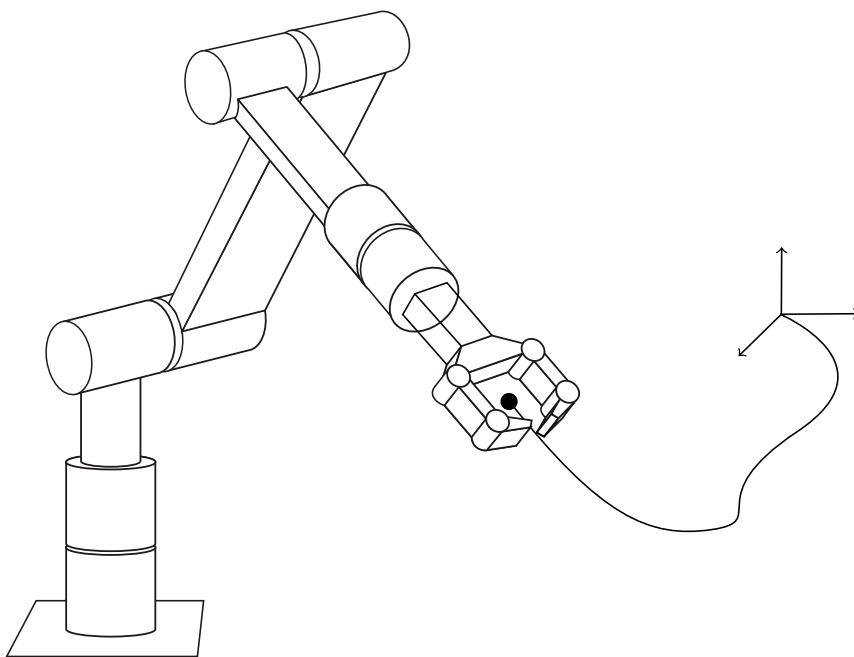


Skript zur Vorlesung

Roboterdynamik



Sommersemester 2017

Prof. dr. ir. Daniel J. Rixen

Changelog:

- 2013-2014 Erste Version Thomas Buschmann, Korrekturen Jörg Baur und Robert Wittmann
- 2015 Ergänzungen und Korrekturen Daniel Rixen und Robert Wittmann
- 2016-2017 Ergänzungen und Korrekturen Daniel Rixen und Felix Sygulla

Inhaltsverzeichnis

1	Einleitung	1
1.1	Literaturhinweise und Links	2
1.1.1	Ausgewählte Hersteller	2
1.1.2	Literaturempfehlungen	3
2	Kinematik	5
2.1	Einleitung	5
2.1.1	Ziele und Aufgaben	5
2.1.2	Modellierung und Annahmen	5
2.2	Koordinatensysteme und Transformationen	6
2.2.1	Koordinatensysteme	6
2.2.2	Konfiguration des Einzelkörpers	6
2.2.3	Koordinatentransformationen	6
2.2.4	Konfiguration des Gesamtsystems	9
2.2.5	DENAVIT-HARTENBERG Konvention	10
2.2.6	Die DENAVIT-HARTENBERG Transformation	11
2.3	Direkte Kinematik	14
2.3.1	Aufgabenstellung	14
2.3.2	Grundlagen Relativkinematik	14
2.3.3	Rekursive Berechnung der direkten Kinematik	16
2.4	Inverse Kinematik	19
2.4.1	Einleitung	19
2.4.2	Numerische Lösung	23
2.4.3	Redundante Roboter	27
3	Dynamik	31
3.1	Einleitung	31
3.2	Grundbegriffe	31
3.2.1	Zwangsbedingungen	31
3.2.2	Virtuelle Verschiebungen	32
3.2.3	Die Prinzipien von D'ALEMBERT und JOURDAIN	33
3.3	Das NEWTON-EULER Verfahren	34
3.3.1	Ausgangspunkt	34
3.3.2	Darstellung als DGL 1. Ordnung	38
3.4	LAGRANGE'sche Gleichungen zweiter Art	39
3.5	Diskussion	40
3.6	Berücksichtigung weiterer Effekte: Antriebsdynamik	40
3.7	Auswertung der Bewegungsgleichungen	42
3.7.1	Direkte Dynamik	42
3.7.2	Inverse Dynamik	42
3.8	Bewegungsgleichungen im Arbeitsraum	43
3.9	Effiziente Berechnung der Dynamik	44
3.9.1	Inverse Dynamik	44

3.9.2	Direkte Dynamik	45
4	Bahnplanung	47
4.1	Einleitung	47
4.1.1	Aufgaben	47
4.1.2	Anmerkungen zur Planung im Arbeitsraum	48
4.1.3	Anmerkungen zur Planung im Gelenkraum	48
4.2	Darstellung der Bahn	48
4.2.1	Kubische Splines	50
5	Optimale Trajektorien	53
5.1	Einleitung	53
5.2	Variationsrechnung	53
5.2.1	Das einfachste Problem (EULER-LAGRANGE Gleichung)	53
5.2.2	Probleme mit Nebenbedingungen	55
5.3	Das PONTRYAGIN Minimum Prinzip	56
5.3.1	Einleitung	56
5.3.2	Problemstellung	56
5.3.3	Optimalitätsbedingungen	56
5.3.4	Allgemeine Lösungsmethode nach dem PMP	59
5.3.5	Beispiel	59
5.4	Numerische Lösung	62
5.4.1	Einleitung	62
5.4.2	Indirekte Verfahren	63
5.4.3	Direkte Verfahren	64
5.5	Reduktion auf einen Bahnparameter	64
6	Regelung	67
6.1	Einleitung	67
6.1.1	Grundbegriffe	67
6.1.2	Aufgaben und Ziele	67
6.1.3	Besonderheiten und Vorgehen	67
6.1.4	Überblick Regelungsstrukturen	68
6.2	Regelung im Gelenkraum	69
6.2.1	Dezentrale, entkoppelte Einzelgelenkregelung	69
6.2.2	Feedback-Entkopplung	72
6.2.3	Feedforward-Entkopplung	73
6.3	Regelung im Arbeitsraum	74
6.3.1	Einleitung	74
6.3.2	Methoden der inversen und transponierten Jacobi-Matrix	75
6.3.3	Feedback-Entkopplung	76
6.3.4	Anmerkungen	76
6.4	Interaktionsregelung	77
6.4.1	Einleitung	77
6.4.2	Hybride Kraft-/Positionsregelung	78
	Akronyme	81
	Symbolverzeichnis	83
	Literatur	85
	Korrekturen zum gedruckten Skript	87

Kapitel 1

Einleitung

Dieses Skript dient als Ergänzung zur Vorlesung „Roboterdynamik“, der begleitenden Übung, den Übungsaufgaben, Folien und Beispielpogrammen. Im Skript werden zentrale Formeln und Zusammenhänge nochmals aufgeführt. Es ersetzt jedoch weder den Besuch der Vorlesung noch den der Übung und deckt den Inhalt des Moduls nicht vollständig ab. **Unter anderem fehlt Teil 1 der Vorlesung fast vollständig (Komponenten und Aufbau von Robotern, etc.)!** Sinn des Skripts ist es, Sie beim Lernen zu unterstützen, indem wesentliche Teile des Stoffs nochmals aufbereitet zur Verfügung gestellt werden. In Bild 1.1 sind wesentliche Teile des Moduls in Form einer kompletten Steuer-/Regelschleife für einen Roboter dargestellt. Dies soll einen Überblick auf die behandelten Themen geben und die Einordnung von Verfahren nach Besuch der Vorlesung erleichtern.

Das Manuskript wurde mit Sorgfalt erstellt. Sollten Sie dennoch Fehler darin finden, bitte ich um eine kurze Nachricht, damit diese behoben werden können. Anregungen und Verbesserungsvorschläge sind natürlich ebenfalls willkommen.

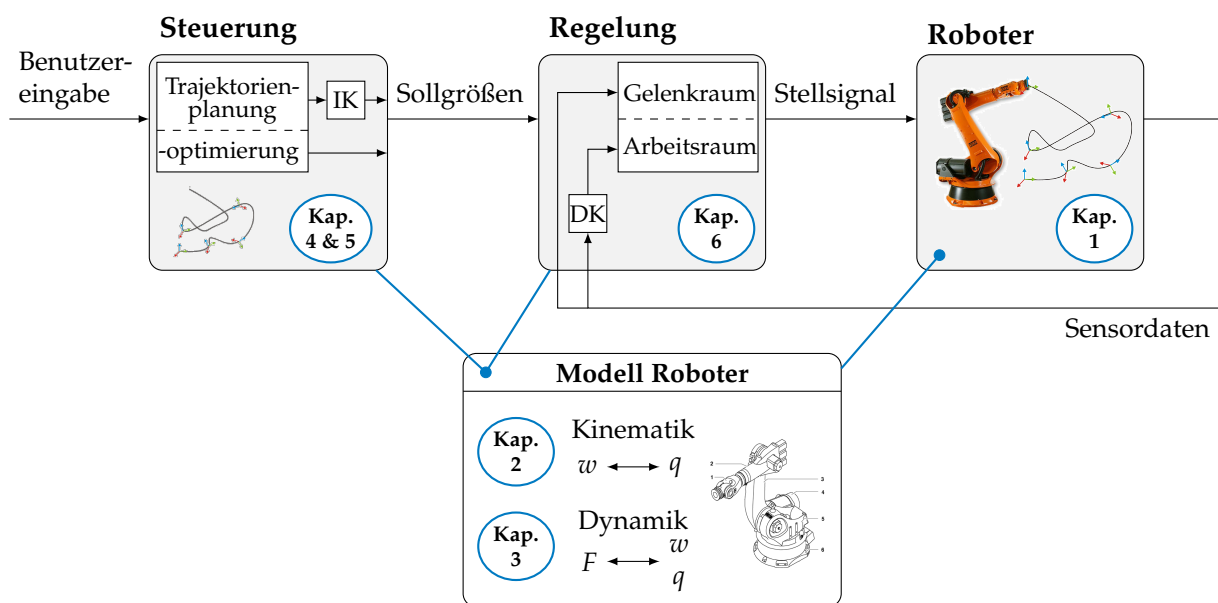


Bild 1.1: Thematische Einordnung der einzelnen Kapitel.

1.1 Literaturhinweise und Links

1.1.1 Ausgewählte Hersteller

Im Folgenden sind die Namen und Webseiten einiger Hersteller von Industrie-Robotern aufgeführt. Auf den Webseiten finden sich zahlreiche Hinweise zu Modellen, Leistungsdaten, sowie viele Applikationsbeispiele.

- **ABB**, Schweiz: <http://www.abb.com/robotics>
- **KUKA**, Augsburg: <http://www.kuka-roboter.de>
- **FANUC**, Japan: <http://www.fanucrobotics.de>
- **ADEPT**, USA: <http://www.adept.com>
- **KAWASAKI ROBOTICS**, USA: <http://www.kawasakirobotics.com>
- **REIS**, Obernburg: <http://www.reisrobotics.de>
- **MOTOMAN**, Allershausen: <http://www.motoman.de>
- **STÄUBLI**, Schweiz: <http://www.staubli.com>
- **MITSUBISHI AUTOMATION**, Japan: <http://www.mitsubishi-automation.com>

1.1.2 Literaturempfehlungen

Hauptlehrbücher

1. **Pfeiffer und Reithmeier**, *Roboterdynamik*. (Pfeiffer und Reithmeier 1987)
Gutes Buch zur Roboterdynamik. Deckt sich in weiten Teilen mit dem Stoff der Vorlesung. Dieses Buch ist aus der ursprünglich von Prof. Pfeiffer gehaltenen Vorlesung hervorgegangen.
2. **Craig**, *Introduction to Robotics, Mechanics & Control*. (Craig 1989)
Gutes Lehrbuch zur Kinematik, Dynamik und Regelung von Robotern.
3. **Siciliano, Sciavicco, Villani und Oriolo**, *Robotics: Modelling, Planning and Control*. (Siciliano, Sciavicco u. a. 2008)
Gutes, neueres Buch zur Kinematik, Dynamik und Regelung. Geht in Teilen über den Stoff der Vorlesung hinaus.

Ergänzend

1. **Pfeiffer**, *Einführung in die Dynamik*. (Pfeiffer 1989)
Gutes Buch zu den Grundlagen der Dynamik (vgl. Grundlagenfach Mechanik/Technische Dynamik)
2. **Bremer**, *Dynamik und Regelung mechanischer Systeme* (Bremer 1988).
Gutes, umfangreiches Buch zur Dynamik und Regelung von Mehrkörpersystemen. Geht in Teilen über den Stoff der Vorlesung hinaus.
3. **Nakamura**, *Advanced Robotics: Redundancy and Optimization*. (Nakamura 1991)
Gutes Buch, v.a. zur Inversen Kinematik und Optimierung zu empfehlen.
4. **Rivin**, *Mechanical Design of Robots*. (Rivin 1988)
Eines der wenigen (guten) Bücher zur mechanischen Konstruktion von Robotern.
5. **Siciliano**, *Handbook of Robotics*. Springer, 2008. (Siciliano und Khatib 2008)
Gutes, umfangreiches Nachschlagewerk und Lehrbuch zu allen Aspekten der Robotik.
6. **Geering**, H. P., *Optimal Control with Engineering Applications*. Springer, 2007. (Geering 2007)
Mathematisch zugängliche Einführung in die Optimalsteuerung.
7. **Featherstone, R.** *Rigid Body Dynamics Algorithms*. Springer, 2008 (Featherstone 2008)
Gutes, sehr vollständiges Buch zur Dynamik starrer Mehrkörpersysteme.

Spezielle Empfehlungen nach Teilen

Teil 1 (Einleitung) Einleitungskapitel der verschiedenen Lehrbücher, insbesondere (Pfeiffer und Reithmeier 1987) und (Siciliano, Sciavicco u. a. 2008).

Teil 2 (Kinematik) Kapitel 1 in (Pfeiffer 1989), Kapitel 2 in (Pfeiffer und Reithmeier 1987), Kapitel 2 und 3 in (Siciliano, Sciavicco u. a. 2008). Kapitel 2, 3 und 4 in (Craig 1989). Speziell zur inversen Kinematik (weiterführend und vertiefend) (Nakamura 1991).

Teil 3 (Dynamik) Grundlagen in Kapitel 1 von (Pfeiffer 1989). Kapitel 3 in (Pfeiffer und Reithmeier 1987). Vertiefend und weitestgehend Kapitel 3 und 4 in (Bremer 1988). Weiterführend auch (Brandl, Johanni und Otter 1986; Featherstone 2008).

Teil 4 (Bahnplanung) Kapitel 4 in (Siciliano, Sciavicco u. a. 2008), Kapitel 7 in (Craig 1989).

Teil 5 (Optimale Bahnen) Kapitel 4 in (Pfeiffer und Reithmeier 1987) (einige Themen in der Vorlesung nicht behandelt!), Kapitel 2 in (Geering 2007). Weiterführend: geometrisches Verfahren für zeitoptimale Bahnen in (Pfeiffer und Johanni 1986).

Teil 6 (Regelung) Kapitel 5 in (Pfeiffer und Reithmeier 1987), Kapitel 9 bis 12 in (Craig 1989), Kapitel 8 und 9 in (Siciliano, Sciavicco u. a. 2008). Speziell zur nichtlinearen Regelung (Feedback-Entkopplung) auch (Slotine und Li 1991).

Kapitel 2

Kinematik

2.1 Einleitung

2.1.1 Ziele und Aufgaben

Die Kinematik ist die Lehre von der Geometrie der Bewegung. Aufgabe der Kinematik ist die eindeutige Beschreibung der geometrischen Konfiguration des Systems in Raum und Zeit. Konkret bedeutet dies für die Roboterdyamik die Beschreibung der Lage und Orientierung aller Komponenten des Systems, sowie der Geschwindigkeiten und Beschleunigungen. Die Ursachen der Bewegung werden in der Kinematik nicht betrachtet.

Diese Größen können auf unterschiedliche Weise beschrieben werden. Zur Beschreibung der Dynamik (siehe Kapitel 3) bietet sich meist die Wahl von Minimalkoordinaten an, die in der Robotik meist mit den Gelenkfreiheitsgraden zusammenfallen (siehe Seite 9). Man spricht von der Darstellung im Konfigurations- oder Gelenkwinkelraum. Zur Beschreibung einer Aufgabe ist dagegen vorwiegend die Lage und Orientierung des Endeffektors relevant. Bei der Beschreibung der Bewegung über den Endeffektor spricht man von einer Darstellung im Arbeitsraum.

Eine wichtige Aufgabe der Kinematik ist auch die Umrechnung zwischen beiden Beschreibungen. Die *direkte Kinematik* ist die Berechnung der Konfiguration des Endeffektors (oder anderer Körper) aus gegebenen Minimalkoordinaten. Die umgekehrte Aufgabe, d.h. die Berechnung der Minimalkoordinaten aus einer gegebenen Endeffektor-Konfiguration, wird als *inverse Kinematik* bezeichnet.

Die direkte Kinematik kann für die hier betrachteten Systeme direkt durch Funktionsauswertungen bestimmt werden: die Arbeitsraumkoordinaten $w \in \mathbb{R}^m$ sind eine Funktion der Minimalkoordinaten $q \in \mathbb{R}^n$. Die Lösung des inversen Problems ist ungleich komplexer, da kein funktioneller Zusammenhang zwischen w und q besteht und darüber hinaus im Allgemeinen die Dimension des Arbeitsraums $\dim(w)$ kleiner derjenigen des Konfigurationsraums $\dim(q)$ ist, sodass keine eindeutige Lösung existiert.

2.1.2 Modellierung und Annahmen

Es wird vereinfachend von idealen Gelenken ausgegangen, die spielfrei, starr und geometrisch ideal sind. Darüber hinaus werden nur starre Körper betrachtet. Für eine Verallgemeinerung auf elastische Systeme sei auf die Literatur verwiesen (z.B. Bremer 1988, 2008; Bremer und Pfeiffer 1992; Shabana 2005).

Die einzelnen Körper sind über Gelenke zu einem Mehrkörpersystem (MKS) verbunden. Es werden nur baumstrukturierte Systeme betrachtet. In Robotern finden sich typischerweise Schub- und Drehgelenke, die jeweils genau eine translatorische bzw. rotatorische Bewegung ermöglichen.

Definition 2.1.1. Baumstruktur In einem baumstrukturierten MKS hat jeder Körper genau einen (oder keinen) Vorgänger und beliebig viele Nachfolger. Genau ein Körper (der Wurzelkörper) hat keinen Vorgänger.

Ein wichtiger Sonderfall der Baumstruktur ist die Kette.

Definition 2.1.2. Kettenstruktur In einem MKS mit Kettenstruktur hat jeder Körper genau einen (oder keinen) Vorgänger und einen oder keinen Nachfolger. Genau ein Körper (der Wurzelkörper) hat keinen Vorgänger.

Klassische Knickarmroboter mit sechs Freiheitsgraden (FhG) haben eine Kettenstruktur.

2.2 Koordinatensysteme und Transformationen

2.2.1 Koordinatensysteme

Es werden zur Berechnung der Kinematik und Dynamik ein inertial festes Koordinatensystem (KOS, Basis B_0), sowie für jeden Körper i ein körperfestes KOS (Basis B_i) eingeführt. Zur eindeutigen Kennzeichnung wird das Bezugssystem eines Vektors oder einer Matrix als tiefgestellter Index vorangestellt. Der Ortsvektor des Punktes P , angeschrieben im KOS B_i wird damit als ${}_i\mathbf{r}_P$ geschrieben. Wird in einer Gleichung das Bezugssystem weggelassen, gilt die Beziehung für ein beliebiges KOS.

2.2.2 Konfiguration des Einzelkörpers

Die geometrische Konfiguration eines einzelnen Starrkörpers kann durch die Position \mathbf{r}_i des körperfesten Systems B_i und die Orientierung von B_i relativ zur inertialen Referenz B_0 eindeutig beschrieben werden. Drei Parameter sind zur Darstellung der Lage des Referenzpunktes notwendig (\mathbf{r}_i), drei für die Orientierung des Koordinatensystems B_i gegenüber B_0 .

Notation. Die relative Position eines Punktes P zu einem Punkt Q wird im Regelfall als \mathbf{r}_{QP} geschrieben. Im Sonderfall der Darstellung eines Punktes bezüglich der inertialen Basis B_0 kann der Bezugspunkt 0 entfallen, d.h. $\mathbf{r}_P \equiv \mathbf{r}_{0P}$.

Die relative Orientierung wird durch die Rotationsmatrix ${}_i\mathbf{A}_0 \in \mathbb{R}^{3 \times 3}$ beschrieben, die Größen von einer Darstellung im B_0 in eine Darstellung im B_i -System transformiert:

$${}_i\mathbf{r}_P = {}_i\mathbf{A}_0 {}_0\mathbf{r}_P \quad (2.2.1)$$

Möglichkeiten zur Parametrierung einer Drehmatrix ${}_i\mathbf{A}_k$ sind im folgenden Abschnitt angegeben.

Zur eindeutigen Beschreibung der Konfiguration eines starren Körpers sind somit insgesamt sechs Parameter erforderlich.

2.2.3 Koordinatentransformationen

Elementardrehungen

Geht eine Basis durch Rotation um eine der drei Achsen x , y , z aus einer anderen hervor, spricht man von einer Elementardrehung. Die entsprechenden Drehmatrizen werden mit \mathbf{A}_x , \mathbf{A}_y , \mathbf{A}_z bezeichnet:

$$\mathbf{A}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (2.2.2)$$

$$\mathbf{A}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \quad (2.2.3)$$

$$\mathbf{A}_z(\gamma) = \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2.4)$$

Die Drehmatrizen können leicht durch folgende Überlegung hergeleitet bzw. verifiziert werden: die i -te Koordinatenachse des Ausgangssystems entspricht im gedrehten System der i -ten Spalte der Drehmatrix. Beispielhaft für A_x und y :

$${}_i y_k = A_{xk} y_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix} \quad (2.2.5)$$

Räumliche Drehungen

Eine allgemeine Drehung im Raum kann durch drei hintereinander geschaltete Elementardrehungen beschrieben werden. Im Folgenden werden ausschließlich orthonormale Basen verwendet, d.h. die Einheitsvektoren $e_i, i = 1, 2, 3$ der Koordinatensysteme haben die Länge „1“ und sind paarweise orthogonal:

$$\|e_i\| = 1, \quad i = 1, 2, 3 \quad (2.2.6)$$

$$e_i^T e_j = 0, \quad i, j = 1, 2, 3 \quad \text{und} \quad i \neq j \quad (2.2.7)$$

Da die Einheitsvektoren die Spalten der Drehmatrix bilden, folgt unmittelbar

$${}_j A_i^T {}_j A_i = E \quad (2.2.8)$$

$$\Rightarrow {}_i A_j^{-1} = {}_i A_j^T \quad (2.2.9)$$

Die Inverse der Drehmatrix entspricht also der Transponierten, was in Kinematik-Berechnungen oft ausgenutzt werden kann.

EULER-Winkel

Durch drei hintereinander geschaltete Drehungen um die z , x und z Achsen ergibt sich die EULER'sche Drehmatrix. Die Drehungen erfolgen dabei jeweils um die bereits gedrehten Richtungsvektoren (siehe Bild 2.1).

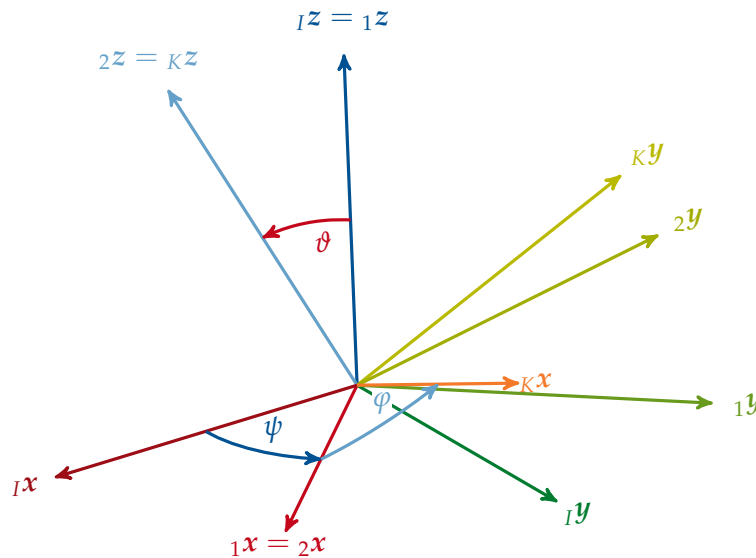


Bild 2.1: EULER-Winkel zur Darstellung räumlicher Drehungen (ψ um ${}_I z$, mit ϑ um ${}_1 x$ und mit φ um ${}_K z$).

Die Vorschrift zur Bestimmung der Drehmatrix lautet beziehungsweise auf Bild 2.1:

1. Drehung mit ψ um ${}_I z$ Achse (entspricht ${}_1 z$)
2. Drehung mit ϑ um ${}_1 x$ Achse (entspricht ${}_2 x$)
3. Drehung mit φ um ${}_K z$ Achse (entspricht ${}_2 z$)

Damit ergibt sich für die Gesamtdrehung zu:

$${}_K A_I = A_z(\varphi) A_x(\vartheta) A_z(\psi) \quad (2.2.10)$$

$$= \begin{pmatrix} c\psi c\varphi - s\psi c\vartheta s\varphi & s\psi c\varphi + c\psi c\vartheta s\varphi & s\vartheta s\varphi \\ -c\psi s\varphi - s\psi s\vartheta c\varphi & -s\psi s\varphi + c\psi s\vartheta c\varphi & s\vartheta c\varphi \\ s\psi s\vartheta & -c\psi s\vartheta & c\vartheta \end{pmatrix} \quad (2.2.11)$$

Dabei wurden die Abkürzungen $\cos(x) = c x$, $\sin(x) = s x$ verwendet. Wenn die zweite Drehung um die (gedrehte) x -Achse verschwindet, vereinfacht sich die Drehmatrix zu:

$${}_K A_I|_{\vartheta=0} = \begin{pmatrix} c\psi c\varphi - s\psi s\varphi & s\psi c\varphi + c\psi s\varphi & 0 \\ -c\psi s\varphi - s\psi c\varphi & -s\psi s\varphi + c\psi c\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2.12)$$

$$= \begin{pmatrix} \cos(\psi + \varphi) & \sin(\psi + \varphi) & 0 \\ -\sin(\psi + \varphi) & \cos(\psi + \varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2.13)$$

Es findet nur eine Elementardrehung um z mit dem Winkel $\psi + \varphi$ statt. Dies bedeutet insbesondere, dass aus der Drehmatrix nicht eindeutig auf die Parameter ψ, ϑ, φ geschlossen werden kann. Die Bedeutung der Singularität wird auch durch Betrachtung der Winkelgeschwindigkeit als Funktion der EULER-Winkel deutlich.

Da die Winkelgeschwindigkeit ein Vektor ist, kann sie als vektorielle Summe der einzelnen Anteile berechnet werden:

$$\omega = \dot{\psi} + \dot{\vartheta} + \dot{\varphi} \quad (2.2.14)$$

Bei Auswertung im körperfesten Koordinatensystem ergibt sich:

$${}_K \omega = A_z(\varphi) A_x(\vartheta) \dot{\psi} z + A_z(\varphi) \dot{\vartheta} x + \dot{\varphi} z \quad (2.2.15)$$

$$= \begin{pmatrix} z & A_z(\varphi)x & A_z(\varphi)A_x(\vartheta)z \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{pmatrix} \quad (2.2.16)$$

$$= \begin{pmatrix} 0 & c\varphi & s\vartheta s\varphi \\ 0 & -s\varphi & s\vartheta c\varphi \\ 1 & 0 & c\vartheta \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{pmatrix} \quad (2.2.17)$$

Die Reihenfolge, in der die Winkelgeschwindigkeiten addiert werden ist gleichgültig – die Winkelgeschwindigkeit ist ein Vektor! Die Reihenfolge endlicher räumlicher Drehungen kann dagegen im Allgemeinen nicht vertauscht werden, ohne das Ergebnis zu ändern.

Die Winkelgeschwindigkeit im I -System kann durch linksseitige Multiplikation mit der Drehmatrix berechnet werden:

$${}_I \omega = \begin{pmatrix} s\psi s\vartheta & c\psi & 0 \\ -c\psi s\vartheta & s\psi & 0 \\ c\vartheta & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{pmatrix} \quad (2.2.18)$$

Diese Abbildung kann invertiert werden, um die zeitliche Ableitung der EULER-Parameter in Abhängigkeit von der Winkelgeschwindigkeit und den momentanen Parametern zu berechnen:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{s\vartheta} \begin{pmatrix} -s\varphi c\vartheta & -c\varphi c\vartheta & s\vartheta \\ s\vartheta c\varphi & -s\vartheta s\varphi & 0 \\ s\varphi & c\varphi & 0 \end{pmatrix} {}_K \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (2.2.19)$$

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{s\vartheta} \begin{pmatrix} s\psi & -c\psi & 0 \\ c\psi s\vartheta & s\psi s\vartheta & 0 \\ -s\psi c\vartheta & c\psi c\vartheta & s\vartheta \end{pmatrix} {}_I \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (2.2.20)$$

In der Nähe der singulären Lage $\vartheta = 0$ wird die Änderungsrate der Parameter aufgrund der Division durch ϑ sehr groß, in der Singularität sind sie nicht definiert.

In konkreten Aufgabenstellungen sollte diese Singularität stets vermieden werden. Dies kann durch eine andere Parametrierung geschehen, die in der entsprechenden Lage keine Singularität aufweist, oder durch redundante, singularitätsfreie Parametrierungen (Achse-Winkel, Euler-Parameter, Quaternionen, etc. siehe etwa Siciliano, Sciavicco u. a. 2008).

KARDAN-Winkel

Eine Alternative zur Parametrierung der Drehmatrix sind die KARDAN-Winkel (siehe Bild 2.2), die durch folgende Vorschrift definiert sind:

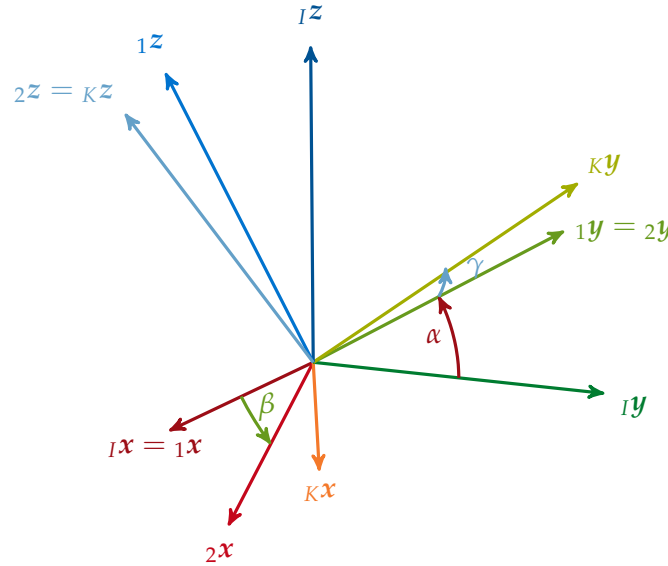


Bild 2.2: KARDAN-Winkel zur Darstellung räumlicher Drehungen (mit α um ${}_1x$, mit β um ${}_1y$ und mit γ um ${}_Kz$).

1. Drehung mit α um ${}_1x$ Achse (entspricht ${}_1x$)
2. Drehung mit β um ${}_1y$ Achse (entspricht ${}_2y$)
3. Drehung mit γ um ${}_Kz$ Achse (entspricht ${}_3z$)

Die Drehmatrix ergibt sich somit zu:

$${}_KA_I = A_z(\gamma)A_y(\beta)A_x(\alpha) \quad (2.2.21)$$

Für die ausmultiplizierte Matrix sei auf die Literatur verwiesen (etwa Pfeiffer und Reithmeier 1987).

2.2.4 Konfiguration des Gesamtsystems

Die momentane geometrische Konfiguration eines MKS mit N Körpern kann im einfachsten Fall durch Angabe der $6N$ Parameter der einzelnen Körper angegeben werden. Man spricht von einer Darstellung in Systemkoordinaten $\mathbf{Z} \in \mathbb{R}^{6N}$. Da die einzelnen Körper jedoch über Gelenke gekoppelt sind, ist die Darstellung im allgemeinen redundant, d.h. die Konfiguration ist bereits mit einer Teilmenge der \mathbf{Z} eindeutig beschrieben. Darüber hinaus sind nicht alle \mathbf{Z} physikalisch möglich, da die Bindungen (Lager, Gelenke, etc., siehe Abschnitt 3.2.1) bei freier Wahl der \mathbf{Z} nicht berücksichtigt werden und somit auch verletzt werden können.

Günstiger ist in der Robotik in aller Regel die Darstellung in Minimalkoordinaten:

Definition 2.2.1. Minimalkoordinaten sind ein minimaler Satz an Positionsparametern, die (gemeinsam mit festen Parametern), die Konfiguration des Systems eindeutig beschreiben.

Im Falle baumstrukturierter Systeme ist die Festlegung der Minimalkoordinaten sehr einfach, da die relativen Freiheitsgrade zwischen den Einzelkörpern gewählt werden können. In der Robotik sind dies meist die Gelenkwinkel bzw. Verschiebungen im Falle prismatischer (translatorischer) Gelenke. Diese Wahl der Minimalkoordinaten ist auch deshalb sinnvoll, da sie meist im Roboter selbst direkt gemessen werden können.

2.2.5 DENAVIT-HARTENBERG Konvention

Die DENAVIT-HARTENBERG Konvention (DH-Konvention) ist eine in der Robotik weit verbreitete Möglichkeit zur Parametrierung der relativen Lage und Orientierung zweier Körper. Die DH-Konvention erfasst mit Dreh- und Schubgelenken die in der Robotik häufigsten Gelenktypen. Ein Vorteil der Konvention ist die Tatsache, dass zur eindeutigen Beschreibung der Kinematik nur vier Parameter erforderlich sind. Im Allgemeinen sind zur eindeutigen Beschreibung der relativen Lage und Orientierung zweier Koordinatensysteme sechs Parameter erforderlich. Die geringere Zahl erforderlicher Parameter wird dadurch ermöglicht, dass nur Schub- und Drehgelenke darstellbar sind und auf die freie Wahl der Position und Orientierung der Koordinatensysteme relativ zu den bewegten Körpern verzichtet wird. Im unten folgenden Beispiel (Seite 12 ff.) wird dies etwa darin deutlich, dass das inertielle Koordinatensystem B_0 nicht in die physikalische Basis des Roboters gelegt werden kann (vgl. Bild 2.4). Der wesentliche Vorteil liegt jedoch in der weiten Verbreitung, sodass der Informationsaustausch mit Hilfe von DH-Parametern erleichtert wird. Zu beachten ist jedoch, dass in der Literatur leichte Variationen der DH-Konvention zu finden sind. Im Rahmen dieser Vorlesung wird die Konvention nach (Craig 1989) verwendet.

In der folgenden Darstellung wird aus Gründen der Einfachheit von einer kinematischen Kette ausgegangen. Eine Verallgemeinerung auf Baumstrukturen ist leicht möglich, erfordert jedoch eine geringfügige Modifikation der Indizierung der DH-Parameter.

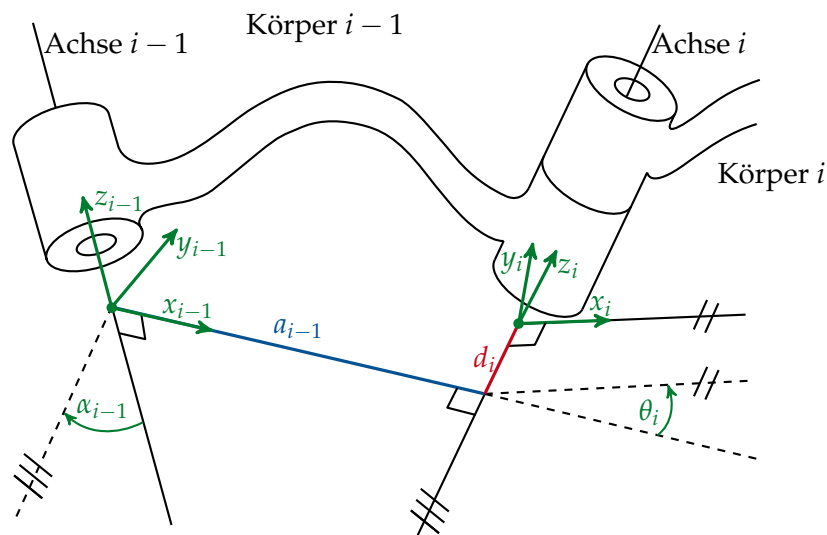


Bild 2.3: Die DENAVIT-HARTENBERG Parameter (Abbildung nach Craig 1989)

Die DH-Konvention gibt die Lage der körperfesten Koordinatensysteme vor, sowie die zur Beschreibung der relativen Lage verwendeten Parameter (siehe Bild 2.3).

Die DH-Parameter sind wie folgt definiert:

- a_{i-1} : Abstand zwischen der z_{i-1} und z_i -Achse entlang der Achse x_{i-1}
- α_{i-1} : Drehwinkel zwischen der z_{i-1} und z_i -Achse um die Achse x_{i-1}
- d_i : Abstand zwischen der x_{i-1} und x_i -Achse entlang der Achse z_i

- θ_i : Drehwinkel zwischen der x_{i-1} und x_i -Achse um die Achse z_i

Die Festlegung der Koordinatensysteme und Bestimmung der Parameter für eine gegebene Kinematik kann nach folgendem Schema durchgeführt werden:

1. Nummerierung der Körper 1 bis N . Körper 0 ist Vorgänger von 1 und fix mit der Umgebung verbunden.
2. Bestimmen der Dreh- oder Schubachsen und Nummerierung der Achsen von 1 bis N
Im Folgenden werden die aufeinanderfolgenden Achsen i und $i + 1$ betrachtet
3. Die Lage des Ursprungs B_i ist definiert durch den:
 - Schnittpunkt der Achse i mit der senkrechten Verbindungslinie der Achsen oder
 - Schnittpunkt der Achsen.
4. Die Koordinatenachse z_i ist identisch mit der Bewegungsachse i . Die Richtung der Koordinatenachse ist durch die positive Dreh- oder Verschiebungsrichtung des Freiheitsgrads vorgegeben.
5. Für die Wahl der Koordinatenachse x_i gilt:
 - entlang der senkrechten Verbindungslinie der Achsen. x_i weist von z_i nach z_{i+1} .
 - bei schneidende Achsen: senkrecht zu beiden Achsen.
6. Die Koordinatenachse y_i ergibt sich aus x_i und z_i (Rechtssystem).
7. Festlegen der Systeme B_0 und B_N :
 - B_0 : identisch zu B_1 wenn der Freiheitsgrad q_1 (θ_1 oder d_1 , je nach Gelenktyp) verschwindet.
 - B_N : Ursprung und x_N -Achse so wählen, dass möglichst viele DH-Parameter zu Null werden.

Falls die Wahl der Koordinatensysteme nicht eindeutig ist, sollten möglichst viele DH-Parameter zu Null gewählt werden.

2.2.6 Die DENAVIT-HARTENBERG Transformation

Die relative Verschiebung und Verdrehung zweier Koordinatensysteme kann kompakt durch eine homogene Transformation beschrieben werden, bei der die relative Drehung und Position einer Basis i relativ zu einer Basis j in einer Matrix ${}^jD_i \in \mathbb{R}^{4 \times 4}$ zusammengefasst wird.

Die Position eines Punktes P relativ zu der Basis B_i kann durch den Ortsvektor ${}^i\mathbf{r}_{iP}$ dargestellt werden. Alternativ kann die Lage des Punktes auch durch die relative Lage bezüglich der Basis B_j beschrieben werden: ${}^j\mathbf{r}_{jP}$. Eine Umrechnung zwischen beiden Darstellungen ist durch die Vektorkette B_j, B_i, P , sowie eine Koordinatentransformation möglich:

$${}^j\mathbf{r}_{jP} = {}^j\mathbf{r}_{ji} + {}^j\mathbf{A}_i {}^i\mathbf{r}_{iP} \quad (2.2.22)$$

Dies kann äquivalent und kompakt mit Hilfe der homogenen Transformation jD_i geschrieben werden:

$$\underbrace{\begin{pmatrix} {}^j\mathbf{r}_{jP} \\ 1 \end{pmatrix}}_{{}^j\mathbf{z}_p} = \underbrace{\begin{pmatrix} {}^j\mathbf{A}_i & {}^j\mathbf{r}_{ji} \\ \mathbf{0} & 1 \end{pmatrix}}_{{}^jD_i} \underbrace{\begin{pmatrix} {}^i\mathbf{r}_{iP} \\ 1 \end{pmatrix}}_{{}^i\mathbf{z}_p} \quad (2.2.23)$$

Die um eine „1“ erweiterten Koordinaten \mathbf{z}_p werden als homogene Koordinaten bezeichnet.

Die homogene Transformation ${}_{i-1}D_i$ von B_i nach B_{i-1} ist durch die DH-Parameter eindeutig beschrieben. In diesem Zusammenhang spricht man auch von der DH-Transformation. Ähnlich

zum Vorgehen bei räumlichen Drehungen, kann auch die DH-Transformation durch vier elementare homogenen Transformationen zusammengesetzt werden, d.h. durch reine Drehungen oder Verschiebungen entlang einer der Koordinatenachsen:

1. Rotation $-\theta_i$ um z_i : $D_1 = \begin{pmatrix} A_z(-\theta_i) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}$
2. Translation $-d_i$ entlang z_i : $D_2 = \begin{pmatrix} E & e_z d_i \\ \mathbf{0} & 1 \end{pmatrix}$
3. Translation $-a_{i-1}$ entlang x_{i-1} : $D_3 = \begin{pmatrix} E & e_x a_{i-1} \\ \mathbf{0} & 1 \end{pmatrix}$
4. Rotation $-\alpha_{i-1}$ um x_{i-1} : $D_4 = \begin{pmatrix} A_x(-\alpha_{i-1}) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}$

Somit ergibt sich die resultierende homogene Transformation zu:

$$\begin{aligned}
 {}_{i-1}D_i &= D_4 D_3 D_2 D_1 \\
 &= \begin{pmatrix} A_x^T(\alpha_{i-1}) A_z^T(\theta_i) & A_x^T(\alpha_{i-1})(a_{i-1}e_x + d_i e_z) \\ \mathbf{0} & 1 \end{pmatrix} \\
 &= \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.2.24}
 \end{aligned}$$

Die Transformation zwischen mehreren Basen kann einfach durch Multiplikation der relativen homogenen Transformationen durchgeführt werden. Für eine kinematische Kette mit n Gliedern berechnen sich die homogenen Koordinaten des Punktes P bzgl. der Basis B_0 (${}_0z_p$) aus denen bzgl. der Basis B_n (${}_nz_p$) durch:

$${}_0z_p = {}_0D_n {}_nz_p = \left(\prod_{i=1}^n {}_{i-1}D_i \right) {}_nz_p \tag{2.2.25}$$

Für einen Roboter mit Kettenstruktur kann somit aus den Koordinaten des Endeffektors relativ zur Basis des letzten Körpers (${}_nz_p$) leicht die Position relativ zum Inertialsystem B_0 berechnet werden.

Beispiel DH-Transformation

Betrachtet wird ein Manipulator mit drei Drehgelenken (RRR), siehe Bild 2.4. Die DH-Parameter ergeben sich nach obigen Regeln zu:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	$\theta_1 = q_1$
2	$-\frac{\pi}{2}$	0	0	$\theta_2 = q_2$
3	0	$a_2 = l$	0	$\theta_3 = q_3$

Die DH-Transformation vom letzten Körper zum Inertialsystem berechnet sich aus den Teiltransformationen gemäß:

$${}_0D_3 = {}_0D_1 {}_1D_2 {}_2D_3. \tag{2.2.26}$$

Mit ${}_{i-1}D_i$ aus (2.2.24) folgt für die einzelnen Matrizen:

$${}_0D_1 = \begin{pmatrix} E A_z^T(q_1) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} c q_1 & -s q_1 & 0 & 0 \\ s q_1 & c q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2.27}$$

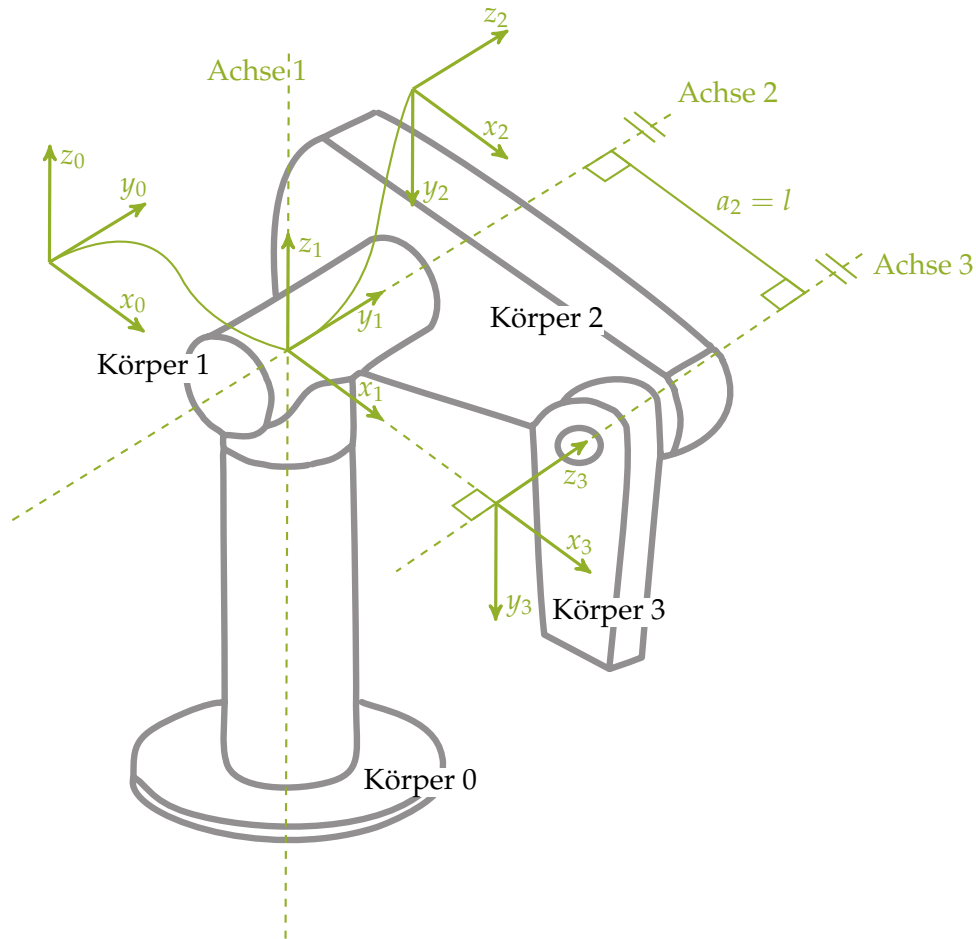


Bild 2.4: Knickarm-Roboter mit drei Drehgelenken als Beispiel zu DH-Parametern und DH-Transformation.

$${}_1D_2 = \begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} & \begin{pmatrix} c q_2 & -s q_2 & 0 \\ s q_2 & c q_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} c q_2 & -s q_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s q_2 & -c q_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2.28)$$

$${}_2D_3 = \begin{pmatrix} c q_3 & -s q_3 & 0 & l \\ s q_3 & c q_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2.29)$$

Die Teil-Transformation ${}_1D_3$ berechnet sich damit zu:

$${}_1D_3 = {}_1D_2 {}_2D_3 = \begin{pmatrix} c q_2 c q_3 - s q_2 s q_3 & -c q_2 s q_3 - s q_2 c q_3 & 0 & c q_2 l \\ 0 & 0 & 1 & 0 \\ -c q_2 s q_3 - s q_2 c q_3 & -c q_2 c q_3 + s q_2 s q_3 & 0 & -s q_2 l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2.30)$$

Mit den Additionstheoremen

$$\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \quad (2.2.31)$$

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta \quad (2.2.32)$$

vereinfacht sich das Ergebnis zu:

$${}_1D_3 = \begin{pmatrix} c q_{23} & -s q_{23} & 0 & c q_2 l \\ 0 & 0 & 1 & 0 \\ -s q_{23} & -c q_{23} & 0 & -s q_2 l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2.33)$$

Das heißt, die Winkel q_2 und q_3 addieren sich in den Argumenten der Sinus- und Kosinusfunktionen. Da die beiden Achsen parallel sind, ist das Ergebnis anschaulich plausibel. Diese Vereinfachung kann bei parallelen Achsen oft ausgenutzt werden.

Das endgültige Ergebnis ergibt sich durch Multiplikation mit 0D_1 :

$${}^0D_3 = {}^0D_1 {}^1D_3 = \begin{pmatrix} c q_1 c q_{23} & -c q_1 s q_{23} & -s q_1 & c q_1 c q_2 l \\ s q_1 c q_{23} & -s q_1 s q_{23} & c q_1 & s q_1 c q_2 l \\ -s q_{23} & -c q_{23} & 0 & -s q_2 l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2.34)$$

$$= \begin{pmatrix} {}^0A_3 & {}^0r_{03} \\ \mathbf{0} & 1 \end{pmatrix} \quad (2.2.35)$$

Mit der homogenen Transformationsmatrix 0D_3 kann direkt die Position des Endeffektors (Tool Center Point, TCP) relativ zum Inertialsystem berechnet werden. Ist die Position des TCPs im körperfesten Koordinatensystem B_3 gegeben

$${}^3r_{3,TCP} = \begin{pmatrix} 0 \\ l \\ 0 \end{pmatrix}, \quad (2.2.36)$$

folgt die Position relativ zu B_0 zu:

$$\begin{pmatrix} {}^0r_{0,TCP} \\ 1 \end{pmatrix} = {}^0D_3 \begin{pmatrix} {}^3r_{3,TCP} \\ 1 \end{pmatrix} = \begin{pmatrix} l c q_1 (-s q_{23} + c q_2) \\ l s q_1 (-s q_{23} + c q_2) \\ -l (c q_{23} + s q_2) \\ 1 \end{pmatrix} \quad (2.2.37)$$

2.3 Direkte Kinematik

2.3.1 Aufgabenstellung

Aufgabe der direkten Kinematik ist die Berechnung der kinematischen Größen der einzelnen Körper aus den gegebenen Minimalkoordinaten q , Minimalgeschwindigkeiten \dot{q} und Minimalbeschleunigungen \ddot{q} . Zu berechnen sind also:

- Positionen und Orientierungen: $r_{i,0}A_i$
- Geschwindigkeiten: ω_i, v_{0i}
- Beschleunigungen: $\dot{\omega}_i, a_{0i}$

2.3.2 Grundlagen Relativkinematik

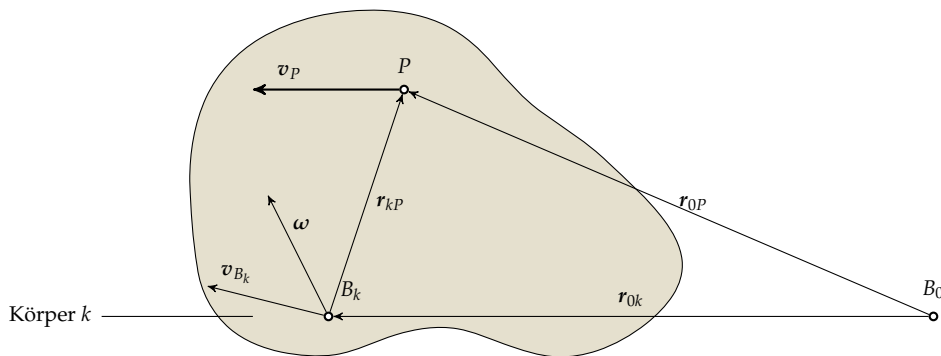


Bild 2.5: Relativkinematik für einen Starrkörper.

Relativkinematik – Position In koordinatenfreier Darstellung ergibt sich die Lage des Punktes P auf dem Körper K abhängig von derjenigen eines körperfesten Systems B_k zu (siehe Bild 2.5):

$$\mathbf{r}_P = \mathbf{r}_k + \mathbf{r}_{kP} \quad (2.3.1)$$

Im Inertialsystem B_0 dargestellt entspricht dies der Gleichung:

$${}_0\mathbf{r}_P = {}_0\mathbf{r}_k + {}_0\mathbf{A}_k {}_k\mathbf{r}_{kP} \quad (2.3.2)$$

Relativkinematik – Geschwindigkeit Zur Berechnung der Dynamik werden die *absoluten* Geschwindigkeiten der Teilkörper benötigt. Die Absolutgeschwindigkeit ist die zeitliche Änderung der Position *relativ zu einem Inertialsystem*.

Formal ergibt sich die Geschwindigkeit durch Ableitung im inertialen B_0 -System, d.h. Differenzierung von (2.3.2):

$${}_0\mathbf{v}_P := {}_0\dot{\mathbf{r}}_P = {}_0\mathbf{v}_k + {}_0\dot{\mathbf{A}}_k {}_k\mathbf{r}_{kP} + {}_0\mathbf{A}_k {}_k\dot{\mathbf{r}}_{kP} \quad (2.3.3)$$

Die Absolutgeschwindigkeit im körperfesten System wird durch Multiplikation mit der Transformationsmatrix berechnet:

$${}_k\mathbf{v}_P = {}_k\mathbf{v}_k + {}_k\mathbf{A}_0 {}_0\dot{\mathbf{A}}_k {}_k\mathbf{r}_{kP} + {}_k\dot{\mathbf{r}}_{kP} \quad (2.3.4)$$

Notation. Ein Punkt über einem Vektor ($\dot{\mathbf{r}}$) stellt hierbei die formale zeitliche Ableitung der einzelnen Komponenten des Vektors im gewählten Bezugssystem dar. Er berechnet i.A. nicht die absolute zeitliche Änderung \mathbf{v} . Folglich gilt in einem Inertialsystem ${}_0\mathbf{v}_P = {}_0\dot{\mathbf{r}}_P$, während in bewegten Bezugssystemen k dieser Zusammenhang im Allgemeinen nicht gilt

$${}_k\mathbf{v}_P \neq {}_k\dot{\mathbf{r}}_P!$$

In der absoluten zeitlichen Ableitung (\mathbf{v}) wird die Rotation der Einheitsvektoren des Bezugssystems berücksichtigt, in der komponentenweisen Ableitung ($\dot{\mathbf{r}}$) nicht.

Die Matrix ${}_k\mathbf{A}_0 {}_0\dot{\mathbf{A}}_k$ in (2.3.4) ist schiefsymmetrisch, wie leicht durch folgende Rechnung, ausgehend von (2.2.8), gezeigt werden kann:

$$\begin{aligned} \frac{d}{dt}({}_0\mathbf{A}_k^T {}_0\mathbf{A}_k) &= \mathbf{0} \\ \Leftrightarrow {}_0\mathbf{A}_k^T {}_0\dot{\mathbf{A}}_k &= -{}_0\dot{\mathbf{A}}_k^T {}_0\mathbf{A}_k \\ &= -({}_0\mathbf{A}_k^T {}_0\dot{\mathbf{A}}_k)^T \end{aligned} \quad (2.3.5)$$

Komponentenweise kann ${}_k\mathbf{A}_0 {}_0\dot{\mathbf{A}}_k$ folgendermaßen geschrieben werden:

$${}_k\mathbf{A}_0 {}_0\dot{\mathbf{A}}_k = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (2.3.6)$$

Die Einträge entsprechen den Komponenten der Winkelgeschwindigkeit ${}_k\boldsymbol{\omega}$ des Körpers relativ zu B_0 . Zur Darstellung der Matrix wird der Tilde-Operator eingeführt, der aus einem Vektor $\boldsymbol{\omega}$ eine entsprechende Matrix generiert:

$${}_k\tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (2.3.7)$$

Aus der Definition des Tilde-Operators ergeben sich unmittelbar folgende Rechenregeln:

$$\tilde{\boldsymbol{\omega}} \mathbf{r} = \boldsymbol{\omega} \times \mathbf{r} \quad (2.3.8)$$

$$\tilde{\boldsymbol{\omega}} \mathbf{r} = \tilde{\mathbf{r}}^T \boldsymbol{\omega} = \boldsymbol{\omega} \times \mathbf{r} \quad (2.3.9)$$

$${}_0\tilde{\boldsymbol{\omega}} = {}_0\mathbf{A}_k {}_k\tilde{\boldsymbol{\omega}} {}_k\mathbf{A}_0 \quad (2.3.10)$$

Die letzte Zeile entspricht der Transformationsvorschrift für Tensoren zweiter Stufe.

Anmerkung. Der Zusammenhang zwischen der Winkelgeschwindigkeit ω , der Matrix $\tilde{\omega}$ und ${}_k\mathbf{A}_{00}\dot{\mathbf{A}}_k$ kann auch über eine Betrachtung der Änderung der Position des Punktes P durch eine kleine Drehung des Koordinatensystems hergestellt werden (TAYLOR-Entwicklung).

Mit Hilfe der Winkelgeschwindigkeit ω kann die Geschwindigkeit des Punktes P kompakter geschrieben werden:

$${}_k\mathbf{v}_P = {}_k\mathbf{v}_k + {}_k\tilde{\omega} {}_k\mathbf{r}_{kP} + {}_k\dot{\mathbf{r}}_{kP} \quad (2.3.11)$$

$$= {}_k\mathbf{v}_k + {}_k\omega \times {}_k\mathbf{r}_{kP} + {}_k\dot{\mathbf{r}}_{kP} \quad (2.3.12)$$

Relativkinematik – Beschleunigung Die Beschleunigung folgt formal aus der Ableitung von (2.3.3) im Inertialsystem:

$${}_0\mathbf{a}_P := {}_0\dot{\mathbf{v}}_P = {}_0\mathbf{a}_k + {}_0\ddot{\mathbf{A}}_k {}_k\mathbf{r}_{kP} + 2{}_0\dot{\mathbf{A}}_k {}_k\dot{\mathbf{r}}_{kP} + {}_0\mathbf{A}_k {}_k\ddot{\mathbf{r}}_{kP} \quad (2.3.13)$$

Und transformiert ins körperfeste System B_k :

$${}_k\mathbf{a}_P = {}_k\mathbf{a}_k + {}_k\mathbf{A}_{00}\ddot{\mathbf{A}}_k {}_k\mathbf{r}_{kP} + 2{}_k\mathbf{A}_{00}\dot{\mathbf{A}}_k {}_k\dot{\mathbf{r}}_{kP} + {}_k\ddot{\mathbf{r}}_{kP} \quad (2.3.14)$$

Mit Hilfe der Winkelgeschwindigkeit und Winkelbeschleunigung ist wiederum eine kompaktere Darstellung möglich. Die Winkelbeschleunigung ergibt sich aus der Ableitung der Geschwindigkeit zu:

$${}_k\dot{\tilde{\omega}} = \frac{d}{dt} ({}_k\mathbf{A}_{00}\dot{\mathbf{A}}_k) = {}_k\dot{\mathbf{A}}_{00}\dot{\mathbf{A}}_k + {}_k\mathbf{A}_{00}\ddot{\mathbf{A}}_k \quad (2.3.15)$$

$$= \underbrace{{}_k\dot{\mathbf{A}}_{00}\mathbf{A}_k}_{{}_k\tilde{\omega}^T = -{}_k\tilde{\omega}} \underbrace{{}_k\mathbf{A}_{00}\dot{\mathbf{A}}_k}_{{}_k\tilde{\omega}} + {}_k\mathbf{A}_{00}\ddot{\mathbf{A}}_k \quad (2.3.16)$$

$$= -{}_k\tilde{\omega} {}_k\tilde{\omega} + {}_k\mathbf{A}_{00}\ddot{\mathbf{A}}_k \quad (2.3.17)$$

$$\Rightarrow {}_k\mathbf{A}_{00}\ddot{\mathbf{A}}_k = {}_k\dot{\tilde{\omega}} + {}_k\tilde{\omega} {}_k\tilde{\omega} \quad (2.3.18)$$

Die Beschleunigung (2.3.14) kann damit auch folgendermaßen geschrieben werden:

$${}_k\mathbf{a}_P = {}_k\mathbf{a}_k + ({}_k\dot{\tilde{\omega}} + {}_k\tilde{\omega} {}_k\tilde{\omega}) {}_k\mathbf{r}_{kP} + 2{}_k\tilde{\omega} {}_k\dot{\mathbf{r}}_{kP} + {}_k\ddot{\mathbf{r}}_{kP} \quad (2.3.19)$$

2.3.3 Rekursive Berechnung der direkten Kinematik

Für baumstrukturierte Systeme bietet sich eine rekursive Berechnung der Kinematik an. Da jeder Körper nur einen Vorgänger hat, können die kinematischen Größen dieses Körpers direkt aus denen des Vorgängers und der relativen Kinematik zum Vorgänger berechnet werden.

Das grundsätzliche Vorgehen kann für die Berechnung der Positionsvektoren bei einer kinematischen Kette veranschaulicht werden. In Pseudocode lautet die rekursive Berechnungsvorschrift:

Berechne: ${}_0\mathbf{r}_0$

für k **von** 1 **bis** N **tue**

 Berechne: ${}_k\mathbf{r}_k = {}_k\mathbf{A}_{v(k)} {}_{v(k)}\mathbf{r}_{v(k)} + {}_k\mathbf{r}_{v(k),k}$ oder: ${}_k\mathbf{r}_k = {}_k\mathbf{A}_{v(k)} \left({}_{v(k)}\mathbf{r}_{v(k)} + {}_{v(k)}\mathbf{r}_{v(k),k} \right)$

Ende

Notation. Der Index des Vorgängers des Körpers k wird als $v(k)$ geschrieben.

Die Rekursionsvorschrift für Baumstrukturen ist etwas komplexer als diejenige für Ketten. Eine einfache Möglichkeit ist die Verwendung des oben beschriebenen Algorithmus in Form einer einfachen Schleife. Dies ist zulässig, wenn bei der Nummerierung der Körper darauf geachtet wird, dass für alle Körper k die Ungleichung $v(k) < k$ gilt. Damit ist sichergestellt, dass ein Vorgänger in der Berechnung stets vor seinen Nachfolgern aufgerufen wird. Eine elegante Möglichkeit unter Verwendung von Zeigern wurde in (Yamane 2004) vorgeschlagen.

Vorteile der rekursiven Berechnung sind die einfache Programmierbarkeit, die allgemeine Anwendbarkeit für beliebige Systeme und die numerische Effizienz.

**Code-
Beispiel**

Ein Beispiel zur Programmierung der rekursiven Berechnung der direkten Kinematik finden Sie im begleitenden Matlab-Skript zur direkten Kinematik (beispiel_dk.m). Matlab ist ein Produkt von The MathWorks Inc. <http://www.themathworks.com>. Sie können über die Campus-Lizenz der TUM die Software kostenfrei beziehen, siehe <https://matlab.rbg.tum.de>. Alle Matlab-Beispiele werden im Verlauf der Vorlesung über <http://moodle.tum.de> bereitgestellt.

Allgemeine Beziehungen

Aufgrund der Baumstruktur kann die rekursive Berechnung, die im vorigen Abschnitt beispielhaft für Ortsvektoren angegeben wurde, allgemein für alle kinematischen Größen durchgeführt werden. Im Folgenden sind die Gleichungen für die zentralen kinematischen Größen angegeben.

$$\text{Ortsvektor} \quad {}^i\mathbf{r}_i = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{r}_{v(i)} + {}^i\mathbf{r}_{v(i)i} \quad (2.3.20)$$

$$\text{Drehmatrix} \quad {}^0\mathbf{A}_i = {}^0\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{A}_i \quad (2.3.21)$$

$$\text{Winkelgeschwindigkeit} \quad {}^i\boldsymbol{\omega}_i = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\boldsymbol{\omega}_{v(i)} + {}^i\boldsymbol{\omega}_{v(i)i} \quad (2.3.22)$$

$$\text{Absolute Geschwindigkeit} \quad {}^i\mathbf{v}_i = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{v}_{v(i)} + {}^i\mathbf{v}_{v(i)i} \quad (2.3.23)$$

Weitere kinematische Größen können aus obigen Gleichungen bestimmt werden. Bei numerischen Verfahren der inversen Kinematik (siehe Abschnitt 2.4) und der Berechnung der Dynamik (siehe Abschnitt 3) wird ein linearer Zusammenhang zwischen den Gelenkgeschwindigkeiten $\dot{\mathbf{q}}$ und den vorgegebenen Geschwindigkeiten im Arbeitsraum $\dot{\mathbf{w}}$ verwendet. Der Vektor $\dot{\mathbf{w}}$ kann dabei sowohl Winkelgeschwindigkeiten $\boldsymbol{\omega}$ als auch translatorische Geschwindigkeiten \mathbf{v} enthalten. Diese Abbildungen können über Jacobi-Matrizen \mathbf{J}_{TO} , \mathbf{J}_R beschrieben werden. Wie später noch gezeigt wird, gelten die folgenden Zusammenhänge:

$$\boldsymbol{\omega} = \mathbf{J}_R \dot{\mathbf{q}} \quad \text{und} \quad \mathbf{v} = \mathbf{J}_{TO} \dot{\mathbf{q}}. \quad (2.3.24)$$

Dabei ist \mathbf{J}_R die Jacobi-Matrix der Rotation und \mathbf{J}_{TO} die Jacobi-Matrix der Translation. Für deren Berechnung können die Rekursionsformeln direkt durch Differentiation bestimmt werden:

$${}^i\mathbf{J}_{Ri} := \frac{\partial {}^i\boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}} = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{J}_{R,v(i)} + \underbrace{\frac{\partial {}^i\boldsymbol{\omega}_{v(i)i}}{\partial \dot{\mathbf{q}}}}_{:= {}^i\mathbf{J}_{R,rel,i}}, \quad (2.3.25)$$

$${}^i\mathbf{J}_{TOi} := \frac{\partial {}^i\mathbf{v}_i}{\partial \dot{\mathbf{q}}} = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{J}_{TO,v(i)} + \underbrace{\frac{\partial {}^i\mathbf{v}_{v(i)i}}{\partial \dot{\mathbf{q}}}}_{:= {}^i\mathbf{J}_{T,rel,i}}. \quad (2.3.26)$$

Bei der späteren Auswertung der Dynamik-Gleichungen werden für die Berechnung von $\frac{d}{dt}(\mathbf{v})$ auch die Matrizen \mathbf{J}_{Rq} und \mathbf{J}_{TqO} benötigt, die wie folgt definiert sind:

$$\begin{aligned} {}^i\mathbf{J}_{Rq,i,[k,l]} &:= \frac{\partial {}^i\omega_{i,[k]}}{\partial \dot{q}_{[l]}} = {}^i\mathbf{A}_{v(i),[k,m]} {}^{v(i)}\mathbf{J}_{Rq,v(i),[m,l]} \\ &\quad + \underbrace{\frac{\partial {}^i\omega_{v(i)i,[k]}}{\partial \dot{q}_{[l]}}}_{:= {}^i\mathbf{J}_{Rq,rel,i,[k,l]}}, \end{aligned} \quad (2.3.27)$$

$$\begin{aligned} {}^i\mathbf{J}_{TqO,i,[k,l]} &:= \frac{\partial {}^i\mathbf{v}_{i,[k]}}{\partial \dot{q}_{[l]}} = {}^i\mathbf{A}_{v(i),[k,m]} {}^{v(i)}\mathbf{J}_{TqO,v(i),[m,l]} \\ &\quad + \underbrace{\frac{\partial {}^i\mathbf{v}_{v(i)i,[k]}}{\partial \dot{q}_{[l]}}}_{:= {}^i\mathbf{J}_{Tq,rel,i,[k,l]}}. \end{aligned} \quad (2.3.28)$$

Notation. Für die Darstellung der Vektorgradienten bezüglich der Minimalkoordinaten \mathbf{q} wird die Index-Schreibweise verwendet. In diesem Fall sind die Indizes der Vektoren und Matrizen in eckige Klammern eingeschlossen – $M_{[i,j]}$ bezeichnet also die i -te Zeile und j -te Spalte der Matrix \mathbf{M} . In diesem Fall gilt die EINSTEIN'sche Summenkonvention, d.h., über gleiche Indizes (in eckigen Klammern) wird summiert. Das Vektor-Matrix-Produkt $\mathbf{a} = \mathbf{M}\mathbf{b}$ wird in diesem Fall als $a_{[i]} = M_{[i,j]}b_{[j]}$ geschrieben.

Relative Größen

Die relativen kinematischen Größen zwischen dem i -ten Körper und seinem Vorgänger $v(i)$, d.h. die Translation ${}_{v(i)}\mathbf{r}_{v(i)i}$, die Rotationen ${}_i\mathbf{A}_{v(i)}$ usf., hängen von dem die beiden Körper verbindenden Gelenk ab. Im folgenden wird auf die Berechnung für Dreh- und Schubgelenke eingegangen. Die Vorgehensweise lässt sich jedoch leicht auf allgemeine Gelenke anwenden. Ausgangspunkt der Berechnungen ist die DH-Transformation (2.2.24), die zur Beschreibung von Dreh- und Schubgelenken geeignet ist. Das prinzipielle Vorgehen ist jedoch unabhängig von der Verwendung der DH-Parameter.

Relative Größen für Drehgelenke Bei Drehgelenken ist der DH-Parameter d_i eine feste, durch die Konstruktion vorgegebene Größe, während $\theta_i = q_i$ als Variable den Freiheitsgrad des Gelenks beschreibt.

Aus (2.2.24) folgen unmittelbar die Drehmatrix und Relativverschiebung:

$${}_{v(i)}\mathbf{r}_{v(i)i} = \mathbf{A}_x^T(\alpha_{i-1})(\mathbf{e}_z d_i + \mathbf{e}_x a_{i-1}) = \begin{pmatrix} a_{i-1} \\ -\sin(\alpha_{i-1})d_i \\ \cos(\alpha_{i-1})d_i \end{pmatrix} = \text{konstant!} \quad (2.3.29)$$

$${}_i\mathbf{A}_{v(i)} = \mathbf{A}_z(q_i)\mathbf{A}_x(\alpha_{i-1}) \quad (2.3.30)$$

Die übrigen Größen werden durch Differentiation berechnet:

$${}_{v(i)}\mathbf{v}_{v(i)i} = {}_{v(i)}\boldsymbol{\omega}_{v(i)} \times {}_{v(i)}\mathbf{r}_{v(i)i} = {}_{v(i)}\tilde{\mathbf{r}}_{v(i)i}^T {}_{v(i)}\boldsymbol{\omega}_{v(i)} \quad (2.3.31)$$

$${}_i\boldsymbol{\omega}_{v(i)i} = \mathbf{e}_z \dot{q}_i \quad (2.3.32)$$

$${}_i J_{R,rel,i,[k,l]} = \begin{cases} 1 & \text{für } l = i \wedge k = 3 \\ 0 & \text{sonst} \end{cases} \quad (2.3.33)$$

$${}_i J_{T,rel,i} = {}_i\mathbf{A}_{v(i)} \left({}_{v(i)}\tilde{\mathbf{r}}_{v(i)i}^T {}_{v(i)} J_{R,v(i)} \right) \quad (2.3.34)$$

$$(2.3.35)$$

Relative Größen für Schubgelenke Für Schubgelenke entspricht d_i dem Freiheitsgrad q_i , während θ_i ein fester Konstruktionsparameter ist. Für die kinematischen Größen erhält man:

$${}_{v(i)}\mathbf{r}_{v(i)i} = \mathbf{A}_x^T(\alpha_{i-1})(\mathbf{e}_z d_i + \mathbf{e}_x a_{i-1}) = \underbrace{\begin{pmatrix} a_{i-1} \\ 0 \\ 0 \end{pmatrix}}_{:= {}_{v(i)}\mathbf{r}_{v(i)i,0} = \text{konstant!}} + \underbrace{\begin{pmatrix} 0 \\ -\sin(\alpha_{i-1}) \\ +\cos(\alpha_{i-1}) \end{pmatrix}}_{:= {}_{v(i)}\Delta\mathbf{r}_{v(i)i}(q_i)} q_i \quad (2.3.36)$$

$${}_i\mathbf{A}_{v(i)} = \mathbf{A}_z(\theta_i)\mathbf{A}_x(\alpha_{i-1}) = \text{konstant!} \quad (2.3.37)$$

$${}_{v(i)}\mathbf{v}_{v(i)i} = \mathbf{A}_x^T(\alpha_{i-1})\mathbf{e}_z \dot{q}_i + {}_{v(i)}\boldsymbol{\omega}_{v(i)} \times {}_{v(i)}\mathbf{r}_{v(i)i} \quad (2.3.38)$$

$${}_i\boldsymbol{\omega}_{v(i)i} = \mathbf{0} \quad (2.3.39)$$

$${}_i J_{R,rel,i} = \mathbf{0} \quad (2.3.40)$$

$${}_i J_{T,rel,i,[k,l]} = {}_i\mathbf{A}_{v(i)} \left({}_{v(i)}\tilde{\mathbf{r}}_{v(i)i}^T {}_{v(i)} J_{R,v(i)} \right)_{[k,l]} + \begin{cases} ({}_i\mathbf{A}_{v(i)} \mathbf{A}_x^T(\alpha_{i-1})\mathbf{e}_z)_{[k]} & \text{für } l=i \\ 0 & \text{sonst} \end{cases} \quad (2.3.41)$$

$$(2.3.42)$$

2.4 Inverse Kinematik

2.4.1 Einleitung

Aufgabenstellung

Aufgabe der inversen Kinematik ist die Berechnung der Minimalkoordinaten \mathbf{q} aus den gegebenen Arbeitsraum-Koordinaten \mathbf{w}_d . Dies ist im einfachsten Fall durch Lösung des Gleichungssystems

$$\mathbf{w}_d = \mathbf{w}(\mathbf{q}) \quad (2.4.1)$$

möglich. Der tiefgestellte Index d bei \mathbf{w} kennzeichnet die Tatsache, dass es sich um eine gewünschte Konfiguration handelt ("desired").

Wie eingangs erwähnt ist die Lösung des inversen Problems komplexer als die des direkten, da kein funktioneller Zusammenhang zwischen \mathbf{w} und \mathbf{q} besteht und darüber hinaus im Allgemeinen die Dimension des Arbeitsraums $\dim(\mathbf{w})$ kleiner derjenigen des Konfigurationsraums $\dim(\mathbf{q})$ ist. Bei der Lösung können je nach Mechanismus und Konfiguration unterschiedliche Fälle auftreten:

- Es gibt genau eine Lösung \mathbf{q}
- Es gibt mehrere Lösungen
- Es gibt keine Lösung
- Es gibt unendlich viele Lösungen

Die Situationen sind beispielhaft in Bild 2.6 dargestellt.

Lösungsverfahren

Das inverse Problem kann durch geometrische Überlegungen, eine analytische Auflösung von (2.4.1) oder über numerische Methoden erfolgen. Jede der Verfahren hat spezifische Vor- und Nachteile.

Die geometrische Methode ist auf einfache, spezielle Fälle beschränkt. Nur in solchen Fällen ist durch „Hinsehen“ die Lösung etwa über die Betrachtung rechtwinkliger Dreiecke unmittelbar ersichtlich. Die Methode eignet sich gut für eine detaillierte Analyse, erleichtern die Einsicht in spezifische Eigenschaften der Kinematik und ist in aller Regel in der numerischen Auswertung sehr effizient. Darüber hinaus können alle möglichen Lösungen angegeben werden, falls das Problem mehrere Lösungen hat.

Die analytische Auflösung von (2.4.1) gelingt ebenfalls nur in einfachen Fällen durch Handrechnung. In diesem Fall ist die Methode ähnlich effizient und ähnlich gut zur Analyse geeignet wie der geometrische Ansatz. Auch hier können alle Lösungen bestimmt werden.

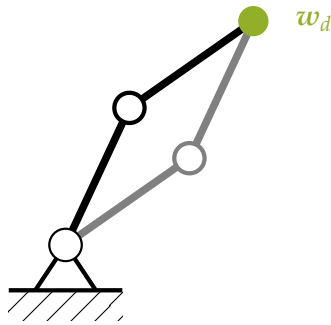
Die numerische Lösung der inversen Kinematik ist allgemein möglich. Hierbei wird jedoch in aller Regel nur eine der möglicherweise zahlreichen Lösungen bestimmt. Die Lösung ist darüber hinaus numerisch aufwändiger und weniger gut für eine tiefere Einsicht in die Eigenheiten der gewählten Kinematik geeignet. Vorteilhaft ist die Möglichkeit zur Berücksichtigung von Nebenbedingungen oder der Wahl einer möglichst günstigen Lösung unter den mitunter unendlich vielen möglichen.

Beispiel zur Analytischen und Geometrischen Lösung

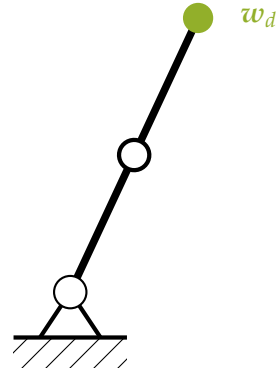
Betrachtet wird ein ebener Roboter mit drei Drehgelenken¹. Die Arbeitsraumkoordinaten setzen sich aus der Position des dritten Gelenks und der Orientierung des dritten Körpers zusammen (vgl. Bild 2.7):

$$\mathbf{w} = \begin{pmatrix} x_2 \\ y_2 \\ \varphi_3 \end{pmatrix}. \quad (2.4.2)$$

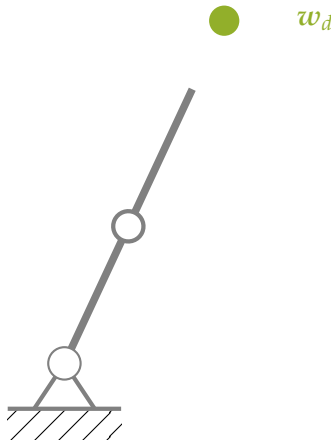
¹Das Beispiel ist (Siciliano, Sciavicco u. a. 2008, Seite 91ff) entnommen.



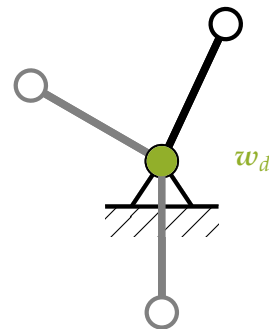
Fall 1: Es gibt zwei Lösungen. w_d liegt im Inneren von \mathcal{W} .



Fall 2: Es gibt genau eine Lösung. w_d auf dem Rand von \mathcal{W} .



Fall 3: Es gibt keine Lösung. w_d liegt außerhalb von \mathcal{W} .



Fall 4: Es gibt unendlich viele Lösungen (der zweite Körper überdeckt genau den ersten). w_d liegt im Innern von \mathcal{W} .

Bild 2.6: Darstellung einiger Situationen, die bei der Lösung eines Problems der inversen Kinematik auftreten können, am Beispiel eines ebenen Knickarm-Roboters (Doppelpendel). Gegeben ist eine gewünschte Position des Endeffektors $w_d^T = (x \ y)$ (grüner Punkt), gesucht sind die Gelenkwinkel $q^T = (\theta_1 \ \theta_2)$. Je nach Lage von w_d im Gebiet \mathcal{W} der erreichbaren Endeffektorpositionen und der Konfiguration des Systems kann es keine, eine, mehrere oder unendlich viele Lösungen geben.

Als generalisierte Koordinaten werden die relativen Gelenkwinkel q_i der Körper zum jeweiligen Vorgänger gewählt.

Direkte Kinematik

$${}^0r_{0P} = l_1 \begin{pmatrix} c q_1 \\ s q_1 \end{pmatrix} + l_2 \begin{pmatrix} c q_{12} \\ s q_{12} \end{pmatrix} \quad (2.4.3)$$

$$\varphi_3 = q_1 + q_2 + q_3 \quad (2.4.4)$$

$$\Rightarrow w = \begin{pmatrix} l_1 c q_1 + l_2 c q_{12} \\ l_1 s q_1 + l_2 s q_{12} \\ q_1 + q_2 + q_3 \end{pmatrix} \quad (2.4.5)$$

Analytische Lösung Das zu lösende Gleichungssystem lautet mit dem Ergebnis des obigen Absatzes:

$$x_d = l_1 c q_1 + l_2 c q_{12} \quad (2.4.6)$$

$$y_d = l_1 s q_1 + l_2 s q_{12} \quad (2.4.7)$$

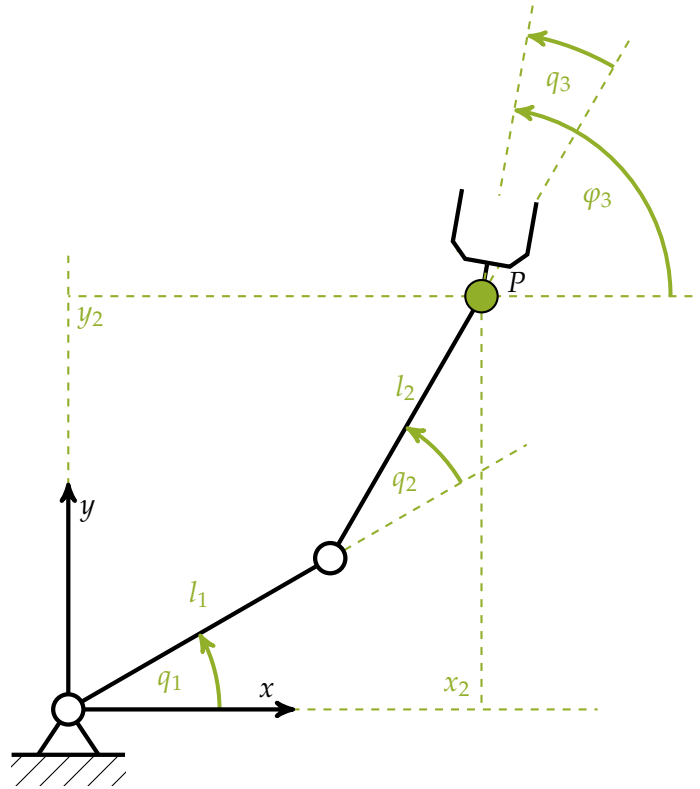


Bild 2.7: Beispielsystem zur Veranschaulichung des analytischen und geometrischen Vorgehens zur Lösung der inversen Kinematik.

$$\varphi_d = q_1 + q_2 + q_3 \quad (2.4.8)$$

Quadrieren und addieren der ersten beiden Gleichungen liefert:

$$x_d^2 + y_d^2 = l_1^2 (\cos q_1)^2 + 2l_1 l_2 \cos q_1 \cos q_{12} + l_2^2 (\cos q_{12})^2 \quad (2.4.9)$$

$$+ l_1^2 (\sin q_1)^2 + 2l_1 l_2 \sin q_1 \sin q_{12} + l_2^2 (\sin q_{12})^2 \quad (2.4.10)$$

$$= l_1^2 + l_2^2 + 2l_1 l_2 (\cos q_1 \cos q_{12} + \sin q_1 \sin q_{12}) \quad [\text{mit: } \sin^2 x + \cos^2 x = 1] \quad (2.4.11)$$

$$= l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2 \quad [\text{mit: } \cos(a - b) = \cos a \cos b + \sin a \sin b] \quad (2.4.12)$$

$$\Rightarrow q_2 = \arccos \left(\frac{x_d^2 + y_d^2 - (l_1^2 + l_2^2)}{2l_1 l_2} \right) \quad (2.4.13)$$

Zur Prüfung der Plausibilität der Lösung kann der Gelenkwinkel in der Strecklage berechnet werden. Hier gilt $x_d^2 + y_d^2 = (l_1 + l_2)^2$ und somit

$$q_2 = \arccos(1) = 0. \quad (2.4.14)$$

Für Punkte außerhalb des Arbeitsbereichs gilt offenbar $x_d^2 + y_d^2 > (l_1 + l_2)^2$ (vgl. Bild 2.6) und somit:

$$\cos(q_2) = \frac{x_d^2 + y_d^2 - (l_1^2 + l_2^2)}{2l_1 l_2} > 1. \quad (2.4.15)$$

Die Nicht-Erreichbarkeit von w_d spiegelt sich darin wieder, dass obige Gleichung keine reelwertige Lösung hat.

Zur Berechnung des ersten Gelenkwinkels q_1 ist es zweckmäßig, die Gleichungen (2.4.6) und (2.4.7) mit Hilfe der trigonometrischen Additionstheoreme ($\sin(a + b) = \sin a \cos b + \cos a \sin b$, $\cos(a + b) = \cos a \cos b - \sin a \sin b$) umzuformen:

$$x_d = l_1 \cos q_1 + l_2 (\cos q_1 \cos q_2 - \sin q_1 \sin q_2) \quad (2.4.16)$$

$$= \cos q_1 (l_1 + l_2 \cos q_2) - l_2 \sin q_1 \sin q_2 \quad (2.4.17)$$

$$y_d = l_1 \sin q_1 + l_2 (\sin q_1 \cos q_2 + \cos q_1 \sin q_2) \quad (2.4.18)$$

$$= l_2 \cos q_1 \sin q_2 + (l_1 + l_2 \cos q_2) \sin q_1 \quad (2.4.19)$$

Auflösen des Gleichungssystems nach $\sin q_1, \cos q_1$ ergibt (mit $\sin^2 x + \cos^2 x = 1$):

$$\sin q_1 = \frac{-l_2 \sin q_2 x_d + (l_1 + l_2 \cos q_2) y_d}{l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2} \quad (2.4.20)$$

$$\cos q_1 = \frac{(l_1 + l_2 \cos q_2) x_d + l_2 \sin q_2 y_d}{l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2} \quad (2.4.21)$$

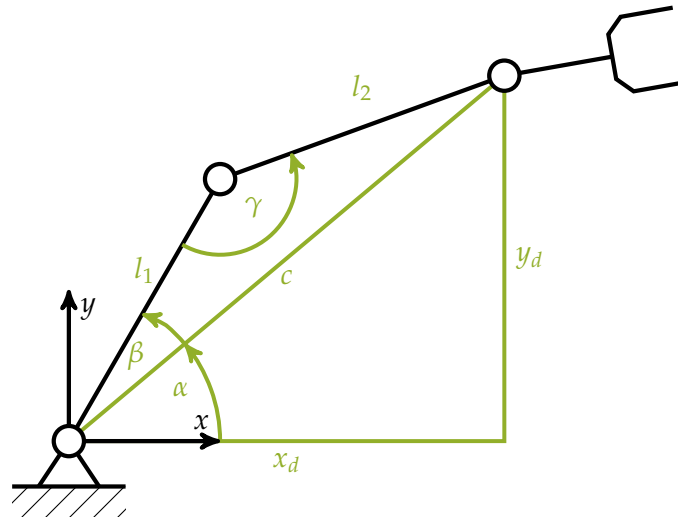
Der gesuchte Winkel ergibt sich damit zu

$$q_1 = \text{Arctan2}(\sin q_1, \cos q_1), \quad (2.4.22)$$

wobei Arctan2 eine Arkustangens Funktion ist, die das Vorzeichen über alle Quadranten korrekt bestimmt. Mit q_1 und q_2 folgt q_3 direkt zu:

$$q_3 = \varphi_d - q_1 - q_2. \quad (2.4.23)$$

Geometrische Lösung Zur Bestimmung der Lösung durch geometrische Überlegungen wird das durch die ersten beiden Körper beschriebene Dreieck betrachtet:



Der Kosinussatz liefert direkt:

$$\begin{aligned} c^2 &= l_1^2 + l_2^2 - 2l_1 l_2 \cos \gamma \\ &= l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - q_2) \\ &= l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2 \\ &\stackrel{!}{=} x_d^2 + y_d^2 \end{aligned} \quad (2.4.24)$$

Die letzte Gleichung folgt über den Satz von Pythagoras. Damit erhält man wiederum:

$$q_2 = \arccos \left(\frac{x_d^2 + y_d^2 - (l_1^2 + l_2^2)}{2l_1 l_2} \right) \quad (2.4.25)$$

Für q_1 gilt offenbar:

$$q_1 = \alpha + \beta, \quad (2.4.26)$$

mit

$$\alpha = \text{Arctan2}(y_d, x_d) \quad (2.4.27)$$

Der Winkel β kann wiederum mit dem Kosinussatz bestimmt werden:

$$l_2^2 = l_1^2 + c^2 - 2l_1c \cos \beta \quad (2.4.28)$$

$$\Rightarrow \cos \beta = \frac{l_1^2 - l_2^2 + (x_d^2 + y_d^2)}{2l_1 \sqrt{x_d^2 + y_d^2}} \quad \text{mit: } c^2 = x_d^2 + y_d^2 \quad (2.4.29)$$

$$(2.4.30)$$

Damit ist β und somit auch q_2 bestimmt.

Für den letzten Winkel gilt wiederum $q_3 = \varphi_d - q_1 - q_2$.

2.4.2 Numerische Lösung

Lösung auf Lageebene

Das einfachste numerische Verfahren besteht in einer iterativen Lösung der nichtlinearen Gleichung $w_d = w(q)$ ausgehend von einem geeigneten Anfangswert q_0 . Einfach und gut geeignete ist das Newton-Raphson Verfahren, das jedoch im klassischen Fall nur für nichtredundante Roboter, also für $(\dim(q) = \dim(w))$ anwendbar ist.

Das Verfahren geht von einer linearen Approximation von (2.4.1) aus:

$$w(q_0 + \Delta q) \approx w(q_0) + \frac{\partial w}{\partial q} \Delta q \quad (2.4.31)$$

Diese Gleichung wird so lange iterativ gelöst, bis die Näherung der Lösung q den Anforderungen an die Genauigkeit $\varepsilon > 0$ genügt. Mit der Bezeichnung $J_w = \frac{\partial w}{\partial q}$ für die Jacobi-Matrix der Arbeitsraum-Koordinaten lautet der Algorithmus:

Setze Startlösung q_0

Setze Index $k = 0$

wiederhole

 Löse $w_d - w(q_k) = J_w(q_k) \Delta q$ nach Δq

 Setze $q_{k+1} = q_k + \Delta q$

 Setze $k \leftarrow k + 1$

bis $\|\Delta q\| < \varepsilon$;

Für die inverse Kinematik relevante Eigenschaften des Verfahrens sind:

- Die Lösung ist vom Startwert abhängig
- Die Konvergenz ist für gute Startwerte sehr gut („superlinear“)
- Divergenz für schlechte Startwerte
- Keine Lösung bei singulärer Matrix J_w
- Nur für nichtredundante Roboter anwendbar

Eine Erweiterung auf unterbestimmte Gleichungssysteme, d.h. auf redundante Roboter, ist durch Anwendung der MOORE-PENROSE Pseudoinverse (siehe Abschnitt 2.4.3) möglich.

Singularitäten

Ist die Jacobi-Matrix J_w singulär, kann das Newton-Verfahren nicht zur Lösung der inversen Kinematik verwendet werden, da J_w nicht invertiert, bzw. das Gleichungssystem $J_w \Delta q = \Delta w$ nicht gelöst werden kann. In diesem Fall verschwindet die Determinante der Jacobi-Matrix, es gilt $\det(J_w) = 0$.

In diesem Fall sind einige Spalten von J_w linear abhängig. Gleichzeitig gilt der Zusammenhang

$$\dot{w} = \frac{\partial w(q)}{\partial q} \dot{q} = J_w \dot{q}. \quad (2.4.32)$$

Die Arbeitsraum-Geschwindigkeiten sind eine Linearkombination der Spalten der Jacobi-Matrix. Sind die Spalten nicht länger linear unabhängig, verringert sich die Dimension des Raums möglicher Arbeitsraum-Geschwindigkeiten: es kommt zu einem Verlust an Freiheitsgraden.

Eine typische Ursache ist eine Strecklage des Roboters, hier ist eine weitere Streckung nicht möglich. Man spricht von einer Randsingularität, da sie am Rand der erreichbaren Punkte im Arbeitsraum liegt. Liegt die Singularität im Inneren, spricht man von einer inneren Singularität.

Lösung auf Geschwindigkeitsebene

Während die kinematischen Beziehungen auf Lageebene nichtlinear sind, sind sie auf Ebene der Geschwindigkeiten linear:

$$\dot{w} = \frac{\partial w(q)}{\partial q} \dot{q} = J_w \dot{q}. \quad (2.4.33)$$

Die Berechnung der Minimalgeschwindigkeiten \dot{q} aus gegebenen Arbeitsraum-Geschwindigkeiten \dot{w} ist bei bekannten Lagekoordinaten q also durch die Lösung eines linearen Gleichungssystems möglich:

$$\dot{q} = J_w^{-1} \dot{w} \quad (2.4.34)$$

Methode der Inversen Jacobi-Matrix Meist wird in der Anwendung nicht eine isolierte Lösung der inversen Kinematik für eine Konfiguration gesucht, sondern für eine ganze Trajektorie $w_d(t) = w(q(t))$. Ist für einen Zeitpunkt t_0 die Lösung bekannt, kann die Lösung $q(t)$ aus (2.4.34) durch Integration bestimmt werden:

$$q(t) = q(t_0) + \int_{t_0}^t J_w^{-1}(q(\tau)) \dot{w}_d(\tau) d\tau \quad (2.4.35)$$

In der Umsetzung wird die Integration meist durch ein explizites Euler-Verfahren umgesetzt. Damit ergibt sich für den Zeitpunkt $k+1$ die Lösung zu:

$$q_{k+1} = q_k + J_w^{-1}(q_k) \dot{w}_d(t_k) \Delta t \quad (2.4.36)$$

Hierbei bezeichnet $\Delta t = t_{k+1} - t_k$ die Länge eines Zeitschritts.

Durch numerische Fehler kommt es bei der direkten Anwendung von (2.4.36) zu einem unbegrenzten Drift. Um das Verfahren in der Praxis anwendbar zu machen, wird daher die Geschwindigkeit \dot{w}_d modifiziert, um den Drift zu begrenzen. Hierbei wird (2.4.36) als zu regelnde Strecke betrachtet, die Arbeitsraum-Koordinaten w als Messgröße, w_d als Sollgröße und \dot{w}_d als Eingang (siehe auch Kapitel 6).

Durch die Einführung der modifizierten Arbeitsraum-Geschwindigkeiten

$$\dot{w}_{d,\text{eff}} = \dot{w}_d + K(w_d - w(q)), \quad (2.4.37)$$

mit einer positiv definiten Verstärkungsmatrix K erhält man ein konvergentes Verfahren ohne numerische Drift. Der Ansatz ist in Bild 2.8 veranschaulicht.

Das Konvergenzverhalten ist unmittelbar durch Einsetzen von (2.4.37) in (2.4.33) ersichtlich:

$$\dot{w}_d - \dot{w} + K(w_d - w(q)) = 0 \quad (2.4.38)$$

Für jedes positiv definite K ergibt sich damit ein asymptotisches Einschwingen auf w_d mit frei wählbarer Zeitkonstante. In einer diskreten Umsetzung wird der Bereich stabiler K durch die endliche Abtastzeit begrenzt. Darüber hinaus ist nur außerhalb von Singularitäten die Stabilität gewährleistet, da sonst wegen $\det(J_w) \rightarrow 0$ auch $\|\dot{q}\| \rightarrow \infty$ gilt.

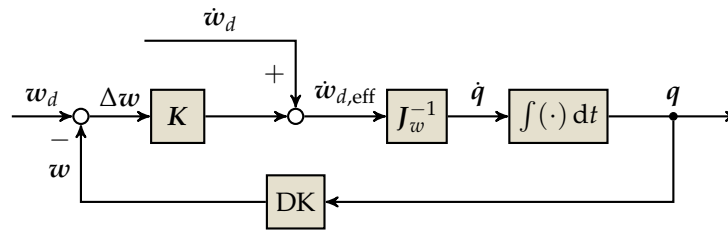


Bild 2.8: Methode der inversen Jacobi-Matrix. DK bezeichnet die direkte Kinematik.

Methode der Transponierten Jacobi-Matrix Bei der Methode der inversen Jacobi-Matrix wird das Problem der inversen Kinematik als regelungstechnische Aufgabe betrachtet, wobei die exakte kinematische Gleichung auf Ebene der Geschwindigkeiten als Modell der Strecke verwendet wird. Dies erfordert die Inversion der Jacobi-Matrix, was numerisch teuer sein kann und in der Nähe von Singularitäten instabil wird.

Aus der Dynamik ist bekannt², dass eine auf einen Körper wirkende Kraft F im Punkt P auf den Roboter die gleiche Wirkung hat, wie die Gelenkmomente $Q = J_T^T F$, wobei J_T die Jacobi-Matrix der Translation des Punktes P bezeichnet (siehe Kapitel 3). Dies motiviert das Verfahren der Transponierten Jacobi-Matrix, bei dem die Beziehung $\dot{q} = J_w^{-1} \dot{w}$ durch

$$\dot{q} = J_w^T \dot{w} \quad (2.4.39)$$

ersetzt wird. Abgesehen davon wird das Verfahren analog aufgebaut, siehe Bild 2.9.

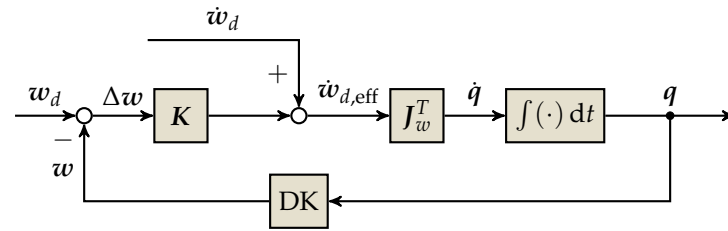


Bild 2.9: Methode der transponierten Jacobi-Matrix. DK bezeichnet die direkte Kinematik.

Die Stabilität ist dabei nicht länger offensichtlich, da die Beziehung $\dot{q} = J_w^T \dot{w}$ nicht exakt ist. Die Stabilität kann jedoch über das Verfahren von LYAPUNOV analysiert werden. Für eine detaillierte Beschreibung der Methode sei auf die Literatur, sowie andere Vorlesungen verwiesen (Khalil 1992; Pfeiffer 1989).

Um Stabilität nachzuweisen muss zunächst eine Funktion $V(e)$ des Arbeitsraum-Fehlers $e = w_d - w(q)$ gefunden werden, für die

$$V(e) > 0 \quad \forall e \neq 0 \quad (2.4.40)$$

$$V(e) = 0 \quad \text{für } e = 0 \quad (2.4.41)$$

gilt. Gelingt dann der Beweis, dass $\dot{V} < 0 \forall e$, ist die Stabilität nachgewiesen.

Mit dem Ansatz

$$V(e) = \frac{1}{2} e^T K e \quad (2.4.42)$$

berechnet sich \dot{V} zu:

$$\dot{V} = e^T K \dot{e} = e^T K (\dot{w}_d - J_w \dot{q}) = e^T K \left[\dot{w}_d - J_w \left(J_w^T (\dot{w}_d + K e) \right) \right]. \quad (2.4.43)$$

Umgeformt ergibt sich schlussendlich

$$\dot{V} = e^T K \left(E - J_w J_w^T \right) \dot{w}_d - e^T K J_w J_w^T K e \quad (2.4.44)$$

²Siehe Seite 38, Gleichungen (3.3.37), (3.3.38) und (3.3.39).

Über den ersten Summanden kann für $\dot{w}_d \neq 0$ keine allgemeine Aussage gemacht werden, während der zweite außerhalb von Singularitäten stets kleiner Null ist. Im statischen Fall ist damit außerhalb von Singularitäten die asymptotische Konvergenz gezeigt, wobei bezüglich diskreter Implementierungen die im Vorigen Abschnitt erwähnten Einschränkungen gelten.

Für singuläre Konfiguration ist die asymptotische Konvergenz zwar nicht gesichert. Es gilt jedoch auch hier $\dot{V} \leq 0$. Somit ist das Verfahren noch stabil, wenn auch nicht asymptotisch stabil.

Eine alternative Herleitung des Verfahrens folgt aus dem Optimierungsproblem für den kleinsten Abstand zur Sollvorgabe w_d :

$$\Phi = \frac{1}{2} e^2 \rightarrow \min!, \quad e = w_d - w(q). \quad (2.4.45)$$

Das Minimum kann über das Verfahren des steilsten Abstiegs berechnet werden. Die Lösung erfolgt iterativ entsprechend dem Gradienten von Φ und einer "Schrittweite" α :

$$q^{k+1} = q^k - \alpha \frac{\partial \Phi}{\partial q} = q^k + \alpha J_w^T (w_d - w(q^k)). \quad (2.4.46)$$

Wird eine Iteration pro Reglertakt durchgeführt, $\alpha = \Delta t$ gesetzt und der Arbeitsraum-Fehler e mit der effektiven Geschwindigkeit $\dot{w}_{d,\text{eff}}$ ersetzt, folgt die oben vorgestellte Methode der Transponierten Jacobi-Matrix:

$$\dot{q}^{k+1} \approx \frac{q^{k+1} - q^k}{\Delta t} = J_w^T \dot{w}_{d,\text{eff}}. \quad (2.4.47)$$

Die Verwendung der „effektiven“ Arbeitsraumvorgabe $\dot{w}_{d,\text{eff}}$ kann dabei als eine Art Vorsteuerung für das Verfahren betrachtet werden, wodurch die sich ändernde Zielvorgabe besser berücksichtigt werden kann. Die Stabilitätseigenschaften des Verfahrens (siehe oben) sind auch aus dieser alternativen Herleitung des Verfahrens heraus verständlich.

Das Verhalten des Verfahrens im Falle einer Singularität soll am Beispiel der Randsingularität des Doppelpendels veranschaulicht werden (siehe Bild 2.10). In der Strecklage mit $q_2 = 0$ lautet

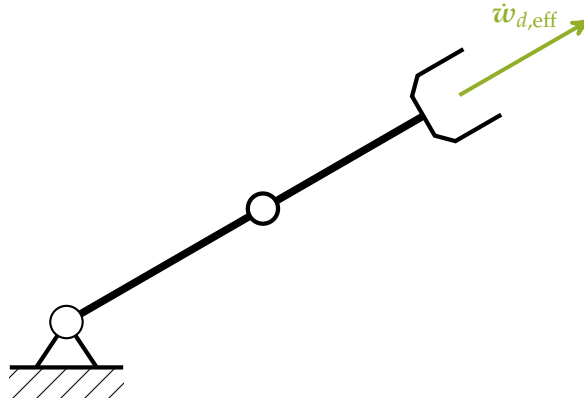


Bild 2.10: Verfahren der transponierten Jacobi-Matrix bei einer Randsingularität.

die Jacobi-Matrix der Endposition des Doppelpendels:

$$J_w|_{q_2=0} = \begin{pmatrix} -(l_1 + l_2) s q_1 & -l_2 s q_1 \\ (l_1 + l_2) c q_1 & l_2 c q_1 \end{pmatrix} \quad (2.4.48)$$

Für eine nach außen gerichtete effektive Soll-Geschwindigkeit mit Betrag v

$$\dot{w}_{d,\text{eff}} = v \begin{pmatrix} c q_1 \\ s q_1 \end{pmatrix} \quad (2.4.49)$$

ergibt sich \dot{q} zu:

$$\dot{q} = J_w^T \dot{w}_{d,\text{eff}} = v \begin{pmatrix} -(l_1 + l_2) s q_1 & (l_1 + l_2) c q_1 \\ -l_2 s q_1 & l_2 c q_1 \end{pmatrix} \begin{pmatrix} c q_1 \\ s q_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.4.50)$$

Der Roboter bleibt also in der Singulären Lage stehen: Der Endeffektor erreicht den Punkt, der den geringsten Abstand zur Sollvorgabe hat.

Im Allgemeinen ergibt sich die Arbeitsraum-Geschwindigkeit durch Einsetzen obiger Gleichung in (2.4.33) zu:

$$\dot{w} = J_w J_w^T \dot{w}_{d,\text{eff}} \quad (2.4.51)$$

Die Bedeutung der Matrix $J_w J_w^T$ soll im Folgenden erläutert werden. Für $\dot{w}_d = \mathbf{0}$ vereinfacht sich die Zeitableitung der Lyapunov-Funktion nach (2.4.44) mit der Wahl $K = E$ zu:

$$\dot{V} = -e^T K J_w J_w^T K e = -(J_w^T e)^T (J_w^T e). \quad (2.4.52)$$

Die Komponenten in $J_w^T e$ sind die Skalarprodukte der Spalten von J_w , die den Raum möglicher Geschwindigkeiten aufspannen, mit dem Fehler im Arbeitsraum. Abweichungen von der Vorgabe w_d , die Außerhalb der möglichen Richtungen liegen, beeinflussen das Ergebnis nicht. Im Beispiel ist dies eine Komponente in Richtung des gestreckten Arms. Sofern noch Anteile in e enthalten sind, die durch die Kinematik verringert werden können (mathematisch: $e \in \text{im}(J_w)$, im : Bild der Matrix), ist $\dot{V} < 0$. Somit ist klar, dass das Verfahren zu einer Lösung konvergiert, die keine weitere Verkleinerung des Fehlers zulässt. Anschaulich gesprochen filtert die Matrix $J_w J_w^T$ diejenigen Komponenten der Vorgabe aus, die nicht realisiert werden können.

Code- Beispiel

Ein Beispiel zur Programmierung der inversen Kinematik mit den Methoden der inversen und transponierten Jacobi-Matrix finden Sie im begleitenden Matlab-Skript zur inversen Kinematik (`beispiel_ik.m`).

2.4.3 Redundante Roboter

Ist die Dimension des Arbeitsraums kleiner als die des Konfigurationsraums, ist der Roboter kinematisch redundant: im Regelfall hat das inverse Problem unendlich viele Lösungen. Ob ein Roboter redundant ist oder nicht, hängt sowohl von der Aufgabe, also der Definition von w , als auch von der kinematischen Struktur des Systems ab. Ein typisches Beispiel eines redundanten Systems ist der menschliche Arm mit sieben Freiheitsgraden, wenn Position und Orientierung der Hand als Arbeitsraum-Koordinaten gewählt werden. Für redundante Systeme können neben der Hauptaufgabe $w_d = w(q)$ auch nachrangige Aufgaben erfüllt werden.

Resolved Motion Rate Control

Die Resolved Motion Rate Control ist eine klassische Methode zur Lösung des inversen Problems für redundante Systeme, die auf WHITNEY zurück geht (Whitney 1969). Die zugrundeliegende Idee ist die Formulierung eines Optimierungsproblems auf Geschwindigkeitsebene:

$$\Phi = \frac{1}{2} \dot{q}^T W \dot{q} \rightarrow \min! \quad (2.4.53)$$

mit: $g = \dot{w}_d - J_w \dot{q} = \mathbf{0}$

Hierbei ist W eine positiv definite Gewichtungsmatrix, die eine Anpassung der Gütefunktion Φ ermöglicht. Durch Lösung des Optimierungsproblems werden unter allen möglichen Lösungen diejenigen gewählt, die bezüglich der Gewichtungsmatrix W zu minimalen Gelenkgeschwindigkeiten führt. Die Nebenbedingung $g = \mathbf{0}$ stellt sicher, dass nur Lösungen gefunden werden, die die Arbeitsraum-Vorgabe erfüllen.

Das Problem (2.4.53) kann durch die Methode der Lagrange-Multiplikatoren gelöst werden. Hierzu werden die Nebenbedingungen g über Lagrange-Multiplikatoren λ zu der Lagrange-Funktion hinzugefügt:

$$L(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T W \dot{q} + \lambda^T (\dot{w}_d - J_w \dot{q}) \quad (2.4.54)$$

Das Optimum kann formal durch Berechnung der stationären Punkte von L bestimmt werden:

$$\left(\frac{\partial L}{\partial \dot{q}} \right)^T = W \dot{q} - J_w^T \lambda = \mathbf{0} \quad (2.4.55)$$

$$\left(\frac{\partial L}{\partial \lambda}\right)^T = \dot{w}_d - J_w \dot{q} = \mathbf{0} \quad (2.4.56)$$

Durch Auflösen von (2.4.55) nach \dot{q} und Einsetzen in (2.4.56) kann λ bestimmt werden. Über λ folgt dann durch (2.4.55) \dot{q} zu:

$$\dot{q} = \underbrace{W^{-1} J_w^T (J_w W^{-1} J_w^T)^{-1}}_{:= J_w^\#} \dot{w}_d \quad (2.4.57)$$

Die Matrix $J_w^\#$ ist die mit W -gewichtete Moore-Penrose Pseudoinverse.

Das Verfahren minimiert zu jedem Zeitpunkt *lokal* die erforderlichen Gelenkgeschwindigkeiten. Dies ist im Allgemeinen nicht gleichbedeutend mit der Minimierung der Geschwindigkeiten für die gesamte Trajektorie.

Anmerkung. In der praktischen Umsetzung sollte \dot{w}_d zur Driftkompensation durch $\dot{w}_{d,\text{eff}}$ entsprechend (2.4.37) ersetzt werden.

Automatic Supervisory Control

Eine Erweiterung des Verfahrens wurde von Liégeois vorgeschlagen (Liégeois 1977). Das Verfahren kann auf eine Erweiterung der Optimierungsaufgabe (2.4.53) um eine lokale Minimierung einer beliebigen Gütefunktion $H(q)$ zurückgeführt werden. Ein beliebiges $H(q)$ ändert sich in einem Zeitschritt durch die Wahl von \dot{q} um:

$$\Delta H = \dot{H} \Delta t = \frac{\partial H}{\partial q} \dot{q} \Delta t \quad (2.4.58)$$

Das verallgemeinerte Problem mit einer gewichteten Summe aus (2.4.58) und (2.4.53) kann mit Hilfe des Parameters $\alpha > 0$ folgendermaßen angegeben werden:

$$\phi = \frac{1}{2} \dot{q}^T W \dot{q} + \alpha \frac{\partial H}{\partial q} \dot{q} \rightarrow \min! \quad (2.4.59)$$

$$g = \dot{w}_d - J_w \dot{q} = \mathbf{0} \quad (2.4.60)$$

Die Lösung erfolgt wiederum über die Methode der Lagrange-Multiplikatoren. Mit der Lagrange-Funktion

$$L = \frac{1}{2} \dot{q}^T W \dot{q} + \alpha \frac{\partial H}{\partial q} \dot{q} + \lambda^T (\dot{w}_d - J_w \dot{q}) \quad (2.4.61)$$

ergeben sich die Optimalitätsbedingungen zu:

$$\left(\frac{\partial L}{\partial \dot{q}}\right)^T = W \dot{q} + \alpha \left(\frac{\partial H}{\partial q}\right)^T - J_w^T \lambda \quad (2.4.62)$$

$$\left(\frac{\partial L}{\partial \lambda}\right)^T = \dot{w}_d - J_w \dot{q} \quad (2.4.63)$$

\dot{q} ergibt sich aus (2.4.62) zu

$$\dot{q} = W^{-1} J_w^T \lambda - \underbrace{W^{-1} \alpha \left(\frac{\partial H}{\partial q}\right)^T}_{:= -z} \quad (2.4.64)$$

Einsetzen in (2.4.63) ergibt:

$$\dot{w}_d = \underbrace{J_w W^{-1} J_w^T}_B \lambda + J_w z. \quad (2.4.65)$$

Durch Auflösen nach λ ergibt sich wiederum eine geschlossene Lösung für \dot{q} :

$$\dot{q} = J_w^\# \dot{w}_d - \alpha N_w W^{-1} \left(\frac{\partial H}{\partial q} \right)^T \quad (2.4.66)$$

Die neu eingeführte Matrix N_w ist die *Nullraum-Projektionsmatrix*:

$$N_w = E - J_w^\# J_w \quad (2.4.67)$$

Die Matrix N_w stellt sicher, dass der gewichtete Gradient $W^{-1} \frac{\partial H}{\partial q}$ keinen Einfluss auf die resultierenden Arbeitsraum-Geschwindigkeiten \dot{w} hat – es gilt für beliebige $x \in \mathbb{R}^n$:

$$J_w N_w x = J_w x - \underbrace{J_w J_w^\#}_{E} J_w x = 0 \quad (2.4.68)$$

Der Gradient von H wird in den *Nullraum* der Matrix J_w projiziert.

Numerisch günstiger als die geschlossene Lösung ist eine Rückbesinnung auf die Optimalitätsbedingungen. Durch numerisches Lösen von (2.4.65) nach λ , Substitution in (2.4.64) und numerische Lösung des resultierenden Gleichungssystems kann \dot{q} schneller berechnet werden (siehe auch Nakamura 1991):

Setze $B = J_w W^{-1} J_w^T$

Setze $p = \dot{w}_d - J_w z$

Löse $B\lambda = p$ nach λ auf

Setze $\dot{q} = W^{-1} (J_w^T \lambda - z)$

Da W in aller Regel diagonal ist, ist die Inversion von W nicht aufwändig.

Anmerkung. In der praktischen Umsetzung sollte \dot{w}_d zur Driftkompensation durch $\dot{w}_{d,\text{eff}}$ entsprechend (2.4.37) ersetzt werden.

Eine typische Anwendung einer Gütefunktion H ist die Einführung einer bevorzugten Konfiguration q^* . Durch Wahl der Gütefunktion

$$H = \frac{1}{2} (q - q^*)^2 \quad (2.4.69)$$

werden Lösungen gewählt, die den Roboter näher an die bevorzugte Konfiguration q^* bringen.

Ein weiterer häufiger Anwendungsfall sind Straffunktionen in Nähe der Grenzen der Arbeitsbereiche der Gelenke. Durch Einführung einer Funktion H , die Nahe am Rand des Arbeitsbereichs stark wächst, können Verletzungen der Grenzen oft vermieden werden.

Für eine vertiefte Behandlung der Themen Redundanz, Optimierung und Vermeidung von Singularitäten sei auf das Buch (Nakamura 1991) verwiesen.

Kapitel 3

Dynamik

3.1 Einleitung

Die Dynamik ist die Lehre der Kräfte und deren Wirkung. In der Dynamik wird also nach dem Zusammenhang zwischen der Bewegung und den verursachenden Kräften gefragt. Über erste Prinzipien wie Impuls- und Drallsatz, die Prinzipien von D’ALEMBERT oder HAMILTON werden die Bewegungsgleichungen hergeleitet, die den Zusammenhang zwischen Kräften und Bewegungen mathematisch ausdrücken. Für die Klasse der hier behandelten Systeme sind die Bewegungsgleichungen (BGLn) Differentialgleichungen (DGLn) zweiter Ordnung.

Die Bewegungsgleichungen können auf zwei grundsätzliche Arten ausgewertet werden. Bei gegebenen Kräften können die resultierenden Bewegungen durch Lösen der BGLn als Anfangswertproblem (AWP) berechnet werden. Man spricht von der *direkten Dynamik*. Die Lösung der direkten Dynamik erlaubt die Programmierung eines „virtuellen Roboters“, der zur Evaluation des Verhaltens und zur Auslegung von Regelungsverfahren vor der Fertigstellung des Systems herangezogen werden kann.

Beim inversen Problem, der *inversen Dynamik*, werden aus einer gegebenen Bewegung die verursachenden Kräfte berechnet. Die Berechnung der zu einer Bewegung gehörenden Kräfte ist in der Regelung, etwa zur Vorsteuerung, wertvoll. Eine erste Auslegung des Roboters während der Konstruktionsphase kann ebenfalls mit Hilfe der inversen Dynamik für ein Kollektiv vorgesehener Bewegungen erfolgen. Dabei muss davon ausgegangen werden, dass die ideale Bewegung tatsächlich ausgeführt wird. Sollen Effekte durch endliche Regelabweichungen berücksichtigt werden, muss die direkte Dynamik des Roboters gemeinsam mit der Regelung und vorgesehenen Bewegung simuliert werden.

Zur Aufstellung der Bewegungsgleichungen bieten sich in der Robotik zwei grundsätzlich unterschiedliche Ansätze an: die LAGRANGE’sche Methode zweiter Art und die NEWTON-EULER Methode (Bremer 1988; Pfeiffer und Reithmeier 1987).

Die LAGRANGE’sche Methode zweiter Art ist ein analytisches Verfahren, das von den Energien des Gesamtsystems ausgehend durch „Zerlegung“ in die Wirkrichtungen entsprechend der Minimalkoordinaten die BGLn erzeugt.

Dem gegenüber steht die synthetische Methode nach NEWTON-EULER. Ausgehend von den grundlegenden Prinzipien der virtuellen Arbeit (bzw. virtuellen Leistung) sowie dem Impuls- und Drallsatz können die BGLn des Gesamtsystems aus Beiträgen der Teilkörper erzeugt bzw. synthetisiert werden.

3.2 Grundbegriffe

3.2.1 Zwangsbedingungen

Die Teilkörper eines Roboters sind über Gelenke miteinander verbunden, die einzelne Bewegungsrichtungen freigeben, andere jedoch sperren. Die Bewegungen der Teilkörper sind daher nicht unabhängig voneinander, sie sind über *Zwangsbedingungen* miteinander gekoppelt.

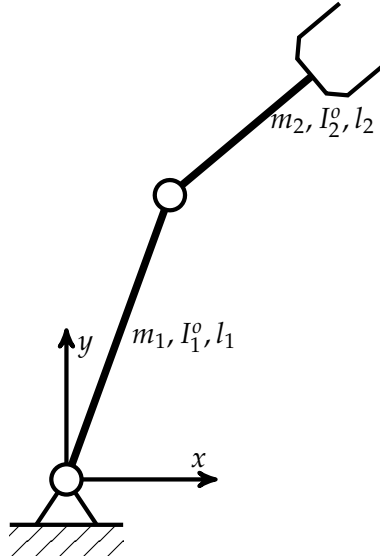


Bild 3.1: Ein ebener Roboter mit zwei rotatorischen Freiheitsgraden als Beispiel eines Systems mit Zwangsbedingungen. Die beiden Körper können sich nicht frei in der Ebene bewegen, da sie über Gelenke miteinander und der Umwelt gekoppelt sind.

Beispielhaft ist in Bild 3.1 ein zweiachsiger Roboter in der Ebene abgebildet. Die Konfiguration des Systems kann durch die Systemkoordinaten $z \in \mathbb{R}^6$ mit den Positionen der Schwerpunkte und der Rotation in der Ebene für jeden Teilkörper beschrieben werden. Im Fall des Systems in Bild 3.1 legen die Zwangsbedingungen fest, dass der Anfang des ersten Körpers $r_{p1}(z)$ im Koordinatenursprung bleibt und der Endpunkt des ersten Körpers $r_{p2}(z)$ mit dem Anfangspunkt des zweiten $r_{p3}(z)$ zusammenfällt:

$$\Phi(z) := \begin{pmatrix} r_{p1}(z) \\ r_{p2}(z) - r_{p3}(z) \end{pmatrix} = \mathbf{0}. \quad (3.2.1)$$

Zwangsbedingungen dieser Form, die nur von Positionsgrößen abhängen, werden holonom genannt. Im Allgemeinen sind auch Zwangsbedingungen der Form $\Phi(z, \dot{z}) = \mathbf{0}$ möglich, in denen Geschwindigkeiten auftreten. Derartige nichtholonome Bedingungen treten oft bei Systemen mit Rollbedingung auf, etwa bei Fahrzeugen. Im Folgenden werden nur holonome Zwangsbedingungen betrachtet.

3.2.2 Virtuelle Verschiebungen

Definition 3.2.1. *Virtuelle Verschiebungen* sind gedachte (infinitesimal kleine) Verschiebungen, die mit den Zwangsbedingungen verträglich sind. Die Zeit bleibt während der Verschiebung konstant.

Definition 3.2.2. *Virtuelle Geschwindigkeiten* sind gedachte, mit den Zwangsbedingungen verträgliche Geschwindigkeiten. Die Zeit und die Lage bleibt konstant. Virtuelle Geschwindigkeiten müssen nicht klein sein.

In der Praxis können virtuelle Verschiebungen und Geschwindigkeiten ähnlich einem totalen Differential gebildet werden, wobei darauf zu achten ist, welche Größen konstant bleiben. Ist für einen Ortsvektor r ein expliziter Zusammenhang mit den Minimalkoordinaten q bekannt, berechnet sich die virtuelle Verschiebung zu:

$$\delta r = \frac{\partial r}{\partial q} \delta q \quad (3.2.2)$$

Das totale Differential wäre hingegen:

$$dr = \frac{\partial r}{\partial q} dq + \frac{\partial r}{\partial t} dt \quad (3.2.3)$$

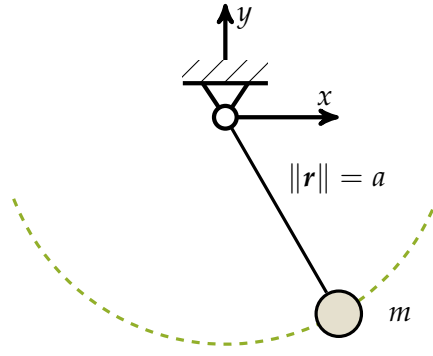


Bild 3.2: Das Fadenpendel als Beispiel eines Systems mit holonomen Zwangsbedingungen.

Virtuelle Verschiebungen und Geschwindigkeiten werden zur Unterscheidung vom totalen Differential durch ein geschwungenes δ gekennzeichnet.

Beispiel Fadenpendel Der Zusammenhang zwischen Zwangsbedingungen und virtuellen Verschiebungen soll beispielhaft anhand des Fadenpendels dargestellt werden (siehe Bild 3.2).

Die Punktmasse m ist über eine Stange¹ konstanter Länge l mit dem Gelenk im Ursprung des Koordinatensystems verbunden. Die Position $\mathbf{r}^T = (x \ y \ z)$ der Punktmasse hat daher immer den Abstand l zum Ursprung:

$$\Phi = \frac{1}{2}(x^2 + y^2 + z^2 - l^2) \quad (3.2.4)$$

Da die Zwangsbedingung $\Phi = 0$ immer gilt, muss auch die Variation $\delta\Phi$ verschwinden:

$$\delta\Phi = \frac{\partial\Phi}{\partial\mathbf{r}}\delta\mathbf{r} = (x \ y \ z) \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} = \mathbf{r}^T \delta\mathbf{r} = 0 \quad (3.2.5)$$

Die virtuellen Verschiebungen sind orthogonal zum Ortsvektor \mathbf{r} , der wiederum die Normale der Fläche $\Phi = 0$ der möglichen Bewegungen ist. Allgemein sind die virtuellen Verschiebungen stets tangential zur durch $\Phi = 0$ definierten Mannigfaltigkeit.

3.2.3 Die Prinzipien von D'ALEMBERT und JOURDAIN

Nach dem Prinzip von D'ALEMBERT leisten die in einem System wirkenden Zwangskräfte keine Arbeit:

$$\int_S (\mathbf{a}_p dm - d\mathbf{F}^e)^T \delta\mathbf{r}_p = \int_S (d\mathbf{F}^z)^T \delta\mathbf{r}_p = 0 \quad (3.2.6)$$

In obiger Gleichung bezeichnet \mathbf{r}_p den Ort des Massenelements dm , S das System, $d\mathbf{F}^z$ die Zwangskräfte und $d\mathbf{F}^e$ die eingepprägten Kräfte. Zwangskräfte werden durch Bindungen verursacht, eingepprägte etwa durch Gravitation, magnetische Kräfte usw. In Worten besagt die Gleichung, dass Zwangskräfte keine Arbeit leisten. Diese Aussage ist geometrisch unmittelbar einsichtig: \mathbf{F}^z sind diejenigen Kräfte, die zur Einhaltung der Zwangsbedingungen $\Phi = 0$ erforderlich sind. Damit müssen sie orthogonal zur Mannigfaltigkeit $\Phi = 0$ sein. Die virtuellen Verschiebungen $\delta\mathbf{r}_p$ sind jedoch tangential hierzu und damit orthogonal zu den Zwangskräften. Damit muss der zweite Term in (3.2.6) wegen $(d\mathbf{F}^z)^T \delta\mathbf{r}_p = 0$ verschwinden.

Analog zum Prinzip von D'ALEMBERT gilt das Prinzip von JOURDAIN „Zwangskräfte erbringen keine Leistung“:

$$\int_S (\mathbf{a}_p dm - d\mathbf{F}^e)^T \delta\mathbf{v}_p = 0 \quad (3.2.7)$$

¹Hinweis: Mit dem Begriff „Fadenpendel“ wird gewöhnlich ein Pendel mit masseloser Aufhängung bezeichnet. Trotz der Verwendung des Wortes „Faden“ wird angenommen, dass Zug- und Druckkräfte übertragen werden können.

δv ist die virtuelle Geschwindigkeit, d.h. die Variation über der Geschwindigkeit bei konstantem Ort und konstanter Zeit.

Für eine detailliertere Ausführung dieser Grundbegriffe sei auf (Bremer 1988; Pfeiffer 1989), sowie das Fach Technische Dynamik verwiesen.

3.3 Das NEWTON-EULER Verfahren

In diesem Abschnitt werden die projektiven NEWTON-EULER Gleichungen hergeleitet.

3.3.1 Ausgangspunkt

Ausgangspunkt ist Prinzip von JOURDAIN:

$$\int_S (\mathbf{a}_p dm - d\mathbf{F}^e)^T \delta \mathbf{v}_p = 0 \quad (3.3.1)$$

Diese Gleichung beschreibt einen Zusammenhang zwischen Beschleunigungen und Kräften für alle möglichen δv_p . Die Gleichung beschreibt also bereits den gesuchten Zusammenhang zwischen Kräften und Bewegungen, ist jedoch für eine direkte Auswertung schlecht geeignet. Gesucht wird eine Darstellung als DGL in den Minimalkoordinaten \mathbf{q} . Hierzu müssen die bekannten kinematischen Zusammenhänge zwischen $\delta \mathbf{r}_p, \mathbf{a}_p$ und den \mathbf{q} ausgenutzt und das Integral aufgelöst werden. Zur Auswertung in der Vorwärtsdynamik muss die DGL numerisch integriert werden. Hierzu müssen für gegebene $\mathbf{q}, \dot{\mathbf{q}}$ und Kräfte die resultierenden $\ddot{\mathbf{q}}$ bestimmt werden. Die Schritte der Herleitung sind in Bild 3.3 nochmals veranschaulicht.

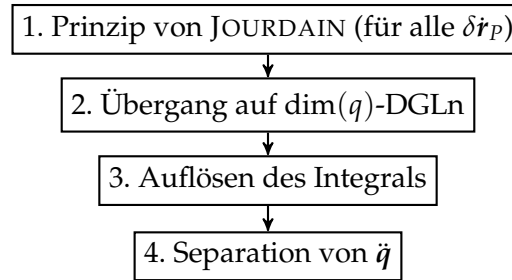


Bild 3.3: Vorgehen zur Herleitung der projektiven Newton-Euler Gleichungen.

2. Schritt: Übergang auf $\dim(\mathbf{q})$ Gleichungen Die virtuelle Geschwindigkeit δv_p kann als Funktion der Variation der Minimalgeschwindigkeiten $\delta \dot{\mathbf{q}}$ geschrieben werden:

$$\delta v_p = \frac{\partial v_p}{\partial \dot{\mathbf{q}}} \delta \dot{\mathbf{q}} \quad (3.3.2)$$

Eingesetzt in (3.3.1) folgt:

$$0 = \int_S (\mathbf{a}_p dm - d\mathbf{F}^e)^T \frac{\partial v_p}{\partial \dot{\mathbf{q}}} \delta \dot{\mathbf{q}} \quad (3.3.3)$$

$$= \delta \dot{\mathbf{q}}^T \int_S \frac{\partial v_p}{\partial \dot{\mathbf{q}}}^T (\mathbf{a}_p dm - d\mathbf{F}^e) \quad (3.3.4)$$

Diese Gleichung gilt für alle möglichen $\delta \dot{\mathbf{q}}$. Da die $\delta \dot{\mathbf{q}}$ jedoch *Minimalgeschwindigkeiten* sind, unterliegen sie keinerlei Einschränkungen. So gilt die Gleichung auch für die Wahl $\delta \dot{\mathbf{q}}^T = (1 \ 0 \ 0 \ \dots)$, $\delta \dot{\mathbf{q}}^T = (0 \ 1 \ 0 \ \dots)$, usf. Dies ist jedoch nur möglich, falls das Integral selbst verschwindet:

$$\int_S \frac{\partial v_p}{\partial \dot{\mathbf{q}}}^T (\mathbf{a}_p dm - d\mathbf{F}^e) = \mathbf{0} \quad (3.3.5)$$

3. Schritt: Auflösen des Integrals Da das betrachtete System ein Mehrkörpersystem ist, kann das Integral durch die Summe der Integrale über die Teilkörper K_i ersetzt werden:

$$\mathbf{0} = \sum_{i=1}^{N_{\text{Körper}}} \int_{K_i} \frac{\partial \mathbf{v}_P}{\partial \dot{\mathbf{q}}}^T (\mathbf{a}_P dm - d\mathbf{F}^e) \quad (3.3.6)$$

Im Folgenden gehen wir davon aus, dass alle Körper starr sind. Eine Verallgemeinerung auf elastische Körper ist möglich und führt in der Konsequenz zu derselben Gleichungsstruktur. Hierzu sei auf die Bücher (Bremer 1988, 2008; Bremer und Pfeiffer 1992) verwiesen.

Für starre Körper ist die Geschwindigkeit \mathbf{v}_P eine Funktion der Geschwindigkeit \mathbf{v}_i und Winkelgeschwindigkeit $\boldsymbol{\omega}_i$ des Körpers (siehe Kapitel 2):

$$\mathbf{v}_P = \mathbf{v}_i + \tilde{\boldsymbol{\omega}} \mathbf{r}_{iP} = \mathbf{v}_i + \tilde{\mathbf{r}}_{iP}^T \boldsymbol{\omega} \quad (3.3.7)$$

Damit folgt für die Jacobi-Matrix in (3.3.6):

$$\frac{\partial \mathbf{v}_P}{\partial \dot{\mathbf{q}}} = \left(\frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} + \tilde{\mathbf{r}}_{iP}^T \frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}} \right) \quad (3.3.8)$$

Einsetzen in (3.3.6) liefert:

$$\mathbf{0} = \sum_{i=1}^{N_{\text{Körper}}} \left\{ \int_{K_i} \left(\frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} \right)^T [\mathbf{a}_P dm - d\mathbf{F}^e] + \int_{K_i} \left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}} \right)^T [\tilde{\mathbf{r}}_{iP} \mathbf{a}_P dm - \tilde{\mathbf{r}}_{iP} d\mathbf{F}^e] \right\} \quad (3.3.9)$$

$$= \sum_{i=1}^{N_{\text{Körper}}} \left\{ \left(\frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} \right)^T \int_{K_i} [\mathbf{a}_P dm - d\mathbf{F}^e] + \left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}} \right)^T \int_{K_i} [\tilde{\mathbf{r}}_{iP} \mathbf{a}_P dm - \tilde{\mathbf{r}}_{iP} d\mathbf{F}^e] \right\} \quad (3.3.10)$$

Die einzelnen Terme (3.3.10) können für Starrkörper exakt aufgelöst werden. Hierzu wird die Gleichung zunächst umgeformt:

$$\mathbf{0} = \sum_{i=1}^{N_{\text{Körper}}} \left\{ \left(\frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} \right)^T \left[\underbrace{\int_{K_i} \mathbf{a}_P dm}_{\langle 1 \rangle} - \underbrace{\int_{K_i} d\mathbf{F}^e}_{\langle 3 \rangle} \right] + \left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}} \right)^T \left[\underbrace{\int_{K_i} \tilde{\mathbf{r}}_{iP} \mathbf{a}_P dm}_{\langle 2 \rangle} - \underbrace{\int_{K_i} \tilde{\mathbf{r}}_{iP} d\mathbf{F}^e}_{\langle 4 \rangle} \right] \right\} \quad (3.3.11)$$

Für den ersten Term $\langle 1 \rangle$ folgt unmittelbar aus der Definition des Schwerpunkts \mathbf{r}_S :

$$\int_{K_i} \mathbf{a}_P dm = m_i \mathbf{a}_{S_i} \quad (3.3.12)$$

Aus dem zweiten Term $\langle 2 \rangle$ folgt mit der Relativkinematik (2.3.19):

$$\int_{K_i} \tilde{\mathbf{r}}_{iP} \mathbf{a}_P dm = \underbrace{\int_{K_i} \tilde{\mathbf{r}}_{iP} \mathbf{a}_i dm}_{\langle 2.1 \rangle} + \underbrace{\int_{K_i} \tilde{\mathbf{r}}_{iP} \tilde{\boldsymbol{\omega}}_i \mathbf{r}_{iP} dm}_{\langle 2.2 \rangle} + \underbrace{\int_{K_i} \tilde{\mathbf{r}}_{iP} \tilde{\boldsymbol{\omega}}_i \tilde{\boldsymbol{\omega}}_i \mathbf{r}_{iP} dm}_{\langle 2.3 \rangle} \quad (3.3.13)$$

Das Integral aus $\langle 2.1 \rangle$ ist wieder über den Schwerpunkt auflösbar:

$$\langle 2.1 \rangle \quad \int_{K_i} \tilde{\mathbf{r}}_{iP} \mathbf{a}_i dm = \int_{K_i} \tilde{\mathbf{r}}_{iP} dm \mathbf{a}_i = m_i \tilde{\mathbf{r}}_{iS_i} \mathbf{a}_i \quad (3.3.14)$$

Hierbei bezeichnet \mathbf{r}_{iS_i} den Schwerpunkt relativ zum Körper-Ursprung (B_i).

Der zweite Term $\langle 2.2 \rangle$ lässt sich folgendermaßen umformen:

$$\langle 2.2 \rangle \quad \int_{K_i} \tilde{\mathbf{r}}_{iP} \tilde{\boldsymbol{\omega}}_i \mathbf{r}_{iP} dm = \int_{K_i} \tilde{\mathbf{r}}_{iP}^T \tilde{\mathbf{r}}_{iP} dm \dot{\boldsymbol{\omega}}_i = \mathbf{I}_i^O \dot{\boldsymbol{\omega}}_i \quad (3.3.15)$$

Die neu eingeführte Größe I_i^O ist der Trägheitstensor des Körpers bezüglich des Ursprungs des körperfesten Systems B_i .

Der letzte Term kann ebenso über den Trägheitstensor ausgedrückt werden:

$$\langle 2.3 \rangle \quad \int_{K_i} \tilde{\mathbf{r}}_{iP} \tilde{\boldsymbol{\omega}}_i \tilde{\boldsymbol{\omega}}_i \tilde{\mathbf{r}}_{iP} dm = \tilde{\boldsymbol{\omega}}_i \int_{K_i} \tilde{\mathbf{r}}_{iP}^T \tilde{\mathbf{r}}_{iP} dm \boldsymbol{\omega}_i = \tilde{\boldsymbol{\omega}}_i I_i^O \boldsymbol{\omega}_i \quad (3.3.16)$$

Die Umformung in obiger Gleichung ist über den Entwicklungssatz $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a}^T \mathbf{c}) - \mathbf{c}(\mathbf{a}^T \mathbf{b})$ möglich.

Die Integrale $\langle 3 \rangle$ und $\langle 4 \rangle$ führen auf die eingepprägten Kräfte \mathbf{F}_i^e und Momente $\mathbf{M}_{O_i}^e$ bezüglich des Körper-Ursprungs:

$$\int_{K_i} d\mathbf{F}^e = \mathbf{F}_i^e \quad (3.3.17)$$

$$\int_{K_i} \tilde{\mathbf{r}}_{iP} d\mathbf{F}^e = \mathbf{M}_{O_i}^e \quad (3.3.18)$$

$$(3.3.19)$$

Einsetzen der Terme in (3.3.10) liefert die Gleichung:

$$\sum_{i=1}^{N_{\text{Körper}}} \left\{ \underbrace{\left(\frac{\partial \mathbf{v}_i}{\partial \dot{\mathbf{q}}} \right)^T}_{J_{TO_i}^T} \left[\underbrace{m_i \mathbf{a}_{S_i}}_{\dot{\mathbf{p}}_i} - \mathbf{F}_i^e \right] + \underbrace{\left(\frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}} \right)^T}_{J_{Ri}^T} \left[m_i \tilde{\mathbf{r}}_{iS_i} \mathbf{a}_i + \underbrace{I_i^O \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i I_i^O \boldsymbol{\omega}_i}_{\dot{\mathbf{L}}_{O_i}} - \mathbf{M}_{O_i}^e \right] \right\} = \mathbf{0} \quad (3.3.20)$$

Mit den eingeführten Abkürzungen für Impulsänderung $\dot{\mathbf{p}}$, Dralländerung $\dot{\mathbf{L}}^O$ und den Jacobi-Matrizen der Translation J_{TO_i} und Rotation J_{Ri} lauten die projektiven NEWTON-EULER Gleichungen:

$$\sum_{i=1}^{N_{\text{Körper}}} \left\{ \begin{pmatrix} J_{TO_i} \\ J_{Ri} \end{pmatrix}^T \begin{pmatrix} \dot{\mathbf{p}}_i - \mathbf{F}_i^e \\ \dot{\mathbf{L}}_{O_i} + m_i \tilde{\mathbf{r}}_{iS_i} \mathbf{a}_i - \mathbf{M}_{O_i}^e \end{pmatrix} \right\} = \mathbf{0} \quad (3.3.21)$$

Falls Körper-Ursprung und Schwerpunkt zusammenfallen gilt $\mathbf{r}_{iS_i} = \mathbf{0}$ und die Gleichungen vereinfachen sich zu:

$$\sum_{i=1}^{N_{\text{Körper}}} \left\{ \begin{pmatrix} J_{T,S_i} \\ J_{Ri} \end{pmatrix}^T \begin{pmatrix} \dot{\mathbf{p}}_i - \mathbf{F}_i^e \\ \dot{\mathbf{L}}_{S_i} - \mathbf{M}_{S_i}^e \end{pmatrix} \right\} = \mathbf{0} \quad (3.3.22)$$

Die Gleichungen enthalten Impuls- und Drallsatz für jeden Körper, die gemeinsam $6N_{\text{Körper}}$ Gleichungen darstellen. Die Spalten von J_{TO_i}, J_{Ri} spannen den Raum der für den i -ten Körper möglichen Geschwindigkeiten auf. Die Linksmultiplikation mit den Jacobi-Matrizen bewirkt somit eine Elimination der Komponenten, die orthogonal zu $\Phi = \mathbf{0}$ stehen: die Gleichungen werden in die zwangsfreien Richtungen projiziert, wodurch die Zwangsbedingungen $\Phi = \mathbf{0}$ automatisch erfüllt werden.

Bei bewegtem Angriffspunkt einer Zwangskraft (etwa in einem Gelenk) verbleibt mitunter eine Komponente nach der Projektion, diese ist jedoch aufgrund des dritten NEWTON'schen Axioms „*actio=reactio*“ gegengleich dem Beitrag ist, durch den Kontaktpartner erbracht wird. In der Summe über alle Teilkomponenten leisten die Zwangskräfte damit keine Arbeit und müssen nicht in der Bewegungsgleichung berücksichtigt werden.

Die Summanden in (3.3.21) können prinzipiell in jedem KOS ausgewertet werden. In jedem Fall müssen jedoch der Implussatz und die zugehörige Jacobi-Matrix der Translation im gleichen System berechnet werden. Gleiches gilt auch für den Drallsatz mit zugehöriger Jacobi-Matrix der Rotation. In der Praxis wird der Drall meist im körperfesten System berechnet, da hier der Trägheitstensor I^O konstant ist. Der Impuls kann je nach Fall im körperfesten (B_i) oder inertialen KOS (B_0) berechnet werden.

4. Schritt: Separation von \ddot{q} In vielen Fällen, etwa in der direkten Dynamik, müssen die Beschleunigungen \ddot{q} für einen gegebenen Zustand \dot{q}, q und gegebene Kräfte berechnet werden. Dies ist jedoch nicht unmittelbar aus (3.3.21) möglich. In einfachen Fällen, in denen die BGLn per Hand aufgestellt werden, können die Beschleunigungen nach Anwendung von (3.3.21) manuell separiert werden. Im allgemeinen Fall ist jedoch ein automatisches Aufstellen der BGLn durch ein Programm anzustreben, sodass zur Separation der Beschleunigungen ein allgemeines Vorgehen benötigt wird. Dies führt überdies zu einer kompakten Darstellung der BGLn, die für Simulation und Regelung günstig ist.

Zur Separation von \ddot{q} werden alle Größen in (3.3.21) mit Hilfe der in Kapitel 2 hergeleiteten Gleichungen der direkten Kinematik als Funktion von q, \dot{q} und \ddot{q} geschrieben.

Für die Geschwindigkeiten und Positionen gilt:

$${}_i r_i = {}_i r_i(q) \quad (3.3.23)$$

$${}_i v_i = \frac{\partial {}_i v_i}{\partial \dot{q}} \dot{q} = {}_i J_{TO,i} \dot{q} \quad (3.3.24)$$

$${}_i \omega_i = \frac{\partial {}_i \omega_i}{\partial \dot{q}} \dot{q} = {}_i J_{Ri} \dot{q} \quad (3.3.25)$$

Die Beschleunigungen folgen daraus durch Bildung der absoluten zeitlichen Ableitung:

$${}_i a_i = \frac{d}{dt}({}_i v_i) = \frac{\partial {}_i v_i}{\partial \ddot{q}} \ddot{q} + \frac{\partial {}_i v_i}{\partial \dot{q}} \dot{q} + \underbrace{{}_i \omega_i \times {}_i v_i}_{\text{Euler-Ableitung}} \quad (3.3.26)$$

$$= {}_i J_{TO,i} \ddot{q} + {}_i J_{TqO,i} \dot{q} + {}_i \omega_i \times {}_i v_i \quad (3.3.27)$$

$${}_i \dot{\omega}_i = \frac{\partial {}_i \dot{\omega}_i}{\partial \ddot{q}} \ddot{q} + \frac{\partial {}_i \dot{\omega}_i}{\partial \dot{q}} \dot{q} + \underbrace{{}_i \omega_i \times {}_i \omega_i}_{=0} \quad (3.3.28)$$

$$= {}_i J_{Ri} \ddot{q} + {}_i J_{Rqi} \dot{q} \quad (3.3.29)$$

Anmerkung. Die Jacobi-Matrix $J_{TO,i}$ wurde für die absolute Geschwindigkeit v_i definiert, daher folgt (3.3.24) unmittelbar. Die Winkelgeschwindigkeit ω_i und Winkelbeschleunigung $\dot{\omega}_i$ sind ebenfalls absolute Größen. Für die Beschleunigungen muss in der absoluten zeitlichen Ableitung die Drehung des Koordinatensystems jedoch durch die „EULER-Ableitung“ ($\tilde{\omega} v_i$) berücksichtigt werden.

Mit obigen kinematischen Zusammenhängen gilt für die Terme in (3.3.21):

$${}_i [\dot{p}_i] = m_i {}_i a_{S_i} = m_i {}_i [a_i + (\tilde{\omega}_i + \tilde{\omega}_i \tilde{\omega}_i) r_{iS_i}] \quad (3.3.30)$$

$$= m_i {}_i \left[J_{T,o} \ddot{q} + J_{Tq,o} \dot{q} + \tilde{\omega}_i v_i + \tilde{\omega}_i \tilde{\omega}_i r_{iS_i} + \tilde{r}_{iS_i}^T (J_{Ri} \ddot{q} + J_{Rqi} \dot{q}) \right] \quad (3.3.31)$$

$${}_i [\dot{L}_{O,i} + m_i \tilde{r}_{iS_i} a_i] = {}_i \left[I_i^o (J_R \ddot{q} + J_{Rq} \dot{q}) + \tilde{\omega}_i I_i^o \omega_i + m_i \tilde{r}_{iS_i} (J_{TO,i} \ddot{q} + J_{TqO,i} \dot{q} + \tilde{\omega} v_i) \right] \quad (3.3.32)$$

Oder, zusammengefasst:

$$\begin{aligned} {}_i \left[\begin{array}{c} \dot{p}_i \\ \dot{L}_{O,i} + m_i \tilde{r}_{iS_i} a_i \end{array} \right] &= \underbrace{\begin{pmatrix} mE & m\tilde{r}_{iS_i}^T \\ m\tilde{r}_{iS_i} & I^o \end{pmatrix}}_{\bar{M}_{O_i}} \underbrace{\begin{pmatrix} J_{TO,i} \\ J_R \end{pmatrix}}_{\bar{J}_i} \ddot{q} + \underbrace{\begin{pmatrix} mE & m\tilde{r}_{iS_i}^T \\ m\tilde{r}_{iS_i} & I^o \end{pmatrix}}_{\bar{M}_{O_i}} \underbrace{\begin{pmatrix} J_{Tqo} \\ J_{Rq} \end{pmatrix}}_{\bar{J}_{qi}} \dot{q} \\ &\quad + \underbrace{\begin{pmatrix} m\tilde{\omega} v_i + m\tilde{\omega} \tilde{\omega} r_{iS_i} \\ \tilde{\omega} I^o \omega + m\tilde{r}_{iS_i} \tilde{\omega} v_i \end{pmatrix}}_{\bar{\xi}_i} \end{aligned} \quad (3.3.33)$$

Für das Gesamtsystem folgt damit:

$$0 = \sum_i^{N_i} \bar{J}_i^T \left\{ \bar{M}_i \bar{J}_i \ddot{q} + \bar{M}_i \bar{J}_{qi} \dot{q} + \bar{\xi}_i - \begin{pmatrix} F_g \\ M_g \end{pmatrix} - \begin{pmatrix} F_u \\ M_u \end{pmatrix} \right\}$$

$$= \underbrace{\sum_i^{N_i} \bar{J}_i^T \bar{M}_i \bar{J}_i}_{\mathbf{M}} \ddot{\mathbf{q}} + \underbrace{\sum_i^{N_i} \bar{J}_i^T \bar{M}_i \bar{J}_{q,i}}_{\mathbf{M}_q} \dot{\mathbf{q}} + \underbrace{\sum_i^{N_i} \bar{J}_i^T \bar{\xi}_i}_{\boldsymbol{\xi}} \quad (3.3.34)$$

$$- \underbrace{\sum_i^{N_i} \bar{J}_i^T \begin{pmatrix} \mathbf{F}_g \\ \mathbf{M}_g \end{pmatrix}}_{\mathbf{Q}_g} - \underbrace{\sum_i^{N_i} \bar{J}_i^T \begin{pmatrix} \mathbf{F}_u \\ \mathbf{M}_u \end{pmatrix}}_{\mathbf{Q}_u} \quad (3.3.35)$$

$$\mathbf{h} = \mathbf{M}_q(\mathbf{q})\dot{\mathbf{q}} + \boldsymbol{\xi}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}_g(\mathbf{q}) \quad (3.3.36)$$

Hierbei wurden die eingprägten Kräfte in Gravitationskräfte $\mathbf{F}_g, \mathbf{M}_g$ und die Antriebskräfte $\mathbf{F}_u, \mathbf{M}_u$ aufgeteilt. Zusätzliche Kräfte sind analog zu berücksichtigen. Die mit den entsprechenden Jacobi-Matrizen multiplizierten Kräfte werden als *generalisierte Kräfte* bezeichnet: es sind die in den Raum der generalisierten Koordinaten (hier Minimalkoordinaten) projizierten Kräfte.

Im Allgemeinen können die generalisierten Kräfte über das Prinzip der virtuellen Arbeit bestimmt werden. Die der Kraft \mathbf{F} entsprechende generalisierte Kraft \mathbf{Q} muss die gleiche virtuelle Arbeit leisten:

$$\delta W = \mathbf{F}^T \delta \mathbf{r} = \mathbf{Q}^T \delta \mathbf{q} \quad (3.3.37)$$

Wegen $\delta \mathbf{r} = \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \delta \mathbf{q}$ folgt unmittelbar:

$$\mathbf{Q} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}} \right)^T \mathbf{F} = \mathbf{J}_T^T \mathbf{F} \quad (3.3.38)$$

Eine analoge Herleitung gilt für Momente über die virtuelle Leistung:

$$\mathbf{Q} = \left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}} \right)^T \mathbf{M} = \mathbf{J}_R^T \mathbf{M} \quad (3.3.39)$$

Mit der Erdbeschleunigung \mathbf{g} folgt für die Gravitationskräfte:

$$\mathbf{F}_g = m_i \mathbf{g} \quad (3.3.40)$$

$$\mathbf{M}_g = m_i \tilde{\mathbf{r}}_{iS_i} \mathbf{g} \quad (3.3.41)$$

Die allgemeine Form der Bewegungsgleichungen lautet damit:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}_u \quad (3.3.42)$$

Hierbei ist \mathbf{M} die Massen-Matrix des Systems und \mathbf{h} der Vektor aller Kräfte ohne \mathbf{Q}_u .

3.3.2 Darstellung als DGL 1. Ordnung

Viele theoretische Verfahren oder numerische Programmpakete oder Bibliotheken erfordern eine Systembeschreibung über eine DGL 1. Ordnung. Die Standardform der Bewegungsgleichungen (3.3.42) ist eine DGL 2. Ordnung, die jedoch leicht als System 1. Ordnung dargestellt werden kann.

Mit der Einführung des Zustandsvektors $\mathbf{x}^T := (\mathbf{q}^T, \dot{\mathbf{q}}^T)$ folgt unmittelbar:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{Q}_u - \mathbf{h}) \end{pmatrix} \quad (3.3.43)$$

Mit $\mathbf{u} = \mathbf{Q}_u$ als Stellgröße folgt die Systemgleichung in Standardform:

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}, \mathbf{u}) \quad (3.3.44)$$

Anmerkung. Im Allgemeinen ist eine explizite Invertierung von \mathbf{M} nicht nötig und nicht sinnvoll. Numerisch effizienter ist die Lösung des Gleichungssystems $\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_u - \mathbf{h}$ nach $\ddot{\mathbf{q}}$.

3.4 LAGRANGE'sche Gleichungen zweiter Art

Die Bewegungsgleichungen des Roboters können auch über die LAGRANGE'schen Gleichungen zweiter Art aufgestellt werden. Diese geben abhängig von der LAGRANGE-Funktion L und den nicht-konservativen generalisierten Kräften Q_{NK} die Bewegungsgleichungen des Systems an:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right)^T - \left(\frac{\partial L}{\partial q} \right)^T = Q_{NK} \quad (3.4.1)$$

Die LAGRANGE-Funktion L ist als $L = T - V$ definiert, wobei T die kinetische Energie und V die Potentielle Energie des Systems ist. Die generalisierten nicht-konservativen Kräfte werden entsprechend (3.3.38) und (3.3.39) berechnet.

Die LAGRANGE'schen Gleichungen (3.4.1) können aus dem Prinzip von D'ALEMBERT hergeleitet werden (Bremer 1988). Alternativ ergeben sie sich aus dem HAMILTON-Prinzip. Dieses besagt, dass das Wirkungsintegral I für die tatsächlich auftretende Bewegung extremal wird:

$$I = \int L dt \rightarrow \text{extrem!} \quad (3.4.2)$$

Über die klassische Variationsrechnung (siehe Kapitel 5) können hieraus unmittelbar die LAGRANGE'sche Gleichungen zweiter Art berechnet werden.

Kinetische und Potentielle Energie Die kinetische Energie des Roboters ist die Summe der Teilenergien der einzelnen Körper:

$$T = \sum_i \left(\frac{1}{2} m_i v_i^T v_i + m_i v_i^T \tilde{\omega}_i r_{iS_i} + \frac{1}{2} \omega_i^T I_i^o \omega_i \right) \quad (3.4.3)$$

Mit $v_i = J_{TO_i} \dot{q}$ und $\omega_i = J_{Ri} \dot{q}$ folgt unmittelbar:

$$T = \frac{1}{2} \dot{q}^T \underbrace{\left(\sum_i m_i J_{TO_i}^T J_{TO_i} + m_i J_{TO_i}^T \tilde{r}_{iS_i}^T J_{Ri} + m_i J_{Ri}^T \tilde{r}_{iS_i} J_{TO_i} + J_{Ri}^T I_i^o J_{Ri} \right)}_M \dot{q} \quad (3.4.4)$$

Potentielle Energie ergibt sich aus der Gravitation und anderen konservativen Kräften, etwa Federkräfte. Im Falle der Gravitation ist das Potential:

$$V = \sum_i (-m_i g^T r_{Si}) \quad (3.4.5)$$

Hierbei können die Gravitationsbeschleunigung g und der Ortsvektor r_{Si} in einem beliebigem KOS aufgestellt werden – es müssen lediglich beide Größen im gleichen System angeschrieben sein. Für die Darstellung anderer Potentiale sowie einer Herleitung der Terme in (3.4.3) sei auf die Grundvorlesung Technische Mechanik und die Literatur verwiesen (z.B. Pfeiffer 1989).

Nach Einsetzen von (3.4.4) und (3.4.5) in die LAGRANGE-Funktion können die für (3.4.1) erforderlichen Differentiationen durchgeführt werden:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right)^T = \frac{d}{dt} (M \dot{q}) = M \ddot{q} + \dot{M} \dot{q} \quad (3.4.6)$$

$$\frac{\partial L}{\partial q}^T = \left[\frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M \dot{q}) \right]^T + \sum_i (+m_i J_{TSi}^T g) \quad (3.4.7)$$

$$= \left[\frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T M \dot{q}) \right]^T + \underbrace{\sum_i (+m_i (J_{TO,i} + \tilde{r}_{iS_i}^T J_{Ri})^T g)}_{Q_G} \quad (3.4.8)$$

Daraus folgt mit (3.4.1) die Bewegungsgleichung in Standardform:

$$\underbrace{M\ddot{q} + \dot{M}\dot{q} - \left[\frac{1}{2} \frac{\partial}{\partial \dot{q}} \left(\dot{q}^T M \dot{q} \right) \right]^T}_{h} - Q_G = Q_{NK} \quad (3.4.9)$$

Aus (3.4.9) und (3.4.4) sind unmittelbar folgende Eigenschaften der Bewegungsgleichungen ersichtlich:

- Die Massen-Matrix ist symmetrisch
- Die Gleichungen sind linear in \ddot{q}
- Die Gleichungen sind maximal quadratisch in \dot{q}
- Die Gleichungen sind allgemein nichtlinear in q

3.5 Diskussion

Offenbar können die Bewegungsgleichungen sowohl über die projektiven NEWTON-EULER Gleichungen (3.3.21) und (3.3.36), als auch über die LAGRANGE'schen Gleichungen zweiter Art (3.4.9) aufgestellt werden. Die hier gezeigte Herleitung lässt den Weg über (3.4.9) sogar deutlich einfacher erscheinen. Tatsächlich sind die LAGRANGE'schen Gleichungen zweiter Art für kleine Systeme bis zu etwa vier Freiheitsgraden sehr effizient. Der Aufwand zur Aufstellung der BGLn steigt jedoch mit der Anzahl der Freiheitsgrade stark. Dies ist durch die Notwendigkeit zur Berechnung partieller und totaler Ableitungen begründet. Dies bedingt für jedes System erneut einen z.T. erheblichen Aufwand. Die Gleichungen können prinzipiell auch mit Hilfe geeigneter Computeralgebra-Programme automatisch erzeugt werden – hier werden jedoch für große Systeme in der Praxis ebenfalls schnell obere Schranken erreicht.

Für größere Systeme ist das Vorgehen über die projektiven NEWTON-EULER Gleichungen zweckmäßiger. Der Formalismus ist sehr gut für die programmtechnische Umsetzung geeignet, die Beiträge der einzelnen Körper können allgemeingültig programmiert werden. Zudem ist die Berechnung durch das rekursive Vorgehen numerisch effizient (Pfeiffer und Reithmeier 1987). Das synthetische Vorgehen über (3.3.21) und (3.3.36) ist daher die erste Wahl in der Roboterdynamik.

3.6 Berücksichtigung weiterer Effekte: Antriebsdynamik

Bisher wurde die Dynamik des starren, baumstrukturierten Mehrkörpersystems betrachtet. Ein Roboter ist jedoch ein mechatronisches Gerät, für dessen Dynamik auch die verwendeten Motoren und Getriebe eine wichtige Rolle spielen. Beispielhaft wird in diesem Abschnitt die Berücksichtigung der Antriebsdynamik für einen bürstenlosen DC-Motor mit vereinfachtem Getriebemodell betrachtet.

Die Bewegungsgleichung des Motors ist durch

$$U = RI + L\dot{I} + k_M\omega_M \quad (3.6.1)$$

mit folgenden Größen gegeben:

U Ankerspannung

I Motorstrom

R Widerstand

L Induktivität

k_M Motorkonstante

ω_M Winkelgeschwindigkeit des Rotors

In vielen Fällen ist die Induktivität sehr klein, sodass die Dynamik des elektrischen Systems gegenüber dem mechanischen vernachlässigt werden kann. Dann gilt näherungsweise:

$$I = \frac{1}{R}(U - k_M \omega_M) \quad (3.6.2)$$

In jedem Fall ist das Motormoment proportional zum Motorstrom:

$$M_M = k_M I \quad (3.6.3)$$

Das Motormoment M_M wirkt über das Getriebe als generalisierte Kraft auf das System. Wir gehen davon aus, dass M_M direkt auf den Gelenkfreiheitsgrad q_i wirkt. Aufgrund der Getriebeübersetzung N_i ist die (relative) Winkelgeschwindigkeit des Rotors $\omega_M = N_i \dot{q}_i$, sodass für die entsprechende generalisierte Kraft gilt:

$$Q_M = \frac{\partial \omega_M}{\partial \dot{q}} M_M = e_i N_i M_M \quad (3.6.4)$$

e_i ist ein Einheitsvektor, der an der i -ten Stelle den Eintrag 1 hat.

Für die in der Robotik üblichen großen Übersetzungen ($N > 50$) und falls der Rotor keine Unwucht hat, dominiert die Trägheit des Rotors durch Drehung relativ zum Stator gegenüber anderen mechanischen Effekten (Kreiselkräfte etc.). In diesem Fall wird die Mechanik des Motorläufers über eine einfache skalare Gleichung beschrieben. Aus Gründen der Einfachheit soll hier von einer viskosen Reibung durch Getriebe und Wälzlager mit einer Reibkonstante b ausgegangen werden. Die BGL ergibt sich dann zu:

$$I_M \dot{\omega}_M + b \omega_M = M_M \quad (3.6.5)$$

Gemeinsam mit (3.6.4) und (3.3.21) ergibt sich die durch die Motoren hinzugekommenen Terme der BGL $\Delta M, \Delta h, Q_u$ zu:

$$\Delta M = \begin{pmatrix} N_1^2 I_{M,1} & 0 & \dots & 0 \\ 0 & N_2^2 I_{M,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & N_n^2 I_{M,n} \end{pmatrix} \quad (3.6.6)$$

$$\Delta h = \begin{pmatrix} N_1^2 b_1 \dot{q}_1 \\ N_2^2 b_2 \dot{q}_2 \\ \vdots \\ N_n^2 b_n \dot{q}_n \end{pmatrix} \quad (3.6.7)$$

$$Q_u = \begin{pmatrix} N_1 M_{M,1} \\ N_2 M_{M,2} \\ \vdots \\ N_n M_{M,n} \end{pmatrix} \quad (3.6.8)$$

Auch wenn die Trägheiten $I_{M,i}$ gegenüber denen der Hauptkörper des Roboters klein sind, ist der Beitrag der Motoren zur Massen-Matrix bei großen Übersetzungen erheblich. Bei einer Übersetzung von $N = 100$ geht die Trägheit des Rotors bereits mit dem Faktor $N^2 = 10000$ in M ein!

Eine höhere Trägheit ist zunächst ungünstig für die Dynamik des Systems. Gleichzeitig kann beobachtet werden, dass bei steigendem N die Massen-Matrix zunehmend diagonal dominant wird, wodurch die Dynamik des Systems weniger stark gekoppelt wird. Für die Implementierung dezentraler Regelungsstrategien ist dies von Vorteil (siehe Kapitel 6).

3.7 Auswertung der Bewegungsgleichungen

3.7.1 Direkte Dynamik

Bei der direkten Dynamik wird aus einem gegebenen Anfangszustand $q(t_a), \dot{q}(t_a)$ zum Zeitpunkt t_a und gegebenen Stellkräften Q_u der zeitliche Verlauf des Zustandsgrößen des Systems ($q(t), \dot{q}(t)$) bestimmt. Die Antriebskräfte Q_u können aus den auch in der Realität eingesetzten Regelungsalgorithmen bestimmt werden. Dann ist Q_u eine Funktion des Zustands und der Zeit: $Q_u = f(q, \dot{q}, t)$. Das Vorgehen zur Lösung der direkten Dynamik ist schematisch in Bild 3.4 dargestellt.

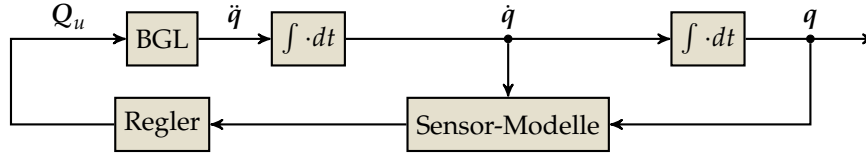


Bild 3.4: Schema zur Berechnung der direkten Dynamik.

Die Zeitintegration der BGL ist nur in einfachsten Fällen geschlossen möglich und muss in aller Regel daher numerisch durchgeführt werden.

Das prinzipielle Vorgehen soll anhand der BGL als DGL 1. Ordnung dargestellt werden (siehe (3.3.44)). Das einfachste Verfahren zur numerischen Lösung ist das explizite Verfahren nach EULER. Die Zeitableitungen werden durch einseitige Finite-Differenzen-Quotienten approximiert:

$$\dot{x} = a(x, u) \approx \frac{x(t_{k+1}) - x(t_k)}{t_{k+1} - t_k} \quad (3.7.1)$$

$$\Rightarrow x(t_{k+1}) \approx x(t_k) + (t_{k+1} - t_k) a(x, u) \quad (3.7.2)$$

Damit kann für diskrete Zeitpunkte t_i jeweils durch Auswertung der BGLn $a(x, t)$ der Zustand zum nächsten Zeitschritt $x(t_{i+1})$ näherungsweise berechnet werden.

In der Praxis sind meist Verfahren höherer Ordnung günstiger, etwa die Runge-Kutta Verfahren. Für eine theoretische Einführung in die numerische Integration der Bewegungsgleichungen sei auf die Literatur (Quarteroni, Sacco und Saleri 2002) und die am Lehrstuhl angebotene Vorlesung „Mehrkörpersimulation“ verwiesen. Für den praktischen Einsatz gibt es mächtige Sammlungen effizienter Integratoren, etwa LSODE/LSODA (Hindmarsh 1983 und <http://www.netlib.org>), oder die in Matlab integrierten Funktionen.

3.7.2 Inverse Dynamik

In der inversen Dynamik ist die Bewegung $q(t), \dot{q}(t), \ddot{q}(t)$ gegeben und es werden die entsprechenden Kräfte gesucht. Im einfachsten Fall werden nur die generalisierten Antriebskräfte Q_u benötigt. Diese können unmittelbar durch Einsetzen in die BGL $M(q)\ddot{q} + h(q, \dot{q}) = Q_u$ berechnet werden. Effizienter ist die direkte Anwendung der projektiven NEWTON-EULER Gleichungen (3.3.21):

$$\sum_{i=1}^{N_{\text{körper}}} \left\{ \begin{pmatrix} J_{TO_i} \\ J_{Ri} \end{pmatrix}^T \begin{pmatrix} \dot{p}_i \\ \dot{L}_{O_i, i} + m_i \tilde{r}_{iS_i} a_i \end{pmatrix} \right\} = \sum_{i=1}^{N_{\text{körper}}} \left\{ \begin{pmatrix} J_{TO_i} \\ J_{Ri} \end{pmatrix}^T \begin{pmatrix} F^e \\ M_{O_i}^e \end{pmatrix} \right\}$$

Dadurch kann auf die aufwändige Berechnung der Massen-Matrix und des h -Vektors verzichtet werden.

Speziell für Auslegungsrechnungen ist die Berechnung der Zwangskräfte erforderlich, da sie zur Dimensionierung der Lager, der Festigkeitsrechnung usf. benötigt werden. Für diese Aufgabe sind die Bewegungsgleichungen (3.3.21), (3.3.42) jedoch nicht geeignet, da die Zwangskräfte durch Projektion eliminiert wurden.

Zur Berechnung der Zwangskräfte ist daher ein Rückgriff auf den Impuls- und Drallsatz für jeden Einzelkörper erforderlich:

$$\dot{p}_i = \sum_k F_{i,k} \quad (3.7.3)$$

$$\dot{L}_{O_i,i} + m_i \tilde{r}_{iS_i} a_i = \sum_k M_{O_i,k} \quad (3.7.4)$$

Wegen des NEWTON'schen Prinzips „actio et reactio“ sind die Zwangskräfte zweier gekoppelter Körper an deren Kopplung gleich. Dadurch sind jeweils zwei der Kräfte und Momente in obiger Gleichung entgegengesetzt gleich. Die Gleichungen (3.7.3), (3.7.4) stellen daher ein lineares gekoppeltes Gleichungssystem in den unbekannten Zwangskräften dar.

Die auf der linken Seite auftretenden Größen können entsprechend dem in Kapitel 2 dargestellten Vorgehen durch eine Vorwärtsrekursion bestimmt werden. Danach kann (3.7.3), (3.7.4) mit einem geeigneten Verfahren gelöst werden.

An dieser Stelle kann jedoch die Baumstruktur des Systems gewinnbringend genutzt werden: Körper, die keine Nachfolger haben, haben maximal 6 unbekannte Zwangskräfte, die unmittelbar aus (3.7.3), (3.7.4) berechnet werden können. Nach Bestimmung der ersten Zwangskräfte gilt die gleiche Argumentation für die im Baum weiter innen liegenden Körper. Somit können die Zwangskräfte ohne Lösung eines gekoppelten Gleichungssystems effizient rekursiv bestimmt werden.

Code- Beispiel

Beispiele zur Programmierung der direkten und inversen Dynamik finden Sie in den begleitenden Matlab-Skripten zur Vorwärtsdynamik (`beispiel_vd.m`) und inversen Dynamik (`beispiel_id.m`).

3.8 Bewegungsgleichungen im Arbeitsraum

Bisher wurde die Dynamik im Konfigurationsraum dargestellt:

$$M\ddot{q} + h = Q_u$$

Diese Darstellung ist natürlich und für die meisten Aufgaben angemessen. Gleichzeitig sind in einer konkreten Aufgabe, die von Roboter auszuführen ist, meist die Bewegungen und Kräfte am Endeffektor relevant, nicht die Gelenkwinkel und Antriebsmomente. Es wird daher eine zu obiger Gleichung analoge Darstellung der Dynamik im Arbeitsraum gesucht:

$$M_w \ddot{w} + h_w = \mathcal{F}_w \quad (3.8.1)$$

Die Größen M_w und h_w sind dabei die zu M und h analogen Größen im Arbeitsraum. Die Kraft \mathcal{F}_w ist die generalisierte Antriebskraft im Arbeitsraum, die entsprechend der Definition der Arbeitsraumkoordinaten w zu bestimmen ist. Aus Gründen der Einfachheit soll im Folgenden angenommen werden, dass $\dim(w) = \dim(q)$ gilt.

Zur Bestimmung der Terme in (3.8.1) steht neben der ursprünglichen BGL (3.3.42) noch die Definition des Arbeitsraums $w(q)$ zur Verfügung. Zweimaliges Differenzieren von $w(q)$ ergibt:

$$\ddot{w} = J_w \ddot{q} + \dot{J}_w \dot{q} \quad (3.8.2)$$

Einsetzen der Gleichung in (3.3.42) ergibt:

$$\underbrace{\left[J_w^{-T} M J_w^{-1} \right]}_{M_w} \ddot{w} + \underbrace{\left[J_w^{-T} h - J_w^{-T} M J_w^{-1} \dot{J}_w \dot{q} \right]}_{h_w} = J_w^{-T} Q_u \quad (3.8.3)$$

Mit der den Arbeitsraumkoordinaten w entsprechenden externen Kraft \mathcal{F}_w und dem Zusammenhang $Q_u = J_w^T \mathcal{F}_w$ aus dem Prinzip der virtuellen Arbeit folgt die gesuchte Gleichung:

$$M_w \ddot{w} + h_w = \mathcal{F}_w \quad (3.8.4)$$

Beispiel zur BGL im Arbeitsraum Es wird der Knickarm-Roboter mit zwei Freiheitsgraden in der Ebene betrachtet (vgl. Bild 3.1), bei dem beide Körper die Länge $l_1 = l_2 = l$ haben. Als generalisierte Koordinaten werden die relativen Gelenkwinkel gewählt. In der Nulllage ($q_1 = q_2 = 0$) liegen beide Körper auf der positiven x -Achse. Die Jacobi-Matrix J_w und Massen-Matrix M des Systems lauten dann:

$$J_w = \begin{pmatrix} -l s q_1 - l s q_{12} & l s q_{12} \\ l c q_1 + l c q_{12} & l c q_{12} \end{pmatrix} \quad (3.8.5)$$

$$M = m \begin{pmatrix} 1/4 l^2 + (5/4 l^2 + l^2 c q_2) & (1/4 l^2 + 1/2 l^2 c q_2) \\ (1/4 l^2 + 1/2 l^2 c q_2) & 1/4 l^2 \end{pmatrix} \quad (3.8.6)$$

Daraus berechnet sich für $q_1 = 0$ die Massen-Matrix im Arbeitsraum M_w und die Inverse der Massen-Matrix zu:

$$M_w(q_1 = 0, q_2) = m \begin{pmatrix} \frac{(c q_2)^2 + 1}{4(-1 + (c q_2)^2)} & \frac{3 c q_2}{4 s q_2} \\ \frac{3 c q_2}{4 s q_2} & \frac{5}{4} \end{pmatrix} \quad (3.8.7)$$

$$M_w^{-1}(q_1 = 0, q_2) = \frac{1}{m} \begin{pmatrix} \frac{-20(s q_2)^2}{(4(c q_2)^2 - 5)} & \frac{12 c q_2 s q_2}{(4(c q_2)^2 - 5)} \\ \frac{12 c q_2 s q_2}{(4(c q_2)^2 - 5)} & \frac{-4((c q_2)^2 + 1)}{(4(c q_2)^2 - 5)} \end{pmatrix} \quad (3.8.8)$$

Das Verhalten in der Randsingularität (Strecklage) kann über den Grenzübergang $q_2 \rightarrow 0$ untersucht werden:

$$\lim_{q_2 \rightarrow 0} M_w(q_1 = 0, q_2) = \begin{pmatrix} \infty & \pm\infty \\ \pm\infty & \frac{5}{4}m \end{pmatrix} \quad (3.8.9)$$

$$\lim_{q_2 \rightarrow 0} M_w^{-1}(q_1 = 0, q_2) = \begin{pmatrix} 0 & 0 \\ 0 & \frac{8}{m} \end{pmatrix} \quad (3.8.10)$$

Die Bedeutung dieses Ergebnisses wird durch Betrachtung der aus einer Kraft F_w resultierenden Beschleunigung \ddot{w} klar:

$$\ddot{w} = M_w^{-1}(F_w) = \begin{pmatrix} 0 & 0 \\ 0 & \frac{8}{m} \end{pmatrix} \begin{pmatrix} F_x \\ F_y \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{8}{m} F_y \end{pmatrix} \quad (3.8.11)$$

Das Ergebnis ist aus der Anschauung heraus unmittelbar plausibel: Die in die Richtung des gestreckten Arms wirkende Kraft F_x führt zu keiner Bewegung und wird durch die Lager aufgenommen. Die vertikale Kraft F_y führt jedoch zu vertikaler Beschleunigung.

3.9 Effiziente Berechnung der Dynamik

In diesem Abschnitt werden kurz Möglichkeiten zur effizienteren Berechnung der inversen und direkten Dynamik angerissen. Für eine detaillierte Darstellung wird auf weiterführende Literatur verwiesen.

3.9.1 Inverse Dynamik

Das bisher dargestellte Vorgehen zur Berechnung der inversen Dynamik erfolgt über das Aufstellen der Bewegungsgleichungen. Der numerische Aufwand wächst hierbei quadratisch mit der Anzahl der Freiheitsgrade n , das Verfahren hat die Ordnung $O(n^2)$. Ein speziell für große Systeme effizienteres Vorgehen ist analog zu dem Verfahren zur Berechnung der Zwangskräfte möglich:

- **Vorwärtsrekursion** Berechnung Ausgehend vom Wurzelkörper zum Endeffektor bzw. den Endkörpern: kinematische Größen (Positionen, Geschwindigkeiten, Beschleunigungen), sowie die Impuls- und Dralländerung berechnet.

- **Rückwärtsrekursion** Berechnung ausgehend von Endeffektor, bzw. Endkörpern zur Wurzel hin: Die am Gelenk wirkenden Kräfte- und Momente aus Impuls- und Drallsatz. Die Antriebskräfte sind hier die z-Komponente der Kraft (bei Schubgelenken) bzw. des Moments (bei Drehgelenken).

Der Aufwand dieses Verfahrens wächst linear in n , es hat die Ordnung $O(n)$. Insbesondere ist die Berechnung der Jacobi-Matrizen und der Massen-Matrix nicht nötig.

3.9.2 Direkte Dynamik

Das bisher dargestellte Vorgehen zur direkten Dynamik wächst kubisch in der Anzahl der Freiheitsgrade, da die Inversion der Massen-Matrix die Ordnung $O(n^3)$ hat. Bereits das Aufstellen der Bewegungsgleichungen hat die Ordnung $O(n^2)$.

Ähnlich dem Vorgehen in der inversen Dynamik kann ein rekursives Vorgehen der Komplexität $O(n)$ angegeben werden. Das Verfahren kann aus Impuls- und Drallsatz für die Einzelkörper und dem Prinzip der virtuellen Arbeit oder virtuellen Leistung hergeleitet werden. Hierbei sind jedoch drei Rekursionen notwendig. Eine Vorwärtsrekursion zur Berechnung der Kinematik, eine Rückwärtsrekursion zur Propagierung der Kraftwirkungen und effektiven Trägheiten, sowie eine letzte Vorwärtsrekursion zur Berechnung der Beschleunigungen \ddot{q} . Hierbei müssen wiederum weder Massen- noch Jacobi-Matrizen berechnet werden. Es gibt leichte Variationen des rekursiven Verfahrens. Für Details sei auf die Literatur verwiesen (Brandl, Johanni und Otter 1986; Featherstone 2008; Walker und Orin 1982).

Kapitel 4

Bahnplanung

4.1 Einleitung

In diesem Abschnitt werden grundlegenden Möglichkeiten zur Beschreibung einer gewünschten Bewegung eines Roboters dargestellt. Zunächst werden zwei grundlegende Begriffe definiert:

Definition 4.1.1. Die *Bahn* ist der Geometrischer Ort der Punkte, entlang derer sich das System bewegt. Sie kann im Arbeits- oder Konfigurationsraum angegeben werden.

Definition 4.1.2. Die *Trajektorie* ist der Zeitverlauf der Zustandsgrößen. Sie beinhaltet die geometrische Bahn gemeinsam mit der zeitlichen Information, wie diese Bahn abgefahren wird.

Die für eine Aufgabe erforderliche Bahn ergibt sich z.T. direkt aus der Anwendung. Beispiele sind das Auftragen von Klebemitteln entlang einer Naht oder Schweißvorgänge.

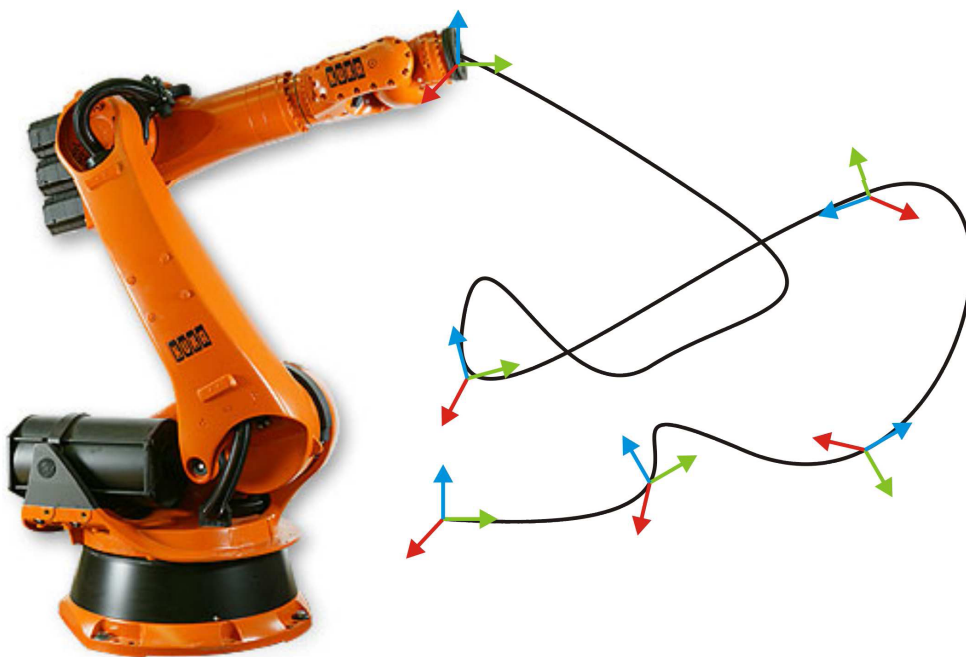


Bild 4.1: Bahn des Endeffektors eines Industrieroboters.

4.1.1 Aufgaben

Die Aufgabe der Bahnplanung sind die Bestimmung der Bahn oder Trajektorie unter gegebenen Randbedingungen. Typische Aufgaben sind:

- Punkt-zu-Punkt Bewegung (PTP):

- Gegeben:
 $q(t_a), q(t_e)$ oder $w(t_a), w(t_e)$
- Alternativ:
 q_a, q_e oder w_a, w_e (ohne zeitliche Information)
- Gesucht: $q(t)$ bzw. $w(t)$ für $t_a \leq t \leq t_e$
- Wegpunkte:
 - Gegeben:
 $q(t_i)$ oder $w(t_i)$ $i = 1, \dots, N$
 - Alternativ:
 q_i, w_i $i = 1, \dots, N$: (ohne zeitliche Information)
 - Gesucht: $q(t)$ bzw. $w(t)$ für $t_a \leq t \leq t_N$

Die Bahnplanung kann prinzipiell im Arbeits- oder Konfigurationsraum angegeben werden.

4.1.2 Anmerkungen zur Planung im Arbeitsraum

Vorteile der Planung im Arbeitsraum sind die

- direkte Vorgabe der gewünschten Werkzeugposition, etc.
- U.U. einfache Kollisionsvermeidung
- Oft einfache Darstellung gewünschter Bahneigenschaften

Der wesentliche Nachteil ist, dass die Bahn im Konfigurationsraum durch Lösung der inversen Kinematik oder einer Regelung im Arbeitsraum bestimmt werden muss. Da die Lösbarkeit der inversen Kinematik bei der Planung in aller Regel nicht als Bedingung berücksichtigt wird, ist die Bahn u.U. nicht ausführbar.

4.1.3 Anmerkungen zur Planung im Gelenkraum

Ein Vorteil der Planung im Gelenkraum ist die Tatsache, dass es nicht zu Singularitäten kommen kann. Es wird in der „natürlichen“ Systembeschreibung geplant, wodurch jede Bahn ausführbar ist (abgesehen von möglichen Kollisionen). Darüber hinaus ist keine Lösung der inverse Kinematik nötig.

Nachteilig ist die Tatsache, dass die Bahn des Endeffektors etwa bei einer PTP-Bewegung im Gelenkraum eine „krumme“ Bahn ausführen wird. Hierdurch kann die Kollisionsvermeidung für den Endeffektor aufwändiger werden. Falls in der Planung noch Ziel-, Start- und Wegpunkte berechnet werden müssen, erfordert dies u.U. doch eine Lösung der inverse Kinematik.

4.2 Darstellung der Bahn

Um konkret nutzbar zu sein, muss die geplante Bahn bzw. Trajektorie mathematisch dargestellt werden. Hierzu bieten sich zahlreiche Möglichkeiten an. Eine allgemeine Anforderung an die Darstellung ist die Stetigkeit bis zur zweiten Ableitung (C^2 -Stetigkeit). Andernfalls wären zur Realisierung Kraftsprünge oder gar Stöße erforderlich. Beides ist physikalisch nicht möglich und würde in der Praxis dazu führen, dass die Antriebsregelung sehr große Kräfte auf das System aufbringt, was vermieden werden sollte.

Geometrische Primitive Zunächst können in einfachen Fällen Geometrische Primitive wie Kreise, Geraden, etc. herangezogen werden. Dies bietet sich insbesondere für Bahnen im Arbeitsraum an.

Trapezförmige Geschwindigkeitsprofile In kommerziellen Antriebsreglern kommen oft trapezförmige Geschwindigkeitsprofile zum Einsatz, wie sie in Bild 4.2 dargestellt sind. Die trapezförmigen Profile selbst sind nicht C^2 -stetig, da es an den Ecken zu Sprüngen in der Beschleunigung kommt. Um dies zu vermeiden kann die Beschleunigung wie in der Abbildung dargestellt „verschliffen“ werden.

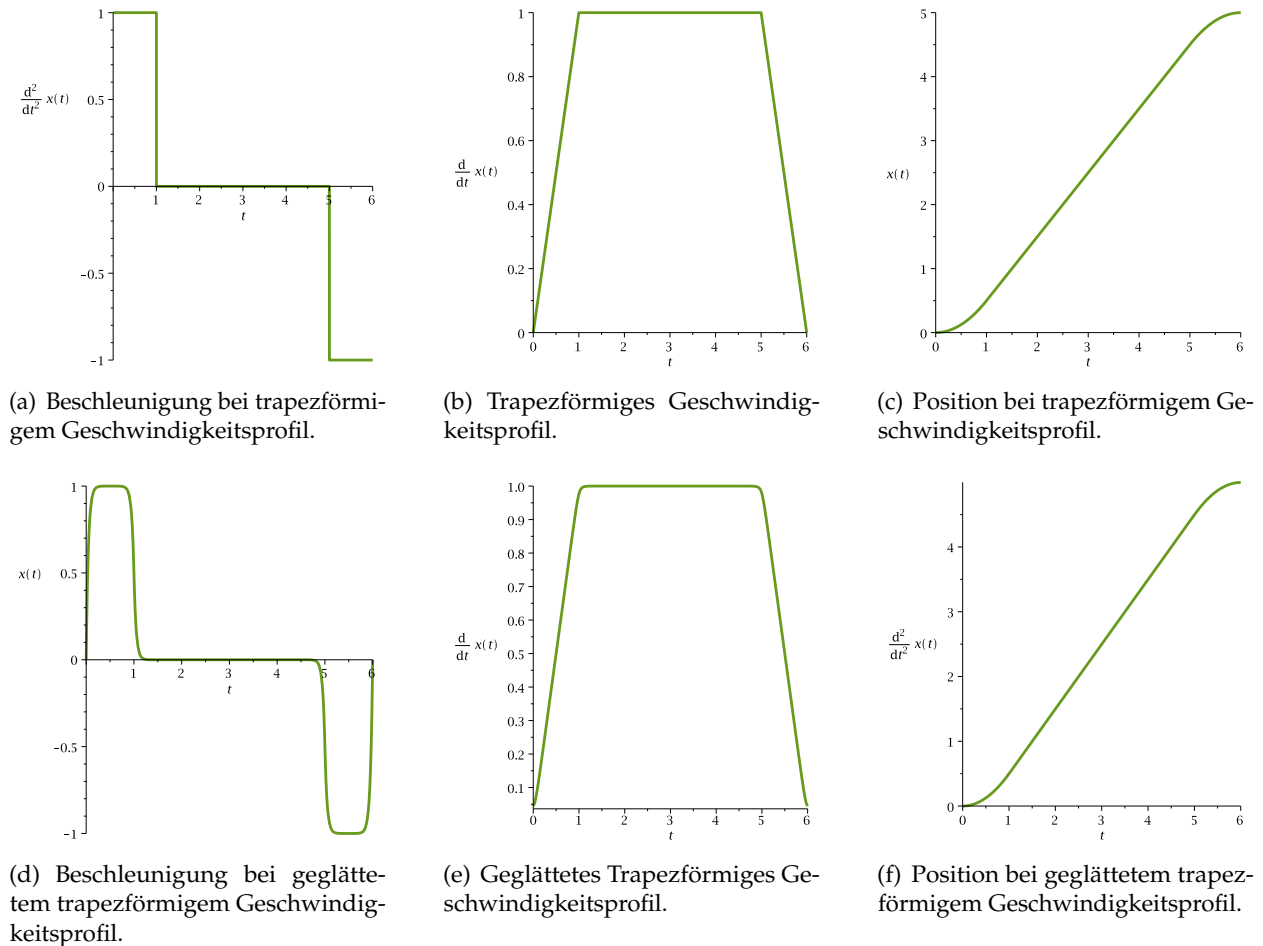


Bild 4.2: Trapezförmige Geschwindigkeitsprofile zur Vorgabe der Bahn.

Polynome Allgemeinere Bahnen können durch Polynome N -ten Grades dargestellt werden:

$$x(t) = \sum_{i=0}^N a_i t^i \quad (4.2.1)$$

Hierbei steht $x(t)$ für eine Komponente von \mathbf{q} oder \mathbf{w} und a_i sind die Parameter des Polynoms. Kritisch an Polynomen zur Darstellung der Bahn ist die Neigung zu Oszillationen für große N . In Abbildung Bild 4.3 sind Interpolationspolynome für unterschiedliche N angegeben. Die zu interpolierenden Punkte sind Funktionswerte der Vorzeichenfunktion σ :

$$\sigma(t) = \begin{cases} -1 & \text{für } t < 0 \\ 0 & \text{für } t = 0 \\ 1 & \text{für } t > 0 \end{cases} \quad (4.2.2)$$

Die Interpolationspolynome sind für $N = 10$ offenbar unbrauchbar, während die interpolierenden kubischen Splines (siehe unten) nahe an den Wegpunkten bleiben.

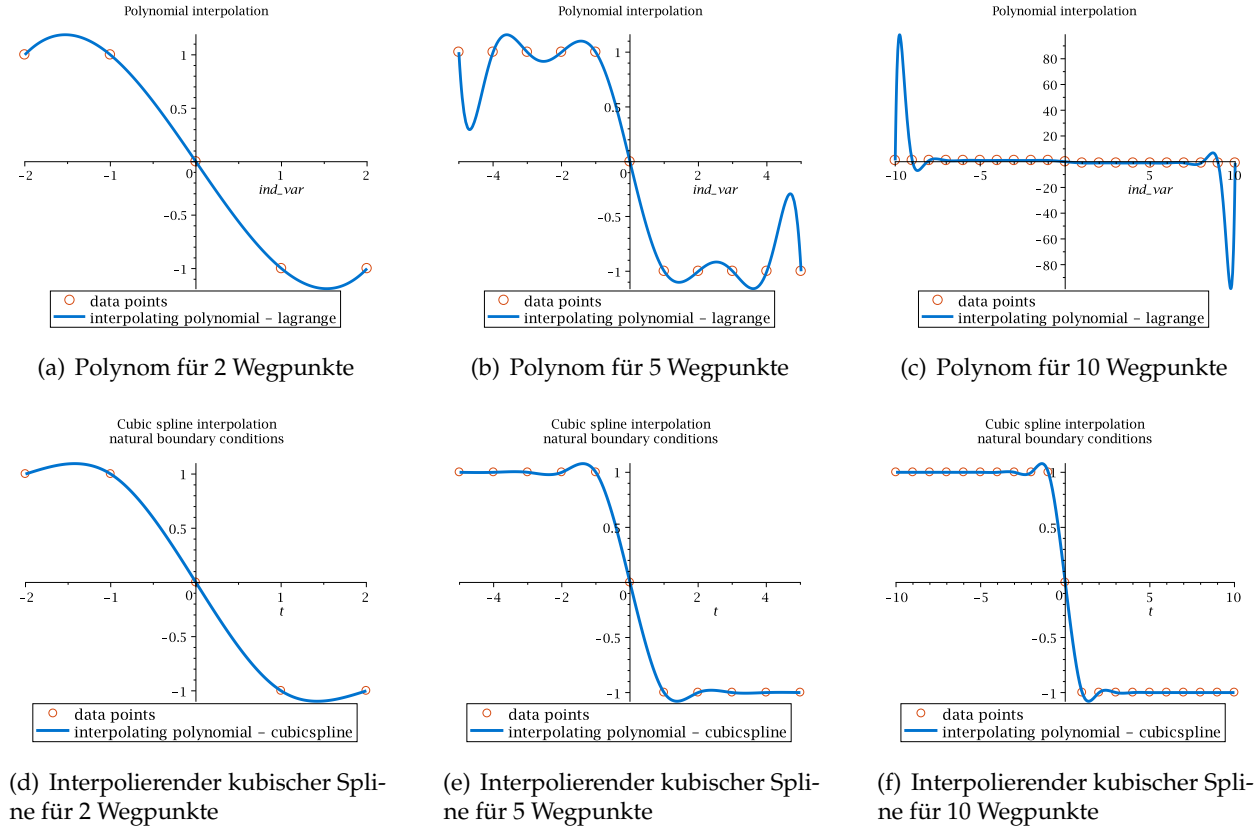


Bild 4.3: Interpolationspolynome und interpolierende kubische Splines für 2, 5 und 10 Wegpunkte.

4.2.1 Kubische Splines

Die Schwierigkeiten bei der Interpolation zahlreicher Punkte durch ein Polynom hoher Ordnung können durch die Verwendung stückweise definierter Funktionen vermieden werden. Eine gut geeignete, oft eingesetzte Variante sind die kubischen Splines.

Bei kubischen Splines wird für jedes Intervall zwischen zwei Wegpunkten p_i ein Polynom dritter Ordnung definiert:

$$s(t) = \begin{cases} s_0(t) & \text{für } t \in [t_0, t_1] \\ s_1(t) & \text{für } t \in [t_1, t_2] \\ \dots & \\ s_{n-2}(t) & \text{für } t \in [t_{n-2}, t_{n-1}] \end{cases} \quad (4.2.3)$$

$$s_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \quad (4.2.4)$$

Damit gibt es $4(n - 1)$ unbekannte Parameter bei n Wegpunkten. Für die Kurve $s(t)$ muss zunächst die Interpolationsbedingung gelten, die n Gleichungen liefert:

$$s(t_i) = p_i \quad \forall i \in \{0, \dots, n - 1\} \quad (4.2.5)$$

Außerdem muss die Stetigkeit auf Lage-, Geschwindigkeits- und Beschleunigungsebene gefordert werden. Dies führt zu weiteren $3(n - 2)$ Gleichungen:

$$s_i(t_{i+1}) = s_{i+1}(t_{i+1}) \quad \forall i \in \{0, \dots, n - 3\} \quad (4.2.6)$$

$$\dot{s}_i(t_{i+1}) = \dot{s}_{i+1}(t_{i+1}) \quad \forall i \in \{0, \dots, n - 3\} \quad (4.2.7)$$

$$\ddot{s}_i(t_{i+1}) = \ddot{s}_{i+1}(t_{i+1}) \quad \forall i \in \{0, \dots, n-3\} \quad (4.2.8)$$

Insgesamt stehen damit $4n - 6$ Gleichungen für $4n - 4$ Unbekannte zur Verfügung. Um eine eindeutige Lösung zu ermöglichen, sind zwei weitere Gleichungen erforderlich. Meist wird noch $\ddot{s}(t_0) = \ddot{s}(t_n) = 0$ gefordert, was zu sogenannten natürlichen Splines führt.

Mit der Abkürzung $h_i := t_{i+1} - t_i$, ergibt sich aus den Stetigkeitsbedingungen für s und \dot{s} :

$$\begin{aligned} a_i &= \frac{1}{6h_i}(\ddot{p}_{i+1} - \ddot{p}_i) \\ b_i &= \frac{1}{2}\ddot{p}_i \\ c_i &= \frac{1}{h_i}(p_{i+1} - p_i) - \frac{h_i}{6}(\ddot{p}_{i+1} + 2\ddot{p}_i) \\ d_i &= p_i \end{aligned} \quad (4.2.9)$$

Stetigkeit für \dot{s} führt schließlich auf folgende Gleichungen für die $n - 2$ Unbekannten $\ddot{p}_i, i = 1 \dots n - 2$:

$$\begin{aligned} h_{i+1}\ddot{p}_{i-1} + 2(h_{i-1} + h_i)\ddot{p}_i + h_i\ddot{p}_{i+1} &= -\frac{6}{h_i}(p_{i+1} - p_i) + \frac{6}{h_{i-1}}(p_i - p_{i-1}) \\ \Leftrightarrow A\ddot{\mathbf{p}} &= \mathbf{r} \end{aligned} \quad (4.2.10)$$

Da die Matrix A tridiagonal ist, kann $\ddot{\mathbf{p}}$ sehr effizient über eine LR-Zerlegung berechnet werden. Die Spline-Koeffizienten werden dann über (4.2.9) berechnet.

Kapitel 5

Optimale Trajektorien

5.1 Einleitung

In aller Regel ist die Trajektorie des Roboters durch die Aufgabenstellung nicht eindeutig festgelegt. Vielmehr sind oft nur die gewünschten Anfangs- und Endpositionen gegeben, eventuell zusätzliche Wegpunkte. Die Fragestellung lautet dann, welche Trajektorie bezüglich der Aufgabenstellung möglichst günstig ist. Können die Vorstellungen einer „guten“ Trajektorie in Form eines skalaren Gütefunktional Φ ausgedrückt werden, kann die diese Funktion minimierende Trajektorie durch Methoden der mathematischen Optimierung bestimmt werden. In diesem Kapitel wird ein Überblick typischer Aufgabenstellungen und Lösungsmethoden gegeben.

Die allgemeine Form der in diesem Kapitel betrachteten Aufgabenstellung lautet: Bestimme diejenige Trajektorie des Zustandsvektors $x(t)$, die alle Randbedingungen erfüllt und das Gütefunktional Φ minimiert. Es sind hier also *Funktionen* zu bestimmen. Man spricht daher von *dynamischer Optimierung*. Im Gegensatz dazu werden in der *statischen Optimierung* endlich viele Variablen durch Minimierung einer Funktion bestimmt.

Zur Lösung dieses Aufgabentyps stehen prinzipiell zwei Wege offen. Die *indirekten Verfahren* suchen zunächst nach Bedingungen an optimale Trajektorien. Diese Optimalitätsbedingungen werden dann gelöst, um die gesuchten Trajektorien zu erhalten. Die klassische Variationsrechnung ist die älteste derartige Methode, sie wird in Abschnitt 5.2 vorgestellt. Anschließend wird auf das PONTYAGIN Minimum Prinzip (PMP)¹ eingegangen, das speziell für technische Aufgabenstellungen gut geeignet ist.

Alternativ kann durch *direkte Verfahren* die Lösung ohne Umweg über die Optimalitätsbedingungen gesucht werden. Hierbei wird zunächst ein allgemeiner Ansatz für die gesuchte Trajektorie gemacht, der endlich viele Unbekannte hat. Dadurch ist das Problem auf eine statische Optimierung zurückgeführt, für die weit entwickelte Algorithmen und effiziente Programme weit verbreitet sind.

5.2 Variationsrechnung

5.2.1 Das einfachste Problem (EULER-LAGRANGE Gleichung)

Die älteste Aufgabenstellung des hier betrachteten Problems lautet:

$$\begin{aligned}\Phi &= \int_{t_a}^{t_e} f(x(t), \dot{x}(t), t) dt \rightarrow \min! \\ x(t_a) &= x_a \\ x(t_e) &= x_e \\ t_a, t_e &\text{ sind gegeben}\end{aligned}\tag{5.2.1}$$

¹Ursprünglich PONTYAGIN-Maximum-Prinzip. Je nach Definition der HAMILTON-Funktion erhält man ein Minimum- oder Maximumprinzip

Im Folgenden wird vorausgesetzt, dass alle Funktionen „hinreichend glatt“ sind, sodass die notwendigen Differentiationen ausgeführt werden können. Die Grundidee zur Lösung der Aufgabe ist wie folgt: Wir betrachten eine kleine mit den Randbedingungen verträgliche Variation δx um die optimale Lösung x_{opt} (siehe Bild 5.1). Eine notwendige Bedingung für ein Optimum ist dann, dass die durch δx verursachte Variation des Gütefunktional verschwindet:

$$\delta\Phi = 0 \quad (5.2.2)$$

Hinreichende Bedingungen können über die zweite Variation hergeleitet werden (siehe Geering 2007). In der Praxis bietet sich wegen der aufwändigen Bedingungen zweiter Ordnung eher eine Plausibilitätsprüfung an, ob tatsächlich ein Minimum oder vielleicht eher ein Maximum vorliegt.

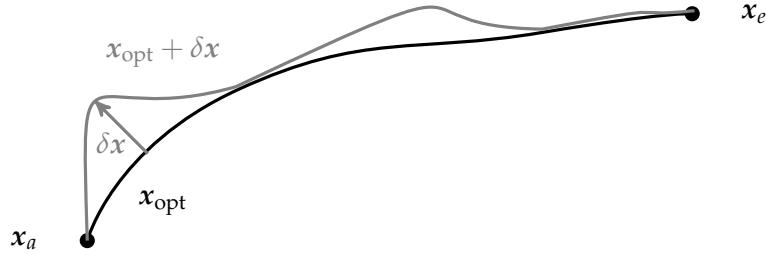


Bild 5.1: Ansatz zur Lösung des Variationsproblems: es wird eine kleine Variation der Trajektorie δx um die optimale Lösung x_{opt} betrachtet.

Die Variation $\delta\Phi$ berechnet sich für (5.2.1) zu:

$$\delta\Phi = \int_{t_a}^{t_e} \frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial \dot{x}} \delta \dot{x} dt \quad (5.2.3)$$

Diese Darstellung ist für die weitere Analyse ungünstig, da die Variationen δx und $\delta \dot{x}$ nicht unabhängig sind. Durch partielle Integration ($\int (uv') = uv - \int u'v$) kann die Variation der Geschwindigkeiten jedoch eliminiert werden:

$$\delta\Phi = \int_{t_a}^{t_e} \frac{\partial f}{\partial x} \delta x - \frac{d}{dt} \left(\frac{\partial f}{\partial \dot{x}} \right) \delta x dt + \left[\frac{\partial f}{\partial \dot{x}} \delta x \right]_{t_a}^{t_b} = \int_{t_a}^{t_e} \left(\frac{\partial f}{\partial x} - \frac{d}{dt} \left(\frac{\partial f}{\partial \dot{x}} \right) \right) \delta x dt + \left[\frac{\partial f}{\partial \dot{x}} \delta x \right]_{t_a}^{t_b} \quad (5.2.4)$$

Gemäß dem Ansatz muss die Variation δx bei t_a und t_b verschwinden, um mit den Randbedingungen verträglich zu sein, sodass der zweite Term verschwindet. Die Forderung $\delta\Phi = 0$ kann (für beliebige δx) nur dann erfüllt werden, wenn der Integrand identisch verschwindet. Es bleibt die EULER-LAGRANGE'sche Differentialgleichung (EL-DGL) der Variationsrechnung:

$$\frac{\partial f}{\partial x} - \frac{d}{dt} \left(\frac{\partial f}{\partial \dot{x}} \right) = 0 \quad (5.2.5)$$

An dieser Stelle wird der Bezug zur Dynamik offensichtlich: Aus dem HAMILTON-Prinzip (3.4.2) folgen über die EL-DGL der Variationsrechnung direkt die LAGRANGE'schen Gleichungen zweiter Art.

Die EL-DGLn liefern n gewöhnliche DGLn zweiter Ordnung für $x \in \mathbb{R}^n$. Gleichzeitig stehen mit $x(t_a) = x_a$, $x(t_e) = x_e$ $2n$ algebraische Gleichungen zur Verfügung, wodurch die $2n$ Integrationskonstanten bestimmt werden können.

Eine analytisch Lösung ist nur in sehr einfachen Fällen möglich. Dann können die Konstanten leicht durch Lösen des algebraischen Gleichungssystems für die Randwerte bestimmt werden. Im Allgemeinen ist jedoch eine numerisch Lösung erforderlich. Dann können die EL-DGLn *nicht* durch eine einfache Zeitintegration bestimmt werden, da die $\dot{x}(t_a)$ unbekannt sind. Es handelt sich um ein *Zweipunkt-Randwertproblem* (ZPRWP), zu dessen Lösung spezielle Verfahren erforderlich sind.

5.2.2 Probleme mit Nebenbedingungen

In vielen Fällen ist die Aufgabenstellung (5.2.1) zu einfach, da neben den Anfangs- und Endwerten noch zusätzliche Nebenbedingungen in Gleichungsform vorhanden sind. Diese können über die Methode der LAGRANGE-Multiplikatoren λ berücksichtigt werden. Im Folgenden werden die Gleichungen zur Behandlung derartiger Probleme kurz zusammengestellt.

Integrale Nebenbedingungen Bei Problemen der Form

$$\begin{aligned}\Phi &= \int_{t_a}^{t_e} f(x, \dot{x}, t) dt \rightarrow \min! \\ \int g(x) dt &= l, \quad l = \text{konstant} \\ x(t_a) &= x_a \\ x(t_e) &= x_e\end{aligned}\tag{5.2.6}$$

kann die Lösung mit Hilfe der Lagrange-Funktion L

$$L(x, \dot{x}, \lambda, t) = f(x, \dot{x}, t) + \lambda^T g(x)\tag{5.2.7}$$

über die EL-DGL für L berechnet werden

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) = 0.\tag{5.2.8}$$

Die Lagrange-Multiplikationen λ sind hier Konstanten und müssen so gewählt werden, dass die zusätzliche Nebenbedingung $\int g(x) dt = l$ erfüllt wird.

Algebraische Nebenbedingungen Bei Problemen der Form

$$\begin{aligned}\Phi &= \int_{t_a}^{t_e} f(x, \dot{x}, t) dt \rightarrow \min! \\ g(x) &= 0 \\ x(t_a) &= x_a \\ x(t_e) &= x_e\end{aligned}\tag{5.2.9}$$

kann die Lösung mit Hilfe der Lagrange-Funktion L

$$L(x, \dot{x}, \lambda, t) = f(x, \dot{x}, t) + \lambda(t)^T g(x)\tag{5.2.10}$$

über die EL-DGL für L berechnet werden. Die Lagrange-Multiplikationen $\lambda(t)$ sind hierbei jedoch Funktionen der Zeit und müssen derart gewählt werden, dass die Nebenbedingung $g(x(t)) = 0$ für alle $t \in [t_a, t_e]$ erfüllt ist.

Differentialgleichungsnebenbedingungen Bei Problemen der Form

$$\begin{aligned}\Phi &= \int_{t_a}^{t_e} f(x, \dot{x}, t) dt \rightarrow \min! \\ \dot{x} &= g(x) \\ x(t_a) &= x_a \\ x(t_e) &= x_e\end{aligned}\tag{5.2.11}$$

kann die Lösung wiederum mit Hilfe der Lagrange-Funktion L

$$L(x, \dot{x}, \lambda, t) = f(x, \dot{x}, t) + \lambda(t)^T (\dot{x} - g(x))\tag{5.2.12}$$

über die EL-DGL für L berechnet werden. Die Lagrange-Multiplikationen $\lambda(t)$ sind wiederum Funktionen der Zeit.

5.3 Das PONTRYAGIN Minimum Prinzip

5.3.1 Einleitung

In vielen technischen Fällen treten nicht nur Nebenbedingungen in Gleichungsform auf, sondern auch Ungleichungsnebenbedingungen. Die häufigsten derartigen Bedingungen sind Stellgrößenbeschränkungen: maximale und minimale Drehmomente von Antrieben in Robotern, maximaler und minimaler Schub der Triebwerke einer Rakete usw. Auch tritt für technische Systeme in aller Regel eine Nebenbedingung an mögliche Lösungen in Form der BGL auf:

$$\dot{x} = a(x, u, t)$$

Die Gleichung zeigt auch, dass im technischen Fall zwischen Zustandsgrößen x und Stellgrößen u unterschieden wird. Dabei ist \dot{x} aufgrund der BGL nicht länger frei variierbar, sondern ergibt sich aus den Stellgrößen u . Eine für diese technischen Anforderungen angepasste Methode wurde in den 1950er Jahren von PONTRYAGIN entwickelt. Wie in der Variationsrechnung gehen wir davon aus, dass alle Funktionen hinreichend glatt sind und das Problem nicht singulär ist (für Details siehe Geering 2007).

5.3.2 Problemstellung

Die durch das PMP angesprochene Problemstellung lautet:

$$\begin{aligned} \Phi &= \Phi_e(t_e) + \int_{t_a}^{t_e} f(x, u, t) dt \rightarrow \min! \\ x(t_a) &= x_a \\ x(t_e) &= x_e \\ \dot{x} &= a(x, u, t) \quad \forall t \in [t_a, t_e] \\ u &\in \mathcal{U} \quad (\text{Beschränkung für } u, \text{ z.B. } u_{\min} \leq u \leq u_{\max}) \\ t_a &\text{ ist fest} \quad t_e \text{ kann frei sein} \end{aligned} \tag{5.3.1}$$

Gegenüber den Aufgabenstellungen der Variationsrechnung wurde hier ein Kostenterm am Ende der Trajektorie Φ_e hinzugefügt. Diese Erweiterung bedeutet keine wesentlichen Komplikationen und kommt technischen Aufgabenstellungen entgegen. Ein Optimum liegt vor, falls für alle zulässigen Variationen das Gütefunktional zunimmt:

$$\delta\Phi \geq 0 \tag{5.3.2}$$

5.3.3 Optimalitätsbedingungen

Ausgangspunkt der Herleitung ist die Berücksichtigung der Nebenbedingungen über LAGRANGE-Multiplikatoren. Das erweiterte Gütefunktional $\bar{\Phi}$ lautet:

$$\begin{aligned} \bar{\Phi} &= \Phi_e(t_e) + \int_{t_a}^{t_e} [f(x, u, t) + \lambda^T(a(x, u) - \dot{x})] dt \\ &\quad + \lambda_a^T(x_a - x(t_a)) + \lambda_e^T(x_e - x(t_e)) \\ &= \Phi_e(t_e) + \int_{t_a}^{t_e} \underbrace{[f + \lambda^T a - \lambda^T \dot{x}]}_H dt + \lambda_a^T(x_a - x(t_a)) + \lambda_e^T(x_e - x(t_e)) \end{aligned} \tag{5.3.3}$$

Die neu eingeführte Funktion $H = f + \lambda^T a$ wird als HAMILTON-Funktion bezeichnet.

Zur übersichtlicheren Bestimmung der Variation des erweiterten Gütefunktionals wird es in Terme mit Integral Φ_I und solche ohne Φ_R aufgespalten:

$$\delta\bar{\Phi} = \underbrace{\delta \int_{t_a}^{t_e} [H - \lambda^T \dot{x}] dt}_{\Phi_I} + \underbrace{\delta \left(\Phi_e(t_e) + \lambda_a^T(x_a - x(t_a)) + \lambda_e^T(x_e - x(t_e)) \right)}_{\Phi_R} \tag{5.3.4}$$

Die Variation der Integralterme berechnet sich zu:

$$\delta\Phi_I = \int_{t_a}^{t_e} \left[\frac{\partial H}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial H}{\partial \boldsymbol{\lambda}} \delta \boldsymbol{\lambda} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} - \delta \boldsymbol{\lambda}^T \dot{\mathbf{x}} - \boldsymbol{\lambda}^T \delta \dot{\mathbf{x}} \right] dt + (H - \boldsymbol{\lambda}^T \dot{\mathbf{x}})|_{t_e} \delta t_e \quad (5.3.5)$$

Der letzte Term mit δt_e beschreibt die Änderung des Integrals, wenn das Integrationsintervall länger oder kürzer wird. Die Variation der Geschwindigkeiten $\delta \dot{\mathbf{x}}$ kann wiederum über partielle Integration eliminiert werden:

$$\begin{aligned} \delta\Phi_I = \int_{t_a}^{t_e} & \left[\left(\frac{\partial H}{\partial \mathbf{x}} + \dot{\boldsymbol{\lambda}}^T \right) \delta \mathbf{x} + \left(\frac{\partial H}{\partial \boldsymbol{\lambda}} - \dot{\mathbf{x}}^T \right) \delta \boldsymbol{\lambda} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \right] dt \\ & - (\boldsymbol{\lambda}^T \delta \mathbf{x})|_{t_e} + (\boldsymbol{\lambda}^T \delta \mathbf{x})|_{t_a} + (H - \boldsymbol{\lambda}^T \dot{\mathbf{x}})|_{t_e} \delta t_e \end{aligned} \quad (5.3.6)$$

Die Variation der übrigen Terme ergibt sich zu:

$$\delta\Phi_R = \frac{\partial \Phi_e}{\partial t_e} \Big|_{t_e} \delta t_e + \delta \boldsymbol{\lambda}_a^T (\mathbf{x}_a - \mathbf{x}(t_a)) - \boldsymbol{\lambda}_a^T \delta \mathbf{x}(t_a) + \delta \boldsymbol{\lambda}_e^T (\mathbf{x}_e - \mathbf{x}(t_e)) - \boldsymbol{\lambda}_e^T (\delta \mathbf{x}(t_e) + \dot{\mathbf{x}}(t_e) \delta t_e) \quad (5.3.7)$$

Der letzte Term ($\dot{\mathbf{x}} \delta t_e$) ist wiederum notwendig, um die Änderung des Endwerts $\mathbf{x}(t_e)$ durch Variation der Endzeit t_e zu berücksichtigen.

Zusammenfassen der beiden Terme liefert die Optimalitätsbedingung $\delta\Phi \geq 0$ in Abhängigkeit von den Variationen $\delta \mathbf{x}, \delta \boldsymbol{\lambda}, \delta \mathbf{u}$:

$$\begin{aligned} 0 \leq \delta\Phi = \int_{t_a}^{t_e} & \left[\underbrace{\left(\frac{\partial H}{\partial \mathbf{x}} + \dot{\boldsymbol{\lambda}}^T \right) \delta \mathbf{x}}_{\langle 1 \rangle} + \underbrace{\left(\frac{\partial H}{\partial \boldsymbol{\lambda}} - \dot{\mathbf{x}}^T \right) \delta \boldsymbol{\lambda}}_{\langle 2 \rangle} + \underbrace{\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u}}_{\langle 3 \rangle} \right] dt \\ & + \underbrace{\delta \boldsymbol{\lambda}_a^T (\mathbf{x}_a - \mathbf{x}(t_a))}_{\langle 4 \rangle} + \underbrace{\delta \boldsymbol{\lambda}_e^T (\mathbf{x}_e - \mathbf{x}(t_e))}_{\langle 5 \rangle} \\ & + \underbrace{\left[(\boldsymbol{\lambda} - \boldsymbol{\lambda}_a)^T \delta \mathbf{x} \right] \Big|_{t_a}}_{\langle 6 \rangle} - \underbrace{\left[(\boldsymbol{\lambda} + \boldsymbol{\lambda}_e)^T (\delta \mathbf{x} + \dot{\mathbf{x}} \delta t_e) \right] \Big|_{t_e}}_{\langle 7 \rangle} \\ & + \underbrace{\left(H + \frac{\partial \Phi_e}{\partial t} \right) \Big|_{t_e} \delta t_e}_{\langle 8 \rangle} \end{aligned} \quad (5.3.8)$$

Die Optimalitätsbedingungen ergeben sich über die Tatsache, dass (5.3.8) für alle zulässigen Variationen gelten muss. Setzt man stets alle Variationen außer einer zu Null, so folgt dass die mit dieser Größe multiplizierten Terme verschwinden müssen. Der Term $\langle 6 \rangle$ verschwindet dabei stets, da nach Voraussetzung $\delta \mathbf{x}(t_a) = \mathbf{0}$ gilt. Ebenso muss Term $\langle 7 \rangle$ verschwinden, da an der (u.U. um δt_e modifizierten!) Endzeit die Variation von \mathbf{x} verschwinden muss².

Setzt man zunächst alle Variationen außer $\delta \mathbf{x}$ zu Null, dann alle außer $\delta \boldsymbol{\lambda}$, ergeben sich die HAMILTON kanonischen DGLn (Teile $\langle 1 \rangle$ und $\langle 2 \rangle$ aus (5.3.8)):

$$\dot{\mathbf{x}}^T = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{a} \quad (5.3.9)$$

$$\dot{\boldsymbol{\lambda}}^T = -\frac{\partial H}{\partial \mathbf{x}} \quad (5.3.10)$$

Aus den Lagrange-Multiplikatoren der Anfangs- und Endbedingungen ergeben sich die ursprünglichen Randbedingungen (Teile $\langle 4 \rangle$ und $\langle 5 \rangle$ aus (5.3.8)):

$$\mathbf{x}(t_a) = \mathbf{x}_a \quad (5.3.11)$$

²Die Variation $\delta \mathbf{x}(t_e)$ beinhaltet die Modifikation der Endzeit um δt_e noch nicht. Diese wird über $\dot{\mathbf{x}} \delta t_e$ berücksichtigt.

$$\mathbf{x}(t_e) = \mathbf{x}_e \quad (5.3.12)$$

Ist die Endzeit t_e frei, so ist auch die Variation δt_e beliebig und somit ((Teil $\langle 8 \rangle$ (5.3.8)))

$$\left(\frac{\partial \Phi_e}{\partial t} + H \right) \Big|_{t_e} = 0 \quad (5.3.13)$$

Im Falle einer fest vorgegebenen Endzeit kann der Ausdruck $\left(\frac{\partial \Phi_e}{\partial t} + H \right) \Big|_{t_e}$ jedoch beliebig sein, da $\delta t_e = 0$ gilt.

Die Variation der Stellgröße $\delta \mathbf{u}$ ist im Allgemeinen *nicht* beliebig, da stets $\mathbf{u} \in \mathcal{U}$ gelten muss. Falls jedoch $\mathcal{U} = \mathbb{R}^m$ gilt, oder \mathbf{u} im Inneren der Menge \mathcal{U} liegt, ist $\delta \mathbf{u}$ beliebig, sodass aus $\langle 3 \rangle$ in (5.3.8) unmittelbar

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \quad (5.3.14)$$

folgt. Falls jedoch \mathbf{u} am Rand von \mathcal{U} liegt, ist $\delta \mathbf{u}$ nicht frei, sondern muss ins Innere der Menge \mathcal{U} zeigen. Für alle zulässigen $\delta \mathbf{u}$ gilt dann

$$\frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \geq 0 \quad (5.3.15)$$

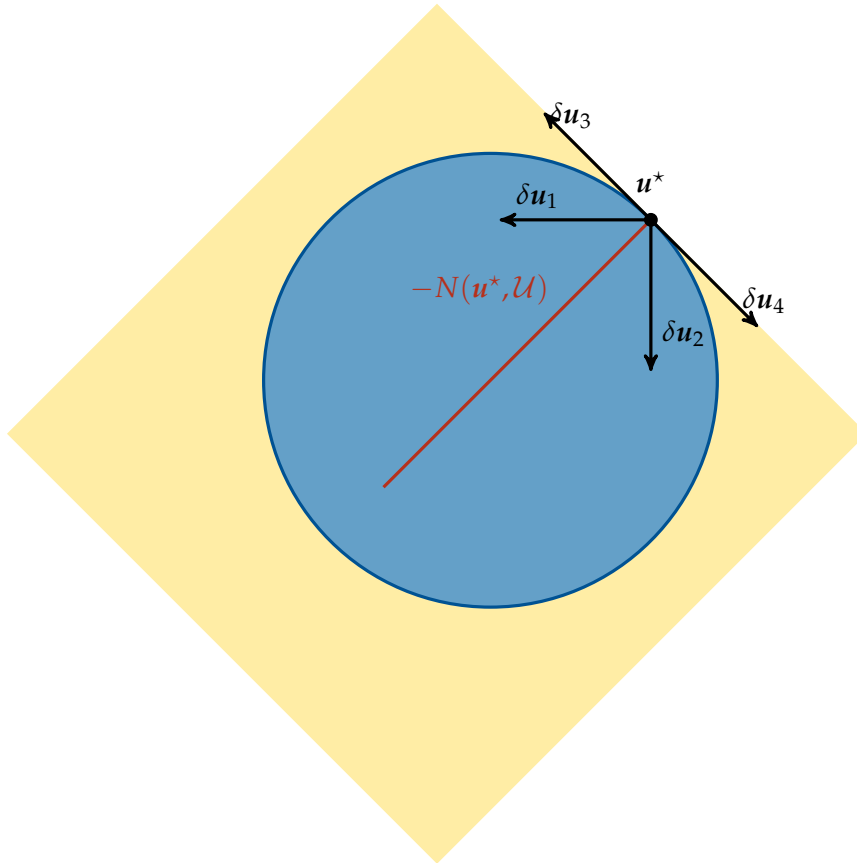


Bild 5.2: Veranschaulichung der Minimumbedingung des PMP (Erläuterungen siehe Text).

Die geometrische Bedeutung dieser Forderung ist in Bild 5.2 dargestellt. Die Pfeile $\delta \mathbf{u}_i$ symbolisieren erlaubte Variationen der Stellgröße am gezeigten Punkt \mathbf{u}^* . Die Menge der erlaubten Variationen bildet in diesem Fall einen Halbraum, der durch die Tangente an den Punkt begrenzt wird (gelb angedeutete Fläche). Im Allgemeinen ergibt sich der *Tangentialkegel* $T(\mathbf{u}^*, \mathcal{U})$ an die Menge \mathcal{U} bei \mathbf{u}^* . Die Ungleichung (5.3.15) bedeutet geometrisch, dass der Winkel zwischen $\frac{\partial H}{\partial \mathbf{u}}$ und $\delta \mathbf{u}$ zwischen $+\pi/2$ und $-\pi/2$ liegen muss. Für alle $\delta \mathbf{u} \in T(\mathbf{u}^*, \mathcal{U})$ kann dies nur gelten, wenn

$-\frac{\partial H}{\partial u}$ im Normalenkegel $N(\mathbf{u}^*, \mathcal{U})$ bei \mathbf{u}^* liegt, bzw. $\frac{\partial H}{\partial u} \in -N(\mathbf{u}^*, \mathcal{U})$. Da $\frac{\partial H}{\partial u}$ gerade die Richtung des steilsten Aufstiegs von H ist, bedeutet (5.3.15) ein lokales Minimum von H bezüglich \mathbf{u} . Für eine optimale Trajektorie muss folglich die HAMILTON-Funktion minimal werden:

$$H(\mathbf{x}_{\text{opt}}, \mathbf{u}_{\text{opt}}, \boldsymbol{\lambda}_{\text{opt}}, t) \leq H(\mathbf{x}_{\text{opt}}, \mathbf{u}, \boldsymbol{\lambda}_{\text{opt}}, t) \quad (5.3.16)$$

Zusammengefasst lauten die Optimalitätsbedingungen:

1. HAMILTON'sche kanonische DGLn:

$$\dot{\mathbf{x}}^T = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{a} \quad (5.3.17)$$

$$\dot{\boldsymbol{\lambda}}^T = -\frac{\partial H}{\partial \mathbf{x}} \quad (5.3.18)$$

2. Randwerte

$$\mathbf{x}(t_a) = \mathbf{x}_a \quad (5.3.19)$$

$$\mathbf{x}(t_e) = \mathbf{x}_e \quad (5.3.20)$$

3. Die Stellgröße minimiert die HAMILTON-Funktion:

$$H(\mathbf{x}_{\text{opt}}, \mathbf{u}_{\text{opt}}, \boldsymbol{\lambda}_{\text{opt}}, t) \leq H(\mathbf{x}_{\text{opt}}, \mathbf{u}, \boldsymbol{\lambda}_{\text{opt}}, t)$$

4. Endzeit

- Bei freier, d.h. nicht vorgegebener Endzeit gilt: $\left(\frac{\partial \Phi_e}{\partial t} + H\right)\Big|_{t_e} = 0$
- Bei einer fest vorgegebenen Endzeit kann $\left(\frac{\partial \Phi_e}{\partial t} + H\right)\Big|_{t_e}$ beliebige Werte annehmen.

5.3.4 Allgemeine Lösungsmethode nach dem PMP

Die Lösung eines Optimierungsproblems mit Hilfe des PMP kann im Allgemeinen nach folgendem Schema durchgeführt werden:

1. Die HAMILTON-Funktion $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = f(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}(t)^T \mathbf{a}(\mathbf{x}, \mathbf{u}, t)$ aufstellen.
2. Minimierende Steuergröße \mathbf{u}_{opt} allgemein als Funktion von $(\mathbf{x}, \boldsymbol{\lambda}, t)$ berechnen (u.U. numerisch):

$$\mathbf{u}_{\text{opt}}(\mathbf{x}, \boldsymbol{\lambda}, t) = \arg \min_{\mathbf{u} \in \mathcal{U}} H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) \quad (5.3.21)$$

3. HAMILTON'sche kanonische DGLn berechnen
4. \mathbf{u}_{opt} aus 2. in DGLn aus 3. einsetzen
5. Lösen der DGLn aus 4. mit gegebenen Randbedingungen

5.3.5 Beispiel

Das Vorgehen zur Bestimmung optimaler Bahnen wird in diesem Abschnitt beispielhaft für die zeitoptimale Bahn einer Punktmasse mit Stellgrößenbeschränkung gezeigt. Die Bewegungsgleichungen der Punktmasse mit Masse $m = 1$ sind gleich einem doppelten Integrator. Die Problemstellung kann folgendermaßen formuliert werden:

$$\Phi = \int_0^T 1 \, dt \rightarrow \min! \quad (5.3.22)$$

$$\frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ u \end{pmatrix} \quad (5.3.23)$$

$$x_1(0) = 0 \quad (5.3.24)$$

$$x_1(T) = x_e \quad (5.3.25)$$

$$x_2(0) = 0 \quad (5.3.26)$$

$$x_2(T) = 0 \quad (5.3.27)$$

$$u \in [-u_m, u_m] \quad (5.3.28)$$

Die Variable u ist die auf die Masse wirkende Kraft, x_1 die Position, x_2 die Geschwindigkeit, T die (unbekannte) Endzeit, u_m die maximal mögliche Kraft und x_e die gewünschte Endposition.

1. Schritt: Aufstellen der Hamilton-Funktion

$$H = f + \lambda^T a = 1 + \lambda_1 x_2 + \lambda_2 u \quad (5.3.29)$$

2. Schritt: Stellgröße u aus Minimumsprinzip als Funktion von λ und x bestimmen

$$u = \arg \min \{1 + \lambda_1 x_2 + \lambda_2 u\} u \in [-u_m, u_m] = \begin{cases} -u_m & , \lambda_2 > 0 \\ u_m & , \lambda_2 < 0 \end{cases} \quad (5.3.30)$$

Im Fall $\lambda_2 = 0$ kann u innerhalb der Beschränkungen frei gewählt werden, da H dann nicht von u abhängt. Mit der Vorzeichenfunktion³

$$\operatorname{sgn}(x) = \begin{cases} 1 & , x > 0 \\ [-1, 1] & , x = 0 \\ -1 & , x < 0 \end{cases} \quad (5.3.31)$$

folgt

$$u = -u_m \operatorname{sgn}(\lambda_2) \quad (5.3.32)$$

3. Schritt: Kanonische Differentialgleichungen

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \quad (5.3.33)$$

$$\Rightarrow \begin{pmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -\lambda_1 \end{pmatrix} \quad (5.3.34)$$

Die Lösung ergibt sich direkt durch Integration zu

$$\lambda_2 = c_1 t + c_2, \quad (5.3.35)$$

(c_i sind Integrationskonstanten).

$$\dot{x} = \frac{\partial H}{\partial \lambda} \quad (5.3.36)$$

$$\Rightarrow \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ u \end{pmatrix} \quad (5.3.37)$$

Mit (5.3.32) und (5.3.35) folgt:

$$\dot{x}_2 = -u_m \operatorname{sgn}(c_1 t + c_2). \quad (5.3.38)$$

Mit dem Zusammenhang $\int \operatorname{sgn}(ax + b) dx = (x + b/a) \operatorname{sgn}(ax + b)$ folgt

$$x_2 = -u_m \left(t + \frac{c_2}{c_1}\right) \operatorname{sgn}(c_1 t + c_2) + c_3 \quad (5.3.39)$$

³In vielen Programmiersprachen, wie z.B. Matlab oder C++ ist $\operatorname{sgn}(0) = 0$.

Die Konstante c_3 folgt aus der Randbedingung bei $t = 0$

$$x_2(0) = 0 = -u_m \frac{c_2}{c_1} \operatorname{sgn}(c_2) + c_3 \quad (5.3.40)$$

$$\Rightarrow c_3 = u_m \frac{c_2}{c_1} \operatorname{sgn}(c_2) \quad (5.3.41)$$

Die Randbedingung bei $t = T$ liefert:

$$x_2(T) = 0 = -u_m \left(T + \frac{c_2}{c_1}\right) \operatorname{sgn}(c_1 T + c_2) + u_m \frac{c_2}{c_1} \operatorname{sgn}(c_2) \quad (5.3.42)$$

$$\Rightarrow 0 = -T \operatorname{sgn}(c_1 T + c_2) + \frac{c_2}{c_1} (-\operatorname{sgn}(c_1 T + c_2) + \operatorname{sgn}(c_2)) \quad (5.3.43)$$

$$\Rightarrow \frac{c_2}{c_1} = T \frac{\operatorname{sgn}(c_1 T + c_2)}{-\operatorname{sgn}(c_1 T + c_2) + \operatorname{sgn}(c_2)} \quad (5.3.44)$$

Es muss $c_1 T + c_2 \neq 0$ gelten, da sonst kein Vorzeichenwechsel in $u(t)$ auftritt (siehe (5.3.38)), wodurch die Masse nicht beschleunigt und wieder abgebremst werden kann (vgl. Randbedingungen). Außerdem muss wegen (5.3.44) $\operatorname{sgn}(c_1 T + c_2) \neq \operatorname{sgn}(c_2)$ gelten, da sonst (für $T > 0$) $\frac{c_2}{c_1} \rightarrow \infty$, wodurch wieder kein Vorzeichenwechsel möglich ist (vgl. (5.3.38)). Damit folgt aus (5.3.44)⁴:

$$\frac{c_2}{c_1} = -\frac{T}{2} \quad (5.3.45)$$

Damit lässt sich c_1 aus obigen Gleichungen eliminieren. Mit (5.3.45) folgt:

$$\operatorname{sgn}(c_1 t + c_2) = \operatorname{sgn}\left(\frac{-2c_2}{T}\left(t - \frac{T}{2}\right)\right) \quad (5.3.46)$$

$$\stackrel{T \geq 0}{=} -\operatorname{sgn}\left(c_2\left(t - \frac{T}{2}\right)\right) \quad (5.3.47)$$

Die Annahme $T > 0$ folgt daraus, dass die Anfangszeit zu 0 gewählt wurde (Zeit kann nur wachsen). Damit folgt aus (5.3.39)

$$x_2 = \operatorname{sgn}(c_2) u_m \left[\left(t - \frac{T}{2}\right) \operatorname{sgn}\left(t - \frac{T}{2}\right) - \frac{T}{2} \right] \quad (5.3.48)$$

$$\Rightarrow x_1 = \operatorname{sgn}(c_2) u_m \left[\frac{1}{2} \left(t - \frac{T}{2}\right)^2 \operatorname{sgn}\left(t - \frac{T}{2}\right) - \frac{T}{2} t \right] + c_4 \quad (5.3.49)$$

Die Randbedingung bei $t = 0$ liefert c_4 :

$$x_1(0) = 0 = \operatorname{sgn}(c_2) u_m \left[\frac{1}{2} \left(-\frac{T}{2}\right)^2 \operatorname{sgn}\left(-\frac{T}{2}\right) \right] + c_4 \quad (5.3.50)$$

$$\Rightarrow c_4 = \operatorname{sgn}(c_2) u_m \frac{T^2}{8} \quad (5.3.51)$$

Die Endzeit folgt aus der Randbedingung bei $t = T$:

$$x_1(T) = x_e = \operatorname{sgn}(c_2) u_m \left[\frac{1}{2} \left(\frac{T}{2}\right)^2 \operatorname{sgn}\left(\frac{T}{2}\right) - \frac{T}{2} T \right] + \operatorname{sgn}(c_2) u_m \frac{T^2}{8} \quad (5.3.52)$$

$$= -\operatorname{sgn}(c_2) u_m \frac{T^2}{4} \quad (5.3.53)$$

$$\Rightarrow T^2 = \frac{-4x_e}{\operatorname{sgn}(c_2) u_m} \quad (5.3.54)$$

⁴Da $\operatorname{sgn}(c_1 T + c_2) \neq \operatorname{sgn}(c_2)$ können zwei Fälle für (5.3.44) auftreten, $\operatorname{sgn} c_2 = 1$ oder $\operatorname{sgn} c_2 = -1$. Für beide Fälle folgt dasselbe Ergebnis.

$$\Rightarrow T = \pm 2 \sqrt{\frac{-x_e}{\operatorname{sgn}(c_2) u_m}} \quad (5.3.55)$$

$$= 2 \sqrt{\frac{\|x_e\|}{u_m}} \quad (5.3.56)$$

In der letzten Gleichung wurde $T > 0$ ausgenutzt, sowie die Tatsache, dass für reelwertige Lösungen offenbar $\operatorname{sgn}(x_e) = -\operatorname{sgn}(c_2)$ gelten muss.

Da die Endzeit T frei ist, muss dort $H(T) = 0$ gelten:

$$H(T) = 0 = 1 + \lambda_1(T)x_2(T) + \lambda_2(T)u(T) \quad (5.3.57)$$

$$= 1 + (c_1 T + c_2)(-u_m \operatorname{sgn}(c_1 T + c_2)) \quad (5.3.58)$$

$$= 1 - c_1 \left(T + \frac{c_2}{c_1} \right) u_m \operatorname{sgn} \left(c_1 \left(T + \frac{c_2}{c_1} \right) \right) \quad (5.3.59)$$

$$= 1 - c_1 \frac{T}{2} u_m \operatorname{sgn} \left(c_1 \frac{T}{2} \right) \quad (5.3.60)$$

$$= 1 - \frac{T}{2} u_m \|c_1\| \quad (5.3.61)$$

$$\Rightarrow \|c_1\| = \frac{2}{T u_m} \quad (5.3.62)$$

$$\stackrel{(5.3.45)}{\Rightarrow} \|c_2\| = \frac{T}{2} \|c_1\| = \frac{1}{u_m} \quad (5.3.63)$$

Aus (5.3.45) folgt $\operatorname{sgn}(c_2) = -\operatorname{sgn}(c_1)$ und somit aus $\operatorname{sgn}(x_e) = -\operatorname{sgn}(c_2)$ (s.o.), $\operatorname{sgn}(c_1) = \operatorname{sgn}(x_e)$:

$$c_1 = \operatorname{sgn}(x_e) \frac{2}{T u_m} \quad (5.3.64)$$

$$\Rightarrow \lambda_2 = \operatorname{sgn}(x_e) \left(\frac{2}{T u_m} t - \frac{1}{u_m} \right) \quad (5.3.65)$$

$$= \operatorname{sgn}(x_e) \left(\sqrt{\frac{1}{\|x_e\| u_m}} t - \frac{1}{u_m} \right) \quad (5.3.66)$$

Damit folgt schließlich die Lösung für die Bahn der Punktmasse zu:

$$x_1(t) = -\operatorname{sgn}(x_e) u_m \left[\frac{1}{2} \left(t - \sqrt{\frac{\|x_e\|}{u_m}} \right)^2 \operatorname{sgn} \left(t - \sqrt{\frac{\|x_e\|}{u_m}} \right) - \left(\sqrt{\frac{\|x_e\|}{u_m}} t + \frac{\|x_e\|}{2 u_m} \right) \right] \quad (5.3.67)$$

Die Lösung ist in Bild 5.3 graphisch dargestellt. Das Ergebnis ist typisch für zeitoptimale Trajektorien: die Stellgröße schaltet zwischen maximal und minimal zulässigen Werten („bang-bang“ Steuerung).

5.4 Numerische Lösung

5.4.1 Einleitung

Wie eingangs erwähnt stehen zur Lösung der Optimierungsaufgabe zwei Klassen an Verfahren zur Verfügung: direkte und indirekte Verfahren. Im Folgenden sind einige Vor- und Nachteile beider Ansätze aufgeführt.

- Indirekte Verfahren
 - Aufwändige Herleitung

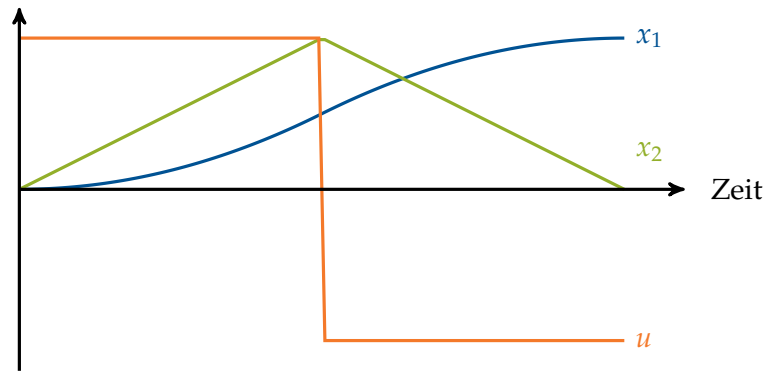


Bild 5.3: Lösung der Zeitoptimalen Bahn einer Punktmasse mit Stellgrößenbeschränkung.

- + Sehr genaue Lösung möglich
- + Erleichtert Analyse des Problems und der Lösung
- Direkte Verfahren
 - + Herleitung Randwertproblem unnötig (s.u.)
 - + Ganz allgemeine Nebenbedingungen möglich, die über das bisher betrachtete hinausgehen.
 - + „Gemischte Probleme“ möglich (z.B. minimale Masse und Energieverbrauch).
 - Es wird nur unter den Ansatzfunktionen gesucht

In der Anwendung werden aktuell oft direkte Verfahren eingesetzt. Die Folgenden Abschnitte erläutern einige Methoden zur numerischen Lösung des Optimierungsproblems.

5.4.2 Indirekte Verfahren

Möglichkeiten zur numerischen Lösung sollen hier beispielhaft nur für das PONTYAGIN-Minimum-Prinzip gezeigt werden. Zunächst wird, wie oben geschildert, die Steuergröße u als Funktion von $z^T = (\lambda^T, x^T)$ geschrieben. Dies muss u.U. stets aufs Neue numerisch durchgeführt werden. Damit folgt:

$$\dot{z} = A(z, t) \quad (5.4.1)$$

$$x(t_a) = x_a \quad (5.4.2)$$

$$x(t_e) = x_e \quad (5.4.3)$$

Dabei fasst $A(z, t)$ die HAMILTON kanonischen DGLn zusammen ($\dot{x}^T = \frac{\partial H}{\partial \lambda}$, $\dot{\lambda}^T = -\frac{\partial H}{\partial x}$). Zu lösen ist also ein ZPRWP.

Im Folgenden werden zwei Möglichkeiten zur Lösung dargestellt. Aus Gründen der Übersichtlichkeit wird von einer festen Endzeit t_e ausgegangen.

Diskretisierung

Eine einfache Möglichkeit zur Lösung ist die direkte Diskretisierung der Steuer- und Zustandsgrößen. Hierzu wird das Intervall $[t_a, t_e]$ in N Teil-Intervalle geteilt. Die Funktionswerte an den $N + 1$ Intervallgrenzen z_k stellen $(N + 1)2n$ unbekannte Variablen dar. Die Zeitableitung in \dot{z} in (5.4.1) kann durch finite Differenzen approximiert werden, etwa durch:

$$\dot{z}_k \approx \frac{z_{k+1} - z_k}{t_{k+1} - t_k} \quad (5.4.4)$$

Dann wird aus (5.4.1) ein nichtlineares *algebraisches* Gleichungssystem in den Unbekannten:

$$z_{k+1} = z_k + (t_{k+1} - t_k)A(z_k), \quad k = 0, \dots, N - 1 \quad : N(2n) \text{ Gleichungen} \quad (5.4.5)$$

Zusätzlich liefern die Randbedingungen $2n$ Gleichungen, sodass die $2n(N + 1)$ Unbekannten eindeutig bestimmt werden können.

Kollokation

Eine Alternative zur direkten Diskretisierung ist die Verwendung von Ansatzfunktionen für die Lösung. Gut geeignet sind oft kubische Splines. Es wird nun gefordert, dass die Approximation (d.h. die Spline-Funktion) die DGL (5.4.1) zumindest an bestimmten „Kollokationsstellen“ erfüllt. Als Kollokationsstellen werden üblicherweise die Stützstellen des Splines verwendet. Dieses Verfahren ist in zahlreichen Variationen weit verbreitet.

5.4.3 Direkte Verfahren

Bei den direkten Verfahren wird auf die Herleitung von Optimalitätsbedingungen verzichtet und unmittelbar von der ursprünglichen Aufgabenstellung ausgegangen, etwa:

$$\Phi = \Phi_e + \int_{t_a}^{t_e} f(\mathbf{x}, \mathbf{u}, t) dt \quad (5.4.6)$$

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}, \mathbf{u}, t) \quad (5.4.7)$$

$$\mathbf{x}(t_a) = \mathbf{x}_a, \quad \mathbf{x}(t_e) = \mathbf{x}_e \quad (5.4.8)$$

Da die Optimalitätsbedingungen in der Regel ebenfalls numerisch gelöst werden müssen, liegt der Gedanke nahe, eine direkte numerische Lösung von (5.4.6) zu versuchen.

Diskretisierung

Ähnlich der Lösung des ZPRWP kann auch das Problem (5.4.6) durch Diskretisierung gelöst werden, wobei hier zusätzlich die Steuergröße zu diskretisieren ist. Man erhält an den t_i , $i = 0 \dots N$ Stützstellen unbekannte Stützwerte \mathbf{x}_i , \mathbf{u}_i . Zeitableitungen $\dot{\mathbf{x}}_i$ werden wiederum durch finite Differenzen angenähert, das Integral über eine Quadraturformel, z.B. die Trapezregel. Man erhält in der Konsequenz ein nichtlineares, statisches Optimierungsproblem:

$$\Phi = \Phi_e + \sum_{k=0}^{N-1} \frac{t_{k+1} - t_k}{2} f(\mathbf{x}_k, \mathbf{u}_k, t_k) \quad (5.4.9)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{a}(\mathbf{x}_k, \mathbf{u}_k, t_k)(t_{k+1} - t_k) \quad \forall k = 0 \dots (N-1) \quad (5.4.10)$$

$$\mathbf{x}(t_0) = \mathbf{x}_a, \quad \mathbf{x}(t_N) = \mathbf{x}_e \quad (5.4.11)$$

Dieses ist zwar in der Regel sehr groß, kann jedoch über Standardverfahren gelöst werden (z.B. SQP-Verfahren). Es gibt für derartige Problemstellungen leicht erhältliche und effiziente Codes (etwa in Matlab, IMSL, SNOPT, etc.). Eine Liste frei verfügbarer Programmcodes mit Entscheidungshilfen ist unter <http://plato.asu.edu/sub/pns.html> zu finden.

Kollokation

Eine Alternative zur direkten Lösung ist wiederum die Kollokation, d.h. die Verwendung von Ansatzfunktionen anstatt diskreter Werte, wobei die DGL wiederum an den Kollokationsstellen erfüllt wird. Es folgt auch hier ein nichtlineares, statisches Optimierungsproblem. Dieser Ansatz ist z.B. in den Programmen DIRCOL von Prof. v. Stryk (TU Darmstadt, <http://www.sim.informatik.tu-darmstadt.de/en/res/sw/dircol.php>) implementiert.

5.5 Reduktion auf einen Bahnparameter

Die dynamische Optimierung ist im Allgemeinen sehr rechenintensiv, wobei der Aufwand stark mit der Anzahl der Freiheitsgrade steigt. Gleichzeitig ist die Bahn des Roboters oftmals bereits durch die Aufgabe zumindest näherungsweise bestimmt. Ein Ansatz zur Reduktion des numerischen Aufwands der Optimierung ist daher die Vorgabe der Bahn als Funktion eines Bahnparameters s : $\mathbf{q} = \mathbf{q}(s)$. Unbekannt ist dann nur die zeitliche Entwicklung des Bahnparameters, also die Funktion $s(t)$. Damit ist die Dimension des Problems auf eins reduziert.

Um eine Optimierungsaufgabe lösen zu können, muss die BGL des Systems in Abhängigkeit des unbekannten Bahnparameters geschrieben werden. Über die bekannte Funktion $q(s)$ folgt über die Kettenregel der Differentialrechnung:

$$\dot{q}(t) = q'(s(t)) \quad (5.5.1)$$

$$\ddot{q}(t) = q''(s) \dot{s} \quad (5.5.2)$$

$$\ddot{q}(t) = q''(s) \dot{s} + q''(s) \dot{s}^2 \quad (5.5.3)$$

Hierbei wurde die Ableitung $\frac{\partial q}{\partial s}$ als q' geschrieben. Eingesetzt in die BGL in Standardform (3.3.42) folgt:

$$M(q' \dot{s} + q'' \dot{s}^2) + h(q, q' \dot{s}) = Q_u \quad (5.5.4)$$

$$(Mq') \dot{s} + (Mq'') \dot{s}^2 + h(q, q' \dot{s}) = Q_u \quad (5.5.5)$$

$$(5.5.6)$$

Mit der plausiblen Annahme, dass s monoton steigt, kann \dot{s} nicht nur als Funktion der Zeit t , sondern auch des Bahnparameter s geschrieben werden: $\dot{s} = \dot{s}(s)$. Damit folgt für die Beschleunigung \ddot{s} :

$$\ddot{s} = \frac{d\dot{s}}{ds} \frac{ds}{dt} = \frac{1}{2} \frac{d}{ds} (\dot{s})^2 = \frac{1}{2} (\dot{s}^2)' \quad (5.5.7)$$

Damit kann \ddot{s} aus (5.5.5) eliminiert werden:

$$\frac{1}{2} (Mq') (\dot{s}^2)' + (Mq'') \dot{s}^2 + h(q, q' \dot{s}) = Q_u \quad (5.5.8)$$

$$\Leftrightarrow A(s) (\dot{s}^2)' + B(s) \dot{s}^2 + C(s) = Q_u \quad (5.5.9)$$

Die Bewegungsgleichungen reduzieren sich also auf ein System erster Ordnung in Abhängigkeit von \dot{s}^2 und s . Dies eröffnet den Weg zu einer sehr effizienten, geometrischen Lösung des Problems und ermöglicht auch den Einsatz von der hier nicht behandelten Methode der Dynamischen Programmierung nach BELLMANN, die nur für Probleme mit wenigen Freiheitsgraden geeignet ist (Pfeiffer und Johanni 1986; Pfeiffer und Reithmeier 1987).

Kapitel 6

Regelung

6.1 Einleitung

6.1.1 Grundbegriffe

In diesem Kapitel werden einige weit verbreitete Methoden zur Roboterregelung vorgestellt. Zunächst sollen in diesem Abschnitt einige Grundbegriffe geklärt werden.

Definition 6.1.1. Bei einer *Regelung* wird eine zu regelnde Größe (die Regelgröße), fortlaufend gemessen und mit einem gewünschten Verhalten (der Führungsgröße) verglichen. Abhängig vom Vergleich wird das System (die Strecke) über eine Stellgröße beeinflusst, um das gewünschte Verhalten zu erzielen.

Definition 6.1.2. Bei einer *Steuerung* wird Stellgröße z.B. auf Basis eines Systemmodells und ohne Rückführung gemessener Größen berechnet und/oder vorgegeben.

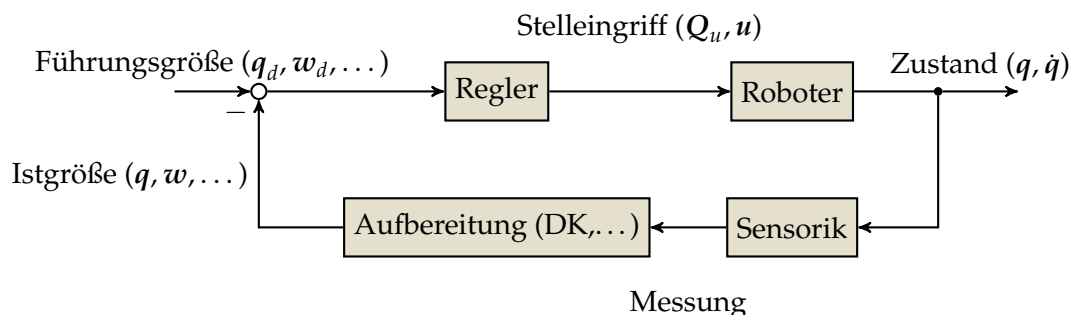


Bild 6.1: Typische Struktur des Regelkreises eines Roboters.

6.1.2 Aufgaben und Ziele

Die typische Aufgabe einer Regelung in der Robotik sind das Nachfahren einer gewünschten Trajektorie (Trajektorienfolgeregelung). Bei Interaktionsaufgaben mit der Umwelt kann auch das Einstellen einer gewünschten Kontaktkraft (Kraftregelung), oder eines bestimmten dynamischen Verhaltens in der Interaktion mit der Umwelt gefordert sein (Impedanzregelung).

Ziele des Reglerentwurfs sind ein möglichst gutes Führungsverhalten, d.h. geringe Abweichungen zwischen Führungsgröße und gemessenem Verhalten. Dabei muss stets die Stabilität des Regelkreises gewährleistet sein. Für die Leistungsfähigkeit in der Anwendung ist eine möglichst wirkungsvolle Unterdrückung von Störgrößen wichtig.

6.1.3 Besonderheiten und Vorgehen

Die Dynamik von (Industrie-) Robotern wird durch die in Kapitel 3 behandelten Bewegungsgleichungen (3.3.42) beschrieben. Das zu regelnde System ist also stark nichtlinear. Neben den bisher

betrachteten Effekten aus der Mehrkörperdynamik kommen in der Anwendung zusätzlich oft schwer zu modellierende und stark veränderliche Reibkräfte in den Getrieben hinzu.

Günstig für die Regelung ist hingegen, dass (Industrie-) Roboter meist vollständig beobachtbar und vollständig steuerbar sind, d.h. der Zustand aus den Messungen rekonstruiert werden kann und der gewünschte Zustand durch die vorhandenen Stellkräfte prinzipiell erreichbar ist. Für eine genaue Erläuterung dieser Begriffe sei auf die einschlägige regelungstechnische Literatur verwiesen (etwa Ludyk 1995a,b).

Bei Entwurf der Regelung wird typischerweise in zwei Schritten vorgegangen. Im ersten Schritt werden ideale Trajektorien geplant (Bahnplanung), d.h. der gewünschte Verlauf der Minimalkoordinaten q_d oder Arbeitsraumkoordinaten w_d . Damit ist stets auch ein idealer Verlauf der Stellkräfte $Q_u(t)$ gegeben. Im zweiten Schritt wird dann eine Regelung entworfen, die es dem System erlaubt, der Solltrajektorie zu folgen. Dabei müssen die immer vorhandenen Modellfehler (z.B. vernachlässigte Nachgiebigkeit der Struktur, etc.), externe Störungen und Parameterunsicherheiten (z.B. nicht exakt bekannte Massen und Trägheitsmomente), kompensiert werden.

6.1.4 Überblick Regelungsstrukturen

Die Regelung eines Roboters kann je nach Aufgabenstellung, Anforderungen und Randbedingungen sehr unterschiedlich gelöst werden. Es gibt bereits eine Fülle unterschiedlicher Regelungsarchitekturen, die jeweils spezifische Vor- und Nachteile bieten. Bild 6.2 zeigt einen Überblick weit verbreiteter Ansätze.

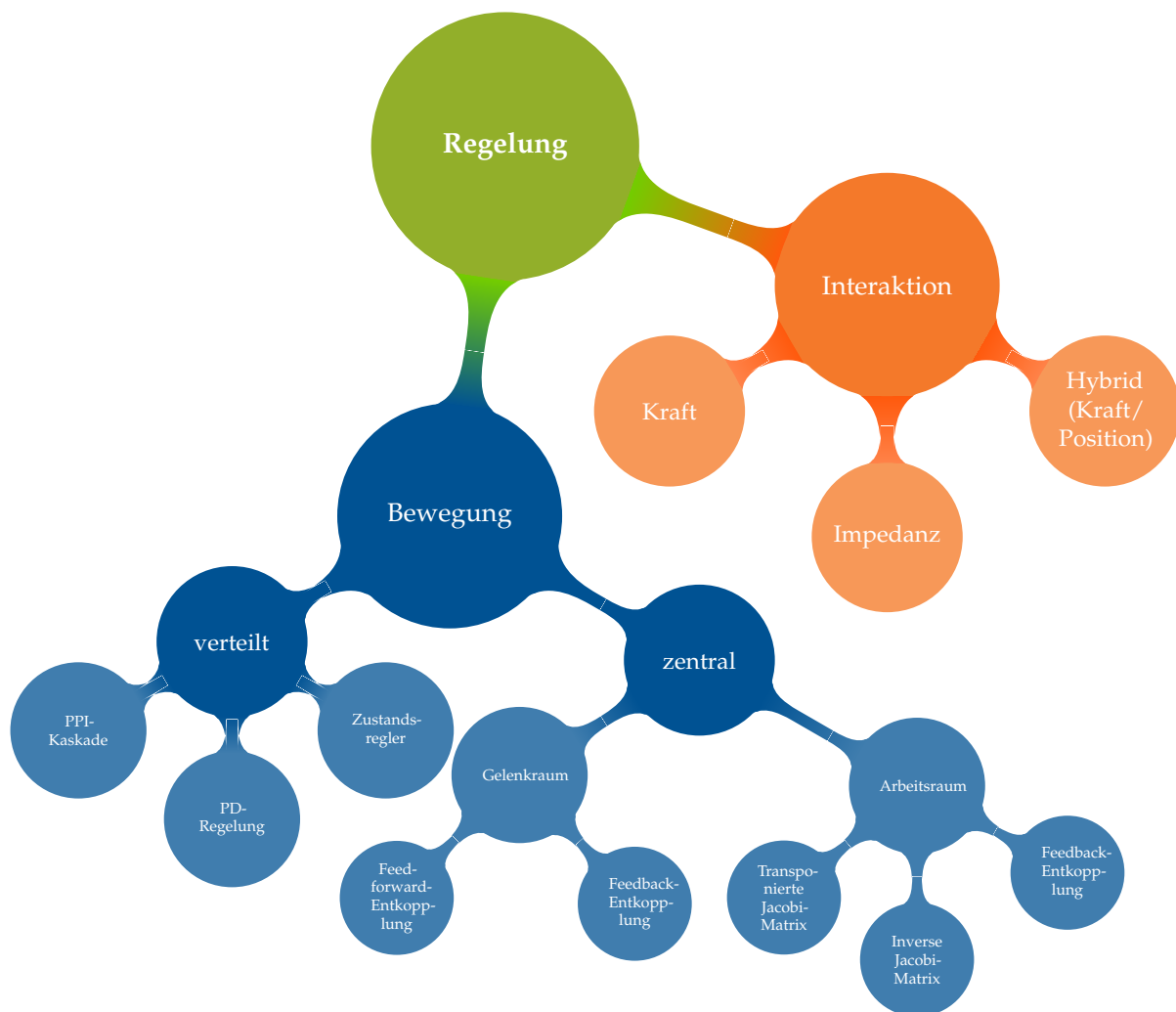


Bild 6.2: Überblick weit verbreiteter Regelungsarchitekturen in der Robotik.

6.2 Regelung im Gelenkraum

6.2.1 Dezentrale, entkoppelte Einzelgelenkregelung

Die einfachste Möglichkeit ist eine entkoppelte, dezentrale Einzelgelenkregelung. Bei dieser oft eingesetzten Methode wird jedes einzelne Gelenk unabhängig voneinander geregelt. Die Mehrkörperdynamik bleibt unberücksichtigt und wird in der Regelung als Störung betrachtet. Da bei der Verwendung hochübersetzender Getriebe die Kopplungen der einzelnen Gelenke reduziert werden, begünstigen hohe Übersetzungen die Verwendung dieses Regelungskonzepts. Die Regelung der einzelnen Antriebe kann in einfachen Fällen bereits über gängige, lineare Antriebsregler ausreichend genau durchgeführt werden. Das Konzept ist in Bild 6.3 veranschaulicht. Ein wesentlicher Vorteil der entkoppelten Regelung ist die Möglichkeit, die Antriebsregelung dezentral mit der Leistungselektronik in der Nähe der Antriebe zu integrieren, da die Regelung lediglich Messwerte eines Gelenks benötigt.

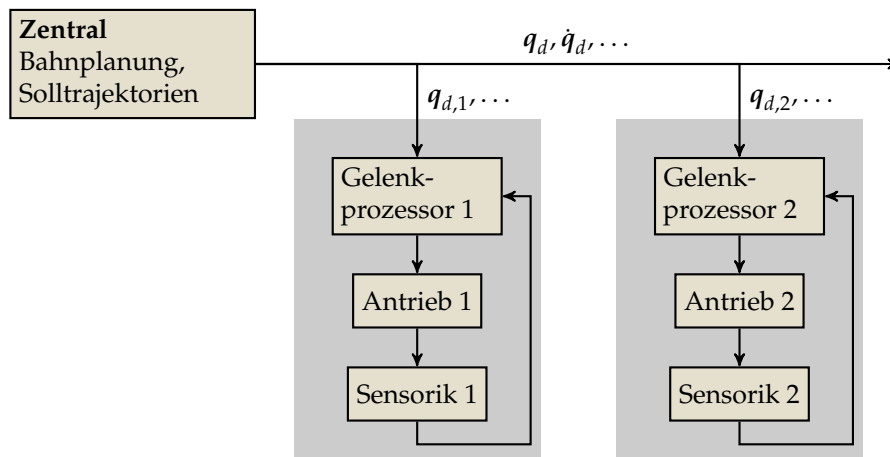


Bild 6.3: Struktur einer entkoppelten, dezentralen Einzelgelenksregelung.

Modellierung der Strecke

Im folgenden werden beispielhaft zwei Varianten einer Einzelgelenkregelung vorgestellt. Als Antrieb wird ein bürstenbehafteter DC-Motor angenommen. Zusätzlich zur Trägheit des Rotors herrscht viskose Reibung. Ohne Einschränkungen wird der Einfachheit halber eine Übersetzung des Getriebes von $N = 1$ angenommen. Das geregelte System ist schematisch in Bild 6.4 dargestellt.

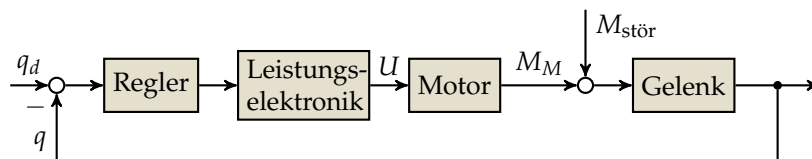


Bild 6.4: Schaltbild zur Einzelgelenkregelung.

Für das elektrische Verhalten gilt die Bewegungsgleichung des DC-Motors (vgl. Abschnitt 3.6):

$$U = RI + L\dot{I} + k_M\dot{q} \quad (6.2.1)$$

Das mechanische Verhalten ist durch

$$M_M = \Theta\ddot{q} + b\dot{q} - M_{\text{stör}} = k_M I \quad (6.2.2)$$

gegeben. Dabei ist M_M das Antriebsmoment des Motors, Θ die Trägheit und b eine Reibkonstante. Meist ist die Dynamik der Leistungselektronik im Vergleich zum restlichen System sehr schnell und kann daher vernachlässigt werden. Andernfalls ist eine Approximation etwa als PT1-Glied möglich.

Kaskadenregelung

Eine weit verbreitete Variante der Antriebsregelung ist die Kaskadenregelung, in der eine Strom-, Geschwindigkeits- und Positionsschleife geschachtelt werden, wobei die weiter innen liegenden Schleifen eine wesentlich höhere Bandbreite aufweisen als die weiter außen liegenden.

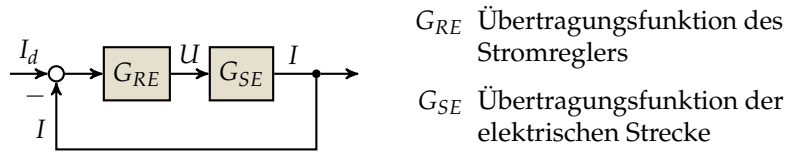


Bild 6.5: Stromschleife der Einzelgelenksregelung.

Stromregelung Im LAPLACE-Bereich berechnet sich aus (6.2.1) die Übertragungsfunktion der elektrischen Strecke G_{SE} zu:

$$G_{SE} = \frac{I}{U} = \frac{1}{R + sL} \quad (6.2.3)$$

Diese Strecke kann gut durch einen Proportional-Integralen (PI) Regler mit der Übertragungsfunktion

$$G_{RE} = K_p + \frac{1}{s} K_I \quad (6.2.4)$$

geregelt werden (K_p, K_I sind die Konstanten des Reglers).

Die Gesamt-Übertragungsfunktion des elektrischen Teils G_E berechnet sich damit zu:

$$G_E = \frac{G_{RE} G_{SE}}{1 + G_{RE} G_{SE}} = \frac{(\frac{1}{s} K_I + K_p)(\frac{1}{sL + R})}{1 + (\frac{1}{s} K_I + K_p)(\frac{1}{sL + R})} = \frac{K_p s + K_I}{s^2 L + (K_p + R)s + K_I} \quad (6.2.5)$$

Zur Analyse des Verhaltens wird die Sprungantwort betrachtet. Mit der Sprungfunktion $\sigma(t)$ und der LAPLACE-Transformierten $\mathcal{L}\{\sigma(t)\} = \frac{1}{s}$ folgt für den Strom durch eine sprungförmige Änderung des Sollstroms I_d :

$$I = \frac{1}{s} G_E \quad (6.2.6)$$

Mit dem Grenzwertsatz

$$\lim_{t \rightarrow \infty} I(t) = \lim_{s \rightarrow 0} s I(s) \quad (6.2.7)$$

folgt:

$$\lim_{t \rightarrow \infty} I(t) = \lim_{s \rightarrow 0} s \left(\frac{1}{s} G_E \right) = \frac{K_I}{K_I} = 1 \quad (6.2.8)$$

Der gewünschte Strom wird ohne bleibende Reglerabweichung erreicht.

Im Sinne der Kaskadenregelung wird der Strom I für die weiter außen liegenden Schleifen als Stellgröße betrachtet.

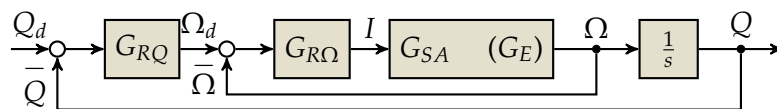


Bild 6.6: PPI-Kaskade zur entkoppelten Einzelgelenkregelung. G_{RQ} ist die Übertragungsfunktion der Positiosregelung, $G_{RO\Omega}$ die der Geschwindigkeitsregelung, G_{SA} die des Antriebs mit Stromregelung (G_E).

Kaskade für Position und Geschwindigkeit Die Struktur der kaskadierten Positions- und Geschwindigkeitsregelung ist in Bild 6.6 dargestellt. Mit der BGL des Antriebs (6.2.2) und $M_M = k_M I$ ergibt sich die Übertragungsfunktion des Antriebs zu:

$$G_{SA} = \frac{\Omega}{I} = \frac{k_M}{(s\Theta + b)} \quad (6.2.9)$$

G_{SA} gleicht also der elektrischen Übertragungsfunktion des Motors G_{SE} , sodass auf Geschwindigkeitsebene ebenfalls eine PI-Regelung eingesetzt wird:

$$G_{R\Omega} = \frac{1}{s}K_I + K_p\Omega \quad (6.2.10)$$

Die Geschwindigkeitsregelung weist ebenfalls keine bleibende Regelabweichung auf.

Zur Regelung der Positionen wird meist eine einfache proportionale (P) Regelung eingesetzt, d.h.

$$G_{RQ} = K_pQ \quad (6.2.11)$$

Grund hierfür ist die Tatsache, dass eine Regelung mit Integralanteil bei ungünstiger Einstellung zum Überschwingen führen kann: der Antrieb fährt kurzfristig über das Ziel hinaus. Dieses Verhalten ist in der Antriebsregelung unerwünscht, da es etwa zu Kollisionen führen kann. Trotz fehlendem Integralanteil weist die Positionsregelung aufgrund des I-Anteils keinen bleibenden Regelfehler auf.

Die Herleitung der Gesamtübertragungsfunktion und der Sprungantwort des in Bild 6.6 dargestellten Systems wird dem Leser als Übungsaufgabe überlassen.

PD-Positionsregler

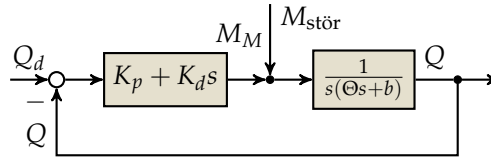


Bild 6.7: PD-Regelung eines einzelnen Gelenks.

Alternativ zur PPI-Kaskade ist auch die direkte Regelung der Position über einen Proportional-Differenzierenden (PD) Regler möglich (siehe Bild 6.7). Das Motormoment berechnet sich dann zu

$$M_M = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}), \quad (6.2.12)$$

woraus die Übertragungsfunktion des Reglers folgt:

$$G_{RA} = K_p + sK_d \quad (6.2.13)$$

Es wird hier angenommen, dass durch die PD-Regelung direkte das Antriebsmoment vorgegeben werden kann. Alternativ kann der hierzu proportionale Motorstrom als Stellgröße gewählt werden, wozu Bild 6.7 um die Motorkonstant k_M ergänzt werden müsste.

Aus (6.2.2) folgt die Übertragungsfunktion der Strecke zu

$$G_{SA} = \frac{1}{s(s\Theta + b)}. \quad (6.2.14)$$

Die Gesamtübertragungsfunktion folgt damit zu:

$$G_{\text{ges}} = \frac{G_{RA}G_{SA}}{1 + G_{RA}G_{SA}} = \frac{K_p + sK_d}{\Theta s^2 + (K_d + b)s + K_p} \quad (6.2.15)$$

Der Grenzwertsatz erlaubt wiederum die Berechnung der bleibenden Regelabweichung bei einem Sprung der Sollposition:

$$\lim_{t \rightarrow \infty} q(t) = \lim_{s \rightarrow 0} s \frac{1}{s} G_{\text{ges}} = 1 \quad (6.2.16)$$

Das Verhalten bezüglich bleibender Regelabweichungen erscheint also zunächst ebenso günstig wie das der PPI-Kaskade. Die Betrachtung der Störungsübertragungsfunktion

$$G_{ST} = \frac{Q}{M_{\text{stör}}} = \frac{1}{\Theta s^2 + (K_d + b)s + K_p} \quad (6.2.17)$$

zeigt jedoch, dass bei einer sprungförmigen Änderung des Störmoments $M_{\text{stör}}$ eine Abweichung bleibt:

$$\lim_{t \rightarrow \infty} q(t) = \lim_{s \rightarrow 0} s \frac{1}{s} G_{\text{stör}} = \frac{1}{K_p} \quad (6.2.18)$$

Da alle Kopplungen in der Dynamik des Roboters bei der entkoppelten Einzelgelenkregelung Regelung als Störung betrachtet werden, sind im realen Betrieb durch eine PD-Regelung stets bleibende Fehler zu erwarten. Allerdings können diese Fehler durch die Wahl sehr hoher Verstärkungen $\|K_p\| \gg 1$ klein gehalten werden. Die PD-Regelung kann bei dynamischen Bewegungen gegenüber der PPI-Kaskade Vorteile haben, da der Integralanteil zwar bleibende Regelabweichungen eliminiert, das Systemverhalten dadurch jedoch etwas träger wird.

Für eine vertiefte Darstellung der Antriebsregelung sei auf (Schröder 2001) verwiesen. Eine kompakte Darstellung der linearen Regelungstechnik und typischer Antriebsregelungen ist in (Lutz und Wendt 2005) zu finden.

6.2.2 Feedback-Entkopplung

In diesem Abschnitt wird ein Regelungskonzept vorgestellt, das die Berücksichtigung des Modellwissens in Form der Bewegungsgleichungen ermöglicht. Dies erfordert jedoch eine zentrale Implementierung der Regelung, da bei einer Berücksichtigung aller Kopplungen auch alle Zustandsgrößen zur Verfügung stehen müssen.

Der Aufwand kann notwendig und lohnend sein, wenn erhöhte Anforderungen an die Bahn-treue gestellt werden und hohe Geschwindigkeiten und Beschleunigungen auftreten. In diesem Fall kann der Einfluss der Kopplungen und Nichtlinearitäten nicht vernachlässigt werden. Ob eine einfache entkoppelte Regelung möglich ist, hängt von den Eigenschaften des jeweiligen Roboters ab, etwa der Massenverteilung und den Getriebeübersetzungen.

Die Idee der Feedback-Entkopplung ist es, durch eine geeignete Transformation der Stellgröße für ein transformiertes System eine lineare Reglerauslegung und lineares Verhalten zu ermöglichen. Kann eine solche Transformation gefunden werden, ist die Reglerauslegung mit den wohlbekannten Werkzeugen der linearen Regelungstechnik leicht möglich. Im Spezialfall vollaktuierter Roboter, der hier behandelt wird, ist die Feedback-Entkopplung besonders einfach. Für eine allgemeinere und leicht zugänglich Behandlung sei auf (Slotine und Li 1991) verwiesen.

Wir gehen von der BGL in Standardform aus:

$$M(q)\ddot{q} + h(q, \dot{q}) = Q_u \quad (6.2.19)$$

Offenbar ist (im vollaktuierten Fall) die Stellkraft Q_u direkt vom momentanen Zustand (q, \dot{q}) und der Minimalbeschleunigung \ddot{q} abhängig. Wählen wir $v = \ddot{q}$ als neue Stellgröße, erhalten wir als transformierte BGL das triviale System entkoppelter, zweifacher Integratoren:

$$\ddot{q} = v \quad (6.2.20)$$

Haben wir für dieses System eine Regelung ausgelegt, erhalten wir über (6.2.19) unmittelbar die tatsächlich aufzubringenden Kräfte zu

$$Q_u = Mv + h \quad (6.2.21)$$

Für die lineare Regelung bietet sich eine PD-Regelung mit Vorsteuerung der Beschleunigungen an:

$$\mathbf{v} = \ddot{\mathbf{q}}_d + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \quad (6.2.22)$$

Die Verstärkungsmatrizen $\mathbf{K}_d, \mathbf{K}_p$ werden in der Regel diagonal gewählt, da sonst über die Regelung Kopplungen zwischen den Freiheitsgraden eingeführt würden. Die Parameter $\mathbf{K}_d, \mathbf{K}_p$ können z.B. so gewählt werden, dass für alle Freiheitsgrade aperiodisches Einschwingen mit der Zeitkonstante $1/\lambda_i$ vorliegt:

$$K_{p,i} = \lambda_i^2 \quad (6.2.23)$$

$$K_{d,i} = 2\lambda_i \quad (6.2.24)$$

Die Funktionsfähigkeit der Entkopplung kann durch Einsetzen des Regelgesetzes (6.2.22) und der Transformation (6.2.21) in die BGL (6.2.19) überprüft werden:

$$\begin{aligned} Q_u &= \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} \\ \Leftrightarrow \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})) + \mathbf{h} &= \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} \\ \Leftrightarrow (\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) &= \mathbf{0} \end{aligned}$$

Die Regelung durch Feedback-Entkopplung ist in Bild 6.8 graphisch dargestellt.

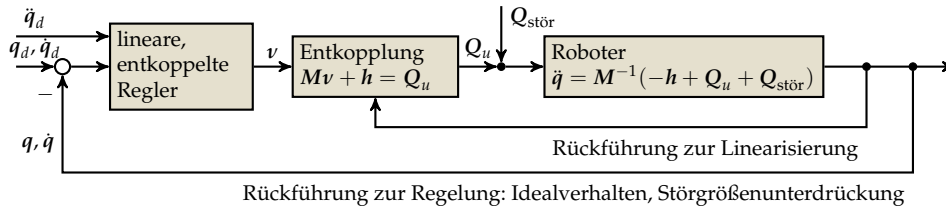


Bild 6.8: Schaltbild für die Regelung durch Feedback-Entkopplung.

Vorteile des Verfahrens sind der einfache Reglerentwurf und die vollständige Berücksichtigung des Modellwissens. Allerdings ist der Rechenbedarf relativ hoch, da zu jedem Zeitschritt die BGL ausgewertet werden muss. Auch ist eine zentrale Implementierung erforderlich. Im Idealfall perfekter Messungen und fehlerfreier Modelle erlaubt das Verfahren ein ideales Systemverhalten. Die Leistungsfähigkeit ist bei Meß- und Modellfehlern jedoch nicht garantiert.

Code- Beispiel

Ein Beispiel zur Programmierung des Verfahrens der Feedback-Entkopplung zur Gelenkregelung finden Sie im begleitenden Matlab-Skript zur Regelung (beispiel_reg.m).

6.2.3 Feedforward-Entkopplung

Eine Alternative zur Berücksichtigung des Modellwissens ist die Feedforward-Entkopplung. Hierbei werden die Nichtlinearitäten nicht in der Regelung kompensiert, sondern in der Vorsteuerung berücksichtigt. Die BGL wird also für die *Sollgrößen* ausgewertet. Störungen, Kopplungen und Modellfehlern werden dann durch einen linearen Mehrgrößenregler kompensiert. Die Stellgröße Q_u wird in Vorsteuerung $Q_{u,S}$ und Regelung $Q_{u,R}$ aufgeteilt:

$$Q_u = \underbrace{\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{h}(\mathbf{q}_d, \dot{\mathbf{q}}_d)}_{Q_{u,S} \text{ Vorsteuerung}} + \underbrace{Q_{u,R}}_{\text{Regelung}} \quad (6.2.25)$$

Da über die Regelung nur kleinere Störungen ausgeglichen werden sollen, kann der Regler für kleine Abweichungen von der Referenztrajektorie $\mathbf{q}_d(t)$ ausgelegt werden. Hierzu wird die BGL um $\mathbf{q}_d(t)$ linearisiert. Um eine kompakte Schreibweise zu ermöglichen, wird die linke Seite der BGL mit $f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ bezeichnet:

$$f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \quad (6.2.26)$$

Mit der Bezeichnung des Regelfehlers als q_e gilt für q

$$q = q_d + q_e \quad (6.2.27)$$

liefert die Taylorentwicklung der BGL um q_d bis zu linearen Termen:

$$f(q, \dot{q}, \ddot{q}) = f|_{q_d} + \frac{\partial f}{\partial \ddot{q}} \Big|_{q_d} \ddot{q}_e + \frac{\partial f}{\partial \dot{q}} \Big|_{q_d} \dot{q}_e + \frac{\partial f}{\partial q} \Big|_{q_d} q_e + \text{HOT} \quad (6.2.28)$$

$$= \underbrace{M(q_d)\ddot{q}_d + h(q_d, \dot{q}_d)}_{\stackrel{!}{=} Q_{u,S}} \quad (6.2.29)$$

$$+ \underbrace{M(q_d)}_{M_e} \ddot{q}_e + \underbrace{\frac{\partial h}{\partial \dot{q}} \Big|_{q_d}}_{D_e} \dot{q}_e + \underbrace{\left[\frac{\partial h}{\partial q} \Big|_{q_d} + \frac{\partial(M\ddot{q})}{\partial q} \Big|_{q_d} \right]}_{K_e} q_e + \text{HOT}$$

Hierbei wurde die Auswertung einer Größe X an der Solltrajektorie ($q = q_d, \dot{q} = \dot{q}_d, \ddot{q} = \ddot{q}_d$) abkürzend durch $X|_{q_d}$ bezeichnet und Terme höherer Ordnung mit „HOT“.

Mit dem Zustandsvektor $x^T = (q_e^T, \dot{q}_e^T)$ und der Umbenennung $u = Q_{u,R}$ folgt die linearisierte Systemdynamik zu:

$$\dot{x} = Ax + Bu \quad (6.2.30)$$

$$\text{mit } A = \begin{pmatrix} 0 & E \\ -M_e^{-1}K_e & -M_e^{-1}D_e \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} 0 \\ M_e^{-1} \end{pmatrix} \quad (6.2.31)$$

Dabei sind K_e, M_e, D_e entsprechend obiger Linearisierung definiert.

Ein Vorteil der Feedforward-Entkopplung ist, dass Vorsteuerung und Systemmatrizen offline berechnet werden können. Andererseits ist die Berechnung der Systemmatrizen aufwändig. Auch muss ein Mehrgrößenregler ausgelegt werden, da in (6.2.30) die (linearisierten) Kopplungen der einzelnen Freiheitsgrade vorhanden sind. Die Auslegung des Mehrgrößenreglers (z.B. LQR) ist ungleich aufwändiger, als die im Falle der Feedback-Entkopplung notwendigen. Im Gegensatz zur dezentralen Regelung muss auch für die Feedforward-Entkopplung eine zentrale Instanz vorhanden sein, in der der Mehrgrößenregler implementiert ist.

Falls im Sinne einer Entlastung des Echtzeitbetriebs Matrizen und Vorsteuerung vorab offline berechnet werden, bedeutet dies auch einen Verlust an Flexibilität und der Möglichkeit, online – etwa aufgrund eines Abstandssensors oder eines Benutzereingriffs – die Bahn zu verändern.

Die Vorsteuerung ist jedoch prinzipiell auch mit einer dezentralen Regelung kombinierbar. Im einfachsten Fall wird die Vorsteuerung nur der Lageabhängigen Terme (z.B. durch die Gravitation) mit entkoppelten Einzelgelenkreglern kombiniert. Dadurch kann bei einem korrekt parametrisierten Modell zumindest die statische Regelabweichung durch einen PD-Regler eliminiert werden, die durch konstante Terme der BGL verursacht werden und in der entkoppelten Regelung nicht berücksichtigt werden können.

Code-Beispiel Ein Beispiel zur Programmierung des Verfahrens der Feedforward-Entkopplung zur Gelenkregelung finden Sie im begleitenden Matlab-Skript zur Regelung (`beispiel_reg.m`). Dort wird eine vereinfachte Variante mit entkoppelten Einzelgelenksreglern umgesetzt.

6.3 Regelung im Arbeitsraum

6.3.1 Einleitung

Die bisher erläuterten Regelungskonzepte finden im Gelenk- bzw. Konfigurationsraum statt: die Solltrajektorie wird in Gelenkkoordinaten q_d vorgegeben und mit den gemessenen Gelenkkoordinaten q verglichen. Gleichzeitig ist das gewünschte Verhalten oft im Arbeitsraum gegeben, da z.B.

die Lage und Orientierung des Endeffektors interessiert, nicht der Gelenkwinkel der einzelnen Achsen. Oft wird daher eine Bahn zwar im Arbeitsraum geplant, dann aber zum Zweck der Regelung über eine inverse Kinematik in den Gelenkraum transformiert. Es liegt daher nahe, eine direkte Regelung im Arbeitsraum zu versuchen, d.h. eine Vorgabe des Sollverhaltens direkt durch eine Arbeitsraum-Trajektorie w_d und eine Bildung des Regelfehlers im Arbeitsraum. Im Folgenden wird bei der Regelung im Arbeitsraum davon ausgegangen, dass der Roboter nicht redundant ist ($\dim(q) = \dim(w)$).¹

6.3.2 Methoden der inversen und transponierten Jacobi-Matrix

Zwei einfache Möglichkeiten ergeben sich direkt aus den Algorithmen zur Lösung der inversen Kinematik auf Geschwindigkeitsebene (siehe Abschnitt 2.4.2, Seite 24 ff.).

Methode der inversen Jacobi-Matrix Die Methode der inversen Jacobi-Matrix ist in Bild 6.9 dargestellt. Der Regelfehler $\Delta w = w_d - w$ wird direkt im Arbeitsraum gebildet. Dabei muss in aller Regel der Istwert w über die direkte Kinematik aus den Minimalkoordinaten q berechnet werden, da selten Sensorik zur direkten Messung der Arbeitsraumkoordinaten vorhanden ist.

Für kleine Regelfehler gilt über die Taylorentwicklung von $w(q)$ die Näherung

$$\Delta w \approx J_w \Delta q \quad (6.3.1)$$

$$(6.3.2)$$

und auch

$$\Delta q = J_w^{-1} \Delta w. \quad (6.3.3)$$

Das Regelgesetz zur Berechnung der Stellgröße Q_u wird dann in Abhängigkeit von Δq gebildet. Oft wird eine PD-Regelung verwendet.

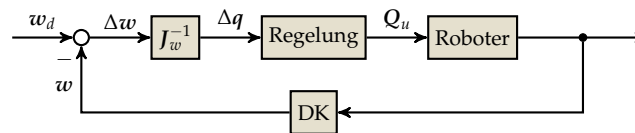


Bild 6.9: Methode der inversen Jacobi-Matrix zur Regelung im Arbeitsraum.

Methode der transponierten Jacobi-Matrix Ähnlich dem Vorgehen im vorigen Absatz kann auch aus dem IK-Algorithmus der transponierten Jacobi-Matrix eine entsprechende Arbeitsraum-Regelung entwickelt werden. Im Unterschied zur Methode der inversen Jacobi-Matrix wird das Regelgesetz jedoch ebenfalls im Arbeitsraum formuliert. Es wird also eine virtuelle Stellgröße

$$F_u(\Delta w) \quad (6.3.4)$$

berechnet, die den Fehler Δw reduzieren soll. Physikalisch können jedoch nur die Gelenkkräfte Q_u aufgebracht werden. Daher werden die zu F_u äquivalenten Kräfte Q_u über die transponierte Jacobi-Matrix (Prinzip der virtuellen Arbeit) bestimmt:

$$Q_u = J_w^T F_u \quad (6.3.5)$$

Es kann wiederum im einfachsten Fall eine PD-Regelung zum Einsatz kommen. Das Konzept ist in Bild 6.10 als Schaltbild dargestellt.

¹Eine Arbeitsraumregelung ist im redundanten Fall ebenfalls möglich, hier können jedoch Komplikationen aufgrund der sogenannten Nulldynamik auftreten. Siehe (Khalil 2000) zu den theoretischen Hintergründen.

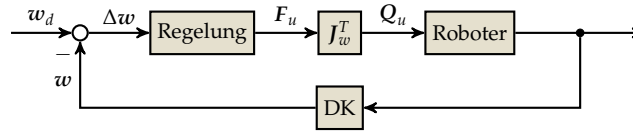


Bild 6.10: Methode der transponierten Jacobi-Matrix zur Regelung im Arbeitsraum.

6.3.3 Feedback-Entkopplung

Eine Arbeitsraum-Regelung unter exakter Berücksichtigung der bekannten Systemdynamik ist durch die Methode der Feedback-Entkopplung möglich. Im Gegensatz zur Feedback-Entkopplung im Gelenkraum wird jedoch eine entkoppelte Dynamik im Arbeitsraum angestrebt.

Um die erforderlichen Umrechnungen durchführen zu können, ist eine Abbildung zwischen Arbeits- und Gelenkraum auf Beschleunigungsebene erforderlich, da auch die BGL auf dieser Ebene formuliert ist. Durch zweimaliges Differenzieren der kinematischen Gleichung $w(q)$ erhält man:

$$\ddot{w} = J_w \ddot{q} + \dot{J}_w \dot{q} \quad (6.3.6)$$

Im Ergebnis wird wiederum ein entkoppeltes, lineares Regelungsverhalten angestrebt, das über positiv definite, diagonale Matrize K_p, K_d einstellbar ist:

$$\ddot{w}_d - \ddot{w} + K_d(\dot{w}_d - \dot{w}) + K_p(w_d - w) = 0 \quad (6.3.7)$$

Für nicht-redundante Roboter kann (6.3.6) nach \ddot{q} aufgelöst werden:

$$\ddot{q} = J_w^{-1} (\ddot{w} - \dot{J}_w \dot{q}) \quad (6.3.8)$$

Mit (6.3.7) folgt dann:

$$\ddot{q} = J_w^{-1} (\ddot{w}_d + K_d(\dot{w}_d - \dot{w}) + K_p(w_d - w) - \dot{J}_w \dot{q}) \quad (6.3.9)$$

Daraus folgt wiederum mit der BGL die gesuchte Stellgröße:

$$Q_u = M \ddot{q} + h \quad (6.3.10)$$

$$= M J_w^{-1} [\ddot{w}_d + K_d(\dot{w}_d - \dot{w}) + K_p(w_d - w) - \dot{J}_w \dot{q}] + h \quad (6.3.11)$$

Bild 6.11 veranschaulicht dieses Regelungsschema.

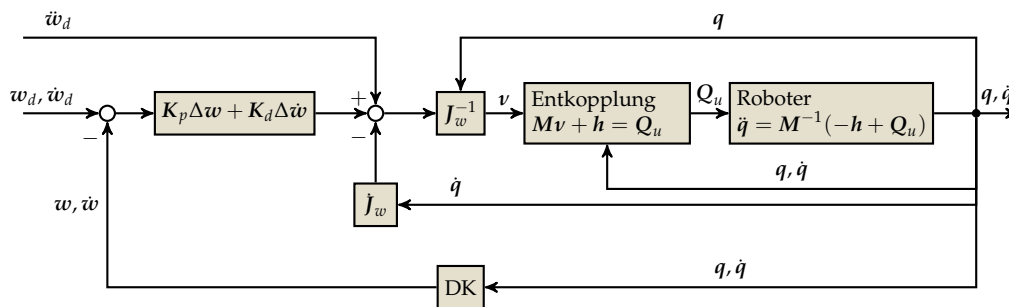


Bild 6.11: Regelung im Arbeitsraum durch Feedback-Entkopplung (die Abhängigkeit von \dot{J}_w von \dot{q} und q ist der Übersichtlichkeit halber nicht eingezeichnet.)

6.3.4 Anmerkungen

Die Verfahren der inversen und transponierten Jacobi-Matrix sind anschaulich motiviert. Die Systemdynamik in Form der BGL bleibt unberücksichtigt. Stabilität und Leistungsfähigkeit beider Verfahren sind daher im allgemeinen nicht garantiert und sowohl von der Sollvorgabe w_d , als

auch vom momentanen Zustand (q, \dot{q}) abhängig. Dennoch ist in vielen Fällen eine ausreichende Leistungsfähigkeit erreichbar.

Speziell die Methode der Transponierten ist anschaulich leicht verständlich: man bestimmt eine Kraft, die im Arbeitsraum in Gegenrichtung des Regelfehlers wirkt. Auf diese Weise können anschaulich (scheinbar) unmittelbar verständlich verschiedene virtuelle Kräfte auf das System aufgebracht werden – es muss nur der Ort der virtuellen Kraft und die entsprechende Jacobi-Matrix vorgegeben werden. Die einzelnen Teilkräfte können dann zu Q_u aufsummiert werden. Zu bedenken bleibt jedoch, dass das tatsächliche Verhalten für dynamische Bewegungen nicht immer unmittelbar einsichtig ist, da die Systemdynamik bei dieser Überlegung nicht berücksichtigt wurde.

Die Methode der Feedback-Entkopplung erlaubt jedoch, im Rahmen der Modell- und Messgenauigkeit, eine exakte Berücksichtigung der Systemdynamik. Der numerische Aufwand ist dabei gegenüber der Feedback-Entkopplung im Gelenkraum nochmals deutlich erhöht, da die Kinematik auf Beschleunigungsebene zu berechnen ist, sowie zahlreiche Vektor-Matrix-Operationen zur Auswertung des Regelgesetzes (6.3.11) erforderlich sind. Zu beachten ist, dass bei der Feedback-Entkopplung und der Methode der inversen Jacobi-Matrix Instabilitäten in der Nähe singulärer Punkte auftreten können.

6.4 Interaktionsregelung

6.4.1 Einleitung

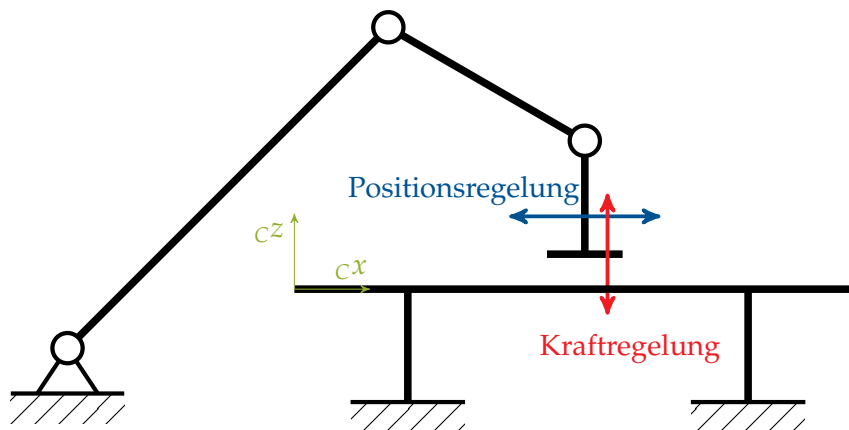


Bild 6.12: Beispiel „Tischabwischen“ zum Einsatz hybrider Kraft-/Positionsregelung bei Interaktionsaufgaben.

Bei der Interaktion mit der Umgebung kann die Erfüllung der Aufgabe durch eine Regelung der Bewegung, ob im Arbeits- oder Konfigurationsraum, oft nur schwer gelöst werden. Bild 6.12 zeigt beispielhaft den Vorgang des Tischabwischens als Aufgabe, die eine Interaktion mit der Umwelt erfordert. Bei einer reinen Positionsregelung sind sehr hohe Anforderungen an die Regelgüte zu stellen. Da der Endeffektor den Kontakt zum Tisch nicht verlieren darf und andererseits eine Beschädigung durch übermäßige Krafteinwirkung zu vermeiden ist, muss der Abstand zwischen Endeffektor und Tischfläche sehr genau geregelt werden. Im Falle einer geometrisch unregelmäßigen Oberfläche etwa mit leichten Vertiefungen, stellen sich hier prinzipielle Fragen hinsichtlich der Durchführbarkeit der Aufgabe.

Auf die extremen Anforderungen an die Genauigkeit der Positionsregelung kann jedoch verzichtet werden, wenn für einige Richtungen eine *Kraftregelung* eingesetzt wird. Im Beispiel des Tischabwischens ist für die Reinigung eine *Bewegung tangential zur Tischfläche* notwendig, sowie eine ausreichende *Kraft normal zur Tischfläche*. Die *natürliche Beschreibung* der Aufgabe besteht daher aus der Vorgabe der:

1. Der Bewegung tangential zum Tisch und

2. Der Kraft normal zum Tisch

Eine Regelung, in der einige Dimensionen kraftgeregelt sind und andere positionsgeregelt, wird als *Hybride Kraft-/Positionsregelung* bezeichnet. Auf die zahlreichen möglichen Variationen dieses Ansatzes, sowie die mathematischen Hintergründe zur Stabilität der Verfahren soll hier nicht eingegangen werden. Als Einstieg können die Literaturstellen (Siciliano und Khatib 2008; Siciliano, Sciavicco u. a. 2008) dienen.

6.4.2 Hybride Kraft-/Positionsregelung

Im Folgenden wird davon ausgegangen, dass die Umwelt als linear-elastisch modelliert werden kann. Durch eine Abweichung des Endeffektors von der Ruhelage (w_u) folgt eine Kraft²:

$$F = K_u(w - w_u) \quad (6.4.1)$$

Wie im einleitenden Beispiel dargelegt, bieten sich bei Interaktionsaufgaben oft Kraft- oder Positionsregelung für bestimmte Dimensionen des Arbeitsraums an. Zu beobachten ist ferner, dass für starre Kontakte nur mit den durch den Kontakt induzierten Zwangsbedingungen kompatible Größen sinnvoll regeln lassen. Im Beispiel des Tischabwischens bedeutet dies, dass die Richtung normal zum Tisch schlecht positionsgeregelt werden kann: bei einem starren Kontakt würde die Position aus der Kontaktbedingung folgen, sodass sie nicht frei geregelt werden kann. Die Kraft in Normalenrichtung folgt jedoch nicht aus der Zwangsbedingung und kann eingeregelt werden. Für die tangentialen Richtungen gilt eine analoge Argumentation: eine Kraftregelung ist schlecht möglich, da mangels Kontaktpartner die Kraft immer gleich Null sein muss (sofern Reibung vernachlässigt wird). Eine Positionsregelung ist hingegen problemlos möglich.

Im Folgenden wird die Thematik im Sinne einer Einführung vereinfacht dargestellt. Es wird angenommen, dass Kraft- und Positionsregelung jeweils im Sinne einer Kaskadenregelung auf einer Geschwindigkeitsregelung im Gelenkraum aufbauen. Eine Regelungsarchitektur für die hybride Kraft-/Positionsregelung ist in Bild 6.13 dargestellt.

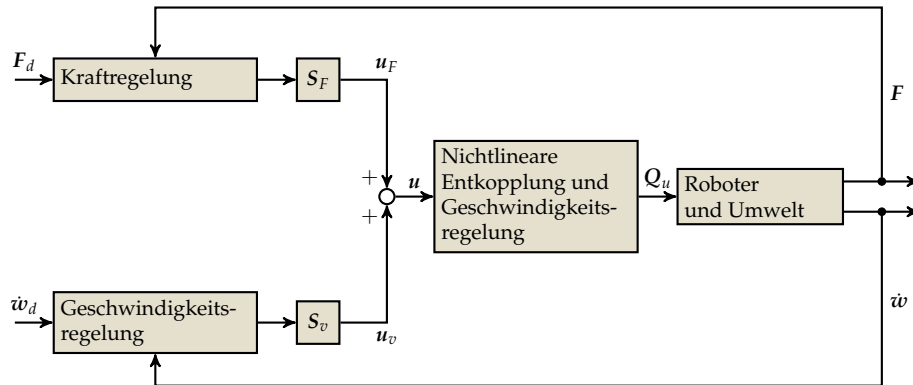


Bild 6.13: Struktur einer hybriden Kraft-/Positionsregelung.

Für Positions- und Kraftregelung wird jeweils aus dem Regelfehler eine Stellgröße u_v, u_F berechnet:

$$u_v = S_v \left[\dot{w}_d + K_{pv}(\dot{w}_d - \dot{w}) + K_{Iv} \int (\dot{w}_d - \dot{w}) dt \right] \quad (6.4.2)$$

$$u_F = S_F \left[\ddot{F}_d + K_{dF}(\dot{F}_d - \dot{F}) + K_{pF}(F_d - F) \right] \quad (6.4.3)$$

Die zu regelnden Positionen und Kräfte werden über die Selektionsmatrizen S_v und S_F vorgegeben, die zur Auswahl derjenigen Komponenten des Ergebnisses von (6.4.2), (6.4.3) dienen, die

²Wir gehen hierbei von kleinen Deformationen aus, sodass auf Komplikationen durch endliche räumliche Drehungen nicht eingegangen werden muss. Weiterführendes siehe Literatur.

in die Stellgröße u einfließen. Der Ausgang der Kraft-/Positionsregelung u dient im Sinne einer Kaskadenregelung als Eingang der Geschwindigkeitsregelung im Arbeitsraum.

Die Wahl der Selektionsmatrizen soll am Beispiel des Tischabwischens verdeutlicht werden (siehe Bild 6.12). Die Tischfläche ist durch die Gleichung:

$${}_C z = 0 \quad (6.4.4)$$

gegeben, wobei der Index C das tischfeste KOS bezeichnet. Damit lauten die Zwangsbedingungen für den Kontakt eines flächigen Endeffektors mit dem Tisch:

$${}_C \dot{z} = 0 \quad \text{Kontakt mit dem Tisch} \quad (6.4.5)$$

$${}_C \omega_x = 0 \quad \text{Parallelität zum Tisch} \quad (6.4.6)$$

$${}_C \omega_y = 0 \quad \text{Parallelität zum Tisch} \quad (6.4.7)$$

$${}_C F_x = 0 \quad \text{Kraft am freien Ende} \quad (6.4.8)$$

$${}_C F_y = 0 \quad \text{Kraft am freien Ende} \quad (6.4.9)$$

$${}_C M_z = 0 \quad \text{Moment um Tisch-Normale} \quad (6.4.10)$$

Mit der Definition der Arbeitsraum-Geschwindigkeiten als

$$\dot{w} = \begin{pmatrix} {}_C \dot{x} \\ {}_C \dot{y} \\ {}_C \dot{z} \\ {}_C \omega_x \\ {}_C \omega_y \\ {}_C \omega_z \end{pmatrix} \quad (6.4.11)$$

lautet die Auswahlmatrix zur Darstellung der zu regelnden Geschwindigkeiten:

$$S_v = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.4.12)$$

Bezeichnet man die Kraft am Endeffektor mit F und das Moment mit M , lautet die den Arbeitsraumkoordinaten w entsprechende Definition der verallgemeinerten Kraft λ :

$$\lambda = \begin{pmatrix} {}_C F_x \\ {}_C F_y \\ {}_C F_z \\ {}_C M_x \\ {}_C M_y \\ {}_C M_z \end{pmatrix} \quad (6.4.13)$$

Damit ergibt sich die Auswahlmatrix S_F zu:

$$S_F = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.4.14)$$

Da die Kraft- und Arbeitsraumrichtungen sich entsprechen und die Zwangsbedingungen für Kräfte und Positionen laut (6.4.5) bis (6.4.10) komplementär sind, gilt auch

$$S_F + S_v = E \quad (6.4.15)$$

Akronyme

BGL Bewegungsgleichung.

DGL Gewöhnliche Differentialgleichung.

DH-Parameter DENAVIT-HARTENBERG Parameter.

DH-Transformation DENAVIT-HARTENBERG Transformation.

EL-DGL EULER-LAGRANGE Differentialgleichung.

FhG Freiheitsgrad.

KOS Koordinatensystem.

MKS Mehrkörpersystem.

TCP Tool Center Point (Endeffektor).

Symbolverzeichnis

${}_iA_k$ Transformationsmatrix zur Umwandlung der Darstellung eines Vektors im B_k System in eine Darstellung im B_i System.

A_x Elementardrehung um die x -Achse.

$a(x)$ Rechte Seite der BGL als System 1. Ordnung.

A_y Elementardrehung um die y -Achse.

A_z Elementardrehung um die z -Achse.

$c x_{12}$ Abkürzung für $\cos(x_1 + x_2)$.

$c x$ Abkürzung für $\cos(x)$.

${}_iD_j$ Homogene Transformation von j nach i .

$\delta \dot{r}$ Virtuelle Geschwindigkeit.

δr Virtuelle Verschiebung.

E Einheitsmatrix.

e_i i -ter Einheitsvektor.

H HAMILTON-Funktion.

h Kraftvektor.

h_w Kraftvektor in Arbeitsraum-Koordinaten.

J_{Ri} Jacobi-Matrix der Rotation des i -ten Körpers ($\frac{\partial \omega_i}{\partial \dot{q}}$).

J_{TO_i} Jacobi-Matrix der Translation für den Ursprung des i -ten Körpers ($\frac{\partial \dot{r}_i}{\partial \dot{q}}$).

J_{TS_i} Jacobi-Matrix der Translation für den Schwerpunkt des i -ten Körpers ($\frac{\partial \dot{r}_{S_i}}{\partial \dot{q}}$).

J_w Jacobi-Matrix der Arbeitsraum-Koordinaten.

L LAGRANGE-Funktion.

λ LAGRANGE-Multiplikator.

$L_{O_i,i}$ Drall des i -ten Körpers bzgl. O_i .

M Massenmatrix.

M_w Massenmatrix in Arbeitsraum-Koordinaten.

O_i Ursprung des körperfesten Koordinatensystems des i -ten Körpers (B_i).

ω_i Winkelgeschwindigkeit des i -ten Körpers.

Φ Gütefunktional.

Φ Zwangsbedingungen.

p_i Impuls des i -ten Körpers.

Q Generalisierte Kraft.

q Minimalkoordinaten ($\in \mathbb{R}^n$).

Q_u Generalisierte Kraft der Antriebe.

\mathbb{R}^n Euklidischer Raum der Dimension n .

${}_j\mathbf{r}_i$ Vektor vom inertialen Koordinatensystem (KOS) (B_0) zum körperfesten KOS des i -ten Körpers, dargestellt im j -System.

${}_j\mathbf{r}_{kl}$ Vektor von k nach l , dargestellt im j -System.

S_i Schwerpunkt des i -ten Körpers.

$s x_{12}$ Abkürzung für $\sin(x_1 + x_2)$.

$s x$ Abkürzung für $\sin(x)$.

T Kinetische Energie.

V Potentielle Energie oder Lyapunov-Funktion.

W Arbeit.

δW Virtuelle Arbeit.

w Arbeitsraum-Koordinaten.

x Einheitsvektor in x -Richtung ($= \mathbf{e}_1$).

x Zustand des Systems ($x^T = [\mathbf{q} \dot{\mathbf{q}}]^T$).

y Einheitsvektor in y -Richtung ($= \mathbf{e}_2$).

z Einheitsvektor in z -Richtung ($= \mathbf{e}_3$).

Literatur

- Brandl, H., R. Johanni und M. Otter (1986). „A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix.“ In: *IFAC/IFIP/IMACS Symposium on Theory of Robots*, S. 95–100.
- Bremer, H. (1988). *Dynamik und Regelung mechanischer Systeme*. Wiesbaden: B.G. Teubner Verlag.
- (2008). *Elastic Multibody Dynamics*. Bd. 35. Intelligent Systems, Control, And Automation: Science And Engineering. Springer Netherlands, S. 29–58. ISBN: 978-1-4020-8679-3. DOI: [10.1007/978-1-4020-8680-9](#).
- Bremer, H. und F. Pfeiffer (1992). *Elastische Mehrkörpersysteme*. Wiesbaden: B.G. Teubner Verlag.
- Craig, J. J. (1989). *Introduction to Robotics, Mechanics & Control*. Addison-Wesley Publishing Company, Inc.
- Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Kluwer international series in engineering and computer science: Robotics. Springer. ISBN: 9780387743158.
- Geering, H. (2007). *Optimal control with engineering applications*. Springer. ISBN: 9783540694373.
- Hindmarsh, A. (1983). „Odepack, a systematized collection of ode solvers“. In: *IMACS Transactions on Scientific Computing*, S. 55–64.
- Khalil, H. K. (2000). *Nonlinear Systems*. 3. Aufl. Pearson Education.
- Khalil, H. (1992). *Nonlinear systems*. Macmillan Pub. Co. ISBN: 9780023635410.
- Liégeois, A. (1977). „Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms“. In: *Systems, Man and Cybernetics, IEEE Transactions on* 7.12, S. 868–871. ISSN: 0018-9472. DOI: [10.1109/TSMC.1977.4309644](#).
- Ludyk, G. (1995a). *Theoretische Regelungstechnik 1*. Springer-Verlag.
- (1995b). *Theoretische Regelungstechnik 2*. Springer-Verlag.
- Lutz, H. und W. Wendt (2005). *Taschenbuch der Regelungstechnik*. H. Deutsch.
- Nakamura, Y. (1991). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley.
- Pfeiffer, F. (1989). *Einführung in die Dynamik*. B.G. Teubner Verlag.
- Pfeiffer, F. und R. Johanni (1986). „A Concept for Manipulator Trajectory Planning“. In: *Proc IEEE Intl Conf Robotics and Automation (ICRA)*. DOI: [10.1109/JRA.1987.1087090](#).
- Pfeiffer, F. und E. Reithmeier (1987). *Roboterdynamik*. Teubner Studienbücher Mechanik. Wiesbaden: B.G. Teubner Verlag.
- Quarteroni, Sacco und Saleri (2002). *Numerische Mathematik 2*. Springer.
- Rivin, E. (1988). *Mechanical design of robots*. McGraw-Hill. ISBN: 9780070529922.
- Schröder, D. (2001). *Elektrische Antriebe – Regelung von Antriebssystemen*. 3. Aufl. Springer-Verlag Berlin Heidelberg.
- Shabana, A. (2005). *Dynamics of Multibody Systems*. Cambridge University Press. ISBN: 9780521850117.
- Siciliano, B. und O. Khatib, Hrsg. (2008). *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg.
- Siciliano, B., L. Sciavicco, L. Villani und G. Oriolo (2008). *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing Series. Springer. ISBN: 9781849966344.
- Slotine, J. und W. Li (1991). *Applied nonlinear control*. Prentice Hall. ISBN: 9780130408907.
- Walker, M. W. und D. E. Orin (1982). „Efficient Dynamic Computer Simulation of Robotic Mechanisms“. In: *Journal of Dynamic Systems, Measurement, and Control* 104.3, S. 205–211. DOI: [10.1115/1.3139699](#).

- Whitney, D. E. (1969). „Resolved Motion Rate Control of Manipulators and Human Prostheses“.
In: *IEEE Transactions on Man Machine Systems* 10.2, S. 47–53. ISSN: 0536-1540. DOI: [10.1109/TMMS.1969.299896](https://doi.org/10.1109/TMMS.1969.299896).
- Yamane, K. (2004). *Simulating and Generating Motions of Human Figures*. Springer-Verlag Berlin Heidelberg.

Korrekturen zum gedruckten Skript

In folgender Tabelle sind Fehler im gedruckten Skript aufgeführt, die in der vorliegenden Datei bereits behoben wurden.

Ort	Beschreibung
Symbolverzeichnis, Gleichung für Jacobi der Rotation	Ableitung $\frac{\partial \omega_i}{\partial \dot{q}}$ korrigiert