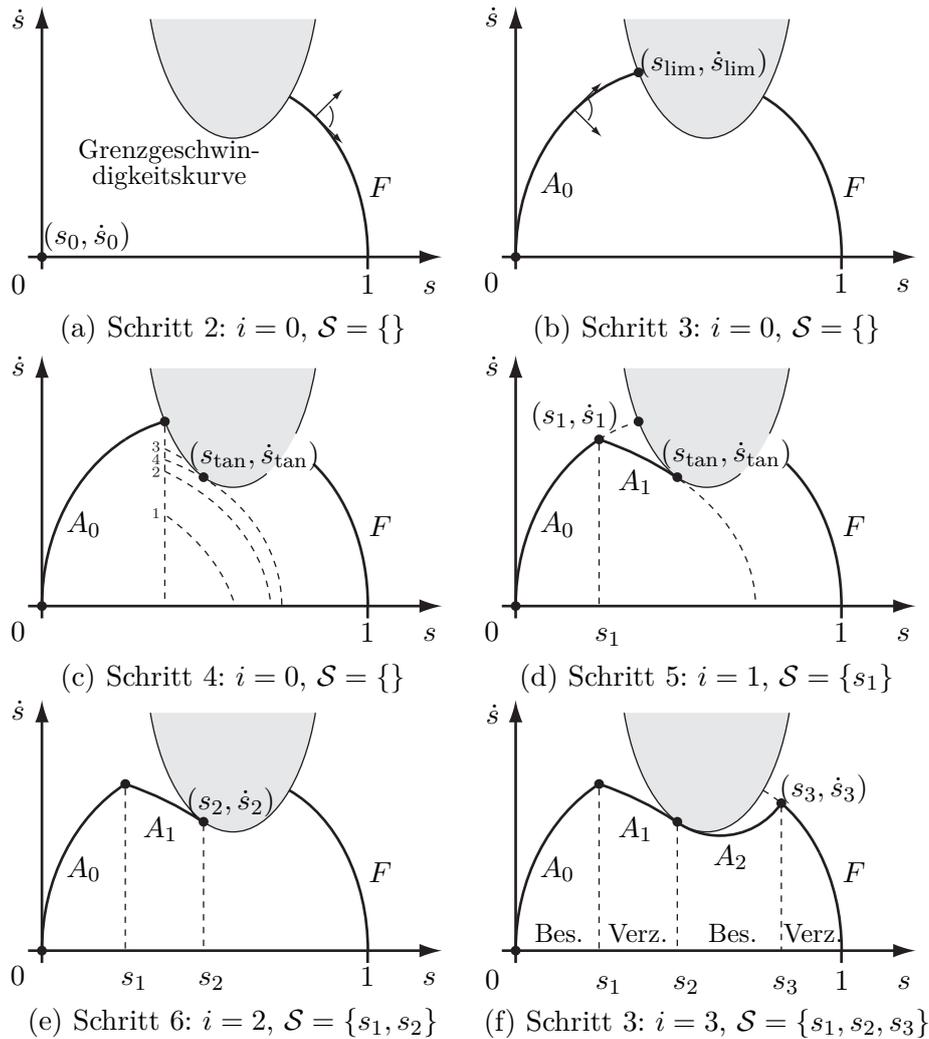


Abbildung 7.11: Beispiel für eine zeitoptimale Parametrierung mit Beschränkungen [7.4].

Die optimalen Umschaltpunkte sind somit $\mathcal{S} = \{s_1, s_2, s_3\}$ und haben folgende Bedeutung, siehe Abb. 7.11f:

- $0 \leq s < s_1$: Maximale Beschleunigung $U(s, \dot{s})$,

- $s_1 \leq s < s_2$: Maximale Verzögerung $L(s, \dot{s})$,
- $s_2 \leq s < s_3$: Maximale Beschleunigung $U(s, \dot{s})$,
- $s_3 \leq s \leq 1$: Maximale Verzögerung $L(s, \dot{s})$,

7.2 Bewegungsplanung

In diesem Abschnitt bezeichnet der Begriff „Bewegungsplanung“ die Aufgabe, eine Roboterbewegung zu finden, welche diesen von einem Anfangszustand in einen Zielzustand unter Berücksichtigung gegebener Randbedingungen und Hindernisse überführt. Die im vorigen Abschnitt gezeigten, geradlinigen Pfade sind häufig nicht ausreichend, um diese Aufgabe zu lösen, vgl. Beispiel 7.1. Daher werden in diesem Abschnitt allgemeinere Methoden vorgestellt, welche eine Vielzahl unterschiedlicher Randbedingungen berücksichtigen können, wie z. B. Hindernisse in der Umgebung oder Beschränkungen bei den Gelenkpositionen, -geschwindigkeiten oder Antriebsmomenten und -kräften.

Bewegungsplanung ist ein sehr aktives Forschungsfeld in der Robotik und entsprechend gibt es eine Vielzahl an unterschiedlichen Methoden, um ein Planungsproblem zu lösen. Es gibt aber nicht „den einen“ Algorithmus, welcher alle Aufgaben lösen kann, weil die Anforderungen an den Algorithmus je nach Anwendungsfall ganz unterschiedlich sein können und sich auch häufig gegenseitig ausschließen.

Im Folgenden werden zunächst die allgemeinen Grundlagen der Bewegungsplanung eingeführt und danach eine Definition für die Bewegungsplanung gegeben. Diese Definition wird in Bezug auf Kategorien und Eigenschaften von Planungsalgorithmen diskutiert. Im Anschluss werden die wichtigsten Planungsmethoden im Detail erklärt.

7.2.1 Grundlagen

In diesem Abschnitt werden elementare Bestandteile, wie sie in den meisten Algorithmen für die Bewegungsplanung enthalten sind, diskutiert.

Hindernisse

Gemäß der Definition 1.1 wird der gesamte Konfigurationsraum als „ C -Space“, kurz C bezeichnet, wobei jede mögliche Roboterkonfiguration $\mathbf{q} \in C \subseteq \mathbb{R}^n$ einen Punkt in diesem Raum darstellt. Dieser Konfigurationsraum wird weiter unterteilt in einen *Freiraum* C_f (engl. *free space*) und einen *Hindernisraum* C_o (engl. *obstacle space*) und es gilt

$$C = C_f \cup C_o . \quad (7.34)$$

Im Hindernisraum C_o überlappt mindestens ein Körper des Roboters geometrisch mit einem Hindernis. Im Freiraum C_f hingegen sind keine Kollisionen vorhanden. Das exakte Abbilden der geometrischen Hindernisse im Hindernisraum kann mitunter eine sehr komplexe, rechenintensive Aufgabe sein, weshalb Hindernisse meist durch vereinfachte Geometrien und konvexe Hüllen approximiert werden. Wenn $C = C_f$ gilt, so wird bei der Planung von einem hindernisfreien Planungsraum ausgegangen.

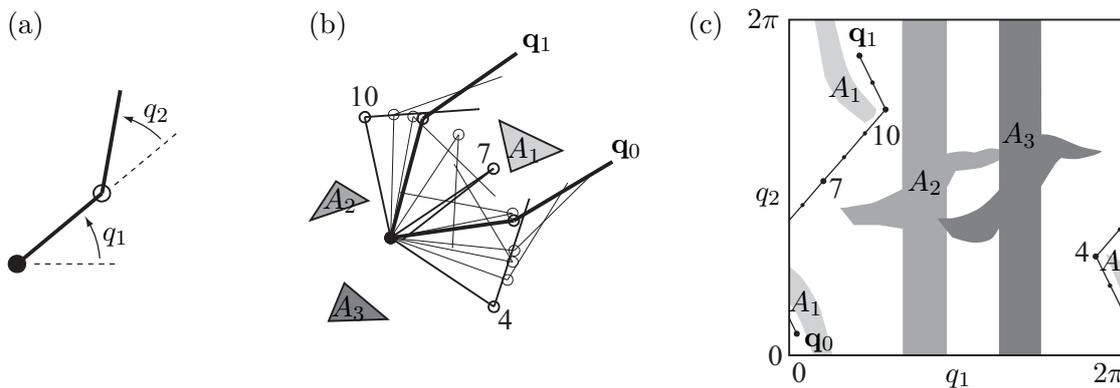


Abbildung 7.12: Bewegungsplanung für einen 2-Achs-Roboter: (a) Schematische Darstellung des 2-Achs-Roboters, (b) Bewegung von der Startkonfiguration \mathbf{q}_0 zur Zielkonfiguration \mathbf{q}_1 zwischen den Hindernissen A_1 , A_2 und A_3 , (c) Konfigurationsraum des Roboters mit den übertragenen Hindernissen sowie der Lösung der Planungsaufgabe [7.4].

Ob ein Roboter in seiner aktuellen Konfiguration \mathbf{q} mit einem Hindernis kollidiert, kann anhand der CAD-Modelle des Roboters und der Umgebung – abhängig vom Detailgrad und der Anzahl an Objekten – mit relativ hohem Aufwand geprüft werden. Um die Bewegungsplanung im Konfigurationsraum ausführen zu können, müssen alle relevanten Objekte und Hindernisse in diesen Raum übertragen werden und darin der Hindernisraum C_o bestimmt werden. Dabei werden Beschränkungen in den Gelenkpositionen direkt als Hindernisse im Konfigurationsraum abgebildet. Dadurch reduziert sich das Planungsproblem auf eine Suche nach einem durchgängigen Pfad innerhalb des Freiraumes C_f .

Als anschauliches Beispiel dient der 2-Achs-Roboter in Abbildung 7.12a, welcher sich im Konfigurationsraum $\mathbf{q} = [q_1 \ q_2]^T \in C = \mathbb{R}^2$ mit unendlich drehenden Achsen q_1 und q_2 bewegt. Der Roboter bewegt sich nun in der Ebene von der Startkonfiguration \mathbf{q}_0 zur Zielkonfiguration \mathbf{q}_1 und muss dabei den Hindernissen A_1 , A_2 und A_3 ausweichen. Dabei ist der Konfigurationsraum in 7.12c als Quadrat dargestellt. Die gegenüberliegenden Kanten sind miteinander verbunden, da der Konfigurationsraum C topologisch ein Torus T^2 ist, vgl. Tabelle 1.2. Weiters wurden die Hindernisse A_1 , A_2 und A_3 in den Konfigurationsraum übertragen. Es ist ersichtlich, dass in jeder Konfiguration, die von einem Hindernis belegt ist, mindestens ein Teil des Roboters mit diesem Objekt kollidieren würde. Durch die Hindernisse wird der Freiraum C_f auch in mehrere disjunkte Teile getrennt: Der Roboter kann nicht von $q_1 = 0$ auf $q_1 = \pi$ fahren, ohne dabei mit dem Hindernis A_2 oder A_3 zu kollidieren.

Als zweites Beispiel wird der kreisförmige, mobile Roboter mit einem dreieckigen Hindernis aus Abbildung 7.13a betrachtet. Der Konfigurationsraum C des Roboters sei die reine Translation in der Ebene, d. h. $\mathbf{q} = [q_1 \ q_2]^T \in C = \mathbb{R}^2$. Das dreieckige Hindernis wird nun in den Konfigurationsraum des Roboters übertragen, indem die Außenkontur um den Radius des Roboters erweitert wird, siehe Abbildung 7.13b. Jeder Punkt außerhalb dieser Kontur gehört zum Freiraum C_f des Roboters.

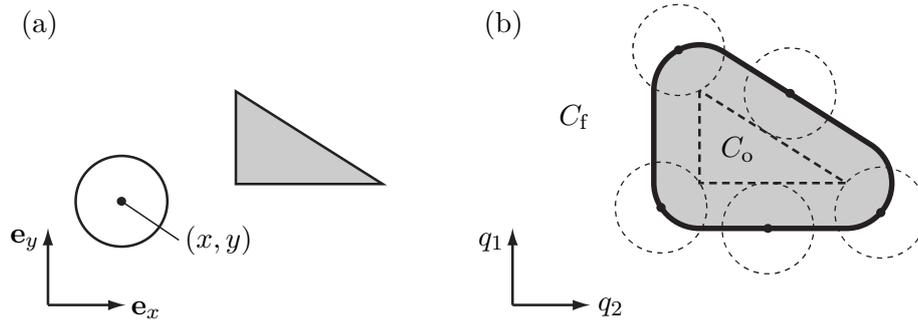


Abbildung 7.13: Bewegungsplanung für einen kreisförmigen, mobilen Roboter: (a) Roboter mit einem dreieckigen Hindernis, (b) Hindernis in den Konfigurationsraum C übertragen [7.4].

Abstand zu Hindernissen

Für die Bewegungsplanung in einem mit Hindernissen besetzten Raum wird insbesondere eine Funktion bzw. ein Algorithmus für die Berechnung der Abstände zu den Hindernissen sowie zur Bestimmung von Kollisionen benötigt. Allgemein wird dazu für einen Roboter in der Konfiguration \mathbf{q} und ein Hindernis A im Konfigurationsraum die skalare Distanzfunktion $d(\mathbf{q}, A)$ gemäß

$d(\mathbf{q}, A) > 0 :$	Kein Kontakt mit dem Hindernis A ,
$d(\mathbf{q}, A) = 0 :$	Kontakt an der Kontur von Hindernis A ,
$d(\mathbf{q}, A) < 0 :$	Roboter dringt in Hindernis A ein.

eingeführt. Diese Distanz kann beispielsweise der geringste Euklidische Abstand zwischen dem Roboter und dem nächsten Punkt am Hindernis sein. Weiters wird zwischen zwei Arten von Algorithmen unterschieden: Einerseits gibt es Algorithmen, die die skalare Distanz $d(\mathbf{q}, A)$ berechnen und andererseits gibt es Algorithmen, welche nur den logischen Ausdruck $d(\mathbf{q}, A) \leq 0$ liefern. Letztere sind nur für die *Kollisionsprüfung* geeignet. Für konkrete Algorithmen und Implementierungen sei auf die Literatur am Ende des Kapitels verwiesen.

Graphen

Viele Algorithmen für die Bewegungsplanung verwenden im Inneren einen *Graphen*, um den Konfigurations- oder Zustandsraum darzustellen. Dabei wird zwischen *gerichteten* und *ungerichteten* Graphen, sowie zwischen *gewichteten* und *ungewichteten* Graphen unterschieden, siehe Abbildungen 7.14a und 7.14b. Eine Sonderform des gerichteten Graphen stellt ein *Baum* (Abbildung 7.14c) dar.

Sobald die Planungsaufgabe als Graph dargestellt ist, kann ein Suchalgorithmus verwendet werden, um einen Pfad vom Start- zum Zielpunkt zu finden. Einer der bekanntesten und leistungsfähigsten Algorithmen dabei ist der A*-Suchalgorithmus². Weitere wichtige

²gesprochen „A star“

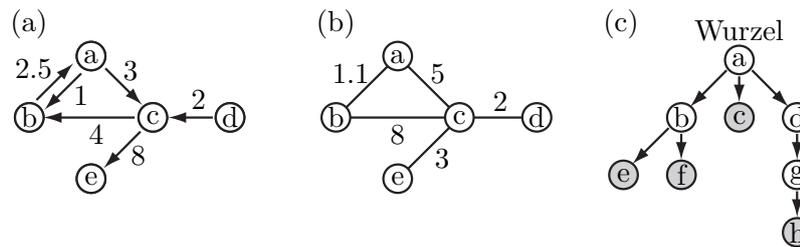


Abbildung 7.14: Beispiele für Graphen: (a) Gerichteter, gewichteter Graph (Digraph), (b) Ungerichteter, gewichteter Graph, (c) Baum [7.4].

Algorithmen sind der *Dijkstra*-Algorithmus, die *Breitensuche* (engl. *Breadth-first search*) und die suboptimale A^* -Suche.

7.2.2 Definition

Definition 7.2 (Bewegungsplanungsproblem). Gegeben seien die Systemgleichungen eines Robotersystems in der allgemeinen Form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (7.35)$$

mit dem Zustandsvektor $\mathbf{x} \in X$ aus dem Zustandsraum X und dem Systemeingang $\mathbf{u} \in \mathbb{R}^u$, sowie ein Anfangszustand \mathbf{x}_0 und ein Zielzustand \mathbf{x}_1 . Dann ist das Bewegungsplanungsproblem die Suche nach dem Zeitverlauf für den Systemeingang $\mathbf{u} : [0, T] \rightarrow U$ aus dem Raum der gültigen Eingangsverläufe U und einer Zeitdauer T , sodass die Systemgleichungen (7.35) erfüllt sind, $\mathbf{x}(T) = \mathbf{x}_1$ erreicht wird und $\mathbf{q}(t) \in C_f \forall t \in [0, T]$ mit dem Freiraum C_f gilt.

7.2.3 Kategorien und Eigenschaften

In diesem Abschnitt werden die Planungsmethoden anhand einiger Kategorien und Eigenschaften unterteilt und anhand dieser Unterteilungen diskutiert.

Planung im Konfigurationsraum/Aufgabenraum Die Bewegungsplanung kann sowohl im Konfigurationsraum ($\mathbf{x} = \mathbf{q}$, $X = \mathbb{R}^n$) als auch im Aufgabenraum (z. B. $\mathbf{x} = \mathbf{x}_e$, $X = \text{SE}(3)$ oder $X = \mathbb{R}^6$) ausgeführt werden. Für die Planung im Aufgabenraum sind häufig Verallgemeinerungen der Algorithmen dieses Abschnitts erforderlich, weshalb hier vorrangig der Konfigurationsraum betrachtet wird.

Kinematische/dynamische Planung Eine weitere Kategorie unterscheidet zwischen kinematischer und dynamischer Planung. Die Systemgleichungen eines allgemeinen Robotersystems mit den Eingängen $\mathbf{u} \in \mathbb{R}^u$ werden als Differentialgleichung (7.35) angeschrieben.

Handelt es sich um eine kinematische Planung, so bleiben die dynamischen Effekte des Robotersystems unberücksichtigt. In diesem Fall wird als Eingang die Gelenkgeschwindigkeit $\mathbf{u} = \dot{\mathbf{q}}$ gewählt, der Zustandsvektor des Systems beinhaltet die

Gelenkpositionen, d. h. $\mathbf{x} = \mathbf{q}$ und die Differenzialgleichung ist ein Einfachintegrator für jedes Gelenk. Gleichermaßen ist es bei der kinematischen Planung möglich, die Gelenksbeschleunigung $\mathbf{u} = \ddot{\mathbf{q}}$ oder höhere Ableitungen als Eingang zu wählen. Entsprechend enthält der Zustandsvektor dann auch höhere Ableitungen, z. B. $\mathbf{x}^T = [\mathbf{q}^T \dot{\mathbf{q}}^T]$ und die Differenzialgleichungen $\mathbf{f}(\mathbf{x}, \mathbf{u})$ werden zu Integratorketten zweiter oder höherer Ordnung.

Bei der dynamischen Planung werden die Kräfte und Momente der Aktoren als Eingang \mathbf{u} gewählt. In diesem Fall ist (7.35) das dynamische Modell des Robotersystems, in welchem die relevanten Massen, Massenträgheitsmomente sowie die Gravitation und exogene Kräfte enthalten sind, siehe Kapitel 6 „Dynamik“. Es handelt sich dann um ein System zweiter Ordnung und der Zustandsvektor ist $\mathbf{x}^T = [\mathbf{q}^T \dot{\mathbf{q}}^T]$.

Pfadplanung/Trajektorienplanung Pfadplanung ist ein Teilproblem der Bewegungsplanung, bei der ein Lösungspfad $\mathbf{q}(s)$, $s \in [0, 1]$ rein geometrisch von einer Startkonfiguration $\mathbf{q}(0) = \mathbf{q}_0$ zu einer Zielkonfiguration $\mathbf{q}(1) = \mathbf{q}_1$ gesucht wird. Dabei bleiben die Systemdynamik, die Dauer der Bewegungen T oder auch Beschränkungen in den Geschwindigkeiten, Beschleunigungen oder Systemeingängen unberücksichtigt. Durch geeignete Zeitparametrierung bzw. Zeitskalierung, siehe Abschnitt 7.1, kann meist eine ausführbare Trajektorie aus dem Pfad erzeugt werden.

Anzahl der Systemeingänge Die Komplexität der Bewegungsplanung steigt, wenn die Dimension u der Systemeingänge kleiner ist als die Dimension m des Aufgabenraums und die Dimension n des Konfigurationsraums. Beispielsweise kann zwar ein Fahrzeug alle Zustände in der Ebene einnehmen (zweidimensionale Translation und Rotation um die Hochachse, $m = n = 3$), jedoch stehen nur zwei Eingänge zur Verfügung ($u = 2$, d. h. Vorwärtsbewegung und Lenken). Eine Seitwärtsbewegung ist hingegen nicht zulässig und dieser Umstand muss in der Bewegungsplanung berücksichtigt werden.

Online-/Offline-Bewegungsplanung Erfolgt die Bewegungsplanung in einer dynamischen Umgebung, in der sich Hindernisse bewegen können, so wird eine Online-Bewegungsplanung benötigt, welche die Planung entsprechend dieser Dynamik schnell genug abschließen kann. Dies hat auch Auswirkungen auf die benötigte Rechenleistung und die verfügbare Rechenzeit des Algorithmus. Sehr komplexe Planungsaufgaben in statischer Umgebung können mit einer Offline-Bewegungsplanung gelöst werden, da das Planungsergebnis nicht sofort durch den Roboter ausgeführt werden muss.

Anforderungserfüllende/Optimale Bewegungsplanung Eine Bewegungsplanung kann so ausgelegt sein, dass sie eine gültige Lösung, die das Ziel $\mathbf{x}(T) = \mathbf{x}_1$ erreicht, liefert – ohne weitere Anforderungen. Darüber hinaus können auch Kriterien der Optimalität berücksichtigt werden, wenn z. B. das Kostenfunktional $J(\mathbf{u}(\cdot))$ gemäß

$$J(\mathbf{u}(\cdot)) = \int_0^T l(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (7.36)$$

minimiert wird. Beispielsweise führt die Wahl $l(\cdot, \cdot) = 1$ zu zeitoptimalen Lösungen der Bewegungsplanung und $l(\cdot, \mathbf{u}(t)) = \mathbf{u}^T(t)\mathbf{u}(t)$ minimiert den Einsatz der Systemeingänge \mathbf{u} .

Exakte/Approximierte Lösung der Bewegungsplanung Beim approximierten Lösen des Bewegungsplanungsproblems werden die strikten Gleichungsbedingungen des Zielzustandes $\mathbf{x}(T) = \mathbf{x}_1$ oder andere Beschränkungen auf Ungleichungsbedingungen gelockert, z. B. $\|\mathbf{x}(T) - \mathbf{x}_1\| < \varepsilon$. Weiters können Gleichungs- und Ungleichungsbedingungen auch approximativ erfüllt werden, wenn sie in die Kostenfunktion einer optimierungsbasierten Planung (7.36) verschoben werden.

Mehrfachabfrage/Einzelabfrage Bewegungsplaner, welche für Mehrfachabfragen (engl. *multi-query*) ausgelegt sind, können die Effizienz steigern, wenn Teilprobleme gelöst und in geeigneten Datenstrukturen abgespeichert werden. Diese Datenstrukturen können anschließend für weitere Abfragen verwendet werden, um das Planungsproblem schneller zu lösen. Bewegungsplaner für Einzelabfragen (engl. *single-query*) beginnen jeden Planungsdurchlauf von Neuem.

„Anytime“-Bewegungsplaner Ein Bewegungsplaner mit der Eigenschaft *anytime* (dt. *jederzeit*) ist dafür ausgelegt, bereits nach kurzer Zeit eine gültige Lösung liefern zu können. Die verbleibende Planungszeit wird verwendet, um bessere Lösungen zu suchen. Der Algorithmus kann jederzeit mit der momentan besten Lösung beendet werden. Solche Planungsalgorithmen werden speziell in Echtzeitanwendungen benötigt, in denen nur ein vorgegebenes Zeitfenster für die Planung zur Verfügung steht.

Vollständigkeit Ein *vollständiger* (engl. *complete*) Algorithmus für die Bewegungsplanung kann garantieren, dass eine Lösung in endlicher Zeit gefunden wird, sofern eine Lösung existiert. Existiert keine gültige Lösung für ein gegebenes Problem, so kann dies ebenfalls nach endlicher Zeit garantiert festgestellt werden. Eine Abschwächung dieser Eigenschaft ist die *Vollständigkeit in Bezug auf die Auflösung* (engl. *resolution completeness*), bei der die Vollständigkeit nur in einem diskretisierten Suchraum garantiert wird. Als *probabilistisch vollständig* (engl. *probabilistically complete*) werden Algorithmen bezeichnet, bei denen die Wahrscheinlichkeit, eine Lösung zu finden, gegen 100 % strebt, wenn die Planungszeit gegen unendlich geht – sofern eine Lösung existiert.

Rechen- und Speicheraufwand Ein Planungsalgorithmus wird anhand der Laufzeit eines Planungsvorganges sowie der Menge an benötigtem Speicher charakterisiert. Dabei hängt dieser Bedarf nicht nur vom Algorithmus ab, sondern auch von der Größe des bearbeiteten Problems, z. B. von der Dimension n des Konfigurationsraumes oder der Anzahl an Knotenpunkten, die für die Darstellung der Hindernisse notwendig sind. Häufig wird die Komplexität eines Algorithmus mithilfe des LANDAU-Symbols $\mathcal{O}(\cdot)$ beurteilt.

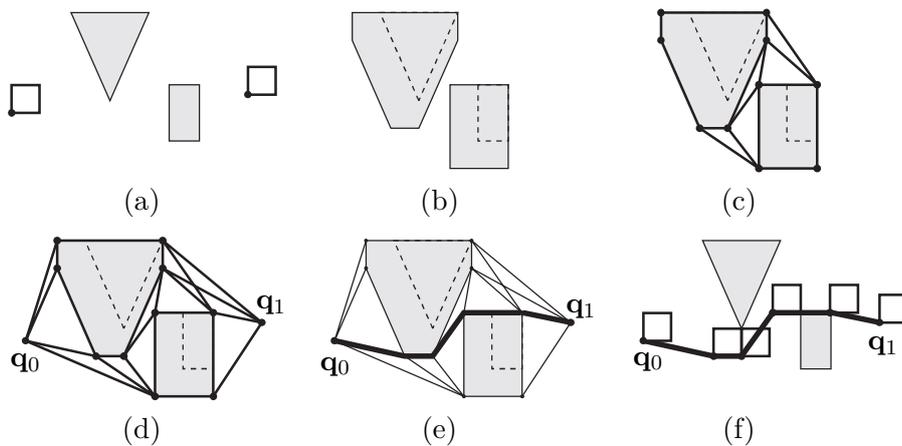


Abbildung 7.15: Schematische Skizze für den Planungsvorgang eines vollständigen Pfadplaners: (a) Start- und Zielkonfigurationen q_0 bzw. q_1 für einen quadratischen Roboter in der Ebene mit zwei Hindernissen, (b) in den Konfigurationsraum übertragene Hindernisse, (c) Roadmap R , (d) vollständiger Graph mit Start- und Zielkonfiguration, (e) Ergebnis der Pfadplanung im Graph, (f) Bewegung des Roboters entlang des geplanten Pfades [7.4].

7.2.4 Methoden

Im Folgenden werden die wichtigsten Methoden für die Planung von Roboterbewegungen aufgelistet und kurz erklärt. Dieser Abschnitt soll einen breiten und kompakten Überblick über das große Forschungsfeld der Bewegungsplanung geben.

Vollständige Pfadplaner

Vollständige Pfadplaner arbeiten mit einer exakten Darstellung des Freiraums C_f und haben daher eine hohe Komplexität. Folglich eignen sich diese nur für sehr einfache Systeme, wie sie in der Praxis selten vorkommen.

Ein Beispiel eines vollständigen Planers verwendet eine sogenannte eindimensionale *Roadmap* R (dt. Fahrplan) für die Darstellung des Freiraums C_f mit folgenden zwei Eigenschaften: (i) *Erreichbarkeit* (engl. *reachability*), d. h. für jeden Punkt $q \in C_f$ gibt es einen Punkt $q' \in R$ auf der Roadmap R , zu welchem ein trivialer Pfad führt, (ii) *Konnektivität* (engl. *connectivity*), d. h. jede Region von C_f ist zumindest an einer Stelle mit der Roadmap R verbunden. Der Algorithmus ist in Abbildung 7.15 grafisch veranschaulicht. Mithilfe der Roadmap sucht der Pfadplaner für den Startpunkt q_0 jenen Punkt $q'_0 \in R$, welcher in der selben Region liegt. Zwischen den Punkten q_0 und q'_0 kann folglich ein trivialer Pfad geplant werden. Der selbe Vorgang wird auch für den Zielpunkt q_1 mit $q'_1 \in R$ durchgeführt. Schließlich wird über die Roadmap R ein Pfad gesucht, welcher die beiden Punkte q'_0 und q'_1 miteinander verbindet.

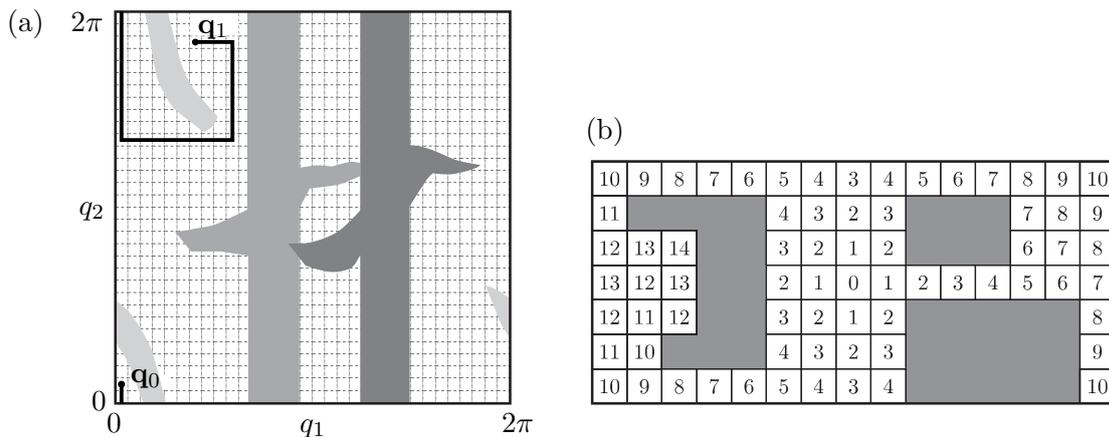


Abbildung 7.16: Beispiele für gitterbasierte Methoden für die Bewegungsplanung: (a) Planungsergebnis für die Aufgabenstellung aus Abbildung 7.12, (b) Planungsgitter und Speicherinhalt für einen Wavefront-Bewegungsplaner [7.4].

Gitterbasierte Methoden

Bei den gitterbasierten Methoden wird der Freiraum C_f diskretisiert und danach eine durchgängige Bewegung von der Startkonfiguration \mathbf{q}_0 zur Zielkonfiguration \mathbf{q}_1 gesucht. Anstatt des Konfigurationsraumes kann auch der Zustandsraum oder der Aufgabenraum diskretisiert werden und es können auch Gitter mit unterschiedlicher Auflösung geeignet miteinander kombiniert werden (*multi-resolution grid representation*). Der limitierende Faktor bei den gitterbasierten Methoden ist der Speicherverbrauch, welcher exponentiell mit der Anzahl an Dimensionen steigt. Wird der Suchraum des Pfadplaners entlang einer Dimension in nur 100 Gitterpunkte geteilt, so sind in einem dreidimensionalen Problem 1 Million Gitterpunkte zu verarbeiten. Bei einem kollaborativen 7-Achs-Roboter sind es bereits 10^{14} Gitterpunkte, was praktisch nicht mehr zu bewältigen ist³.

Abbildung 7.16a zeigt das Planungsergebnis zur Aufgabenstellung aus Abbildung 7.12. Dabei wurden 32 Gitterpunkte im Konfigurationsraum $\mathbf{q} = [q_1 \ q_2]^T \in C \subseteq \mathbb{R}^2$ verwendet, d. h. die Winkelauflösung für jedes Gelenk liegt bei $360^\circ/32 = 11.25^\circ$. Die Planung erfolgte mit der sogenannten *Manhattan-Distanz*, welche nur orthogonale Bewegungen über die Gitterpunkte zulässt.

Ein sogenannter *Wellenfront-Bewegungsplaner* (engl. *wavefront planner*) ist eine Variante der gitterbasierten Bewegungsplaner, welche für Mehrfachabfragen konzipiert ist. Ein Beispiel ist in Abbildung 7.16 dargestellt. In jedem Gitterpunkt die minimale Manhattan-Distanz zu einem festgelegten Startpunkt eingetragen. Dies geschieht vom Startpunkt ausgehend, ähnlich einer Wellenfront. Anhand dieser Karte können nun sehr schnell viele Pfade von diesem Startpunkt ausgehend geplant werden. Der Planer muss lediglich von Gitterpunkt zu Gitterpunkt in Richtung der absteigenden Zahlenwerte planen.

³In der Literatur wird dies auch als der „Fluch der Dimensionalität“ (engl. *curse of dimensionality*) bezeichnet.

Sampling-Methoden

Sampling-Methoden für die Bewegungsplanung verwenden ausgewählte Punkte im Konfigurations- oder Zustandsraum des Roboters, um einen Graphen oder Baum aufzubauen. Folgende Elemente werden im Allgemeinen für einen Sampling-Algorithmus benötigt:

- Eine Zufallsfunktion oder deterministische Funktion zum Auswählen eines Samples,
- eine Funktion zum Prüfen, ob ein gewählter Punkt im Freiraum C_f liegt,
- eine Funktion zum Ermitteln des „nächstgelegenen“ Sample im Freiraum C_f ,
- ein lokaler Planer, der die Samples kollisionsfrei verbindet.

Im Allgemeinen sind Sampling-Methoden einfach zu implementieren, sind probabilistisch Vollständig und sind auch für höherdimensionale Planungsprobleme noch geeignet. Üblicherweise sind die Lösungen eines Algorithmus aus dieser Gruppe nicht optimal und erfordern einen nachgelagerten Prozess zum Glätten des Lösungspfades. Die Sampling-Methoden werden wie folgt in zwei Kategorien unterteilt.

Rapidly-exploring Random Tree (RRT) Bei einem RRT wird ein neues Sample $\mathbf{x}_s \in X$ zufällig aus dem Zustandsraum des Systems mit einer Gleichverteilung gewählt. Häufig wird zusätzlich ein Bias in Richtung des Zielzustandes \mathbf{x}_1 in die Zufallsverteilung eingebracht. Anschließend wird jener Knoten \mathbf{x}_t des Suchbaumes T bestimmt, welcher die geringste (euklidische) Distanz aufweist. Nun wird ein neuer Punkt \mathbf{x}_n erzeugt, welcher maximal eine kleine Distanz d vom Baumknoten \mathbf{x}_t in Richtung des Samples \mathbf{x}_s liegt. Dieser neue Punkt \mathbf{x}_n wird dann in den Suchbaum T aufgenommen, wenn das Robotersystem kollisionsfrei von \mathbf{x}_t nach \mathbf{x}_n fahren kann. Da der Abstand d klein ist, kann ein einfacher, lokaler Planer diese Überprüfung ausführen. Die obigen Schritte werden so lange wiederholt, bis eine Verbindung zwischen dem Startzustand \mathbf{x}_0 und dem Zielzustand \mathbf{x}_1 gefunden wurde. Ein Beispiel eines RRT in einem hindernisfreien Konfigurationsraum $C = \mathbb{R}^2$ zeigt Abbildung 7.17a. Abgesehen von dieser grundlegenden Methode gibt es noch eine Reihe von Modifikationen, wie z. B. Bidirectional RRT und RRT*.

Probabilistic Roadmap (PRM) Beim PRM-Algorithmus wird durch zufällige Auswahl von neuen Knotenpunkten aus dem Konfigurations- oder Zustandsraum eine Roadmap R generiert, analog zu den vollständigen Planern. Ein lokaler Planer versucht, die neuen Knotenpunkte zur Roadmap R hinzuzufügen, indem jene Knotenpunkte mit der geringsten Distanz in der Roadmap gesucht werden und eine Verbindung zum neuen Punkt hergestellt wird. Diese Schritte werden solange wiederholt, bis der Graph den Freiraum C_f dicht genug ausfüllt. Abschließend werden noch der Start- und der Zielzustand \mathbf{x}_0 bzw. \mathbf{x}_1 hinzugefügt. Ein Beispiel für eine fertiggestellte PRM zeigt Abbildung 7.17b, welche für eine Mehrfachabfrage geeignet ist. Mithilfe eines Suchalgorithmus, typischerweise A*, können nun Pfade geplant werden.

Virtuelle Potenzialfelder

Ein physikalisches Potenzialfeld $V(\mathbf{q})$ führt zu einer Kraft $\mathbf{f} = -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}}$, welche ein Objekt von Bereichen mit höherem Potenzial zu niedrigerem Potenzial treibt. In der Bewegungs-

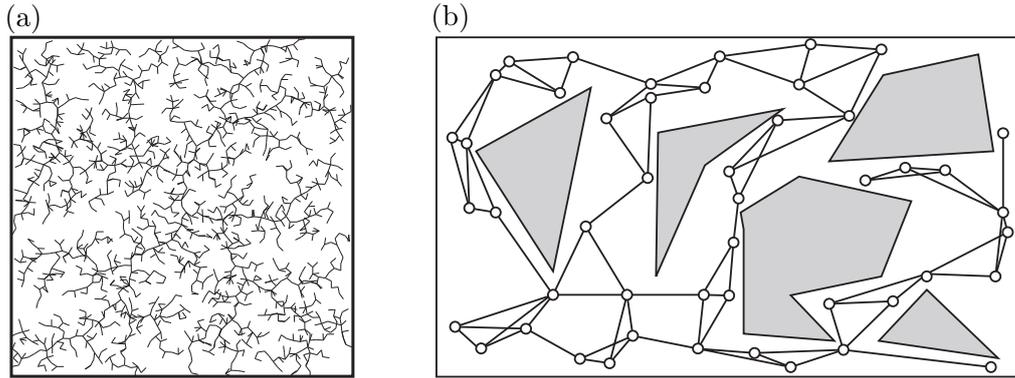


Abbildung 7.17: Beispiele für Sampling-Methoden: (a) Rapidly-exploring Random Tree (RRT), (b) Probabilistic Roadmap (PRM) für einen punktförmigen Roboter im Konfigurationsraum $C = \mathbb{R}^2$ [7.4].

planung werden *virtuelle Potenzialfelder* verwendet, um die Trajektorie eines (mobilen) Roboters zu berechnen. Diese virtuellen Potenzialfelder rufen während der Planung virtuelle Kräfte hervor, welche den Roboter in Richtung des Ziels ziehen sowie ihn von Hindernissen wegstoßen. Die dabei entstehende Trajektorie ist das Ergebnis der Bewegungsplanung.

Aufgrund der Einfachheit ist eine Implementierung in höherdimensionalen Systemen sowie auch in Online-Bewegungsplanern möglich. Aufgrund der begrenzten (lokalen) Wirkung der Potenzialfelder kann es aber vorkommen, dass der Roboter zwischen den Potenzialfeldern „hängen bleibt“, weil sich lokal die anziehenden und abstoßenden Kräfte gegenseitig aufheben.

Um die Methode der virtuellen Potenzialfelder zu illustrieren, werden im Folgenden die grundlegenden Zusammenhänge für einen punktförmigen Roboter im Konfigurationsraum C angeschrieben. Ein virtuelles Potenzial für einen Zielpunkt \mathbf{q}_1 wird meist als quadratisches Potenzial in der Form

$$V_1(\mathbf{q}) = \frac{1}{2}(\mathbf{q} - \mathbf{q}_1)^T \mathbf{K}(\mathbf{q} - \mathbf{q}_1) \quad (7.37)$$

mit der symmetrischen, positiv definiten Gewichtungsmatrix \mathbf{K} angesetzt. Die zugehörige Kraft ist

$$\mathbf{f}_1(\mathbf{q}) = -\frac{\partial V_1(\mathbf{q})}{\partial \mathbf{q}} = \mathbf{K}(\mathbf{q}_1 - \mathbf{q}) . \quad (7.38)$$

Darin ist ersichtlich, dass die anziehende Kraft proportional zur Distanz zum Ziel ist. Für ein Hindernis A im Konfigurationsraum wird ein repulsives Potenzial mithilfe der Distanzfunktion $d(\mathbf{q}, A)$ gemäß

$$V_A(\mathbf{q}) = \frac{k}{2d^2(\mathbf{q}, A)} \quad (7.39)$$

mit der Kraft

$$\mathbf{f}_A(\mathbf{q}) = -\frac{\partial V_A(\mathbf{q})}{\partial \mathbf{q}} = \frac{k}{2d^3(\mathbf{q}, A)} \frac{\partial d}{\partial \mathbf{q}} \quad (7.40)$$

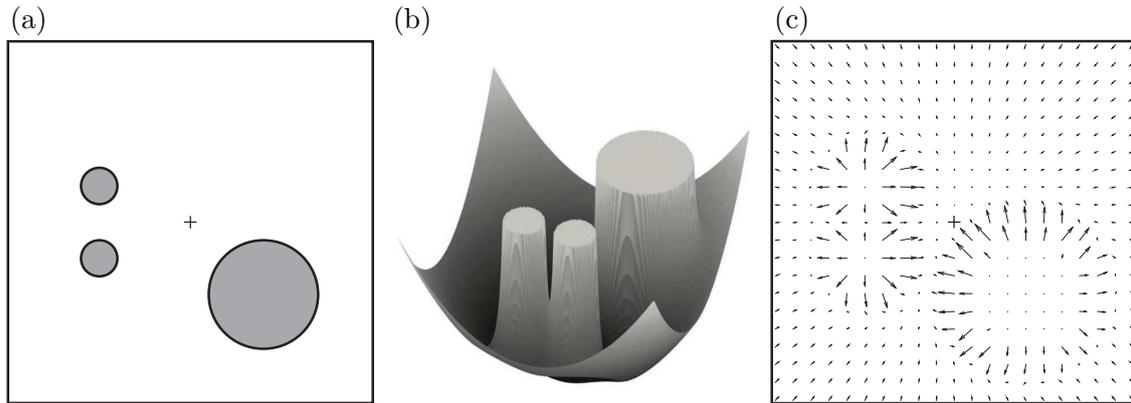


Abbildung 7.18: Beispiel für einen zweidimensionalen Konfigurationsraum mit drei kreisförmigen Hindernissen. (a) Schematische Darstellung des Konfigurationsraums, (b) Potenzialfunktion als Summe einer quadratischen Funktion, um den Roboter zum Ziel zu ziehen, sowie drei abstoßenden Feldern für die Hindernisse, (c) lokale Kräfte, welche durch die Potenzialfelder hervorgerufen werden [7.4].

eingeführt. Darin ist k ein Skalierungsfaktor. Weiters ist dieses Potenzial nur für Punkte außerhalb des Hindernisses $d(\mathbf{q}, A) > 0$ definiert.

In einem Planungsraum mit mehreren Hindernissen A_i , $i = 1, \dots, N$, findet sich das gesamte virtuelle Potenzialfeld bzw. die gesamte auf den Roboter wirkende Kraft als Summe der Einzelkomponenten gemäß

$$V(\mathbf{q}) = V_1(\mathbf{q}) + \sum_{i=1}^N V_{A_i}(\mathbf{q}) \quad (7.41)$$

bzw.

$$\mathbf{f}(\mathbf{q}) = \mathbf{f}_1(\mathbf{q}) + \sum_{i=1}^N \mathbf{f}_{A_i}(\mathbf{q}) . \quad (7.42)$$

Der Ausdruck (7.42) kann nun als Eingang \mathbf{u} für die Systemgleichungen (7.35) verwendet werden, damit die Bewegung des Robotersystems durch die Potenzialfelder geführt wird. Ein Beispiel für ein Potenzialfeld für einen Konfigurationsraum mit drei kreisförmigen Hindernissen zeigt Abbildung 7.18.

Nichtlineare Optimierung

Eine Aufgabe in der Bewegungsplanung kann auch als Optimierungsproblem bzw. Optimalsteuerungsproblem aufgefasst und formuliert werden und anschließend mit bekannten Algorithmen aus der nichtlinearen Optimierung gelöst werden. Dabei wird die gesuchte Lösung mit einer endlichen Anzahl an Optimierungsvariablen dargestellt und bezüglich einer Kostenfunktion (7.36), Beschränkungen und weiteren Kriterien optimiert. Dabei werden Gradientenmethoden, sequentielle quadratische Programmierung (SQP) oder auch

ableitungsfreie Methoden wie das NELDER-MEAD-Verfahren verwendet. Die Herausforderung liegt meist im Finden und Formulieren der Kostenfunktion J . Aufgrund der Nichtlinearität und Nichtkonvexität des Problems, kann nicht garantiert werden, dass eine (gültige) Lösung gefunden wird. Weiters wird eine Anfangslösung benötigt.

Eine Beispielformulierung für ein allgemeines Optimalsteuerungsproblem ist

$$\min_{\mathbf{u}(\cdot)} J(\mathbf{u}(\cdot)) = \varphi(T, \mathbf{x}(T)) + \int_0^T l(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad (7.43a)$$

$$\text{u.B.v. } \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) , \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (7.43b)$$

$$\boldsymbol{\psi}(T, \mathbf{x}(T)) = \mathbf{0} \quad (7.43c)$$

$$\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} , \quad \forall t \in [0, T] , \quad (7.43d)$$

mit der Kostenfunktion (7.43a), den Systemgleichungen (7.43b), den Endbedingungen (7.43c) sowie den Ungleichungsnebenbedingungen (7.43d). Für weiterführende Informationen sei auf die Vorlesung *Optimierung* [7.5] verwiesen.

7.3 Literatur

- [7.1] B. Siciliano und O. Khatib, Hrsg., *Springer Handbook of Robotics*. Springer, 2008. Adresse: <https://link.springer.com/book/10.1007/978-3-540-30301-5>.
- [7.2] A. Lehmann, R. Mikut und T. Asfour, „Petri nets for task supervision in humanoid Robots,“ *VDI Berichte*, Jg. 1956, S. 71, 2006.
- [7.3] J. Jeon, H.-r. Jung, T. Luong, F. Yumbla und H. Moon, „Combined task and motion planning system for the service robot using hierarchical action decomposition,“ *Intelligent Service Robotics*, Jg. 15, Nr. 4, S. 487–501, 2022.
- [7.4] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [7.5] A. Steinböck, *Skriptum zur VU Optimierung (WS 2023/24)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2023. Adresse: <https://www.acin.tuwien.ac.at/master/optimierung/>.

8 Mobile Roboter

Dieses Kapitel beschäftigt sich mit der Modellbildung und Systemanalyse von radbasierten mobilen Robotern. Radbasierte mobile Roboter lassen sich grundsätzlich in *omnidirektionale* und *nichtholonome* mobile Roboter unterteilen. Der Fokus dieses Kapitels liegt auf der Kinematik, insbesondere auf dem Zusammenhang zwischen den Geschwindigkeiten der einzelnen Räder und der Gesamtbewegung des mobilen Roboters. Weiters werden die Themen Planung und Regelung für beide Robotertypen kurz diskutiert.

8.1 Starrkörperbewegungen in der Ebene

In diesem Abschnitt werden die wichtigsten Ergebnisse aus Kapitel 3 auf Bewegungen in der Ebene übertragen. Die Pose des Starrkörpers mit dem körperfesten Koordinatensystem $\{k\}$ in der Ebene mit dem Inertialkoordinatensystem $\{w\}$, siehe Abb. 8.1, wird durch eine homogene Transformation ${}^w\mathbf{H}_k \in \text{SE}(2)$ gemäß folgender Definition beschrieben.

Definition 8.1. Die *spezielle euklidische Gruppe im 2-dimensionalen Raum* $\text{SE}(2)$ umfasst alle reellen 3×3 -Matrizen der Bauform

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}(\phi) & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (8.1)$$

mit der Rotationsmatrix $\mathbf{R}(\phi) \in \text{SO}(2)$ gemäß

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (8.2)$$

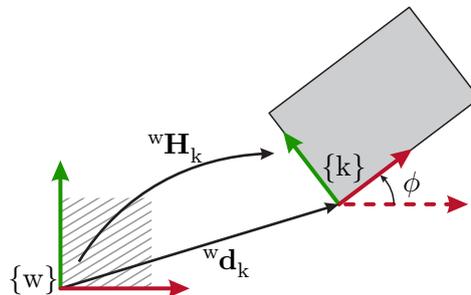


Abbildung 8.1: Pose eines Starrkörpers mit dem körperfesten Koordinatensystem $\{k\}$ in der Ebene mit dem Inertialkoordinatensystem $\{w\}$.

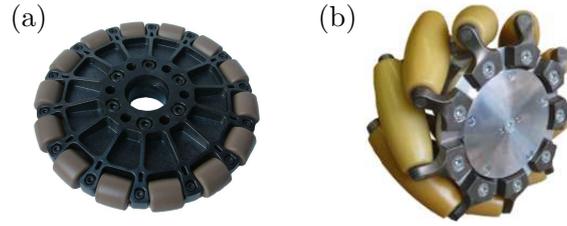


Abbildung 8.2: Beispiele für omnidirektionale Räder: (a) Allseitenrad, (b) Mecanum-Rad.

und dem Verschiebungsvektor $\mathbf{d} = [d_x \ d_y]^T \in \mathbb{R}^2$. Die Rotationsmatrix $\mathbf{R}(\phi)$ entspricht stets der linken oberen 2×2 -Matrix von $\mathbf{R}_{z,\phi}$, vgl. (3.17a), da die einzig zulässige Drehung die Rotation um die Normale der Bewegungsebene ist. Die Matrizen \mathbf{H} werden *homogene Transformationen* bzw. *Transformationsmatrizen im 2-dimensionalen Raum* genannt.

Analog zu (3.36), (3.48) und (3.49) kann die Geschwindigkeit eines Starrkörpers in der Ebene als *Vektor der räumlichen Geschwindigkeit im körperfesten Koordinatensystem im 2-dimensionalen Raum* ${}^k\mathbf{t}_k^w = [({}^k\mathbf{v}_k^w) \quad {}^k\omega_k^w]^T$ in der Form

$${}^w\mathbf{H}_k^{-1} {}^w\dot{\mathbf{H}}_k = \begin{bmatrix} {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k & {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{d}}_k \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{S}({}^k\omega_k^w) & {}^k\mathbf{v}_k^w \\ \mathbf{0} & 0 \end{bmatrix} \quad (8.3)$$

mit der schiefsymmetrischen Matrix

$${}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k = \mathbf{R}^T(\phi) \dot{\mathbf{R}}(\phi) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \dot{\phi} = \mathbf{S}({}^k\omega_k^w) \quad (8.4)$$

angeschrieben werden.

8.2 Omnidirektionale mobile Roboter

Es wird angenommen, dass sich der mobile Roboter in der Ebene auf einem undeformierbaren, horizontalen Untergrund bewegt. Omnidirektionale mobile Roboter erlauben zu jedem Zeitpunkt eine Bewegung in jede beliebige Richtung des $SE(2)$. Die Räder von omnidirektionalen Robotern sind üblicherweise nicht lenkbar, d.h. die Drehachsen der Räder sind fest mit dem Fahrgestell verbunden. Bei omnidirektionalen mobilen Robotern wird eine seitliche Bewegung des Rades in eine Richtung orthogonal zur Radebene durch eine besondere Konstruktion der Räder ermöglicht.

Abbildung 8.2 zeigt zwei typische Bauformen von omnidirektionalen Rädern. Im Allseitenrad (Abb. 8.2a) sind entlang des Radumfangs zusätzliche Rollen angebracht. Diese Rollen werden nicht aktiv angetrieben und erlauben eine passive Bewegung orthogonal zur Radebene. Mecanum-Räder (Abb. 8.2b) haben einen ähnlichen Aufbau, wobei die Drehachsen der passiven Rollen nicht in der Radebene liegen, sondern um einen Winkel

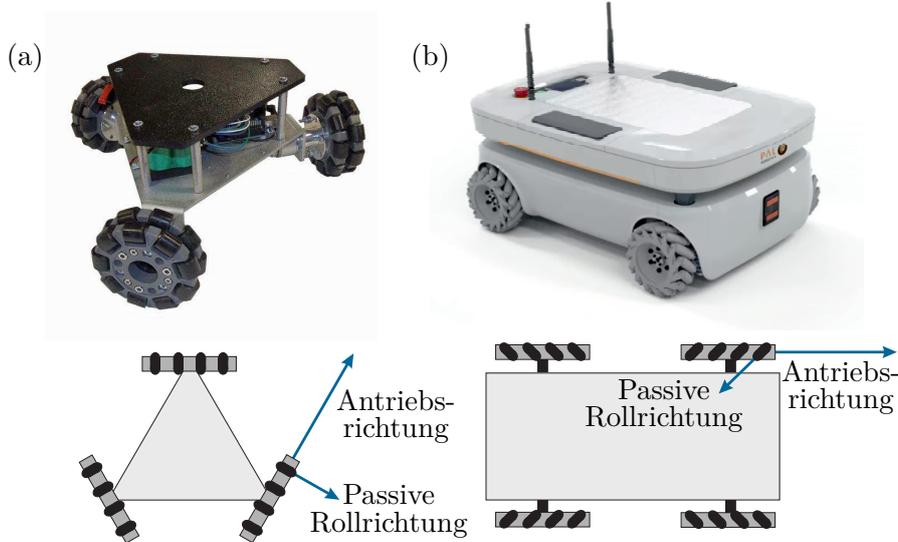


Abbildung 8.3: Beispiele für omnidirektionale mobile Roboter: (a) Programmable Triangular Omni Wheel Vectoring Robot [8.1] (b) TIAGo OMNI Base [8.2].

von üblicherweise 45° versetzt sind. Aufgrund der relativ kleinen Durchmesser der äußeren Rollen haben mobile Roboter mit omnidirektionalen Rädern Einschränkungen auf unebenem oder nachgiebigem Untergrund.

Um jede beliebige Geschwindigkeit ${}^k\mathbf{t}_k^w$ – d. h. zwei translatorische Geschwindigkeiten ${}^k\mathbf{v}_k^w$ und eine Drehwinkelgeschwindigkeit ${}^k\omega_k^w$ – voneinander unabhängig zu erzeugen, sind mindestens drei Räder erforderlich. Abbildung 8.3 zeigt zwei typische Ausführungsformen von omnidirektionalen Robotern mit drei bzw. vier omnidirektionalen Rädern.

8.2.1 Kinematisches Modell

Im Folgenden bezeichnet $r \geq 3$ die Anzahl der omnidirektionalen Räder des Roboters und θ_i , $i = 1, \dots, r$, sind die Drehwinkel der r Räder. Die Drehwinkelgeschwindigkeiten der Räder werden als Systemeingang betrachtet. Dazu wird

$$\mathbf{u} = [u_1 \ u_2 \ \dots \ u_r]^T = [\dot{\theta}_1 \ \dot{\theta}_2 \ \dots \ \dot{\theta}_r]^T \quad (8.5)$$

gewählt. Die Konfiguration eines omnidirektionalen mobilen Roboters ist durch die Koordinaten des Basiskörpers gegeben, d. h. $\mathbf{q} = [d_x \ d_y \ \phi]^T$, siehe Abb. 8.4. Die absoluten Drehungen θ_i der einzelnen Räder sind für die folgende Analyse nicht relevant und werden daher nicht als zusätzliche Koordinaten im Konfigurationsraum behandelt. Im Folgenden wird gezeigt, welche Drehgeschwindigkeiten u_i an den Rädern erforderlich sind, um eine gewünschte Gesamtgeschwindigkeit $\dot{\mathbf{q}}_d$ im Konfigurationsraum bzw. eine räumliche Geschwindigkeit ${}^k\mathbf{t}_k^w$ zu erhalten.

Die Pose des Roboters kann durch die homogene Transformation ${}^w\mathbf{H}_k$ und gleichermaßen

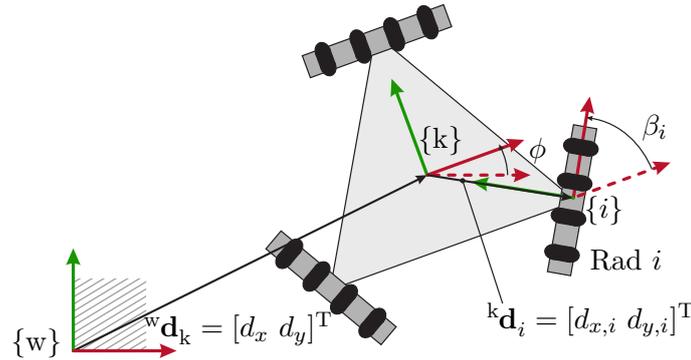


Abbildung 8.4: Kinematik eines radbasierten mobilen Roboters mit dem körperfesten Koordinatensystem $\{k\}$ im Inertialkoordinatensystem $\{w\}$. Die Lage der Räder i , $i = 1, \dots, r$, wird durch die Koordinatensystem $\{i\}$ beschrieben, welche körperfest in $\{k\}$ liegen.

durch die Konfiguration des Roboters

$$\mathbf{q} = [d_x \quad d_y \quad \phi]^T \Leftrightarrow {}^w\mathbf{H}_k = \begin{bmatrix} {}^w\mathbf{R}_k & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\phi) & \begin{bmatrix} d_x \\ d_y \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \quad (8.6)$$

mit ${}^w\mathbf{d}_k = [d_x \quad d_y]^T$ beschrieben werden¹. Die Zusammenhänge mit der Geschwindigkeit im Konfigurationsraum $\dot{\mathbf{q}} = [\dot{d}_x \quad \dot{d}_y \quad \dot{\phi}]^T$ folgen mit (8.3) zu

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{d}_x \\ \dot{d}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\phi) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} {}^k\mathbf{t}_k^w, \quad (8.7a)$$

$${}^k\mathbf{t}_k^w = \begin{bmatrix} {}^k\mathbf{v}_k^w \\ {}^k\omega_k^w \end{bmatrix} = \begin{bmatrix} \mathbf{R}^T(\phi) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \dot{d}_x \\ \dot{d}_y \\ \dot{\phi} \end{bmatrix}. \quad (8.7b)$$

Die Geschwindigkeit ${}^i\mathbf{v}_i^w$ eines einzelnen Rades i ergibt sich aus der Bewegung des Roboters $\{k\}$ im Inertialkoordinatensystem $\{w\}$, siehe Abb. 8.4. Im Koordinatensystem $\{k\}$ sind die Position ${}^k\mathbf{d}_i = [d_{x,i} \quad d_{y,i}]^T$ und Orientierung β_i der Räder $i = 1, \dots, r$ konstant. Die Koordinatensysteme $\{i\}$ der Räder sind körperfest in $\{k\}$ verankert. Die homogene Transformation des Koordinatensystems $\{i\}$ des Rades i bezüglich des Inertialkoordinatensystems $\{w\}$ lautet damit

$${}^w\mathbf{H}_i = {}^w\mathbf{H}_k {}^k\mathbf{H}_i = \begin{bmatrix} {}^w\mathbf{R}_k & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^k\mathbf{R}_i & {}^k\mathbf{d}_i \\ \mathbf{0} & 1 \end{bmatrix}, \quad (8.8)$$

¹Bei mobilen Robotern mit lenkbaren Rädern ist auch der Lenkwinkel in der Konfiguration \mathbf{q} enthalten.

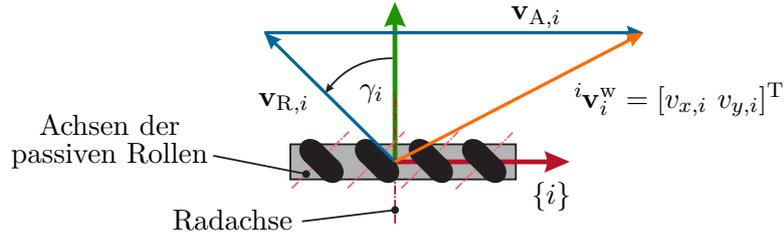


Abbildung 8.5: Aufteilung der Geschwindigkeit ${}^i\mathbf{v}_i^w$ in die Komponenten entlang der Antriebsrichtung $\mathbf{v}_{A,i}$ und der Querrollen $\mathbf{v}_{R,i}$.

woraus die räumliche Geschwindigkeit ${}^i\mathbf{t}_i^w$ gemäß

$${}^i\mathbf{t}_i^w = \begin{bmatrix} {}^i\mathbf{v}_i^w \\ {}^i\omega_i^w \end{bmatrix} = \begin{bmatrix} {}^w\mathbf{R}_i^T {}^w\dot{\mathbf{d}}_i \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} ({}^w\mathbf{R}_k {}^k\mathbf{R}_i)^T ({}^w\dot{\mathbf{d}}_k + {}^w\dot{\mathbf{R}}_k {}^k\mathbf{d}_i) \\ \dot{\phi} \end{bmatrix} \quad (8.9)$$

folgt, vgl. (8.3) und (8.4). Die translatorische Geschwindigkeit des Rades ${}^i\mathbf{v}_i^w$ in (8.9) wird mithilfe der räumlichen Geschwindigkeit des Roboters ${}^k\mathbf{t}_k^w = [({}^k\mathbf{v}_k^w)^T \ \dot{\phi}]^T$ in der Form

$$\begin{aligned} {}^i\mathbf{v}_i^w &= {}^k\mathbf{R}_i^T ({}^w\mathbf{R}_k^T {}^w\dot{\mathbf{d}}_k + {}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k {}^k\mathbf{d}_i) = \\ &= {}^k\mathbf{R}_i^T \left(\underbrace{{}^w\mathbf{R}_k^T {}^w\dot{\mathbf{d}}_k}_{= {}^k\mathbf{v}_k^w} + \underbrace{{}^w\mathbf{R}_k^T {}^w\dot{\mathbf{R}}_k}_{= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}} \dot{\phi} \right) \\ &= {}^k\mathbf{R}_i^T \left({}^k\mathbf{v}_k^w + \begin{bmatrix} -d_{y,i} \\ d_{x,i} \end{bmatrix} \dot{\phi} \right) \end{aligned} \quad (8.10)$$

ausgedrückt.

Im nächsten Schritt wird aus der Geschwindigkeit ${}^i\mathbf{v}_i^w$ eines einzelnen Rades i die Drehwinkelgeschwindigkeit $u_i = \dot{\theta}_i$ dieses Rades berechnet. Abbildung 8.5 zeigt die Aufteilung der Geschwindigkeit ${}^i\mathbf{v}_i^w$ in die Geschwindigkeitsanteile in Antriebsrichtung $\mathbf{v}_{A,i}$ sowie in Richtung der passiven Rollen $\mathbf{v}_{R,i}$. Die Abrollrichtung der passiven Rollen ist um einen Winkel γ_i zur Radachse gedreht. Im Falle eines Allseitenrades gilt $\gamma_i = 0$, beim Mecanum-Rad ist häufig $\gamma_i = 45^\circ$. Die translatorische Gesamtgeschwindigkeit ${}^i\mathbf{v}_i^w$ setzt sich aus den Anteilen $\mathbf{v}_{A,i}$ und $\mathbf{v}_{R,i}$ gemäß

$${}^i\mathbf{v}_i^w = \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix} = \mathbf{v}_{A,i} + \mathbf{v}_{R,i} = v_{A,i} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_{R,i} \begin{bmatrix} -\sin(\gamma_i) \\ \cos(\gamma_i) \end{bmatrix} \quad (8.11)$$

zusammen. Die Beträge $v_{A,i}$ und $v_{R,i}$ der beiden Geschwindigkeitsanteile finden sich durch Umstellung von (8.11) gemäß

$$v_{A,i} = v_{x,i} + v_{y,i} \tan(\gamma_i) \quad (8.12a)$$

$$v_{R,i} = \frac{v_{y,i}}{\cos(\gamma_i)}. \quad (8.12b)$$

Unter der Annahme, dass alle Räder den selben Radius R besitzen, ergibt sich daraus die Winkelgeschwindigkeit u_i des Antriebes zu

$$u_i = \frac{v_{A,i}}{R} = \frac{1}{R} \begin{bmatrix} 1 & \tan(\gamma_i) \end{bmatrix} \begin{bmatrix} v_{x,i} \\ v_{y,i} \end{bmatrix}. \quad (8.13)$$

Schließlich können die Gleichungen (8.7b), (8.10) und (8.13) in der Form

$$u_i = \underbrace{\frac{1}{R} \begin{bmatrix} 1 & \tan(\gamma_i) \end{bmatrix} \begin{bmatrix} \cos(\beta_i) & \sin(\beta_i) \\ -\sin(\beta_i) & \cos(\beta_i) \end{bmatrix}}_{\mathbf{v}_i^T} \begin{bmatrix} 1 & 0 & -d_{y,i} \\ 0 & 1 & d_{x,i} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{R}^T(\phi) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{k}_k^w} \begin{bmatrix} \dot{d}_x \\ \dot{d}_y \\ \dot{\phi} \end{bmatrix} \quad (8.14)$$

zusammengefasst werden. Durch geeignete trigonometrische Umformungen^{2,3} kann $\mathbf{v}_i \in \mathbb{R}^3$, $i = 1, \dots, r$, zu

$$\mathbf{v}_i = \frac{1}{R \cos(\gamma_i)} \begin{bmatrix} \cos(\gamma_i + \beta_i) \\ \sin(\gamma_i + \beta_i) \\ d_{x,i} \sin(\gamma_i + \beta_i) - d_{y,i} \cos(\gamma_i + \beta_i) \end{bmatrix} \quad (8.15)$$

vereinfacht werden. Schließlich werden die Vektoren \mathbf{v}_i zu einer Matrix $\mathbf{V} \in \mathbb{R}^{r \times 3}$ zusammengefasst und damit ergibt sich der Vektor der Antriebsgeschwindigkeiten $\mathbf{u} = [u_1 \dots u_r]^T$ gemäß

$$\mathbf{u} = \underbrace{\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_r^T \end{bmatrix}}_{\mathbf{V}} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{q}}_d. \quad (8.16)$$

An dieser Stelle sein angemerkt, dass die Vektoren \mathbf{v}_i , $i = 1, \dots, r$, aus (8.15) bzw. die Matrix \mathbf{V} aus (8.16) ausschließlich konstante kinematische Parameter des Roboters bezüglich $\{k\}$ enthält, d. h. die Radpositionen ${}^k\mathbf{d}_i = [d_{x,i} \ d_{y,i}]^T$, die Radorientierungen β_i und die Winkeln γ_i für die Abrollrichtung der passiven Rollen, $i = 1, \dots, r$. Bei der Konstruktion eines omnidirektionalen mobilen Roboters müssen diese Parameter so gewählt werden, dass die Matrix \mathbf{V} den Rang 3 besitzt, um jede beliebige Richtung des SE(2) ausführen zu können. Wären bei einem Roboter beispielsweise die Antriebsrichtungen aller Räder gleich und die passiven Abrollrichtungen auch zueinander parallel, also z. B. Abb. 8.3b mit vier identischen Rädern, so hätte \mathbf{V} nur Rang 2. Damit könnten keine aktiven translatorischen Bewegungen entlang der passiven Abrollrichtung erzeugt werden.

² $\sin(\alpha + \beta) = \sin(\alpha) \cos(\beta) + \cos(\alpha) \sin(\beta)$

³ $\cos(\alpha + \beta) = \cos(\alpha) \cos(\beta) - \sin(\alpha) \sin(\beta)$

Beispiel 8.1 (Omnidirektionaler mobiler Roboter mit drei Allseitenrädern). In Abb. 8.6 ist ein omnidirektionaler mobiler Roboter mit drei Allseitenrädern mit dem Radius R gegeben. Die Kinematikparameter sind in Tabelle 8.1 aufgeführt. Damit ergibt sich die Matrix \mathbf{V} aus (8.16) zu

$$\mathbf{V} = \frac{1}{R} \begin{bmatrix} 1 & 0 & -d \\ -\frac{1}{2} & -\sin(\frac{\pi}{3}) & -d \\ -\frac{1}{2} & \sin(\frac{\pi}{3}) & -d \end{bmatrix}.$$

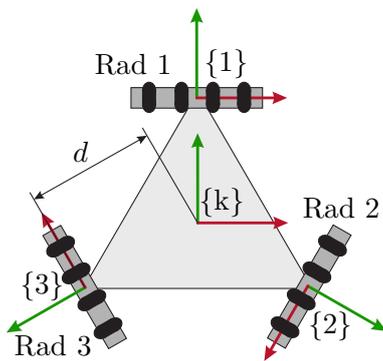


Abbildung 8.6: Omnidirektionaler mobiler Roboter mit drei Allseitenrädern.

i	γ_i	β_i	$d_{x,i}$	$d_{y,i}$
1	0	0	0	d
2	0	$-\frac{2\pi}{3}$	$d \sin(\frac{\pi}{3})$	$-d \cos(\frac{\pi}{3})$
3	0	$\frac{2\pi}{3}$	$-d \sin(\frac{\pi}{3})$	$-d \cos(\frac{\pi}{3})$

Tabelle 8.1: Kinematikparameter.

Beispiel 8.2 (Omnidirektionaler mobiler Roboter mit vier Mecanum-Rädern). In diesem Beispiel wird der omnidirektionale mobile Roboter aus Abb. 8.7 betrachtet. Die Kinematikparameter sind durch Tabelle 8.2 gegeben und der Radius aller Räder ist R . Somit folgt die Matrix \mathbf{V} aus (8.16) zu

$$\mathbf{V} = \frac{1}{R} \begin{bmatrix} 1 & 1 & -l - w \\ 1 & -1 & -l - w \\ 1 & 1 & l + w \\ 1 & -1 & l + w \end{bmatrix}.$$

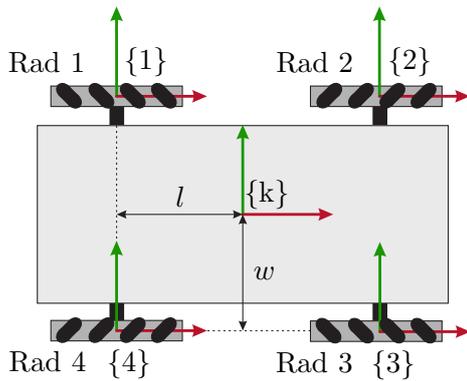


Abbildung 8.7: Omnidirektionaler mobiler Roboter mit vier Mecanum-Rädern.

i	γ_i	β_i	$d_{x,i}$	$d_{y,i}$
1	$\frac{\pi}{4}$	0	$-l$	w
2	$-\frac{\pi}{4}$	0	l	w
3	$\frac{\pi}{4}$	0	l	$-w$
4	$-\frac{\pi}{4}$	0	$-l$	$-w$

Tabelle 8.2: Kinematikparameter.

8.2.2 Planung

Omnidirektionale mobile Roboter erlauben eine instantane Bewegung in jede beliebige Richtung. Daher können die kinematischen Planungsmethoden aus Kapitel 7 direkt angewendet werden.

8.2.3 Regelung

Eine kinematische Trajektorienfolgeregelung für eine gegebene Trajektorie $\mathbf{q}_d(t)$ lässt sich durch

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \mathbf{K}(\mathbf{q} - \mathbf{q}_d) \quad (8.17a)$$

$$\mathbf{u} = \mathbf{V} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{q}}_r \quad (8.17b)$$

mit einer positiv definiten Reglermatrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ realisieren.

8.3 Nichtholonome mobile Roboter

Bei nichtholonomen mobilen Robotern ist die mögliche Geschwindigkeit $\dot{\mathbf{q}} = [\dot{d}_x \ \dot{d}_y \ \dot{\phi}]^T$ durch eine oder mehrere Pfaff'sche Zwangsbedingung, siehe Abschnitt 5.6, eingeschränkt. Die nichtholonomen Zwangsbedingungen ergeben sich aus der Forderung, dass die Räder eines mobilen Roboters ohne seitlichen Versatz in Bewegungsrichtung abrollen⁴. Trotzdem können nichtholonome mobile Roboter durch geeignetes Manövrieren jede beliebige

⁴Weiters wird angenommen, dass die Räder ohne Schlupf abrollen.

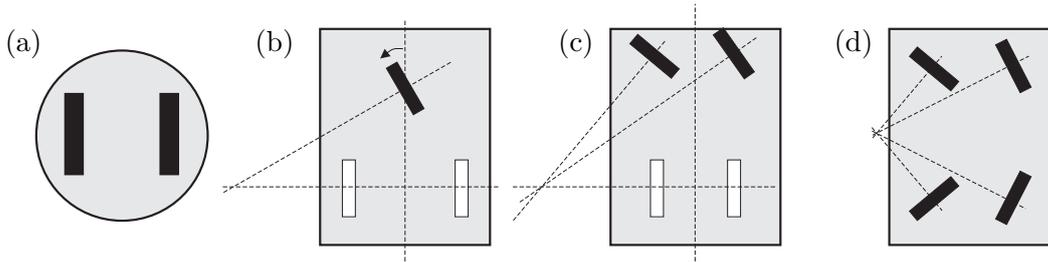


Abbildung 8.8: Typische Kinematiken von nichtholonomen mobilen Robotern mit (a) null, (b) einem, (c) zwei, (d) vier gelenkten Rädern. Angetriebene Räder sind schwarz, Folgeräder sind weiß eingezeichnet.

Konfiguration $\mathbf{q} = [d_x \ d_y \ \phi]^T$ im $SE(2)$ erreichen. Ein bekanntes Beispiel für eine Bewegungsplanung mit nichtholonomen Zwangsbedingungen ist das Parallelparken eines Autos. Auch wenn sich ein Auto in jedem Zeitpunkt nicht orthogonal zur Radstellung bewegen kann, ist das seitliche Einparken durch eine Abfolge geeigneter Lenk- und Fahrbewegungen natürlich durchaus möglich. Abbildung 8.8 zeigt vier typische Radkonfigurationen in mobilen Robotern.

8.3.1 Nichtholonome Zwangsbedingungen

In diesem Abschnitt bezeichnet $\mathbf{q} \in \mathbb{R}^n$ die n -dimensionale Konfiguration bestehend aus der Position ${}^w\mathbf{d}_k = [d_x \ d_y]^T$ und der Orientierung ϕ des Fahrgestells mit dem Koordinatensystem $\{k\}$ sowie den Lenkwinkeln aller gelenkten Räder.

Im Folgenden wird angenommen, dass die Zwangsbedingungen in Form von $k < n$ linear unabhängigen Vektorfunktionen $\mathbf{a}_i(\mathbf{q}) \in \mathbb{R}^{1 \times n}$ bzw. der Matrix $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{k \times n}$ gemäß, vgl. Abschnitt 5.6

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{a}_1(\mathbf{q}) \\ \mathbf{a}_2(\mathbf{q}) \\ \vdots \\ \mathbf{a}_k(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{0} \quad (8.18)$$

gegeben sind. Da die Vektorfunktionen als linear unabhängig angenommen wurden, hat die Matrix $\mathbf{A}(\mathbf{q})$ den Rang k . Durch $\mathbf{A}(\mathbf{q})$ wird die Geschwindigkeit $\dot{\mathbf{q}}$ auf den Nullraum bzw. Kern der Matrix $\mathbf{A}(\mathbf{q})$ eingeschränkt. Im Gegensatz dazu schränken holonome Zwangsbedingungen die Bewegung im *Konfigurationsraum* ein und werden gemäß Abschnitt 5.6 mit einer vektorwertigen Funktion $\phi(\mathbf{q}) \in \mathbb{R}^k$ beschrieben werden.

Für eine gegebene Zwangsbedingung in der Form (8.18) stellt sich nun die Frage, ob es sich um eine holonome oder nichtholonome Zwangsbedingung handelt. Hierfür wird exemplarisch der Fall einer einzelnen Bindung, d.h. $k = 1$ mit $\mathbf{a}_1(\mathbf{q}) = [a_{1,1}(\mathbf{q}) \ a_{1,2}(\mathbf{q}) \ \dots \ a_{1,n}(\mathbf{q})]^T$, betrachtet. Ist die Zwangsbedingung holonom, so muss eine Funktion $\phi(\mathbf{q}) \in \mathbb{R}^1$ existieren, so dass der Zusammenhang, vgl. (5.53)

$$\frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}} = \alpha(\mathbf{q})\mathbf{a}_1(\mathbf{q}) \quad (8.19)$$

mit einem Faktor $\alpha(\mathbf{q}) \neq 0$ erfüllt ist. Dies bedeutet dass eine Funktion $\phi(\mathbf{q})$ passend zur gegebenen Zwangsbedingung $\mathbf{a}_1(\mathbf{q})\dot{\mathbf{q}} = 0$ existieren muss. Da für die skalare, zweifach partiell nach \mathbf{q} differenzierbare Funktion $\phi(\mathbf{q})$ der *Satz von Schwarz* erfüllt sein muss, d. h.

$$\frac{\partial^2 \phi(\mathbf{q})}{\partial q_i \partial q_j} = \frac{\partial^2 \phi(\mathbf{q})}{\partial q_j \partial q_i} \quad (8.20)$$

müssen daher die Bedingungen

$$\frac{\partial(\alpha(\mathbf{q})a_{1,i}(\mathbf{q}))}{\partial q_j} = \frac{\partial(\alpha(\mathbf{q})a_{1,j}(\mathbf{q}))}{\partial q_i} \quad \forall i, j = 1, \dots, n \quad (8.21)$$

erfüllt sein, damit $\mathbf{a}_1(\mathbf{q})\dot{\mathbf{q}} = 0$ für den Fall $k = 1$ eine holonome Zwangsbedingung darstellt. Für ein System von $k > 1$ Zwangsbedingungen $\mathbf{A}(\mathbf{q})$ sind die Integrationsbedingungen allgemeiner⁵ und können schnell relativ komplex werden. Stattdessen ist es in vielen Fällen einfacher, das kinematische Modell direkt über die freien Bewegungsrichtungen anstatt der Bewegungseinschränkung (8.18) zu beschreiben.

Beispiel 8.3. Als Beispiel wird ein System mit einem dreidimensionalen Konfigurationsraum $\mathbf{q} \in \mathbb{R}^3$ und der Zwangsbedingung

$$\mathbf{a}_1(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} 1 & q_1 & 1 \end{bmatrix} \dot{\mathbf{q}} = \dot{q}_1 + q_1 \dot{q}_2 + \dot{q}_3 = 0 \quad (8.22)$$

betrachtet. Aus (8.21) folgen die Bedingungen

$$\begin{aligned} \frac{\partial(\alpha(\mathbf{q})a_{1,1}(\mathbf{q}))}{\partial q_2} = \frac{\partial(\alpha(\mathbf{q})a_{1,2}(\mathbf{q}))}{\partial q_1} &\implies \frac{\partial\alpha(\mathbf{q})}{\partial q_2} = q_1 \frac{\partial\alpha(\mathbf{q})}{\partial q_1} + \alpha(\mathbf{q}), \\ \frac{\partial(\alpha(\mathbf{q})a_{1,1}(\mathbf{q}))}{\partial q_3} = \frac{\partial(\alpha(\mathbf{q})a_{1,3}(\mathbf{q}))}{\partial q_1} &\implies \frac{\partial\alpha(\mathbf{q})}{\partial q_3} = \frac{\partial\alpha(\mathbf{q})}{\partial q_1}, \\ \frac{\partial(\alpha(\mathbf{q})a_{1,2}(\mathbf{q}))}{\partial q_3} = \frac{\partial(\alpha(\mathbf{q})a_{1,3}(\mathbf{q}))}{\partial q_2} &\implies q_1 \frac{\partial\alpha(\mathbf{q})}{\partial q_3} = \frac{\partial\alpha(\mathbf{q})}{\partial q_2}. \end{aligned}$$

Durch Einsetzen der zweiten und dritten Gleichung in die erste wird ersichtlich, dass dieses Gleichungssystem die eindeutige Lösung $\alpha(\mathbf{q}) = 0$ hat. Somit ist die Zwangsbedingung (8.22) nichtholonom.

Bemerkung 8.1. Neben mobilen Robotern kommen nichtholonome Zwangsbedingungen auch in weiteren robotischen Systemen vor. Darunter fallen z. B. künstliche Hände mit abgerundeten Fingern, wobei die nichtholonomen Bindungen durch das Abrollen eines gegriffenen Objektes an den Fingeroberflächen entstehen. Ein weiteres Beispiel für nichtholonome Systeme sind freischwebende orbitale Weltraumroboter, wobei die Drehimpulserhaltung als nichtholonome Zwangsbedingung betrachtet werden kann.

⁵Das *Frobenius-Theorem* in der Differentialgeometrie liefert eine hinreichende und notwendige Bedingung für die Integrierbarkeit einer Verteilung (Bündel von Tangentenvektorfeldern).

8.3.2 Kinematisches Modell

Durch die k Pfaff'schen Zwangsbedingungen wird die Geschwindigkeit $\dot{\mathbf{q}}$ des Systems auf den Nullraum bzw. Kern von $\mathbf{A}(\mathbf{q})$ mit der Dimension $m = n - k$ eingeschränkt. Ist eine Basis für diesen Nullraum durch die Vektorfunktionen $\mathbf{g}_i(\mathbf{q}) \in \mathbb{R}^n$, $i = 1, \dots, m$, bekannt, so gilt $\text{kern}(\mathbf{A}) = \text{span}\{\mathbf{g}_1, \dots, \mathbf{g}_m\}$. Damit kann die zulässige Bewegung als Lösung des dynamischen Systems

$$\dot{\mathbf{q}} = \sum_{i=1}^m \mathbf{g}_i(\mathbf{q}) u_i = \begin{bmatrix} \mathbf{g}_1(\mathbf{q}) & \mathbf{g}_2(\mathbf{q}) & \cdots & \mathbf{g}_m(\mathbf{q}) \end{bmatrix} \mathbf{u} = \mathbf{G}(\mathbf{q}) \mathbf{u} \quad (8.23)$$

dargestellt werden. Analog zu Abschnitt 8.2.1 ist $\mathbf{u} = [u_1 \ u_2 \ \dots \ u_m]^T$ der Systemeingang, über welchen die Geschwindigkeit des Systems entlang der Basisvektoren $\mathbf{g}_i(\mathbf{q})$ vorgegeben werden kann. Die Basisvektoren $\mathbf{g}_i(\mathbf{q})$ – und somit die Matrix $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n \times m}$ – sind nicht eindeutig und die entsprechenden Eingänge u_i haben daher im Allgemeinen unterschiedliche Bedeutungen. Üblicherweise lässt sich aber eine Darstellung finden, in der die Eingänge u_i eine interpretierbare physikalische Bedeutung haben. Da die Vektorfunktionen $\mathbf{g}_i(\mathbf{q})$ eine Basis des Nullraumes von $\mathbf{A}(\mathbf{q})$ bilden, muss in jedem Fall

$$\mathbf{A}(\mathbf{q}) \mathbf{G}(\mathbf{q}) = \mathbf{0} \quad (8.24)$$

gelten. In dieser Darstellung lässt sich die Frage nach der Integrierbarkeit der Zwangsbedingungen (8.18) mit der Erreichbarkeit des dynamischen Systems in Verbindung bringen. Aus dieser Argumentation lassen sich die folgenden Eigenschaften ableiten:

- Ist das System (8.23) vollständig erreichbar, so kann aus einer beliebigen Anfangskonfiguration \mathbf{q}_0 jede beliebige Zielkonfiguration \mathbf{q}_1 über ein geeignetes Stellsignal $\mathbf{u}(t)$ erreicht werden. In diesem Fall existiert eine mit den Zwangsbedingungen kompatible Trajektorie, die zwei beliebige Punkte \mathbf{q}_0 und \mathbf{q}_1 miteinander verbindet. Die Zwangsbedingung ist in diesem Fall nichtholonom, da aus der vorhandenen Einschränkung des Geschwindigkeitsraumes *keine* Einschränkung des Konfigurationsraums folgt.
- Ist nur ein m -dimensionaler Unterraum erreichbar, so sind die Zwangsbedingungen holonom.
- In allen anderen Fällen liegt die Dimension $\nu \in \mathbb{N}$ des erreichbaren Unterraumes im Bereich $m < \nu < n$. Die Zwangsbedingungen enthalten dann eine Kombination aus holonomen und nichtholonomen Zwangsbedingungen.
- Besitzt das System genau $n - 1$ Zwangsbedingungen, dann ist das kinematische System (8.23) nur durch einen einzelnen Basisvektor definiert. Der Roboter kann sich in jeder Konfiguration also nur in genau eine Richtung bewegen. Die $n - 1$ (linear unabhängigen) Zwangsbedingungen sind holonom.
- Sind die Zwangsbedingungen konstant, d.h. unabhängig von der Konfiguration \mathbf{q} , so können auch die Basisvektoren als konstante Vektoren gewählt werden. In diesem Fall ist die gesamte Bewegung des Systems auf eine m -dimensionale Hyperebene des Konfigurationsraum eingeschränkt. Zwangsbedingungen mit konstanten Koeffizienten sind also holonom.

- Ein dynamisches System der Form (8.23) wird *driftfrei* genannt, da die Bewegung für $\mathbf{u} = \mathbf{0}$ unmittelbar zum Stillstand $\dot{\mathbf{q}} = \mathbf{0}$ kommt.

8.3.3 Erreichbarkeit

Um die Erreichbarkeit eines Systems (8.23) zu analysieren, wird untersucht, wie sich das System bei einer zeitlichen Abfolge von Steuerbefehlen entlang der Basisvektoren verhält. Ergibt eine Bewegung in Richtung \mathbf{g}_i gefolgt von einer Bewegung in Richtung \mathbf{g}_j mit $j \neq i$, einen anderen Zielwert als eine Bewegung in umgekehrter Reihenfolge, so hängt die Gesamtbewegung von der Reihenfolge der Bewegungen entlang der Basisvektoren ab. In diesem Fall spricht man davon, dass die Basisvektoren *nicht kommutieren*. Für die Erreichbarkeitsanalyse ist nun interessant, ob durch derartige Abfolgen von Elementarbewegungen neue Bewegungsrichtungen entstehen, die nicht bereits in dem ursprünglichen Satz von Basisvektoren enthalten waren.

Im Folgenden bezeichnet $\mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0) \in \mathbb{R}^n$ die Lösung des Systems $\dot{\mathbf{q}} = \mathbf{g}_i(\mathbf{q})$ zum Zeitpunkt t ausgehend von der Startkonfiguration $\mathbf{q}(0) = \mathbf{q}_0$. Die Differenz

$$\Delta \mathbf{Q} = \mathbf{Q}_\varepsilon^{\mathbf{g}_j}(\mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0)) - \mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{Q}_\varepsilon^{\mathbf{g}_j}(\mathbf{q}_0)) \quad (8.25)$$

beschreibt die Abweichung zwischen einer Abfolge der Bewegung entlang \mathbf{g}_i gefolgt von \mathbf{g}_j und der umgekehrten Reihenfolge, jeweils um ein kleines Zeitintervall ε . Die Berechnung von $\Delta \mathbf{Q}$ kann für kleine Werte von ε durch eine Taylorreihenentwicklung angenähert werden. Dieses Vorgehen ergibt

$$\begin{aligned} \mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0) &= \mathbf{q}(0) + \varepsilon \dot{\mathbf{q}}(0) + \frac{1}{2} \varepsilon^2 \ddot{\mathbf{q}}(0) + \mathcal{O}(\varepsilon^3) \\ &= \mathbf{q}(0) + \varepsilon \mathbf{g}_i(\mathbf{q}(0)) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}} \mathbf{g}_i(\mathbf{q}(0)) + \mathcal{O}(\varepsilon^3) \end{aligned}$$

sowie

$$\begin{aligned} \mathbf{Q}_\varepsilon^{\mathbf{g}_j}(\mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0)) &= \mathbf{Q}_\varepsilon^{\mathbf{g}_j}(\mathbf{q}_0) + \varepsilon \mathbf{g}_j(\mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0)) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{Q}_\varepsilon^{\mathbf{g}_i}(\mathbf{q}_0)) + \mathcal{O}(\varepsilon^3) \\ &= \mathbf{q}(0) + \varepsilon \mathbf{g}_j(\mathbf{q}(0)) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{q}(0)) + \\ &\quad \varepsilon \mathbf{g}_j(\mathbf{q}(0) + \varepsilon \mathbf{g}_i(\mathbf{q}(0))) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{q}(0)) + \mathcal{O}(\varepsilon^3) \\ &= \mathbf{q}(0) + \varepsilon \mathbf{g}_j(\mathbf{q}(0)) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{q}(0)) + \\ &\quad \varepsilon \mathbf{g}_j(\mathbf{q}(0)) + \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_i(\mathbf{q}(0)) + \frac{1}{2} \varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{q}(0)) + \mathcal{O}(\varepsilon^3) . \end{aligned}$$

Für den zweiten Term in $\Delta \mathbf{Q}$ ergibt sich dasselbe Ergebnis mit den vertauschten Indizes i und j . In der Differenz bleibt lediglich der Term $\varepsilon^2 \frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_i(\mathbf{q}(0))$ übrig und man erhält

$$\Delta \mathbf{Q} = \varepsilon^2 \underbrace{\left(\frac{\partial \mathbf{g}_j}{\partial \mathbf{q}} \mathbf{g}_i(\mathbf{q}(0)) - \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}} \mathbf{g}_j(\mathbf{q}(0)) \right)}_{[\mathbf{g}_i, \mathbf{g}_j](\mathbf{q})} + \mathcal{O}(\varepsilon^3) . \quad (8.26)$$

Der Ausdruck $[\mathbf{g}_i, \mathbf{g}_j](\mathbf{q})$ in (8.26) wird als *Lie-Klammer* der Vektorfelder \mathbf{g}_i und \mathbf{g}_j bezeichnet und bildet selbst wieder ein Vektorfeld. Eine Bewegung in Richtung des Vektorfeldes $[\mathbf{g}_i, \mathbf{g}_j](\mathbf{q})$ ist somit möglich, indem man in kurzem Abstand zwischen den beiden Vektorfeldern \mathbf{g}_i und \mathbf{g}_j wechselt.

Um zu ermitteln, ob ein kinematisches System der Form (8.23) erreichbar ist, müssen neben den Vektorfeldern $\mathbf{g}_i(\mathbf{q})$ auch die Vektorfelder in Richtung der Lie-Klammern berücksichtigt werden. Durch Verschachtelung der Lie-Klammer, z. B. $[\mathbf{g}_i, [\mathbf{g}_i, \mathbf{g}_j]](\mathbf{q})$ oder $[[\mathbf{g}_i, \mathbf{g}_j], \mathbf{g}_i](\mathbf{q})$, können noch weitere Richtungen erzeugt werden. Im Allgemeinen müssten hierbei beliebig tief verschachtelte Lie-Klammern ausgewertet werden. Ist es möglich durch Mitnahme dieser zusätzlichen Lie-Klammern eine vollständige Basis des \mathbb{R}^n zu erzeugen⁶, so ist das kinematische System (lokal) erreichbar. Die entsprechenden Zwangsbedingungen sind dann nichtholonom. In der Praxis stellt sich jedoch oft heraus, dass eine tiefere Verschachtelung keine neuen Bewegungsrichtungen ergibt. Dies ist z. B. der Fall, wenn die um die (verschachtelten) Lie-Klammern erweiterte Matrix $\mathbf{G}(\mathbf{q})$ vollen Rang hat. In diesem Falle ist das System lokal erreichbar und die Zwangsbedingungen sind nichtholonom.

Beispiel 8.4 (Einrad-Modell). Abbildung 8.9 zeigt ein auf der Ebene ohne Schlupf abrollendes Einrad. Das Rad bewegt sich in der Ebene mit der Position ${}^w\mathbf{d}_k = [d_x \ d_y]^T$ durch eine reine Abrollbewegung mit einem Lenkwinkel ϕ . Die Konfiguration des Rades lässt sich durch die Koordinaten $\mathbf{q} = [d_x \ d_y \ \phi]^T$ beschreiben. Grundsätzlich könnte auch die Drehbewegung des Rades um die Rollachse als zusätzliche Koordinate ergänzt werden. Diese Koordinate spielt jedoch für die folgende Analyse keine Rolle und wird daher vernachlässigt.

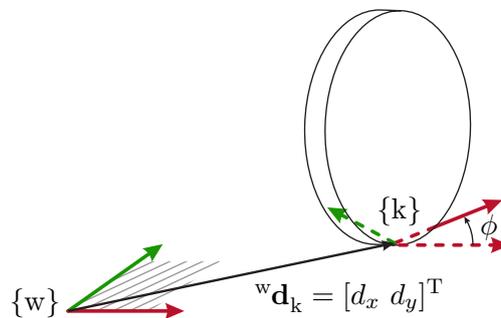


Abbildung 8.9: Auf der Ebene abrollendes Einrad mit der Position des Kontaktpunktes ${}^w\mathbf{d}_k = [d_x \ d_y]^T$ und dem Lenkwinkel ϕ .

Eine Bewegung orthogonal zur Radebene ist nicht möglich. Dies lässt sich durch

⁶Dieses hier etwas informell dargestellte Vorgehen lässt sich durchaus formalisieren und führt auf die Überprüfung der Involutivität der durch die Vektorfunktionen aufgespannten Distribution.

die Zwangsbedingung

$$\dot{d}_x \sin(\phi) - \dot{d}_y \cos(\phi) = \underbrace{\begin{bmatrix} \sin(\phi) & -\cos(\phi) & 0 \end{bmatrix}}_{\mathbf{a}_1(\mathbf{q})} \dot{\mathbf{q}} = 0$$

beschreiben. Für dieses System ist unmittelbar einsichtig, dass die Zwangsbedingung nichtholonom sein muss. Das Rad kann durch eine Abfolge von einer Lenkbewegung, einer Translation, und einer abschließenden Lenkbewegung in jede beliebige Konfiguration gebracht werden. Das kinematische Modell kann durch die beiden Vektorfelder

$$\mathbf{g}_1(\mathbf{q}) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \end{bmatrix}^T, \quad \mathbf{g}_2(\mathbf{q}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

analog zu (8.23) gemäß

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{d}_x \\ \dot{d}_y \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (8.27)$$

beschrieben werden. Es sei angemerkt, dass die Vektorfelder die Bedingungen $\mathbf{a}_1(\mathbf{q})\mathbf{g}_1 = \mathbf{0}$ und $\mathbf{a}_1(\mathbf{q})\mathbf{g}_2 = \mathbf{0}$ erfüllen. Hierbei haben die Eingangsgrößen u_1 und u_2 eine physikalisch interpretierbare Bedeutung. Der Eingang u_1 entspricht der Vorwärtsgeschwindigkeit des Rades und u_2 entspricht der Drehwinkelgeschwindigkeit der Lenkbewegung.

Die Lie-Klammer der Vektorfelder ergibt sich zu

$$[\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \begin{bmatrix} \sin(\phi) & -\cos(\phi) & 0 \end{bmatrix}^T \quad (8.28)$$

und ist immer linear unabhängig von $\mathbf{g}_1(\mathbf{q})$ und $\mathbf{g}_2(\mathbf{q})$. Die um die Lie-Klammer ergänzten Vektorfelder stellen also eine vollständige Basis des Konfigurationsraumes dar.

$$\mathbf{q} \in \mathbb{R}^3 = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]\} = \text{span}\left\{ \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \sin(\phi) \\ -\cos(\phi) \\ 0 \end{bmatrix} \right\} \quad (8.29)$$

Dies bestätigt die Erreichbarkeit des kinematischen Modells und somit die Nichtholonomie der Zwangsbedingung.

Ferner kann das Modell des Einrades auch als Ersatzmodell für einen Roboter mit differentiellem Antrieb (Abb. 8.8a) herangezogen werden, indem die Vorwärtsgeschwindigkeiten der beiden Räder v_R und v_L in die Eingangsgrößen $u_1 = (v_R + v_L)/2$ und $u_2 = (v_R - v_L)/d$ transformiert werden, wobei d der Abstand der beiden Räder ist.

Die Ergebnisse aus Beispiel 8.4 werden im Folgenden nochmals grafisch dargestellt

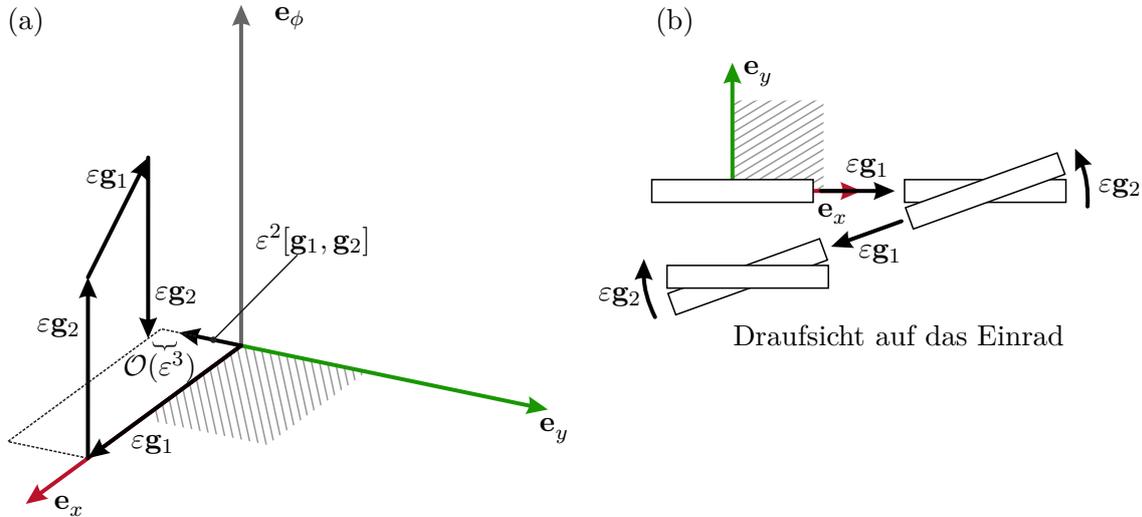


Abbildung 8.10: Visuelle Darstellung der Lie-Klammer am Beispiel des Einrad-Modells aus Beispiel 8.4. (a) Bewegungen im 3-dimensionalen Konfigurationsraum $\mathbf{q} = [d_x \ d_y \ \phi]^T \in \mathbb{R}^3$ mithilfe der Vektorfelder $\mathbf{g}_1(\mathbf{q})$ und $\mathbf{g}_2(\mathbf{q})$, (b) Posens des Einrades während der Bewegungen.

und anhand von (8.25) diskutiert. Abbildung 8.10a zeigt eine 3-dimensionale Darstellung des Konfigurationsraums $\mathbf{q} = [d_x \ d_y \ \phi]^T$ des Systems (8.27). In dieser Darstellung ist die Vorwärtsbewegung des Roboters $\mathbf{g}_1(\mathbf{q})$ eine Bewegung in der xy -Ebene entlang der momentanen Orientierung, während die Drehbewegung mit $\dot{\phi}$ durch einen vertikalen Pfeil entlang $\mathbf{g}_2(\mathbf{q})$ parallel zu e_ϕ entspricht.

Der Roboter startet in der Konfiguration $\mathbf{q} = \mathbf{0}$. Durch die Abfolgen von Bewegungen $\varepsilon\mathbf{g}_1$, $\varepsilon\mathbf{g}_2$, $-\varepsilon\mathbf{g}_1$ und $-\varepsilon\mathbf{g}_2$ wird aber eine andere Konfiguration $\mathbf{q} \neq \mathbf{0}$ erreicht, da die beiden Basisrichtungen $\mathbf{g}_1(\mathbf{q})$ und $\mathbf{g}_2(\mathbf{q})$ nicht kommutieren. Die resultierende Bewegung im Konfigurationsraum wird entsprechend durch den Ausdruck (8.25) mit der Lie-Klammer $[\mathbf{g}_1, \mathbf{g}_2](\mathbf{q})$ und Termen höherer Ordnung $\mathcal{O}(\varepsilon^3)$ beschrieben.

In Abb. 8.10 wird die Pose des Einrades nach den jeweiligen Teilbewegungen dargestellt. Darin ist ersichtlich, dass der Roboter durch die Abfolge von Bewegungen eine neue Pose erreicht, welche wiederum $\phi = 0$ aufweist, aber deutlich entlang der negativen e_y -Richtung verschoben ist – eine Richtung, die durch die Einzelbewegungen $\mathbf{g}_1(\mathbf{q})$ und $\mathbf{g}_2(\mathbf{q})$ nicht realisierbar ist. Diese negative e_y -Richtung wird (hauptsächlich) durch die Lie-Klammer $[\mathbf{g}_1, \mathbf{g}_2](\mathbf{q})$ beschrieben, siehe (8.26).

Beispiel 8.5 (Einspur-Modell). In diesem Beispiel wird das kinematische Modell aus Abb. 8.11 betrachtet. Im Folgenden wird angenommen, dass das Vorderrad gelenkt und angetrieben ist, während die Achse des Hinterrades fest mit dem Fahrgestell verbunden ist. Das System kann bei einem gegebenen Lenkwinkel ψ eine Drehung um den Punkt C , den sogenannten *Momentanpol der Bewegung*, ausführen. Durch Verstellen des Lenkwinkels lässt sich die Lage des Momentanpols verändern. Dieses Modell kann z. B. als Ersatzmodell für die in Abb. 8.8b und Abb. 8.8c gezeigten

Kinematiken dienen.

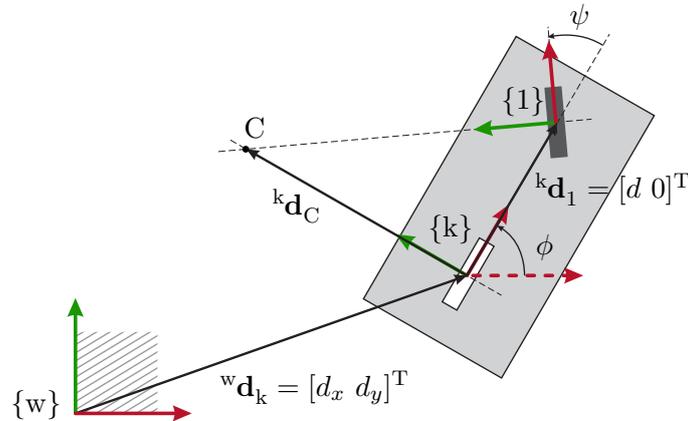


Abbildung 8.11: Koordinaten für ein Robotermodell mit einem gelenkten Rad (sog. „Einspur-Modell“).

Neben der Position ${}^w\mathbf{d}_k = [d_x \ d_y]^T$ und Orientierung ϕ des Fahrgestells, wird nun auch der Lenkwinkel ψ als Koordinate im Konfigurationsraum $\mathbf{q} = [d_x \ d_y \ \phi \ \psi]^T$ benötigt. Durch die beiden Räder sind die beiden Zwangsbedingungen

$$\dot{d}_{x,1} \sin(\phi + \psi) - \dot{d}_{y,1} \cos(\phi + \psi) = 0 \quad (8.30a)$$

$$\dot{d}_x \sin(\phi) - \dot{d}_y \cos(\phi) = 0 \quad (8.30b)$$

gegeben, wobei ${}^w\mathbf{d}_1 = [d_{x,1} \ d_{y,1}]^T$ die Position des Vorderrades im Inertialkoordinatensystem $\{w\}$ beschreibt. Das Vorderrad mit dem Koordinatensystem $\{1\}$ ist gegenüber dem körperfesten Koordinatensystem $\{k\}$ um den Abstandsvektor ${}^k\mathbf{d}_1 = [d \ 0]^T$ verschoben. Die Position des Vorderrades ${}^w\mathbf{d}_1$ wird mithilfe der homogenen Transformation ${}^w\mathbf{H}_k$ und dem homogenen Abstandsvektor zu ${}^k\mathbf{d}_1$ über den Zusammenhang, siehe (3.24)

$$\begin{bmatrix} d_{x,1} \\ d_{y,1} \\ 1 \end{bmatrix} = {}^w\mathbf{H}_k \begin{bmatrix} {}^k\mathbf{d}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\phi) & {}^w\mathbf{d}_k \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^k\mathbf{d}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} d_x + d \cos(\phi) \\ d_y + d \sin(\phi) \\ 1 \end{bmatrix}$$

berechnet. Somit kann die Zwangsbedingung (8.30a) in den Koordinaten des Konfigurationsraums \mathbf{q} in der Form

$$\dot{d}_x \sin(\phi + \psi) - \dot{d}_y \cos(\phi + \psi) - d\dot{\phi} \cos \psi = 0 \quad (8.31)$$

geschrieben werden. Die Matrix $\mathbf{A}(\mathbf{q})$ der Zwangsbedingungen ergibt sich nun aus (8.30b) und (8.31) zu

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \sin(\phi) & -\cos(\phi) & 0 & 0 \\ \sin(\phi + \psi) & -\cos(\phi + \psi) & -d \cos(\psi) & 0 \end{bmatrix}.$$

Der Nullraum der Matrix $\mathbf{A}(\mathbf{q})$ hat eine Dimension 2. Die zulässigen Geschwindigkeiten können z. B. in der Form

$$\begin{bmatrix} \dot{d}_x \\ \dot{d}_y \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) \cos(\psi) \\ \sin(\phi) \cos(\psi) \\ \sin(\psi)/d \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2$$

geschrieben werden. Hierbei stellen u_1 und u_2 die Vorwärts- und Lenkgeschwindigkeit des Vorderrades dar.

Aus den Vektorfeldern $\mathbf{g}_1(\mathbf{q})$ und $\mathbf{g}_2(\mathbf{q})$ können die Lie-Klammern

$$[\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \begin{bmatrix} \cos(\phi) \sin(\psi) \\ \sin(\phi) \sin(\psi) \\ -\cos(\psi)/d \\ 0 \end{bmatrix}$$

und

$$[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]](\mathbf{q}) = \begin{bmatrix} -\sin(\phi)/d \\ \cos(\phi)/d \\ 0 \\ 0 \end{bmatrix}$$

gebildet werden. Mit diesen beiden Lie-Klammern erhält man eine vollständige Basis des Konfigurationsraumes, womit die Nichtholonomie der Zwangsbedingungen (8.30) nachgewiesen ist.

8.3.4 Planung

Grundsätzlich lassen sich viele der Methoden aus Kapitel 7 auch auf nichtholonome mobile Roboter anwenden. Da der Konfigurationsraum mobiler Roboter eine relativ kleine Dimension hat, eignen sich auch gitterbasierte Methoden. Auch Sampling-Methoden wie z. B. RRT und PRM werden oft für die kollisionsfreie kinematische Bahnplanung mobiler Roboter eingesetzt.

Für die Trajektorienplanung aus Abschnitt 7.1 müssen jedoch zusätzliche Nebenbedin-

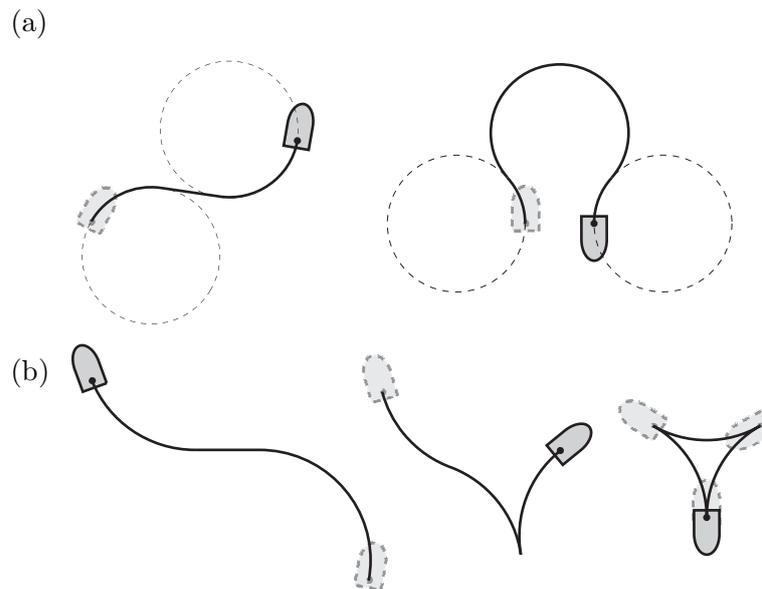


Abbildung 8.12: Beispiele für die Trajektorienplanung für nichtholonome mobile Roboter. Kürzeste Pfade für einen mobilen Roboter mit gelenktem Vorderrad, (a) welcher nur vorwärtsfahren kann, (b) mit Vorwärts- und Rückwärtsfahrt.

dungen ergänzt werden, um nichtholonome Zwangsbedingungen zu berücksichtigen. Dies bedeutet z. B. für einen mobilen Roboter mit differentiellem Antrieb (Abb. 8.8a), dass die Orientierung ϕ immer an die Bewegungsrichtung angepasst werden muss und nicht unabhängig geplant werden kann.

Die Erzeugung optimaler Trajektorien hinsichtlich unterschiedlicher Kostenfunktionen erfordert eine Behandlung mittels nichtlinearer Optimierungstheorie. Für viele praktisch relevante Klassen mobiler Roboter können auf diese Weise auch qualitative Eigenschaften optimaler Trajektorien hergeleitet werden. So lässt sich z. B. zeigen, dass der kürzeste Pfad eines mobilen Roboters mit einem gelenktem Vorderrad (wie in Abb. 8.11), welcher nur vorwärtsfahren kann, aus einer Sequenz von Geraden und Kurven mit maximalem Lenkwinkel besteht, siehe Abb. 8.12a. Werden diese Bahnen mit der jeweils maximalen Geschwindigkeit abgefahren, so erhält man in diesem Fall auch zeitoptimale Trajektorien. Ist eine Rückwärtsfahrt für diesen Roboter erlaubt, so entstehen stückweise Pfade durch das Reversieren des Fahrzeuges, siehe Abb. 8.12b. Auch bei differentiell angetriebenen mobilen Robotern besteht der kürzeste Pfad aus einer Sequenz von reinen Drehbewegungen und geradlinigen Bewegungen. Hierbei kann sich der kürzeste Pfad von der zeitoptimalen Trajektorie unterscheiden.

8.3.5 Regelung

Die nichtholonomen Zwangsbedingungen haben eine deutliche Auswirkung auf die regelungstechnischen Eigenschaften mobiler Roboter. Während bei Manipulatoren mit serieller kinematischer Kette die Stabilisierung eines Arbeitspunktes als Sonderfall der Trajektorienfolgeregelung aufgefasst werden kann, bestehen für mobile Roboter wesentli-

che Unterschiede. Der folgende Satz beschreibt eine grundlegende Einschränkung in der Stabilisierung einer Ruhelage.

Satz 8.1. *Für ein driftfreies kinematisches System mit einem n -dimensionalen Konfigurationsraum $\mathbf{q} \in \mathbb{R}^n$ der Form $\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u}$ mit $\text{rang}(\mathbf{G}(\mathbf{q})) < n \forall \mathbf{q} \in \mathbb{R}^n$ existiert kein stetiges zeitinvariantes Regelgesetz der Form $\mathbf{u} = \mathbf{u}(\mathbf{q})$ das in der Lage ist, einen gewünschten Arbeitspunkt \mathbf{q}_d asymptotisch zu stabilisieren^a.*

^aEs lässt sich zeigen, dass für ein derartiges System die notwendigen Bedingungen des Brockett-Theorems zur stetigen Stabilisierbarkeit nichtlinearer Systeme nicht erfüllt sind.

Aus Satz 8.1 folgt, dass jedes Regelgesetz, das einen nichtholonomen mobilen Roboter in einem Punkt asymptotisch stabilisieren soll, entweder Unstetigkeiten oder eine zeitvariable Komponente $\mathbf{u}(\mathbf{q}, t)$ beinhalten muss. Somit bezieht sich Satz 8.1 explizit nicht auf das Trajektorienfolgeproblem, in dem die Zeitvarianz durch die zeitabhängige Referenztrajektorie gegeben ist. Eine lokale Trajektorienfolgeregelung kann z. B. durch Linearisierung des Modells entlang der Referenztrajektorie aufgebaut werden. Außerdem sei angemerkt, dass sich Satz 8.1 auf die asymptotische Stabilität bezieht, während diese Einschränkung für nicht-asymptotische Stabilität in dieser Form nicht gilt.

Für weiterführende Informationen zur nichtlinearen Regelung mobiler Roboter wird auf die Literatur [8.3–8.6] verwiesen.

8.4 Odometrie

Bei Robotern mit feststehender Basis kann die Konfiguration \mathbf{q} üblicherweise direkt gemessen werden. Hingegen besitzen mobile Roboter ihre propriozeptiven Sensoren in den Rädern, während die Gesamtbewegung des Fahrwerks aus diesen Messungen ermittelt werden muss. Die Berechnung der absoluten Bewegung des Fahrwerkes aus den Radgeschwindigkeiten wird als *Odometrie* bezeichnet.

Im Falle von omnidirektionalen Robotern wird die Odometrie aus (8.16) ermittelt, indem die Matrix \mathbf{V} (pseudo-)invertiert wird, um eine Schätzung für die Geschwindigkeit des Fahrzeuges zu erhalten. Diese Geschwindigkeit wird dann zur absoluten Position und Orientierung aufintegriert. Im Falle von nichtholonomen mobilen Robotern kann die Odometrie in ähnlicher Weise aus dem kinematischen Modell (8.23) berechnet werden.

Aufgrund von Unsicherheiten wie Schlupf, seitlichem Rutschen der Räder, sowie numerischer Fehler in der Integration kommt es in der Odometrie über einen längeren Zeitraum jedoch zu einer oftmals erheblichen Drift. Daher wird die Odometrieinformation in der Praxis entweder von Zeit zu Zeit re-initialisiert oder mit exterozeptiven Sensorinformationen fusioniert. Hierfür kommen typischerweise Kalman-Filter oder Partikelfilter zum Einsatz.

8.5 Literatur

- [8.1] SuperDroid Robots, *PROGRAMMABLE TRIANGULAR OMNI WHEEL VECTORING ROBOT*, (8. Jänner, 2024). Adresse: <https://www.superdroidrobots.com/>.
- [8.2] PAL Robotics, *TIAGO OMNI BASE*, (8. Jänner, 2024). Adresse: <https://www.pal-robotics.com/>.
- [8.3] K. M. Lynch und F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, Cambridge, 2017. Adresse: <http://hades.mech.northwestern.edu/images/7/7f/MR.pdf>.
- [8.4] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [8.5] B. Siciliano und O. Khatib, Hrsg., *Springer Handbook of Robotics*. Springer, 2008. Adresse: <https://link.springer.com/book/10.1007/978-3-540-30301-5>.
- [8.6] C. de Wit, B. Siciliano und G. Bastin, Hrsg., *Theory of Robot Control (Communications and Control Engineering)*. Springer-Verlag, 1996, ISBN: 3-540-76054-7.