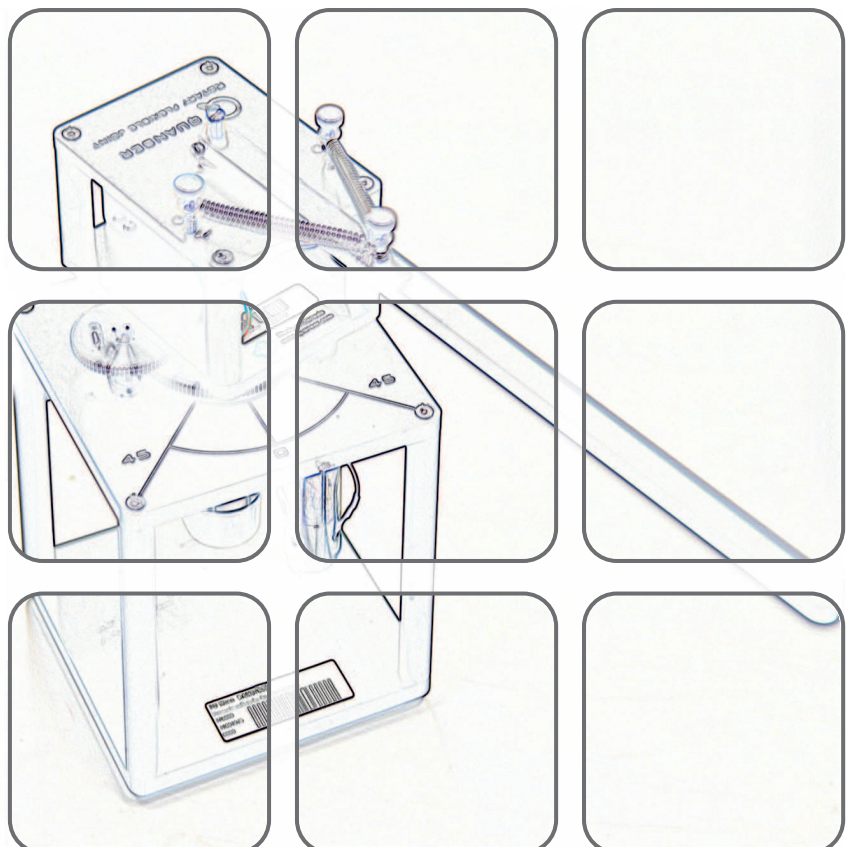


# AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Fachvertiefung  
WS 2021/2022

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



## **Automatisierungs- und Regelungstechnik**

Fachvertiefung  
WS 2021/2022

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

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

<b>1</b>	<b>Grundbegriffe der Robotik</b>	<b>1</b>
1.1	Freiheitsgrade eines Roboters . . . . .	1
1.1.1	Gelenke . . . . .	2
1.1.2	Kinematische Ketten . . . . .	3
1.1.3	Formel von Grübler . . . . .	4
1.1.4	Generalisierte Koordinaten . . . . .	6
1.2	Kinematische Räume . . . . .	6
1.2.1	Konfigurationsraum . . . . .	6
1.2.2	Arbeitsraum . . . . .	6
1.2.3	Vorwärtskinematik und inverse Kinematik . . . . .	8
1.3	Reglerstrukturen . . . . .	9
1.3.1	Regelung im Konfigurationsraum . . . . .	9
1.3.2	Regelung im Arbeitsraum . . . . .	9
1.4	Absolutgenauigkeit und Wiederholgenauigkeit . . . . .	10
1.5	Literatur . . . . .	12
<b>2</b>	<b>Roboterkinematik</b>	<b>13</b>
2.1	Vorwärtskinematik . . . . .	13
2.1.1	Homogene Transformationen . . . . .	13
2.1.2	Serielle kinematische Ketten . . . . .	15
2.1.3	Parametrierung der Orientierung . . . . .	17
2.1.4	6-Achs-Roboter mit sphärischem Handgelenk . . . . .	19
2.2	Inverse Kinematik . . . . .	21
2.2.1	Geschlossene analytische Lösungen . . . . .	21
2.2.2	Numerische Lösungen . . . . .	23
2.2.3	Beispiele . . . . .	23
2.3	Differentielle Kinematik . . . . .	27
2.3.1	Analytische und geometrische Manipulator Jacobi-Matrix . . . . .	27
2.3.2	Singularitäten . . . . .	31
2.3.3	Manipulierbarkeitsellipsoide . . . . .	35
2.3.4	Manipulierbarkeit . . . . .	37
2.3.5	Statik . . . . .	37
2.4	Differentielle inverse Kinematik . . . . .	39
2.4.1	Kinematisch nicht-redundante Roboter $n = m$ . . . . .	39
2.4.2	Kinematisch redundante Roboter $n > m$ . . . . .	41
2.4.3	Höhere Ableitungen . . . . .	43
2.5	Literatur . . . . .	45

<b>3</b>	<b>Trajektorienplanung</b>	<b>45</b>
3.1	Pfade und Trajektorien . . . . .	45
3.2	Geradlinige Pfade . . . . .	46
3.3	Zeitparametrierung . . . . .	48
3.3.1	Polynom 3. Ordnung . . . . .	49
3.3.2	Polynom 5. Ordnung . . . . .	50
3.3.3	Trapezförmige Bewegungsprofile . . . . .	51
3.4	Trajektorien mit Via-Punkten . . . . .	52
3.5	Literatur . . . . .	57
<b>4</b>	<b>Roboterregelung</b>	<b>55</b>
4.1	Konfigurationsraum . . . . .	56
4.1.1	Dynamisches Modell im Konfigurationsraum . . . . .	56
4.1.2	Dezentrale Regler . . . . .	56
4.1.3	PD-Regler mit Gravitationskompensation . . . . .	57
4.1.4	Computed-Torque-Regler . . . . .	58
4.2	Arbeitsraum . . . . .	60
4.2.1	Dynamisches Modell im Arbeitsraum . . . . .	60
4.2.2	Computed-Torque-Regler . . . . .	61
4.2.3	Impedanzregelung . . . . .	62
4.2.4	Kinematisch redundante Roboter $n > m$ . . . . .	63
4.3	Implementierung . . . . .	63
4.4	Literatur . . . . .	70

# 1 Grundbegriffe der Robotik

Dieses einleitende Kapitel beschäftigt sich mit einer Reihe von Grundbegriffen in der Robotik, welche in den weiteren Abschnitten dieses Kapitels im Detail beschrieben werden.

Ein Roboter – auch *Manipulator* genannt – ist ein mechanisches System bestehend aus einer Reihe von Körpern, den *Gliedern* (engl. *Links*), welche durch unterschiedliche Typen von *Gelenken* (engl. *Joints*) miteinander verbunden sind. Die Glieder des Roboters werden mithilfe von *Aktuatoren* wie z. B. Elektromotoren oder Hydraulikzylinder bewegt. Eines der Glieder wird als *Roboterbasis* bezeichnet und dieses kann entweder inertialfest, wie z. B. bei einem klassischen Industrieroboter oder einem Turmdrehkran, aber auch mobil sein, beispielsweise bei einem autonomen Fahrzeug oder einem humanoiden Roboter. Im Rahmen dieser Vorlesung werden Robotersysteme mit inertialfester Basis behandelt und die Glieder des Roboters werden als *Starrkörper* betrachtet.

## 1.1 Freiheitsgrade eines Roboters

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 1.1. Wird diese Münze aber auf einen Tisch gelegt, so entstehen durch den Tisch drei Zwangsbedingungen: Es wird ein translatorischer Freiheitsgrad und zwei rotatorische Freiheitsgrade 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

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

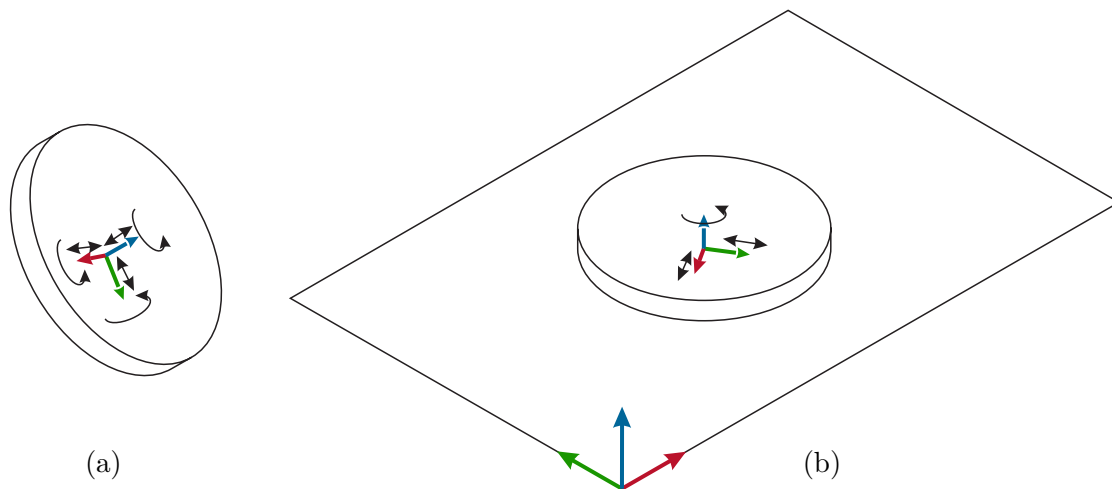


Abbildung 1.1: Freiheitsgrade einer Münze: (a) Münze ohne Zwangsbedingungen mit 6 Freiheitsgraden, (b) Münze auf einem Tisch mit 3 Freiheitsgraden.

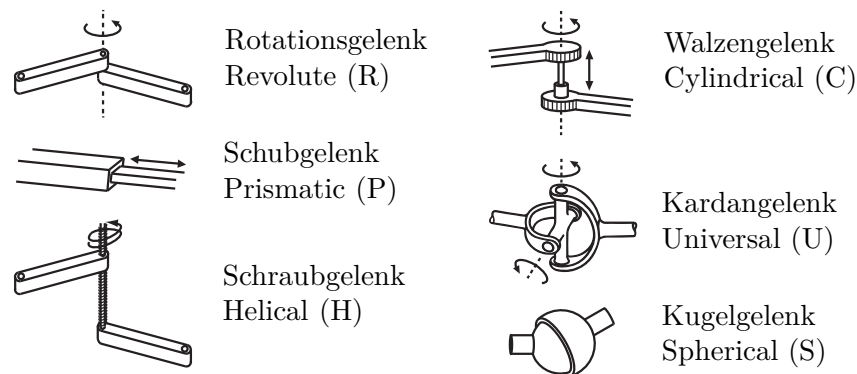


Abbildung 1.2: Unterschiedliche Gelenkstypen [1.1].

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.

### 1.1.1 Gelenke

In Abbildung 1.2 sind einige Gelenkstypen dargestellt, welche häufig in Robotern verbaut werden. Das *Rotationsgelenk* (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

Gelenkstyp	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.

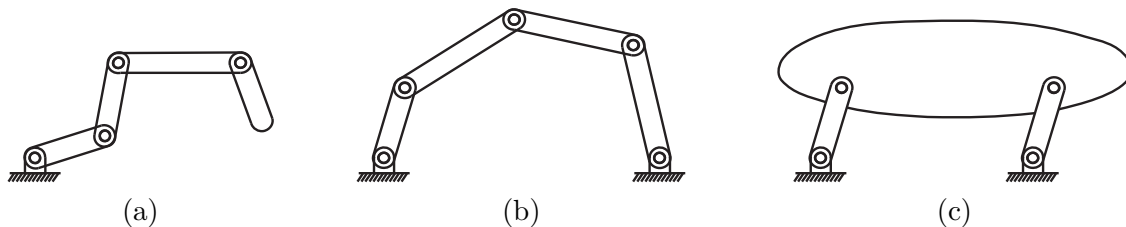


Abbildung 1.3: Kinematische Ketten [1.1]: (a) Offene bzw. serielle Kette, (b) geschlossene Kette, (c) Parallelkinematik.

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. *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 ein 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.

### 1.1.2 Kinematische Ketten

Werden mehrere Glieder durch Gelenke zu einem Mechanismus verbunden, so bilden diese eine *kinematische Kette*. Eine kinematische Kette wird als *offen* oder *seriell* bezeichnet, wenn sie frei von Schleifen ist. Das heißt, sie lässt sich als Baumstruktur darstellen, wie das Beispiel in Abbildung 1.3a zeigt. Eine *geschlossene* kinematische Kette enthält eine oder

mehrere Schleifen, siehe Abbildung 1.3b. Hier ist zu beachten, dass auch der inertialfeste Boden (schraffiert dargestellt) als Teil des Mechanismus zählt.

Ein Spezialfall der geschlossenen Kinematik stellt die *Parallelkinematik* (Abbildung 1.3c) dar, bei der zwei Teile einer kinematischen Kette durch eine Parallelogrammstruktur miteinander verbunden sind. Roboter mit Parallelkinematik 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.

### 1.1.3 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*

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

*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 1.1.2 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 1.4a 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 1.4b zeigt einen Schieber–Kurbel-Me-



chanismus, 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 1.4c, hat eine starre Plattform (oben) und eine bewegte Plattform (unten), welche durch drei Arme mit Parallelkinematik verbunden sind. Jeder Arm 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 Armen 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.

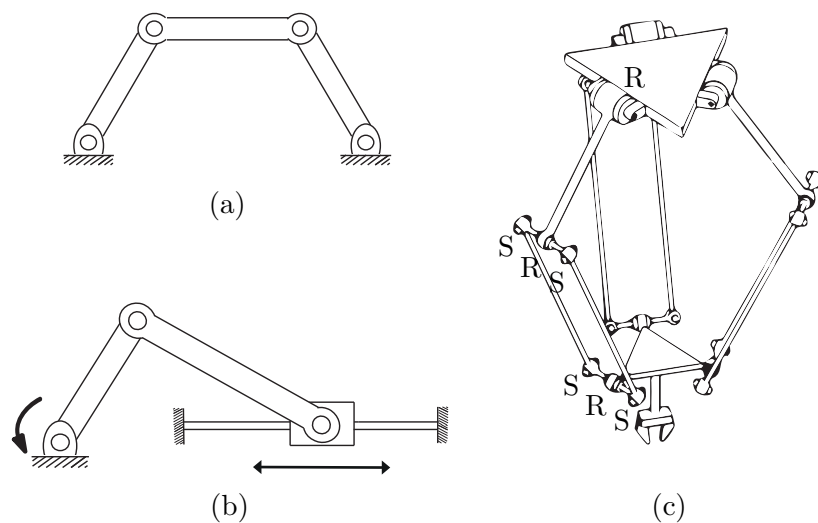


Abbildung 1.4: Beispiele für kinematische Ketten [1.1]: (a) Vier-Balken-Kette (der inertialfeste Boden ist einer der Balken), (b) Schieber-Kurbel-Mechanismus, (c) Delta-Roboter.

### 1.1.4 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 1.1.1. 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. Die unabhängigen Koordinaten  $\mathbf{q}$  werden *generalisierte Koordinaten* genannt und alle Bewegungen zufolge dieser Koordinaten erfüllen stets alle Zwangsbedingungen.

## 1.2 Kinematische Räume

Ein Roboter mit serieller Kinematik weist am letzten Glied meist einen standardisierten Flansch auf, an den ein sogenannter *Endeffektor* – beispielsweise ein Werkzeug oder ein Greifer – montiert wird. Mit dem Endeffektor interagiert der Roboter mit seiner Umgebung. 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 1.1) relativ zum Roboterflansch charakterisiert.

Die Bewegung des Roboters bzw. des Endeffektors wird in unterschiedlichen kinematischen Räumen beschrieben, nämlich dem *Konfigurationsraum* und dem *Arbeitsraum*. Die mathematische Verbindung zwischen diesen beiden Räumen wird durch die *Vorwärtskinematik* und die *inverse Kinematik* hergestellt.

### 1.2.1 Konfigurationsraum

Zunächst werden die beiden Begriffe *Konfiguration* (engl. *Configuration*) und *Konfigurationsraum* (engl. *Configuration Space*, C-Space) formal definiert.

**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.

### 1.2.2 Arbeitsraum

Der Arbeitsraum (engl. *Task Space*) dient der Beschreibung des Endeffektors bzw. des TCP, während sich der Konfigurationsraum auf die Beschreibung des gesamten Roboters

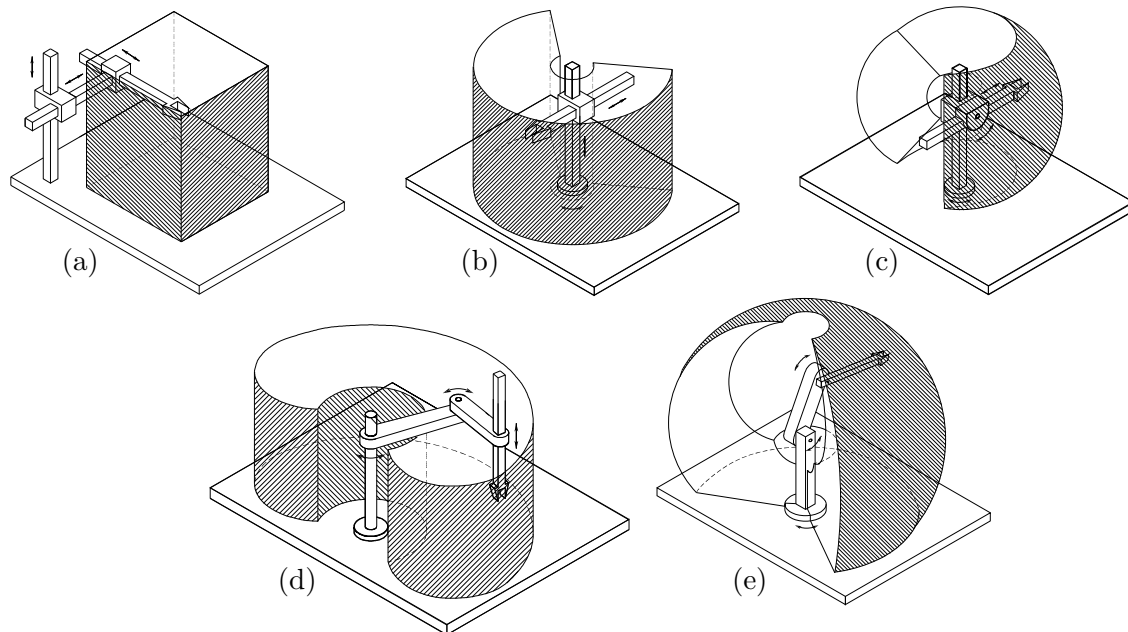


Abbildung 1.5: Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsbereichen [1.2]: (a) Kartesischer Roboter (PPP-Roboter), (b) Roboter mit zylindrischem Arbeitsbereich (RPP-Roboter), (c) Roboter mit kugelförmigem Arbeitsbereich (RRP-Roboter), (d) SCARA-Roboter (Selective Compliance Assembly Robot Arm, RRP-Roboter), (e) Kinematik eines klassischen Industrieroboters mit drei Achsen (RRR-Roboter).

mit all seinen Starrkörpern bezieht. Der Arbeitsraum hat im Folgenden die Dimension  $m$  und wird im Allgemeinen so gewählt, dass die Aufgabe des Roboters geeignet beschrieben werden kann. Beispielsweise für das Zeichnen mit einem Stift auf ein Blatt Papier ist  $m = 2$  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  $m = 6$ .

Der *Arbeitsbereich* (engl. *Work Space*) eines Roboters umfasst nun den gesamten Bereich, den der Roboter mit seinem Endeffektor bzw. TCP erreichen kann. Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsbereichen sind in Abbildung 1.5 dargestellt.

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

Falls der Roboter mehr Freiheitsgrade besitzt als im Arbeitsraum benötigt, 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 Pose des Endeffektors im Arbeitsraum ändert.

### 1.2.3 Vorwärtskinematik und inverse Kinematik

Mit der *Vorwärtskinematik* wird für eine  $n$ -dimensionale Roboterkonfiguration  $\mathbf{q}$  aus dem Konfigurationsraum die  $m$ -dimensionale Pose  $\mathbf{x}_e$  im Arbeitsraum des Endeffektors bzw. TCP berechnet, d. h.

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \quad (1.3)$$

mit der (im Allgemeinen nichtlinearen) Funktion  $\mathbf{f}(\cdot)$ . Sie gibt also an, wie der Endeffektor  $\mathbf{x}_e$  im Arbeitsraum steht, wenn der Roboter eine bestimmte Konfiguration  $\mathbf{q}$  einnimmt. Analog wird die *inverse Kinematik* als *formale* Umkehrung von (1.3) in der Form

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_e) \quad (1.4)$$

eingeführt. Hier wird eine Pose  $\mathbf{x}_e$  für den Endeffektor vorgegeben und die zugehörige Roboterkonfiguration  $\mathbf{q}$  ermittelt. An dieser Stelle sei erwähnt, dass sowohl (1.3) als auch (1.4) im Allgemeinen nicht als geschlossene Funktion oder in analytischer Form angeschrieben werden können. Dies wird in Kapitel 2 näher beschrieben.

Die Lösbarkeit und Eindeutigkeit von (1.3) und (1.4) hängt von der Art der kinematischen Kette ab: Offene kinematische Ketten haben eine eindeutige und immer lösbare Vorwärtskinematik, während die inverse Kinematik nur für Punkte innerhalb des Arbeitsbereichs des Roboters lösbar ist und eine, mehrere oder unendlich viele Lösungen liefert. Andererseits ist die inverse Kinematik von geschlossenen Ketten (im Arbeitsraum) eindeutig lösbar, während die Vorwärtskinematik mehrere oder unendlich viele Lösungen hat.

Die inverse Kinematik von seriellen kinematischen Ketten kann auf zwei unterschiedliche Arten gelöst werden: analytisch und numerisch. Eine analytische inverse Kinematik kann nur für relativ einfache oder für spezielle kinematische Ketten formuliert werden. So gibt es beispielsweise eine analytische Lösung für die inverse Kinematik von klassischen 6-Achs-Industrierobotern und auch von speziellen 7-Achs-Robotern, z.B. dem KUKA LBR iiwa. Im Vergleich zu numerischen Lösungen sind analytische Ansätze sehr effizient, schnell und liefern meist alle möglichen Roboterkonfigurationen zu einer gegebenen Endeffektorpose. Auf der anderen Seite gewinnen numerische Algorithmen durch die heute verfügbare Rechenleistung zunehmend an Bedeutung. Dabei wird (1.4) als Nullstellensuche aufgefasst und mit iterativen Verfahren gelöst. Dies ist für beliebig komplexe Roboterkinematiken möglich. Allerdings zieht dies eine Reihe von Nachteilen mit sich: Iterative Algorithmen benötigen einen Startwert für die erste Iteration, es wird durch die Iterationen mehr Rechenzeit benötigt und die numerische Stabilität hängt von vielen Faktoren ab. Schließlich liefern solche Methoden nur eine einzelne Lösung pro Startwert.

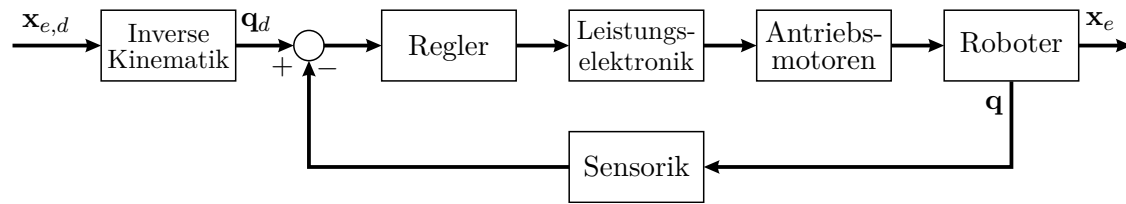


Abbildung 1.6: Regelung im Konfigurationsraum [1.2].

## 1.3 Reglerstrukturen

Die Aufgabe der Regelkreise eines Roboters besteht darin, mit dem Endeffektor möglichst genau – in Zeit und Ort – eine gewünschte vorgegebene Bewegung auszuführen oder den Endeffektor auf eine bestimmte Pose im Raum zu stabilisieren. Als Stellgrößen stehen dabei typischerweise die Motormomente in den Antriebsmotoren der Gelenke zur Verfügung. Das heißt, dass die Regelungsaufgabe im Allgemeinen im Arbeitsraum spezifiziert wird, wohingegen deren Ausführung sowohl im Konfigurationsraum (engl. *Joint Space Control*) als auch im Arbeitsraum (engl. *Task Space Control*) erfolgen kann.

### 1.3.1 Regelung im Konfigurationsraum

Soll der Roboter eine Bewegung im Arbeitsraum  $\mathbf{x}_{e,d}(t)$  (Index  $d$  für *desired*) ausführen, so wird bei der Regelung im Konfigurationsraum zunächst die inverse Kinematik für jeden Punkt dieser Bewegung gelöst, um die zugehörigen Verläufe im Konfigurationsraum  $\mathbf{q}_d(t)$  zu erhalten. Dies ist in Abbildung 1.6 als Blockdiagramm dargestellt. Die Regelung selbst wird als geschlossener Regelkreis über die gemessenen Gelenkpositionen des Roboters  $\mathbf{q}$  entworfen. Der Nachteil dieser Methode besteht darin, dass die Endeffektorpose  $\mathbf{x}_e$  nur indirekt geregelt wird, da diese nicht direkt rückgekoppelt wird. Somit wirkt sich jeder Fehler der in der inversen Kinematik auftritt unmittelbar auf die Regelgüte aus.

Die Regelung im Konfigurationsraum kann nun *dezentral* oder *zentral* erfolgen. Bei der dezentralen Regelung wird jedes Gelenk des Roboters unabhängig voneinander geregelt und die Kopplungen der einzelnen Glieder des Roboters während der Roboterbewegung werden als Störung aufgefasst. Bei der zentralen Regelung werden die mechanischen Kopplungen und die Nichtlinearitäten des Roboters systematisch berücksichtigt. In Abschnitt 4.1 werden die typischen dezentralen und zentralen Regler für den Konfigurationsraum vorgestellt.

### 1.3.2 Regelung im Arbeitsraum

Bei der *Regelung im Arbeitsraum* gemäß Abbildung 1.7 ist die Kinematik implizit in den Regelkreis eingebettet und es wird direkt auf die Endeffektorpose geregelt. Dazu müssen jedoch die Vorwärtskinematik (1.3) und deren Ableitungen bei der Auswertung des Stellgesetzes berechnet werden, welche in der Regel komplexe Ausdrücke sind. Die Regelung im Arbeitsraum muss zwingend als zentraler Regler implementiert werden, da für die Berechnung der Vorwärtskinematik die Positionen aller Gelenke benötigt werden. Der Reglerentwurf für kinematisch nicht-redundante und redundante Roboter wird im Abschnitt 4.1 erklärt.

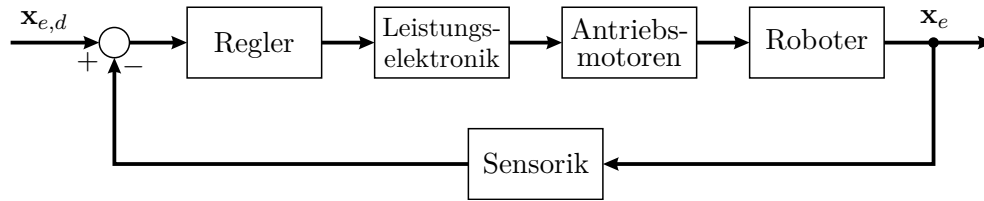


Abbildung 1.7: Regelung im Arbeitsraum [1.2].

## 1.4 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 im Raum innerhalb des Arbeitsraumes anfahren kann. Im Gegensatz dazu gibt die Wiederholgenauigkeit lediglich an, wie genau ein Roboter den selben Punkt im Arbeitsraum wiederholt anfahren kann.

Die beiden Begriffe sind in Abbildung 1.8 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 1.8 um den Abstand  $\Delta l$  von der vorgegebenen Position  $\mathbf{x}_{e,d}$  ab. Bei Robotern mit guter Absolutgenauigkeit ist  $\Delta l$  für beliebige Sollpositionen  $\mathbf{x}_{e,d}$  im Arbeitsraum 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 Vorwärtskinematik  $\mathbf{f}(\mathbf{q})$  mit der 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 kleiner als die Absolutgenauigkeit, d.h. es gilt  $r \ll \Delta l$ .

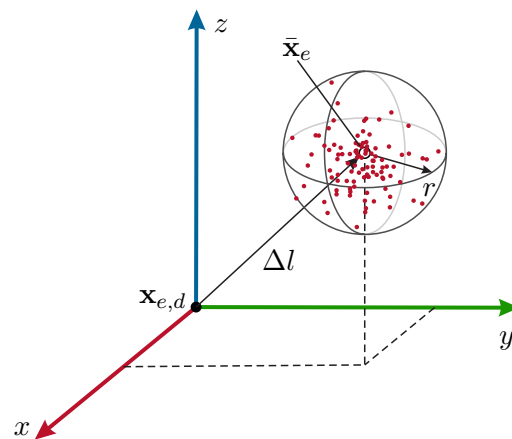


Abbildung 1.8: Absolutgenauigkeit und Wiederholgenauigkeit [1.3].

## 1.5 Literatur

- [1.1] 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>.
- [1.2] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [1.3] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik und M. Munih, *Robotics*, Ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, Dordrecht, 2010, Bd. 43.



## 2 Roboterkinematik

Die *Kinematik* beschäftigt sich ausschließlich geometrisch mit der Bewegung von Körpern unter Verwendung der Größen Zeit, Ort, Geschwindigkeit und Beschleunigung. Dabei bleiben wirkende Kräfte sowie die Massen und Massenträgheiten der Körper unberücksichtigt – diese sind Teil der *Dynamik*.

Dieses Kapitel beschäftigt sich mit der Geometrie von Robotern mit serieller Kinematik und vertieft dabei die Begriffe Vorwärtskinematik und inverse Kinematik aus Kapitel 1. Zusätzlich werden auch die Geschwindigkeiten im Konfigurations- und Arbeitsraum mithilfe der differentiellen Kinematik betrachtet.

### 2.1 Vorwärtskinematik

In diesem Abschnitt wird die allgemeine Beziehung der Vorwärtskinematik (1.3) auf Roboter mit serieller Kinematik eingeschränkt. Dazu werden zunächst die wichtigsten Zusammenhänge zu *homogenen Transformationen*, welche in der Vorlesung *Modellbildung* [2.1] eingeführt wurden, zusammengefasst. Unter Verwendung von homogenen Transformationen wird danach ein Formalismus für die Vorwärtskinematik serieller Roboter hergeleitet. Weiters werden unterschiedliche Möglichkeiten für die Parametrierung der Orientierung im dreidimensionalen Raum diskutiert. Schließlich wird der eingeführte Formalismus auf einen klassischen 6-Achs-Industrieroboter angewendet.

Im Folgenden werden die kinematischen Strukturen von Robotern mit schematischen Gelenken und Verbindungslinien entsprechend Abbildung 2.1 dargestellt.

#### 2.1.1 Homogene Transformationen

Die homogene Transformation  $\mathbf{H}_0^1$  beschreibt die Lage des Koordinatensystems  $(0_1x_1y_1z_1)$  in Bezug auf das Koordinatensystem  $(0_0x_0y_0z_0)$  (ausgedrückt im Koordinatensystem 0)

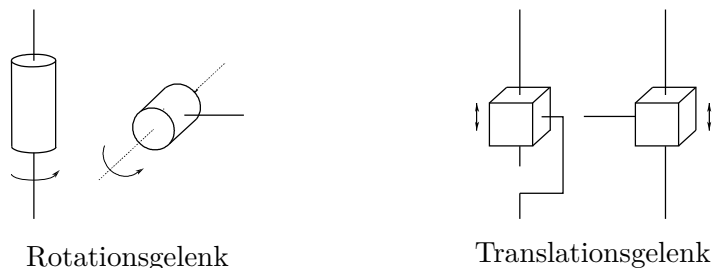


Abbildung 2.1: Schematische Darstellung eines Rotations- und eines Translationsgelenkes [2.2].

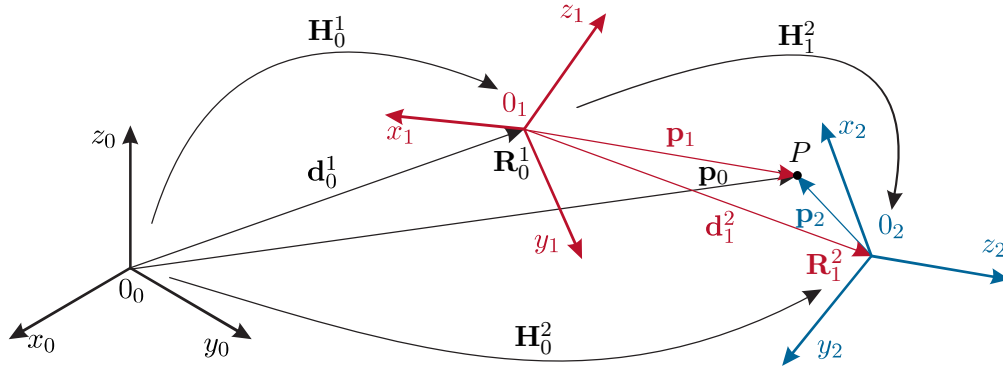


Abbildung 2.2: Homogene Transformationen am Beispiel der Koordinatensysteme  $(0_0 x_0 y_0 z_0)$ ,  $(0_1 x_1 y_1 z_1)$  und  $(0_2 x_2 y_2 z_2)$  [2.1].

und setzt sich aus der Position des Koordinatenursprungs  $\mathbf{d}_0^1 \in \mathbb{R}^3$  und der orthogonalen Rotationsmatrix  $\mathbf{R}_0^1 \in \text{SO}(3)$ <sup>1</sup> gemäß

$$\mathbf{H}_0^1 = \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{d}_0^1 \\ \mathbf{0} & 1 \end{bmatrix} \in \text{SE}(3) \quad (2.1)$$

zusammen<sup>2</sup>. Das Symbol  $\mathbf{0}$  bezeichnet dabei allgemein einen Nullvektor oder eine Nullmatrix der geeigneten Dimension. In (2.1) ist  $\mathbf{0}$  demnach ein  $1 \times 3$ -Nullvektor. Ein Koordinatensystem  $(0_2 x_2 y_2 z_2)$  wird bezüglich des Koordinatensystems  $(0_0 x_0 y_0 z_0)$  durch aufeinanderfolgende Transformationen mithilfe des Koordinatensystems  $(0_1 x_1 y_1 z_1)$  in der Form

$$\mathbf{H}_0^2 = \mathbf{H}_0^1 \mathbf{H}_1^2 \quad (2.2)$$

beschrieben, siehe Abbildung 2.2.

Homogene Transformationen können nicht nur zum Darstellen von Koordinatensystemen, sondern auch zum Berechnen von Vektoren in einem neuen Koordinatensystem benutzt werden. Als Beispiel wird der Punkt  $P$  in Abbildung 2.2 betrachtet. Ausgedrückt im Koordinatensystem  $(0_2 x_2 y_2 z_2)$  wird dieser Punkt mithilfe des homogenen Vektors  $\mathbf{P}_2^T = \begin{bmatrix} \mathbf{p}_2^T & 1 \end{bmatrix}$  beschrieben. Analog dazu beschreiben die Vektoren  $\mathbf{P}_i^T = \begin{bmatrix} \mathbf{p}_i^T & 1 \end{bmatrix}$ ,  $i \in \{0, 1\}$ , den selben Punkt  $P$  in Bezug auf das jeweilige Koordinatensystem  $(0_i x_i y_i z_i)$ . Der homogene

<sup>1</sup> $\text{SO}(n)$  steht für „spezielle orthogonale Gruppe im  $n$ -dimensionalen Raum“ und wird auch als *Drehgruppe* bezeichnet. Sie umfasst alle Drehungen. Diese Drehungen werden durch orthogonale Matrizen  $\mathbf{R}$  mit der Determinante  $\det(\mathbf{R}) = +1$  beschrieben.

<sup>2</sup> $\text{SE}(n)$  ist die „spezielle Euklidische Gruppe des  $n$ -dimensionalen Raums“, welche alle Verschiebungen und Rotationen enthält, jedoch Spiegelungen ausschließt. Diese Transformationen werden auch als *Euklidische Bewegungen* bzw. *Starrkörperbewegungen* bezeichnet.

Vektor  $\mathbf{P}_0$  berechnet sich gemäß

$$\begin{aligned}\mathbf{P}_0 &= \begin{bmatrix} \mathbf{p}_0 \\ 1 \end{bmatrix} = \mathbf{H}_0^1 \mathbf{p}_1 = \mathbf{H}_0^1 \mathbf{H}_1^2 \mathbf{p}_2 = \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{d}_0^1 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^2 & \mathbf{d}_1^2 \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{p}_2 \\ &= \begin{bmatrix} \mathbf{R}_0^1 \mathbf{R}_1^2 & \mathbf{R}_0^1 \mathbf{d}_1^2 + \mathbf{d}_0^1 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0^1 \mathbf{R}_1^2 \mathbf{p}_2 + \mathbf{R}_0^1 \mathbf{d}_1^2 + \mathbf{d}_0^1 \\ 1 \end{bmatrix}.\end{aligned}\quad (2.3)$$

Der Vektor  $\mathbf{p}_0$  setzt sich also aus den Verschiebungsvektoren  $\mathbf{d}_0^1$  und  $\mathbf{d}_1^2$  sowie dem Vektor  $\mathbf{p}_2$  zusammen. Es ist zu beachten, dass die Vektoren durch die homogenen Transformationen stets automatisch in das selbe Koordinatensystem  $(0_0 x_0 y_0 z_0)$  transformiert wurden, bevor diese addiert werden.

Weiters wird die elementare Transformation  $\mathbf{H}_{T_i,d}$  eingeführt, welche die Translation um die Distanz  $d$  entlang der lokalen Koordinatenachse  $i \in \{x, y, z\}$  beschreibt, d.h.

$$\mathbf{H}_{T_x,d} = \begin{bmatrix} \mathbf{E} & \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{H}_{T_y,d} = \begin{bmatrix} \mathbf{E} & \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{H}_{T_z,d} = \begin{bmatrix} \mathbf{E} & \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (2.4)$$

Eine Rotation um die lokale Achse  $i \in \{x, y, z\}$  mit dem Winkel  $\varphi$  wird im Folgenden mit der homogenen Transformation  $\mathbf{H}_{R_i,\varphi}$  bezeichnet, welche sich gemäß

$$\mathbf{H}_{R_i,\varphi} = \begin{bmatrix} \mathbf{R}_{i,\varphi} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.5)$$

mit der zugehörigen Rotationsmatrix  $\mathbf{R}_{i,\varphi}$  zusammensetzt.

### 2.1.2 Serielle kinematische Ketten

In Abbildung 2.3 ist schematisch eine allgemeine serielle kinematische Kette bestehend aus Rotations- und Translationsgelenken mit dem Inertialkoordinatensystem  $(0_0 x_0 y_0 z_0)$  und dem Koordinatensystem des Endeffektors  $(0_e x_e y_e z_e)$  dargestellt. Gemäß (2.1) wird die homogene Transformation des Endeffektorkoordinatensystems  $(0_e x_e y_e z_e)$  in Bezug auf das Inertialkoordinatensystem  $(0_0 x_0 y_0 z_0)$  durch

$$\mathbf{H}_0^e(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^e(\mathbf{q}) & \mathbf{d}_0^e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.6)$$

als Funktion der generalisierten Koordinaten  $\mathbf{q} \in \mathbb{R}^n$  beschrieben. Weiters wird jedes Glied des Roboters mit einem Koordinatensystem  $(0_i x_i y_i z_i)$ ,  $i = 1, \dots, n$ , ausgestattet. Dabei ist es von Vorteil, jeweils eine Achse der Koordinatensysteme in die jeweilige Rotations- bzw. Translationsachse zu legen. Wie in Abbildung 2.4 gezeigt, entsteht dadurch eine Kette von Koordinatensystemen, welche mithilfe der homogenen Transformationen in der Form

$$\mathbf{H}_0^n(\mathbf{q}) = \mathbf{H}_0^1(q_1) \mathbf{H}_1^2(q_2) \cdots \mathbf{H}_{i-1}^i(q_i) \cdots \mathbf{H}_{n-1}^n(q_n) \quad (2.7)$$

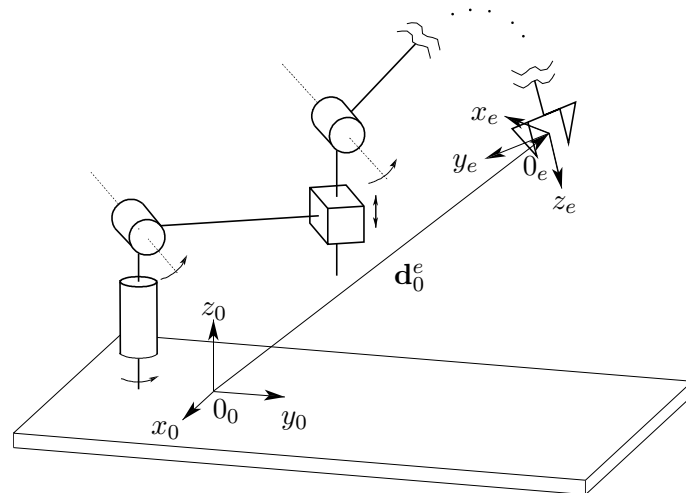


Abbildung 2.3: Schematische Darstellung einer seriellen kinematischen Kette [2.2].

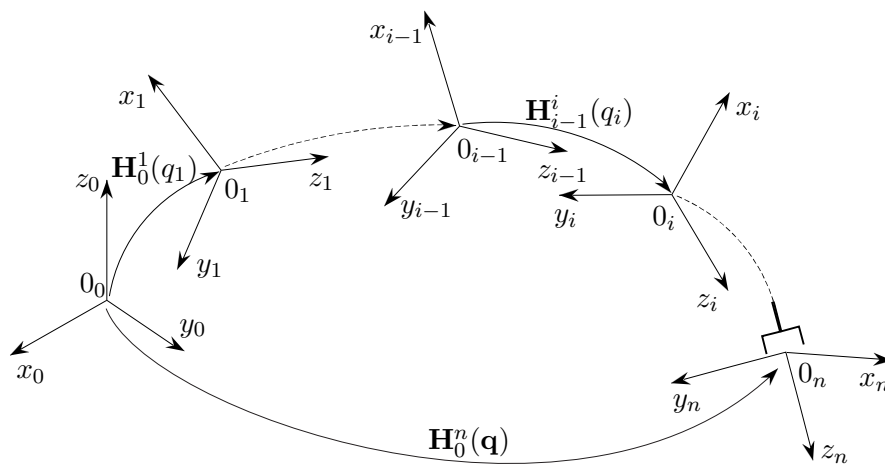


Abbildung 2.4: Koordinatensysteme eines Roboters mit serieller Kinematik [2.2].

beschrieben wird, siehe (2.2). Jede der Transformationen  $\mathbf{H}_{i-1}^i(q_i)$ ,  $i = 1, \dots, n$ , in (2.7) 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  $\mathbf{H}_b^0$  und  $\mathbf{H}_n^e$  zu (2.2) hinzugefügt, welche das Koordinatensystem der Roboterbasis  $b$  in Bezug auf das Inertialkoordinatensystem  $0$  bzw. das Endeffektorkoordinatensystem  $e$  in Bezug auf das Koordinatensystem des letzten Roboter gliedes  $n$  beschreibt. Damit folgt die Beschreibung der Roboterkinematik als homogene Transformation zu

$$\mathbf{H}_b^e(\mathbf{q}) = \mathbf{H}_b^0 \mathbf{H}_0^n(\mathbf{q}) \mathbf{H}_n^e = \begin{bmatrix} \mathbf{R}_b^e(\mathbf{q}) & \mathbf{d}_b^e(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix}. \quad (2.8)$$

Um die Vorwärtskinematik in der Form (1.3) zu erhalten, wird  $\mathbf{H}_b^e(\mathbf{q})$  schließlich in Form der  $m$  Koordinaten des Arbeitsraums

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} \mathbf{d}_b^e(\mathbf{q}) \\ \phi(\mathbf{R}_b^e(\mathbf{q})) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_e \\ \phi_e \end{bmatrix} \quad (2.9)$$

dargestellt. In (2.9) setzt sich die Pose des Endeffektors  $\mathbf{x}_e$  im Arbeitsraum aus dem Vektor  $\mathbf{d}_b^e$  zum Ursprung des Endeffektorkoordinatensystems und einer Minimalkoordinatendarstellung der Orientierung  $\phi(\cdot)$  zusammen. Weiters wurden die Kurzbezeichnungen  $\mathbf{p}_e$  und  $\phi_e$  für die Position und Orientierung des Endeffektors im Arbeitsraum eingeführt. Im nächsten Abschnitt werden Minimalkoordinatendarstellungen für die Orientierung im Detail diskutiert.

### 2.1.3 Parametrierung der Orientierung

Im dreidimensionalen Raum besitzt ein Starrkörper 6 Freiheitsgrade, wovon 3 Freifreiheitsgrade die Orientierung charakterisieren. Bei einer *Minimalkoordinatendarstellung*  $\phi(\cdot)$  wird die Orientierung ebenfalls mit genau drei Winkeln parametrisiert. Dazu zählen die EULER-Winkel und die TAIT-BRYAN-Winkel, zu denen auch die *Roll-Nick-Gier-Winkel* (engl. *Roll-Pitch-Yaw*, RPY) zählen. Darüber hinaus kann die Orientierung auch mit mehr als drei Parametern angegeben werden, wie z. B. durch die Rotationsmatrix (9 Parameter), Quaternionen (4 Parameter) und der Achse-Winkel-Darstellung (4 Parameter). In Abschnitt 2.3.2 werden die Eigenschaften von Minimal- und Nicht-Minimalkoordinatendarstellungen beschrieben. Minimalkoordinatendarstellungen werden in der industriellen Robotik am häufigsten verwendet und praktisch jede Robotersteuerung verwendet sie, um Endeffektorposen im Raum zu parametrisieren. Im Folgenden werden die mathematischen Grundlagen für die Minimalkoordinatendarstellungen eingeführt.

Im Allgemeinen erhält man Minimalkoordinatendarstellungen aus einer Rotationsmatrix  $\mathbf{R}$  in der Form

$$\phi = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \phi_{ijk}(\mathbf{R}), \quad i, j, k \in \{x, y, z\}, \quad (2.10)$$

wobei  $\mathbf{R}$  aus einer Sequenz von drei Rotationen besteht, d. h.

$$\mathbf{R}(\phi) = \mathbf{R}_{i,\phi} \mathbf{R}_{j,\theta} \mathbf{R}_{k,\psi} . \quad (2.11)$$

Für die EULER-Winkel gilt  $i = k$  in (2.10) und für die TAIT-BRYAN-Winkel  $i \neq j \neq k$ . An dieser Stelle sei erwähnt, dass Minimalkoordinatendarstellungen mit  $i = j$  oder  $j = k$  degeneriert sind und nicht den gesamten  $\text{SO}(3)$  beschreiben können. Während (2.11) direkt ausgewertet werden kann, verbergen sich hinter (2.10) komplexere Zusammenhänge mit Fallunterscheidungen.

Die klassischen EULER-Winkel verwenden  $i = z$ ,  $j = y$  und  $k = z$  als Sequenz der drei Rotationen, also  $\phi_{zyz}(\cdot)$ . Die zugehörige Rotationsmatrix lautet

$$\mathbf{R}(\phi) = \mathbf{R}_{z,\phi} \mathbf{R}_{y,\theta} \mathbf{R}_{z,\psi} = \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} , \quad (2.12)$$

wobei  $s_x = \sin(x)$  und  $c_x = \cos(x)$  kompakte Notationen für die trigonometrischen Funktionen sind. 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 (2.12) mit

$$\phi_{zyz}(\mathbf{R}) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(r_{23}, r_{13}) \\ \text{atan2}\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \\ \text{atan2}(r_{32}, -r_{31}) \end{bmatrix} \quad (2.13)$$

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

$$\phi_{zyz}(\mathbf{R}) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(-r_{23}, -r_{13}) \\ \text{atan2}\left(-\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \\ \text{atan2}(-r_{32}, r_{31}) \end{bmatrix} . \quad (2.14)$$

In (2.13) und (2.14) ist  $\text{atan2}(y, x)$  die vorzeichenkorrekte Auswertung des Arkustangens  $\arctan(y/x)$  in allen vier Quadranten.

Die RPY-Winkel verwenden die Konvention  $\phi_{zyx}(\cdot)$  und die Rotationsmatrix  $\mathbf{R}(\phi)$  ergibt sich zu

$$\mathbf{R}(\phi) = \mathbf{R}_{z,\phi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\psi} = \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} . \quad (2.15)$$

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

$$\phi_{zyx}(\mathbf{R}) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(r_{21}, r_{11}) \\ \text{atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \\ \text{atan2}(r_{32}, r_{33}) \end{bmatrix} \quad (2.16)$$

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

$$\phi_{zyx}(\mathbf{R}) = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(-r_{21}, -r_{11}) \\ \text{atan2}\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right) \\ \text{atan2}(-r_{32}, -r_{33}) \end{bmatrix}. \quad (2.17)$$

### 2.1.4 6-Achs-Roboter mit sphärischem Handgelenk

Die Grundlagen zur Vorwärtskinematik werden in diesem Abschnitt am Beispiel eines 6-Achs-Roboters mit sphärischem Handgelenk veranschaulicht. Ein *sphärisches Handgelenk* (engl. *spherical wrist*) ist ein gängiger Begriff in der industriellen Robotik und bezeichnet eine Konstellation von drei aufeinanderfolgenden Rotationsgelenken, deren Achsen sich in *einem* Punkt schneiden. Dies ist in Abbildung 2.5 mit den Gelenken 4, 5 und 6 schematisch dargestellt. Ähnlich wie ein Kugelgelenk bietet diese Anordnung 3 Freiheitsgrade und ermöglicht es, den Endeffektor beliebig im gesamten  $\text{SO}(3)$  zu orientieren. Im Gegensatz zu einem Kugelgelenk kann allerdings ein mechanisches Blockieren (engl. *gimbal lock*) auftreten wenn gilt  $q_5 = 0$ . In diesem Fall sind die Achsen  $z_4$  und  $z_6$  parallel zueinander und eine Drehung um die  $x_4$ -Achse ist nicht möglich.

**Beispiel 2.1 (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  $(0_i x_i y_i z_i)$ ,  $i = 1, \dots, 6$  ist schematisch in Abbildung 2.5 dargestellt. Das Inertialkoordinatensystem wird mit 0 bezeichnet und jedes weitere Koordinatensystem  $i$  liegt körperfest im Glied  $i$  des Roboters. Die 6 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 der Abbildung können die geometrischen Zusammenhänge als homogene Transformationen abgelesen werden, welche sich aus den elementaren Translationen und Rotationen

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

zusammensetzen. Analog zu (2.7) und (2.9) folgt mit

$$\mathbf{H}_0^6(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^6(\mathbf{q}) & \mathbf{d}_0^6(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{H}_0^1(q_1)\mathbf{H}_1^2(q_2)\mathbf{H}_2^3(q_3)\mathbf{H}_3^4(q_4)\mathbf{H}_4^5(q_5)\mathbf{H}_5^6(q_6) \quad (2.18)$$

die Vorwärtskinematik gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} \mathbf{d}_0^6(\mathbf{q}) \\ \phi_{zyz}(\mathbf{R}_0^6(\mathbf{q})) \end{bmatrix}, \quad (2.19)$$

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 ( $0_6x_6y_6z_6$ ) liegt auf der  $z_0$ -Achse und hat die selbe Orientierung wie das Inertialkoordinatensystem. Die Koordinaten im Arbeitsraum lauten

$$\mathbf{x}_e^T = [0, 0, d_1 + d_2 + d_4 + d_6, 0, 0, 0] . \quad (2.20)$$

Für  $\mathbf{q}^T = [\frac{\pi}{2}, 0, \frac{\pi}{2}, 0, 0, 0]$  ergibt sich

$$\mathbf{x}_e^T = [0, d_4 + d_6, d_1 + d_2, \frac{\pi}{2}, \frac{\pi}{2}, 0] . \quad (2.21)$$

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  $z_0$ -Achse sind und der restliche Arm in Richtung der  $y_0$ -Achse zeigt. Das Endeffektorkoordinatensystem ist gegenüber dem Inertialkoordinatensystem entsprechend der Konvention  $\phi_{zyz}(\cdot)$  zuerst um  $\frac{\pi}{2}$  um die  $z_0$ -Achse und dann um  $\frac{\pi}{2}$  um die neue  $y$ -Achse gedreht. Damit ist die  $z_6$ -Achse parallel zur  $y_0$ -Achse und die  $x_6$ -Achse ist antiparallel zur  $z_0$ -Achse.



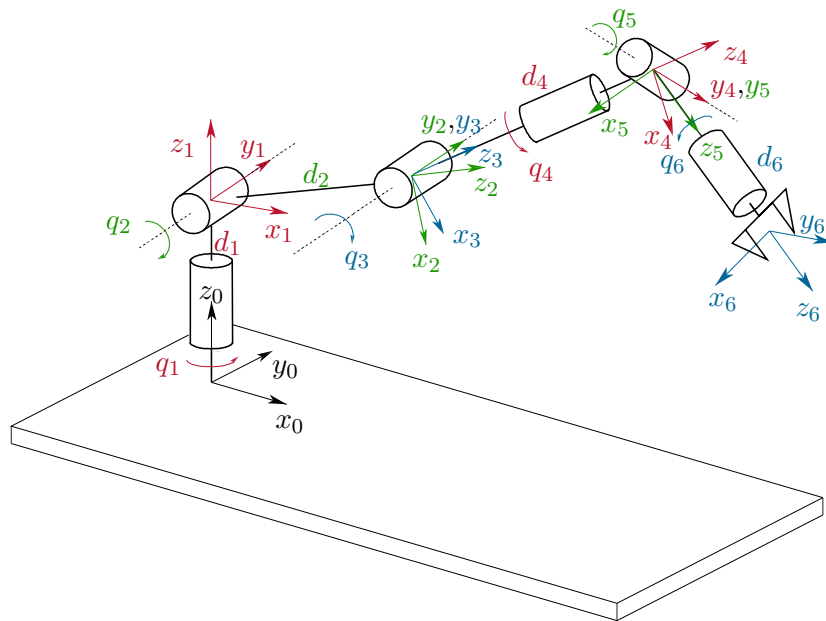


Abbildung 2.5: 6-Achs-Roboter mit sphärischem Handgelenk [2.2].

## 2.2 Inverse Kinematik

Für Roboter mit serieller Kinematik kann die Lage des Endeffektors mithilfe von (2.7) eindeutig für jede Konfiguration  $\mathbf{q}$  berechnet werden. Andererseits ist das Problem der inversen Kinematik wesentlich komplexer, da die Gleichung (1.3) invertiert werden muss (siehe (1.4)), um für eine gegebene Pose im Arbeitsraum  $\mathbf{x}_e$  die zugehörige Konfiguration  $\mathbf{q}$  zu berechnen. Die zu lösende Gleichung (1.3) ist im Allgemeinen nichtlinear und es ist nicht immer möglich, geschlossene analytische Ausdrücke als Lösung zu finden. Dies ist nur für einfache kinematische Ketten der Fall bzw. für kinematische Ketten mit spezieller Struktur. Es benötigt daher entweder eine algebraische Intuition, um geeignete Gleichungen für die unbekannten Größen 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.

### 2.2.1 Geschlossene analytische Lösungen

Mithilfe von geschlossenen analytischen Ausdrücken ist es möglich, alle existierenden Lösungen 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 Gelenkwinkel des Roboters enthalten. Anschließend werden diese in eine geschlossen lösbare Form umgeformt. Häufig werden dabei transzendente Gleichungen in einer Variable  $q_i$  der Form

$$C_1 \cos(q_i) + C_2 \sin(q_i) + C_3 = 0, \quad (2.22)$$

mit den Konstanten  $C_1$ ,  $C_2$  und  $C_3$  gebildet. Die Lösung dieser Gleichung lautet mithilfe der WEIERSTRASS-Substitution  $x = \tan\left(\frac{q_i}{2}\right)$

$$q_i = 2 \arctan\left(\frac{C_2 \pm \sqrt{C_2^2 - C_3^2 + C_1^2}}{C_1 - C_3}\right). \quad (2.23)$$

Eine weitere häufige Form ist das Gleichungspaar

$$C_1 \cos(q_i) + C_2 \sin(q_i) + C_3 = 0 \quad (2.24)$$

$$C_1 \sin(q_i) - C_2 \cos(q_i) + C_4 = 0, \quad (2.25)$$

für welches die eindeutige Lösung

$$q_i = \text{atan2}(-C_1 C_4 - C_2 C_3, C_2 C_4 - C_1 C_3) \quad (2.26)$$

existiert.

### Geometrische Methoden

Bei den geometrischen Methoden werden ausgezeichnete Punkte am Roboter identifiziert, die durch eine reduzierte Anzahl an Gelenkwinkeln 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 Abbildung 2.5. Es kann gezeigt werden, dass die inverse Kinematik 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 Endeffektorposition von der Lösung der Orientierung entkoppelt. Es lässt sich ein Punkt im Koordinatensystem  $(0_3 x_3 y_3 z_3)$  finden, welcher durch

$$\mathbf{H}_0^6 \left( \mathbf{H}_3^6(q_4, q_5, q_6) \right)^{-1} = \mathbf{H}_0^3(q_1, q_2, q_3) \quad (2.27)$$

beschrieben werden kann, siehe (2.18) sowie Abbildung 2.5.

### 2.2.2 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 existierenden Lösungen der inversen Kinematik.

Zu den wichtigsten numerischen Lösungen zählen die *differentielle inverse Kinematik*, welche in Abschnitt 2.4 ausführlich behandelt werden, und die *iterativen Methoden*. Bei den iterativen Methoden wird die Vorwärtskinematik (1.3) als Nullstellensuche in der Form

$$\mathbf{x}_e - \mathbf{f}(\mathbf{q}) = \mathbf{0} \quad (2.28)$$

aufgefasst und mithilfe des NEWTON-RAPHSON-Verfahrens iterativ gelöst. Hier ist zu beachten, dass eine Startlösung  $\mathbf{q}_0$  benötigt wird, welche großen Einfluss auf die gefundene Lösung  $\mathbf{q}$  hat.

### 2.2.3 Beispiele

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

**Beispiel 2.2 (Inverse Kinematik eines 3-Achs-Roboters in RRR-Struktur).** Der in Abbildung 2.6 gezeigte 3-Achs-Roboter in RRR-Struktur entspricht dem 6-Achs-Roboter aus Abbildung 2.5 ohne sphärischem Handgelenk, d. h. ohne die Gelenke 4, 5 und 6. Das Endeffektorkoordinatensystem des 3-Achs-Roboters ( $0_4x_4y_4z_4$ ) 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.

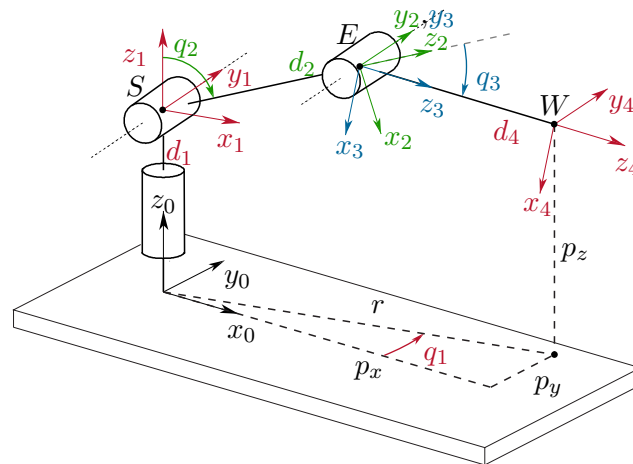


Abbildung 2.6: 3-Achs-Roboter in RRR-Struktur [2.2].

Der dreidimensionale Konfigurationsraum  $\mathbf{q}^T = [q_1 \ q_2 \ q_3]$ ,  $n = 3$ , erlaubt es, einen

dreidimensionalen Arbeitsraum ( $m = 3$ ) zu wählen. Als Koordinaten im Arbeitsraum wird der Ursprung des Endeffektorkoordinatensystems mit  $\mathbf{x}_e = \mathbf{p}_0^4 = [p_x \ p_y \ p_z]^T$  gewählt, siehe Abbildung 2.6. In der  $(x_0y_0)$ -Ebene 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  $z_0$ -Achse und die Gerade  $r$ ) mit  $q_1$  gedreht werden kann.

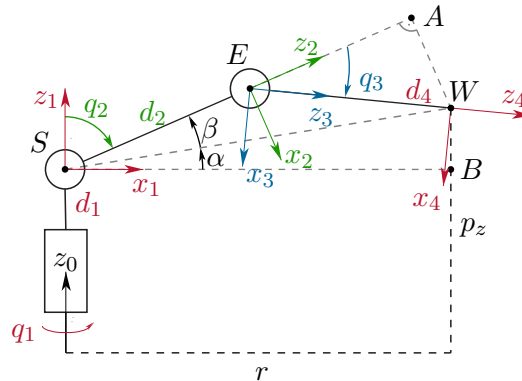


Abbildung 2.7: Ansicht der  $z_0$ - $r$ -Ebene des 3-Achs-Roboters.

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

$$q_1 = \text{atan2}(p_y, p_x) \text{ .} \quad (2.29)$$

Für das Dreieck  $(SEW)$ , siehe Abbildung 2.7, lautet der Kosinussatz für den Winkel  $\pi - q_3$

$$r^2 + (p_z - d_1)^2 = d_2^2 + d_4^2 - 2d_2d_4 \cos(\pi - q_3) . \quad (2.30)$$

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

$$c_{q_3} = \cos(q_3) = \frac{r^2 + (p_z - d_1)^2 - d_2^2 - d_4^2}{2d_3d_4} \quad (2.31)$$

und damit

$$q_3 = \pm \arccos(c_{q_3}) \ , \quad (2.32)$$

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  $\alpha$  gemäß

$$\alpha = \text{atan2}(p_z - d_1, r) \quad (2.33)$$

berechnet und der Zusammenhang

$$\cos(\beta)\sqrt{r^2 + (p_z - d_1)^2} = d_2 + d_4 \cos(q_3) \quad (2.34)$$

ist durch zweimalige Projektion im Dreieck (*SWA*) ersichtlich. Unter Verwendung von (2.31) berechnet sich die Zwischengröße  $\beta$  zu

$$\beta = \arccos\left(\frac{r^2 + (p_z - d_1)^2 + d_2^2 - d_4^2}{2d_2\sqrt{r^2 + (p_z - d_1)^2}}\right) \quad (2.35)$$

und es folgt der verbleibende Winkel  $q_2$  gemäß

$$q_2 = \frac{\pi}{2} - \alpha \mp \beta. \quad (2.36)$$

Bisher wurden also zwei unterschiedliche Lösungen gefunden: Die *Elbow-Up*-Lösung hat ein negatives Vorzeichen in (2.36) und ein positives Vorzeichen in (2.32) und für die *Elbow-Down*-Lösung ist es umgekehrt. Insgesamt besitzt der 3-Achs-Roboter mit RRR-Struktur vier unterschiedliche Lösungen, wie in Abbildung 2.8 gezeigt ist. Wird die erste Achse um  $180^\circ$  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.

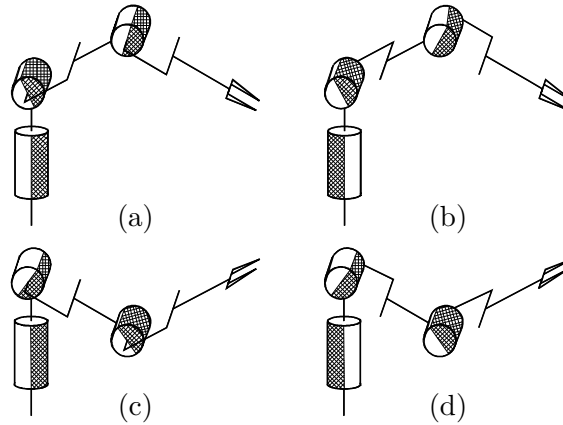


Abbildung 2.8: Vier Konfigurationen für die inverse Kinematik des 3-Achs-Roboters: (a) Shoulder-Right/Elbow-Up, (b) Shoulder-Left/Elbow-Up, (c) Shoulder-Right/Elbow-Down, (d) Shoulder-Left/Elbow-Down [2.2].

**Beispiel 2.3 (Inverse Kinematik des sphärischen Handgelenks).** Abbildung 2.9 zeigt eine schematische Darstellung eines sphärischen Handgelenks und den zugehörigen Koordinatensystemen  $(0_i x_i y_i z_i)$ , 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

$$\mathbf{H}_0^6(\mathbf{q}) = \mathbf{H}_0^4(q_4)\mathbf{H}_4^5(q_5)\mathbf{H}_5^6(q_6) \quad (2.37)$$

mit

$$\mathbf{H}_0^4(q_4) = \mathbf{H}_{Rz,q_4} \mathbf{H}_{Tz,d_4} \quad (2.38)$$

$$\mathbf{H}_4^5(q_5) = \mathbf{H}_{Ry,q_5} \quad (2.39)$$

$$\mathbf{H}_5^6(q_6) = \mathbf{H}_{Rz,q_6} \mathbf{H}_{Tz,d_6} \quad (2.40)$$

beschrieben, welche direkt aus Abbildung 2.9 abgelesen werden können. Es ist zu beachten, dass sich die Rotationsmatrix  $\mathbf{R}_0^6(\mathbf{q})$  in (2.37) zu

$$\mathbf{R}_0^6(\mathbf{q}) = \begin{bmatrix} c_{q_4} c_{q_5} c_{q_6} - s_{q_4} s_{q_6} & -c_{q_4} c_{q_5} s_{q_6} - s_{q_4} c_{q_6} & c_{q_4} s_{q_5} \\ s_{q_4} c_{q_5} c_{q_6} + c_{q_4} s_{q_6} & -s_{q_4} c_{q_5} s_{q_6} + c_{q_4} c_{q_6} & s_{q_4} s_{q_5} \\ -s_{q_5} c_{q_6} & s_{q_5} s_{q_6} & c_{q_5} \end{bmatrix} \quad (2.41)$$

ergibt, welche die gleiche Form aufweist wie die Rotationsmatrix für die klassischen EULER-Winkel in (2.12) mit  $\phi = q_4$ ,  $\theta = q_5$  und  $\psi = q_6$ . Dies kommt durch die Wahl der Koordinatensysteme und durch die gewählte Sequenz von Drehungen in der kinematischen Struktur zustande. Durch diesen Zusammenhang folgt sofort mit den Gleichungen (2.13) und (2.14) die inverse Kinematik für das sphärische Handgelenk eines Roboters für eine gegebene Orientierung  $\phi$  bzw. für eine gegebene Rotationsmatrix in der Form (2.12). Für jede Orientierung im Arbeitsraum existieren somit zwei Sätze von Lösungen  $\mathbf{q}$ .

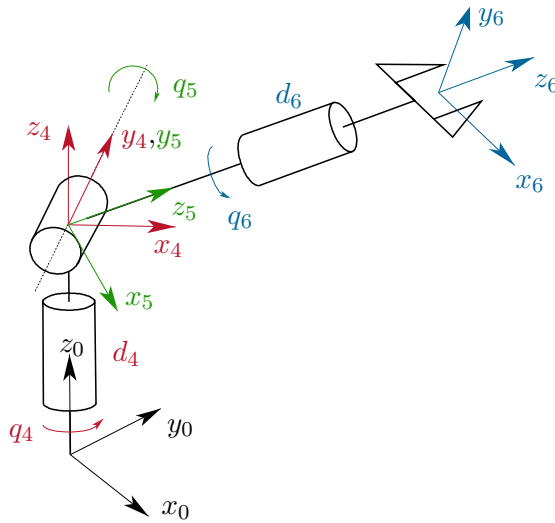


Abbildung 2.9: Sphärisches Handgelenk [2.2].

**Beispiel 2.4** (Inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk). Aufbauend auf den drei Beispielen 2.1, 2.2 und 2.3, soll in diesem Beispiel die inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk in Abbildung

2.5 analytisch gelöst werden.

Im 6-dimensionalen Arbeitsraum wird nun eine Endeffektorpose  $\mathbf{x}_e^T = [\mathbf{p}_e^T \ \phi_e^T]$  vorgegeben und die zugehörigen Konfigurationen  $\mathbf{q}^T = [q_1 \ \dots \ q_6]$  werden gesucht. Es ist daher  $\mathbf{R}_0^6 = \mathbf{R}(\phi_e)$  gemäß (2.12) und  $\mathbf{d}_0^6 = \mathbf{p}_e$ . Für die inverse Kinematik des 6-Achs-Roboters mit sphärischem Handgelenk wird der Handgelenkspunkt  $0_4$  als Bezugspunkt gewählt und die Position dieses Punktes folgt zu

$$\mathbf{d}_0^4 = \mathbf{d}_0^6 - \mathbf{R}_0^6 \begin{bmatrix} 0 & 0 & d_6 \end{bmatrix}^T. \quad (2.42)$$

Wie anhand von (2.18) nachvollzogen werden kann, ist der Vektor  $\mathbf{d}_0^4$  ausschließlich eine Funktion der ersten drei Gelenke, also gilt  $\mathbf{d}_0^4 = \mathbf{d}_0^4(q_1, q_2, q_3)$ . Es ist zu beachten, dass dies nicht für die Rotationsmatrix  $\mathbf{R}_0^4 = \mathbf{R}_0^4(q_1, q_2, q_3, q_4)$  gilt, wie auch in Abbildung 2.5 ersichtlich ist:  $q_4$  dreht um die Längsachse des Gliedes 4 und bewegt damit den Ursprung des Koordinatensystems ( $0_4 x_4 y_4 z_4$ ) nicht.

Damit kann die inverse Kinematik in folgenden Schritten gelöst werden.

1. Berechnen des Handgelenkspunktes  $\mathbf{d}_0^4(q_1, q_2, q_3)$  gemäß (2.42).
2. Lösen der inversen Kinematik für  $(q_1, q_2, q_3)$  analog zu Beispiel 2.2.
3. Berechnen von  $\mathbf{R}_0^3(q_1, q_2, q_3)$  mithilfe von (2.18).
4. Berechnen von  $\mathbf{R}_3^6(q_4, q_5, q_6) = \mathbf{R}_0^3(q_1, q_2, q_3)^T \mathbf{R}_0^6$ .
5. Lösen der inversen Kinematik für  $(q_4, q_5, q_6)$  analog zu Beispiel 2.3.

## 2.3 Differentielle Kinematik

Die vorigen beiden Abschnitte 2.1 und 2.2 zur Vorwärtskinematik bzw. inversen Kinematik haben den Zusammenhang zwischen den Positionen im Konfigurationsraum und dem Arbeitsraum eines Roboters hergestellt. Die *differentielle Kinematik* beschäftigt sich nun mit den *Geschwindigkeiten* im Konfigurations- und Arbeitsraum und den mathematischen Beziehungen zwischen den Räumen. Die wichtigsten Werkzeuge dabei sind die *analytische* und die *geometrische Manipulator Jacobi-Matrix*. Zusätzlich können mit diesen Matrizen kinematische Eigenschaften von Robotern analysiert, wie z. B. Singularitäten und Redundanzen, und auch numerische Lösungen der inversen Kinematik gefunden werden.

### 2.3.1 Analytische und geometrische Manipulator Jacobi-Matrix

Die Geschwindigkeit des Endeffektors beschrieben durch die Pose im  $m$ -dimensionalen Arbeitsraum  $\mathbf{x}_e$  wird durch zeitliches Ableiten der Vorwärtskinematik (2.9) gemäß

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\phi}_e \end{bmatrix} = \underbrace{\frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q})}_{\mathbf{J}_A(\mathbf{q})} \dot{\mathbf{q}} = \mathbf{J}_A(\mathbf{q}) \dot{\mathbf{q}} \quad (2.43)$$

mit der  $m \times n$  *analytischen Manipulator Jacobi-Matrix* (engl. *analytical Jacobian*)

$$\mathbf{J}_A(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) \quad (2.44)$$

berechnet. In (2.43) wurde die translatorische Endeffektorgeschwindigkeit  $\dot{\mathbf{p}}_e$  und die rotatorische Endeffektorgeschwindigkeit in Minimalkoordinatendarstellung  $\dot{\phi}_e$  eingeführt. Direktes berechnen von  $\partial \mathbf{f}(\mathbf{q}) / \partial \mathbf{q}$  ist für die Koordinaten der Orientierung meist nicht möglich, weil  $\phi_e(\mathbf{q})$  nicht in Form von geschlossenen Ausdrücken vorliegt, sondern über die Rotationsmatrix  $\mathbf{R}_0^e(\mathbf{q})$  berechnet werden muss, siehe (2.10) bzw. (2.13) und (2.14).

An dieser Stelle soll hervorgehoben werden, dass sich die analytische Manipulator Jacobi-Matrix von der (*geometrischen*) *Manipulator Jacobi-Matrix* (engl. *geometric Jacobian* oder *body Jacobian*) unterscheidet, welche in der Vorlesung *Modellbildung* [2.1] eingeführt wurde. Diese ist in der Form

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_0^e \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} (\mathbf{J}_v)_0^e(\mathbf{q}) \\ (\mathbf{J}_\omega)_0^e(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \\ \frac{\partial \boldsymbol{\omega}_0^e}{\partial \mathbf{q}} \end{bmatrix} \dot{\mathbf{q}} \quad (2.45)$$

definiert und beschreibt die rotatorische Geschwindigkeit als Vektor der Drehwinkelgeschwindigkeiten  $\boldsymbol{\omega}_0^e$  des Endeffektorkoordinatensystems  $e$  gegenüber dem Inertialkoordinatensystem 0. Im Gegensatz zur analytischen Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  muss für (2.45) bei der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  *keine* Darstellung für die Orientierung gewählt bzw. festgelegt werden. Der Vektor mit den Komponenten  $\boldsymbol{\omega}_0^e = [\omega_{0,1}^e \ \omega_{0,2}^e \ \omega_{0,3}^e]^T$  ergibt sich aus der Rotationsmatrix des Endeffektors  $\mathbf{R}_0^e$  und dessen Zeitableitung mit

$$\mathbf{S}(\boldsymbol{\omega}_0^e) = \dot{\mathbf{R}}_0^e(\mathbf{R}_0^e)^T = \begin{bmatrix} 0 & -\omega_{0,3}^e & \omega_{0,2}^e \\ \omega_{0,3}^e & 0 & -\omega_{0,1}^e \\ -\omega_{0,2}^e & \omega_{0,1}^e & 0 \end{bmatrix}, \quad (2.46)$$

unter Verwendung des schiefsymmetrischen Operators  $\mathbf{S}(\boldsymbol{\omega})$  [2.1].

Während sowohl  $\dot{\phi}_e$  als auch  $\boldsymbol{\omega}_0^e$  Drehgeschwindigkeiten des Endeffektorkoordinatensystems beschreiben, führt das zeitliche Integral von  $\boldsymbol{\omega}_0^e$ , also  $\int \boldsymbol{\omega}_0^e dt$ , zu keinem sinnvollen Ausdruck für die Orientierung. Im Gegensatz dazu ist  $\phi_e = \int \dot{\phi}_e dt$  ein gültiger Zusammenhang, denn gemäß (2.43) ist  $\dot{\phi}_e$  eine echte Zeitableitung der Minimalkoordinatendarstellung  $\phi_e$  und beschreibt die momentane Drehwinkelgeschwindigkeit in diesen Koordinaten. Dies illustriert das folgende Beispiel.

**Beispiel 2.5 (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

$$\boldsymbol{\omega}_1(t) = \begin{cases} \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^T & \text{für } 0 \leq t < 1 \\ \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^T & \text{für } 1 \leq t \leq 2 \end{cases} \quad (2.47)$$



und

$$\boldsymbol{\omega}_2(t) = \begin{cases} \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^T & \text{für } 0 \leq t < 1 \\ \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^T & \text{für } 1 \leq t \leq 2 \end{cases}. \quad (2.48)$$

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

$$\int_0^2 \boldsymbol{\omega}_i(t) dt = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & 0 \end{bmatrix}^T, \quad i = 1, 2, \quad (2.49)$$

jedoch kommt in beiden Fällen eine ganz unterschiedliche Lage für den Starrkörper zum Zeitpunkt  $t = 2$  heraus, wie dies in Abbildung 2.10 dargestellt ist.

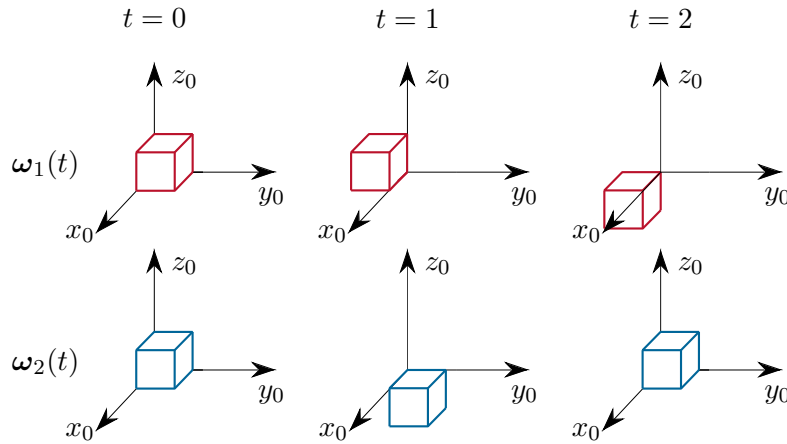


Abbildung 2.10: Bewegung des Starrkörpers, welcher mit  $\boldsymbol{\omega}_1(t)$  bzw. mit  $\boldsymbol{\omega}_2(t)$  rotiert wird [2.2].

Der Zusammenhang zwischen  $\dot{\phi}_e$  und  $\boldsymbol{\omega}_0^e$  wird mithilfe von (2.12) mit  $\mathbf{R}_0^e = \mathbf{R}(\phi_e)$  und (2.46) über eine Transformationsmatrix  $\mathbf{T}(\phi_e)$  gemäß

$$\boldsymbol{\omega}_0^e = \mathbf{T}(\phi_e) \dot{\phi}_e \quad (2.50)$$

beschrieben. Diese Matrix  $\mathbf{T}(\phi_e) = \mathbf{T}_{ijk}(\phi_e)$  folgt für die gewählte Konvention der Minimalkoordinatendarstellung  $i, j, k \in \{x, y, z\}$  aus dem Vektor der Drehwinkelgeschwindigkeiten  $\boldsymbol{\omega}_0^e$  in der Form

$$\mathbf{T}(\phi_e) = \mathbf{T}_{ijk}(\phi_e) = \frac{\partial}{\partial \dot{\phi}} \boldsymbol{\omega}_0^e \quad (2.51)$$

und lautet für die klassischen EULER-Winkel

$$\mathbf{T}_{zyz}(\phi_e) = \begin{bmatrix} 0 & -\sin(\phi) & \cos(\phi) \sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \sin(\theta) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \quad (2.52)$$

und für die RPY-Winkel

$$\mathbf{T}_{zyx}(\phi_e) = \begin{bmatrix} 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 1 & 0 & -\sin(\theta) \end{bmatrix}. \quad (2.53)$$

Schließlich folgt mit (2.50) der Zusammenhang zwischen den Manipulator Jacobi-Matrizen  $\mathbf{J}(\mathbf{q})$  und  $\mathbf{J}_A(\mathbf{q})$

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \omega_0^e \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \mathbf{T}(\phi_e) \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\phi_e) \end{bmatrix} \underbrace{\begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\phi}_e \end{bmatrix}}_{\dot{\mathbf{x}}_e} = \underbrace{\begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\phi_e) \end{bmatrix}}_{\mathbf{J}(\mathbf{q})} \mathbf{J}_A(\mathbf{q}) \dot{\mathbf{q}} \quad (2.54)$$

mit der Einheitsmatrix  $\mathbf{E}$ . Der Zusammenhang (2.54) zeigt also, dass die geometrische und die analytische Manipulator 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  $\mathbf{T}(\phi_e)$  zum Umrechnen benötigt, siehe (2.50).

**Beispiel 2.6 (Konstante Drehwinkelgeschwindigkeiten).** Dieses Beispiel untersucht den nichtlinearen Zusammenhang (2.50) zwischen dem Vektor der Drehwinkelgeschwindigkeiten  $\omega_0^e$  und der Drehrate  $\dot{\phi}_e$  in Minimalkoordinatendarstellung mit RPY-Winkel. Dazu wird ein Starrkörper mit der Orientierung  $\phi_e^T(0) = [0, 0, 0]$  zum Zeitpunkt  $t = 0$  in Rotation mit konstanter Drehrate versetzt.

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

$$\phi_e(t) = \int_0^t \dot{\phi}_e(\tau) d\tau = \begin{bmatrix} t \\ 0.5t \\ 0 \end{bmatrix} \text{ rad}, \quad (2.55)$$

während  $\omega_0^e(t)$  zu jedem Zeitpunkt mithilfe von  $\phi_e(t)$  und  $\dot{\phi}_e$  aus (2.50) berechnet werden kann. Die Zeitverläufe dieser drei Vektoren sind in Abbildung 2.11 dargestellt. Durch die nichtlineare Abbildung über die Matrix  $\mathbf{T}_{zyx}(\phi_e)$  aus (2.53) schwankt der Vektor der Drehwinkelgeschwindigkeit  $\omega_0^e(t)$  sinusförmig zufolge der konstanten Drehrate in RPY-Winkel.

Im zweiten Schritt wird ein konstanter Vektor der Drehwinkelgeschwindigkeiten  $\omega_0^e = [1, 0.5, 0]^T$  rad/s vorgegeben und die zugehörige Drehung in Minimalkoordinatendarstellung berechnet. Dazu wird (2.50) auf  $\dot{\phi}_e$  umgeformt und gemäß

$$\phi_e(t) = \int_0^t \dot{\phi}_e(\tau) d\tau = \int_0^t \mathbf{T}_{zyx}^{-1}(\phi_e(\tau)) \omega_0^e d\tau \quad (2.56)$$

integriert. In Abbildung 2.12 sind die Zeitverläufe für diese Drehung veranschaulicht. Darin ist deutlich zu sehen, dass der konstante Vektor der Drehwinkelgeschwindigkeiten  $\omega_0^e$  auf eine schwankende Drehrate  $\dot{\phi}_e$  in Minimalkoordinatendarstellung führt. Weiters entsteht durch die Verkopplung über die Matrix  $\mathbf{T}_{zyx}(\phi_e)$  in (2.56) eine Drehung in allen drei Dimensionen von  $\phi_e$ .

Dieses Beispiel zeigt, dass durch den nichtlinearen Zusammenhang (2.50) konstante Geschwindigkeiten  $\omega_0^e$  *nicht* auf konstante Geschwindigkeiten  $\dot{\phi}_e$  abgebildet werden und umgekehrt.

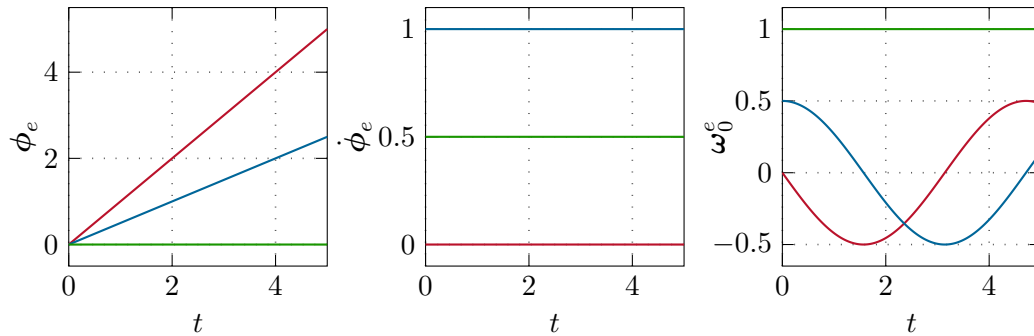


Abbildung 2.11: Drehung eines Starrkörpers mit konstanter Drehrate  $\dot{\phi}_e$  in Minimalkoordinatendarstellung mit RPY-Winkel.

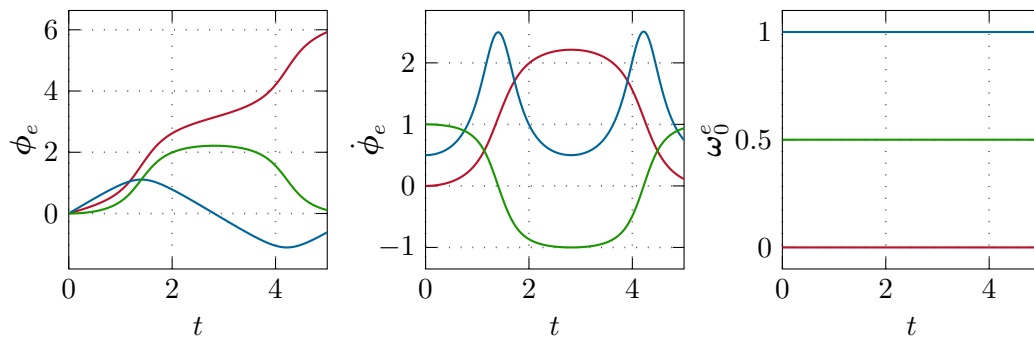


Abbildung 2.12: Drehung eines Starrkörpers mit konstantem Vektor der Drehwinkelgeschwindigkeiten  $\omega_0^e$ .

### 2.3.2 Singularitäten

In der Robotik bezeichnet der Begriff *Singularität* einen Punkt oder Bereich im Konfigurations- oder Arbeitsraum des Roboters bei dem 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 (2.45) gezeigt, bildet die geometrische Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  die Geschwindigkeiten im Konfigurationsraum  $\dot{\mathbf{q}}$  linear auf die Geschwindigkeiten im Arbeitsraum  $[\dot{\mathbf{p}}_e^T (\boldsymbol{\omega}_0^e)^T]^T$  ab, wobei die Jacobi-Matrix selbst im Allgemeinen *nichtlinear* von der Konfiguration  $\mathbf{q}$  abhängt. *Kinematische Singularitäten* bezeichnen jene Konfigurationen  $\mathbf{q}$ , an denen ein Rangverlust im Zeilenrang der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  auftritt. Das bedeutet, dass die Abbildung (2.45) nicht mehr alle Geschwindigkeiten im Arbeitsraum  $[\dot{\mathbf{p}}_e^T (\boldsymbol{\omega}_0^e)^T]^T$  linear unabhängig voneinander erzeugen kann. Für den Fall  $m = n$  ist  $\mathbf{J}(\mathbf{q})$  quadratisch und Konfigurationen  $\mathbf{q}$  mit Rangverlust können direkt aus der Gleichung  $\det(\mathbf{J}(\mathbf{q})) = 0$  ermittelt werden. In der Nähe einer Singularität hat die geometrische Manipulator Jacobi-Matrix zwar vollen Rang, ist aber schlecht konditioniert. Dadurch können in der Nähe einer Singularität große Geschwindigkeiten im Konfigurationsraum auftreten, um bestimmte Bewegungen im Arbeitsraum hervorzurufen.

- **Singularität am Rand des Arbeitsbereichs:** Diese Art von Singularität tritt immer dann auf, wenn der Roboter ganz ausgestreckt oder ganz eingezogen ist. Dies stellt aber keine zusätzliche Einschränkung in der Bewegungsfreiheit des Roboters dar, da der Arbeitsbereich ohnehin durch die mechanische Konstruktion des Roboters beschränkt ist.
- **Singularität im Inneren des Arbeitsbereichs:** Kinematische Singularitäten können auch im Inneren des Arbeitsbereiches auftreten, z. B. wenn die Rotationsachsen von zwei oder mehreren Rotationsgelenken kollinear werden. Eine übergeordnete Pfadplanung muss daher im Vorhinein verhindern, dass der Roboter in solche kinematischen Singularitäten hineinfährt oder in die Nähe von solchen Singularitäten kommt.

**Beispiel 2.7 (Kinematische Singularitäten des 3-Achs-Roboters).** In diesem Beispiel werden die kinematischen Singularitäten des 3-Achs-Roboters aus Beispiel 2.2 betrachtet. Der Roboter mit drei Rotationsgelenken und den körperfesten Koordinatensystemen  $(0_i x_i y_i z_i)$ ,  $i = 1, \dots, 4$  ist in Abbildung 2.6 dargestellt. Aus dieser Abbildung werden die homogenen Transformationen zwischen den Koordinatensystemen in der Form

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

bestimmt und es folgt aus

$$\mathbf{H}_0^4(\mathbf{q}) = \mathbf{H}_0^1(q_1) \mathbf{H}_1^2(q_2) \mathbf{H}_2^3(q_3) \mathbf{H}_3^4 = \begin{bmatrix} \mathbf{R}_0^4(\mathbf{q}) & \mathbf{d}_0^4(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.57)$$

die Vorwärtskinematik des 3-Achs-Roboters gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \mathbf{d}_0^4(\mathbf{q}) = \begin{bmatrix} (d_4 \sin(q_2 + q_3) + d_2 \sin(q_2)) \cos(q_1) \\ (d_4 \sin(q_2 + q_3) + d_2 \sin(q_2)) \sin(q_1) \\ d_4 \cos(q_2 + q_3) + d_2 \cos(q_2) + d_1 \end{bmatrix}. \quad (2.58)$$

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

$$\mathbf{J}(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) = \frac{\partial \mathbf{d}_0^4}{\partial \mathbf{q}} \quad (2.59)$$

und dessen Determinante

$$\det(\mathbf{J}(\mathbf{q})) = d_2 d_4 \underbrace{\sin(q_3)}_{S_1} \underbrace{(d_4 \sin(q_2 + q_3) + d_2 \sin(q_2))}_{S_2} \quad (2.60)$$

berechnet. In (2.60) 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 Abbildung 2.13a) 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 ( $p_x = p_y = 0$ ) und damit ist der Ausdruck  $S_2$  gleich Null. Dies ist in Abbildung 2.13b dargestellt.

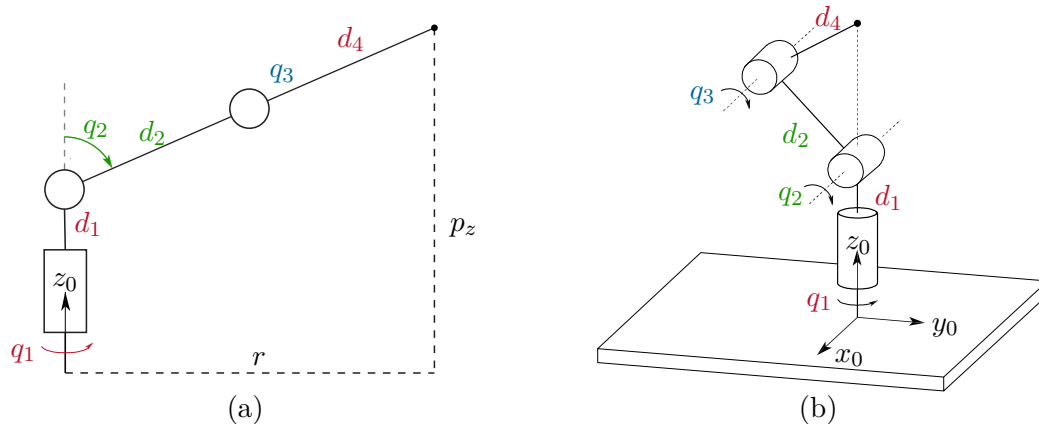


Abbildung 2.13: Kinematische Singularitäten des 3-Achs-Roboters: (a) Ausgestreckter Roboterarm ( $S_1 = 0$ ), (b) Endeffektor liegt auf  $z_0$ -Achse ( $S_2 = 0$ ) [2.2].

**Beispiel 2.8 (Kinematische Singularitäten des sphärischen Handgelenks).** Ein sphärisches Handgelenk weist kinematische Singularitäten auf, welche bei einem 6-Achs-

Roboter mit sphärischem Handgelenk auch im Inneren des Arbeitsbereichs auftreten können. Zur Analyse der kinematischen Kette, siehe Abbildung 2.9 wird daher der Vektor der Drehwinkelgeschwindigkeiten mithilfe von (2.46) aus  $\mathbf{R}_0^e(\mathbf{q})$  zu

$$\boldsymbol{\omega}_0^6(\mathbf{q}) = \begin{bmatrix} \cos(q_4) \sin(q_5) \dot{q}_6 - \dot{q}_5 \sin(q_4) \\ \sin(q_4) \sin(q_5) \dot{q}_6 + \dot{q}_5 \cos(q_4) \\ \dot{q}_6 \cos(q_5) + \dot{q}_4 \end{bmatrix} \quad (2.61)$$

berechnet. Es folgt die zugehörige geometrische Manipulator Jacobi-Matrix zu

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} 0 & -\sin(q_4) & \cos(q_4) \sin(q_5) \\ 0 & \cos(q_4) & \sin(q_4) \sin(q_5) \\ 1 & 0 & \cos(q_5) \end{bmatrix} \quad (2.62)$$

mit der Determinante  $\det(\mathbf{J}(\mathbf{q})) = -\sin(q_5)$ . 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  $x_5$ -Achse (siehe Abbildung 2.9 für  $q_5 = 0$ ) erzeugt werden, da alle Rotationsgelenke orthogonal dazu stehen.

### Repräsentationssingularität

Eine sogenannte *Repräsentationssingularität* ist eine Singularität, die aufgrund der gewählten Koordinatendarstellung im Arbeitsraum auftritt. Die Ursache für diese Art der Singularität liegt daher nicht in der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  (die ausschließlich die kinematische Kette beschreibt), sondern in der analytischen Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  bzw. in der Transformationsmatrix zwischen den beiden Matrizen.

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

$$\det(\mathbf{T}_{zyz}(\boldsymbol{\phi}_e)) = -\sin(\theta) \quad (2.63)$$

und für die RPY-Winkel folgt

$$\det(\mathbf{T}_{zyx}(\boldsymbol{\phi}_e)) = -\cos(\theta) . \quad (2.64)$$

Eine Repräsentationssingularität liegt vor, wenn  $\det(\mathbf{T}(\boldsymbol{\phi}_e)) = 0$  gilt und damit die Inverse  $\mathbf{T}^{-1}(\boldsymbol{\phi}_e)$  nicht existiert. In diesem Fall kann aus einem gegebenen Vektor der Drehwinkelgeschwindigkeiten  $\boldsymbol{\omega}_0^e$  keine zugehörige Winkelgeschwindigkeit der EULER-Winkel  $\dot{\boldsymbol{\phi}}_e$  berechnet werden, siehe (2.54). Zusätzlich degenerieren in einer Repräsentationssingularität auch die Gleichungen (2.13) und (2.14) bzw. (2.16) und (2.17). Es können  $\phi$  und  $\psi$  nicht mehr unabhängig voneinander berechnet werden – nur noch die Summe bzw. die

Differenz von  $\phi$  und  $\psi$ . Diese Singularitäten lassen sich durch eine andere Parametrierung der Drehung (z. B. Quaternionen) beheben.

### 2.3.3 Manipulierbarkeitsellipsoid der Geschwindigkeit

Die geometrische Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  beschreibt den *linearen* Zusammenhang zwischen den Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}$  und den Endeffektorgeschwindigkeiten als translatorische Geschwindigkeit  $\dot{\mathbf{p}}_e$  und Vektor der Drehwinkelgeschwindigkeit  $\boldsymbol{\omega}_0^e$ , siehe (2.45). Die Geschwindigkeit  $\dot{q}_i$  jedes Gelenks  $i = 1, \dots, n$  liefert über den zugehörigen Spaltenvektor  $\mathbf{j}_i(\mathbf{q})$  einen Beitrag zur Gesamtgeschwindigkeit gemäß

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_0^e \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{j}_1(\mathbf{q}) & \mathbf{j}_2(\mathbf{q}) & \cdots & \mathbf{j}_n(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \sum_{i=1}^n \mathbf{j}_i(\mathbf{q}) \dot{q}_i. \quad (2.65)$$

Im Folgenden wird die Menge der Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}$  betrachtet, die die Norm 1 aufweisen. Betrachtet wird daher

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1, \quad (2.66)$$

wodurch eine Einheitskugel im Raum der Gelenksgeschwindigkeiten beschrieben wird. Werden diese Gelenksgeschwindigkeiten auf einen Roboter aufgeschaltet, so folgen unter Verwendung von (2.45) die Endeffektorgeschwindigkeiten im Arbeitsraum des Roboters gemäß

$$\begin{bmatrix} \dot{\mathbf{p}}_e^T & (\boldsymbol{\omega}_0^e)^T \end{bmatrix} \left( \mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}) \right)^{-1} \begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_0^e \end{bmatrix} = 1. \quad (2.67)$$

Es ist zu beachten, dass Gleichung (2.67) eine quadratische Form darstellt und im Allgemeinen ein Ellipsoid im Raum der Endeffektorgeschwindigkeiten beschreibt. Es wird daher auch als *Manipulierbarkeitsellipsoid der Geschwindigkeit* (engl. *velocity manipulability ellipsoid*) bezeichnet. Die Form und die Orientierung des Ellipsoids wird durch die symmetrische Matrix  $\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})$  bestimmt. Dabei sind die Richtungen der Hauptachsen durch die Eigenvektoren gegeben und die Ausdehnung des Ellipsoids entlang jeder Hauptachse berechnet sich als Singulärwert  $\sigma_i$  der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$ , d.h.  $\sigma_i = \sqrt{\lambda_i(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))}$  mit dem  $i$ -ten Eigenwert  $\lambda_i(\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))$ .

Aus den obigen Zusammenhängen kann abgeleitet werden, dass der Roboter entlang der größten Hauptachse die größten Endeffektorgeschwindigkeiten erzielen kann. Bewegungen entlang kleiner Hauptachsen führen nur zu kleinen Endeffektorgeschwindigkeiten. Wird die Länge einer Hauptachse Null, so kann der Roboter in bestimmte Richtungen keine Bewegung mehr ausführen. In diesem Fall befindet sich der Roboter in einer kinematischen Singularität und die geometrische Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  ist singulär. Das folgende Beispiel illustriert diesen Sachverhalt anhand eines 2-Achs-Roboters.

**Beispiel 2.9 (Manipulierbarkeitsellipsen des 2-Achs-Roboters).** In diesem Beispiel wird die differentielle Kinematik eines planaren 2-Achs-Roboters (Abbildung 2.14a) untersucht.

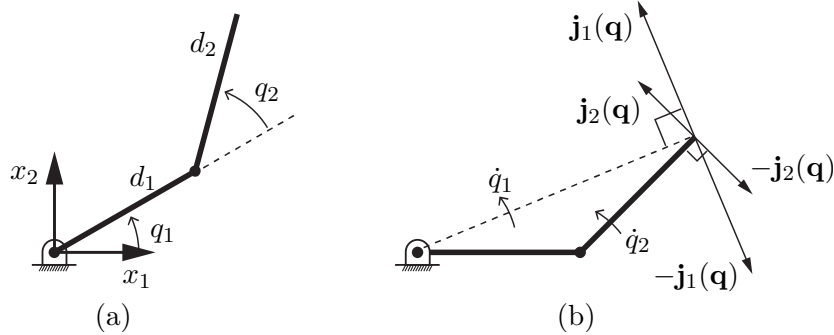


Abbildung 2.14: 2-Achs-Roboter: (a) Roboterkinematik und Gelenkwinkel, (b) Spaltenvektoren der geometrischen Jacobi-Matrix  $\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{j}_1(\mathbf{q}) & \mathbf{j}_2(\mathbf{q}) \end{bmatrix}$  [2.3].

Die Vorwärtskinematik des Roboters lautet

$$\mathbf{p}_e = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} d_1 \cos(q_1) + d_2 \cos(q_1 + q_2) \\ d_1 \sin(q_1) + d_2 \sin(q_1 + q_2) \end{bmatrix} \quad (2.68)$$

und die Endeffektorgeschwindigkeit folgt zu, siehe (2.45) und (2.65)

$$\dot{\mathbf{p}}_e = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{j}_1(\mathbf{q}) \dot{q}_1 + \mathbf{j}_2(\mathbf{q}) \dot{q}_2, \quad (2.69)$$

mit der geometrischen  $2 \times 2$  Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{j}_1(\mathbf{q}) & \mathbf{j}_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} -d_1 \sin(q_1) - d_2 \sin(q_1 + q_2) & -d_2 \sin(q_1 + q_2) \\ d_1 \cos(q_1) + d_2 \cos(q_1 + q_2) & d_2 \cos(q_1 + q_2) \end{bmatrix}. \quad (2.70)$$

Gleichung (2.69) zeigt, dass sich die Endeffektorgeschwindigkeit  $\dot{\mathbf{p}}_e$  als Linearkombination der beiden Spaltenvektoren  $\mathbf{j}_1(\mathbf{q})$  und  $\mathbf{j}_2(\mathbf{q})$  zusammensetzen lässt. Die Spaltenvektoren  $\mathbf{j}_i(\mathbf{q})$  entsprechen den Endeffektorgeschwindigkeiten  $\dot{\mathbf{p}}_e$  für  $\dot{\mathbf{q}} = \mathbf{e}_i$  mit dem Einheitsvektor  $\mathbf{e}_i$ . In Abbildung 2.14b sind die Spaltenvektoren (2.70) für eine Konfiguration  $\mathbf{q}$  des 2-Achs-Roboters eingezeichnet. Solange die beiden Spaltenvektoren nicht parallel sind, können alle Endeffektorgeschwindigkeiten  $\dot{\mathbf{p}}_e$  im Arbeitsraum des Roboters durch eine geeignete Wahl von  $\dot{q}_1$  und  $\dot{q}_2$  hervorgerufen werden. Wie sich leicht zeigen lässt, werden die Spaltenvektoren der geometrischen Manipulator Jacobi-Matrix (2.70) für  $q_2 = k\pi$ ,  $k \in \mathbb{Z}$  parallel. In diesem Fall ist  $\mathbf{J}(\mathbf{q})$  singulär und es handelt sich um eine kinematische Singularität des Roboters.

Gleichung (2.66) definiert einen Einheitskreis im  $\mathbb{R}^2$  im Raum der Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}$ , welcher in Abbildung 2.15a dargestellt ist. Dieser Einheitskreis wird über



die geometrische Manipulator Jacobi-Matrix (2.69) in den Raum der Endeffektorgeschwindigkeit  $\dot{\mathbf{p}}_e$  abgebildet, wodurch die Manipulierbarkeitsellipse (im  $\mathbb{R}^2$ ) entsteht. Abhängig von der Konfiguration  $\mathbf{q}$  des Roboters, nimmt sie eine andere Form ein, siehe Abbildungen 2.15b und 2.15c. Wenn der Roboter in eine Singularität fährt, so reduziert sich die Manipulierbarkeitsellipse auf eine Linie. In diesem Fall kann der Roboter den Endeffektor in eine Richtung orthogonal zu dieser Linie nicht mehr bewegen.

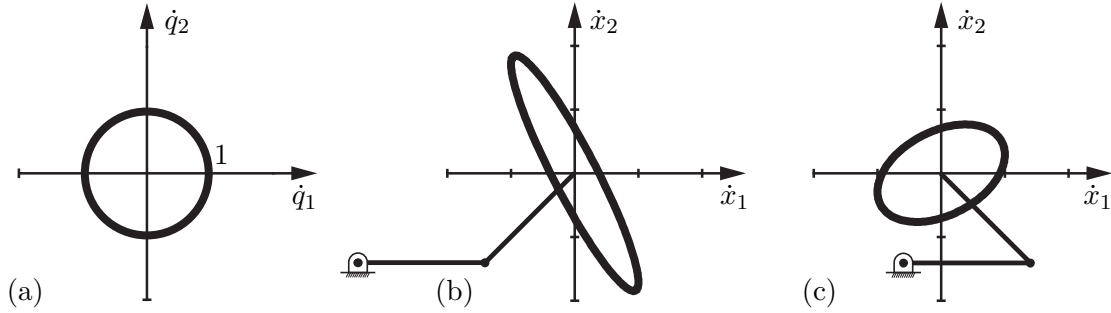


Abbildung 2.15: Manipulierbarkeitsellipsen für einen 2-Achs-Roboter: (a) Einheitskreis im Raum der Gelenksgeschwindigkeiten, (b) und (c) Manipulierbarkeitsellipsen in zwei unterschiedlichen Konfigurationen [2.3].

### 2.3.4 Manipulierbarkeit

Motiviert durch die Manipulierbarkeitsellipsoide (2.67) im vorigen Abschnitt kann die Manipulator Jacobi-Matrix auch dazu benutzt werden, um die sogenannte *Manipulierbarkeit* (engl. *manipulability*) des Roboters in der Form

$$w(\mathbf{q}) = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))} \quad (2.71)$$

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 Arbeitsraums ausführen kann – dies schließt also sowohl Translationen als auch Rotationen mit ein. Der Ausdruck (2.71) wird an den kinematischen Singularitäten Null, siehe Abschnitt 2.3.2. Wie in Abschnitt 2.4.2 gezeigt wird, kann der Gradient von  $w(\mathbf{q})$  dazu benutzt werden, um den Abstand zu Singularitäten bei einem kinematisch redundanten Roboter zu maximieren. Anstelle der geometrischen Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  kann in (2.71) auch die analytische Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  verwendet werden.

### 2.3.5 Manipulierbarkeitsellipsoid der Kraft

Für einen Roboter spielt die Manipulator Jacobi-Matrix auch eine wichtige Rolle bei der Analyse der Statik. Wie in [2.1] gezeigt wurde, gilt der Zusammenhang

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F} \quad (2.72)$$

zwischen den generalisierten Momenten  $\boldsymbol{\tau}$  im Konfigurationsraum und der generalisierten Kraft  $\mathbf{F} \in \mathbb{R}^6$  im Arbeitsraum. Es ist zu beachten, dass dieser Zusammenhang über die *geometrische* Manipulator Jacobi-Matrix  $\mathbf{J}(\mathbf{q})$  hergestellt wird und damit die in  $\mathbf{F}$  enthaltenen Drehmomente korrekt auf  $\boldsymbol{\tau}$  abgebildet werden. Wird nun – analog zu (2.66) – die Einheitskugel im Raum der generalisierten Momente  $\boldsymbol{\tau}$  betrachtet, d.h.

$$\boldsymbol{\tau}^T \boldsymbol{\tau} = 1, \quad (2.73)$$

so folgt mit (2.72)

$$\mathbf{F}^T (\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})) \mathbf{F} = 1. \quad (2.74)$$

Gleichung (2.74) beschreibt das sogenannte *Manipulierbarkeitsellipsoid der Kraft* (engl. *force manipulability ellipsoid*). Im Vergleich zu (2.67) tritt in (2.74) die symmetrische Matrix  $\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q})$  ohne Inversion auf. Dies führt zu dem bemerkenswerten Ergebnis, dass die Hauptachsen der Manipulierbarkeitsellipsoide für die Geschwindigkeit und die Kraft zusammenfallen. Die Längen der jeweiligen Achsen sind jedoch umgekehrt proportional, da die Eigenwerte der inversen Matrix genau den Kehrwerten der ursprünglichen Eigenwerte entsprechen. Es folgt daher, dass Richtungen für gute Manipulierbarkeit der Geschwindigkeit gleichzeitig schlechte Manipulierbarkeit der Kraft zufolge haben.

**Beispiel 2.10 (Manipulierbarkeitsellipsen des 2-Achs-Roboters).** In diesem Beispiel werden die Manipulierbarkeitsellipsen der Geschwindigkeit und der Kraft für den 2-Achs-Roboter, siehe Abbildung 2.14 sowie (2.68) und (2.70), betrachtet. Abbildung 2.16a zeigt den Einheitskreis im Raum der Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}$  und der generalisierten Kräfte  $\boldsymbol{\tau}$ . Für zwei unterschiedliche Konfigurationen  $\mathbf{q}$  sind die Manipulierbarkeitsellipsen in den Abbildungen 2.16b und 2.16c dargestellt. Der Vergleich der beiden Ellipsen zeigt, dass die zwei Hauptachsen beider Ellipsen zueinander ausgerichtet sind, jedoch die Größe der Ellipsen entlang der Hauptachsen jeweils umgekehrt proportional ist.

In einer singulären Konfiguration des 2-Achs-Roboters ( $q_2 = k\pi$ ,  $k \in \mathbb{Z}$ ) wird die Manipulierbarkeitsellipse der Kraft unendlich lang parallel zu den Robotergliedern und sehr schmal orthogonal dazu. Dies bedeutet, dass Kräfte parallel zu den Robotergliedern in diesem Fall ausschließlich über die Struktur des Roboters abgeleitet werden. Es werden keine Drehmomente in den Antrieben  $\boldsymbol{\tau}$  benötigt, um das statische Gleichgewicht zu diesen Kräften zu halten.

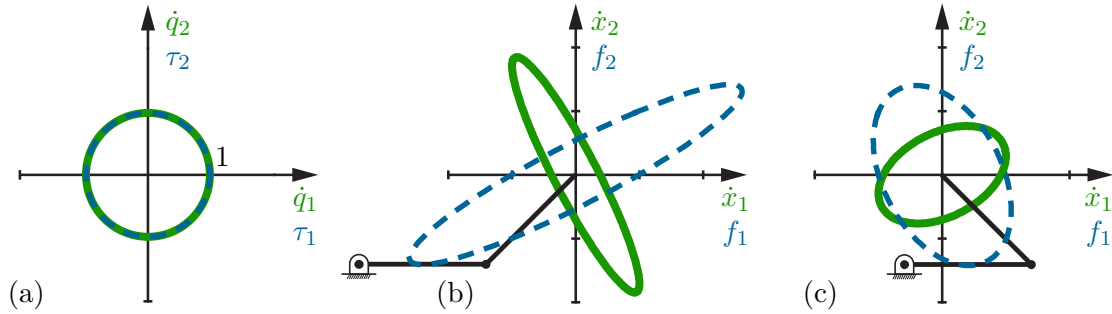


Abbildung 2.16: Manipulierbarkeitsellipsen der Geschwindigkeit (grün) und Kraft (blau strichliert) für einen 2-Achs-Roboter: (a) Einheitskreis im Raum der Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}$  und der generalisierten Kräfte  $\boldsymbol{\tau}$ , (b) und (c) Manipulierbarkeitsellipsen in zwei unterschiedlichen Konfigurationen [2.3].

## 2.4 Differentielle inverse Kinematik

Mit der inversen Kinematik in Abschnitt 2.2 werden Konfigurationen  $\mathbf{q}$  im Konfigurationsraum für eine gegebene Pose  $\mathbf{x}_e$  im Arbeitsraum gesucht. Bei der *differentiellen inversen Kinematik* werden nun zusätzlich die Geschwindigkeiten und Beschleunigungen mit berücksichtigt. Dadurch können nicht nur Lösungen für statische Posen  $\mathbf{x}_e$  gefunden werden, sondern es können ganze Zeitverläufe  $\mathbf{x}_e(t)$ ,  $\dot{\mathbf{x}}_e(t)$  und  $\ddot{\mathbf{x}}_e(t)$  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 nicht-redundante Roboter, d. h. für  $n = m$ , vorgestellt. Dieses Verfahren erlaubt es, die Geschwindigkeiten im Konfigurationsraum zu einer gegebenen Bewegung des Endeffektors im Arbeitsraum  $\mathbf{x}_e(t)$  zu berechnen. Im darauffolgenden Abschnitt wird das Verfahren auf kinematisch redundante Roboter mit  $n > m$  erweitert und im letzten Abschnitt eine Erweiterung auf Beschleunigungen gezeigt.

### 2.4.1 Kinematisch nicht-redundante Roboter $n = m$

Wenn die Anzahl  $m = \dim(\mathbf{x}_e)$  der Koordinaten des Endeffektors im Arbeitsraum  $\mathbf{x}_e$  gleich der Anzahl  $n = \dim(\mathbf{q})$  der mechanischen Freiheitsgrade  $\mathbf{q}$  des Roboters ist, dann ist die analytische Manipulator 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 (2.43)

$$\dot{\mathbf{q}} = \mathbf{J}_A^{-1}(\mathbf{q})\dot{\mathbf{x}}_e \quad (2.75)$$

aus den Geschwindigkeiten im Arbeitsraum  $\dot{\mathbf{x}}_e$  bestimmt werden. Gleichung (2.75) kann nun direkt zur *numerischen Berechnung der inversen Kinematik* herangezogen werden. Im Folgenden bezeichnet der Index  $d$  (engl. für *desired*) eine vorgegebene Sollgröße. Außerdem

wird der Schätzwert einer Größe mit dem Hutsymbol gekennzeichnet, also die Schätzung von  $\mathbf{q}_d$  ist  $\hat{\mathbf{q}}_d$ .

Betrachtet man nun die zeitliche Ableitung der vorgegebenen Bewegung des Endeffektors im Arbeitsraum  $\dot{\mathbf{x}}_{e,d}(t)$  als Eingangsgröße, dann folgt aus (2.75) die Schätzgleichung

$$\frac{d}{dt}\hat{\mathbf{q}}_d = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)\dot{\mathbf{x}}_{e,d} \quad (2.76a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} \quad (2.76b)$$

mit  $\mathbf{q}_{d,0}$  als Wert von  $\hat{\mathbf{q}}_d$  zum Anfangszeitpunkt  $t = 0$ . Dabei wird vorausgesetzt, dass  $\mathbf{x}_{e,d}(t)$  stetig differenzierbar ist. Ersetzt man nun in (2.76) den Differentialquotienten  $\frac{d}{dt}\hat{\mathbf{q}}_d$  durch den Differenzenquotienten

$$\left(\frac{d}{dt}\hat{\mathbf{q}}_d\right)(kT_a) \approx \frac{\hat{\mathbf{q}}_d(kT_a) - \hat{\mathbf{q}}_d((k-1)T_a)}{T_a} \quad (2.77)$$

mit der Abtastzeit  $T_a$  und  $k = 1, 2, 3, \dots$ , dann lautet die zeitdiskrete Schätzgleichung

$$\hat{\mathbf{q}}_d(kT_a) = \hat{\mathbf{q}}_d((k-1)T_a) + T_a \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d(kT_a))\dot{\mathbf{x}}_{e,d}(kT_a) \quad (2.78a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} . \quad (2.78b)$$

Man beachte, dass die Verwendung des einfachen Differenzenquotienten (2.77) in der Literatur auch unter dem Namen *Integration nach dem expliziten Euler-Verfahren* bekannt ist. Natürlich kann man an dieser Stelle auch aufwändigere Integrationsverfahren höherer Ordnung einsetzen, um von (2.76) zu einem zeitdiskreten System mit der Abtastzeit  $T_a$  zu kommen. Der Nachteil dieser Vorgehensweise besteht aber darin, dass die offene Integration von (2.76) bzw. (2.78) im Allgemeinen zu einer *Drift* führt. Dies kann bekanntermaßen durch den Entwurf eines vollständigen Beobachters gelöst werden. Dazu betrachtet man den Fehler  $\hat{\mathbf{e}}_x$  zwischen der vorgegebenen Bewegung im Arbeitsraum des Endeffektors  $\mathbf{x}_{e,d}(t)$  und dem geschätzten Verlauf  $\hat{\mathbf{x}}_{e,d} = \mathbf{f}(\hat{\mathbf{q}}_d)$ , siehe (2.9)

$$\hat{\mathbf{e}}_x = \mathbf{x}_{e,d}(t) - \hat{\mathbf{x}}_{e,d} = \mathbf{x}_{e,d}(t) - \mathbf{f}(\hat{\mathbf{q}}_d) , \quad (2.79)$$

und dessen Zeitableitung, siehe (2.43)

$$\frac{d}{dt}\hat{\mathbf{e}}_x = \dot{\mathbf{x}}_{e,d}(t) - \mathbf{J}_A(\hat{\mathbf{q}}_d)\frac{d}{dt}\hat{\mathbf{q}}_d . \quad (2.80)$$

Wählt man nun als *Beobachter*

$$\frac{d}{dt}\hat{\mathbf{q}}_d = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\hat{\mathbf{e}}_x) = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)(\dot{\mathbf{x}}_{e,d} + \mathbf{K}(\mathbf{x}_{e,d} - \mathbf{f}(\hat{\mathbf{q}}_d))) \quad (2.81a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} \quad (2.81b)$$

mit der positiv definiten Matrix  $\mathbf{K}$ , dann folgt mit (2.81) eingesetzt in (2.80) die asymptotisch stabile Schätzfehlerdynamik

$$\frac{d}{dt}\hat{\mathbf{e}}_x = -\mathbf{K}\hat{\mathbf{e}}_x \quad (2.82a)$$

$$\hat{\mathbf{e}}_x(0) = \hat{\mathbf{e}}_{x,0} = \mathbf{x}_{e,d}(0) - \mathbf{f}(\mathbf{q}_{d,0}) . \quad (2.82b)$$

Zusätzlich werden die Fehlerdynamiken (2.82a) entkoppelt, wenn  $\mathbf{K} = \text{diag}(k_1, \dots, k_m)$  mit positiven Einträgen  $k_j$ ,  $j = 1, \dots, m$  gewählt wird und es folgt

$$\frac{d}{dt} \hat{e}_{x,j} = -k_j \hat{e}_{x,j} , \quad \hat{e}_x(0) = \hat{e}_{x,0} \quad (2.83)$$

Damit nehmen die Fehler mit den frei vorgebbaren Zeitkonstanten  $k_j > 0$  exponentiell gemäß

$$\hat{e}_{x,j}(t) = \hat{e}_{x,0} \exp(-k_j t) . \quad (2.84)$$

ab. Der Beobachter (2.81) kann analog zu (2.78) wiederum zeitdiskret implementiert werden.

### 2.4.2 Kinematisch redundante Roboter $n > m$

Für den Fall eines kinematisch redundanten Roboters, d. h.  $n > m$ , hat die analytische Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  gemäß (2.44)  $n$  Spalten und  $m$  Zeilen und kann nicht mehr wie in (2.75) direkt invertiert werden. Die zusätzlichen Freiheitsgrade werden nun so genutzt, dass nach wie vor das Gleichungssystem (2.43) erfüllt wird, aber gleichzeitig auch die Minimierungsaufgabe

$$\min_{\dot{\mathbf{q}}} V_n(\dot{\mathbf{q}}) = \min_{\dot{\mathbf{q}}} \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n) \quad (2.85)$$

mit dem noch frei zu wählenden Vektor der Gelenksgeschwindigkeiten  $\dot{\mathbf{q}}_n$  gelöst wird. Diese Minimierungsaufgabe mit der Gleichungsnebenbedingung (2.43) löst man durch Minimierung der Funktion

$$L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n) + \boldsymbol{\lambda}^T (\dot{\mathbf{x}}_e - \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}) \quad (2.86)$$

bezüglich  $\dot{\mathbf{q}}$  und  $\boldsymbol{\lambda}$ . 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

$$\frac{\partial}{\partial \dot{\mathbf{q}}} L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = (\dot{\mathbf{q}} - \dot{\mathbf{q}}_n)^T - \boldsymbol{\lambda}^T \mathbf{J}_A(\mathbf{q}) = \mathbf{0} \quad (2.87a)$$

$$\frac{\partial}{\partial \boldsymbol{\lambda}} L_n(\dot{\mathbf{q}}, \boldsymbol{\lambda}) = (\dot{\mathbf{x}}_e - \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}) = \mathbf{0} \quad (2.87b)$$

gegeben. Löst man (2.87a) nach  $\dot{\mathbf{q}}$  gemäß

$$\dot{\mathbf{q}} = \mathbf{J}_A^T(\mathbf{q})\boldsymbol{\lambda} + \dot{\mathbf{q}}_n \quad (2.88)$$

auf und setzt dies in (2.87b) ein, so erhält man

$$\dot{\mathbf{x}}_e = \mathbf{J}_A(\mathbf{q})\mathbf{J}_A^T(\mathbf{q})\boldsymbol{\lambda} + \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}_n . \quad (2.89)$$

Außerhalb von Singularitäten ist die analytische Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  für einen redundanten Roboter *zeilenregulär*, d. h. die  $m$  Zeilen sind im Arbeitsraum 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 (2.89) gemäß

$$\boldsymbol{\lambda} = \left( \mathbf{J}_A(\mathbf{q})\mathbf{J}_A^T(\mathbf{q}) \right)^{-1} (\dot{\mathbf{x}}_e - \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}}_n) \quad (2.90)$$

berechnen. Eingesetzt in (2.88) folgt damit die Lösung der Minimierungsaufgabe (2.85) unter der Gleichungsnebenbedingung (2.43) zu

$$\dot{\mathbf{q}} = \mathbf{J}_A^\dagger \dot{\mathbf{x}}_e + (\mathbf{E} - \mathbf{J}_A^\dagger \mathbf{J}_A) \dot{\mathbf{q}}_n \quad (2.91)$$

mit der Einheitsmatrix  $\mathbf{E}$  und der *rechten Pseudoinversen*

$$\mathbf{J}_A^\dagger = \mathbf{J}_A^T (\mathbf{J}_A \mathbf{J}_A^T)^{-1}. \quad (2.92)$$

**Bemerkung 2.1.** Die rechte Pseudoinverse  $\mathbf{J}_A^\dagger$  in (2.91) wird oft auch durch  $\mathbf{J}_A^* = \mathbf{J}_A^T (\mathbf{J}_A \mathbf{J}_A^T + k^2 \mathbf{E})^{-1}$  mit dem Dämpfungsfaktor  $k$  ersetzt, damit in der Nähe von Singularitäten die Invertierbarkeit von  $\mathbf{J}_A \mathbf{J}_A^T$  sichergestellt wird.

Der Term  $(\mathbf{E} - \mathbf{J}_A^\dagger \mathbf{J}_A) \dot{\mathbf{q}}_n$  in (2.91) 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. Zum Beispiel können so die Manipulierbarkeit, der Abstand zu Hindernissen oder die Abstände der Gelenkwinkel zu Gelenkwinkelbeschränkungen maximiert werden. Ein allgemeiner Ansatz für  $\dot{\mathbf{q}}_n$  in (2.91) ist

$$\dot{\mathbf{q}}_n = k_0 \left( \frac{\partial}{\partial \mathbf{q}} w(\mathbf{q}) \right)^T, \quad k_0 > 0 \quad (2.93)$$

mit der Funktion  $w(\mathbf{q})$ , welche das Kriterium für die jeweilige Anwendung definiert. 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 (2.91) erfüllt wird. Um die Manipulierbarkeit (und damit den Abstand zu Singularitäten) zu maximieren, wird  $w(\mathbf{q})$  gemäß (2.71) gewählt. Mit

$$w(\mathbf{q}) = -\frac{1}{2n} \sum_{i=1}^n \left( \frac{q_i - \bar{q}_i}{q_{i,max} - q_{i,min}} \right)^2 \quad (2.94)$$

werden die Abstände der einzelnen Gelenkwinkel  $q_i$  zu den jeweiligen Gelenkwinkelbeschränkungen  $q_{i,max}$  und  $q_{i,min}$ ,  $i = 1, \dots, n$  maximiert. Weiters bezeichnet  $\bar{q}_i$  in Gleichung (2.94) den Mittelwert des Gelenkwinkelbereichs für das Gelenk  $i$ .

Mit der gewünschten Bewegung des Endeffektors  $\dot{\mathbf{x}}_{e,d}(t)$  als Eingangsgröße kann analog zu (2.76) basierend auf (2.91) eine Schätzgleichung für den kinematisch redundanten Fall

in der Form

$$\frac{d}{dt}\hat{\mathbf{q}}_d = \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d)\dot{\mathbf{x}}_{e,d} + \left(\mathbf{E} - \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d)\mathbf{J}_A(\hat{\mathbf{q}}_d)\right)\dot{\mathbf{q}}_n \quad (2.95a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} \quad (2.95b)$$

angegeben werden mit der rechten Pseudoinversen  $\mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d)$  und dem Vektor der Geschwindigkeiten im Nullraum  $\dot{\mathbf{q}}_n$ . Der Beobachter für (2.95) lautet analog zu (2.81) für den kinematisch redundanten Roboter

$$\frac{d}{dt}\hat{\mathbf{q}}_d = \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d)(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\hat{\mathbf{e}}_x) + \left(\mathbf{E} - \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d)\mathbf{J}_A(\hat{\mathbf{q}}_d)\right)\dot{\mathbf{q}}_n \quad (2.96a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} \quad (2.96b)$$

mit  $\mathbf{K} = \text{diag}(k_1, \dots, k_m)$ ,  $k_j > 0$ ,  $j = 1, \dots, m$ . Sowohl der Schätzer (2.95) als auch der Beobachter (2.96) für den kinematisch redundanten Fall lassen sich entsprechend (2.78) zeitdiskret implementieren.

### 2.4.3 Höhere Ableitungen

Mithilfe der beiden vorgestellten Beobachter (2.81) und (2.96) kann eine vorgegebene Bewegung des Endeffektors  $\mathbf{x}_{e,d}(t)$  in den zugehörigen zeitlichen Verlauf im Konfigurationsraum  $\mathbf{q}_d(t)$  für einen kinematisch nicht-redundanten bzw. redundanten Roboter umgerechnet werden. Die Beobachter liefern dabei eine Schätzung für die Position  $\hat{\mathbf{q}}_d(t)$  und die Geschwindigkeit  $\dot{\hat{\mathbf{q}}}_d(t)$  im Konfigurationsraum. Viele modellbasierte Regelungskonzepte, Trajektorien- und Pfadplanungsalgorithmen benötigen aber zusätzlich auch die Beschleunigungen im Konfigurationsraum  $\ddot{\hat{\mathbf{q}}}_d(t)$ . Daher wird im Folgenden ein Beobachter zweiter Ordnung hergeleitet.

Der Zusammenhang zwischen den Beschleunigungen im Konfigurationsraum und im Arbeitsraum wird als Zeitableitung von (2.43) gemäß

$$\ddot{\mathbf{x}}_e = \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \quad (2.97)$$

gefunden. Für kinematisch nicht-redundante Roboter lässt sich die analytische Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  außerhalb von Singularitäten invertieren und es folgt die Schätzgleichung für die Beschleunigungen im Konfigurationsraum

$$\frac{d^2}{dt^2}\hat{\mathbf{q}}_d = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)\left(\ddot{\mathbf{x}}_{e,d} - \dot{\mathbf{J}}_A(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}}_d)\dot{\hat{\mathbf{q}}}_d\right) \quad (2.98a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0}, \quad \dot{\hat{\mathbf{q}}}_d(0) = \dot{\mathbf{q}}_{d,0} \quad (2.98b)$$

Nun lässt sich der Beobachter

$$\frac{d^2}{dt^2}\hat{\mathbf{q}}_d = \mathbf{J}_A^{-1}(\hat{\mathbf{q}}_d)\left(\ddot{\mathbf{x}}_{e,d} + \mathbf{K}_1\dot{\hat{\mathbf{e}}}_x + \mathbf{K}_0\hat{\mathbf{e}}_x - \dot{\mathbf{J}}_A(\hat{\mathbf{q}}_d, \dot{\hat{\mathbf{q}}}_d)\dot{\hat{\mathbf{q}}}_d\right) \quad (2.99a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0}, \quad \dot{\hat{\mathbf{q}}}_d(0) = \dot{\mathbf{q}}_{d,0}, \quad (2.99b)$$

mit dem Schätzfehler  $\hat{\mathbf{e}}_x$  nach (2.79) und dessen Zeitableitung  $\dot{\hat{\mathbf{e}}}_x$  nach (2.80) sowie den diagonalen Verstärkungsmatrizen mit positiven Einträgen  $\mathbf{K}_0 = \text{diag}(k_{0,1}, k_{0,2}, \dots, k_{0,m})$

und  $\mathbf{K}_1 = \text{diag}(k_{1,1}, k_{1,2}, \dots, k_{1,m})$  anschreiben. Wird dieser Beobachter (2.99) in (2.97) eingesetzt, so folgt die Fehlerdynamik zu

$$\ddot{\mathbf{e}}_x + \mathbf{K}_1 \dot{\mathbf{e}}_x + \mathbf{K}_0 \mathbf{e}_x = \mathbf{0} , \quad (2.100)$$

bzw.

$$\ddot{e}_{x,j} + k_{1,j} \dot{e}_{x,j} + k_{0,j} e_{x,j} = 0 , \quad j = 1, \dots, m , \quad (2.101)$$

welche mit  $k_{0,j} > 0$ ,  $k_{1,j} > 0$ ,  $j = 1, \dots, m$  asymptotisch stabil ist.

Wie in Abschnitt 2.4.2 vorgestellt wurde, kann auch der Beobachter zweiter Ordnung (2.99) mithilfe einer Projektionsmatrix  $(\mathbf{E} - \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d) \mathbf{J}_A(\hat{\mathbf{q}}_d))$  für kinematisch redundante Roboter in der Form

$$\frac{d^2}{dt^2} \hat{\mathbf{q}}_d = \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d) (\ddot{\mathbf{x}}_d + \mathbf{K}_1 \dot{\mathbf{e}} + \mathbf{K}_0 \mathbf{e} - \dot{\mathbf{J}}_A(\hat{\mathbf{q}}_d, \dot{\hat{\mathbf{q}}}_d) \dot{\hat{\mathbf{q}}}_d) + (\mathbf{E} - \mathbf{J}_A^\dagger(\hat{\mathbf{q}}_d) \mathbf{J}_A(\hat{\mathbf{q}}_d)) \ddot{\mathbf{q}}_n \quad (2.102a)$$

$$\hat{\mathbf{q}}_d(0) = \mathbf{q}_{d,0} , \quad \dot{\hat{\mathbf{q}}}_d(0) = \dot{\mathbf{q}}_{d,0} , \quad (2.102b)$$

erweitert werden. Die Beschleunigungen im Nullraum in (2.102a) werden analog zu (2.93) gemäß

$$\ddot{\mathbf{q}}_n = k_0 \left( \frac{\partial}{\partial \mathbf{q}} w(\mathbf{q}) \right)^\top , \quad k_0 > 0 , \quad (2.103)$$

mit einem (anwendungsspezifisch gewählten) sekundären Kriterium  $w(\mathbf{q})$  berechnet.



## 2.5 Literatur

- [2.1] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2020)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2020. Adresse: <https://www.acin.tuwien.ac.at/bachelor/modellbildung/>.
- [2.2] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.
- [2.3] 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>.



## 3 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 im Arbeitsraum 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. Arbeitsraum 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 Gelenkwinkel und Gelenkwinkelgeschwindigkeiten sowie die Beschleunigungen und Drehmomente müssen eingehalten werden.
- Die Berechnung einer Trajektorie kann viel Zeit in Anspruch nehmen, wenn viele Randbedingungen wie Hindernisse im Arbeitsraum oder Zeitoptimalität berücksichtigt werden sollen. Die berechneten Trajektorien müssen hingegen einfach auszuwerten sein, damit dies während der Ausführung in Echtzeit passieren kann.

In diesem Kapitel wird eine allgemeine Trajektorie aus zwei Teilen zusammengesetzt, nämlich einem *Pfad*, also einer rein geometrischen Beschreibung der Bewegung im Konfigurations- oder Arbeitsraum, und einer *Zeitparametrierung*. In den weiteren Abschnitten werden einerseits geradlinige Bewegungen zwischen einem Start- und Zielpunkt berechnet und andererseits Bewegungen mit Zwischenpunkten, sogenannten *Via-Punkten*, geplant.

### 3.1 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 Arbeitsraum 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.

Ohne Beschränkung der Allgemeinheit wird die Trajektorienplanung im Weiteren für eine skalare Funktion  $x(t)$  vorgestellt. Die folgenden Konzepte können gleichermaßen auf Zeitfunktionen im  $n$ -dimensionalen Konfigurationsraum  $\mathbf{q}(s)$  und  $m$ -dimensionalen Arbeitsraum  $\mathbf{x}_e(s)$  angewendet werden.

Ein Pfad  $x(s)$  zusammen mit einer Zeitparametrierung  $s(t)$  spezifizieren eine *Trajektorie*  $x(s(t))$  für eine Koordinate im Konfigurations- oder Arbeitsraum. Die zugehörige Geschwindigkeit und Beschleunigung als Funktion der Zeit werden unter Verwendung der Kettenregel gemäß

$$x = x(s(t)) \quad (3.1a)$$

$$\dot{x} = \frac{dx}{ds} \dot{s} \quad (3.1b)$$

$$\ddot{x} = \frac{dx}{ds} \ddot{s} + \frac{d^2x}{ds^2} \dot{s}^2 \quad (3.1c)$$

berechnet.

## 3.2 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. 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. Dadurch ist es immer möglich, die mechanischen Beschränkungen der Gelenkwinkel  $q_{i,min} \leq q_i \leq q_{i,max}$ ,  $i = 1, \dots, n$ , direkt zu berücksichtigen. Solche geradlinigen Bewegungen sind auch im Arbeitsraum des Roboters möglich und diese werden als *Linearbewegung* (engl. *linear motion*, LIN) bezeichnet. Bei einer Linearbewegung ist hingegen nicht garantiert, dass die Gelenkwinkelbeschränkungen immer erfüllt sind.

Für eine skalare Koordinate  $x$  im Konfigurations- oder Arbeitsraum wird ein geradliniger Pfad  $x(s)$  von einem Startpunkt  $x_0$  zu einem Zielpunkt  $x_1$  mit  $s \in [0, 1]$  in der Form

$$x(s) = x_0 + s(x_1 - x_0), \quad s \in [0, 1] \quad (3.2)$$

mit den Ableitungen, vgl. (3.1)

$$\frac{dx}{ds} = x_1 - x_0 \quad (3.3a)$$

$$\frac{d^2x}{ds^2} = 0 \quad (3.3b)$$

definiert. Eine geradlinige Bewegung im Konfigurationsraum führt im Allgemeinen *nicht* zu einer geradlinigen Bewegung im Arbeitsraum aufgrund des im Allgemeinen nichtlinearen Zusammenhangs über die Roboterkinematik (1.3) bzw. (1.4). Dies soll im Folgenden anhand des Beispiels eines 2-Achs-Roboters gezeigt werden.

**Beispiel 3.1 (Geradlinige Pfade eines 2-Achs-Roboters).** In diesem Beispiel sollen geradlinige Pfade für einen 2-Achs-Roboter sowohl im Konfigurationsraum als auch im Arbeitsraum 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 3.1a 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 (3.2) für jedes Gelenk einzeln geplant, d. h.

$$\mathbf{q}(s) = \begin{bmatrix} q_1(s) \\ q_2(s) \end{bmatrix} = \begin{bmatrix} q_{0,1} + s(q_{1,1} - q_{0,1}) \\ q_{0,2} + s(q_{1,2} - q_{0,2}) \end{bmatrix}, \quad s \in [0, 1], \quad (3.4)$$

bzw. in Vektorschreibweise

$$\mathbf{q}(s) = \mathbf{q}_0 + s(\mathbf{q}_1 - \mathbf{q}_0), \quad s \in [0, 1]. \quad (3.5)$$

In den Abbildungen 3.1b und 3.1c ist der durch die Gelenkwinkelbeschränkungen zulässige Konfigurationsraum bzw. Arbeitsraum als hellgraue Fläche dargestellt. In Abbildung 3.1b 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 Arbeitsraum, siehe Abbildung 3.1c.

**Arbeitsraum** Mithilfe der Vorwärtskinematik des 2-Achs-Roboters (2.68) werden zunächst der Start- und der Endpunkt  $\mathbf{q}_0$  bzw.  $\mathbf{q}_1$  in den Arbeitsraum gemäß

$$\mathbf{x}_0 = \mathbf{f}(\mathbf{q}_0) = [x_{0,1} \ x_{0,2}]^T \quad (3.6)$$

$$\mathbf{x}_1 = \mathbf{f}(\mathbf{q}_1) = [x_{1,1} \ x_{1,2}]^T \quad (3.7)$$

übertragen. Analog zu (3.2) wird nun im Arbeitsraum ein geradliniger Pfad für den Endeffektor  $\mathbf{x}_e(s)$

$$\mathbf{x}_e(s) = \begin{bmatrix} x_{e,1}(s) \\ x_{e,2}(s) \end{bmatrix} = \begin{bmatrix} x_{0,1} + s(x_{1,1} - x_{0,1}) \\ x_{0,2} + s(x_{1,2} - x_{0,2}) \end{bmatrix} = \mathbf{x}_0 + s(\mathbf{x}_1 - \mathbf{x}_0), \quad s \in [0, 1], \quad (3.8)$$

geplant. Um diese Bewegung auszuführen, kann einerseits der Beobachter in Abschnitt 2.4.1 verwendet werden. Damit wird zunächst die Roboterbewegung in den Konfigurationsraum umgerechnet und anschließend kann diese mithilfe eines Reglers im Konfigurationsraum (siehe Abschnitt 1.3.1) ausgeführt werden. Andererseits kann die Regelung auch im Arbeitsraum erfolgen (siehe Abschnitt 1.3.2) und der Pfad (3.8) zusammen mit einer geeigneten Zeitparametrierung  $s(t)$  kann direkt im geschlossenen Regelkreis vorgegeben werden.

Der geradlinige Pfad im Arbeitsraum des 2-Achs-Roboters ist in Abbildung 3.1e dargestellt, während der zugehörige kurvige Pfad im Konfigurationsraum in Abbildung 3.1d 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 Arbeitsraum auf die Einhaltung der mechanischen und physikalischen Beschränkungen der Gelenkwinkel, -geschwindigkeiten und -beschleunigungen geachtet werden muss.

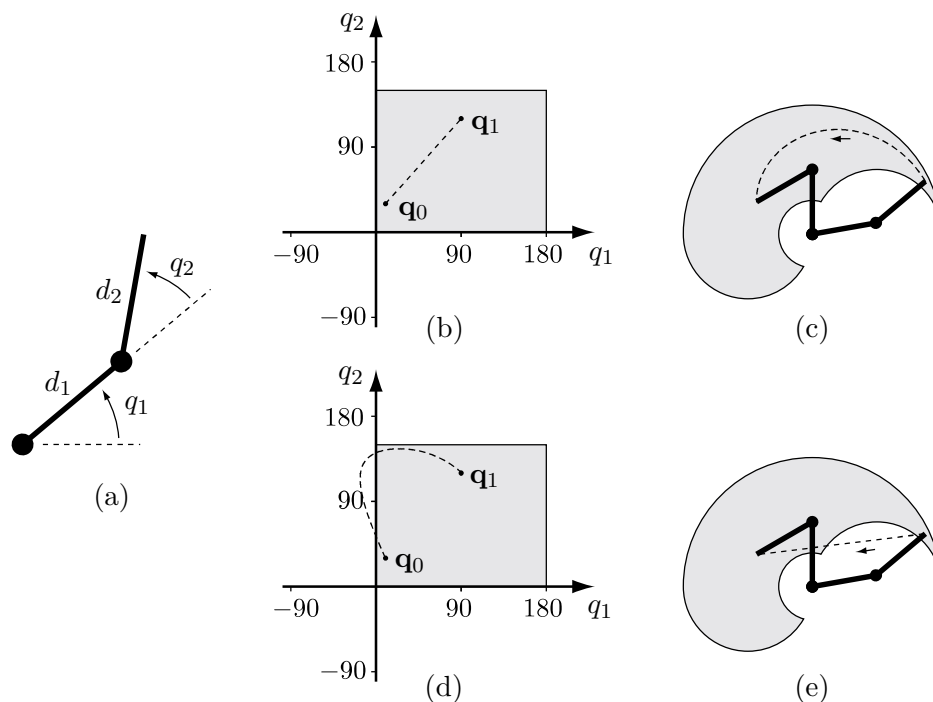


Abbildung 3.1: Geradlinige Pfade eines 2-Achs-Roboters: (a) Roboterkinematik und Gelenkwinkel, (b) geradliniger Pfad im Konfigurationsraum, (c) zugehöriger kurviger Pfad im Arbeitsraum, (d) kurviger Pfad im Konfigurationsraum welcher aus einem geradlinigen Pfad im Arbeitsraum resultiert, (e) geradliniger Pfad im Arbeitsraum, welcher die Gelenkwinkelbeschränkungen verletzt [3.1].

### 3.3 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

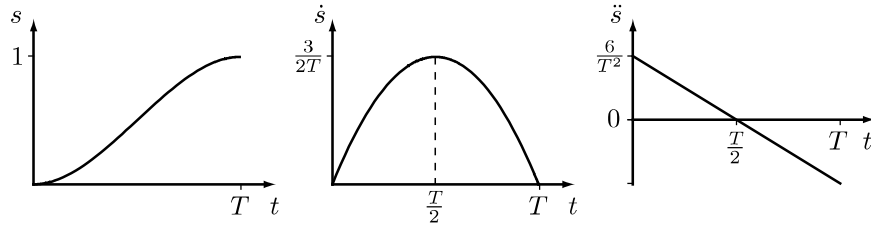


Abbildung 3.2: Schematische Darstellung der Funktionen  $s(t)$ ,  $\dot{s}(t)$  und  $\ddot{s}(t)$  für die Zeitparametrierung mithilfe eines Polynoms 3. Ordnung [3.1].

auch die Beschränkungen für die Geschwindigkeit und die Beschleunigung aller Gelenke eingehalten werden.

Mit einer allgemeinen Zeitparametrierung  $s(t)$  ergibt sich aus (3.1) für den geradlinigen Pfad (3.2) mit (3.3) die Trajektorie

$$x = x_0 + s(x_1 - x_0) \quad (3.9a)$$

$$\dot{x} = \dot{s}(x_1 - x_0) \quad (3.9b)$$

$$\ddot{x} = \ddot{s}(x_1 - x_0) . \quad (3.9c)$$

Im Folgenden werden zwei unterschiedliche Zeitparametrierungen  $s(t)$  vorgestellt. Einerseits wird eine Zeitparametrierung mithilfe von Polynomen 3. und 5. Ordnung konstruiert, mit welchen bestimmte Anfangs- und Endbedingungen erfüllt werden. Andererseits wird in der zweiten Methode eine Bewegung intuitiv durch ein trapezförmiges Geschwindigkeitsprofil bestehend aus einer Beschleunigungsphase, einer Phase konstanter Geschwindigkeit und einer Verzögerungsphase (Bremsphase) beschrieben.

### 3.3.1 Polynom 3. Ordnung

Für die Zeitparametrierung mithilfe eines Polynoms 3. Ordnung in der Form

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (3.10a)$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (3.10b)$$

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

$$a_0 = 0 , \quad a_1 = 0 , \quad a_2 = \frac{3}{T^2} , \quad a_3 = -\frac{2}{T^3} . \quad (3.11)$$

Die Trajektorie (3.9) zusammen mit (3.10) und (3.11) ist damit vollständig spezifiziert und in Abbildung 3.2 schematisch dargestellt. Aus (3.10b) wird der Zeitpunkt  $t = T/2$  bestimmt, bei dem die maximale Geschwindigkeit

$$|\dot{x}_{max}| = \frac{3}{2T} |x_1 - x_0| \quad (3.12)$$

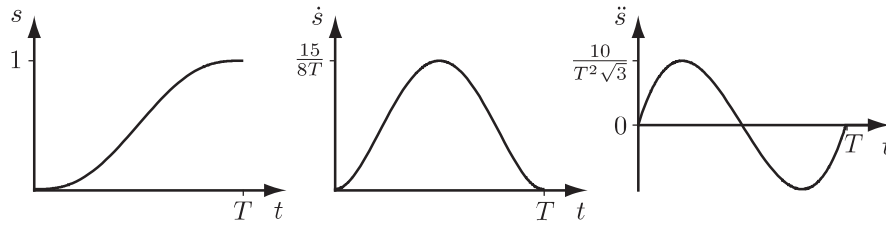


Abbildung 3.3: Schematische Darstellung der Funktionen  $s(t)$ ,  $\dot{s}(t)$  und  $\ddot{s}(t)$  für die Zeitparametrierung mithilfe eines Polynoms 5. Ordnung [3.1].

auftritt, siehe (3.9b) Die maximale Beschleunigung ist

$$|\ddot{x}_{max}| = \frac{6}{T^2} |x_1 - x_0|, \quad (3.13)$$

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 (3.12) bzw. (3.13) berechnet wird.

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 die Vorgabe einer Geschwindigkeit, aber nicht für die Vorgabe einer Beschleunigung. Weiters haben alle polynomialen Ansätze die Eigenschaft, dass die maximale Geschwindigkeit nur an einem Zeitpunkt erreicht wird und sie eine lange Beschleunigungs- und Verzögerungsphase aufweisen. Damit wird die Leistungsfähigkeit des Robotersystems nicht optimal ausgenutzt. Diese nachteiligen Eigenschaften werden durch trapezförmige Bewegungsprofile behoben, wie in Abschnitt 3.3.3 gezeigt wird.

### 3.3.2 Polynom 5. Ordnung

Wird die Zeitparametrierung als Polynom 5. Ordnung in der Form

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3.14a)$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (3.14b)$$

$$\ddot{s}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad (3.14c)$$

gewählt, so können im Vergleich zum Polynom 3. Ordnung noch zusätzlich die zweite Ableitung am Anfang und am Ende der Roboterbewegung mit  $\ddot{s}(0) = \ddot{s}(T) = 0$  vorgegeben werden. Aus diesen Bedingungen werden die Koeffizienten  $a_0, \dots, a_5$  des Polynoms (3.14a) gemäß

$$a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = \frac{10}{T^3}, \quad a_4 = -\frac{15}{T^4}, \quad a_5 = \frac{6}{T^5} \quad (3.15)$$

berechnet. In der schematischen Darstellung in Abbildung 3.3 ist ersichtlich, dass die zweite Ableitung  $\ddot{s}(t)$  nun ebenfalls bei Null beginnt und endet. Damit ist diese Zeitparametrierung geeignet, um Beschleunigungen für die Bewegungen eines Robotersystems vorzugeben.



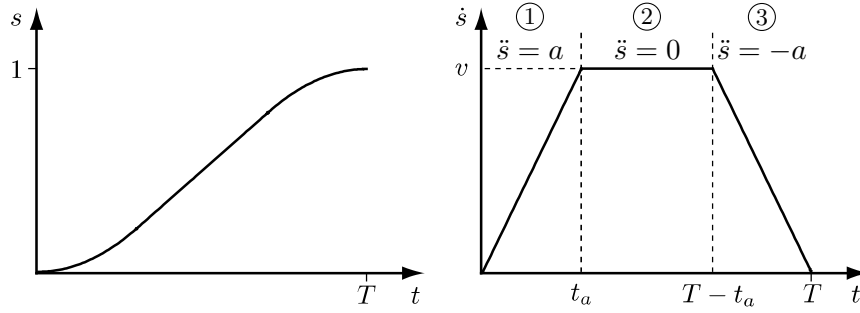


Abbildung 3.4: Zeitparametrierung mit trapezförmigen Bewegungsprofil: ① Beschleunigungsphase, ② Phase konstanter Geschwindigkeit  $v$ , ③ Verzögerungsphase [3.1].

### 3.3.3 Trapezförmige Bewegungsprofile

Bei den trapezförmigen Bewegungsprofilen weist die erste Ableitung der Zeitparametrierung  $\dot{s}(t)$  oder die Geschwindigkeit der Trajektorie  $\dot{x}(t)$  einen trapezförmigen Verlauf auf, wie Abbildung 3.4 zeigt. Trapezförmige Bewegungsprofile bestehen aus drei Phasen, nämlich einer Beschleunigungsphase ①, einer Phase konstanter Geschwindigkeit ② und einer Verzögerungsphase ③. Das zugehörige Beschleunigungsprofil lautet daher

$$\ddot{s}(t) = \begin{cases} a & 0 \leq t < t_a \\ 0 & t_a \leq t < T - t_a \\ -a & T - t_a \leq t \leq T \\ 0 & \text{sonst} \end{cases} \quad (3.16)$$

und durch einmalige Integration folgt das Geschwindigkeitsprofil zu

$$\dot{s}(t) = \begin{cases} at & 0 \leq t < t_a \\ v & t_a \leq t < T - t_a \\ a(T - t) & T - t_a \leq t \leq T \\ 0 & \text{sonst} \end{cases}, \quad (3.17)$$

wobei in (3.16) und (3.17) die Geschwindigkeit  $v = at_a$  und die Beschleunigungs- und Verzögerungszeit  $t_a$  eingeführt wurde. Nochmalige Integration liefert die Zeitparametrierung  $s(t)$  in der Form

$$s(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2}at^2 & 0 \leq t < t_a \\ vt - \frac{v^2}{2a} & t_a \leq t < T - t_a \\ vT - \frac{v^2}{a} - \frac{a}{2}(T - t)^2 & T - t_a \leq t \leq T \\ vT - \frac{v^2}{a} & \text{sonst} \end{cases}. \quad (3.18)$$

Das trapezförmige Bewegungsprofil bietet den Vorteil, dass die Geschwindigkeits- und Beschleunigungsbeschränkungen des Roboters optimal ausgenutzt werden können. Aus

(3.16) und (3.17) eingesetzt in (3.9) 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

$$|v(x_1 - x_0)| \leq \dot{x}_{max} \quad (3.19)$$

$$|a(x_1 - x_0)| \leq \ddot{x}_{max} \quad (3.20)$$

bestimmen.

Die geradlinige Trajektorie (3.9) mit dem trapezförmigen Bewegungsprofil (3.16)–(3.18) 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. Die Beschleunigungs- und Verzögerungszeit ist mit  $t_a = v/a$  festgelegt. 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

$$T = \frac{1}{v} + \frac{v}{a} . \quad (3.21)$$

Weiters können 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 3 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 3.3) kann auch ein trapezförmiges Bewegungsprofil mit 7 Phasen konstruiert werden, welches diese Aufgabe erfüllt, siehe Abbildung 3.5. Anstatt sprunghaft die maximale Beschleunigung  $a$  aufzuschalten, wird die Beschleunigung linear mit beschränktem Ruck hoch- und heruntergefahren.

## 3.4 Trajektorien mit Via-Punkten

Die bisher vorgestellten Methoden erlauben es lediglich eine Punkt-zu-Punkt-Bewegung mit einem geradlinigen Pfad zu planen. Häufig ist aber die Aufgabe eines Roboters, eine Bahn mit mehreren Zwischenpunkten, den sogenannten *Via-Punkten*, abzufahren. Im einfachsten Fall können dazu eine Reihe von polynomialen Trajektorien verwendet werden, welche an den Via-Punkten mit Stetigkeitsbedingungen verbunden werden. Im Folgenden wird eine Methode für Polynome 3. Ordnung vorgestellt mit welcher eine Trajektorie  $x(t)$  direkt geplant wird, also ohne expliziter Zeitparametrierung  $s(t)$ .

Die gesuchte Trajektorie  $x(t)$  wird anhand der Positionen  $x(t_i) = x_i$  zu den Zeitpunkten  $t_i$  sowie den Geschwindigkeiten  $\dot{x}(t_i) = \dot{x}_i$  für alle Via-Punkte  $i = 0, \dots, N-1$  spezifiziert. Dabei sind  $t_0 = 0$  und  $t_{N-1} = T$  der Anfangs- und Endzeitpunkt dieser Trajektorie. Zwischen den  $N$  Via-Punkten liegen  $N-1$  Trajektoriensegmente mit den Laufzeiten

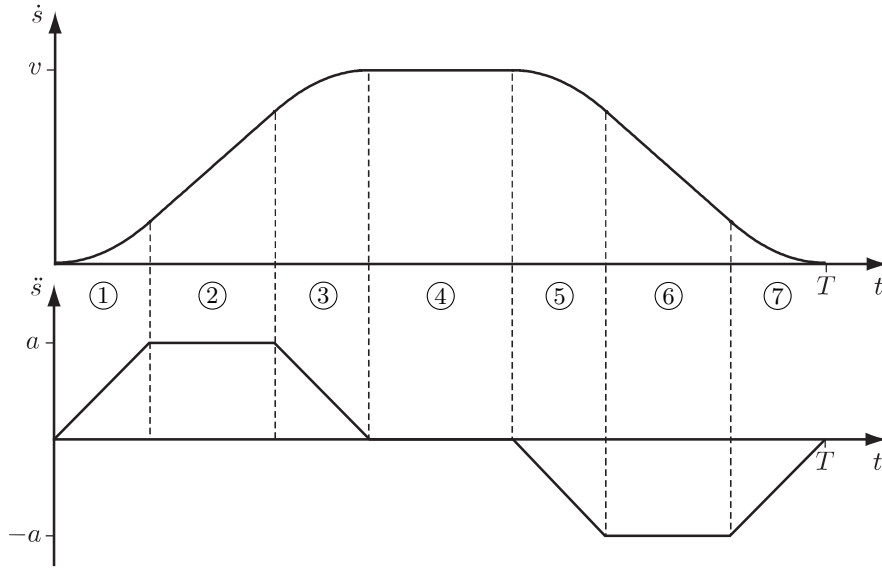


Abbildung 3.5: Zeitparametrierung für trapezförmiges Bewegungsprofil mit 7 Phasen:  
 ① Hochfahren der Beschleunigung, ② konstante Beschleunigung  $a$ , ③ herunterfahren der Beschleunigung, ④ konstante Geschwindigkeit  $v$ , ⑤ hochfahren der Verzögerung, ⑥ konstante Verzögerung  $-a$ , ⑦ herunterfahren der Verzögerung [3.1].

$T_i = t_{i+1} - t_i$ ,  $i = 0, \dots, N-2$ . Jedes Trajektoriensegment  $i$  wird als Polynom 3. Ordnung in der Form

$$x(t_i + \Delta t) = a_{i0} + a_{i1}\Delta t + a_{i2}\Delta t^2 + a_{i3}\Delta t^3 \quad (3.22)$$

angesetzt, wobei  $\Delta t$  die verstrichene Zeit innerhalb des Segments bezeichnet mit  $0 \leq \Delta t \leq T_i$ . Die Anfangs- und Endwerte jedes Trajektoriensegmentes werden nun anhand der vier Bedingungen gemäß

$$\begin{aligned} x(t_i) &= x_i & \dot{x}(t_i) &= \dot{x}_i \\ x(t_i + T_i) &= x_{i+1} & \dot{x}(t_i + T_i) &= \dot{x}_{i+1} \end{aligned}$$

vorgegeben, sodass die Übergänge zwischen zwei aufeinanderfolgenden Trajektoriensegmenten  $i$  und  $i+1$  einen stetigen Positions- und Geschwindigkeitsverlauf aufweisen. Aus diesen Bedingungen werden die Koeffizienten  $a_{i0}, \dots, a_{i3}$  der Polynome  $i = 0, \dots, N-1$  gelöst und lauten

$$a_{i0} = x_i \quad (3.23a)$$

$$a_{i1} = \dot{x}_i \quad (3.23b)$$

$$a_{i2} = \frac{3x_{i+1} - 3x_i - 2\dot{x}_i T_i - \dot{x}_{i+1} T_i}{T_i^2} \quad (3.23c)$$

$$a_{i3} = \frac{2x_i + (\dot{x}_i + \dot{x}_{i+1})T_i - 2x_{i+1}}{T_i^3} \quad (3.23d)$$

**Beispiel 3.2 (Trajektorie mit Via-Punkten).** In diesem Beispiel soll die Methode aus Abschnitt 3.4 anhand einer konkreten Trajektorie gezeigt werden. Die Trajektorie  $x(t)$  soll  $N = 4$  Via-Punkte mit den Zeitpunkten  $t_i$  passieren sowie die zugehörigen Positionen  $x_i$  und Geschwindigkeiten  $\dot{x}_i$ ,  $i = 0, \dots, 3$ , gemäß

$t_0 = 0 \text{ s}$	$x(t_0) = x_0 = 0$	$\dot{x}(t_0) = \dot{x}_0 = 0$
$t_1 = 1 \text{ s}$	$x(t_1) = x_1 = 0$	$\dot{x}(t_1) = \dot{x}_1 = 1$
$t_2 = 2 \text{ s}$	$x(t_2) = x_2 = 1$	$\dot{x}(t_2) = \dot{x}_2 = 0$
$t_3 = 3 \text{ s}$	$x(t_3) = x_3 = 1$	$\dot{x}(t_3) = \dot{x}_3 = 0$

aufweisen. Mit diesen Vorgaben werden die  $N - 1 = 3$  Trajektoriensegmente als Polynome 3. Ordnung in der Form (3.22) mit den Koeffizienten (3.23) und den vorgegebenen Positionen und Geschwindigkeiten berechnet. Es ergibt sich die zusammengesetzte Trajektorie mit Via-Punkten  $x(t)$  und die zugehörige Geschwindigkeit  $\dot{x}(t)$  welche in Abbildung 3.6 dargestellt ist.

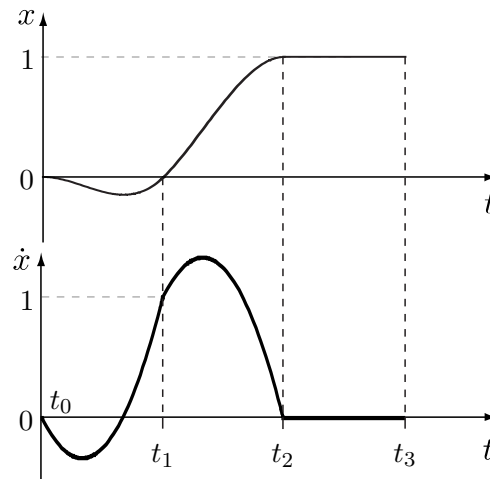


Abbildung 3.6: Zusammengesetzte Trajektorie mit Via-Punkten  $x(t)$  mit zugehöriger Geschwindigkeit  $\dot{x}(t)$  [3.1].

## 3.5 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>.



## 4 Roboterregelung

Ein Roboter kann eine Reihe von fundamental unterschiedlichen Aufgaben ausführen, wobei für jede Aufgabe ein geeignetes Regelungskonzept eingesetzt werden muss.

Zunächst kann ein Roboter vorgegebene Bewegungen – beschrieben im Konfigurations- oder im Arbeitsraum – ausführen, um so ein Objekt zu manipulieren oder ein Werkzeug entlang der Oberfläche eines Objektes zu führen. Dies wird durch die sogenannte *Bewegungsregelung* bewerkstelligt.

Ist der Endeffektor des Roboters im physischen Kontakt mit der Umgebung, so können mithilfe einer *Kraftregelung* gewünschte Kräfte und Drehmomente auf die Umgebung ausgeübt werden. Dies ist zum Beispiel notwendig, wenn zwei Objekte mit definierten Kräften zusammengefügt werden sollen oder wenn die Oberfläche eines Werkstücks mit vorgegebenen Kräften geschliffen oder poliert werden soll.

Darüber hinaus wird häufig bei kollaborativen Robotern die sogenannte *Impedanzregelung* eingesetzt. Eine mechanische Impedanz beschreibt den Zusammenhang zwischen der Positionsabweichung eines mechanischen Systems aus einer Ruhelage und den daraus folgenden Reaktionskräften. Ein Beispiel für ein mechanisches System ist das Feder-Masse-Dämpfer-System: Wird die Masse aus der Ruhelage heraus bewegt, so entstehen Reaktionskräfte, die der Auslenkung entgegenwirken. Mithilfe der Impedanzregelung kann sich ein Roboter im Konfigurations- oder im Arbeitsraum wie ein virtuelles mechanisches System verhalten, z. B. wie ein Feder-Masse-Dämpfer-System mit vorgegebener Masse, Dämpfung und Steifigkeit. Dies ist nützlich, wenn ein Roboter ein Objekt bearbeiten soll, dessen Form nicht exakt bekannt ist, oder wenn der Roboter als haptische Benutzerschnittstelle verwendet wird.

Schließlich können die obigen Regelungskonzepte auch kombiniert werden, indem unterschiedliche Regler für die jeweiligen Koordinaten des Arbeitsraums verwendet werden. So kann ein Werkzeug bei der Oberflächenbearbeitung durch den Roboter mithilfe einer Kraftregelung angeedrückt und mit einer Bewegungsregelung entlang der Oberfläche geführt werden.

Dieses Kapitel befasst sich mit der Bewegungsregelung von Robotersystemen für kinematisch nicht-redundante und redundante Roboter. In den beiden folgenden Abschnitten 4.1 und 4.2 werden zunächst geeignete Beschreibungen für das dynamische Robotermodell im Konfigurationsraum bzw. im Arbeitsraum eingeführt. Anschließend werden häufig verwendete Regler für die Bewegungsregelung und Impedanzregelung vorgestellt. Im Abschnitt 4.3 wird die Implementierung der vorgestellten Regler in einem Digitalrechner diskutiert.

## 4.1 Konfigurationsraum

### 4.1.1 Dynamisches Modell im Konfigurationsraum

Die Bewegungsgleichungen eines Roboters mit  $n = \dim(\mathbf{q})$  Freiheitsgraden lassen sich im Konfigurationsraum in der Form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (4.1)$$

anschreiben [4.1]. Es wird angenommen, dass der Roboter *vollaktuiert* ist, d.h. es gibt für jeden mechanischen Freiheitsgrad einen unabhängigen Stelleingang  $\boldsymbol{\tau}^T = [\tau_1, \tau_2, \dots, \tau_n]$ . Hier ist zu beachten, dass sich die generalisierten Kräfte  $\boldsymbol{\tau}$  im Allgemeinen noch in die dissipativen Kräfte und Momente  $\boldsymbol{\tau}_d$ , die Stelleingänge  $\boldsymbol{\tau}_c$  und die extern auf den Roboter wirkenden Kräfte und Momente  $\boldsymbol{\tau}_{ext}$  aufspalten lassen, d.h.

$$\boldsymbol{\tau} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext} . \quad (4.2)$$

Für die Massenmatrix  $\mathbf{M}(\mathbf{q})$ , die Matrix der Zentrifugal- und Coriolisterme  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  und den Vektor der Potentialkräfte  $\mathbf{g}(\mathbf{q})$  gelten folgende Eigenschaften und Zusammenhänge:

- $\mathbf{M}(\mathbf{q})$  ist eine positiv definite und damit invertierbare Matrix. D. h. für alle  $\dot{\mathbf{q}} \neq \mathbf{0}$  gilt  $\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} > 0$ , denn die kinetische Energie ist immer positiv.
- $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  ist eine schiefsymmetrische Matrix, d.h.  $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{N}^T(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ . Damit gilt  $\dot{\mathbf{q}}^T \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = 0$  für alle  $\dot{\mathbf{q}}$  sowie  $\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})$ .
- $\mathbf{g}^T(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} V(\mathbf{q})$  mit der potenziellen Energie  $V(\mathbf{q}) > 0$ .

### 4.1.2 Dezentrale Regler

Wie bereits in Abschnitt 1.3.1 eingeführt wurde, kann die Regelung im Konfiguration nun zentral oder dezentral erfolgen. Bei der *dezentralen Regelung* wird jedes Gelenk des Roboters getrennt voneinander geregelt und die mechanische Kopplung der Glieder bei der Roboterbewegung wird als Störung aufgefasst. Dadurch reduziert sich der Reglerentwurf auf den Eingrößenfall (SISO) mit der jeweiligen (verallgemeinerten) Gelenkskoordinate  $q_j$  als Ausgangsgröße und der generalisierten Kraft bzw. dem Moment  $\tau_j$ ,  $j = 1, \dots, n$  als Stelleingang.

Typischerweise liegt dem Entwurf ein einfaches Modell zweiter Ordnung zu Grunde. In der Praxis hat es sich bewährt, hochdynamische unterlagerte Geschwindigkeitsregler einzusetzen, welche etwa um den Faktor 5-10 schneller als der äußere Regelkreis sind. Diese unterlagerten Regler werden meist in Form von einfachen PID- oder PD-Reglern mit Reibungskompensation implementiert. Im Sinne einer klassischen Kaskadenregelung lassen sich dann die unterlagerten Geschwindigkeitsregelkreise idealisiert als Durchschaltung

$$\dot{q}_j = \dot{q}_{d,j} = u_j , \quad j = 1, \dots, n \quad (4.3)$$

mit den gewünschten Gelenksgeschwindigkeiten  $u_j = \dot{q}_{d,j}$  als neue Eingangsgrößen betrachten. Mit den idealisierten unterlagerten Geschwindigkeitsregelkreisen reduziert sich



das mathematische Modell des Roboters (4.1) zum sogenannten *kinematischen Robotermodell*

$$\dot{\mathbf{q}} = \mathbf{u} , \quad (4.4)$$

welches im Wesentlichen  $n$  unabhängige Integratoren darstellt. Um nun auf die gewünschte Trajektorie im Konfigurationsraum  $q_{d,j}(t)$  zu regeln, kann man das einfache PI-Regelgesetz

$$u_j = -k_1(q_j - q_{d,j}(t)) - k_0 \int (q_j - q_{d,j}(t)) dt \quad (4.5)$$

mit geeigneten Koeffizienten  $k_0 > 0$  und  $k_1 > 0$  verwenden. Liegt auch die Sollgeschwindigkeit  $\dot{q}_{d,j}(t)$  der Trajektorie vor, so kann das Regelgesetz noch weiter verbessert werden indem der Term  $\dot{q}_{d,j}(t)$  als Vorsteuerung zu (4.5) addiert wird.

#### 4.1.3 PD-Regler mit Gravitationskompensation

Ein sehr bekanntes und einfaches Verfahren zur Stabilisierung eines *Arbeitspunktes* im Konfigurationsraum (d. h.  $\mathbf{q} = \mathbf{q}_d$  und  $\dot{\mathbf{q}} = \dot{\mathbf{q}}_d = \mathbf{0}$ ) ist die *PD-Regelung mit Gravitationskompensation*

$$\boldsymbol{\tau} = \underbrace{\mathbf{g}(\mathbf{q})}_{\text{Gravitationskomp.}} - \underbrace{\mathbf{K}_d \dot{\mathbf{e}}_q}_{\text{D-Anteil}} - \underbrace{\mathbf{K}_p \mathbf{e}_q}_{\text{P-Anteil}} \quad (4.6)$$

mit dem Regelfehler  $\mathbf{e}_q = \mathbf{q} - \mathbf{q}_d$  und  $\dot{\mathbf{e}}_q = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d = \dot{\mathbf{q}}$  und den positiv definiten (symmetrischen) Reglermatrizen  $\mathbf{K}_d$  und  $\mathbf{K}_p$ . Aufgrund des Terms  $\mathbf{g}(\mathbf{q})$  in (4.6) muss das Regelgesetz von einer zentralen Recheneinheit ausgewertet werden, in welcher die Messwerte aller Gelenkwinkel  $\mathbf{q}$  vorliegen. Damit zählt dieses Verfahren zu den zentralen Reglern im Konfigurationsraum.

Setzt man (4.6) in (4.1) ein, so erhält man für den geschlossenen Kreis

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \dot{\mathbf{q}} - \mathbf{K}_p \mathbf{e}_q . \quad (4.7)$$

Man erkennt, dass durch das Regelgesetz (4.6) der Gravitationsterm  $\mathbf{g}(\mathbf{q})$  gekürzt und mit  $\boldsymbol{\tau}_p = \mathbf{K}_p \mathbf{e}_q$  ein weiterer Term hinzugefügt wird, der als Federkraft mit der Steifigkeitsmatrix  $\mathbf{K}_p$  interpretiert werden kann. Im Speziellen lassen sich die Einträge von  $\mathbf{K}_p$  als Federsteifigkeiten von fiktiven Federelementen im Konfigurationsraum interpretieren. Diese Elemente sind zwischen der Konfiguration des Roboters  $\mathbf{q}$  und der gewünschten Konfiguration des Roboters  $\mathbf{q}_d$  gespannt. Für die Kraft  $\boldsymbol{\tau}_p$  gilt

$$\boldsymbol{\tau}_p = \mathbf{K}_p \mathbf{e}_q = \left( \frac{\partial}{\partial \mathbf{q}} V_p \right)^T \quad \text{mit} \quad V_p = \frac{1}{2} \mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q , \quad (4.8)$$

d. h.  $\boldsymbol{\tau}_p$  stellt eine Potenzialkraft dar, wie sie in der Vorlesung *Modellbildung* eingeführt wurde, siehe [4.1].

Aus (4.7) ist nicht unmittelbar einsichtig, dass die Ruhelage  $\mathbf{q} = \mathbf{q}_d$  und  $\dot{\mathbf{q}} = \mathbf{0}$  des geschlossenen Kreises auch tatsächlich asymptotisch stabil ist. Um dies zu zeigen, wird in einem ersten Schritt für den geschlossenen Kreis (4.7) eine Energiefunktion der Form

$$E_c = \underbrace{\frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}}_{\text{kinetische Energie}} + \underbrace{\frac{1}{2}\mathbf{e}_q^T \mathbf{K}_p \mathbf{e}_q}_{\text{potentielle Energie der Feder}} > 0 \quad (4.9)$$

mit der potentiellen Energie der Feder  $V_p$  aus (4.8) angesetzt. Bildet man die zeitliche Ableitung von (4.9), so folgt

$$\begin{aligned} \dot{E}_c &= \frac{1}{2}\ddot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{e}}_q^T \mathbf{K}_p \mathbf{e}_q + \frac{1}{2}\mathbf{e}_q^T \mathbf{K}_p \dot{\mathbf{e}}_q \\ &= \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{e}}_q^T \mathbf{K}_p \mathbf{e}_q \\ &= \dot{\mathbf{q}}^T \underbrace{(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{K}_d \dot{\mathbf{q}} - \mathbf{K}_p \mathbf{e}_q)}_{\text{gemäß (4.7)}} + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{K}_p \mathbf{e}_q \\ &= -\dot{\mathbf{q}}^T \mathbf{K}_d \dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \underbrace{(\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}))}_{\substack{\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{N}^T(\mathbf{q}, \dot{\mathbf{q}}) \\ = 0}} \dot{\mathbf{q}} \leq 0. \end{aligned} \quad (4.10)$$

Man erkennt, dass wegen der positiven Definitheit von  $\mathbf{K}_d$  die Energie  $E_c$  von (4.9) im geschlossenen Kreis nur kleiner werden kann und daraus lässt sich über die *Lyapunovtheorie* die asymptotische Stabilität des Arbeitspunktes  $\mathbf{q} = \mathbf{q}_d$  und  $\dot{\mathbf{q}} = \mathbf{0}$  beweisen.

**Bemerkung 4.1.** 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 (4.6) eingesetzt wird. Im Weiteren gibt es auch adaptive Regelungsverfahren, die es erlauben, gewisse Parameter des Roboters (z. B. die Lastmasse) in Echtzeit mitzuschätzen.

#### 4.1.4 Computed-Torque-Regler

Der PD-Regler mit Gravitationskompensation in Abschnitt 4.1.3 kann nur für die Stabilisierung von Arbeitspunkten  $\mathbf{q}_d$  oder für langsam variierende Sollkonfigurationen  $\mathbf{q}_d(t)$  eingesetzt werden. Eine Bewegungsregelung im eigentlichen Sinn ist hingegen eine Trajektorienfolgeregelung, beispielsweise mit dem *Computed-Torque-Regler*. Mit diesem Verfahren folgt der Roboter im Konfigurationsraum einer vorgegebenen Solltrajektorie  $\mathbf{q}_d(t)$ . Dabei wird angenommen, dass die Solltrajektorie  $\mathbf{q}_d(t)$  zumindest zweifach stetig differenzierbar ist, d.h.  $\dot{\mathbf{q}}_d(t)$  und  $\ddot{\mathbf{q}}_d(t)$  sind zumindest stetig. Der Computed-Torque-Regler lautet

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (4.11)$$

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 Reglermatritzen  $\mathbf{K}_{1q}$  und  $\mathbf{K}_{2q}$ . Ein Vergleich zwischen (4.1) und (4.11) zeigt die strukturelle Ähnlichkeit des Computed-Torque-Reglers zum Robotermodell im Konfigurationsraum. Tatsächlich werden durch den Computed-Torque-Regler die nichtlinearen Terme des Robotermodells  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  und  $\mathbf{g}(\mathbf{q})$  kompensiert bzw. gekürzt und es bleibt ein lineares Systemverhalten übrig. Deshalb wird dieses Verfahren auch *inverse*

*Dynamik* bezeichnet. Dazu müssen allerdings alle statischen und dynamischen Parameter (Abmessungen, Schwerpunkte, Massen, Massenträgheitsmomente) des Roboters sehr genau bekannt sein, was in der Praxis eine Herausforderung darstellt. Weiters handelt es sich beim Computed-Torque-Regler ebenfalls um ein zentrales Regelungskonzept im Konfigurationsraum.

Setzt man (4.11) in (4.1) ein, so ergibt sich

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (4.12)$$

bzw.

$$\mathbf{M}(\mathbf{q})\underbrace{(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d(t))}_{\ddot{\mathbf{e}}_q} + \mathbf{K}_{2q}\dot{\mathbf{e}}_q + \mathbf{K}_{1q}\mathbf{e}_q = \mathbf{0} . \quad (4.13)$$

Da  $\mathbf{M}(\mathbf{q})$  für alle  $\mathbf{q}$  positiv definit und damit invertierbar ist, muss für die Fehlerdynamik

$$\ddot{\mathbf{e}}_q + \mathbf{K}_{2q}\dot{\mathbf{e}}_q + \mathbf{K}_{1q}\mathbf{e}_q = \mathbf{0} \quad (4.14)$$

gelten. Wählt man für  $\mathbf{K}_{1q}$  und  $\mathbf{K}_{2q}$  Diagonalmatrizen in der Form  $\mathbf{K}_{1q} = \text{diag}(k_{1q,1}, \dots, k_{1q,n})$  und  $\mathbf{K}_{2q} = \text{diag}(k_{2q,1}, \dots, k_{2q,n})$  mit positiven Einträgen  $k_{1q,j}, k_{2q,j} > 0, j = 1, \dots, n$ , dann sind die Fehlerdynamiken der einzelnen Gelenkskoordinaten  $q_j$  mit

$$\ddot{e}_{q,j} + k_{2q,j}\dot{e}_{q,j} + k_{1q,j}e_{q,j} = 0, \quad j = 1, \dots, n \quad (4.15)$$

entkoppelt. Schreibt man (4.15) als System von Differenzialgleichungen erster Ordnung (Zustandsdarstellung) in der Form

$$\frac{d}{dt} \begin{bmatrix} e_{q,j} \\ \dot{e}_{q,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_{1q,j} & -k_{2q,j} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} e_{q,j} \\ \dot{e}_{q,j} \end{bmatrix}, \quad (4.16)$$

so erkennt man, dass die Matrix  $\mathbf{A}$  in erster Standardform vorliegt und das Polynom  $p_j(s) = s^2 + k_{2q,j}s + k_{1q,j}$  das charakteristische Polynom von  $\mathbf{A}$  darstellt. Für  $k_{2q,j} > 0$  und  $k_{1q,j} > 0$  ist  $p_j(s)$  ein Hurwitzpolynom und mit den Koeffizienten  $k_{2q,j}$  und  $k_{1q,j}$  lassen sich die *Eigenwerte der Trajektorienfehlerdynamik* wie gewünscht vorgeben (Polvorgabe).

**Bemerkung 4.2 (Computed-Torque-Regler mit Integralanteil).** Man könnte in (4.11) noch einen Integralanteil im Regler hinzufügen, womit das erweiterte Regelgesetz

$$\boldsymbol{\tau} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{M}(\mathbf{q})\left(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q - \mathbf{K}_{0q} \int \mathbf{e}_q dt\right). \quad (4.17)$$

lautet. Wählt man auch  $\mathbf{K}_{0q} = \text{diag}(k_{0q,1}, \dots, k_{0q,n})$  als Diagonalmatrix mit positiven Einträgen  $k_{0q,j}, j = 1, \dots, n$ , dann müssen die Koeffizienten  $k_{0q,j}, k_{1q,j}$  und  $k_{2q,j}$  so festgelegt werden, dass die entkoppelten Fehlerdynamiken gewünschte Eigenwerte

aufweisen (Polvorgabe) und die charakteristischen Polynome

$$p_j(s) = s^3 + k_{2q,j}s^2 + k_{1q,j}s + k_{0q,j} \quad (4.18)$$

Hurwitzpolynome sind.

**Bemerkung 4.3 (Kompensation von dissipativen und externen Kräften und Momenten).**

Man beachte, dass sich  $\tau$  gemäß (4.2) im Allgemeinen aus dissipativen Kräften und Momenten  $\tau_d$ , Stelleingängen  $\tau_c$  und extern auf den Roboter wirkenden Kräften und Momenten  $\tau_{ext}$  zusammensetzt. Sind Schätzungen  $\hat{\tau}_d$  bzw.  $\hat{\tau}_{ext}$  von  $\tau_d$  bzw.  $\tau_{ext}$  oder zumindest von Teilen bekannt, dann können diese im Regelgesetz berücksichtigt werden und  $\tau_c$  lautet dann

$$\tau_c = \tau - \hat{\tau}_d - \hat{\tau}_{ext} \quad (4.19)$$

mit  $\tau$  gemäß (4.6), (4.11) oder (4.17).

## 4.2 Arbeitsraum

### 4.2.1 Dynamisches Modell im Arbeitsraum

Das dynamische Modell des Roboters (4.1) wurde als Funktion der Koordinaten im Konfigurationsraum  $\mathbf{q}$  und  $\dot{\mathbf{q}}$  angeschrieben. Eine Bewegungsregelung im Arbeitsraum findet allerdings in den zugehörigen Koordinaten des Arbeitsraums  $\mathbf{x}_e$  gemäß (1.3) und  $\dot{\mathbf{x}}_e$  gemäß (2.43) statt. Es ist daher vorteilhaft, das dynamische Modell des Roboters (4.1) in den Koordinaten des Arbeitsraums anzuschreiben.

Der Zusammenhang zwischen den Koordinaten im Konfigurationsraum und den Koordinaten im Arbeitsraum sowie deren Ableitungen sind durch (1.3) und (2.43) gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \quad (4.20a)$$

$$\dot{\mathbf{x}}_e = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}} \quad (4.20b)$$

$$\ddot{\mathbf{x}}_e = \mathbf{J}_A(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \quad (4.20c)$$

gegeben. Aus (4.20c) folgt unter der Annahme der Regularität von  $\mathbf{J}_A(\mathbf{q})$ , d. h. außerhalb der Singularitäten, die Beziehung

$$\ddot{\mathbf{q}} = \mathbf{J}_A^{-1}(\mathbf{q})\left(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\right). \quad (4.21)$$

Setzt man (4.21) in (4.1) ein, so erhält man

$$\mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q})\ddot{\mathbf{x}}_e + \left(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q})\dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\right)\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau \quad (4.22)$$

und durch Multiplikation der Gleichung mit  $\mathbf{J}_A^T(\mathbf{q}) = \left(\mathbf{J}_A^{-1}\right)^T(\mathbf{q}) = \left(\mathbf{J}_A^T\right)^{-1}(\mathbf{q})$  von links ergibt sich das *mathematische Modell des Roboters im Arbeitsraum* zu

$$\Lambda(\mathbf{x}_e)\ddot{\mathbf{x}}_e + \boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)\dot{\mathbf{x}}_e + \mathbf{F}_g(\mathbf{x}_e) = \mathbf{F} \quad (4.23)$$

mit

$$\Lambda(\mathbf{x}_e) = \mathbf{J}_A^{-T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q}) \quad (4.24a)$$

$$\boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e) = \mathbf{J}_A^{-T}(\mathbf{q})\left(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q})\dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\right)\mathbf{J}_A^{-1}(\mathbf{q}) \quad (4.24b)$$

$$\mathbf{F}_g(\mathbf{x}_e) = \mathbf{J}_A^{-T}(\mathbf{q})\mathbf{g}(\mathbf{q}) \quad (4.24c)$$

und der generalisierten Kraft  $\mathbf{F}$  im Arbeitsraum

$$\mathbf{F} = \mathbf{J}_A^{-T}(\mathbf{q})\boldsymbol{\tau} , \quad (4.25)$$

welche am Endeffektor des Roboters wirkt. Die Beziehung zwischen den Kräften und Momenten im Konfigurationsraum und jenen im Arbeitsraum über die Manipulator Jacobi-Matrix wurde in der Vorlesung *Modellbildung* hergeleitet [4.1]. Man beachte, dass in (4.24) der Zusammenhang  $\dot{\mathbf{q}} = \mathbf{J}_A^{-1}(\mathbf{q})\dot{\mathbf{x}}_e$  gilt und  $\mathbf{q}$  noch *formal* durch  $\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_e)$  ersetzt werden muss, siehe (4.20). Handelt es sich jedoch um einen kinematisch redundanten Roboter, d.h.  $m < n$ , so können die Vektoren  $\mathbf{x}_e, \dot{\mathbf{x}}_e \in \mathbb{R}^m$  nicht den gesamten Zustand  $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$  des Roboters darstellen. In diesem Fall müssen die Vektoren und Matrizen in (4.24) als Funktion von  $\mathbf{q}$  und  $\dot{\mathbf{q}}$  anstatt von  $\mathbf{x}_e$  und  $\dot{\mathbf{x}}_e$  ausgedrückt werden.

Die Multiplikation von (4.22) mit  $\mathbf{J}_A^{-T}(\mathbf{q})$  von links wurde ausgeführt, um beim Robotermodell im Arbeitsraum (4.23) eine ähnliche Struktur zu erhalten wie im Konfigurationsraum (4.1). In (4.23) ist  $\Lambda(\mathbf{x}_e)$  die Massenmatrix,  $\boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)$  ist die Coriolismatrix und  $\mathbf{F}_g(\mathbf{x}_e)$  ist der Vektor der Gravitationskräfte im Arbeitsraum. Es kann gezeigt werden, dass folgende Eigenschaften auch für das Robotermodell (4.23) erhalten bleiben:

- $\Lambda(\mathbf{x}_e)$  ist eine positiv definite und damit invertierbare Matrix.
- $\boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)$  ist eine schiefsymmetrische Matrix, d.h.  $\boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e) + \boldsymbol{\mu}^T(\mathbf{x}_e, \dot{\mathbf{x}}_e) = \mathbf{0}$ .

#### 4.2.2 Computed-Torque-Regler

Der *Computed-Torque-Regler im Arbeitsraum* basiert direkt auf dem mathematischen Modell des Roboters in den Koordinaten des Arbeitsraums (4.23) und lautet

$$\mathbf{F} = \Lambda(\mathbf{x}_e)(\ddot{\mathbf{x}}_{e,d}(t) - \mathbf{K}_{2x}\dot{\mathbf{e}}_x - \mathbf{K}_{1x}\mathbf{e}_x) + \boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)\dot{\mathbf{x}}_e + \mathbf{F}_g(\mathbf{x}_e) \quad (4.26)$$

mit dem Trajektorienfehler im Arbeitsraum  $\mathbf{e}_x = \mathbf{x}_e - \mathbf{x}_{e,d}(t)$  bzw.  $\dot{\mathbf{e}}_x = \dot{\mathbf{x}}_e - \dot{\mathbf{x}}_{e,d}(t)$  und den Reglermatrizen  $\mathbf{K}_{1x}$  und  $\mathbf{K}_{2x}$ . Die Fehlerdynamik des geschlossenen Regelkreises folgt durch Einsetzen des Reglers (4.26) in die Gleichung des Robotermodells (4.23) zu

$$\underbrace{\ddot{\mathbf{x}}_e - \ddot{\mathbf{x}}_{e,d}}_{\ddot{\mathbf{e}}_x} + \mathbf{K}_{2x}\dot{\mathbf{e}}_x + \mathbf{K}_{1x}\mathbf{e}_x = \mathbf{0} . \quad (4.27)$$

Werden weiters die Reglermatrizen als Diagonalmatrizen in der Form  $\mathbf{K}_{1x} = \text{diag}(k_{1x,1}, \dots, k_{1x,m})$  und  $\mathbf{K}_{2x} = \text{diag}(k_{2x,1}, \dots, k_{2x,m})$  mit positiven Einträgen  $k_{1x,j}, k_{2x,j}, j = 1, \dots, m$ , gewählt, so sind die  $m$  Fehlerdynamiken des  $m$ -dimensionalen Arbeitsraums

$$\ddot{e}_{x,j} + k_{2x,j}\dot{e}_{x,j} + k_{1x,j}e_{x,j} = 0 , \quad j = 1, \dots, m \quad (4.28)$$

entkoppelt. Dies bedeutet, dass die Fehlerdynamik in jeder Koordinate individuell vorgegeben werden kann und dass die Regelfehler unabhängig voneinander mit der jeweiligen Dynamik asymptotisch abklingen.

Während (4.23) und (4.26) das Regelungskonzept mathematisch klar darstellen, ist die Formulierung (4.26) des Computed-Torque-Reglers im Arbeitsraum jedoch ungeeignet für eine Implementierung in der Praxis. Die Gründe dafür sind, dass in den Matrizen (4.24) mehrmals die Inverse der analytischen Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  auftritt und die Gleichungen (4.20) nicht eindeutig umgekehrt werden können, siehe dazu Abschnitt 2.2. Weiters liegen meist nur die Gelenkwinkel  $\mathbf{q}$  als Messung am Roboter vor, nicht aber die Pose des Endeffektors im Arbeitsraum  $\mathbf{x}_e$ . Daher kann die Messung der Gelenkwinkel  $\mathbf{q}$  direkt im Regelgesetz verwendet werden. Setzt man nun (4.24) und (4.25) in (4.26) ein, so erhält man den Computed-Torque-Regler im Arbeitsraum *ausgedrückt im Konfigurationsraum* in der Form

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{J}_A^T(\mathbf{q})\mathbf{F} \\ &= \mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q})\left(\ddot{\mathbf{x}}_{e,d}(t) - \mathbf{K}_{2x}\dot{\mathbf{e}}_x - \mathbf{K}_{1x}\mathbf{e}_x - \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\right) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})\end{aligned}\quad (4.29)$$

mit

$$\mathbf{e}_x = \mathbf{f}(\mathbf{q}) - \mathbf{x}_{e,d}(t), \quad \dot{\mathbf{e}}_x = \mathbf{J}_A(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{x}}_{e,d}(t). \quad (4.30)$$

Man beachte, dass in der Formulierung (4.29) und (4.30) lediglich die Vorwärtskinematik  $\mathbf{f}(\mathbf{q})$  und die differentielle Kinematik mit  $\mathbf{J}_A(\mathbf{q})$  des Roboters benötigt werden. Auch diese Formulierung führt zur selben Fehlerdynamik (4.27).

### 4.2.3 Impedanzregelung

Bei der Impedanzregelung im Arbeitsraum wird nicht nur eine gewünschte Roboterbewegung  $\mathbf{x}_{e,d}(t)$  vorgegeben, sondern ein gewünschtes mechanisches Verhalten bezüglich der Koordinaten im Arbeitsraum. Ein Beispiel für ein mechanisches Verhalten ist das *Feder-Masse-Dämpfer-System*

$$\mathbf{M}_d\ddot{\mathbf{e}}_x + \mathbf{D}_d\dot{\mathbf{e}}_x + \mathbf{K}_d\mathbf{e}_x = \mathbf{F}_{ext} \quad (4.31)$$

mit  $\mathbf{e}_x = \mathbf{x}_e - \mathbf{x}_{e,d}$  und den vorgegebenen positiv definiten Matrizen  $\mathbf{M}_d$ ,  $\mathbf{D}_d$  und  $\mathbf{K}_d$  für die Masse, Dämpfung bzw. Steifigkeit des Systems [4.2]. Die externe Kraft  $\mathbf{F}_{ext}$  in (4.31) wirkt auf das System und lenkt die virtuelle Masse von der (zeitveränderlichen) virtuellen Ruhelage  $\mathbf{x}_{e,d}$  aus.

Das Regelgesetz für die Impedanzregelung im Arbeitsraum lautet

$$\mathbf{F} = \Lambda(\mathbf{x}_e)\left(\ddot{\mathbf{x}}_{e,d}(t) + \mathbf{M}_d^{-1}(\mathbf{F}_{ext} - \mathbf{D}_d\dot{\mathbf{e}}_x - \mathbf{K}_d\mathbf{e}_x)\right) + \boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)\dot{\mathbf{x}}_e + \mathbf{F}_g(\mathbf{x}_e) \quad (4.32)$$

und ausgedrückt im Konfigurationsraum ist dies

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{M}(\mathbf{q})\mathbf{J}_A^{-1}(\mathbf{q})\left(\ddot{\mathbf{x}}_{e,d}(t) + \mathbf{M}_d^{-1}(\mathbf{F}_{ext} - \mathbf{D}_d\dot{\mathbf{e}}_x - \mathbf{K}_d\mathbf{e}_x) - \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\right) \\ &\quad + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}).\end{aligned}\quad (4.33)$$

Beide Formulierungen führen zum gewünschten Impedanzverhalten und weisen (4.31) als Fehlerdynamik des geschlossenen Regelkreises auf. Dies bedeutet, dass eine externe

Kraft  $\mathbf{F}_{ext}$ , welche am Endeffektor des Roboters angreift, eine Auslenkung  $\mathbf{e}_x$  aus der virtuellen Ruhelage  $\mathbf{x}_{e,d}$  gemäß der vorgegebenen Dynamik (4.31) hervorruft. Für die Impedanzregelung ist allerdings die Messung (oder Schätzung) der externen Kräfte und Drehmomente  $\mathbf{F}_{ext}$  in allen  $m$  Dimensionen des Arbeitsraums notwendig.

#### 4.2.4 Kinematisch redundante Roboter $n > m$

In diesem Abschnitt werden die vorgestellten Regelungskonzepte im Arbeitsraum auf kinematisch redundante Roboter erweitert, d. h. es gilt  $n > m$ . Damit ist die analytische Manipulator Jacobi-Matrix  $\mathbf{J}_A(\mathbf{q})$  nicht mehr quadratisch und die Inverse ist nicht mehr definiert. Analog zu den Abschnitten 2.4.2 und 2.4.3 wird daher die Pseudoinverse  $\mathbf{J}_A^\dagger(\mathbf{q})$  aus (2.92) anstelle der Inversen  $\mathbf{J}_A^{-1}(\mathbf{q})$  verwendet. Weiters wird ein zusätzlicher Stelleingang  $\tau_n$  eingeführt, über welchen die Bewegung des Roboters im Nullraum geregelt wird.

Mit diesen Erweiterungen folgt der Computed-Torque-Regler für kinematisch redundante Roboter (ausgedrückt im Konfigurationsraum) zu, siehe (4.29)

$$\begin{aligned} \boldsymbol{\tau} = & \mathbf{M}(\mathbf{q})\mathbf{J}_A^\dagger(\mathbf{q})\left(\ddot{\mathbf{x}}_{e,d}(t) - \mathbf{K}_{2x}\dot{\mathbf{e}}_x - \mathbf{K}_{1x}\mathbf{e}_x - \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\right) \\ & + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_n \end{aligned} \quad (4.34)$$

und der Impedanzregler lautet, siehe (4.33)

$$\begin{aligned} \boldsymbol{\tau} = & \mathbf{M}(\mathbf{q})\mathbf{J}_A^\dagger(\mathbf{q})\left(\ddot{\mathbf{x}}_{e,d}(t) + \mathbf{M}_d^{-1}(\mathbf{F}_{ext} - \mathbf{D}_d\dot{\mathbf{e}}_x - \mathbf{K}_d\mathbf{e}_x) - \dot{\mathbf{J}}_A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\right) \\ & + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_n . \end{aligned} \quad (4.35)$$

Die Beschleunigungen des Nullraumreglers  $\ddot{\mathbf{q}}_n$  gemäß (2.103) werden über eine Projektionsmatrix in der Form

$$\boldsymbol{\tau}_n = \left(\mathbf{E} - \mathbf{J}_A^\dagger(\mathbf{q})\mathbf{J}_A(\mathbf{q})\right)\ddot{\mathbf{q}}_n \quad (4.36)$$

aufgeschaltet. Wie in den Abschnitten 2.4.2 und 2.4.3 kann mit dem Nullraumregler ein (anwendungsspezifisch gewähltes) sekundäres Kriterium erfüllt werden.

### 4.3 Implementierung

In diesem Skriptum wurden eine Reihe von mathematischen Zusammenhängen zur Roboterkinematik, -dynamik, Trajektorienplanung und Regelung vorgestellt. Damit ein geschlossener Regelkreis für einen Roboter implementiert werden kann, sind noch weitere Schritte notwendig, die in diesem Abschnitt erläutert werden sollen.

Einen Überblick über eine solche Implementierung gibt Abbildung 4.1. Der eigentliche Roboter ist auf der rechten Seite dargestellt und besteht aus den Aktoren, dem mechanischen Aufbau und den Sensoren. Die Ansteuerung der Aktoren erfolgt über D/A-Wandler und die Sensoren werden mithilfe von A/D-Wandlern ausgelesen. In praktisch allen heutigen Automatisierungssystemen erfolgt die Kommunikation mit der zentralen Steuereinheit (Digitalrechner) über ein Bussystem, über welches die digitalen Eingangs- und Ausgangsdaten (digitale I/Os) ausgetauscht werden. Die digitalen I/Os stellen die Schnittstelle zu den Aktoren und Sensoren dar. Die zugehörigen Signale haben noch keine



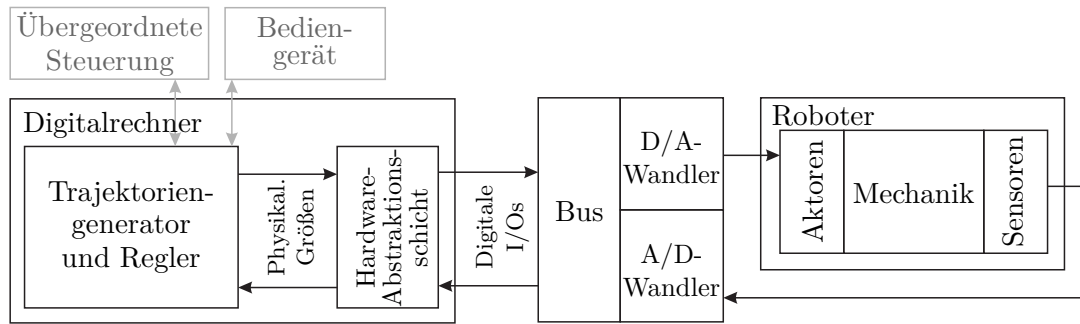


Abbildung 4.1: Blockschaltbild eines digitalen Regelkreises für einen Roboter.

physikalische Dimension: Beispielsweise liefert ein Inkrementalencoder  $2^k$  Inkremente pro Umdrehung des Motors, wobei  $k$  der Auflösung des Encoders in bit entspricht. Weiters ist in einem Industrieroboter in jedem Gelenk ein Getriebe mit einem Übersetzungsverhältnis  $u \gg 1$  eingebaut. Die digitale Position eines Gelenks  $\hat{q}$  wird daher als eine Ganzzahl im Bereich  $0 \leq \hat{q} < 2^k u$  dargestellt, während der Regler mit einer physikalischen Position  $0 \leq q < 2\pi$  arbeitet. Mithilfe der Hardware-Abstraktionsschicht werden die digitalen Sensorsignale in physikalische Größen umgerechnet, z.B. für den Inkrementalencoder mit Getriebe im Antriebsstrang gilt

$$q = 2\pi \frac{\hat{q}}{2^k u} . \quad (4.37)$$

Gleichermaßen werden auch die Stellgrößen für die Motoren zwischen physikalischen Größen und digitale Ausgangsdaten umgerechnet. Eine weitere Aufgabe für die Hardware-Abstraktionsschicht liegt beim Prüfen und Validieren der Sensorgrößen und Stellgrößen. Wird vom Regler eine ungültige oder zu große Stellgröße angefordert, so können Sicherheitsfunktionen ausgeführt und der Roboter zu einem sicheren Halt gebracht werden. Der Regler und der Trajektoriengenerator müssen in Echtzeit mit einer vorgegebenen Abtastzeit  $T_a$  am Digitalrechner laufen. In jedem Zyklus wird die laufende Trajektorie ausgewertet und durch den Regler wird die neue physikalische Stellgröße unter Verwendung der physikalischen Sensorgrößen und des Robotermodells berechnet. Eine übergeordnete Steuerung bzw. ein Bediengerät (engl. *Teach-Pendant*) parametriert schließlich den Trajektoriengenerator und den Regler.

Der Block „Trajektoriengenerator und Regler“ ist in Abbildung 4.2 im Detail dargestellt. Die physikalischen Sensorgrößen werden zunächst mithilfe geeigneter Sensorfilter aufbereitet und alle notwendigen Zustandsgrößen davon abgeleitet. Die Gelenksgeschwindigkeit  $\dot{\mathbf{q}}$  kann nicht direkt gemessen werden, sondern muss über einen Differenzierfilter aus der Gelenkposition  $\mathbf{q}$  berechnet werden. Mithilfe der gemessenen und berechneten Zustandsgrößen des Roboters wird das Robotermodell ausgewertet, d.h. die Matrizen  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  sowie der Gravitationsvektor  $\mathbf{g}(\mathbf{q})$ . Die Trajektoriengeneratoren sowie die Reglerauswahl werden über Befehle von einer übergeordneten Steuerung bzw. einem Bediengerät gesteuert. Der gewählte Regler wird für die laufende Trajektorie ausgewertet und zusammen mit der Reibungskompensation wird die Stellgröße berechnet.



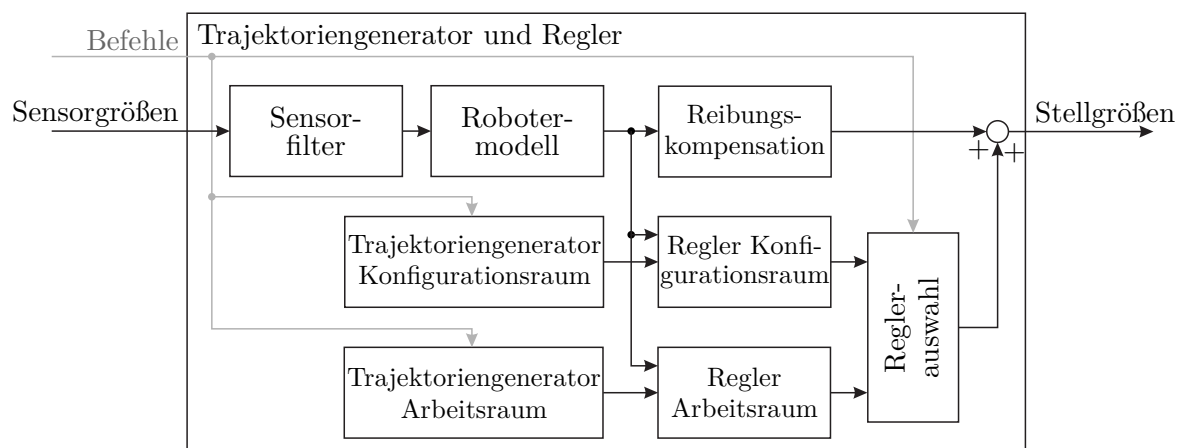


Abbildung 4.2: Blockschaltbild der Implementierung eines Trajektoriengenerators und Reglers für einen Industrieroboter.

## 4.4 Literatur

- [4.1] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2020)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2020. Adresse: <https://www.acin.tuwien.ac.at/bachelor/modellbildung/>.
- [4.2] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Berlin Heidelberg: Springer, 2008.
- [4.3] 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>.
- [4.4] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.