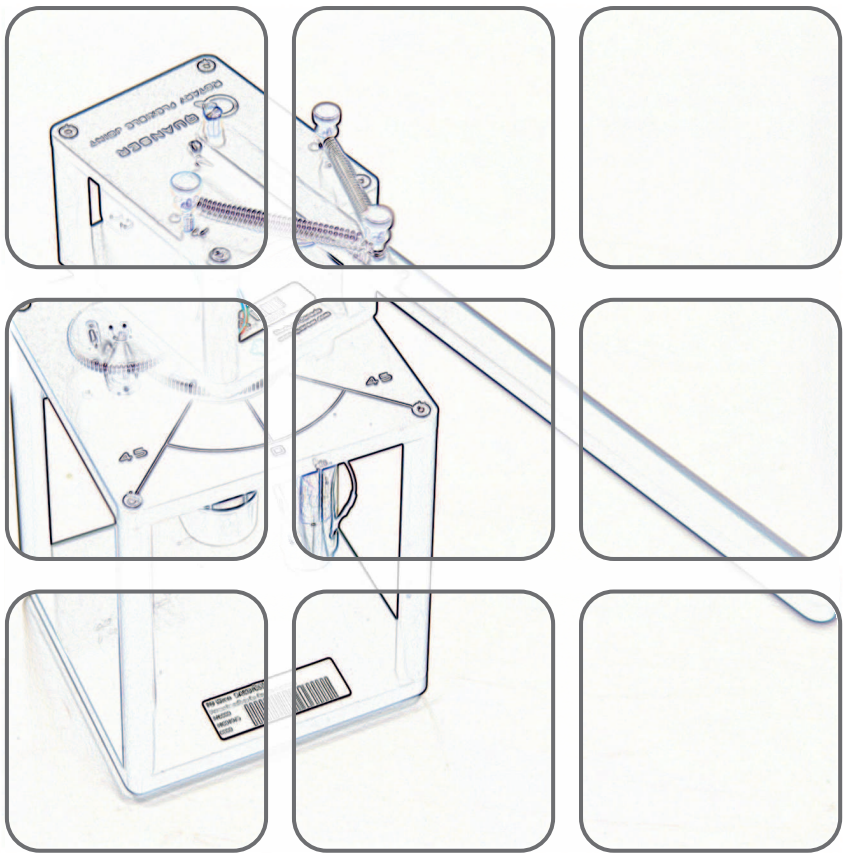
TECHNISCHE UNIVERSITÄT WIEN

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Fachvertiefung
2022W

AUTOMATISIERUNGS- & REGELUNGSTECHNIK

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



Automatisierungs- und Regelungstechnik

Fachvertiefung
2022W

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

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 (d. h. die vertikale Position) und zwei rotatorische Freiheitsgrade (d. h. die Rotation um die beiden Achsen der Tischebene) durch den Tisch vorgegeben. Es verbleiben daher zwei translatorische Freiheitsgrade, das ist die zweidimensionale Position der Münze am Tisch, und ein rotatorischer Freiheitsgrad, das ist die Rotation um den Oberflächennormalenvektor der Tischplatte.

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

$$\begin{aligned} n = &\text{ (Summe der Freiheitsgrade der Starrkörper) } - \\ &\text{ (Anzahl der unabhängigen Zwangsbedingungen) } . \end{aligned} \tag{1.1}$$

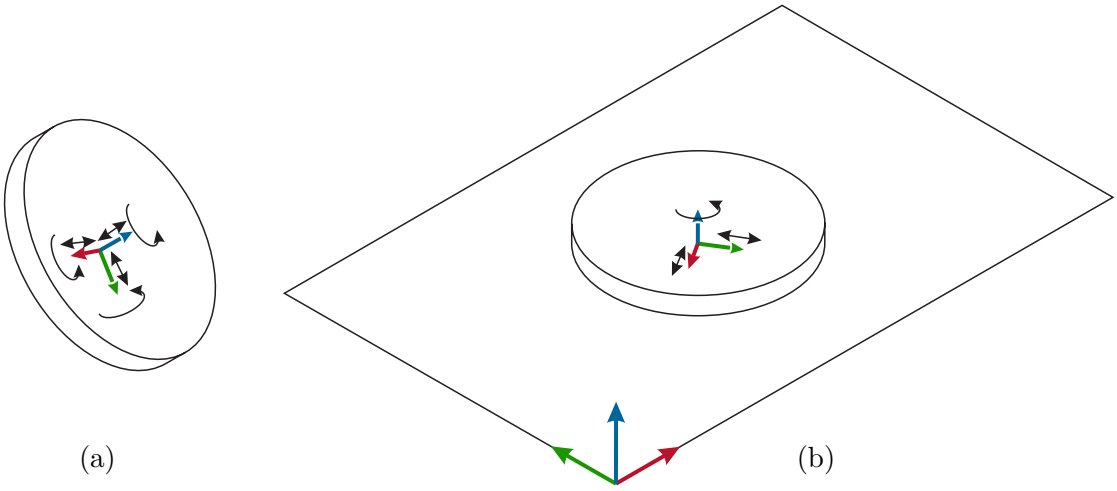


Abbildung 1.1: Freiheitsgrade einer Münze. (a) Münze ohne Zwangsbedingungen mit 6 Freiheitsgraden, (b) Münze auf einem Tisch mit 3 Freiheitsgraden

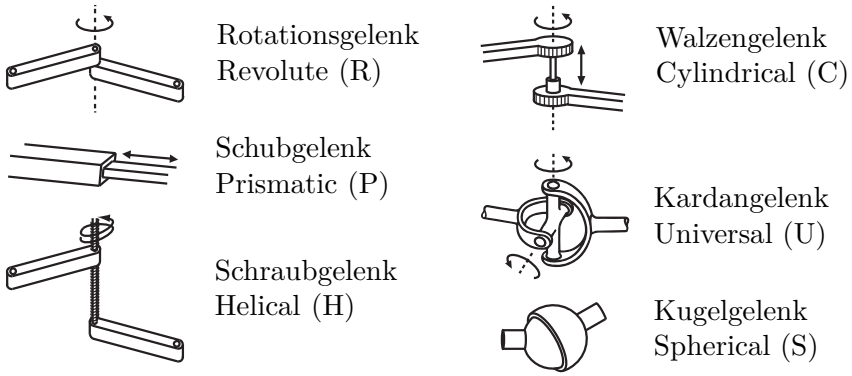


Abbildung 1.2: Unterschiedliche Gelenkstypen [1.1].

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* (auch *Drehgelenk*, engl. *Revolute*, R) ermöglicht die Rotation um eine Achse, während das *Schubgelenk* (engl. *Prismatic*, P) eine lineare Translationsbewegung ausführt. Mit dem *Schraubgelenk* (engl. *Helical*, H) wird eine simultane Rotation und lineare Translation entlang einer Schraubenachse ausgeführt. Diese drei Gelenkstypen bieten genau einen Freiheitsgrad. Weiters gibt es auch Gelenkstypen mit mehreren Freiheitsgraden. Dazu zählt das *Walzengelenk* (engl. *Cylindrical*, C), welches zwei Freiheitsgrade besitzt und im Gegensatz zum Schraubgelenk eine unabhängige Rotations-

Gelenkstyp	Freiheits-grade	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.

und Translationsbewegung entlang einer Gelenksachse erlaubt. Das *Kardangelenk* (engl. *Universal*, U) besteht aus zwei sich orthogonal schneidenden Rotationsgelenken und wird durch zwei Freiheitsgrade beschrieben. Drei Freiheitsgrade hat das *Kugelgelenk* (engl. *Spherical*, S), welches eine Bewegung ähnlich dem menschlichen Schultergelenk ermöglicht.

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

1.1.2 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. Diese unabhängigen Koordinaten $\mathbf{q}$ werden *generalisierte Koordinaten* genannt und alle Bewegungen zufolge dieser Koordinaten erfüllen stets alle Zwangsbedingungen.

1.1.3 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

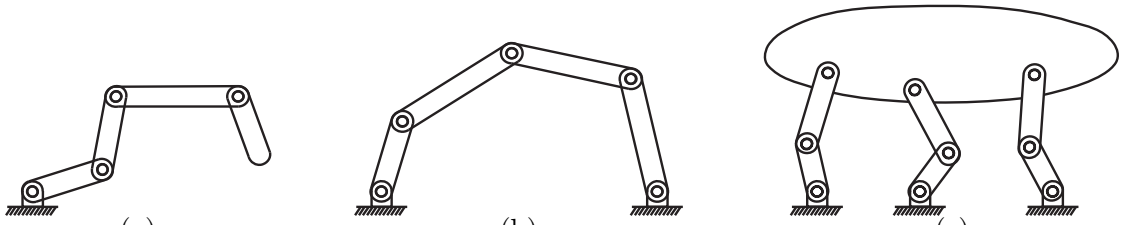


Abbildung 1.3: Kinematische Ketten [1.1]: (a) Offene bzw. serielle Kette, (b) geschlossene Kette, (c) Parallele Kinematik.

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 kinematischen Kette stellt die *parallele Kinematik* (Abbildung 1.3c) dar. Diese besteht aus einer bewegten Basis, welche über mehrere gleichartige *Stränge* (engl. *legs*) mit dem inertialfesten Boden verbunden sind. Die Stränge sind wiederum kinematische Ketten, welche für sich eine serielle oder parallele Kinematik aufweisen können. Roboter mit paralleler Kinematik weisen eine hohe strukturelle Steifigkeit auf und ermöglichen daher höhere Bewegungsgeschwindigkeiten im Vergleich zu Robotern mit serieller Kinematik. Als Nachteil erweist sich allerdings eine deutliche Einschränkung in der Bewegungsfreiheit des Roboters.

1.2 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 *Konfigurationraum* und dem *Aufgabenraum*. 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 *Konfigurationraum* (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 Aufgabenraum

Im Aufgabenraum (engl. *task space*) wird die Lage des Endeffektors bzw. des TCP beschrieben, während sich der Konfigurationsraum auf die Beschreibung des gesamten Roboters mit all seinen Starrkörpern bezieht. Der Aufgabenraum hat im Folgenden die Dimension m und wird im Allgemeinen so gewählt, dass die Aufgabe des Roboters geeignet beschrieben werden kann. Für das Zeichnen mit einem Stift auf ein Blatt Papier ist beispielsweise $m = 2$ ausreichend, um die Position des Stiftes am Papier zu charakterisieren. Soll der Roboter mit einer Schneidklinge entlang einer Kontur auf dem Papier schneiden, so wird auch die Orientierung der Klinge und somit $m = 3$ benötigt. Um einen Starrkörper mit einem Greifer im dreidimensionalen Raum zu manipulieren, sind alle Freiheitsgrade dieses Raums notwendig, also $m = 6$.

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

Der *Arbeitsraum* (engl. *workspace*, auch Arbeitsbereich) eines Roboters umfasst nun den gesamten Bereich, den der Roboter mit seinem Endeffektor erreichen kann. Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsräumen sind in Abbildung 1.4 dargestellt.

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

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 Aufgabenraum des Endeffektors bzw. TCP berechnet, d. h.

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \tag{1.2}$$

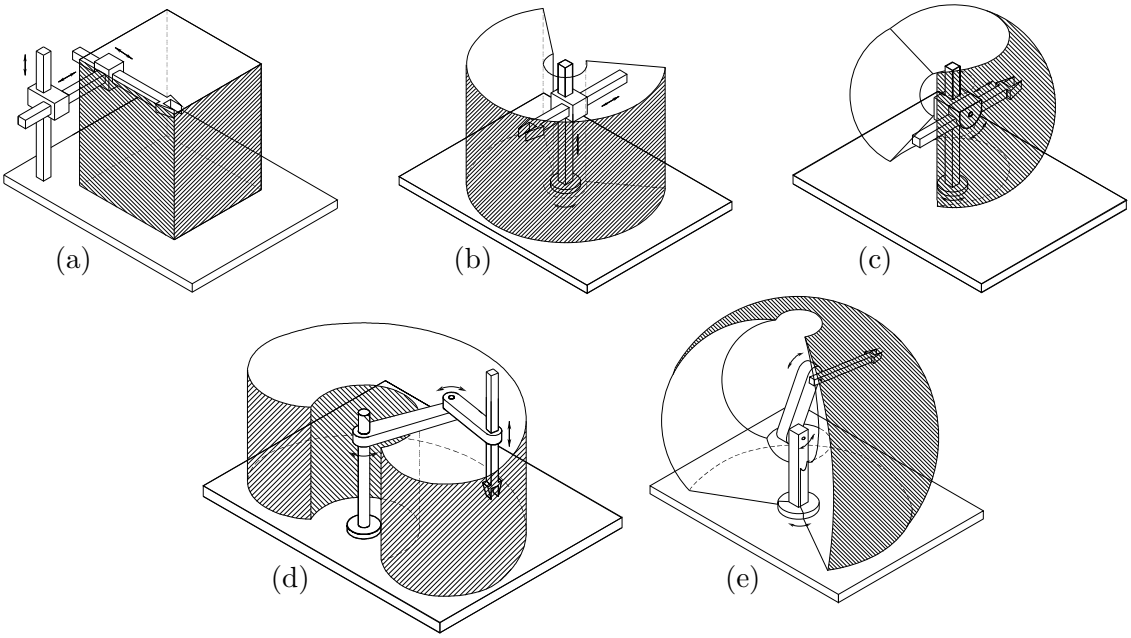


Abbildung 1.4: Beispiele für Roboter mit serieller Kinematik mit deren Arbeitsräumen [1.2]: (a) Kartesischer Roboter (PPP-Roboter), (b) Roboter mit zylindrischem Arbeitsraum (RPP-Roboter), (c) Roboter mit kugelförmigem Arbeitsraum (RRP-Roboter), (d) SCARA-Roboter (Selective Compliance Assembly Robot Arm, RRP-Roboter), (e) Kinematik eines klassischen Industrieroboters mit drei Achsen (RRR-Roboter).

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 (1.2) in der Form

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}_e) \tag{1.3}$$

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 (1.2) als auch (1.3) 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 (1.2) und (1.3) 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 Arbeitsraums des Roboters lösbar ist und eine, mehrere oder unendlich viele Lösungen liefert. Andererseits ist die inverse Kinematik von geschlossenen Ketten eindeutig lösbar (innerhalb des Arbeitsraums), während die Vorwärtskinematik mehrere oder unendlich viele Lösungen hat.

Die inverse Kinematik von seriellen kinematischen Ketten kann auf zwei unterschiedliche Arten gelöst werden, nämlich analytisch und numerisch. Eine analytische inverse Kinematik kann nur für relativ einfache oder für spezielle kinematische Ketten formu-

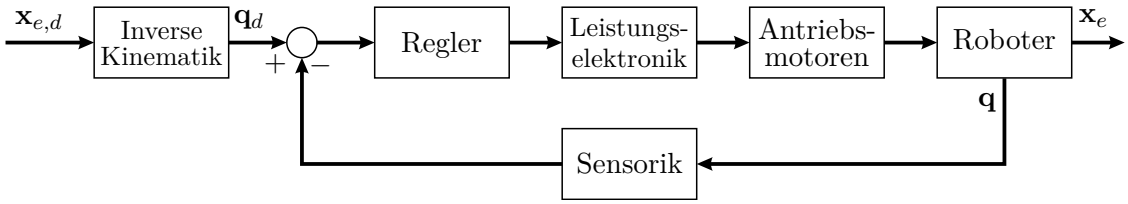


Abbildung 1.5: Regelung im Konfigurationsraum [1.2].

liert 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 (1.3) 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.

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 Aufgabenraum spezifiziert wird, wohingegen deren Ausführung sowohl im Konfigurationsraum (engl. *Joint Space Control*) als auch im Aufgabenraum (engl. *Task Space Control*) erfolgen kann.

1.3.1 Regelung im Konfigurationsraum

Soll der Roboter eine Bewegung im Aufgabenraum $\mathbf{x}_{e,d}(t)$ (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 1.5 als Blockdiagramm dargestellt. Die Regelung selbst wird als geschlossener Regelkreis über die gemessenen Gelenkspositionen 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

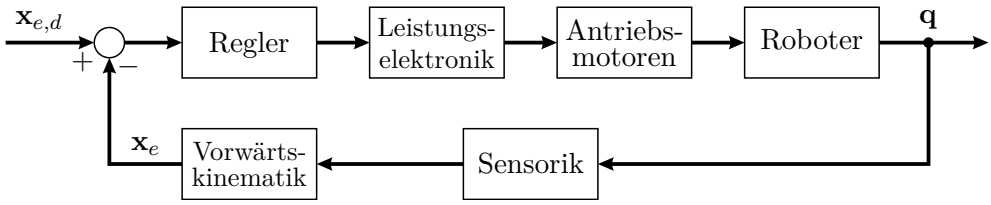


Abbildung 1.6: Regelung im Aufgabenraum, angepasst von [1.2].

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 Aufgabenraum

Bei der *Regelung im Aufgabenraum* gemäß Abbildung 1.6 ist die Kinematik implizit in den Regelkreis eingebettet und es wird direkt auf die Endeffektorpose geregelt. Dazu müssen jedoch die Vorwärtskinematik (1.2) und deren Ableitungen bei der Auswertung des Stellgesetzes berechnet werden, welche im Allgemeinen komplexe Ausdrücke sind. Die Regelung im Aufgabenraum muss zwingend als zentraler Regler implementiert werden, da für die Berechnung der Vorwärtskinematik die Positionen aller Gelenke benötigt werden. Der Reglerentwurf für kinematisch nichtredundante Roboter wird im Abschnitt 4.1 erklärt.

1.4 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.

2 Roboterkinematik

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

Dieses Kapitel 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 Aufgabenraum mithilfe der differentiellen Kinematik betrachtet.

2.1 Vorwärtskinematik

In diesem Abschnitt wird die allgemeine Beziehung der Vorwärtskinematik (1.2) 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.

Rotationsgelenk Translationsgelenk

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

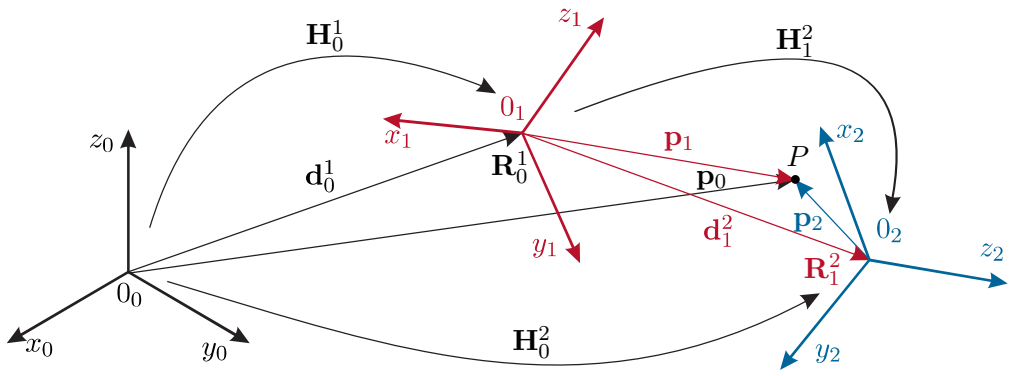


Abbildung 2.2: Homogene Transformationen am Beispiel der Koordinatensysteme $(0_0x_0y_0z_0)$, $(0_1x_1y_1z_1)$ und $(0_2x_2y_2z_2)$ [2.1].

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) und setzt sich aus der Position des Koordinatenursprungs $\mathbf{d}_0^1 \in \mathbb{R}^3$ und der orthogonalen Rotationsmatrix $\mathbf{R}_0^1 \in \mathrm{SO}(3)$[1] gemäß

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

zusammen[2]. Das Symbol $\mathbf{0}$ bezeichnet dabei allgemein einen Nullvektor oder eine Nullmatrix der geeigneten Dimension. In (2.1) ist $\mathbf{0}$ demnach ein 1×3-Nullvektor. Ein Koordinatensystem $(0_2x_2y_2z_2)$ wird bezüglich des Koordinatensystems $(0_0x_0y_0z_0)$ durch aufeinanderfolgende Transformationen mithilfe des Koordinatensystems $(0_1x_1y_1z_1)$ in der Form

$$\mathbf{H}_0^2 = \mathbf{H}_0^1\mathbf{H}_1^2 \tag{2.2}$$

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 $(0_2x_2y_2z_2)$ wird dieser Punkt mithilfe des homogenen Vektors $\tilde{\mathbf{p}}_2^{\mathrm{T}} = \begin{bmatrix} \mathbf{p}_2^{\mathrm{T}} & 1 \end{bmatrix}$ beschrieben. Analog dazu beschreiben die Vektoren $\tilde{\mathbf{p}}_i^{\mathrm{T}} = \begin{bmatrix} \mathbf{p}_i^{\mathrm{T}} & 1 \end{bmatrix}$, $i \in \{0, 1\}$, den selben Punkt P in Bezug auf das jeweilige Koordiantensystem $(0_ix_iy_iz_i)$. Der homogene

[1] $\mathrm{SO}(n)$ 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.

[2] $\mathrm{SE}(n)$ 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 $\tilde{\mathbf{p}}_0$ berechnet sich gemäß

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

Der Vektor $\mathbf{p}_0$ selbst, d. h. die ersten drei Einträge des Vektors $\tilde{\mathbf{p}}_0$ in (2.3), 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 $(0_0x_0y_0z_0)$ transformiert wurden, bevor diese addiert werden.

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

$$\mathbf{H}_{\mathrm{T}x,d} = \begin{bmatrix}\mathbf{E} & \begin{bmatrix}d\\0\\0\end{bmatrix}\\ \mathbf{0} & 1\end{bmatrix}, \qquad \mathbf{H}_{\mathrm{T}y,d} = \begin{bmatrix}\mathbf{E} & \begin{bmatrix}0\\d\\0\end{bmatrix}\\ \mathbf{0} & 1\end{bmatrix}, \qquad \mathbf{H}_{\mathrm{T}z,d} = \begin{bmatrix}\mathbf{E} & \begin{bmatrix}0\\0\\d\end{bmatrix}\\ \mathbf{0} & 1\end{bmatrix}, \tag{2.4}$$

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

$$\mathbf{H}_{\mathrm{R}i,\varphi} = \begin{bmatrix}\mathbf{R}_{i,\varphi} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix} \tag{2.5}$$

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 $(0_0x_0y_0z_0)$ und dem Koordinatensystem des Endeffektors $(0_ex_ey_ez_e)$ dargestellt. Gemäß (2.1) wird die homogene Transformation des Endeffektorkoordinatensystems $(0_ex_ey_ez_e)$ in Bezug auf das Inertialkoordinatensystem $(0_0x_0y_0z_0)$ durch

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

als Funktion der generalisierten Koordinaten $\mathbf{q} \in \mathbb{R}^n$ beschrieben. Weiters wird jedes Glied des Roboters mit einem Koordinatensystem $(0_ix_iy_iz_i)$, $i = 1, \ldots, n$, ausgestattet. Dabei ist es von Vorteil, jeweils eine Achse der Koordinatensysteme in die jeweilige Rotations- bzw.

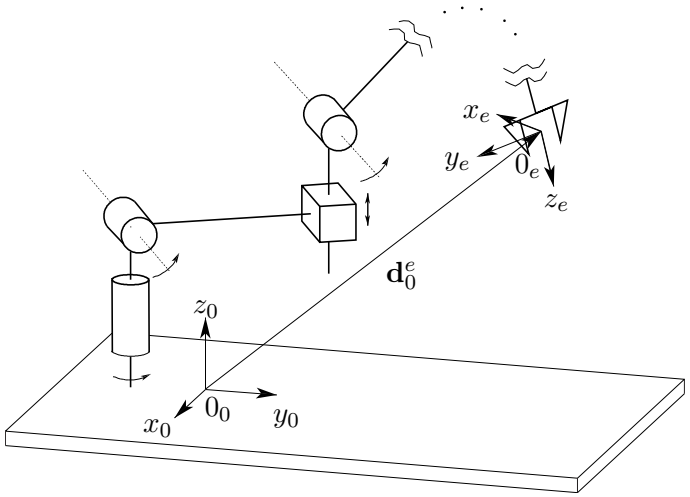


Abbildung 2.3: Schematische Darstellung einer seriellen kinematischen Kette [2.2].

Translationsachse zu legen. Wie in Abbildung 2.4 gezeigt, entsteht dadurch eine Kette von Koordinatensystemen, welche mithilfe der homogenen Transformationen in der Form

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

beschrieben wird, siehe (2.2). Jede der Transformationen $\mathbf{H}_{i-1}^i(q_i)$, $i = 1, \ldots, n$, 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 Robotergliedes n beschreibt. Damit folgt die Beschreibung der Roboterkinematik als homogene Transformation zu

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

Um die Vorwärtskinematik in der Form (1.2) zu erhalten, wird $\mathbf{H}_b^e(\mathbf{q})$ schließlich in Form der m Koordinaten des Aufgabenraums

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

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

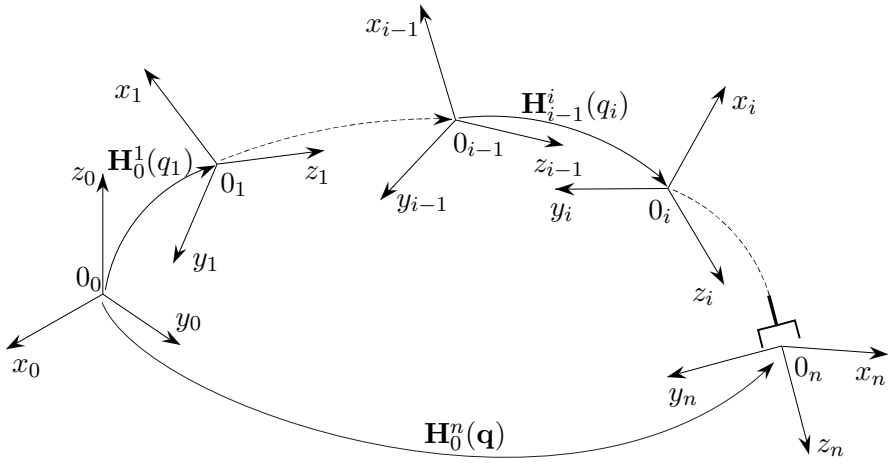


Abbildung 2.4: Koordinatensysteme eines Roboters mit serieller Kinematik [2.2].

2.1.3 Parametrierung der Orientierung

Im dreidimensionalen Raum besitzt ein Starrkörper 6 Freiheitsgrade, wovon 3 Freifreiheitsgrade die Orientierung beschreiben. Die Parametrierung dieser Orientierung kann auf unterschiedliche Weise mit unterschiedlicher Anzahl an Parametern erfolgen. Bei einer *Minimalkoordinatendarstellung* wird die Orientierung ebenfalls mit genau drei Winkeln parametriert. Beispiele dafür sind die *klassischen* Euler*-Winkel*, die *Roll-Nick-Gier-Winkel* und die *exponentiellen Koordinaten*. Darstellungen mit mehr als drei Parametern sind *Einheitsquaternionen* und die *Achse-Winkel-Darstellung* (engl. *axis-angle representation*, 4 Parameter) sowie die Rotationsmatrix (9 Parameter). Im Folgenden werden die mathematischen Grundlagen für die Minimalkoordinatendarstellungen eingeführt.

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

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

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

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

In (2.10) gilt für die Euler-Winkel $i = k$ 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 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 $\boldsymbol{\phi}_{zyz}(\cdot)$. Die zugehörige Rotationsmatrix $\mathbf{R}(\boldsymbol{\phi})$ lautet

$$\mathbf{R}(\boldsymbol{\phi}) = \mathbf{R}_{z,\varphi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\psi} = \begin{bmatrix} \mathrm{c}_\varphi\mathrm{c}_\theta\mathrm{c}_\psi - \mathrm{s}_\varphi\mathrm{s}_\psi & -\mathrm{c}_\varphi\mathrm{c}_\theta\mathrm{s}_\psi - \mathrm{s}_\varphi\mathrm{c}_\psi & \mathrm{c}_\varphi\mathrm{s}_\theta \\ \mathrm{s}_\varphi\mathrm{c}_\theta\mathrm{c}_\psi + \mathrm{c}_\varphi\mathrm{s}_\psi & -\mathrm{s}_\varphi\mathrm{c}_\theta\mathrm{s}_\psi + \mathrm{c}_\varphi\mathrm{c}_\psi & \mathrm{s}_\varphi\mathrm{s}_\theta \\ -\mathrm{s}_\theta\mathrm{c}_\psi & \mathrm{s}_\theta\mathrm{s}_\psi & \mathrm{c}_\theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{2.12}$$

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

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

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

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

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

Die RPY-Winkel verwenden die Konvention $\boldsymbol{\phi}_{zyx}(\cdot)$. Statt den Winkeln (2.10) wird aber häufig die Winkelkonvention $\boldsymbol{\phi}^{\mathrm{T}} = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}$ verwendet. Die Rotationsmatrix $\mathbf{R}(\boldsymbol{\phi})$ ergibt sich damit zu

$$\mathbf{R}(\boldsymbol{\phi}) = \mathbf{R}_{z,\alpha}\mathbf{R}_{y,\beta}\mathbf{R}_{x,\gamma} = \begin{bmatrix} \mathrm{c}_\alpha\mathrm{c}_\beta & \mathrm{c}_\alpha\mathrm{s}_\beta\mathrm{s}_\gamma - \mathrm{s}_\alpha\mathrm{c}_\gamma & \mathrm{c}_\alpha\mathrm{s}_\beta\mathrm{c}_\gamma + \mathrm{s}_\alpha\mathrm{s}_\gamma \\ \mathrm{s}_\alpha\mathrm{c}_\beta & \mathrm{s}_\alpha\mathrm{s}_\beta\mathrm{s}_\gamma + \mathrm{c}_\alpha\mathrm{c}_\gamma & \mathrm{s}_\alpha\mathrm{s}_\beta\mathrm{c}_\gamma - \mathrm{c}_\alpha\mathrm{s}_\gamma \\ -\mathrm{s}_\beta & \mathrm{c}_\beta\mathrm{s}_\gamma & \mathrm{c}_\beta\mathrm{c}_\gamma \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{2.15}$$

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

$$\boldsymbol{\phi}_{zyx}(\mathbf{R}) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}(r_{21}, r_{11}) \\ \operatorname{atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \\ \operatorname{atan2}(r_{32}, r_{33}) \end{bmatrix} \tag{2.16}$$

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

$$\boldsymbol{\phi}_{zyx}(\mathbf{R}) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}(-r_{21}, -r_{11}) \\ \operatorname{atan2}\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right) \\ \operatorname{atan2}(-r_{32}, -r_{33}) \end{bmatrix}. \tag{2.17}$$

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 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 $\mathbf{q}^{\mathrm{T}} = [q_1 \ q_2 \ \cdots \ q_6]$ 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

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

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

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

die Vorwärtskinematik gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} \mathbf{d}_0^6(\mathbf{q}) \\ \boldsymbol{\phi}_{zyz}(\mathbf{R}_0^6(\mathbf{q})) \end{bmatrix}, \tag{2.19}$$

wobei mit $\boldsymbol{\phi}_{zyz}(\cdot)$ 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 $(0_6 x_6 y_6 z_6)$ liegt auf der z_0-Achse und hat die selbe Orientierung wie das

Inertialkoordinatensystem. Die Koordinaten im Aufgabenraum lauten

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

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

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

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 $\boldsymbol{\phi}_{zyz}(\cdot)$ 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.

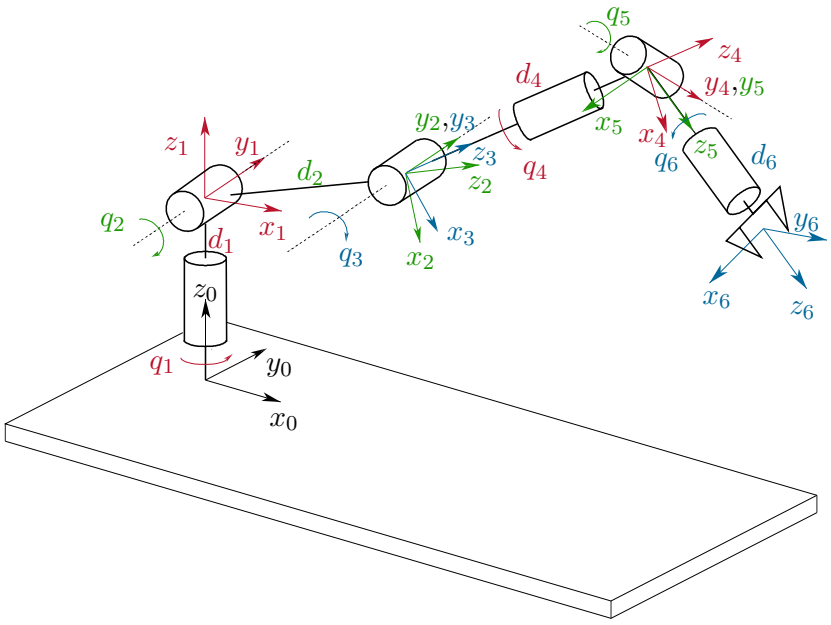


Abbildung 2.5: 6-Achs-Roboter mit sphärischem Handgelenk [2.2].

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 (1.2) invertiert werden muss (siehe (1.3)), um für eine gegebene Pose im Aufgabenraum $\mathbf{x}_e$ die zugehörige Konfiguration $\mathbf{q}$ zu berechnen. Die zu lösende Gleichung (1.2) 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 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.

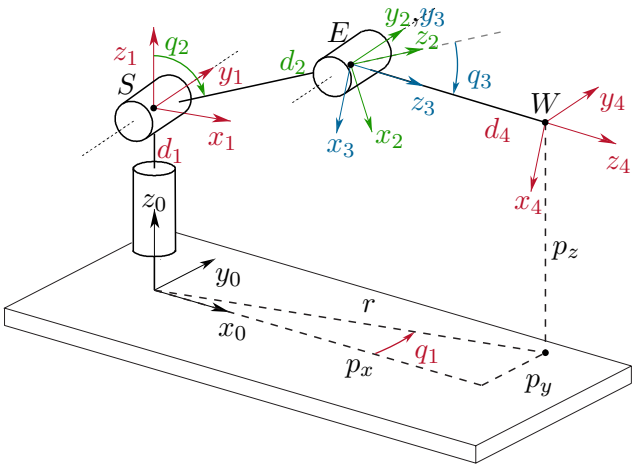


Abbildung 2.6: 3-Achs-Roboter in RRR-Struktur [2.2].

Der dreidimensionale Konfigurationsraum $\mathbf{q}^{\mathrm{T}} = [q_1 \; q_2 \; q_3]$, $n = 3$, erlaubt es, einen dreidimensionalen Aufgabenraum ($m = 3$) zu wählen. Als Koordinaten im Aufgabenraum wird der Ursprung des Endeffektorkoordinatensystems mit $\mathbf{x}_e = \mathbf{p}_0^4 = [p_x \; p_y \; p_z]^{\mathrm{T}}$ 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.

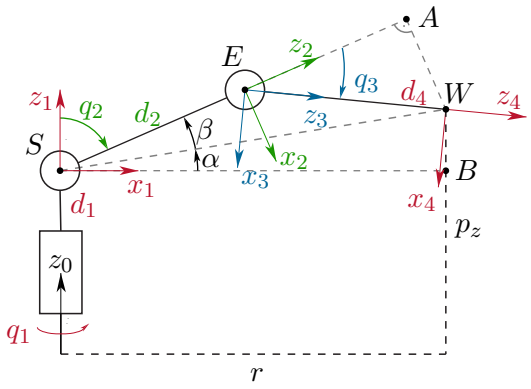


Abbildung 2.7: Ansicht der z_0-r-Ebene des 3-Achs-Roboters.

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

$$q_1 = \operatorname{atan2}(p_y, p_x) \ . \tag{2.22}$$

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

$$r^2 + (p_z - d_1)^2 = d_2^2 + d_4^2 - 2d_2 d_4 \cos(\pi - q_3) \ . \tag{2.23}$$

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

$$c_{q_3} = \cos(q_3) = \frac{r^2 + (p_z - d_1)^2 - d_2^2 - d_4^2}{2d_2 d_4} \tag{2.24}$$

und damit

$$q_3 = \pm \arccos(c_{q_3}) \ , \tag{2.25}$$

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

$$\alpha = \operatorname{atan2}(p_z - d_1, r) \tag{2.26}$$

berechnet und der Zusammenhang

$$\cos(\beta)\sqrt{r^2 + (p_z - d_1)^2} = d_2 + d_4 \cos(q_3) \tag{2.27}$$

ist durch zweimalige Projektion im Dreieck (SWA) ersichtlich. Unter Verwendung von (2.24) berechnet sich die Zwischengröße β zu

$$\beta = \arccos\left(\frac{r^2 + (p_z - d_1)^2 + d_2^2 - d_4^2}{2d_2\sqrt{r^2 + (p_z - d_1)^2}}\right) \tag{2.28}$$

und es folgt der verbleibende Winkel q_2 gemäß

$$q_2 = \frac{\pi}{2} - \alpha \mp \beta \ . \tag{2.29}$$

Bisher wurden also zwei unterschiedliche Lösungen gefunden: Die *Elbow-Up*-Lösung hat ein negatives Vorzeichen in (2.29) und ein positives Vorzeichen in (2.25) 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° 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.

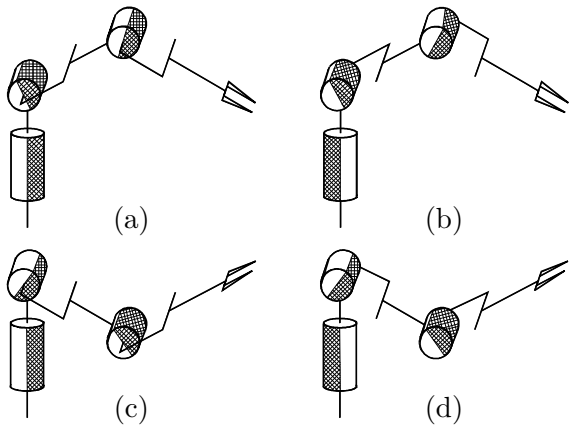


Abbildung 2.8: Vier Konfigurationen für die inverse Kinematik des 3-Achs-Roboters: (a) Shoulder-Right/Elbow-Up, (b) Shoulder-Left/Elbow-Up, (c) Shoulder-Right/Elbow-Down, (d) Shoulder-Left/Elbow-Down [2.2].

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 $\mathbf{q}^{\mathrm{T}} = [q_4 \ q_5 \ q_6]$ verbunden. In diesem Beispiel wird die inverse Kinematik des sphärischen Handgelenks betrachtet, d. h. es werden die Gelenkswinkel q_4, q_5 und q_6 für eine vorgegebene Orientierung $\mathbf{R}_{e,d}$ bzw. $\boldsymbol{\phi}_{e,d}$ des Endeffektorkoordinatensystems $(0_6 x_6 y_6 z_6)$ gesucht.

Zunächst wird die kinematische Kette anhand der homogenen Transformationen

$$\mathbf{H}_0^6(\mathbf{q}) = \mathbf{H}_0^4(q_4)\mathbf{H}_4^5(q_5)\mathbf{H}_5^6(q_6) = \begin{bmatrix} \mathbf{R}_0^6(\mathbf{q}) & \mathbf{d}_0^6(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.30}$$

mit

$$\mathbf{H}_0^4(q_4) = \mathbf{H}_{\mathrm{R}z,q_4}\mathbf{H}_{\mathrm{T}z,d_4} \tag{2.31}$$
$$\mathbf{H}_4^5(q_5) = \mathbf{H}_{\mathrm{R}y,q_5} \tag{2.32}$$
$$\mathbf{H}_5^6(q_6) = \mathbf{H}_{\mathrm{R}z,q_6}\mathbf{H}_{\mathrm{T}z,d_6} \tag{2.33}$$

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 (2.30) zu

$$\mathbf{R}_0^6(\mathbf{q}) = \begin{bmatrix} c_{q_4}c_{q_5}c_{q_6} - s_{q_4}s_{q_6} & -c_{q_4}c_{q_5}s_{q_6} - s_{q_4}c_{q_6} & c_{q_4}s_{q_5} \\ s_{q_4}c_{q_5}c_{q_6} + c_{q_4}s_{q_6} & -s_{q_4}c_{q_5}s_{q_6} + c_{q_4}c_{q_6} & s_{q_4}s_{q_5} \\ -s_{q_5}c_{q_6} & s_{q_5}s_{q_6} & c_{q_5} \end{bmatrix} \tag{2.34}$$

ausgewertet. Diese weist die gleiche Form auf wie die Rotationsmatrix für die klassischen Euler-Winkel in (2.12) mit $\varphi = q_4$, $\theta = q_5$ und $\psi = q_6$. Der Zusammenhang kommt durch die Wahl der Koordinatensysteme und durch die gewählte Sequenz von Drehungen in der kinematischen Struktur zustande. Durch diese mathematische Übereinstimmung folgt sofort mit den Gleichungen (2.13) und (2.14) die inverse Kinematik für das sphärische Handgelenk eines Roboters für eine gegebene Orientierung $\boldsymbol{\phi}_{e,d}$ bzw. für eine gegebene Rotationsmatrix $\mathbf{R}_{e,d}$ in der Form (2.12). Für jede Orientierung im Aufgabenraum existieren somit zwei Sätze von Lösungen $\mathbf{q}$.

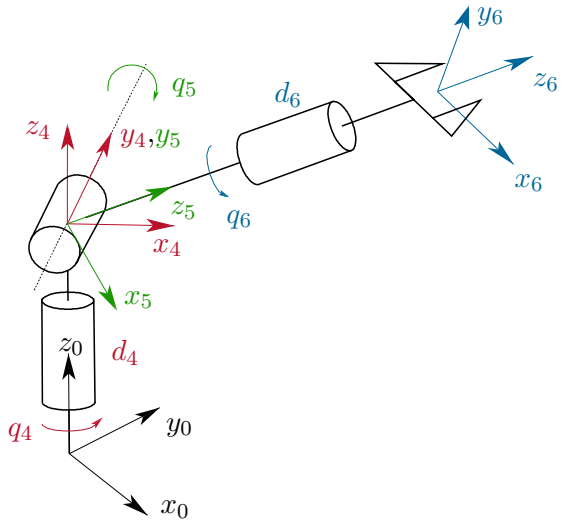


Abbildung 2.9: Sphärisches Handgelenk [2.2].

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 Aufgabenraum wird nun eine Endeffektorpose $\mathbf{x}_{e,d}^{\mathrm{T}} = [\mathbf{p}_{e,d}^{\mathrm{T}}\ \boldsymbol{\phi}_{e,d}^{\mathrm{T}}]$ vorgegeben und die zugehörigen Konfigurationen $\mathbf{q}^{\mathrm{T}} = [q_1\ \cdots\ q_6]$ werden gesucht. Es ist daher $\mathbf{R}_{0,d}^{6} = \mathbf{R}(\boldsymbol{\phi}_{e,d})$ gemäß (2.12) und $\mathbf{d}_{0,d}^{6} = \mathbf{p}_{e,d}$. 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

$$\mathbf{d}_0^4 = \mathbf{d}_0^6 - \mathbf{R}_{0,d}^{6}\begin{bmatrix}0 & 0 & d_6\end{bmatrix}^{\mathrm{T}}. \tag{2.35}$$

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 (2.35).
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 $\mathbf{R}_3^6(q_4, q_5, q_6) = \left(\mathbf{R}_0^3(q_1, q_2, q_3)\right)^{\mathrm{T}}\mathbf{R}_{0,d}^{6}$.
5. Lösen der inversen Kinematik für (q_4, q_5, q_6) analog zu Beispiel 2.3.

2.3 Differentielle Kinematik

Der vorigen beiden Abschnitte 2.1 und 2.2 zur Vorwärtskinematik bzw. inversen Kinematik haben den Zusammenhang zwischen den Punkten im Konfigurationsraum $\mathbf{q} \in \mathbb{R}^n$ und dem Aufgabenraum eines Roboters $\mathbf{x}_e \in \mathbb{R}^m$ hergestellt. Die *differentielle Kinematik* beschäftigt sich nun mit den *Geschwindigkeiten* im Konfigurations- und Aufgabenraum und den mathematischen Beziehungen zwischen den Räumen. Die wichtigsten Werkzeuge dabei sind die *analytische* und die *geometrische Jacobi-Matrix*. Zusätzlich können mit diesen Matrizen kinematische Eigenschaften von Robotern analysiert werden, wie z. B. Singularitäten, und auch Lösungen der inversen Kinematik für Roboterbewegungen im Aufgabenraum gefunden werden.

2.3.1 Analytische und geometrische Jacobi-Matrix

Die Pose des Endeffektors wird durch den Vektor $\mathbf{x}_e$ im m-dimensionalen Aufgabenraum beschrieben. Die Geschwindigkeit des Endeffektors $\dot{\mathbf{x}}_e$ wird durch zeitliches Ableiten der Vorwärtskinematik (2.9) gemäß

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix} = \underbrace{\frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q})}_{\mathbf{J}_a(\mathbf{q})} \dot{\mathbf{q}} = \mathbf{J}_a(\mathbf{q}) \dot{\mathbf{q}} \tag{2.36}$$

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

$$\mathbf{J}_a(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) \in \mathbb{R}^{m \times n} \tag{2.37}$$

berechnet. In (2.36) wurde die translatorische Endeffektorgeschwindigkeit $\dot{\mathbf{p}}_e$ und die rotatorische Endeffektorgeschwindigkeit in Minimalkoordinatendarstellung $\dot{\boldsymbol{\phi}}_e$ eingeführt. Die analytische Jacobi-Matrix gibt also an, welche zeitlichen Änderungsraten sich in den Koordinaten des Aufgabenraumes $\dot{\mathbf{x}}_e$ zufolge der Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$ ergeben. Direktes Berechnen von $\partial \mathbf{f}(\mathbf{q}) / \partial \mathbf{q}$ ist aber für die Koordinaten der Orientierung meist nicht möglich, weil $\boldsymbol{\phi}_e(\mathbf{q})$ nicht in Form von geschlossenen Ausdrücken vorliegt, sondern über die Rotationsmatrix $\mathbf{R}_0^e(\mathbf{q})$ berechnet werden muss, siehe (2.10) bzw. (2.13) und (2.14).

Die momentanen Endeffektorgeschwindigkeiten können auch in Form der translatorischen Geschwindigkeit $\dot{\mathbf{p}}_e = \dot{\mathbf{d}}_0^e$ und dem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e = \boldsymbol{\omega}_0^e$, beschrieben im Inertialkoordinatensystem $(0_0 x_0 y_0 z_0)$, ausgedrückt werden. Der Vektor mit den Komponenten $\boldsymbol{\omega}_0^e = [\omega_{0,1}^e \ \omega_{0,2}^e \ \omega_{0,3}^e]^{\mathrm{T}}$ ergibt sich aus der Rotationsmatrix des Endeffektors $\mathbf{R}_0^e$ und dessen Zeitableitung mit

$$\mathbf{S}(\boldsymbol{\omega}_0^e) = \dot{\mathbf{R}}_0^e (\mathbf{R}_0^e)^{\mathrm{T}} = \begin{bmatrix} 0 & -\omega_{0,3}^e & \omega_{0,2}^e \\ \omega_{0,3}^e & 0 & -\omega_{0,1}^e \\ -\omega_{0,2}^e & \omega_{0,1}^e & 0 \end{bmatrix}, \tag{2.38}$$

unter Verwendung des schiefsymmetrischen Operators $\mathbf{S}(\boldsymbol{\omega})$ [2.1]. Identisch zu (2.36) berechnet sich die translatorische Geschwindigkeit gemäß

$$\dot{\mathbf{p}}_e = \dot{\mathbf{d}}_0^e(\mathbf{q}) = \sum_{i=1}^{n} \left(\frac{\partial}{\partial q_i} \mathbf{d}_0^e(\mathbf{q}) \right) \dot{q}_i \ , \tag{2.39}$$

während sich die schiefsymmetrische Matrix zum Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ mithilfe von (2.38) zu

$$\mathbf{S}(\boldsymbol{\omega}_e) = \dot{\mathbf{R}}_0^e(\mathbf{q}) (\mathbf{R}_0^e(\mathbf{q}))^{\mathrm{T}} = \sum_{i=1}^{n} \left(\frac{\partial}{\partial q_i} \mathbf{R}_0^e(\mathbf{q}) \right) (\mathbf{R}_0^e(\mathbf{q}))^{\mathrm{T}} \dot{q}_i \tag{2.40}$$

ergibt. Bei den obigen Gleichungen (2.39) und (2.40) wird sichtbar, dass sich die Geschwindigkeiten $\dot{\mathbf{p}}_e$ und $\boldsymbol{\omega}_e$ als Produkt einer Matrix, welche nur von $\mathbf{q}$ abhängig ist, und den Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$ zusammensetzen. D. h. diese Gleichungen sind lediglich linear in den Gelenksgeschwindigkeiten $\dot{\mathbf{q}}$. Folglich kann Gleichung (2.39) in der Form

$$\dot{\mathbf{p}}_e = \mathbf{J}_v(\mathbf{q}) \dot{\mathbf{q}} = \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \dot{\mathbf{q}} \tag{2.41}$$

mit einer Matrix $\mathbf{J}_v(\mathbf{q})$ dargestellt werden. Ausgehend von (2.40) wird die Matrix $\mathbf{J}_\omega(\mathbf{q})$ gemäß

$$\boldsymbol{\omega}_e = \mathbf{J}_\omega(\mathbf{q})\dot{\mathbf{q}} \tag{2.42a}$$

$$\mathbf{J}_\omega(\mathbf{q}) = \begin{bmatrix} \frac{\partial \boldsymbol{\omega}_e}{\partial \dot{q}_1} & \dots & \frac{\partial \boldsymbol{\omega}_e}{\partial \dot{q}_n} \end{bmatrix} = \frac{\partial \boldsymbol{\omega}_e}{\partial \dot{\mathbf{q}}} \tag{2.42b}$$

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

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_g(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_v(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \\ \frac{\partial \boldsymbol{\omega}_e}{\partial \dot{\mathbf{q}}} \end{bmatrix} \dot{\mathbf{q}} \tag{2.43}$$

zusammengefasst. Die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ unterscheidet sich bei den rotatorischen Geschwindigkeiten von der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$. Während bei der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ in (2.37) implizit eine Minimalkoordinatendarstellung $\boldsymbol{\phi}_e = \boldsymbol{\phi}_{ijk}(\mathbf{R}_0^e)$ gewählt wurde, siehe (2.10), ist hingegen die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ aus (2.43) unabhängig von der Wahl der Koordinatendarstellung der Orientierung.

Sowohl $\dot{\boldsymbol{\phi}}_e$ als auch $\boldsymbol{\omega}_e$ beschreiben Drehgeschwindigkeiten des Endeffektorkoordinatensystems. Das zeitliche Integral von $\boldsymbol{\omega}_e$, also $\int \boldsymbol{\omega}_e \,\mathrm{d}t$, führt allerdings zu keinem sinnvollen Ausdruck für die Orientierung. Im Gegensatz dazu ist $\boldsymbol{\phi}_e = \int \dot{\boldsymbol{\phi}}_e \,\mathrm{d}t$ ein gültiger Zusammenhang, denn gemäß (2.36) ist $\dot{\boldsymbol{\phi}}_e$ eine echte Zeitableitung der Minimalkoordinatendarstellung $\boldsymbol{\phi}_e$ und beschreibt die momentane Drehwinkelgeschwindigkeit in diesen Koordinaten. Dies illustriert das folgende Beispiel.

Beispiel 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

$$\boldsymbol{\omega}_1(t) = \begin{cases} \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^\mathrm{T} & \text{für } 0 \le t < 1 \\ \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^\mathrm{T} & \text{für } 1 \le t \le 2 \end{cases} \tag{2.44}$$

und

$$\boldsymbol{\omega}_2(t) = \begin{cases} \begin{bmatrix} 0 & \frac{\pi}{2} & 0 \end{bmatrix}^\mathrm{T} & \text{für } 0 \le t < 1 \\ \begin{bmatrix} \frac{\pi}{2} & 0 & 0 \end{bmatrix}^\mathrm{T} & \text{für } 1 \le t \le 2 \end{cases} . \tag{2.45}$$

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

$$\int_0^2 \boldsymbol{\omega}_i(t) \,\mathrm{d}t = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & 0 \end{bmatrix}^\mathrm{T}, \quad i = 1, 2 , \tag{2.46}$$

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

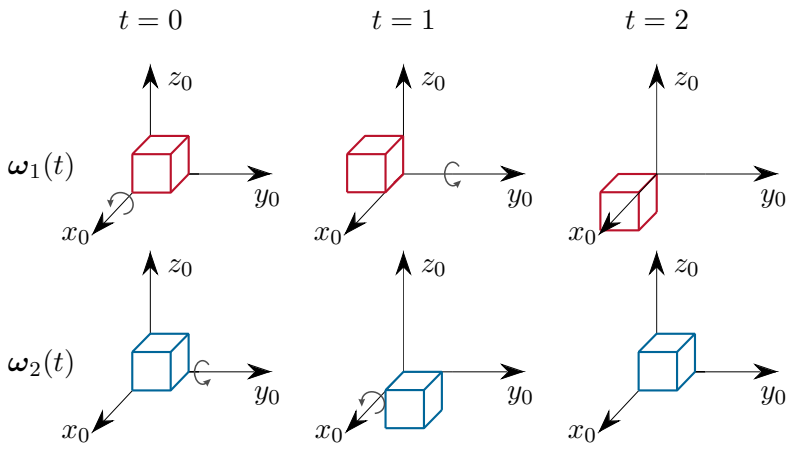


Abbildung 2.10: Bewegung des Starrkörpers, welcher mit $\boldsymbol{\omega}_1(t)$ bzw. mit $\boldsymbol{\omega}_2(t)$ rotiert wird [2.2].

Im Folgenden wird der Zusammenhang zwischen $\dot{\boldsymbol{\phi}}_e$ und $\boldsymbol{\omega}_e$ diskutiert. Zunächst wird der Vektor der Drehwinkelgeschwindigkeiten des Endeffektors $\boldsymbol{\omega}_e$ mithilfe von (2.11), (2.38) und gemäß

$$\mathbf{S}(\boldsymbol{\omega}_e) = \dot{\mathbf{R}}(\boldsymbol{\phi}_e)\mathbf{R}^{\mathrm{T}}(\boldsymbol{\phi}_e) \tag{2.47}$$

ausgedrückt. Wie in (2.40) zeigt sich, dass $\boldsymbol{\omega}_e$ linear in $\dot{\boldsymbol{\phi}}_e$ ist. Dies bedeutet, dass ein linearer Zusammenhang zwischen $\dot{\boldsymbol{\phi}}_e$ und $\boldsymbol{\omega}_e$ mit einer Transformationsmatrix $\mathbf{T}(\boldsymbol{\phi}_e)$ gemäß

$$\boldsymbol{\omega}_e = \mathbf{T}(\boldsymbol{\phi}_e)\dot{\boldsymbol{\phi}}_e \tag{2.48}$$

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

$$\mathbf{T}(\boldsymbol{\phi}_e) = \mathbf{T}_{ijk}(\boldsymbol{\phi}_e) = \frac{\partial}{\partial \dot{\boldsymbol{\phi}}_e}\boldsymbol{\omega}_e \tag{2.49}$$

und lautet für die klassischen Euler-Winkel

$$\mathbf{T}_{zyz}(\boldsymbol{\phi}_e) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi)\sin(\theta) \\ 0 & \cos(\varphi) & \sin(\varphi)\sin(\theta) \\ 1 & 0 & \cos(\theta) \end{bmatrix} \tag{2.50}$$

und für die RPY-Winkel

$$\mathbf{T}_{zyx}(\boldsymbol{\phi}_e) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi)\cos(\theta) \\ 0 & \cos(\varphi) & \sin(\varphi)\cos(\theta) \\ 1 & 0 & -\sin(\theta) \end{bmatrix}. \tag{2.51}$$

Schließlich folgt mit (2.48) der Zusammenhang zwischen den Jacobi-Matrizen $\mathbf{J}_g(\mathbf{q})$ und $\mathbf{J}_a(\mathbf{q})$

$$\begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \mathbf{T}(\boldsymbol{\phi}_e)\dot{\boldsymbol{\phi}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\phi}_e) \end{bmatrix} \underbrace{\begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\boldsymbol{\phi}}_e \end{bmatrix}}_{\dot{\mathbf{x}}_e} = \underbrace{\begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\phi}_e) \end{bmatrix} \mathbf{J}_a(\mathbf{q})}_{\mathbf{J}_g(\mathbf{q})} \dot{\mathbf{q}} \,. \tag{2.52}$$

Der Zusammenhang (2.52) zeigt also, dass die geometrische und die analytische Jacobi-Matrix für translatorische Geschwindigkeiten $\dot{\mathbf{p}}_e$ identisch sind. Da die beiden Matrizen lediglich unterschiedliche Darstellungen für die Drehrate verwenden, wird hierfür die Matrix $\mathbf{T}(\boldsymbol{\phi}_e)$ zum Umrechnen benötigt, siehe (2.48).

Beispiel 2.6 (Konstante Drehwinkelgeschwindigkeiten). Dieses Beispiel untersucht den nichtlinearen Zusammenhang (2.48) zwischen dem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ und der Drehrate $\dot{\boldsymbol{\phi}}_e$ in Minimalkoordinatendarstellung mit RPY-Winkel. Dazu wird ein Starrkörper mit der Orientierung $\boldsymbol{\phi}_e^{\mathrm{T}}(0) = [0,\ 0,\ 0]$ zum Zeitpunkt $t = 0$ in Rotation mit konstanter Drehrate versetzt.

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

$$\boldsymbol{\phi}_e(t) = \int_0^t \dot{\boldsymbol{\phi}}_e(\tau)\,\mathrm{d}\tau = \begin{bmatrix} t \\ 0.5t \\ 0 \end{bmatrix} \mathrm{rad} \,, \tag{2.53}$$

während $\boldsymbol{\omega}_e(t)$ zu jedem Zeitpunkt mithilfe von $\boldsymbol{\phi}_e(t)$ und $\dot{\boldsymbol{\phi}}_e$ aus (2.48) berechnet werden kann. Die Zeitverläufe dieser drei Vektoren sind in Abbildung 2.11 dargestellt. Durch die nichtlineare Abbildung über die Matrix $\mathbf{T}_{zyx}(\boldsymbol{\phi}_e)$ aus (2.51) schwankt der Vektor der Drehwinkelgeschwindigkeit $\boldsymbol{\omega}_e(t)$ sinusförmig zufolge der konstanten Drehrate $\dot{\boldsymbol{\phi}}_e$ in RPY-Winkel.

Im zweiten Schritt wird ein konstanter Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e = [1,\ 0.5,\ 0]^{\mathrm{T}}\,\mathrm{rad/s}$ vorgegeben und die zugehörige Drehung in Minimalkoordinatendarstellung berechnet. Dazu wird (2.48) auf $\dot{\boldsymbol{\phi}}_e$ umgeformt und gemäß

$$\boldsymbol{\phi}_e(t) = \int_0^t \dot{\boldsymbol{\phi}}_e(\tau)\,\mathrm{d}\tau = \int_0^t \mathbf{T}_{zyx}^{-1}(\boldsymbol{\phi}_e(\tau))\boldsymbol{\omega}_e\,\mathrm{d}\tau \tag{2.54}$$

integriert. In Abbildung 2.12 sind die Zeitverläufe für diese Drehung veranschaulicht. Darin ist deutlich zu sehen, dass der konstante Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$ auf eine schwankende Drehrate $\dot{\boldsymbol{\phi}}_e$ in Minimalkoordinatendarstellung führt. Weiters entsteht durch die Verkopplung über die Matrix $\mathbf{T}_{zyx}(\boldsymbol{\phi}_e)$ in (2.54) eine Drehung in allen drei Dimensionen von $\boldsymbol{\phi}_e$.

Dieses Beispiel zeigt, dass durch den nichtlinearen Zusammenhang (2.48) konstante Geschwindigkeiten $\boldsymbol{\omega}_e$ *nicht* auf konstante Geschwindigkeiten $\dot{\boldsymbol{\phi}}_e$ abgebildet werden und umgekehrt.

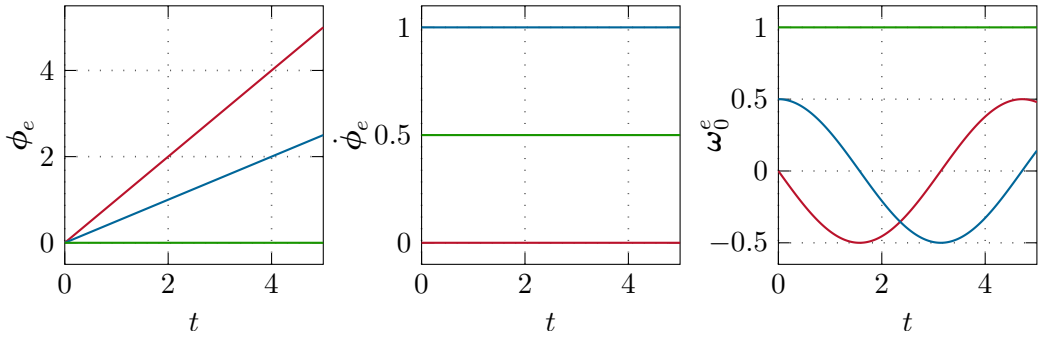


Abbildung 2.11: Drehung eines Starrkörpers mit konstanter Drehrate $\dot{\boldsymbol{\phi}}_e$ in Minimalkoordinatendarstellung mit RPY-Winkel.

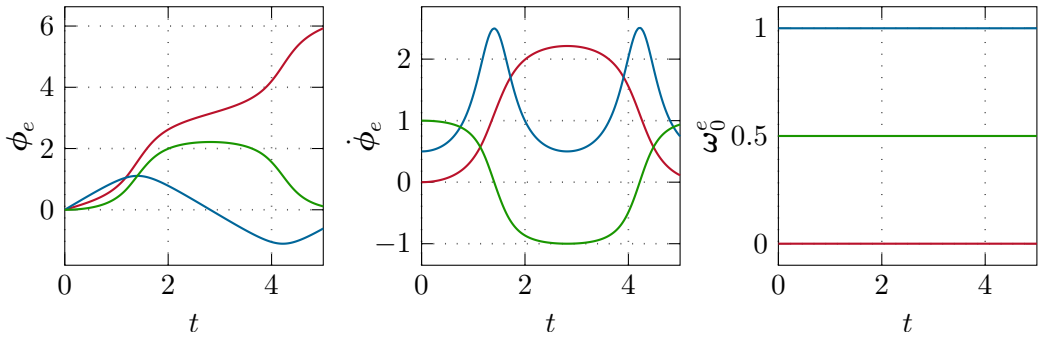


Abbildung 2.12: Drehung eines Starrkörpers mit konstantem Vektor der Drehwinkelgeschwindigkeiten $\boldsymbol{\omega}_e$.

2.3.2 Singularitäten

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

Kinematische Singularität

Wie in (2.43) gezeigt, bildet die geometrische Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ die Geschwindigkeiten im Konfigurationsraum $\dot{\mathbf{q}}$ *linear* auf die Geschwindigkeiten im Aufgabenraum $[\dot{\mathbf{p}}_e^{\mathrm{T}}\ (\boldsymbol{\omega}_e)^{\mathrm{T}}]^{\mathrm{T}}$ ab, wobei die Jacobi-Matrix selbst im Allgemeinen *nichtlinear* von der Konfiguration $\mathbf{q}$ abhängt.

Kinematische Singularitäten treten bei jenen Konfigurationen $\mathbf{q}$ auf, an denen es zu einem Rangverlust im Zeilenrang der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ kommt. Das bedeutet, dass die Abbildung (2.43) nicht mehr alle Geschwindigkeiten im Aufgabenraum $[\dot{\mathbf{p}}_e^{\mathrm{T}}\ (\boldsymbol{\omega}_e)^{\mathrm{T}}]^{\mathrm{T}}$ linear unabhängig voneinander erzeugen kann. Für einen kinematisch nichtredundanten Roboter mit $m = n$ ist $\mathbf{J}_g(\mathbf{q})$ quadratisch und Konfigurationen $\mathbf{q}$ mit Rangverlust können direkt aus der Gleichung $\det(\mathbf{J}_g(\mathbf{q})) = 0$ ermittelt werden. In der Nähe einer Singularität hat die geometrische Jacobi-Matrix zwar vollen Rang, ist aber schlecht konditioniert. Dadurch können in der Nähe einer Singularität große Geschwindigkeiten $\dot{\mathbf{q}}$ im Konfigurationsraum in einer oder mehreren Achsen auftreten, um bestimmte Bewegungen $\dot{\mathbf{p}}_e$ bzw. $\boldsymbol{\omega}_e$ im Aufgabenraum hervorzurufen.

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

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 $(0_i x_i y_i z_i)$, $i = 1, \ldots, 4$ ist in Abbildung 2.6 dargestellt. Aus dieser Abbildung werden die homogenen Transformationen zwischen den Koordinatensystemen in der Form

$$\mathbf{H}_0^1(q_1) = \mathbf{H}_{\mathrm{R}z,q_1}\mathbf{H}_{\mathrm{T}z,d_1} \qquad \mathbf{H}_1^2(q_2) = \mathbf{H}_{\mathrm{R}y,q_2}\mathbf{H}_{\mathrm{T}z,d_2}$$
$$\mathbf{H}_2^3(q_3) = \mathbf{H}_{\mathrm{R}y,q_3} \qquad \mathbf{H}_3^4 = \mathbf{H}_{\mathrm{T}z,d_4}$$

bestimmt und es folgt aus

$$\mathbf{H}_0^4(\mathbf{q}) = \mathbf{H}_0^1(q_1)\mathbf{H}_1^2(q_2)\mathbf{H}_2^3(q_3)\mathbf{H}_3^4 = \begin{bmatrix} \mathbf{R}_0^4(\mathbf{q}) & \mathbf{d}_0^4(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.55}$$

die Vorwärtskinematik des 3-Achs-Roboters gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) = \mathbf{d}_0^4(\mathbf{q}) = \begin{bmatrix} (d_4 \sin(q_2+q_3) + d_2 \sin(q_2)) \cos(q_1) \\ (d_4 \sin(q_2+q_3) + d_2 \sin(q_2)) \sin(q_1) \\ d_4 \cos(q_2+q_3) + d_2 \cos(q_2) + d_1 \end{bmatrix}. \tag{2.56}$$

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

$$\mathbf{J}_g(\mathbf{q}) = \frac{\partial \mathbf{d}_0^e}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \mathbf{f}(\mathbf{q}) = \mathbf{J}_a(\mathbf{q}) \tag{2.57}$$

und deren Determinante

$$\det(\mathbf{J}_g(\mathbf{q})) = d_2 d_4 \underbrace{\sin(q_3)}_{S_1} \Big(\underbrace{d_4 \sin(q_2+q_3) + d_2 \sin(q_2)}_{S_2} \Big) \tag{2.58}$$

berechnet. Da die Koordinaten des Aufgabenraums $\mathbf{x}_e$ nur die Endeffektorposition $\mathbf{d}_0^e(\mathbf{q})$ und keine Orientierung beinhalten, ist in (2.57) ersichtlich, dass die geometrische und die analytische Jacobi-Matrix identisch sind. In (2.58) 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 Roboter auf der z_0-Achse ($p_x = p_y = 0$, siehe Abbildung 2.6) und damit ist der Ausdruck S_2 gleich Null. Dies ist in Abbildung 2.13b dargestellt.

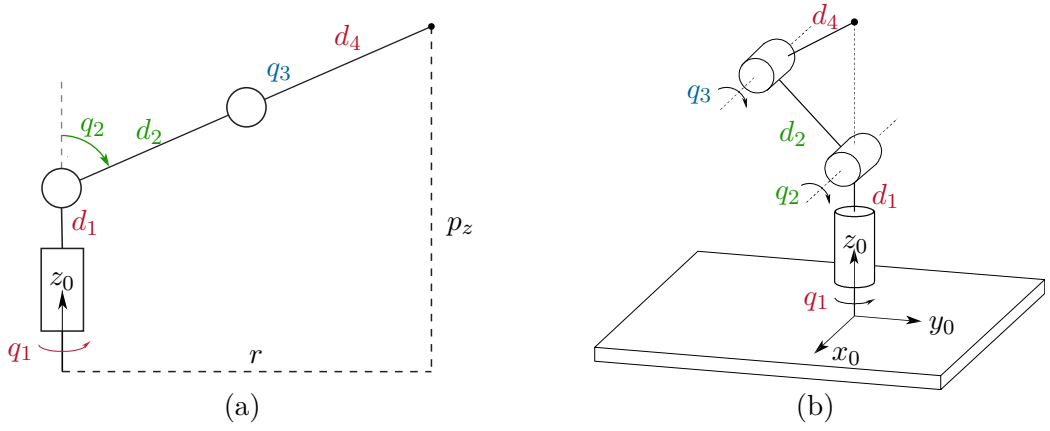


Abbildung 2.13: Kinematische Singularitäten des 3-Achs-Roboters: (a) Ausgestreckter Roboterarm ($S_1 = 0$), (b) Endeffektor liegt auf z_0-Achse ($S_2 = 0$) [2.2].

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 Arbeitsraums auftreten können. Zur Analyse der kinematischen Kette, siehe Abbildung 2.9 wird daher der Vektor der Drehwinkelgeschwindigkeiten mithilfe von (2.38) aus $\mathbf{R}_0^e(\mathbf{q})$ zu

$$\boldsymbol{\omega}_0^6(\mathbf{q}) = \begin{bmatrix} \cos(q_4)\sin(q_5)\dot{q}_6 - \dot{q}_5\sin(q_4) \\ \sin(q_4)\sin(q_5)\dot{q}_6 + \dot{q}_5\cos(q_4) \\ \dot{q}_6\cos(q_5) + \dot{q}_4 \end{bmatrix} \tag{2.59}$$

berechnet. Es folgt die zugehörige geometrische Jacobi-Matrix zu

$$\mathbf{J}_g(\mathbf{q}) = \begin{bmatrix} 0 & -\sin(q_4) & \cos(q_4)\sin(q_5) \\ 0 & \cos(q_4) & \sin(q_4)\sin(q_5) \\ 1 & 0 & \cos(q_5) \end{bmatrix} \tag{2.60}$$

mit der Determinante $\det(\mathbf{J}_g(\mathbf{q})) = -\sin(q_5)$. Singularitäten treten also für $q_5 = k\pi$, $k \in \mathbb{Z}$ auf. In diesen Fällen sind die Rotationsachsen der Gelenke 4 und 6 kollinear und die Bewegungsfreiheit des Roboters ist damit eingeschränkt: Ist $\dot{q}_4 = -\dot{q}_6$, so wird keine Bewegung am Endeffektor hervorgerufen. Weiters kann in dieser Singularität keine Rotationsgeschwindigkeit um die x_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 Aufgabenraum auftritt. Die Ursache für diese Art der Singularität liegt daher nicht in der geometrischen Jacobi-Matrix $\mathbf{J}_g(\mathbf{q})$ (die ausschließlich die kinematische Kette beschreibt), sondern in der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ bzw. in der Transformationsmatrix $\mathbf{T}(\boldsymbol{\phi})$.

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, siehe (2.50)

$$\det(\mathbf{T}_{zyz}(\boldsymbol{\phi}_e)) = -\sin(\theta) \tag{2.61}$$

und für die RPY-Winkel, siehe (2.51), folgt

$$\det(\mathbf{T}_{zyx}(\boldsymbol{\phi}_e)) = -\cos(\theta)\ . \tag{2.62}$$

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 des Endeffektors $\boldsymbol{\omega}_e$ *keine* zugehörige Winkelgeschwindigkeit der Euler-Winkel $\dot{\boldsymbol{\phi}}_e$ berechnet werden, siehe (2.52). Zusätzlich degenerieren in einer

Repräsentationssingularität auch die Gleichungen (2.13) und (2.14) bzw. (2.16) und (2.17). Es können φ und ψ nicht mehr unabhängig voneinander berechnet werden – nur noch die Summe bzw. die Differenz von φ und ψ. Diese Singularitäten lassen sich durch eine Nicht-Minimalkoordinatendarstellung der Orientierung (z. B. Quaternionen) beheben.

2.4 Inverse differentielle Kinematik

Mit der inversen Kinematik in Abschnitt 2.2 werden Konfigurationen $\mathbf{q}$ im Konfigurationsraum für eine gegebene Pose $\mathbf{x}_{e,d}$ im Aufgabenraum gesucht. Bei der *differentiellen inversen Kinematik* werden nun zusätzlich die Geschwindigkeiten und Beschleunigungen berücksichtigt. Dadurch können nicht nur Lösungen für statische Posen $\mathbf{x}_{e,d}$ gefunden werden, sondern es können gewünschte Zeitverläufe $\mathbf{x}_{e,d}(t)$, $\dot{\mathbf{x}}_{e,d}(t)$ und $\ddot{\mathbf{x}}_{e,d}(t)$ in Zeitverläufe im Konfigurationsraum $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ und $\ddot{\mathbf{q}}(t)$ übertragen werden.

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

2.4.1 Inverse differentielle Kinematik für Geschwindigkeiten

Wenn die Anzahl $m = \dim(\mathbf{x}_e)$ der Koordinaten des Endeffektors im Aufgabenraum $\mathbf{x}_e$ gleich der Anzahl $n = \dim(\mathbf{q})$ der mechanischen Freiheitsgrade $\mathbf{q}$ des Roboters ist, d. h. $m = n$, dann ist die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ quadratisch. Außerhalb von Singularitäten ist $\mathbf{J}_a(\mathbf{q})$ regulär und es können daher die Geschwindigkeiten im Konfigurationsraum $\dot{\mathbf{q}}$ direkt durch Inversion von $\mathbf{J}_a(\mathbf{q})$ in der Form, siehe (2.36)

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})\dot{\mathbf{x}}_{e,d} \tag{2.63}$$

aus den gewünschten Geschwindigkeiten im Aufgabenraum $\dot{\mathbf{x}}_{e,d}$ bestimmt werden. Die Trajektorie der generalisierten Koordinaten $\mathbf{q}(t)$ wird nun einfach über Zeitintegration aus (2.63) mit der Anfangsposition $\mathbf{q}(0)$ gemäß

$$\mathbf{q}(t) = \int_0^t \dot{\mathbf{q}}(\tau)\,\mathrm{d}\tau + \mathbf{q}(0) \tag{2.64}$$

berechnet. Bei der Implementierung in einem Digitalrechner muss allerdings eine Zeitdiskretisierung eingeführt werden und (2.64) mit numerischen Methoden integriert werden, z. B. dem *expliziten* EULER*-Verfahren*. Die resultierende Gleichung lautet

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + T_a\mathbf{J}_a^{-1}(\mathbf{q}(t_k))\dot{\mathbf{x}}_{e,d}\ , \tag{2.65}$$

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

Um das obige Problem zu beheben, wird der *Fehler in den Koordinaten des Aufgabenraums* gemäß

$$\mathbf{e} = \mathbf{x}_{e,d} - \mathbf{x}_e = \mathbf{x}_{e,d} - \mathbf{f}(\mathbf{q}) \tag{2.66}$$

eingeführt und dessen Zeitableitung ist

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_{e,d} - \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} \ , \tag{2.67}$$

unter Verwendung der differentiellen Kinematik (2.37). Anstatt die Gelenksgeschwindigkeit $\dot{\mathbf{q}}$ gemäß (2.63) vorzugeben, wird nun der Fehler $\mathbf{e}$ in der Form

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})(\dot{\mathbf{x}}_{e,d} + \mathbf{K}\mathbf{e}) \tag{2.68}$$

mit der Verstärkungsmatrix $\mathbf{K}$ rückgeführt. Eingesetzt in (2.67) ergibt sich das lineare Fehlersystem

$$\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = \mathbf{0} \ , \quad \mathbf{e}(0) = \mathbf{e}_0 = \mathbf{x}_{e,d}(0) - \mathbf{f}(\mathbf{q}(0)) \ . \tag{2.69}$$

Dieses Fehlersystem ist asymptotisch stabil, wenn die Verstärkungsmatrix $\mathbf{K}$ positiv definit gewählt wird und die Konvergenzrate hängt von den Eigenwerten dieser Matrix ab. Zusätzlich werden die Fehlerdynamiken (2.69) entkoppelt, wenn $\mathbf{K} = \mathrm{diag}(k_1, \ldots, k_m)$ mit positiven Einträgen k_i, $i = 1, \ldots, m$ gewählt wird und es folgt

$$\frac{\mathrm{d}}{\mathrm{d}t} e_i = -k_i e_i \ , \quad e(0) = e_{0,i} \tag{2.70}$$

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

$$e_i(t) = e_{0,i} \exp(-k_i t) \ . \tag{2.71}$$

ab. Für systemtheoretische Betrachtungen sei auf die Vorlesung *Automatisierung* [2.3] verwiesen.

Analog zu (2.65) kann auch für (2.68) eine zeitdiskrete Formulierung mithilfe des expliziten EULER-Integrationsverfahrens gefunden werden. Mit (2.66) folgt daraus die *zeitdiskrete differentielle inverse Kinematik*

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + T_a \mathbf{J}_a^{-1}(\mathbf{q}(t_k))\Big(\dot{\mathbf{x}}_{e,d}(t_k) + \mathbf{K}(\mathbf{x}_{e,d}(t_k) - \mathbf{f}(\mathbf{q}(t_k)))\Big) \ , \tag{2.72}$$

mit der positiv definiten Verstärkungsmatrix $\mathbf{K}$.

2.4.2 Inverse differentielle Kinematik für Beschleunigungen

Mithilfe der vorgestellten Rückführung des Fehlers $\mathbf{e}$ in (2.68) und (2.72) kann eine vorgegebene Bewegung des Endeffektors $\mathbf{x}_{e,d}(t)$ in den zugehörigen zeitlichen Verlauf im Konfigurationsraum $\mathbf{q}(t)$ für einen kinematisch nichtredundanten Roboter umgerechnet werden. Die Algorithmen liefern dabei die Position $\mathbf{q}(t)$ und die Geschwindigkeit $\dot{\mathbf{q}}(t)$ im Konfigurationsraum. Viele modellbasierte Regelungskonzepte, Trajektorien- und Pfadplanungsalgorithmen benötigen aber zusätzlich auch die Beschleunigungen im Konfigurationsraum $\ddot{\mathbf{q}}(t)$. Daher wird der obige Algorithmus im Folgenden auf Beschleunigungen erweitert.

Der Zusammenhang zwischen den Beschleunigungen im Konfigurationsraum und im Aufgabenraum wird als Zeitableitung von (2.36) gemäß

$$\ddot{\mathbf{x}}_e = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \ . \tag{2.73}$$

gefunden. Für kinematisch nichtredundante Roboter lässt sich die analytische Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ außerhalb von Singularitäten invertieren. Analog zu (2.63) wird diese Gleichung auf $\ddot{\mathbf{q}}$ zu

$$\ddot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})\Big(\ddot{\mathbf{x}}_{e,d} - \dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\Big) \tag{2.74}$$

umgeformt. Wird nun der Fehler $\mathbf{e}$ und der Geschwindigkeitsfehler $\dot{\mathbf{e}}$ in der Form

$$\ddot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})\Big(\ddot{\mathbf{x}}_{e,d} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_0\mathbf{e} - \dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\Big) \tag{2.75}$$

zurückgeführt, so folgt die Fehlerdynamik zu

$$\ddot{\mathbf{e}} + \mathbf{K}_1\dot{\mathbf{e}} + \mathbf{K}_0\mathbf{e}_x = \mathbf{0} \tag{2.76a}$$

$$\mathbf{e}(0) = \mathbf{e}_0 = \mathbf{x}_{e,d}(0) - \mathbf{f}(\mathbf{q}(0)) \tag{2.76b}$$

$$\dot{\mathbf{e}}(0) = \dot{\mathbf{e}}_0 = \dot{\mathbf{x}}_{e,d}(0) - \mathbf{J}_a(\mathbf{q}(0))\dot{\mathbf{q}}(0) \tag{2.76c}$$

Mit den diagonalen Verstärkungsmatrizen $\mathbf{K}_0 = \mathrm{diag}(k_{0,1}, k_{0,2}, \ldots, k_{0,m})$ und $\mathbf{K}_1 = \mathrm{diag}(k_{1,1}, k_{1,2}, \ldots, k_{1,m})$ folgen die m entkoppelten Gleichungen der Fehlerdynamik

$$\ddot{e}_i + k_{1,i}\dot{e}_i + k_{0,i}e_i = 0 \ , \quad i = 1, \ldots, m \ , \tag{2.77}$$

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

2.5 Literatur

[2.1] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2022)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2022. 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] A. Kugi, *Skriptum zur VU Automatisierung (WS 2022/2023)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2022. Adresse: https://www.acin.tuwien.ac.at/bachelor/automatisierung/.

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 Aufgabenraum mit $\mathbf{x}_e(t)$ beschreiben. Bei manchen Anwendungen ist die Trajektorie des Roboters vollständig durch die zu lösende Aufgabe vorgegeben, wenn beispielsweise der Endeffektor einem Objekt folgen soll. Häufig wird aber auch nur eine Start- und Zielkonfiguration oder eine Start- und Zielpose für eine Roboterbewegung vorgegeben und die Bewegung zwischen diesen Punkten im Konfigurations- bzw. Aufgabenraum kann frei gewählt werden. Die Aufgabe der *Trajektorienplanung* ist es nun, Trajektorien zu berechnen, welche folgende Anforderungen erfüllen:

- Die Trajektorie muss eine hinreichend glatte Funktion der Zeit sein, damit sie durch das Starrkörpersystem mit den vorhandenen Aktuatoren bzw. den unterlagerten Regelkreisen ausgeführt werden kann.
- Die physikalischen Beschränkungen für die Gelenkspositionen und Gelenksgeschwindigkeiten sowie die Beschleunigungen und Drehmomente müssen eingehalten werden.
- Die Berechnung einer Trajektorie kann viel Zeit in Anspruch nehmen, wenn viele Randbedingungen wie Hindernisse im Aufgabenraum oder Zeitoptimalität berücksichtigt werden sollen. Die berechneten Trajektorien müssen hingegen einfach auszuwerten sein, damit dies während der Ausführung in Echtzeit passieren kann.

In diesem Kapitel wird eine allgemeine Trajektorie aus zwei Teilen zusammengesetzt, nämlich einem *Pfad*, also einer rein geometrischen Beschreibung der Bewegung im Konfigurations- oder Aufgabenraum, und einer *Zeitparametrierung*. In den weiteren Abschnitten werden einerseits geradlinige Bewegungen zwischen einem Start- und Zielpunkt berechnet und andererseits 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 Aufgabenraum des Roboters definiert werden. Dabei wird angenommen, dass $s = 0$ beim Startpunkt gilt und mit $s = 1$ der Zielpunkt erreicht wird. Weiters spezifiziert nun die Zeitparametrierung $s(t)$ den Zeitverlauf von s und weist jedem Zeitpunkt t zwischen dem Anfangszeitpunkt $t = 0$ und dem Endzeitpunkt $t = T$ einen Wert zwischen 0 und 1 zu. Damit ist die Zeitparametrierung als Funktion $s : [0, T] \to [0, 1]$ definiert.

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

$$x = x(s(t)) \tag{3.1a}$$

$$\dot{x} = \frac{\partial x}{\partial s}\dot{s} \tag{3.1b}$$

$$\ddot{x} = \frac{\partial x}{\partial s}\ddot{s} + \frac{\partial^2 x}{\partial s^2}\dot{s}^2 \tag{3.1c}$$

berechnet.

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

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 und die genaue Bewegung dazwischen „frei" ist. Diese Art der Bewegung wird *Punkt-zu-Punkt-Bewegung* (engl. *point-to-point motion*, PTP) genannt und ist im einfachsten Fall ein geradliniger Pfad im Konfigurationraum. Mit dieser Wahl des Pfades ist immer gewährleistet, dass die mechanischen Beschränkungen der Gelenkspositionen $q_{i,min} \leq q_i \leq q_{i,max}$, $i = 1, \ldots, n$, stets eingehalten werden. Solche geradlinigen Bewegungen sind auch im Aufgabenraum des Roboters möglich und diese werden als *Linearbewegung* (engl. *linear motion*, LIN) bezeichnet. Bei einer Linearbewegung ist hingegen nicht garantiert, dass die Beschränkungen in den Gelenkspositionen erfüllt sind.

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

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

mit den Ableitungen, vgl. (3.1)

$$\frac{\partial x}{\partial s} = x_1 - x_0 \tag{3.3a}$$

$$\frac{\partial^2 x}{\partial s^2} = 0 \tag{3.3b}$$

definiert. Eine geradlinige Bewegung im Konfigurationsraum führt im Allgemeinen *nicht* zu einer geradlinigen Bewegung im Aufgabenraum aufgrund des im Allgemeinen nichtlinearen Zusammenhangs über die Roboterkinematik (1.2) bzw. (1.3). 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 Aufgabenraum geplant werden. Der 2-Achs-Roboter mit den Gelenkswinkeln $\mathbf{q}^{\mathrm{T}} = [q_1 \; q_2]$ und den Abmessungen d_1 und d_2 ist in Abbildung 3.1a dargestellt. Die Gelenkswinkel sind mechanisch mit $0° \leq q_1 \leq 180°$ und $0° \leq q_2 \leq 150°$ beschränkt. Der Startpunkt des Pfades ist $\mathbf{q}_0^{\mathrm{T}} = [q_{0,1} \; q_{0,2}]$ und der Endpunkt wird mit $\mathbf{q}_1^{\mathrm{T}} = [q_{1,1} \; q_{1,2}]$ bezeichnet.

Konfigurationsraum Im Konfigurationsraum werden geradlinige Pfade in der Form (3.2) für jedes Gelenk einzeln geplant, d. h.

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

bzw. in Vektorschreibweise

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

In den Abbildungen 3.1b und 3.1c ist der durch die Gelenkswinkelbeschränkungen zulässige Konfigurations- bzw. Aufgabenraum 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 Aufgabenraum, siehe Abbildung 3.1c.

Aufgabenraum Mithilfe der Vorwärtskinematik des 2-Achs-Roboters $\mathbf{f}(\mathbf{q})$ gemäß

$$\mathbf{x}_e = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{p}_e = \mathbf{f}(\mathbf{q}) = \begin{bmatrix} d_1 \cos(q_1) + d_2 \cos(q_1 + q_2) \\ d_1 \sin(q_1) + d_2 \sin(q_1 + q_2) \end{bmatrix} \tag{3.6}$$

werden zunächst der Start- und der Endpunkt $\mathbf{q}_0$ bzw. $\mathbf{q}_1$ in den Aufgabenraum gemäß

$$\mathbf{x}_0 = \mathbf{f}(\mathbf{q}_0) = [x_{0,1} \; x_{0,2}]^{\mathrm{T}} \tag{3.7}$$

$$\mathbf{x}_1 = \mathbf{f}(\mathbf{q}_1) = [x_{1,1} \; x_{1,2}]^{\mathrm{T}} \tag{3.8}$$

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

$$\mathbf{x}_e(s) = \begin{bmatrix} x_{e,1}(s) \\ x_{e,2}(s) \end{bmatrix} = \begin{bmatrix} x_{0,1} + s(x_{1,1} - x_{0,1}) \\ x_{0,2} + s(x_{1,2} - x_{0,2}) \end{bmatrix} = \mathbf{x}_0 + s(\mathbf{x}_1 - \mathbf{x}_0) , \quad s \in [0,1] , \tag{3.9}$$

geplant. Um diese Bewegung auszuführen, kann einerseits die inverse differentielle Kinematik aus Abschnitt 2.4 verwendet werden. Damit wird zunächst die

Roboterbewegung in den Konfigurationsraum umgerechnet und anschließend kann diese mithilfe eines Reglers im Konfigurationsraum ausgeführt werden. Andererseits kann die Regelung auch im Aufgabenraum erfolgen (siehe Abschnitt 1.3.2) und der Pfad (3.9) zusammen mit einer geeigneten Zeitparametrierung $s(t)$ kann direkt im geschlossenen Regelkreis vorgegeben werden.

Der geradlinige Pfad im Aufgabenraum des 2-Achs-Roboters ist in Abbildung 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 Roboter verlässt und die untere Gelenkswinkelbeschränkung von q_1 verletzt wird. Dies zeigt, dass besonders bei der Pfadplanung im Aufgabenraum auf die Einhaltung der mechanischen und physikalischen Beschränkungen der Gelenkswinkel, -geschwindigkeiten und -beschleunigungen geachtet werden muss.

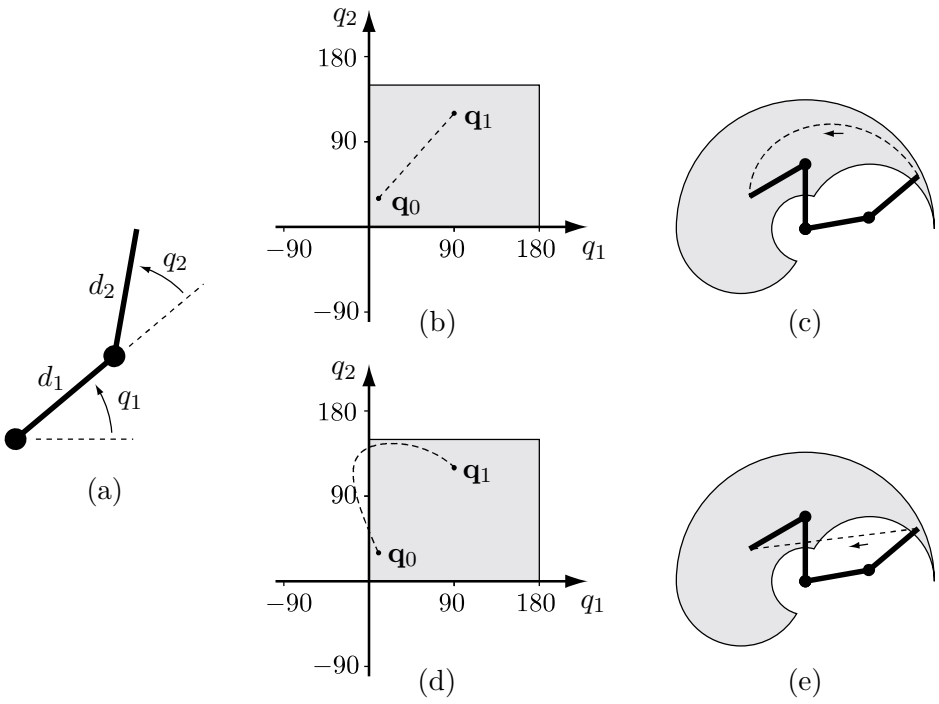


Abbildung 3.1: Geradlinige Pfade eines 2-Achs-Roboters: (a) Roboterkinematik und generalisierte Koordinaten $\mathbf{q}$, (b) geradliniger Pfad im Konfigurationsraum, (c) zugehöriger kurviger Pfad im Aufgabenraum, (d) kurviger Pfad im Konfigurationsraum, welcher aus einem geradlinigen Pfad im Aufgabenraum resultiert, (e) geradliniger Pfad im Aufgabenraum, welcher die Gelenkswinkelbeschränkungen 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 parametriert wird und 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

$$x = x_0 + s(x_1 - x_0) \tag{3.10a}$$
$$\dot{x} = \dot{s}(x_1 - x_0) \tag{3.10b}$$
$$\ddot{x} = \ddot{s}(x_1 - x_0) \ . \tag{3.10c}$$

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 eine Bewegung intuitiv durch ein trapezförmiges Geschwindigkeitsprofil bestehend aus einer Beschleunigungsphase, einer Phase konstanter Geschwindigkeit und einer Verzögerungsphase beschrieben.

3.3.1 Polynom 3. Ordnung

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

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{3.11a}$$
$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{3.11b}$$

können insgesamt 4 Randbedingungen vorgegeben werden, um die vier Koeffizienten $a_0, \ldots, a_3$ 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

$$a_0 = 0 \ , \quad a_1 = 0 \ , \quad a_2 = \frac{3}{T^2} \ , \quad a_3 = -\frac{2}{T^3} \ . \tag{3.12}$$

Die Trajektorie (3.10) zusammen mit (3.11) und (3.12) ist damit vollständig spezifiziert und in Abbildung 3.2 schematisch dargestellt. Aus (3.11b) wird der Zeitpunkt $t = T/2$ bestimmt, bei dem die maximale Geschwindigkeit

$$|\dot{x}_{max}| = \frac{3}{2T}|x_1 - x_0| \tag{3.13}$$

auftritt, vgl. (3.10b). Die maximale Beschleunigung ist

$$|\ddot{x}_{max}| = \frac{6}{T^2}|x_1 - x_0| \ , \tag{3.14}$$

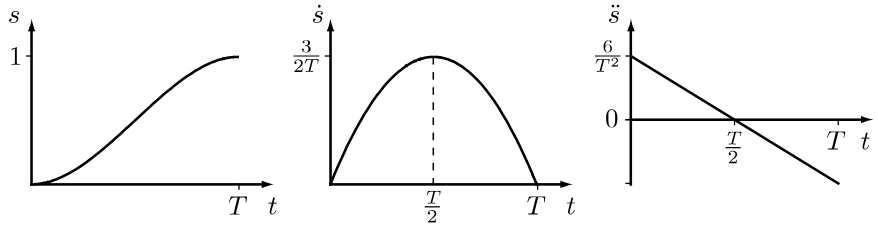


Abbildung 3.2: Schematische Darstellung der Funktionen $s(t)$, $\dot{s}(t)$ und $\ddot{s}(t)$ für die Zeitparametrierung mithilfe eines Polynoms 3. Ordnung [3.1].

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 (3.13) bzw. (3.14) 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 Systeme, bei denen die Geschwindigkeit der Systemeingang ist, aber nicht für Systeme mit einem Beschleunigungs- bzw. Drehmomenteingang. Weiters haben alle polynomialen Ansätze die Eigenschaft, dass die maximale Geschwindigkeit nur zu *einem* Zeitpunkt erreicht wird und sie eine lange Beschleunigungs- und Verzögerungsphase aufweisen. Damit wird die Leistungsfähigkeit des Robotersystems nicht optimal ausgenützt.

3.3.2 Polynom 5. Ordnung

Wird die Zeitparametrierung als Polynom 5. Ordnung in der Form

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3.15a}$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{3.15b}$$

$$\ddot{s}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{3.15c}$$

gewählt, so können im Vergleich zum Polynom 3. Ordnung noch zusätzlich die zweiten Ableitungen am Anfang und am Ende der Roboterbewegung mit $\ddot{s}(0) = \ddot{s}(T) = 0$ vorgegeben werden. D. h. es wird gefordert, dass die zweite Ableitung zu Beginn und am Ende stetig ist. Mit diesen Bedingungen werden die Koeffizienten $a_0, \ldots, a_5$ des Polynoms (3.15a) gemäß

$$a_0 = 0\ , \quad a_1 = 0\ , \quad a_2 = 0\ , \quad a_3 = \frac{10}{T^3}\ , \quad a_4 = -\frac{15}{T^4}, \quad a_5 = \frac{6}{T^5} \tag{3.16}$$

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

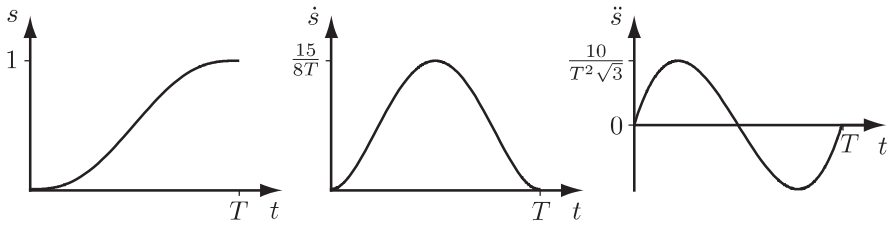


Abbildung 3.3: Schematische Darstellung der Funktionen $s(t)$, $\dot{s}(t)$ und $\ddot{s}(t)$ für die Zeitparametrierung mithilfe eines Polynoms 5. Ordnung [3.1].

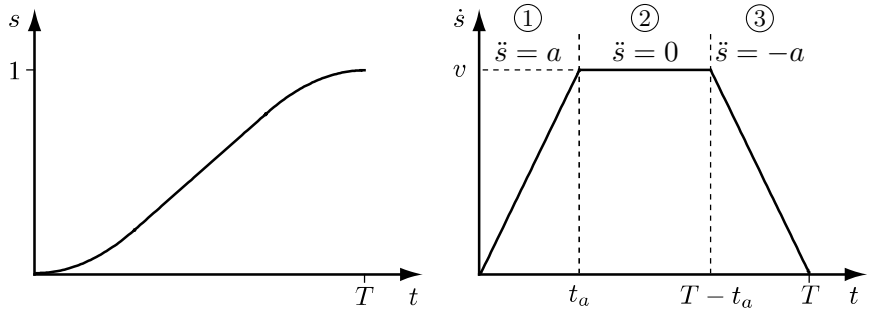


Abbildung 3.4: Zeitparametrierung mit trapezförmigen Bewegungsprofil: ① Beschleunigungsphase, ② Phase konstanter Geschwindigkeit v, ③ Verzögerungsphase [3.1].

3.3.3 Trapezförmige Bewegungsprofile

Bei den trapezförmigen Bewegungsprofilen weist die erste Ableitung der Zeitparametrierung $\dot{s}(t)$ oder die Geschwindigkeit der Trajektorie $\frac{\mathrm{d}}{\mathrm{d}t}x(s(t))$ eines geradlinigen Pfades $x(s)$ 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 mit dem konstanten Beschleunigungsparameter a lautet daher

$$\ddot{s}(t) = \begin{cases} a & 0 \leq t < t_a \\ 0 & t_a \leq t < T - t_a \\ -a & T - t_a \leq t \leq T \\ 0 & \text{sonst} \end{cases} \tag{3.17}$$

und durch einmalige Integration folgt das Geschwindigkeitsprofil zu

$$\dot{s}(t) = \begin{cases} at & 0 \leq t < t_a \\ v & t_a \leq t < T - t_a \\ a(T-t) & T - t_a \leq t \leq T \\ 0 & \text{sonst}\ , \end{cases} \tag{3.18}$$

wobei in (3.17) und (3.18) die Geschwindigkeit $v = at_a$ und die Beschleunigungs- und Verzögerungszeit t_a eingeführt wurde. Nochmalige Integration liefert die Zeitparametrierung

$s(t)$ in der Form

$$s(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2}at^2 & 0 \leq t < t_a \\ vt - \frac{v^2}{2a} & t_a \leq t < T - t_a \\ vT - \frac{v^2}{a} - \frac{a}{2}(T-t)^2 & T - t_a \leq t \leq T \\ vT - \frac{v^2}{a} & \text{sonst .} \end{cases} \tag{3.19}$$

Das trapezförmige Bewegungsprofil bietet den Vorteil, dass die Geschwindigkeits- und Beschleunigungsbeschränkungen des Roboters optimal ausgenützt werden können. Aus (3.17) und (3.18), eingesetzt in (3.10), lassen sich die Parameter v und a aus der maximalen Geschwindigkeit $\dot{x}_{max}$ und der maximalen Beschleunigung $\ddot{x}_{max}$ in der Koordinate x in der Form

$$|v(x_1 - x_0)| \leq \dot{x}_{max} \tag{3.20}$$
$$|a(x_1 - x_0)| \leq \ddot{x}_{max} \tag{3.21}$$

bestimmen.

Die geradlinige Trajektorie (3.10) mit dem trapezförmigen Bewegungsprofil (3.17)–(3.19) wird durch die drei Variablen v, a und T parametriert. Es können aber nur zwei davon unabhängig gewählt werden, denn es wird die Bedingungen $s(T) = 1$ gefordert und die Beschleunigungs- und Verzögerungszeit t_a steht über $t_a = v/a$ im Zusammenhang. Werden beispielsweise die Geschwindigkeit v und die Beschleunigung a vorgegeben – unter der Bedingung $v^2/a \leq 1$, sodass alle drei Phasen des Bewegungsprofils vorhanden sind – so ergibt sich aus der Bedingung $s(T) = 1$ die erforderliche Zeitdauer T für die Trajektorie zu

$$T = \frac{1}{v} + \frac{v}{a} \; . \tag{3.22}$$

Weiters kann auch durch Vorgabe von v und T die erforderliche Beschleunigung a bestimmt werden und wenn a und T vorgegeben wird, kann die zugehörige Geschwindigkeit v berechnet werden.

Das trapezförmige Bewegungsprofil mit drei Phasen weist allerdings den gleichen Nachteil auf wie die Zeitparametrierung mittels eines Polynoms 3. Ordnung, nämlich die Unstetigkeit der Beschleunigung. Um Vibrationen und Schwingungen im Robotersystem zu minimieren, wird meist die Änderungsrate der Beschleunigung, der sogenannte *Ruck* (engl. *jerk*), beschränkt. Analog zur Zeitparametrierung mit einem Polynom 5. Ordnung (siehe Abbildung 3.3) kann auch ein trapezförmiges Bewegungsprofil mit sieben Phasen konstruiert werden, welches diese Aufgabe erfüllt, siehe Abbildung 3.5. Anstatt sprungförmig 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

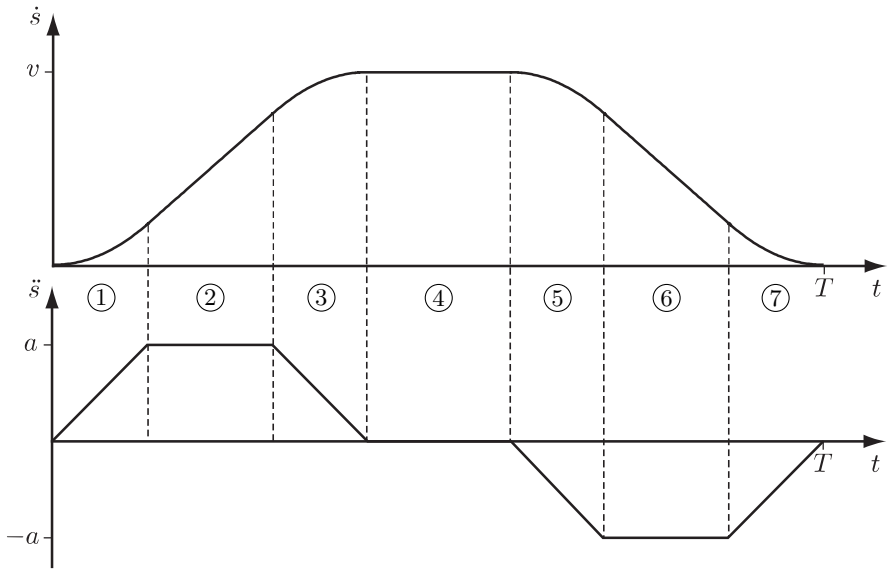


Abbildung 3.5: Zeitparametrierung für trapezförmiges Bewegungsprofil mit 7 Phasen: ① Hochfahren der Beschleunigung, ② konstante Beschleunigung a, ③ herunterfahren der Beschleunigung, ④ konstante Geschwindigkeit v, ⑤ hochfahren der Verzögerung, ⑥ konstante Verzögerung $-a$, ⑦ herunterfahren der Verzögerung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

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

$$x(t_i + \Delta t) = a_{i0} + a_{i1}\Delta t + a_{i2}\Delta t^2 + a_{i3}\Delta t^3 \tag{3.23}$$

angesetzt, wobei Δt 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 $a_{i0}, \ldots, a_{i3}$ der Polynome $i = 0, \ldots, N-1$

gelöst und lauten

$$a_{i0} = x_i \tag{3.24a}$$
$$a_{i1} = \dot{x}_i \tag{3.24b}$$
$$a_{i2} = \frac{3x_{i+1} - 3x_i - 2\dot{x}_i T_i - \dot{x}_{i+1} T_i}{T_i^2} \tag{3.24c}$$
$$a_{i3} = \frac{2x_i + (\dot{x}_i + \dot{x}_{i+1})T_i - 2x_{i+1}}{T_i^3} \; . \tag{3.24d}$$

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 $\dot{x}_i$, $i = 0, \ldots, 3$, gemäß

$$\begin{aligned}
t_0 &= 0\,\mathrm{s} & x(t_0) &= x_0 = 0 & \dot{x}(t_0) &= \dot{x}_0 = 0 \\
t_1 &= 1\,\mathrm{s} & x(t_1) &= x_1 = 0 & \dot{x}(t_1) &= \dot{x}_1 = 1 \\
t_2 &= 2\,\mathrm{s} & x(t_2) &= x_2 = 1 & \dot{x}(t_2) &= \dot{x}_2 = 0 \\
t_3 &= 3\,\mathrm{s} & x(t_3) &= x_3 = 1 & \dot{x}(t_3) &= \dot{x}_3 = 0
\end{aligned}$$

aufweisen. Mit diesen Vorgaben werden die $N - 1 = 3$ Trajektoriensegmente als Polynome 3. Ordnung in der Form (3.23) mit den Koeffizienten (3.24) 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.

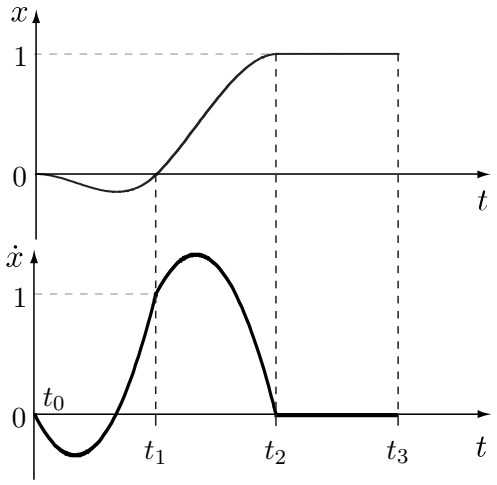


Abbildung 3.6: Zusammengesetzte Trajektorie mit Via-Punkten $x(t)$ mit zugehöriger Geschwindigkeit $\dot{x}(t)$ [3.1].

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 Aufgabenraum – 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 Aufgabenraum 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 Aufgabenraums verwendet werden. So kann ein Werkzeug bei der Oberflächenbearbeitung durch den Roboter mithilfe einer Kraftregelung angedrückt und mit einer Bewegungsregelung entlang der Oberfläche geführt werden.

Dieses Kapitel befasst sich mit der Bewegungsregelung von Robotersystemen für kinematisch nichtredundante Roboter. In den beiden folgenden Abschnitten 4.1 und 4.2 werden zunächst geeignete Beschreibungen für das dynamische Robotermodell im Konfigurationsraum bzw. im Aufgabenraum eingeführt. Anschließend werden häufig verwendete Regler für die Bewegungsregelung 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

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

anschreiben [4.1]. Es wird angenommen, dass der Roboter *vollaktuiert* ist, d.h. es gibt für jeden mechanischen Freiheitsgrad einen unabhängigen Stelleingang $\boldsymbol{\tau}^{\mathrm{T}} = [\tau_1, \tau_2, \ldots, \tau_n]$. 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.

$$\boldsymbol{\tau} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_c + \boldsymbol{\tau}_{ext} \ . \tag{4.2}$$

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 $\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{M}(\mathbf{q})\dot{\mathbf{q}} > 0$, 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. $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{N}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$. Damit gilt $\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0$ für alle $\dot{\mathbf{q}}$ sowie $\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}})$.
- $\mathbf{g}^{\mathrm{T}}(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} V(\mathbf{q})$ 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 τ_j, $j = 1, \ldots, n$ 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

$$\dot{q}_j = \dot{q}_{d,j} = u_j \ , \quad j = 1, \ldots, n \tag{4.3}$$

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*

$$\dot{\mathbf{q}} = \mathbf{u} \ , \tag{4.4}$$

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

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

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*

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

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 Gelenkspositionen $\mathbf{q}$ 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

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

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

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

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. Für einen Stabilitätsbeweis sei auf die Vorlesung „VU Grundlagen der Robotik" verwiesen.

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

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{4.9}$$

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 (4.9) 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 (4.9) in (4.1) ein, so ergibt sich

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{4.10}$$

bzw.

$$\mathbf{M}(\mathbf{q})\Big(\underbrace{\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d(t)}_{\ddot{\mathbf{e}}_q} + \mathbf{K}_{2q}\dot{\mathbf{e}}_q + \mathbf{K}_{1q}\mathbf{e}_q\Big) = \mathbf{0}\ . \tag{4.11}$$

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

$$\ddot{\mathbf{e}}_q + \mathbf{K}_{2q}\dot{\mathbf{e}}_q + \mathbf{K}_{1q}\mathbf{e}_q = \mathbf{0} \tag{4.12}$$

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

$$\ddot{e}_{q,j} + k_{2q,j}\dot{e}_{q,j} + k_{1q,j}e_{q,j} = 0\ , \quad j = 1,\ldots,n \tag{4.13}$$

entkoppelt. Schreibt man (4.13) als System von Differenzialgleichungen erster Ordnung (Zustandsdarstellung) in der Form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} e_{q,j} \\ \dot{e}_{q,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_{1q,j} & -k_{2q,j} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} e_{q,j} \\ \dot{e}_{q,j} \end{bmatrix}\ , \tag{4.14}$$

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 4.1 (Computed-Torque-Regler mit Integralanteil). Man könnte in (4.9) noch einen Integralanteil im Regler hinzufügen, womit das erweiterte Regelgesetz

$$\boldsymbol{\tau} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{M}(\mathbf{q})\left(\ddot{\mathbf{q}}_d(t) - \mathbf{K}_{2q}\dot{\mathbf{e}}_q - \mathbf{K}_{1q}\mathbf{e}_q - \mathbf{K}_{0q}\int \mathbf{e}_q \,\mathrm{d}t\right) . \tag{4.15}$$

lautet. Wählt man auch $\mathbf{K}_{0q} = \mathrm{diag}(k_{0q,1}, \ldots, k_{0q,n})$ als Diagonalmatrix mit positiven Einträgen $k_{0q,j}$, $j = 1, \ldots, n$, 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

$$p_j(s) = s^3 + k_{2q,j}s^2 + k_{1q,j}s + k_{0q,j} \tag{4.16}$$

Hurwitzpolynome sind.

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

$$\boldsymbol{\tau}_c = \boldsymbol{\tau} - \hat{\boldsymbol{\tau}}_d - \hat{\boldsymbol{\tau}}_{ext} \tag{4.17}$$

mit $\boldsymbol{\tau}$ gemäß (4.6), (4.9) oder (4.15).

4.2 Aufgabenraum

4.2.1 Dynamisches Modell im Aufgabenraum

Das dynamische Modell des Roboters (4.1) wurde als Funktion der Koordinaten im Konfigurationsraum $\mathbf{q}$ und $\dot{\mathbf{q}}$ angeschrieben. Eine Bewegungsregelung im Aufgabenraum findet allerdings in den zugehörigen Koordinaten des Aufgabenraums $\mathbf{x}_e$ gemäß (1.2) und $\dot{\mathbf{x}}_e$ gemäß (2.36) statt. Es ist daher vorteilhaft, das dynamische Modell des Roboters (4.1) in den Koordinaten des Aufgabenraums anzuschreiben.

Der Zusammenhang zwischen den Koordinaten im Konfigurationsraum und den Koordinaten im Aufgabenraum sowie deren Ableitungen sind durch (1.2) und (2.36) gemäß

$$\mathbf{x}_e = \mathbf{f}(\mathbf{q}) \tag{4.18a}$$

$$\dot{\mathbf{x}}_e = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} \tag{4.18b}$$

$$\ddot{\mathbf{x}}_e = \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_a(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \tag{4.18c}$$

gegeben. Aus (4.18c) folgt unter der Annahme der Regularität von $\mathbf{J}_a(\mathbf{q})$, d. h. außerhalb der Singularitäten, die Beziehung

$$\ddot{\mathbf{q}} = \mathbf{J}_a^{-1}(\mathbf{q})\Big(\ddot{\mathbf{x}}_e - \dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\Big) \ . \tag{4.19}$$

Setzt man (4.19) in (4.1) ein, so erhält man

$$\mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\ddot{\mathbf{x}}_e + \Big(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\Big)\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{4.20}$$

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

$$\boldsymbol{\Lambda}(\mathbf{x}_e)\ddot{\mathbf{x}}_e + \boldsymbol{\mu}(\mathbf{x}_e,\dot{\mathbf{x}}_e)\dot{\mathbf{x}}_e + \mathbf{F}_g(\mathbf{x}_e) = \mathbf{F} \tag{4.21}$$

mit

$$\boldsymbol{\Lambda}(\mathbf{x}_e) = \mathbf{J}_a^{-\mathrm{T}}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q}) \tag{4.22a}$$

$$\boldsymbol{\mu}(\mathbf{x}_e,\dot{\mathbf{x}}_e) = \mathbf{J}_a^{-\mathrm{T}}(\mathbf{q})\Big(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\dot{\mathbf{J}}_a(\mathbf{q},\dot{\mathbf{q}})\Big)\mathbf{J}_a^{-1}(\mathbf{q}) \tag{4.22b}$$

$$\mathbf{F}_g(\mathbf{x}_e) = \mathbf{J}_a^{-\mathrm{T}}(\mathbf{q})\mathbf{g}(\mathbf{q}) \tag{4.22c}$$

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

$$\mathbf{F} = \mathbf{J}_a^{-\mathrm{T}}(\mathbf{q})\boldsymbol{\tau} \ , \tag{4.23}$$

welche am Endeffektor des Roboters wirkt. Die Beziehung zwischen den Kräften und Momenten im Konfigurationsraum und jenen im Aufgabenraum über die Manipulator Jacobi-Matrix wurde in der Vorlesung *Modellbildung* hergeleitet [4.1]. Man beachte, dass in (4.22) 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 (4.18). Handelt es sich jedoch um einen kinematisch redundanten Roboter, d.h. $n > m$, 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 (4.22) als Funktion von $\mathbf{q}$ und $\dot{\mathbf{q}}$ anstatt von $\mathbf{x}_e$ und $\dot{\mathbf{x}}_e$ ausgedrückt werden.

Die Multiplikation von (4.20) mit $\mathbf{J}_a^{-\mathrm{T}}(\mathbf{q})$ von links wurde ausgeführt, um beim Robotermodell im Aufgabenraum (4.21) eine ähnliche Struktur zu erhalten wie im Konfigurationsraum (4.1). In (4.21) ist $\boldsymbol{\Lambda}(\mathbf{x}_e)$ 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 Aufgabenraum. Es kann gezeigt werden, dass folgende Eigenschaften auch für das Robotermodell (4.21) erhalten bleiben:

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

4.2.2 Computed-Torque-Regler

Der *Computed-Torque-Regler im Aufgabenraum* basiert direkt auf dem mathematischen Modell des Roboters in den Koordinaten des Aufgabenraums (4.21) und lautet

$$\mathbf{F} = \boldsymbol{\Lambda}(\mathbf{x}_e)(\ddot{\mathbf{x}}_{e,d}(t) - \mathbf{K}_{2x}\dot{\mathbf{e}}_x - \mathbf{K}_{1x}\mathbf{e}_x) + \boldsymbol{\mu}(\mathbf{x}_e, \dot{\mathbf{x}}_e)\dot{\mathbf{x}}_e + \mathbf{F}_g(\mathbf{x}_e) \tag{4.24}$$

mit dem Trajektorienfehler im Aufgabenraum $\mathbf{e}_x = \mathbf{x}_e - \mathbf{x}_{e,d}(t)$ 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 (4.24) in die Gleichung des Robotermodells (4.21) zu

$$\underbrace{\ddot{\mathbf{x}}_e - \ddot{\mathbf{x}}_{e,d}}_{\ddot{\mathbf{e}}_x} + \mathbf{K}_{2x}\dot{\mathbf{e}}_x + \mathbf{K}_{1x}\mathbf{e}_x = \mathbf{0} \ . \tag{4.25}$$

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

$$\ddot{e}_{x,j} + k_{2x,j}\dot{e}_{x,j} + k_{1x,j}e_{x,j} = 0 \ , \quad j = 1, \ldots, m \tag{4.26}$$

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 (4.21) und (4.24) das Regelungskonzept mathematisch klar darstellen, ist die Formulierung (4.24) des Computed-Torque-Reglers im Aufgabenraum jedoch ungeeignet für eine Implementierung in der Praxis. Die Gründe dafür sind, dass in den Matrizen (4.22) mehrmals die Inverse der analytischen Jacobi-Matrix $\mathbf{J}_a(\mathbf{q})$ auftritt und die Gleichungen (4.18) nicht eindeutig umgekehrt werden können, siehe dazu Abschnitt 2.2. Weiters liegen meist nur die Gelenkspositionen $\mathbf{q}$ als Messung am Roboter vor, nicht aber die Pose des Endeffektors im Aufgabenraum $\mathbf{x}_e$. Daher kann die Messung der Gelenkspositionen $\mathbf{q}$ direkt im Regelgesetz verwendet werden. Setzt man nun (4.22) und (4.23) in (4.24) ein, so erhält man den Computed-Torque-Regler im Aufgabenraum *ausgedrückt im Konfigurationsraum* in der Form

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{J}_a^{\mathrm{T}}(\mathbf{q})\mathbf{F} \\ &= \mathbf{M}(\mathbf{q})\mathbf{J}_a^{-1}(\mathbf{q})\Big(\ddot{\mathbf{x}}_{e,d}(t) - \mathbf{K}_{2x}\dot{\mathbf{e}}_x - \mathbf{K}_{1x}\mathbf{e}_x - \dot{\mathbf{J}}_a(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\Big) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \end{aligned} \tag{4.27}$$

mit

$$\mathbf{e}_x = \mathbf{f}(\mathbf{q}) - \mathbf{x}_{e,d}(t) \tag{4.28a}$$

$$\dot{\mathbf{e}}_x = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{x}}_{e,d}(t) \ . \tag{4.28b}$$

Man beachte, dass in der Formulierung (4.27) und (4.28) 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 (4.25).

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.

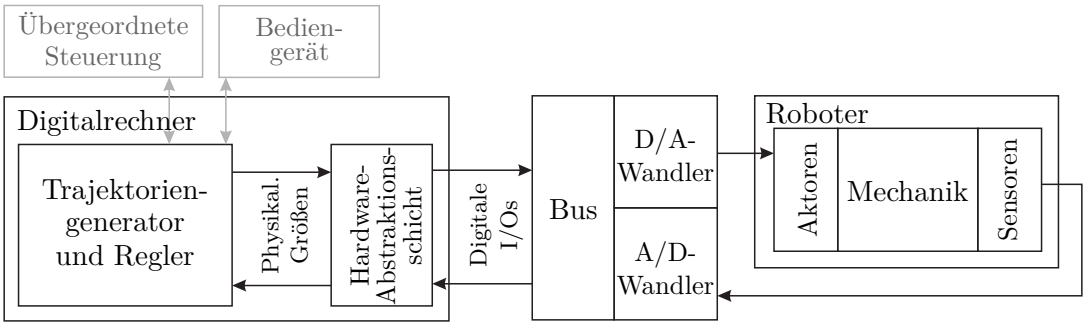


Abbildung 4.1: Blockschaltbild eines digitalen Regelkreises für einen Roboter.

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 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

$$q = 2\pi \frac{\hat{q}}{2^k u} \ . \tag{4.29}$$

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 Gelenksgeschwin-

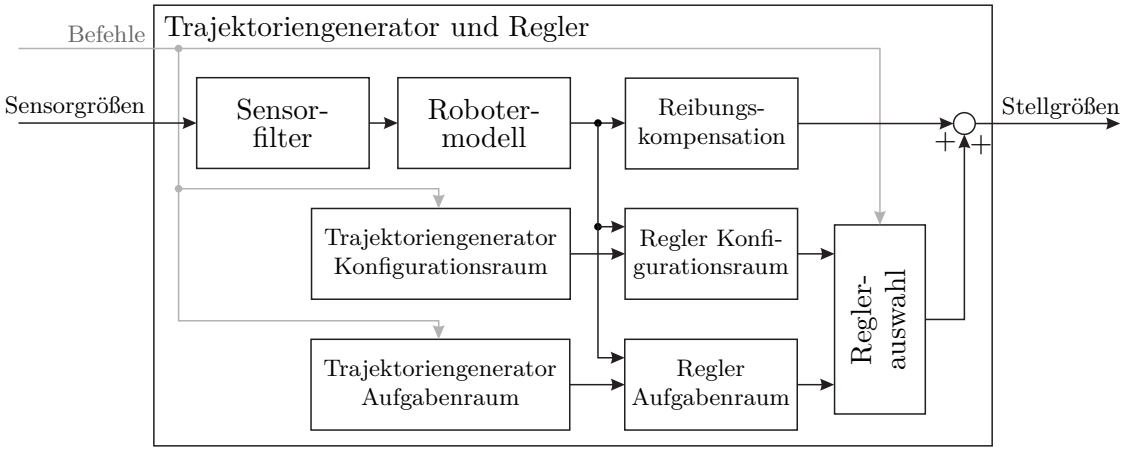


Abbildung 4.2: Blockschaltbild der Implementierung eines Trajektoriengenerators und Reglers für einen Industrieroboter.

digkeit $\dot{\mathbf{q}}$ kann nicht direkt gemessen werden, sondern muss über einen Differenzierfilter aus der Gelenksposition $\mathbf{q}$ 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.

4.4 Literatur

[4.1] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Modellbildung (SS 2022)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2022. Adresse: https://www.acin.tuwien.ac.at/bachelor/modellbildung/.

[4.2] 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.

[4.3] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Berlin Heidelberg: Springer, 2008.

[4.4] B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, London, 2009.